



Instituto Politécnico Nacional

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y TECNOLOGIAS AVANZADAS

Trabajo Terminal II

**“Prototipo de dispensador de comprimidos y cápsulas
medicinales configurable y administrable desde una aplicación
móvil”**

Que para obtener el título de

“Ingeniero en Mecatrónica”

Presentan:

García Miranda Carlos Ignacio

Mondragón Zenteno José Rodolfo

Rojas Granada Edgar Leonel

Asesores:

Dr. Salazar Silva Gastón Hugo

M. en C. Fonseca Campos Jorge

Dr. Mollinedo Ponce de León Helvio Ricardo



Mayo 2022



Instituto Politécnico Nacional

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN
INGENIERIA Y TECNOLOGIAS AVANZADAS

Trabajo Terminal II

“Prototipo de dispensador de comprimidos y cápsulas
medicinales configurable y administrable desde una aplicación
móvil”

Que para obtener el título de

“Ingeniero en Mecatrónica”

Presentan:

García Miranda Carlos Ignacio

Monragón Zenteno José Rodolfo

Rojas Granada Edgar Leonel

M. en C. Fonseca Campos Jorge

Asesores:

Dr. Salazar Silva Gastón Hugo

Dr. Mollinedo Ponce de León Helvio Ricardo

Presidente del Jurado

M. en C. Gutiérrez Begovich David Arturo

Profesor titular

M. en E. Rivas Bonilla Elizabeth



Autorización de uso de obra

Instituto Politécnico Nacional

Presente

Bajo protesta de decir verdad el que suscribe García Miranda Carlos Ignacio (se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los derechos morales y patrimoniales de la obra titulada Prototipo de dispensador de comprimidos y cápsulas medicinales configurable y administrable desde una aplicación móvil.

_____, en adelante "La Tesis" y de la cual se adjunta copia, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal del Derecho de Autor, otorgo a el Instituto Politécnico Nacional, en adelante El IPN, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales, Plataforma de la Dirección de Bibliotecas del IPN y/o consulta directa en la Coordinación de Biblioteca de la UPIITA "La Tesis" por un periodo de 5 años contado a partir de la fecha de la presente autorización, dicho periodo se renovará automáticamente en caso de no dar aviso expreso a "El IPN" de su terminación.

En virtud de lo anterior, "El IPN" deberá reconocer en todo momento mi calidad de autor de "La Tesis". Adicionalmente, y en mi calidad de autor y titular de los derechos morales y patrimoniales de "La Tesis", manifiesto que la misma es original y que la presente autorización no contraviene ninguna otorgada por el suscrito respecto de "La Tesis", por lo que deslindo de toda responsabilidad a El IPN en caso de que el contenido de "La Tesis" o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumo las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México, a 19 de Mayo de 2022

Atentamente



Autorización de uso de obra

Instituto Politécnico Nacional

Presente

Bajo protesta de decir verdad el que suscribe Montejari Zenaro José Redolfo
(se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los derechos morales y patrimoniales de la obra titulada Prototipo de dispensador de comprimidos y cápsulas medicinales configurable y administrable desde una aplicación móvil, en adelante "La Tesis" y de la cual se adjunta copia, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal del Derecho de Autor, otorgo a el Instituto Politécnico Nacional, en adelante El IPN, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales, Plataforma de la Dirección de Bibliotecas del IPN y/o consulta directa en la Coordinación de Biblioteca de la UPIITA "La Tesis" por un periodo de 5 años contado a partir de la fecha de la presente autorización, dicho periodo se renovará automáticamente en caso de no dar aviso expreso a "El IPN" de su terminación.

En virtud de lo anterior, "El IPN" deberá reconocer en todo momento mi calidad de autor de "La Tesis". Adicionalmente, y en mi calidad de autor y titular de los derechos morales y patrimoniales de "La Tesis", manifiesto que la misma es original y que la presente autorización no contraviene ninguna otorgada por el suscrito respecto de "La Tesis", por lo que deslindo de toda responsabilidad a El IPN en caso de que el contenido de "La Tesis" o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumo las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México, a 19 de Mayo de 2022

Atentamente



Autorización de uso de obra

Instituto Politécnico Nacional

Presente

Bajo protesta de decir verdad el que suscribe Rojas Granada Edgar Leonel (se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los derechos morales y patrimoniales de la obra titulada Prototipo de dispensador de comprimidos y capsulas medicinales configurable y administrable desde una aplicacion movil, en adelante "La Tesis" y de la cual se adjunta copia, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal del Derecho de Autor, otorgo a el Instituto Politécnico Nacional, en adelante El IPN, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales, Plataforma de la Dirección de Bibliotecas del IPN y/o consulta directa en la Coordinación de Biblioteca de la UPIITA "La Tesis" por un periodo de 5 años contado a partir de la fecha de la presente autorización, dicho periodo se renovará automáticamente en caso de no dar aviso expreso a "El IPN" de su terminación.

En virtud de lo anterior, "El IPN" deberá reconocer en todo momento mi calidad de autor de "La Tesis". Adicionalmente, y en mi calidad de autor y titular de los derechos morales y patrimoniales de "La Tesis", manifiesto que la misma es original y que la presente autorización no contraviene ninguna otorgada por el suscrito respecto de "La Tesis", por lo que deslindo de toda responsabilidad a El IPN en caso de que el contenido de "La Tesis" o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumo las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México, a 19 de Mayo de 2022

Atentamente





Contenido

Resumen.....	xiii
Palabras clave.....	xiii
Abstract.....	xiv
Nomenclatura y simbología.....	xv
Nomenclatura.....	xv
Simbología.....	xvi
Introducción.....	xxi
Planteamiento del Problema.....	xxii
Propuesta de solución.....	xxiii
Justificación.....	xxiii
Objetivos.....	xxv
Objetivo General.....	xxv
Objetivos Particulares.....	xxv
Antecedentes.....	27
Organización del documento.....	30
I. Marco Teórico.....	32
Concepto de formas farmacéuticas.....	32
Condiciones de conservación y almacenamiento de productos farmacéuticos.....	32
Materiales utilizados en la construcción de contenedores primarios de medicamentos.....	33
Redes inalámbricas.....	35
Comunicación Wi-Fi.....	36
Teoría de clasificaciones.....	36
Tamaños y formas de comprimidos para administración oral.....	37
Tamaño, forma y otros atributos físicos de las cápsulas y tabletas genéricas.....	37
Transportadores Helicoidales Verticales.....	39
Medio paso, helicoidal sencillo.....	46
Flujo de fluido debido a una carga descendente.....	46
Coeficiente de resistencia K para pérdidas de flujo de volumen de fluido en un componente hidráulico.....	47
Control automático.....	48



Bases de Datos en la Nube	50
Material desecante.....	51
2. Metodología	53
3. Diseño del Sistema.....	56
Requerimientos y especificaciones del sistema	56
Análisis de Requerimientos	56
PDS	57
Diseño Conceptual.....	60
Matriz morfológica	60
Selección de rutas	64
Parámetros y atributos para la evaluación de diseños conceptuales	64
Evaluación de las rutas seleccionadas.....	64
Boceto general del dispositivo en CAD.....	65
IDEF A-0.....	66
IDEF A0	68
Diseño detallado.....	69
Diseño del Contenedor de Medicamentos.....	69
Validaciones del Contenedor de Medicamentos	81
Diseño del Contenedor de Agua	87
Sistema de Dispensado de Agua.....	91
Sistema de Dispensado de Medicamentos	102
Acoplamiento de los sistemas de dispensado	126
Sistema de entrada de medicamentos.....	128
Sistema de salida de medicamentos.....	138
Sistema de control de temperatura interna del dispensador	148
Selección de unidad central de procesamiento y tarjeta de control de actuadores	161
Sistema de alarma para ingesta de comprimido o cápsula.....	164
Operación del Sistema	168
Interfaces de Usuario.....	175
Sistema de energía	182
4. Manufactura.....	188



Material de construcción.....	188
Manufactura de las piezas del dispositivo	191
Pruebas de impresión 3D	192
Proceso de impresión 3D de las piezas del dispositivo	194
Piezas Impresas.....	195
5. Implementación.....	207
Etapas de potencia	207
Aplicación móvil.....	210
Software de la tarjeta de control.....	220
Integración de software.....	225
Mecanismo de ingreso de medicamentos.....	227
Mecanismo de extracción de medicamentos	229
Mecanismo de dispensado de agua.....	230
Mecanismo de ventilación y disipación de humedad.....	230
6. Resultados	233
Prueba de PWM para los ventiladores.....	233
Prueba de la secuencia para el motor a pasos bipolar.....	234
Prueba de la secuencia para el motor a pasos unipolar	236
Prueba del sensor de temperatura y humedad.....	237
Pruebas de la aplicación móvil	238
Modelo final en CAD del prototipo.....	245
Ensamble General del dispositivo.....	247
7. Análisis de resultados	248
8. Costos.....	252
9. Cronogramas.....	254
10. Conclusiones	256
Trabajo a futuro.....	257
11. Referencias	258
12. Apéndices.....	263
13. Anexos.....	356



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria en Ingeniería y
Tecnologías Avanzadas





Índice de imágenes

Imagen 1. Ecuación para obtener la capacidad equivalente [20].	40
Imagen 2. Ecuación para calcular los rpm requeridos para una carga. [20].	41
Imagen 3. Ecuación de la potencia para superar la fricción. [20].	43
Imagen 4. Ecuación de la potencia para transportar la capacidad de material especificada. [20].	43
Imagen 5. Ecuación de la potencia requerida para transportar un sólido dado. [20].	43
Imagen 6. Ecuación de la deflexión para secciones más largas que el estándar. [20].	45
Imagen 7. Ecuación de par al que estarán sujetos los componentes. [20].	45
Imagen 8. Identificación de los parámetros a tomar en cuenta en el análisis de un fluido descendente. [22].	46
Imagen 9. Lazo Cerrado de Control. [24].	48
Imagen 10. Ejemplo Lazo Cerrado de Control de Temperatura [25].	48
Imagen 11. Acciones de Control de los controladores ON-OFF, P y PID [27].	49
Imagen 12. Gráficas de capacidad de equilibrio (H ₂ O) y tasa de absorción (H ₂ O). [30].	51
Imagen 13. Modelo V para el desarrollo de sistemas mecatrónicos. [33].	54
Imagen 14. Modelo V para el desarrollo del dispensador.	55
Imagen 15. Boceto general del dispositivo (vista isométrica).	65
Imagen 16. Boceto general del dispositivo (vista superior).	66
Imagen 17. Cálculo del número de hélices mínimo en el tornillo.	71
Imagen 18. Diagramas de cuerpo libre de una sección de la hélice del tornillo sinfín.	72
Imagen 19. Tope de las pastillas.	75
Imagen 20. Cubierta del contenedor del tornillo 1, isométrico.	77
Imagen 21. Vista frontal de la cubierta del tornillo 1.	77
Imagen 22. Vista isométrica de la base de la cubierta del tornillo 1.	77
Imagen 23. Cubierta del contenedor del tornillo 2, isométrico.	78
Imagen 24. Vista frontal de la cubierta del tornillo 2.	78
Imagen 25. Vista isométrica de la base de la cubierta del tornillo 2.	78
Imagen 26. Cubierta del contenedor del tornillo 3, isométrico.	79
Imagen 27. Vista frontal de la cubierta del tornillo 3.	79
Imagen 28. Vista isométrica de la base de la cubierta del tornillo 3.	79
Imagen 29. Vista isométrica del tornillo 1.	80
Imagen 30. Vista isométrica del tornillo 2.	80
Imagen 31. Vista isométrica del tornillo 3.	80
Imagen 32. Validación de esfuerzo calculado en el tornillo 1.	81
Imagen 33. Validación de esfuerzo calculado en el tornillo 2.	81
Imagen 34. Validación de esfuerzo calculado en el tornillo 3.	82
Imagen 35. Comprimido a un ángulo de 18° donde no se desliza.	83
Imagen 36. Comprimido a un ángulo de 30° donde se desliza.	83
Imagen 37. Comprimido a un ángulo de 28.5° donde se empieza a deslizar.	84
Imagen 38. Tableta a un ángulo de 25° donde no se desliza.	84
Imagen 39. Tableta a un ángulo de 30° donde se desliza.	85
Imagen 40. Tableta a un ángulo de 30° donde ya se deslizó.	85



Imagen 41. Cápsula a un ángulo de 25° donde no se desliza.....	85
Imagen 42. Cápsula a un ángulo de 30° donde se desliza.....	86
Imagen 43. Cápsula a un ángulo de 30° donde ya se deslizó.....	86
Imagen 44. Vista superior del boceto general del dispositivo.....	87
Imagen 45. Vista superior de la parte trasera del dispositivo concepto.....	88
Imagen 46. Geometría Base del contenedor.....	88
Imagen 47. Medición del área del contenedor.....	89
Imagen 48. Vista superior del contenedor.....	90
Imagen 49. Vista isométrica del contenedor.....	90
Imagen 50. Esquema del sistema de dispensado de agua.....	91
Imagen 51. Electroválvula seleccionada con su respectivo circuito neumático. [34].....	92
Imagen 52. Sensor de nivel de agua sin contacto, XKC-Y25. [35].....	97
Imagen 53. Sensor de nivel de agua sin contacto, XKC-Y26-NPN. [36].....	97
Imagen 54. Sensor de nivel de agua sin contacto, tipo switch. [37].....	97
Imagen 55. Presión hidrostática aplicada al contenedor de agua.....	99
Imagen 56. Resultado de la deformación en el tanque.....	100
Imagen 57. Resultado del esfuerzo en el tanque.....	100
Imagen 58. Vista isométrica del contenedor con la válvula.....	101
Imagen 59. Sensor XKC-Y25-V con sus respectivas especificaciones.....	101
Imagen 60. Nueva electroválvula 2/2 N. C. de 1/4"......	102
Imagen 61. Vista isométrica superior del revolver.....	103
Imagen 62. Vista isométrica inferior del revolver.....	103
Imagen 63. Masa indicada del revolver con SolidWorks.....	104
Imagen 64. Masa indicada del contenedor 1 con SolidWorks.....	104
Imagen 65. Masa indicada del contenedor 2 con SolidWorks.....	105
Imagen 66. Masa indicada del contenedor 3 con SolidWorks.....	105
Imagen 67. Sistema ensamblado con 3 tipos diferentes de contenedor.....	106
Imagen 68. Vista inferior del ensamblaje del revolver y los contenedores.....	106
Imagen 69. Propiedades físicas del ensamble del revolver y los contenedores.....	107
Imagen 70. Resultado del análisis de deformación para el revolver.....	107
Imagen 71. Resultado del análisis de esfuerzo de Von mises para el revolver.....	107
Imagen 72. Resultado del análisis de esfuerzos máximos para el revolver.....	108
Imagen 73. Vista inferior del revolver para el sistema de salida de las pastillas.....	108
Imagen 74. motor 28BYJ-48. 5 VCD, 40 mA/fase, tamaño: 35 mm, cuenta con driver, torque: 0.02942 Nm [38].....	109
Imagen 75. motor PM55L-048. 24 VCD, 600 mA/fase, tamaño: 65 mm, cuenta con driver, torque: 0.11376 Nm [39].....	109
Imagen 76. motor NEMA17. 12 VCD, 1.7 A, tamaño: 42 mm, cuenta con driver, torque: 0.3923 Nm [40]. ...	109
Imagen 77. Compartimento 1 de almacenamiento de pastillas.....	114
Imagen 78. Compartimento 2 de almacenamiento de pastillas.....	114
Imagen 79. Compartimento 3 de almacenamiento de pastillas.....	114
Imagen 80. CAD de los compartimentos 1, 2 y 3, respectivamente, estos son los modelos actuales.....	115



Imagen 81. Modelos CAD actuales de los contenedores 3,1 y 2, respectivamente	117
Imagen 82. Tapa 1 vista de la cara inferior.....	117
Imagen 83. Tapa 2 vista de la cara inferior.	118
Imagen 84. Tapa 3 vista de la cara inferior.	118
Imagen 85. Tapa 1 vista de la cara superior.	118
Imagen 86. Tapa 2 vista de la cara superior.....	119
Imagen 87. Tapa 3 vista de la cara superior.	119
Imagen 88. CAD de las tapas de los contenedores 2,1 y 3, respectivamente, que son los modelos actuales. .	120
Imagen 89. Tornillo 2 actual en comparación con el tornillo 2 anterior.	121
Imagen 90. Tornillo 3 actual en comparación con el tornillo 3 anterior.	121
Imagen 91. Componentes por ensamblar para el tornillo 1.....	122
Imagen 92. Ensamble del tornillo 1 junto con su reductor.	122
Imagen 93. CAD de los tornillos sinfin 3,1 y 2, respectivamente, que son los modelos actuales.	123
Imagen 94. Vista superior e inferior respectivamente del revolver.	124
Imagen 95. Rieles de sujeción para cada compartimento.....	124
Imagen 96. Ejemplo de ensamble de un compartimento al revolver.	124
Imagen 97. CAD del modelo del revolver anterior.....	126
Imagen 98. CAD del modelo del revolver actual.	126
Imagen 99. Vista inferior de la tapa del contenedor de agua.....	127
Imagen 100. Vista isométrica de la tapa del contenedor de agua.	127
Imagen 101. Vista isométrica de la tapa del tanque de agua acoplada a la carcasa del dispensador.	127
Imagen 102. Vista isométrica de la tapa del contenedor de agua cuando el tanque se encuentra descubierto, se puede apreciar las ranuras por donde se acopla la tapa.	128
Imagen 103. Vista lateral del sistema de dispensado de agua, se puede apreciar el acople interno del tanque con la electroválvula.....	128
Imagen 104. Entrada de la pastilla, ranura y rampa.....	129
Imagen 105. Cámara Raspberry Pi3 Ov5647. [41]	129
Imagen 106. Soporte de cámara Raspberry Pi 3 y de embudo.	130
Imagen 107. Momento en el que la compuerta impide el paso de las pastillas.	131
Imagen 108. Momento en el que la compuerta permite el paso de las pastillas.	131
Imagen 109. Obtención de la masa de la compuerta de paso mediante SolidWorks.	131
Imagen 110. Solenoide push pull Rantec de 12 V. [42]	132
Imagen 111. Solenoide push pull Tecneu de 5 V. [43]	132
Imagen 112. Solenoide push pull Filect de 12 V. [44]	132
Imagen 113. Embudo de posicionamiento de pastillas.....	134
Imagen 114. Vista de sección del sistema de entrada de pastillas.....	134
Imagen 115. Vista de la carcasa del sistema de entrada de medicamentos.....	135
Imagen 116. Vista isométrica del embudo anterior.....	136
Imagen 117. Vista isométrica del embudo actual de entrada.	136
Imagen 118. Sujetador del embudo de entrada de medicamentos.....	137
Imagen 119. Carcasa de entrada de medicamentos.	138
Imagen 120. Mecanismo de entrada/dispensado de medicamentos.	139



Imagen 121. Vista de una compuerta en la parte inferior, el embolo que tiene es para el acople al actuador.	140
Imagen 122. Vista de sección transversal del revolver acoplado a los contendores y a la compuerta.	141
Imagen 123. Acople del tornillo 3, el diámetro de cada acople en la parte inferior es diferente para garantizar la transmisión de movimiento.....	141
Imagen 124. Embudo de salida del medicamento, el final de esta se acopla a la parte de la salida lateral de la carcasa.....	142
Imagen 125. Actuador alineado al embolo de la compuerta en posición de salida, también se aprecia la disposición de la rampa.....	142
Imagen 126. Sistema de transmisión de movimiento, se aprecia la posición tangencial del transmisor con el acople del tornillo.	143
Imagen 127. Cubierta del segundo actuador adaptada a la carcasa de salida.	143
Imagen 128. Carcasa del mecanismo de salida.....	144
Imagen 129. Acople del sistema de agua con la carcasa y con el mecanismo de entrada.....	144
Imagen 130. Acople de los sistemas de entrada y salida, se aprecia la disposición del embudo con el contenedor lateral izquierdo del sistema.	145
Imagen 131. Acople final del dispensador de comprimidos y capsulas, se aprecia el cambio notable con respecto al boceto original.	145
Imagen 132. Nuevo mecanismo de empuje de las compuertas, se aprecia la composición del mecanismo denominado “Ascensor”.	146
Imagen 133. CAD de las compuertas 1, 2 y 3 respectivamente, modelos actualizados.	147
Imagen 134. Lazo cerrado de control de temperatura interna del dispensador.	149
Imagen 135. Ventilador Cooler Master: 12 VCD, 0.15 A, 44.73 CFM, \$299.00MXN. [45].....	150
Imagen 136. Ventilador WINSINN USB: 5 VCD, 0.25 A, 72 CFM, \$323.54MXN. [46]	150
Imagen 137. Ventilador Arctic I2 VCD, 0.20 A, 56.3 CFM, \$474.64MXN. [47].....	151
Imagen 138. Sensor de temperatura y humedad con driver DHT11 [48].	152
Imagen 139. Sensor de temperatura y humedad con driver DHT22 [49].	152
Imagen 140. Sensor de temperatura y humedad ATM 1001 [50].	152
Imagen 141. Conexiones eléctricas del sensor de temperatura y humedad.	154
Imagen 142. Circuito eléctrico de la etapa de potencia para el control de temperatura.	154
Imagen 143. Símbolo de MOSFET canal n D=Drain, G= Gate, S=Source.....	155
Imagen 144. Regiones de trabajo del MOSFET con V_DS=cte y V_GS=variable.....	155
Imagen 145. Configuración de Barrido de segunda configuración.	156
Imagen 146. Segunda configuración de fuentes de tensión del circuito.	156
Imagen 147. Curva característica de la configuración del MOFSET.	157
Imagen 148. Comparación de modelo teórico y modelo de Multisim.	157
Imagen 149. Símbolo que se usará para indicar la etapa de potencia para cada actuador.....	158
Imagen 150. Sensor de temperatura y humedad con driver DHT11 [48].	159
Imagen 151. Ventilador BDM8025S.....	160
Imagen 152. Tarjeta de procesamiento de datos Raspberry Pi 3 b+. [51].....	161
Imagen 153. Tarjeta de procesamiento de datos Núcleo-F446RE. [52]	162
Imagen 154. Tarjeta de procesamiento de datos Arduino Mega 2560. [53]	162
Imagen 155. Composición del buzzer.[54].....	165



Imagen 156. Buzzer comercial que se pretende usar. [55].....	165
Imagen 157. Especificaciones técnicas del buzzer comercial. [55].....	166
Imagen 158. Bocina altavoz comercial tentativa para usar. [56]	166
Imagen 159. Especificaciones técnicas de la bocina altavoz comercial. [56].....	166
Imagen 160. Diagrama de flujo para el sistema de alarma.	167
Imagen 161. Diagrama eléctrico de la conexión del buzzer al microcontrolador.....	167
Imagen 162. Interfaz de la aplicación móvil.....	175
Imagen 163. Interfaces de ingresar nuevo tratamiento y de gestionar tratamientos.	176
Imagen 164. Interfaces de observar funcionamiento del dispositivo y de modo viaje.	177
Imagen 165. Pantalla de 5" con conexión HDMI-DSI Waveshare [57].	178
Imagen 166. Pantalla de 3.5" con conexión HDMI-SPI Raspberry [58]......	179
Imagen 167. Pantalla de 7" con conexión HDMI-DSI Waveshare [59].	179
Imagen 168. Interfaz gráfica de usuario del dispensador al encender el dispensador.	180
Imagen 169. Interfaz de usuario en el dispensador en operación.	181
Imagen 170. Ventana de selección de forma del medicamento en la aplicación móvil.	182
Imagen 171. Fuente de poder CoolerMAster MWE 450 [60]......	185
Imagen 172. Fuente de poder Getttech FG500W BX [61]......	186
Imagen 173. Fuente de poder Ateck integra 5R500 [62].	186
Imagen 174. Cotización de la impresión de las piezas con el GRUPO XDS.....	189
Imagen 175. Cotización de la impresión de las piezas en resina similar al polipropileno, con el GRUPO XDS.	190
Imagen 176. Segmento de reductor en CAD e impreso.....	192
Imagen 177. Segmento del revolver en CAD e impreso.....	193
Imagen 178. Segmento de ensamble de tornillo sinfín en CAD e impreso.	193
Imagen 179. Tapa de contenedor en CAD e impreso.	193
Imagen 180. Cubierta del contenedor en CAD e impreso.....	193
Imagen 181. Proceso de impresión en 3D de las piezas del dispositivo.....	194
Imagen 182. Software para el proceso de impresión 3D de las piezas del dispositivo.....	195
Imagen 183. Carcasa entrada rediseñada, vista frontal.....	196
Imagen 184. Carcasa de entrada, vista superior.	196
Imagen 185. Carcasa principal superior, vista superior, se puede observar el compartimento del tanque de agua y en la parte superior las ranuras para el soporte de entrada.	197
Imagen 186. Carcasa principal superior, vista frontal, se aprecia la ranura de la tapa superior que recubre el tanque de agua.	197
Imagen 187. Vista superior de la carcasa principal inferior.	198
Imagen 188. Vista frontal de la carcasa principal inferior.	198
Imagen 189. Tapa del compartimento 1.	199
Imagen 190. Tapa del compartimento 2.	199
Imagen 191. Tapa del compartimento 3.	199
Imagen 192. Tapa del compartimento 4.	199
Imagen 193. Compuertas de los compartimentos.....	200
Imagen 194. Vista superior del revólver, se logra apreciar los espacios de ensamble.	200



Imagen 195. Vista lateral de uno de los compartimentos, en la parte inferior se aprecian las ranuras de anclaje al revolver.	201
Imagen 196. Vista frontal del tanque de agua.	201
Imagen 197. Vista frontal del embudo de entrada.	202
Imagen 198. Soporte del embudo de entrada.	202
Imagen 199. Paredes del mecanismo del ascensor que pertenecen al sistema de salida.	203
Imagen 200. Pieza de unión de las paredes del mecanismo de ascensor.	203
Imagen 201. Embudo de salida.	204
Imagen 202. Soporte del embudo de salida.	204
Imagen 203. Soporte del solenoide de dispensado.	205
Imagen 204. Soporte del motor a pasos unipolar para la transmisión de movimiento a los tornillos.	205
Imagen 205. Rampa de entrada de las pastillas.	206
Imagen 206. Diagrama esquemático del circuito de potencia.	207
Imagen 207. PCB del circuito de potencia.	208
Imagen 208. Modelo en 3D del PCB del circuito de potencia, vista superior.	208
Imagen 209. Modelo en 3D del PCB del circuito de potencia y pistas para planchado, vista inferior.	208
Imagen 210. Pistas del circuito de la etapa de potencia.	209
Imagen 211. Circuito de etapa de potencia para los actuadores.	209
Imagen 212. Conexión de los pines GPIO de la Raspberry al convertidor de nivel, y este, a la etapa de potencia.	210
Imagen 213. Ventana principal.	211
Imagen 214. Ventana de selección de tipo de medicamento.	212
Imagen 215. Ventana de ingreso de nuevos tratamientos.	213
Imagen 216. Ventana de ingreso de medicamentos.	214
Imagen 217. Ventana de gestión de tratamientos.	215
Imagen 218. Ventana para borrar un tratamiento.	215
Imagen 219. Ventana para modificar un tratamiento.	216
Imagen 220. Ventana para reabastecer un tratamiento.	217
Imagen 221. Ventana de monitoreo del dispensador.	218
Imagen 222. Ventana de modo viaje.	218
Imagen 223. Ventana de notificación de ingesta de medicamento.	219
Imagen 224. Diagrama de flujo de funcionamiento de la aplicación móvil.	220
Imagen 225. Lazo cerrado de control de temperatura interna del dispensador.	221
Imagen 226. Diagrama de posicionamiento de los contenedores de pastillas y el desplazamiento que deben recorrer.	222
Imagen 227. Diagrama de flujo ilustrativo del funcionamiento del programa en la tarjeta Raspberry.	224
Imagen 228. Base de datos generada para la comunicación del dispensador con la aplicación móvil.	225
Imagen 229. Rama Tratamientos de la base de datos.	226
Imagen 230. Rama BanderasControl de la base de datos.	226
Imagen 231. Diagrama de conexión de la base de datos con la aplicación móvil y la Raspberry Pi 3.	227
Imagen 232. Ensamble de los contenedores con el revólver.	227
Imagen 233. Ensamble del revólver con los contenedores al motor a pasos bipolar.	228



Imagen 234. Ensamble de la carcasa principal de entrada, junto con la rampa de entrada de medicamentos.	228
Imagen 235. Ensamble del embudo de entrada de pastillas junto con su soporte.	229
Imagen 236. Ensamble del motor a pasos unipolar con los transmisores de movimiento de los tornillos sinfín.	230
Imagen 237. Ensamble de la caja para desecante con la carcasa inferior.	231
Imagen 238. Prueba del PWM en el caso de una temperatura baja.	233
Imagen 239. Prueba del PWM en el caso de una temperatura media.	234
Imagen 240. Prueba del PWM en el caso de una temperatura alta.	234
Imagen 241. Código de la secuencia para el motor a pasos bipolar.	235
Imagen 242. Secuencia del paso completo representado con los leds.	235
Imagen 243. Código de la secuencia para el motor a pasos unipolar.	236
Imagen 244. Secuencia de medio paso representado con los leds.	237
Imagen 245. Código para medir temperatura y humedad con el sensor DHT11.	238
Imagen 246. Prueba física del sensor de temperatura y humedad, DHT11.	238
Imagen 247. Ventana de ingreso de nuevo tratamiento con información recopilada y la bandera de IngreNTrat activada.	239
Imagen 248. Recopilación de los datos de un nuevo tratamiento de pastillas redondas en Firebase.	239
Imagen 249. Ventana de modificar tratamiento con nuevos datos recopilados.	240
Imagen 250. Cambio en los datos del tratamiento uno tras haber solicitado guardar los cambios realizados.	240
Imagen 251. Ventana de reabastecer tratamientos habiendo consultado reabastecer tratamiento 1.	241
Imagen 252. Ventana para borrar tratamientos seleccionando eliminar tratamiento 1.	242
Imagen 253. Cambio en los datos del tratamiento uno y la bandera borrar en la base de datos.	242
Imagen 254. Ventana de monitoreo consultando información más reciente de Firebase.	243
Imagen 255. Información consultada de Frirebase por la aplicación móvil.	244
Imagen 256. Ventana de modo viaje de la aplicación, consulta de pastillas necesarias para el viaje.	244
Imagen 257. Bandera "ModoViaje".	245
Imagen 258. Diseño final del prototipo de dispensador de comprimidos y cápsulas.	245
Imagen 259. Explosionado del diseño del prototipo de dispensador de comprimidos y cápsulas.	246
Imagen 260. Ensamble final del dispensador.	247
Imagen 261. Cronograma de actividades en Trabajo Terminal I.	254
Imagen 262. Cronograma de actividades en Trabajo Terminal II.	255



Índice de tablas

Tabla 1. Paso especial del tornillo transportador Factor CF1. [20].....	40
Tabla 2. Capacidad especial del filete del tornillo transportador Factor CF2. [20]	40
Tabla 3. Capacidad especial de la paleta mezcladora del tornillo transportador Factor CF3. [20].....	40
Tabla 4. Tamaños y capacidades de transportadores de tornillo sinfín. [20]	41
Tabla 5. Factor de rodamiento, Fb. [20]	42
Tabla 6. Factor de diámetro del filete, Fd. [20].....	43
Tabla 7. Factor de filete, Ff. [20]	43
Tabla 8. Factor de álabe, Fp. [20].....	43
Tabla 9. Factor de sobrecarga, Fo. [20].....	44
Tabla 10. Especificaciones de diseño del producto.	60
Tabla 11. Matriz morfológica de formas para el diseño conceptual del dispensador.	63
Tabla 12. Rutas solución de la matriz morfológica propuestas por el equipo.	64
Tabla 13. Evaluación de las rutas seleccionadas.	65
Tabla 14. Características de interés de cada sensor de nivel de agua sin contacto.	98
Tabla 15. Evaluación de los sensores de nivel de agua sin contacto, mediante criterios ponderados.	98
Tabla 16. Evaluación de los motores a pasos para el revolver, mediante criterios ponderados.	110
Tabla 17. Características de los solenoides push pull propuestos.....	132
Tabla 18. Evaluación de los solenoides push pull para la extracción de la pastilla, mediante criterios ponderados.....	133
Tabla 19. Evaluación de los motores a pasos para la extracción de la pastilla, mediante criterios ponderados.	139
Tabla 20. Evaluación de los ventiladores mediante criterios ponderados.	151
Tabla 21. Principales características técnicas de los sensores propuestos.	153
Tabla 22. Evaluación de los sensores de temperatura y humedad con criterios ponderados.....	153
Tabla 23. Características principales de las tarjetas de procesamiento propuestas.	162
Tabla 24. Evaluación de las tarjetas de procesamiento de datos para ser la unidad central de procesamiento, mediante criterios ponderados.	163
Tabla 25. Principales características de las tarjetas que se proponen para ser la tarjeta de control.	163
Tabla 26. Evaluación de las tarjetas de procesamientos de datos para ser la tarjeta de control, mediante criterios ponderados.	164
Tabla 27. Características de interés de las pantallas propuestas.	179
Tabla 28. Evaluación de las pantallas propuestas mediante criterios ponderados de acuerdo a las necesidades del proyecto.....	180
Tabla 29. Características de interés de las fuentes de alimentación.	186
Tabla 30. Evaluación de las fuentes de alimentación, mediante criterios ponderados.	187
Tabla 31. Cotización de las piezas a imprimir con el material de PLA, por un independiente.....	191
Tabla 32. Costos de los materiales usados para el prototipo.	252
Tabla 33. Costo de manufactura de las piezas del prototipo.	253
Tabla 34. Costo final del prototipo.	253



Resumen

En este documento, se presenta el diseño y construcción de un prototipo siguiendo una metodología recursiva de diseño mecatrónico en V, cuyo fin, será atender la problemática de la falta de adherencia terapéutica de los pacientes que se encuentran bajo tratamientos médicos permanentes debido a enfermedades crónicas, bajo tratamientos médicos intermitentes y también de quienes se encuentran bajo tratamientos médicos temporales y que, además, consumen medicamentos en forma de comprimidos y capsulas.

El proyecto consiste en el diseño y construcción de un prototipo de dispensador de comprimidos y cápsulas que asista a las personas durante sus tratamientos médicos llevando el control de qué medicamento debe ingerir y en qué momento. El diseño realizado logra que dicho dispositivo tenga la capacidad de almacenar hasta tres tipos de comprimidos y cápsulas de diferentes tamaños y geometrías las cuales deberán ser ingresadas manualmente al aparato una a la vez; los tratamientos del usuario y sus respectivos horarios de ingesta de medicamento son configurables y administrables desde una aplicación móvil y, cuando llega el momento de ingerir un medicamento, despliega una notificación de recordatorio en el móvil del usuario, hace sonar una alarma en el dispensador y le entrega el medicamento requerido, así como agua potable para poder ingerirlo.

Además, este dispositivo cuenta con alarmas para indicar cuando un medicamento se está agotando y cuando el tanque de agua se está quedando vacío. Otro punto a destacar es que se ha integrado una modalidad de viaje que permite al usuario llevarse consigo las pastillas que necesite durante un periodo de tiempo de uno hasta tres días en un pastillero portátil.

El dispensador cuenta con contenedores con una capacidad de almacenamiento de hasta quince pastillas cada uno y en total tres diferentes tipos de contenedores teniendo una capacidad general de hasta 60 pastillas, así como un contenedor de agua con capacidad de hasta un litro. Adicionalmente, cuenta con un sistema de control de temperatura que mantiene las pastillas almacenadas en un ambiente con una temperatura inferior a los 30°C. En la aplicación móvil del celular del usuario también se pueden observar los principales parámetros de operación del dispensador.

Palabras clave

Dispensador, temperatura, mecanismo, sistema, tratamientos, aplicación móvil, medicamentos, diseño.



Abstract

This document presents the design and development of a prototype following the V recursive mechatronics design methodology, which purpose is to address the problem of the lack of therapeutic adherence of patients who are undergoing permanent medical treatments due to chronic diseases, undergoing intermittent medical treatments and even those who are undergoing temporary medical treatments, and use medicines in pill form

The project consists of the design and construction of a prototype of a medical pill dispenser which looks forward to assist people who are undergoing medical treatments by remembering to the users which pill they have to take and when. The device has the possibility to stock three different kinds of pills of different shape and size, the pills are to enter into the device manually one by one, and then, the dispenser stock them into a proper container. The user has the chance to configure and administrate his treatments and their intake period from a mobile app. When the intake time of a pill takes place, the user is notified in his phone and the dispenser rings an alarm, and also provides the required pill and water to take it. The device has alarms to alert the user that he needs to refill the water tank or a pill container with low amount of medicine. This device will have a travel mode which allows the user to take with him the required number of pills for an absence period from one up to three days in a portable pill organizer.

The device has 3 different containers, each container is able to stock 15 pills, this means that in general it has the capacity to stock 60 pills, and also has a water container which has a 1L capacity. The dispenser has a temperature control to preserve the pills environment under 30°C. Finally, the mobile app also gives the user the chance to look the main operation variables of the pill dispenser.



Nomenclatura y simbología

Nomenclatura

FCO	Fármaco
NOM	Norma Oficial Mexicana
ISO	International Organization for Standardization
PVC	Cloruro de Polivinilo
LDPE	Low Density Polyethylene
HDPE	High Density Polyethylene
PS	Poliestireno
PDA	Personal Digital Assintant
WPAN	Wireless Personal Area Network
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WWAN	Wireless Wide Area Network
WiFi	Wireless Fidelity
HIPERLAN	High Performance Radio Local Area Network
IEEE	Institue of Electrical and Electronics Engineers
RLD	Reference Listed Drug
QTPP	Quality Target Product Profile
CTD	Comparative Toxicogenomics Database
PWM	Pulse Width Modulation
USB	Universal Serial Bus
HDMI	High Definition Multimedia Interface
VDC	Tensión de corriente directa
VCA	Tensión de corriente alterna



CD	Corriente Directa
MOSFET	Transistor de Efecto de Campo Metal-Óxido-Semiconductor
DSI	Display Serial Interface
SPI	Serial Peripheral Interface
PID	Proporcional Integral Derivativo
IA	Inteligencia Artificial
RNA	Red Neuronal Artificial
CAD	Computer-Aided Design
CBP	Cuanto Baste Para
N.C.	Normalmente Cerrado
mpp	Motor paso a paso
pps	Pulso por segundo
TT2	Trabajo Terminal 2
V_{GS}	Tensión de Gate a Source
V_{DS}	Tensión de Drain a Souce
IP	Internet Protocol
iOS	iPhone Operative System
CSS	Cascading Style Sheets
GUI	Graphical User Interface
TT1	Trabajo Terminal 1
GPIO	General Purpose Input/Output
ATX	Advanced Technology Extended

Simbología

mm	Milímetros
cm	Centímetros
m	Metros



N	Newton
Pa	Pascales
s	Segundos
rad	Radianes
°C	Grados Celsius
A	Ampere
V	Volt/Velocidad (según contexto)
gr, g	Gramo
kg	Kilogramo
l	Litro
bar	Bar
h, hr	Hora
RPM	Revoluciones Por Minuto
rev	Revolución
paso	Paso (de motor a pasos)
gf	Gramo fuerza
Hz	Hertz
π	3.1415
°	Grado sexagesimal
MXN	Moneda Nacional Mexicana
CFM	Pies cúbicos por minuto
ft	Pie
in	Pulgada
lb	Libra fuerza
psi	Libra por pulgada cuadrada
%	Porcentaje
dB	Decibeles



CF_1	Special Conveyor Pitch Factor
CF_2	Special Conveyor Flight Capacity Factor
CF_3	Special Conveyor Mixing Paddle Capacity Factor
F_b	Hanger Bearing Factor
F_d	Conveyor Flight Diameter Factor
HP	Potencia en caballos de fuerza
HPf	Potencia para superar la fricción
HPm	Potencia para transportar la capacidad de material especificada
F_f	Flight factor
F_m	Material factor
F_o	Overload factor
F_p	Paddle factor
C	Capacity
e	Drive efficiency
L	Total length of conveyor/Length of screw between bearings
N	Operating speed
W	Apparent density of material as conveyed/Total weight of assembled screw
Δ	Deflection at mid-span
E	Modulo de elasticidad
I	Momento de inercia/Corriente eléctrica (según contexto)
T	Torque
M	Momento flector
v_b	Velocidad de salida en la boquilla
g	Gravedad
h	Altura del fluido
Q	Caudal
A_b	Área de la boquilla



dt	Diferencial de tiempo
dh	Diferencial de altura
∇	Flujo volumétrico
A_t	Área del tanque
Δp	Pérdida de presión del componente
K	Coefficiente de resistencia
ρ	Densidad
v	Velocidad media en la sección transversal
d	Diámetro
C_v	Coefficiente de caudal, sistema inglés
K_v	Coefficiente de caudal, sistema métrico
A	Área
R_1	Ruta 1
R_2	Ruta 2
R_3	Ruta 3
F	Fuerza
l_{pastilla}	Longitud de la pastilla
d_m	Diámetro medio
ΣF	Suma de fuerzas
R	Reacción en soporte fijo
σ_{axial}	Esfuerzo axial
c	Distancia al centroide de la sección transversal de la hélice
ΣM	Suma de momentos
σ_{flexor}	Esfuerzo flector
σ_{TS}	Esfuerzo total parte superior de la viga
σ_{TI}	Esfuerzo total parte inferior de la viga



$\mu_{comprimidos}$	Coeficiente de fricción de los comprimidos
$\mu_{tabletas}$	Coeficiente de fricción de las tabletas
$\mu_{cápsulas}$	Coeficiente de fricción de las cápsulas
$\lambda_{hélice}$	Ángulo de avance del tornillo
$\beta_{hélice}$	Ángulo de inclinación de la hélice
V	Volumen de un cuerpo
Z	Altura de punto de análisis respecto al nivel de referencia
$Z_{válvula}$	Altura de la válvula respecto al nivel de referencia
$P_{válvula}$	Presión hidrostática en la válvula cuando está cerrada
α	Aceleración rotacional del motor
ω	Velocidad angular del motor
θ	Desplazamiento angular
t	Tiempo
ω_{rpm}	Velocidad angular en rpm
I_z	Momento de inercia en el eje z de la pieza
T_i	Torque de inercia
$T_{Pull-In}$	Torque de Pull-In del motor
f_{pps_mpp}	Frecuencia de pulso por segundo del motor a pasos
m_s	Masa trasladable por el solenoide
I_{DTmax}	Corriente teórica de drenador máxima
I_{DMmax}	Corriente de simulación de drenador
I_{Dmin}	Corriente mínima de drenador
P	Potencia eléctrica
$P_{min\ fuente}$	Potencia mínima de la fuente



Introducción

Los dispensadores de medicamentos tienen su origen desde los pastilleros de plástico o cerámica, los cuales servían principalmente para almacenar las pastillas y evitar que se pierdan o se maltraten. Luego se procedió a crear pastilleros que ayudaran a las personas a seguir sus tratamientos médicos, esto mediante separaciones y marcando los días de la semana en cada separación. Esto obviamente fue de utilidad para poner los medicamentos correspondientes para cada día, pero existía el error de que el usuario se equivocara al momento de guardar una pastilla en la separación que no debía ser, y también que el mismo usuario debía ponerse una alarma, lo cual no era seguro que pasara.

Entonces empezaron a diseñar dispositivos que fueran capaces de organizar los medicamentos, y que avisara al usuario para que tomara la pastilla indicada, de aquí se partió para crear los dispensadores de medicamentos, a los cuales se les han agregado más funciones para que el usuario siga al pie de la letra el tratamiento médico indicado.

Ahora se procede a dar datos alentadores para que se pueda apreciar la utilidad que tiene un dispensador de comprimidos y cápsulas.

México es uno de los países con más hipertensos del mundo, según el Instituto Nacional de Salud Pública (INSP), entre el 20% y 40% de los mexicanos adultos padecen de esta condición médica [1]. Además, de acuerdo al Instituto Mexicano del Seguro Social, el país ocupa el segundo lugar en prevalencia de la obesidad [2] y según la Universidad Nacional Autónoma de México, en el año 2019 ocupó el noveno lugar en prevalencia de diabetes y se estima que para el 2025 ascenderá al séptimo [3]. Debido a dichas condiciones médicas, los mexicanos se ven obligados a mantenerse bajo tratamientos médicos constantes, sin embargo, suele ser una tarea que olvidan llevar a cabo durante el día, debido a sus actividades, suelen omitir la ingesta de sus píldoras y/o tomar medicamentos a deshora o confundir los horarios de los mismos. Pero, esa situación, no solo acontece en las personas que permanentemente se están medicando, sino también en las personas que debido a una enfermedad pasajera deben someterse a un tratamiento médico, además, esto ocurre en pacientes de todas las edades, aunque suele ser más común a partir de los 45 años.

Según la Sociedad Española de Farmacéutica Comunitaria, uno de cada cuatro pacientes olvida tomar su medicación, es decir, el 25% de los pacientes confiesan haber olvidado la hora de su medicación o tomarla al menos en una ocasión [4].

Además, un problema subyacente que genera la mala gestión de la ingesta de medicamentos es la resistencia bacteriana. De acuerdo a la OMS “la resistencia a los antibióticos es hoy una de las mayores amenazas para la salud mundial, la seguridad alimentaria y el desarrollo” [5]. Los pacientes generalmente creen que el hecho de sentirse mejor después de unos días de tomar antibióticos es indicador de la erradicación de la infección que padecían y, por lo tanto, dejan de tomar los medicamentos recomendados permitiendo así que la infección bacteriana permanezca en su organismo y desarrolle un grado de resistencia a ese antibiótico en específico, agravando después la



infección, teniendo que recurrir a antibióticos más fuertes [6]. Un ejemplo de esta situación es que “en Paraguay, entre las resistencias que más preocupan se encuentra la emergencia del *Staphylococcus aureus* resistente a la meticilina” [7], que antes era tratable con Cefalexina, pero hoy, ni ese antibiótico ni la amoxicilina, sulbactam, tazobactam tienen efecto sobre la bacteria.

Por las razones anteriores, se ha decidido realizar un dispositivo dispensador de comprimidos configurable y administrable desde una aplicación en el teléfono celular del usuario, pues consideramos que su construcción ayudaría a las personas a tomar los medicamentos que le corresponden a la hora que debe ingerirlos, esto contribuye a que el tratamiento sea constante y efectivo. Además, que incentiva a que los pacientes no suspendan voluntariamente sus tratamientos, sino que los terminen cuando es indicado y en el caso de tratamientos permanentes, a que no se interrumpa debido a algún olvido por parte del usuario.

Planteamiento del Problema

Como ya se ha mencionado en la introducción es de suma importancia que las personas que se encuentran bajo tratamiento médico ingieran sus medicamentos en tiempo y forma, ya que el no hacerlo de dicha manera, puede significar un daño a su salud, así como un riesgo para sus vidas debido a que pueden desarrollar una resistencia bacteriana.

Para tratar dicho problema en la actualidad se han desarrollado algunos dispensadores comerciales con características avanzadas, tales como: comunicación con el teléfono móvil del usuario a través de aplicaciones, recopilación de datos sobre la ingesta de medicamentos, sistemas de recuento de comprimidos disponibles, medidores de ritmo cardíaco, sistemas de horario de tratamiento autoajustables, interfaces de usuario táctiles; siendo estas las principales funciones en esos dispositivos.

Conociendo el contexto actual sobre la problemática y las características de las soluciones existentes, se ha diseñado y construido un dispositivo dispensador de comprimidos y capsulas que cumpla con ciertas características avanzadas, tales como: comunicación con el teléfono móvil para configurarlo y administrarlo, un sistema de horario auto-ajustable, así como un sistema para almacenar, gestionar y dispensar los medicamentos y agua potable para que el usuario pueda ingerirlos; con la finalidad de ayudar a las personas a llevar sus tratamientos médicos de forma más efectiva y correcta.

Uno de los principales retos es la comunicación con el teléfono móvil del usuario, pues deberá desarrollarse una aplicación móvil que tenga algún protocolo de comunicación con el dispensador, ya que se enviarán datos de forma bidireccional.

Otro reto a abordar fue el mecanismo para dispersar los comprimidos y capsulas que contenga el dispensador, dado que existen de diferentes tamaños y formas, lo cual requirió de un diseño que permitiese entregar al usuario la cantidad correcta de medicamento sin importar el tamaño o forma de la capsula o comprimido.



También se diseñó y desarrolló un sistema con horario autoajutable para situaciones en las que se deba re-agendar el horario de ingesta de medicamentos debido a omisiones o retrasos por imprevistos del usuario.

Por otra parte, la forma y tamaño del dispositivo son de diseño propio, lo que representó otro reto particularmente complejo de atender ya que se buscó uno que contemplase un tamaño suficiente para contener los sistemas electrónicos y electromecánicos ya diseñados. El equipo priorizó la funcionalidad sobre la estética.

Propuesta de solución

De acuerdo a la problemática que se busca atender, mencionada en la introducción y planteamiento del problema, y buscando resolverla de forma innovadora con respecto a las tecnologías existentes presentadas con anterioridad en este documento, se propone como solución, el diseño y construcción de un dispositivo dispensador de comprimidos y cápsulas configurable y administrable desde una aplicación móvil que ayude al usuario a llevar su tratamiento médico con la mejor adherencia terapéutica posible y sin interrupciones involuntarias o indebidas. Para ello, se realizará un análisis de elementos que se deben tomar en cuenta para el desarrollo del prototipo, las cuestiones técnicas, requerimientos de diseño y funciones que debe cumplir. Además de establecer un plan de trabajo para tener orden en las áreas que se abordarán y que los resultados sean acordes a los objetivos establecidos.

Para poder realizar un proyecto de forma adecuada y estructurada, es necesario apegarse a alguna metodología de trabajo, en este caso, se empleará una basada en el “modelo en V” con la finalidad de lograr un resultado satisfactorio en el desarrollo del proyecto.

Para poder realizar un proyecto de forma adecuada y estructurada, es necesario apegarse a alguna metodología de trabajo, en este caso, se empleará una basada en el “modelo en V” con la finalidad de lograr un resultado satisfactorio en el desarrollo del proyecto.

Justificación

Teniendo en cuenta los datos y cifras sobre diabetes, hipertensión y obesidad en México ya mencionados en la introducción, podemos notar que existe una gran cantidad de personas que, debido a dichas condiciones de salud, se encuentran bajo tratamientos médicos constantemente o incluso de forma permanente. Adicionalmente, como se indica en esa misma parte de este documento, se estima que el 25% de los pacientes confiesan olvidar tomar su medicación o ingerirla en el horario indicado. Por otra parte, se sabe que la falta de adherencia terapéutica resulta ser aún más



predominante en pacientes crónicos, en quienes se estima que el 50% de los pacientes no sigue de forma adecuada sus tratamientos y dicha cifra suele ser mayor en los países en desarrollo [8].

Como se ha mencionado anteriormente, la falta de adherencia terapéutica es un problema de atención médica de carácter mundial para la ONU y existen múltiples investigaciones y artículos sobre las repercusiones que este mal hábito puede provocar, no solo en los pacientes, sino en las comunidades y en general sobre la humanidad, el ejemplo más claro sobre esto es la resistencia bacteriana.

En la actualidad, en el mercado mexicano, existen algunos dispositivos que buscan atacar ese problema, los cuales son diferentes tipos de pastilleros, como los electrónicos, los cuales avisan al usuario la hora a la que se deben tomar sus medicamentos, por medio de una alarma; también existen los pastilleros convencionales donde el usuario organiza sus medicamentos y él debe llevar control del tratamiento. Sin embargo, hasta el momento no se ha encontrado un producto comercial o proyecto académico con las mismas especificaciones como que el que busca desarrollar el equipo, pues en este caso particular, no se trata de un pastillero sencillo o electrónico sino de un dispensador con muchas más funciones, las cuales también ya han sido descritos en el planteamiento del problema.

Por otra parte, en el mercado estadounidense y europeo si se han encontrado tanto productos comerciales como proyectos académicos similares, sin embargo, en el caso de los productos comerciales, aunque comparten varias funciones, estos no cuentan con la función de dispensar junto con los comprimidos o capsulas, el agua que el usuario requiere para ingerirlas, y tampoco cuentan con la función de entregar al usuario en un contenedor los medicamentos que requerirá para un día completo o más días en caso de no estar en casa por algún viaje u otra situación y, por supuesto, el precio de dichos aparatos ronda los \$500.00 USD más gastos de importación. Las diferencias con los proyectos académicos son las mismas y además se agrega que algunos modelos no cuentan con ninguna forma de comunicación con dispositivos móviles o la comunicación con dispositivos es de carácter informativo. En el caso del que se busca desarrollar en este proyecto la comunicación con dispositivos móviles será para la administración y gestión del dispensador.

Teniendo en consideración que aún no se ha encontrado un producto comercial idéntico o de prestaciones cercanas en el mercado nacional y que los costos de los productos de importación son muy elevados, se ha decidido desarrollar un dispensador de comprimidos y capsulas configurable que pueda entregar al usuario el medicamento que debe ingerir, pueda dispensar el agua potable para el usuario y además tenga la función de entregar al usuario el medicamento que requiere para uno o más días en un organizador de viaje, siendo estas las principales ventajas sobre los dispositivos comerciales existentes, además de la búsqueda de la reducción de costos.



Objetivos

Objetivo General

Diseñar y construir un dispositivo dispensador de comprimidos y cápsulas configurable y administrable desde una aplicación móvil para ayudar a las personas a mejorar su adherencia terapéutica.

Objetivos Particulares

Objetivos Específicos de trabajo terminal 1

- Diseñar un mecanismo para identificar y separar tres tipos de comprimidos y cápsulas de diferente tamaño y geometría en compartimentos separados adecuados para su almacenamiento.
- Diseñar una aplicación móvil en un lenguaje de alto nivel para que el usuario pueda configurar cada cuanto tiempo ingiere sus distintos medicamentos, que despliegue notificaciones cada vez que sea tiempo de ingerir la dosis y cuando esta haya sido omitida y, además, funcione como interfaz gráfica para el usuario brindándole información y parámetros de operación del dispensador.
- Diseñar el programa de un sistema de alarma integrado en el dispositivo que avise al usuario que es el momento de la ingesta de un medicamento y que insista hasta en 3 ocasiones para que el usuario tome su medicamento.
- Diseñar los mecanismos para proveer al usuario de la cápsula o comprimido que le corresponde ingerir, así como el agua que requiere para tomarla, en recipientes que formarán parte del mismo aparato.
- Diseñar un programa que permita administrar y gestionar el medicamento para hacerle saber al usuario que le quedan tres dosis y pueda abastecer el dispositivo de forma oportuna.
- Diseñar un módulo que permita al usuario llevarse los medicamentos que le toca consumir a lo largo de uno a tres días en un pastillero portátil (para situaciones de viaje o cualquier otra que le impida estar en casa).
- Validar computacionalmente los módulos electrónicos y mecánicos en un software especializado para cada área.

Objetivos Específicos de trabajo terminal 2

- Construir el mecanismo para la identificación y separación de capsulas y comprimidos en recipientes separados para el almacenamiento de los mismos, así como el set de compartimentos requeridos.
- Desarrollar una aplicación móvil en un lenguaje de alto nivel para que el usuario pueda configurar cada cuanto tiempo ingiere sus distintos medicamentos, que despliegue



notificaciones cada vez que sea tiempo de ingerir la dosis y cuando esta haya sido omitida y, además, funcione como interfaz gráfica para el usuario brindándole información y parámetros de operación del dispensador.



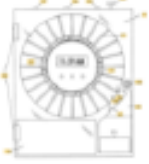
- Implementar un sistema de alarma integrado en el dispositivo que avise al usuario que es el momento de la ingesta de un medicamento y que insista hasta en 3 ocasiones para que el usuario tome su medicamento. Así como implementar un esquema de control para regular la temperatura del dispositivo y mantenerla menor a los 30° Celsius.
- Construir los mecanismos que puedan proveer al usuario de los comprimidos o cápsulas que le corresponde ingerir, así como el agua para tomarla, en recipientes que formarán parte del mismo aparato.
- Desarrollar el programa que permita administrar y gestionar el medicamento para hacerle saber al usuario que le quedan tres dosis y pueda abastecer el dispositivo de forma oportuna.
- Implementar un módulo que permita al usuario llevarse los medicamentos que le toca consumir a lo largo de uno a tres días en un pastillero portátil.
- Verificar el sistema mecatrónico mediante la realización de pruebas de funcionamiento de los diferentes sistemas que integran el dispositivo.

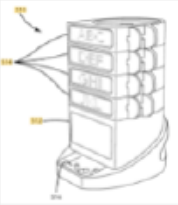
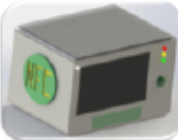

Objetivos Específicos de trabajo terminal 2 en el caso de continuar con clases no presenciales por el virus covid-19

- Desarrollar una aplicación móvil en un lenguaje de alto nivel para que el usuario pueda configurar cada cuanto tiempo ingiere sus distintos medicamentos, que despliegue notificaciones cada vez que sea tiempo de ingerir la dosis y cuando esta haya sido omitida y, además, funcione como interfaz gráfica para el usuario.
- Desarrollar el circuito electrónico del sistema de alarma que avise al usuario que es el momento de la ingesta de un medicamento y que insista hasta en 3 ocasiones para que el usuario tome su dosis, y cuente con un botón de paro.
- Desarrollar un prototipo del sistema de control de temperatura que permita mantener un espacio con un volumen no mayor a 10 cm³ a una temperatura menor a los 30° Celsius.
- Desarrollar el programa que permita administrar y gestionar el medicamento para hacerle saber al usuario que le quedan tres dosis.
- Construir el módulo para la identificación de 3 tipos de capsulas y comprimidos de tamaños y geometrías diferentes para su futura separación.

Antecedentes

A continuación, se presenta una tabla de antecedentes referentes a los dispositivos que tienen ciertas características similares a las que se pretende tenga el dispensador propuesto.

Item	Nombre	Descripción	Características	País	Instituto	Tipo	Referencia	Año
1	Pill dispenser 	Dispensador de píldoras que incluye una carcasa que define una abertura, un mecanismo dispensador de píldoras dispuesto dentro de la carcasa y acoplado operativamente a la abertura, un receptáculo acoplado operativamente a la carcasa, una primera cámara de visualización de píldoras colocada para capturar una imagen del receptáculo, una cámara de identificación colocada para capturar una imagen de un área adyacente a la carcasa, al menos un procesador en comunicación operativa con el mecanismo dispensador, la primera cámara y la cámara de identificación, y un medio de almacenamiento.	<ul style="list-style-type: none"> Camara de identificación de medicamento y usuarios autorizados Altavoces dispuestos para uso y configuración deseada por el usuario y control de voz Pantalla táctil para el uso y configuración del usuario Sistema de posicionamiento global y tecnología NFC Sistema de carga inalámbrico 	Estados Unidos	DEKA Products Limited Partnership	Patente	DEKA Products Limited Partnership. (2016). Pill dispenser. Manchester, NH.	2016
2	Electronic pill dispenser 	Un dispensador de píldoras electrónico incluye un recipiente y una tapa unida al recipiente que se puede quitar. Los componentes del dispensador de píldoras incluyen una fuente de alimentación, circuitos del dispensador de píldoras, un reloj en tiempo real, un contador, una pantalla, un mecanismo de dispensación, un sensor, un indicador visual, un indicador audible, una interfaz de entrada / salida, un puerto de entrada y salida, y un bus de comunicación que interconecta eléctricamente los componentes. El dispensador de píldoras también puede incluir un indicador físico, un mecanismo de bloqueo, un transceptor, una antena y un módem.	<ul style="list-style-type: none"> Reloj en tiempo real y contador Pantalla con interfaz entrada/salida Transceptor, antena y modem integrado 	Estados Unidos		Patente	Varvarelis, N. M., & Samuelson, J. (2008). Electronic pill dispenser. Philadelphia, PA.	2008
3	Programmable automatic pill dispenser 	Un dispositivo que puede cargarse con las píldoras adecuadas y programarse para dispensar automáticamente la cantidad y tipo adecuado de píldoras a la hora asignada cada día. El dispositivo incluye un sistema para alertar a la persona que toma la píldora de que se han dispensado las píldoras, un sistema para proporcionar mensajes de voz para entrenar a la persona que toma la píldora a usar el dispositivo y consumir las píldoras, y un sistema para alertar a un cuidador fuera del sitio cuando el usuario toma la píldora, cuando el usuario no ha respondido como se requiere o cuando hay un problema con el funcionamiento del dispositivo.	<ul style="list-style-type: none"> Temporizador programable Compartimiento para almacenar botellas de píldoras Sistema de grabación y reproducción de mensajes de voz Sistema marcador telefónico automático 	Estados Unidos		Patente	Lim, J. (2008). Programmable automatic pill dispenser. Oakland, CA.	2003

4	<p>Pill dispenser</p> 	<p>Un sistema dispensador de píldoras que incluye paquetes de píldoras que se pueden usar para dispensar píldoras manualmente o con un sistema dispensador para proporcionar una funcionalidad mejorada. Los envases se pueden proporcionar con información relativa a las píldoras envasadas o al uso de las píldoras envasadas. Cuenta con capacidad de leer la información del paquete e interpretarla. Proporciona recordatorios de cuándo se deben tomar las píldoras. Puede tener la capacidad de conectar un dispositivo electrónico específico a un usuario específico y el sistema dispensador puede requerir que el dispositivo electrónico esté cerca del sistema dispensador antes de dispensar píldoras para ese usuario.</p>	<p>Sistema de envases de píldoras apilables</p> <hr/> <p>Base del sistema usada como gestor de datos de entrada y salida</p>	Estados Unidos	ACCESS BUSINESS GROUP INTERNATIONA L LLC	Patente	Baarman, D. W., Eulich, S. T., Mollema, S. A., Dean, C. D., Kuyvenhoven, N. W., Runyon, M. K., Van Den, J. C., & Schamper, R. D. (2015). Pill dispenser. Ada, MI.	2015
5	<p>Proyecto de dosificador de pastillas para personas con autonomía reducida</p> 	<p>Sistema programable que avisa al usuario la hora de la ingesta de medicamento al menos 3 veces al día. Pantalla LCD con 6 botones, alternativamente se implementa una pantalla táctil. Sistema de alarma por bocina además de una pulsera vibratoria que almacena datos como alarma alternativa. Sistema de reconocimiento por voz. Sistema de energía por batería así como por corriente eléctrica.</p>	<p>Conexión con telefonía e internet y tecnología NFC</p> <hr/> <p>Pantalla LCD con botones</p> <hr/> <p>Altavoz y reconocimiento de voz</p> <hr/> <p>Pulsera vibratoria como método de aviso adicional con controlador</p> <hr/> <p>Batería de litio recargable y sistema de carga provisional con corriente AC</p>	España	Universidad Politécnica de Cataluña	Tesis	Muñoz Gonzalez, T. (2016). Proyecto de dosificador de pastillas para personas con autonomía reducida. Cataluña, España.	2016
6	<p>Prototipo dispensador automático de medicamentos para personas de la tercera edad</p> 	<p>Sistema dispensador programable. Se puede cargar con hasta 6 tipos de medicamentos diferentes en un rango de 1,8 -2,5 cm de largo y 0,6 y 1 cm de ancho. Dispensador de agua de hasta 5 onzas. Alertas sonoras con comunicación a dispositivos móviles. Conexión a internet. Sistema conectado permanentemente a una fuente de suministro eléctrico. Interfaz a través de pantalla LCD táctil. El sistema se carga con un máximo de 20 pastillas y un mínimo de 2.</p>	<p>Sistema dispensador de agua</p> <hr/> <p>Control de humedad y temperatura</p> <hr/> <p>Pantalla LCD interfaz táctil</p> <hr/> <p>Conexión a internet y tecnología NFC</p>	Colombia	Universidad del bosque	Tesis	Velandia Llanos, C. C., & Rocancio Rojas, A. F. (2020). Prototipo dispensador automático de medicamentos para personas de la tercera edad. Bogotá, Colombia.	2020

7	<p>Diseño e implementación de un prototipo de máquina dispensadora de medicamentos en forma de pastillas para personas de la tercera edad que padecen enfermedades no transmisibles</p> 	<p>Sistema trabaja en una red de 100v con un regulador de voltaje. Microcontrolador ATmega 2560. Comunicación con puerto serial y con la pantalla táctil. Interfaz amigable con el usuario y configurable a horarios deseados. Peseee actuadores y un retroalimentado a través de 4 sensores luminicos. Registro de dispensado a través de notificaciones.</p>	<p>Pantalla táctil para el uso y configuración del usuario</p> <p>Sistema de Alarma auditiva y vibratorias</p> <p>Tecnología NFC</p>	Ecuador	Universidad Politécnica Salesiana	Tesis	<p>Morcoso Nagra, S. Javier, & Milaces Miranda, F. (2020). Diseño e implementación de un prototipo de máquina dispensadora de medicamentos en forma de pastillas para personas de la tercera edad que padecen enfermedades no transmisibles. Ecuador</p>	2020
8	<p>Dispensador automatico de pastillas</p> 	<p>Sistema dispensador programable. Regulador de corriente directa. Los datos ingresados y opciones seleccionadas son almacenados y procesados a través del controlador. El proceso de dosificación consiste en posicionar el contenedor con las pastillas requeridas en la parte central inferior</p>	<p>Alimentación 220v AC</p> <p>Pantalla táctil para el uso y configuración del usuario</p> <p>Sistema de operación Beagle Bone Black</p>	Peru	Universidad Católica del Peru	Tesis	<p>Sánchez Noriega, J. L. (2015). Dispensador automatico de pastillas. Peru.</p>	2015
9	<p>Hero</p> 	<p>Dispensador comercial de píldoras que recuerda al usuario la toma de su dosis programada. Tecnología NFC que notifica al usuario directamente a sus dispositivos móviles. Cartuchos limitados con cantidad de pastillas limitadas. Acepta únicamente pastillas, estas pueden ser de cualquier tamaño pero no medias píldoras o píldoras en polvo. Acceso a internet, WiFi y computadora.</p>	<p>Tecnología NFC</p> <p>Conexión a internet</p> <p>Cartuchos de pastillas</p> <p>Actualización de software</p> <p>Pantalla LCD</p> <p>Sistema de zumbidos y alertadores</p>	Estados Unidos	Hero Health, Inc	Producto		2020
10	<p>Meda Cube</p> 	<p>Dispensador comercial de medicamento, puede manejar píldoras, Inhaladores o insulina. Sistema de seguridad desbloqueable por PIN. Sistema de inclinación que bloque al sistema en caso de estar colocado incorrectamente. Notificaciones al celular o por correo electrónico a través de Wi Fi. Manejo de 16 compartimentos de diferentes tamaños. Almacenamiento de hasta 90 días. Horarios programables por el usuario. Sistemas de alarma por audios y audios personalizados.</p>	<p>Pantalla LCD táctil</p> <p>Conexión a internet</p> <p>Cartuchos de diferente tamaño</p> <p>Almacenamiento de información</p> <p>Tecnología NFC</p> <p>Sistema de seguridad</p> <p>Sensor de inclinación</p> <p>Funcionalidad para otros medicamentos</p> <p>Mensajes de audio personalizados</p> <p>Batería de respaldo</p>	Estados Unidos	PharmAdva	Producto		2020
11	<p>Pria</p> 	<p>Betiquin comercial de dosis de medicamentos. Almacenamiento de 28 dosis por hasta 90 días. Pantalla LCD táctil para interactuar con el usuario. El sistema puede enviar notificaciones a dispositivos móviles.</p>	<p>Pantalla LCD táctil</p> <p>Temporizador programable</p> <p>Reconocimiento facial o por PIN</p> <p>Tecnología NFC</p>	Estados Unidos	Pilo Health	Producto		2019

Dispensador de comprimidos y capsulas



Organización del documento

Este reporte se encuentra conformado principalmente por trece capítulos, en los cuales se encuentran repartidos y contenidos los aspectos esenciales para el desarrollo de este proyecto.

El capítulo primero, consiste principalmente en las definiciones, conceptos y teoría consultada respecto a sistemas de almacenamiento de medicamentos, sus condiciones de almacenamiento, materiales aptos para el almacenamiento de medicamentos. También se incluye teoría respecto a procesos de diseño mecánico de mecanismos de transporte de carga, sobre métodos de clasificación de objetos y conceptos esenciales sobre sistemas de visión artificial, así como mecánica de fluidos, conectividad a internet. Estos temas resultan de amplio interés para este proyecto pues serán aplicados a lo largo de su diseño y desarrollo.

El capítulo segundo describe a detalle la metodología de diseño empleada por el equipo para el desarrollo de este proyecto, así como su forma de implementación.

En el capítulo tercero, se detalla el proceso de diseño del sistema, desde el diseño conceptual que implica la búsqueda de los requerimientos y la propuesta del concepto del sistema, hasta el diseño detallado en el que se rediseñan y desarrollan las ideas conceptuales del proyecto y, además, se describen los procesos de evaluación y validación empleados para la selección de componentes y los diseños desarrollados por el equipo. Así como las actualizaciones de diseño realizadas a lo largo del proyecto.

El capítulo cuarto, describe los procesos de manufactura empleados por el equipo para la fabricación de las piezas que conforman el prototipo, así como las pruebas realizadas durante dicho proceso

El capítulo quinto detalla el proceso de ensamble, construcción y desarrollo de los sub sistemas del prototipo y explica sus funcionamientos, así como la integración del software.

El capítulo sexto detalla los resultados de los experimentos realizados sobre los diferentes módulos y códigos desarrollados por el equipo, también en este se muestran los resultados del ensamble general del prototipo.

El capítulo séptimo está constituido por los análisis de los resultados obtenidos durante y al finalizar la construcción e integración de los módulos del dispositivo diseñado, respecto de los objetivos específicos planteados y los alcances de proyecto establecidos.

El capítulo octavo contiene los costos de fabricación del dispositivo; mientras que el capítulo noveno detalla los cronogramas de trabajo seguidos por el equipo a lo largo del desarrollo del proyecto.



El capítulo decimo está constituido por las conclusiones del equipo respecto al proyecto desarrollado considerando los análisis de resultados realizados.

El capítulo onceavo contiene las fuentes de información y de imágenes consultadas por el equipo a lo largo del desarrollo de este proyecto.

El capítulo doceavo son los apéndices, en el cual, se adjunta información adicional sobre el diseño y desarrollo del sistema como son los planos de las piezas diseñadas, planos de ensamble y diagramas eléctricos, códigos de la aplicación móvil y programa de la tarjeta de control.

Por último, encontramos el capítulo treceavo denominado anexos, en este, se adjuntan las hojas de datos de los componentes seleccionados en el diseño detallado del sistema, las cuales contienen las características de operación, así como dimensiones e imágenes del componente.



I. Marco Teórico

Concepto de formas farmacéuticas

Se entienden como la forma en la que los componentes activos se acoplan para obtener un medicamento que se pueda administrar al cuerpo humano. Para dicho acoplar los componentes, se realiza un proceso en el que se le da al medicamento ciertas características como lo son: “dosificación, eficacia terapéutica y estabilidad en el tiempo” [9].

Según su estado físico existen las siguientes formas terapéuticas: sólidas, semisólidas y líquidas. Y por la vía de administración se clasifican en: oral, rectal y vaginal, tópica y subcutánea, oftálmica y ótica, parenteral e inhalatoria. Por lo que, la vía más usada es la vía oral. Ahora bien, referente a las formas orales sólidas, se pueden diferenciar principalmente en dos grupos: los comprimidos (son el resultado del proceso de comprimir los granos, esto mecánicamente) y cápsulas (son de una cubierta gelatinosa y dentro de éstas está dosificado el fco).

Dentro de los comprimidos se encuentran: los normales, efervescentes, masticables, bucales, sublinguales, grageas y los de recubrimiento pelicular. Para las capsulas hay dos tipos: las capsulas duras o las blandas [9].

Condiciones de conservación y almacenamiento de productos farmacéuticos

Medicamentos, así como otros insumos para el cuidado de la salud, deben cumplir con estándares de conservación, los cuales ayudan a mantener su calidad a lo largo de su vida útil.

En general, los productos farmacéuticos deben cumplir con el requerimiento de estar a temperatura ambiente, la cual no debe ser mayor a 30°C, el sitio debe estar seco y sin luz directa del sol. Cabe mencionar que existen otros productos que por sus propiedades específicas tanto físicas, químicas o terapéuticas, requieren condiciones particulares.

“Las condiciones de almacenamiento deberán conservarse de acuerdo a las recomendaciones del fabricante, las cuales están definidas con base a los resultados de los estudios de estabilidad realizados conforme a la norma oficial mexicana vigente que corresponda” (NOM-073-SSA1-2015, Estabilidad de fármacos y medicamentos, así como de remedios herbolarios): Farmacopea de los Estados Unidos Mexicanos, décima edición, generalidades, página 17 [10].

Para la conservación de los medicamentos y evitar su deterioro es necesario que la humedad relativa a la que están expuestos no sea superior al 65%, cabe mencionar que el aire es también un factor de degradación, pues contiene oxígeno y humedad, por lo que los envases deben permanecer cerrados para resguardar los comprimidos y evitar su degradación. [11]



La humedad relativa es “la relación entre la cantidad de vapor de agua contenida en el aire y la máxima cantidad que el aire sería capaz de contener a esa temperatura”. [12]

Debido a que la luz también contribuye al deterioro de los medicamentos, los envases no son completamente transparentes, pero si llegan a ser translúcidos, estos filtran la luz por su tonalidad opaca o incluso llegan a ser completamente opacos lo cual impide la entrada de luz al contenedor primario. [11]

Para el almacenamiento de los medicamentos, suelen emplearse diferentes tipos de contenedores y materiales, sin embargo, el envase principal es aquel que está en contacto directo con el medicamento, al cual se le conoce como envase primario. Este puede llegar a ser de vidrio, plástico, elastómero o metal, esto de acuerdo a la norma ISO 15378 del 2017 para materiales de envases primarios de medicamentos.

En la industria farmacéutica, normalmente el contenedor primario está, a su vez, contenido en un envase o empaque de cubierta al cual se le denomina como envase secundario. [13]

Materiales utilizados en la construcción de contenedores primarios de medicamentos

Los registros legales de normas médicas recopilan información acerca de los materiales que deben ser utilizados en la fabricación de envases de uso farmacéutico, no obstante, otros polímeros aprobados podrán ser empleados en cada caso por la Autoridad competente responsable de la autorización y venta del preparado contenido en el envase (RFE, 2011).

De manera general, los principales materiales empleados para la fabricación de envases de tipo farmacéutico son el vidrio, el plástico, materiales elastómeros y metales.

En el caso de los envases primarios fabricados a partir de plástico, algunos de los polímeros más usados en la industria farmacéutica suelen ser el polietileno, polipropileno, cloruro de polivinilo (PVC) y poliestireno.

Para entender mejor las características de dichos materiales primero se debe definir de forma general qué son empezando por el tipo de material, en este caso el plástico que “son moléculas muy polimerizadas que una vez solidificadas pueden quedar con las más variadas cualidades de dureza, flexibilidad, transparencia y opacidad, insolubilidad total o solubilidad condicionada a determinados disolventes, con lo que se cubre un amplio campo de aplicaciones. Además de cumplir condiciones físicas de impermeabilidad, de resistencia mecánica y de temperatura (esterilización) hay que exigirles absoluta inocuidad y suficiente inercia química, la cual no resulta siempre satisfactoria en contacto con las formas farmacéuticas líquidas.” [14]

A continuación, se presentan las características generales de los polímeros frecuentemente usados para los envases primarios:



Poliétileno: es un material termoplástico, blanquecino, puede ser transparente o translúcido, usualmente fabricado en láminas transparentes muy finas. Para las secciones gruesas tienden a ser translúcidas y tienen una apariencia de cera. Mediante la polimerización de etileno se obtienen productos con propiedades físicas variadas, estos productos tienen en común la estructura química fundamental.

Algunas de las propiedades que hacen del polietileno un material primario tan conveniente de utilizar para una variedad de artículos manufacturados son, su bajo peso, flexibilidad, tenacidad y alta resistencia química. “La estabilidad e inercia químicas son adecuadas para las soluciones acuosas, por otra parte, no resulta tan fiable su contacto con disolventes del tipo orgánico o bien con sustancias grasas.” [14]

“Generalmente se describen dos tipos de polietileno: de baja densidad (LDPE) y de alta densidad (HDPE). El de baja densidad posee una estructura de cadena enramada, mientras que el polietileno de alta densidad tiene esencialmente una estructura de cadena recta.” [14]

Polipropileno: es un polímero de estructura similar al polietileno. Es un plástico de alta dureza y resistencia, con color opaco y gran resistencia al calor ya que solo se ablanda a una temperatura superior a los 150 °C. Posee una mayor resistencia impermeable al vapor de agua que el polietileno. Es altamente resistente a los golpes, aunque por otro lado posee muy baja densidad (densidad=0,9 g/ml) y se dobla muy fácilmente, resistiendo múltiples esfuerzos de fatiga. También es resistente a los productos corrosivos. No se puede esterilizar usando rayos gamma, ya que alteran su estructura y composición, sin embargo, es posible esterilizar por óxido de etileno. Es fácilmente oxidable, por lo tanto, debe evitarse el contacto con sustancias oxidantes.

Cloruro de polivinilo: es un polímero termoplástico que se describe como un material blanco que comienza a ablandarse aproximadamente de los 80°C y se descompone alrededor a los 140°C.

Entre sus características con respecto a la utilización en el envasado y acondicionamiento de medicamentos, destacan las siguientes: para el caso de la esterilización, es necesario hacerse con óxido de etileno, los plastificantes líquidos presentes en su composición tienen tendencia a migrar hacia el exterior provocando problemas de cesión de sustancias que pueden alterar o afectar el contenido del recipiente; absorbe el ácido bencílico y la mayoría de los conservadores. Por otra parte, no resiste el ataque químico de ésteres, cetonas, hidrocarburos aromáticos e hidrocarburos clorados. Presenta una baja permeabilidad al oxígeno y al agua, por lo que es posible preservar de manera satisfactoria el contenido de la oxidación.

Poliestireno: “El poliestireno (PS) es un polímero termoplástico que se obtiene de la polimerización del estireno. Existen cuatro tipos principales: el PS cristal que es transparente, rígido y quebradizo; el poliestireno de alto impacto resistente y opaco, el



poliestireno expandido muy ligero, y el poliestireno extrusionado, similar al expandido, pero más denso e impermeable.” [14]

Entre las principales aplicaciones del PS choque y el PS cristal están la fabricación de envases mediante la extrusión-termoformado, y la fabricación de diversos objetos a través del moldeo por inyección. Es más ligero, rígido y termoestable que el polietileno, y de la misma manera presenta características de inercia química. Tiene prácticamente las mismas aplicaciones que el polietileno como material para envasado y puede ser esterilizado en autoclave. Posee alta resistencia química en general, a pesar de que es incompatible con ácidos y álcalis diluidos, ésteres, cetonas, hidrocarburos, disolventes, esencias y aceites minerales.

Por otro lado, es altamente resistente a la acción de las sales (excepto la del sulfato de cobre y la del arseniato de plomo), de los alcoholes (a temperaturas inferiores a 50°C), etilenglicoles, propilenglicoles, glicerina, agua oxigenada, hidroquinona y trietanolamina. Además, es un material ligero, es fácil de transportar y manipular. Es poco permeable al vapor de agua, oxígeno y generalmente todo tipo de vapores y gases.

Redes inalámbricas

Las redes inalámbricas son un modelo de comunicación entre diferentes dispositivos PDA (Dispositivos de Uso Personal, por sus siglas en inglés) libre de cables, estas permiten la transferencia de datos, crear áreas de trabajo remotas, sustituir redes cableadas o crear redes de comunicación de bajo costo.

Estas redes funcionan a través de ondas de radio y, de hecho, existen diferentes modelos y tecnologías para esta forma de comunicación, los cuales utilizan diferentes frecuencias velocidades o alcance.

Una forma de categorizar las redes inalámbricas es por alcance, existiendo Redes Inalámbricas de: Área personal (WPAN), Área local (WLAN), Área metropolitana (WMAN) y Área amplia (WWAN).

Área local (WLAN): Estas redes son empleadas para la comunicación de dispositivos en espacios de hasta 100 metros, su uso se da principalmente para la creación de áreas de trabajo compartidas y transmisión de datos, además son comúnmente las redes locales que sirven como punto de acceso, es decir, fungen como puerta para comunicarse a Internet. La tecnología o estándares que comúnmente se emplea en estas redes de comunicación es la Wi-Fi, aunque existen otras como la HiperLAN.

Normalmente estas redes son utilizadas en escuelas, oficinas, casas y edificios, son de las redes más comunes hoy en día debido a que son las que permiten la conexión a internet, suelen trabajar con bandas de 2.4GHz o 5GHz, y pueden trabajar a diferentes velocidades de carga y descarga de datos. [15]



Comunicación Wi-Fi

Es el conjunto de especificaciones de comunicación para redes inalámbricas de área local (WLAN por sus siglas en inglés) las cuales están establecidas por el IEEE como estándar 802.11. Son redes de comunicación que permiten la conexión de varios PDA de forma simultánea y que además suelen tener comunicación a internet, es decir, cuentan con un punto de acceso.

La conectividad a internet de estas redes es lo que ha sido un detonante de gran importancia, pues permite tener comunicación no solo de forma local, sino con todo el mundo, permitiendo conectar con otros dispositivos, redes locales o servidores fuera de la red en la que se encuentra un usuario, esto significa que pueden crearse redes de trabajo remoto y transmisión de datos a larga distancia.

Dentro de los estándares de comunicación IEEE 802.11 existen otros subestándares relacionados con la velocidad de transmisión de datos, aunque en todos los modelos las bandas de transmisión son de 2.4 o 5 GHz, sobre todo en las redes comerciales. [15]

Teoría de clasificaciones

El método para la clasificación de objetos consiste en asignar un objeto específico en algún tipo de las clases que se van a manejar. Estos objetos pueden ser definidos por sus características propias, algunos ejemplos pueden ser el tamaño, la forma o el color de estos. Para lograr llevar a cabo la clasificación se deben definir las fronteras que existen entre las diferentes clases. Clasificar un objeto desconocido consiste en determinar sus características para posteriormente poder asignarlo en la clase correspondiente.

La clasificación suele ser utilizada preferentemente a otras técnicas cuando los objetos involucrados tienen bastante similitud pero que a su vez están sujetas a variaciones desconocidas. Los clasificadores se utilizan para:

- Segmentación de imágenes
- Reconocimiento de objetos
- Control de calidad
- Detección de novedad
- El proceso de clasificación, independientemente del tipo de clasificador que se elija, consta de la siguiente serie de pasos:
- Se reúnen muestras de objetos de clases conocidas, después de eso es elegido un juego de características para esa clase (vector de características) y se calculan las características de las muestras.
- Los vectores de características se usan para entrenar al sistema clasificador y establecer las fronteras entre sus clases.
- Se utilizan objetos desconocidos para extraer las mismas características y clasificarlos.

- El clasificador a través de las fronteras calculadas puede decidir a qué clases pertenecen los vectores de características de los objetos nuevos que se quieran reconocer. [16]

Tamaños y formas de comprimidos para administración oral

El peso, las formas y el tamaño de estos varían entre sí. Por lo general, el tamaño está en el rango entre 5 y 17 mm, mientras que el peso está entre 0.1 y 1.0 g. La forma puede ser redonda, oblonga, ovoide, biconvexa, etc. En la superficie suelen tener alguna inscripción y una ranura, esto para fraccionarlos y hacer más fácil el ajuste posológico (este término se refiere a la forma en que se determina la dosis con la cual se debe administrar el medicamento) a las necesidades individuales. [17]

Tamaño, forma y otros atributos físicos de las cápsulas y tabletas genéricas

Los comprimidos y cápsulas se fabrican y prescriben ampliamente y pueden proporcionar una serie de ventajas sobre otras formas de dosificación, incluida la facilidad de almacenamiento, portabilidad, facilidad de administración y precisión en la dosificación. Si bien se requiere que las formulaciones genéricas de estos productos farmacéuticos sean tanto farmacéutica como terapéuticamente equivalentes a un medicamento listado de referencia (RLD por sus siglas en inglés), es importante considerar las diferencias en las características físicas (Tamaño y forma de la tableta o cápsula) que puedan afectar el cumplimiento del tratamiento del paciente y aceptabilidad de los regímenes de medicación o bien podría dar lugar a errores de medicación. Estas preocupaciones sobre la seguridad del paciente son importantes y se recomienda que los fabricantes de medicamentos genéricos consideren los atributos físicos cuando desarrollen perfiles de productos objetivo de calidad (QTPP) para sus candidatos de productos genéricos.

Consideraciones

Tamaño: los estudios en adultos que evalúan el efecto del tamaño de los comprimidos y las cápsulas sobre la facilidad para tragar sugieren que los aumentos de tamaño están asociados con un aumento de las quejas de los pacientes relacionados con dificultades para tragar en comprimidos de más de aproximadamente 8 mm de diámetro.

El tamaño de la tableta o cápsula influye en el tránsito esofágico, independientemente de los factores del paciente y las técnicas de administración (es decir, uso de líquidos, posición del paciente). En general, se ha demostrado que las tabletas más pequeñas tienen tiempos de tránsito significativamente más rápidos en estos estudios.

Channer y Virjee compararon específicamente el tiempo de tránsito de los comprimidos redondos de 8 mm de diámetro con los comprimidos redondos de 11 mm de diámetro y los comprimidos ovalados de 14 mm x 9 mm y encontraron que los tiempos de tránsito de los comprimidos redondos de 8 mm eran significativamente más cortos que los de los



comprimidos redondos de 11 mm y 14 comprimidos ovalados de 9 mm x 9 mm ($p < 0,02$ y $p < 0,04$, respectivamente). Además, un número significativamente mayor de pacientes notaron que los comprimidos redondos más grandes (> 8 mm) se pegaban en el esófago en comparación con los comprimidos redondos de 8 mm.

Aunque ha habido menos investigación que cuantifique los efectos de la diferencia de tamaño en la fase orofaríngea de la deglución, se cree que el aumento del tamaño de la tableta o cápsula se correlaciona con una mayor dificultad con la transferencia orofaríngea.

Forma: para cualquier tamaño determinado, ciertas formas pueden ser más fáciles de tragar que otras. Los estudios in vitro sugieren que los comprimidos planos tienen una mayor adherencia al esófago que los comprimidos en forma de cápsula. Los estudios en humanos también han sugerido que las tabletas ovaladas pueden ser más fáciles de tragar y tienen tiempos de tránsito esofágico más rápidos que las tabletas redondas del mismo peso. El cumplimiento del paciente con los regímenes de medicación puede verse influido por el tamaño y la forma de una tableta o cápsula. [18]

Recomendaciones

Tamaño: La Agencia recomienda que las tabletas y cápsulas orales genéricas destinadas a tragarse intactas sean de un tamaño similar al RLD correspondiente. La Agencia recomienda limitar las diferencias de tamaño entre tabletas terapéuticamente equivalentes de la siguiente manera:

Si el RLD tiene menos de 17 mm en su dimensión más grande, el producto genérico debe ser:

- No más del 20 por ciento más grande que el RLD en cualquier dimensión única (la dimensión única resultante del genérico no debe exceder los 17 mm).
- No más del 40 por ciento más grande que el volumen del RLD.

Si el RLD es igual o superior a 17 mm en su dimensión más grande, el producto genérico debe ser:

1. No más grande que el RLD en una sola dimensión.
2. No mayor que el volumen del RLD.

Se recomienda que la dimensión más grande de una tableta o cápsula no exceda los mm y que las cápsulas no excedan un tamaño estándar de 00.

Se puede dar flexibilidad adicional para productos de 8 mm o más pequeños en su dimensión más grande, pero se deben hacer esfuerzos para desarrollar tabletas y cápsulas que sean de un tamaño y forma similares al RLD.

Según la convención de tamaño de cápsula estándar, las tolerancias descritas anteriormente generalmente permitirán un aumento de un tamaño de cápsula, cuando la



cápsula de RLD es de tamaño 3 o menor. Cuando la cápsula de RLD es de tamaño 2 o mayor, solo se debe considerar un aumento del tamaño de una cápsula cuando se pueda proporcionar una justificación adecuada para el aumento de tamaño. Estas recomendaciones permitirían aumentar el tamaño de una cápsula cuando el tamaño de la cápsula sea menor que el tamaño de la cápsula 00.

La Agencia reconoce que dos productos farmacéuticos pueden tener diferentes límites superiores de tamaño recomendados, pero el tamaño debe considerarse como parte de un perfil de riesgo / beneficio de un solo producto. Al establecer la equivalencia terapéutica, el solicitante debe comparar su producto genérico solo con el RLD.

Forma: Además de las recomendaciones de tamaño descritas anteriormente, se recomienda fabricar tabletas y cápsulas que tengan una forma similar o que tengan una forma que resulte más fácil de tragar en comparación con la forma del RLD. Evaluar y comparar las áreas transversales más grandes del RLD y el producto genérico es una estrategia para cuantificar los cambios de forma.

Las tabletas y cápsulas que tienen un área de sección transversal más grande (por ejemplo, tabletas que son más redondas) generalmente serían más difíciles de tragar que las tabletas o cápsulas del mismo volumen, pero con áreas de sección transversal más pequeñas.

Existe una variedad de técnicas que pueden usarse para determinar las medidas de volumen de una tableta o cápsula, incluido el uso de picnómetros, o cálculos basados en medidas físicas de la tableta o el troquel utilizado para producir la tableta.

El tamaño de una tableta o cápsula debe proporcionarse en el documento técnico común (formato CTD, sección 3.2.P.1, Descripción y composición del producto farmacéutico de la ANDA. Cualquier estudio y / o información relacionada debe proporcionarse en el CTD sección 5.3.1.2, Informes de estudios comparativos de biodisponibilidad y bioequivalencia.[18]

Transportadores Helicoidales Verticales

Es una manera eficiente de elevar o descender materiales en pendientes pronunciadas o completamente verticales. Estos son de diseño compacto y se pueden adaptar de forma sencilla en espacios reducidos. [19]

Procedimiento de diseño para transportadores helicoidales

- Determinación del Material: Esto para saber las propiedades del material a transportar y con ello saber si afectará de forma negativa a los componentes del transportador.
- Limitación del tamaño del bulto: La relación que existe entre el radio de los grumos y el volumen total del material, dado que puede haber un problema en el espacio libre entre el tornillo y el canal debido a un alto porcentaje de presencia de muestras.

- Capacidad: Manejar la capacidad máxima a la que estará expuesto. Esto usando unidades de volumen y tiempo.
- Capacidad equivalente: Los filetes especiales son menos eficientes en el transporte de material y deben compensarse. Esto se hace multiplicando la capacidad por los factores CF1, CF2 y CF3. La capacidad corregida se denomina capacidad equivalente y se utiliza para determinar el tamaño del tornillo, esta se puede calcular a partir de la siguiente ecuación de la imagen 1. [20]

$$\text{Equivalent Capacity} = \text{Required Capacity}(CF_1)(CF_2)(CF_3)$$

Imagen 1. Ecuación para obtener la capacidad equivalente [20].

Dichos factores se pueden obtener en las siguientes tablas.

Pitch	Description	CF ₁
Standard	Pitch = Diameter	1.00
Short	Pitch = 2/3 Diameter	1.50
Half	Pitch = 1/2 Diameter	2.00
Long	Pitch = 1 1/2 Diameter	0.67

Tabla 1. Paso especial del tornillo transportador Factor CF1.[20]





Type of Flight	Conveyor Loading		
	15%	30%	45%
Standard screw	1.00	1.00	1.00
Cut flight	1.95	1.57	1.43
Cut and folded flight	N.R.	3.75	2.54
Ribbon flight	1.04	1.37	1.62

Tabla 2. Capacidad especial del filete del tornillo transportador Factor CF2. [20]

No. of Paddles per Pitch (Set at 45° Reverse Pitch)					
	None	1	2	3	4
CF ₃	1.00	1.08	1.16	1.24	1.32

Tabla 3. Capacidad especial de la paleta mezcladora del tornillo transportador Factor CF3. [20]

- Tamaño del tornillo: Para obtener el tamaño de tornillo requerido, primero se determina a qué clase pertenece el material usando la primera columna en la tabla 4. Se elige el tornillo de este grupo que tenga la capacidad máxima más grande más cercana a su capacidad requerida. [20]

Material Class Code	Degree of Trough Loading	Screw Dia. (in.)	Maximum RPM*	Capacity, Cubic Feet per Hour	
				At Max. RPM	At 1 RPM
Class I	 45 PERCENT	6	165	368	2.23
A-15		9	155	1,270	8.2
A-25		12	145	2,820	19.4
B-15		14	140	4,370	31.2
B-25		16	130	6,060	46.7
C-15		18	120	8,120	67.6
C-25		20	110	10,300	93.7
		24	100	16,400	164.0
Class II	 30 PERCENT	6	120	180	1.49
A-35 D-35		9	100	545	5.45
A-45 D-45		12	90	1160	12.9
B-35 E-15		14	85	1770	20.8
B-45 E-25		16	80	2500	31.2
C-35 E-35		18	75	3380	45.0
C-45 E-45		20	70	4370	62.5
D-15		24	65	7100	109.0
D-25					
Class III		 15 PERCENT	6	60	90
A-16 C-36	9		55	300	5.45
A-26 C-46	12		50	645	12.9
A-36 D-16	14		50	1040	20.8
A-46 D-26					
B-16 D-36					
B-26 D-46	16		45	1400	31.2
B-36 E-16	18		45	2025	45.0
B-46 E-26	20		40	2500	62.5
C-16 E-36	24		40	4360	109.0
C-26 E-46					
Class IV	 15 PERCENT	6	60	45	0.75
A-17 C-37		9	55	150	2.72
A-27 C-47		12	50	325	6.46
A-37 D-17		14	50	520	10.4
A-47 D-27					
B-17 D-37					
B-27 D-47		16	45	700	15.6
B-37 E-17		18	45	1010	22.5
B-47 E-27		20	40	1250	31.2
C-17 E-37		24	40	2180	54.6
C-27 E-47					

*Maximum recommended RPM.

■ Dark portions represent average level of loading of material for percent indicated.

□ Areas above dark portion represent approximate maximum height of material at carrying side.

Tabla 4. Tamaños y capacidades de transportadores de tornillo sinfin. [20]

- Requerimiento de buje: Un transportador largo requerirá varios bujes intermedios. Si el transportador es corto, los bujes no son requeridos.
- Velocidad de operación: Las revoluciones por minuto requeridas para transportar una capacidad dada se pueden calcular por la siguiente ecuación de la imagen 2.

$$N = \frac{\text{Required Capacity (cu ft/hr)}}{\text{cu ft/hr at 1 RPM}}$$

Imagen 2. Ecuación para calcular los rpm requeridos para una carga. [20]

Donde:

N = revolución por minuto.

La capacidad a 1 rpm se obtiene de la tabla 4. Si un transportador no requiere sujetadores y no transporta materiales abrasivos, puede funcionar a velocidades más altas. Las velocidades máximas se establecen para evitar el ruido y el desgaste excesivos de los bujes.

- **Requerimiento de caballos de potencia:** Para determinar la potencia requerida para transportar un sólido, primero se determinan una serie de factores empíricos para el material y los componentes del transportador. Se busca el material transportado, se encuentra el número de grupo de componentes y su factor de potencia del material, F_m .

El número de grupo de componentes se utiliza para realizar la selección preliminar del rodamiento. Teniendo el número de grupo de componentes, se busca el tipo de rodamiento en la tabla 5, donde también se obtendrá el factor de rodamiento, F_b . Este factor se deriva empíricamente y está relacionado con la fricción en el buje. El factor de material, F_m , es un factor enteramente empírico determinado por la experiencia. No tiene relación medible con ninguna propiedad física del material transportado.

Se obtiene el factor de diámetro del filete, F_d , de la tabla 6. Este factor ha sido calculado de forma proporcional al peso medio por pie de las piezas giratorias más pesadas. [20]

Component Group	Bearing Type	F_b
Group A	Ball	1.0
	Babbitt	
Group B	Bronze	1.7
	Graphite bronze*	
	Canvas base phenolic*	
	Oil-impregnated bronze*	
	Oil-impregnated wood*	
Group C	Plastic*	2.0
	Nylon*	
	Teflon*	
Group D	Chilled, hardened iron*	
	Hardened alloy sleeve*	

*Nonlubricated bearings or bearings not additionally lubricated.

Tabla 5. Factor de rodamiento, F_b . [20]

Screw Diameter (in.)	F_d	Screw Diameter (in.)	F_d
4	12.0	14	78.0
6	18.0	16	106.0
9	31.0	18	135.0
10	37.0	20	165.0
12	55.0	24	235.0

Tabla 6. Factor de diámetro del filete, F_d . [20]

El factor del filete, F_f , y el factor de álabe, F_p , se pueden obtener de las tablas 7 y 8, respectivamente. Son factores de corrección para las distintas formas de filete del tornillo transportador. Se derivan empíricamente, pero tiene relación con las áreas netas efectivas del filete del tornillo.

La potencia requerida para transportar un sólido dado es el total de la potencia para superar la fricción, HP_f , y la potencia para transportar la capacidad de material especificada, HP_m . Luego, el total se corrige con un factor de sobrecarga para unidades de poca potencia y se divide por un factor de eficiencia:

$$HP_f = \frac{LNF_d F_b}{1,000,000}$$

Imagen 3. Ecuación de la potencia para superar la fricción. [20]

$$HP_m = \frac{CLWF_f F_m F_p}{1,000,000}$$

Imagen 4. Ecuación de la potencia para transportar la capacidad de material especificada. [20]

$$\text{Total HP} = \frac{(HP_f + HP_m)F_o}{e}$$

Imagen 5. Ecuación de la potencia requerida para transportar un sólido dado. [20]

Type of Flight	Conveyor Loading			
	15%	30%	45%	95%
Standard	1.0	1.0	1.0	1.0
Cut flight	1.10	1.15	1.20	1.3
Cut and folded flight	N.R.*	1.50	1.70	2.20
Ribbon flight	1.05	1.14	1.20	—

*Not recommended.

Tabla 7. Factor de filete, F_f . [20]

F_p	No. of Paddles per Pitch Set at 45° Reverse Pitch				
	None	1	2	3	4
	1.00	1.29	1.58	1.87	2.16

Tabla 8. Factor de álabe, F_p . [20]



$HP_f + HP_m$	F_o
< 1	2
$\geq 1 < 2$	1.5
$\geq 2 < 4$	1.25
$\geq 4 < 5$	1.1
≥ 5	1

Tabla 9. Factor de sobrecarga, F_o . [20]

Donde:

C = capacidad (cu ft/hr)

e = factor de eficiencia

Fb = Factor de rodamiento (Tabla 5)

Fd = Factor de diámetro del filete (Tabla 6).

Ff = Factor de filete (Tabla 7)

Fm = Factor del material

Fo = Factor de sobrecarga (Tabla 9)

Fp = Factor de álabe (Tabla 8)

L = longitud total del transportador (ft)

N = velocidad de operación (RPM)

W = densidad aparente del material transportado (lb/cu ft)

El denominador en las ecuaciones de las imágenes 2 y 3, se ha cambiado a 1,000,000 por conveniencia; los factores utilizados en las ecuaciones se han ajustado para compensar.

El factor de sobrecarga, F_o , es una corrección para potencias calculadas de menos de 5 HP. Este factor es necesario porque los transportadores de tornillo sinfín a menudo requieren un rango de par mayor que el que pueden proporcionar los motores pequeños. Pequeñas sobrecargas y condiciones de estrangulamiento menores podrían detener fácilmente las unidades más pequeñas. La Tabla 9 enumera el valor de F_o apropiado basado en la suma de HP_f y HP_m .

Varios equipos de reducción de velocidad tienen una pérdida por fricción interna que puede corregirse utilizando un factor de eficiencia, e. Para la selección preliminar, se usa 0.88, que es la pérdida de eficiencia para un accionamiento de transportador de tornillo montado en eje.

Arrancar un transportador completamente cargado puede causar una sobrecarga del motor. Los motores suelen estar sobredimensionados para evitar este tipo de situaciones. También se debe considerar la característica del material transportado.

Todos los elementos de transmisión de potencia de un transportador de tornillo deben tener el tamaño adecuado para soportar el par motor completo. Si, por ejemplo, un transportador de tornillo requiere un motor de 4.5 HP según lo

determinado por cálculo y se utiliza un motor de 5 HP, todos los elementos transmisores de potencia deben ser capaces de manejar los 5 HP completos. [20]

- Deflexión admisible del eje: A menudo, se utilizan longitudes de tornillo más grandes que el estándar, ya sea para eliminar los bujes o para compensar las extrañas longitudes de transporte. La deflexión de las longitudes estándar de los tornillos no suele ser un problema; las tolerancias permitidas están predeterminadas. Para secciones más largas que el estándar, la deflexión debe determinarse utilizando la fórmula de la viga:

$$\Delta = \frac{5WL^3}{384EI}$$

Imagen 6. Ecuación de la deflexión para secciones más largas que el estándar. [20]

Donde:

Δ = deflexión en la mitad del tramo (in).

W = peso total del tornillo ensamblado (lb).

L = longitud del tornillo entre cojinetes (in).

E = módulo de elasticidad del acero (psi).

I = momento de inercia del eje de tubo hueco o macizo (in⁴).

La deflexión no debe exceder 0.25 in. Si la deflexión excede este valor, se pueden producir daños en el canal y los cojinetes, y el sello del eje no será efectivo.

Si la deflexión es superior a 0.25 in. el eje debe actualizarse a un tamaño mayor.

- Clasificaciones de torsión de componentes: Una vez determinada la potencia, se puede calcular el par al que estarán sujetos los componentes:

$$T = \frac{HP \times 63025}{N}$$

Imagen 7. Ecuación de par al que estarán sujetos los componentes. [20]

Donde:

T = Torque (lb)

HP = Potencia calculada

N = velocidad (RPM)

Los transportadores de tornillo sinfín están limitados en longitud total por la cantidad de torque que se puede transmitir de manera segura a través de sus componentes. Los pernos, los acoplamientos y las tuberías que se utilizan en los transportadores de tornillo tienen límites de torsión que deben tenerse en cuenta cuando se diseña el transportador. Los componentes están sujetos a las siguientes tensiones:

1. esfuerzo cortante en los pernos
2. aplastamiento o carga de tensión en los pernos
3. tensión de torsión en la tubería

4. tensión de torsión en el eje de acoplamiento en áreas reducidas por los orificios de los pernos

Se enumera la clasificación de torsión de cada tamaño de componente. El componente con la calificación más baja para un lote de tamaño determinado será el que gobierne la cantidad de caballos de fuerza que se pueden transmitir de manera segura. Para los de 2 in. tubería, el acoplamiento estándar es el factor limitante a 3070 in-lb. Si este valor es demasiado bajo, el acoplamiento estándar se puede sustituir por uno endurecido. Si los dos pernos estándar son el factor limitante, se pueden usar tres. El grosor de la pared de la tubería y el buje también se puede aumentar para requisitos más altos. [20]

Medio paso, helicoidal sencillo

El paso se reduce a 1/2 del diámetro. Este tipo de paso se recomienda para aplicaciones verticales o inclinadas, ya que también se utilizan en alimentadores helicoidales y para poder manejar materiales altamente fluidos. [21]

Flujo de fluido debido a una carga descendente

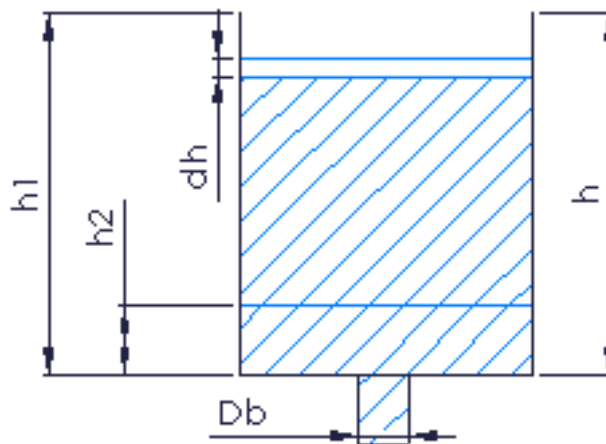


Imagen 8. Identificación de los parámetros a tomar en cuenta en el análisis de un fluido descendente. [22]

Para una altura h de fluido como se muestra en la imagen 8, la velocidad de salida v_b en la boquilla es:

$$v_b = \sqrt{2gh}$$

La rapidez de flujo de volumen es el caudal, el cual es:

$$Q = A_b v_b dt$$

El volumen que fluye a través de la boquilla es:

$$\forall = Q dt = A_b v_b dt$$

El volumen que fluye a través de la boquilla es igual al volumen extraído del tanque:

$$\forall = -A_t dh$$

Ambos volúmenes son iguales:

$$\forall = A_b v_b dt = -A_t dh$$

$$dt = \frac{A_t}{A_b} \frac{dh}{v_b} = - \frac{A_t}{A_b} \frac{dh}{\sqrt{2gh}}$$
$$\int_{t_1}^{t_2} dt = - \frac{A_t}{A_b} \frac{1}{\sqrt{2g}} \int_{h_1}^{h_2} h^{-\frac{1}{2}} dh$$
$$t_2 - t_1 = \frac{A_t}{A_b} \frac{2}{\sqrt{2g}} \left(h_1^{\frac{1}{2}} - h_2^{\frac{1}{2}} \right)$$

Por lo que, se obtiene el tiempo requerido para vaciar un tanque. [22]

Coeficiente de resistencia K para pérdidas de flujo de volumen de fluido en un componente hidráulico.

$$\Delta p = K * \frac{\rho}{2} * \frac{Q^2}{A^2} \quad (1)$$

$$v = \frac{Q}{A} \quad (2)$$

Donde:

Δp = Pérdida de presión del componente

K = Coeficiente de resistencia (Factor de fricción) sin dimensiones del componente.

ρ = Densidad

v = Velocidad media en la sección transversal característica

Q = Flujo de volumen

A = Sección transversal característica

d = Diámetro interior de la sección transversal característica. Para válvulas en general es diámetro nominal DN.

La ecuación (1) es una estimación general para la pérdida de carga de tuberías y componentes en analogía con la fórmula Darcy-Weisbach.

El coeficiente de resistencia se supone constante para el rango de caudal considerado aquí. Esta hipótesis es correcta para un flujo turbulento en el componente. En otros muchos casos es aceptable esta inexactitud. Con K=const., la ecuación (1) puede escribir como sigue:

$$\frac{\Delta p}{\Delta p_0} = \frac{\rho}{\rho_0} * \frac{Q^2}{Q_0^2} \quad (3)$$

$$Q = Q_0 \sqrt{\frac{\Delta p}{\Delta p_0} * \frac{\rho_0}{\rho}} \quad (4)$$

El fabricante de válvulas Mason-Neilan (MA, USA) introdujo en los años 40 el coeficiente de caudal Cv. En los años 50 se representó como Kv en el sistema métrico. Estos coeficientes determinan el punto de referencia ($Q_0, \Delta p_0, \rho_0$) de una válvula. Se selecciona el agua como medio de referencia.

Donde:

Kv: $Q = Kv[m^3/h]$ @ $\Delta p_0 = 1 [bar]$

Después de resolver la ecuación (1) para K y fijado $A=\pi/4d^2$, se obtiene con la siguiente ecuación la relación entre el coeficiente de resistencia K y el coeficiente de caudal Kv. [23]

$$K = \frac{(d [m])^4}{\left(Kv\left[\frac{m^3}{h}\right]\right)^2} (0.001599)$$

Control automático

En la actualidad, los sistemas de control son una de las aplicaciones de la ingeniería más propagadas en la industria y han tenido un gran impacto en el avance de la ingeniería. El control automático hoy en día forma parte integral de los procesos aeroespaciales, aeronáuticos, automotrices, de robótica industrial, entre otras ramas.

Los controladores, son un conjunto operaciones que permiten manipular o regular una o más características de operación de un sistema mediante un esquema de control que puede tratarse de un lazo de control cerrado o abierto.

Los lazos de control cerrado, de acuerdo a Ogata (2010), son aquellos que mantienen una relación determinada entre la salida y la entrada de referencia del sistema, comparándolas y usando la diferencia como medio de control que ingresa a una Ley de Control la cual se encarga de corregir dicha diferencia o error y llevar a la salida que se está controlando del sistema al valor de referencia deseado. Las imágenes 9 y 10, muestran el esquema de un lazo de control cerrado o retroalimentado.



Imagen 9. Lazo Cerrado de Control. [24]

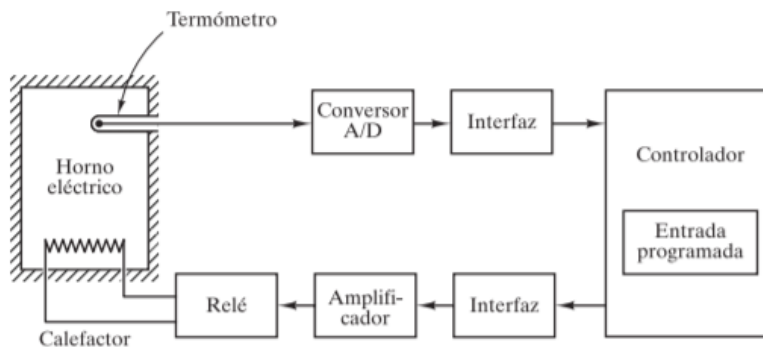


Imagen 10. Ejemplo Lazo Cerrado de Control de Temperatura [25]

Existe una diversidad de controladores que permiten establecer un control sobre una o más salidas de un sistema, normalmente su selección y diseño depende de la complejidad del problema a tratar y del sistema a controlar. En el caso particular de este proyecto, el problema consiste en el control de la temperatura interna del dispensador de medicamento. Lo que se busca realizar es que mediante un proceso de control se pueda regular la temperatura dentro del dispositivo de tal forma que se pueda mantener por debajo de los 30°C ya que, como se mencionó en el marco teórico, los medicamentos no pueden permanecer expuestos a temperaturas superiores a ese valor por mucho tiempo si se desea que se conserven adecuadamente y no se alteren sus propiedades.

Controlador ON-OFF

Es una de las formas más elementales, por no decir la más elemental, del control clásico. Como su nombre lo indica, estos controladores ejercen una acción de tipo Todo o Nada sobre el actuador o conjunto de actuadores que se busca controlar a partir de la información de retroalimentación que se recibe de los sensores del sistema, es decir, se implementan bajo un esquema de control de lazo cerrado. Un caso de ejemplificación es que, si se tiene una diferencia de un determinado valor “n” entre señal de referencia y la señal de salida del sistema, entonces se enciende (o apaga según sea requerido) uno o más actuadores hasta que la diferencia sea nula. [26]

En la imagen 11, se muestran ejemplos de las gráficas de las señales de salida de un controlador P, PID y un controlador ON-OFF para ilustrar de forma más clara su acción de control sobre la salida de un sistema.

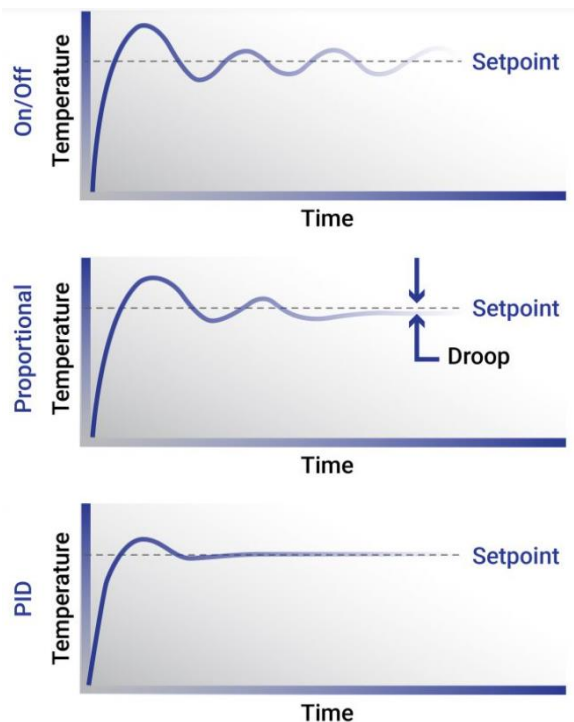


Imagen 11. Acciones de Control de los controladores ON-OFF, P y PID [27].



También existe la posibilidad de reacondicionar los controladores ON-OFF para que no necesariamente se tenga una acción todo o nada sobre el actuador a controlar, sino que se tenga acciones que, dependiendo de si se presentan un cierto valor o rango de valores de interés en la diferencia entre la señal de salida y referencia del sistema, se haga funcionar a un actuador a un porcentaje específico de su capacidad, lo cual, normalmente, se logra mediante la implementación de un PWM como señal de control, la cual posteriormente deberá tener una etapa de potencia o acoplamiento.

Bases de Datos en la Nube

Como su nombre lo indica, son bases de datos alojadas y ejecutadas en los servidores de un tercero, es decir, de un prestador de servicios. Estas, pueden realizar las mismas operaciones que una base de datos normal, con la ventaja de que todas las operaciones y computación se realizan en la nube, aligerando las operaciones que deben realizarse de manera local y, desde luego, eliminando o minimizando el desarrollo de “back end” de la base de datos.

Existen diferentes tipos de servicios en la actualidad, sin embargo, todas comparten ciertas características:

- Son un servicio que se crea y al cual se accede en la nube
- Permite el alojamiento de bases de dato sin comprobación de requerimientos de hardware
- Puede ser gestionado por el usuario o por el proveedor a manera de servicio
- Se accede por medio de una interfaz web o una API proporcionada por el proveedor
- Llegan a soportar bases de datos SQL, NoSQL e incluso MySQL
- Permiten la posibilidad de tener bases de datos en tiempo real con comunicación multiplataforma. [28]

Uno de los mayores proveedores de servicios en la nube en la actualidad es Google, quien cuanta con Google Firebase y Google Cloud, siendo ambos servicios de desarrollo, analítica, bases de datos y almacenamiento en la nube.

Google Firebase Realtime Database

Es el servicio de base de datos en tiempo real de Google Firebase, la cual, funciona bajo un formato de almacenamiento de datos JSON, es decir, un árbol de nodos y sub nodos. Estas bases de datos son accesibles desde múltiples plataformas como Android, IOS y JavaScript por medio de la implementación de SDKs para la vinculación de las mismas. Dado a que os usuarios están conectados por medio de clientes en línea, todos pueden acceder a los datos más recientes en la base de datos o realizar modificaciones. Además, es posible acceder a los datos de manera offline ya que el SDK de esta plataforma guarda

los datos de forma persistente en el disco local y, al reconectarse, de forma automática el cliente actualiza los datos. [29]

Material desecante

Un material desecante es aquel que, debido a su composición química, tiene la capacidad de absorber las moléculas de agua del aire, es decir, se comporta como un agente deshidratante que absorbe la humedad del ambiente y la mantiene consigo. Los más comunes son la arcilla, el gel de silicio o silica y el tamiz molecular.

Estos materiales se implementan en múltiples industrias debido a las cualidades antes mencionadas, pues calculando la cantidad adecuada de material, es posible asegurar un ambiente seco o con bajos porcentajes de humedad relativa para la conservación de diversos productos. Para evitar el contacto del material con productos que requieran mantenerse libres de agentes químicos o contaminantes debido a su uso o forma de consumo, los desecantes son empaquetados con envoltorios seguros aprobados por la Administración de Alimentos y Drogas de Estados Unidos (FDA por sus siglas en inglés).

Para la selección del material desecante, los expertos sugieren tomar en cuenta las condiciones de humedad relativa y temperatura del espacio en que se aplicará, y realizar una comparación con respecto a los datos de absorción que se presentan en las gráficas de la imagen 12. [30]

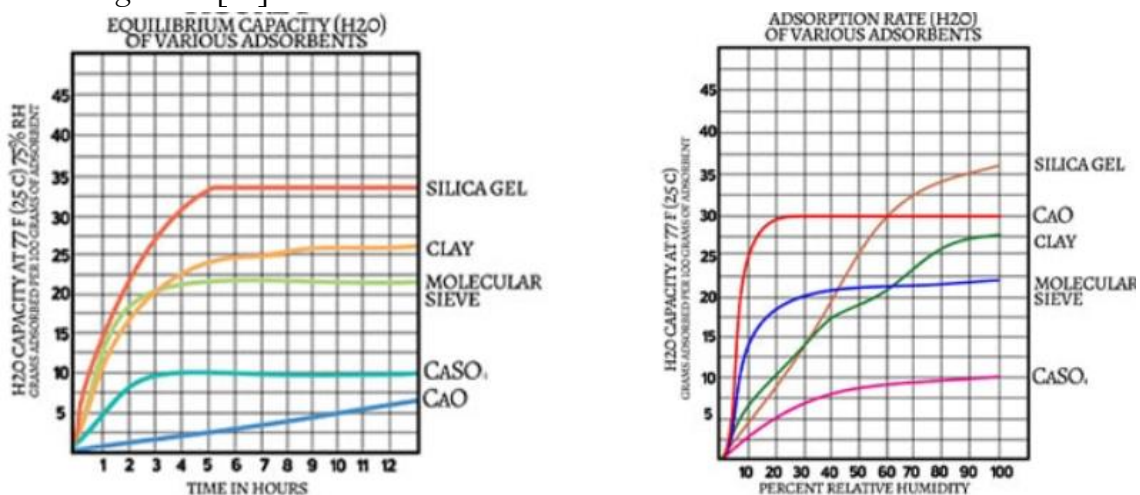


Imagen 12. Gráficas de capacidad de equilibrio (H₂O) y tasa de absorción (H₂O). [30]

En la gráfica del lado izquierdo de la imagen 12, podemos observar la capacidad de equilibrio de los desecantes más comunes a temperatura ambiente y 75% de humedad relativa, en ella, se grafican gramos de agua absorbidos por cada 100g de desecante contra el tiempo en horas. En la gráfica del lado derecho se muestran las tasas de absorción de los desecante, en esta, se grafican los gramos de agua absorbidos por cada 100g de desecante contra el porcentaje de humedad.



Una vez se selecciona el material desecante, se debe calcular la cantidad de material, dependiendo de la aplicación, es posible determinar la cantidad en gramos por medio de métodos termodinámicos complejos como el método CNEA o la Norma MIL que se emplean para espacios que contendrán equipo electrónico en condiciones ambientales adversas de humedad relativa alta y buscan evitar la corrosión de los componentes. Sin embargo, si la aplicación no requiere tanto nivel de precisión, se puede emplear un cálculo simplificado para contenedores de almacenamiento rígidos como una aproximación válida.

El cálculo simplificado de desecante para contenedores rígidos se realiza mediante la expresión:

$$\text{Unidades de Desecante} = K * V$$

Donde $K = 0.161 \frac{1}{gal}$ ó $0.0007 \frac{1}{in^3}$ ó $1.2 \frac{1}{ft^3}$ y $V = \text{Volumen del contenedor}$

Este cálculo basado en la norma DIN 55474 permite conocer cuántas unidades DIN de desecante se requieren para mantener el volumen del contenedor rígido seco. De acuerdo a la norma DIN, una unidad equivale a la absorción aproximada de 6g de agua, que, a su vez, equivale a aproximadamente 30g de desecante. [31]



2. Metodología

El modelo V es un modelo de desarrollo de sistemas diseñado para simplificar la comprensión de la complejidad asociada con el desarrollo de sistemas. El modelo V se basa en el ciclo del proyecto, que se muestra de izquierda a derecha para representar el tiempo y la madurez del proyecto. Junto con esta representación está el reconocimiento de los niveles de descomposición, que se ilustran, en la dimensión vertical de arriba a abajo. El usuario está en el nivel más alto y las partes y líneas de código en el más bajo. El plano ortogonal al plano de la V ilustra el número de entidades en cada nivel de descomposición, lo que se relaciona con la complejidad del sistema.

El concepto de una línea de base en evolución, aumentando progresivamente en profundidad y bajo control de cambios, crea la forma de V, que se llama el núcleo de la V. La pierna izquierda de la V representa la descomposición y la definición y la pierna derecha representa la integración y la verificación.[31]

La verificación y validación de los requisitos son una parte fundamental de la ingeniería de sistemas y software. La importancia de la verificación y validación (especialmente las pruebas) es una de las principales razones por las que el ciclo de desarrollo en cascada tradicional se sometió a una modificación menor para crear el modelo V que vincula las actividades de desarrollo temprano con sus correspondientes actividades de prueba posteriores.

El modelo V tiene ventajas y desventajas. En el lado positivo, representa claramente las actividades de ingeniería primarias en un flujo lógico que es fácilmente comprensible y equilibra las actividades de desarrollo con sus actividades de prueba correspondientes. Por otro lado, el modelo V es una simplificación excesiva en la que estas actividades se ilustran cómo fases secuenciales en lugar de actividades que típicamente ocurren de manera incremental, iterativa y concurrente, especialmente en proyectos que utilizan enfoques de desarrollo evolutivo (ágil).

Los desarrolladores de software pueden disminuir el impacto de esta limitación de fase secuencial si consideran que el desarrollo consta de muchas V de corta duración en lugar de una pequeña cantidad de V grandes, una por cada incremento iterativo simultáneo. Sin embargo, cuando los programadores aplican un modelo V al desarrollo ágil de un sistema grande y complejo, se encuentran con algunas complicaciones potenciales que requieren más que una simple colección de modelos V pequeños, incluidos los siguientes:

- Los requisitos arquitectónicamente significativos y la arquitectura asociada deben diseñarse y estabilizarse tan rápidamente como sea posible. Todos los incrementos posteriores dependen de la arquitectura, que se vuelve difícil y costosa de modificar después de que los incrementos iniciales se hayan basado en ella.
- Múltiples equipos ágiles multifuncionales trabajarán en diferentes componentes y subsistemas simultáneamente, por lo que sus incrementos deben coordinarse

entre los equipos para producir componentes y subsistemas consistentes y probables que se pueden integrar y lanzar.

Otro problema con el modelo V es que la distinción entre prueba de unidad, integración y sistema no es tan clara como implica el modelo. [32]

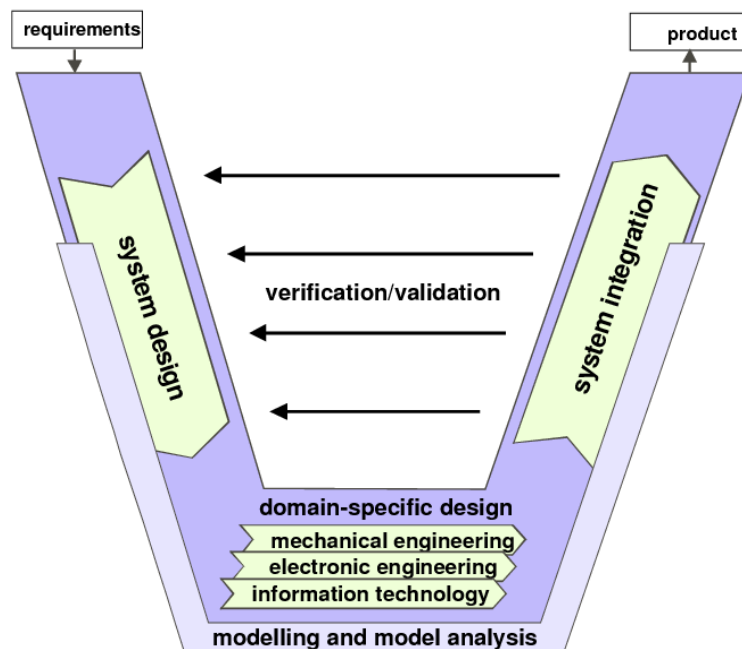


Imagen 13. Modelo V para el desarrollo de sistemas mecatrónicos. [33]

La metodología implementada para el desarrollo del prototipo fue el llamado Modelo V que se describe en la imagen 13. En esta se pueden ver los pasos a seguir para obtener un producto a partir de una lista de requerimientos, como el diseño del sistema, el modelamiento y análisis y la integración del sistema.

La estructura básica de un sistema mecatrónico propuesta consta de: fuente, actuadores, sistema base, sensores, procesamiento de la información, ambiente y usuario.

Usando de esta estructura, se hace referencia a los pasos a seguir para el desarrollo del dispositivo dispensador, el cual se muestra en la imagen 14.

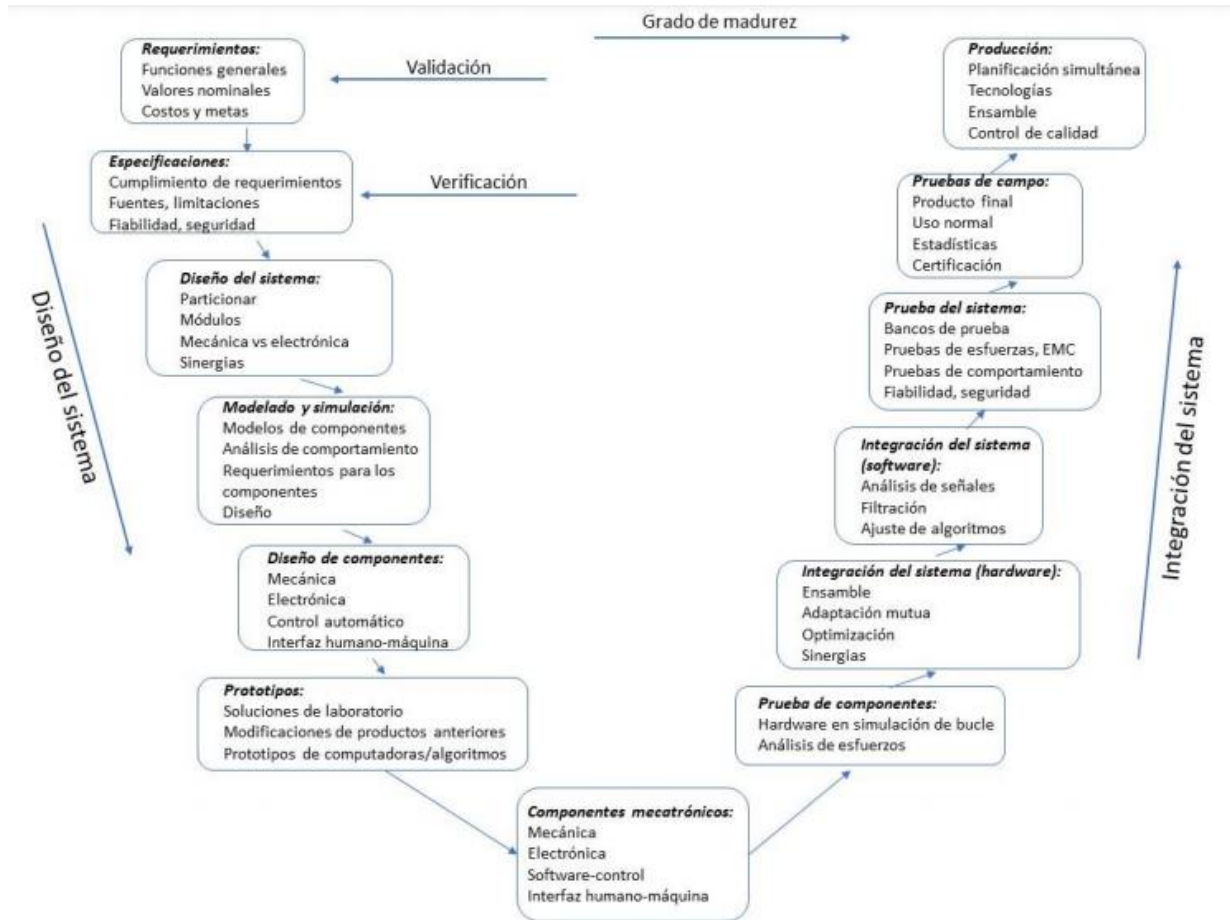


Imagen 14. Modelo V para el desarrollo del dispensador.



3. Diseño del Sistema

Es necesario, para el desarrollo de este y cualquier proyecto, conocer cuáles son las necesidades y requerimientos que implica el desarrollo del sistema, ya que, al final, estas permiten establecer tanto las delimitaciones que se fijaran, como cuáles son las problemáticas que el aparato o producto debe atender para dar solución a un problema.

En nuestro caso, es imprescindible conocer cuáles son las necesidades y requerimientos para que un sistema cuyo fin es el de llevar el tratamiento médico de un paciente con algún padecimiento ya sea crónico o temporal, pueda satisfacerse adecuadamente.

Requerimientos y especificaciones del sistema

Esta actividad supone como tarea a resolver la búsqueda y definición de los requerimientos que debe cumplir el proyecto de acuerdo con las necesidades que éste mismo busca cubrir a través de un análisis.

Por otra parte, parte del problema a resolver es plantear de acuerdo con los requerimientos, la serie de especificaciones de diseño que el aparato deberá seguir para poder satisfacer y cubrir las necesidades debido a las cuales se ha decidido desarrollarlo. Esto, mediante la creación de una tabla de especificaciones de diseño también conocida como PDS.

Análisis de Requerimientos

Se pretende diseñar un prototipo de dispositivo dispensador de medicamentos el cual deberá ser posible configurar y administrar desde una aplicación móvil por medio de una conexión Wi-Fi. Deberá tener un sistema de alarma integrado el cual pueda ajustar horarios de ingesta de medicamentos, y tenga la capacidad de almacenar, gestionar y dispensar tanto medicamentos como agua potable para los usuarios.

Deberá contar con compartimentos para 3 tipos de medicamentos de diferente forma y tamaño. Los contenedores para medicamentos deberán tener la capacidad de almacenar al menos 15 cápsulas o comprimidos. Deberá contar también con un contenedor de agua potable de al menos 1 l.

A su vez, debe conservar una temperatura menor a 30°C dentro del dispositivo para mantener en condiciones necesarias los comprimidos y cápsulas.

El dispositivo deberá tener la capacidad de organizar las pastillas de forma autónoma, por lo que el usuario solamente deberá ingresarlas para que el aparato reconozca qué tipo de medicamento es y lo almacene en un contenedor adecuado.

Otra cualidad requerida es que el aparato cuente con un módulo que permita al usuario llevarse los medicamentos que va a consumir a lo largo de uno a tres días en un pastillero portátil.



Para funcionar, el dispositivo deberá conectarse a un enchufe de pared de red eléctrica doméstica con parámetros de funcionamiento: 120-127 Vca - 60Hz.

PDS

A continuación, se presentan las principales especificaciones de diseño con las que el prototipo deberá cumplir, acorde al previo análisis de requerimientos, esto se muestra en la siguiente tabla 10.

No.	Obligatorio (D) Opcional (W)	Descripción	Requerimiento
1	D	El aparato debe contar con al menos 3 contenedores apropiados para 3 diferentes pastillas.	3 contenedores: - 1 para comprimidos de geometría redonda con diámetros de 6 a 10 mm. - 1 para tabletas de geometría elíptica (ovalada) con un ancho de 4 a 6 mm y un largo máximo de 17 mm. - 1 para cápsulas con un ancho de 3 a 5 mm y un largo máximo de 17 mm.
2	D	El dispositivo debe tener la capacidad de conectarse a una red WiFi y comunicarse a través de ella con un teléfono inteligente.	Conexión a red WLAN con especificaciones de comunicación Wi-Fi para una banda de transmisión de 2.4GHz
3	D	El dispositivo debe contar con un contenedor capaz de almacenar agua potable.	1 litro de agua.
4	D	El aparato debe medir la temperatura y humedad relativa interna.	Sensores de temperatura y humedad.
5	D	El dispositivo debe llevar el conteo de las pastillas disponibles en los contenedores.	Sensor óptico de barrera.
6	W	Los contenedores de los medicamentos deben estar fabricados de un material de grado farmacéutico.	Material: Cloruro de Polivinilo (PVC)



7	D	El dispositivo deberá contar con un sistema dispensador de agua.	Mangueras y un actuador electromecánico ya sea electroválvula o bomba.
8	D	El aparato debe contar con un sistema de reconocimiento para los medicamentos que emplee visión artificial.	Reconocimiento de geometría y tamaño. Cámara web comercial.
9	D	El dispositivo deberá implementar un sistema de alarma para el usuario.	Alarma sonora (bocina o buzzer) y visual (led) y botón de paro.
10	D	Deberá desarrollarse una aplicación móvil que permita configurar y administrar el dispositivo a través de una interfaz de usuario.	Aplicación desarrollada en un lenguaje basado en Java o compatible con Android.
11	W	La interfaz de usuario de la aplicación móvil debe ser atractiva para el usuario.	Animaciones de transición. Colores amigables con la vista. Mensajes de ayuda. Menús de configuración. Ventana de información general.
12	D	El dispositivo deberá implementar un esquema de control que permita regular la temperatura interna y mantenerla menor a los 30°C.	Actuador: Ventilador de C.D.
13	D	El dispositivo debe dispensar los medicamentos que el usuario requiera en el horario correspondiente.	Sistema electromecánico con sincronización de horario.
14	D	Deberá de implementarse al menos una unidad de procesamiento de datos (CPU), la cual controlará todos los sistemas del dispositivo	Uso de una o más tarjetas embebidas o miniPC con capacidad de procesamiento de datos. (Raspberry, Núcleo o similares)
16	D	El dispositivo deberá contar con una fuente de poder que pueda conectarse a una red eléctrica doméstica.	Parámetros de entrada: 127V-60Hz
18	D	El aparato debe contar con recipientes propios para dispensar el agua y los medicamentos del usuario.	- 1 vaso con capacidad de 100 ml para agua. - 1 contenedor con capacidad para 5 pastillas



19	D	El dispositivo debe contar con una modalidad para que el usuario pueda almacenar pastillas que requiere ingerir a lo largo de uno a tres días.	- Pastillero portátil con 3 espacios de almacenamiento. Cada compartimento debe almacenar hasta 10 pastillas.
20	D	Debe de proporcionarle al usuario la posibilidad de que pueda solicitar al dispositivo las pastillas que necesita a lo largo de uno a tres días consecutivos en caso de viaje.	Aplicación móvil ofrece selección de modo de viaje. El dispositivo entrega medicamentos necesarios de acuerdo con días ausente del usuario.
21	W	El sistema de reconocimiento debe tener la capacidad de detectar si se ha ingresado más de un tipo de medicamento y evitar mezclarlos.	Organizar los medicamentos de acuerdo con la geometría y tamaño en su respectivo compartimento. El sistema cuenta con acciones correctivas en caso de que se ingrese más de un tipo de medicamento.
22	D	La programación del dispositivo deberá llevar un conteo de los medicamentos disponibles y advertir cuando se deba abastecer.	Advertencia de reabastecimiento cuando solo quedan tres dosis.
23	D	La programación del aparato deberá tener la capacidad de determinar cuándo le corresponde al usuario ingerir un medicamento, proporcionarle la dosis correspondiente, así como agua para ingerirla. En caso de omisión o retraso, el dispositivo deberá ser capaz de autodeterminar nuevos horarios de ingesta del medicamento.	Código fuente robusto y solvente de acuerdo con las necesidades descritas
24	D	La aplicación móvil deberá tener la capacidad de enviar y recibir datos del dispensador para administrarlo y gestionarlo, mostrar datos de operación del mismo y activar notificaciones al usuario como recordatorio de ingesta de medicamentos.	Comunicación con el dispensador a través de una red WLAN con especificaciones de comunicación Wi-Fi para una banda de transmisión de 2.4GHz.
25	D	Los contenedores de medicamentos deben tener la capacidad de almacenar una cantidad mínima de pastillas.	Cantidad mínima: 15 por contenedor.

			Cantidad máxima: 20 por contenedor
26	W	Volumen aproximado del dispensador.	0.343 metros cúbicos.

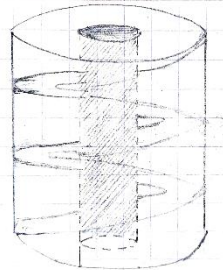
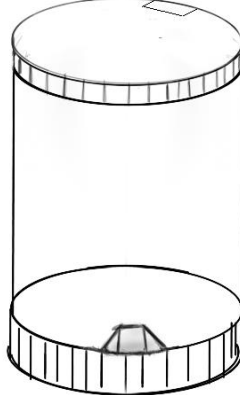

Tabla 10. Especificaciones de diseño del producto.

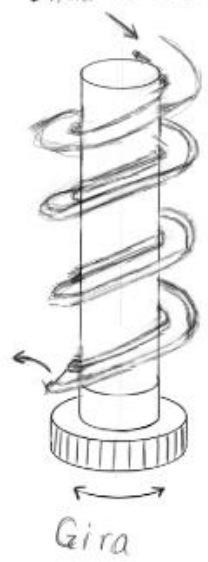
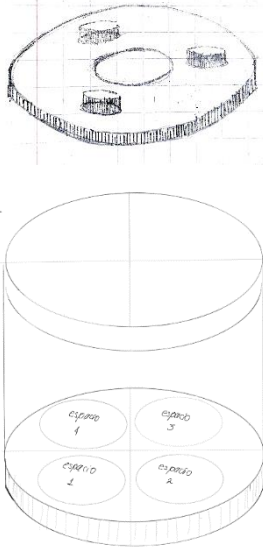
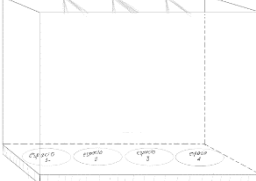
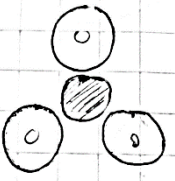
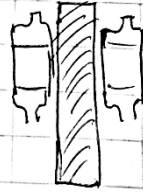
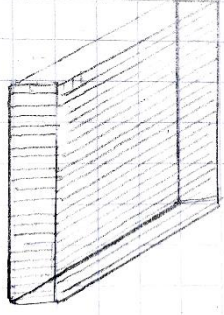

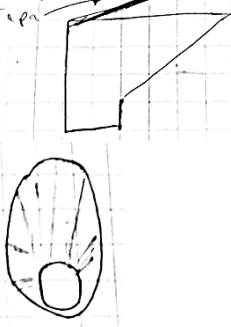
Diseño Conceptual

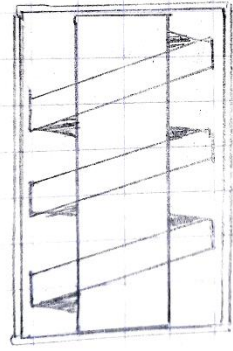
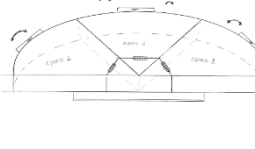
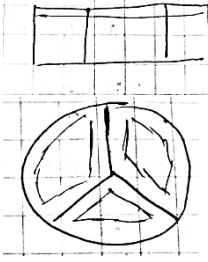
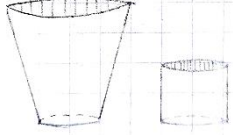
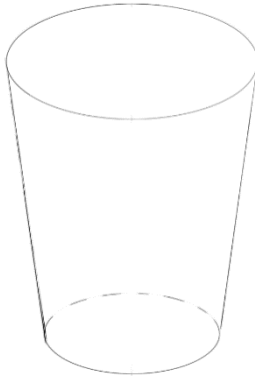
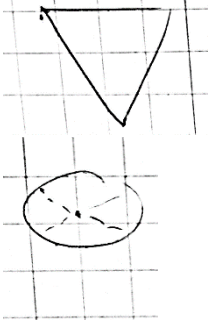
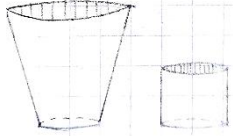
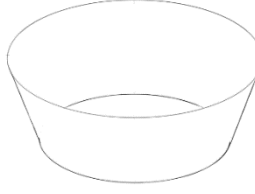

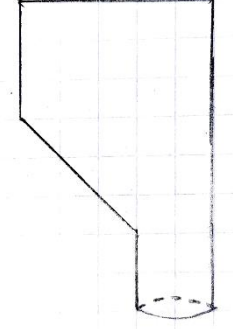

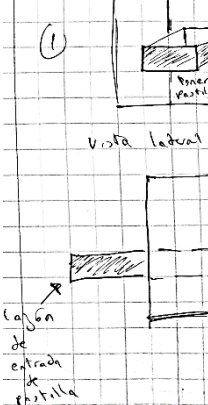
Supone como principal problema a resolver el proponer un primer diseño (concepto) para el dispositivo que se ajuste a las necesidades, requerimientos, especificaciones de diseño y que además pueda ofrecer una perspectiva de cómo podría lucir el producto final, esto a partir de una matriz morfológica que se puede visualizar en la tabla 11. Este problema, tiene una serie de subactividades que también generan sus retos, dichas tareas son:

- Definición de los atributos y parámetros para evaluar los diseños conceptuales, así como la construcción de una matriz binaria con los pesos de los parámetros.
- Realizar propuestas de diseño para algunas de las partes más relevantes del dispositivo.
- Construcción de una matriz morfológica con las propuestas de diseño
- Seleccionar las posibles rutas solución de la matriz morfológica
- Evaluar las rutas solución empleando la matriz binaria de parámetros y atributos y seleccionar la más adecuada.
- Proponer un primer diseño (modelo) general del aparato.

Matriz morfológica

Parte/Pieza	Opción 1	Opción 2	Opción 3
Compartimento de Medicamentos [A]			

		<p style="text-align: center;">Entra Pastilla</p>  <p style="text-align: center;">Gira</p>	
<p>Disposición de Compartimentos de Medicamento</p> <p>[B]</p>			<p style="text-align: center;">plano. alzado</p>  <p style="text-align: center;">plano lateral</p> 
<p>Compartimento de agua</p> <p>[C]</p>			<p style="text-align: center;">Tapa</p> 

<p>Organizador de Pastillas Portátil</p> <p>[D]</p>			
<p>Recipiente para tomar agua</p> <p>[E]</p>			
<p>Recipiente para tomar medicamento</p> <p>[F]</p>			
<p>Entrada de Medicamentos</p> <p>[G]</p>			


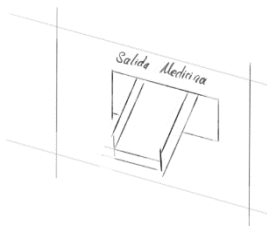

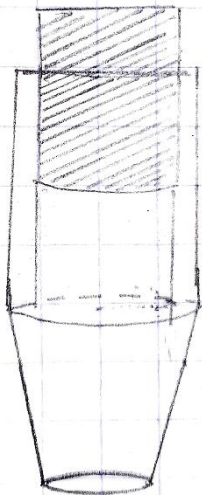
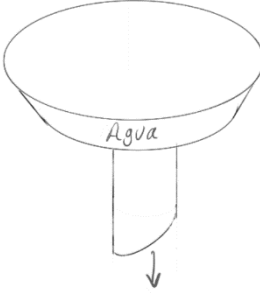
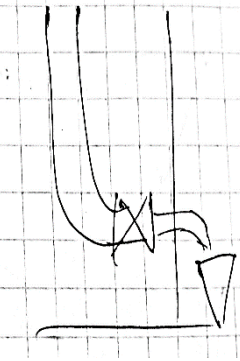
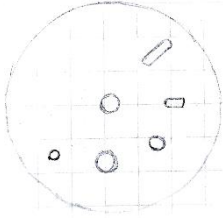
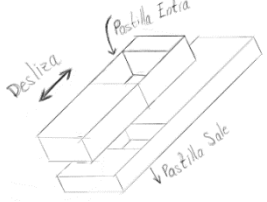
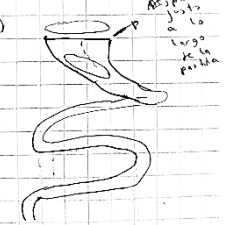
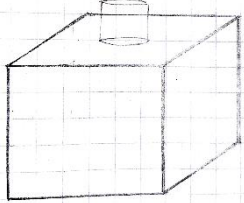
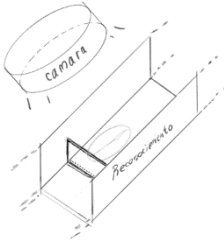

<p>Salida final de Medicamentos</p> <p>[H]</p>			
<p>Salida final de Agua</p> <p>[I]</p>			
<p>Extractor de Medicina</p> <p>[J]</p>			
<p>Identificador de pastilla</p> <p>[K]</p>			

Tabla 11. Matriz morfológica de formas para el diseño conceptual del dispensador.

Selección de rutas

Ruta	Combinaciones
1	A2,B1,C3,D2,E1,F2,G2,H2,I1,J2,K2
2	A1,B3,C2,D3,E3,F1,G3,H3,I2,J1,K1
3	A3,B2,C1,D3,E2,F3,G1,H1,I3,J1,K3

Tabla 12. Rutas solución de la matriz morfológica propuestas por el equipo.

Parámetros y atributos para la evaluación de diseños conceptuales

Atributos:

- **Facilidad de manufactura:** la producción de las piezas para la máquina no implica una gran complicación.
- **Materiales:** estos deben ser de grado farmacéutico.
- **Vida del producto:** teniendo en cuenta que es un prototipo, la vida útil sería de 2 años.
- **Facilidad de uso:** que el usuario no tenga problemas al momento de usar el dispositivo dado que contará con una interfaz intuitiva y descriptiva.
- **Comunicación Wi-Fi:** tendrá la capacidad de ser configurado y administrado por un teléfono inteligente enviando datos a través de una red WLAN de 2.4GHz
- **Conservación de medicamentos:** las condiciones dentro del dispositivo serán las adecuadas para almacenamiento de comprimidos y cápsulas.
- **Organización de medicamentos:** la máquina es capaz de organizar correctamente diferentes tipos de medicamentos.
- **Estética:** Que el dispositivo tiene un diseño agradable. para el usuario.

Parámetros:

- Costo
- Facilidad de implementación
- Velocidad de manufactura
- Número de componentes
- Disponibilidad de mercado
- Simplicidad de ensamble
- Confiabilidad

Evaluación de las rutas seleccionadas

Parámetro	Porcentaje (%)	R1		R2		R3	
		Calificación	Peso	Calificación	Peso	Calificación	Peso
A	12.50	8	10	7	8.75	5	6.25
B	16.66	8	13.328	6	9.996	6	9.996

C	8.88	9	7.992	7	6.216	6	5.328
D	20.83	9	18.747	7	14.581	8	16.664
E	12.5	8	10	8	10	7	8.75
F	12.5	9	11.25	8	10	8	10
G	16.66	9	14.994	9	14.994	9	14.994
TOTAL	100	60	86.311	51	74.537	49	71.982

Tabla 13. Evaluación de las rutas seleccionadas.

A partir de los resultados de la tabla 13, es posible observar que la ruta mejor evaluada resulta ser la denominada R1, teniendo una diferencia de peso de más de 10 puntos respecto a las otras alternativas, por lo cual, es elegido como el Diseño Conceptual de este proyecto.

Boceto general del dispositivo en CAD

En las imágenes 15 y 16 se muestra el posible diseño que tendría el dispositivo completo teniendo en cuenta las formas elegidas en la matriz morfológica.

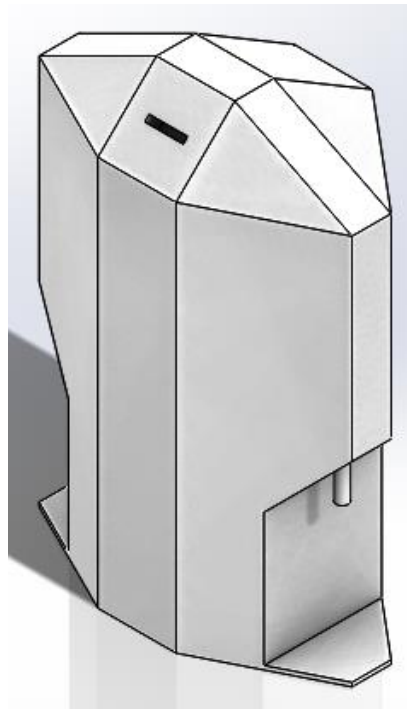


Imagen 15. Boceto general del dispositivo (vista isométrica).

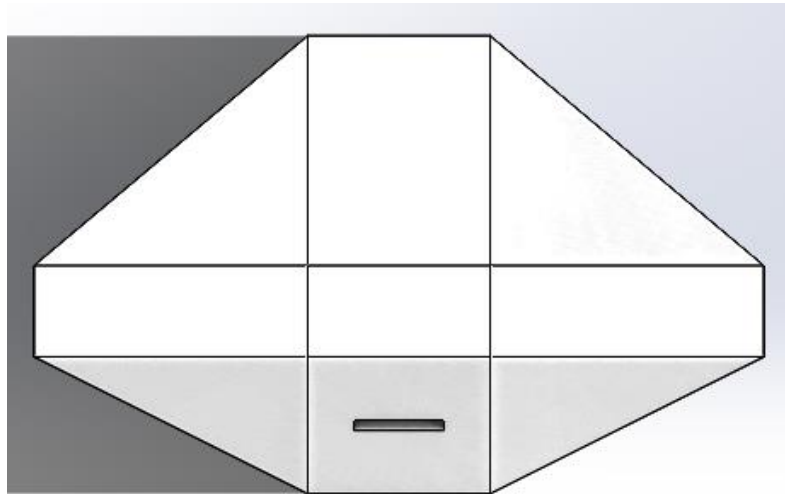
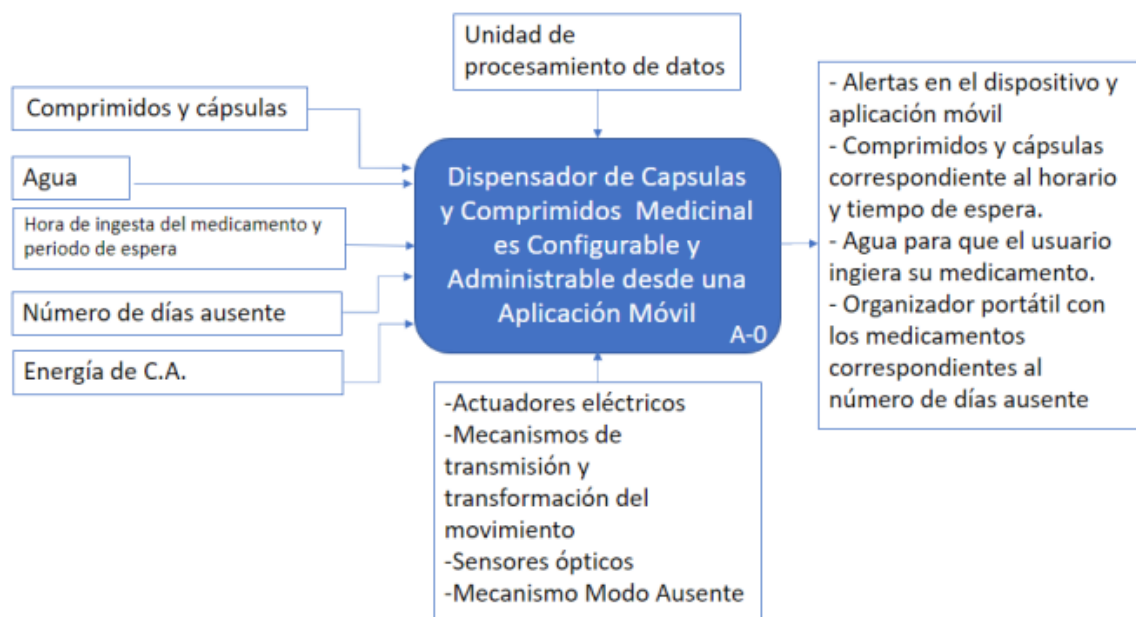


Imagen 16. Boceto general del dispositivo (vista superior).

IDEF A-0



Unidad de procesamiento de datos: Este componente se propone como controlador ya que se utilizará para determinar y activar, por medio de algoritmos de control, las acciones de los mecanismos de todas las áreas funcionales, esto con el fin de conseguir disminuir el error en cada sistema antes mencionado.

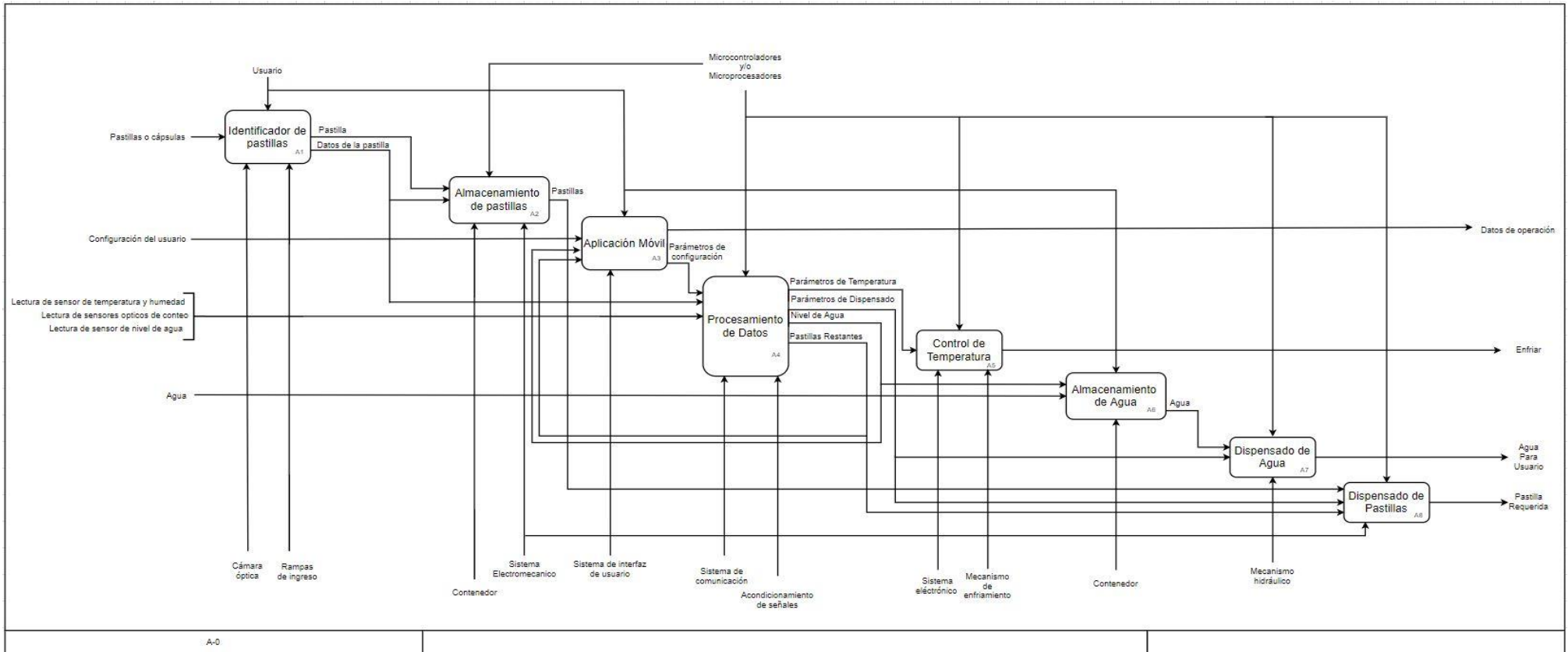
Actuadores eléctricos / Mecanismos de transmisión y transformación del movimiento / Sensores ópticos / Mecanismo modo ausente: Se proponen estos elementos como mecanismos debido a que se consideran necesarios en cumplir con la función que se desea realizar, además de que darán señales de entrada para poder hacer un control de la acción que hacen estos mecanismos.

Agua / Hora de ingesta del medicamento y tiempo de espera / Número de días ausente / Energía de CA: Estos elementos se han considerado como entradas al sistema



ya que son las variables que deben ser suministradas para que el dispositivo pueda funcionar. Son los elementos que el aparato usa para llevar a cabo sus tareas principales.

Alertas en el dispositivo y aplicación móvil / Comprimidos y cápsulas correspondiente al horario y tiempo de espera / Agua para que el usuario ingiera el comprimido o capsula / Organizador portátil con los medicamentos correspondientes al número de días ausente: Estas acciones y/o elementos son considerados salidas porque son los resultados esperados a partir de todas las acciones que realiza el dispositivo y que son gobernadas por los sistemas de control y ejecutadas por los mecanismos del sistema.



IDEF A0



Diseño detallado

Diseño del Contenedor de Medicamentos

Ésta es la primera actividad del diseño detallado del proyecto, que implica el comienzo de todo el proceso de este con una de las piezas claves para el funcionamiento del aparato y sobre la cual gira el diseño de muchas otras de sus partes y mecanismos.

El principal reto consiste en lograr un diseño que permita almacenar de manera organizada los medicamentos y, a su vez, permita una extracción ordenada y efectiva de las pastillas para su dispensado.

Debido a que las pastillas no suponen grandes cargas en el contenedor, la principal preocupación del diseño no radica en los cálculos estructurales sino en alcanzar un diseño que cumpla con una cinemática para poder dispensar el medicamento deseado.

Para efecto del diseño de los contenedores, es preciso primero mencionar que, estos, están delimitados a cierto tipo y tamaño de pastilla. Siguiendo los objetivos específicos del proyecto y los requerimientos de sistema, se han seleccionado las siguientes:

Pastilla 1: Comprimidos de geometría redonda con diámetros 8 a 10 mm

Pastilla 2: Tabletas de geometría elíptica (ovalada) con un ancho de 4 a 6 mm y un largo máximo de 17 mm

Pastilla 3: Cápsulas con un ancho de 3 a 5 mm y un largo máximo de 17 mm

Posteriormente, se propusieron los valores de las dimensiones de los contenedores considerando las geometrías y dimensiones de las pastillas que el aparato será capaz de dispensar (Ver el apartado de validación de contenedores en la página 31). Esto también, de acuerdo a la selección realizada en la ruta solución de la matriz morfológica, se diseñaron los contenedores de medicamentos.

Se propone un contenedor cilíndrico que en su interior contenga un tornillo transportador. Dicho tornillo permitirá una correcta organización de las pastillas en el contenedor y a su vez permitirá tener un mecanismo que admita una extracción ordenada de las mismas al momento de su dispensado. Este contenedor tendrá capacidad para al menos 15 pastillas.

Cálculos analíticos de los contenedores de pastillas:

Dimensiones conocidas:

Como datos conocidos tenemos las dimensiones de las pastillas, por lo que, a su vez, conocemos también los anchos mínimos que deben tener las hélices de los tornillos dispensadores, pues estos dependen directamente del tamaño de las piezas que el tornillo debe transportar.



Para tener un diámetro exterior en el tornillo de forma que las pastillas puedan caber sin que exista riesgo de que se caigan, se propone que sean del ancho máximo de las pastillas que va a transportar:

- Ancho de la Hélice del Tornillo 1: 10 mm.
- Ancho de la Hélice del Tornillo 2: 6 mm.
- Ancho de la Hélice del Tornillo 3: 5 mm.

Para el eje del eje del tornillo sinfín, se propone un diámetro de 20 mm, con la finalidad de tener un soporte fuerte para la cuerda. Debido a que las cargas a las que se someterá el tornillo consisten en únicamente cargas puntuales sobre la hélice y, además, dichas cargas son muy bajas:

- Peso máximo de una pastilla que puede almacenar el dispositivo: 1.1 g

Y el hecho de que no existirá un momento torsional sobre el eje. No se considera estructuralmente necesario que se calcule un ancho del eje.

Tomando en cuenta las consideraciones antes mencionadas, también nos es posible establecer el espesor de la hélice, el cual se propone de 5 mm.

Debido a que hemos definido el diámetro del eje del tornillo, podemos entonces definir lo siguiente:

- Diámetro interior de los tornillos: 20 mm

Tornillo 1:

- Diámetro medio: 30 mm
- Diámetro exterior: 40 mm

Tornillo 2:

- Diámetro medio: 26 mm
- Diámetro exterior: 32 mm

Tornillo 3:

- Diámetro medio: 25 mm
- Diámetro exterior: 30 mm

Respecto a la cubierta y la base del contenedor, debido a que el diámetro exterior del tornillo es un dato conocido, podemos definir el diámetro interno de ambas piezas, pues son iguales, además, el grosor de ambas piezas es propuesto de 3 mm debido a que éstas no se encontrarán sometidas a ningún tipo de carga o esfuerzo. Una consideración más cuenta es que debe existir holgura o espacio entre la parte más externa de la hélice del tornillo y la pared de la cubierta. Para tener dicha



holgura, al diámetro exterior de la hélice se le agrega un milímetro, de manera que:

Contenedor 1:

- Diámetro interno de la base: 41 mm
- Diámetro exterior de la base: 47 mm

Contenedor 2:

- Diámetro interno de la base: 33 mm
- Diámetro exterior de la base: 39 mm

Contenedor 3:

- Diámetro interno de la base: 31 mm
- Diámetro exterior de la base: 37 mm

Otra dimensión importante será la altura tanto del tornillo como de la cubierta del contenedor, los cuales son valores propuestos:

Respecto a la altura del tornillo sinfín, sabemos que una revolución de la hélice (un filete) entonces se calcula el número de hélices, como se muestra en la imagen 17:

$$\text{Num de hélices mínimo} = \frac{15 \text{ pastillas}}{4.6 \text{ pastillas}} = 3.26 \cong 3.5 \text{ hélices}$$

Imagen 17. Cálculo del número de hélices mínimo en el tornillo.

Debido a que el tipo de tornillo transportador es vertical, y siguiendo las recomendaciones de las bibliografías citadas respecto a este tema en el marco teórico, el paso de la hélice será un Paso Corto, lo cual implica que el paso será la mitad del diámetro exterior del tornillo, por lo que se tiene un paso:

$$\text{Paso diametral} = 20 \text{ mm}$$

De aquí, sabemos que el paso es la distancia entre un filete y otro, por lo que para que existan cuatro hélices en el tornillo deberá tener al menos 80 mm de longitud, sin embargo, para este proyecto se propone que los tornillos tengan 130 mm de longitud.

Una vez justificado el largo del tornillo, podemos intuir que el largo de la cubierta deberá ser ligeramente superior al largo del tornillo, ya que se busca evitar rozamiento entre el eje del tornillo y la tapa del contenedor, por lo que se propone una altura de la cubierta de 123 mm, considerando que 1 centímetro del tornillo queda por fuera de la base del contenedor.

Quedan aquí, definidos y justificados, todas las dimensiones propuestas y conocidas empleadas en el diseño y validación de los compartimientos de almacenamiento de pastillas.

Cálculos del tornillo transportador:

1. Cálculos Estructurales:

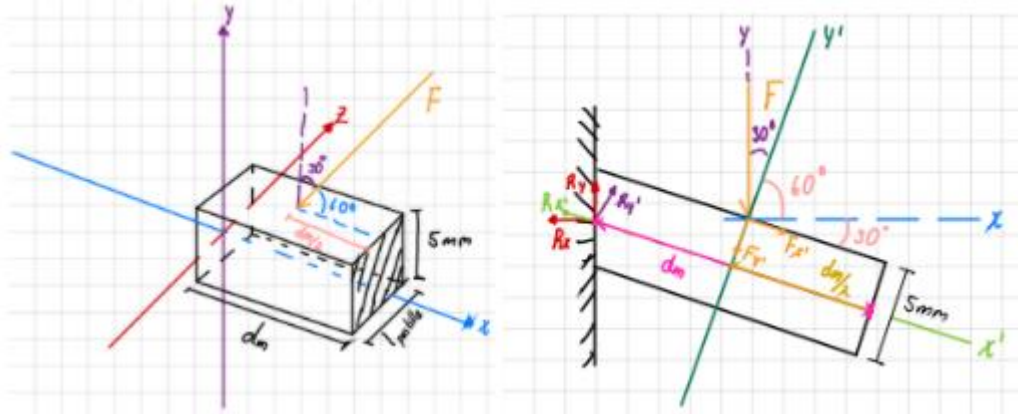


Imagen 18. Diagramas de cuerpo libre de una sección de la hélice del tornillo sin fin.

De la imagen 18, se sabe que:
 lpastilla= Longitud de la pastilla.
 dm = diámetro medio.

o Tornillo 1:

Ecuaciones de equilibrio de fuerza.

$$\Sigma F_{x'} = -R_{x'} + F_{x'} = 0$$

$$\Sigma F_{y'} = R_{y'} - F_{y'} = 0$$

Obtención del peso de la pastilla.

$$F = \left(\frac{9.81m}{s^2}\right) (0.0011kg) = 10.791 \times 10^{-3} N$$

Obtención de las componentes del peso de la pastilla.

$$F_{x'} = (10.791 \times 10^{-3} N) \text{sen}(30) = 5.3955 \times 10^{-3} N$$

$$F_{y'} = (10.791 \times 10^{-3} N) \text{cos}(30) = 9.3452 \times 10^{-3} N$$

Obtención de las reacciones en los ejes coordenados primos.

$$R_{x'} = 5.3955 \times 10^{-3} N$$

$$R_{y'} = 9.3452 \times 10^{-3} N$$

Sistema de ecuaciones para obtener las reacciones Rx y Ry.

$$R_{y'} \text{cos}(30) - R_{x'} \text{sen}(30) = 9.3452 \times 10^{-3} N$$

$$R_{y'} \text{cos}(60) + R_{x'} \text{cos}(30) = 5.3955 \times 10^{-3} N$$



Valores de las reacciones en los ejes coordenados.

$$R_y = 10.791 \times 10^{-3} N$$

$$R_x = 8.0132 \times 10^{-8} N$$

Esfuerzo axial

Donde:

A = Área de la sección transversal de la hélice.

Rx' = Fuerza aplicada a lo largo de la hélice.

Obtención del esfuerzo axial.

$$\sigma_{axial} = \frac{Rx'}{A} = \frac{5.3955 \times 10^{-3} N}{(5 \times 10^{-3} m)(10 \times 10^{-3} m)} = 107.91 Pa$$

Esfuerzo flector

Donde:

M = Momento flector.

I = Momento de inercia.

c = Distancia al centroide de la sección transversal de la hélice.

$$\Sigma M = 0$$

$$M - Fy'(5 \times 10^{-3} m) = M - (9.3452 \times 10^{-3} N)(5 \times 10^{-3} m) = M - 46.726 \times 10^{-6} Nm = 0$$

$$M = 46.726 \times 10^{-6} Nm$$

Cálculo del Momento flector, M.

$$\sigma_{flector} = \frac{Mc}{I} = \frac{(46.726 \times 10^{-6} Nm)(2.5 \times 10^{-3} m)}{\frac{1}{12}(10 \times 10^{-3} m)(5 \times 10^{-3} m)^3} = 1121.424 Pa$$

Cálculo del esfuerzo flector.

Esfuerzo total parte superior de la viga

$$\sigma_{TS} = \sigma_{axial} + \sigma_{flector} = 107.91 Pa + 1121.424 Pa = 1229.334 Pa$$

Cálculo del esfuerzo total de la parte superior de la viga, TS.

Esfuerzo total parte inferior de la viga

$$\sigma_{TI} = \sigma_{axial} - \sigma_{flector} = 107.91 Pa - 1121.424 Pa = -1013.514 Pa$$

Cálculo del esfuerzo total de la parte inferior de la viga, TI.

- Tornillo 2:

Dado que el peso máximo de una pastilla que puede almacenar el dispositivo es de 1.1 g, el análisis de las fuerzas estáticas en el diseño de cada uno de los tornillos es el mismo, en cuestión de resultados. Siendo el análisis de esfuerzos lo que cambia en cada uno, por lo que se presenta a continuación.

Esfuerzo axial

$$\sigma_{axial} = \frac{Rx'}{A} = \frac{5.3955 \times 10^{-3} N}{(5 \times 10^{-3} m)(6 \times 10^{-3} m)} = 179.85 Pa$$

Obtención del esfuerzo axial, axial.

Esfuerzo flector

$$\Sigma M = 0$$

$$M - Fy'(3 \times 10^{-3} m) = M - (9.3452 \times 10^{-3} N)(3 \times 10^{-3} m) = M - 2.80356 \times 10^{-5} Nm = 0$$

$$M = 2.80356 \times 10^{-5} Nm$$

Cálculo del Momento flector, M.

$$\sigma_{flector} = \frac{Mc}{I} = \frac{(2.80356 \times 10^{-5} Nm)(2.5 \times 10^{-3} m)}{\frac{1}{12} (6 \times 10^{-3} m)(5 \times 10^{-3} m)^3} = 1121.424 Pa$$

Cálculo del esfuerzo flector.

Esfuerzo total parte superior de la viga

$$\sigma_{TS} = \sigma_{axial} + \sigma_{flector} = 179.85 Pa + 1121.424 Pa = 1301.274 Pa$$

Cálculo del esfuerzo total de la parte superior de la viga, TS.

Esfuerzo total parte inferior de la viga

$$\sigma_{TI} = \sigma_{axial} - \sigma_{flector} = 179.85 Pa - 1121.424 Pa = -941.574 Pa$$

Cálculo del esfuerzo total de la parte inferior de la viga, TI.

o Tornillo 3:

Como en el caso anterior, el peso máximo de una pastilla que puede almacenar el dispositivo es de 1.1 g, el análisis de las fuerzas estáticas en el diseño de cada uno de los tornillos es el mismo, en cuestión de resultados. Siendo el análisis de esfuerzos lo que cambia en cada uno, por lo que se presenta a continuación.

Esfuerzo axial

$$\sigma_{axial} = \frac{Rx'}{A} = \frac{5.3955 \times 10^{-3} N}{(5 \times 10^{-3} m)(5 \times 10^{-3} m)} = 215.82 Pa$$

Obtención del esfuerzo axial.

Esfuerzo flector

$$\Sigma M = 0$$

$$M - Fy'(2.5 \times 10^{-3} m) = M - (9.3452 \times 10^{-3} N)(2.5 \times 10^{-3} m) = M - 2.3363 \times 10^{-5} Nm = 0$$

$$M = 2.3363 \times 10^{-5} Nm$$

Cálculo del Momento flector, M.

$$\sigma_{Flector} = \frac{Mc}{I} = \frac{(2.3363 \times 10^{-5} Nm)(2.5 \times 10^{-3} m)}{\frac{1}{12}(5 \times 10^{-3} m)(5 \times 10^{-3} m)^3} = 1121.424 Pa$$

Cálculo del esfuerzo flector.

Esfuerzo total parte superior de la viga

$$\sigma_{TS} = \sigma_{axial} + \sigma_{Flector} = 215.82 Pa + 1121.424 Pa = 1337.244 Pa$$

Cálculo del esfuerzo total de la parte superior de la viga, TS.

Esfuerzo total parte inferior de la viga

$$\sigma_{TI} = \sigma_{axial} - \sigma_{Flector} = 215.82 Pa - 1121.424 Pa = -905.604 Pa$$

Cálculo del esfuerzo total de la parte inferior de la viga, TI.

Las idealizaciones realizadas en estos procedimientos tienen fundamento en la teoría de tornillos de potencia del libro: "Diseño en Ingeniería Mecánica de Shigley" 9na Edición, Editorial McGrawHill. Autores: Richard G. Budynas & J. Keith Nisbett, Año de Publicación: 2012

- Especificaciones de Diseño adicionales para los Tornillos:

En el diseño propuesto, el tornillo transportador debe contar con un tope o sostén al final de la hélice, el cual, cumple con la función de impedir que las pastillas se salgan de los filetes y se acumulen en la base y atasquen el sistema de dispensado.

La barrera o tope se ubica a una distancia de la mitad del largo máximo de la pastilla que almacena el contenedor y su superficie es paralela al vértice de la última hélice.

Dicho lo anterior, el tope para las pastillas debe nacer del eje y orientarse de la siguiente manera, como lo muestra la imagen 19:

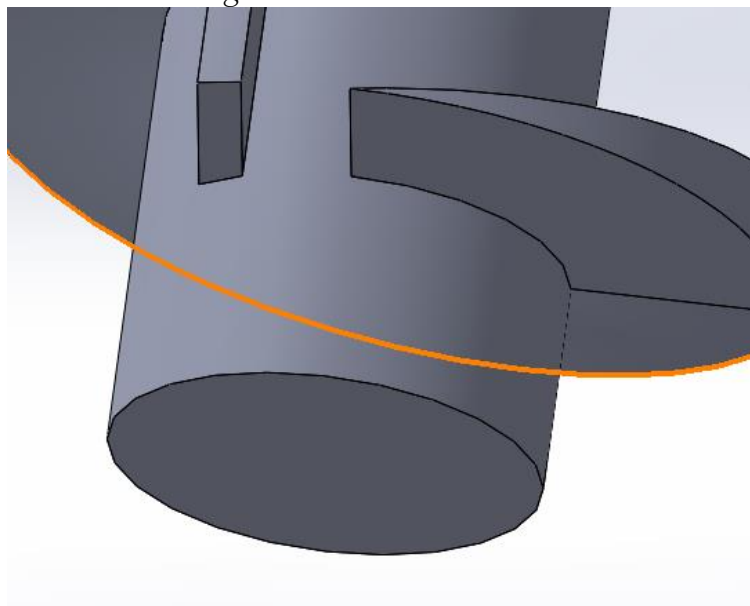


Imagen 19. Tope de las pastillas.



Entonces, las especificaciones generales de diseño serían:

- Altura del Tope: distancia entre el final de la cuerda y la superficie inferior de la penúltima cuerda.
- Ancho del Tope: igual al ancho de la hélice del tornillo en el que se esté aplicando.
- Distancia entre el tope y el final de la hélice: un medio de la longitud máxima de la pastilla que puede almacenar el contenedor.

Especificaciones de diseño adicionales del contenedor (Base y Cubierta):

Recordemos que se ha diseñado un contenedor por cada tipo de medicamento que el dispensador es capaz de almacenar. Por lo mismo, cada uno tiene orificios para entrada y salida de las pastillas de diferentes tamaños.

En general, en las cubiertas de los contenedores, el agujero de sus tapas es coincidente con el inicio de la cuerda del tornillo sinfín, su ancho corresponde al ancho de la hélice del tornillo en el que se aplican y su longitud es de 8 milímetros para garantizar la entrada de las pastillas al contenedor. Es preciso mencionar que las pastillas ingresarán en diagonal al contenedor y no de forma vertical, por lo que el largo del agujero no impedirá el fácil ingreso de los medicamentos.

Respecto a la base, debido a que el eje del tornillo debe atravesarla, ésta deberá tener un agujero. Para evitar el rozamiento entre las paredes del orificio y el eje del tornillo, se agrega un milímetro de diámetro a la magnitud del diámetro interior del tornillo, es decir, el diámetro del agujero será de 21 mm para todas las bases de los contenedores.

Por otra parte, la base deberá tener un agujero por el cual las pastillas serán extraídas del contenedor, de forma similar con el orificio de la tapa, el de las bases debe ser coincidente con el inicio de la cuerda del tornillo sinfín, su ancho corresponde al ancho de la hélice del tornillo en el que se aplican y su longitud es el espesor de las pastillas que puede almacenar el contenedor.

A continuación, se muestran los modelos en CAD de los contenedores diseñados:

- Contenedor 1 (para la Pastilla 1):
De este contenedor se muestran las partes que lo conforman, como es la cubierta que se muestra en las imágenes 20 y 21, y la base que tapa a la cubierta que se ve en la imagen 22.



Imagen 20. Cubierta del contenedor del tornillo 1, isométrico.

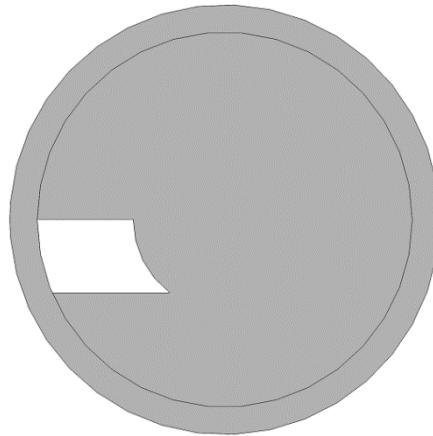


Imagen 21. Vista frontal de la cubierta del tornillo 1.

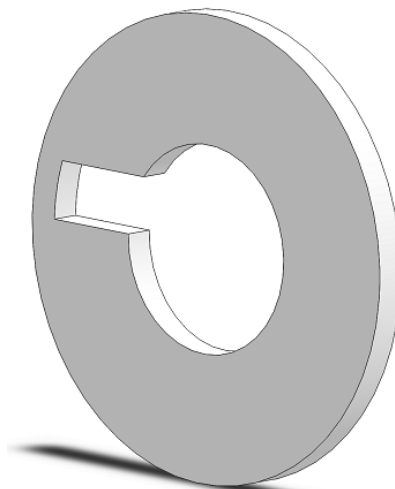


Imagen 22. Vista isométrica de la base de la cubierta del tornillo 1.

- Contenedor 2 (para la Pastilla 2):
De este contenedor se muestran las partes que lo conforman, como es la cubierta que se muestra en las imágenes 23 y 24, y la base que tapa a la cubierta que se ve en la imagen 25.

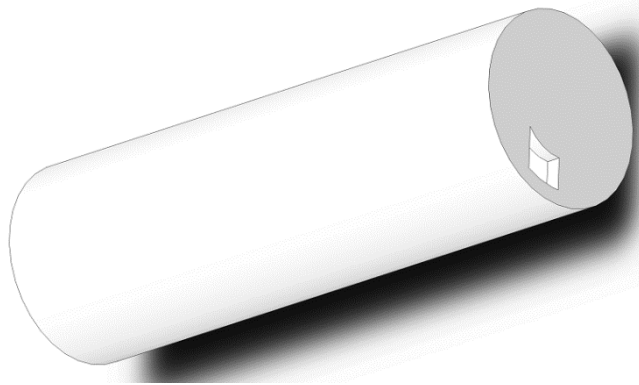


Imagen 23. Cubierta del contenedor del tornillo 2, isométrico.

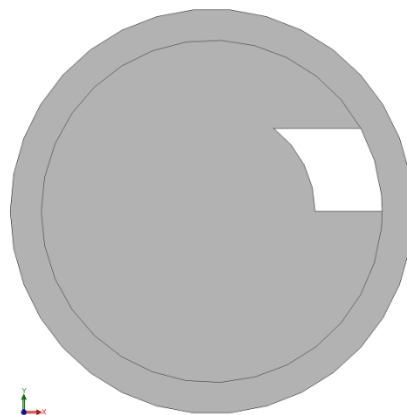


Imagen 24. Vista frontal de la cubierta del tornillo 2.

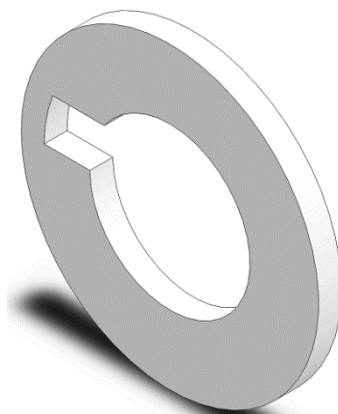


Imagen 25. Vista isométrica de la base de la cubierta del tornillo 2.

- Contenedor 3 (para la Pastilla 3):
De este contenedor se muestran las partes que lo conforman, como es la cubierta que se muestra en las imágenes 26 y 27, y la base que tapa a la cubierta que se ve en la imagen 28.

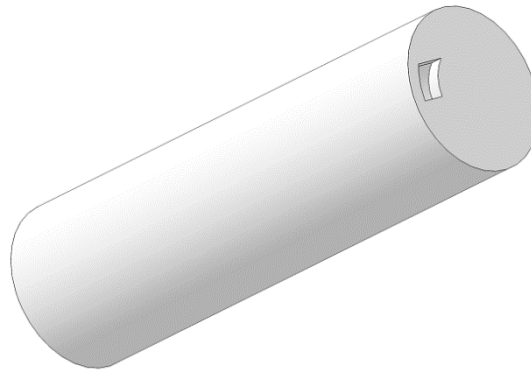


Imagen 26. Cubierta del contenedor del tornillo 3, isométrico.

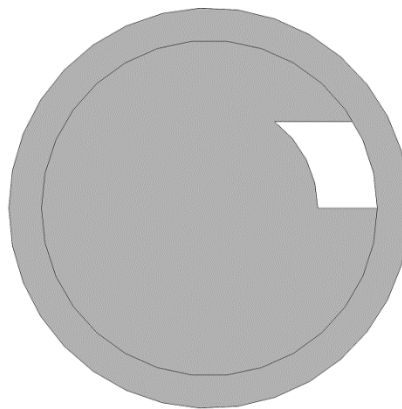


Imagen 27. Vista frontal de la cubierta del tornillo 3.

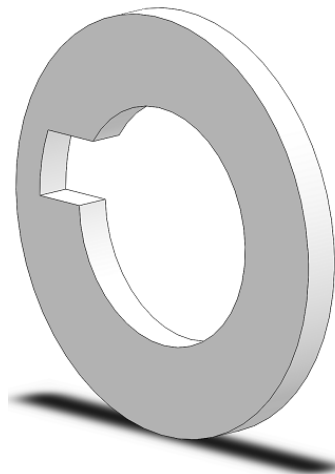


Imagen 28. Vista isométrica de la base de la cubierta del tornillo 3.

A su vez, los contenedores albergan en su interior un tornillo sin fin que permite la correcta organización de las pastillas, así como su dispensado, estos se pueden visualizar en las imágenes 29, 30 y 31, cada uno referente a su contenedor:

Tornillo del contenedor 1:

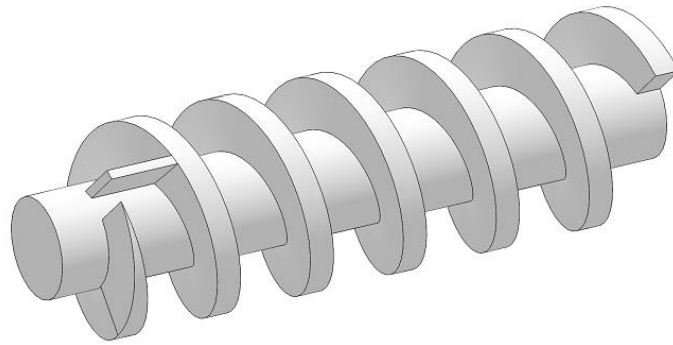


Imagen 29. Vista isométrica del tornillo 1.

Tornillo del contenedor 2:

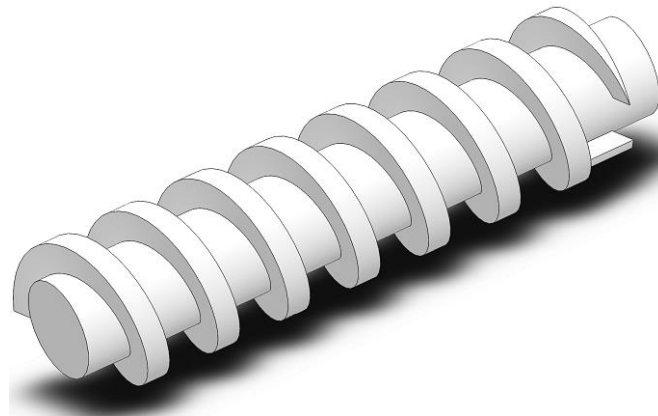


Imagen 30. Vista isométrica del tornillo 2.

Tornillo del contenedor 3:

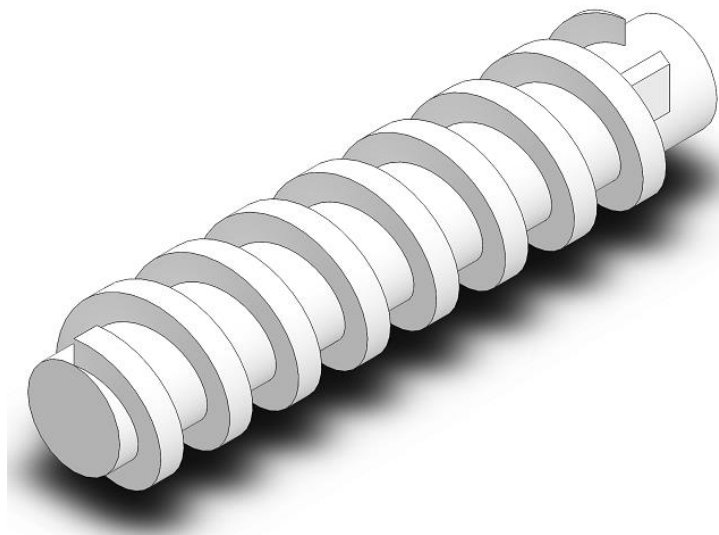


Imagen 31. Vista isométrica del tornillo 3.

Las validaciones computacionales para estos diseños pueden ser consultadas en el apartado de Validaciones de los Contenedores de Medicamento.

Validaciones del Contenedor de Medicamentos

En este apartado se mostrarán las siguientes validaciones computacionales de los tornillos debido a que son los que soportan el peso de las pastillas, lo que se podrá notar en las siguientes imágenes que es prevalece el color azul dado que representa que el esfuerzo existente es bajo.

Como se puede observar en la imagen 32 el esfuerzo representado en MPa, y que resultó de la validación es de 1.1284×10^{-3} MPa y es menor en un 8.21% del valor que se obtuvo en cálculos analíticos, lo cual quiere decir que el análisis que realizó se puede aceptar como correcto. Además de que son esfuerzos de magnitud baja, por lo que, no debería existir una deformación importante.

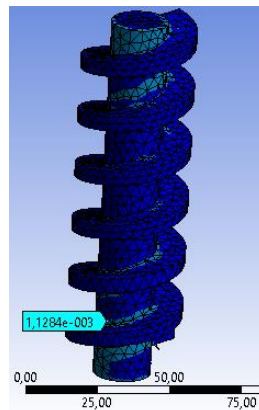


Imagen 32. Validación de esfuerzo calculado en el tornillo 1.

Como se puede observar en la imagen 33 el esfuerzo representado en MPa, y que resultó de la validación es de 1.3408×10^{-3} MPa y sobrepasa en un 3.037% del valor que se obtuvo en cálculos analíticos, lo cual quiere decir que el análisis que realizó se puede aceptar como correcto. Y que, de igual forma, los esfuerzo que se manejan se pueden considerar soportables por el diseño.

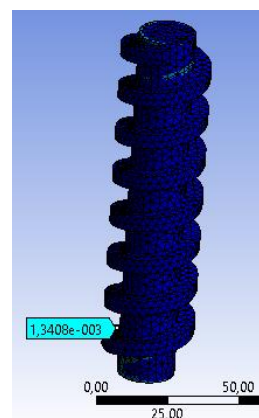


Imagen 33. Validación de esfuerzo calculado en el tornillo 2.

Como se puede observar en la imagen 34 el esfuerzo representado en MPa, y que resultó de la validación es de $1.3003e-003$ MPa y es menor en un 2.76% del valor que se obtuvo en cálculos analíticos, lo cual quiere decir que el análisis que realizó se puede aceptar como correcto. Y que, de igual forma, los esfuerzo que se manejan se pueden considerar soportables por el diseño.

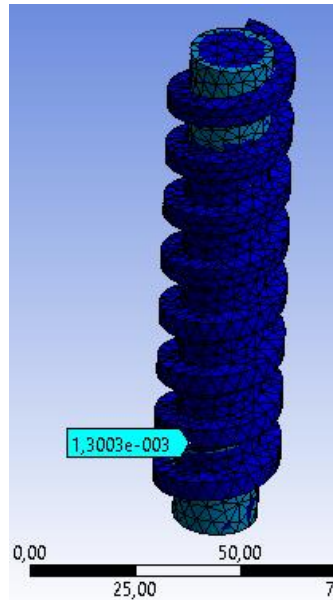


Imagen 34. Validación de esfuerzo calculado en el tornillo 3.

Obtención experimental del ángulo y coeficiente de fricción estática entre Polivinilo de Cloruro (PVC) y Excipiente CBP, así como Polivinilo de Cloruro (PVC) y Glicerina de las Cápsulas:

Material:

- Rampa de PVC
- Pastillas de Excipiente CBP
- Cápsulas a base de gelatina (Glicerina)
- Transportador Virtual (Aplicación Móvil)

Procedimiento:

El procedimiento que se describe a continuación es el mismo para el cálculo de ambos coeficientes buscados:

Se tomó la rampa de PVC y se colocó de forma horizontal sobre una superficie plana, después, sobre la rampa se colocó una pastilla y, poco a poco, se fue incrementando el ángulo entre el extremo derecho de la rampa y la superficie plana que sirve como eje horizontal "X", así hasta encontrar el ángulo en el que la pastilla comienza a deslizarse en la rampa.

Una vez hallado el ángulo, se tomó nota y fotografía de su valor y el procedimiento se repitió en otras 5 ocasiones.

De los 5 valores encontrados, se elige el valor más grande con la intención de asegurar que el ángulo garantice que se ha roto la fuerza fricción estática.

Para conocer el coeficiente de fricción estática, basta con calcular el arcotangente del ángulo encontrado.

- *Excipiente CBP*
 - *Comprimido:*
Como se puede apreciar en la imagen 35, el comprimido se encuentra en un ángulo de 18° , en el cual aún no presenta deslizamiento. Por lo que este ángulo no es el indicado para lo que se desea.



Imagen 35. Comprimido a un ángulo de 18° donde no se desliza.

En la imagen 36, el comprimido se encuentra en un ángulo de 30° , en el cual se desliza el comprimido, pero su deslizamiento fue en el momento, eso quiere decir que, debe empezar a un ángulo menor.



Imagen 36. Comprimido a un ángulo de 30° donde se desliza.

Ahora se puede observar en la imagen 37 que a un ángulo de 28.5° el comprimido justo comienza a deslizarse, esto da como resultado que el ángulo de 30° que se propuso para el diseño del tornillo sinfín es correcto para que el comprimido se pueda deslizar sin algún problema.



Imagen 37. Comprimido a un ángulo de 28.5° donde se empieza a deslizar.

El ángulo obtenido fue de 28.5 grados aproximadamente, por lo tanto:

$$\mu_{\text{comprimidos}} = \tan(28.5^\circ) = 0.5429$$

Cálculo del coeficiente de fricción de los comprimidos.

- *Tabletas:*
Para las tabletas se realizó el mismo proceso de medición del ángulo como se realizó con los comprimidos. En la imagen 38, se probó con un ángulo menor al que se propuso para observar si empezaba a deslizarse la tableta; el ángulo fue de 25° en el cual la tableta no se deslizó.



Imagen 38. Tableta a un ángulo de 25° donde no se desliza.

Ahora se procede a medir con el ángulo de 30° de inclinación para la tableta, como se muestra en las imágenes 39 y 40, donde la tableta se desliza sin complicaciones.



Imagen 39. Tableta a un ángulo de 30° donde se desliza.



Imagen 40. Tableta a un ángulo de 30° donde ya se deslizó.

El ángulo obtenido fue de 30 grados aproximadamente, por lo tanto:

$$\mu_{tabletas} = \tan(30^\circ) = \frac{\sqrt{3}}{3} = 0.7573$$

Cálculo del coeficiente de fricción de las tabletas.

- *Glicerina (Gelatina) de Cápsulas*

Para las cápsulas se realizó el mismo proceso anterior para comprobar el ángulo propuesto, y se empezó con un ángulo de 25°, que como se puede visualizar en la imagen 41, donde la cápsula no se puede deslizar.



Imagen 41. Cápsula a un ángulo de 25° donde no se desliza.

Ahora se prueba con el ángulo de 30° , como se ve en las imágenes 42 y 43, que la cápsula se puede deslizar con libertad y sin algún problema, por lo que igual se valida que el ángulo propuesto cumple también para las cápsulas.



Imagen 42. Cápsula a un ángulo de 30° donde se desliza.



Imagen 43. Cápsula a un ángulo de 30° donde ya se deslizó.

El ángulo obtenido fue de 30 grados aproximadamente, por lo tanto:

$$\mu_{capsulas} = \tan(30^\circ) = 0.5095$$

Cálculo del coeficiente de fricción de las cápsulas.

El ángulo máximo encontrado fue de 30° y debido a que dicho ángulo el resto de las pastillas también ya habrían superado la fricción estática se tomará como el ángulo de inclinación para la hélice respecto al eje horizontal:

$$\gamma_{hélice} = 30^\circ$$

Ángulo de avance del tornillo.

Es decir, el ángulo de la hélice será:

$$\beta_{hélice} = 90^\circ - \gamma_{hélice} = 90^\circ - 30^\circ = 60^\circ$$

Ángulo de inclinación de la hélice.

Diseño del Contenedor de Agua

El contenedor de agua del dispositivo es una de las piezas esenciales para el funcionamiento del aparato, pues este contendrá el agua que será dispensada al usuario para que este último pueda ingerir sus medicamentos.

De acuerdo a los objetivos, se busca que el contenedor tenga capacidad de almacenamiento para un litro de agua potable, lo cual será una de las bases para el diseño del mismo. Adicionalmente, se busca tener apego a la geometría del diseño conceptual del dispositivo. Otra cuestión que también resulta relevante de mencionar es que el contenedor deberá ser desmontable para que sea de fácil limpieza.

Teniendo lo anterior en cuenta, el equipo desarrollador infiere que, en pro de facilitar los análisis y la aplicación, el contenedor debería ubicarse en la parte más alta del aparato de tal forma que el dispensado pueda realizarse por gravedad. Dicho esto, se retoma la vista superior del dispositivo para inspirar la forma del contenedor, la cual se puede ver en la imagen 44.

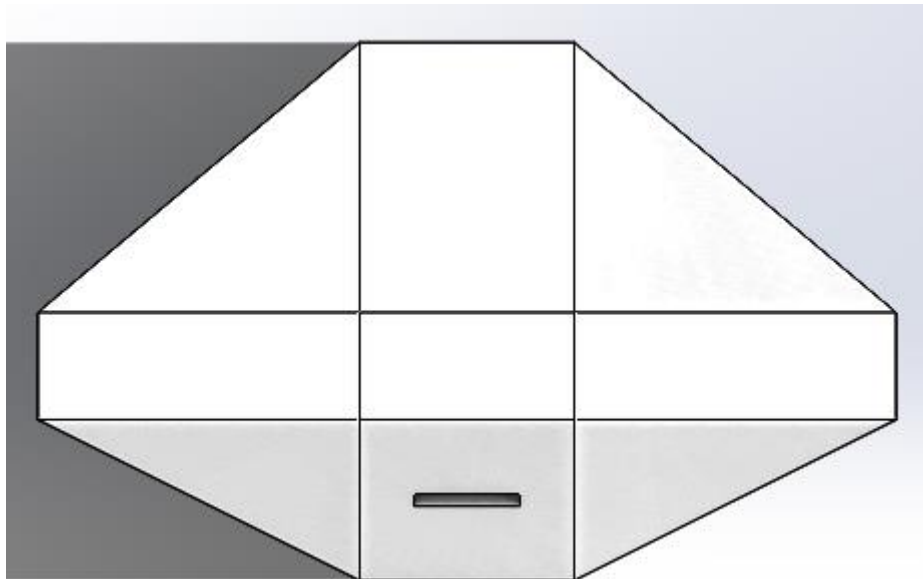


Imagen 44. Vista superior del boceto general del dispositivo.

Buscando realizar una buena administración de espacio, se propone que la forma de la parte superior del contenedor sea de la forma de la mitad trasera del dispositivo, la cual se puede apreciar en la Imagen 45.

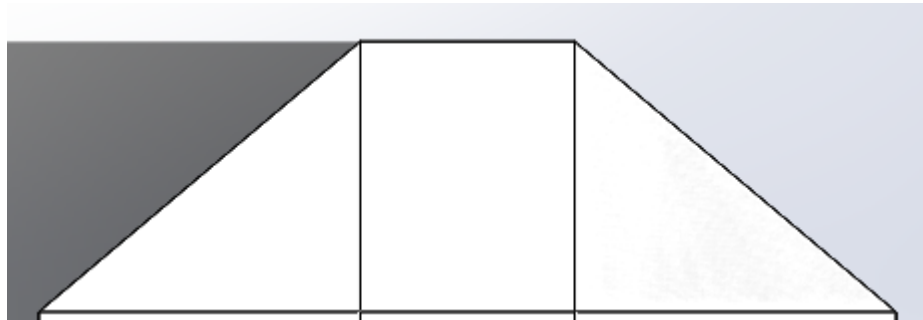


Imagen 45. Vista superior de la parte trasera del dispositivo concepto.

Habiendo aclarado lo anterior, se procedió a proponer las medidas de la pieza, esto considerando que el contenedor ocupará un área de la mitad del aparato. Recordemos también que, para evitar problemas de dimensiones y espacios, la “carcasa” o cuerpo del dispositivo será lo último en ser diseñado y se acoplará las dimensiones y ocupabilidad de espacio de los elementos que integran el aparato.

A continuación, se indican las medidas propuestas para la geometría base del contenedor que es un trapecio isósceles:

- Altura: 80 mm
- Base Mayor: 180 mm
- Base Menor: 60 mm

Las dimensiones de la geometría propuesta serían las medidas exteriores del contenedor, se propone adicionalmente que las paredes tengan un espesor de tres milímetros, el cual es un grosor bastante robusto para la aplicación en cuestión, de forma que las medidas del trapecio interior serían:

- Altura: 74 mm
- Base Mayor: 168 mm
- Base Menor: 57 mm

En la imagen 46, podemos ver la geometría dibujada en SolidWorks con las acotaciones correspondientes a la parte exterior del contenedor de agua.

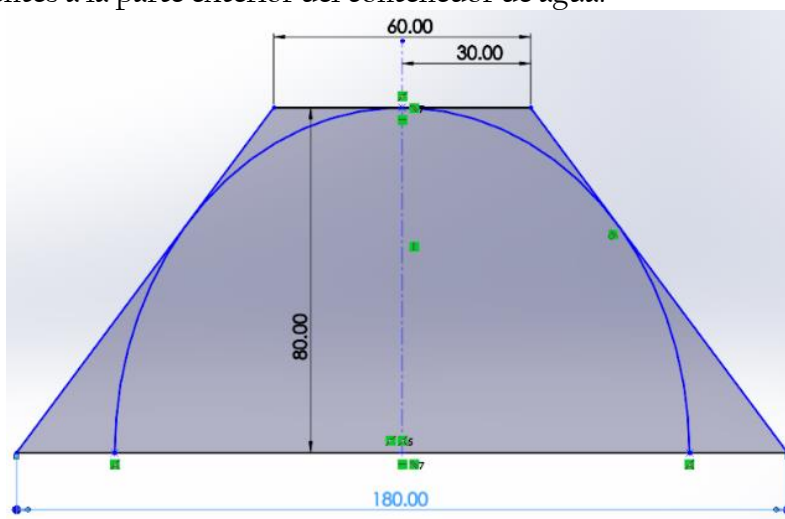


Imagen 46. Geometría Base del contenedor.

Sin embargo, el equipo considera que la geometría trapezoidal tiene una forma poco ergonómica y además con múltiples aristas que representan concentradores de esfuerzos, por lo que se propone que dicha geometría se conserve únicamente en la parte superior del contenedor y el cuerpo principal de la pieza y donde se almacena el agua tenga una geometría semi elíptica que se acople a las dimensiones del trapecio, esto con la intención de que el cuerpo tenga una forma semicilíndrica.

Para acotar dicha forma, se propone que el centro de la elipse se ubique en la mitad de la base mayor del trapecio exterior y que uno de sus vértices sea coincidente con el punto medio de la base menor del trapecio exterior, dicha curva definiría la cara exterior del cuerpo del contenedor y esa pared también cuenta con un espesor de tres milímetros.

Hasta este punto, hemos definido la sección transversal del compartimento para almacenar agua, cuyo valor podemos conocer usando las herramientas de SolidWorks como se muestra a continuación en la imagen 47:

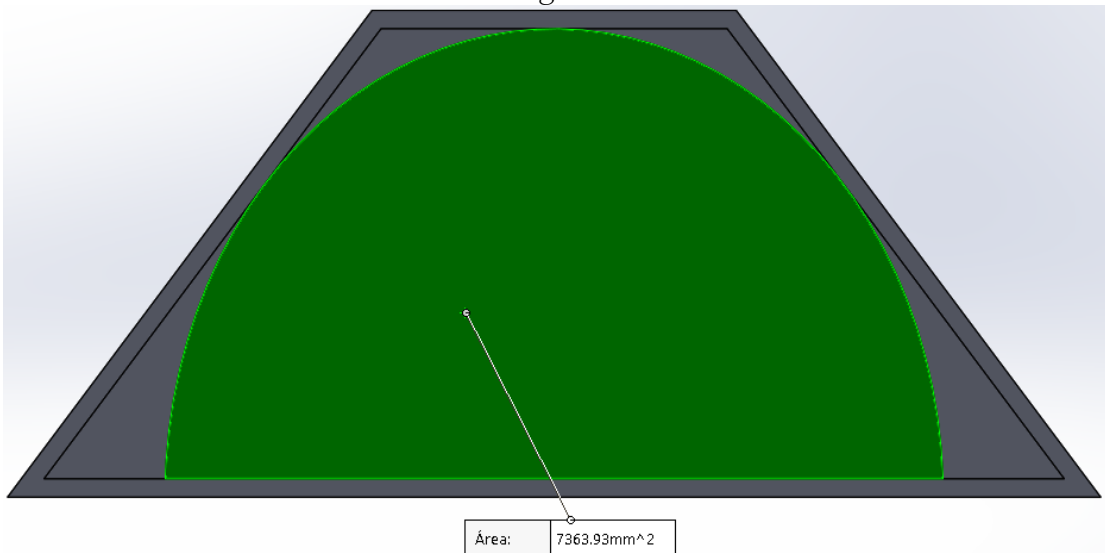


Imagen 47. Medición del área del contenedor.

Ahora, debemos definir la altura del cuerpo del contenedor para que quepa un litro de agua, para ello sabemos que:

$$V = A * L$$

donde V es el volumen de un cuerpo, A la sección transversal y L la longitud del cuerpo.

De tal manera que, si despejamos la longitud, obtendremos la altura que debe tener un elemento cuando se tiene como datos conocidos su volumen y área transversal.

$$L = \frac{V}{A} = \frac{1 \text{ l}}{7363.9293 \text{ mm}^2} = \frac{1000000 \text{ mm}^3}{7363.9293 \text{ mm}^2} = 135.7970 \text{ mm} \approx 136 \text{ mm}$$

Adicionalmente, la parte del trapecio será únicamente como acople al resto del aparato por lo que su altura será de diez milímetros. El espesor que tiene la base inferior del tanque es de tres milímetros, por lo que el contenedor tendrá una altura total de 149 milímetros o 14.9 centímetros.

Para el dispensado de agua se emplearán mangueras de nivel de PVC de $\frac{1}{4}$ pulgada, por lo que adicionalmente al tanque se agrega una salida de agua compatible con el diámetro de las mangueras.

- Diámetro Interior de la salida: 5.38 mm
- Diámetro Exterior de la salida: 6.38 mm
- Diámetro interior de la manguera: 6.35 mm
- Diámetro Exterior de la manguera: 7.75 mm

Finalmente, en las imágenes 48 y 49 podemos observar las vistas superior e isométrica del contenedor.

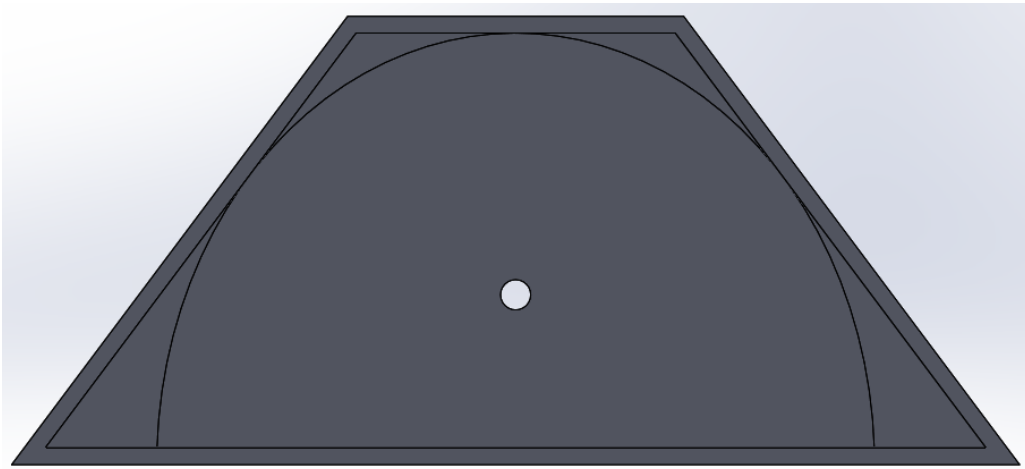


Imagen 48. Vista superior del contenedor.



Imagen 49. Vista isométrica del contenedor.

Sistema de Dispensado de Agua

Para el correcto funcionamiento del dispositivo, resulta imprescindible la existencia de un sistema que pueda cumplir con la función de dispensar agua a los usuarios para que ellos puedan ingerir sus medicamentos.

Para cumplir con este objetivo, se busca emplear el tanque de agua previamente diseñado, así como una electroválvula, de tal manera que se use a favor la gravedad en este proceso. Adicionalmente, se requerirá de mangueras de PVC flexible para los conductos por los cuales se desplazaría el líquido.

Para controlar adecuadamente el dispensado de agua, también resultará imprescindible un sensor de nivel.

Debido a que se espera una altura máxima del dispositivo de 60 centímetros, se considera que el sistema de dispensado de agua no tenga una longitud vertical superior a los 30 centímetros con la finalidad de ahorrar espacios.

Por otra parte, el equipo considera que, debido a que el agua que se planea entregar al usuario es únicamente para lograr ingerir una pastilla, no es necesario brindar más de 100 mililitros al usuario, por lo que se establece como volumen aproximado a entregar al usuario.

Otra cuestión importante de diseño resulta el tiempo de llenado del vaso, el equipo considera que, debido a que se busca que el usuario tenga agilidad en la ingesta de sus medicinas, el debería tener acceso al agua tan pronto como sea posible, por lo que se considera que cinco segundos de espera puede ser el tiempo máximo tolerable

Finalmente, se debe recordar que el aparato es de uso doméstico, por lo que las mangueras y su disposición interna deben permitir un fácil manejo, así como la mayor posibilidad de ahorro de espacios. Ya que se requiere un caudal bastante bajo, pues se busca servir agua en un vaso, se propone el uso de mangueras de $\frac{1}{4}$ de pulgada.

Sabiendo esto, se muestra a manera de esquema, la disposición que tendría el sistema en la Imagen 50, con la finalidad de ilustrar gráficamente la idea.

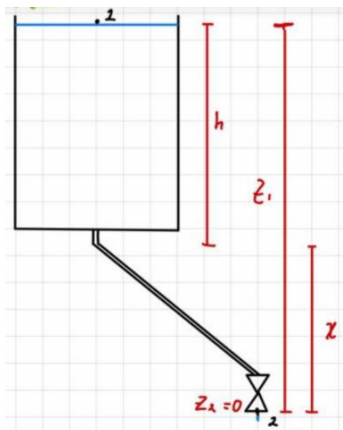


Imagen 50. Esquema del sistema de dispensado de agua.

Viendo la imagen anterior, ahora es posible continuar con otros parámetros de diseño necesarios, entre ellos, la selección de una electroválvula que se acople a nuestras necesidades. Para ello, es preciso conocer la presión hidrostática del sistema a la altura de la válvula cuando está cerrada, dicho cálculo se presenta a continuación:

$$P_{\text{válvula}} = \rho g Z_{\text{válvula}} = 1000 \frac{\text{kg}}{\text{m}^3} * 9.81 \frac{\text{m}}{\text{s}^2} * (0.3 \text{ m} - 40 * 10^{-3} \text{ m}) = 2550.6 \text{ Pa}$$

Como es posible observar en cálculo anterior, este es un sistema de baja presión hidrostática, por lo que se requiere una válvula que no tenga una presión mínima de trabajo y cuyo diámetro interno sea ¼ de pulgada.

Dicho lo anterior y tras exhaustiva búsqueda, la única alternativa viable encontrada fue la electroválvula 2/2 vías N.C. de acción directa 4030, con diámetro interno de ¼” cuyo fabricante es Genebre. En seguida, la imagen 51 de la válvula en cuestión.

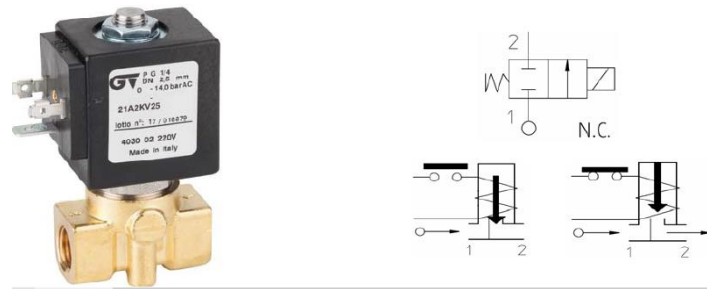


Imagen 51. Electroválvula seleccionada con su respectivo circuito neumático. [34]

A continuación, se exponen todos los cálculos analíticos para la validación del sistema:

Tiempo deseado de llenado: 5 s.

Cantidad de agua a dispensar por ingesta de pastilla: 100 ml.

Peso específico del agua: $9810 \frac{\text{kg}}{\text{kg}^2\text{s}^2}$

$1 \text{ ml} = 1 * 10^{-6} \text{ m}^3$

Caudal deseado:

$$Q_{\text{deseado}} = \frac{20 \text{ ml}}{\text{s}} \left[\frac{1 * 10^{-6} \text{ m}^3}{1 \text{ ml}} \right] = \left[\frac{20 * 10^{-6} \text{ m}^3}{\text{s}} \right]$$

Altura estimada desde el nivel superior:

$$Z_{1D} = 30 \text{ cm} = 0.3 \text{ m}$$

Cálculo del coeficiente de caudal

El coeficiente de caudal Kv que es proporcionada por la hoja de datos de la electroválvula que se usará, es de 3.2 L/min, se convierte a m³/h.



$$Kv = 3.2 \frac{L}{\text{min}} \left(\frac{60 \text{ min}}{1 \text{ h}} \right) \left(\frac{1 \text{ m}^3}{1000 \text{ L}} \right) = 0.192 \frac{\text{m}^3}{\text{h}}$$

Dado que existe una relación entre el coeficiente de caudal Kv y el coeficiente de resistencia K, esta mediante la siguiente ecuación.

$$K = \frac{d^4}{Kv^2} (0.001599)$$

Donde:

d^4 = Diámetro interior de la sección transversal característica. Para las válvulas es el diámetro nominal. Este valor dado en milímetros.

Ahora, se sustituyen los valores, teniendo el valor del diámetro interior que es de $\frac{1}{4}$ ", el valor de Kv. Por lo que se obtiene lo siguiente.

$$K = \frac{(6.35 * 10^{-3} \text{ m})^4}{\left(0.192 \frac{\text{m}^3}{\text{h}}\right)^2} (0.001599) = 7.0524 * 10^{-11}$$

Otros datos conocidos para el análisis:

$$P_1 = P_2 = 0$$

$$P_2 = 0$$

$$V_1 = 0 \frac{\text{m}}{\text{s}}$$

$$d_{\text{tubería}} = \frac{1}{4} \text{ in} = 6.35 * 10^{-3} \text{ m}$$

Se asume que debido a que la válvula es pequeña y la longitud de las tuberías también son muy pequeñas, la velocidad de salida del agua es constante a lo largo de la tubería.

Conociendo el caudal para el tiempo mínimo de llenado propuesto en el diseño, es posible calcular cual sería la dimensión mínima de la altura Z_1 para alcanzar la velocidad requerida para el caudal propuesto, para ello, primero:

De la fórmula del caudal:

$$Q = VA \therefore V_2 = \frac{Q_{\text{deseado}}}{A_{\text{tubería}}} = \frac{20 * 10^{-6} \text{ m}^3/\text{s}}{\frac{\pi d_{\text{tubería}}^2}{4}} = \frac{20 * 10^{-6} \text{ m}^3/\text{s}}{\frac{\pi (6.35 * 10^{-3} \text{ m})^2}{4}} = 0.631528 \frac{\text{m}}{\text{s}}$$

Ahora, para poder conocer la altura se procede a emplear el teorema de Bernoulli, aplicando el análisis entre los puntos 1 y 2 del esquema.

Utilizando la fórmula de Bernoulli:



$$\frac{1}{2}\rho V_1^2 + \rho g Z_1 + P_1 = \frac{1}{2}\rho V_2^2 + \rho g Z_2 + P_2 + \Delta P$$

$$\rho g Z_1 = \frac{1}{2}\rho V_2^2 + \Delta P$$

$$Z_1 = \frac{1}{2g} V_2^2 + \frac{\Delta P}{\rho g} = \frac{0.631528 \frac{m}{s}}{2 * 9.81 \frac{m}{s^2}} + \frac{k * 0.631528 \frac{m}{s}}{2 * 9.81 \frac{m}{s^2}}$$

$$Z_1 = \frac{0.631528 \frac{m}{s}}{2 * 9.81 \frac{m}{s^2}} + \frac{7.0524 * 10^{-11} * 0.631528 \frac{m}{s}}{2 * 9.81 \frac{m}{s^2}} = 0.03218 m$$

Se ha corroborado que se cumple la condición de que la altura de diseño es mayor que la calculada, es decir:

$$Z_{1D} > Z_1$$

Por lo tanto, se puede establecer que la dimensión de diseño es más que adecuada.

Ahora, se procede a calcular cual sería la velocidad de salida V_2 del sistema de dispensado cuando el tanque está lleno y se tiene como dato la altura de diseño. Para ello, como datos conocidos se tienen:

$$P_1 = P_2 = 0 \text{ Pa}$$

$$P_2 = 0$$

$$V_1 = 0 \frac{m}{s}$$

$$d_{tuberia} = \frac{1}{4} in = 6.35 * 10^{-3} m$$

$$Z_1 = 0.3 m$$

$$Z_2 = 0 m$$

Ahora, se aplica nuevamente el análisis con teorema de Bernoulli entre los puntos 1 y 2 del esquema del sistema

$$\frac{1}{2}\rho V_1^2 + \rho g Z_1 + P_1 = \frac{1}{2}\rho V_2^2 + \rho g Z_2 + P_2 + \Delta P$$

$$\rho g Z_1 = \frac{1}{2}\rho V_2^2 + \Delta P$$

$$\rho g Z_1 = \frac{1}{2}\rho V_2^2 + \frac{1}{2}\rho K V_2^2 = \frac{1}{2}\rho V_2^2 (1 + k)$$



Hecho lo anterior, es posible despejar de la fórmula a la velocidad de salida de la siguiente forma:

$$V_2 = \sqrt{\frac{2gZ_1}{1+K}} = \sqrt{\frac{2 * 9.81 \frac{m}{s^2} * 0.3 m}{1 + 7.0524 * 10^{-11}}}$$

La velocidad máxima de salida entonces sería de:

$$V_2 = 2.4261 \frac{m}{s}$$

Ahora se determina el caudal final que se lograra con el sistema:

$$Q_{real} = V_2 A_{tuberia} = 2.4261 \frac{m}{s} * \frac{\pi(6.35 * 10^{-3} m)^2}{4} = 76.8329 * 10^{-6} \frac{m^3}{s}$$

El tiempo de llenado de 100 ml entonces correspondería a la siguiente expresión:

$$Tiempo\ minimo\ de\ llenado = \frac{100 * 10^{-6} m^3}{Q_{real}} = \frac{100 * 10^{-6} m^3}{76.8329 * 10^{-6} \frac{m^3}{s}} = 1.3015 s$$

El resultado, permite observar un tiempo de llenado de no más de 1 segundo por lo que se considera adecuado.

También es imprescindible el cálculo del tiempo de llenado máximo que tendría el dispositivo para brindar al usuario los 100 mililitros de agua requeridos, dichas operaciones se muestran a continuación.

El sistema de dispensado de agua cuenta con un sensor para reabastecimiento del tanque cuando el nivel de líquido alcanza un cuarto de la altura máxima de llenado, es decir, la cantidad mínima de agua que se puede tener en el tanque es:

$$h_{minima} = \frac{h}{4} = \frac{0.2030 m}{4} = 0.05075 m$$

De tal forma que la altura mínima del suelo a la superficie de la columna de líquido sería:

$$Z_{1min} = Z_1 - (h - h_{minima}) = 0.3 - (0.1490 - 0.03725)m = 0.11175 m$$

El resto de los datos conocidos es el mismo:

$$P_1 = P_2 = 0 \text{ Pa}$$

$$P_2 = 0$$

$$V_1 = 0 \frac{m}{s}$$

$$d_{tuberia} = \frac{1}{4} in = 6.35 * 10^{-3} m$$



$$Z_1 = Z_{1min} = 0.11175 \text{ m}$$

$$Z_2 = 0 \text{ m}$$

Se aplica una vez más el análisis con teorema de Bernoulli entre los puntos 1 y 2 del esquema del sistema

$$\frac{1}{2}\rho V_1^2 + \rho g Z_1 + P_1 = \frac{1}{2}\rho V_2^2 + \rho g Z_2 + P_2 + \Delta P$$

$$\rho g Z_1 = \frac{1}{2}\rho V_2^2 + \Delta P$$

$$\rho g Z_1 = \frac{1}{2}\rho V_2^2 + \frac{1}{2}\rho K V_2^2 = \frac{1}{2}\rho V_2^2 (1 + k)$$

Hecho lo anterior, es posible despejar de la formula a la velocidad de salida de la siguiente forma:

$$V_2 = \sqrt{\frac{2gZ_1}{1+K}} = \sqrt{\frac{2 * 9.81 \frac{m}{s^2} * 0.11175 \text{ m}}{1 + 7.0524 * 10^{-11}}}$$

La velocidad máxima de salida entonces sería de:

$$V_2 = 1.4807 \frac{m}{s}$$

Ahora se determina el caudal final que se lograra con el sistema:

$$Q_{real} = V_2 A_{tuberia} = 1.70260 \frac{m}{s} * \frac{\pi (6.35 * 10^{-3} m)^2}{4} = 46.6893 * 10^{-6} \frac{m^3}{s}$$

El tiempo de llenado de 100 ml entonces correspondería a la siguiente expresión:

$$Tiempo \text{ máximo de llenado} = \frac{100 * 10^{-6} m^3}{Q_{real}} = \frac{100 * 10^{-6} m^3}{53.92008 * 10^{-6} \frac{m^3}{s}} = 2.1325 \text{ s}$$

El resultado, permite observar un tiempo de llenado máximo de aproximadamente dos segundos, por lo que se considera adecuado.

Lo que procede para asegurar el correcto dispensado de agua en el vaso correspondiente, se propone utilizar un sensor que sea capaz de medir el nivel del agua dentro del vaso, dicho sensor no debe tener contacto con el agua ni con la parte interior del vaso, por lo que, se sugieren sensores de nivel del agua sin contacto. Estos se muestran en las siguientes imágenes 52, 53 y 54.



Imagen 52. Sensor de nivel de agua sin contacto, XKC-Y25. [35]



Imagen 53. Sensor de nivel de agua sin contacto, XKC-Y26-NPN. [36]



Imagen 54. Sensor de nivel de agua sin contacto, tipo switch. [37]

De los sensores mostrados en las imágenes anteriores, se puede ver que son de diferentes formas, y sus características específicas diversas. Las características que se tomaron a consideración para realizar la respectiva evaluación de estos componentes por criterios ponderados son las siguientes:

- Tensión nominal
- Corriente nominal
- Distancia para sensado
- Valor monetario
- Dimensiones del sensor

De estas, las que tienen mayor peso son la de distancia para sensado (sensibilidad), el tamaño y la tensión nominal; estas porque lo que se busca es que tenga una distancia de sensado adecuada para no tener que mantener el vaso tan cerca del sensor, además de que debe ser de un tamaño pequeño que sea discreto al diseño del dispensador y la tensión de trabajo no supere los 24V para evitar componentes que necesiten más potencia de la esperada. En la siguiente tabla 14, se pueden ver estos valores de cada sensor para poder realizar una comparación entre ellos.

	XKC-Y25 Sensor nivel de agua sin contacto	XKC-Y26-NPN Sensor nivel de agua sin contacto	Sensor de nivel de líquidos sin contacto - tipo switch
Tensión	5V - 24V	5V - 24V	5V
Corriente	5 mA	5 mA	25 uA
Sensibilidad	0 mm - 13 mm	20 mm	1-13 mm
Precio	\$201.58	\$510.50	\$533.00
Tamaño	28mm x 16.9mm	40mm x 34.5mm	10mm x 10 mm

Tabla 14. Características de interés de cada sensor de nivel de agua sin contacto.

Ahora en la siguiente tabla 15, muestra las calificaciones que cada sensor obtiene para cada uno de los criterios en cuestión.

Criterio	Peso	Descripción	Unidad	Opciones:					
				XKC-Y25 Sensor nivel de agua sin contacto		XKC-Y26-NPN Sensor nivel de agua sin contacto		Sensor de nivel de líquidos sin contacto - tipo switch	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Tensión	0.2	Tensión nominal	V	9	1.8	9	1.8	10	2
Corriente	0.15	Corriente nominal	A	8	1.2	8	1.2	9	1.35
Sensibilidad	0.3	Distancia para sensado	mm	8	2.4	10	3	7	2.1
Precio	0.15	Valor monetario	MXN	9	1.35	7	1.05	6	0.9
Tamaño	0.2	Dimensiones del sensor	mm	8	1.6	7	1.4	9	1.8
Total:	1			42	8.35	41	8.45	41	8.15

Tabla 15. Evaluación de los sensores de nivel de agua sin contacto, mediante criterios ponderados.

De la tabla 15, resulta con mejor calificación el sensor XKC-Y26-NPN como el elegido para ser el sensor que mida el nivel del agua al momento que el vaso comience a llenarse.

Para realizar la validación estructural de este contenedor, debemos primero tomar en consideración que un litro de agua es realmente una carga muy baja y, debido a que las paredes tienen un grosor de 3 milímetros, los esfuerzos y deformaciones resultan ser muy bajos.

Por otra parte, debido a que la superficie es curva y dicha curva es una media elipse, el cálculo de presión, esfuerzo y deformaciones resultarían muy complejos por lo que la validación estructural se realiza de forma computacional en 2 softwares, Ansys Workbench y Solidworks.

En ambos softwares, se emplea un análisis hidrostático aplicando una presión linealmente variable sobre las paredes del cuerpo del contenedor que se mojan y colocando restricciones fijas en la superficie inferior del contenedor y en las superficies inferiores del trapecio que quedan libres.

Los resultados solicitados fueron los análisis de esfuerzos y deformaciones del contenedor, cuyos resultados se exponen en las imágenes de a continuación. En la imagen 55 se muestra la distribución de la presión en el tanque mientras que las imágenes 56 y 57 muestran los resultados de deformación y esfuerzos respectivamente.

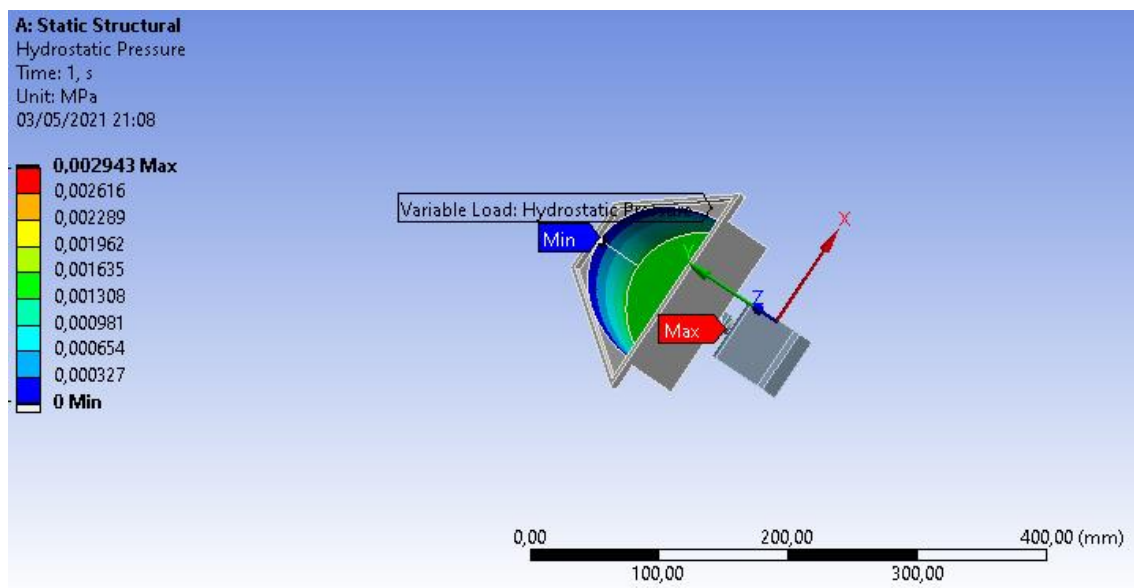


Imagen 55. Presión hidrostática aplicada al contenedor de agua.

De la imagen 55, se puede observar que la máxima presión hidrostática se encuentra justo en la entrada a la electroválvula, teniendo un valor de 0.002943 MPa como máxima de presión, y la región en azul es donde la presión 0 MPa, esto debido que es la superficie libre a la atmósfera del líquido.

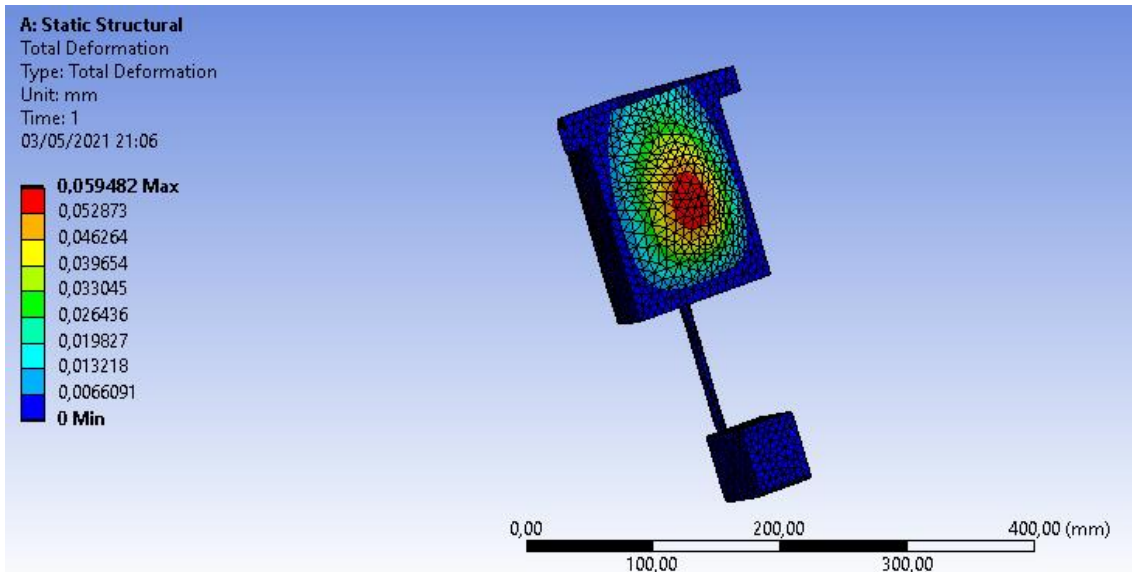


Imagen 56. Resultado de la deformación en el tanque.

En la imagen 56, se visualiza la deformación que sufriría el tanque debido al líquido que estará contenido en éste. La parte en color rojo es la deformación máxima debido a que ahí se concentrarían los esfuerzos, donde dicha deformación tendría un valor de 0.059482 mm aproximadamente en el centro de la pared plana del tanque, dicha deformación es mínima, por lo que el diseño la soportaría.

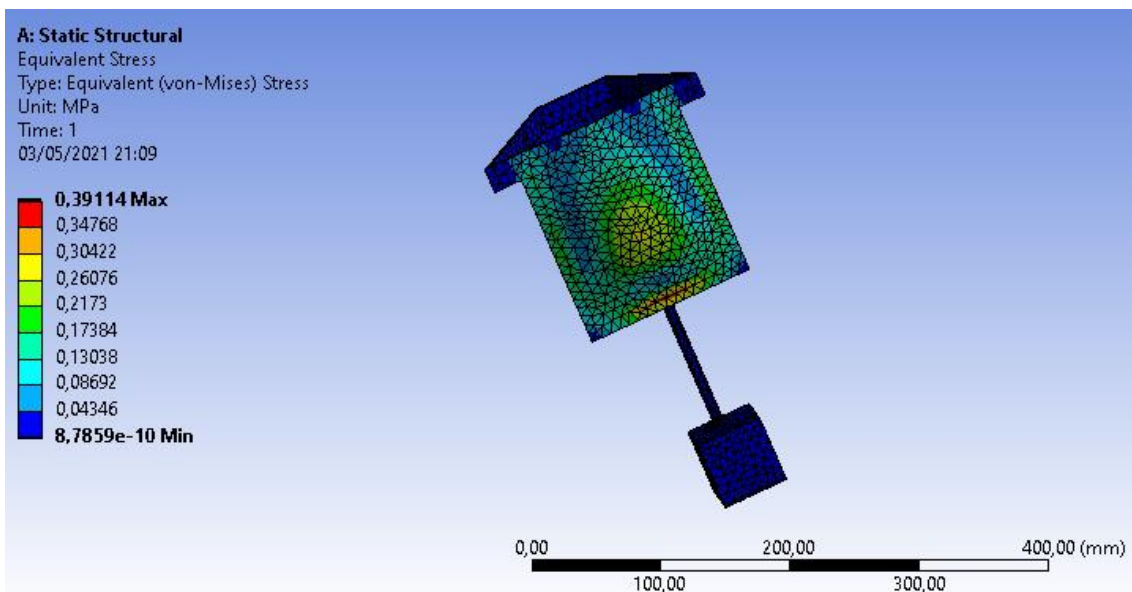


Imagen 57. Resultado del esfuerzo en el tanque.

Como se puede apreciar en la imagen 57, el esfuerzo máximo en color rojo se encuentra en la esquina del fondo del tanque de agua, que abarca una superficie pequeña en comparación a toda el área de la pared plana del tanque. Este esfuerzo tiene un valor de 0.39114 MPa y teniendo un esfuerzo mínimo de 8.7859×10^{-10} MPa que se representa en color azul, el cual abarca la mayoría del área del contenedor. Dichos esfuerzos son pequeños por lo que este diseño es capaz de soportarlos.

En la siguiente imagen 58, se puede apreciar el diseño final del contenedor de agua junto con su respectivo sistema de dispensado.

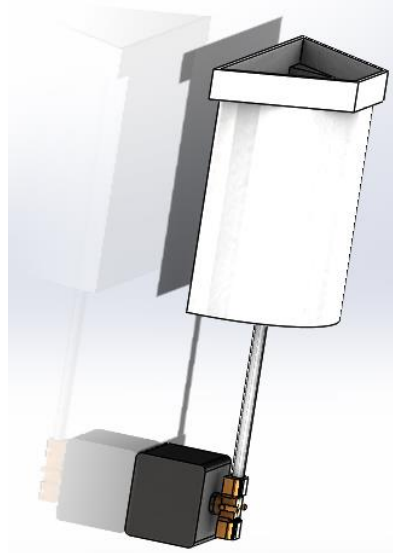


Imagen 58. Vista isométrica del contenedor con la válvula.

Actualización de sistema de dispensado de agua

- **Reemplazo del sensor XKX-Y26**

El sensor XKC-Y26 es poco comercializado en la Ciudad de México, además, su adquisición por internet implicaría costear gastos de envío adicionales, encareciendo aún más el dispositivo, sin embargo, existe un homólogo más comercial, el sensor XKC-Y25, el cual cuenta con casi las mismas características, con la diferencia de que es posible alimentarlo con 5 o 12 v y su rango de detección llega a los 11mm de separación del agua, sin embargo, dado que las paredes que separarán a este sensor de nivel libre de contacto son de tan solo 3 mm, y realiza la misma función que el propuesto anteriormente, se ha decidido usarlo como reemplazo para este proyecto.



Imagen 59. Sensor XKC-Y25-V con sus respectivas especificaciones.

En la imagen 59 podemos observar el sensor XKC-Y25, así como algunas de sus especificaciones.

- **Reemplazo de la electroválvula GENEBRE-4030**

Otro cambio que el equipo se ha visto en la necesidad de realizar es el de la electroválvula, esto, principalmente debido a que el fabricante de la marca propuesta ha cesado operaciones en México y solo opera actualmente en Brasil y España, por lo que continuar con dicha válvula implicaría un proceso de envío largo y costoso y aunque sería una importación. Sin embargo, se ha encontrado una electroválvula de la marca Tecsol, la cual, cumple con el requisito principal de que su diafragma se abra y permita el paso de agua sin la necesidad una presión hidrostática mínima.



Imagen 60. Nueva electroválvula 2/2 N. C. de 1/4".

La imagen 60 muestra la nueva electroválvula a emplear en este proyecto, la cual cabe resaltar, que también cumple con la característica de ser de un orificio de $\frac{1}{4}$ de pulgada y ser de alimentación de corriente directa, sin embargo, ésta se activa con 12V. La matrícula de este nuevo componente es Tecsol-Q2BBC02, el resto de sus especificaciones se encuentran adjuntas en los anexos de este documento.

Sistema de Dispensado de Medicamentos

Se comienza ha propuesto una estructura que pueda mover los contenedores de medicamentos y que pueda soportarlos fijamente, dicha estructura se ha denominado “revolver”. Para ello, se procede a mostrar en las imágenes 61 y 62 el isométrico de la pieza mencionada.

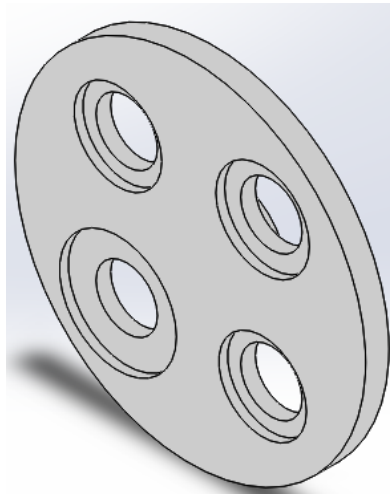


Imagen 61. Vista isométrica superior del revolver.

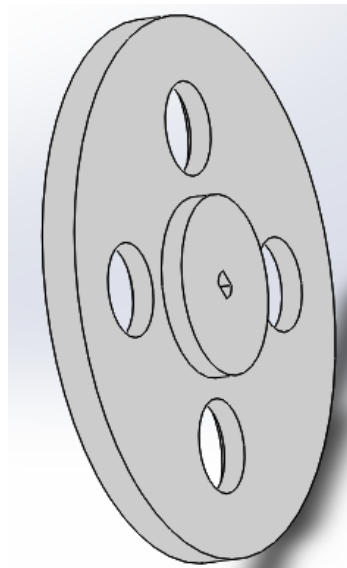


Imagen 62. Vista isométrica inferior del revolver.

Como se puede apreciar en las imágenes anteriores, tiene cuatro espacios para los diferentes contenedores que se han diseñado y los cuales serán empotrados en los orificios. Además, de que girará por medio de un actuador, esto debido a que, en la parte inferior, como se ve en la imagen 62, hay hueco para unir a un eje de un motor.

Ahora se proporcionan las dimensiones de esta pieza:

Diámetro del revolver: 140 mm.

Espesor del revolver: 8 mm.

Espesor de la parte inferior del revolver para eje del motor: 5 mm.

Diámetro de la parte inferior del revolver para eje del motor: 40 mm.

Dado que se requiere que gire para que realice el dispensado de los medicamentos, se debe calcular el torque con el que se empezaría a moverse este revolver. Para esto, mediante SolidWorks, se obtiene la masa del revolver y de los diferentes contenedores, donde a estos últimos se les agrega la masa esperada cuando estén al máximo de capacidad de

pastillas. La masa del revolver indicada por el programa es de 103.00 gr, como se puede observar en la imagen 63.

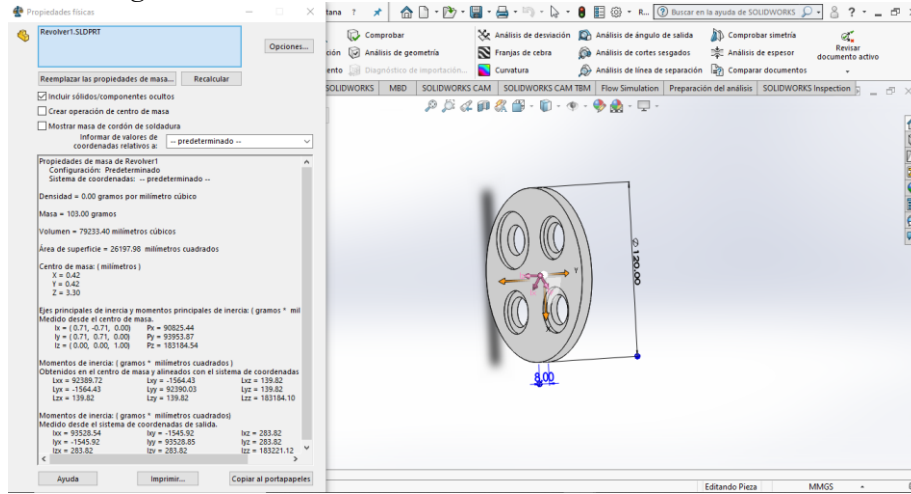


Imagen 63. Masa indicada del revolver con SolidWorks.

A continuación, se muestran las masas de los contenedores 1, 2 y 3 dadas por SolidWorks que son: 166.07 gr, 137.29 gr y 131.20 gr, respectivamente, que se ven en las imágenes 64, 65 y 66.

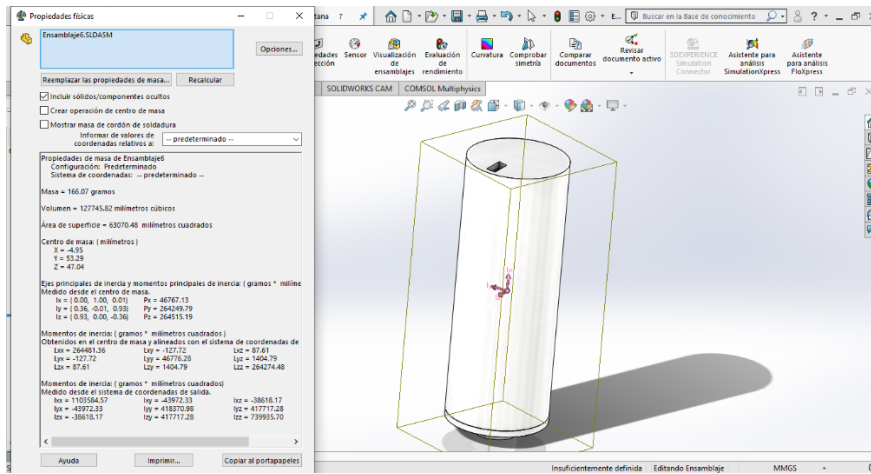


Imagen 64. Masa indicada del contenedor 1 con SolidWorks.

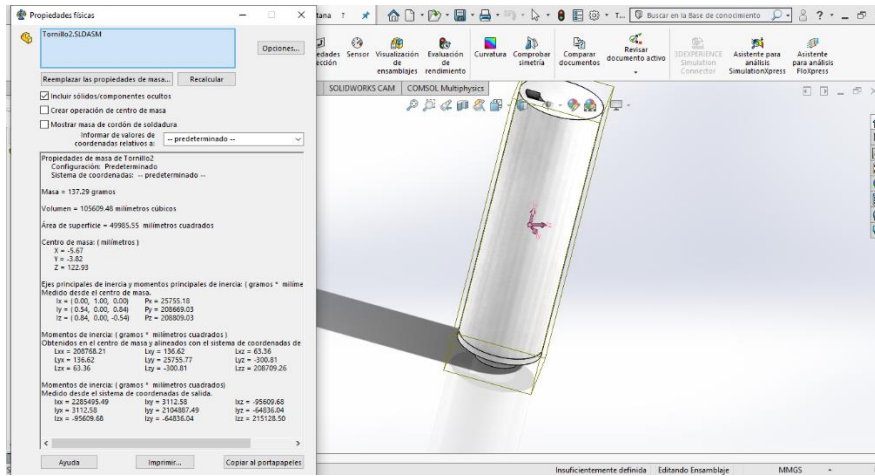


Imagen 65. Masa indicada del contenedor 2 con SolidWorks.

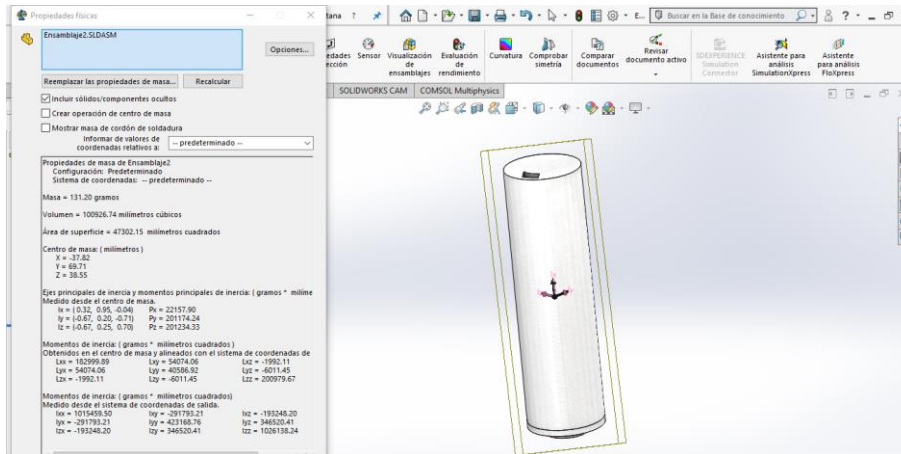


Imagen 66. Masa indicada del contenedor 3 con SolidWorks.

La masa obtenida para cada contenedor cuando están en su máxima capacidad de pastillas 16.5 gr, esto teniendo en cuenta que el peso máximo de un comprimido o cápsula es de 1.1 gr. El valor obtenido de 16.5 gr fue a partir de que cada contenedor tiene la capacidad para 15 pastillas. Ahora, se multiplica por 4 que son los contenedores que se colocarán en el revolver, y da como resultado 66 gr distribuidos en los cuatro contenedores.

Se ensambla el sistema como se aprecia en las imágenes 67 y 68 para obtener el momento de inercia respecto al eje por donde pasara el motor.

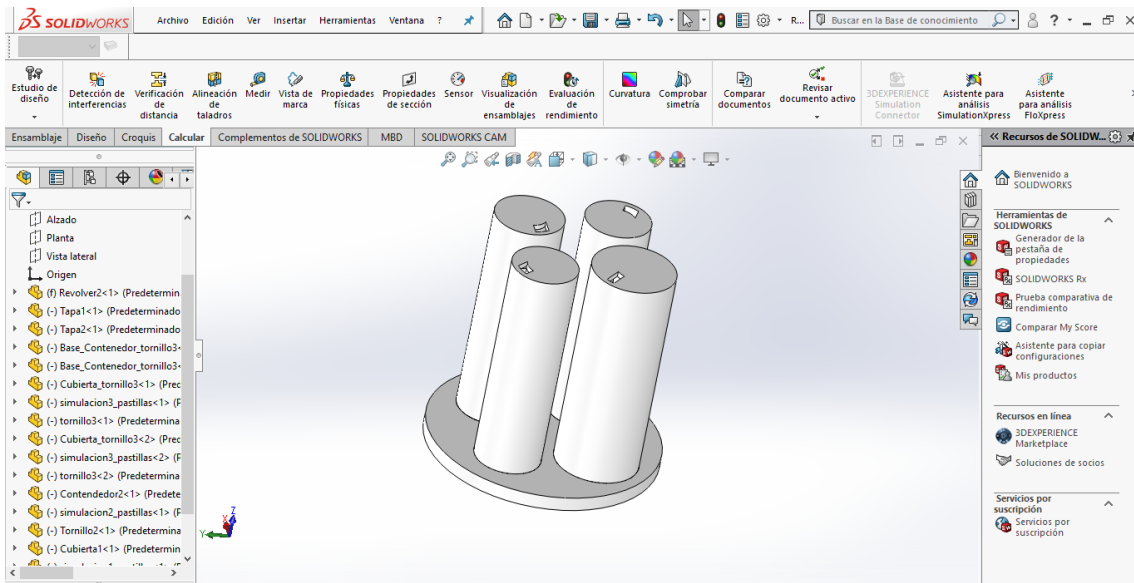


Imagen 67. Sistema ensamblado con 3 tipos diferentes de contenedor.

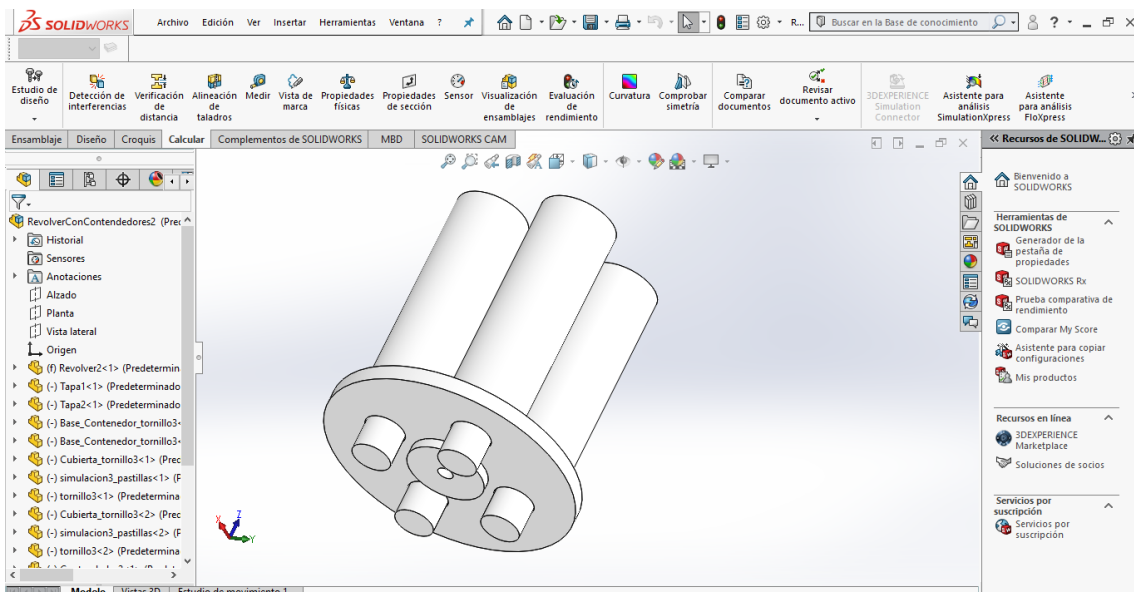


Imagen 68. Vista inferior del ensamblaje del revolver y los contenedores.

Una vez establecido el sistema de coordenadas para el cálculo de los momentos de inercia se obtuvieron los siguientes valores mostrados en la imagen 69.

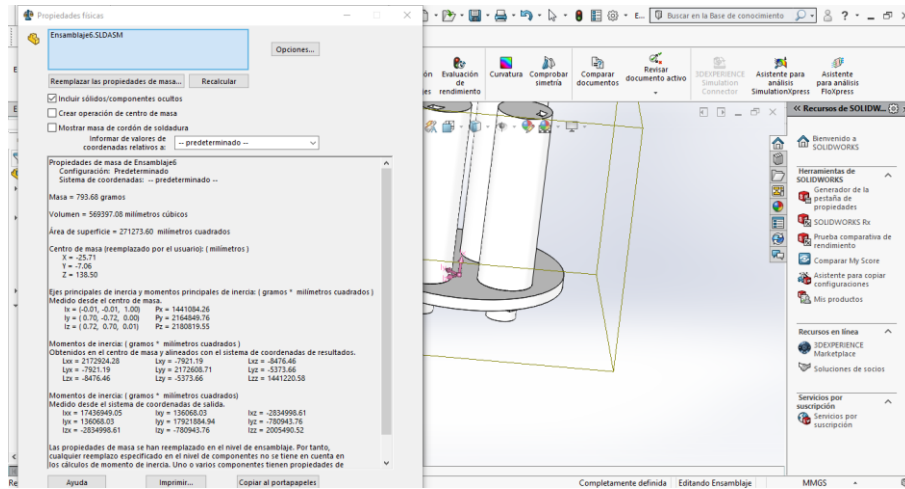


Imagen 69. Propiedades físicas del ensamble del revolver y los contenedores.

Para la validación numérica del revolver se realizaron los cálculos de esfuerzos en Workbench para comprobar que la pieza puede soportar las cargas de los contenedores sin sufrir una deformación considerable, estas se visualizan en las imágenes 70, 71 y 72.

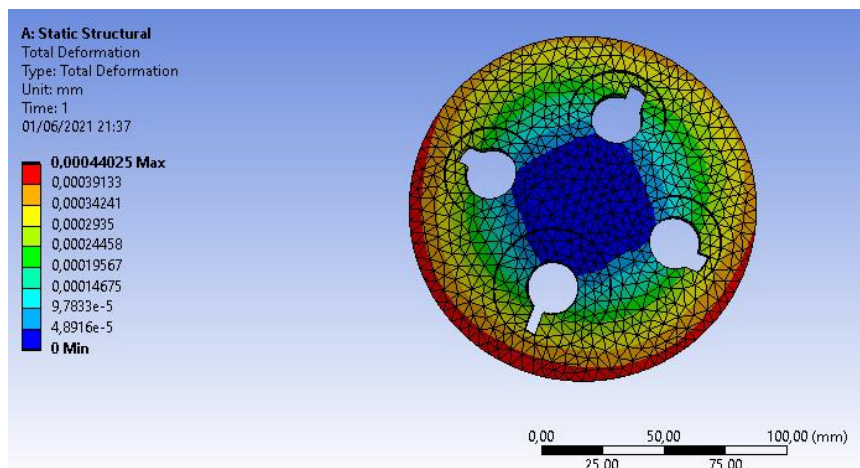


Imagen 70. Resultado del análisis de deformación para el revolver.

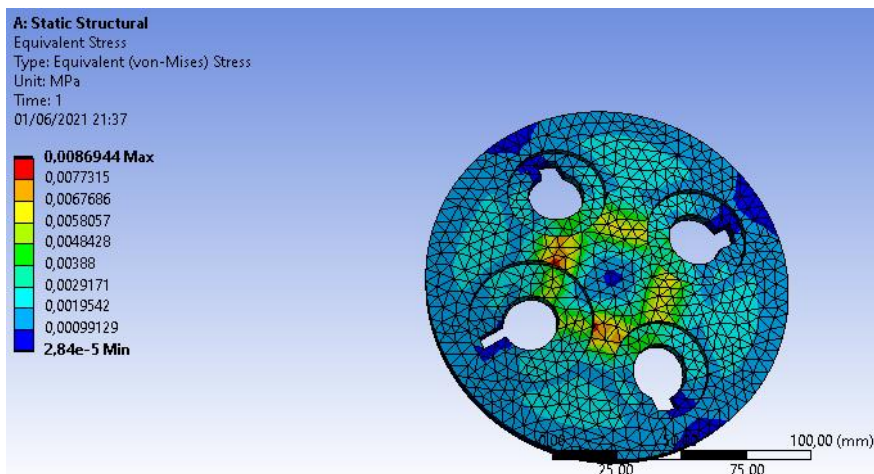


Imagen 71. Resultado del análisis de esfuerzo de Von mises para el revolver.

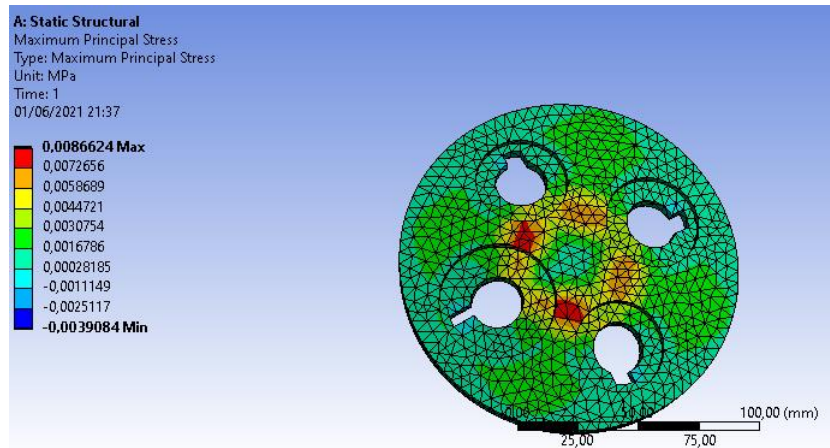


Imagen 72. Resultado del análisis de esfuerzos máximos para el revolver.

Como se ve en las imágenes anteriores, el valor de la deformación para la pieza es de 0.00044 mm, lo cual es un valor muy pequeño y por consiguiente aceptable. De la misma manera los esfuerzos principales no son muy grandes al ser la carga muy pequeña. Para el esfuerzo de Von Mises se obtiene un valor de 0.0086 MPa mientras que para el esfuerzo máximo principal 0.0086 MPa.

Al modelo anterior se le agrego un sistema de compuertas en la parte inferior del revolver que será parte del sistema de salida de las pastillas, en donde además se acoplaran unos pequeños resortes que regresaran la compuerta a la posición original. El detalle de diseño se presenta en la imagen 73.

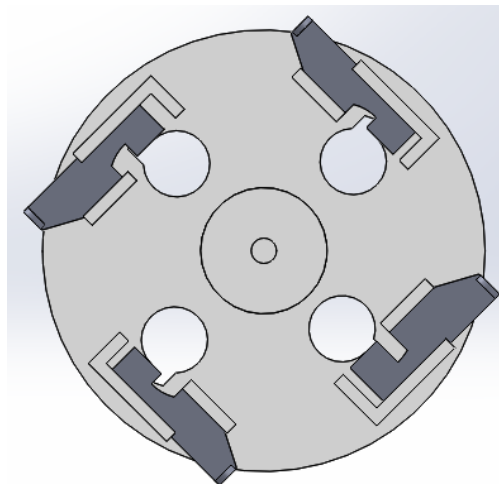


Imagen 73. Vista inferior del revolver para el sistema de salida de las pastillas.

Una vez obtenidos estos datos se realizó el proceso de selección del motor para mover el sistema del revolver, tomando en cuenta que de este se pudiese controlar su posición angular de una manera altamente precisa. Dicho lo anterior, se opta por proponer como actuador un motor a pasos.

Se proponen como actuadores a los motores a pasos de corriente directa que se muestran en las imágenes 74, 75 y 76.

Motor 28BYJ-48:



Imagen 74. motor 28BYJ-48. 5 VCD, 40 mA/fase, tamaño: 35 mm, cuenta con driver, torque: 0.02942 Nm [38].

Motor PM55L-048:



Imagen 75. motor PM55L-048. 24 VCD, 600 mA/fase, tamaño: 65 mm, cuenta con driver, torque: 0.11376 Nm [39].

Motor NEMA17:



Imagen 76. motor NEMA17. 12 VCD, 1.7 A, tamaño: 42 mm, cuenta con driver, torque: 0.3923 Nm [40].

Los motores propuestos, fueron evaluados por medio de criterios ponderados para determinar cual resulta el más adecuado para nuestra aplicación usando la siguiente tabla 16.

Criterio	Peso	Descripción	Unidad	Opciones:					
				PM55L-048		28BYJ-48		NEMA17	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Tension	0.15	Tensión nominal	V	9	1.35	6	0.9	7	1.05
Corriente	0.15	Corriente nominal	A	7	1.05	9	1.35	6	0.9
Tamaño	0.15	Dimensiones del motor	mm	7	1.05	9	1.35	8	1.2
Driver	0.2	Grado de complejidad de implementación	SI/NO	9	1.8	10	2	9	1.8
Torque	0.2	Torque de Pull-In	Nm	9	1.8	6	1.2	10	2
Precio	0.15	Valor monetario	MXN	8	1.2	9	1.35	6	0.9
Total:	1			49	8.25	49	8.15	46	7.85

Tabla 16. Evaluación de los motores a pasos para el revolver, mediante criterios ponderados.

De acuerdo a las especificaciones del fabricante, el motor mejor evaluado cuenta con un ángulo de zancada o paso (Stride Angle) de 7.5° por paso. Asu vez, las especificaciones establecen que el motor cuenta con 48 pasos por revolución y no cuenta con caja de reducción de velocidad, es decir no tiene un ratio de reducción de velocidad.

Esto a su vez nos sugiere que:

$$Relacion\ Desplazamiento = \frac{360^\circ}{\frac{1\ rev}{7.5^\circ}} = 48 \frac{paso}{rev}$$

Debido a que no tenemos un ratio de reducción de velocidad:

$$Relacion\ Desplazamiento\ Final = 48 \frac{paso}{revolucion} = 7.5 \frac{grado}{paso}$$

Para calcular el torque requerido para mover el revolver junto con los contenedores y suponiendo que cada uno de estos está a su máxima capacidad de pastillas, se ocupará la siguiente ecuación.

$$T = I\alpha$$

Donde:

T: Es el torque necesario para mover el revolver y contenedores.

I: Es el momento de inercia respecto al eje de rotación.

α : Es la aceleración rotacional del motor.

Partiendo de la ecuación anterior, se propone una velocidad angular donde se conoce el desplazamiento del motor y se elige un tiempo para que recorra ese desplazamiento angular. Esto se representa en la siguiente ecuación.



$$\omega = \frac{\theta}{t}$$

Donde:

ω : Es la velocidad angular a la que debería ir el motor.

θ : Es el desplazamiento angular que debe recorrer el eje de salida del motor.

t : Es el tiempo que tarda en recorrer el desplazamiento angular.

El desplazamiento deseado debido a que los contenedores se encuentran dentro de dos líneas perpendiculares, semejantes a los ejes de un plano cartesiano, donde cada uno de los contenedores están en los extremos de dichos ejes. Es por esta razón que se desea un desplazamiento de 90° para mantener esa disposición de los contenedores.

Referente al tiempo con el que se va a cumplir el desplazamiento mencionado, se propone un tiempo de 5 segundos. Ahora teniendo estos datos, se sustituyen en la ecuación anterior.

$$\omega = \frac{90^\circ}{5 \text{ s}} = \frac{\frac{\pi}{2} \text{ rad}}{5 \text{ s}} = \frac{\pi \text{ rad}}{10 \text{ s}}$$

Teniendo ahora la velocidad angular, se hace el cálculo de la aceleración angular, que se muestra con la siguiente ecuación.

$$\alpha = \frac{\omega_f - \omega_0}{t}$$

Donde:

α : Es la aceleración angular.

ω_f : Es la velocidad angular final que es la que se calculó de $\frac{\pi \text{ rad}}{10 \text{ s}}$.

ω_0 : Es la velocidad angular inicial, que en este caso es cero debido a que parte del reposo.

t : Es el tiempo en el que alcanza dicha velocidad final, el cual es de 5 s.

Por lo que, se sustituyen los valores y se obtiene la siguiente aceleración angular.

$$\alpha = \frac{\frac{\pi \text{ rad}}{10 \text{ s}}}{5 \text{ s}} = \frac{\pi \text{ rad}}{50 \text{ s}^2}$$

Considerando la aceleración calculada, sabemos que la velocidad de operación del motor sería:

$$\omega_{rpm} = \frac{\pi \text{ rad}}{10 \text{ s}} \left(\frac{60 \text{ s}}{1 \text{ min}} \right) \left(\frac{1 \text{ rev}}{2\pi \text{ rad}} \right) = 3 \text{ rpm}$$



Empleando las herramientas virtuales de simulación y medición de Solid Works, se obtuvo el momento de inercia en el eje Z de la pieza, el cual se muestra en la siguiente expresión:

$$I_z = 1441084.26 \text{ g} \cdot \text{mm}^2 \left(\frac{1 \text{ cm}^2}{(10 \text{ mm})^2} \right) = 14410.8426 \text{ g} \cdot \text{cm}^2$$

Con este dato se tiene lo necesario para calcular el torque de inercia que deberá vencer el motor a pasos en el arranque, el cual se calcula a continuación:

$$T_i = (14410.8426 \text{ g} \cdot \text{cm}^2) \left(\frac{\pi \text{ rad}}{50 \text{ s}^2} \right) = 902.459 \text{ gf} \cdot \text{cm} = 902.46 * 10^{-4} \text{ Nm}$$

Una vez más, se recurre a la hoja de datos del fabricante, pero esta ocasión para observar los puntos de operación del motor y determinar si es posible emplearlo en esta aplicación específica y de ser así bajo qué condiciones.

Como es posible observar en la hoja de datos del motor a pasos seleccionado, éste es un bipolar, tiene diferentes puntos de operación dependiendo de la frecuencia a la cual se envían los pulsos para hacer girar el motor.

Se sabe que torque de Pull-In es el torque máximo al cual el motor gira al arrancar, mientras que el de Pull-Out es el torque máximo que puede entregar el motor una vez ya se encuentra girando, por lo cual, buscaremos que la frecuencia de pulsos por segundo converja con un torque de Pull-In que sea superior al torque inercial calculado previamente, es decir, debe cumplirse que:

$$T_{\text{Pull-In}} > T_i$$

Dicho lo anterior, vemos que la condición se cumple para el rango frecuencias:

$$f_{\text{pps_mpp}} = [200,300,400] \text{ Hz}$$

En las cuales se tienen los respectivos torques de Pull-In:

$$T_{\text{Pull-In}} = [1137.6; 994.4; 764.9] * 10^{-4} \text{ Nm} > T_i$$

Estos torques representan lo máximo que se puede exigir al motor en el arranque y no debe ser superado ya que de lo contrario perdería sincronía.

Con estos cálculos queda comprobado que el motor a pasos seleccionado es adecuado para la aplicación requerida.

Para su implementación, se empleará un driver A4988, el cual está diseñado para controlar motores bipolares a paso. Éste, nos permite establecer control de giro y dirección sobre el motor a partir de su conexión.

El integrado, permite un paso de corriente en sus terminales a las bobinas del motor de hasta 2 A, la cual es prácticamente el doble de la corriente máxima por fase del motor a pasos seleccionado, dato que puede ser corroborado en las hojas de datos de su fabricante.



Este circuito integrado, requiere de una tensión de alimentación de 3.3 a 5 V y adicionalmente la conexión de la fuente para el encendido del motor.

Para lograr tener control sobre el motor, este driver implementa 2 conexiones, una es el paso del motor mientras que la otra es la dirección de giro del motor cuyos nombres en la hoja de datos son STEP y DIR, cada una deberá ser conectada a un pin digital GPIO de la tarjeta de control.

Finalmente, este driver debe ser caracterizado para el motor al que se va a aplicar, para ello, debemos considerar la ecuación del fabricante:

$$I_{MAX\ driver} = V_{refDriver} / 8 * R_s$$

$$V_{refDriver} = I_{MASdriver} * 8 * R_s$$

De donde:

$V_{refDriver}$: es la tensión a la que deberá ajustarse el preset regulador de corriente del driver para garantizar proveer una corriente superior a la requerida por las fases del motor.

$I_{MAX\ driver}$: Corriente máxima que entregará el driver en los pines de salida a las bobinas del motor dada la tensión de preset.

$R_s = 0.1 \Omega$: Resistencia de salida de los pines del driver.

Consultando la hoja de datos del motor PM55L-048, se observa que la corriente por fase que demanda es de 800 mA, entonces, interesa extraer al menos esa cantidad de corriente del driver, por lo que se propone como $I_{MAX\ driver} = 1.2 A$, con la intención de que si se demanda mayor corriente debido al arranque con carga en el motor el driver no interrumpa su funcionamiento. Entonces:

$$V_{refDriver} = 1.2 A * 8 * (0.1 \Omega) = 0.96 V$$

De manera que cuando se realice la conexión, antes habrá que fijar el preset del driver a 0.96 V.

La conexión requerida para implementar el motor y driver en este proyecto se muestra en los apéndices de este documento, en la página 251.

Actualización de diseño del sistema de dispensado de medicamentos

Las actualizaciones al sistema de dispensado son principalmente hacia las piezas encargadas de almacenar los medicamentos y algunas de las encargadas de su organización y dispensado.

Principalmente, estas actualizaciones son mejoras a las piezas previamente diseñadas, es decir, su forma es mayoritariamente la misma y solamente se han añadido características que tienen como finalidad mejorar su funcionamiento o cinemática.

- *Compartimentos de almacenamiento*

En éstos, se ha añadido un par de muescas las cuales servirán como mecanismo de ensamble y sujeción con el revolver donde se concentran. Esta nueva característica puede observarse en las imágenes que se presentan a continuación.

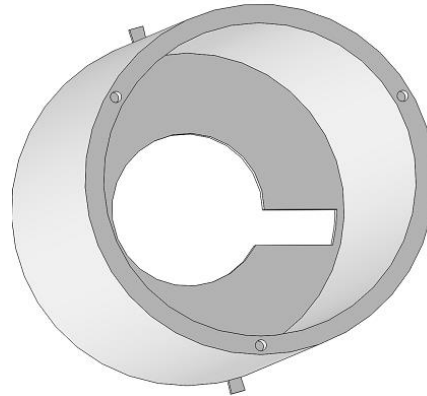


Imagen 77. Compartimento 1 de almacenamiento de pastillas.

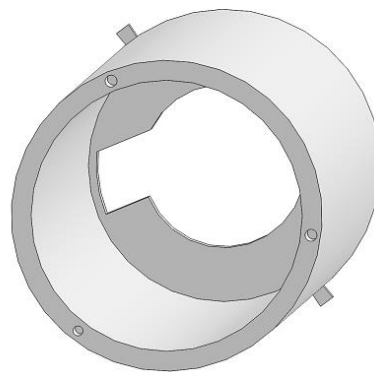


Imagen 78. Compartimento 2 de almacenamiento de pastillas.

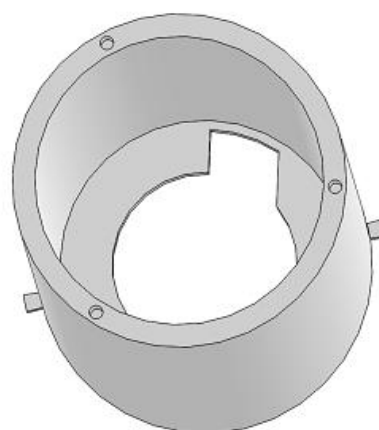


Imagen 79. Compartimento 3 de almacenamiento de pastillas.

Adicionalmente, las imágenes 77, 78 y 79 nos permiten observar otra de las características añadidas, la cual consiste en una serie de tres barrenos repartidos cada uno a 120° en la parte superior del compartimento, su finalidad es permitir el ensamble de sus respectivas tapas.

Dimensiones de las muescas:

Compartimento 1

Base menor: 1.72 mm

Base mayor: 1.87 mm

Laterales: 1.85 mm

Espesor: 1.8 mm

Compartimento 2

Base menor: 1.7 mm

Base mayor: 1.88 mm

Laterales: 1.85 mm

Espesor: 1.8 mm

*Compartimento 3

Base menor: 1.7 mm

Base mayor: 1.88 mm

Laterales: 1.85 mm

Espesor: 1.8 mm

Dimensiones de los barrenos:

Para todos los compartimentos

Barreno: 5 mm

Diámetro: 1.5 mm

Basados en los resultados de las pruebas de impresión relatadas en la parte de Implementación, se decidió modificar las dimensiones de las muescas añadidas con la finalidad de que estas sean impresas en mejor calidad y sean más resistentes, por lo que a continuación se muestran en la imagen 80, las nuevas muescas para ensamble en cada contenedor, así como sus nuevas dimensiones.

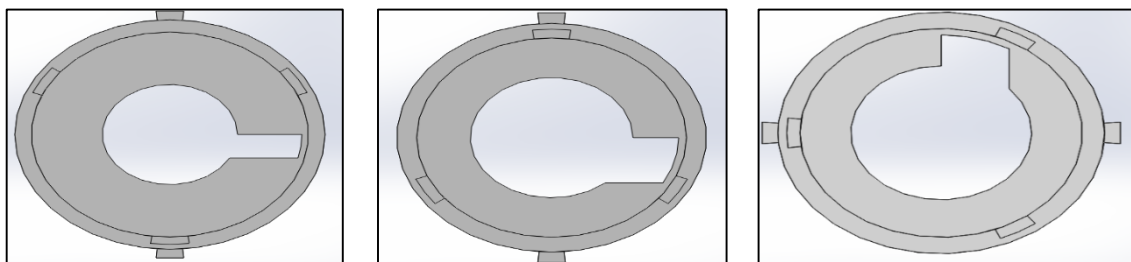


Imagen 80. CAD de los compartimentos 1, 2 y 3, respectivamente, estos son los modelos actuales.

Dimensiones de las muescas:

Compartimento 1 (imagen 80, la de la izquierda)

Base menor: 1.72 mm

Base mayor: 1.87 mm



Laterales: 1.85 mm

Espesor: 1.8 mm

Compartimento 2 (imagen 80, la de en medio)

Base menor: 1.7 mm

Base mayor: 1.88 mm

Laterales: 1.85 mm

Espesor: 1.8 mm

Compartimento 3 (imagen 80, la de la derecha)

Base menor: 1.7 mm

Base mayor: 1.88 mm

Laterales: 1.85 mm

Espesor: 1.8 mm

La imagen 80, a su vez permite observar que se ha modificado el mecanismo de unión para las tapas, pues antes eran barrenos, pero dado que estos resultaban en barrenos de mala calidad con diámetros internos inexactos y los ejes muy frágiles, se apuesta ahora por un ensamble a partir de pestañas y ranuras, permitiendo dimensiones más aptas dada la resolución de la impresora. Las dimensiones de las ranuras en los contenedores se muestran a continuación.

Dimensiones de la ranura:

Para todos los compartimentos

Profundidad: 5 mm

Espesor: 1.43 mm

Longitud de arco: 4.3 mm

Por otra parte, debido al cambio de espesor del revolver, fue necesario incrementar la altura del “Contenedor 1” debido a que este se ensambla a una mayor profundidad que el resto, esta condición se produce debido a que el largo de las pastillas redondas es menor que el de las pastillas alargadas (capsulas y tabletas) y se busca que en la ranura de salida del revolver solo quepa una pastilla en posición vertical, para lograrlo, el contenedor debe meterse más en el revolver compensando la diferencia de longitudes, e incrementa su tamaño para compensar la pérdida de altura. Su altura final es ahora de 129 mm, una diferencia de 6 mm respecto de su versión anterior. Los demás contenedores mantienen sus dimensiones pese a la variación de longitud ya que esta es compensada en la parte libre de cuerda de tornillo de estos y dicha sección de los tornillos no es contenida dentro de estas cubiertas. Esto se puede observar en la imagen 81.

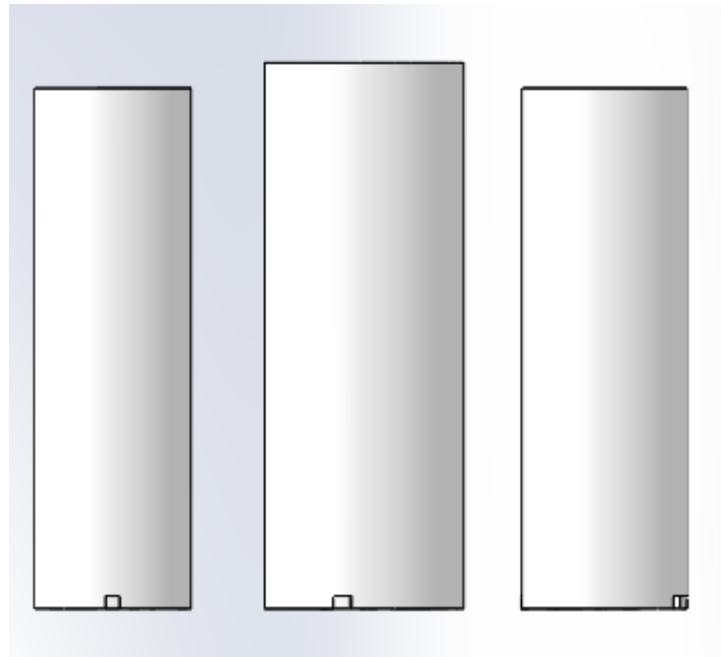


Imagen 81. Modelos CAD actuales de los contenedores 3,1 y 2, respectivamente.

- *Tapas de los compartimientos de almacenamiento*

Como consecuencia a la modificación a los compartimientos, se ha agregado a las tapas una serie de 3 ejes ubicados cada uno a 120° en sus caras inferiores, estos tienen la finalidad de ensamblarlas y fijarlas adecuadamente a sus respectivos contenedores y permitir que sean removibles. A continuación, se puede ver la modificación realizada en las imágenes 82, 83 y 84.

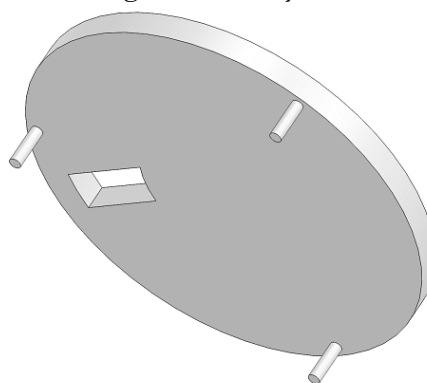


Imagen 82. Tapa 1 vista de la cara inferior.

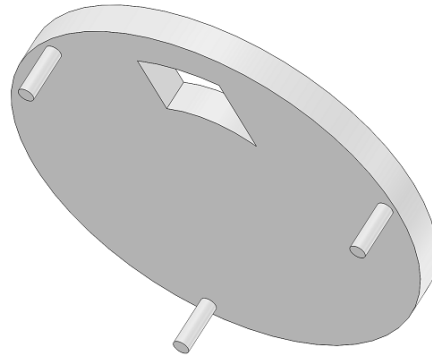


Imagen 83. Tapa 2 vista de la cara inferior.

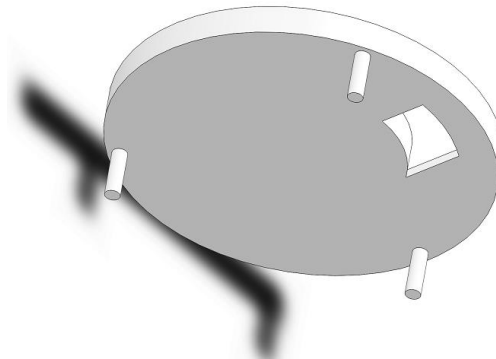


Imagen 84. Tapa 3 vista de la cara inferior.

Otra modificación realizada ha sido en sus caras superiores, donde se han agregado chaflanes a una de las aristas los orificios de entrada de los medicamentos, en el caso de la tapa del compartimento de pastillas redondas su ángulo es de 30° y de 60° para las de los compartimentos de tabletas y capsulas. Estas se pueden apreciar en las imágenes 85, 86 y 87.

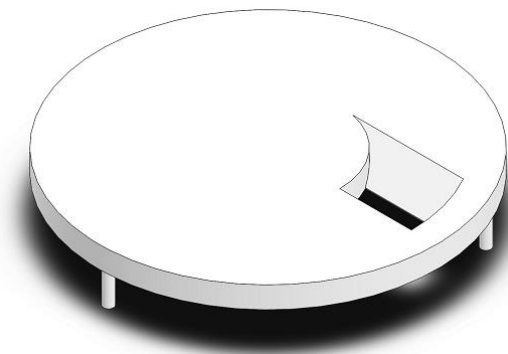


Imagen 85. Tapa 1 vista de la cara superior.

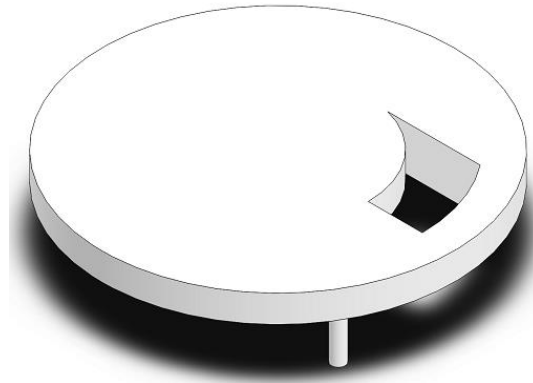


Imagen 86. Tapa 2 vista de la cara superior.

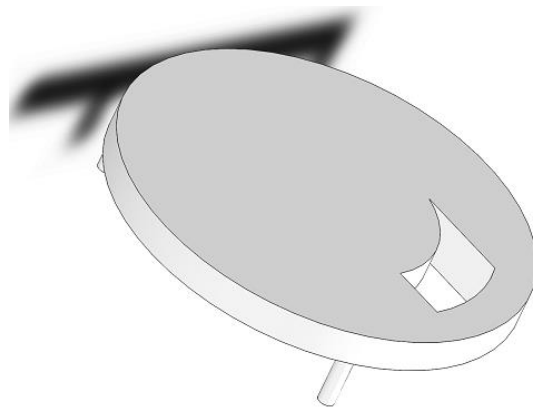


Imagen 87. Tapa 3 vista de la cara superior.

En las imágenes 85, 86 y 87, se puede ver que los chafflanes se han hecho de forma que, cuando los medicamentos ingresen a los contenedores, estas, puedan seguir más fácilmente la trayectoria de la cuerda de los tornillos sin fin, la cual tiene un ángulo de inclinación de 30° respecto al eje horizontal, en caso de que no caigan directamente en el orificio de entrada.

Como bien se mencionó en la sección de los compartimentos de almacenamiento, debido a que los agujeros y, en el caso de las tapas, los ejes del mecanismo de ensamble propuesto resultaban impresos con mala calidad debido a su espesor tan delgado, se optó por cambiar a un mecanismo de ensamble a partir de ranuras y pestañas.

Las pestañas se ubican en las tapas de compartimentos reemplazando a los ejes y se corresponden con las ranuras que están en los contenedores. Su finalidad es la de permitir ensamblar las tapas a los contenedores de forma sencilla.

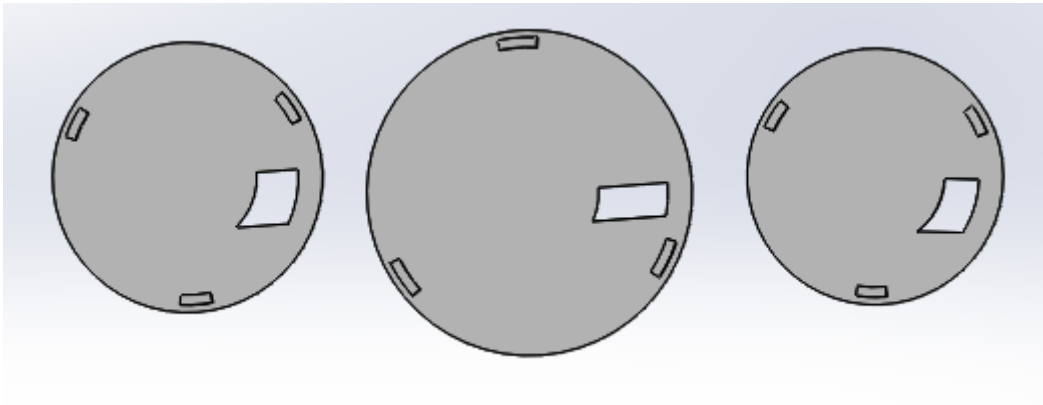


Imagen 88. CAD de las tapas de los contenedores 2,1 y 3, respectivamente, que son los modelos actuales.

La imagen 88, permite observar las pestañas diseñadas en cada una de las tapas de los contenedores.

Las dimensiones de las pestañas son las mismas para todos los contenedores y son las siguientes:

Profundidad: 5 mm

Espesor: 1.43 mm

Longitud de arco: 4.3 mm

- **Tornillos Sinfín**

Estos sufrieron una actualización más notoria que el resto de las piezas, pues los tornillos para los compartimentos de capsulas y tabletas fueron fusionados con sus respectivos “Reductores”. Los reductores son piezas que tenían la finalidad de permitir la transmisión de movimiento rotacional de un motor a pasos monopolar a los tornillos y fueron diseñados en TT1.

La fusión de estas piezas se da debido a que inicialmente estas tenían que ensamblarse, sin embargo, después de analizar su arquitectura, el equipo notó que era posible que fuesen una sola ya que el hecho de combinarlas no afecta el proceso de ensamble del resto de las piezas que componen el mecanismo de dispensado de medicamentos.

A continuación, se muestra una comparativa de los diseños realizados en TT1 y la actualización realizada en TT2. En las imágenes 89 y 90, se visualizan las piezas a comparar.

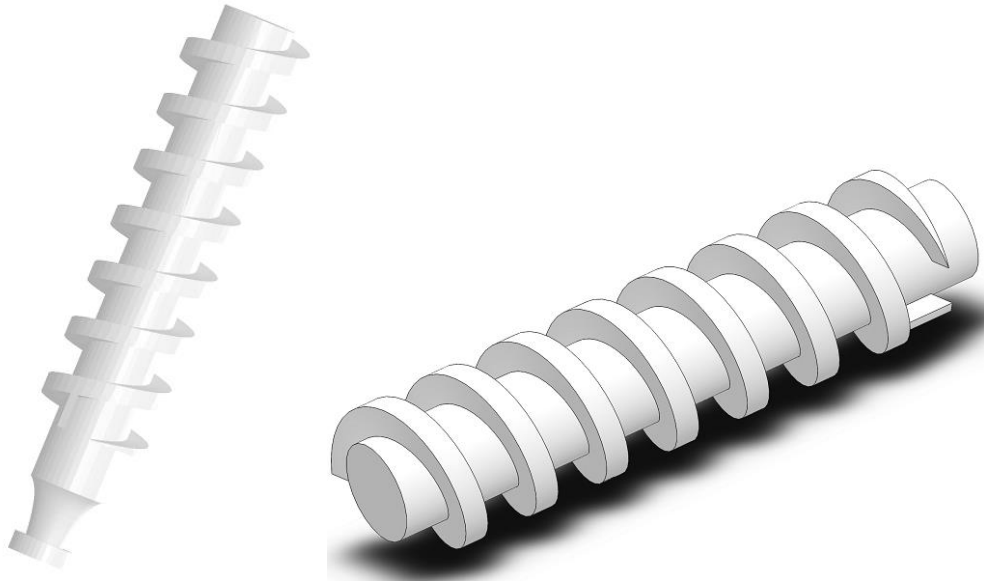


Imagen 89. Tornillo 2 actual en comparación con el tornillo 2 anterior.

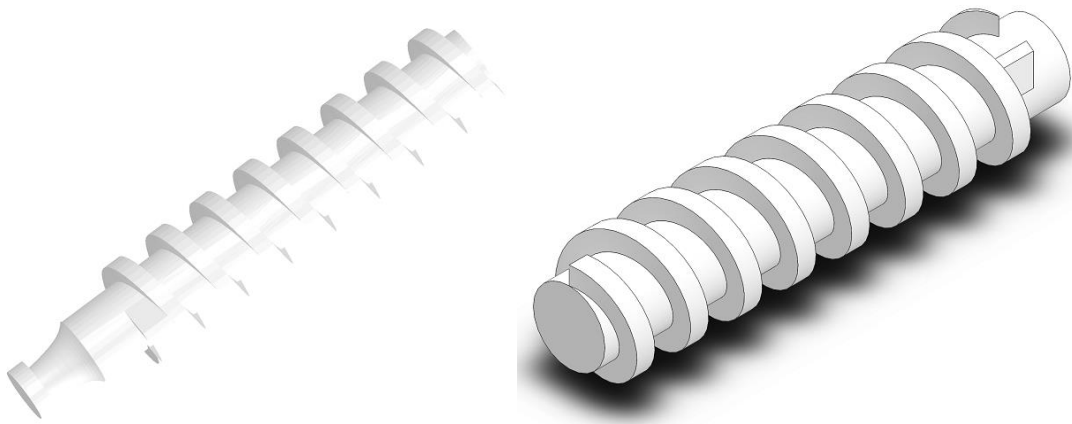


Imagen 90. Tornillo 3 actual en comparación con el tornillo 3 anterior.

Por otra parte, en el caso del tornillo para pastillas redondas, se realizó una reducción de diámetro a la parte inferior del eje del tornillo, con la finalidad de que esta pueda ensamblar su respectivo reductor, el cual fue diseñado en TTI. En consecuencia, al reductor de dicho tornillo se le realizó un agujero para que pudiese ensamblar con el tornillo. Dicha modificación y ensamble se pueden observar en las imágenes 91 y 92.

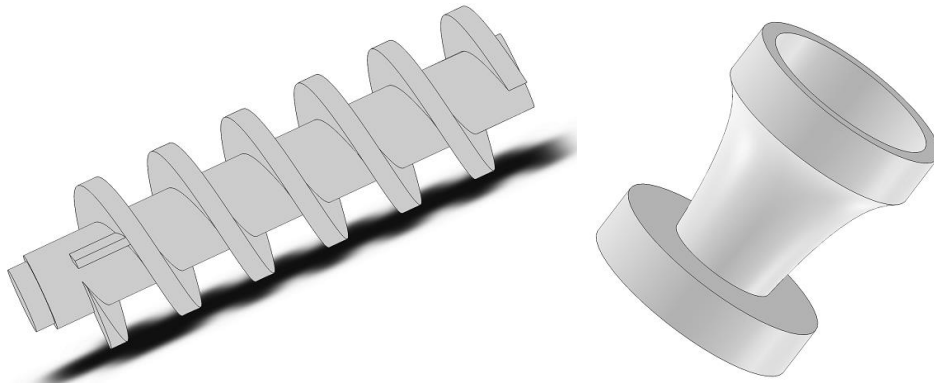


Imagen 91. Componentes por ensamblar para el tornillo 1.

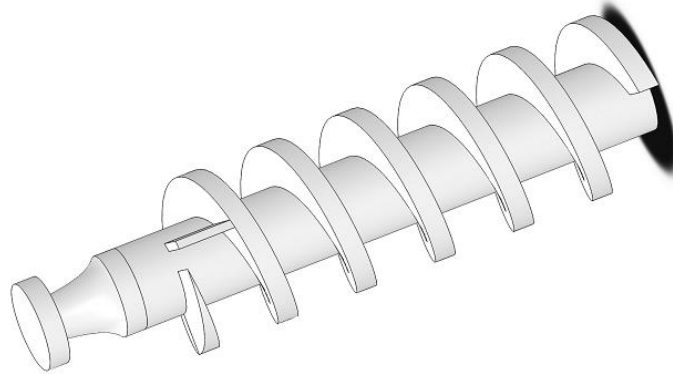


Imagen 92. Ensamble del tornillo 1 junto con su reductor.

Las imágenes 91 y 92 nos permiten observar el desbaste realizado al eje del tornillo sin fin y el agujero o barreno con caja efectuado en el reductor, así como su forma de ensamble.

Diámetro del eje del tornillo 1: 20 mm

Diámetro del eje de ensamble del tornillo 1: 16.85 mm

Diámetro del agujero del reductor del tornillo 1: 17 mm

Debido a la variación en el espesor del revolver, se requirió modificar la longitud de los tornillos sin fin, agregando en el caso de los tornillos de los contenedores dos y tres a la parte libre de cuerda 9 mm. Esto solo para compensar la variación de espesor.

Para el caso del “Tornillo 1”, desafortunadamente se tuvo que agregar más cuerda y altura de eje del tornillo. Esto se debió principalmente a que cuando su contenedor, y por ende el tornillo, se ensamblan con el revolver, se fijan a una mayor profundidad que los contenedores dos y tres, esto debido a su vez a que el largo de las pastillas redondas es menor que el de una tableta o capsula alargada y, con la finalidad de respetar el diseño de ranura de salida de medicamentos en la que quepa una sola pastilla, se requirió introducir 21 mm más en el revolver a este

contenedor en especial. Dicho esto, la longitud final del “Tornillo 1” una vez acoplado a su reductor es de 164mm.

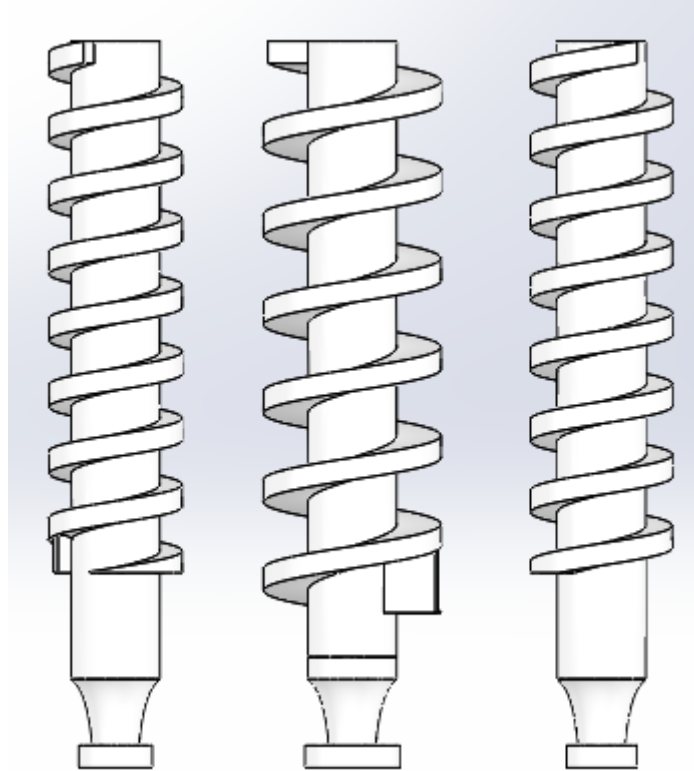


Imagen 93. CAD de los tornillos sinfín 3,1 y 2, respectivamente, que son los modelos actuales.

Como se puede ver en la imagen 93, el cambio es poco perceptible ya que únicamente se incrementó su altura, sin embargo, a continuación, se presentan las nuevas dimensiones de los tornillos:

Longitud del Tornillo 1 sin reductor: 144 mm

Longitud del Tornillo 1 con reductor: 164 mm

Longitud del Tornillo 2: 164 mm

Longitud del Tornillo 3: 164 mm

- **Revolver del mecanismo de dispensado**

La modificación a esta pieza se ha realizado principalmente en los agujeros para los compartimentos de medicamentos. A cada orificio se le ha agregado un riel en forma de Z invertida, el cual, sigue la trayectoria diametral del agujero, este, tiene la función de un seguro que permita ensamblar y fijar el compartimento al revolver por medio de sus muescas evitando así su movimiento rotacional en el eje de cotas y su deslizamiento en los ejes de las abscisas y ordenadas.

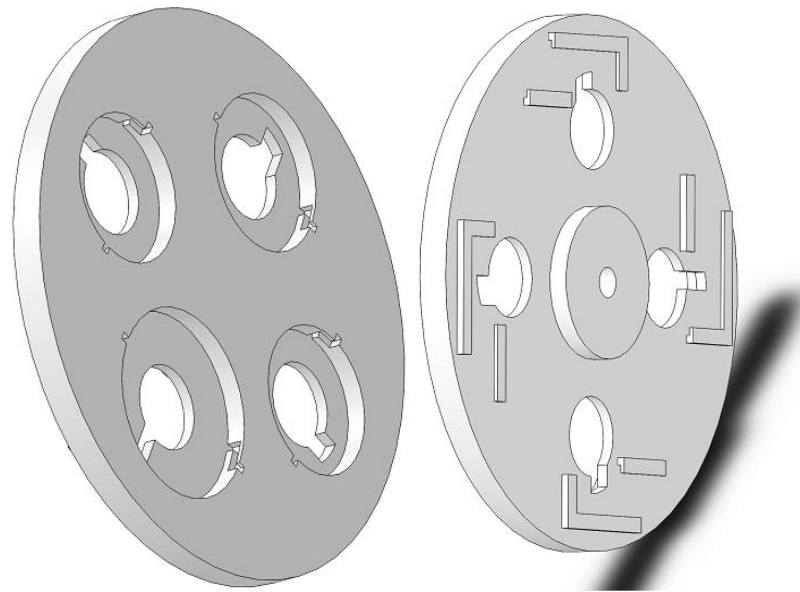


Imagen 94. Vista superior e inferior respectivamente del revolver.

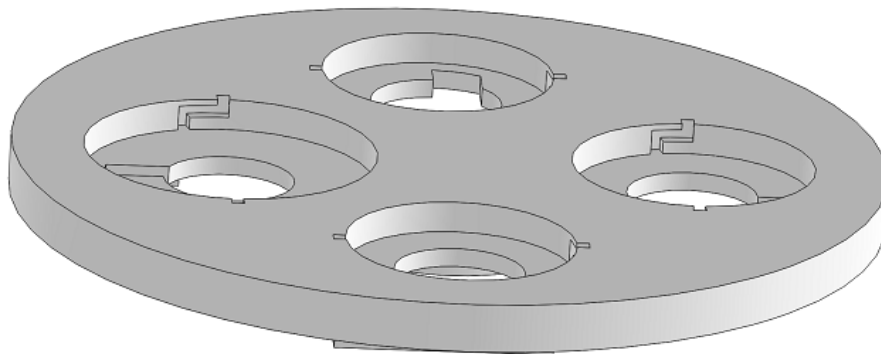


Imagen 95. Rieles de sujeción para cada compartimento.

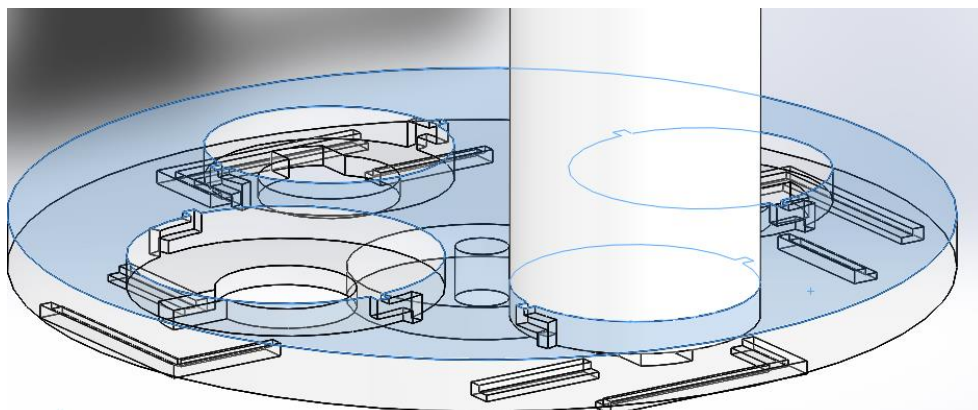


Imagen 96. Ejemplo de ensamble de un compartimento al revolver.

En las imágenes 94, 95 y 96, se observan los rieles de los orificios del revolver. Así como un ejemplo del ensamble de un compartimento a su orificio por medio del riel.



De acuerdo al diseño conceptual del revólver, se busca que, al momento de extraer una pastilla de un contenedor, ésta, quepa en posición vertical en su ranura de salida correspondiente, esto para evitar que, al momento de caer la pastilla en turno, la siguiente caiga accidentalmente por inercia generando una obstrucción en la compuerta de extracción. El equipo revisó el diseño y requirió modificar el espesor del revólver, ya que su valor anterior solo permitía que cupiese en forma vertical una pastilla redonda de 10 mm de radio, sin embargo, las tabletas y capsulas tienen un largo máximo de 17 mm, por lo que estas no cabrían en la disposición deseada.

Tomando como referencia que, la altura de la sección transversal de la ranura de salida de medicamentos en el revólver de los contenedores de tabletas y capsulas debe tener 14 mm, ya que las propias bases de los contenedores tienen 3 mm de espesor, se incrementó su espesor modificando su forma final teniendo un espesor total de 17 mm. Dado que los contenedores de capsulas y tabletas comparten la misma altura de ranura de salida, la zona de ensamble de ambos al revolver mantiene su profundidad de 3 mm. Sin embargo, el contenedor de pastillas redondas requiere una sección transversal con menor altura, para compensarlo, el contenedor deberá introducirse 9 mm en el revólver, de esta forma la sección tendrá 8 mm de altura y agregando los 3 mm de espesor del contenedor se logran 11 mm de altura total en los que cabe de forma vertical la pastilla redonda. Esto a su vez hizo que los compartimentos del medicamento y las piezas que cada uno poseen en su interior tuvieran nuevas características (más adelante se mencionan).

Considerando que ahora el “Compartimento 1” se encuentra a una altura diferente a los compartimentos 2 y 3 en relación con las compuertas, la sección trasversal del revolver se hizo homogénea, para que no posea cambios de sección diametral y así simplificarla.

Además, la parte inferior en donde se colocarán las compuertas para el paso de medicamento se amplió a 3 mm, esto para poder colocar una compuerta de un espesor de 2.85 mm, sumado a que se agregó un barreno de 2 mm para colocar un eje de guía para el sistema de compuertas de extracción del cual hablaremos en la sección de “Compuertas de paso de pastillas”.

A continuación, en las imágenes 97 y 98, se muestran tanto el modelo anterior y nuevo modelo del revólver, respectivamente:

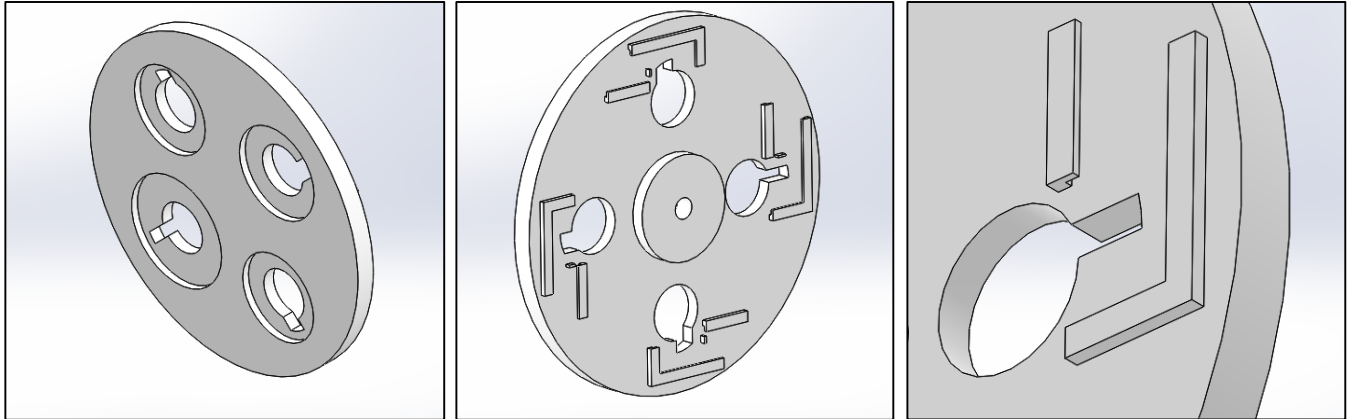


Imagen 97. CAD del modelo del revolver anterior.

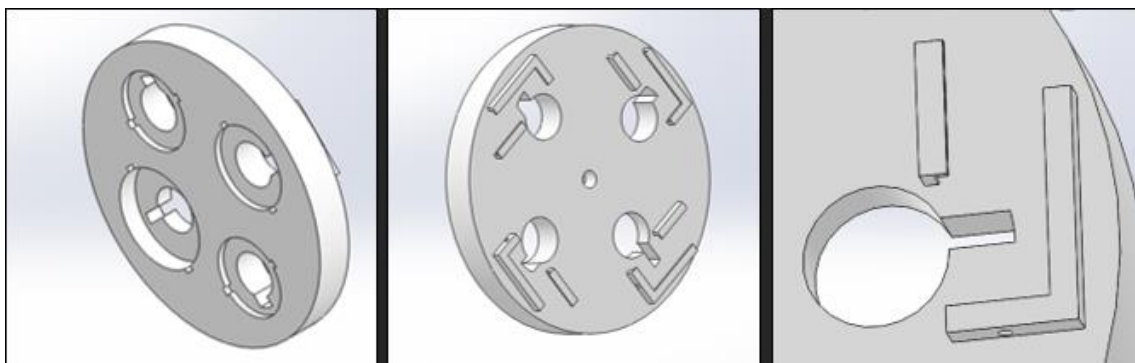


Imagen 98. CAD del modelo del revolver actual.

Acoplamiento de los sistemas de dispensado

Una vez diseñado tanto el sistema de dispensado de agua y el sistema de dispensado de pastillas se valida el espacio que ocuparían con relación a la carcasa exterior para poder estimar el espacio libre para otros módulos del sistema.

En el sistema de agua se ha optado por diseñar una tapa deslizable hacia los costados a manera de que el usuario pueda removerla en el momento en el que sea necesario reabastecer el tanque y de igual forma para que el limpiado de la misma tapa sea practico como se muestra en las imágenes 99 y 100.

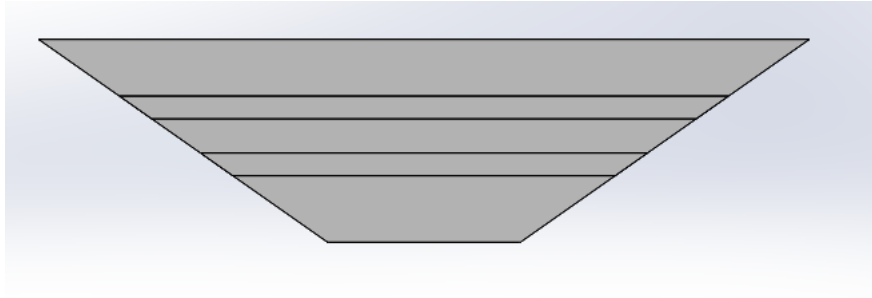


Imagen 99. Vista inferior de la tapa del contenedor de agua.

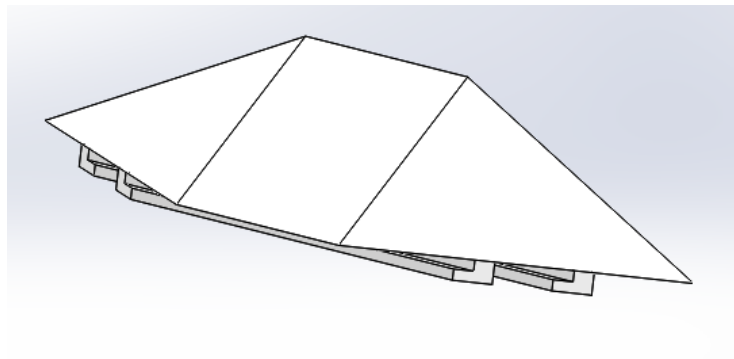


Imagen 100. Vista isométrica de la tapa del contenedor de agua.

Así, el sistema de agua acoplado al dispensador queda de la siguiente manera mostrada en las imágenes 101, 102 y 103.

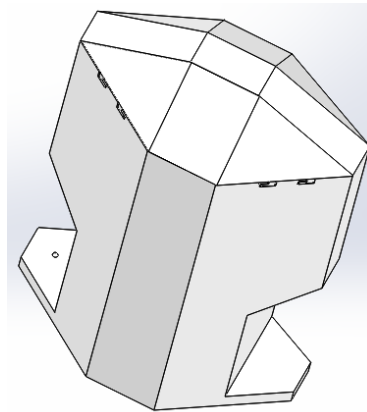


Imagen 101. Vista isométrica de la tapa del tanque de agua acoplada a la carcasa del dispensador.

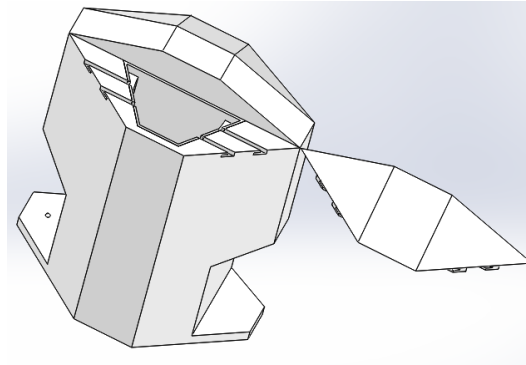


Imagen 102. Vista isométrica de la tapa del contenedor de agua cuando el tanque se encuentra descubierto, se puede apreciar las ranuras por donde se acopla la tapa.

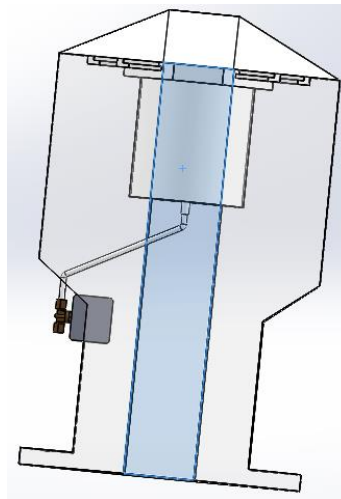


Imagen 103. Vista lateral del sistema de dispensado de agua, se puede apreciar el acople interno del tanque con la electroválvula.

Sistema de entrada de medicamentos

Ahora referente al sistema de entrada, se ha propuesto una ranura de forma rectangular de 10 mm de altura y 12 mm de base, esto para asegurar que puedan entrar tanto cápsulas, comprimidos y tabletas, cabe mencionar que debe ingresar una a una y del mismo medicamento, solo hasta que haya terminado de ingresar todo ese medicamento, es entonces cuando puede ingresar de otro tipo. Esto para asegurar que el funcionamiento correcto del dispensador.

Retomando la ranura de entrada, una vez que haya entrado la pastilla, se desliza por una rampa de material PVC, este material debido a que no afecta la composición del medicamento, dicha rampa mide 12 mm de ancho y 10 mm de alto, con una longitud de 60 mm. En la cual, podrán entrar al menos 3 pastillas tomando en cuenta que la máxima longitud que pueden tener son 17 mm. Esta parte de la ranura y la rampa se puede ver la imagen 104.

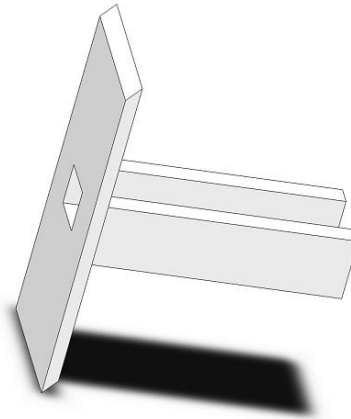


Imagen 104. Entrada de la pastilla, ranura y rampa.

La siguiente parte del sistema, es una cámara que se usará para el reconocimiento visual de cada uno de los medicamentos que se ingresen. Esta es una cámara Raspberry Pi 3 Ov5647, la cual se puede apreciar en la imagen 105.

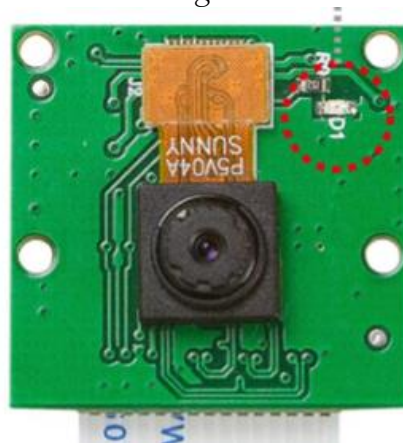


Imagen 105. Cámara Raspberry Pi3 Ov5647. [41]

Esta cámara está montada en una pieza, logrando que esté inclinada 40° donde el lente queda viendo paralelamente a la rampa y con ello asegurar que vea cada pastilla. Esta inclinación es debida a que en esta se deslizan los comprimidos, tabletas y cápsulas sin presentar fricción que las detenga. Entonces el diseño de la pieza que mantendrá en esa posición la cámara es el que se muestra en la imagen 106. También esta pieza soportará un embudo, del cual se hablará más adelante.

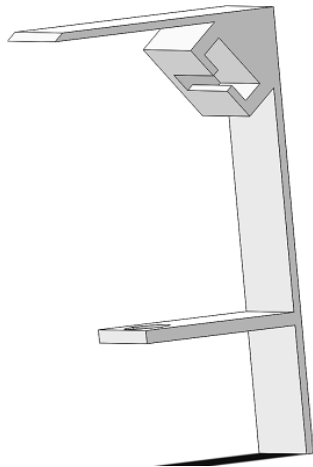


Imagen 106. Soporte de cámara Raspberry Pi 3 y de embudo.

Como se puede ver en la imagen 106, la ranura en la parte superior está hecha para que pueda entrar justo la cámara y se mantenga a un ángulo de 40° , el lente está a una distancia de aproximadamente 7 cm de la rampa y con ello asegurar que pueda ver todas las pastillas que fueron introducidas.

Después se encuentra una compuerta que impide o da paso a los medicamentos que estén acomodados en la rampa. Mientras no se hayan reconocido las pastillas, que esto se da por la cámara y el algoritmo de identificación de comprimidos, tabletas y cápsulas, entonces la compuerta permanece impidiendo el paso de éstas. Una vez sean identificadas, la compuerta es jalada hacia abajo por un actuador solenoide push pull, el cual está montado y dispuesto de tal forma que haga que la compuerta cubra todo el final de la rampa, y que, al momento de abrirse, la vista superior de la compuerta se alinee con la parte de la rampa en cual las pastillas se encuentran soportadas. Estos dos momentos se pueden visualizar en las imágenes 107 y 108, representando el momento que está cerrado y la otra el momento en que se abre. La compuerta es de material PVC, y sus medidas son de 32.25 mm de alto, 22 mm de ancho y 9 mm de espesor, por lo que, se procedió a obtener su masa, la cual se puede obtener en el software de SolidWorks y se puede apreciar en la imagen 109. El solenoide push pull se puede ver en las imágenes mencionadas, éste es de color gris, siendo un diseño representativo.

El tipo actuador mencionado se eligió debido a que, en el diseño, el equipo decidió que fuera un movimiento lineal el que moviera una compuerta ya que facilita su construcción y reduce la complejidad del sistema, además dado que su longitud se mantiene en rangos de 3 a 7 cm de largo, permite ahorrar espacio en comparación con otros mecanismos de transmisión de movimiento como los Manivela-Biela-Corredera.

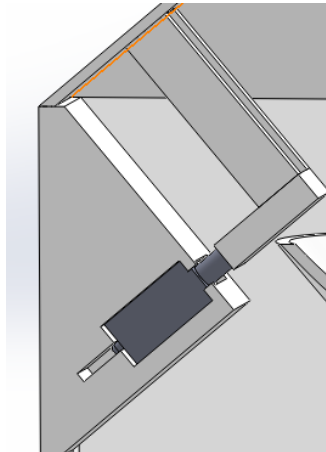


Imagen 107. Momento en el que la compuerta impide el paso de las pastillas.

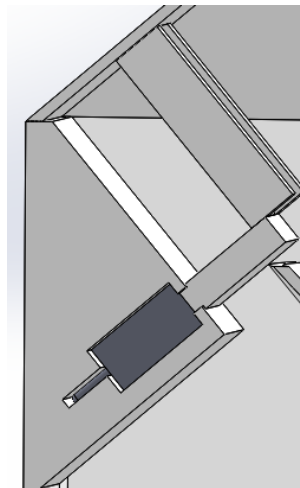


Imagen 108. Momento en el que la compuerta permite el paso de las pastillas.

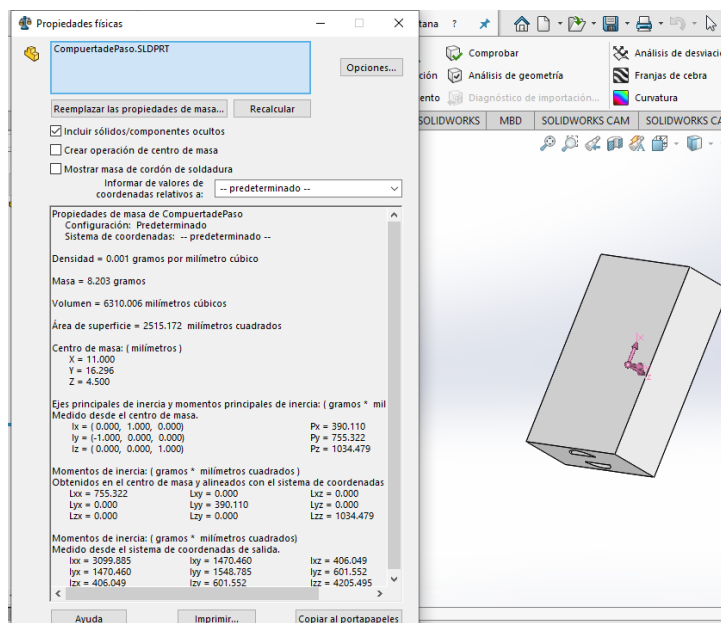


Imagen 109. Obtención de la masa de la compuerta de paso mediante SolidWorks.

A continuación, se proponen tres diferentes solenoides push pull como posibles actuadores para nuestro sistema, los cuales se muestran en las imágenes 110, 111 y 112:

Solenoides push pull Rantec:



Imagen 110. Solenoide push pull Rantec de 12 V. [42]

Solenoides push pull Tecneu:



Imagen 111. Solenoide push pull Tecneu de 5 V. [43]

Solenoides push pull Fielect:

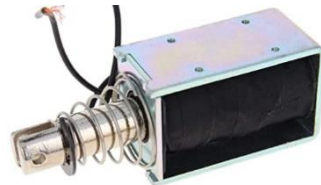


Imagen 112. Solenoide push pull Fielect de 12 V. [44]

Las características principales de estos actuadores para la aplicación que se les desea dar en este proyecto se mencionan en la tabla 17:

	Rantec	Tecneu	Fielect
Tensión	12 V	5 V	12 V
Corriente	1 A	1.1 A	0.5A
Fuerza	0.5 N	0.3 N	0.5 N a 55 N
Tamaño	3 cm	2.9 cm	6.4 cm
Precio	\$140.00	\$139.00	\$559.81

Tabla 17. Características de los solenoides push pull propuestos.

Para seleccionar el actuador más adecuado, se realiza una evaluación de los mismos mediante una tabla de criterios ponderados, esta es la tabla 18, la cual nos indicará cual es la mejor alternativa de acuerdo a nuestras necesidades.

Criterio	Peso	Descripción	Unidad	Opciones:					
				Rantec		Tecneu		Fielect	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Tensión	0.25	Tensión nominal	V	8	2	10	2.5	8	2
Corriente	0.15	Corriente nominal	A	7	1.05	6	0.9	9	1.35
Fuerza	0.25	Fuerza de empuje	N	7	1.75	5	1.25	9	2.25
Tamaño	0.2	Dimensiones del solenoide	mm	9	1.8	9	1.8	5	1
Precio	0.15	Valor monetario	MXN	9	1.35	9	1.35	3	0.45
Total:	1			40	7.95	39	7.8	34	7.05

Tabla 18. Evaluación de los solenoides push pull para la extracción de la pastilla, mediante criterios ponderados.

De la tabla anterior, el solenoide que resultó mejor ponderado fue el de la marca Rantec, el cual cumple satisfactoriamente con los criterios de interés para realice la acción establecida sin problema alguno.

Como se puede observar en la imagen anterior, la masa de la compuerta es de 8.203 gr, y de acuerdo con las especificaciones del solenoide push pull que se eligió, éste brinda una fuerza de 0.5N. Entonces se procede a validar que efectivamente este actuador es capaz de mover linealmente la compuerta de paso, calculado la masa que puede mover y para ello se divide la fuerza que brinda entre la gravedad.

$$m_s = \frac{0.5 \frac{Kg \cdot m}{s^2}}{9.81 \frac{m}{s^2}} = 0.05096 Kg = 50.9683 gr$$

Ahora bien, debido a que puede mover una masa de 50.9683 gr y la masa que se desea que mueva es de 8.203 gr, por lo tanto, queda validado que puede desplazar la compuerta de paso.

Pasando al último punto del sistema de entrada, este es un embudo que se coloca en la pieza que también soporta a la cámara de reconocimiento. Este embudo donde su principal función es que se acomoden las pastillas de forma vertical y lo hagan en fila, esto para el caso de las cápsulas y tabletas, para el caso de los comprimidos pueden acomodarse en fila, pero en su forma plana. Y al momento de entrar a los respectivos contenedores, ya vayan organizados en fila y en cierta posición. Lo importante de su diseño es que tiene una profundidad de 38 mm hasta el punto medio del tubo inmediato de salida, además de que el cono del embudo tiene una inclinación de 40° respecto al eje Z. La parte del tubo de salida, tiene un diámetro interior de 11.5 mm esto principalmente para que los comprimidos puedan entrar sin problema al tubo de salida, este tubo también tiene una inclinación y al final está cortado de forma que la boquilla de salida sea paralela a la cara superior de los contenedores de medicamentos, y que al momento de hacer contacto con la entrada de estos, asegurar que las pastillas entren ya con cierta inclinación y donde debe ser, evitando que alguna pastilla no ingrese al contenedor

porque se haya salido del tubo del embudo antes de los debido. Ahora este mencionado embudo se representa en la imagen 113.

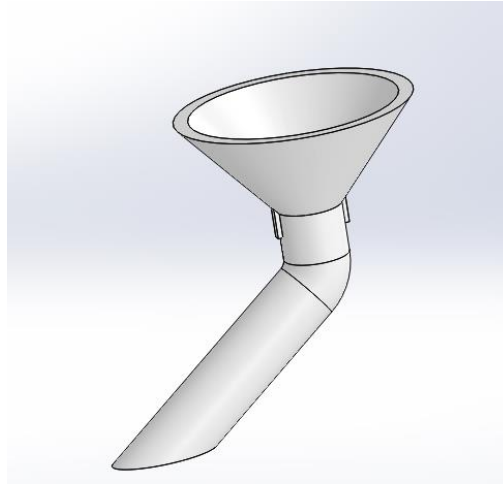


Imagen 113. Embudo de posicionamiento de pastillas.

Finalmente, el diseño del sistema de entrada de pastillas queda mostrado en las imágenes 114 y 115. En la primera, se muestra el sistema con una vista de sección para observar la disposición de los componentes mencionados y, en la segunda, se enseña la carcasa del sistema.

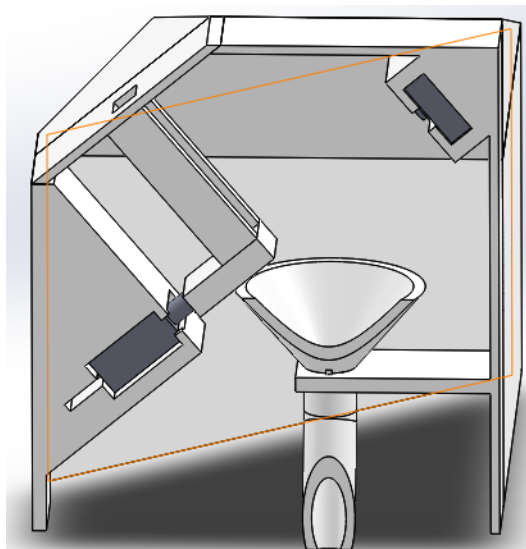


Imagen 114. Vista de sección del sistema de entrada de pastillas.

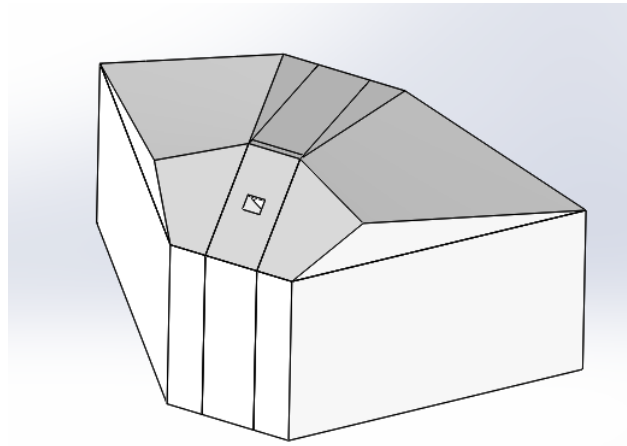


Imagen 115. Vista de la carcasa del sistema de entrada de medicamentos.

Actualización de diseño del sistema de entrada

- *Mecanismo de identificación de medicamentos*

En el caso de este mecanismo en particular, se ha decidido rediseñar el método mediante el cual se realice la identificación de las pastillas, dejando atrás la propuesta de emplear un sistema de visión artificial. Este cambio se debe principalmente a que, dado que el usuario debe ingresar los medicamentos al dispensador de forma manual y una pastilla a la vez, resulta más sencillo que la identificación del medicamento se realice de forma manual por el usuario a través de la interfaz gráfica que proporcionará la aplicación móvil.

Puesto que las diferentes formas de medicamentos no se mezclarán al ingresarlos al dispensador y los tratamientos también se ingresan uno a la vez, un sistema complejo de visión artificial haría que el funcionamiento fuese más lento y el programa más susceptible a fallos.

Dicho lo anterior, se remueven del diseño la cámara, su sujetador, y sus conexiones y se reemplazará por una ventana adicional en el proceso de configuración de nuevos tratamientos que se realizará desde la aplicación móvil, la cual, contará con imágenes descriptivas de cada forma de medicamento que funcionaran como botones para que el usuario seleccione cual corresponde con el medicamento que va a ingresar, logrando así una comunicación intuitiva y sencilla con el usuario y asegurando la correcta separación de los fármacos en el dispensador.

- *Cono de entrada de medicamentos*

Esta pieza funciona como receptor de los medicamentos que el usuario ingresa al dispensador y, a su vez, como puente hacia su contenedor final. Sin embargo, el equipo considera que la forma propuesta en T11 podría no ser tan efectiva, sobre todo en el tramo final en el que la pastilla sale del ducto del cono hacia su contenedor final como se muestra en la imagen 116.

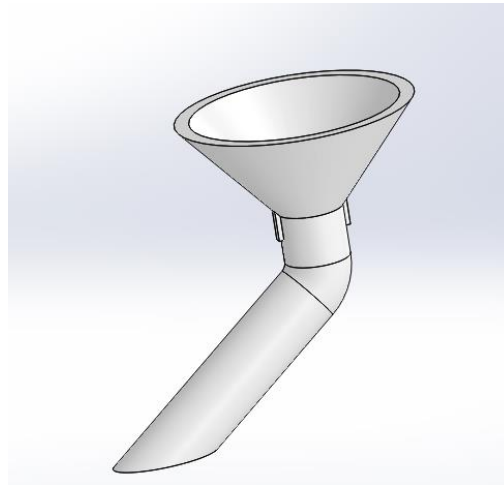


Imagen 116. Vista isométrica del embudo anterior.

El equipo considera que, dado que la salida hacia los contenedores es prácticamente vertical, existen grandes posibilidades de que los medicamentos de forma plana como las tabletas o las pastillas redondas caigan en una posición perpendicular a la cuerda de los tornillos sinfín y que, por ende, los medicamentos no logren entrar a su contenedor o se atasquen en la entrada.

La modificación efectuada, permite que los medicamentos descendan por un ducto hasta alcanzar una postura que no comprometa su entrada al contenedor y siguiendo un ángulo de deslizamiento igual o superior al de la cuerda de los tornillos. Esto puede observarse en la imagen 117.

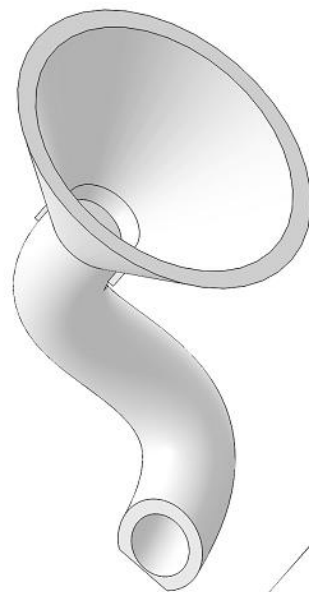


Imagen 117. Vista isométrica del embudo actual de entrada.

- *Sujetador del embudo de entrada de medicamentos*

Dado que el ingreso de medicamentos pasó a ser un proceso manual asistido por la aplicación móvil eliminando por completo los procesos de visión artificial, como ya se mencionó anteriormente, el equipo buscó maneras de aprovechar el espacio liberado tras remover los componentes que dicho proceso implicaba. La finalidad es pues, ahorrar espacio y, sobre todo, reducir en medida de lo posible la altura total del dispensador que, a su vez, permite ahorrar material de impresión reduciendo así el costo final del prototipo.

Dicho esto, el equipo modificó la pieza llamada “soporte para cámara Raspberry Pi3 y embudo”, se eliminó de ella el sujetador para la cámara web. Adicionalmente, se eliminó de ella la cubierta superior que conformaba parte de la carcasa, reduciéndose a pieza a una “T” que se ensamblará a la estructura central del dispensador para que actúe como repisa sobre la cual descansará el embudo de entrada de medicamentos. Como se visualiza en la imagen 118.

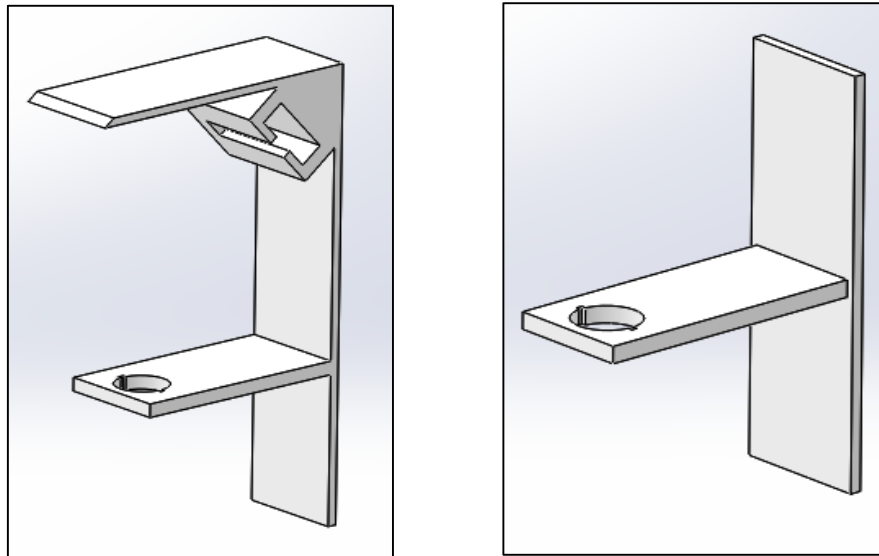


Imagen 118. Sujetador del embudo de entrada de medicamentos.

En la imagen 119, se puede observar a la izquierda la versión inicial y a la derecha la versión actual de la pieza en cuestión.

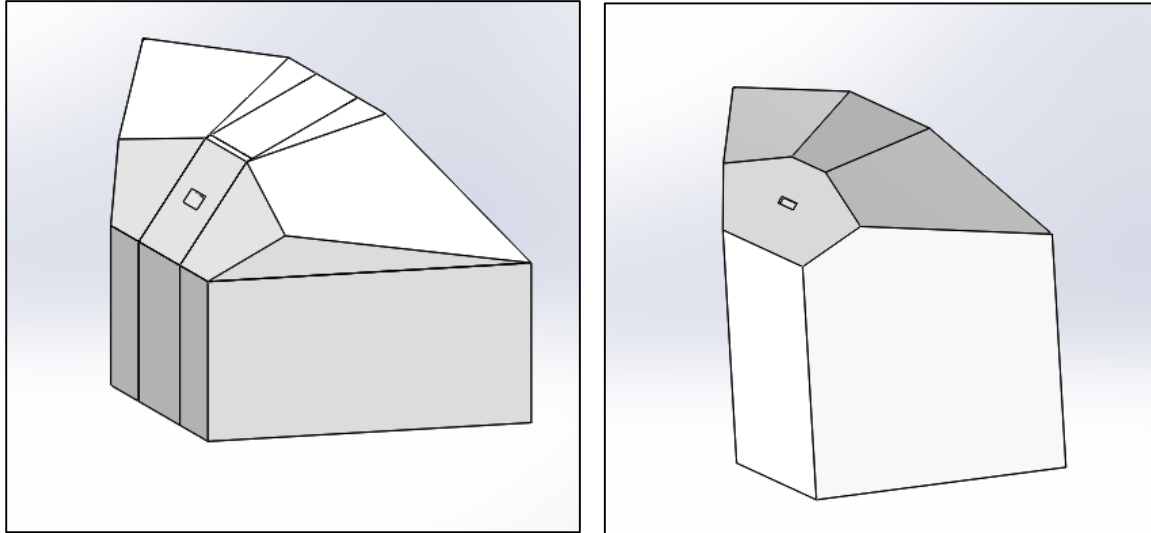


Imagen 119. Carcasa de entrada de medicamentos.

La sección superior eliminada del sujetador y que fungía como parte de la carcasa ahora pasará a ser parte de la tapa superior de la misma. De esta forma se simplifica la impresión de ambas piezas. Es por eso, que la carcasa de entrada de medicamentos quedó como la que se muestra a la derecha de la imagen 7, mientras que a la izquierda se observa la carcasa de entrada anterior.

Sistema de salida de medicamentos

Éste, se diseñó a manera de que pudiera ser acoplado fácilmente al sistema de entrada. En primera instancia cuenta con el mecanismo de entrada que consiste en los contenedores acoplados al revolver como se muestra en la imagen 120. Al momento de ingresar los medicamentos el contenedor requerido para ese tipo específico de comprimido o cápsula se alinea a la parte lateral izquierda (vista de manera superior) del sistema para ser abastecido. Para la operación de dispensado el contenedor indicado se alinea a la parte lateral derecha.

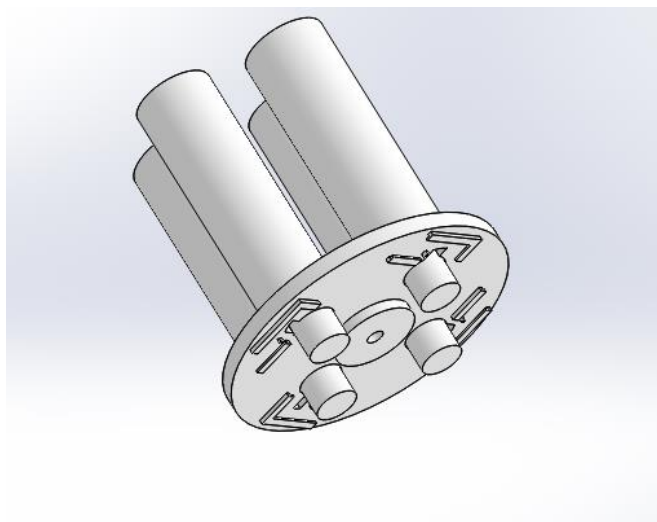


Imagen 120. Mecanismo de entrada/dispensado de medicamentos.

Al momento de extraer el medicamento del contenedor, se ocupa un motor a pasos el cual hará girar el eje del tornillo sinfín acoplado dentro de dicho contenedor; para esto se evaluaron los mismos motores propuestos en la parte de selección de motor para mover el revólver. Para dicha selección, ahora lo que interesa es que la tensión nominal se la menor posible, esto debido a que no es necesario más de 12V esto debido a todos los componentes eléctricos que se manejan en el proyecto.

En la siguiente tabla 19, se puede apreciar la evaluación que se realizó, en la cual el motor a pasos seleccionado por los criterios ponderados fue el 28BYJ-48. El cual resultó el mejor evaluado y que cumple con la función de mover el eje el tornillo.

Criterio	Peso	Descripción	Unidad	Opciones:					
				PM55L-048		28BYJ-48		NEMA17	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Tension	0.15	Tensión nominal	V	6	0.9	9	1.35	7	1.05
Corriente	0.15	Corriente nominal	A	7	1.05	9	1.35	6	0.9
Tamaño	0.15	Dimensiones del motor	mm	7	1.05	9	1.35	8	1.2
Driver	0.2	Grado de complejidad de implementación	Si/NO	9	1.8	10	2	9	1.8
Torque	0.2	Torque de Pull-In	Nm	9	1.8	6	1.2	10	2
Precio	0.15	Valor monetario	MXN	8	1.2	9	1.35	6	0.9
Total:	1			46	7.8	52	8.6	46	7.85

Tabla 19. Evaluación de los motores a pasos para la extracción de la pastilla, mediante criterios ponderados.

Para su implementación se contempla el uso del driver UNL2003, el cual está diseñado para controlar el giro de un motor a pasoso monopolar. Este, requiere de una alimentación de 5 a 12 V, así como conexión a cuatro pines GPIO de la tarjeta de control los cuales permiten pasar la tensión y corriente para el encendido de las bobinas del motor a pasos.

A diferencia del driver A4988, éste no requiere de previa caracterización, únicamente se requiere que, en caso de tener alimentación externa para el motor se tenga tierra

compartida con la tarjeta de control. El diagrama eléctrico de la implementación de éste driver junto al motor a pasos 28BYJ-48 se encuentra en los apéndices de este documento en la página 195.

Para controlar la salida del medicamento se utilizan las denominadas “compuertas”, que están dispuestas en la parte inferior de cada salida del tornillo (imagen 121), estas bloquean la salida del comprimido o capsula que se encuentra en ese momento en los “racks” y su posición original se reestablece por medio de dos resortes que estarán colocados en la cara opuesta al embolo donde se acopla el actuador. Esta posición está calculada para que al moverse la compuerta 5 mm y ésta deje pasar el medicamento a través de los racks.

Los racks se describen como 2 aberturas por donde pasara el comprimido o capsula y que, para este fin, ambos deberán estar alineados como se muestra en la imagen 122. Uno de ellos es la abertura misma de la tapa del contenedor de comprimidos. El otro es la parte del revolver.

Cabe señalar que para poder acoplar de manera eficiente los mecanismos se diseñó una extensión en la parte inferior de cada tornillo, esto a manera de que, al momento de llegar a la posición de salida, todos ellos presentaran una misma distancia de manera tangencial desde una posición específica, con el objetivo de transmitir movimiento tangencial al tornillo, dicha extensión se puede visualizar en la imagen 123.

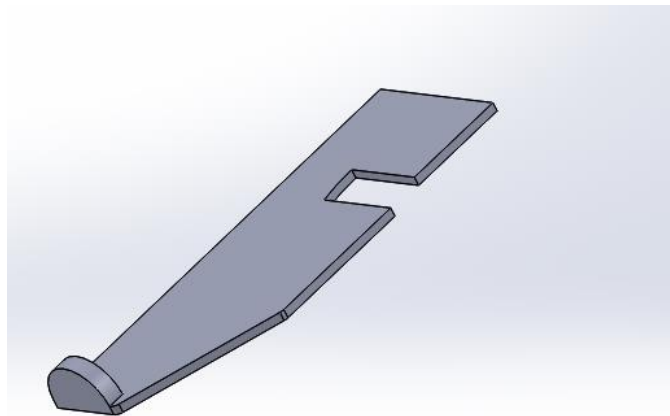


Imagen 121. Vista de una compuerta en la parte inferior, el embolo que tiene es para el acople al actuador.

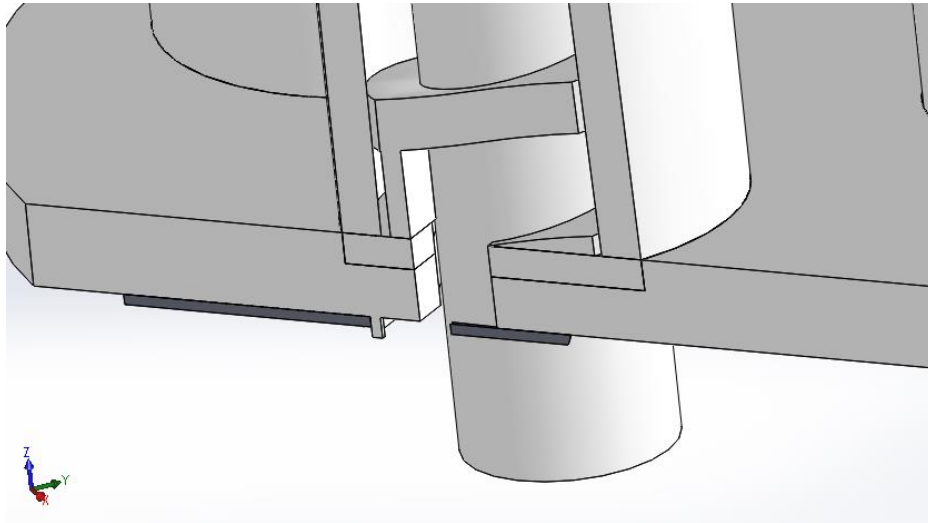


Imagen 122. Vista de sección transversal del revolver acoplado a los contendores y a la compuerta.

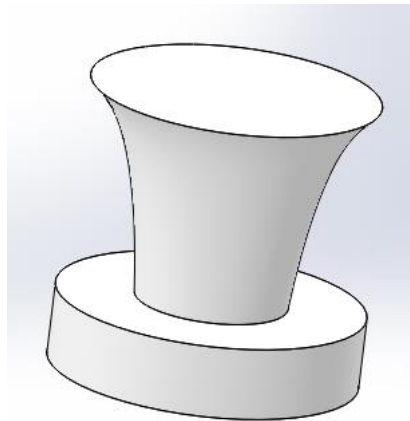


Imagen 123. Acople del tornillo 3, el diámetro de cada acople en la parte inferior es diferente para garantizar la transmisión de movimiento.

Una vez liberada la pastilla, esta cae en un medio cono que termina en el lado opuesto a donde se dispensara el agua, está diseñada para ser del mismo material que la mayoría del dispensador (PVC), esto para cumplir con la normativa al trabajar con medicamento. El segundo cono posee un ángulo de 40 grados de inclinación, esto garantiza un rápido deslizamiento. Termina en una sección circular de 11.5 mm de diámetro, con un espesor de 2.5mm. Este sistema de salida esta sobre la parte inferior del revolver, a una distancia cercana pero que no posee contacto con los tornillos, diseñado así ya que al tener los tornillos diferente excentricidad, esto hace inviable una rampa que se pueda alinear a una única distancia con la salida inmediata, ya que chocaría con los mismos tornillos al momento de girar el sistema del revolver. La rampa se muestra en la imagen 124.

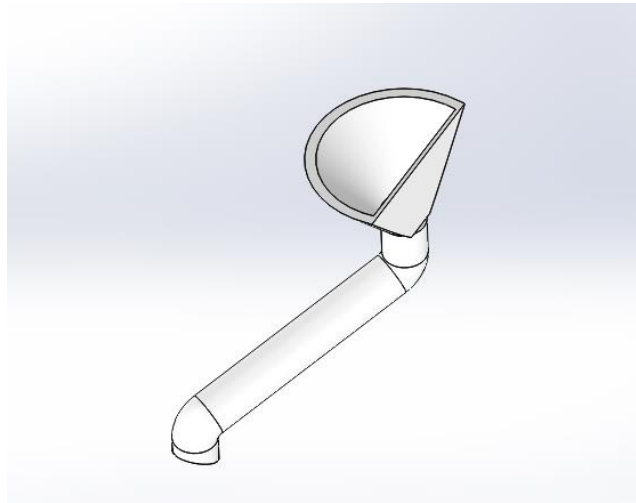


Imagen 124. Embudo de salida del medicamento, el final de esta se acopla a la parte de la salida lateral de la carcasa.

El actuador que permite la salida de medicamento de los contenedores se fija a una base que se encuentra sobre la carcasa en la parte interna y desde ahí, el embolo del actuador se alinea con los émbolos de las compuertas a una distancia aproximada de 5 mm, esto para evitar chocar con estos últimos, ya que la carrera de los actuadores es de aproximadamente 10 mm. La posición de estos se muestra en la imagen 125, mientras que el tipo de actuador usado se muestra en la imagen 110.

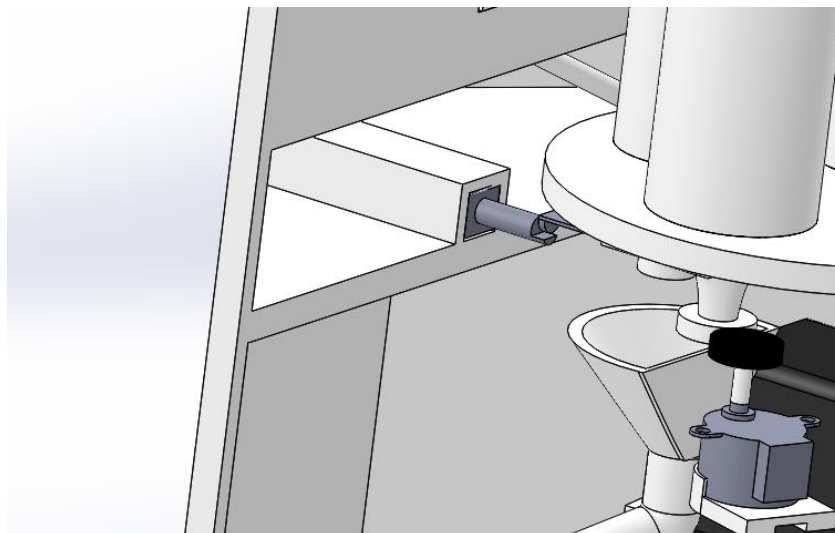


Imagen 125. Actuador alineado al embolo de la compuerta en posición de salida, también se aprecia la disposición de la rampa.

El movimiento de rotación en los tornillos es generado a través de una rueda de material que al contacto con el tornillo permita una transmisión de movimiento tangencial, esto impulsado por un motor a pasos montado sobre una pieza que de igual manera se puede desplazar sobre un riel. El riel puede moverse linealmente impulsado por un actuador del mismo tipo que el utilizado en la parte de las compuertas. Cabe señalar que aquí se aprecia el uso de los acoples realizados sobre los tornillos. Se calculó una posición óptima para que este sistema de trasmisión de movimiento al desplazarse la misma distancia

podiera contactar con los acoples de manera tangencial o semi tangencial. Esto como ya se menciono es debido a que los centros de cada tornillo no poseen la misma excentricidad. Este sistema se observa en la siguiente imagen 126, se aprecia también que el riel sobre el que está montado el seguidor es parte de la misma carcasa de salida.

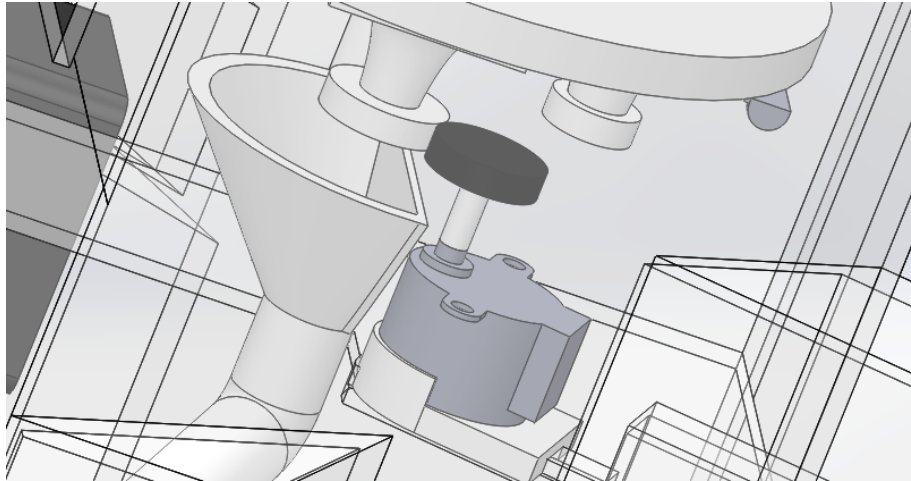


Imagen 126. Sistema de transmisión de movimiento, se aprecia la posición tangencial del transmisor con el acople del tornillo.

También se necesitó modificar la carcasa de salida, ya que la se requirió más espacio para poder acoplar el actuador, de esta manera la carcasa presenta una sección rectangular alineada a 110 grados con la horizontal de la carcasa y con un espesor de 2mm, esto se muestra en la imagen 127.

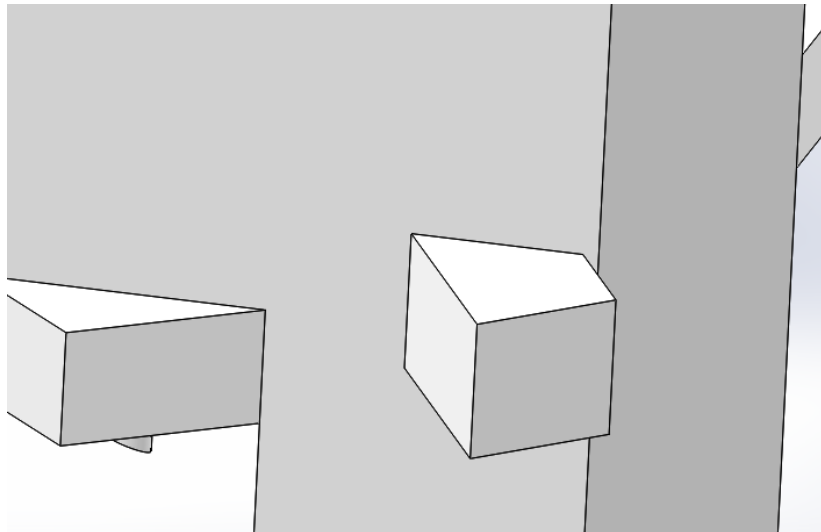


Imagen 127. Cubierta del segundo actuador adaptada a la carcasa de salida.

Finalmente, la carcasa de este subsistema es la que facilita el acople con el sistema de entrada y que se adapta al cambio de forma de la carcasa en la parte inferior, mostrada en la imagen 128.

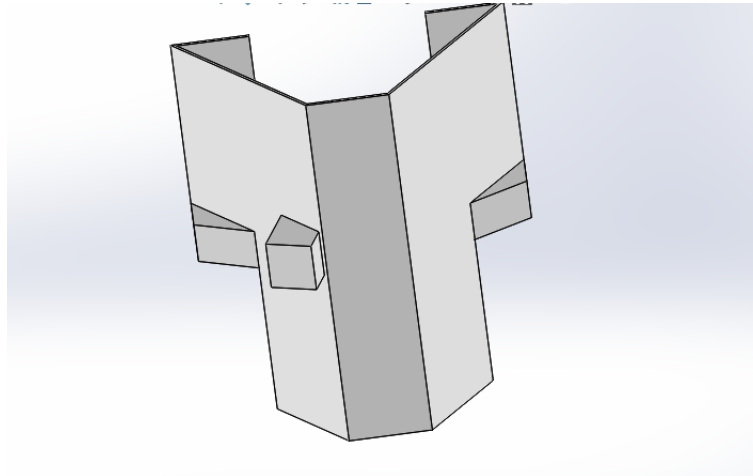


Imagen 128. Carcasa del mecanismo de salida.

El ensamble final de los sistemas mencionados anteriormente se muestra a continuación. Se aprecia el acople del sistema del tanque de agua con el mecanismo de entrada (imagen 129), el detalle de los sistemas de entrada y salida mencionados anteriormente (imagen 130), así como la vista final del dispensador (imagen 131).

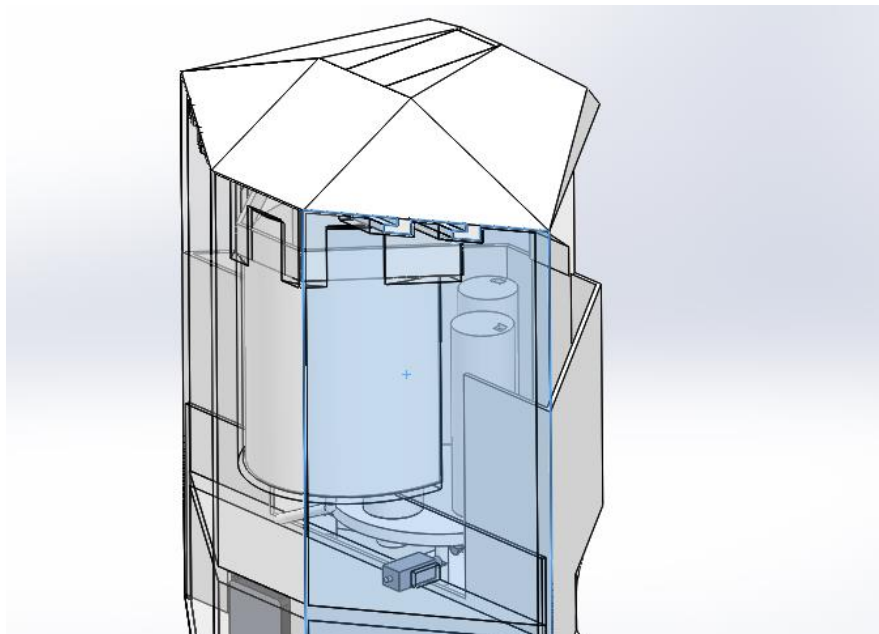


Imagen 129. Acople del sistema de agua con la carcasa y con el mecanismo de entrada.

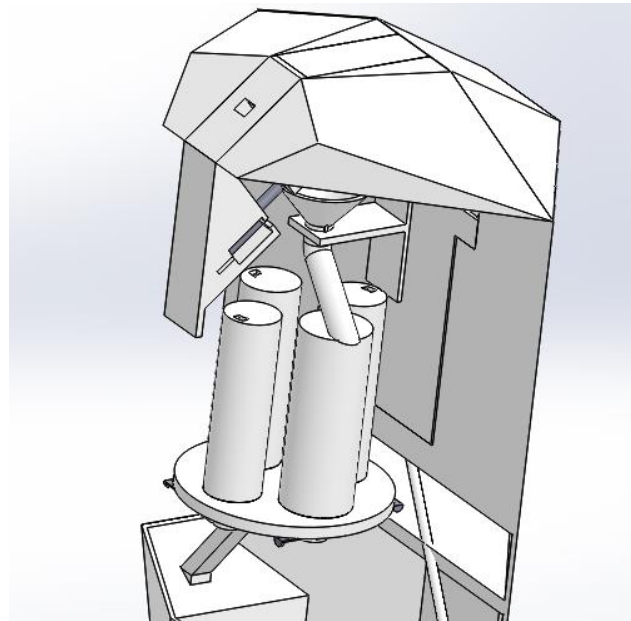


Imagen 130. Acople de los sistemas de entrada y salida, se aprecia la disposición del embudo con el contenedor lateral izquierdo del sistema.

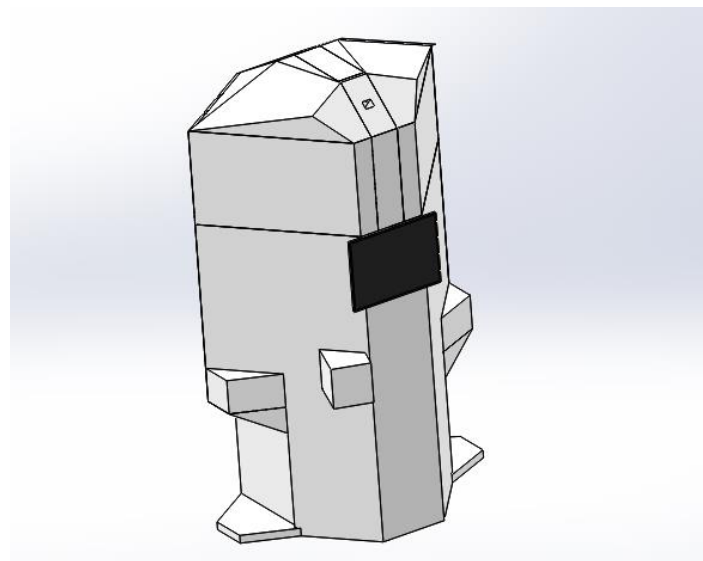


Imagen 131. Acople final del dispensador de comprimidos y capsulas, se aprecia el cambio notable con respecto al boceto original.

Respecto a la validación de todo el funcionamiento del sistema de entrada de pastillas, no es posible realizar una validación por un software sino que se debe hacer la prueba física, y debido a que el equipo de trabajo no cuenta con los componentes que lo conforman, solo se tienen las pruebas que se hicieron en el ensamble en el caso de la carrera del solenoide usado, las pruebas en las que se validaron el ángulo al cual los medicamentos sobrepasan el coeficiente de fricción y comienzan a deslizarse, y la validación de la capacidad de mover la compuerta de paso por el solenoide. Es por eso que la validación sería explícitamente por pruebas en físico, para asegurar el correcto

funcionamiento de la cámara de reconocimiento, y si no hay atasco de pastillas en el embudo por una posición de caída inesperada.

Actualización del sistema de Salida

Se actualizó el mecanismo para la transmisión de movimiento a las compuertas del sistema de salida, si bien el principio es el mismo, se requirió el uso de un segundo solenoide para elevar una base donde estará colocado el primero.

Esto se debió a que había un problema de contacto al momento del giro del revolver con el solenoide de empuje de la compuerta, por lo que se diseñó un ascensor para poder elevar y descender este primer actuador.

Este mecanismo consiste en 2 piezas principales denominadas paredes del ascensor, 1 pieza de acople para unir ambas y que además brinde estabilidad al compartimento para el solenoide de ascenso y finalmente la base que sostiene al solenoide de empuje de la compuerta. Dicho mecanismo se puede ver en la imagen 132.

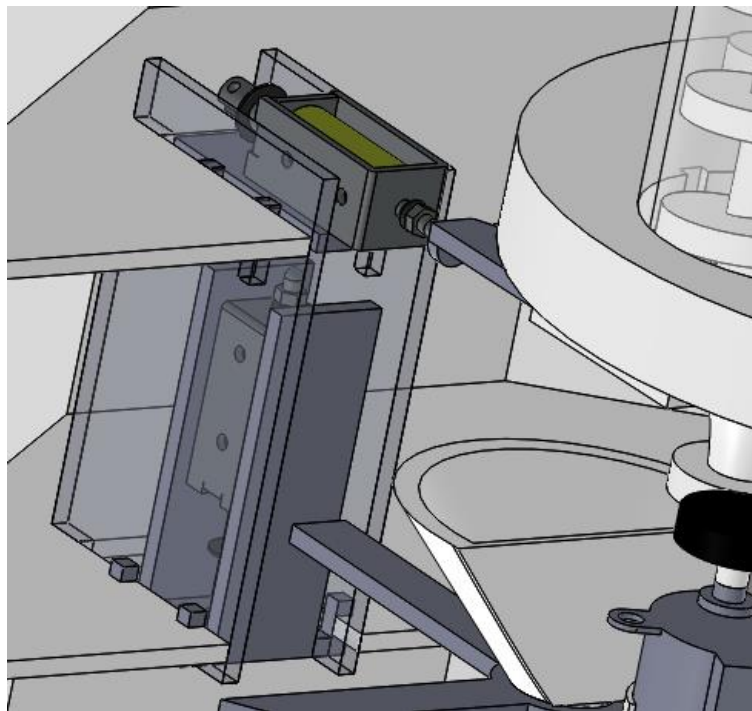


Imagen 132. Nuevo mecanismo de empuje de las compuertas, se aprecia la composición del mecanismo denominado "Ascensor".

Para esto se utilizó la carcasa principal inferior en la parte de las plataformas para darle un acople al mecanismo en la parte de las paredes del ascensor.

- **Compuertas de paso de pastillas**

Las compuertas, como se mencionó en T1, serán las que impidan la caída continua de pastillas del revolver durante el proceso de extracción, y son activadas por un solenoide push pull cuando se busca extraer una pastilla de un contenedor, sin embargo, como

recordará, el solenoide solo empuja para que la pastilla salga y esta vuelve a acomodarse en posición de bloqueo de la ranura de salida gracias a un resorte de compresión.

El resorte que busca implementarse tiene las siguientes medidas:

Diámetro exterior: 0.120 mm

Diámetro interior: 0.105 mm

Longitud inicial: 1.3 mm

Longitud en máxima compresión: 3 mm

Dicho lo anterior, el equipo considera que resulta más efectivo colocar un eje o guía para la compresión del resorte, pues así se puede evitar una situación en la que éste se doble durante su compresión, por lo cual, se ha añadido un eje a las compuertas. Además, se ha aumentado el espesor de las mismas a 2.85 mm.

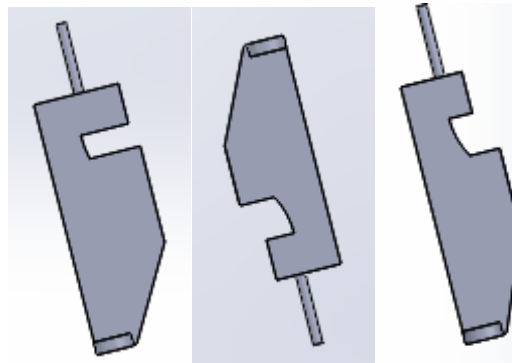


Imagen 133. CAD de las compuertas 1, 2 y 3 respectivamente, modelos actualizados.

La imagen 133, permite ver estas nuevas características en las piezas, por otro lado, es preciso indicar las nuevas dimensiones básicas de las compuertas.

Compuerta 1 (Para pastillas redondas):

Largo máximo: 64 mm

Ancho máximo: 18.01 mm

Diámetro del eje del resorte: 1.85 mm

Altura del embolo: 7.27 mm

Radio del embolo: 3.63 mm

Compuerta 2 (Para tabletas):

Largo máximo: 64 mm

Ancho máximo: 16.2 mm



Diámetro del eje del resorte: 1.85 mm

Altura del embolo: 6.46 mm

Radio del embolo: 3.61 mm

Compuerta 3 (Para capsulas):

Largo máximo: 64 mm

Ancho máximo: 15.21 mm

Diámetro del eje del resorte: 1.85 mm

Altura del embolo: 6.59 mm

Radio del embolo: 3.59 mm

Sistema de control de temperatura interna del dispensador

El problema de control del equipo se simplifica debido a que no es necesario que los medicamentos se encuentren a un valor de temperatura específico, es decir, no existe como tal un valor de referencia al cual la temperatura interna del dispensador se debe de establecer, basta con que su valor se encuentre por debajo de los 30°C la mayor parte de tiempo posible mientras se encuentran almacenados. Dicho esto, el equipo logró concluir que, al no existir la necesidad del establecimiento de la temperatura interna del aparato a un valor de referencia, emplear un controlador P, PD, PI o PID resulta innecesario pues, además, no se requiere tener control sobre la respuesta estacionaria ni transitoria de la salida del sistema. Basta con el diseño de un esquema de control ON-OFF para poder regular el sistema de la forma que el problema lo requiere ya que, estos controladores, permiten realizar una regulación sobre la variable a controlar sin la necesidad de establecerla a un valor de referencia; la desventaja del controlador ON-OFF es que da como resultado una señal de salida con oscilaciones, sin embargo, ya que para la aplicación requerida en este proyecto no es relevante el establecimiento de la señal a ningún valor de referencia, esta condición no representa un problema para su implementación.

Sin embargo, debido a que no existe una temperatura de establecimiento, sino simplemente la intención de no rebasar un límite máximo en el valor de temperatura, el equipo propuso un controlador por comparaciones, el cual permitirá realizar más de una acción de control sobre el actuador de enfriamiento del sistema, su intención sería la de poder activar el actuador a tres diferentes valores de operación dependiendo de qué tan próxima está la temperatura a los 30°C considerados como temperatura interna límite. Lo anterior sería realizado mediante la implementación de tres comparadores que permitan definir el rango de error en el que se encuentra la temperatura leída por medio de un sensor y en base ello activar un conjunto de ventiladores de enfriamiento a un

porcentaje de su velocidad máxima para tratar de aumentar la diferencia entre la temperatura actual y la límite. A continuación, se muestra el diagrama de bloques del lazo de control, en la imagen 134.

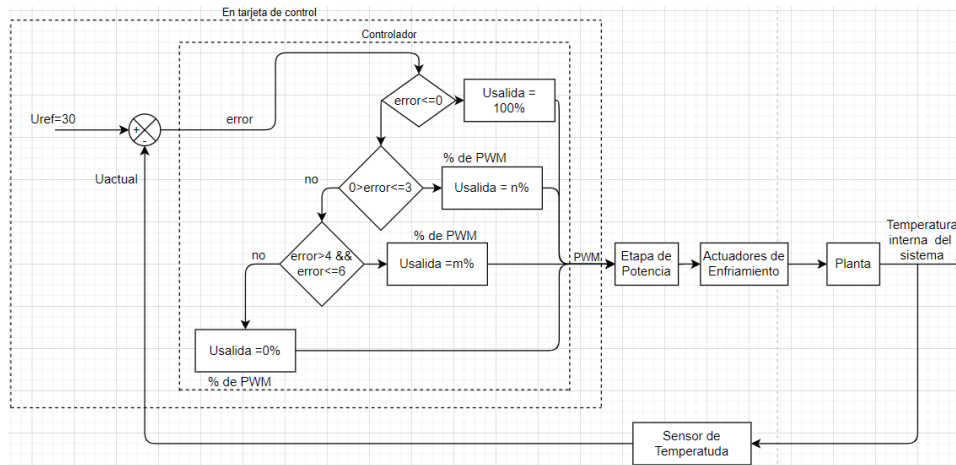


Imagen 134. Lazo cerrado de control de temperatura interna del dispensador.

Como se puede apreciar en la imagen 134, el diseño del controlador contempla emplear tres diferentes porcentajes de PWM para la activación de los actuadores de enfriamiento, sin embargo, dos de ellos no se encuentran definidos, lo anterior se debe a que para poder determinar el porcentaje de PWM de dichos casos, primero se deberá caracterizar el actuador a implementar para así poder determinar con qué porcentaje se produce el arranque de los actuadores y así tener idea de cuál será el porcentaje mínimo posible. Además, debido a que el dispositivo será construido en PVC que es un polímero, y se sabe que los polímeros son buenos conductores de calor, se desconoce cuál será la inercia térmica del aparato, razón por la cual resultaría imposible establecer la potencia a la cual se requiere que se encuentren girando los enfriadores para poder disipar el calor dentro del dispensador, para poder establecerlo, se requerirá de pruebas de caracterización en laboratorio, lo cual deberá ser realizado en TT2.

Una vez diseñado el lazo de control, se procedió a proponer los actuadores de enfriamiento, sensor de temperatura y etapa de potencia requeridos para su funcionamiento.

Actuadores de enfriamiento

Debido a que la parte donde se interesa mantener un control de temperatura se resume a todo el apartado donde se encuentran los contenedores de medicamento y que, debido al diseño del dispositivo, comparte espacios con partes de electrónica, se descarta el uso de sistemas de enfriamiento líquido, por otra parte, se considera que debido a que interesa tener acción de control sobre un volumen amplio de espacio, un sistema de enfriamiento por ventilación es el más adecuado para nuestra aplicación, razón por la cual los actuadores de enfriamiento serán ventiladores de enfriamiento.

Dado que, como ya se mencionó, se desconoce cuál será la inercia térmica del dispositivo, así como cuáles serán las condiciones ambientales reales de operación del dispositivo, el

principal factor para la selección de los actuadores será en base a su potencia requerida, su capacidad de enfriamiento, y su voltaje de alimentación ya que debemos tener en cuenta que se está implementando una fuente de poder conmutada de PC que tiene restricciones de tensión y corriente de alimentación.

Se proponen tres diferentes actuadores, mostrados en las imágenes 135, 136 y 137, los cuales serán evaluados para determinar cuál sería el más adecuado al sistema, sin embargo, la selección del actuador deberá esperar al periodo de desarrollo en TT2, pues los factores más cruciales que determinarán cuál seleccionar serán la temperatura máxima interna del dispositivo en condiciones ambientales reales, su inercia térmica y la capacidad de enfriamiento requerida para poder controlar dicha temperatura interna, datos que como se mencionó antes, resulta muy complicado o casi imposible poder establecer de forma adecuada mediante simulaciones.

Ventilador enfriador Cooler Master R4-S2S-124K-GP:



Imagen 135. Ventilador Cooler Master: 12 VCD, 0.15 A, 44.73 CFM, \$299.00MXN. [45]

Ventilador enfriador WINSINN 5V USB



Imagen 136. Ventilador WINSINN USB: 5 VCD, 0.25 A, 72 CFM, \$323.54MXN. [46]

Ventilador Arctic P12



Imagen 137. Ventilador Arctic 12 VCD, 0.20 A, 56.3 CFM, \$474.64MXN. [47]

A continuación, se muestra el proceso de evaluación de los ventiladores propuestos, por una tabla de evaluación de criterios ponderados, la cual es la tabla 20.

Criterio	Peso	Descripción	Unidad	Opciones:					
				Ventilador enfriador Cooler Master R4-S2S-124K-GP		Ventilador enfriador WINSINN 5V USB		Ventilador Arctic P12	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Tension	0.2	Voltaje nominal de trabajo	V	7	1.4	9	1.8	7	1.4
Flujo de aire	0.3	Cantidad de volumen de aire por unidad de tiempo	CFM	7	2.1	10	3	8	2.4
Corriente	0.2	Corriente eléctrica nominal de trabajo	A	9	1.8	7	1.4	7	1.4
Velocidad	0.15	Velocidad angular máxima	RPM	7	1.05	8	1.2	10	1.5
Costo	0.15	Valor monetario del	MXN	8	1.2	7	1.05	6	0.9
Total:	1			38	7.55	41	8.45	38	7.6

Tabla 20. Evaluación de los ventiladores mediante criterios ponderados.

Sensor de temperatura:

Debido a que interesa mantener informado al usuario sobre la temperatura y humedad relativa interna del dispensador, se buscó proponer sensores que tuviesen la capacidad de entregar ambos datos cuando realizan la medición, esto con la intención de reducir gastos y consumo de espacio en el dispositivo.

Se propusieron los siguientes sensores, que se visualizan en las imágenes 138, 139 y 140, de los cuales se realiza un proceso de selección ponderada considerando principalmente su precisión, costo, disponibilidad de mercado y facilidad de implementación.

Sensor DHT11:

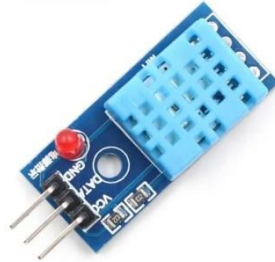


Imagen 138. Sensor de temperatura y humedad con driver DHT11 [48].

Sensor DHT22:

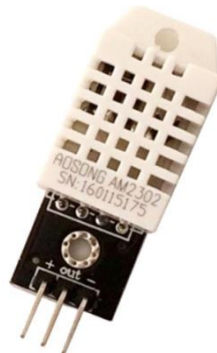


Imagen 139. Sensor de temperatura y humedad con driver DHT22 [49].

Sensor ATM1001:



Imagen 140. Sensor de temperatura y humedad ATM 1001 [50].

En la tabla siguiente (tabla 21), se exponen las características principales de los sensores propuestos:

	DHT11	DHT22	ATM1001
Tension	3.5 a 5 V	3.5 a 5 V	4 a 5.5 V
Corriente	2.5 mA	2.5 mA	2.0 mA
Precio	\$53.00	\$242.00	\$98.50
Temperatura	0 a 50 °C	-40 a 80 °C	0 a 50 °C
Presición	+ - 2°C	+ - 0.5°C	+ - 0.5°C
Humedad	20%-90%	0%-100%	20%-90%

Tabla 21. Principales características técnicas de los sensores propuestos.

A continuación, se muestra el proceso de evaluación de los ventiladores propuestos, visualizado en la tabla 22.

Criterio	Peso	Descripción	Unidad	Opciones:					
				DHT11		DHT22		ATM1001	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Tension	0.15	Tensión nominal	V	7	1.05	7	1.05	6	0.9
Corriente	0.15	Corriente nominal	A	7	1.05	7	1.05	8	1.2
Precio	0.15	Valor monetario del sensor	\$MXN	9	1.35	5	0.75	7	1.05
Temperatura	0.2	Rango de temperatura	°C	7	1.4	9	1.8	7	1.4
Presición	0.2	Precisión de medición de temperatura	°C	7	1.4	9	1.8	9	1.8
Humedad	0.15	Rango de humedad	%HR	7	1.05	9	1.35	7	1.05
Total:	1			44	7.3	46	7.8	44	7.4

Tabla 22. Evaluación de los sensores de temperatura y humedad con criterios ponderados.

Sensor seleccionado: Sensor de temperatura y humedad DHT22

Dado a que el sensor seleccionado cuenta con su propio driver electrónico, no se requiere de un proceso de acondicionamiento de señal físico, sino de un procesado de la señal que entrega por medio de software para poder extraer los datos de interés.

La conexión eléctrica del sensor sería entonces directa a la tarjeta de control y estaría dada de la siguiente forma, como se observa en la imagen 141:

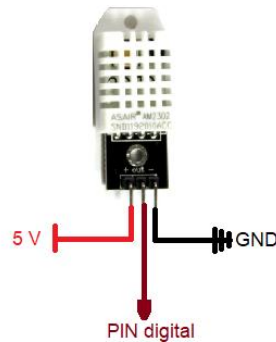


Imagen 141. Conexiones eléctricas del sensor de temperatura y humedad.

Etapa de potencia para los actuadores de enfriamiento

Debido a que se tratan de actuadores que funcionan a partir de motores eléctricos de C.D, no es posible encenderlos directamente a partir de la señal PWM que entrega una etapa control, principalmente, debido a que estas no tienen la capacidad de suministrar la corriente que estos motores consumen, por ello es necesario implementar un circuito de potencia.

Considerando que la etapa de control resulta de complicado reemplazo para este proyecto, se propone una etapa de potencia que integre un opto-acoplamiento de la señal de control, para así poder protegerla contra picos de corriente o tensión que pudieran dañarla.

El diagrama eléctrico propuesto como etapa de potencia es el siguiente (imagen 142):

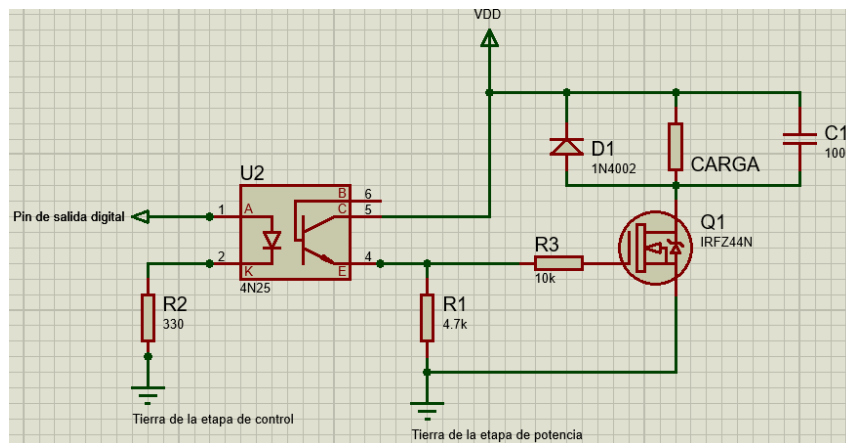


Imagen 142. Circuito eléctrico de la etapa de potencia para el control de temperatura.

Como se aprecia en el diagrama eléctrico anterior (imagen 142), se hace uso de transistores MOSFET IRFZ44N en una configuración que permite polarizarlo en sus diferentes regiones de trabajo a partir de la señal de control opto-acoplada que recibe en su compuerta de entrada “G”, esto a su vez permite encender a un nivel de tensión y corriente a los actuadores el cual depende del porcentaje de PWM que se esté entregando a la etapa de potencia.

Suponiendo un 0% de PWM, se tendría un cero lógico en la compuerta Gate del transistor, lo que a su vez implica que debido a la configuración del MOSFET, el transistor se ubicaría en la región de corte de forma que el actuador no recibe tensión de la compuerta Drain y se mantiene apagado. Por el contrario, suponiendo un 100% de PWM se tendrá un uno lógico en la compuerta Gate, por lo que el transistor se polariza en Saturación, entregando la tensión de alimentación a la que está la compuerta Drain del transistor al actuador y, en consecuencia, encendiéndolo al máximo de su capacidad. A continuación, se muestran las consideraciones que permiten validar esta configuración, para mostrar el símbolo del MOSFET en cuestión, se muestra la imagen 143.

Regiones de operación MOSFET (fórmulas):

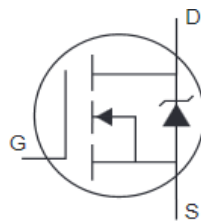


Imagen 143. Símbolo de MOSFET canal n D=Drain, G= Gate, S=Source.

- **Región de Corte:**

$$I_{DS} = 0 \quad \text{si} \quad V_{GS} < V_T$$

- **Región Óhmica**

$$I_{DS} = K_n (V_{GS} - V_T)^2 \quad \text{si} \quad 0 < (V_{GS} - V_T) < V_{DS}$$

- **Región de Saturación**

$$I_{DS} = K_n [2 * (V_{GS} - V_T)V_{DS} - V_{DS}^2] \quad \text{si} \quad (V_{GS} - V_T) > V_{DS}$$

Gráfica de Regiones de Operación MOSFET cuando se tiene una tensión de alimentación de la compuerta Drain a la compuerta Source es constante, esta gráfica se puede ver en la imagen 144:

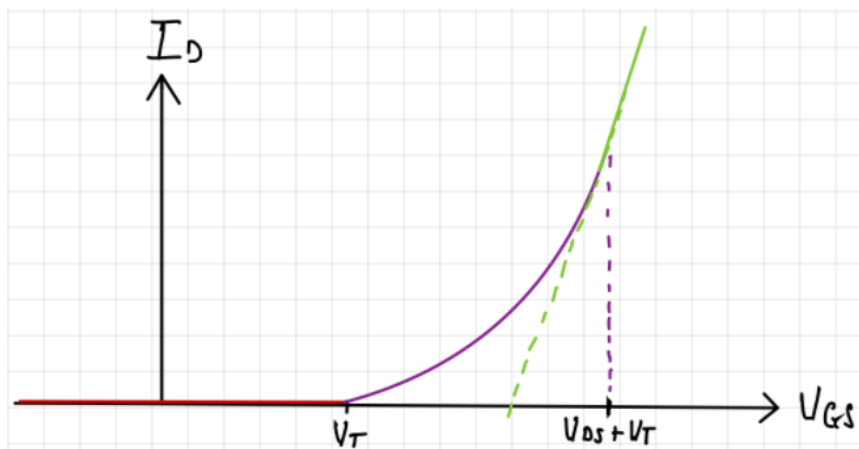


Imagen 144. Regiones de trabajo del MOSFET con $V_{DS} = \text{cte}$ y $V_{GS} = \text{variable}$.

En la imagen 144, se ve que las líneas de la gráfica se dividen en tres colores diferentes, la parte de color rojo corresponde con la región de trabajo denominada Corte, se puede ver que, si la tensión de entrada en la compuerta Gate no supera la tensión de umbral del transistor, la corriente de Drain se mantiene en cero. La línea violeta refiere a la región Óhmica, en esta región a medida que la tensión aumenta, más corriente habrá en Drain. Finalmente, la línea verde corresponde con la región de saturación, en esta región, se alcanza el valor máximo de corriente que puede entregar de acuerdo al valor de tensión de polarización de Gate y Drain.

Para comprobar esto, se realiza la simulación en MULTISIM de un transistor MOSFET con una configuración: $V_{DS} = cte = 10\text{ V}$ y $V_{GS} = variable = 0, \dots, 15\text{ V}$

La configuración antes mencionada se traslada a Multisim, estableciendo el valor adecuado de tensión para la fuente V_{DS} y en el “DC Sweep” configurando el intervalo de barrido de tensión para la fuente V_{GS} . (Ver imágenes 145 y 146)

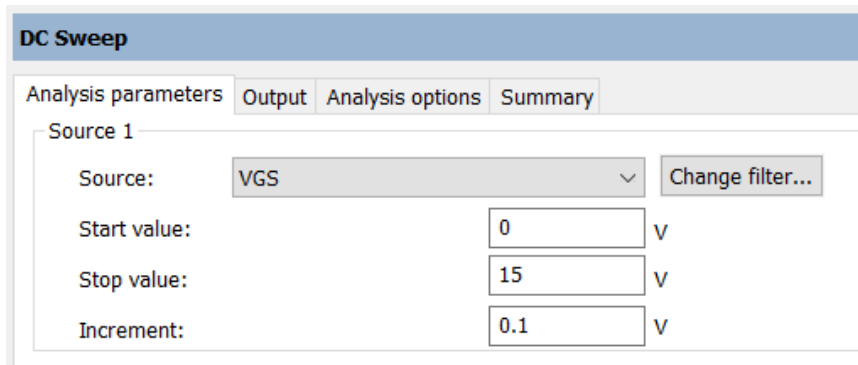


Imagen 145. Configuración de Barrido de segunda configuración.

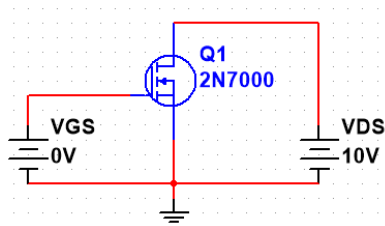


Imagen 146. Segunda configuración de fuentes de tensión del circuito.

Al correr la simulación, se obtiene la siguiente gráfica (imagen 147):

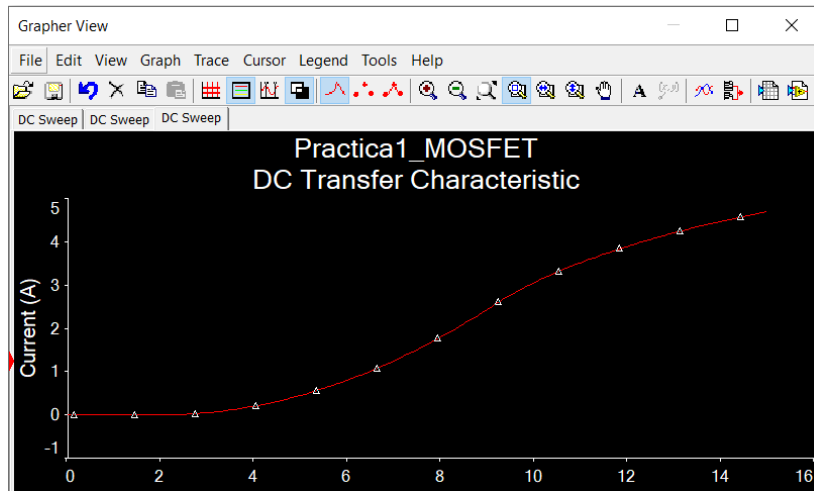


Imagen 147. Curva característica de la configuración del MOSFET.

Los datos de esta simulación pueden ser comparados con los del comportamiento establecido en las ecuaciones correspondientes para cada región descritas anteriormente con la finalidad de corroborar su veracidad. Esto empleando Matlab para ingresar el modelo matemático que describe el modelo del transistor e importando los datos obtenidos en Multisim de forma que se obtiene la siguiente comparación de gráficas (imagen 148):

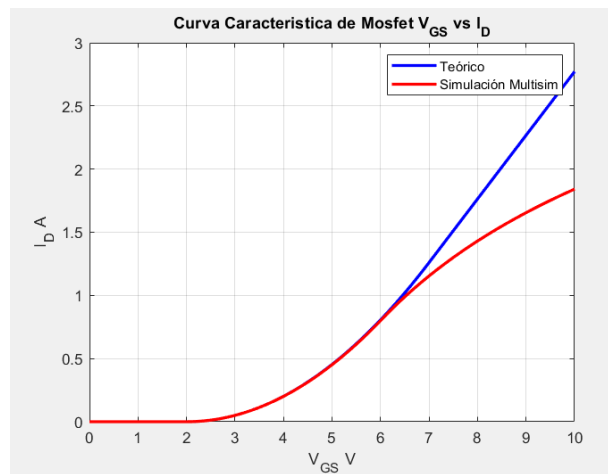


Imagen 148. Comparación de modelo teórico y modelo de Multisim.

La imagen 148 permite apreciar que con un $V_{DS} = 10\text{ V}$ y un $V_{GS} = 0, \dots, 10\text{ V}$ se tiene como corriente teórica máxima $I_{DTmax} \cong 2.7\text{ A}$, mientras que se tiene una corriente de simulación de $I_{DMmax} \cong 1.8\text{ A}$. Como corriente mínima se tiene evidentemente una $I_{Dmin} = 0\text{ A}$. Además, se tienen los valores de corriente intermedios para el resto de los valores de tensión de entrada de la compuerta Gate y vemos que la configuración del circuito permite pasar por las tres regiones de operación a medida que V_{GS} aumenta.

Como es posible observar, los modelos se comportan de forma muy aproximada salvo en la región de saturación, sin embargo, esto acontece debido a que el modelo de Multisim

es uno mucho más detallado que el modelo idealizado descrito en las ecuaciones de las regiones de trabajo anteriormente, sin embargo, esto nos permite validar que efectivamente las regiones de operación y sus comportamientos en sus regiones de trabajo descritas de forma teórica son correctas

Adicionalmente, de acuerdo con su hoja de datos, el MOSFET IRFZ44N tiene una capacidad de corriente continua de drenaje de hasta 49 A y una tensión máxima de hasta 55 V de entrada en la compuerta Drain. De tal forma que el uso del circuito propuesto en la imagen 142 como etapa de potencia para cualquiera de los actuadores de este proyecto no se ve comprometido, ya que ninguno tiene un consumo de corriente o tensión superior a los valores indicados por el fabricante.

En las simulaciones, se emplea un MOSFET 2N700 debido a que en Multisim no se encuentra el empaquetado del MOSFET IRFZ44N, sin embargo, ambos son canal n y por ende responden al mismo comportamiento y forma de operación.

El circuito de opto-acoplamiento emplea un acoplador 4N25, el cual está compuesto por un diodo y un fototransistor, el cual, dependiendo del valor de tensión que recibe el foto diodo permite el paso de corriente y tensión hacia su compuerta de emisor, de esta forma se logra transmitir el PWM a la etapa de potencia sin que exista contacto o conexión directa con la etapa de control, ya que la tierra del fototransistor es la de la etapa de potencia y la del foto diodo la tierra de la etapa de control y la alimentación del opto-acoplador proviene exclusivamente de la etapa de control, mientras que la alimentación para los componentes de la etapa de potencia está separada. Dicho esto, queda establecido que las etapas están aisladas una de la otra quedando protegida la de control contra picos, descargas o cualquier malfuncionamiento en la de potencia.

Recapitulando de la teoría, podemos darnos cuenta de que, efectivamente, esta etapa de potencia puede ser aplicada para una acción de control todo o nada sobre un actuador o para un control mediante PWM, si mantenemos una tensión de drenador constante y controlamos la tensión de la compuerta Gate. Para simplificar su futuro uso en este documento, entiéndase que el símbolo siguiente corresponde al circuito de la imagen 149:

Etapa de Potencia

Imagen 149. Símbolo que se usará para indicar la etapa de potencia para cada actuador.

Actualización del sistema de control de temperatura

- **Reemplazo del sensor de temperatura y humedad**

Debido a la falta de existencia del sensor DHT22 en los puntos de venta más conocidos de la CDMX, el equipo decidió optar por reemplazarlo por un sensor equivalente en cuanto a características de funcionamiento, eligiendo el segundo puesto de la tabla de evaluación de sensores de temperatura y humedad con criterios ponderados (tabla 22), el cual es el DHT11. Este sensor, al igual que el DHT22, cuenta con un driver integrado, y tiene los mismos parámetros de

conexión, cuenta con protocolo de comunicación I2C, las únicas diferencias entre ellos son: el rango de temperaturas y humedad detectables y la precisión de la lectura las cuales pueden observarse en la tabla 21 de comparación de las características de los sensores propuestos.

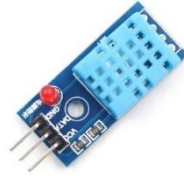


Imagen 150. Sensor de temperatura y humedad con driver DHT11 [48].

El nuevo sensor (imagen 150), permite realizar lecturas desde los 0°C hasta los 48°C, rango dentro del cual se encuentran las temperaturas a las cuales normalmente se encuentra expuesto el dispositivo. Además, cuenta con un rango de detección de humedad relativa del 20% al 90% el cual es también un rango aceptable para la aplicación en curso ya que el dispositivo está diseñado para mantenerse en interiores en los cuales las condiciones de humedad no serán tan agresivas.

- **Reemplazo del actuador de enfriamiento**

Se buscó el adquirir el ventilador seleccionado en TTI, sin embargo, el proveedor que lo comercializaba suspendió la venta del producto por razones que resultan desconocidas. Razón por la cual, el equipo procedió a adquirir un ventilador más comercial en la CDMX, sin embargo, los proveedores como Pigma, no especifican el flujo de aire en pies cúbicos por minuto (CMF) de sus ventiladores, por la que el equipo procedió a determinar primero, de forma aproximada, que capacidad de flujo se requiere en el sistema para tener una acción de enfriamiento, así como el procedimiento para determinar de forma aproximada cuanto flujo de aire tiene un ventilador de acuerdo a sus especificaciones de tamaño y velocidad.

De acuerdo con la bibliografía consultada, para calcular el flujo de aire necesario para tener una acción de enfriamiento y renovación de aire adecuada, se requieren dos valores: el número de cambios de aire por hora (ACH) y el volumen del espacio a ventilar (V). A partir de ellos, puede determinarse los CMF necesarios mediante la expresión:

$$CMF = \frac{ACH * V}{60}$$

Adicionalmente, la bibliografía sugiere que para habitaciones o espacios donde se almacenan medicamentos, se tenga una tasa de al menos seis a diez cambios de aire por hora en habitaciones farmacéuticas y el doble en espacios de almacenamiento. Por otra parte, el volumen total interno dentro del cual se encuentran los contenedores de medicamento es de 0.978 ft³, valor que fue

obtenido por medio de SolidWorks. De forma que empleando la expresión obtenemos lo siguiente:

$$CFM = \frac{20 * 0.978}{60} \left[\frac{ACH * ft^3}{\frac{h}{min}} \right] = 19.56 CFM \cong 20 CFM$$

Teniendo este dato, se consultó a los proveedores sobre un ventilador con la capacidad requerida, sin embargo, sus tamaños eran demasiado grandes para el dispensador, por lo que se optó a emplear 2 ventiladores con una capacidad de flujo de aire de 10 CFM cada uno ya que el flujo total de aire cumple con el teorema de superposición, pudiéndose sumar los flujos para obtener el flujo total. De esta forma el proveedor proporciono un ventilador de 8 cm por lado y un espesor de 2.5 cm cuyo diámetro interno es de 7.8 cm y tiene una velocidad máxima de 2400 rpm. Con estos datos, se puede calcular de forma aproximada el flujo de aire del ventilador, ya que sabemos que, por revolución, el ventilador moverá un volumen de aire:

$$V = 2.5 \text{ cm} * \pi * \left(\frac{7.8 \text{ cm}}{2} \right)^2 = 119.4590 \text{ cm}^3 * \frac{3.53147 * 10^{-5}}{\text{cm}^3}$$

$$V = 4.2186 * 10^{-3} \text{ ft}^3$$

Si consideramos que por cada revolución de mueve el volumen de aire V, entonces:

$$CFM \cong V * 2400 \text{ rpm} \cong 10.1247 \frac{\text{ft}^3}{\text{min}}$$

Por lo tanto, el flujo total del aire teniendo 2 ventiladores iguales sería

$$CFM_{total} \cong 2 * 10.1247 \frac{\text{ft}^3}{\text{min}} \cong 20.2495 \frac{\text{ft}^3}{\text{min}} \cong 20.25 CFM$$

De esta forma, al emplear 2 ventiladores se cumple el requisito de 20 CFM que requiere el sistema de control de temperatura. Cabe mencionarse que el cálculo es aproximado ya que no se toman en cuenta las perdidas intrínsecas del ventilador, sin embargo, dado que la aplicación no quiere una extrema precisión en la regulación de la temperatura, se considera un cálculo adecuado para la implementación en curso.

Finalmente, el ventilador empleado es el BDM8025S que se puede observar en la imagen 151, el cual requiere de una tensión de entrada de 12V y demanda una corriente de 0.16A.



Imagen 151. Ventilador BDM8025S.

Selección de unidad central de procesamiento y tarjeta de control de actuadores

Resulta crucial para el funcionamiento del dispensador el contar con una tarjeta de procesamiento de datos. Esta cumplirá la función de comunicarse con el dispositivo móvil del usuario para que, por medio de este último, sea posible configurar y administrar el prototipo, transmitirse los parámetros de operación de temperatura y humedad a las interfaces de usuario, y activar/desactivar las subrutinas que controlan los actuadores del aparato.

Dado que se tienen procesos que requieren conexión a red de internet, adquisición y procesamiento de imágenes, e interfaces de usuario gráficas, se busca implementar una tarjeta de procesamiento que cuente con los atributos necesarios para soportar dichas operaciones.

Las características técnicas deseadas en la tarjeta serían las listadas a continuación:

- Capacidad de conexión a internet
- Capacidad de comunicación serial
- Soporte para conexión de cámaras
- Conexiones para transmisión de video (DSI y HDMI)
- Conexiones USB tipo A para transmisión de datos

Otros aspectos a considerar para su selección serán la tensión requerida para encender la tarjeta, así como su costo ya que el presupuesto del proyecto saldrá de la bolsa de los desarrolladores.

Se propusieron las siguientes tarjetas de procesamiento de datos, que se observan en las imágenes 152, 153 y 154, las cuales a la vez se consideran para seleccionar la tarjeta de control de actuadores. Empezando con la selección de la tarjeta de procesamiento de datos, esto mediante un proceso de selección ponderada considerando principalmente si cuenta con conexión a internet, comunicación serial, conexión para pantalla touch y conexión para cámara.

Tarjeta Raspberry Pi 3 b+:



Imagen 152. Tarjeta de procesamiento de datos Raspberry Pi 3 b+. [51]

Tarjeta Nucleo-F446RE:

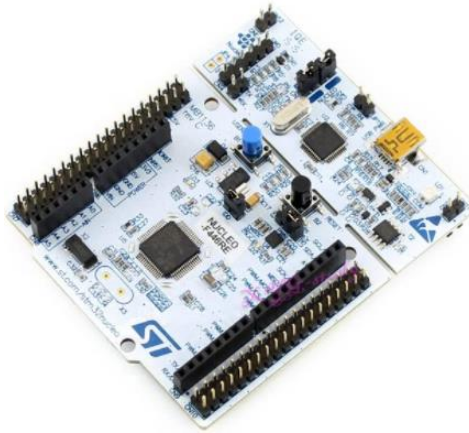


Imagen 153. Tarjeta de procesamiento de datos Núcleo-F446RE. [52]

Tarjeta Arduino Mega 2560:



Imagen 154. Tarjeta de procesamiento de datos Arduino Mega 2560. [53]

En la tabla 23, que se muestra a continuación, se exhiben las características de las tarjetas de procesamiento de datos propuestas para su evaluación.

	Nucleo f466	Arduino MEGA 2560	Raspberry Pi3 b+
Tensión	3.3 - 5 V	5 V	5 V
HDMI	No	No	Si
WiFi	No	Modulo externo	Si
Seríal COM	Si	Si	SI
Precio	\$538.73	\$792.88	\$1,040
DSI (Pantalla touch)	No	No	Si
Conexión Cámara	No	No	Si
USB A	No	No	Si

Tabla 23. Características principales de las tarjetas de procesamiento propuestas.

La evaluación de las tarjetas propuestas se realiza empleando el método de criterios ponderados, de esta manera, se logra determinar cuál sería la más adecuada para el

proyecto de acuerdo a las necesidades del mismo. La evaluación obtenida se puede apreciar en la siguiente tabla 24.

Criterio	Peso	Descripción	Unidad	Opciones:					
				Nucleo f466		Arduino MEGA 2560		Raspberry Pi3 b+	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Tensión	0.05	Tensión nominal	V	9	0.45	8	0.4	8	0.4
HDMI	0.1	Conexión de salida HDMI	Si/No	0	0	0	0	10	1
WiFi	0.2	Conexión de internet	Si/Modulo externo/No	7	1.4	8	1.6	0	0
Serial COM	0.15	Comunicación serial	Si/No	10	1.5	10	1.5	10	1.5
Precio	0.1	Valor monetario	MXN	7	0.7	9	0.9	5	0.5
DSI	0.15	Cuenta con conexión para Pantalla	Si/No	0	0	0	0	10	1.5
Camara	0.15	Soporta conexión de cámara	Si/No	0	0	0	0	10	1.5
USB A	0.1	Cuenta con puertos USB tipo A	Si/No	0	0	0	0	10	1
Total:	1			33	4.05	35	4.4	63	7.4

Tabla 24. Evaluación de las tarjetas de procesamiento de datos para ser la unidad central de procesamiento, mediante criterios ponderados.

De la tabla 24, se puede observar que la opción con mejor evaluación es la tarjeta Raspberry Pi 3 b+, por lo que, ésta será la que se usará para la unidad central de procesamiento.

Ahora para lo que refiere a la tarjeta de control, la cual debe tener manejo de cada actuador en los diferentes sistemas del dispensador. Para esto se proponen las mismas tarjetas que fueron evaluadas para la selección de la unidad central de procesamiento, pero tomando en cuenta a los siguientes criterios:

- Tensión nominal.
- Pines usados como entrada o salida.
- Pines usados para la salida de PWM.
- Comunicación serial.
- Valor monetario.

En la siguiente tabla 25, se muestran los datos referentes a los criterios de las tarjetas a evaluar.

	Nucleo f466	Arduino MEGA 2560	Raspberry Pi3 b+
Tensión	3.3 - 5 V	5 V	5 V
Pines GPIO	50GPIO	54GPIO	40 GPIO
Pines PWM	9 PWM	14 PWM	No
Serial COM	Si	Si	Si
Precio	\$538.73	\$792.88	\$1,040

Tabla 25. Principales características de las tarjetas que se proponen para ser la tarjeta de control.

De los criterios mencionados, los que más son de interés son: pines GPIO, pines PWM y la comunicación serial. Estos debido a que es de suma importancia tener pines para controlar las señales de salida de los actuadores y tener los suficientes pines de conexión para los mismos. Por lo que, en la tabla 26, se realiza la evaluación de las tarjetas antes mencionadas, pero ahora con diferentes criterios y ponderaciones.

Criterio	Peso	Descripción	Unidad	Opciones:					
				Nucleo f466		Arduino MEGA 2560		Raspberry Pi3 b+	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Tensión	0.1	Tensión nominal	V	9	0.9	8	0.8	8	0.8
Pines GPIO	0.25	Pines usados como entrada o salida	Pines	8	2	9	2.25	6	1.5
Pines PWM	0.25	Pines usados para salida de PWM	Pines	7	1.75	8	2	0	0
Serial COM	0.25	Comunicación serial	Si/No	10	2.5	10	2.5	10	2.5
Precio	0.15	Valor monetario	MXN	7	1.05	9	1.35	5	0.75
Total:	1			41	8.2	44	8.9	29	5.55

Tabla 26. Evaluación de las tarjetas de procesamientos de datos para ser la tarjeta de control, mediante criterios ponderados.

De la tabla mostrada (tabla 26), se puede visualizar que la tarjeta que resultó mejor evaluada para ser la tarjeta de control es el Arduino Mega 2560, dado que cumple con cierta cantidad de pines para los actuadores a utilizar, además de que el precio comercial es más accesible por el equipo que para las demás.

Actualización de unidad de procesamiento de datos

El equipo ha revalorado el diseño del sistema de procesamiento de datos siguiendo el consejo del jurado calificador y, dado que la tarjeta Raspberry Pi3B+ cuenta con una gran variedad de puertos GPIO, se ha decidido remover la tarjeta de control Arduino ATMEGA2560 y realizar las operaciones que esta última realizaba en la tarjeta de las cerezas. Esto reducirá los tiempos de operación entre funciones pues el programa se comunicará de forma interna en un solo procesador, además, se recupera espacio en el dispositivo y permite que el programa sea más ordenado y eficiente.

Nota: Dicho lo anterior, se están realizando modificaciones a los circuitos eléctricos para adaptarlos a las nuevas conexiones de la tarjeta Raspberry, pues esta trabaja a una tensión de 3.3V y no cuenta con convertidor analógico digital (ADC) a diferencia de la Arduino que funciona con 5V en sus puertos GPIO y si cuenta con puertos con ADC.

Sistema de alarma para ingesta de comprimido o cápsula

Para este sistema, lo que se propone es usar un buzzer el cual se activará mediante la señal del microcontrolador al momento que llegue la hora de tomar un comprimido, tableta o cápsula correspondiente. Para lograr que a una hora en específico el buzzer empiece a

sonar, lo que se hará es comparar la hora a la que se empezó el tratamiento con el medicamento en cuestión, y sumarle las horas de espera para volver a tomar el medicamento; entonces se tomará la hora en tiempo real que marque el celular del usuario y la comparará con el resultado de la suma de la hora de inicio del tratamiento más las horas de esperas. Por lo que, si coinciden, en ese momento se dispensará la pastilla y el agua para ingerirla, para después activar el buzzer y realice la secuencia siguiente: comenzará a emitir sonido para alertar al usuario y que tome su medicamento, dicho alerta durará un tiempo de 3 segundos en los cuales se emitirá un sonido durante 500 ms, dejará de sonar 500 ms y después volverá a sonar cumpliendo el mismo patrón hasta cumplir con los 3 segundos; al terminar esta secuencia, dejará de sonar 2 segundos y luego repetirá el patrón 2 veces más. Como resultado, el tiempo que estará funcionando el sistema de alarma durará un total de 13 segundos. Esperando que este tiempo el usuario ya haya ingerido su pastilla correspondiente y el programa principal del dispensador asigne como tiempo de inicio la hora a la que fue dispensado el medicamento para volver a sumar el tiempo de esperar, y así sucesivamente hasta que el tratamiento esté cumplido.

El buzzer que se pretende utilizar es un transductor piezoeléctrico, estos últimos tienen la propiedad de variar su volumen al ser atravesados por una corriente eléctrica, este hecho es aprovechado para hacer vibrar una membrana al pasar el material piezoeléctrico con una señal eléctrica. Esta composición de los buzzers se puede apreciar en la siguiente imagen 155.

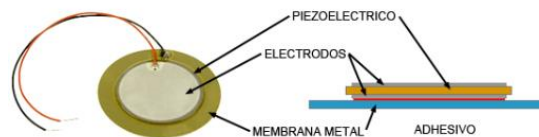


Imagen 155. Composición del buzzer.[54]

Como se puede notar son dispositivos compactos y pequeños, con bajo consumo eléctrico y la calidad de sonido es reducida. En las imágenes 156 y 157, se muestra un buzzer comercial y las especificaciones técnicas para determinar su potencia y observar los decibeles que ofrece.



Imagen 156. Buzzer comercial que se pretende usar. [55]

Mini Buzzer 3.3 a 5 V especificaciones técnicas:

- Voltaje recomendado de funcionamiento: 3.3 a 5 V
- Voltaje rango de operación: 3 a 7 V
- Consumo de corriente: <30mA
- Salida de sonido: >85 dB
- Frecuencia generada: 2.3 KHz +- 400 Hz
- Temperatura de Operación: -40° a 85° C
- Diámetro de 1.2 cm
- Peso: 2g

Imagen 157. Especificaciones técnicas del buzzer comercial. [55]

De la imagen 157 cabe resaltar su tensión de funcionamiento, que tomando la máxima que es de 5V, y la corriente que consume la cual para fines de cálculo de potencia se tomará el valor de 29 mA. Ahora calculando la potencia que consume:

$$P = V * I = 5V * 29mA = 0.145 W$$

Ahora, consultando una bocina altavoz comercial con la cual se comparará con el buzzer mencionado. Se muestra tanto la bocina y sus respectivas especificaciones técnicas para poder ver su potencia y los decibeles que otorga. Esto se muestra en las imágenes 158 y 159.



Imagen 158. Bocina altavoz comercial tentativa para usar. [56]

Acerca de este artículo

- Diámetro: 52mm
- Respuesta de frecuencia: 160Hz-20kHz
- Impedancia: 4 Ω
- Potencia nominal: 10W
- Sensibilidad: 82 (dB / W)

Imagen 159. Especificaciones técnicas de la bocina altavoz comercial. [56]

Como se puede apreciar, de las imágenes 158 y 159, la bocina tiene mayor potencia que el buzzer, los decibeles que ofrece son mayores también y trabaja a un rango de frecuencias mayor que las del buzzer, la única desventaja es el tamaño que son 52 mm de diámetro contra los 12 mm de diámetro del buzzer, que para beneficio del proyecto no debe ser grande este componente.

Dado que la bocina tiene mayor tamaño, mayor calidad de sonido y mayor potencia, debido a su composición de un imán permanente acoplado a una carcasa y además una bobina móvil que se acopla a una membrana flexible. Se tomó la decisión de elegir un buzzer debido a que cumple la función de emitir un sonido, que no importar la calidad, solo el hecho de que emita sonido y mejor que sea de menor tamaño. [54]

Para ejemplificar mejor este algoritmo de alarma se presenta el siguiente diagrama de flujo (Imagen 160) explicando los pasos a seguir para activar la alarma y su respectiva secuencia de sonido.

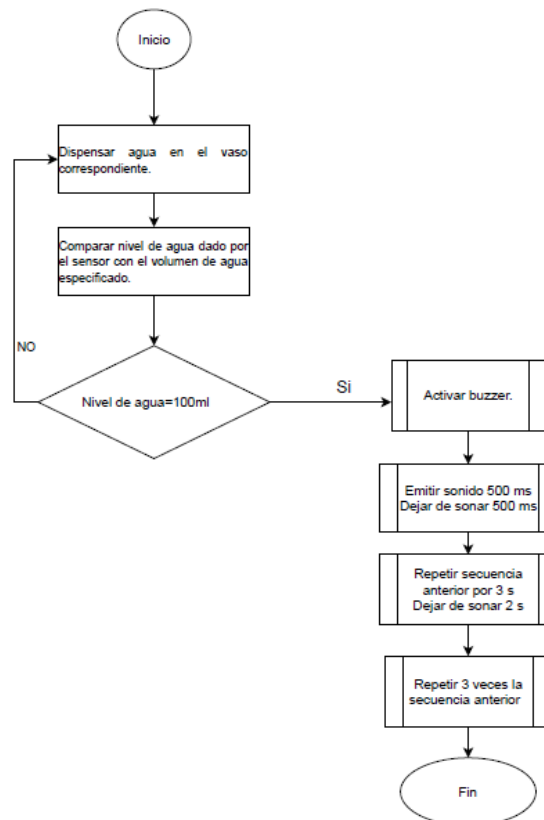


Imagen 160. Diagrama de flujo para el sistema de alarma.

La conexión del buzzer con el microcontrolador es sencilla dado que se conecta directamente, solo el pin positivo del buzzer a la salida digital y el pin negativo a la tierra del microcontrolador. Esto se puede apreciar en la siguiente imagen 161.

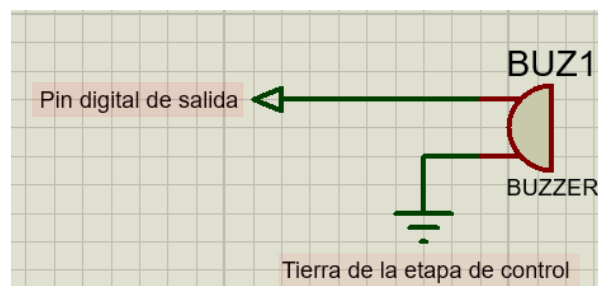


Imagen 161. Diagrama eléctrico de la conexión del buzzer al microcontrolador.



Cabe mencionar que debido a que no se cuenta con los materiales para determinar la frecuencia y escuchar un volumen de sonido al cual lo pueda escuchar el usuario y no le incomode, esto a partir de pruebas experimentales que no se pueden realizar en estos momentos. Por esta razón, la validación de sonido por frecuencia para el buzzer en cuestión se realizará en el trabajo terminal 2.

Operación del Sistema

Pseudocódigo de operación del dispensador

1. Encendido del Dispensador
 - Encendido general del dispositivo
 - Activar los sensores de temperatura y humedad, nivel de agua del tanque de almacenamiento
2. Ventana emergente de inicio (solicitud de vinculación a dispositivo móvil)
 - Mostrar dirección IP del dispositivo y las siguientes opciones
 - [1] No conectar
 - Deshabilitar transmisión de datos
 - Pasar a proceso 3
 - [2] Aceptar
 - 2.1- Esperar solicitud de vinculación del teléfono móvil vía dirección IP
 - 2.2- Aceptar automáticamente la vinculación
 - 2.3- Pasar a proceso 3
3. Mostrar datos en la interfaz de usuario informativa
 - 3.1- Lectura de temperatura interna, porcentaje de humedad relativa, nivel de agua del contenedor
 - 3.2- Lectura de datos de tratamientos actualmente registrados
 - 3.3- Desplegar información de 3.1 y 3.2 en la interfaz
 - 3.4- Si temperatura máxima- lectura de temperatura=error=10
 - si $\text{error} \leq 0$ no activar ventilación
 - si $0 < \text{error} < 4$ activar ventilación a %PWM 1
 - si $4 < \text{error} < 8$ activar ventilación a %PWM 2
 - si $8 < \text{error}$ activar ventilación a 100%
 - 3.5- Si lectura de nivel de agua < 330 ml (1/3 de la capacidad del tanque)
 - + Activar buzzer con secuencia de sonido preestablecida
 - 3.6- Una vez desplegada la interfaz se mantiene siempre activa
4. Esperar solicitud de información, registro de tratamiento o modo viaje
 - {1} Solicitud de registro de tratamiento por parte del usuario
 - Pasar a proceso 5
 - {2} Solicitud de gestionar tratamientos activos por parte del usuario
 - Pasar a proceso 10
 - {3} Solicitud de información por parte del usuario
 - Pasar a proceso 6
 - {4} Solicitud de modo viaje por parte del usuario
 - Pasar a proceso 8

Si no existe ninguna solicitud no se realiza ninguno de los procesos vinculados.



5. Activar secuencia de registro de tratamiento
 - 5.1-Solicitar datos de la aplicación móvil (Nombre del tratamiento, número de pastillas a ingresar, periodo de ingesta, duración del tratamiento, hora de registro del tratamiento, tratamiento permanente [Si/No])
 - 5.2-Crear el nuevo tratamiento con sus atributos (nombre de tratamiento, pastillas disponibles=pastillas a ingresar, duración del tratamiento, hora de ingesta= hora de registro del tratamiento + periodo de ingesta)
 - 5.2-Encender cámara de reconocimiento
 - 5.3-Mostrar mensaje para solicitar ingreso de medicamento en la interfaz del dispensador
 - 5.4-Sensar ingreso de medicamento (Si no se detecta el ingreso, volver a paso anterior)
 - 5.5-Quitar mensaje de ingreso de medicamento
 - 5.6-Inicia reconocimiento de medicamento (ir a proceso 7)
 - 5.7- Secuencia If-else-if para procesar los resultados del reconocimiento
 - Si es pastilla
 - Encender motor a pasos del revolver
 - Secuencia de pasos para colocar contenedor de pastillas en modo entrada
 - Si es tableta
 - Encender motor a pasos del revolver
 - Secuencia de pasos para colocar contenedor de pastillas en modo entrada
 - Si es capsula
 - Encender motor a pasos del revolver
 - Secuencia de pasos para colocar contenedor de pastillas en modo entrada
 - Ninguna
 - Encender motor a pasos del revolver
 - Secuencia de pasos para colocar contenedor de desechos en modo entrada
 - Desplegar mensaje temporal de 2 segundos de “objeto invalido”
 - 5.8- Activar solenoide para permitir el paso del medicamento a su contenedor
 - 5.9- Incrementar en una unidad el registro de pastillas ingresadas
 - 5.10- Comparar si el número de pastillas ingresadas el igual al número de pastillas requeridas
 - Si Pastillas ingresadas < Pastillas a ingresar
 - Volver a paso 5.3
 - Si pastillas ingresadas >= Pastillas a ingresar
 - Fin de la subrutina de ingreso de medicamento
6. Envío de información a aplicación móvil
 - 6.1- Recabar información de monitoreo
 - Temperatura, porcentaje de humedad
 - Información de tratamientos activos



- + Nombre del tratamiento
- + Numero de pastillas disponibles
- 6.2- Enviar datos recabados al dispositivo móvil del usuario a través de la conexión IP establecida (Temperatura, porcentaje de humedad, Información de tratamientos activos)
- 6.3- Fin de subrutina de envío de información
- 7. Reconocimiento de medicamento
 - 7.1- Adquirir imagen de la cámara
 - 7.2- Filtrado y tratamiento de la imagen
 - 7.3- Obtención de los momentos invariantes de Hu de la imagen filtrada
 - 7.4- Procesar momentos de Hu mediante red neuronal simple de una capa para distinguir entre: pastilla, tableta o capsula.
 - 7.5- Volver a proceso 5
- 8. Modo viaje
 - 8.1- Recibir información de la aplicación
 - Número de días ausente
 - 8.2- Calcular pastillas necesarias de cada tratamiento para los días ausentes considerando sus periodos de ingesta
($\text{int}(\text{Número de días} * 24 / \text{periodo de ingesta}) = \text{pastillas necesarias}$)
 - 8.3- Dispensar las pastillas requeridas
 - Repetir para número de tratamientos activos
 - + Activar motor a pasos del revolver para colocar los contenedores de los tratamientos activos en modo de salida.
 - + Repetir para número de pastillas necesarias:
 - * Activar secuencia del motor a pasos del extractor para un giro completo (permite extraer un comprimido o capsula del contenedor donde están almacenadas)
 - * Activar solenoide de extracción de pastilla (permite entregar el comprimido o capsula extraída al usuario)
 - + Descontar a los contadores de pastillas disponibles de los tratamientos el número correspondiente de pastillas necesarias para el modo ausente
 - Fin de la subrutina de modo viaje
- 9. Dispensado de medicamentos y agua
 - 9.1- Verificar si hora actual == Horario de ingesta de medicamento de tratamientos
 - Si se cumple la igualdad pasar a subproceso 9.2
 - Si no se cumple la igualdad no se realiza ningún proceso de dispensado
 - 9.2- Extracción de medicamento requerido
 - Identificar contenedor del medicamento requerido
 - Activar motor a pasos del Revolver para colocar el contenedor identificado en modo de salida
 - Activar motor a pasos del extractor para un giro completo (permite extraer un comprimido o capsula del contenedor donde están almacenadas)
 - Activar solenoide de extracción de pastilla (permite entregar el comprimido o capsula extraída al usuario)
 - Reducir el contador de pastillas disponibles del tratamiento



- Horario de ingesta de medicamento de tratamientos+/-periodo de ingesta
- Pasar a subproceso 9.3
- 9.3- Dispensado de agua para el usuario
 - Activar sensor de nivel de agua del vaso para el usuario
 - + Desactivar válvula para cerrar el paso de agua
 - + Desactivar sensor de nivel de agua del vaso
 - Pasar a subproceso 9.4
- 9.4- Activación de alarma de ingesta de medicamento
 - Activar buzzer con secuencia de sonido preestablecida
- Fin de la secuencia de dispensado de medicamentos y agua
- 10. Gestionar tratamientos activos
 - Recibir información de la aplicación del usuario:
 - {1} Solicitud de borrar tratamiento
 - Pasar al proceso 11.
 - {2} Solicitud de Reabastecer tratamiento
 - Pasar al proceso 12
 - {3} Solicitud de editar configuración de un tratamiento
 - Pasar al proceso 13
- Si no existe ninguna solicitud no se realiza ninguno de los procesos vinculados.
- 11. Borrar tratamiento
 - Recibir información de la aplicación móvil (Nombre del tratamiento)
 - Si pastillas disponibles del tratamiento a borrar > 0
 - + Activar motor a pasos del revolver para posicionar el contenedor del tratamiento a borrar en modo de salida
 - + Repetir para número de pastillas disponibles:
 - * Activar secuencia del motor a pasos del extractor para un giro completo (permite extraer un comprimido o capsula del contenedor donde están almacenadas)
 - * Activar solenoide de extracción de pastilla (permite entregar el comprimido o capsula extraída al usuario)
 - + Igualar el contador de pastillas disponibles del tratamiento a cero
 - Destruir tratamiento y sus atributos
- 12. Reabastecer tratamiento
 - Recibir información de la aplicación móvil (Nombre del tratamiento, pastillas a ingresar)
 - Activar secuencia del motor a pasos del revolver para posicionar el contenedor del tratamiento a reabastecer en modo de entrada
 - Encender solenoide de compuerta de entrada de medicamentos
 - Mostrar en la interfaz de usuario mensaje de ingresar medicamentos
 - Repetir el número de pastillas a ingresar:
 - Censar ingreso de medicamento
 - +incrementar en una unidad pastillas disponibles
 - Desactivar el solenoide de ingreso de medicamentos



- Quitar el mensaje de ingresar medicamentos de la interfaz de usuario
 - Mostrar mensaje de pastillas almacenadas con éxito en la interfaz de usuario por 2 segundos
 - Fin del proceso de abastecer medicamentos
13. Editar configuración de un tratamiento
- Recibir información de la aplicación móvil (Nombre del tratamiento recibido=Nombre del tratamiento, periodo de ingesta, duración del tratamiento, tratamiento permanente [Si/No])
 - Encontrar el tratamiento a modificar comparando el nombre del tratamiento recibido con los nombres de los tratamientos activos.
 - Modificar los atributos del tratamiento con nombre coincidente con los datos recibidos de la aplicación móvil

Pseudocódigo de operación de la aplicación móvil

1. Inicio de la Aplicación
2. Ventana emergente de inicio (solicitud de vinculación a dispositivo móvil)
 - Solicitar dirección IP del dispensador y mostrar las siguientes opciones
 - [1] Cancelar
 - Deshabilitar opciones del menú principal
 - Preservar funciones de alertas
 - [2] Aceptar
 - Enviar solicitud de vinculación IP a través de la red local del usuario
 - Al aceptarse la vinculación, habilitar opciones y funciones del menú principal
3. Menú principal
 - [1] Ingresar nuevo tratamiento
 - Mostrar ventana de ingreso de nuevo tratamiento
 - Solicitar ingreso de los datos necesarios (Nombre del tratamiento, número de pastillas a ingresar, periodo de ingesta, duración del tratamiento, hora de registro del tratamiento, tratamiento permanente [Si/No])
 - Al clicar botón de guardar cambios pasar al subproceso 5.1
 - Al clicar botón de atrás regresa a menú principal (si no se clicó la opción de guardar cambios, se descartan automáticamente)
 - [2] Gestionar tratamientos activos
 - Mostrar ventana de gestión de tratamientos activos
 - [2.1] Borrar tratamiento
 - Solicitar seleccionar tratamiento en la ventana
 - Al clicar botón de borrar pasa al subproceso 5.1
 - [2.2] Reabastecer tratamiento



- Solicitar seleccionar tratamiento en la ventana
- Al clicar botón de borrar pasa al subproceso 5.2
- [2.3] Editar configuración del tratamiento
 - Solicitar que se ingrese en la ventana los nuevos datos de configuración del tratamiento (Seleccionar un tratamiento, periodo de ingesta, duración del tratamiento, tratamiento permanente [Si/No])
 - Al clicar botón de guardar cambios pasa al subproceso 5.3
- [3] Observar funcionamiento del dispositivo
 - Mostrar ventana de monitoreo del dispensador
 - Pasa al proceso 6
- [4] Modo viaje
 - Mostrar ventana para configurar el modo de viaje
 - Solicitar en la ventana el número de días fuera del hogar
 - Al clicar botón de aceptar pasa a proceso 7
- {1} Alerta de reabastecimiento
 - Si el número de pastillas disponibles de algún tratamiento es ≤ 3
 - + Activar notificación en el teléfono del usuario que le indique los tratamientos con 3 pastillas o menos y que debe reabastecerlos
 - Si no existe ningún tratamiento con menos de 3 pastillas disponibles no se activa ninguna notificación.
- {2} Alerta de ingesta de medicamento
 - Verificar si hora actual == Horario de ingesta de medicamento de tratamientos
 - + Activar notificación en el teléfono del usuario que le indique que es momento de ingerir un medicamento.
 - + Horario de ingesta de medicamento de tratamientos \pm periodo de ingesta
 - Mientras la igualdad no se cumpla no se activa ninguna notificación.
- 4. Ingresar nuevo tratamiento al sistema
 - Crear nuevo tratamiento local tomando como atributos los datos solicitados en la ventana de ingreso de medicamentos
 - Enviar solicitud de ingresar nuevo tratamiento al dispensador vía red local del usuario
 - Al aceptarse solicitud, enviar los datos del nuevo tratamiento al dispensador a través la red local del usuario. (Nombre del tratamiento, número de pastillas a ingresar, periodo de ingesta, duración del tratamiento, hora de registro del tratamiento-hora actual del teléfono del usuario, tratamiento permanente [Si/No])
- 5. Gestionar tratamientos activos
 - 5.1 Borrar un tratamiento
 - Enviar solicitud de borrar un tratamiento



- Al aceptarse solicitud, enviar el nombre del tratamiento a borrar al dispensador a través de la red local de usuario
 - Destruir el tratamiento local y todos sus atributos
 - Fin del subproceso de borrar tratamiento
- 5.2 Reabastecer un tratamiento
- Obtener las pastillas necesarias para reabastecer el tratamiento (pastillas a ingresar = 15-Pastillas disponibles)
 - Enviar solicitud de reabastecer un tratamiento al dispensador vía red local del usuario
 - Al aceptarse solicitud, enviar al dispensador los datos de reabastecimiento (Nombre del tratamiento, pastillas a ingresar)
 - fin del subproceso de reabastecer tratamiento
- 5.3 Editar configuración de un tratamiento
- Enviar solicitud de editar la configuración de un tratamiento
 - Al aceptarse solicitud, enviar al dispensador los nuevos datos de configuración del tratamiento (Nombre del tratamiento seleccionado, periodo de ingesta, duración del tratamiento, tratamiento permanente [Si/No])
 - Actualizar los atributos locales del tratamiento seleccionado con los nuevos datos (Periodo de ingesta, duración del tratamiento, tratamiento permanente [Si/No])
 - Fin del subproceso de editar configuración del tratamiento
6. Observar funcionamiento del dispositivo
- Enviar solicitud de información al dispensador
 - Recibir datos del dispensador (Temperatura, porcentaje de humedad, Información de tratamientos activos)
 - Desplegar datos recibidos en la ventana de monitoreo de la interfaz del teléfono del usuario
 - Este proceso se repite cada 5 segundos hasta que el usuario abandone la ventana.
 - Al clicar botón de atrás
 - + Eliminar los últimos datos recibidos de temperatura y humedad
 - + Actualizar los atributos de los tratamientos activos con los últimos datos recibidos de los tratamientos activos
 - + Regresar al menú principal
7. Modo viaje
- Enviar solicitud de información al dispensador
 - Al aceptarse solicitud, enviar número de días ausente al dispensador a través de la red local de usuario
 - Calcular número de pastillas necesarias de cada tratamiento activo ($\text{int}(\text{Número de días ausente} * 24 / \text{periodo de ingesta}) = \text{pastillas necesarias}$)
 - Restar a los contadores de pastillas disponibles de los tratamientos el número correspondiente de pastillas necesarias para el modo ausente
 - Regresar al menú principal

Interfaces de Usuario

En un sistema o dispositivo, sobre todo si debe interactuar con un ser humano para configurarse y/o funcionar, es siempre indispensable que exista un medio de comunicación que funcione como puente entre el usuario y el aparato que es a lo que se conoce como Interfaz de Usuario (UI por sus siglas en inglés).

Existen diversos tipos de interfaz de usuario, sin embargo, para el caso particular de este proyecto, se considera que emplear Interfaces de Usuario Gráficas (GUI por sus siglas en inglés) es la opción más adecuada, sobre todo teniendo en cuenta que se tiene como objetivo que el dispensador pueda ser configurado y monitoreado desde un teléfono celular y, para ello, se requerirá desarrollar una aplicación móvil de fácil uso que tenga la posibilidad de interactuar con el usuario a través de la pantalla de su teléfono.

Además, se contempla la posibilidad de que exista una interfaz gráfica en el dispensador cuyo fin sería principalmente de monitoreo. Se busca que al igual que la interfaz de la app, esta interfaz sea visual e intuitiva para la mayoría de los usuarios.

Actualmente, el diseño de interfaces de usuario gráficas ha evolucionado mucho, de tal forma que no es estrictamente necesario construirlas a partir de código fuente en CSS, Android o IOS, sino que se pueden emplear plataformas de diseño como Figma que permite desarrollarlas a partir de recursos gráficos en un estilo Canva y que además permite obtener el código de lo que se haya diseñado. El equipo consideró que esta era la mejor alternativa para diseñar las interfaces requeridas por el proyecto, por lo que se procedió a diseñarlas en Figma, a continuación, se presentan los resultados obtenidos.

o Interfaz de la Aplicación Móvil

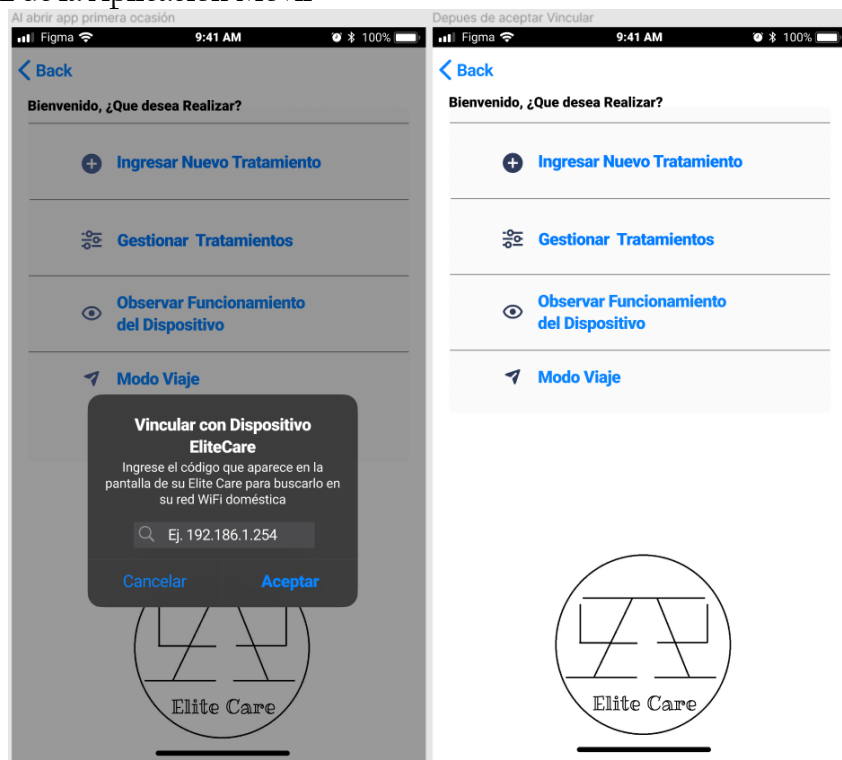


Imagen 162. Interfaz de la aplicación móvil.

Como se aprecia en la imagen 162, las primeras 2 pantallas muestran la interfaz de usuario principal, en ella el usuario puede seleccionar entre las distintas funciones disponibles para configurar u observar el funcionamiento del dispensador.

La imagen de la izquierda muestra cómo se presentaría la pantalla al iniciar la aplicación, se muestra que lo primero que realiza el dispositivo móvil es tratar de establecer conexión a través de la red local del usuario con el dispensador para poder posteriormente acceder a las diferentes funciones del aparato.

La imagen de la derecha muestra la interfaz de usuario principal la vez establecida la conexión entre el teléfono del usuario y el dispensador. En ésta también es posible observar que se cuenta con un menú selector que permite elegir cuál de las funciones principales se desea realizar.

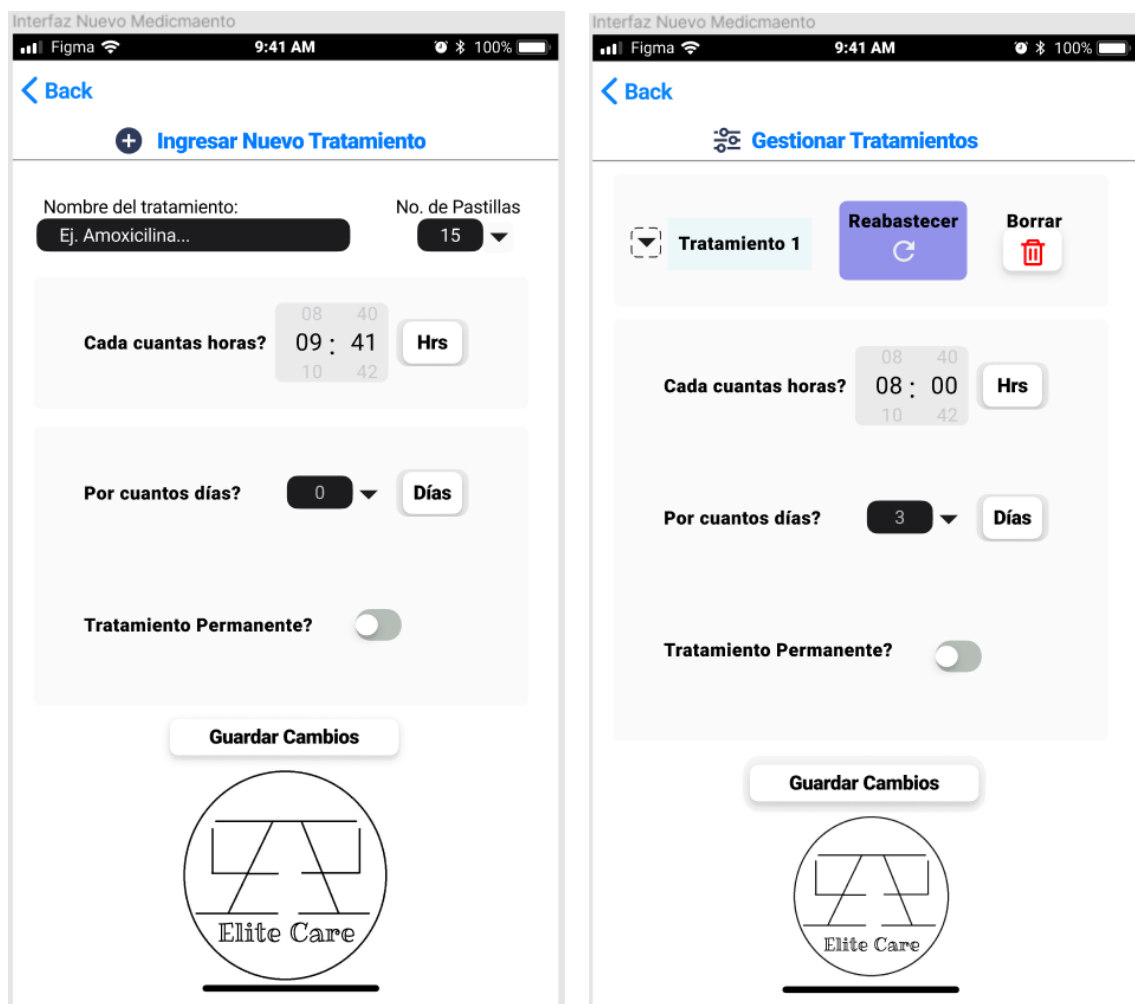


Imagen 163. Interfaces de ingresar nuevo tratamiento y de gestionar tratamientos.

De la imagen 163, en la imagen de la izquierda, se observa la sub interfaz de la función de “Ingresar Nuevo Tratamiento”, esta interfaz permitirá al usuario configurar todo lo relacionado a un nuevo tratamiento al cual está siendo sometido. La interfaz permite ingresar entre otras cosas, el número de pastillas que se desea almacenar, el periodo de

tiempo en que debe tomar el medicamento, por cuantos días deberá consumirlo o, incluso, habilitar la opción de tratamiento permanente para descartar un número de días y que se considere como un tratamiento siempre activo.

Por otra parte, la imagen de la derecha se muestra la sub interfaz de la función de “Gestionar Tratamientos”, ésta, permitirá al usuario realizar modificaciones sobre la configuración de los tratamientos activos del usuario. Como se observa, la interfaz permite al usuario seleccionar el tratamiento a modificar, eliminarlo si así lo desea o reabastecerlo de medicamentos, modificar cada cuanto tiempo debe ingerir el medicamento, los días que debe durar el tratamiento o incluso volverlo un tratamiento permanente.

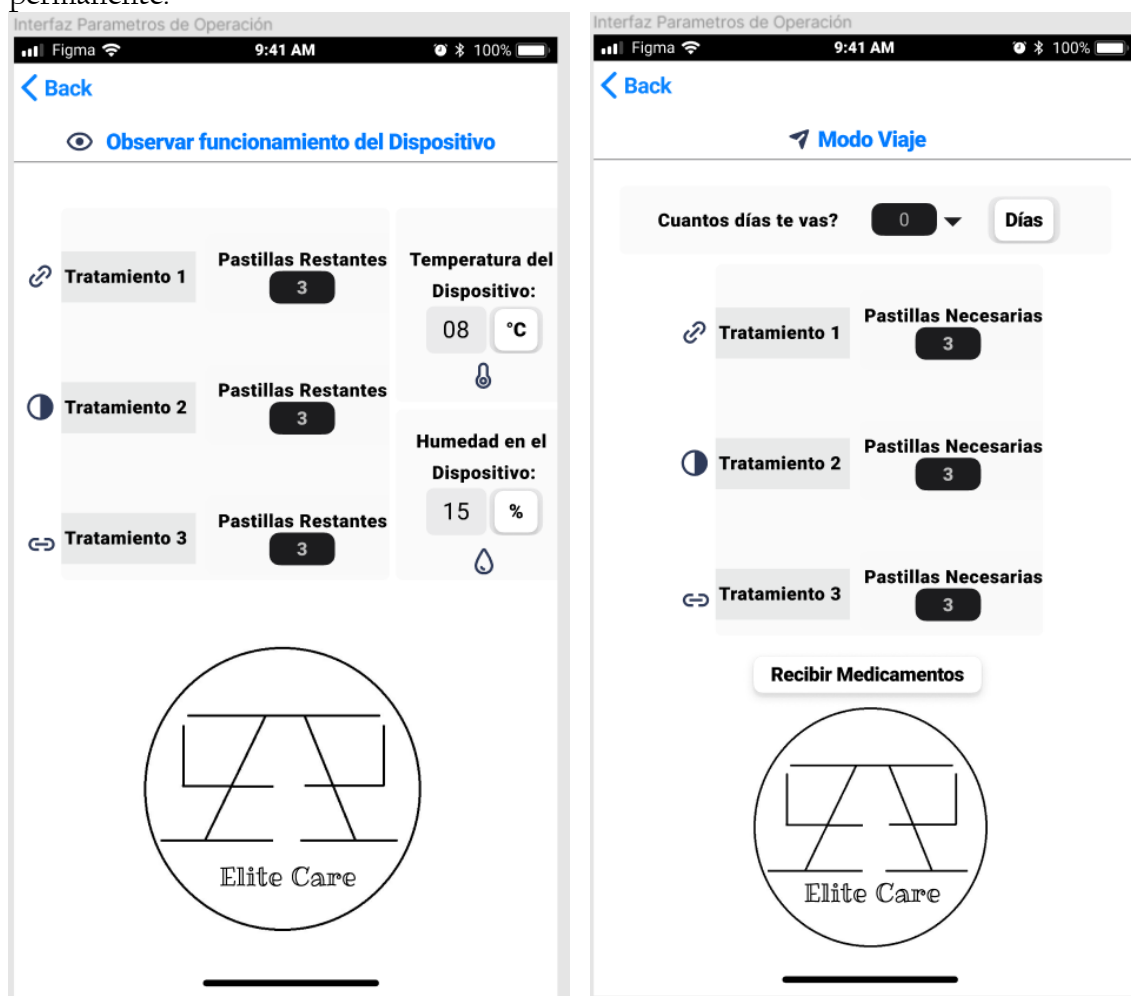


Imagen 164. Interfaces de observar funcionamiento del dispositivo y de modo viaje.

De la imagen 164, en la imagen de la izquierda, se observa la sub interfaz para la función “Observar Funcionamiento del Dispositivo”. Esta interfaz es meramente de monitoreo, permitirá al usuario revisar la temperatura y el porcentaje de humedad relativa interna del aparato, así como el número de pastillas restantes de sus tratamientos activos.

Finalmente, en la imagen de la derecha, se muestra la sub interfaz para la función de “Modo Viaje”, la cual, está desarrollada con la intención de que, si el usuario va a

ausentarse de su hogar por un periodo de tiempo que le impida tomar sus tratamientos en casa, pueda llevárselos en el pastillero portátil que incluye el dispositivo. La interfaz permitirá al usuario indicar cuantos días va a ausentarse y observar cuantas pastillas deberá recibir del dispensador y con un botón confirmar que desea recibir los medicamentos para llevar.

En todas las pantallas existe un botón de retroceso, el cual permite salir de las sub interfaces de las funciones del dispensador y regresar a la interfaz principal, descartando cualquier modificación que no se haya guardado o confirmado.

- Interfaz en el Dispensador

Aunque no forma parte de los objetivos específicos de TTI, se considera como una posibilidad el agregar al dispensador una interfaz software-hardware para que el usuario pueda observar los parámetros de funcionamiento del dispensador, así como el estatus actual de sus tratamientos aun cuando no cuente con su teléfono móvil. Su fin sería principalmente de monitoreo del dispositivo.

Para poder presentar una interfaz gráfica, se requiere implementar una pantalla que pueda conectarse a la unidad central de procesamiento y a través de dicha conexión pueda mostrar la GUI (Interfaz Gráfica de Usuario por sus siglas en inglés).

De acuerdo a los tipos de conexión con los que cuenta la unidad central de procesamiento, que son conexiones HDMI-DSI y HDMI-SPI, se dan como propuesta las siguientes pantallas, que se aprecian en las imágenes 165, 166 y 167:

Pantalla touch Waveshare:



Imagen 165. Pantalla de 5" con conexión HDMI-DSI Waveshare [57].

Pantalla touch Raspberry:



Imagen 166. Pantalla de 3.5' con conexión HDMI-SPI Raspberry [58].

Pantalla touch Head Sun:



Imagen 167. Pantalla de 7' con conexión HDMI-DSI Waveshare [59].

Las características de las pantallas propuestas como posible solución se muestran en la siguiente tabla 27.

	Pantalla touch Waveshare	Pantalla touch Raspberry	Pantalla touch Head Sun
Precio	\$995.65	\$636.14	\$1,552
Tamaño	5'	3.5'	7'
Resolución	800*480 pixeles	320*480 pixeles	1600*600 pixeles
Conexión	HDMI-DSI	HDMI-SPI	HDMI-DSI

Tabla 27. Características de interés de las pantallas propuestas.

Para poder evaluar y seleccionar una de las alternativas propuestas, una vez más se recurre al método de criterios ponderados, como resultado se obtuvo la siguiente tabla 28, donde se hizo la evaluación de los criterios ponderados a cada pantalla.

Criterio	Peso	Descripción	Unidad	Opciones:					
				Pantalla touch Waveshare		Pantalla touch Rapberry		Pantalla touch Head Sun	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Precio	0.2	Costo de la pantalla	\$MXN	8	1.6	9	1.8	3	0.6
Tamaño	0.3	Tamaño de la pantalla	pulgada	8	2.4	5	1.5	10	3
Resolución	0.3	Resolución de la pantalla	píxeles	9	2.7	6	1.8	10	3
Conexión	0.2	Tipo de conexión de la pantalla	HDMI-DSI ó HDMI-SPI	9	1.8	10	2	9	1.8
Total:	1			34	8.5	30	7.1	32	8.4

Tabla 28. Evaluación de las pantallas propuestas mediante criterios ponderados de acuerdo a las necesidades del proyecto.

Como es posible observar en la tabla 28, la alternativa mejor evaluada resulta ser la pantalla touch Waveshare de cinco pulgadas. Resulta ser una gran alternativa debido a sus características y costo. Además, ya que cuenta con conexión HDMI-DSI no requiere de ningún cableado especial o alimentación separada para ser conectada a la unidad de procesamiento y poder mostrar imagen, basta con acoplar los pines GPIO de la tarjeta a las entradas de la pantalla y conectar los puertos HDMI de ambas para que pueda operar.

Al ya contar con una pantalla definida como interfaz de hardware para el dispositivo, el equipo procedió a diseñar su interfaz gráfica de usuario acorde al tipo de software que se empleará en la programación del dispensador, la cual estará basada en Python. Además, se diseña con un tamaño de 800*480 píxeles el cual corresponde a la resolución de la pantalla seleccionada como hardware de interacción con usuario, dicha interfaz se visualiza en la imagen 168.

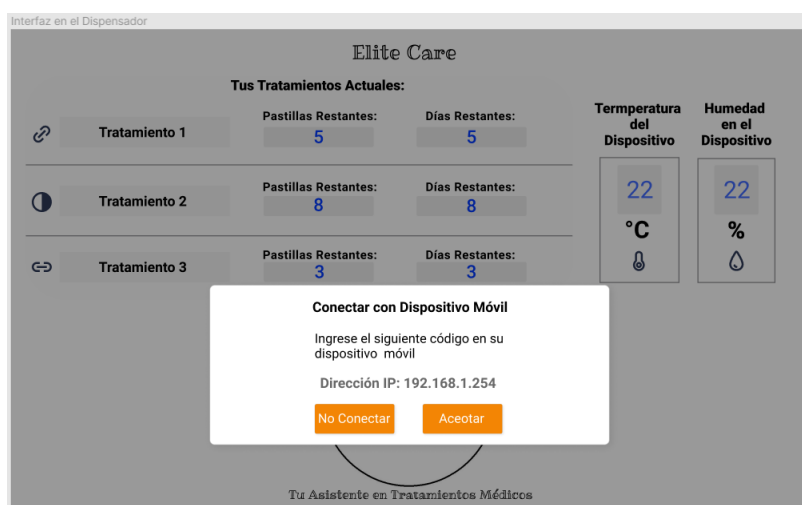


Imagen 168. Interfaz gráfica de usuario del dispensador al encender el dispensador.

La imagen 168 muestra la pantalla de inicio del dispensador, en esta se presenta una ventana que aparecerá al encenderlo o cuando se desconecte del teléfono móvil del usuario para poder vincularlo nuevamente, aunque también permitirá que, en caso de no querer vincular el dispensador con ningún teléfono exista la posibilidad de indicarlo.

Por otra parte, en la imagen 169, se muestra la interfaz principal, ésta, como se mencionó anteriormente, tendrá un fin meramente informativo para el usuario, pues únicamente permitirá al usuario observar los parámetros de funcionamiento de su dispensador, es decir, la temperatura y porcentaje de humedad internos, así como la situación actual de sus tratamientos activos o, dicho de otro modo, cuantos días y pastillas le restan de cada tratamiento activo.

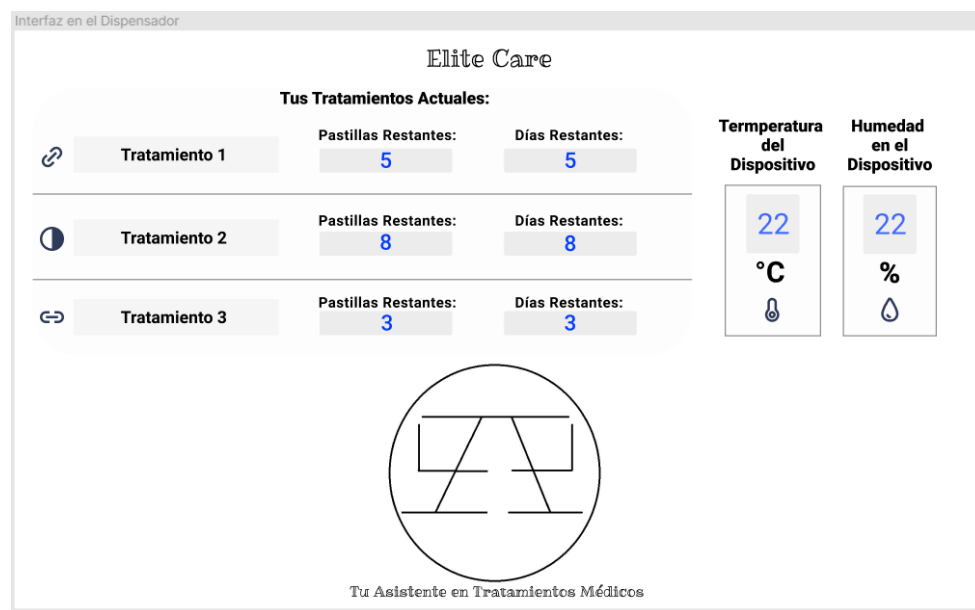


Imagen 169. Interfaz de usuario en el dispensador en operación.

Actualización de diseño de las interfaces de usuario

- *Diseño de la ventana de identificación de medicamentos*

Como parte de las actualizaciones, se ha agregado una nueva ventana a la aplicación móvil, la cual aparecerá cuando el usuario desee agregar un nuevo tratamiento al dispensador y, cuya función será solicitar al usuario que indique que forma tiene el medicamento del nuevo tratamiento para que este sea almacenado apropiadamente en el dispensador. Una vez el usuario indique la forma del medicamento, será enviado a la ventana de configuración de tratamiento, donde podrá indicar el resto de las características del mismo, como su duración, periodo de ingesta, nombre, etcétera.

El equipo busca que la ventana sea intuitiva y de fácil interacción para un manejo accesible por parte del usuario.

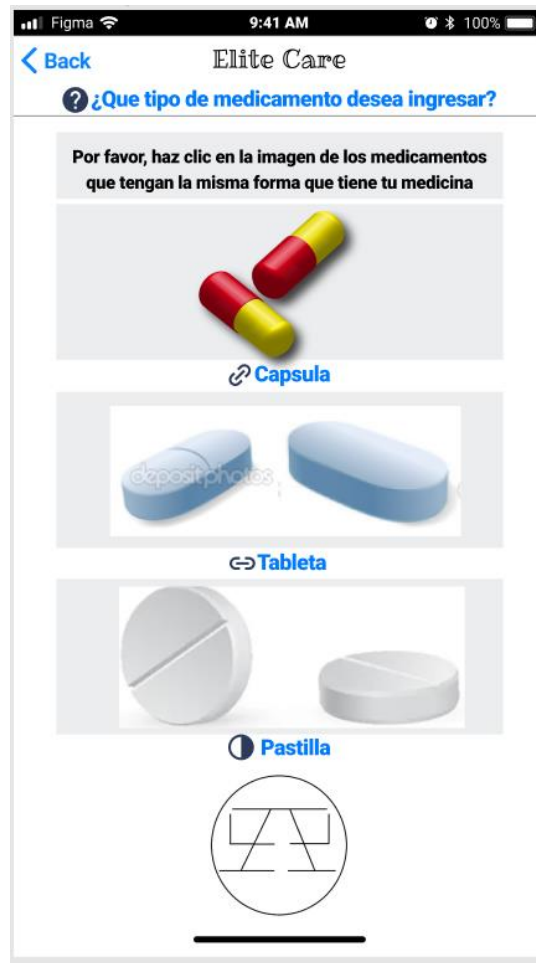


Imagen 170. Ventana de selección de forma del medicamento en la aplicación móvil.

En la imagen 170, se puede observar cómo se ha creado una ventana cuyos botones selectores son imágenes que ilustran de forma clara las formas admitidas por el dispensador y, además, agregan el nombre de dichas formas para evidenciar aún más cuál es cada una. Adicionalmente, se muestra un texto con la instrucción al usuario de tocar la imagen correspondiente a la forma del medicamento a ingresar.

Cabe resaltarse que los colores, y formas de las ventanas finales en la aplicación pueden llegar a variar respecto al diseño propuesto, sin embargo, si seguirán el mismo concepto y funcionamiento.

Sistema de energía

Para poder diseñar el sistema de energía se necesita conocer la potencia que el sistema electrónico demanda en el caso más extremo, esto para poder tener un factor de seguridad y asegurar el correcto funcionamiento de este. Para ello se consideraron las potencias eléctricas de cada sensor y/o actuador previamente seleccionado, esto a su vez dividido por el proceso que llevan a cabo y se estimó la máxima potencia que demanda el dispensador.



Potencias eléctricas

Solenoides 5V Push Pull: 1.1 A, 5.5 W

Electroválvula 2/2 Vías N.C. Acción Directa: 12V DC, 8W

Procesos:

Ingresar Medicamentos y agua

- Ingreso de agua (proceso manual)
- Ingreso de pastillas
- Reconocimiento y separación de pastillas
 - Activar sistema de reconocimiento de pastillas
(Activo hasta ingresar carga de medicamentos configurada)

Organizar Medicamentos

- Ingresar pastillas en contenedores
- Conteo de pastillas disponibles

Dispensar medicamentos y agua:

- Ubicar contenedor de medicamento requerido por el usuario
- Colocar contenedor en área de dispensado
 - Activar Push Pull para retraer motor de tornillo sinfín
 - Activar motor a pasos y mover revolver a posición requerida
- Extraer medicamento
 - Activar actuador para poner en contacto el motor a pasos con el eje del tornillo sinfín
 - Activar motor a pasos para movimiento del tornillo sinfín del contenedor
 - Activar Push Pull de mecanismo de stock
- Dispensar agua
 - Activar electroválvula y sensor de nivel de agua capacitivo
 - Activos hasta llenado deseado del vaso de agua
- Activar alarma para ingerir medicamentos y agua.
 - Activar buzzer para alertar al usuario (activo por 13s)
 - Enviar notificación al teléfono del usuario para ingesta de medicamento

Activos siempre: Unidad de procesamiento de datos, sistema de alarma, control de temperatura, conexión de red y tráfico de datos, interfaz de usuario en el dispositivo.

Todos los actuadores quedan desactivados al terminar sus respectivas tareas, es decir, no tienen consumo de energía.

Actuadores y sensores coincidentes:



Debemos considerar que, realmente, los procesos suceden en escalera, por lo que los actuadores casi nunca coinciden y más bien se da un trabajo conjunto entre los siempre activos y los procesos intermitentes.

Dispositivos siempre activos:

- Tarjeta de control (250mW)
- Sensor de temperatura y humedad relativa (110 mW)
- Unidad central de procesamiento de datos (12.5W)
 - Pantalla de interfaz de usuario
 - Cámara
 - Tarjeta
- Ventiladores de sistema de enfriamiento (1.25W)
- Sensor de nivel de agua de tanque de almacenamiento (125 mW)

Potencia acumulada de dispositivos siempre activos: Watt

Dispositivos intermitentes:

Coincidentes Alarma:

- Buzzer (150mW)

Potencia total: 150mW.

Coincidentes Reconocimiento:

- Motor a pasos de revolver (19.2W)
- Solenoide Push Pull para extracción de motor a pasos de tornillo sin fin (12W)

Potencia total= (19.2 + 12) W = 31.2 W

Coincidentes Colocación de Contenedor para extracción de medicamento:

- Motor a pasos del revólver (19.2W)
- Solenoide Push Pull para extracción de motor a pasos de tornillo sin fin (12W)

Potencia total= (19.2 + 12) W = 31.2 W

Coincidentes Extracción de medicamento:

- Motor a pasos de tornillos sin fin (200mW)
- Solenoide Push Pull de mecanismo de stock (12W)

Potencia total=12.2W

Coincidentes Dispensado de agua:

- Electroválvula (8W)
- Sensor de nivel de agua capacitivo (120 mW)

Potencia total = (8+0.12) W=8.12W

Las potencias de los elementos listados fueron determinadas consultando las hojas de datos de los fabricantes. En los casos en los que el consumo de potencia no figura como dato en dichos documentos, ésta se obtiene mediante ley de Ohm:

$$P = I * V$$

Donde P: Potencia consumida por el componente en Watt; I= Corriente máxima consumida por el componente; V=tensión de alimentación del componente.

Ahora, para determinar la potencia de consumo máxima de los actuadores y sensores, debemos sumar la potencia del grupo de coincidentes de mayor consumo con la potencia de los dispositivos siempre activos, esto garantiza que podamos seleccionar una fuente cuya potencia nominal de trabajo sea superior a la requerida y así podamos mantener una correcta operación del dispositivo independientemente de los procesos activos.

Cómo es posible observar, el grupo de coincidentes de mayor consumo son los de “reconocimiento” y los de “sistemas de colocación de contenedor para extracción de medicamento” que tienen el mismo valor de consumo de potencia, por lo que sumamos la potencia consumida por los siempre activos siguiendo el teorema de superposición:

Potencia total = Potencia Siempre Activos + Potencia Coincidentes Reconocimiento

Potencia total = 250mW + 110mW + 12.5W + 1.25w + 0.125mW + 19.2W + 12W

Potencia total = **45.435 Watt = $P_{\min \text{ fuente}}$**

Acorde a estos cálculos, se realizará la propuesta y selección de una fuente de factor de forma ATX de potencia nominal superior a la potencia total calculada, de tal forma que su marco de operación sea superior al requerido, garantizando la correcta operación del dispositivo y sus subprocesos. Un aspecto importante para su selección es que cuente con la variedad de tensiones de salida requeridas por los actuadores y sensores seleccionados para este proyecto, las cuales son 3.3 V, 5 V, 12 V, y -12 V ya que al conectar las salidas de -12 V y 12 V se puede lograr obtener una fuente de 24 V.

A partir de esta justificación, se procede a mostrar las fuentes que superan la potencia mínima necesaria y cumplen con las características de salidas de tensión y corriente requeridas. Para esto, se eligieron tres fuentes de alimentación las cuales se pueden ver en las imágenes 171, 172 y 173, donde se mostrará su nombre comercial y su forma.



Imagen 171. Fuente de poder CoolerMAster MWE 450 [60].



Imagen 172. Fuente de poder Getttech FG500W BX [61].



Imagen 173. Fuente de poder Ateck integra 5R500 [62].

De las fuentes mencionadas, se toman en cuenta ciertas características técnicas de éstas, que serán comparadas entre sí, mediante los siguientes criterios de evaluación:

- Potencia nominal
- Variedad de salidas de tensión
- Corriente máxima de salida
- Valor monetario
- Dimensiones de la fuente

Con estos criterios se realizó una tabla comparativa para observar dichas características de cada fuente, esto se visualiza en la siguiente tabla 29.

	Cooler Master MWE 450	Getttech FG500WBX	ACTECK INTEGRA 5R500
Potencia	450 W	480 W	500 W
Fuentes de salida	3.3 V, 5 V, 12 V, -12 V	3.3 V, 5 V, 12 V, -12 V	3.3 V, 5 V, 12 V, -12 V
Corrientes de salida	15 A, 15 A, 37A, 0.3 A	18 A, 18 A, 40 A, 0.3A	12 A, 30 A, 25 A, 0.8 A
Precio	\$999.00	\$269.00	\$428.99
Tamaño	140*150*86 mm	140*150*86 mm	140*150*85 mm

Tabla 29. Características de interés de las fuentes de alimentación.

Observando las especificaciones de interés de cada fuente de alimentación, se realizó la siguiente tabla 30, en la cual se evaluaron los criterios de selección para elegir a la fuente que mejor cumpla con cada criterio establecido.

Criterio	Peso	Descripción	Unidad	Opciones:					
				Cooler Master MWE 450		Getttech FG500WBX		ACTECK INTEGRA 5R500	
				Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado	Valor del rubro	Valor ponderado
Potencia	0.25	Potencia nominal de la fuente	W	8	2	9	2.25	10	2.5
Fuentes de salida	0.2	Variedad de salidas de tensión	V	10	2	10	2	10	2
Corriente	0.25	Corriente máxima de salida	A	7	1.75	8	2	9	2.25
Precio	0.15	Valor monetario	MXN	6	0.9	10	1.5	8	1.2
Tamaño	0.15	Dimensiones de la fuente	mm	9	1.35	9	1.35	10	1.5
Total:	1			40	8	46	9.1	47	9.45

Tabla 30. Evaluación de las fuentes de alimentación, mediante criterios ponderados.

Como resultado de la evaluación en la tabla mostrada, se elige la fuente llamada ACTECK INTEGRA 5R500, la cual tuvo mayor ponderación dado que cumple satisfactoriamente con los criterios de interés para dar la energía necesaria para hacer funcionar el dispensador.

Es preciso aclarar que el equipo es consciente de que Lafuente seleccionado supera por 10 veces la potencia requerida, sin embargo, la fuente no se selecciona con ese criterio específicamente. Tal y como se muestra en la tabla 30, existe una evaluación de criterios ponderados para su selección, además, la forma de la fuente ATX se elige para evitar el implementar más de una fuente o tener que agregar la complejidad de diseñar circuitos adicionales como divisores de tensión o corriente al proyecto.



4. Manufactura

Material de construcción

El primer obstáculo hacia la implementación es, ciertamente, el material de construcción de las piezas. Esto se debe principalmente a que el equipo propuso realizar las piezas en filamento de PVC con la finalidad de apegarse a la normatividad oficial sobre almacenamiento y conservación de medicamentos, desafortunadamente, la empresa 3D PRYNX, el único proveedor en México del filamento buscado (Vinyl 303), se encuentra sin stock. Dicho lo anterior, el equipo realizó un segundo esfuerzo por conseguir un segundo material de impresión que fuese aceptado por la norma oficial, el material de remplazo en cuestión fue el Polipropileno.

Con el nuevo material, se repitieron los estudios estructurales cuyos resultados indicaron que el cambio de material no suponía ningún riesgo de falla por esfuerzos o deformaciones mecánicas, por lo cual, se procedió a buscar una empresa que realizara la impresión 3D en este material.

La impresión 3D en materiales especializados implica el uso de máquinas sofisticadas, por lo que hallar quien imprima con un filamento poco usual resulta una labor complicada. La única empresa encontrada por el equipo que cuenta con el servicio de impresión en filamento de polipropileno es Grupo XDS, con quienes se realizó la cotización de todas las piezas que tienen un contacto directo con los medicamentos, como se muestra a continuación, en la imagen 174.



Valida por 5 Días Habiles

Fecha 14/09/2021

ID Cotizacion CT-01272

Grupo Soluciones Variables de Diseño, S.A.S de C.V.

RFC: GSV1709062V4
Calle B 26, Potrero del Llano
Azcapotzalco, CDMX, C.P. 02650
Tel.: 55 62748755
E-mail: servicios@grupoxds.com
WA: 55 17977606
www.grupoxds.com

Tu contacto para esta cotización es:

Manufactura Aditiva
servicios@grupoxds.com

Informacion de Contacto:

Rodolfo Mondragon
null
0
jmondragonz1501@alumno.ipn.mx

Enviar a:

Envío incluido por mensajería(2-5 días Habiles)
null
0
0
0
null
Mexico
0

TERMINOS Y CONDICIONES DE PAGO:

Agradecemos tu preferencia. ¡Es un placer trabajar en tu proyecto!
Para realizar el pago de esta orden con transferencia o depósito bancario, utiliza los siguientes datos, recuerda que debes cubrir el 100% del monto correspondiente a tu Cotización CT-01272 para iniciar la fabricación:

A nombre de Grupo Soluciones Variables de Diseño S.A.S de C.V

Cinbanamex	BBVA Bancomer
SUCURSAL: 7012	SUCURSAL: 0058
CUENTA: 5499188	CUENTA: 0116006795
CLABE: 002180701254991881	CLABE: 012180001160067955

Como referencia coloca tu nombre o el número de tu Cotización (CT-01272).

Si deseas realizar tu pago con tarjeta de crédito o débito, da click [aquí](#) y sigue las instrucciones
Una vez realizado el pago envía tu comprobante de pago a facturacion@grupoxds.com, en caso de requerir factura adjunta tus datos fiscales para emitirla, cuentas con 3 días para revisar y solicitar cualquier cambio en la factura. NO SE FACTURARÁN ordenes de meses anteriores.
Tu pedido será procesado una vez acreditado el pago y recibirás actualizaciones de estado vía correo electrónico.

TOTAL DE ITEMS COTIZADOS: 19

Precio Neto:	9933.07mxn
Envío incluido por mensajería(2-5 días Habiles):	0.00mxn
Descuento:	0.00mxn
IVA:	1589.29mxn
Total :	11522.36 mxn

© Grupo XDS 2009-2020 - Todos los derechos reservados

Imagen 174. Cotización de la impresión de las piezas con el GRUPO XDS.

Grupo XDS también sugirió la posibilidad de imprimir en un material más novedoso, una resina cuyas propiedades fisicoquímicas emulan las del polipropileno y, además, permite obtener una mejor calidad estética en las piezas.

La cotización en ese material se muestra a continuación, en la imagen 175:



Valida por 5 Días Hábiles

Fecha 14/09/2021

ID Cotización CT-01272

Grupo Soluciones Variables de Diseño, S.A.S de C.V.

RFC: GSV1709062V4
Calle B 26, Potrero del Llano
Azcapotzalco, CDMX, C.P. 02650
Tel.: 55 62748755
E-mail: servicios@grupoxds.com
WA: 55 17977606
www.grupoxds.com

Tu contacto para esta cotización es:

Manufatura Aditiva
servicios@grupoxds.com

Información de Contacto:

Rodolfo Mondragon
null
0
jmondragonz1501@alumno.ipn.mx

Enviar a:

Envío incluido por mensajería(2-5 días Hábiles)
null
0
0
0
null
Mexico
0

TERMINOS Y CONDICIONES DE PAGO:

Agradecemos tu preferencia. ¡Es un placer trabajar en tu proyecto!
Para realizar el pago de esta orden con transferencia o depósito bancario, utiliza los siguientes datos, recuerda que debes cubrir el 100% del monto correspondiente a tu Cotización CT-01272 para iniciar la fabricación:

A nombre de Grupo Soluciones Variables de Diseño S.A.S de C.V

Ciudadanex	BIVA Bancomer
SUCURSAL: 7012	SUCURSAL: 0058
CUENTA: 5499188	CUENTA: 0116006795
CLABE: 002180701254991881	CLABE: 01218000116006795

Como referencia coloca tu nombre o el número de tu Cotización (CT-01272).

Si deseas realizar tu pago con tarjeta de crédito o débito, da click [aquí](#) y sigue las instrucciones
Una vez realizado el pago envía tu comprobante de pago a facturacion@grupoxds.com, en caso de requerir factura adjunta tus datos fiscales para emitirla, cuentas con 3 días para revisar y solicitar cualquier cambio en la factura. **NO SE FACTURARÁN** ordenes de meses anteriores.
Tu pedido será procesado una vez acreditado el pago y recibirás actualizaciones de estado vía correo electrónico.

TOTAL DE ITEMS COTIZADOS: 19

Precio Neto:	25049.85mxn
Envío incluido por mensajería(2-5 días Hábiles):	0.00mxn
Descuento:	0.00mxn
IVA:	4007.98mxn
Total :	29057.83 mxn

© Grupo XDS 2009-2020 - Todos los derechos reservados

Imagen 175. Cotización de la impresión de las piezas en resina similar al polipropileno, con el GRUPO XDS.

Desafortunadamente, ambos precios resultan extremadamente elevados, el costo de las impresiones en estos materiales supera por mucho el presupuesto del equipo y se debe tener en consideración que las piezas cotizadas no representan el total de piezas requeridas, es decir, aun se tendrían piezas pendientes de imprimir así como el resto de componentes del dispositivo, razón por la cual el equipo se ve obligado a optar por un material de impresión no especializado, esto, con la finalidad de reducir el costo del proyecto a valores que sean aptos a nuestra situación económica.

Dicho lo anterior, el equipo ha decidido implementar el filamento polimérico de PLA, el cual es un material con buenas propiedades mecánicas para la aplicación en cuestión como dureza, resiliencia y resistencia, lo cual, se puede corroborar en los estudios estructurales de las piezas, mismos que se han repetido con este material comprobando que no existe ningún compromiso en las mismas debido a esfuerzos o deformaciones.



La cotización que se muestra a continuación (tabla 31), donde la impresión ha sido realizada por un independiente.

Nombre	Cantidad	Densidad	Peso (g)	Costo (MXN)
Compuerta1_Ultimate	1	50%	1	\$1.50
Compuerta2_Ultimate	1	50%	1	\$1.50
Compuerta3_Ultimate	2	50%	2	\$3.00
ContenedorAgua_Ultimate	1	50%	413	\$619.50
CubiertaDeTornilo1_Ultimate	1	50%	5	\$7.50
CubiertaDeTornilo2_Ultimate	1	50%	4	\$6.00
CubiertaDeTornilo3_Ultimate	2	50%	3	\$4.50
EntradaEmbudo_Ultimate	1	50%	40	\$60.00
Reductor1_Ultimate	1	80%	5	\$7.50
RevolverUltimateC	1	80%	116	\$174.00
Tapa1_UltimateC	1	80%	64	\$96.00
Tapa2_UltimateC	1	80%	51	\$76.50
Tapa3_UltimateC	2	80%	94	\$141.00
Tornillo1_Ultimate	1	80%	85	\$127.50
Tornillo2_Ultimate	1	80%	75	\$112.50
Tornillo3_Ultimate	2	80%	140	\$210.00
Totales ->	20		1099	\$1,648.50

Tabla 31. Cotización de las piezas a imprimir con el material de PLA, por un independiente.

Como se puede observar, el costo de impresión resulta ser un 85.69% más barato que la impresión en polipropileno. La desventaja será que no se logrará el grado farmacéutico en este caso práctico.

El equipo está consciente de que, para lograr la mejor implementación posible, se debería emplear el filamento de PVC o polipropileno, sin embargo, debido a la escasez de ese tipo de filamentos y altos costos de impresión nos resulta imposible imprimir en dichos materiales, sobre todo considerando la situación económica actual debido al contexto de la pandemia por Covid-19, por lo que, para los efectos prácticos de este proyecto se empleará filamento de PLA.

Manufactura de las piezas del dispositivo

Como ya se ha hecho evidente en capítulos anteriores, el equipo decidió emplear la tecnología de impresión 3D como principal proceso de manufactura para las piezas diseñadas del dispositivo debido a que permite obtener replicas precisas de los modelos 3D realizados en CAD en un material polimérico, lo cual es su principal ventaja sobre otros procesos como el corte laser que solo permite obtener piezas planas; además, es más fácil de acceder que a otros procesos industriales como el vaciado, extruido o inyección de plásticos.

Una vez teniendo claro que, para evitar un muy alto gasto en la impresión de las piezas, se ha optado por cambiar a un material de impresión comercial como lo es el PLA, se procedió a contratar al servidor privado de la última cotización.

El equipo optó por imprimir primero piezas de prueba para tener un primer acercamiento con la calidad de impresión, así como para comprobar el manejo de tolerancias y características de ensamble, de esta forma, se obtuvo información útil para mejorar las características de las piezas críticas del dispositivo.

Pruebas de impresión 3D

Dado que el proceso de manufactura de impresión 3D es uno que resultaba desconocido para el equipo, sobre todo en los aspectos referentes a la precisión y calidad, se decidió realizar pruebas con algunas piezas y segmentos de piezas, con la finalidad de observar si era posible obtener las características de diseño planeadas. Las características más importantes a verificar en estas pruebas eran: la consistencia de las piezas, la calidad de impresión en las muescas y ranuras de ensamble y, a su vez, corroborar que las tolerancias empleadas en los mecanismos de unión fuesen adecuadas y permitiesen ensamblar los componentes.

Para llevar esto a cabo, se optó por tomar 2 piezas que cuentan con mecanismos de ensamble, las cuales son: el contenedor de pastillas redondas y la tapa de dicho contenedor. Por otra parte, para evitar un gasto mayor en este proceso, se obtuvieron versiones segmentadas del revólver, el tornillo euclidiano para pastillas redondas y su respectivo reductor de diámetro, pues las partes que se requieren probar son solamente las que ensamblan entre sí.

A continuación, en las imágenes 176, 177, 178, 179 y 180, se muestran las piezas en CAD empleadas para realizar las pruebas, así como su resultado de impresión.

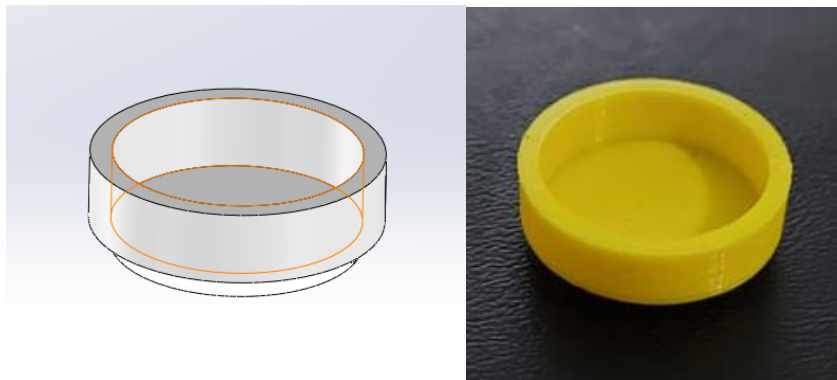


Imagen 176. Segmento de reductor en CAD e impreso.

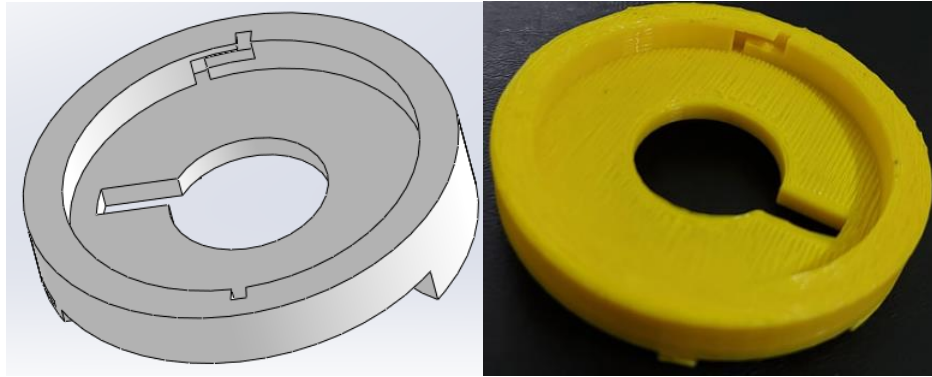


Imagen 177. Segmento del revolver en CAD e impreso.

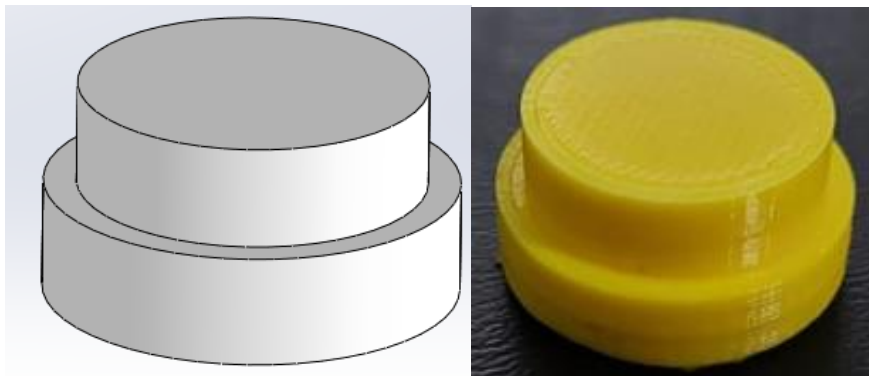


Imagen 178. Segmento de ensamble de tornillo sinfín en CAD e impreso.

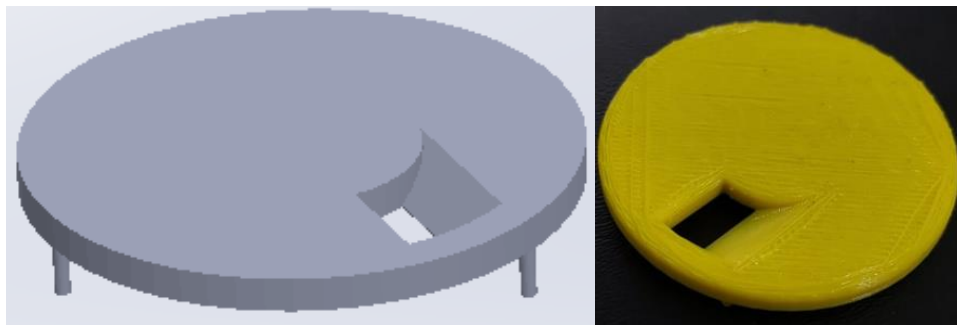


Imagen 179. Tapa de contenedor en CAD e impreso.

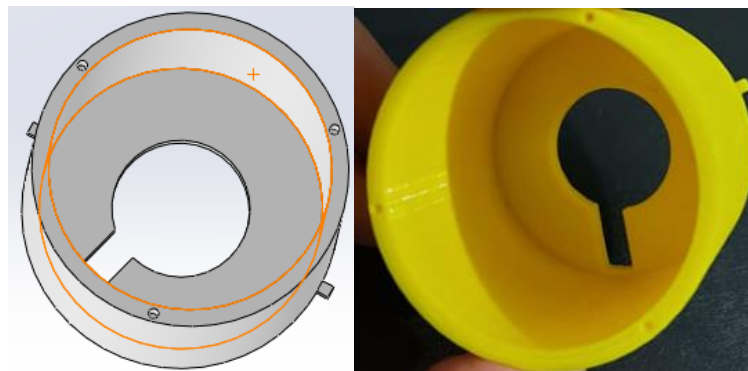


Imagen 180. Cubierta del contenedor en CAD e impreso.

De estas pruebas se logró concluir que, de acuerdo a como lo sugieren las bibliografías, una tolerancia para que exista deslizamiento en el ensamble debe ser de al menos 0.15mm, para una pieza que tendrá rotación, la tolerancia debe ser de al menos 0.25 mm, y para un ensamble fijo la tolerancia debe ser de 0.1 mm que sería equivalente a un ajuste fijo.

Otra cuestión importante fue que la precisión de impresión es de 0.2mm por lo que no es posible imprimir con la calidad suficiente características delgadas y/o pequeñas inferiores a los 2 milímetros en las piezas, pues estas pierden calidad y resultan ser muy frágiles, como es el caso de los ejes de las tapas de los contenedores, por lo que se realizarán modificaciones a dichas características de ensamble en esas piezas así como a los contenedores y el revólver, esto, con la finalidad de lograr que sean más resistentes y efectivas.

Proceso de impresión 3D de las piezas del dispositivo

Una vez realizados los ajustes pertinentes en los modelos CAD de acuerdo a los resultados obtenidos en las pruebas, se imprimieron todas aquellas piezas funcionales de los mecanismos de ingreso, almacenamiento y dispensado de medicamentos, pues de esta forma se pudieron realizar las pruebas necesarias para corroborar su funcionamiento individual y colectivo. Dejando al último las piezas que conforman la cubierta o carcasa del dispensador.

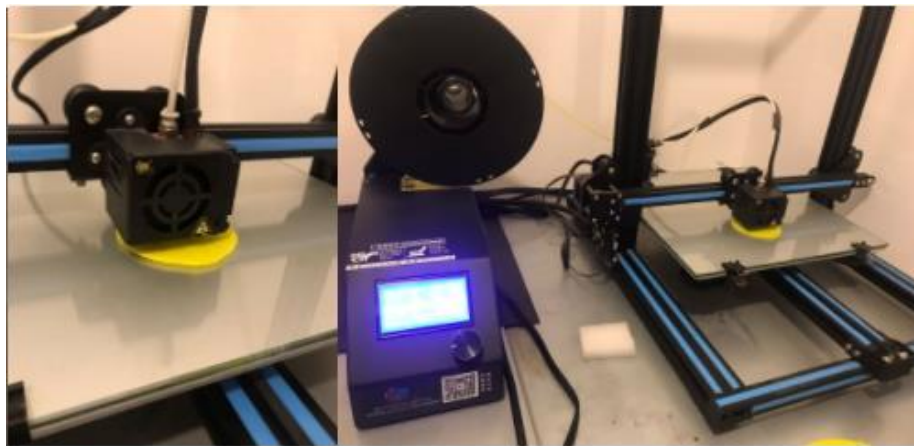


Imagen 181. Proceso de impresión en 3D de las piezas del dispositivo.

En la imagen 181, se puede ver parte del proceso de impresión, donde se observa como la máquina está fabricando una de las piezas sobre la cama de impresión, por otro lado, en la imagen 182, podemos ver el seguimiento del software de impresión sobre las piezas en proceso.

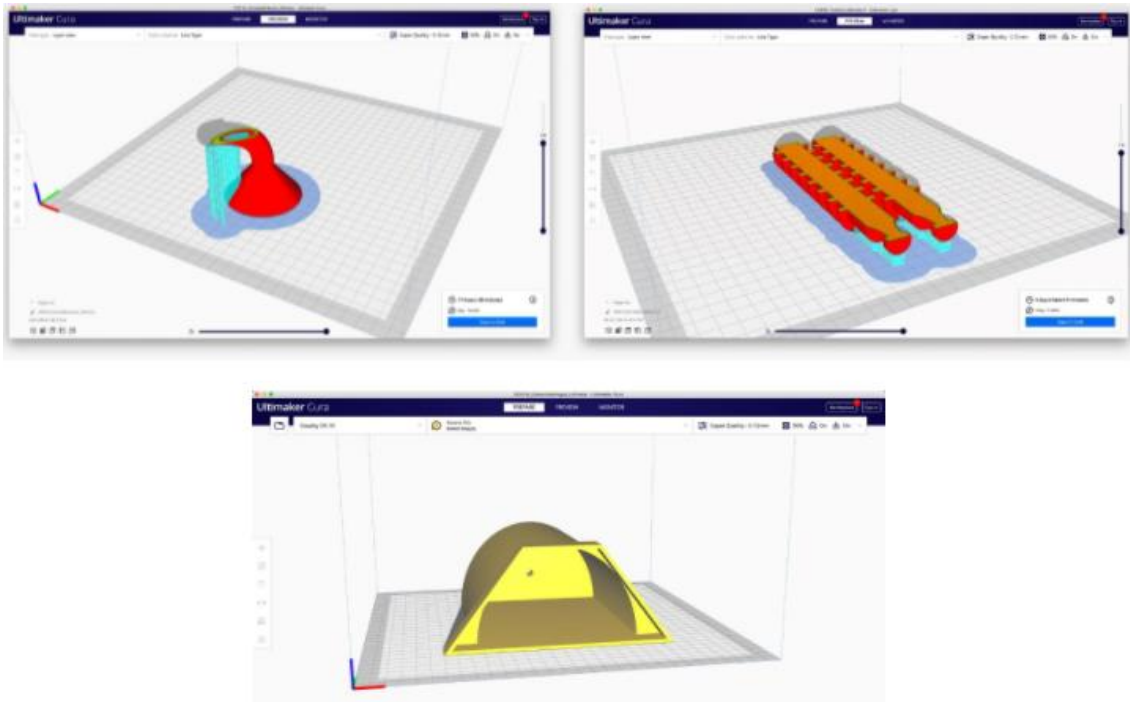


Imagen 182. Software para el proceso de impresión 3D de las piezas del dispositivo.

En la parte superior izquierda de la imagen 182 se puede notar como, debido a su forma irregular, el software de impresión marca en color azul los soportes que deberá construir para poder manufacturar la pieza, que se trata del cono de entrada de medicamentos.

El proceso de impresión fue llevado a cabo para todas las piezas diseñadas por el equipo, desafortunadamente, fue también el que más retrasos sufrió debido, primero, al cambio de material y, posteriormente, debido a que algunas piezas tienen geometrías complejas, asimétricas o huecas, haciendo que el tiempo de manufactura incrementa debido a la necesidad de la máquina de construir de soportes de apoyo durante la elaboración de las mismas.

Los soportes son estructuras impresas junto a la pieza que permiten sostener geometrías que se encuentran en voladizo o son huecas, ya que, al imprimir con hilos de polímero caliente, sin un soporte en dichas condiciones tendería a escurrirse o doblarse.

Piezas Impresas

Después de un considerable proceso de impresión, el equipo logró recabar las piezas terminadas las cuales fueron fabricadas acorde a los modelos realizados en la etapa de diseño detallado. Las piezas obtenidas se pueden observar de la imagen 183 a la 205.

En la imagen 183 se observa como las piezas independientes que componían la carcasa de la entrada que originalmente eran 4 se fusionaron en 1 sola carcasa, dejando solo como pieza de acople la rampa observada en la imagen 184.



Imagen 183. Carcasa entrada rediseñada, vista frontal.

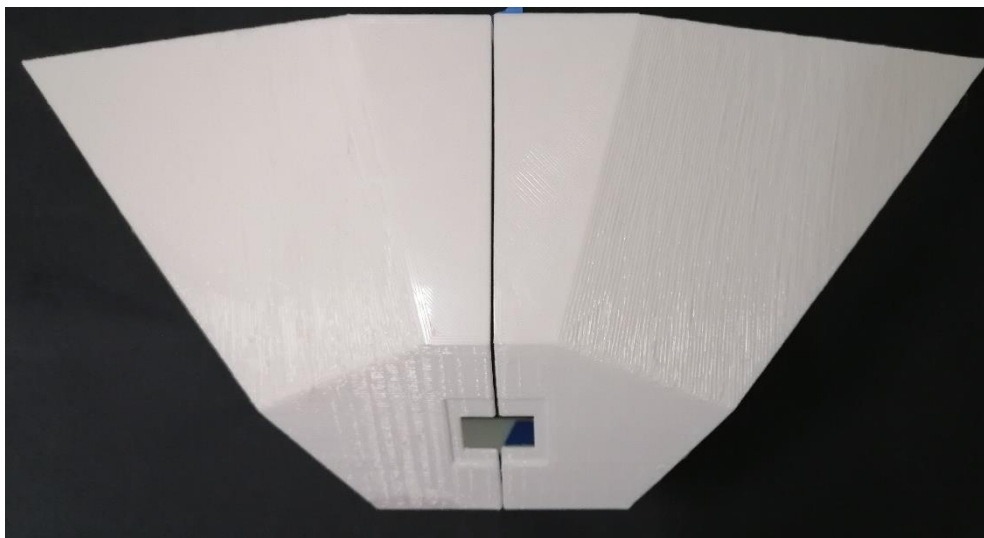


Imagen 184. Carcasa de entrada, vista superior.

En las imágenes 185 y 186, se puede observar la carcasa principal superior que forma parte del contenedor de agua donde se visualiza su forma en la vista superior y en la vista frontal se presenta la unión de 2 carcasas para armar la carcasa principal superior para resguardar el contenedor de agua.



Imagen 185. Carcasa principal superior, vista superior, se puede observar el compartimento del tanque de agua y en la parte superior las ranuras para el soporte de entrada.



Imagen 186. Carcasa principal superior, vista frontal, se aprecia la ranura de la tapa superior que recubre el tanque de agua.

Para la parte de la carcasa principal inferior se agregaron 2 plataformas para poder colocar componentes electrónicos, así como la actualización del sistema de salida del medicamento. Esto se observa mejor en la imagen 187.

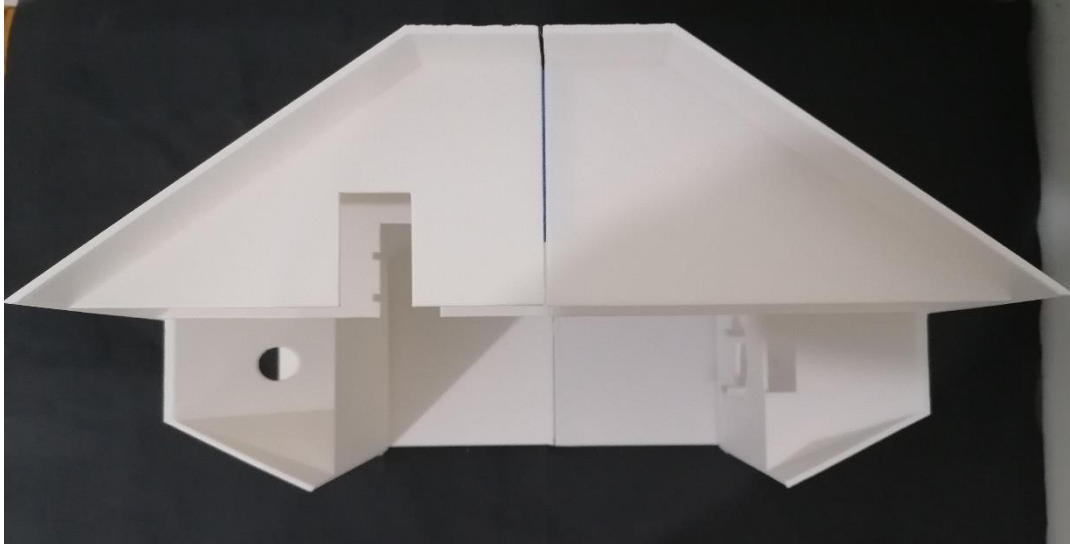


Imagen 187. Vista superior de la carcasa principal inferior.

Al cambiar los sensores de detección de agua se cambió también la forma en donde se acoplan estos a la carcasa principal, siendo uno el que se observa en la imagen 188 en la parte inferior izquierda, además del que estará colocado en la parte posterior del soporte para el tanque de agua. En esta imagen además se observa nuevamente la parte de las plataformas con el corte para la actualización del sistema de dispensado en el cual actúan los solenoides que empujan las compuertas.

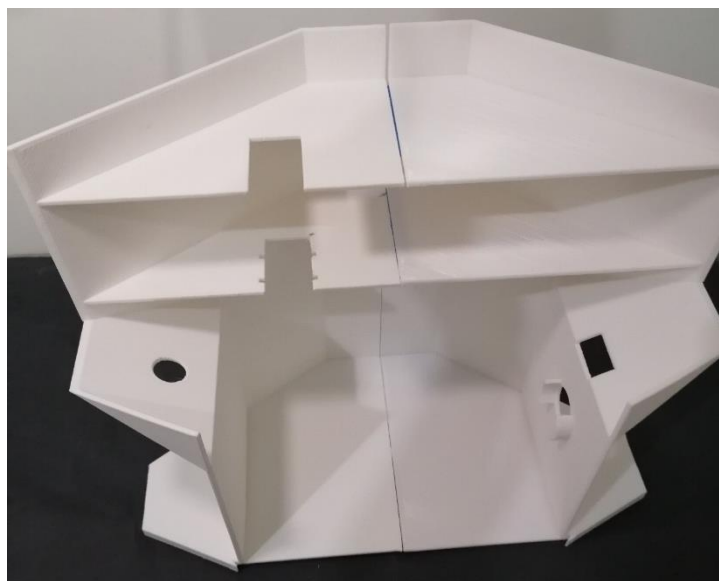


Imagen 188. Vista frontal de la carcasa principal inferior.

A continuación, se muestran las tapas de cada contenedor, ya con su respectivo chaflán para que la pastilla correspondiente pueda deslizarse con mayor facilidad y entrar sin problema a la hélice del tornillo sinfín. Estas se visualizan de la imagen 189 a 192.



Imagen 189. Tapa del compartimento 1.



Imagen 190. Tapa del compartimento 2.



Imagen 191. Tapa del compartimento 3.



Imagen 192. Tapa del compartimento 4.

Las piezas que se observan a continuación en la imagen 193 son las compuertas que permiten la salida de las pastillas de sus correspondientes contenedores, cabe mencionar que tienen un eje al cual se le coloca un resorte con el fin de poder cerrar el paso de la pastilla anterior a la que se dispensó.

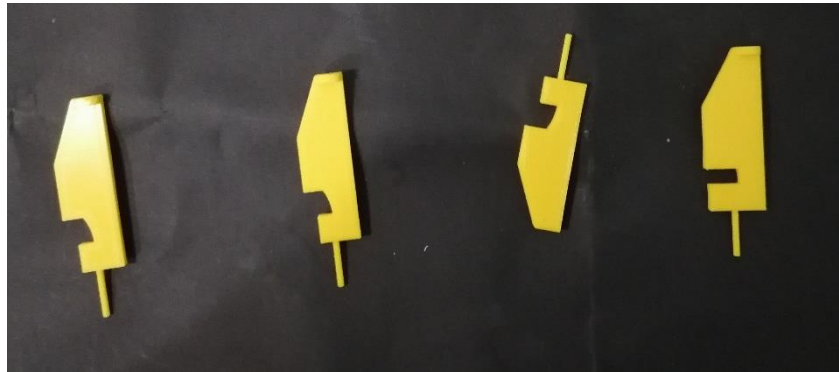


Imagen 193. Compuertas de los compartimentos.

Ahora se puede ver el revolver impreso en la imagen 194, en el que se posicionan los compartimentos de las pastillas donde se encuentran los orificios justos a la medida para colocarlos.



Imagen 194. Vista superior del revolver, se logra apreciar los espacios de ensamble.

Uno de los compartimentos se puede ver en la imagen 195, todos los que se imprimieron se ven relativamente iguales solo cambian en tamaño. La textura es grata para la mano y presentan la dureza suficiente para soportar la presión al colocarlos en el revolver, esto gracias a las ranuras que lleva.



Imagen 195. Vista lateral de uno de los compartimentos, en la parte inferior se aprecian las ranuras de anclaje al revolver.

El contenedor de agua impreso se observa en la imagen 196, para este se decidió imprimirlo en color café esto debido a la falta de filamentos. Este es capaz de soportar un litro de agua que será el que se dispensará a lo largo de los tratamientos que lleve el usuario.

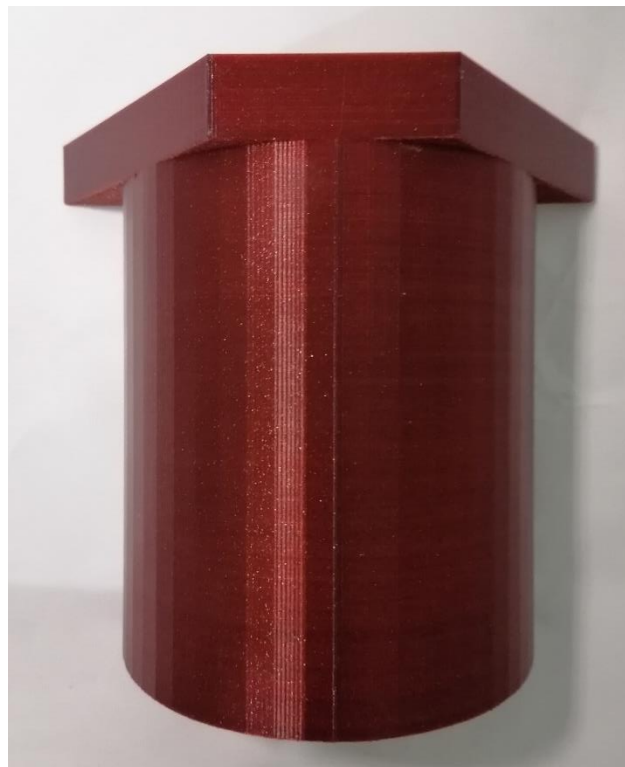


Imagen 196. Vista frontal del tanque de agua.

Para poder garantizar que entre una a una pastilla a cada contenedor, se imprimieron las siguientes piezas: embudo de entrada y su respectivo soporte, que se aprecian en las imágenes 197 y 198. El color fue por la misma razón que en el contenedor de agua. Los medicamentos se deslizan debidamente a través de este embudo y se puede fijar dado que tiene pestañas que se pueden anclar a su soporte. El soporte se fija a la parte interna de la carcasa.



Imagen 197. Vista frontal del embudo de entrada.



Imagen 198. Soporte del embudo de entrada.

Las piezas impresas que se muestran en la imagen 199 y 200, son las paredes de un mecanismo de ascensor y la pieza que une a dichas paredes, este mecanismo se utiliza para subir o bajar un solenoide que es el que ubica a otro solenoide justo para mover la compuerta de salida y así permitir la salida del medicamento.

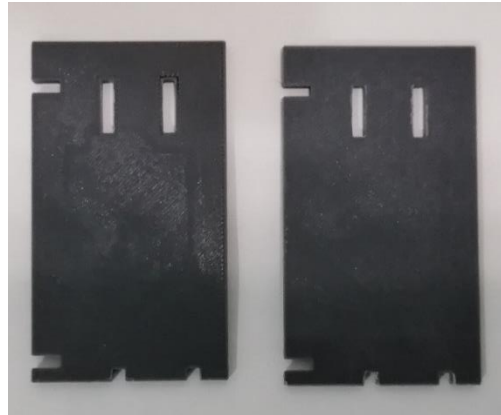


Imagen 199. Paredes del mecanismo del ascensor que pertenecen al sistema de salida.



Imagen 200. Pieza de unión de las paredes del mecanismo de ascensor.

A continuación, se muestran en las imágenes 201 y 202, las piezas impresas del embudo de salida y su correspondiente soporte. Este embudo es la parte final donde se desliza la pastilla hasta llegar al vaso donde el usuario podrá tomarla; el soporte está fijado a una pared interna de la carcasa y en este se posiciona el embudo de salida que se puede fijar ya que tiene muescas que se acoplan perfecto con las ranuras del soporte.



Imagen 201. Embudo de salida.



Imagen 202. Soporte del embudo de salida.

El soporte para el solenoide que mueve las compuertas de salida de medicamentos, se aprecia en la imagen 203, dicha pieza está sujeta entre las paredes del ascensor y hace el movimiento de subir y bajar, por lo que solo soporta ese solenoide.



Imagen 203. Soporte del solenoide de dispensado.

Para la pieza impresa visualizada en la imagen 204, se trata del soporte del motor a pasos que transmite movimiento a los tornillos sinfín que se encuentran dentro de los contenedores. Esta pieza va ensamblada a un solenoide que la jala junto con el motor, esto al momento que se mueve el revolver y así evitar que se muevan tornillos involuntariamente.



Imagen 204. Soporte del motor a pasos unipolar para la transmisión de movimiento a los tornillos.

En la imagen 205, se aprecia el resultado de la impresión de la rampa de entrada de pastillas, la cual se empotra en la carcasa superior de entrada que al momento de ponerla queda con una inclinación de 30° y con eso, los tres tipos de pastillas se puedan deslizar sin dificultad.



Imagen 205. Rampa de entrada de las pastillas.

Debido a que las impresiones 3D se realizan por capas de hilo de polímero, la superficie de las piezas no resulta ser liza, es decir, tiene una rugosidad la cual depende directamente del ancho del filamento y boquilla empleados en el proceso. En el caso de estas piezas se empleó una boquilla y filamento de 0.2mm que es la mejor calidad posible, sin embargo, no resulta posible eliminar la rugosidad pues esta es intrínseca del proceso.

5. Implementación

Etapas de potencia

Para poder activar los diferentes actuadores que previamente están dispuestos en el diseño del dispensador, se elaboró el siguiente circuito de potencia, en el cual se compone principalmente de un optoacoplador 4N25 que permite una conexión eléctricamente aislada entre dos circuitos que operan a distintas tensiones, como es el caso de la tarjeta de control y los actuadores. También consta de un transistor MOSFET IRFZ44N dado que soporta una corriente de drenaje de 49 A y una tensión de drenaje a fuente de 55V, esto es ideal para lograr manejar las cargas a usar. Este diagrama esquemático del circuito se puede apreciar en la siguiente imagen 206.

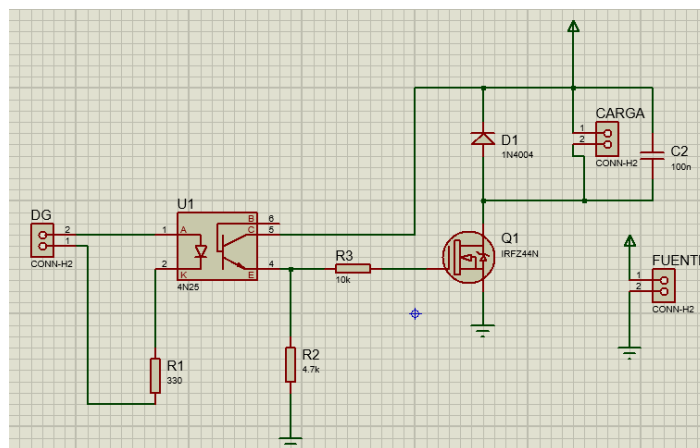


Imagen 206. Diagrama esquemático del circuito de potencia.

Se procedió a realizar el PCB del diagrama esquemático de la imagen 206, con lo que se buscó ahorrar espacio y acomodar los componentes para que las pistas fueran estéticas y pudiesen cumplir la conducción deseada. La placa tiene un ancho de 2.15 in y un alto de 1.2 in y se incluyeron las etiquetas de cada componente para identificar fácilmente los lugares en los que se soldarían. Las etiquetas que dicen “FUENTE”, “CARGA” y “DG” son borneras donde se colocarán los cables de alimentación de la fuente, los cables de la carga, ya sea los ventiladores o los solenoides, y los cables de alimentación tomados de la tarjeta de control, en este caso le corresponde la etiqueta “DG”. En la siguiente imagen 207, se presenta el PCB de la etapa de potencia para los ventiladores y los solenoides que se usarán.

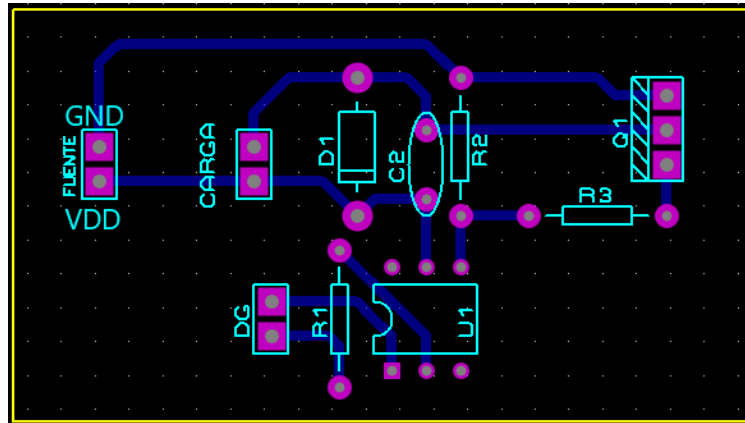


Imagen 207. PCB del circuito de potencia.

A continuación, se muestran en las imágenes 208 y 209, la vista en 3D del PCB ya con los componentes que se soldarían. De igual forma, en las imágenes se pueden ver las etiquetas mencionadas en la parte superior de la placa fenólica, mientras que en la parte inferior se observan las pistas de cobre con los pines que le corresponden a cada componente.

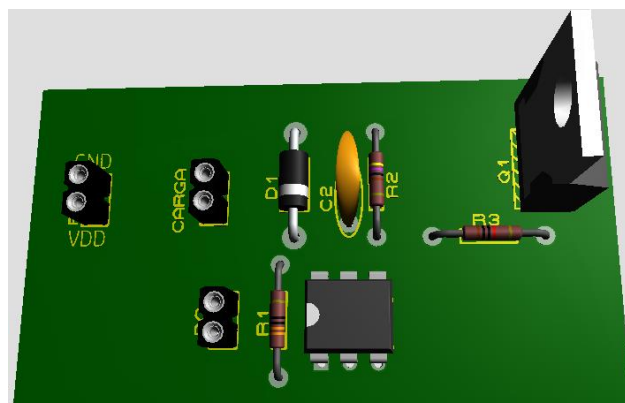


Imagen 208. Modelo en 3D del PCB del circuito de potencia, vista superior.

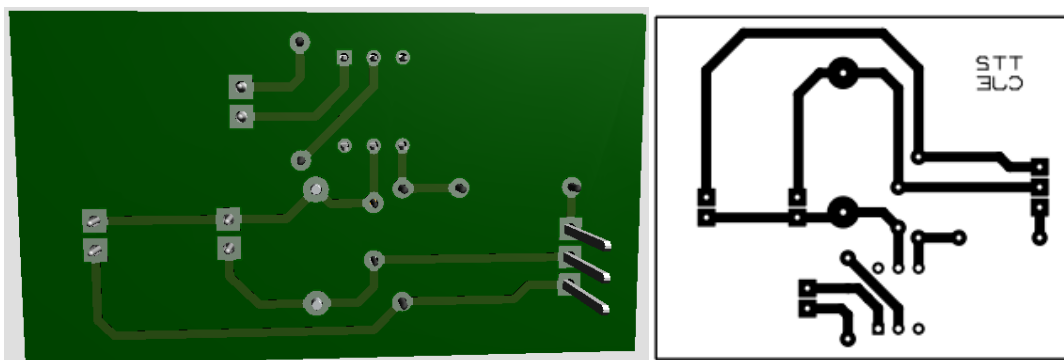


Imagen 209. Modelo en 3D del PCB del circuito de potencia y pistas para planchado, vista inferior.

Esta vista previa del circuito electrónico permite observar con precisión la disposición de las pistas y las etiquetas en las respectivas placas fenólicas, con la finalidad de determinar si son adecuadas para su impresión y planchado sobre las placas fenólicas para finalmente

después soldar los componentes correspondientes. El número de placas que se usarán serán 8.

El equipo fabricó las placas de PCB mediante el proceso de planchado sobre placas fenólicas debido a que es un proceso que no requiere equipo especial y puede ser llevado a cabo sin la necesidad de un laboratorio, como resultado se obtuvieron las placas que se muestran en la imagen 210. Posteriormente, sobre estas se soldaron los componentes electrónicos obteniendo como resultado la placa que se muestra en la imagen 211. Este procedimiento fue repetido 8 veces que equivale al número de actuadores que requieren una etapa de potencia en este dispositivo.

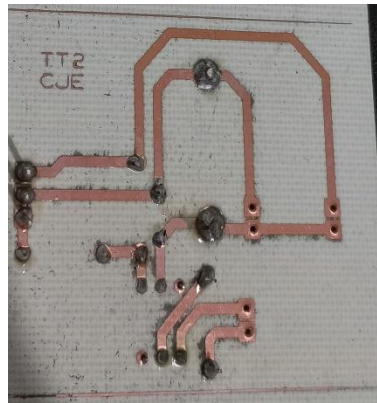


Imagen 210. Pistas del circuito de la etapa de potencia.

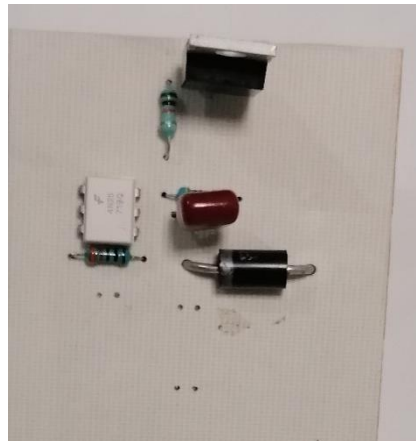


Imagen 211. Circuito de etapa de potencia para los actuadores.

Convertidores de nivel

Estos se utilizaron principalmente porque las salidas digitales de la Raspberry brindan una tensión de 3.3 V y para activar los actuadores en cuestión se necesitan 5V. Los convertidores de nivel hacen posible que una salida digital de 3.3 V, al pasar a éstos, puedan salir a 5V por sus pines y viceversa, dado que son convertidores bidireccionales.

La matrícula de estos es HW-221. Para mostrar su forma de conexión se presenta un diagrama representativo, donde se observan las conexiones necesarias con la Raspberry y los actuadores en forma general, esto se puede ver en la imagen 212.

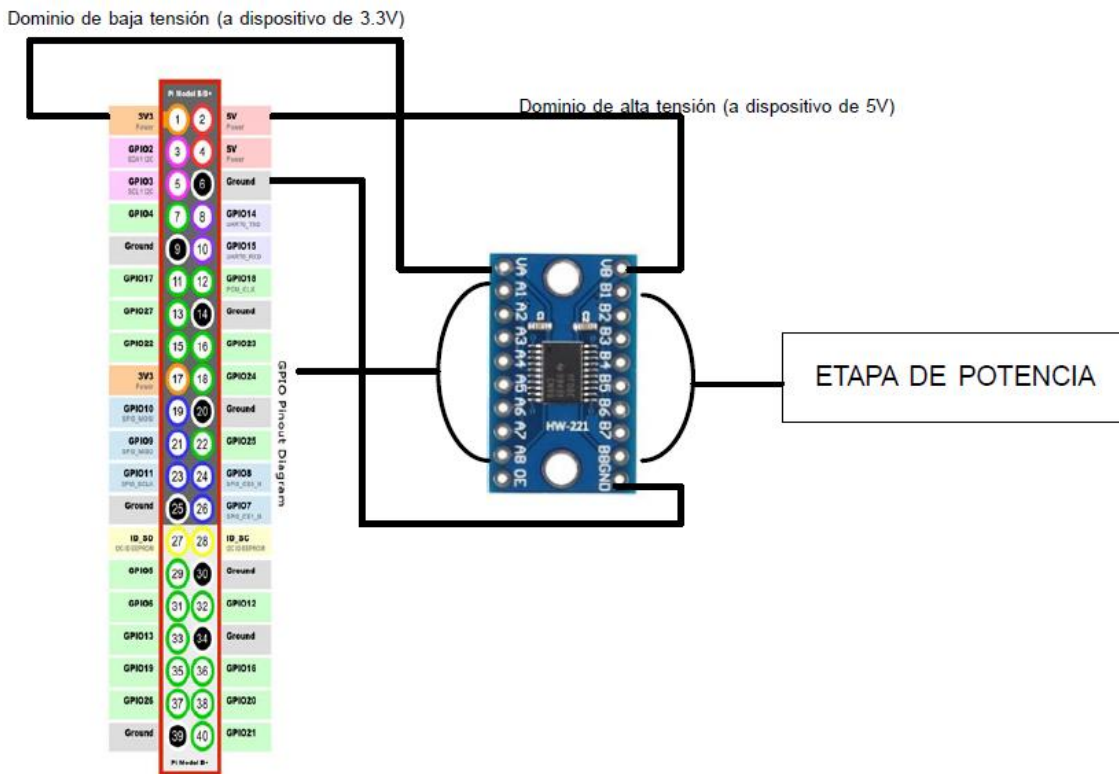


Imagen 212. Conexión de los pines GPIO de la Raspberry al convertidor de nivel, y este, a la etapa de potencia.

Como se puede apreciar en la imagen 212, la alimentación de 3.3 V de la Raspberry se conecta al pin VA del convertidor, dado que éste gobierna los pines de baja tensión que se indican con la letra A y su respectivo número. A estos se conectan los pines GPIO que se ocupan para el manejo de los actuadores. Ahora, lo que se conecta es la alimentación de 5V de la Raspberry al pin VB del convertidor, el cual es el que gobierna los pines de alta tensión, indicados con la letra B y su respectivo número. A estos se conectan los pines de entrada digital de cada etapa de potencia utilizada para los actuadores y funcionan a la tensión debida.

Aplicación móvil

Habiendo revisado el diseño propuesto de la aplicación móvil, así como sus respectivos objetivos específicos y alcances de proyecto, el equipo decidió desarrollarla comenzando con las interfaces o ventanas para las distintas actividades que el usuario podrá realizar con la app. Para llevar a cabo esta tarea, se decidió emplear la IDE de Android Studio bajo el lenguaje de programación Kotlin que es el lenguaje oficial de desarrollo de aplicaciones móviles para Android.

El equipo ha desarrollado en XML las ventanas con las que podrá interactuar el usuario, siguiendo las propuestas realizadas en TTI, sin embargo, debido a que se busca lograr que la interacción con el usuario sea fácil e intuitiva, se ha optado por agregar o modificar características sin que se pierda el concepto. Cabe recalcar que se ha premiado a la funcionalidad de la aplicación sobre la estética

- **Ventana principal**

Esta es la ventana de inicio de la aplicación, a partir de ella se puede acceder al resto de ventanas que permitirán al usuario gestionar el dispensador de medicamentos.

Como es posible observar en la imagen 213, se mantiene el concepto inicial, aunque se agrega un botón de salir para permitir al usuario abandonar la app de forma segura y correcta.



Imagen 213. Ventana principal.

- **Ventana de selección de tipo de medicamento**

Cuando el usuario selecciona en la ventana principal que desea ingresar un nuevo tratamiento, es llevado a esta ventana, en la cual, debe indicar que forma tiene de medicamento del mismo con la finalidad de que las pastillas sean almacenadas en un contenedor adecuado. Dicha ventana se puede visualizar en la imagen 214.

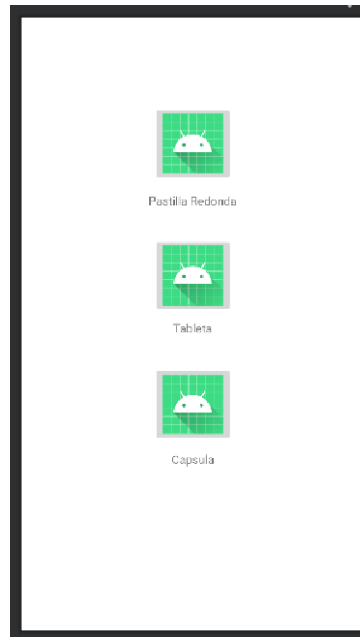


Imagen 214. Ventana de selección de tipo de medicamento.

Como se puede observar en la imagen 214, se busca tener 3 botones con formato de imagen para que el usuario seleccione la que más asemeja a la forma de su medicamento, además, estos botones pictográficos cuentan con una descripción que enuncia que tipo de medicamento es el mostrado en sus respectivas imágenes. Al presionar el botón, se configura el contenedor correspondiente y se lleva al usuario a la ventana de ingreso de nuevos tratamientos para que brinde la información respecto al que está ingresando.

- **Ventana de ingreso de nuevos tratamientos**

Esta ventana será la encargada de solicitar al usuario los datos e información sobre el tratamiento que desean agregar.

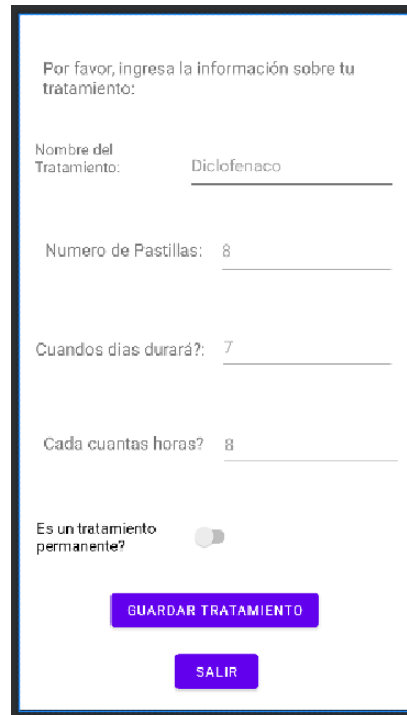


Imagen 215. Ventana de ingreso de nuevos tratamientos.

Como se observa en la imagen 215, el concepto procura mantener la organización de la información de forma oportuna y busca recopilarla de forma fácil e intuitiva a través de cajas de texto, selectores y un botón para almacenar la información en variables globales dentro de la aplicación de tal manera que sean accesibles por las diferentes actividades de la app que pudiesen requerirla como la de modificar o borrar tratamientos y otro para volver a la ventana inicial.

- **Ventana de ingreso de medicamentos**

Esta es una ventana auxiliar para el proceso de ingreso de medicamentos al dispositivo, su propósito es el de indicar al usuario que debe esperar a que el prototipo coloque el contenedor adecuado para su tratamiento en posición de ingreso de pastillas. Cuando el contenedor se coloca, la aplicación recibe la instrucción de indicar con un texto al usuario que ya puede ingresar sus pastillas y que haga clic en el botón de listo cuando termine de meterlos, este último solo es visible después de que el contenedor de pastillas requerido se ha colocado.

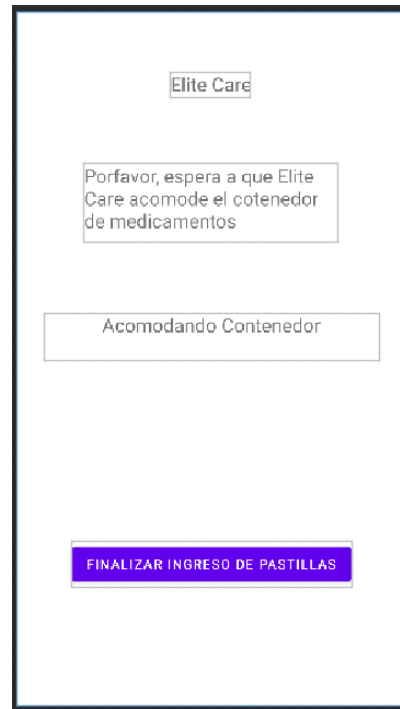


Imagen 216. Ventana de ingreso de medicamentos.

En la imagen 216 podemos ver que la interfaz cuenta con textos descriptivos, así como el botón de listo mencionado en la descripción de la ventana.

- **Ventana de gestión de tratamientos**

En el caso de esta ventana, el equipo decidió rediseñar su formato, pues el propuesto podría llegar a resultar confuso y conflictivo para el usuario ya que buscaba poder realizar cualquiera de las acciones desde la misma ventana, lo cual agrupaba demasiados botones y opciones en un mismo espacio.

Ahora, esta venta funcionará como enlace a otras tres sub-ventanas que permiten realizar tres actividades diferentes de forma aislada, mejor ordenada y mucho más intuitiva que antes, a las cuales el usuario accede clicando el botón con la actividad que sea que desee realizar.



Imagen 217. Ventana de gestión de tratamientos.

Como se puede ver en la imagen 217, se cuenta con los tres botones antes mencionados, así como con un botón para regresar a la ventana principal si es que no se quiere realizar ninguna acción.

- **Ventana para borrar un tratamiento**

Como su nombre lo indica esta ventana está diseñada para ser la encargada de borrar un tratamiento que el usuario ya no desea continuar, busca mantener la operación simple e intuitiva.

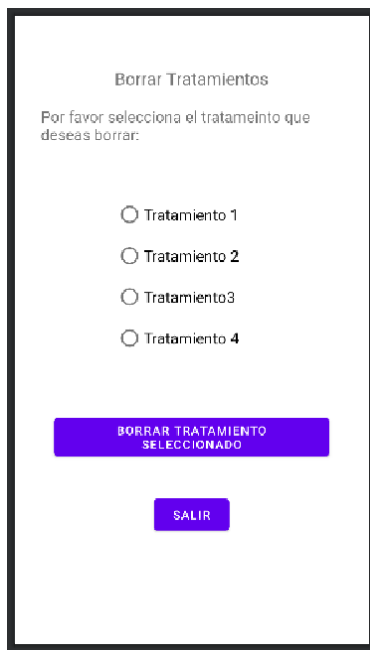


Imagen 218. Ventana para borrar un tratamiento.

Como se muestra en la imagen 218, la ventana cuenta con un selector con cuatro opciones pues es la máxima capacidad de tratamientos que tiene el dispositivo. A su vez, el botón de borrar tratamiento elimina los datos del tratamiento seleccionado de la base de datos en la nube, así como su respectiva alarma, mientras que el salir devuelve al menú principal.

- **Ventana para modificar un tratamiento**

De forma similar a la ventana anterior, esta fue diseñada para lograr que el usuario tenga un espacio exclusivo para poder editar las características de su tratamiento.

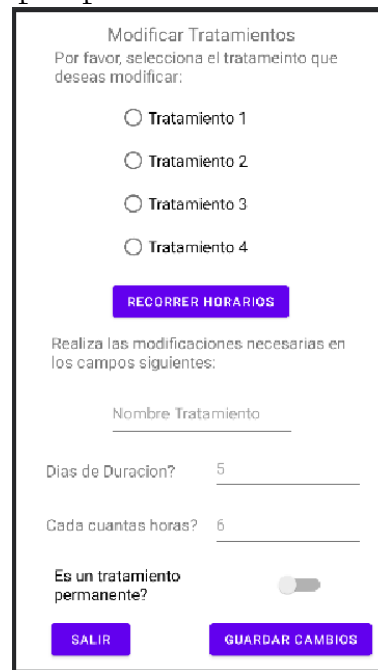


Imagen 219. Ventana para modificar un tratamiento.

La imagen 219, muestra como esta ventana permite seleccionar cual tratamiento se modificará con un selector y a su vez ingresar los nuevos datos por medio de cajas de texto y selectores sencillos. Finalmente, cuenta con tres botones uno que permite recorrer los horarios de ingesta en caso de omisión, otro para guardar los cambios ingresados y otro para salir a la ventana inicial en caso de no querer modificar ningún dato.

- **Ventana para reabastecer un tratamiento**

Como su nombre indica, el equipo diseño esta ventana para facilitar al usuario seleccionar cual tratamiento desea reabastecer, indicarle cuantas pastillas deberá ingresar y decidir si hacerlo o no.

Esta ventana, en lugar de un selector, cuenta con botones, cada uno con un nombre del tratamiento que, al ser presionados, indican al usuario la cantidad de pastillas necesarias para reabastecerlo en su totalidad. Además, cuenta con dos botones, uno para salir a ventana principal en caso de prescindir de la operación

y otro para ingresar los medicamentos en caso de continuar con el reabastecimiento del tratamiento seleccionado

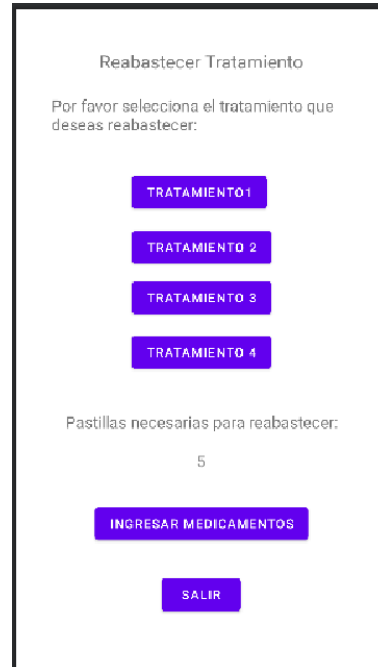


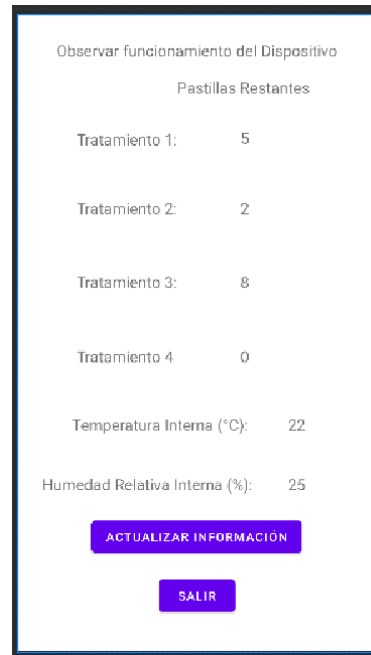
Imagen 220. Ventana para reabastecer un tratamiento.

La imagen 220, nos muestra la interfaz que, en perspectiva, semeja a la de borrar tratamientos, pues siguen el mismo concepto de diseño.

- **Ventana de monitoreo del dispensador**

Esta ventana es la que permitirá al usuario observar la temperatura y humedad internas de operación del dispensador, así como saber el número de pastillas disponible en cada tratamiento, será una ventana completamente informativa.

Como se puede ver en la imagen 221, la ventana cuenta con textos informativos para las pastillas restantes de cada tratamiento, así como para la temperatura y humedad internas del dispensador, además, cuenta con dos botones uno para salir a la ventana principal y otro para actualizar los datos mostrados. El equipo procuró apegarse en medida de lo posible al diseño conceptual inicial.



Observar funcionamiento del Dispositivo

Pastillas Restantes

Tratamiento 1:	5
Tratamiento 2:	2
Tratamiento 3:	8
Tratamiento 4:	0
Temperatura Interna (°C):	22
Humedad Relativa Interna (%):	25

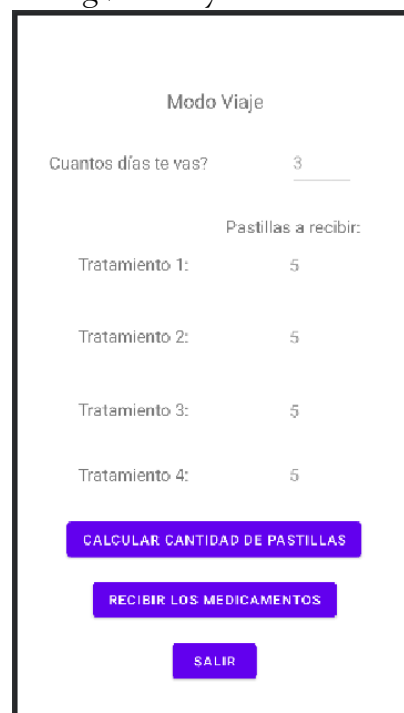
ACTUALIZAR INFORMACIÓN

SALIR

Imagen 221. Ventana de monitoreo del dispensador.

- **Ventana de modo viaje**

En el caso de la ventana de modo viaje, es la que permitirá al usuario solicitar dosis de medicamentos de sus tratamientos para hasta tres días en caso de que vaya a ausentarse un tiempo de casa y no pueda llevarse consigo el dispensador. El equipo procuró apegarse lo más posible al diseño conceptual modificando ligeramente para garantizar legibilidad y facilidad de interacción.



Modo Viaje

Cuantos días te vas?

Pastillas a recibir:

Tratamiento 1:	5
Tratamiento 2:	5
Tratamiento 3:	5
Tratamiento 4:	5

CALCULAR CANTIDAD DE PASTILLAS

RECIBIR LOS MEDICAMENTOS

SALIR

Imagen 222. Ventana de modo viaje.

Como se observa en la imagen 222, se cuenta con una caja de texto para indicar los días de ausencia, textos informativos sobre el número de pastillas que recibirá de cada tratamiento y dos botones, uno para recibir los medicamentos necesarios y otro para salir a la pantalla principal en caso de prescindir de la operación.

- **Ventana de notificación de ingesta de medicamento**

Esta es una venta auxiliar, la cual es desplegada al usuario cuando es momento de ingerir un medicamento junto a una notificación. Esta tiene la finalidad de informarle que le cuenta con una hora para ingerir la dosis o de lo contrario tendrá que recorrer los horarios de su tratamiento o esperar hasta la próxima dosis. El usuario deberá consultar con el medico a cargo del tratamiento cuál de las opciones es más adecuada de acuerdo a su padecimiento.

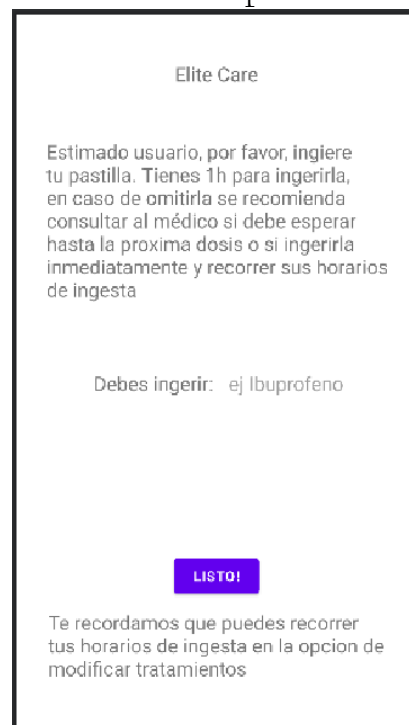


Imagen 223. Ventana de notificación de ingesta de medicamento.

Como se muestra en la imagen 223, la ventana cuenta con texto descriptivo e instrucciones para el usuario, así como un botón de listo, el cual deberá ser presionado por el usuario cuando haya tomado su medicamento y que lo devolverá al menú principal.

Todas estas ventanas se han desarrollado en XML agregando identificadores a los distintos tipos de botones, selectores, cajas de texto y textos fijos para que a su vez puedan ser vinculados con sus respectivas actividades basadas en Kotlin y puedan así tener su funcionamiento correspondiente. En el caso de los botones, estos son vinculados a variables que permiten el enlace entre ambos códigos permitiendo acceder a los eventos OnClick y así realizar acciones cuando sean presionados. En el caso de las cajas de texto

y textos fijos, no es necesario crear variables, pero su identificación permite acceder a sus propiedades internas como su texto actual y modificarlo o guardarlo para realizar alguna operación, lo cual es útil para cuando se busca solicitar o consultar un dato al usuario o hacerle llegar un mensaje o notificación. Adicionalmente, al proyecto se añadieron las librerías y paquetes para obtener los servicios de google Firebase Realtime Database y así poder emplearla como base de datos principal y puente de comunicación entre la tarjeta de control y el teléfono móvil del usuario. Los códigos de las actividades desarrolladas en Kotlin y sus respectivos archivos XML pueden visualizarse de manera completa en los apéndices de este documento.

A continuación, en la imagen 224, podemos ver el diagrama de flujo de funcionamiento de la aplicación móvil.

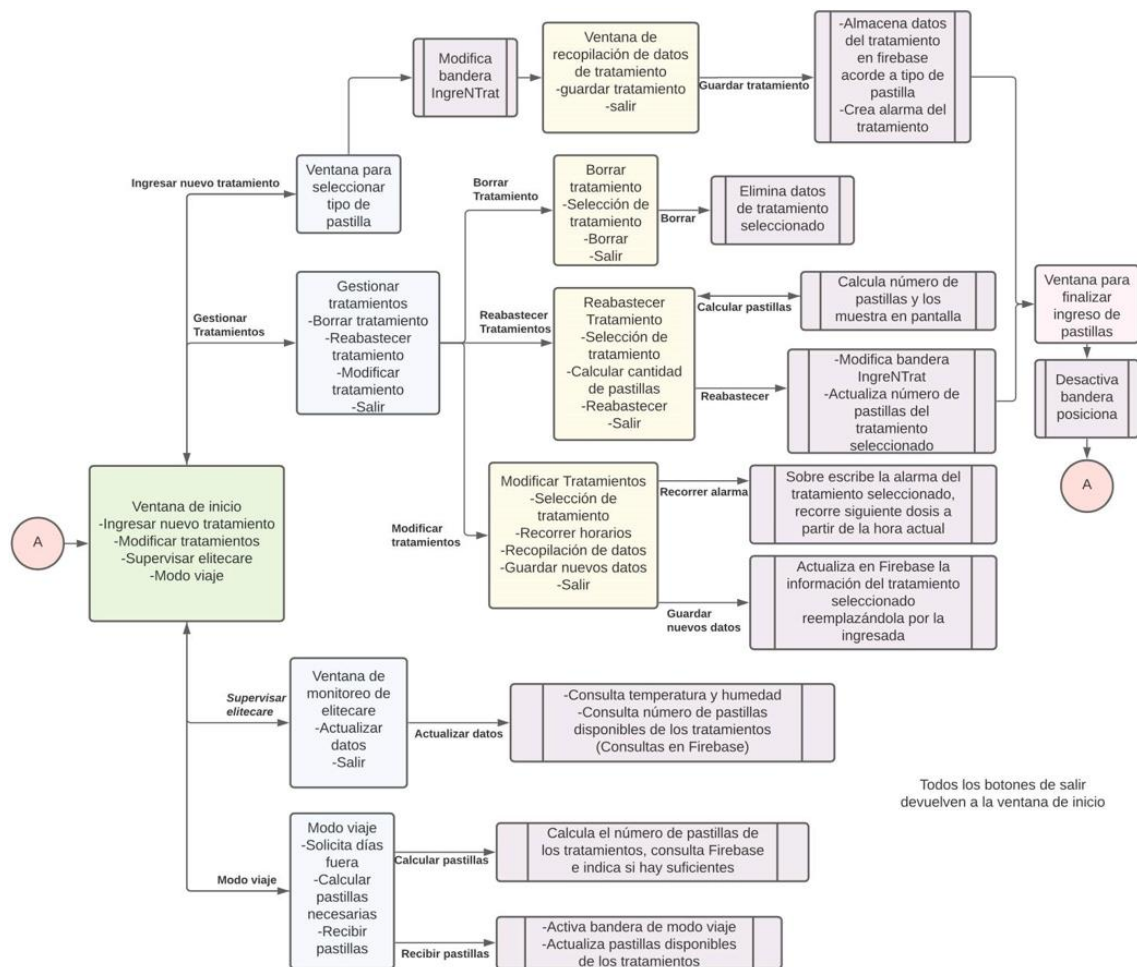


Imagen 224. Diagrama de flujo de funcionamiento de la aplicación móvil.

Software de la tarjeta de control

Atendiendo los objetivos específicos y los alcances del proyecto, se desarrolló, con base en el pseudocódigo de comportamiento propuesto del dispositivo propuesto en TTL, un programa en microPython, que es la extensión de Python para RaspberryOS, para

controlar los actuadores del dispositivo y realizar lecturas con el sensor de humedad y temperatura seleccionado por medio de los diversos puertos GPIO de la tarjeta.

Para ello se instalaron en la tarjeta el sistema operativo más reciente disponible para la tarjeta adquirida, posteriormente, lenguaje de programación Python y las librerías necesarias tanto para acceder al uso de los puertos GPIO para activar los actuadores y realizar lecturas con el sensor, como para la conexión a los servicios de nube Firebase de google.

Una vez recopilados los recursos antes mencionados, el equipo procedió a desarrollar el programa en el ambiente de desarrollo Tony que es la IDE oficial de Raspberry. Se implementó una programación basada en funcione con la finalidad de poder desarrollar los segmentos de código de manera organizada y además tener la posibilidad de ejecutar las funciones por separado para realizar pruebas de funcionamiento.

Por otra parte, el programa se desarrolló de tal manera que se mantiene en ejecución continua tras haber sido iniciado, además para la ejecución de los diferentes subprocesos que el dispositivo puede realizar, se cuenta con banderas enlazadas a la base de datos de Firebase y que son activadas por la aplicación móvil, estas indican al programa que subproceso se debe ejecutar.

Los subprocesos principales son: Ingreso de medicamentos al dispositivo, dispensado de medicamento, borrar un tratamiento y actualizar parámetros de operación. Como proceso principal, el programa siempre se mantiene consultando el valor de las banderas de Firebase, leyendo la temperatura y humedad relativa del dispensador y aplicando la acción de control de temperatura correspondiente de acuerdo al lazo cerrado de control que se aprecia en la imagen 225.

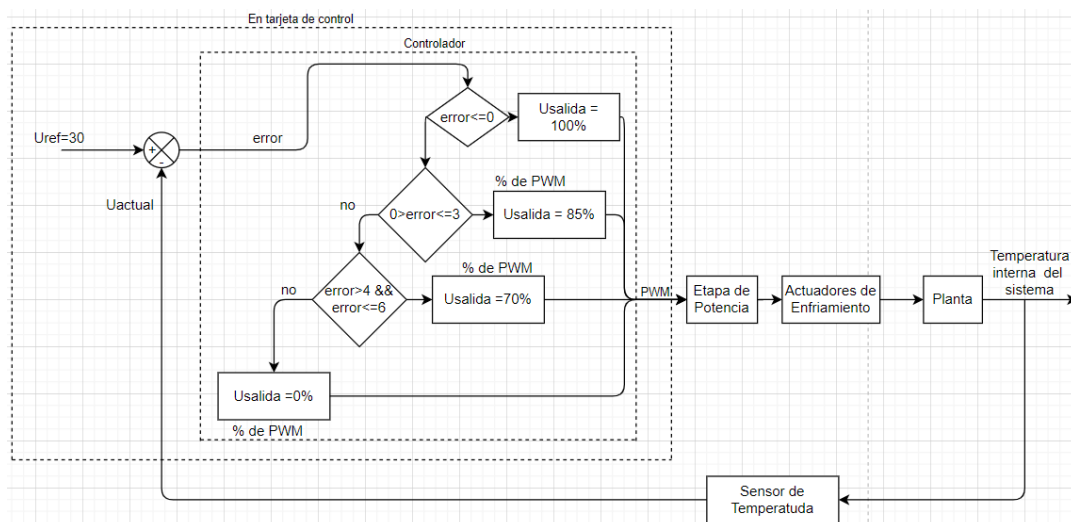


Imagen 225. Lazo cerrado de control de temperatura interna del dispensador.

Ingreso de medicamentos al dispositivo

Esta función se activa cuando la bandera de ingresar un nuevo tratamiento “IngreNTrat” es activada por el usuario. Para el desarrollo de este código, es preciso recordar que el

motor a pasos empleado para posicionar los contenedores mediante el mecanismo de revolver es uno bipolar de precisión y que, además, cuenta con 48 pasos por revolución. Por otra parte, también resulta importante recordar que la distribución de los contenedores de medicamento en el revólver hace que sus centros se encuentren a 12 pasos entre sí, es decir, existe una separación de noventa grados entre ellos.

En el caso del ingreso de medicamentos, se cuenta con una bandera que activa el proceso, esta puede adquirir valores de 0 a 4 haciendo referencia a los números de tratamientos que se almacenan en el dispositivo, si el valor de la bandera es cero, no se ejecuta el proceso, pero si es un valor entre uno y cuatro, se coloca en posición de entrada el contenedor correspondiente al número de tratamiento que indica la bandera. Esta acción permite que, dependiendo del tipo de tratamiento, se coloque un contenedor adecuado para el tipo de pastilla que ingresará el usuario.

Para tener una referencia de inicio, se cuenta con una posición denominada origen, en esta el contenedor de pastillas redondas también nombrado contenedor 1 se encuentra en posición de entrada de medicamentos, mientras que el contenedor de tabletas nombrado contenedor 2, que se encuentra a 24 pasos del contenedor 1, se encuentra en posición de salida. Esto se puede apreciar con mayor detalle en la imagen 226.

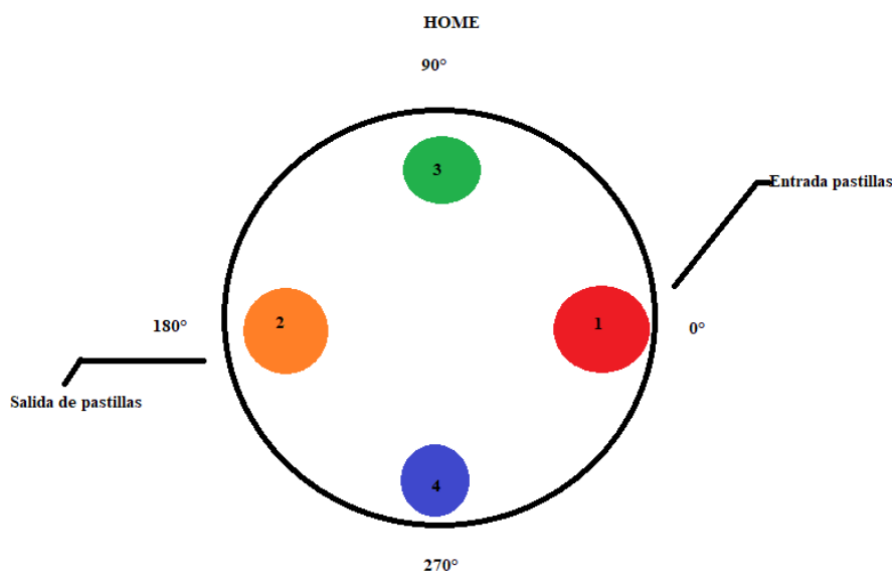


Imagen 226. Diagrama de posicionamiento de los contenedores de pastillas y el desplazamiento que deben recorrer.

Dicho lo anterior, para colocar el contenedor 2 en modo entrada se requieren veinticuatro pasos, para el contenedor de capsulas 3 se requieren doce pasos y para el contenedor de capsulas 4 se requieren treinta y seis pasos. El programa repite una secuencia de pasos completos y dado que la secuencia es repetitiva cada cuatro pasos, el programa toma el valor de la bandera para determinar el número de repeticiones necesarias para colocar el contenedor requerido. Es decir, la secuencia se repite 6 veces para colocar el contenedor 2, nueve veces para colocar el contenedor 3 y se repite tres veces para colocar el contenedor 4.



Una vez que el contenedor se ha colocado, se queda a la espera de una segunda bandera, que le indica cuando el usuario ha terminado de ingresar los medicamentos, cuando esta adquiere un valor de uno, automáticamente el programa vuelve a la posición de 'home' recorriendo el restante de pasos para terminar una revolución.

Dado que se cuenta con un motor a pasos de precisión y un driver para el movimiento del mismo, no se considera necesaria una retroalimentación de lazo cerrado para confirmar cuantos pasos se han recorrido, aunque podría agregarse como trabajo a futuro.

Extracción de medicamentos

Cuando se activa la bandera para dispensar un tratamiento llamada "numTrata" en Firebase, automáticamente se ejecuta esta función. Tal como se explicó en la función de ingresar un nuevo tratamiento, el código ubica el contenedor requerido en posición de salida siguiendo una secuencia de pasos de cuatro en cuatro. Una vez se ha ubicado el contenedor en la posición de salida, la tarjeta activa la electroválvula de dispensado de agua y se desactiva hasta que el sensor de nivel de agua del vaso marca llenado. Hecho lo anterior, se activa el solenoide elevador para alinear el solenoide extractor, éste último es activado de manera consecuyente, su movimiento lineal permite abrir la compuerta de salida de pastillas del contenedor posicionado, en otras palabras, funge como una compuerta electromecánica. Posteriormente, la tarjeta desactiva el solenoide extractor permitiendo que la compuerta del contenedor restituya su posición cerrada por la acción de un resorte, después, se desactiva el solenoide elevador para evitar colisiones y se hace sonar una alarma al usuario.

Una vez que se han ejecutado las acciones mencionadas en el párrafo anterior, el programa procede a colocar el motor a pasos unipolar en posición de transmisión de movimiento mediante la activación de otro actuador push-pull denominado solenoide de transmisor, posteriormente, se realiza una revolución completa con el motor a pasos para hacer girar al tornillo sinfín del contenedor y de esta manera dejar una nueva pastilla lista para su futura extracción. Finalmente, la función devuelve el revolver a la posición de home.

Borrar un tratamiento

Esta función se encarga de eliminar las pastillas restantes de un tratamiento que el usuario ha decidido borrar si es que aún queda alguna almacenada en su contenedor. Se activa cuando la bandera "borrar" es activada en Firebase.

Consiste en la misma lógica de posicionamiento del contenedor requerido a partir del valor indicado por la bandera. Una vez que el revolver lo ha posicionado en modo de salida, el programa activa el solenoide elevador luego el solenoide extractor y, con estos permaneciendo activos, procede a activar también el solenoide transmisor y el motor a pasos unipolar para hacer rotar el tornillo sinfín el mismo número de revoluciones que pastillas restantes en el contenedor extrayéndolas por completo del mismo.

Una vez se han extraído las pastillas, desactiva todos los solenoides push-pull y finalmente lleva el revolver a la posición de home.

Actualizar datos de operación del dispensador

Esta función es el subproceso más sencillo pues lo único que realiza es la actualización de los valores de los parámetros de operación que son la temperatura y humedad relativa internas del dispensador en la base de datos de Firebase para que estos puedan ser consultados por el usuario desde la aplicación móvil.

Los valores son actualizados durante la ejecución continua de forma automática pero no en tiempo real ya que eso consumiría demasiado rápido los datos de descarga disponibles para la versión gratuita de Firebase Realtime Database. Por ello, los valores se actualizan aproximadamente cada 2 minutos.

Modo viaje

Es la función encargada de extraer las dosis necesarias de cada tratamiento existente en el dispensador. Lo anterior se realiza primero colocando los contenedores uno a uno en posición de salida, posteriormente, se activan el solenoide elevador, el solenoide extractor y finalmente el solenoide de transmisor, finalmente se hace girar el tornillo del contenedor el número de veces equivalente al número de dosis requeridas extrayendo así las pastillas necesarias del tratamiento.

Esta función coloca todos los contenedores en posición de salida incluso si no se extrae ninguna pastilla, de esta forma al final queda en la posición de home.

La imagen 227 se muestra un diagrama de flujo ilustrativo del funcionamiento del programa en la tarjeta Raspberry, esto con la finalidad de tener una forma más didáctica de presentar el funcionamiento. imagen n se muestra un diagrama de flujo ilustrativo del funcionamiento del programa en la tarjeta Raspberry, esto con la finalidad de tener una forma más didáctica de presentarlo.

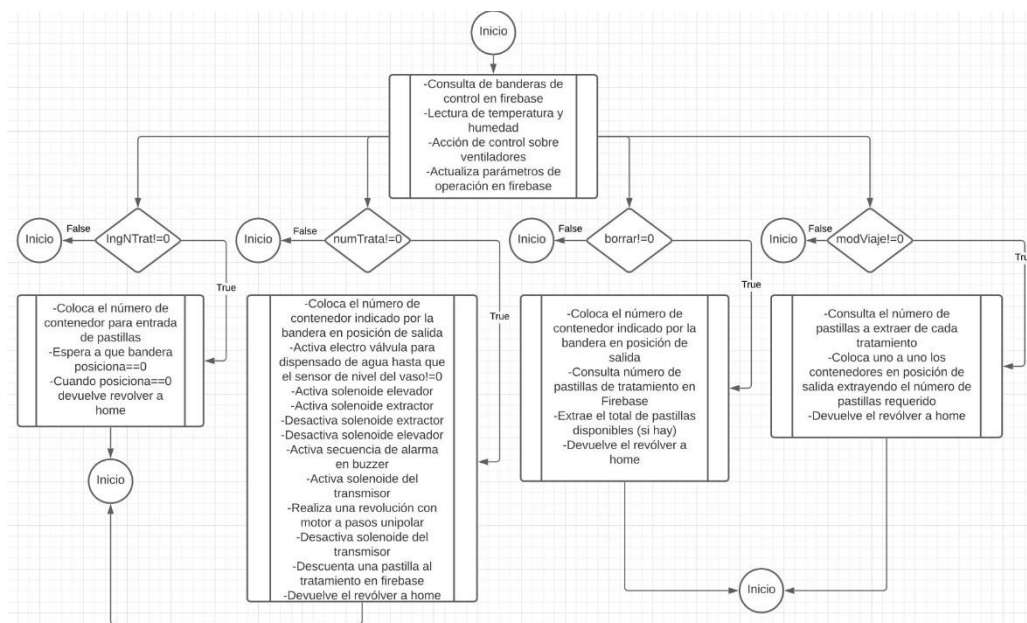


Imagen 227. Diagrama de flujo ilustrativo del funcionamiento del programa en la tarjeta Raspberry.

El código desarrollado en Python de este programa puede ser consultado en los apéndices de este documento.

Integración de software

Como ya se mencionó anteriormente, el equipo desarrollo una aplicación móvil y un programa para la tarjeta de control y procesamiento de datos, sin embargo, estas deben operar de manera sincronizada, para ello, se ha implementado una base de datos en la nube empleando los servicios de Google Firebase. Esta nube permite enlazar sistemas desarrollados en diferentes plataformas y, en específico, el servicio de Realtime Database de Firebase permite que la conexión sea en tiempo real ya que todas las instancias conectadas a la base de datos por medio de un cliente a través de conexión a internet comparten una instancia de la base de datos y reciben actualizaciones automáticamente con los datos más reciente.

Para esto el equipo añadió los SDK, librerías y paquetes del servicio tanto en la aplicación como en el programa de Python y, además, creó una base de datos en la consola de Firebase con las diferentes variables que requiere para el correcto funcionamiento del sistema. Esta base de datos, tal como se menciona en el marco teórico, tiene un formato de árbol tipo JSON (JavaScript Object Notation por sus siglas en ingles), el cual permite organizar información en ramas y nodos que pueden contener los siguientes tipos de datos: numéricos, boléanos, cadenas de texto, null, arreglos y objetos.

En las imágenes 228, se puede apreciar la base de datos generada, así como sus contenidos y sus respectivos valores.

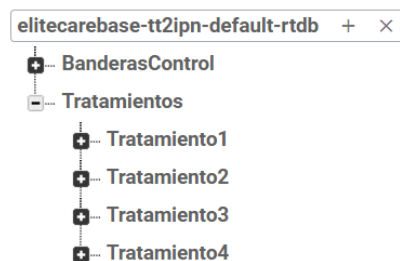


Imagen 228. Base de datos generada para la comunicación del dispensador con la aplicación móvil.

En la imagen 228, se puede ver la base de datos creada en formato de árbol y sus dos principales ramas llamadas “BanderasControl” y “Tratamientos” respectivamente, sus contenidos se pueden apreciar a detalle en las imágenes 229 y 230.

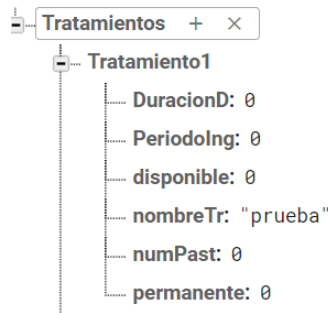


Imagen 229. Rama Tratamientos de la base de datos.

La imagen 229 se muestra la organización de la información de los tratamientos que almacena el usuario, esta rama llamada tratamientos contiene cuatro nodos que a su vez contienen la información del tratamiento. Esta rama no ejerce acciones de control sobre el dispensador y se usa para poder guardar de forma persistente fuera de la aplicación la información que el usuario ingresa cuando crea un nuevo tratamiento.

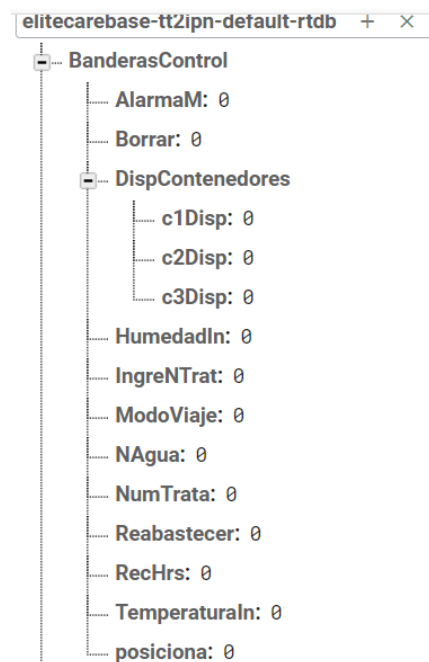


Imagen 230. Rama BanderasControl de la base de datos.

Como se observa en la imagen 230, se cuenta con una rama dedicada exclusivamente a las banderas de control del sistema, estas, en su mayoría, se emplean para comunicar a la tarjeta de control que debe realizarse un subproceso específico.

La base de datos funciona como puente de comunicación entre la aplicación móvil y la tarjeta Raspberry, pues, dependiendo de las acciones que ejecuta el usuario en la app, las banderas de control adquieren valores que activan los subprocesos del dispensador.

A continuación, en la imagen 231, se logra observar un diagrama ilustrativo que ejemplifica la conexión entre ambos sistemas y la transferencia de datos que estos realizan.

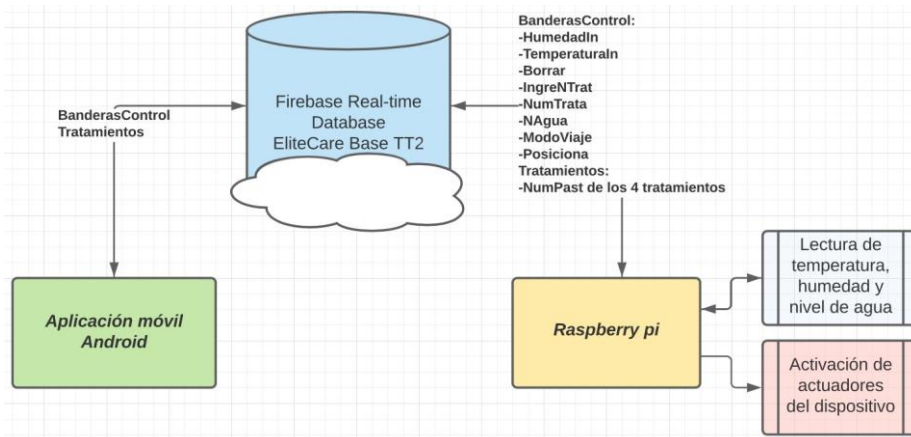


Imagen 231. Diagrama de conexión de la base de datos con la aplicación móvil y la Raspberry Pi 3.

Las acciones de control que se efectúan en el dispensador cuando las banderas de control son modificadas se encuentran descritas en el apartado de “Software de la tarjeta de control”. Por otro lado, la aplicación móvil no realiza acciones de control, más bien es la encargada de activar las banderas para que se realicen los procesos correspondientes a las acciones que el usuario efectúe en la misma y de recopilar y gestionar la información de los tratamientos del usuario.

Mecanismo de ingreso de medicamentos

Las piezas de este mecanismo fueron impresas acorde a los modelos CAD de diseño detallado, por lo que cuentan con muescas y orificios o rieles para asegurar su ensamblaje y garantizar la fijación necesaria durante la operación del dispositivo.

El equipo ensambló primero el revólver, los contenedores de medicamento y sus respectivos tornillos, tapas y compuertas de extracción de medicamentos. En la imagen 232, se logra observar el ensamblaje de los contenedores y sus respectivos tornillos y tapas.



Imagen 232. Ensamblaje de los contenedores con el revólver.

En la imagen 232, se visualiza a su vez se han ensamblado las compuertas de extracción de medicamento al revolver siendo cada una correspondiente a un tipo de pastilla diferente.

Posteriormente, como se ve en la imagen 233, se ensamblaron los contenedores y el motor a pasos bipolar PM55L al revolver, teniendo así, completamente ensamblado el mecanismo principal.



Imagen 233. Ensamble del revolver con los contenedores al motor a pasos bipolar.

Por otra parte, se ensambló a la carcasa correspondiente la rampa de entrada de medicamentos como se aprecia en la imagen 234, mientras que en la imagen 235, se observa el ensamble del cono de entrada, su sujetador y la carcasa que lo sostiene.



Imagen 234. Ensamble de la carcasa principal de entrada, junto con la rampa de entrada de medicamentos.



Imagen 235. Ensamble del embudo de entrada de pastillas junto con su soporte.

Mecanismo de extracción de medicamentos

Debido a cuestiones de facilidad de ensamble, las compuertas de extracción fueron ensambladas al revolver junto con las piezas del mecanismo de ingreso de medicamentos. Adicionalmente, sabemos que el mecanismo de entrada comparte la mayoría de sus piezas con el de extracción de medicamentos, en sí, todo el mecanismo del revolver incluyendo los contenedores y sus tornillos. Dicho esto, El equipo procedió a ensamblar el resto de los actuadores que pertenecen solo al sistema de extracción de pastillas en sus espacios correspondientes.

En la imagen 235, se muestra el ensamble de los solenoides push-pull en el mecanismo de elevador diseñado por el equipo. Este mecanismo es el encargado de abrir las compuertas de extracción de medicamentos cuando es momento de dispensar una pastilla.

La imagen 236, muestra el ensamble del motor a pasos unipolar 28BYJ-48, y su transmisor de movimiento a su mecanismo de riel junto con el actuador push-pull que lo desplaza para hacer contacto con los tornillos. Como se menciona en el diseño detallado, este mecanismo se encarga de transmitir el movimiento rotacional a los tornillos por medio de contacto tangencial cuando se requiere extraer una pastilla. El mecanismo de riel hace que el motor y el transmisor sean retractiles para evitar colisiones cuando el revolver gira para colocar un contenedor en posición de entrada o salida de medicamento.



Imagen 236. Ensamble del motor a pasos unipolar con los transmisores de movimiento de los tornillos sin fin.

Finalmente, se ensambló el cono de salida de medicamentos a su sujetador y a su carcasa correspondiente como se aprecia en la imagen 236.

Mecanismo de dispensado de agua

Este mecanismo consta únicamente de 3 piezas, la electroválvula, el contenedor de agua y su manguera de conexión, las cuales a su vez se ensamblan al dispositivo. Primero, para poder conectar la manguera a la electroválvula se ha implementado un racor de $\frac{1}{4}$ ", este, permite que la manguera entre a presión y se tenga un lado roscado para conectar a la electroválvula de forma segura y evitando fugas, esto puede apreciarse en la imagen 235.

Posteriormente, se ensambló el contenedor de agua a su compartimiento especial en la carcasa posterior del dispositivo con su respectivo sensor XKCY25 para detectar el nivel de agua y finalmente se conectó la manguera a su salida de agua como se muestra en la imagen 235.

Mecanismo de ventilación y disipación de humedad

Para el sistema de ventilación el equipo empleó 2 ventiladores sin escobillas de 12V de CD, estos ventiladores fueron descritos en las actualizaciones de diseño del sistema, requieren una conexión directa a sus respectivas etapas de potencia y fueron colocados en la parte superior frontal del dispensador.

El equipo diseñó e imprimió un compartimiento para material desecante, el cual se ensambla como una gaveta en la parte inferior frontal de la carcasa del prototipo, este cajón permite al usuario colocar bolsas de material desecante para garantizar un ambiente libre de humedad en el interior del dispositivo. En la imagen 237, se puede observar cómo se ensambla a la carcasa.



Imagen 237. Ensamble de la caja para desecante con la carcasa inferior.

Cabe mencionarse que, para calcular una cantidad de material desecante adecuada, se empleó la expresión:

$$\text{Unidades de Desecante} = K * V$$

$$\text{Donde } K = 0.161 \frac{1}{\text{gal}} \text{ ó } 0.0007 \frac{1}{\text{in}^3} \text{ ó } 1.2 \frac{1}{\text{ft}^3}; \quad V = \text{Volumen del contenedor}$$

De acuerdo con la expresión anterior entonces, se obtiene que:

$$\text{Unidades de Desecante} = 1.2 \left[\frac{1}{\text{ft}^3} \right] * 0.978 \text{ ft}^3 = 1.1736 \cong 1.2$$

Recordemos que una unidad DIN equivale aproximadamente a 30g de desecante, por lo que podemos obtener los gramos necesarios:

$$\text{Gramos de Desecante} = 1.2 * 30g = 36g$$

Dependiendo de la presentación en que se aplica el material, esto puede equivaler a 4 sobres de gel sílice, ya que es el desecante más común y comercial y comúnmente su presentación es de 10 gramos. El cajón tiene la capacidad para almacenar hasta 5 sobres de 10 gramos, por lo que el usuario solo debe colocar el gel de silicato en la presentación que mejor prefiera en el cajón hasta una línea indicada.



6. Resultados

Prueba de PWM para los ventiladores

Para realizar la prueba del PWM, se hizo el código en lenguaje de Python para ejecutarlo en RaspberryOS, todo mediante el uso la tarjeta de Raspberry. Se conectaron tres push botón y un led de color azul. Cada botón se conectó a una resistencia de pull-down de 1 K Ω y a la alimentación de 3.3V, entre la terminal del botón y positivo, se conectó al pin GPIO de entrada correspondiente. Los pines GPIO de entrada que se utilizaron fueron: 11, 18 y 29. El led se conectó a una resistencia de pull-down de 1 K Ω y su ánodo se conectó al pin de salida GPIO respectivo. El pin GPIO de salida que se usó fue: 32.

A continuación, se presentan en las imágenes x1, x2 y x3, las pruebas realizadas a diferentes porcentajes del PWM, los cuales fueron: 70, 85 y 100. Primero se presenta el caso donde la temperatura es baja, alrededor de 24 °C (imagen 238), en este se observa en la pantalla del compilador el mensaje: 'Temperatura Baja: Porcentaje Mínimo' y la intensidad luminosa del led es poca.

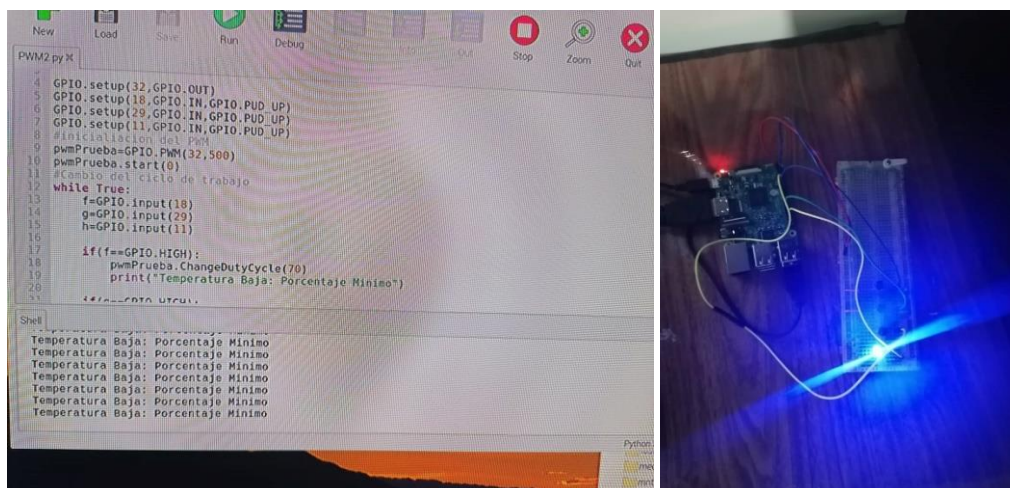


Imagen 238. Prueba del PWM en el caso de una temperatura baja.

En el segundo caso, la temperatura es media, alrededor de 27 °C (imagen 239), es por lo que ahora se mostró en la pantalla el mensaje: 'Temperatura Media: Porcentaje Medio' y la intensidad del led incrementó ligeramente.

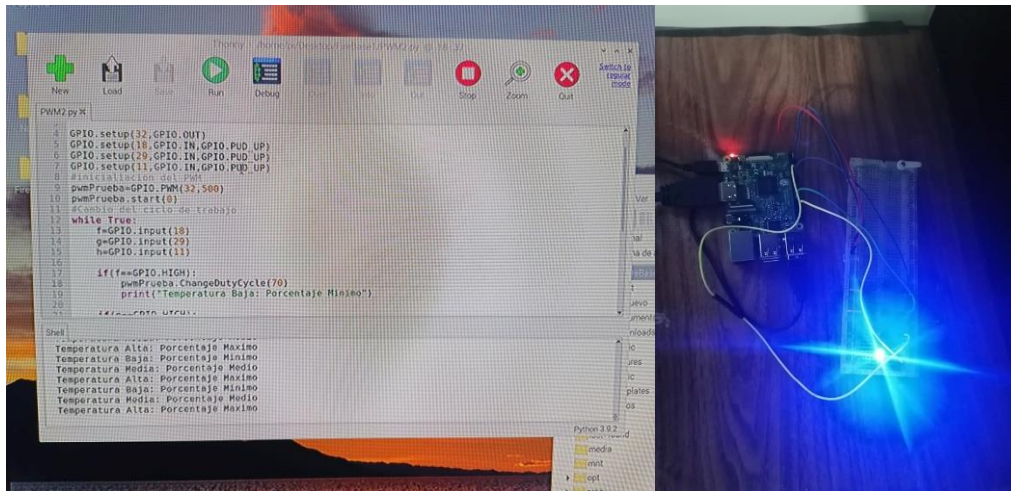


Imagen 239. Prueba del PWM en el caso de una temperatura media.

Para el último caso, donde la temperatura es alta, suponiendo estar a 30 °C (imagen 240), en la pantalla se mostró el mensaje: ‘Temperatura Alta: Porcentaje Máximo’ y la intensidad del led era la más alta.

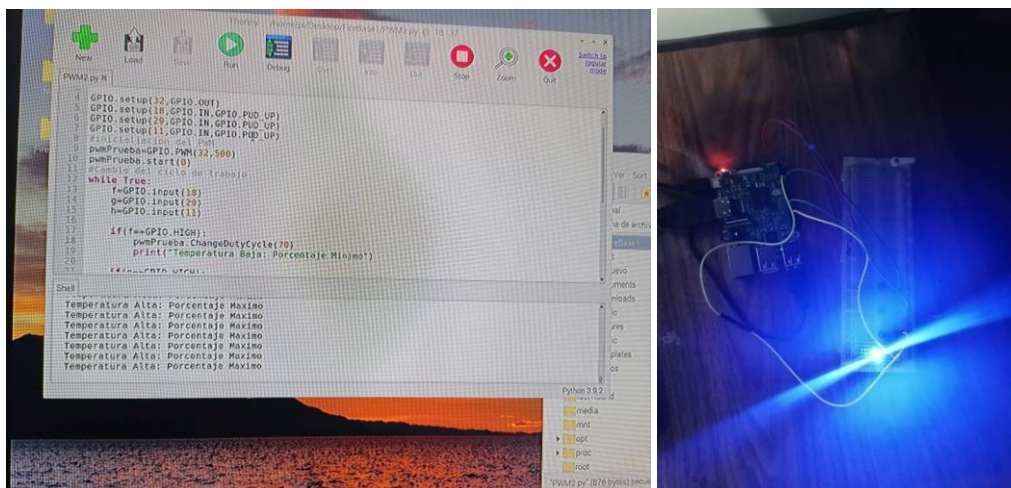
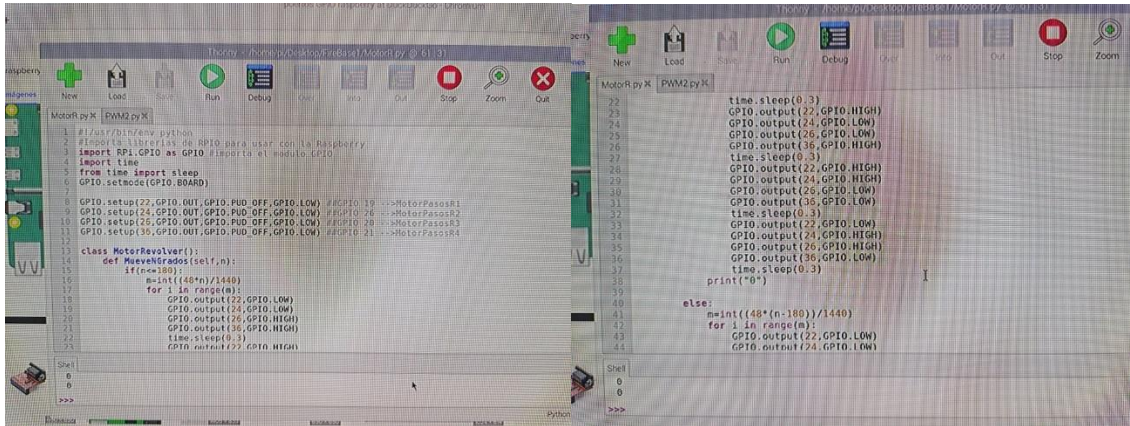


Imagen 240. Prueba del PWM en el caso de una temperatura alta.

Prueba de la secuencia para el motor a pasos bipolar

Para realizar esta prueba se declararon los pines GPIO: 22, 24, 26 y 36, como salidas; y se sustituyó el motor a pasos por 4 leds de diversos colores. Entonces dichos leds se conectaron junto con una resistencia del pull-down de 1 KΩ y su ánodo se le conectó la salida que le correspondía.

En la imagen 241, se puede visualizar el código elaborado para que los leds encendieran de dos en dos imitando la secuencia de paso completo que se le debe enviar al motor a pasos.



```
1 #!/usr/bin/env python
2 # Import librerías de GPIO para usar con la Raspberry
3 import RPi.GPIO as GPIO # Importa el módulo GPIO
4 import time
5 from time import sleep
6 GPIO.setmode(GPIO.BOARD)
7 GPIO.setup(22, GPIO.OUT, GPIO.PUD_OFF, GPIO.LOW) #GPIO 19 --> MotorPasosS1
8 GPIO.setup(24, GPIO.OUT, GPIO.PUD_OFF, GPIO.LOW) #GPIO 26 --> MotorPasosR2
9 GPIO.setup(26, GPIO.OUT, GPIO.PUD_OFF, GPIO.LOW) #GPIO 20 --> MotorPasosR3
10 GPIO.setup(30, GPIO.OUT, GPIO.PUD_OFF, GPIO.LOW) #GPIO 21 --> MotorPasosR4
11
12 class MotorRevolucion():
13     def MueveGrados(self, n):
14         if n <= 180:
15             #int((40*n)/1440)
16             for i in range(n):
17                 GPIO.output(22, GPIO.LOW)
18                 GPIO.output(24, GPIO.LOW)
19                 GPIO.output(26, GPIO.HIGH)
20                 GPIO.output(30, GPIO.HIGH)
21                 time.sleep(0.3)
22                 GPIO.output(22, GPIO.HIGH)
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Imagen 241. Código de la secuencia para el motor a pasos bipolar.

Ahora, se muestra en la imagen 242, que los leds se encienden correctamente siguiendo la secuencia de paso completo. Esto indica que debería funcionar al momento de implementarlo con el motor a pasos bipolar.

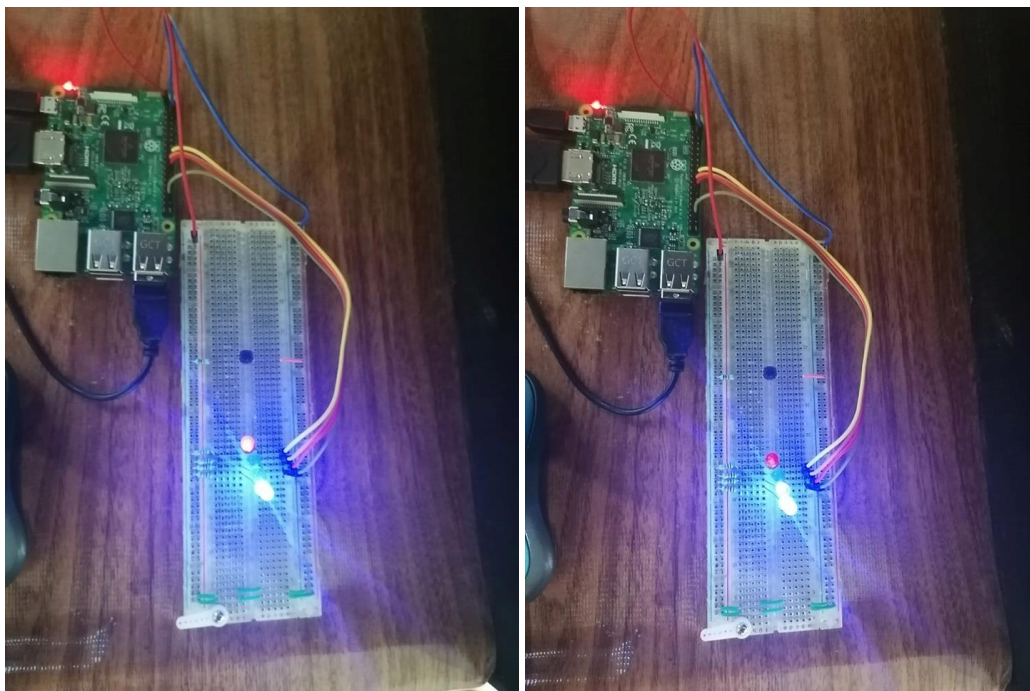


Imagen 242. Secuencia del paso completo representado con los leds.

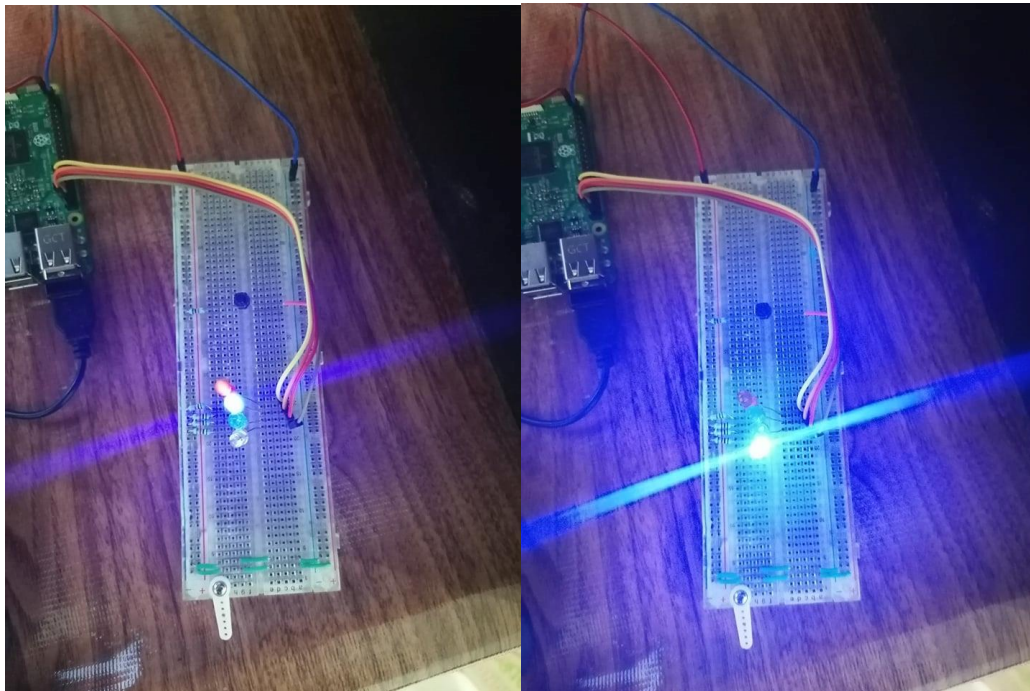


Imagen 244. Secuencia de medio paso representado con los leds.

Como salida se utilizaron 4 leds para visualizar las señales digitales que irán conectadas al controlador y posterior fase de potencia con una resistencia de 1 K Ω . La diferencia de esta secuencia es que el motor solo debe girar hacia una dirección y no en dos sentidos como el motor del revolver, a su vez al tener los tornillos diferentes diámetros el número de pasos que debe girar es variable por cada tipo (ver imagen 244).

Prueba del sensor de temperatura y humedad

Para poder hacer esta prueba se tuvo que instalar la librería correspondiente para el sensor DHT11, la cual se llama Adafruit_DHT. Entonces lo que se hace es declarar el sensor con un método de la librería mencionada y declarar el pin 23 como entrada de la señal mandada por el sensor. Después de eso, se declaran las variables de temperatura y humedad que son las que se leerán del sensor. Esto se aprecia en el código de la imagen 245.

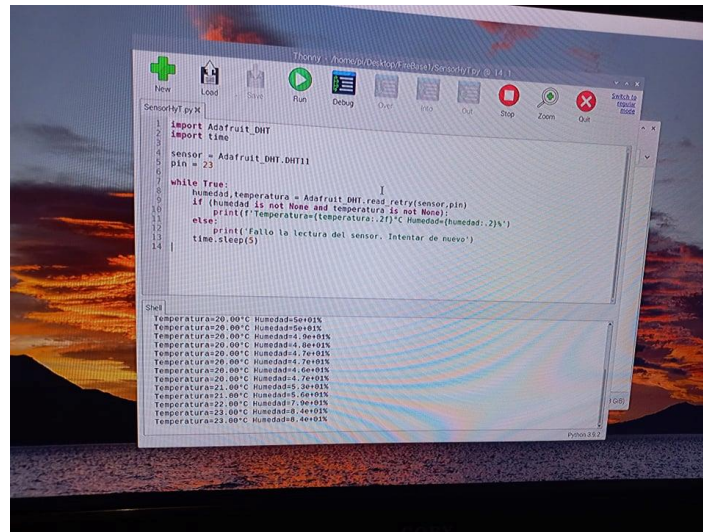


Imagen 245. Código para medir temperatura y humedad con el sensor DHT11.

Al momento de medir, en la pantalla del compilador aparece la temperatura medida en °C, así como también la humedad medida en porcentaje. Para asegurarnos de que sí lograba medir diferentes temperaturas, se colocó encima del mueble para que mostrara la temperatura en el lugar donde se probó, luego se encerró con la mano y empezó a subir la temperatura. Esto se puede observar en la imagen 246.

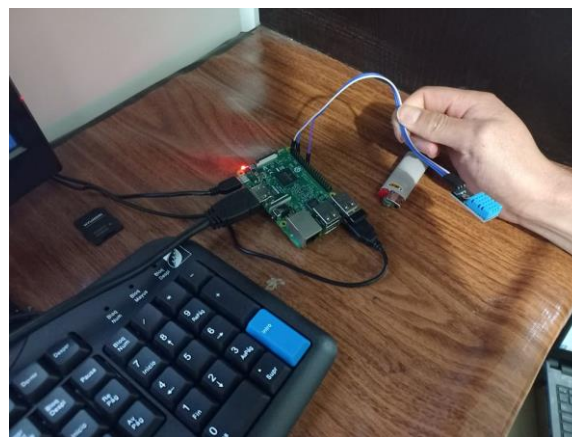


Imagen 246. Prueba física del sensor de temperatura y humedad, DHT11.

Pruebas de la aplicación móvil

Se realizaron pruebas de funcionamiento sobre las principales ventanas de interacción del usuario para corroborar la operación de la aplicación, así como su conexión con la base de datos de Firebase.

Ventana para ingresar nuevos tratamientos:

Se seleccionó el tipo de pastilla redonda, por lo que la bandera deberá para ingresar nuevo tratamiento “IngreNTrat” deberá tener el valor de uno. A continuación, en la imagen 247, se muestra la ventana en cuestión y los datos ingresados para la prueba.

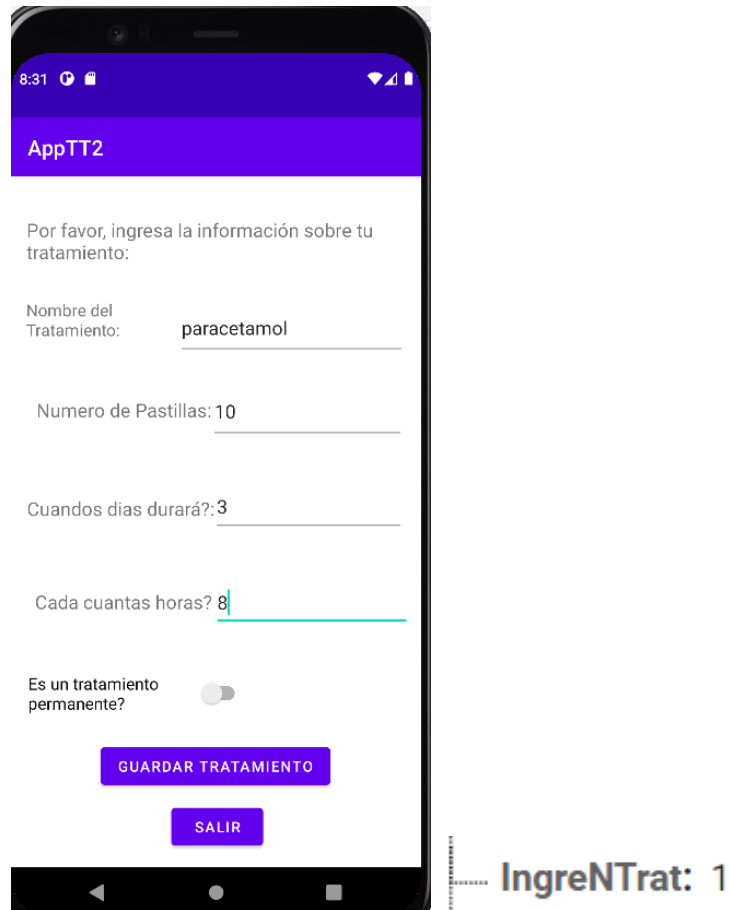


Imagen 247. Ventana de ingreso de nuevo tratamiento con información recopilada y la bandera de IngreNTrat activada.

En la imagen 248, se puede observar los datos de tratamiento antes y después de la prueba, lo cual muestra como la información ha sido recabada en la base de datos.

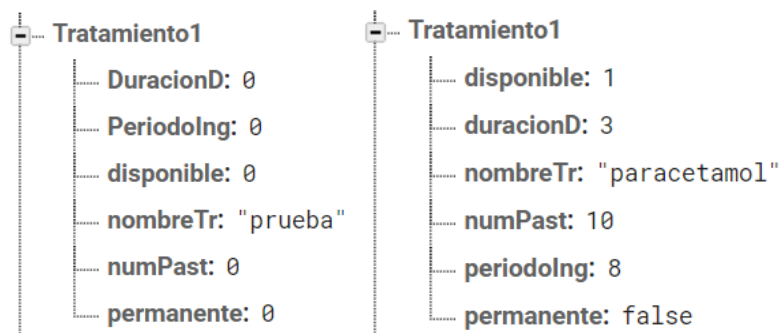


Imagen 248. Recopilación de los datos de un nuevo tratamiento de pastillas redondas en Firebase.

Ventana de modificar tratamiento

En la imagen 249, se muestra que se seleccionó modificar el tratamiento 1 de pastillas redondas, al modificar los parámetros del tratamiento en la app y haber solicitado que se guarden, los nuevos datos son actualizados en la base de datos.

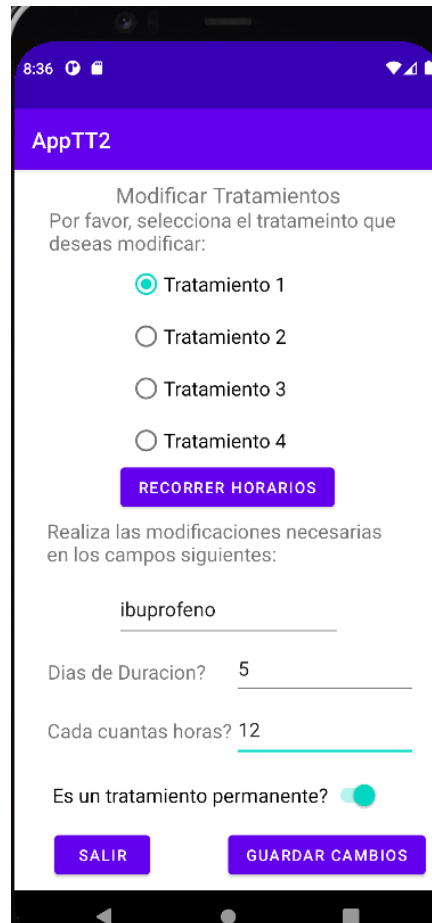


Imagen 249. Ventana de modificar tratamiento con nuevos datos recopilados.

La imagen 250, muestra los datos del tratamiento antes y después de ser modificados

Tratamiento1	Tratamiento1
disponible: 1	DuracionD: 3
duracionD: 3	PeriodoIng: 8
nombreTr: "paracetamol"	disponible: 1
numPast: 10	nombreTr: "ibuprofeno"
periodoIng: 8	numPast: 10
permanente: false	permanente: true

Imagen 250. Cambio en los datos del tratamiento uno tras haber solicitado guardar los cambios realizados.

Ventana de reabastecer tratamiento

En esta ventana se seleccionó reabastecer el tratamiento uno, por lo que vemos el número de pastillas necesarias a ingresar, al clicar ingresar pastillas la bandera “IngreNTrat” adquiere el número de tratamiento a reabastecer. Esto se visualiza en la imagen 251.

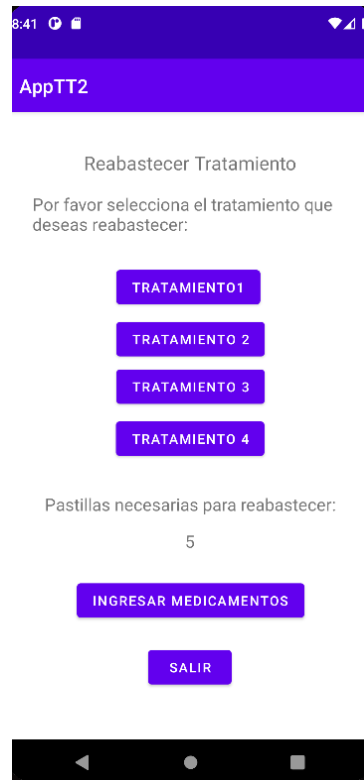


Imagen 251. Ventana de reabastecer tratamientos habiendo consultado reabastecer tratamiento 1.



Ventana de borrar tratamiento

Esta ventana nos muestra que se seleccionó eliminar el tratamiento 1, por lo que, al confirmar, sus datos son borrados de la base de datos y se sustituyen por valores de default. (Ver imagen 252).

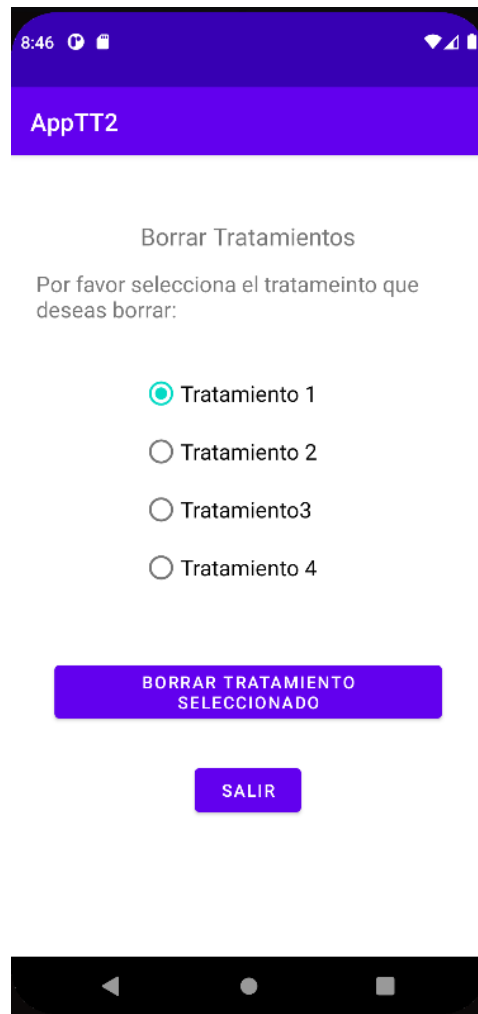


Imagen 252. Ventana para borrar tratamientos seleccionando eliminar tratamiento 1.

En la imagen 253 se logra observar cómo los datos del tratamiento pasaron de tener un contenido a valores default y que la bandera de control “borrar” adquiere el valor del tratamiento a ser eliminado.

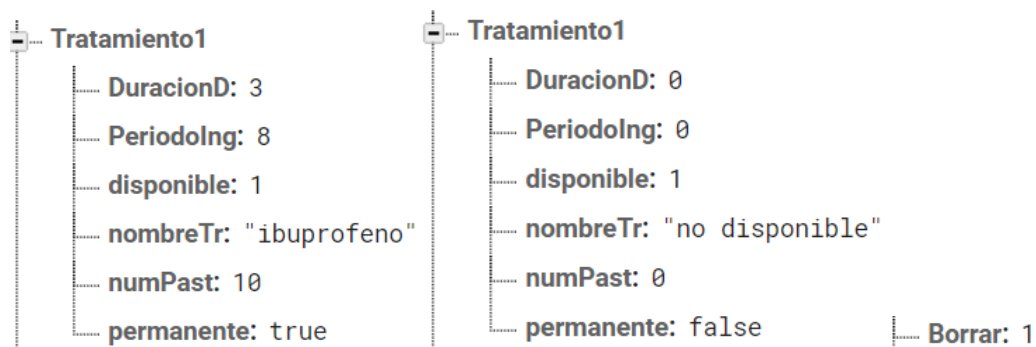


Imagen 253. Cambio en los datos del tratamiento uno y la bandera borrar en la base de datos.

Ventana de monitoreo del dispensador

Esta ventana es puramente informativa, se aprecia, en la imagen 254 que se presionó actualizar datos y en la ventana se despliegan los datos referentes a los tratamientos, así como los parámetros de operación del dispensador y que se corresponden con los datos existentes en Firebase.

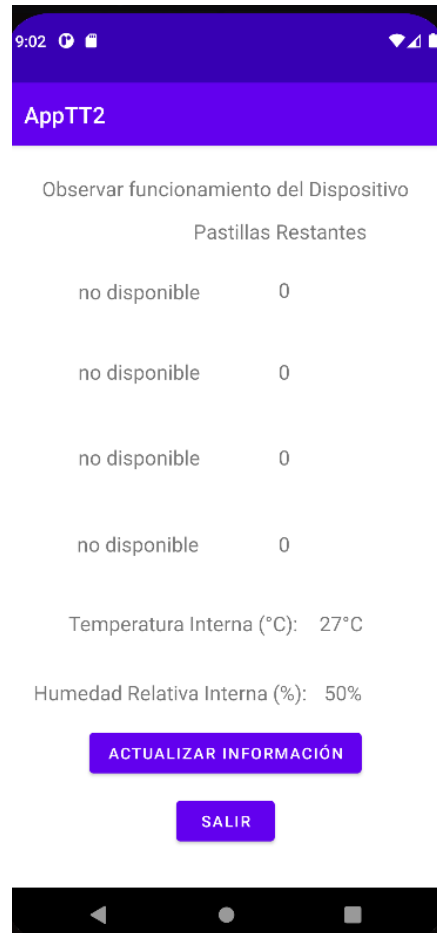


Imagen 254. Ventana de monitoreo consultando información más reciente de Firebase.

En las imágenes 255, se logran ver los valores de los datos de los tratamientos no disponibles y de los parámetros de operación del dispositivo.

Tratamiento1	DuracionD: 0	
	PeriodoIng: 0	
	nombreTr: "no disponible"	
	numPast: 0	
	permanente: false	
Tratamiento2	DuracionD: 0	
	PeriodoIng: 0	
	nombreTr: "no disponible"	
	numPast: 0	HumedadIn: "50%"
	permanente: false	Temperaturaln: "27°C"

Imagen 255. Información consultada de Frirebase por la aplicación móvil.

Ventana de modo viaje

La imagen 256, muestra que el usuario ha indicado en la ventana que estará ausente 2 días y se han calculado las pastillas que requiere para el periodo de tiempo.

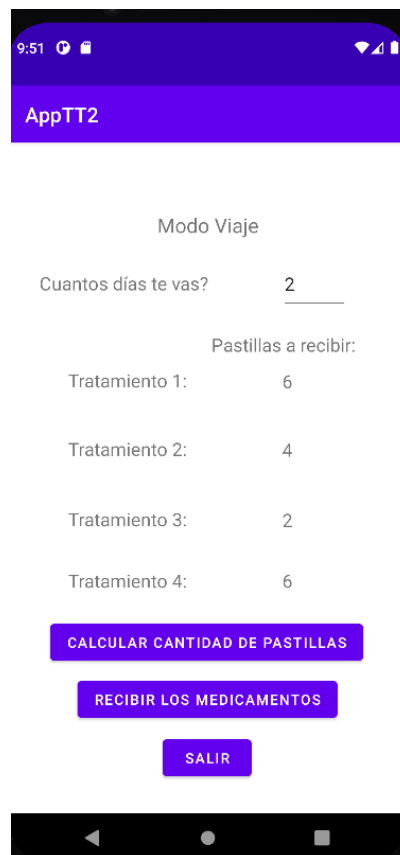


Imagen 256. Ventana de modo viaje de la aplicación, consulta de pastillas necesarias para el viaje.

El usuario también presionó que desea recibir los medicamentos del dispensador por lo que la bandera “ModoViaje” está activada en Firebase como se muestra en la imagen 257.

ModoViaje: 1

Imagen 257. Bandera "ModoViaje".

Como se pudo observar en estas pruebas, la conexión entre la aplicación móvil y la base de datos funciona correctamente, permitiendo modificar los valores de las banderas de control y los datos de los tratamientos en tiempo real.

Modelo final en CAD del prototipo

En la imagen 258, se muestran los resultados obtenidos del diseño final del prototipo, cuentan con una vista de corte en la que se puede apreciar el ensamblaje de las piezas internas del dispositivo que integran los mecanismos de ingreso y dispensado de medicamentos y agua.

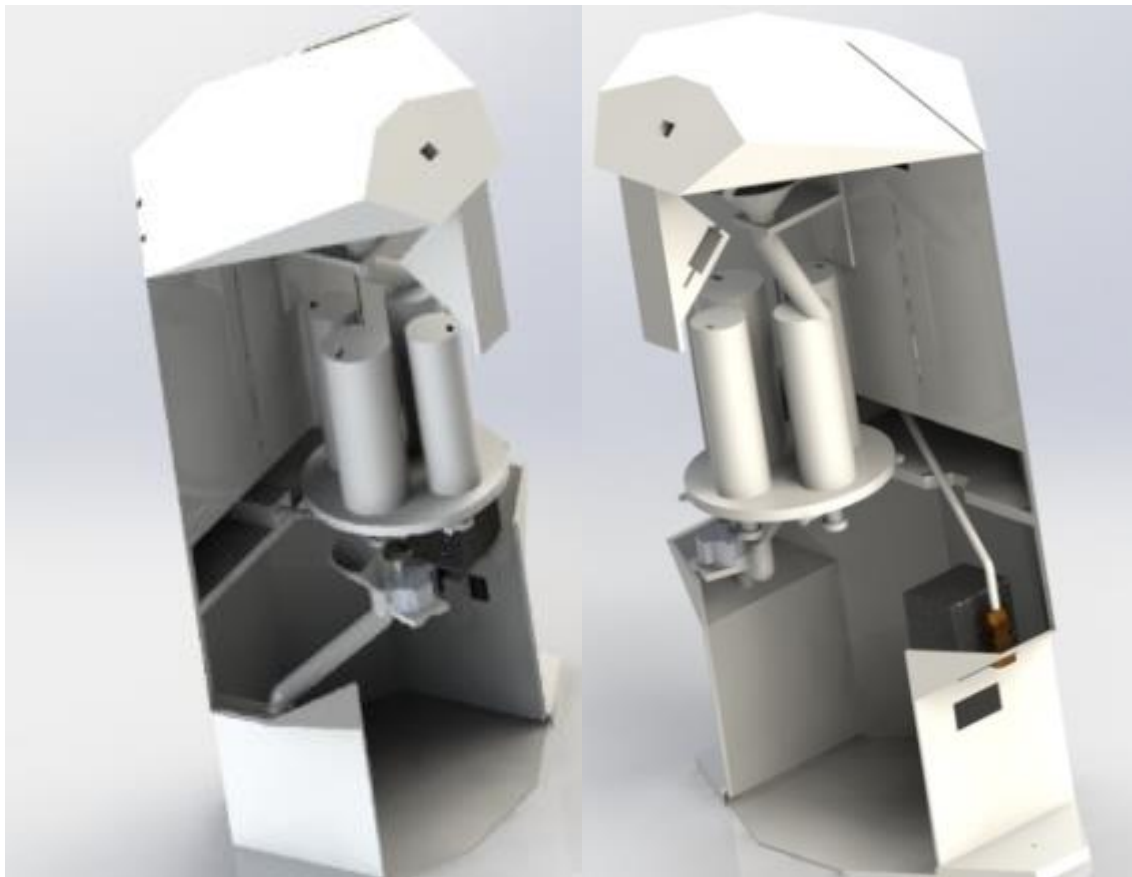


Imagen 258. Diseño final del prototipo de dispensador de comprimidos y cápsulas.

En la imagen 259, que se muestra en la página siguiente, se puede apreciar la vista explosionada del diseño final, en ella también se pueden observar tanto las piezas

diseñadas por el equipo como los principales componentes que se integrarán en el dispositivo.



Imagen 259. Explosionado del diseño del prototipo de dispensador de comprimidos y cápsulas.

Ensamble General del dispositivo

Una vez se tuvieron ensamblados los distintos mecanismos que integran el dispositivo, el equipo procedió a ensamblarlos para tener el prototipo completamente integrado, para ello, primero se realizaron las conexiones de los actuadores a sus etapas de potencia siguiendo el diagrama de conexión visible en la imagen 206. Es preciso recordar que la etapa de potencia diseñada puede y fue empleada para la activación de todos los actuadores salvo por los motores a pasos que cuentan con drivers propios, sus diagramas de conexión pueden consultarse en los apéndices en las páginas 286 y 287. También vale la pena mencionar que, para poder activar las etapas de potencia con la tarjeta de control, se implementan convertidores de nivel, las salidas digitales de la tarjeta pasan por estos convertidores para poder alcanzar los 5V requeridos por el optoacoplador 4N25 ya que la Raspberry no supera los 3.3V de salida.

Además, se conectaron los sensores del dispositivo siguiendo sus respectivos diagramas de conexión visibles en el diseño detallado. Cabe recalcar que los sensores de nivel de agua debieron pasar sus pines digitales de salida/datos por convertidores de nivel pues estos operan a 5V y la tarjeta Raspberry no soporta más de 3.3V de entrada. Otra cuestión importante de recordar es que como fuente se emplea una de tipo ATX que cuenta con cuatro tipos de valores de salida de tensión: 3.3V, 5V, 12V y -12V. Esto resulta de gran utilidad pues con una sola fuente se pueden obtener los diferentes valores de tensión requeridos por los componentes electrónicos implementados.

En la imagen 260 se puede observar el dispensador completamente ensamblado, cuya forma final guarda estrecho parentesco al modelo final obtenido en CAD.

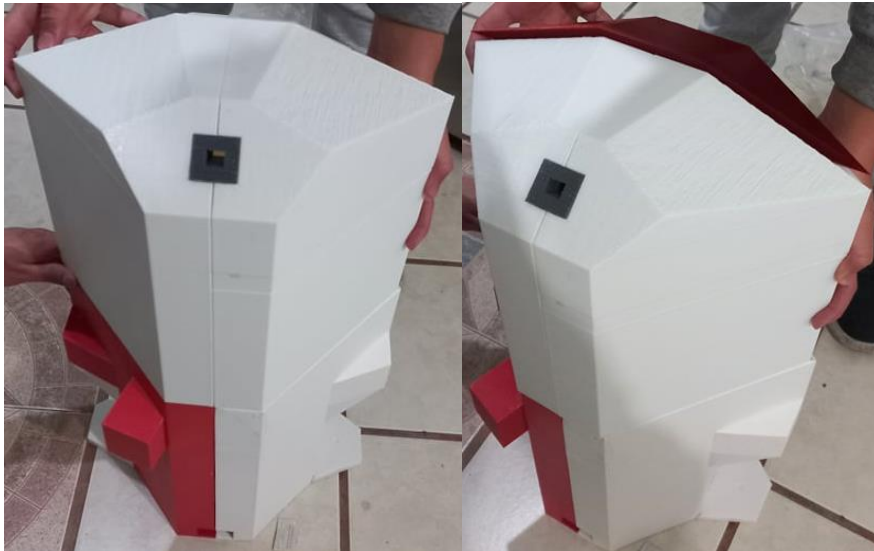


Imagen 260. Ensamble final del dispensador.



7. Análisis de resultados

Objetivo: Construir el mecanismo para la identificación y separación de capsulas y comprimidos en recipientes separados para el almacenamiento de los mismos, así como el set de compartimentos requeridos.

Se logró diseñar y construir el mecanismo acorde a los alcances de proyecto establecidos, sobre todo, gracias a la implementación de la selección del tipo de pastillas a través de la aplicación móvil, pues a partir de esos datos, el mecanismo de almacenamiento de revolver diseñado logra separar en un contenedor adecuado las pastillas que ingresan al dispositivo.

Este mecanismo en especial sufrió cambios considerables respecto del diseño propuesto en TTI, principalmente respecto al método de identificación que paso de ser uno mediante visión artificial a ser uno asistido por el usuario a través de una interfaz gráfica, además, como fue posible observar en su respectivo apartado de actualizaciones de diseño se modificaron características de las piezas del mecanismo para mejorar su cinemática.

Finalmente, las piezas impresas, pese a la rugosidad agregada debido al procesos de manufactura y que no estaba contemplada por el equipo, permiten el correcto deslizamiento hacia los contenedores.

Rendimiento: 100%

Objetivo: Desarrollar una aplicación móvil en un lenguaje de alto nivel para que el usuario pueda configurar cada cuanto tiempo ingiere sus distintos medicamentos, que despliegue notificaciones cada vez que sea tiempo de ingerir la dosis y cuando esta haya sido omitida y, además, funcione como interfaz gráfica para el usuario brindándole información y parámetros de operación del dispensador.

Se logró desarrollar la aplicación móvil en el ambiente de desarrollo de Android Studio implementando el lenguaje de programación Kotlin y el uso de lenguaje XML para diseñar las interfaces de usuario.

Respecto del diseño propuesto en TTI hubo ligeras modificaciones las cuales fueron principalmente en la disposición de las ventanas y sus botones, esto buscando conservar el concepto de diseño. Además, se agregaron nuevas ventanas para volver la interacción más amigable con el usuario.

Además, la implementación de los servicios de Firebase Realtime Database permitieron lograr una conexión y sincronización efectiva con la tarjeta Raspberry ahorrando tiempo de desarrollo pues no se requirió de la configuración de un servidor privado o del desarrollo en backend de una base de datos, pues ambos servicios son provistos por Google.

Rendimiento: 100%



Objetivo: Implementar un sistema de alarma integrado en el dispositivo que avise al usuario que es el momento de la ingesta de un medicamento y que insista hasta en 3 ocasiones para que el usuario tome su medicamento. Así como implementar un esquema de control para regular la temperatura del dispositivo y mantenerla menor a los 30° Celsius.

Referente al sistema de alarma, este fue construido respecto al diseño propuesto en TTI empleando un buzzer para alertar al usuario al momento que un medicamento es dispensado. El algoritmo garantiza que la alarma suena solo cuando el prototipo realiza el dispensado de una pastilla pues es parte de la secuencia de extracción de medicamentos y emplea una secuencia de sonidos que cumple con las especificaciones del objetivo.

Respecto al control de temperatura, este módulo funciona siempre y cuando el prototipo se encuentre en ambientes no extremos, es decir, principalmente en interiores pues el dispositivo está pensado para su uso dentro del hogar. Además, se considera que el ambiente al que se encontrara expuesto tiene las consideraciones de que el dispositivo debe ubicarse fuera del alcance de la luz del sol y en un espacio con bajos índices de humedad relativa.

Por otra parte, bajo las condiciones antes mencionadas, el sistema logra mantener la temperatura del dispensador controlada y bajo los 30°C requeridos, sin embargo, si las condiciones ambientales dentro del hogar comienzan a volverse más extremas, es decir, con temperaturas mayores a los 35°C y humedades relativas superiores al 60%, el módulo pierde efectividad siendo incapaz de lograr mantener la temperatura interna por debajo de la temperatura requerida dado que los ventiladores implementados no tienen la capacidad suficiente para disipar el calor por convección forzada a la velocidad que las condiciones ameritarían.

Rendimiento: 75%

Objetivo: Construir los mecanismos que puedan proveer al usuario de los comprimidos o cápsulas que le corresponde ingerir, así como el agua para tomarla, en recipientes que formarán parte del mismo aparato.

Se diseñaron y construyeron dos mecanismos, uno para la extracción de los medicamentos de los contenedores de almacenamiento y otro para dispensar el agua de un contenedor con capacidad de un litro.

El mecanismo de dispensado de agua funciona por gravedad e implementa una electroválvula diafragma NC la cual es activada por la tarjeta de control del dispositivo, esto permite accionarla cuando es momento de ingerir un medicamento y a partir de un sensor de detección de agua libre de contacto es posible indicar su cierre al alcanzar un nivel adecuado para ingerir la pastilla. Además, la válvula no requiere de una presión mínima de trabajo, por lo que se garantiza su funcionamiento aún con poco líquido en el tanque.



Respecto al mecanismo de dispensado, los actuadores implementados y el mecanismo diseñado cumplen con el objetivo de extraer los medicamentos de los contenedores si existe una pastilla colocada en el orificio de salida de los mismos, sin embargo, se encontraron fallos en el uso del tornillo sinfín, el problema radica en que, debido a que las piezas impresas no tienen superficies lisas, las pastillas no siempre deslizan en conjunto hasta el fondo del contenedor de almacenamiento, por lo que pueden darse situaciones en que no se logre extraer la pastilla. Desafortunadamente, el ángulo en que se rompe el coeficiente de fricción entre la pastilla y el material implementado se obtuvo en base a pruebas experimentales en las que la superficie plástica era lisa. Sin embargo, debido a la rotación del tornillo y su vibración, las pastillas logran descender, aunque no se puede asegurar que siempre estarán disponibles. Cabe aclararse que este detalle resulta ser un problema que es intrínseco del proceso de manufactura y no uno de diseño y que era desconocido por el equipo.

Rendimiento del mecanismo de dispensado de agua: 100%

Rendimiento del mecanismo de extracción de pastillas: 65%

Rendimiento en general: $(100+65) / 2 = 82.5\%$

Objetivo: Desarrollar el programa que permita administrar y gestionar el medicamento para hacerle saber al usuario que le quedan tres dosis y pueda abastecer el dispositivo de forma oportuna.

Se desarrolló el software requerido en el lenguaje de Python, este programa se encuentra alojado en la tarjeta de control y es el encargado tanto de actualizar los parámetros de operación en la base de datos como el número de pastillas restantes disponibles de cada tratamiento. Además, es el encargado de ejecutar los diferentes procesos que puede realizar el prototipo, ya sea dispensar un medicamento, ingresar o reabastecer tratamientos, eliminarlos o entregar las pastillas que el usuario requiere para de uno a tres días de ausencia.

El programa desarrollado logra satisfactoriamente su cometido, en gran medida gracias a la implementación de los servicios de Firebase que permiten conectar con la aplicación móvil y gestionar la base de datos en la nube.

Rendimiento: 100%

Objetivo: Implementar un módulo que permita al usuario llevarse los medicamentos que le toca consumir a lo largo de uno a tres días en un pastillero portátil.

Este módulo está compuesto por la función de modo viaje desarrollada en Python, la ventana de acceso al modo viaje desarrollada en la aplicación móvil que en conjunto permiten al usuario poder extraer del dispositivo las pastillas que necesitará durante su periodo de ausencia.

Por otro lado, se diseñó y construyó un pequeño pastillero portátil para que el usuario pueda llevar consigo las pastillas que necesita, el cual, consta con compartimentos de



almacenamiento y una tapa para evitar que contaminantes entren en contacto con sus medicinas.

Su eficacia se ve reducida debido a los detalles mencionados respecto al sistema de extracción de medicamentos, sin embargo, fuera de dichos aspectos, el módulo se ejecuta correctamente.

Rendimiento: 75%

8. Costos

En esta sección se muestra el costo de este proyecto desglosado en 2 rublos:

- **Materiales:** Son los costos de los diferentes materiales que se utilizaron para la implementación del proyecto, ya sean componentes electrónicos, hidráulicos o herramientas.
- **Manufactura:** Se presentan cada uno de los costos por pieza impresa en 3D.

A continuación, se muestra la información correspondiente a cada rublo mencionado en las siguientes tablas (ver tablas 32 a 33) y al final de la sección se observa el resumen de cada área y el total del costo del Prototipo de dispensador de comprimidos y cápsulas medicinales configurable y administrable desde una aplicación móvil (tabla 34).

Nombre	Descripción	Costo unitario (MXN)	Cantidad de piezas	Costo total (MXN)
Solenoid Push-Push 5V	Actuador lineal eléctrico	\$140.00	4	\$560.00
Raspberry Pi3 b+	Unidad de control y procesamiento de datos	\$1,040.00	1	\$1,040.00
IFRZ44N	Transistor MOSFET	\$18.00	8	\$144.00
4N25	Optoacoplador	\$7.00	8	\$56.00
HW-221	Convertidor de niveles bidireccional	\$46.00	2	\$92.00
FR307	Diodo de recuperación rápida	\$7.00	8	\$56.00
Capacitor 100nF	Capacitor cerámico	\$3.00	8	\$24.00
DHT11	Sensor de temperatura y humedad	\$60.00	1	\$60.00
Buzzer	Buzzer mini	\$14.00	1	\$14.00
PM55L-048	Motor a pasos bipolar	\$70.00	1	\$70.00
28BYJ-48	Motor a pasos unipolar	\$55.00	1	\$55.00
A4988	Driver para el motor a pasos bipolar	\$45.00	1	\$45.00
UNL2003	Driver para el motor a pasos unipolar	\$40.00	1	\$40.00
XKC-Y25	Sensor de nivel agua sin contacto	\$202.00	2	\$404.00
ACTECK INTEGRA 5R500	Fuente de alimentación	\$429.00	1	\$429.00
Electroválvula	2/2 N. C. para agua 1/4"	\$520.00	1	\$520.00
Placas fenólicas	5 cm x 10 cm	\$9.00	4	\$36.00
			TOTAL	\$3,645.00

Tabla 32. Costos de los materiales usados para el prototipo.



Nombre de la pieza	Cantidad	Densidad	Costo (MXN)
Compuerta1_Ultimate	1	30%	\$3.00
Compuerta2_Ultimate	1	30%	\$3.00
Compuerta3_Ultimate	2	30%	\$6.00
ContenedorAgua_Ultimate	1	50%	\$648.00
CubiertaDeTornilo1_Ultimate	1	40%	\$9.00
CubiertaDeTornilo2_Ultimate	1	40%	\$9.00
CubiertaDeTornilo3_Ultimate	2	40%	\$18.00
EntradaEmbudo_Ultimate	1	50%	\$64.50
Reductor1_Ultimate	1	80%	\$9.00
RevolverUltimateC	1	80%	\$373.50
Tapa1_UltimateC	1	80%	\$103.50
Tapa2_UltimateC	1	80%	\$79.50
Tapa3_UltimateC	2	80%	\$150.00
Tornillo1_Ultimate	1	80%	\$135.00
Tornillo2_Ultimate	1	80%	\$123.00
Tornillo3_Ultimate	2	80%	\$234.00
VasoAgua_Ultimate	1	20%	\$105.00
VasoPastilla_Ultimate	1	20%	\$90.00
TapaPastilleroPortatil	1	20%	\$54.00
TapaSuperior_Ultimate	1	15%	\$312.00
TransmisorRevolver	1	30%	\$27.00
Cajon_Ultimate	1	20%	\$22.50
CarcasaSecundaria_Ultimate2	1	15%	\$1,021.50
PastilleroPortatil_UltimateF	1	15%	\$135.00
RampaPastilla_Ultimate	1	15%	\$75.00
SoporteEmbudoS_Ultimate2	1	15%	\$75.00
AscensorPared	2	15%	\$83.22
AscensorParedContenedor	1	15%	\$45.44
AscensorPlataforma	1	15%	\$20.23
Riel	1	15%	\$32.52
SalidaEmbudo	1	15%	\$111.70
SoporteEmbudo	1	15%	\$24.52
Transmisor	1	15%	\$19.45
CarcasaEntrada_Ultimate1	1	25%	\$358.00
CarcasaEntrada_Ultimate2	1	25%	\$358.00
CarcasaPrincipal_Ultimate1	1	25%	\$358.00
CarcasaPrincipal_Ultimate2	1	25%	\$358.00
CarcasaPrincipal_Ultimate3	1	25%	\$358.00
CarcasaPrincipal_Ultimate4	1	25%	\$358.00
CarcasaSecundaria_Ultimate1	1	25%	\$358.00
		TOTAL	\$6,728.08

Tabla 33. Costo de manufactura de las piezas del prototipo.

Área	Costo (MXN)
Materiales	\$3,645.00
Manufactura	\$6,728.08
TOTAL	\$10,373.08

Tabla 34. Costo final del prototipo.

9. Cronogramas

El cronograma de actividades propuesto para las unidades de aprendizaje de Trabajo Terminal I (imagen 261) y Trabajo Terminal II (imagen 262) durante los periodos escolares 2021/2 y 2022/1 respectivamente son los siguientes:

Item	Actividad	Responsable	Semanas																	
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	Analisis de requerimientos																			
1.1	Análisis de Requerimientos	A1 ,A2, A3	█																	
1.2	Especificaciones de Diseño	A1 ,A2, A3	█																	
2	Diseño conceptual																			
2.1	Parametros y Atributos de Evaluación para Diseños conceptuales	A1 ,A2, A3	█																	
2.2	Matriz Morfológica de Diseño Conceptual	A1 ,A2, A3		█																
2.3	Selección de Rutas de la Matriz Morfológica	A1 ,A2, A3		█																
2.4	Evaluación de Rutas Seleccionadas usando Parametros Propuestos	A1 ,A2, A3		█																
2.5	Boceto general (Concepto)	A3		█																
3	Diseño detallado																			
3.1	Selección de componentes para todos los sistemas que lo requieran																	█		
3.2	Diseño compartimentos	A1,A2,A3			█															
3.3	Diseño de los sistemas de dosificación	A1,A2,A3			█															
3.3.1	<i>Diseño del sistema de dosificación de agua</i>				█															
3.3.2	<i>Diseño del sistema de dosificación de medicamentos</i>				█															
3.3	Diseño del sistema de energía	A1,A2,A3				█														
3.4	Diseño del sistema de control de temperatura	A1,A2,A3					█													
3.5	Diseño del sistema de alarma	A1,A2,A3						█												
3.6	Diseño de algoritmos de control y procesamiento de datos	A1,A2,A3							█											
3.7	Diseño de interfaz de usuario	A1,A2,A3								█										
3.8	Selección de materiales comerciales	A1,A2,A3									█									
4	Validacion																			
4.1	Validacion Mecánica de los Elementos Diseñados	A1,A2,A3																	█	
4.2	Validacion Electronica de los Circuitos Diseñados	A1,A2,A3																	█	
5	Ensamble de Piezas en Software																			
5.1	Ensamble General	A1																	█	
5.2	Ensamble Explosionado	A2																	█	
5.3	Diagrama de Ensamble	A3																	█	

Imagen 261. Cronograma de actividades en Trabajo Terminal I.



10. Conclusiones

Recapitulando los análisis de resultados, el equipo considera que el funcionamiento de los diversos módulos que constituyen al dispositivo son adecuados tanto de forma separada como integrada, sin embargo, existen problemas en el mecanismo de extracción de medicamentos siendo este el único con fallos intermitentes debidos a la rugosidad de la superficie de los tonillos sinfín impresos que provocan que las pastillas no deslicen adecuadamente y, por consiguiente, puedan existir situaciones en que las pastillas no estén ordenadas adecuadamente dentro del contenedor y se presenten ocasiones en las que no se acomode la pastilla en el orificio de salida cuando este se libera para que otra esté lista para ser dispensada. Es un fallo que impide garantizar que se dispensará el medicamento el 100% de las veces, sin embargo, este problema no radica en el diseño del mecanismo sino en el proceso de manufactura implementado ya que es debido a este que la superficie de la cuerda del tornillo no es lisa y recae en la calidad de impresión y precisión de la máquina del proveedor del servicio de impresión, desafortunadamente el equipo no cuenta con los recursos económicos para acceder a un servicio de impresión de calidad industrial o un proceso de manufactura más robusto como lo pueden ser la inyección o el extruido de plástico pues además de ser especializados no se ponen a disposición de un cliente que requiere piezas unitarias.

En el caso del módulo de control de temperatura, aunque su capacidad de mantener bajo la temperatura requerida el sistema se reduce en temperaturas ambientales severas (temperaturas superiores a los 35°), bajo temperaturas ambientales normales (temperaturas de hasta 32°C) funciona sin ningún problema y dado que el prototipo está diseñado para emplearse en interiores y ser ubicado en espacios alejados de la luz del sol y por consiguiente frescos y secos, se considera que su funcionamiento es aceptable para los fines académicos de este proyecto. Además, gracias al contenedor de material desecante implementado en el dispositivo, es posible mantener la humedad relativa interna del dispositivo en valores dentro del rango aceptado el cual va del 0% hasta un 65% de acuerdo a las normas oficiales de almacenamiento de medicamentos

El resto de módulos construidos y diseñados por el equipo funcionan adecuadamente sin ningún inconveniente.

Considerando lo anterior y realizando un promedio de los rendimientos logrados en los objetivos específicos se obtiene que se logra satisfacer en un 88.75% el objetivo general del proyecto.



Trabajo a futuro

Se considera que, como principal trabajo a futuro, debe buscarse un proceso de manufactura que permita obtener las piezas que conforman al mecanismo de extracción de medicamentos con superficies lisas, pues el problema de la rugosidad en la cuerda de los tornillos de dicho mecanismo representó el mayor causante de fallos en el dispositivo.

Implementar ventiladores con un mayor flujo de aire como los inicialmente propuestos por el quipo en TTI y a los que no se pudo acceder debido a falta de existencias del proveedor, en lugar de los actualmente instalados permitirá tener una mayor capacidad de enfriamiento incluso ante temperaturas adversas. Por otra parte, el cálculo de flujo de aire necesario para el enfriamiento del volumen interno podría realizarse por medio de cálculos termodinámicos que consideren los coeficientes de transferencia de calor del material para tener un valor más exacto, aunque el cálculo empleado actualmente da una aproximación adecuada sobre todo teniendo en cuenta que el volumen a ventilar es de tan solo un pie cúbico.

El cálculo de material desecante para mantener bajos niveles de humedad el dispositivo dentro del rango adecuado es una aproximación, sin embargo, recurrir a un método más robusto que considere las propiedades termodinámicas específicas del sistema permitiría mantenerlo seco durante mayor periodo de tiempo.

El orificio de entrada de medicamentos, así como los orificios de salida de aire del prototipo, aunque son pequeños, exponen a los contenedores de medicamento a la humedad y polvo, por lo que agregarles una tapa y filtros respectivamente permitiría aislar de mejor forma el interior del dispositivo.

Mejorar la estética de las interfaces de usuario de la aplicación móvil para tener un aspecto más llamativo e intuitivo que, a su vez, permita que un mayor número y tipo de usuarios puedan manipular el dispositivo.



II. Referencias

- [1]- M. Piña-Pozas, “Hipertensión arterial un problema de salud pública en México” Julio 2020. [En línea]. Disponible en: <https://www.insp.mx/avisos/5398-hipertension-arterial-problema-salud-publica.html>
- [2]- J. Dávila, J. Gonzales & A. Barrera, “Panorama de la obesidad en México”. Noviembre 2014. [En línea]. Disponible en: http://revistamedica.imss.gob.mx/editorial/index.php/revista_medica/article/viewFile/21/54
- [3]- A. Andrade, “MÉXICO OCUPA EL 90 LUGAR MUNDIAL EN DIABETES”. Mayo 2019. [En línea]. Disponible en: https://www.dgcs.unam.mx/boletin/bdboletin/2019_305.html
- [4]- ABCSALUD. “Uno de cada cuatro pacientes olvida tomar su medicación”. Diciembre 2013. [En línea]. Disponible en: <https://www.abc.es/salud/noticias/20131115/abci-cada-cuatro-pacientes-olvida-201311291109.html?ref=https://www.google.com>
- [5]- OMS. “Resistencia a los antibióticos”. Julio 2020. [En línea]. Disponible en: <https://www.who.int/es/news-room/fact-sheets/detail/resistencia-a-los-antibi%C3%B3ticos>
- [6]- L. Garbayo, “¿Qué es la resistencia bacteriana y cómo afecta a nuestra salud?”. Enero 2017. [En línea]. Disponible en: <https://www.ceac.es/blog/que-es-la-resistencia-bacteriana-y-como-afecta-nuestra-salud>
- [7]- Sicelo. “Resistencia bacteriana: un problema de salud pública mundial de difícil solución”. Abril 2016. [En línea]. Disponible en: http://scielo.iics.una.py/scielo.php?script=sci_arttext&pid=S1812-95282016000100001
- [8]- J. Ortega, D. Sánchez, O. Rodríguez & J. Ortega, “Adherencia terapéutica: un problema de atención médica”. Septiembre 2018. [En línea]. Disponible en: [http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-72032018000300226#:text=La%20Organizaci%C3%B3n%20Mundial%20de%20la%20Salud%20\(OOMS\)%20define%20E2%80%9Cadherencia,un%20prestador%20de%20asistencia%20sanitaria%20E2%80%9D](http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-72032018000300226#:text=La%20Organizaci%C3%B3n%20Mundial%20de%20la%20Salud%20(OOMS)%20define%20E2%80%9Cadherencia,un%20prestador%20de%20asistencia%20sanitaria%20E2%80%9D)
- [9]- V. Ferrandis Tébar, “Formas farmacéuticas y vías de administración”. Mayo 2013. [En línea]. Disponible en: https://saludyejercicio.co/wp-content/uploads/2018/02/Form_farmacuticas.pdf
- [10]- códigof. La revista de la CANIFARMA. “Estándares para la conservación de productos farmacéuticos”. Junio 2019. [En línea]. Disponible en: <https://codigof.mx/estandares-para-la-conservacion-de-productos-farmacuticos/?fbclid=IwAR3D8L8w4OU4OOMAIZkQZCXVExpGpnoMBLUiu00OL7A4mJlrZsSureHXG90>
- [11]- Medecins Sans Frontieres. “Calidad y conservación de los medicamentos”. Noviembre 2020. [En línea]. Disponible en: https://medicalguidelines.msf.org/viewport/EssDr/spanish/calidad-y-conservacion-de-los-medicamentos-22287420.html?fbclid=IwAR24SZGhxaxTM_yk7LWZUO0AUsd4wjQM46nCrIfTrirHRwBU0NTkHS3p_Ms
- [12]- S&P. El blog de la ventilación eficiente. “Humedad relativa, específica y absoluta [Carta Psicométrica]”. Abril 2018. [En línea]. Disponible en: <https://www.solerpalau.com/es-es/blog/humedad-relativa-especifica-absoluta/>
- [13]- M. C. Soriano & C. Sánchez Lafuente, “Acondicionamiento de medicamentos: funciones y tipo de envasado”. Junio 2000. [En línea]. Disponible en: <http://dica.minec.gob.sv/inventa/attachments/article/2581/11articulos.pdf>
- [14]- I. Rodríguez Tejonero, “Material de acondicionamiento primario de medicamentos”. Marzo 2014. [En línea]. Disponible en: https://eprints.ucm.es/28811/1/T35869.pdf?fbclid=IwAR0MwAEsoHF5ntq-X3gNAAD_g0FFo8Sqsuc3c6V0CLG4GZ0d83d7yAZT4
- [15]- J. Salazar, “Redes Inalámbricas”, Noviembre 2020. [En línea]. Disponible en: https://upcommons.upc.edu/bitstream/handle/2117/100918/LM01_R_ES.pdf



- [16]- Bibing. “Teoría de las clasificaciones”. Noviembre 2020. [En línea]. Recuperado de: http://bibing.us.es/proyectos/abreproy/70448/fichero/05_Capitulo4.pdf?fbclid=IwAR3szq06NYU_0_lTatLz0_kGhFqBEoo5PtvMgR6vhfvf3LYFSAhKPhSdk4
- [17]- F. Hernandez. (2001, Dic). Notas galénicas: comprimido (Vol. 2, n.º 6.). [En línea]. Disponible en: https://www.tremedica.org/wp-content/uploads/n6_NotasGalenicas.pdf?fbclid=IwAR0rq0_ZDc3n5Jy4w0P5QciBMpA7nTbpZtkoPWy5JEsjTER2n3WLMO07c50
- [18]- U.S. Department of Health and Human Services Food and Drug Administration Center for Drug Evaluation and Research (CDER). (2015, June). Size, Shape, and Other Physical Attributes of Generic Tablets and Capsules. [En línea]. Disponible en: https://www.fda.gov/files/drugs/published/Size--Shape--and-Other-Physical-Attributes-of-Generic-Tablets-and-Capsules.pdf?fbclid=IwAR0F9eUIGQBc6c5Xunf2M5nfR56yAHgvWemrOa8IGUf_lrypaEePleNhK4s
- [19]- bega helicoidales. “Transportador Helicoidal Vertical (Elevador Helicoidal)/Tornillo Helicoidal Vertical/Gusano Helicoidal Vertical”. Marzo 2021. [En línea]. Disponible en: <https://www.begahelicoidales.com/transportador-helicoidal-vertical>
- [20]- M. E. Fayed. Mechanical Conveyors: Selection and Operation. Western Hemisphere: Technomic Publishing Company, Inc, 1997.
- [21]- L. Dan. (2021, Mar). Sección H: Transportadores helicoidales. [En línea]. Disponible en: https://www.academia.edu/?fbclid=IwAR1TfDaes-ZmMJ_VtbjKOR7pkGnFvvVYhcc3dE_ib-jwmqzK3Spwv-sVXs8
- [22]- Flujo debido a una carga descendente”, apuntes de clase para M212, Departamento de mecánica, Instituto Politécnico Nacional en la Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, primera 2019.
- [23]- TECCINESS. “Coeficiente de resistencia K”. Abril 2021. [En línea]. Disponible en: https://www.es.tecciness.com/hilfe/druckverlustbeiwert_zeta.php
- [24]- «L. Vélez» [En línea]. Disponible en: <https://makinandovelez.wordpress.com/2018/02/15/sistemas-de-control-de-lazo-cerrado/>. [Último acceso: 15 02 2018].
- [25]- «K. Ogata» [En Línea]. Disponible en: https://cursos.cl/usuario/78303fe04da8e4eb340eae09f1840b2/mi_blog/r/Ingenieria_de_Control_Moder_na_Ogata_5a_ed.pdf. [Último acceso: 2010].
- [26]- «E. Ávila Perona» “Sistemas de Control”. [En línea]. Disponible en: <http://dea.unsj.edu.ar/ceal/Sistemas.pdf?fbclid=IwAR0YTqUfwu8vO7LcelXT4Kx9f0ISvjLbslmtUrHtStHaHBim2XuuAGqGhS0>.
- [27]- «Autmix Industrial Group» [En línea]. Disponible en: <https://pragma-a.com/controlador-pid/>.
- [28]- IBM. (S/A). ¿Qué es una Base de Datos en la Nube?. noviembre 28, 2021, de International Business Machines Corporation Sitio web: <https://www.ibm.com/mx-es/cloud/learn/what-is-cloud-database>
- [29]- Google. (S/A). Firebase Realtime Database. noviembre 28, 2021, de Google Sitio web: <https://firebase.google.com/docs/database?authuser=0>
- [30]- AMG Container. ((S/A)). Seleccionar el Desecante Adecuado. noviembre 28, 2021, de AMG Container Sitio web: <https://www.agmcontainer.com/spanish/seleccionar-el-desecante/>
- [31]- Prado J.. (2017). Sales Desecantes, Capacidad de Absorción y Normalización. noviembre 28, 2021, de Encaja, Embalajes y Trading Sitio web: <https://blog.cajaeco.com/sales-desecantes-deshidratantes-envios-maritimos-embalaje-contenedor/>
- [31]- International Council on System Engineering. Modelo Vee de Diseño e Integración de Ingeniería de Sistemas. (2018, Sep). [En línea]. Disponible en: https://web.archive.org/web/20070927005332/http://g2sebok.incose.org/app/qualsys/view_by_id.cfm?ID=INCOSE%20G2SEBOK%203.30&ST=F



- [32]- Firesmith, Donald. "Uso de modelo V para pruebas". (2013, Nov). Instituto de ingeniería de software. [En línea]. Disponible en: <https://insights.sei.cmu.edu/blog/using-v-models-for-testing/>
- [33]- Enfoque UTE. "Diseño de un sistema inteligente y compacto de iluminación". (2018, Jan). Universidad UTE. [En línea]. Disponible en: http://ingenieria.ute.edu.ec/enfoqueute/public/journals/1/html_v9n1/art020.html
- [34]- «genebre» [En línea]. Disponible en: <http://www.genebre.es/electrovalvula-de-accion-directa-2-vias-nc>.
- [35]- «mercado libre» [En línea]. Disponible en: https://articulo.mercadolibre.com.mx/MLM-727506884-xkc-y25-sensor-nivel-liquido-agua-sin-contacto-5-24v-_JM?matt_tool=48904454&matt_word=&matt_source=google&matt_campaign_id=11714912137&matt_ad_group_id=113017550359&matt_match_type=&matt_network=g&matt_device=c&matt_creative=482511924687&matt_keyword=&matt_ad_position=&matt_ad_type=pla&matt_merchant_id=117915919&matt_product_id=MLM727506884&matt_product_partition_id=353037831509&matt_target_id=pla-353037831509&gclid=Cj0KCQjwnueFBhChARIsAPu3YkT2tZCzdcRXI-8dT-YY9Dbbmm_ddw5XZFUridlOXqPq3N9EWGVuY-YaAkg8EALw_wcB. [Último acceso: 26 05 2021].
- [36]- «mercado libre» [En línea]. Disponible en: https://articulo.mercadolibre.com.mx/MLM-814367329-sensor-de-nivel-de-agua-sin-contacto-_JM?matt_tool=82503379&matt_word=&matt_source=google&matt_campaign_id=11767285381&matt_ad_group_id=114578050056&matt_match_type=&matt_network=g&matt_device=c&matt_creative=483756468691&matt_keyword=&matt_ad_position=&matt_ad_type=pla&matt_merchant_id=275874715&matt_product_id=MLM814367329&matt_product_partition_id=420985601641&matt_target_id=pla-420985601641&gclid=Cj0KCQjwnueFBhChARIsAPu3YkRNcTyjU3xtr0edC7vREdiyJOHl5fQNNvLxFkcrkMpuYTKzPavUDyIaAn7mEALw_wcB. [Último acceso: 01 04 2021].
- [37]- «mercado libre» [En línea]. Disponible en: https://articulo.mercadolibre.com.mx/MLM-766505537-sensor-de-nivel-de-liquidos-sin-contacto-tipo-switch-_JM?matt_tool=48904454&matt_word=&matt_source=google&matt_campaign_id=11714912137&matt_ad_group_id=113017550359&matt_match_type=&matt_network=g&matt_device=c&matt_creative=482511924687&matt_keyword=&matt_ad_position=&matt_ad_type=pla&matt_merchant_id=114074559&matt_product_id=MLM766505537&matt_product_partition_id=353037831509&matt_target_id=pla-353037831509&gclid=Cj0KCQjwnueFBhChARIsAPu3YkSC_Yjo-5ad5SOTBgyYNDCzudAC3V5t4oTns9Ed7tkCCcueDyKT6loaAv_UEALw_wcB. [Último acceso: 03 11 2020].
- [38]- «mouser electronics» [En línea]. Disponible en: <https://www.mouser.mx/datasheet/2/272/step-motor-5v-28byj48-datasheet-1864397.pdf?fbclid=IwAR2UsPiRQsrh0kiJkdyzm6aDHI2E2xYlzlF54tfyEqwqvQ2YcrBVo7s9N8>.
- [39]- «eminebea» [En línea]. Disponible en: https://www.eminebea.com/en/product/rotary/steppingmotor/pm/standard/pm55l-048.shtml?fbclid=IwAR2zgEARnPz9y26mMGMY8kwps4dz4la5DdKDRoscpSFG8-dbRUYT_I0ghCA.
- [40]- «components101» [En línea]. Disponible en: <https://components101.com/motors/nema17-stepper-motor>. [Último acceso: 19 08 2019].
- [41]- «amazon» [En línea]. Disponible en: <https://www.amazon.com/-/es/Arducam-peque%C3%B1o-megapixeles-pixeles-Raspberry/dp/B012VIHEP4>. [Último acceso: 12 12 2014].
- [42]- «mercado libre» [En línea]. Disponible en: https://articulo.mercadolibre.com.mx/MLM-695624536-solenoid-push-pull-12v-robot-arduino-carrera-10mm-_JM#position=1&search_layout=grid&type=item&tracking_id=1f4c5504-62ef-4e04-a4bc-5d42a0229bcd. [Último acceso: 29 04 2021].



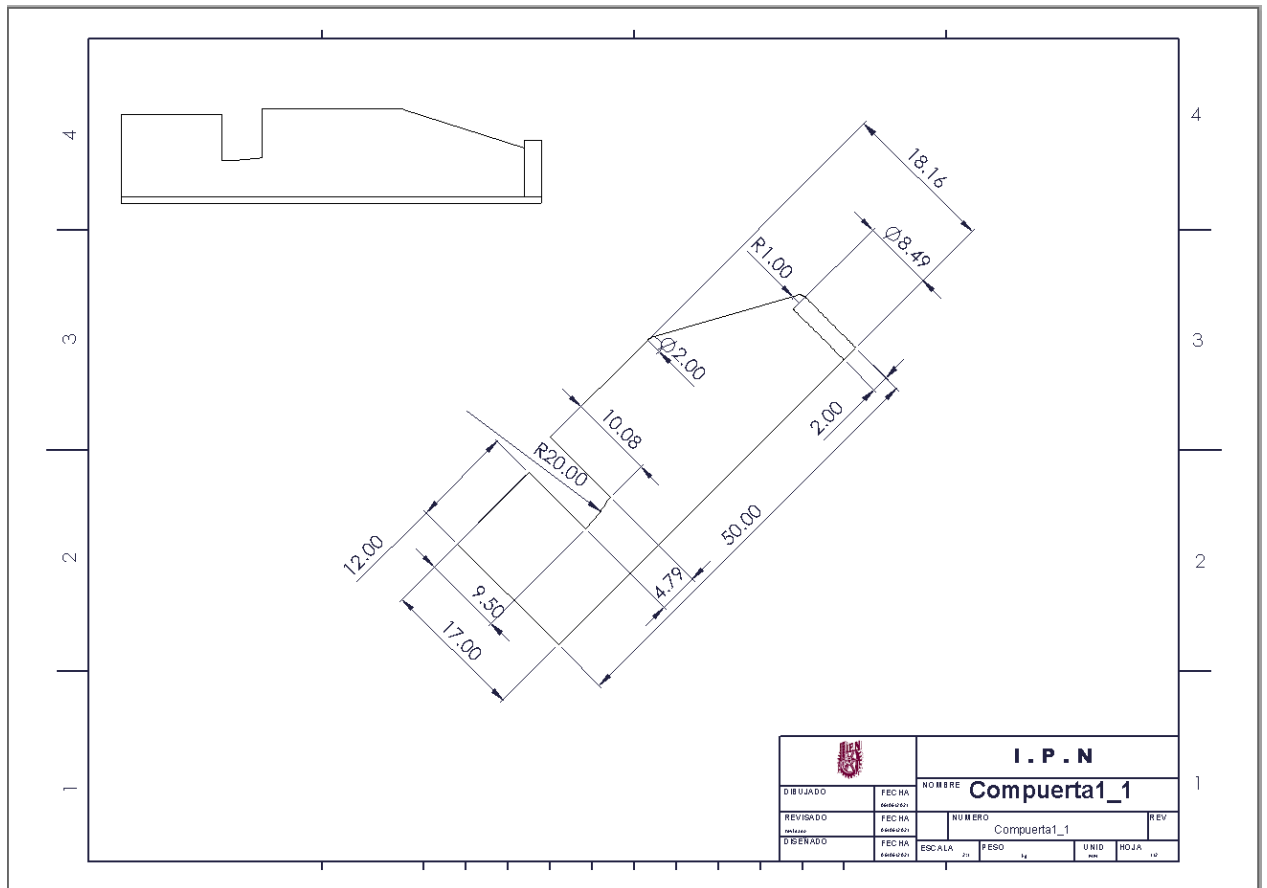
- [43]- «mercado libre» [En línea]. Disponible en: https://articulo.mercadolibre.com.mx/MLM-776147079-solenoid-5v-push-pull-electroiman-actuador-lineal-arduino-_JM#position=1&search_layout=grid&type=item&tracking_id=4e77a26f-7fad-40bd-a0df-cc0e374e19f6. [Último acceso: 14 04 2021].
- [44]- «amazon» [En línea]. Disponible en: https://www.amazon.com.mx/Fielect-electroim%C3%A1n-solenoid-movimiento-JF-1264-12/dp/B07Z3RRDX4/ref=asc_df_B07Z3RRDX4/?tag=gledskshopmx-20&linkCode=df0&hvadid=450930989940&hvpos=&hvnetw=g&hvrnd=7758051746446518385&hvpon=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1010204&hvtargid=pla-1214218293541&psc=1. [Último acceso: 10 05 2021].
- [45]- «mercado libre» [En línea]. Disponible en: https://articulo.mercadolibre.com.mx/MLM-775423882-ventilador-12cm-cooler-master-silent-fan-r4-s2s-124k-gp-_JM?matt_tool=49451585&matt_word=&matt_source=google&matt_campaign_id=11714883592&matt_ad_group_id=113337169826&matt_match_type=&matt_network=g&matt_device=c&matt_creative=482511935163&matt_keyword=&matt_ad_position=&matt_ad_type=pla&matt_merchant_id=255140501&matt_product_id=MLM775423882&matt_product_partition_id=422742614611&matt_target_id=pla-422742614611&gclid=Cj0KCQjw--GFBhDeARIsACH_kdaR_STOSM2VuYjT6WxbJPseF85HvqTOduU6ifgpx7Ij3BLPOMdT9dp0aAsF3EALw_wcB. [Último acceso: 20 04 2021].
- [46]- «amazon» [En línea]. Disponible en: <https://www.amazon.com.mx/WINSINN-140mm-USB-refrigeraci%C3%B3n-Enfriadores/dp/B07J9NKSTR>. [Último acceso: 05 07 2019].
- [47]- «amazon» [En línea]. Disponible en: https://www.amazon.com.mx/ARCTIC-P12-Silent-Pressure-optimised-Extra/dp/B07GSTMRFB/ref=asc_df_B07GSTMRFB/?tag=gledskshopmx-20&linkCode=df0&hvadid=389543661254&hvpos=&hvnetw=g&hvrnd=6446106321381794915&hvpon=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1010029&hvtargid=pla-634402018738&psc=1. [Último acceso: 21 02 2021].
- [48]- «mercado libre» [En línea]. Disponible en: https://articulo.mercadolibre.com.mx/MLM-897816434-sensor-de-temperatura-y-humedad-dht11-arduino-pic-_JM?matt_tool=48904454&matt_word=&matt_source=google&matt_campaign_id=11714912137&matt_ad_group_id=113017550359&matt_match_type=&matt_network=g&matt_device=c&matt_creative=482511924687&matt_keyword=&matt_ad_position=&matt_ad_type=pla&matt_merchant_id=181740647&matt_product_id=MLM897816434&matt_product_partition_id=353037831509&matt_target_id=pla-353037831509&gclid=Cj0KCQjw--GFBhDeARIsACH_kdb6KyUnOdcEDlOPax4XAqGPI_O1FRj4W6gZAqgVcDZVBMFFW1vPYaQaAmcGEALw_wcB.
- [49]- «Components101» [En línea]. Disponible en: <https://components101.com/sensors/dht22-pinout-specs-datasheet>. [Último acceso: 19 04 2018].
- [50]- «mercado libre» [En línea]. Disponible en: https://articulo.mercadolibre.com.mx/MLM-754130622-amt1001-sensor-de-humedad-y-temperatura-modulo-arduino-_JM#position=6&search_layout=grid&type=item&tracking_id=a05d6431-696d-4f5f-9a54-269e5708e037. [Último acceso: 29 10 2020].
- [51]- «330ohms» [En línea]. Disponible en: <https://www.330ohms.com/products/raspberry-pi-3-modelo-b?src=raspberrypi>. [Último acceso: 23 04 2021].
- [52]- «amazon» [En línea]. Disponible en: <https://www.amazon.com.mx/XYGStudy-Desarrollo-NUCLEO-F446RE-STM32F446RET6-Programador/dp/B07ZYCWMLM>. [Último acceso: 11 11 2019].
- [53]- «store arduino» [En línea]. Disponible en: <https://store.arduino.cc/usa/mega-2560-r3>.
- [54]- L. Llamas. “REPRODUCIR SONIDOS CON ARDUINO Y UN BUZZER PASIVO O ALTAVOZ”. Julio 2016. [En línea]. Disponible en: <https://www.luisllamas.es/reproducir-sonidos-arduino-buzzer-pasivo-altavoz/>
- [55]- «Geek Factory» [En línea]. Disponible en: <https://www.geekfactory.mx/tienda/bocinas-zumbadores-y-avisadores-acusticos/mini-buzzer-zumbador-activo-5-v/>.

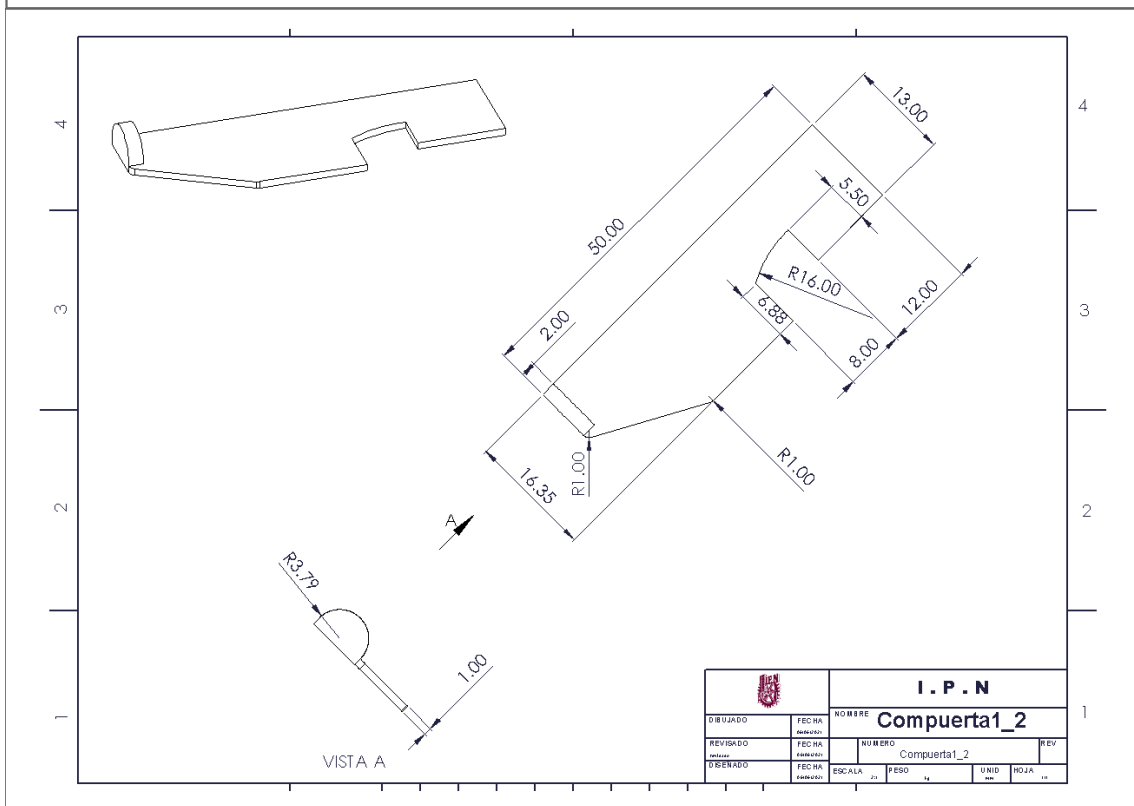
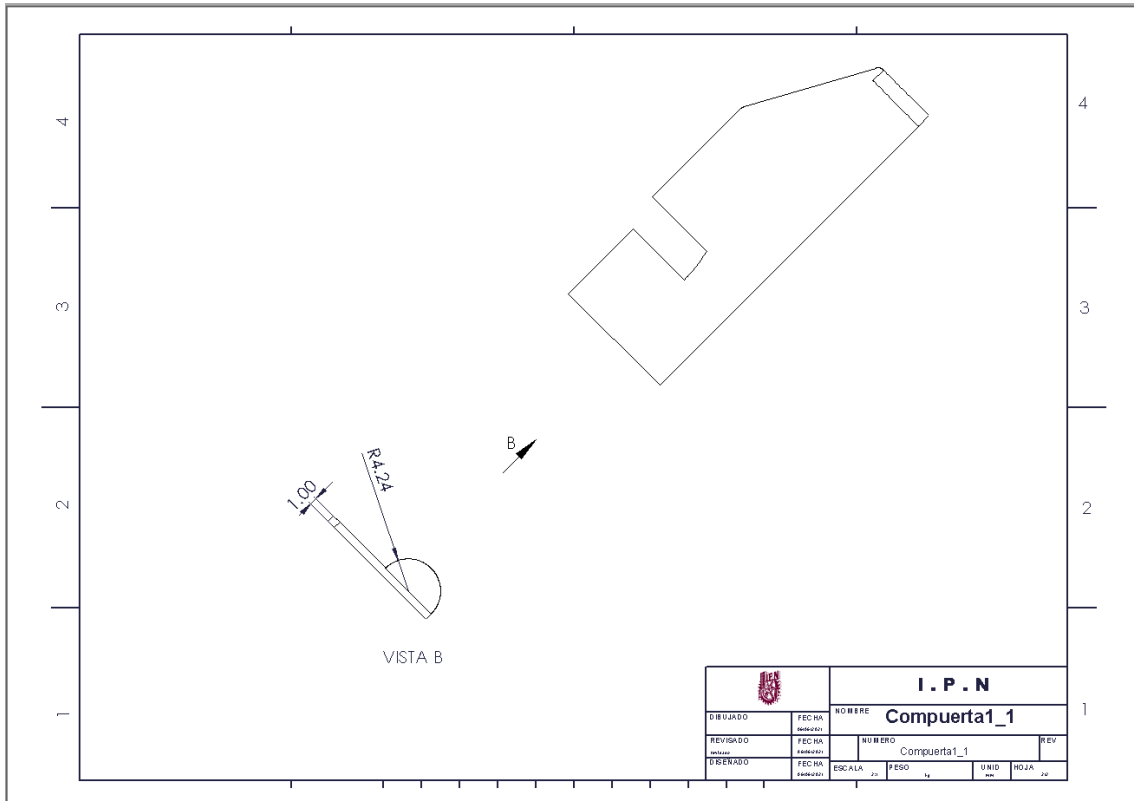


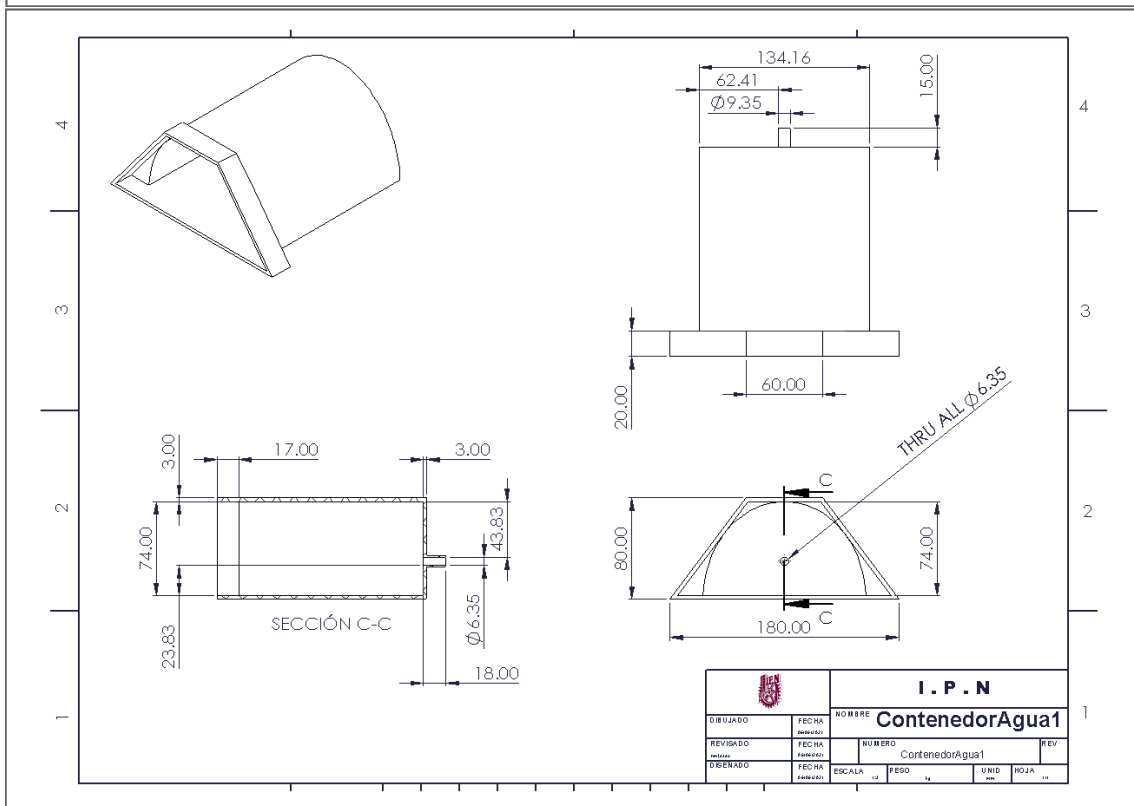
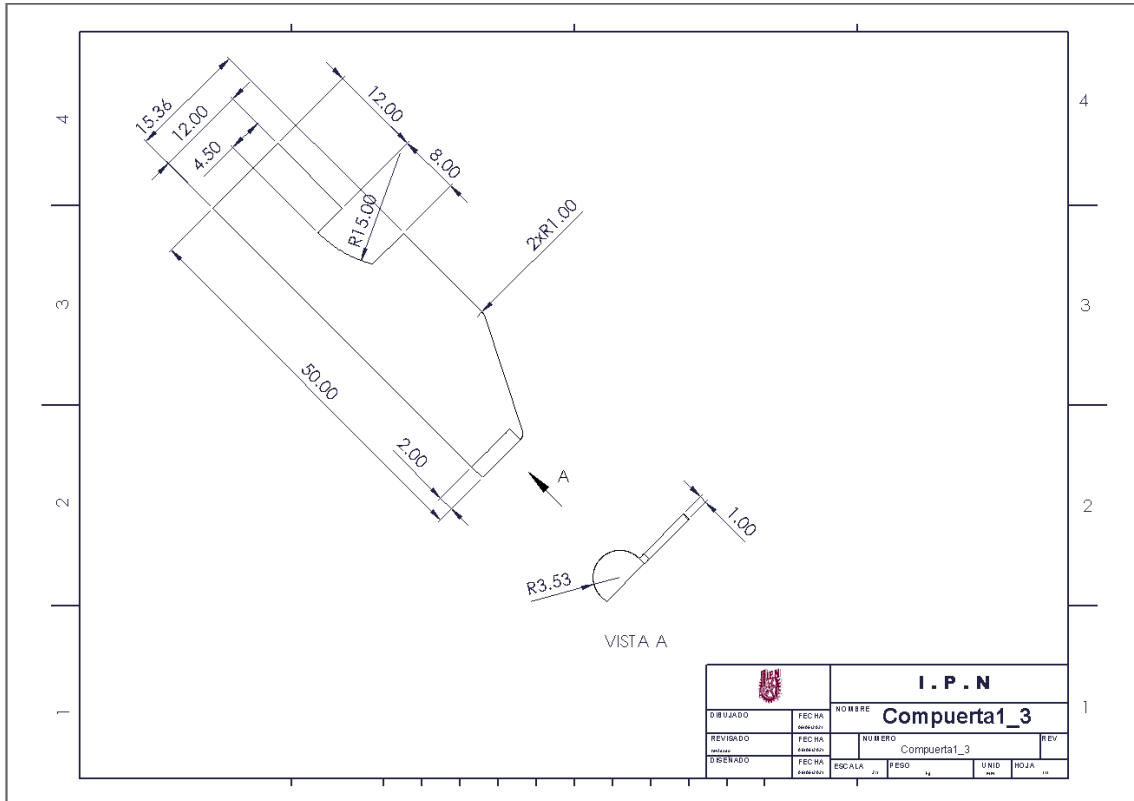
- [56]- «amazon» [En línea]. Disponible en: https://www.amazon.com.mx/ADUCI-Completa-Altavoces-port%C3%B3n/dp/B08RXLMC79/ref=asc_df_B08RXLMC79/?tag=gledskshopmx-20&linkCode=df0&hvadid=486955325997&hvpos=&hvnetw=g&hvrnd=1788657429116861532&hvpon=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1010204&hvtargid=pla-1254841239750&psc=1. [Último acceso: 02 01 2021].
- [57]- «amazon» [En línea]. Disponible en: https://www.amazon.com.mx/pulgadas-Raspberry-pantalla-compatible-Raspbian/dp/B091FYFNV8/ref=asc_df_B091FYFNV8/?tag=gledskshopmx-20&linkCode=df0&hvadid=522669728157&hvpos=&hvnetw=g&hvrnd=14738299792796248451&hvpon=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1010029&hvtargid=pla-1284684734062&psc=1. [Último acceso: 10 05 2021].
- [58]- «mercado libre» [En línea]. Disponible en: https://articulo.mercadolibre.com.mx/MLM653434390-display-touch-35-pulgadas-raspberry-pi-3b-3b-con-carcasa-c-_JM?matt_tool=48904454&matt_word=&matt_source=google&matt_campaign_id=11714912137&matt_ad_group_id=113017550359&matt_match_type=&matt_network=g&matt_device=c&matt_t_creative=482511924687&matt_keyword=&matt_ad_position=&matt_ad_type=pla&matt_merchant_id=117915919&matt_product_id=MLM653434390&matt_product_partition_id=353037831509&matt_target_id=pla-353037831509&gclid=Cj0KCQjwnueFBhChARIsAPu3YkQzbcbfwJx0fd4KRgfzvlGEQGcVF--8as4OJeRcOeVd1p_OB7lr5zoaAqCeEALw_wcB. [Último acceso: 05 05 2021].
- [59]- «jefe sun co» [En línea]. Disponible en: http://headsun.net/self-capacitive-production_gf_55617.html.
- [60]- «ddtech» [En línea]. Disponible en: https://ddtech.mx/producto/fuente-de-poder-cooler-master-mwe-450-450w-80-plus-white-mpe-4501-acaaw-us?id=8938&fbclid=IwARlr2uazXBlcssGSusoK4gKqt27AGhmWilyVsZf0VTan_VU6UIz9ve2gHCw.
- [61]- «ddtech» [En línea]. Disponible en: [https://ddtech.mx/producto/fuente-de-poder-getttech-500w-fg500wbx?id=8912&fbclid=IwARIMgELIzdJmEB4jiYsr4UjnGmzfrNMKWPv5E4qyI\]-sd0Dtymlu_ek4U7k](https://ddtech.mx/producto/fuente-de-poder-getttech-500w-fg500wbx?id=8912&fbclid=IwARIMgELIzdJmEB4jiYsr4UjnGmzfrNMKWPv5E4qyI]-sd0Dtymlu_ek4U7k).
- [62]- «mercado libre» [En línea]. Disponible en: https://www.mercadolibre.com.mx/fuente-de-poder-acteck-power-5-r500-500w-negra-115v230v/p/MLM10896269?pdp_filters=category:MLM1695#searchVariation=MLM10896269&position=2&search_layout=stack&type=product&tracking_id=6566c9e8-38ca-42e6-b091-428b093301a1.

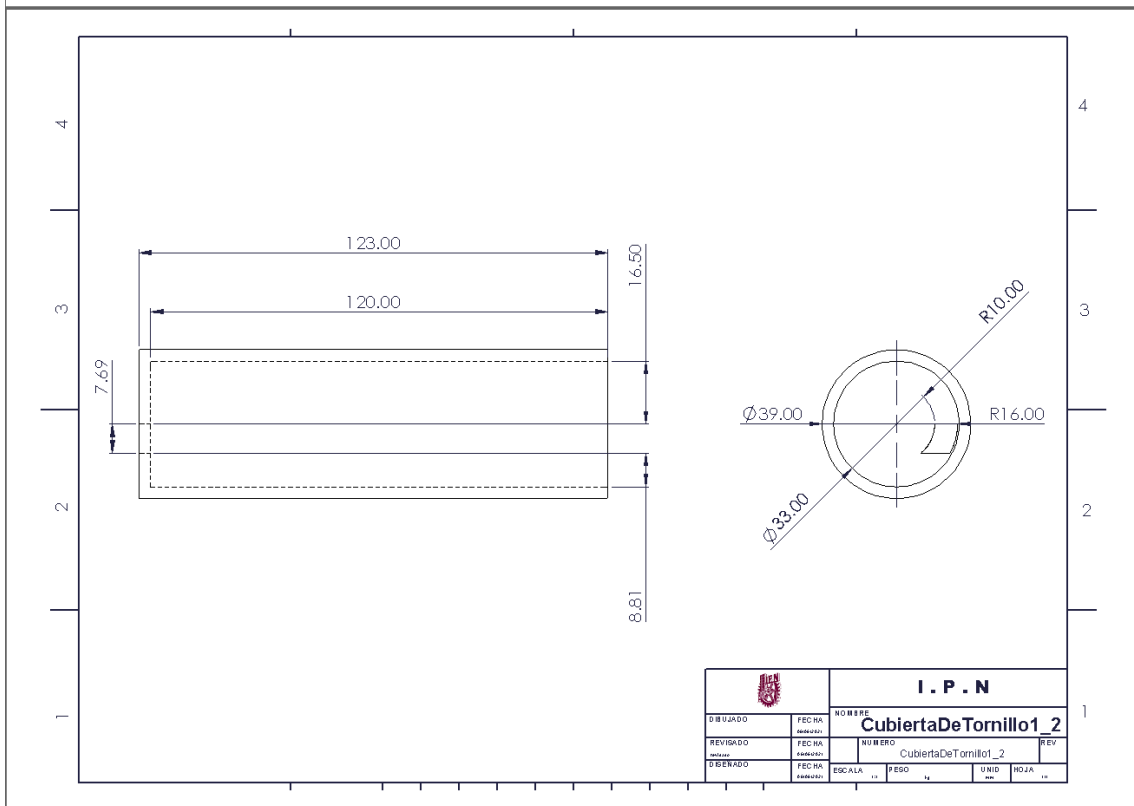
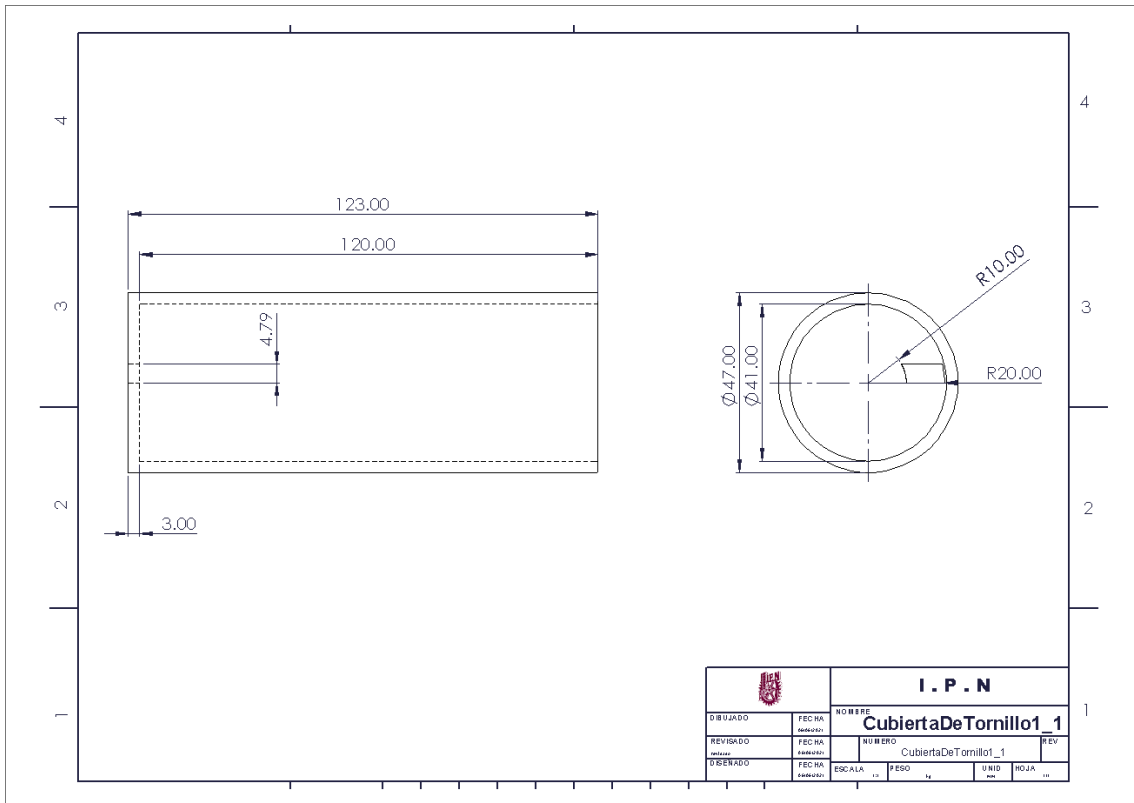


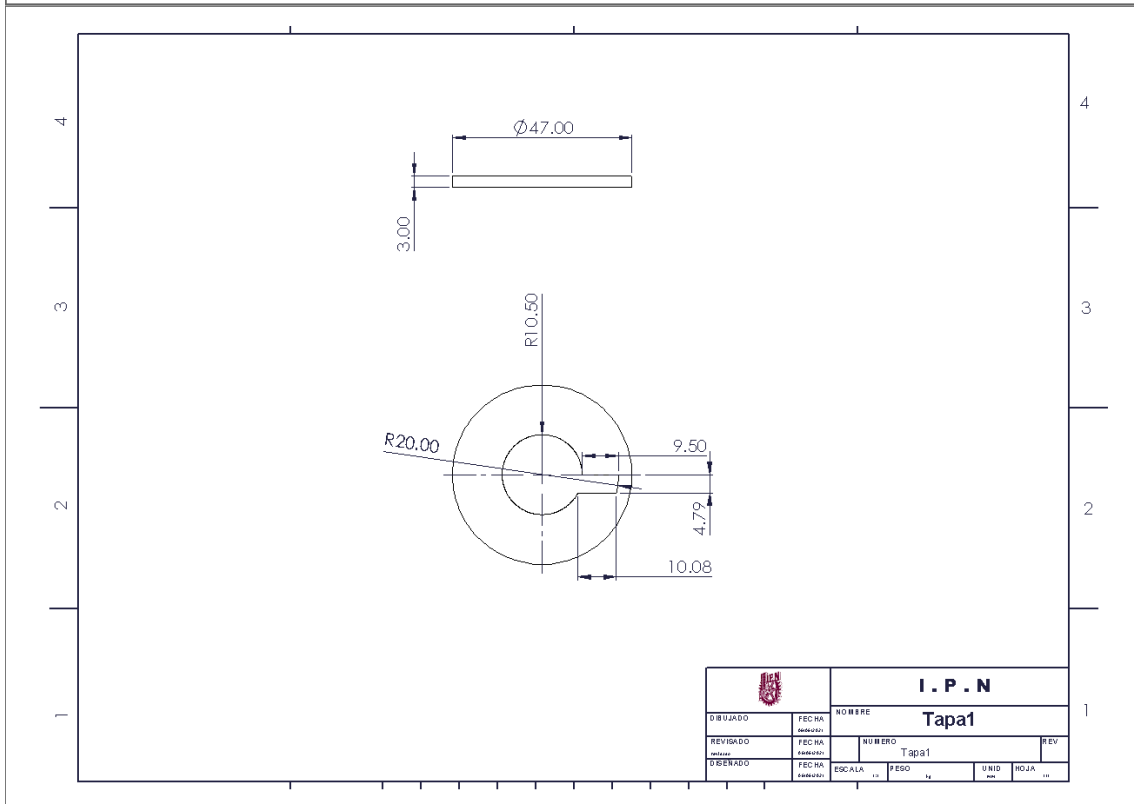
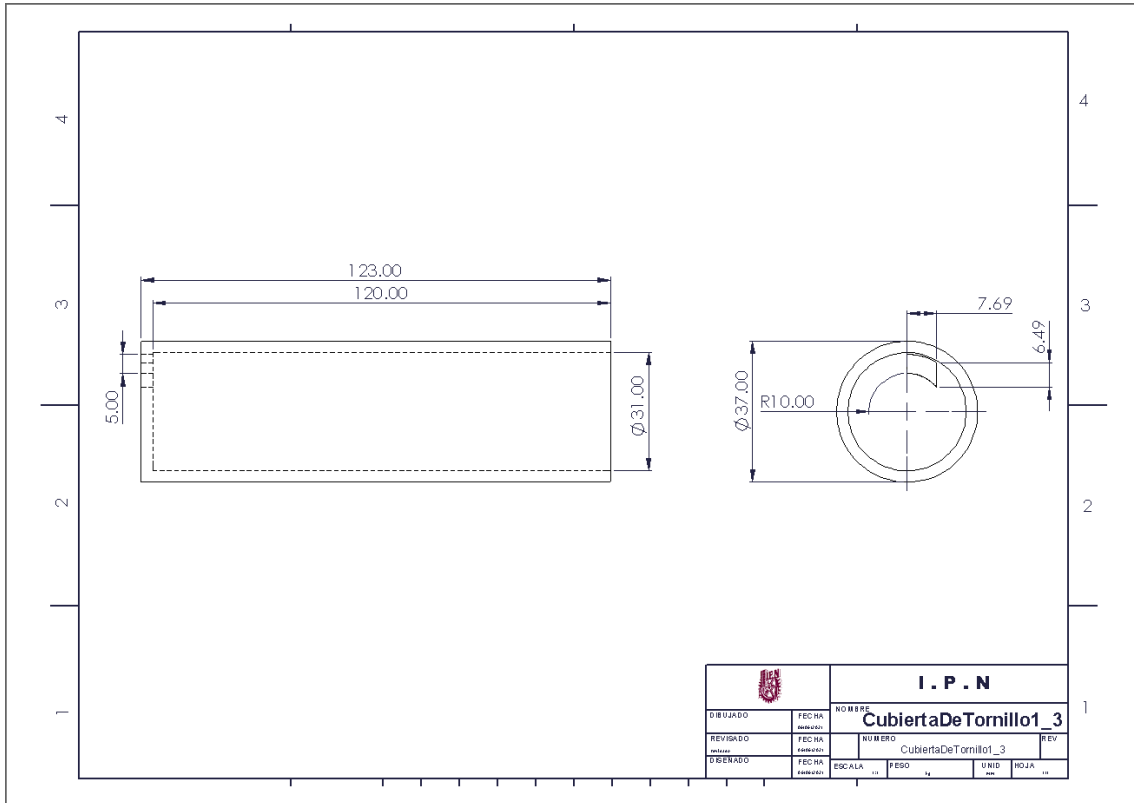
12. Apéndices

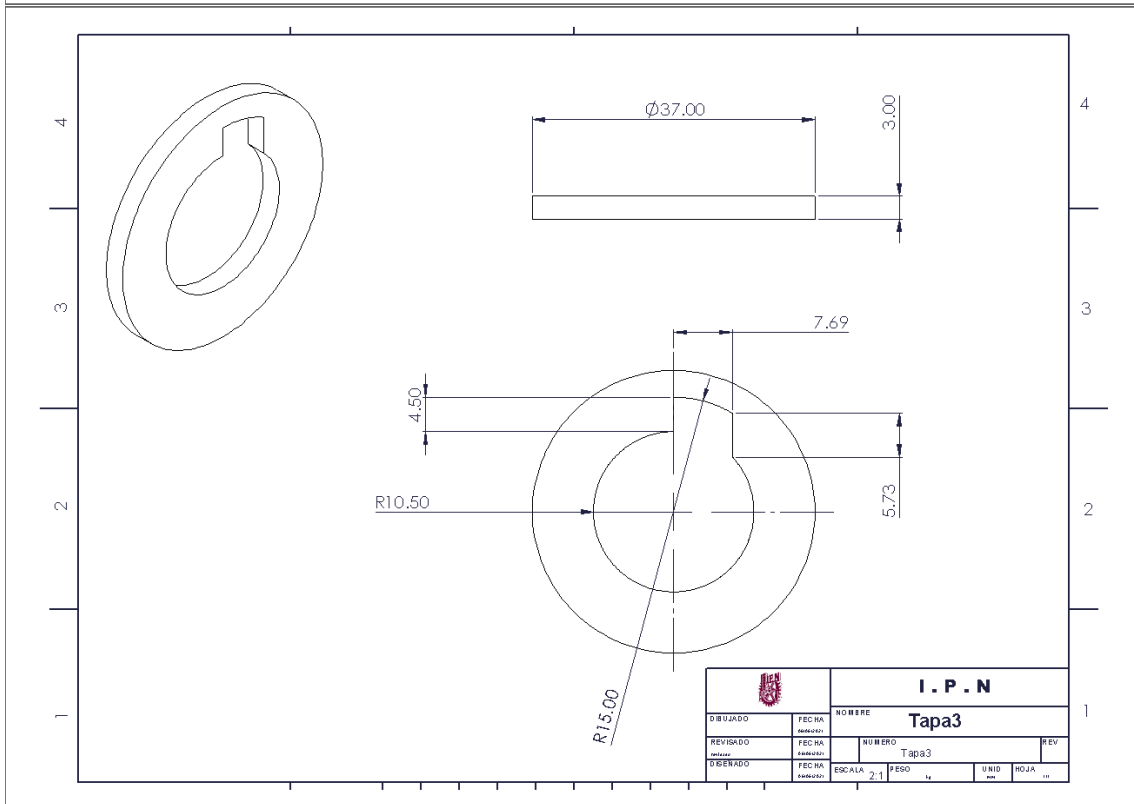
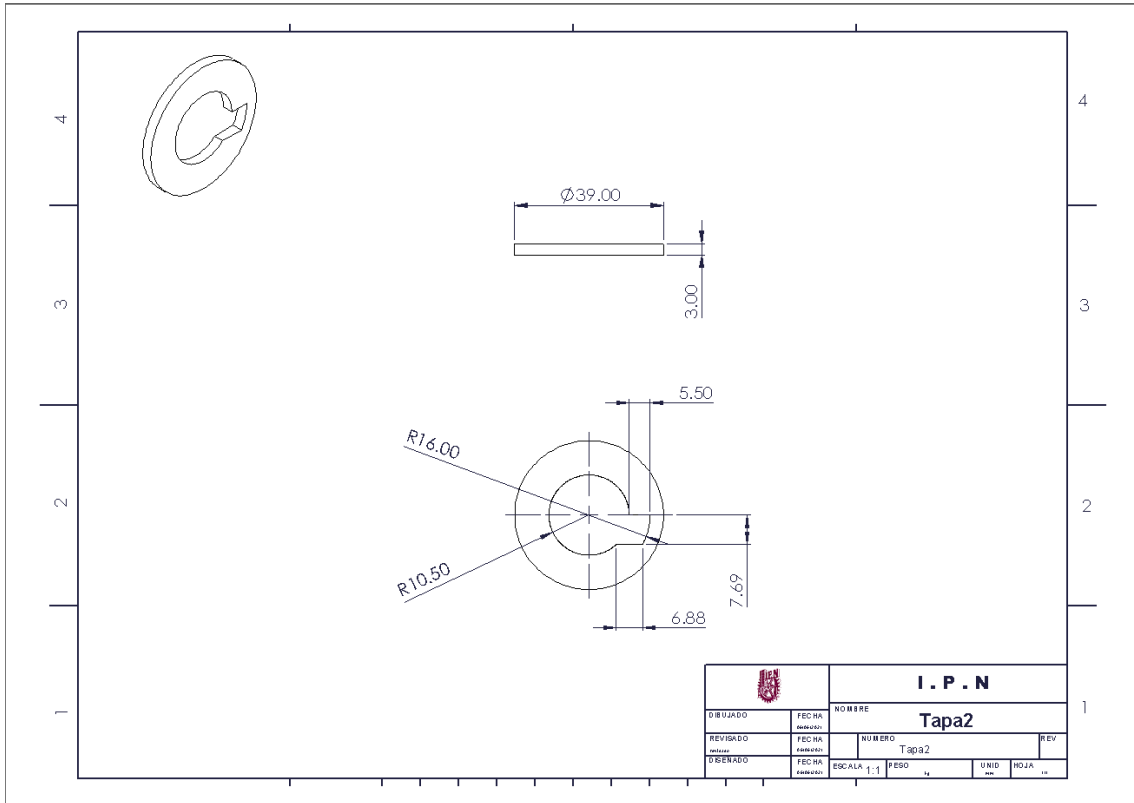


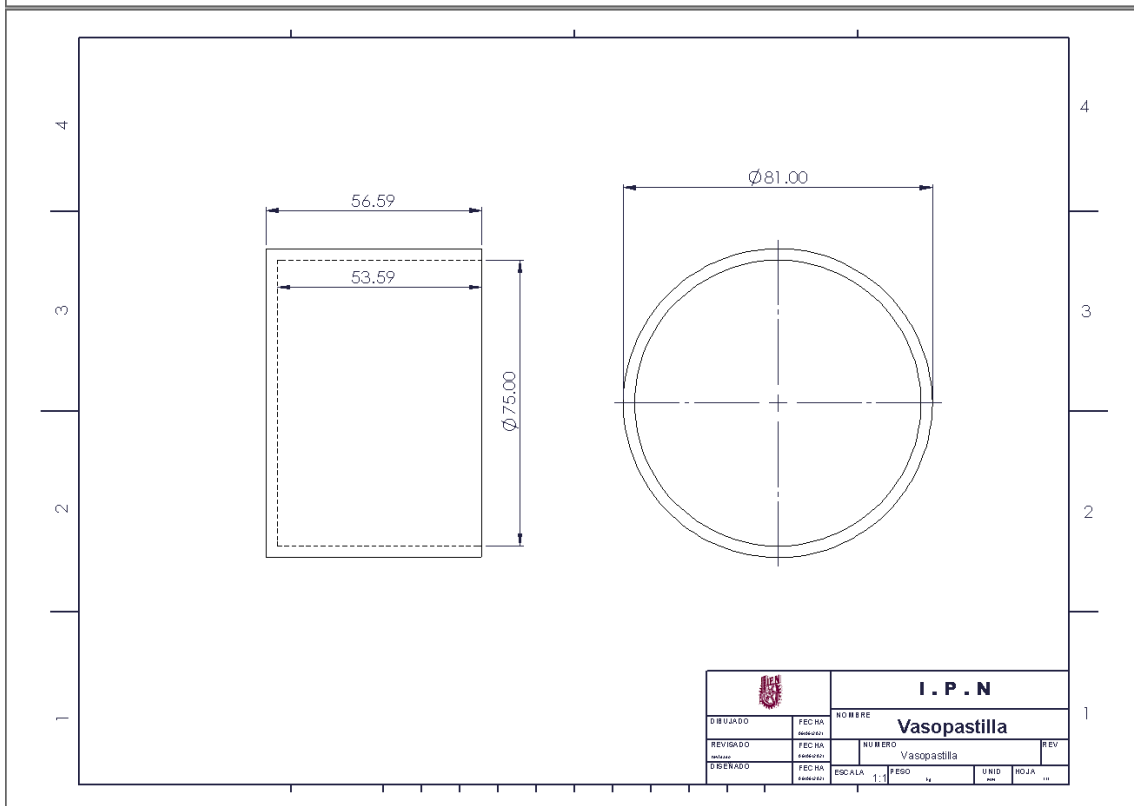
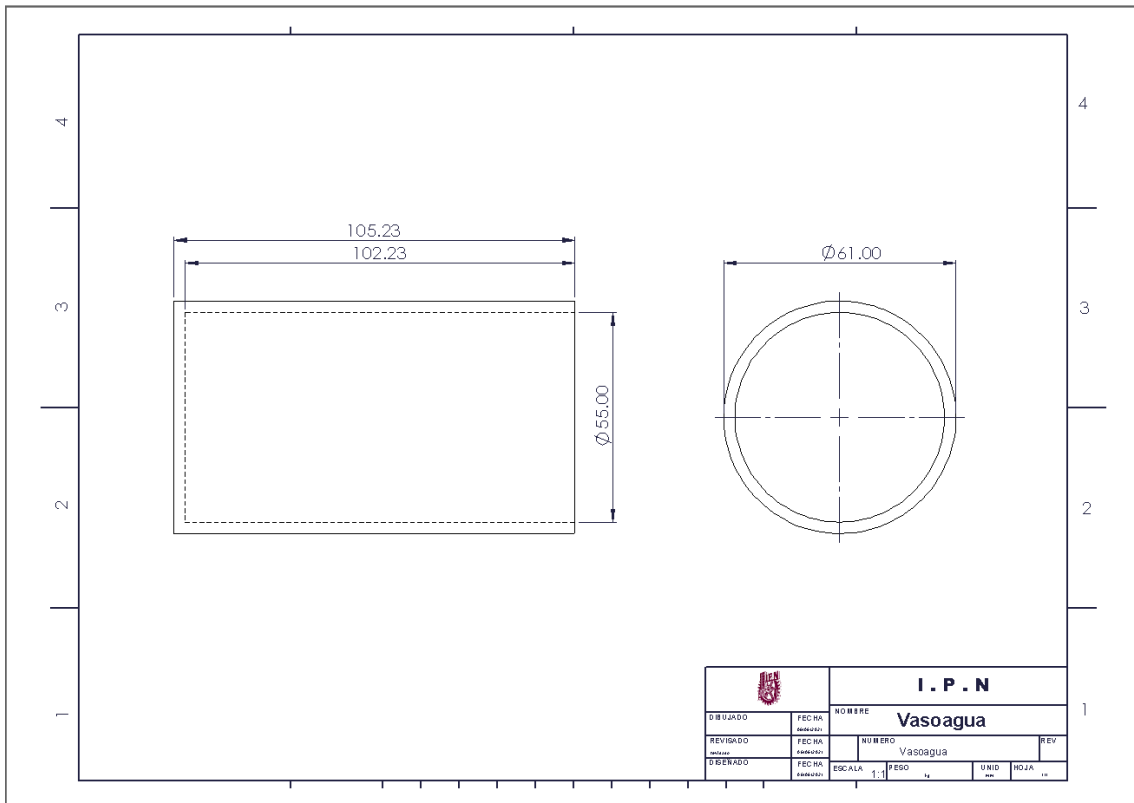


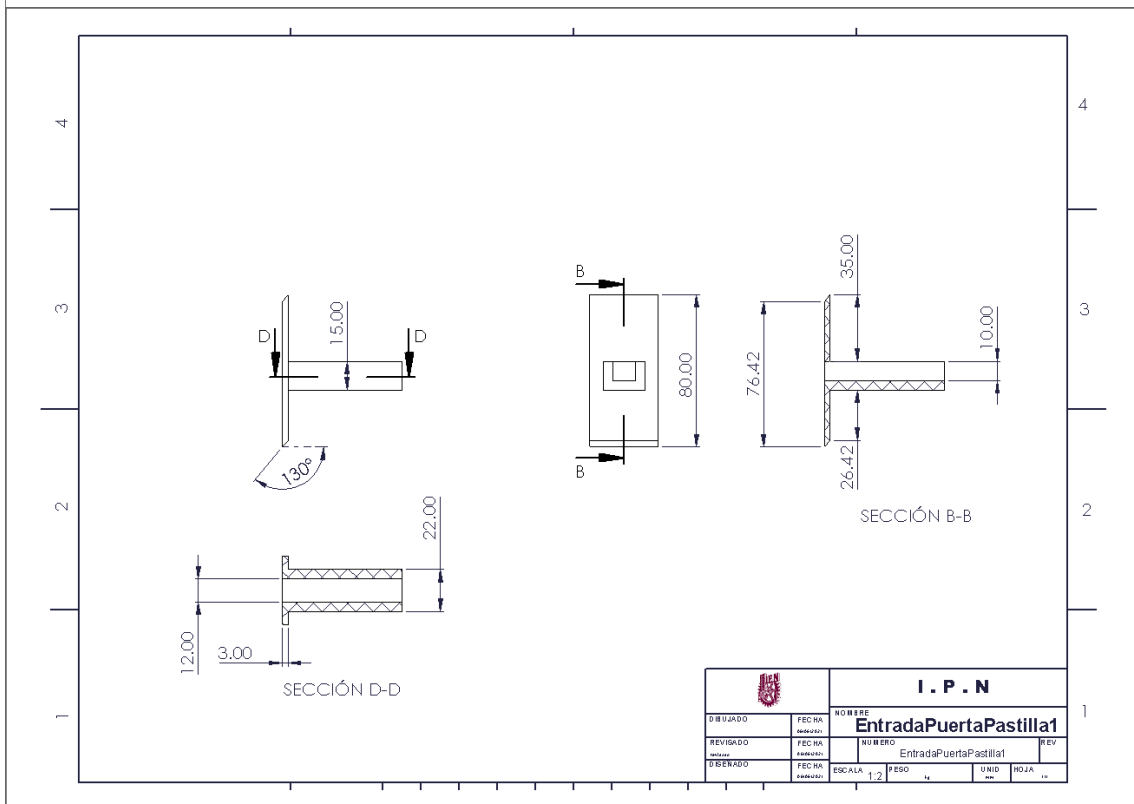
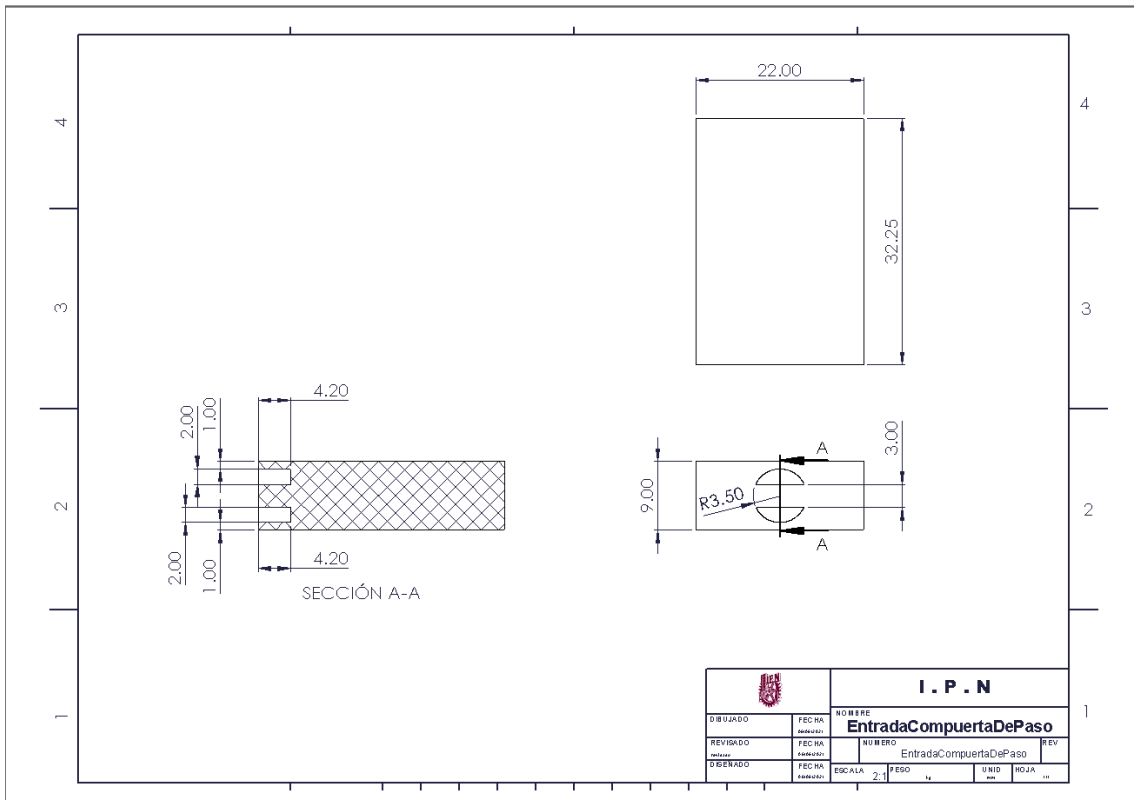


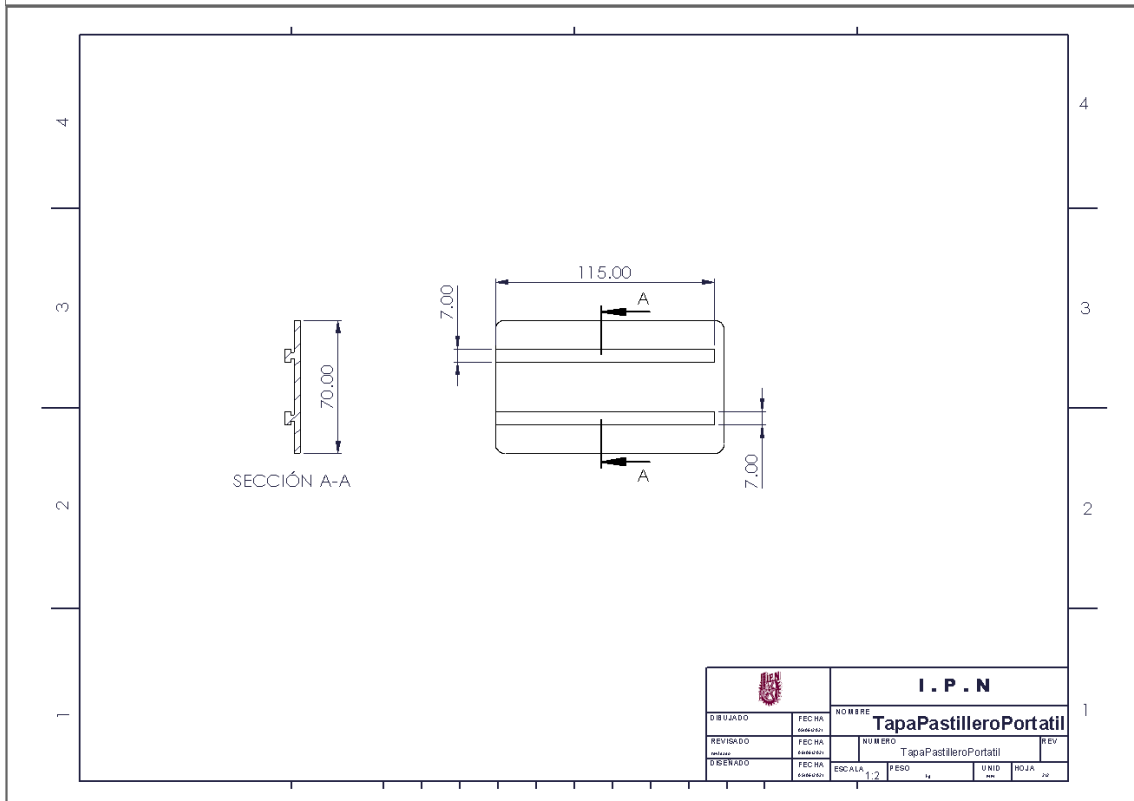
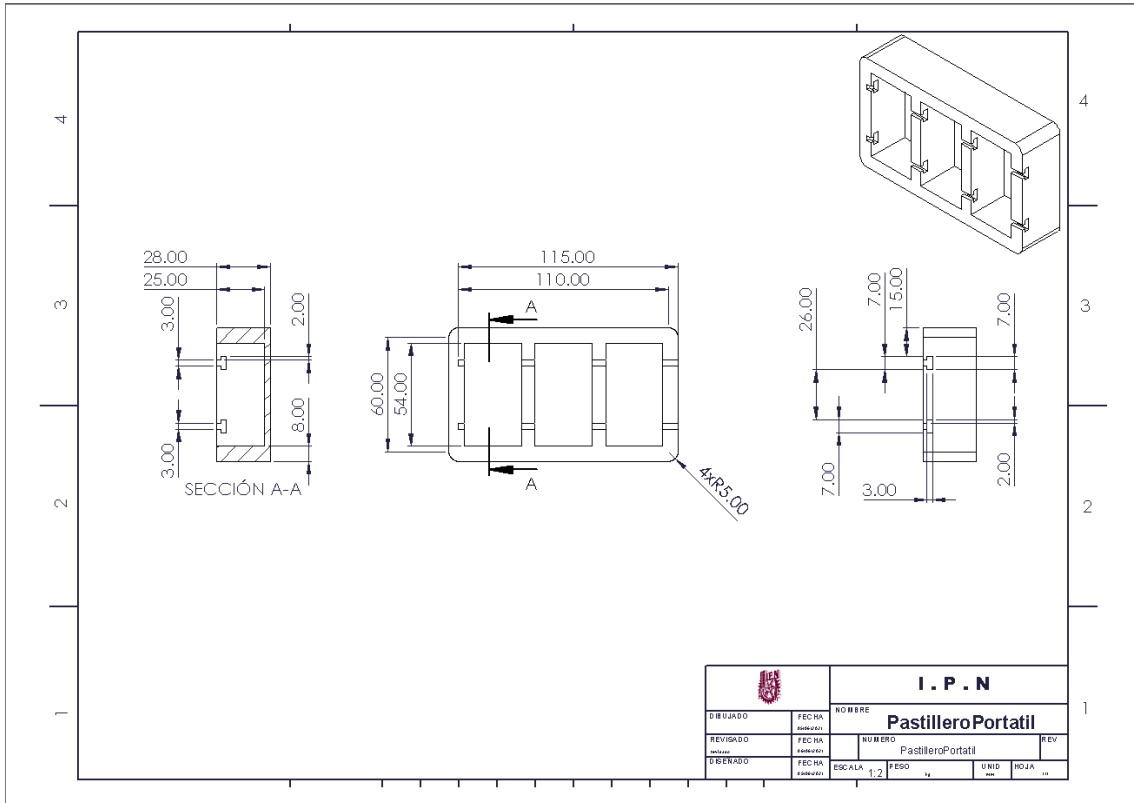


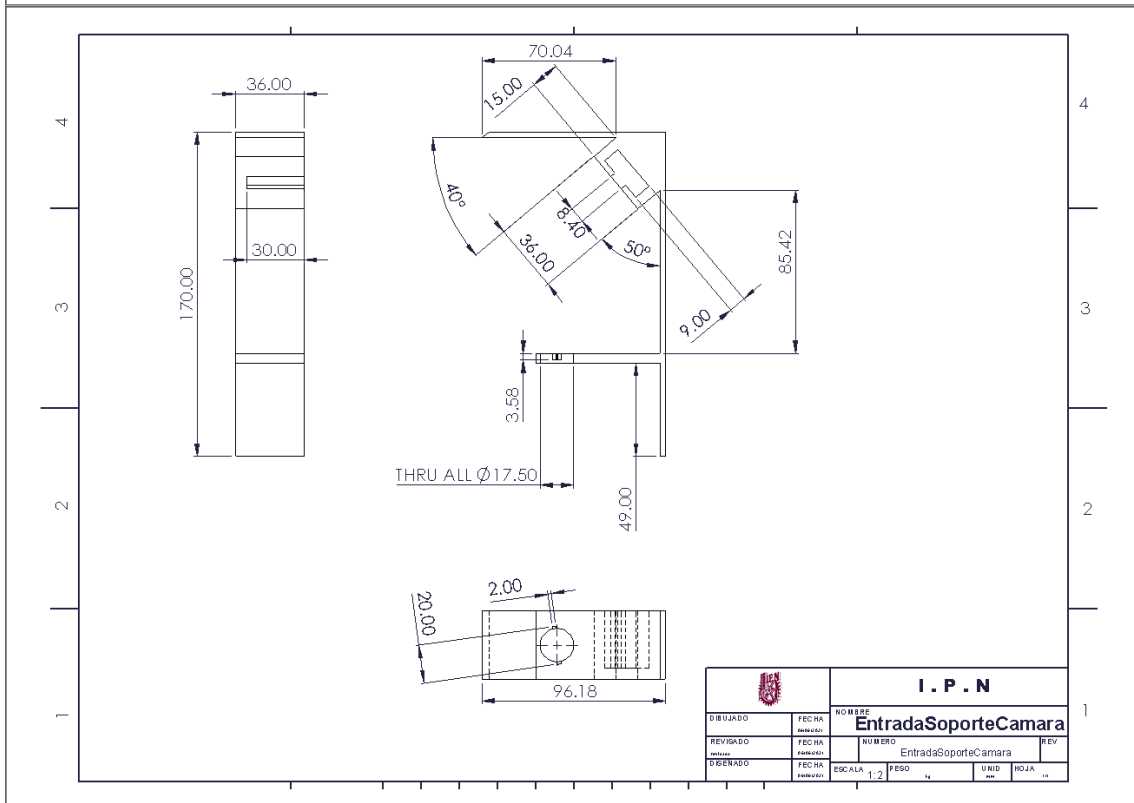
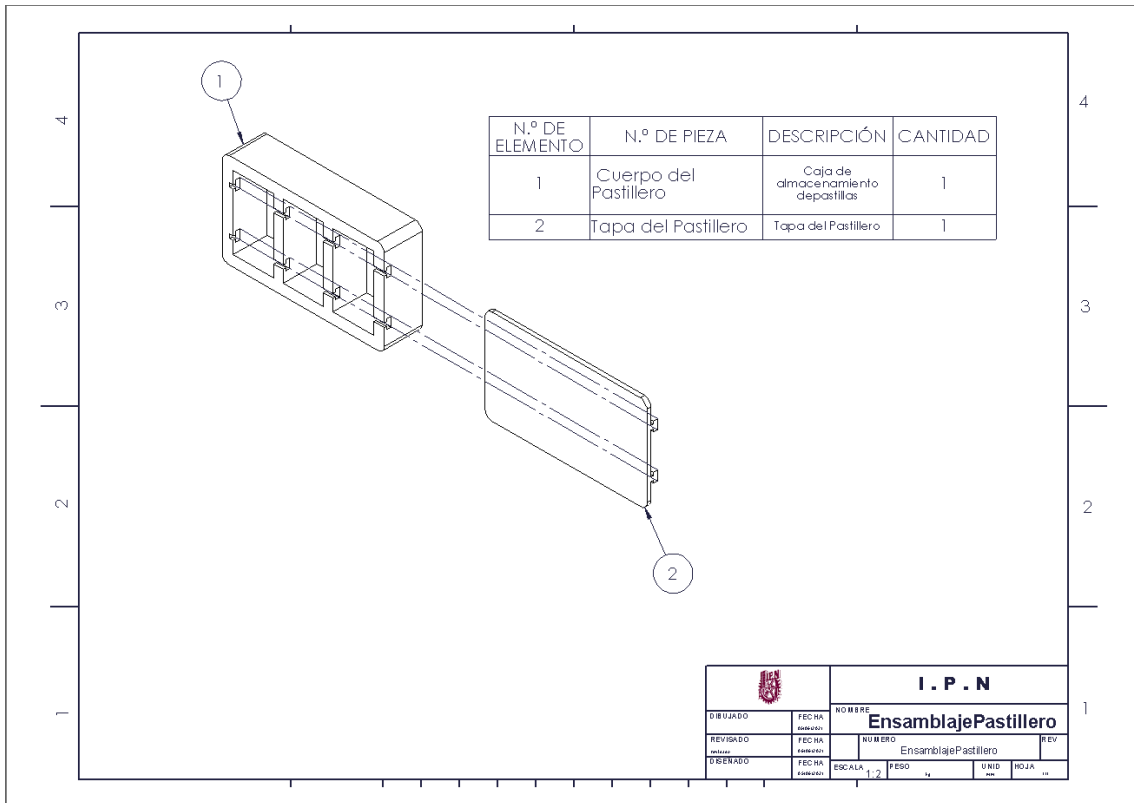


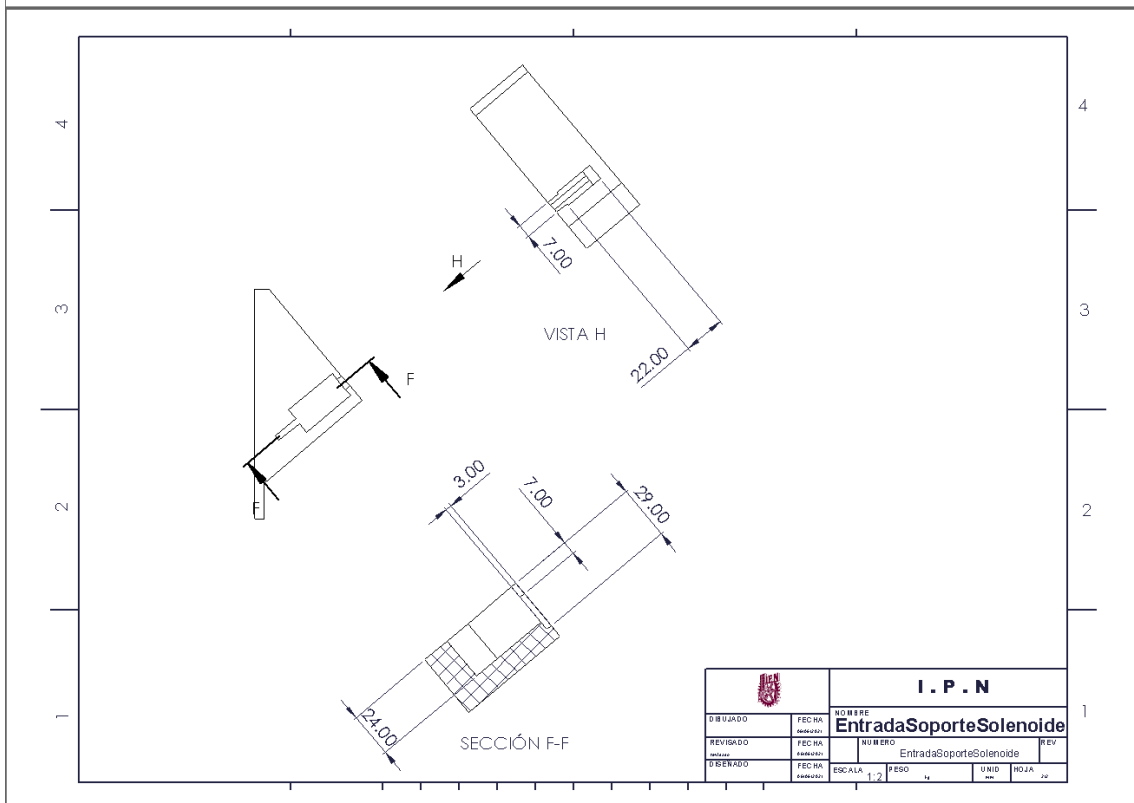
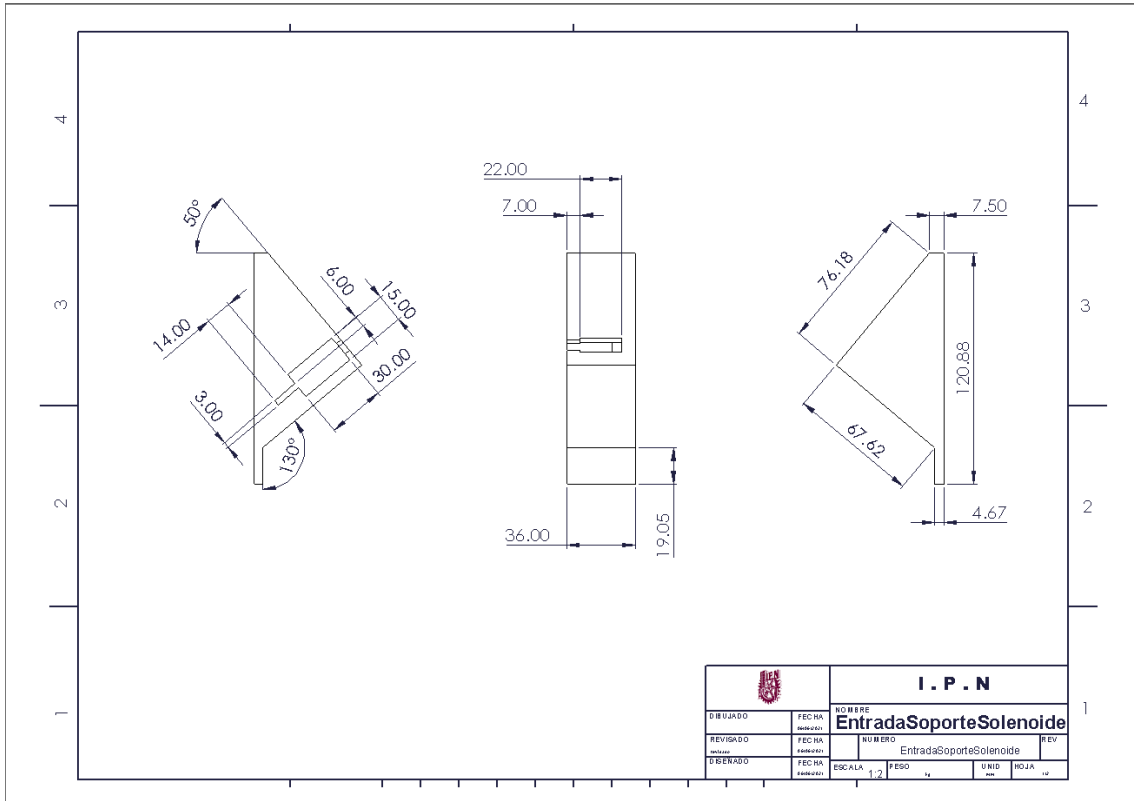


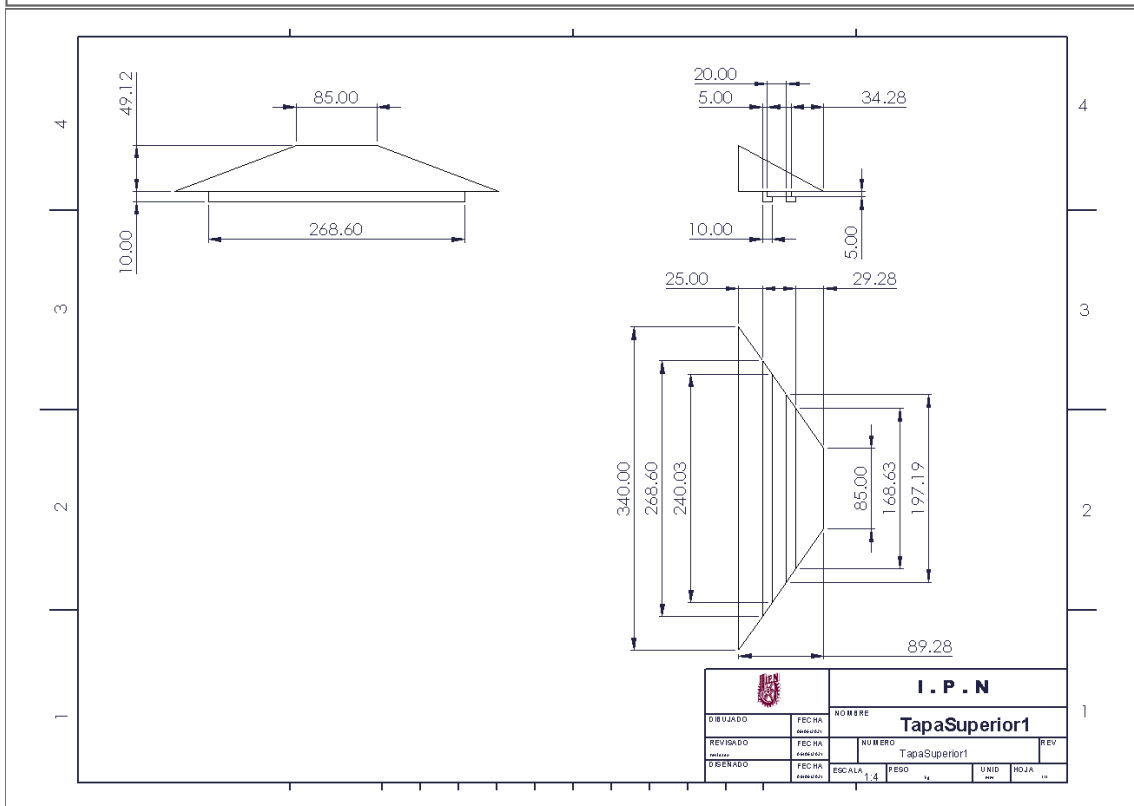
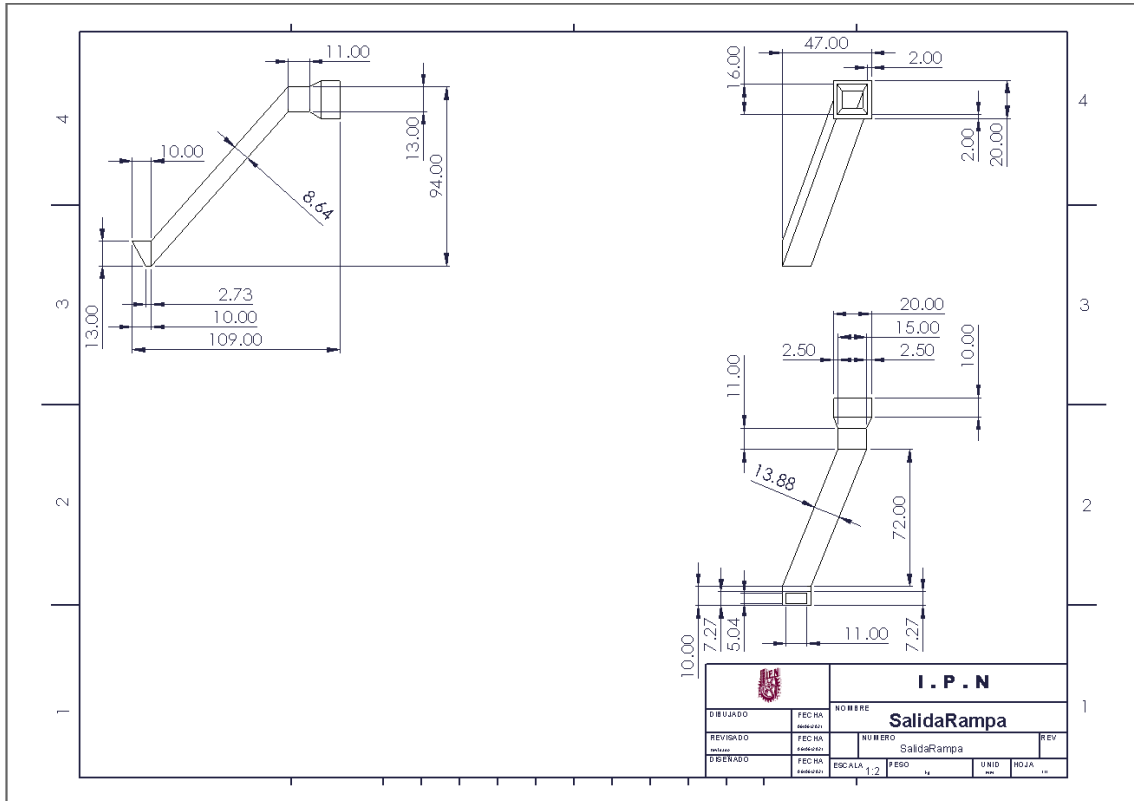


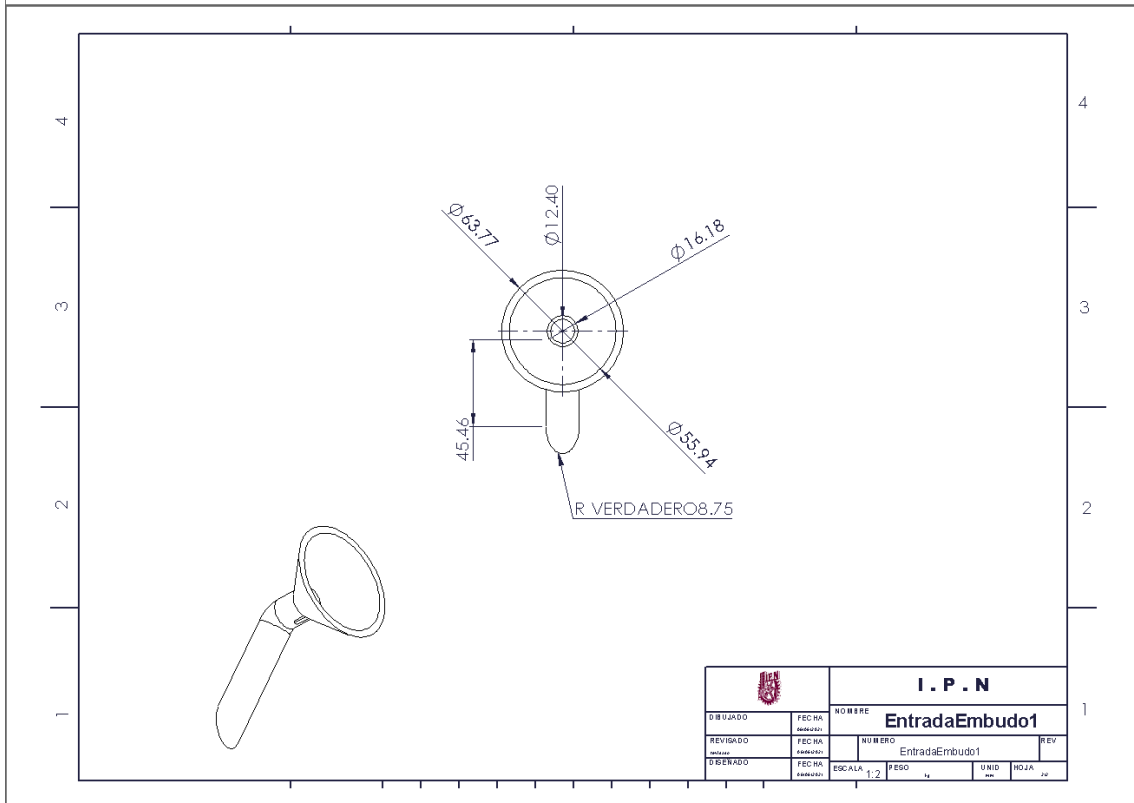
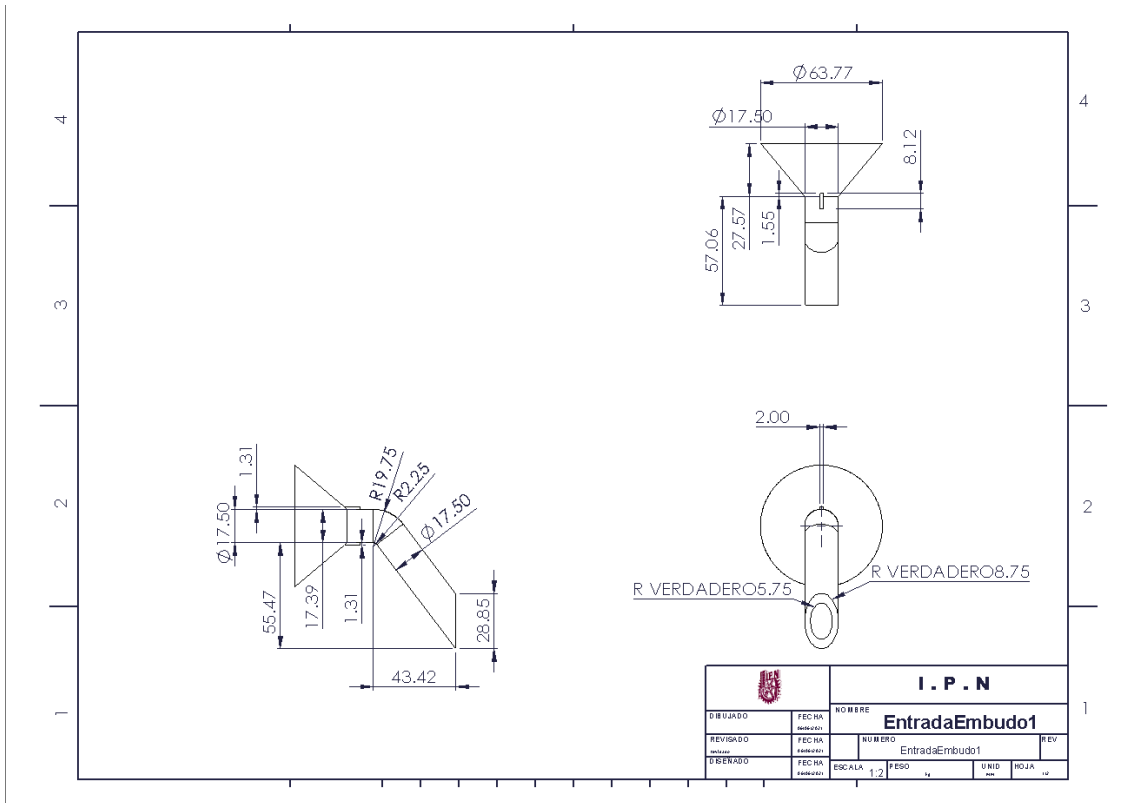


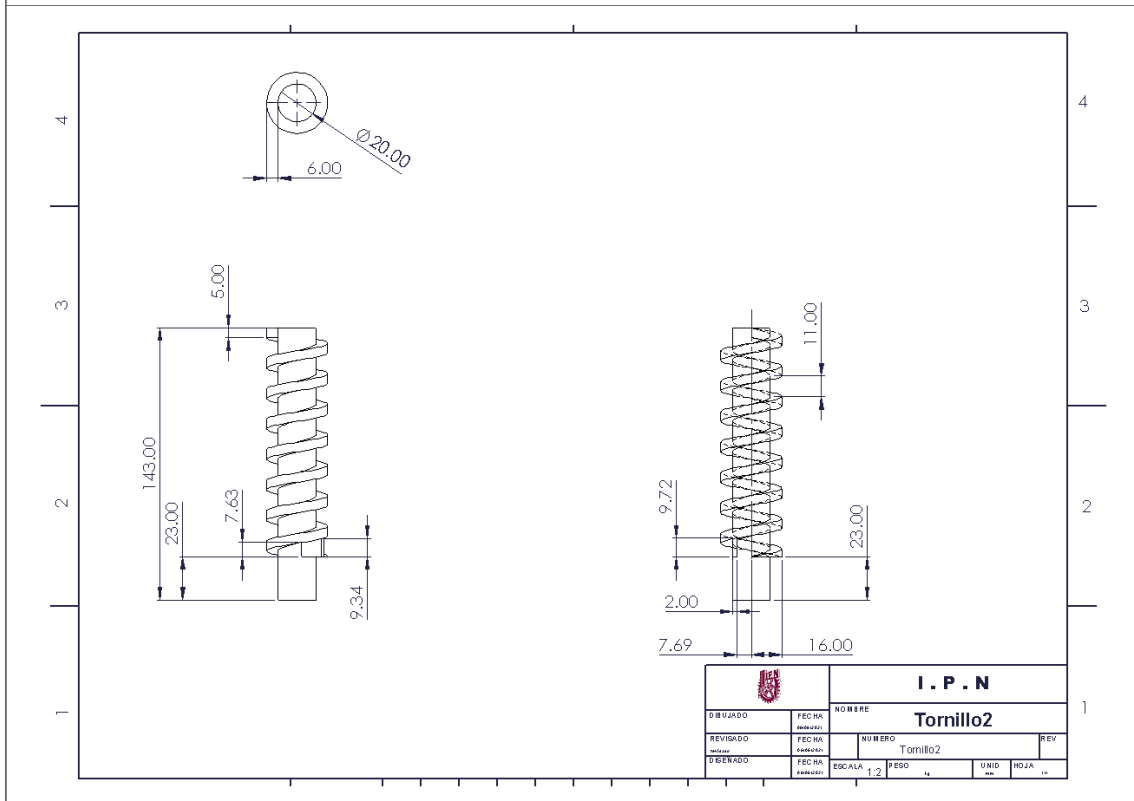
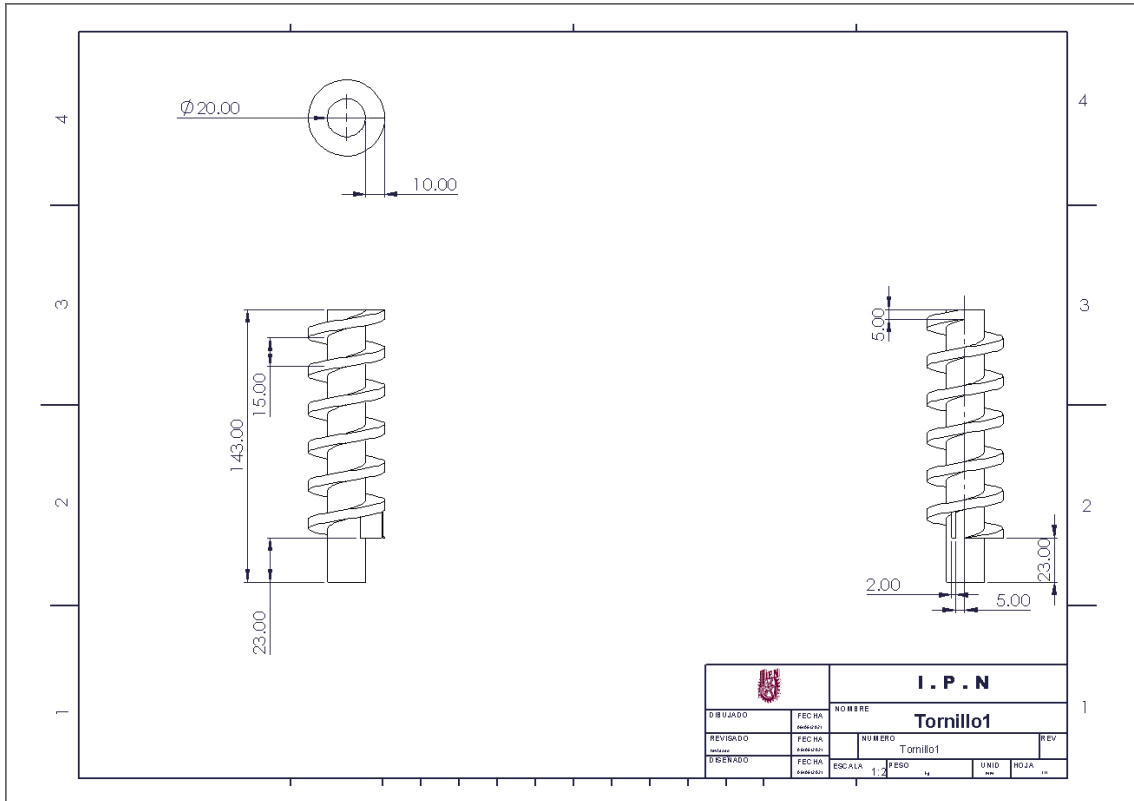


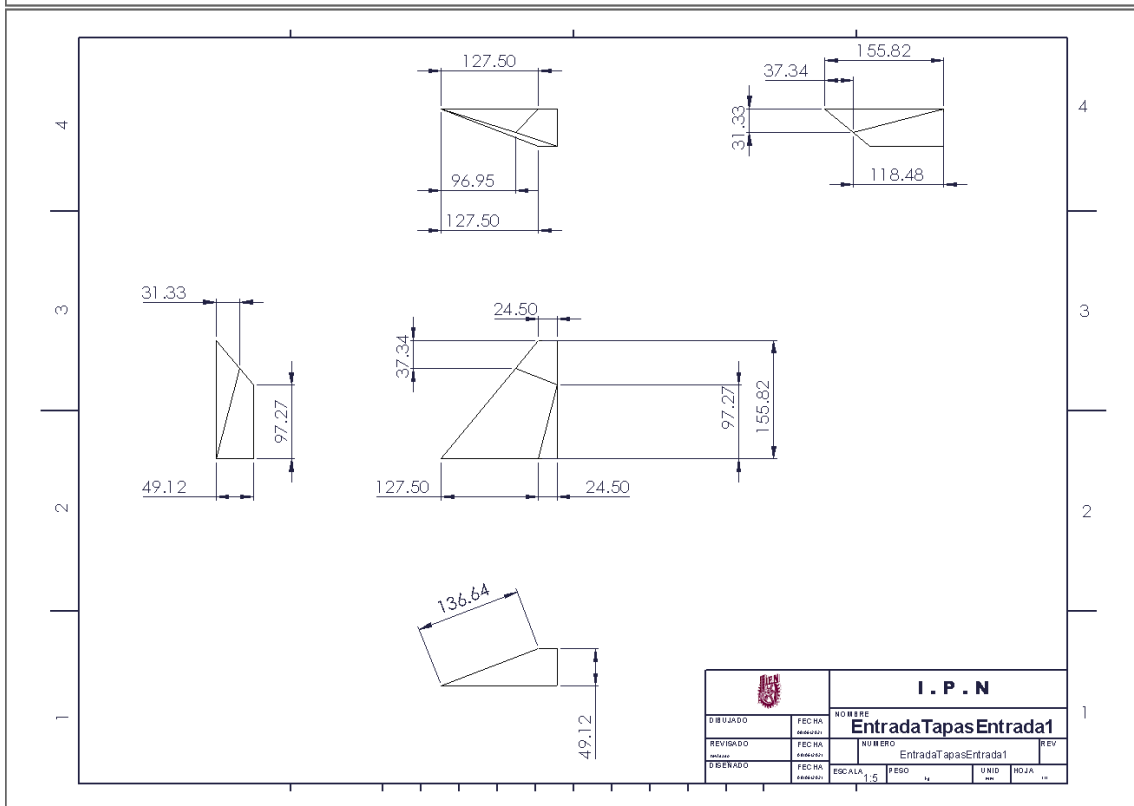
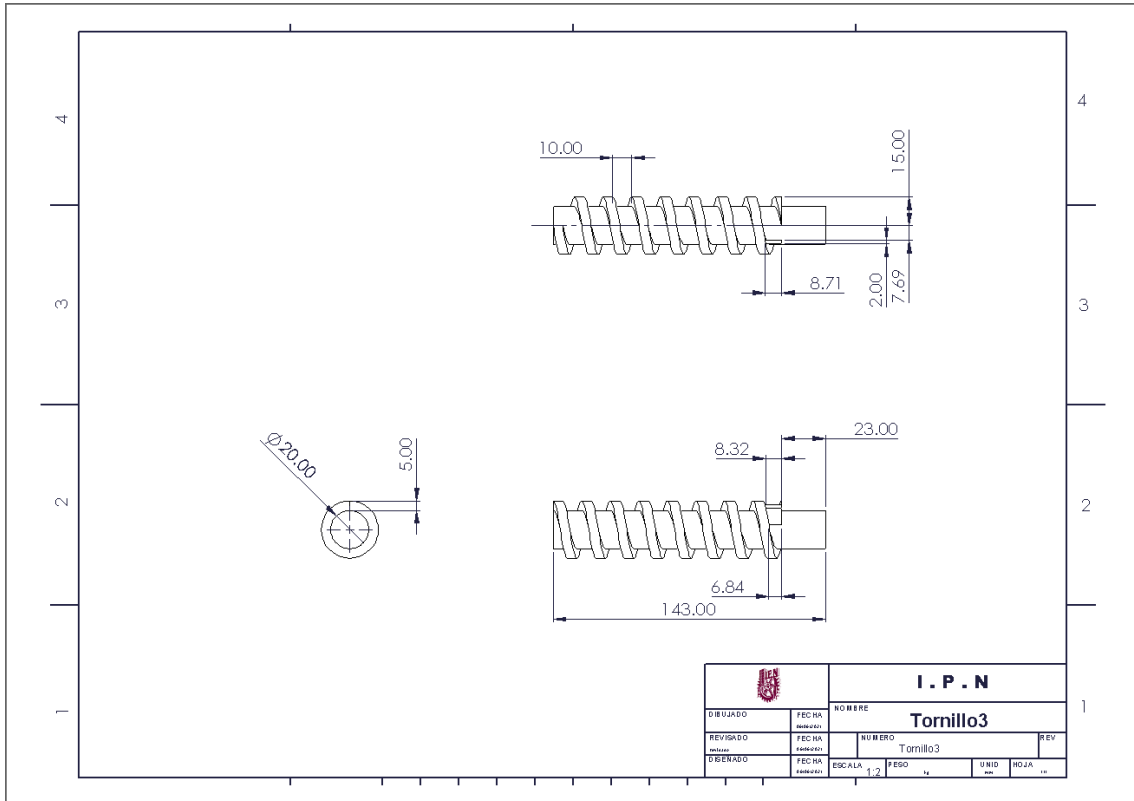


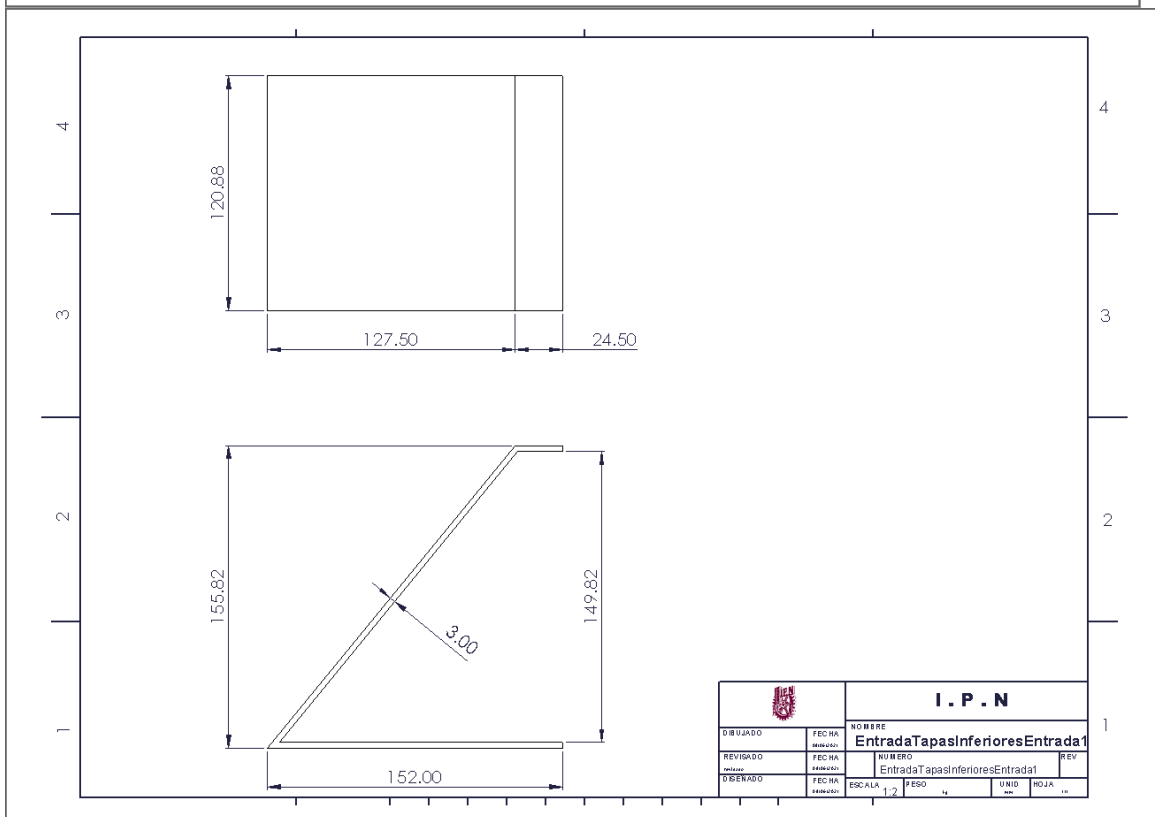
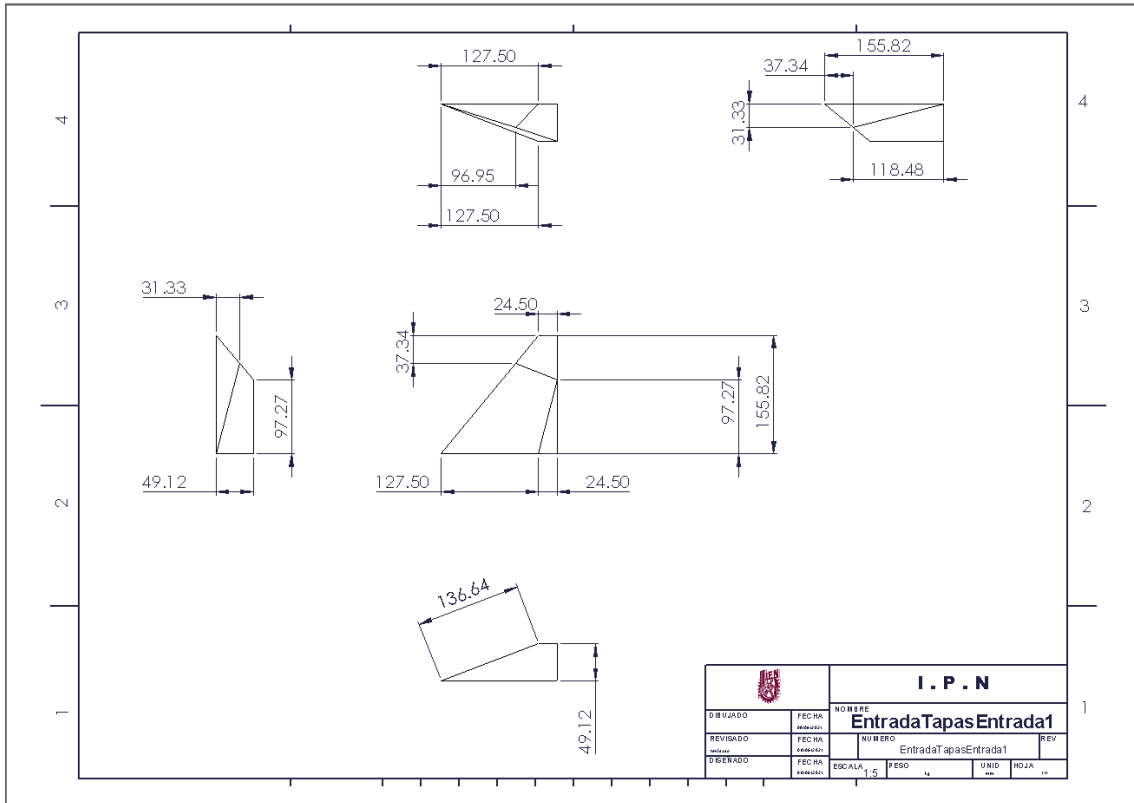


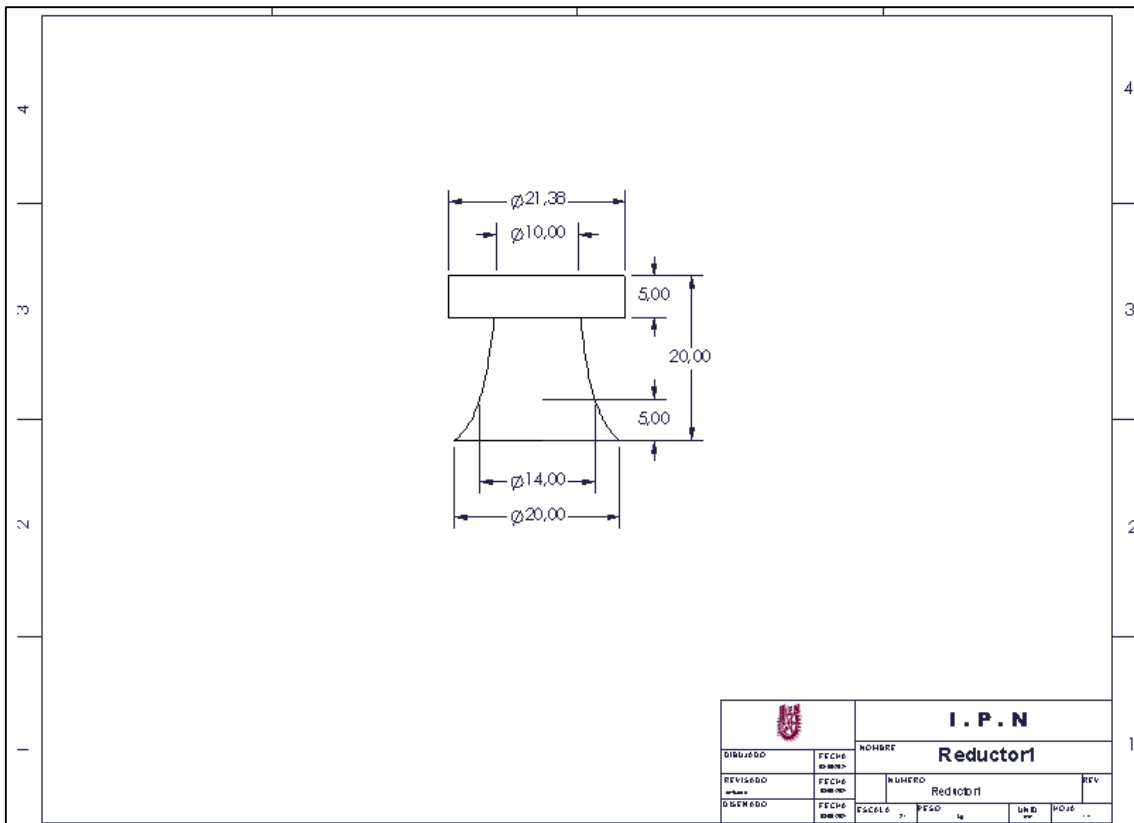
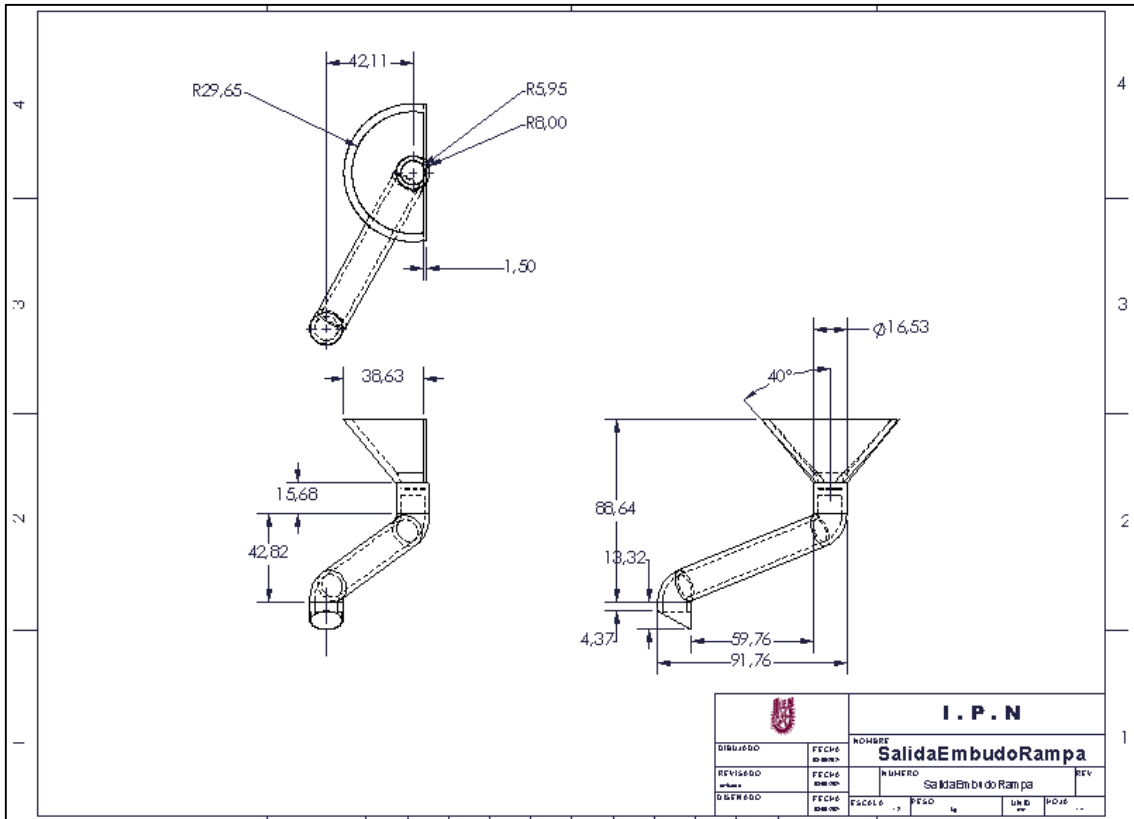


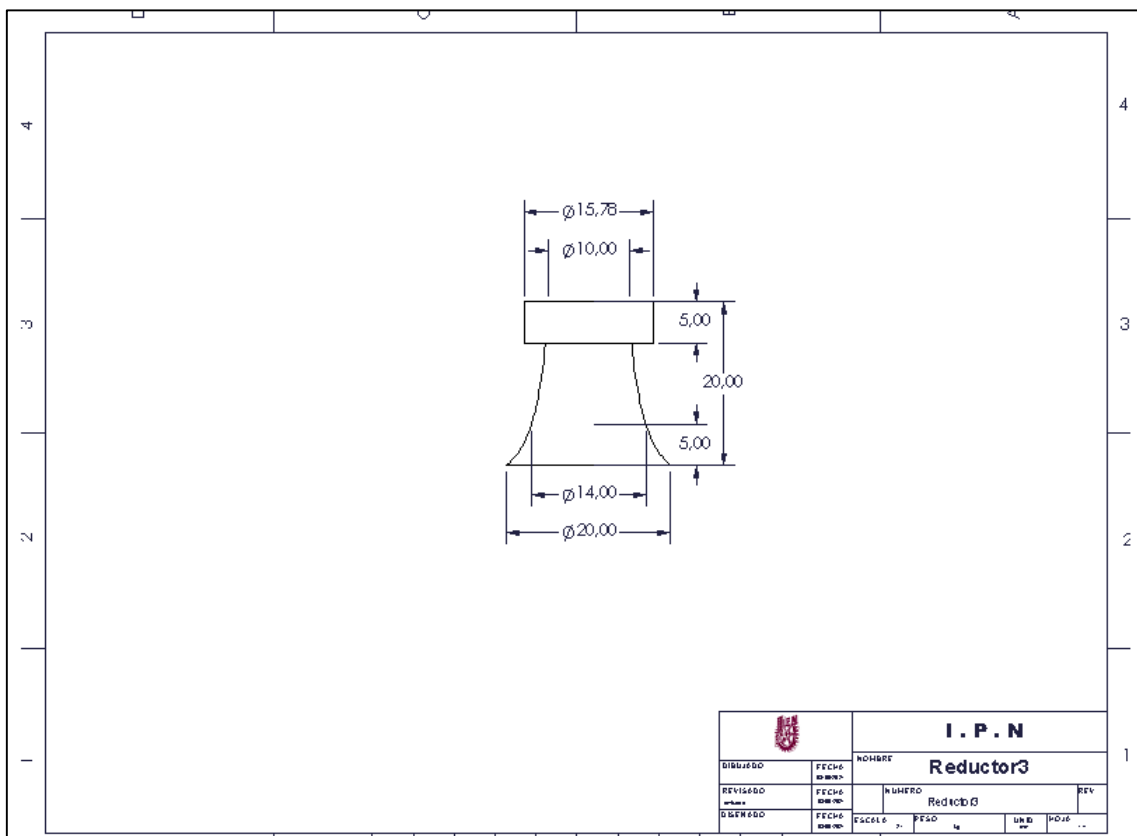
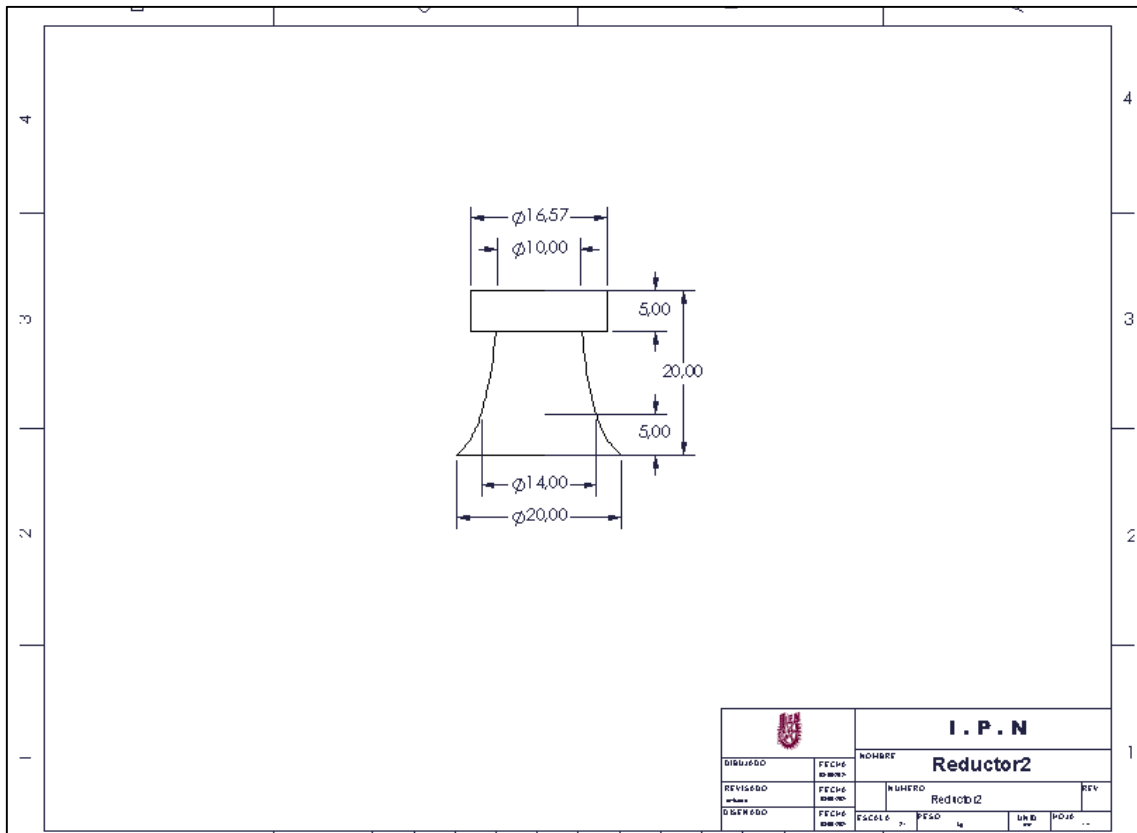


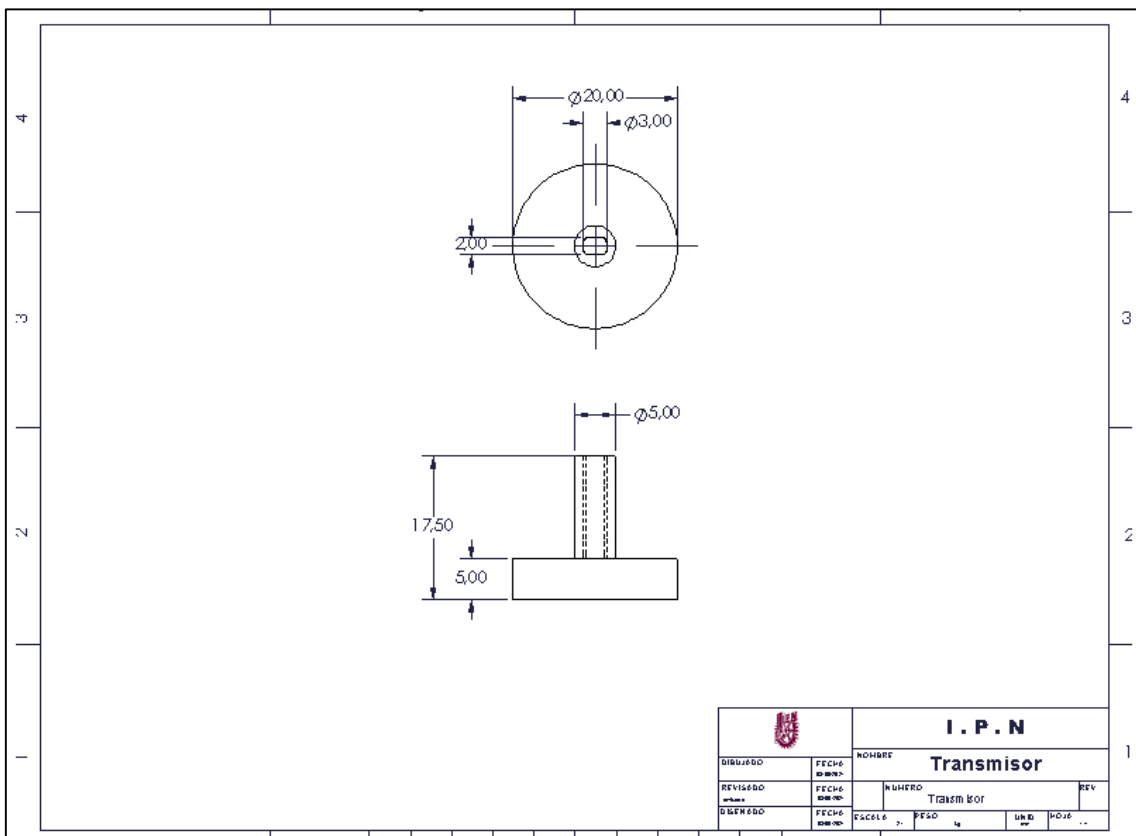
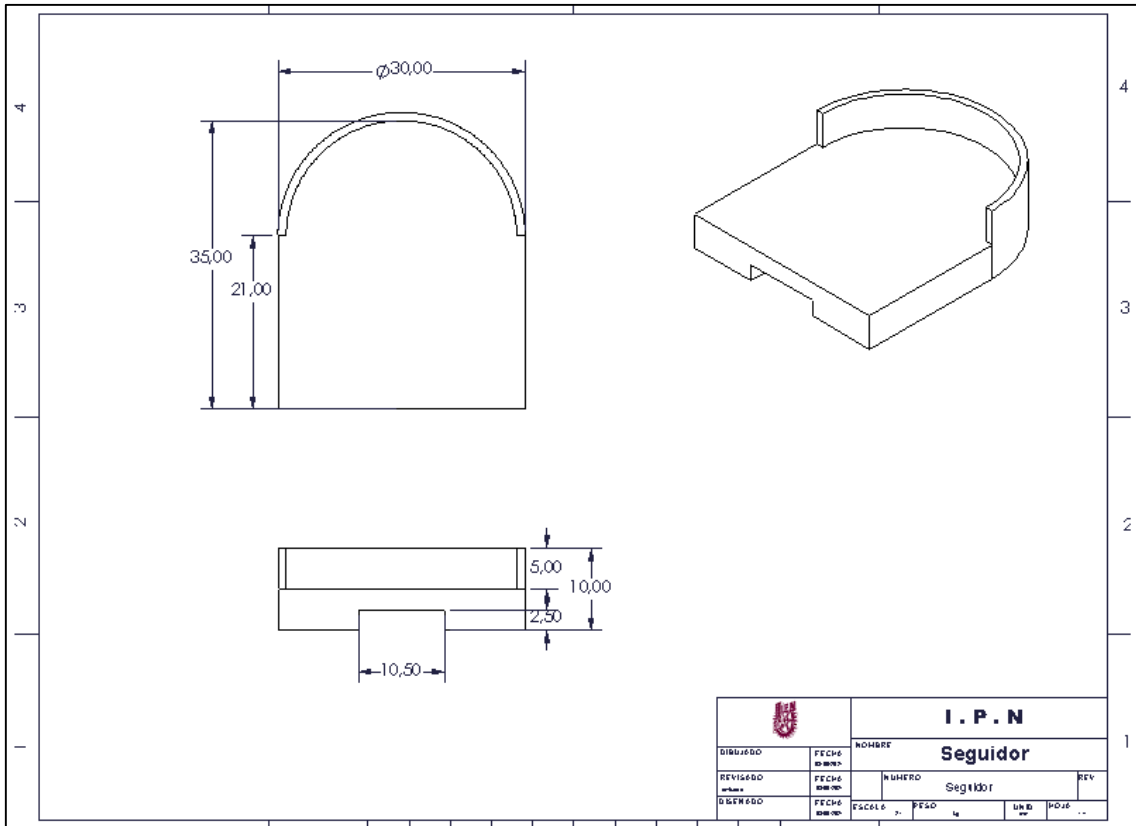


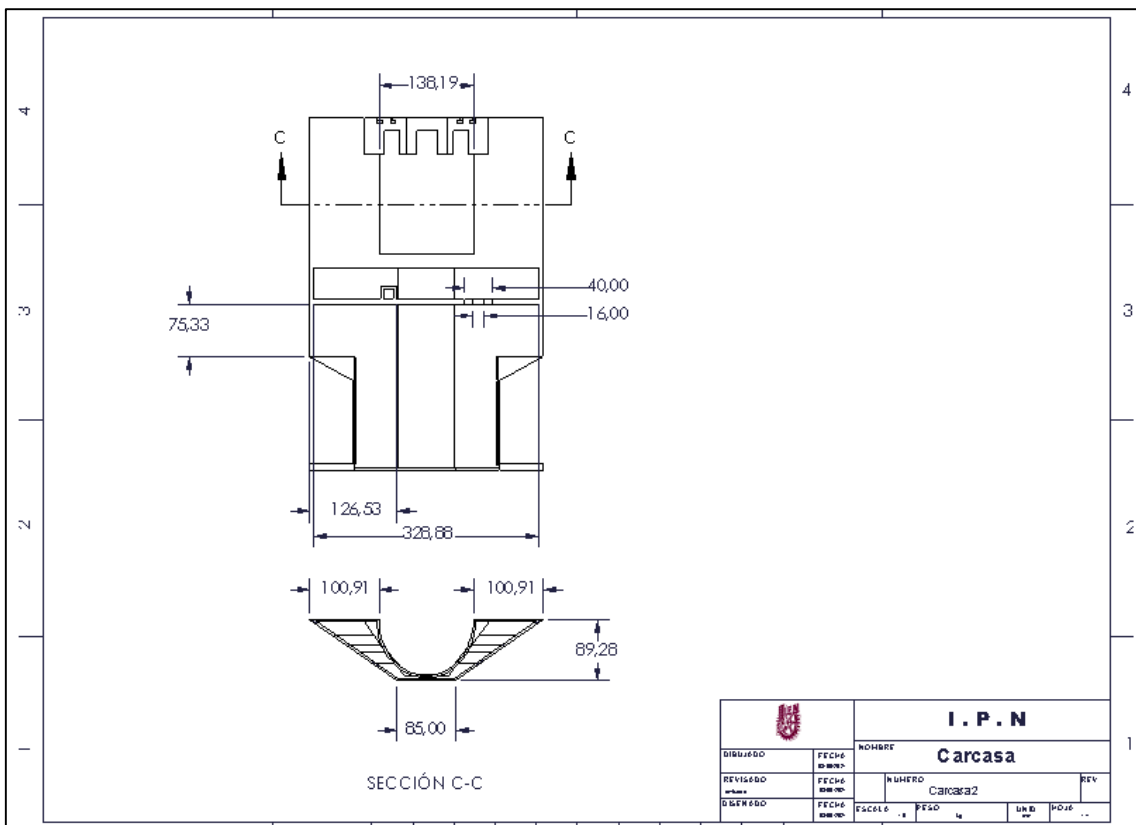
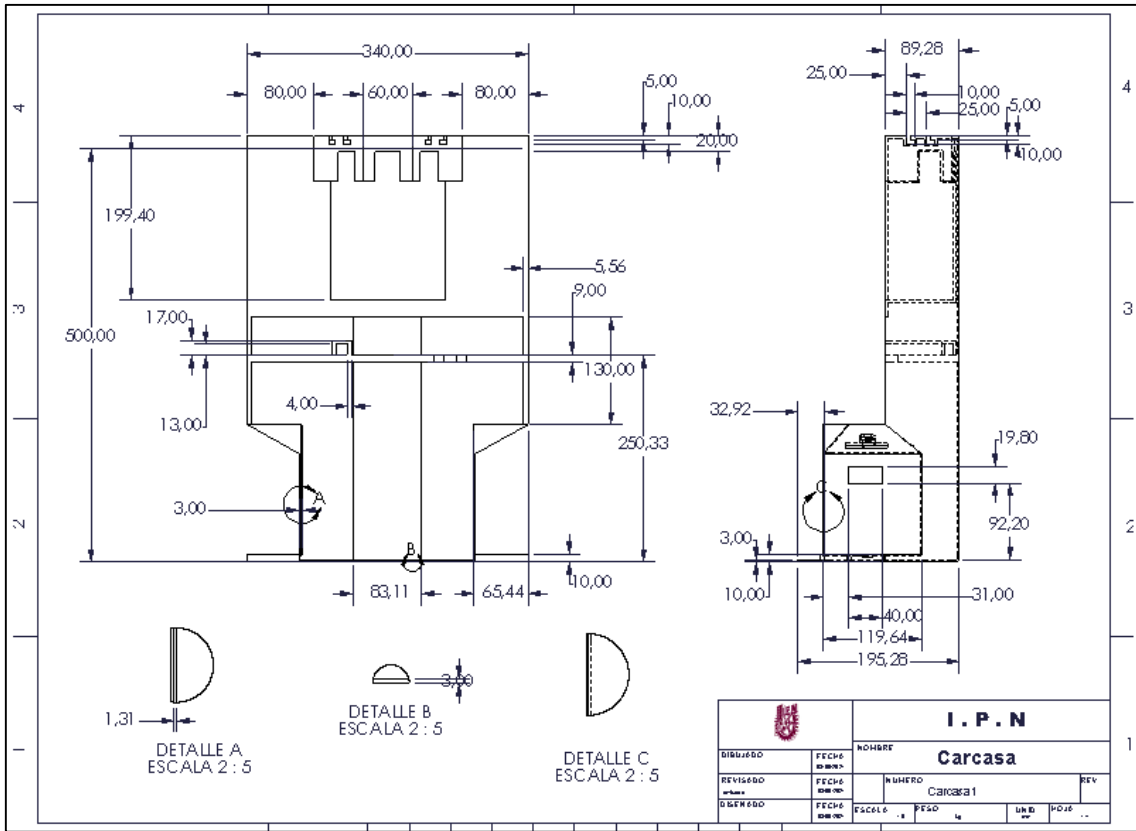


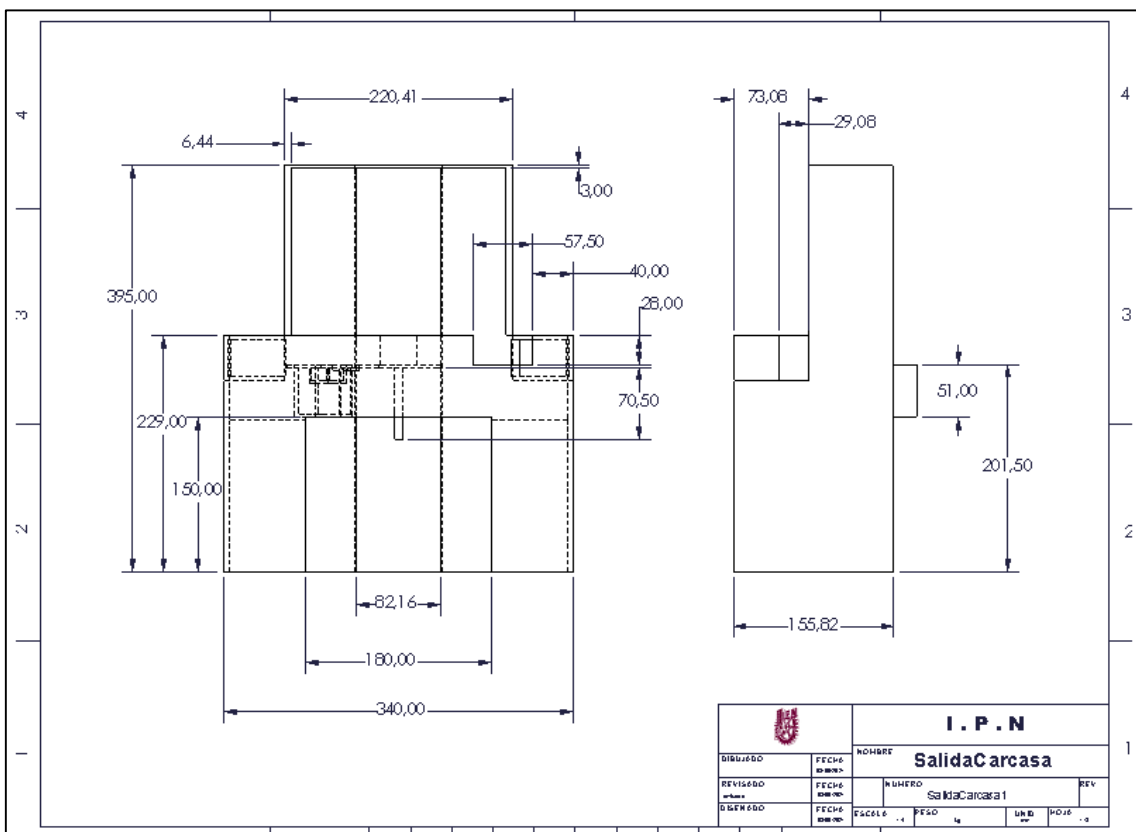
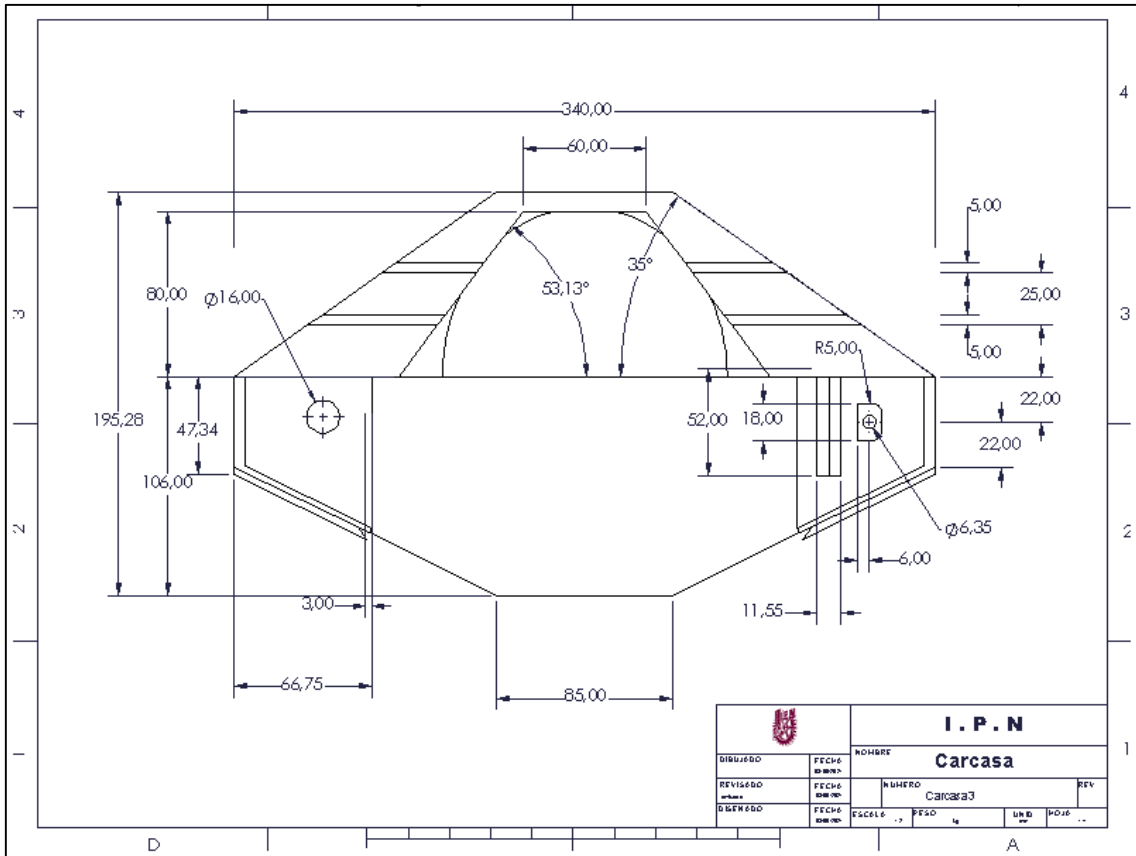


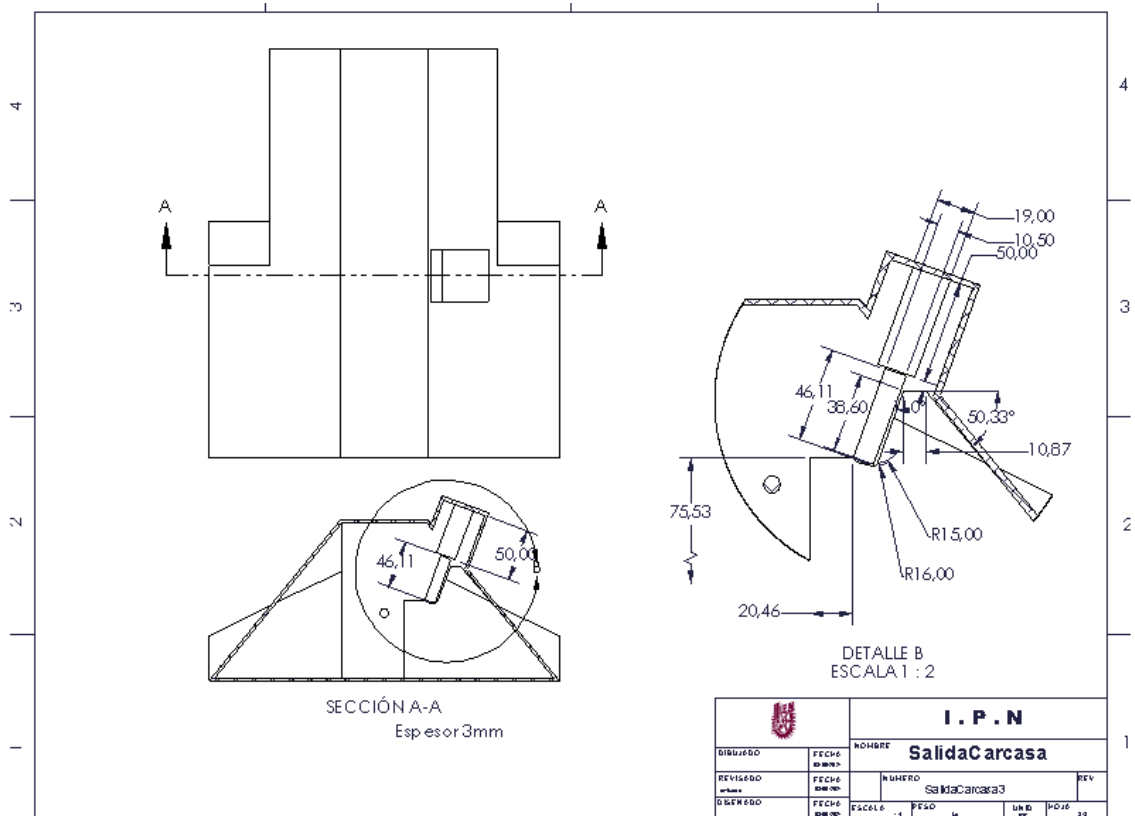
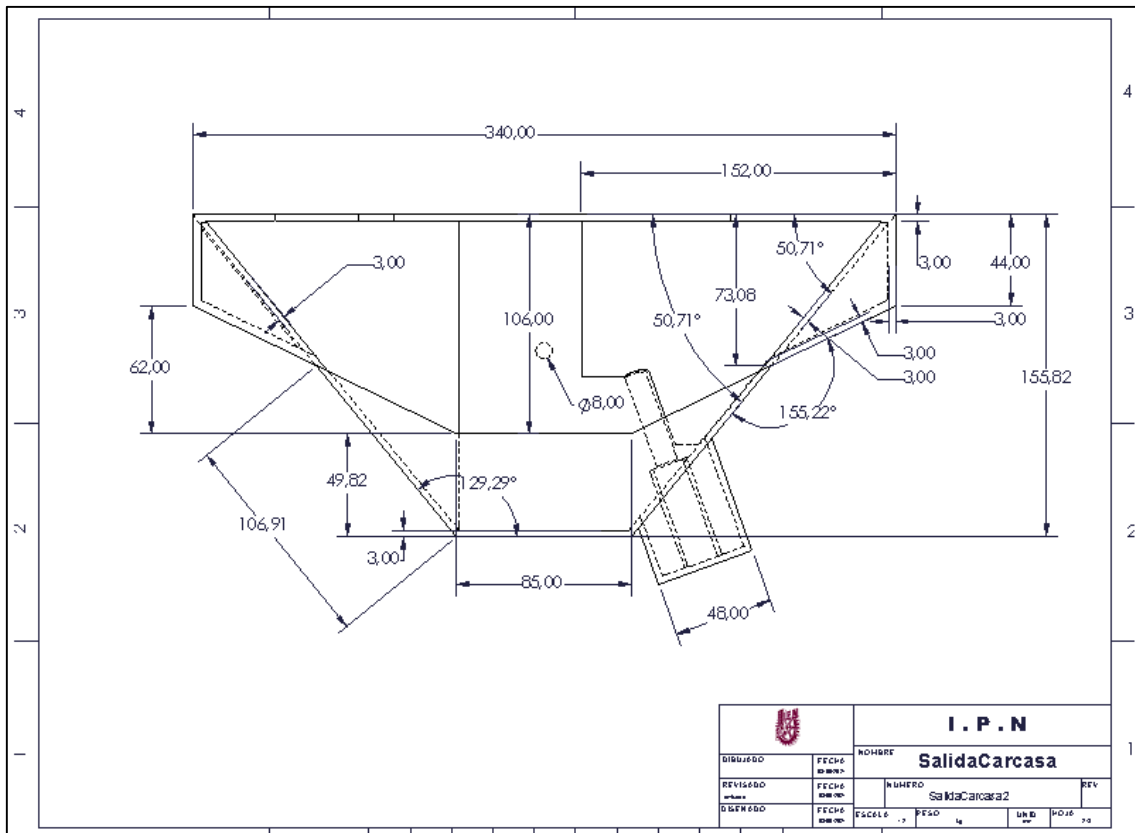


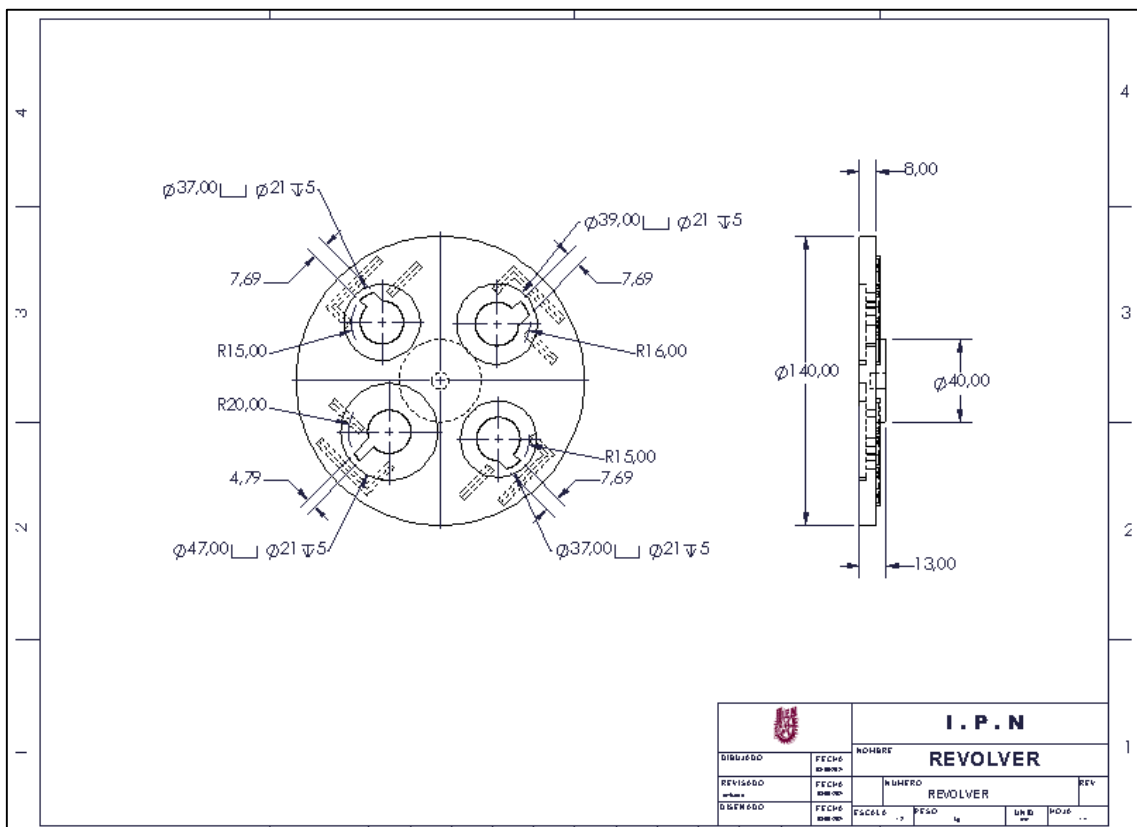
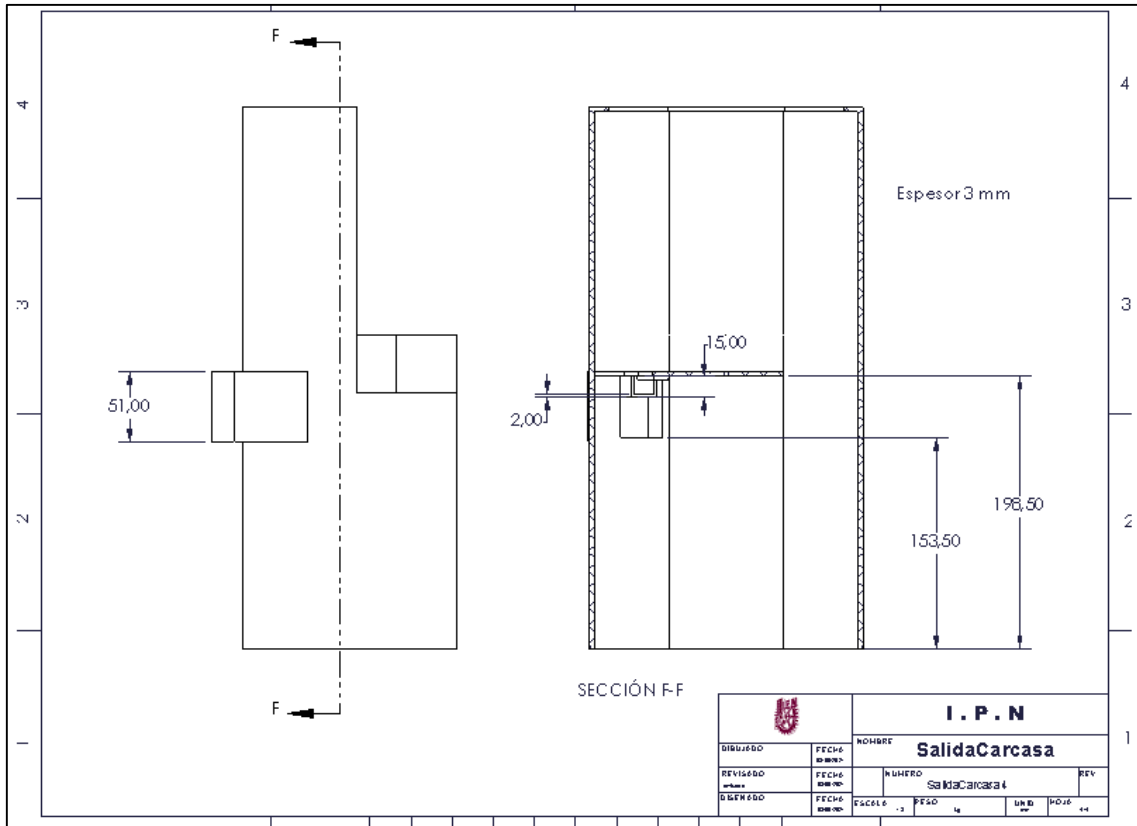




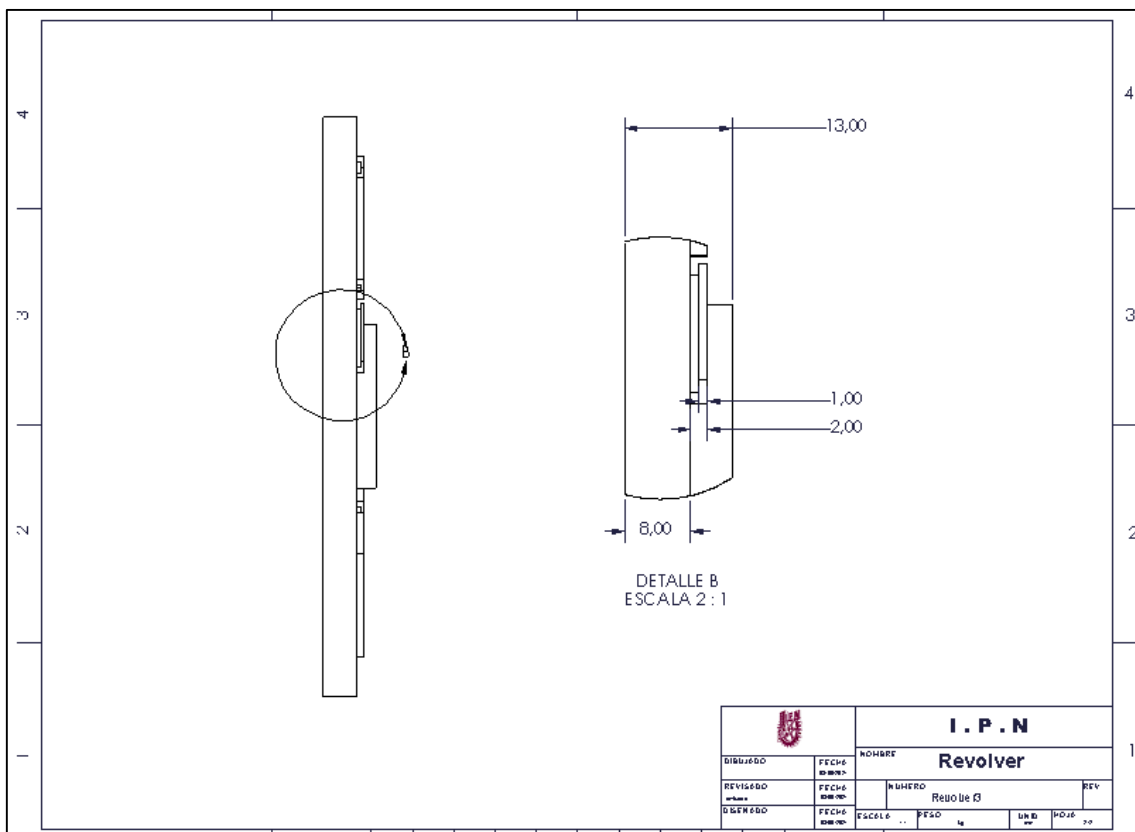
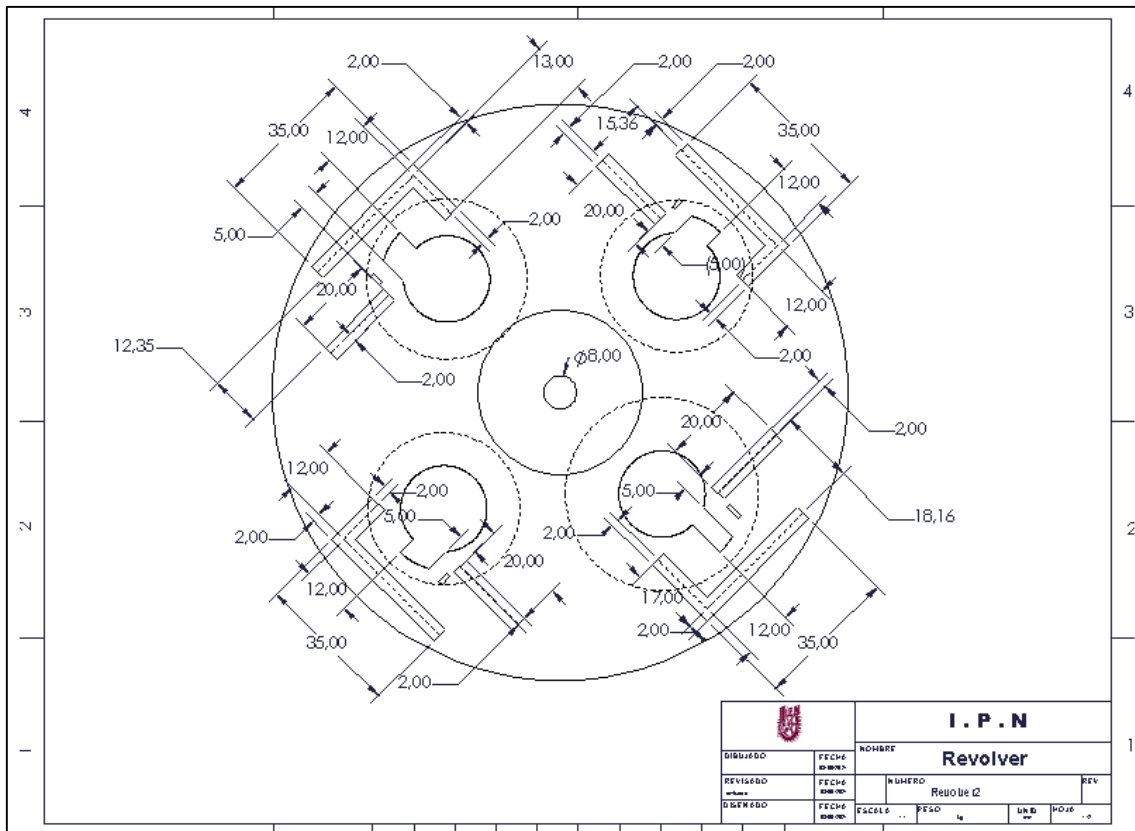


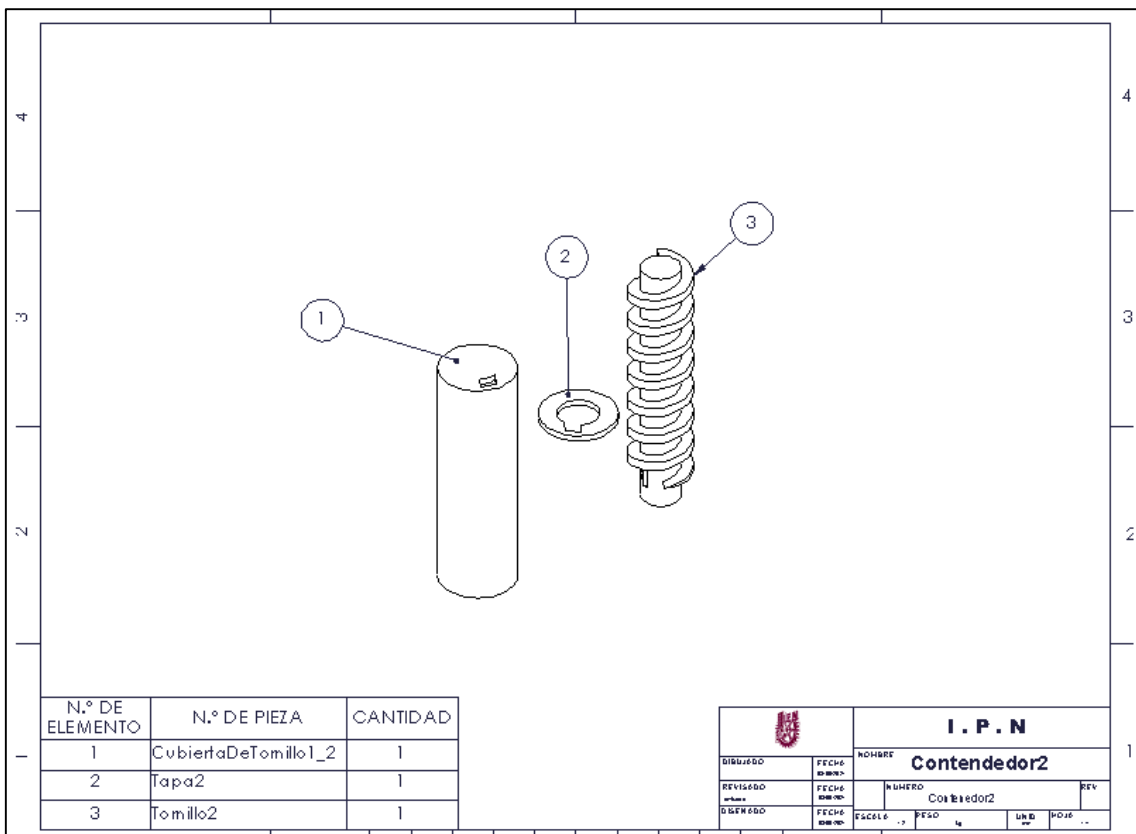
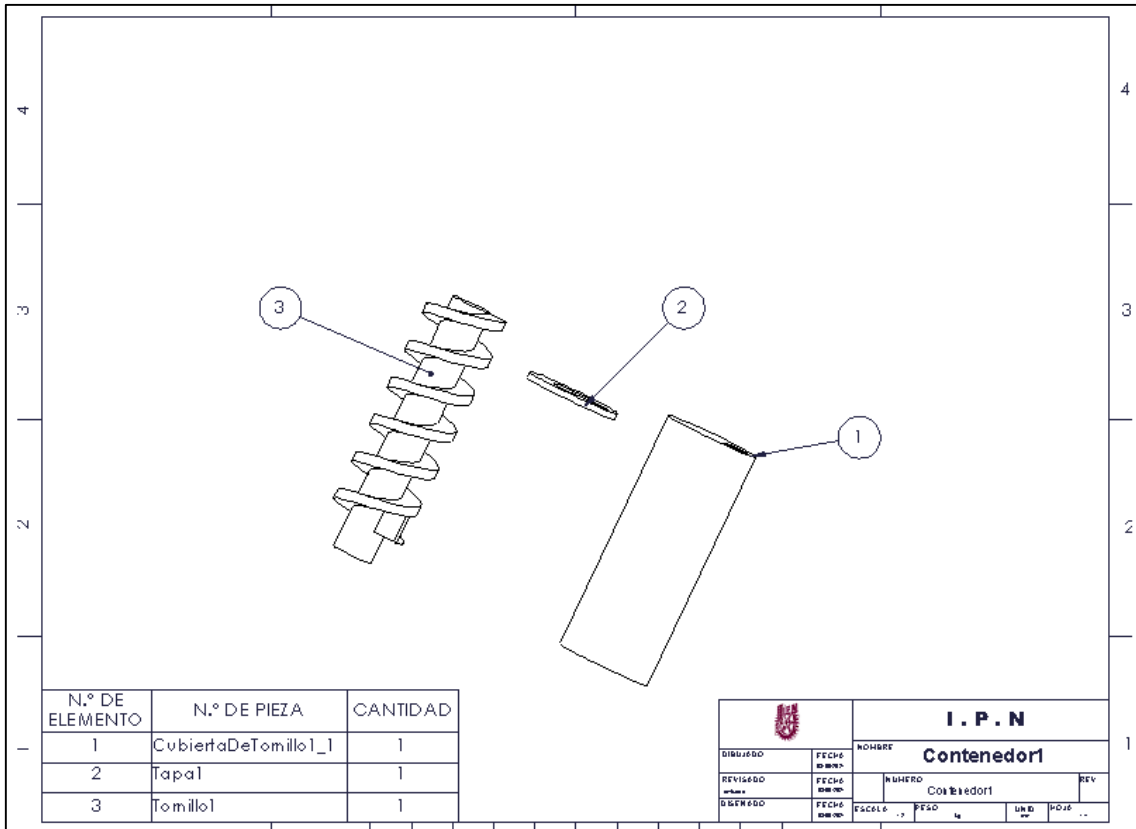


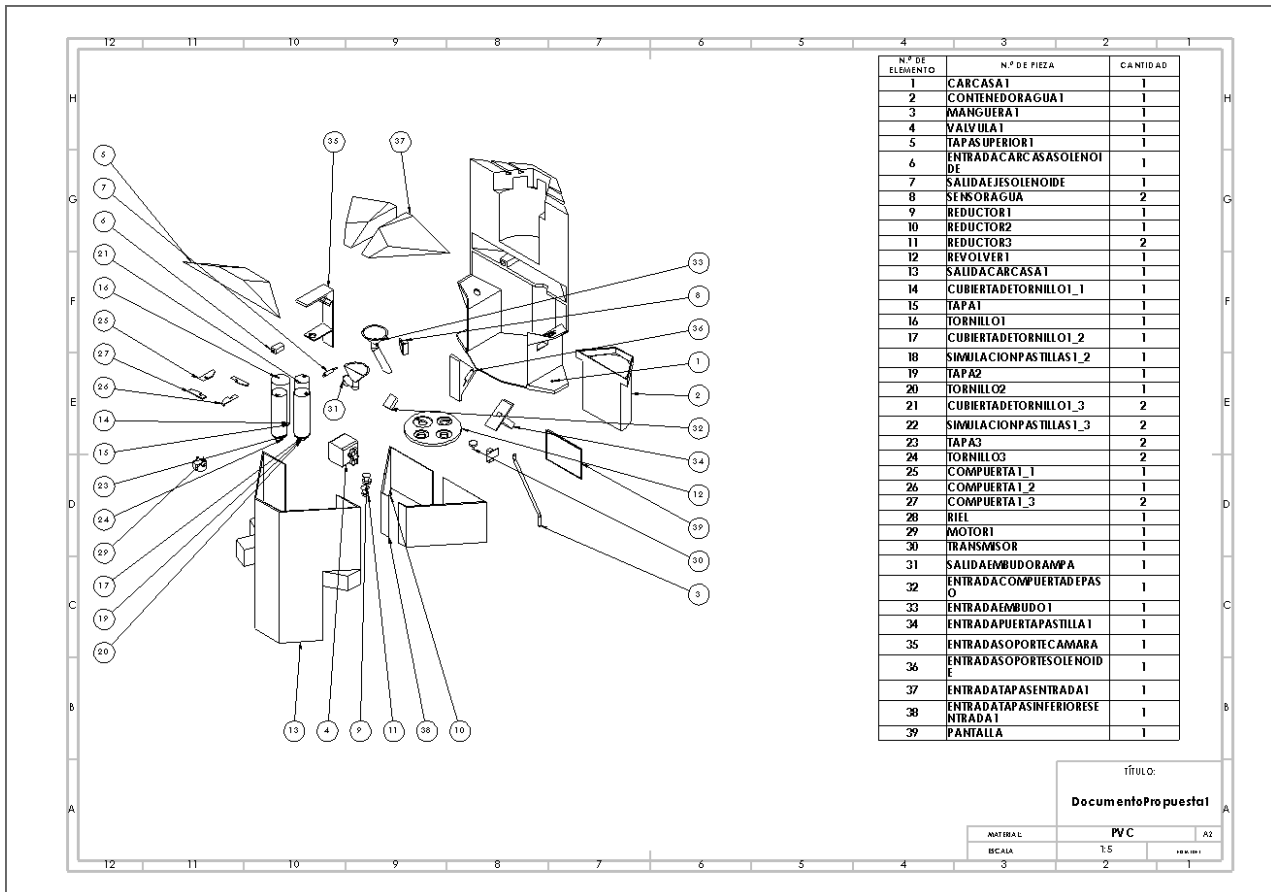
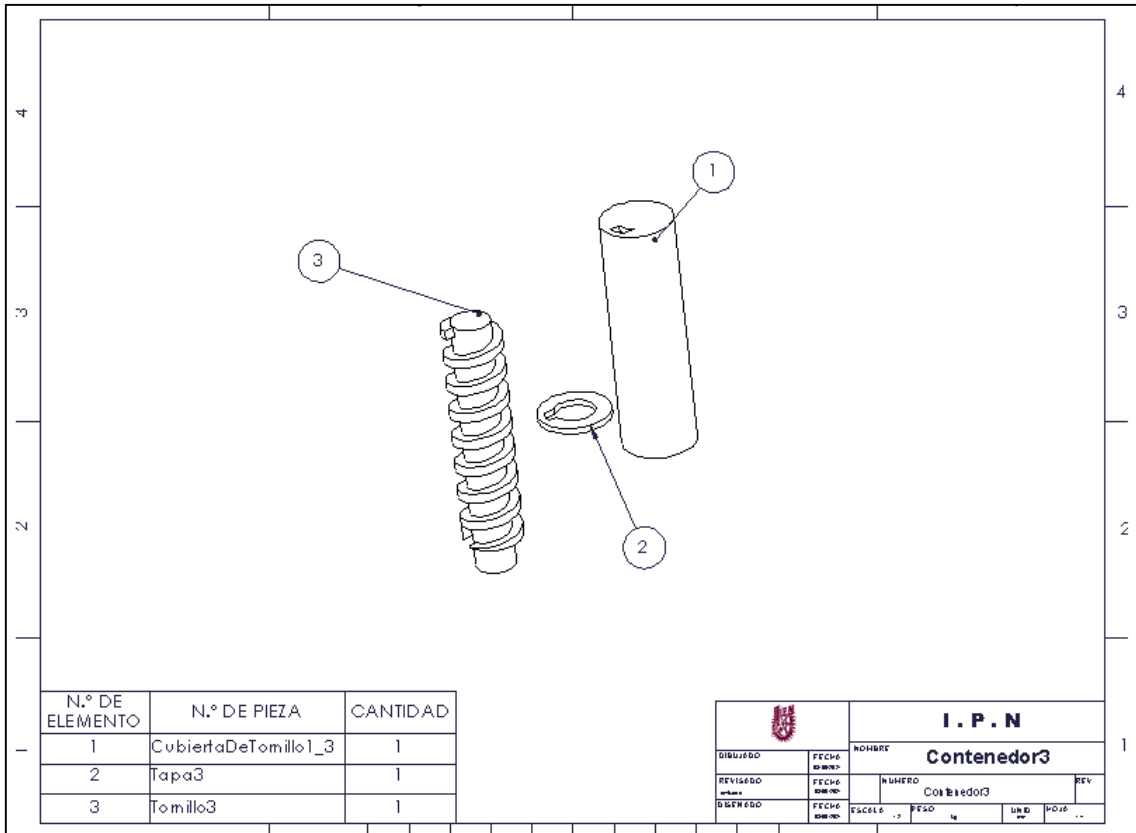


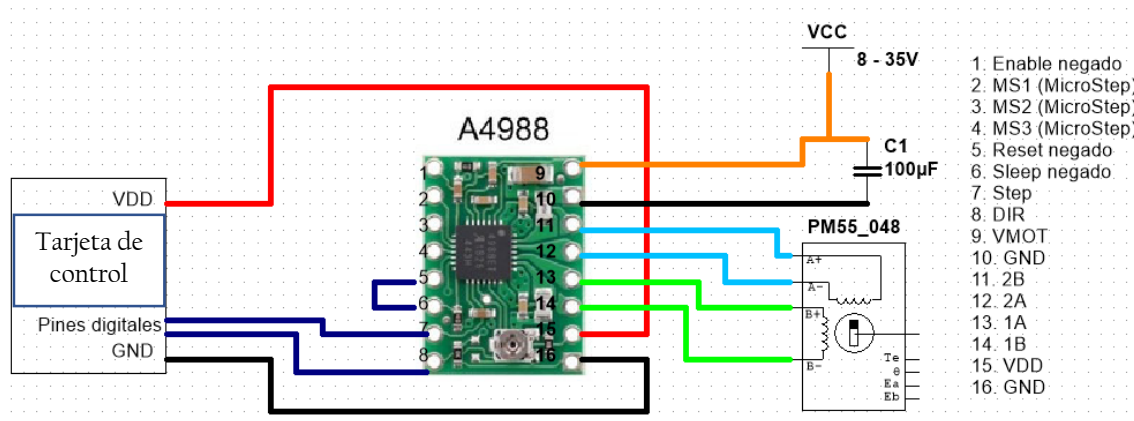
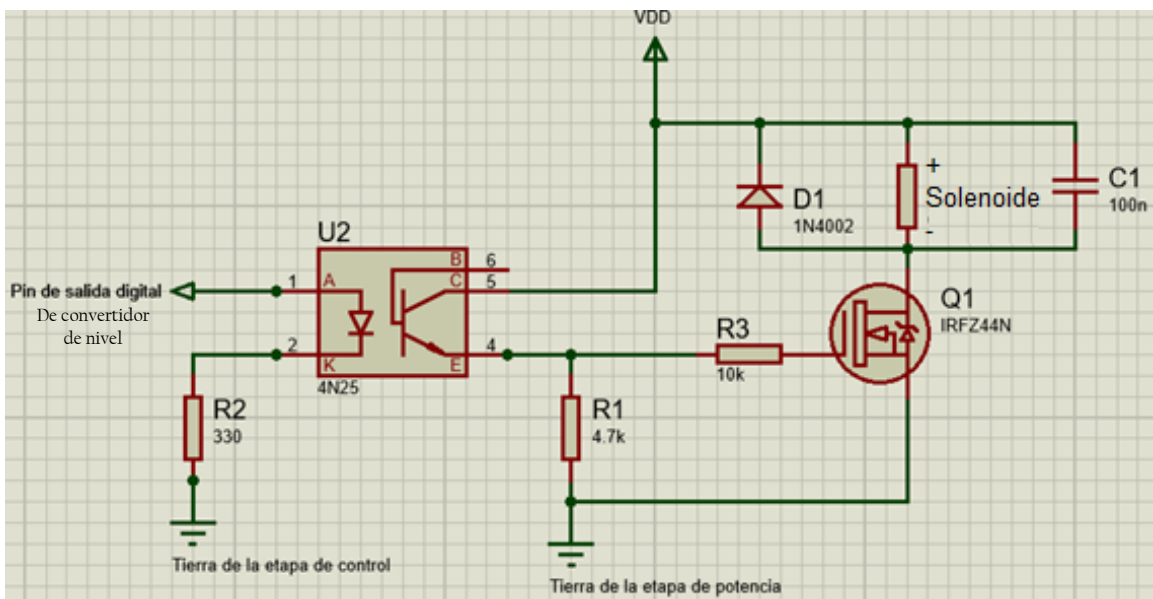
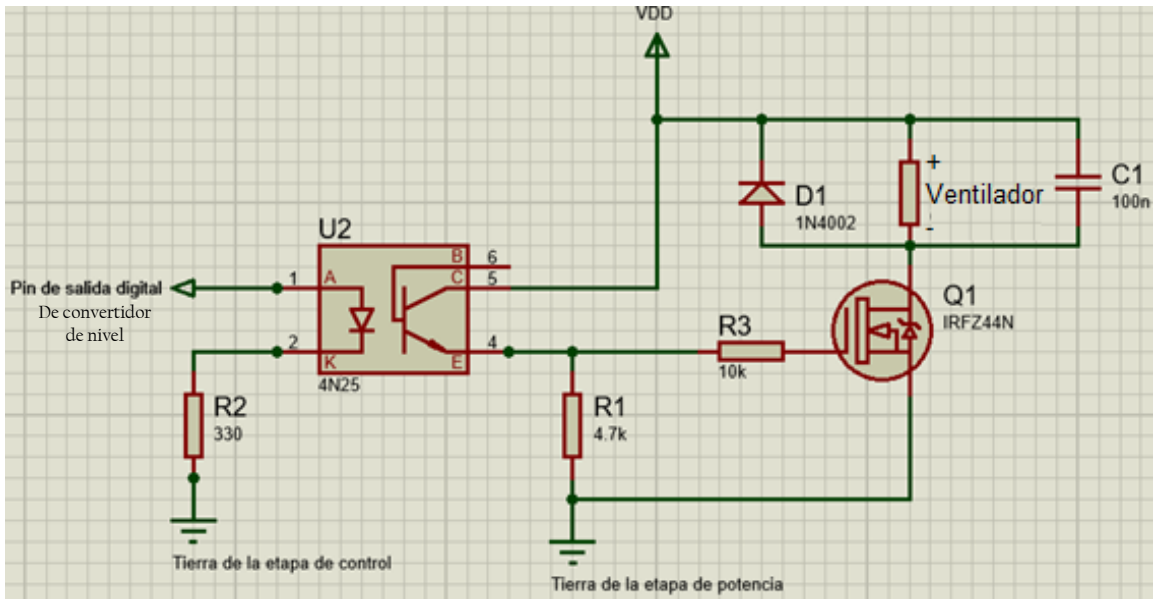


Dispensador de comprimidos y capsulas

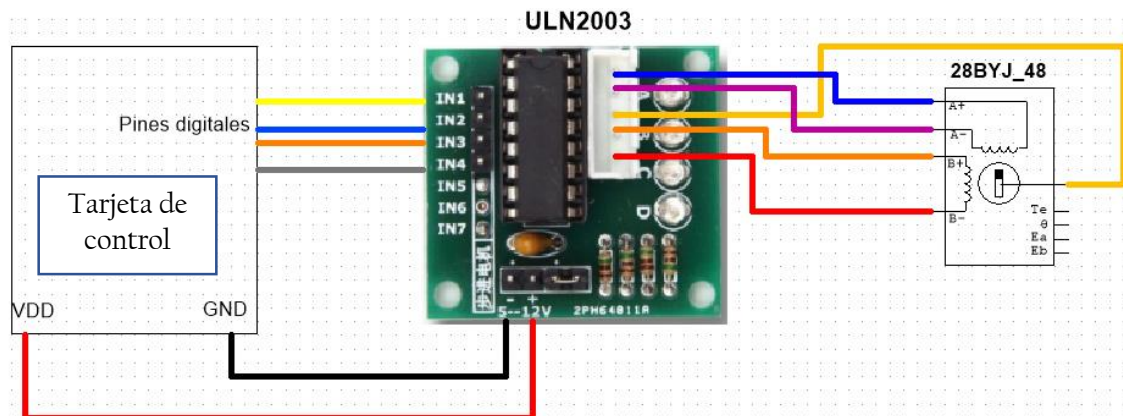








Dispensador de comprimidos y capsulas



Código de programa de la tarjeta de control desarrollado en Python

```
#!/usr/bin/env python
#SEÑALES DE FIREBASE
#- Borrar -->(0-4) se encarga de arrojar todas las pastillas de
un compartimento
#- numPast-->(0-15) contador del número de pastillas en un
contenedor y son 4
#- numPast-->
#- numPast-->
#- numPast-->
#- HumedadIn -->(0-n)-->entrada del valor de la humedad
#- IngreNTrat-->(0-4)-->posicionamiento de entrada
#- ModoViaje -->(0-1)-->saca cantidad de pastillas de todos los
compartimentos
#- NAgua -->(0-1)Nivel de agua del tanque
#- NumTrata -->(0-4)Indica salida de un tratamiento
#- Reabastecer-->(0-4)proceso para introducir pastillas nueva
#- TemperaturaIn-->(0-n) entrada del valor de la temperatura
#- posicion-->(0-1)Indica en que momento el usuario puede meter
pastillas y en
#que momento debe regresar el revolver a home
#COMPONENES FISICOS
#Entradas
#- Sensor de agua (2)
#- Sensor de humedad(1)----->
#Salidas
#- Señal electroválvula
#- Señal motor revolver-----> 48 pasos para una revolución
completa
#- Señal motor transmisor-----> 4096 medios pasos para un
revolución o 64 pasos completos
#- Señal solenoides(3)
#- Señal led
#- Señal alarma
#- Ventiladores (2)----->Señal PWM a 3 valores
diferentes de operación
```



```
#!/usr/bin/env python
#Importa librerias de RPIO para usar con la Raspberry
import RPi.GPIO as GPIO #importa el modulo GPIO

#Importa librerias de firebase
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

#Importa otras librerias :v
import sys
from threading import Thread
import signal
import time
from time import sleep

#Libreria para el sensor
import Adafruit_DTH

GPIO.setmode(GPIO.BOARD)
#Entradas
GPIO.setup(31,GPIO.IN,GPIO.PUD_UP)##GPIO    -->SensorVaso
GPIO.setup(36,GPIO.IN,GPIO.PUD_UP)##GPIO 16  -->SensorTanque
#Salidas
GPIO.setup(11,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 17 --
>Electrovalvula
GPIO.setup(13,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 27 --
>SolenoidT
GPIO.setup(15,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 22 --
>SolenoidE2
GPIO.setup(16,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 23 --
>SolenoidE1
GPIO.setup(18,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 24 -->Led
indicador
GPIO.setup(29,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 5  -->Alarma
GPIO.setup(35,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 19 --
>MotorPasosR1
GPIO.setup(37,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 26 --
>MotorPasosR2
GPIO.setup(38,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 20 --
>MotorPasosR3
GPIO.setup(40,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 21 --
>MotorPasosR4
GPIO.setup(22,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 25 --
>MotorPasosT1
GPIO.setup(24,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 8  --
>MotorPasosT2
GPIO.setup(26,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 7  --
>MotorPasosT3
```



```
GPIO.setup(28,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)##GPIO 1  --
>MotorPasosT4
sensorT=Adafruit_DTH.DTH11
pinT=5                                     ##GPIO 3  --
>SensorHumedadyTemperatura
GPIO.setup(32,GPIO.OUT)##GPIO 12 -->Ventilador1
GPIO.setup(33,GPIO.OUT)##GPIO 13 -->Ventilador2
pwm1=GPIO.PWM(32,500) #Inicializa PWM1
pwm2=GPIO.PWM(33,500)

class Fuego():
    def __init__(self):
        cred =
credentials.Certificate("path/to/serviceAccountKey.json")
        firebase_admin.initialize_app(cred,
                                       {'databaseURL':
'https://elitecarebase-tt2ipn-default-rtdb.firebaseio.com/'})

        self.Banderas = db.reference('BanderasControl')
        self.Borrar = self.Banderas.child('Borrar')
        self.Humedad = self.Banderas.child('HumedadIn')
        self.IngresarT = self.Banderas.child('IngreNTrat')
        self.ModoViaje = self.Banderas.child('ModoViaje')
        self.NivelA = self.Banderas.child('NAGua')
        self.NumeroTratamiento = self.Banderas.child('NumTrata')
        self.Reabastecer = self.Banderas.child('Reabastecer')
        self.Temperatura = self.Banderas.child('TemperaturaIn')
        self.Posicionamiento = self.Banderas.child('posicion')

        self.Tratamientos = db.reference('Tratamientos')
        self.Tratamiento1 =
self.Tratamientos.child('Tratamiento1')
        self.NumeroPastillas1=self.Tratamiento1.child('numPast')
        self.Tratamiento2 =
self.Tratamientos.child('Tratamiento2')
        self.NumeroPastillas2=self.Tratamiento2.child('numPast')
        self.Tratamiento3 =
self.Tratamientos.child('Tratamiento3')
        self.NumeroPastillas3=self.Tratamiento3.child('numPast')
        self.Tratamiento4 =
self.Tratamientos.child('Tratamiento4')
        self.NumeroPastillas4=self.Tratamiento4.child('numPast')

class SistemaAGua():
    def Dispensado(self):
        while(GPIO.input(31)==GPIO.LOW):#Sensor del vaso este en
bajo
            GPIO.output(11,GPIO.HIGH)#Activa la electrovalvula
            print("Llenando vaso")

    def IndicaNivel(self):
```



```
if(GPIO.input(36)==GPIO.LOW):#Si el sensor no detecta
agua
    self.NivelA.set(1)#Manda el valor 1 al firebase
else:
    self.NivelA.set(0)#Manda el valor 0 al firebase

class Actuadores():
    def SolenoideCompuerta():
        GPIO.output(15,GPIO.HIGH)#Activa el solenoideA1
        time.sleep(500)
        GPIO.output(16,GPIO.HIGH)#Activa el solenoideA2
        time.Sleep(500)
        GPIO.output(16,GPIO.LOW)#Desactiva el solenoideA2
        time.sleep(500)
        GPIO.output(15,GPIO.LOW)#Desactiva el solenoideA1

    def ActivaSolTransmisor():
        GPIO.output(13,GPIO.HIGH)#Activa el solenoideT

    def DesactivaSolTransmisor():
        GPIO.output(13,GPIO.LOW)#Activa el solenoideT

class MotorRevolver():
    def MueveNGrados(self,n):
        if(n<=180):
            m=(48*n)/1440
            for i in range(m):
                GPIO.output(35,GPIO.LOW)
                GPIO.output(37,GPIO.LOW)
                GPIO.output(38,GPIO.HIGH)
                GPIO.output(40,GPIO.HIGH)
                time.sleep(0.3)
                GPIO.output(35,GPIO.HIGH)
                GPIO.output(37,GPIO.LOW)
                GPIO.output(38,GPIO.LOW)
                GPIO.output(40,GPIO.HIGH)
                time.sleep(0.3)
                GPIO.output(35,GPIO.HIGH)
                GPIO.output(37,GPIO.HIGH)
                GPIO.output(38,GPIO.LOW)
                GPIO.output(40,GPIO.LOW)
                time.sleep(0.3)
                GPIO.output(35,GPIO.LOW)
                GPIO.output(37,GPIO.HIGH)
                GPIO.output(38,GPIO.HIGH)
                GPIO.output(40,GPIO.LOW)
                time.sleep(0.3)
            self.posicionamiento.set(0)

        else:
            m=(48*(n-180))/1440
            for i in range(m):
```



```
GPIO.output(35,GPIO.LOW)
GPIO.output(37,GPIO.LOW)
GPIO.output(38,GPIO.HIGH)
GPIO.output(40,GPIO.HIGH)
time.sleep(0.3)
GPIO.output(35,GPIO.LOW)
GPIO.output(37,GPIO.HIGH)
GPIO.output(38,GPIO.HIGH)
GPIO.output(40,GPIO.LOW)
time.sleep(0.3)
GPIO.output(35,GPIO.HIGH)
GPIO.output(37,GPIO.HIGH)
GPIO.output(38,GPIO.LOW)
GPIO.output(40,GPIO.LOW)
time.sleep(0.3)
GPIO.output(35,GPIO.HIGH)
GPIO.output(37,GPIO.LOW)
GPIO.output(38,GPIO.LOW)
GPIO.output(40,GPIO.HIGH)
time.sleep(0.3)
self.posicionamiento.set(0)
```

```
class MotorTransmisor():
    def MueveContenedor(self,c):
        if(c==1):
            x0=90
        if(c==2):
            x0=180
        if(c==3):
            x0=270
        if(c==4):
            x0=360
        m=(4096*x0)/2880
        for i in range(m):
            GPIO.output(22,GPIO.HIGH)
            GPIO.output(24,GPIO.LOW)
            GPIO.output(26,GPIO.LOW)
            GPIO.output(28,GPIO.LOW)
            time.sleep(0.3)
            GPIO.output(22,GPIO.HIGH)
            GPIO.output(24,GPIO.HIGH)
            GPIO.output(26,GPIO.LOW)
            GPIO.output(28,GPIO.LOW)
            time.sleep(0.3)
            GPIO.output(22,GPIO.LOW)
            GPIO.output(24,GPIO.HIGH)
            GPIO.output(26,GPIO.LOW)
            GPIO.output(28,GPIO.LOW)
            time.sleep(0.3)
            GPIO.output(22,GPIO.LOW)
            GPIO.output(24,GPIO.HIGH)
```




```
GPIO.output(26,GPIO.HIGH)
GPIO.output(28,GPIO.LOW)
time.sleep(0.3)
GPIO.output(22,GPIO.LOW)
GPIO.output(24,GPIO.LOW)
GPIO.output(26,GPIO.HIGH)
GPIO.output(28,GPIO.LOW)
time.sleep(0.3)
GPIO.output(22,GPIO.LOW)
GPIO.output(24,GPIO.LOW)
GPIO.output(26,GPIO.HIGH)
GPIO.output(28,GPIO.HIGH)
time.sleep(0.3)
GPIO.output(22,GPIO.LOW)
GPIO.output(24,GPIO.LOW)
GPIO.output(26,GPIO.LOW)
GPIO.output(28,GPIO.HIGH)
time.sleep(0.3)
GPIO.output(22,GPIO.HIGH)
GPIO.output(24,GPIO.LOW)
GPIO.output(26,GPIO.LOW)
GPIO.output(28,GPIO.HIGH)
time.sleep(0.3)

class Ventilador():
    def IniciaPWMBajo():
        pwm1.ChangeDutyCycle(70)
        pwm2.ChangeDutyCycle(70)
    def IniciaPWMMedio():
        pwm1.ChangeDutyCycle(85)
        pwm2.ChangeDutyCycle(85)
    def IniciaPWMAlto():
        pwm1.ChangeDutyCycle(100)
        pwm2.ChangeDutyCycle(100)
    def ApagaPWM():
        GPIO.cleanup()

class SensorHyT():
    def MandaStatus(self):
        hum,tem = Adafruit_DHT.read_retry(sensorT,pinT)
        self.Humedad.set(hum)
        self.Temperatura.set(tem)

    def RegulaTemp(self):
        hum1,tem1 = Adafruit_DHT.read_retry(sensorT,pinT)
        V=Ventilador()
        Error=30-hum1
        if(Error>4 and Error<=8):
            V.IniciaPWMBajo()
        if(Error>4 and Error<=8):
            V.IniciaPWMMedio()
        if(Error>8):
```



```
V.IniciaPWMAlto()
if(Error<0):
    V.ApagaPWM()

class Alarma():
    def Secuencial():
        GPIO.output(29,GPIO.HIGH)
        time.sleep(100)
        GPIO.output(29,GPIO.LOW)
        time.sleep(100)
        GPIO.output(29,GPIO.HIGH)
        time.sleep(200)
        GPIO.output(29,GPIO.LOW)
        time.sleep(50)
        GPIO.output(29,GPIO.HIGH)
        time.sleep(500)
        GPIO.output(29,GPIO.LOW)

    def Secuencia2():
        GPIO.output(29,GPIO.HIGH)
        time.sleep(100)
        GPIO.output(29,GPIO.LOW)
        time.sleep(100)
        GPIO.output(29,GPIO.HIGH)
        time.sleep(200)
        GPIO.output(29,GPIO.LOW)
        time.sleep(50)
        GPIO.output(29,GPIO.HIGH)
        time.sleep(500)
        GPIO.output(29,GPIO.LOW)

#Loop en lapsos grandes-----
----->
StatusTanque = SistemaAGua() #Inicializa variable clase sistema
de agua
StatusTanque.IndicaNivel() #Manda el status actual del nivel de
agua
                                #en el tanque
StatusTemperatura = SensorHyT()#Inicializa variable clase
SensorHyT
StatusTemperatura.MandaStatus()#Manda el status de la humedad y
temperatura
StatusTemperatura.RegulaTemp() #Regula la temperatura interna

#Loop en tiempo real-----
----->
FireB=Fuego()#Inicializa las variables actuales

#Proceso de introduccion de un nuevo tratamiento-----
----->
BIT=FireB.IngresarT    #BanderaIndicaTratamiento
BIT=BIT.get()          #Obten valor actual
```



```
BIP=FireB.Posicionamiento #BanderaIndicaPosicionamiento
MotorR=MotorRevolver()#Inicializa variable clase MotorRevolver

if(BIT==1):
    MotorR.MueveNGrados(0) #Se queda en Home y manda un 0 a
posicionamiento
    Flag1=BIP.get() #Bandera para retornar
    if(Flag1 == 1): #Cuando el usuario meta pastillas
        MotorR.MueveNGrados(0)#Regresa a su posicion inicial

if(BIT==2):
    MotorR.MueveNGrados(180) #Rota 180 grados horarios
    Flag1=BIP.get() #Bandera para retornar
    if(Flag1 == 1): #Cuando usuario meta pastillas
        MotorR.MueveNGrados(180)#Regresa a su posicion inicial

if(BIT==3):
    MotorR.MueveNGrados(90) #Rota 90 grados horario
    Flag1=BIP.get() #Bandera para retornar
    if(Flag1 == 1): #Cuando usuario meta pastillas
        MotorR.MueveNGrados(270)#Regresa 90 grados antihorario

if(BIT==4):
    MotorR.MueveNGrados(270) #Rota 90 grados antihorario
    Flag1=BIP.get() #Bandera para retornar
    if(Flag1 == 1): #Cuando usuario meta pastillas
        MotorR.MueveNGrados(90)#Regresa 90 grados horario

#Proceso reabastecimiento de un tratamiento-----
----->
BRT=FireB.Reabastecer #BanderaReabasteceTratamiento
BRT=BRT.get() #Obten valor actual
BIPR=FireB.Posicionamiento #BanderaIndicaPosicionamientoR
MotorD=MotorRevolver() #Inicializa variable clase
MotorRevolver

if(BRT==1):
    MotorD.MueveNGrados(0) #Se queda en Home y manda un 0 a
posicionamiento
    Flag2=BIPR.get() #Bandera para retornar
    if(Flag2 == 1): #Cuando el usuario meta pastillas
        MotorD.MueveNGrados(0)#Regresa a su posicion inicial

if(BRT==2):
    MotorD.MueveNGrados(180) #Rota 180 grados horarios
    Flag2=BIPR.get() #Bandera para retornar
    if(Flag2 == 1): #Cuando usuario meta pastillas
        MotorD.MueveNGrados(180)#Regresa a su posicion inicial

if(BRT==3):
    MotorD.MueveNGrados(90) #Rota 90 grados horario
```



```
Flag2=BIPR.get() #Bandera para retornar
if(Flag2 == 1): #Cuando usuario meta pastillas
    MotorD.MueveNGrados(270)#Regresa 90 grados antihorario

if(BRT==4): #Caso tratamiento contenedor 4
    MotorD.MueveNGrados(270) #Rota 90 grados antihorario
    Flag2=BIPR.get() #Bandera para retornar
    if(Flag2 == 1): #Cuando usuario meta pastillas
        MotorD.MueveNGrados(90)#Regresa 90 grados horario

#Proceso de salida de pastillas-----
----->
BDT=FireB.NumeroTratamiento #BanderaDosificaTratamiento
BDT=BDT.get() #Obten valor actual
MotorG=MotorRevolver() #Inicializa variable clase
MotorRevolver
Actua=Actuadores() #Inicializa variable clase
Actuadores
MotorT=MotorTransmisor() #Inicializa variable clase
MotorTransmisor
DispensaA=SistemaAGua() #Inicializa variable clase
SistemaAgua
Alarm=Alarma() #Inicializa variable clase Alarma

if(BDT==1): #Caso tratamiento contenedor 1
    NAP1=FireB.NumeroPastillas1
    n=(NAP1.get())-1
    NAP1=NAP1.set(n)
    Alarm.Secuencial() #Inicia secuencia de alarma
    DispensaA.Dispensado() #Dispensa agua
    Actua.SolenoideCompuerta() #Dispensa partilla en turno
    MotorG.MueveNGrados(180) #Rota 180 grados a la posicion de
salida
    Actua.ActivaSolTransmisor()#Despliega el motor transmisor
    MotorT.MueveContenedor(1) #Mueve tornillo 1
    Actua.DesactivaSolTransmisor()#Retrae el motor transmisor
    MotorG.MueveNGrados(180) #Regresa a posicion inicial

if(BDT==2): #Caso tratamiento contenedor 2
    NAP2=FireB.NumeroPastillas2
    n=(NAP2.get())-1
    NAP2=NAP2.set(n)
    DispensaA.Dispensado() #Dispensa agua
    Actua.SolenoideCompuerta() #Inicia proceso de dispensado
    MotorG.MueveNGrados(0) #Se queda en su lugar
    Actua.ActivaSolTransmisor()#Despliega el motor transmisor
    MotorT.MueveContenedor(2) #Mueve tornillo 2
    Actua.DesactivaSolTransmisor()#Retrae el motor transmisor
    MotorG.MueveNGrados(180) #Regresa a posicion inicial

if(BDT==3): #Caso tratamiento contenedor 3
```



```
NAP3=FireB.NumeroPastillas3
n=(NAP3.get())-1
NAP3=NAP3.set(n)
DispensaA.Dispensado() #Dispensa agua
Actua.SolenoideCompuerta() #Inicia proceso de dispensado
MotorG.MueveNGrados(90) #Rota 90 grados horario
Actua.ActivaSolTransmisor()#Despliega el motor transmisor
MotorT.MueveContenedor(3) #Mueve tornillo 1
Actua.DesactivaSolTransmisor()#Retrae el motor transmisor
MotorG.MueveNGrados(270) #Regresa a posicion inicial

if(BDT==4): #Caso tratamiento contenedor 4
    NAP4=FireB.NumeroPastillas4
    n=(NAP4.get())-1
    NAP4=NAP4.set(n)
    DispensaA.Dispensado() #Dispensa agua
    Actua.SolenoideCompuerta() #Inicia proceso de dispensado
    MotorG.MueveNGrados(270) #Rota 90 grados antihorario
    Actua.ActivaSolTransmisor()#Despliega el motor transmisor
    MotorT.MueveContenedor(4) #Mueve tornillo 1
    Actua.DesactivaSolTransmisor()#Retrae el motor transmisor
    MotorG.MueveNGrados(90) #Regresa a posicion inicial

#Proceso borrado de un tratamiento-----
----->
BBT=FireB.Borrar #BanderaBorraTratamiento
BBT=BBT.get() #Obten valor del tratamiento a
borrar
MotorB=MotorRevolver() #Inicializa variable clase
MotorRevolver
ActuaB=Actuadores() #Inicializa variable clase
Actuadores
MotorTB=MotorTransmisor() #Inicializa variable clase
MotorTransmisor
Alarm2=Alarma() #Inicializa variable clase Alarma

if(BBT==1): #Caso tratamiento contenedor 1
    NP1=FireB.NumeroPastillas1
    NP1=NP1.get() #Obtiene numero de pastillas
actual
    Alarm2.Secuencia2() #Secuencia de alarma 2
    for i in range(NP1):
        Actua.SolenoideCompuerta() #Dispensa pastilla en turno
        MotorG.MueveNGrados(180) #Rota 180 grados a la
posicion de salida
        Actua.ActivaSolTransmisor()#Despliega el motor
transmisor
        MotorT.MueveContenedor(1) #Mueve tornillo 1
        Actua.DesactivaSolTransmisor()#Retrae el motor
transmisor
        MotorG.MueveNGrados(180) #Regresa a posicion inicial
    NP1=FireB.NumeroPastillas1
```



```
NP1=NP1.set(0) #Actualiza el numero actual de
pastillas

if(BBT==2): #Caso tratamiento contenedor 2
    NP2=FireB.NumeroPastillas2
    NP2=NP2.get() #Obtiene numero de pastillas
actual
    Alarm2.Secuencia2() #Secuencia de alarma 2
    for i in range(NP2):
        Actua.SolenoideCompuerta() #Inicia proceso de dispensado
        MotorG.MueveNGrados(0) #Se queda en su lugar
        Actua.ActivaSolTransmisor() #Despliega el motor
transmisor
        MotorT.MueveContenedor(2) #Mueve tornillo 2
        Actua.DesactivaSolTransmisor() #Retrae el motor
transmisor
        MotorG.MueveNGrados(180) #Regresa a posicion inicial
    NP2=FireB.NumeroPastillas2
    NP2=NP2.set(0) #Actuliza el numero actual de
pastillas

if(BBT==3): #Caso tratamiento contenedor 3
    NP3=FireB.NumeroPastillas3
    NP3=NP3.get() #Obtiene numero de pastillas
actual
    Alarm2.Secuencia2() #Secuencia de alarma 2
    for i in range(NP3):
        Actua.SolenoideCompuerta() #Inicia proceso de dispensado
        MotorG.MueveNGrados(90) #Rota 90 grados horario
        Actua.ActivaSolTransmisor() #Despliega el motor
transmisor
        MotorT.MueveContenedor(3) #Mueve tornillo 1
        Actua.DesactivaSolTransmisor() #Retrae el motor
transmisor
        MotorG.MueveNGrados(270) #Regresa a posicion inicial
    NP3=FireB.NumeroPastillas3
    NP3=NP3.set(0) #Actualiza el numero actual de
pastillas

if(BBT==4): #Caso tratamiento contenedor 4
    NP4=FireB.NumeroPastillas4
    NP4=NP4.get() #Obtiene numero de pastillas
actual
    Alarm2.Secuencia2() #Secuencia de alarma 2
    for i in range(NP4):
        DispensaA.Dispensado() #Dispensa agua
        Actua.SolenoideCompuerta() #Inicia proceso de dispensado
        MotorG.MueveNGrados(270) #Rota 90 grados antihorario
        Actua.ActivaSolTransmisor() #Despliega el motor
transmisor
        MotorT.MueveContenedor(4) #Mueve tornillo 1
```



```
Actua.DesactivaSolTransmisor() #Retrae el motor
transmisor
    MotorG.MueveNGrados(90)      #Regresa a posicion inicial
    NP4=FireB.NumeroPastillas4
    NP4=NP4.set(0)                 #Actualiza el numero de
pastillas
```

Código de los archivos XML de las interfaces de usuario

Ventana de borrar tratamiento

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BorrarTratamientoActive">

    <TextView
        android:id="@+id/TituloBorrarTratamientos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Borrar Tratamientos"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.084" />

    <TextView
        android:id="@+id/InstruccionesBorrarTratamiento"
        android:layout_width="350dp"
        android:layout_height="51dp"
        android:text="Por favor selecciona el tratameinto que deseas
borrar:"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.491"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.154" />

    <Button
        android:id="@+id/BotonBorrarTratamiento"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Borrar tratamiento seleccionado"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.494"
        app:layout_constraintStart_toStartOf="parent"
```




```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.685" />

<Button
    android:id="@+id/BotonSalirBorrarTratamiento"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Salir"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.815" />

<RadioGroup
    android:id="@+id/RadioGroupTratam"
    android:layout_width="177dp"
    android:layout_height="197dp"
    android:layout_marginBottom="120dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <RadioButton
        android:id="@+id/RadioBorrarTratamiento1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Tratamiento 1"
        android:textSize="18sp" />

    <RadioButton
        android:id="@+id/RadioBorrarTratamiento2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Tratamiento 2"
        android:textSize="18sp" />

    <RadioButton
        android:id="@+id/RadioBorrarTratamiento3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Tratamiento3"
        android:textSize="18sp" />

    <RadioButton
        android:id="@+id/RadioBorrarTratamiento4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Tratamiento 4"
        android:textSize="18sp" />

</RadioGroup>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad de kotlin correspondiente



```
package com.example.apptt2

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.RadioGroup
import android.widget.Toast
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase

class BorrarTratamientoActive : AppCompatActivity() {

    //Creando variables para enlazar xml con kotlin
    var botonEliminarTratamiento:Button ?= null
    var botonSalirEliminarTratamiento:Button ?= null
    //var RadioGroupTratamientos:RadioGroup =
    findViewById(R.id.RadioGroupTratam)
    private lateinit var databaseBorrar:DatabaseReference
    private lateinit var databaseBorrar2:DatabaseReference
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_borrar_tratamiento_active)

        //Vinculando botones xml con kotlin

        botonEliminarTratamiento=findViewById(R.id.BotonBorrarTratamiento)

        botonSalirEliminarTratamiento=findViewById(R.id.BotonSalirBorrarTratamiento)

        var tratIDB=" "
        var tratIDBnum=0
        val RadioGroupTratamientos:RadioGroup =
        findViewById(R.id.RadioGroupTratam)

        //Evento del RadioGroup de seleccion de tratamiento
        RadioGroupTratamientos.setOnCheckedChangeListener { group,
        checkedId ->
            when (checkedId) {
                R.id.RadioBorrarTratamiento1 -> {
                    tratIDB="Tratamiento1"
                    tratIDBnum=1
                    Toast.makeText(this,"Eliminaras Tratamiento
1",Toast.LENGTH_SHORT).show() }
                R.id.RadioBorrarTratamiento2 -> {
                    tratIDB="Tratamiento2"
                    tratIDBnum=2
                    Toast.makeText(this,"Eliminaras Tratamiento
2",Toast.LENGTH_SHORT).show() }
                R.id.RadioBorrarTratamiento3 -> {
                    tratIDB="Tratamiento3"
                    tratIDBnum=3
                    Toast.makeText(this,"Eliminaras Tratamiento
3",Toast.LENGTH_SHORT).show() }
                R.id.RadioBorrarTratamiento4 -> {
                    tratIDB="Tratamiento4"
                    tratIDBnum=4
                    Toast.makeText(this,"Eliminaras Tratamiento
```



```
4", Toast.LENGTH_SHORT).show() }
        } //fin del when
    } //fin evento Radio Group

    //Eventos OnClick de los botones

    botonEliminarTratamiento!!.setOnClickListener {
        val intentToAlarmas=Intent(this, AlarmasSCTr::class.java)
        //Eliminando registros de firebase empleando identificador
deregistro TratName
        val BorrarUserData=Tratamientos(0,0,"No
disponible",0,false)
        if (tratIDB.isNotBlank()) {

databaseBorrar2=FirebaseDatabase.getInstance().getReference("BanderasC
ontrol")

databaseBorrar=FirebaseDatabase.getInstance().getReference("Tratamient
os")

databaseBorrar.child(tratIDB).setValue(BorrarUserData).addOnSuccessLis
tener {

databaseBorrar2.child("Borrar").setValue(tratIDBnum).addOnSuccessLis
tener {

            Toast.makeText(this, "$tratIDB eliminado",
Toast.LENGTH_SHORT).show()
                //startActivity(intentToAlarmas)
                }.addOnFailureListener {Toast.makeText(this,
"Intente mas tarde", Toast.LENGTH_SHORT).show() }
                }.addOnFailureListener {
                    Toast.makeText(this, "Intente mas tarde",
Toast.LENGTH_SHORT).show()
                }
            }else{
                Toast.makeText(this, "seleccione un tratamiento",
Toast.LENGTH_SHORT).show()
            }
        }
        botonSalirEliminarTratamiento!!.setOnClickListener {
            //Volver a menu principal
            val BorrarToMainIntent:Intent=Intent(this,
MainActivity::class.java)
            startActivity(BorrarToMainIntent)
        }
    }
}
}
```

Ventana de fin de ingreso de medicamentos

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
```



```
tools:context=".FinIngresoPastillas">

<TextView
    android:id="@+id/txtFinPas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Elite Care"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.089" />

<TextView
    android:id="@+id/txtFinPasBod"
    android:layout_width="269dp"
    android:layout_height="82dp"
    android:text="Porfavor, espera a que Elite Care acomode el
cotenedor de medicamentos"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.246" />

<TextView
    android:id="@+id/txtFinPasAlert"
    android:layout_width="353dp"
    android:layout_height="50dp"
    android:text="Acomodando Contenedor"
    android:textAlignment="center"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.465" />

<Button
    android:id="@+id/bFinInPas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Finalizar ingreso de pastillas"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.818" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad correspondiente de kotlin

```
package com.example.apptt2
```



```
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.core.view.isVisible
import com.google.firebase.database.*

class FinIngresoPastillas : AppCompatActivity() {
    var btnFIpas:Button?=null
    private lateinit var databasefin:DatabaseReference
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_fin_ingreso_pastillas)
        btnFIpas=findViewById(R.id.bFinInPas)
        btnFIpas!!.isVisible=false
        val txtAlerta:TextView=findViewById(R.id.txtFinPasAlert)
        var txtlata="Colocando el contenedor adecuado"
        txtAlerta.text=txtlata
        var flag=0

        databasefin=FirebaseDatabase.getInstance().getReference("BanderasControl")
        databasefin.child("posiciona").addValueEventListener(object
: ValueEventListener{
            override fun onDataChange(snapshot: DataSnapshot) {
                flag=(snapshot.value).toString().toInt()
                if(flag==1){
                    txtlata="Ingresa las pastillas y, al finalizar,
presiona listo"
                    txtAlerta.text=txtlata
                    btnFIpas!!.isVisible=true
                    Toast.makeText(this@FinIngresoPastillas, "Ingresar
pastillas", Toast.LENGTH_SHORT).show()
                }
            }

            override fun onCancelled(error: DatabaseError) {
                Toast.makeText(this@FinIngresoPastillas, "Error
cancel", Toast.LENGTH_SHORT).show()
            }
        })

        btnFIpas!!.setOnClickListener {
            databasefin.child("posiciona").setValue(0).addOnSuccessListener {
                Toast.makeText(this, "Gracias!",
                Toast.LENGTH_SHORT).show()
            }
            val intFinToMain=Intent(this,MainActivity::class.java)
            startActivity(intFinToMain)
        }
    }
}
```



Ventana de gestionar tratamientos

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".GestionarTratamientosActive">

    <Button
        android:id="@+id/BorrarTratamiento"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Borrar un Tratamiento"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.284" />

    <Button
        android:id="@+id/ReabastecerTratamiento"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Reabastecer un Tramaiento"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.409" />

    <Button
        android:id="@+id/ModificarTratamiento"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Modificar un Tratamiento"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.519" />

    <TextView
        android:id="@+id/TituloGestinartratamientos1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Que deaseas Realizar?"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.16" />
```



```
<Button
    android:id="@+id/BotonSalirGesTratam"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Salir"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.733" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad correspondiente de kotlin

```
package com.example.apptt2

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button

class GestionarTratamientosActive : AppCompatActivity() {
    //Creando variables tipo boton
    var botonBorrarTratamiento : Button ?= null
    var botonGestionarTratamiento : Button ?= null
    var botonReabastecerTratamiento: Button ?= null
    var botonSalirGesTratamiento: Button ?= null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_gestionar_tratamientos_active)

        //Conectando botones del xml a kotlin
        botonBorrarTratamiento=findViewById(R.id.BorrarTratamiento)
        botonGestionarTratamiento=findViewById(R.id.ModificarTratamiento)
        botonReabastecerTratamiento=findViewById(R.id.ReabastecerTratamiento)
        botonSalirGesTratamiento=findViewById(R.id.BotonSalirGesTratam)

        //Eventos Onclick para cambiar a las respectivas ventanas de
        las opciones
        botonBorrarTratamiento!!.setOnClickListener {
            val BorrarTratIntent:Intent= Intent(this,
            BorrarTratamientoActive::class.java)
            startActivity(BorrarTratIntent)
        }
        botonGestionarTratamiento!!.setOnClickListener {
            val GestTratIntent:Intent=Intent(this,
            ModificarTratamientoActive::class.java)
            startActivity(GestTratIntent)
        }
        botonReabastecerTratamiento!!.setOnClickListener {
            val ReabTratIntent:Intent=Intent(this,
            ReabastecerTratamientoActive::class.java)
            startActivity(ReabTratIntent)
        }
    }
}
```




```
    }  
    botonSalirGesTratamiento!!.setOnClickListener {  
        val GesToMainIntent:Intent=Intent(this,  
MainActivity::class.java)  
        startActivity(GesToMainIntent)  
    }  
}  
}
```

Ventana de ingesar nuevo tratamiento

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:app="http://schemas.android.com/apk/res-auto"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
tools:context=".IngresarNuevoTratamientoActive">  
  
    <EditText  
        android:id="@+id/ETNombreTratamiento"  
        android:layout_width="219dp"  
        android:layout_height="56dp"  
        android:ems="10"  
        android:hint="Diclofenaco"  
        android:inputType="textPersonName"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="0.916"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintVertical_bias="0.194" />  
  
    <TextView  
        android:id="@+id/NombreTratamiento"  
        android:layout_width="144dp"  
        android:layout_height="58dp"  
        android:text="Nombre del Tratamiento:"  
        android:textSize="16sp"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="0.059"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintVertical_bias="0.194" />  
  
    <EditText  
        android:id="@+id/ETNumeroPastillas"  
        android:layout_width="186dp"  
        android:layout_height="57dp"  
        android:ems="10"  
        android:hint="8"  
        android:inputType="number"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="0.924"  
        app:layout_constraintStart_toStartOf="parent"
```



```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.326" />

<TextView
    android:id="@+id/NumeroPastillas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Numero de Pastillas:"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.109"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.332" />

<EditText
    android:id="@+id/ETDiasTratamiento"
    android:layout_width="184dp"
    android:layout_height="52dp"
    android:ems="10"
    android:hint="7"
    android:inputType="number"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.925"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.477" />

<TextView
    android:id="@+id/DiasTratamiento"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Cuandos dias durará?:"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.072"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.48" />

<Button
    android:id="@+id/GuardarTratamiento"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Guardar Tratamiento"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.881" />

<androidx.appcompat.widget.SwitchCompat
    android:id="@+id/TratamientoPermanente"
    android:layout_width="209dp"
```



```
android:layout_height="58dp"
android:text="Es un tratamiento permanente?"
android:textSize="16sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.089"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.772" />

<TextView
    android:id="@+id/TituloIngresarTratamiento"
    android:layout_width="362dp"
    android:layout_height="57dp"
    android:text="Por favor, ingresa la información sobre tu
tratamiento:"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.489"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.065" />

<EditText
    android:id="@+id/ETNperiodo"
    android:layout_width="189dp"
    android:layout_height="49dp"
    android:ems="10"
    android:hint="8"
    android:inputType="number"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.95"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.627" />

<TextView
    android:id="@+id/PeriodoTiempoNuevo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Cada cuantas horas?"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.103"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.62" />

<Button
    android:id="@+id/BotonSalirNuevoT"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Salir"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```



```
app:layout_constraintHorizontal_bias="0.504"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.976" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad correspondiente de Kotlin

```
package com.example.apptt2  
  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import android.widget.Switch  
import android.widget.Toast  
import androidx.appcompat.widget.SwitchCompat  
import com.google.firebase.database.DatabaseReference  
import com.google.firebase.database.FirebaseDatabase  
  
class IngresarNuevoTratamientoActive : AppCompatActivity() {  
    //Creando variables para conectar xml con kotlin  
    var botonGuardar: Button? = null  
    //var switchPermanente: SwitchCompat? = null  
    var bSalirNT: Button? = null  
    private lateinit var databaseNTata: DatabaseReference //Referencia a  
base de datos en firebase  
    private lateinit var databaseNTata2: DatabaseReference //Referencia  
a base de datos en firebase  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
  
        setContentView(R.layout.activity_ingresar_nuevo_tratamiento_active)  
  
        //vinculando el botones con kotlin  
        botonGuardar = findViewById(R.id.GuardarTratamiento)  
        //switchPermanente = findViewById(R.id.TratamientoPermanente)  
        bSalirNT = findViewById(R.id.BotonSalirNuevoT)  
        val NTrat: EditText = findViewById(R.id.ETNombreTratamiento)  
        val NumPas: EditText = findViewById(R.id.ETNumeroPastillas)  
        val NDias: EditText = findViewById(R.id.ETDiasTratamiento)  
        val NTime: EditText = findViewById(R.id.ETNperiodo)  
        val  
NTratPerma: SwitchCompat = findViewById(R.id.TratamientoPermanente)  
        val posRevoInt: Intent = intent  
        val posRevoS = posRevoInt.getIntExtra("revolverPos", 0)  
        var nomTratID = "Tratamiento1"  
  
        //Evetos de los botones para guardar y salir de la activity  
        botonGuardar!!.setOnClickListener {  
            //obteniendo datos ingresados  
            val nombreTramieto: String = NTrat.text.toString()  
            val numeroPastillas: Int = NumPas.text.toString().toInt()  
            val diasTratamiento: Int = NDias.text.toString().toInt()  
            val periodoIngesta: Int = NTime.text.toString().toInt()  
            val trataPermanente: Boolean = NTratPerma.isChecked
```



```
//Asignando ID de almacenamiento del Tratamiento
when (posRevoS) {
    1 -> nomTratID = "Tratamiento1"
    2 -> nomTratID = "Tratamiento2"
    3 -> nomTratID = "Tratameinto3"
    4 -> nomTratID = "Tratamiento4"
    else -> {
        Toast.makeText(this, "fallo critico",
Toast.LENGTH_SHORT).show()
        finish()
    }
}
//Enlazando a la base de datos en el nodo de tratamientos
databaseNTata =
FirebaseDatabase.getInstance().getReference("Tratamientos")
if ((nombreTramiento.isNotBlank()) and
(NumPas.text.toString().isNotBlank()) and
(NDias.text.toString().isNotBlank()) and
(NTime.text.toString().isNotBlank())
and
(numeroPastillas>0)and(diasTratamiento>0)and(periodoIngesta>0)){
//Enlazando a la base de datos en el nodo de
tratamientos
databaseNTata =
FirebaseDatabase.getInstance().getReference("Tratamientos")
//Creando objeto tratamiento con los atributos
ingresados por el usuario
val tratamientoGuardar = Tratamientos(
    diasTratamiento,
    periodoIngesta,
    nombreTramiento,
    numeroPastillas,
    trataPermanente)
//Guardando datos del tratamiento en firebase
databaseNTata.child(nomTratID).setValue(tratamientoGuardar).addOnSucce
ssListener {
    Toast.makeText(this, "Tratamiento Guardado",
Toast.LENGTH_SHORT).show()
//Activando bandera de ingreso de tratamientos
para raspberry
databaseNTata2 =
FirebaseDatabase.getInstance().getReference("BanderasControl")
databaseNTata2.child("IngreNTrat").setValue(posRevoS).addOnSuccessList
ener {
    Toast.makeText(this, "Ingrese pastillas",
Toast.LENGTH_SHORT).show()
val intentToAlarmas: Intent = Intent(this,
AlarmasSCTr::class.java)
intentToAlarmas.putExtra("revolverPos",
posRevoS)
//startActivity(intentToAlarmas)
}.addOnFailureListener {
    Toast.makeText(this, "Fallo Control",
Toast.LENGTH_SHORT).show()
}
}.addOnFailureListener {
```



```
        Toast.makeText(this, "Fallo de conexión",
Toast.LENGTH_SHORT).show()
    }
    }else{
        Toast.makeText(this, "faltan datos!",
Toast.LENGTH_SHORT).show()
    }
    }
    bSalirNT!!.setOnClickListener {
        val
ingresarToMenuIntent:Intent=Intent(this,MainActivity::class.java)
        startActivity(ingresarToMenuIntent)
    }
}
}
```

Ventana de menú principal

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <Button
android:id="@+id/boton1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Ingresar Nuevo Tratamiento"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.26" />

    <Button
android:id="@+id/boton2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Gestionar tratamientos"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.383" />

    <Button
android:id="@+id/boton3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Monitorear EliteCare"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```



```
app:layout_constraintVertical_bias="0.499" />

<Button
    android:id="@+id/boton4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Modo Viaje"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.625" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Bienvenido, ¿Que deseas realizar?"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.495"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.166" />

<Button
    android:id="@+id/BotonSalirMain"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Salir"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.822" />

<TextView
    android:id="@+id/TituloMain1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Elite Care"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.119" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad de Kotlin correspondiente

```
package com.example.apptt2
```




```
import android.app.NotificationChannel
import android.app.NotificationManager
import android.content.Intent
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase

class MainActivity : AppCompatActivity() {

    //Creando variables tipo buttton para enlazarlas a los botones del
xml
    var btn: Button ?= null
    var btn2 : Button ?= null
    var btn3: Button ?= null
    var btn4 : Button ?= null
    var btn5: Button ?= null
    //private lateinit var databasep:DatabaseReference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //Creando canales para notificaciones
        craateNotificationChanel1()
        craateNotificationChanel2()
        //Enlazando los botones del XML con Kotlin
        btn=findViewById(R.id.boton1)
        btn2=findViewById(R.id.boton2)
        btn3=findViewById(R.id.boton3)
        btn4=findViewById(R.id.boton4)
        btn5=findViewById(R.id.BotonSalirMain)

        //Eventos OnClick Para cambiar de activities (ventanas)
        btn!!.setOnClickListener{
            //variable intent para cambio de actividad
            val
SelecPastIntent:Intent=Intent(this,SeleccionarTipoDePastilla::class.java)

            //iniciando la actividad correspondiente
            startActivity(SelecPastIntent)
        }
        btn2!!.setOnClickListener{
            //variable intent para cambio de actividad
            val
GesTratIntent:Intent=Intent(this,GestionarTratamientosActive::class.java)

            //iniciando la actividad correspondiente
            startActivity(GesTratIntent)
        }
        btn3!!.setOnClickListener{
            //variable intent para cambio de actividad
            val
MonitoreoIntent:Intent=Intent(this,MonitorearEliteCare::class.java)

            //iniciando la actividad correspondiente
            startActivity(MonitoreoIntent)
        }
    }
}
```



```
}
btn4!!.setOnClickListener{
    //variable intent para cambio de actividad
    val MViajeIntent:Intent=Intent(this,ModoViaje::class.java)
    //iniciando la actividad correspondiente
    startActivity(MViajeIntent)
}
btn5!!.setOnClickListener{
    //sale de la aplicación
    finish()
}

}

private fun craateNotificationChanel2() {
    if (Build.VERSION.SDK_INT>=Build.VERSION_CODES.O) {
        val name:CharSequence="EliteCareOmissionChanel"
        val description="Canal para omisiones"
        val importance=NotificationManager.IMPORTANCE_HIGH
        val canal2=NotificationChannel("ECareCh2",name,importance)
        canal2.description=description
        val
notificationManager=getSystemService(NotificationManager::class.java)
        notificationManager.createNotificationChannel(canal2)
    }
}

private fun craateNotificationChanel1() {
    if (Build.VERSION.SDK_INT>=Build.VERSION_CODES.O) {
        val name:CharSequence="EliteCareReminderChanel"
        val description="Canal para alarmas"
        val importance=NotificationManager.IMPORTANCE_HIGH
        val canal=NotificationChannel("ECareCh",name,importance)
        canal.description=description
        val
notificationManager=getSystemService(NotificationManager::class.java)
        notificationManager.createNotificationChannel(canal)
    }
}

}
}
```

Actividad de modificar un tratamiento

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ModificarTratamientoActive">

    <TextView
        android:id="@+id/IndicacionesModTrata2"
        android:layout_width="332dp"
```



```
        android:layout_height="56dp"
        android:text="Realiza las modificaciones necesarias en los
campos siguientes:"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.493"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.527" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Cada cuantas horas?"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.136"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.789" />

<Button
    android:id="@+id/BotonSalirModTratamientos"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Salir"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.12"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.986" />

<EditText
    android:id="@+id/ModIngP"
    android:layout_width="168dp"
    android:layout_height="54dp"
    android:ems="10"
    android:hint="6"
    android:inputType="number"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.897"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.802" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Dias de Duracion?"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```



```
app:layout_constraintHorizontal_bias="0.123"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.704" />  
  
<EditText  
    android:id="@+id/ModDiasTr"  
    android:layout_width="168dp"  
    android:layout_height="52dp"  
    android:ems="10"  
    android:hint="5"  
    android:inputType="number"  
    android:textSize="18sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.897"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.709" />  
  
<androidx.appcompat.widget.SwitchCompat  
    android:id="@+id/ModPermanenteSw"  
    android:layout_width="300dp"  
    android:layout_height="56dp"  
    android:text="Es un tratamiento permanente?"  
    android:textSize="18sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.378"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.902" />  
  
<Button  
    android:id="@+id/GuardarModificacionesTratam"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Guardar Cambios"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.93"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.986" />  
  
<RadioGroup  
    android:id="@+id/RadioModTratam"  
    android:layout_width="183dp"  
    android:layout_height="197dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.174">  
  
<RadioButton  
    android:id="@+id/RadioModTratal"  
    android:layout_width="match parent"
```



```
        android:layout_height="wrap_content"
        android:text="Tratamiento 1"
        android:textSize="18sp"
        tools:layout_editor_absoluteX="114dp"
        tools:layout_editor_absoluteY="167dp" />

<RadioButton
    android:id="@+id/RadioModTrata2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tratamiento 2"
    android:textSize="18sp"
    tools:layout_editor_absoluteX="114dp"
    tools:layout_editor_absoluteY="215dp" />

<RadioButton
    android:id="@+id/RadioModTrata3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tratamiento 3"
    android:textSize="18sp"
    tools:layout_editor_absoluteX="114dp"
    tools:layout_editor_absoluteY="263dp" />

<RadioButton
    android:id="@+id/RadioModTrata4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tratamiento 4"
    android:textSize="18sp"
    tools:layout_editor_absoluteX="114dp"
    tools:layout_editor_absoluteY="311dp" />
</RadioGroup>

<TextView
    android:id="@+id/IndicacionesModTratamiento"
    android:layout_width="336dp"
    android:layout_height="47dp"
    android:text="Por favor, selecciona el tratameinto que deseas
modificar:"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.56"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.055" />

<TextView
    android:id="@+id/TituloModificarTratamiento"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Modificar Tratamientos"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    />
```



```
app:layout_constraintVertical_bias="0.015" />

<Button
    android:id="@+id/recorrerH"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Recorrer horarios"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.436" />

<EditText
    android:id="@+id/NTratModn"
    android:layout_width="207dp"
    android:layout_height="56dp"
    android:ems="10"
    android:hint="Nombre Tratamiento"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.62" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad de Kotlin correspondiente

```
package com.example.apptt2

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.*
import androidx.appcompat.widget.SwitchCompat
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase

class ModificarTratamientoActive : AppCompatActivity() {
    //variables para enlazar xml con kotlin
    var bSalirModificarTratamiento:Button ?= null
    var botonGuardarDatosMod:Button ?=null
    var botonRecorrerHorarios:Button?=null
    //var PermanenteSwitchMod: SwitchCompat ?= null
    //var RadioGroupModTrata: RadioGroup =
    findViewById(R.id.RadioModTratam)
    private lateinit var databaseModT:DatabaseReference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_modificar_tratamiento_active)

        //Enlazando botones con kotlin
        bSalirModificarTratamiento=findViewById(R.id.BotonSalirModTratamientos
    )
```



```
botonGuardarDatosMod=findViewById(R.id.GuardarModificacionesTratam)
    botonRecorrerHorarios=findViewById(R.id.recorrerH)
    val
    PermanenteSwitchMod:SwitchCompat=findViewById(R.id.ModPermanenteSw)
    val RadioGroupModTrata: RadioGroup =
    findViewById(R.id.RadioModTratam)
    val nombreTratModificar:EditText=findViewById(R.id.NTratModn)
    val duracionTrMod:EditText=findViewById(R.id.ModDiasTr)
    val periodoTrMod:EditText=findViewById(R.id.ModIngP)
    var modTrID=" "
    var modAlm=0
    //Evento de RadioButton Group
    RadioGroupModTrata.setOnCheckedChangeListener { group,
    checkedId1 ->
        when(checkedId1) {
            //Casos
            R.id.RadioModTrata1 -> {
                modTrID="Tratamiento1"
                modAlm=1
                Toast.makeText(this,"Modificar Tratamiento 1",
                Toast.LENGTH_SHORT).show() }
            R.id.RadioModTrata2 -> {
                modTrID="Tratamiento2"
                modAlm=2
                Toast.makeText(this,"Modificar Tratamiento 2",
                Toast.LENGTH_SHORT).show() }
            R.id.RadioModTrata3 -> {
                modTrID="Tratamiento3"
                modAlm=3
                Toast.makeText(this,"Modificar Tratamiento 3",
                Toast.LENGTH_SHORT).show() }
            R.id.RadioModTrata4 -> {
                modTrID="Tratamiento4"
                modAlm=4
                Toast.makeText(this,"Modificar Tratamiento 4",
                Toast.LENGTH_SHORT).show() }
        } //finaliza when
    } //finaliza evento de RadioGroup

    //Eventos OnCLik de botones
    botonGuardarDatosMod!!.setOnClickListener {
        //recuperando información ingresada porel usuario
        val nombreTratamientoM=nombreTratModificar.text.toString()
        val
        duracionTratamientoM=duracionTrMod.text.toString().toInt()
        val
        periodoTratamientoM=periodoTrMod.text.toString().toInt()
        val permTratamientoM=PermanenteSwitchMod.isChecked
        //Actualizando datos en firebase

        databaseModT=FirebaseDatabase.getInstance().getReference("Tratamientos
        ")

        databaseModT.child(modTrID).child("DuracionD").setValue(duracionTratam
        ientoM).addOnFailureListener {
            Toast.makeText(this, "fallo de conexión",
```




```
Toast.LENGTH_SHORT).show()
    }

databaseModT.child(modTrID).child("PeriodoIng").setValue(periodoTratamientoM).addOnFailureListener {
    Toast.makeText(this, "fallo de conexión",
Toast.LENGTH_SHORT).show()
    }

databaseModT.child(modTrID).child("nombreTr").setValue(nombreTratamientoM).addOnFailureListener {
    Toast.makeText(this, "fallo de conexión",
Toast.LENGTH_SHORT).show()
    }

databaseModT.child(modTrID).child("permanente").setValue(permTratamientoM).addOnSuccessListener {
    Toast.makeText(this, "datos actualizados",
Toast.LENGTH_SHORT).show()
    }.addOnFailureListener {
    Toast.makeText(this, "fallo de conexión",
Toast.LENGTH_SHORT).show()
    }
    val intentRecAlarm=Intent(this,AlarmasSCTr::class.java)
    intentRecAlarm.putExtra("modHrs",modAlm)
    startActivity(intentRecAlarm)
}

botonRecorrerHorarios!!.setOnClickListener {
    val intentRecAlarm=Intent(this,AlarmasSCTr::class.java)
    intentRecAlarm.putExtra("modHrs",modAlm)
    startActivity(intentRecAlarm)
}

bSalirModificarTratamiento!!.setOnClickListener {
    //Regresar a manú principal
    val ModtToMainIntent: Intent
=Intent(this,MainActivity::class.java)
    startActivity(ModtToMainIntent)
}
//fin eventos onclick
}
}
```

Ventana de modo viaje

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ModoViaje">

    <TextView
        android:id="@+id/TituloModoViaje"
```



```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Modo Viaje"  
android:textSize="20sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.107" />
```

<EditText

```
android:id="@+id/ETDiasFuera"  
android:layout_width="67dp"  
android:layout_height="54dp"  
android:ems="10"  
android:hint="3"  
android:inputType="number"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.822"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.186" />
```

<TextView

```
android:id="@+id/DiasFuera"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Cuantos días te vas?"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.134"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.199" />
```

<TextView

```
android:id="@+id/Tratamiento4Viaje"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Tratamiento 4:"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.212"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.659" />
```

<TextView

```
android:id="@+id/Tratamiento2Viaje"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Tratamiento 2:"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"
```



```
app:layout_constraintHorizontal_bias="0.212"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.455" />

<TextView
    android:id="@+id/Tratamiento1Viaje"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tratamiento 1:"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.212"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.349" />

<TextView
    android:id="@+id/Tratamiento3Viaje"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tratamiento 3:"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.212"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.564" />

<TextView
    android:id="@+id/Tratamiento4CantidadViaje"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="5"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.705"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.659" />

<TextView
    android:id="@+id/Tratamiento1CantidadViaje"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="5"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.705"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.35" />

<TextView
```



```
android:id="@+id/Tratamiento3CantidadViaje"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:hint="5"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.705"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.565" />
```

<TextView

```
android:id="@+id/Tratamiento2CantidadViaje"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:hint="5"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.705"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.456" />
```

<Button

```
android:id="@+id/ExtraerPastillasViaje"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Recibir los Medicamentos"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.496"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.856" />
```

<TextView

```
android:id="@+id/TotalPastillasViaje"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Pastillas a recibir:"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.8"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.294" />
```

<Button

```
android:id="@+id/BotonSalirMTravel"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Salir"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.498"
```



```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.95" />  
  
<Button  
    android:id="@+id/CalcularPastillasViaje"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Calcular Cantidad de Pastillas"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.488"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.764" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad de Kotlin correspondiente

```
package com.example.apptt2  
  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import android.widget.TextView  
import android.widget.Toast  
import androidx.core.widget.doAfterTextChanged  
import androidx.core.widget.doOnTextChanged  
import com.google.firebase.database.DatabaseReference  
import com.google.firebase.database.FirebaseDatabase  
  
class ModoViaje : AppCompatActivity() {  
    //Variables para enlazar XML con Kotlin  
    var BotonRecibirMedicamentoViaje:Button ?= null  
    var bSalirModViaje: Button ?= null  
    var bCalcPasV : Button ?= null  
    private lateinit var databasem:DatabaseReference //referencia para  
base de datos firebase  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_modos_viaje)  
  
        //Enlazando boton con Kotlin  
  
        BotonRecibirMedicamentoViaje=findViewById(R.id.ExtraerPastillasViaje)  
        bSalirModViaje=findViewById(R.id.BotonSalirMTravel)  
        bCalcPasV=findViewById(R.id.CalcularPastillasViaje)  
        val diasFuera:EditText=findViewById(R.id.ETDiasFuera)  
        val  
t1Cviaje:TextView=findViewById(R.id.Tratamiento1CantidadViaje)  
        val  
t2Cviaje:TextView=findViewById(R.id.Tratamiento2CantidadViaje)  
        val  
t3Cviaje:TextView=findViewById(R.id.Tratamiento3CantidadViaje)  
        val  
t4Cviaje:TextView=findViewById(R.id.Tratamiento4CantidadViaje)  
        var per1=2
```



```
var per2=2
var per3=1
var per4=1
var cantPasTravel1=0
var cantPasTravel2=0
var cantPasTravel3=0
var cantPasTravel4=0
var cantPasDisp1=0
var cantPasDisp2=0
var cantPasDisp3=0
var cantPasDisp4=0
//Evento para mostrar cantidades de pastillas necesarias para
viaje
bCalcPasV!!.setOnClickListener {
    val prueban=diasFuera.text.toString()
    if((prueban.toInt()<=3)and(prueban.toInt()>0)){
        val hrsTot=diasFuera.text.toString().toInt() *24

databasem=FirebaseDatabase.getInstance().getReference("Tratamientos")
databasem.child("Tratamiento1").get().addOnSuccessListener {
per1=(it.child("PeriodoIng").value).toString().toInt()

cantPasDisp1=(it.child("numPast").value).toString().toInt()
    }.addOnFailureListener{
        Toast.makeText(this,"Tratamiento 1 no
disponible",Toast.LENGTH_SHORT).show()
    }

databasem.child("Tratamiento2").get().addOnSuccessListener {
per2=(it.child("PeriodoIng").value).toString().toInt()

cantPasDisp2=(it.child("numPast").value).toString().toInt()
    }.addOnFailureListener{
        Toast.makeText(this,"Tratamiento 2 no
disponible",Toast.LENGTH_SHORT).show()
    }

databasem.child("Tratamiento3").get().addOnSuccessListener {
per3=(it.child("PeriodoIng").value).toString().toInt()

cantPasDisp3=(it.child("numPast").value).toString().toInt()
    }.addOnFailureListener{
        Toast.makeText(this,"Tratamiento 3 no
disponible",Toast.LENGTH_SHORT).show()
    }

databasem.child("Tratamiento4").get().addOnSuccessListener {
per4=(it.child("PeriodoIng").value).toString().toInt()

cantPasDisp4=(it.child("numPast").value).toString().toInt()
    }.addOnFailureListener{
        Toast.makeText(this,"Tratamiento 4 no
```



```
disponible", Toast.LENGTH_SHORT).show()
    }
    cantPasTravel1=hrsTot/(per1)
    cantPasTravel2=hrsTot/(per2)
    cantPasTravel3=hrsTot/(per3)
    cantPasTravel4=hrsTot/(per4)

    t1Cviaje.text = cantPasTravel1.toString()
    t2Cviaje.text = cantPasTravel2.toString()
    t3Cviaje.text = cantPasTravel3.toString()
    t4Cviaje.text = cantPasTravel4.toString()

    }else{
        Toast.makeText(this, "Número de dias debe ser >0 ó
<3", Toast.LENGTH_SHORT).show()
    }
}
//Evento Onclick para activar bandera de extracción de
Medicamentos
BotonRecibirMedicamentoViaje!!.setOnClickListener {
    // lógica para advertencia de pastillas insuficientes
    if (cantPasDisp1-cantPasTravel1<0) {
        Toast.makeText(this, "Tratamiento 1 entregaré:
"+cantPasDisp1, Toast.LENGTH_SHORT).show()
        cantPasTravel1=cantPasDisp1
        t1Cviaje.text = cantPasTravel1.toString()
    }
    if (cantPasDisp2-cantPasTravel2<0) {
        Toast.makeText(this, "Tratamiento 2 entregare:
"+cantPasDisp2, Toast.LENGTH_SHORT).show()
        cantPasTravel2=cantPasDisp2
        t2Cviaje.text = cantPasTravel2.toString()
    }
    if (cantPasDisp3-cantPasTravel3<0) {
        Toast.makeText(this, "Tratamiento 3 entregare:
"+cantPasDisp3, Toast.LENGTH_SHORT).show()
        cantPasTravel3=cantPasDisp3
        t3Cviaje.text = cantPasTravel3.toString()
    }
    if (cantPasDisp4-cantPasTravel4<0) {
        Toast.makeText(this, "Tratamiento 4 entregare:
"+cantPasDisp4, Toast.LENGTH_SHORT).show()
        cantPasTravel4=cantPasDisp4
        t4Cviaje.text = cantPasTravel4.toString()
    }
    // activando bandera de firebase para modo viaje
    databasem=
    FirebaseDatabase.getInstance().getReference("BanderasControl")
    databasem.child("ModoViaje").setValue(1).addOnSuccessListener {
        Toast.makeText(this, "iniciando modo viaje",
    Toast.LENGTH_SHORT).show()
    }.addOnFailureListener {
        Toast.makeText(this, "modo viaje fail",
    Toast.LENGTH_SHORT).show()
    }
    //Actualización de datos en firebase
    databasem=
```




```
FirebaseDatabase.getInstance().getReference("Tratamientos")

databasem.child("Tratamiento1").child("numPast").setValue(cantPasDisp1-
cantPasTravel1)

databasem.child("Tratamiento2").child("numPast").setValue(cantPasDisp2-
cantPasTravel2)

databasem.child("Tratamiento3").child("numPast").setValue(cantPasDisp3-
cantPasTravel3)

databasem.child("Tratamiento4").child("numPast").setValue(cantPasDisp4-
cantPasTravel4)
        Toast.makeText(this, "Volviendo al menú",
Toast.LENGTH_SHORT).show()
        val modViajeIntent=Intent(this,MainActivity::class.java)
        //startActivity(modViajeIntent)
    }
    bSalirModViaje!!.setOnClickListener {
        //Boton para regresar a menú principal
        val modViajeIntent=Intent(this,MainActivity::class.java)
        startActivity(modViajeIntent)
    }
}
}
```

Actividad de monitoreo del dispositivo

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MonitorearEliteCare">

    <TextView
        android:id="@+id/TituloMonitorear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Observar funcionamiento del Dispositivo"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.494"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.041" />

    <TextView
        android:id="@+id/PRestantesT2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="2"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
```



```
app:layout_constraintHorizontal_bias="0.635"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.297" />

<TextView
    android:id="@+id/RestantesGral"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pastillas Restantes"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.702"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.101" />

<Button
    android:id="@+id/ActualizarDatos"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Actualizar Información"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.843" />

<TextView
    android:id="@+id/PastillasRestantesT1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tratamiento 1:"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.226"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.185" />

<TextView
    android:id="@+id/PastillasRestantesT3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tratamiento 3:"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.226"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.417" />

<TextView
    android:id="@+id/ValorHumedad"
```



```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="25"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.79"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.746" />
```

<TextView

```
android:id="@+id/PastillasRestantesT2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Tratamiento 2:"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.226"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.297" />
```

<TextView

```
android:id="@+id/HumedadInterna"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Humedad Relativa Interna (%):"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.164"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.746" />
```

<TextView

```
android:id="@+id/PRestantesT1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="5"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.635"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.183" />
```

<TextView

```
android:id="@+id/Tempreatura"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Temperatura Interna (°C):"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"
```



```
app:layout_constraintHorizontal_bias="0.296"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.649" />  
  
<TextView  
    android:id="@+id/PRestantesT3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="8"  
    android:textSize="18sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.635"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.417" />  
  
<TextView  
    android:id="@+id/PRestantesT4"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="0"  
    android:textSize="18sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.635"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.538" />  
  
<TextView  
    android:id="@+id/ValorTemperatura"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="22"  
    android:textSize="18sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.79"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.649" />  
  
<TextView  
    android:id="@+id/PastillasRestantesT4"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Tratamiento 4"  
    android:textSize="18sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.223"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.538" />  
  
<Button
```



```
android:id="@+id/BotonSalirMonitoreoEC"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Salir"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.498"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.942" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad correspondiente de Kotlin

```
package com.example.apptt2  
  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.TextView  
import android.widget.Toast  
import com.google.firebase.database.DatabaseReference  
import com.google.firebase.database.FirebaseDatabase  
  
class MonitorearEliteCare : AppCompatActivity() {  
    //Variables para enlazar xml con kotlin  
    var botonActualizarDatosMonitoreo:Button ?= null  
    var bSalirMonitorELiteCare:Button ?= null  
    private lateinit var databasemec:DatabaseReference//referencia  
para base de datos firebase  
    private lateinit var databasemec2:DatabaseReference  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_monitorear_elite_care)  
  
        //Enlazando boton con kotlin  
  
        botonActualizarDatosMonitoreo=findViewById(R.id.ActualizarDatos)  
  
        bSalirMonitorELiteCare=findViewById(R.id.BotonSalirMonitoreoEC)  
        val pasRes1Tv:TextView=findViewById(R.id.PRestantesT1)  
        val pasRes2Tv:TextView=findViewById(R.id.PRestantesT2)  
        val pasRes3Tv:TextView=findViewById(R.id.PRestantesT3)  
        val pasRes4Tv:TextView=findViewById(R.id.PRestantesT4)  
        val tratname1:TextView=findViewById(R.id.PastillasRestantesT1)  
        val tratname2:TextView=findViewById(R.id.PastillasRestantesT2)  
        val tratname3:TextView=findViewById(R.id.PastillasRestantesT3)  
        val tratname4:TextView=findViewById(R.id.PastillasRestantesT4)  
        val  
        temperaturaInterna:TextView=findViewById(R.id.ValorTemperatura)  
        val humedadInterna:TextView=findViewById(R.id.ValorHumedad)  
  
        databasemec=FirebaseDatabase.getInstance().getReference("Tratamientos"  
)  
  
        databasemec2=FirebaseDatabase.getInstance().getReference("BanderasCont
```



```
rol")

        //Evento onclick de actualización de datos consultando desde
firebase
        botonActualizarDatosMonitoreo!!.setOnClickListener {

databasemec.child("Tratamiento1").get().addOnSuccessListener {
        pasRes1Tv.text=it.child("numPast").value.toString()
        tratname1.text=it.child("nombreTr").value.toString()
        }.addOnFailureListener {
            Toast.makeText(this, "mala conexión",
Toast.LENGTH_SHORT).show()
        }

databasemec.child("Tratamiento2").get().addOnSuccessListener {
        pasRes2Tv.text=it.child("numPast").value.toString()
        tratname2.text=it.child("nombreTr").value.toString()
        }.addOnFailureListener {
            Toast.makeText(this, "mala conexión",
Toast.LENGTH_SHORT).show()
        }

databasemec.child("Tratamiento3").get().addOnSuccessListener {
        pasRes3Tv.text=it.child("numPast").value.toString()
        tratname3.text=it.child("nombreTr").value.toString()
        }.addOnFailureListener {
            Toast.makeText(this, "mala conexión",
Toast.LENGTH_SHORT).show()
        }

databasemec.child("Tratamiento4").get().addOnSuccessListener {
        pasRes4Tv.text=it.child("numPast").value.toString()
        tratname4.text=it.child("nombreTr").value.toString()
        }.addOnFailureListener {
            Toast.makeText(this, "mala conexión",
Toast.LENGTH_SHORT).show()
        }
        //Datos de Humedad y Temperatura
databasemec2.get().addOnSuccessListener {

humedadInterna.text=it.child("HumedadIn").value.toString()

temperaturaInterna.text=it.child("TemperaturaIn").value.toString()
        }.addOnFailureListener {
            Toast.makeText(this, "mala conexión",
Toast.LENGTH_SHORT).show()
        }
    }
    //Evento onclick para volver a menú principal
    bSalirMonitorELiteCare!!.setOnClickListener{
        val
MonitorToMainIntent:Intent=Intent(this,MainActivity::class.java)
        startActivity(MonitorToMainIntent)
    }
}
}
```



Ventana de notificación de ingesta de medicamento

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".OmissionNoti">

    <TextView
        android:id="@+id/NotiTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Elite Care"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.067" />

    <TextView
        android:id="@+id/NotiAlerta"
        android:layout_width="351dp"
        android:layout_height="205dp"
        android:text="Estimado usuario, por favor, ingiere tu
pastilla. Tienes 1h para ingerirla, en caso de omitirla se recomienda
consultar al médico si debe esperar hasta la proxima dosis o si
ingerirla inmediatamente y recorrer sus horarios de ingesta"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.241" />

    <Button
        android:id="@+id/bNotif"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Listo!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.808" />

    <TextView
        android:id="@+id/NotiRem"
        android:layout_width="351dp"
        android:layout_height="80dp"
        android:text="Te recordamos que puedes recorrer tus horarios
de ingesta en la opcion de modificar tratamientos"
        android:textSize="20sp"

```




```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.533"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.937" />

<TextView
    android:id="@+id/bodNoti"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Debes ingerir:"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.274"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.518" />

<TextView
    android:id="@+id/nomTrNoti"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="ej Ibuprofeno"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.74"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.518" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad correspondiente de Kotlin

```
package com.example.apptt2

import android.app.AlarmManager
import android.app.PendingIntent
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.SystemClock
import android.widget.Button
import android.widget.TextView

class OmissionNoti : AppCompatActivity() {
    var btnNoti:Button?=null
    private lateinit var alarmmanager5:AlarmManager
    private lateinit var pendingIntentOm:PendingIntent
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_omision_noti)
        //vinculando boton
        btnNoti=findViewById(R.id.bNotif)
        //Vinculando TextView para nombre de tratamiento
        val nameTrat:TextView=findViewById(R.id.nomTrNoti)
        //creando intents para recuperar información
```



```
val intNoti=intent
var nombreInt=intNoti.getStringExtra("nombreTr")
//Creando intents para alarma en caso de omisión
val intent5=Intent(this,AlarmReciver5::class.java)
intent5.putExtra("nomNotif",nombreInt)
pendingIntentOm=PendingIntent.getBroadcast(this,0,intent5,0)
//Indicando que tratameinto se debe ingerir
nameTrat.text=nombreInt
//Obteniendo servicios de alarma de sistema
alarmmanager5=getSystemService(ALARM_SERVICE) as AlarmManager
//Estableciendo alarma para sonar dentro de una hora
alarmmanager5.set(
    AlarmManager.ELAPSED_REALTIME,
    SystemClock.elapsedRealtime()+AlarmManager.INTERVAL_HOUR,
    pendingIntentOm
)

btnNoti!!.setOnClickListener {
    //Canelando la alarma de omisión
    alarmmanager5.cancel(pendingIntentOm)
    //Volviendo a menú principal
    val notiToMain=Intent(this,MainActivity::class.java)
    startActivity(notiToMain)
}

}
```

ventana de reabastecer tratamientos

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/NumeroPastillasReabastecer"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".ReabastecerTratamientoActive">

    <TextView
        android:id="@+id/IndicacionesReabastecer"
        android:layout_width="333dp"
        android:layout_height="56dp"
        android:text="Por favor selecciona el tratamiento que deseas
reabastecer:"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.14" />

    <Button
        android:id="@+id/ModTratamiento3"
        android:layout_width="wrap_content"
```



```
android:layout_height="wrap_content"  
android:text="Tratamiento 3"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.433" />
```

<Button

```
android:id="@+id/ModTratamiento2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Tratamiento 2"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.351" />
```

<TextView

```
android:id="@+id/TituloReabastecerTratamiento"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Reabastecer Tratamiento"  
android:textSize="20sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.497"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.063" />
```

<Button

```
android:id="@+id/ModTratamiento1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Tratamiento1"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.49"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.262" />
```

<TextView

```
android:id="@+id/TextPastillasNecesarias"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Pastillas necesarias para reabastecer:"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.63" />
```

<TextView

```
android:id="@+id/ValorPastillasNecesariasTxt"
```



```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="5"  
android:textSize="18sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.498"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.691" />
```

<Button

```
android:id="@+id/BotonIngresarMedicamento"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Ingresar Medicamentos"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.8" />
```

<Button

```
android:id="@+id/BotonSalirReabTratamientos"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Salir"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.498"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.915" />
```

<Button

```
android:id="@+id/ModTratamiento4"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="TRATAMIENTO 4"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.498"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.522" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

Actividad correspondiente de Kotlin

```
private lateinit var databaseReab2: DatabaseReference  
  
//var  
PastillasCantidad: TextView = findViewById(R.id.TextPastillasNecesarias)  
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
  
    setContentView(R.layout.activity_reabastecer_tratamiento_active)
```



```
//Vinculando xml con kotlin

botonReabastecerTratamiento1=findViewById(R.id.ModTratamiento1)
botonReabastecerTratamiento2=findViewById(R.id.ModTratamiento2)
botonReabastecerTratamiento3=findViewById(R.id.ModTratamiento3)
botonReabastecerTratamiento4=findViewById(R.id.ModTratamiento4)
botonIngresarMedicamento=findViewById(R.id.BotonIngresarMedicamento)
bSalirReabTratamientos=findViewById(R.id.BotonSalirReabTratamientos)
    val
PastillasCantidad:TextView=findViewById(R.id.ValorPastillasNecesariasT
xt)
    var numTrataReab=0

databaseReab=FirebaseDatabase.getInstance().getReference("Tratamientos
")

    //eventos onclick de los botones de la activity
    botonReabastecerTratamiento1!!.setOnClickListener {
        //Obteniendo pastillas actuales de firebase y calculando
pastillas

databaseReab.child("Tratamiento1").get().addOnSuccessListener {
    val x1 =(it.child("numPast").value)
    NumPastillasreab= 15 - (x1.toString().toInt())
    //Desplegando pastillas a ingresar
    PastillasCantidad.text=NumPastillasreab.toString()
    numTrataReab=1
}.addOnFailureListener {
    Toast.makeText(this, "fallo conexión 1",
Toast.LENGTH_SHORT).show()
}

}
    botonReabastecerTratamiento2!!.setOnClickListener {
        //Obteniendo pastillas actuales de firebase y calculando
pastillas

databaseReab.child("Tratamiento2").get().addOnSuccessListener {
    val x2 =(it.child("numPast").value)
    NumPastillasreab= 15 - (x2.toString().toInt())
    //Desplegando pastillas a ingresar
    PastillasCantidad.text=NumPastillasreab.toString()
    numTrataReab=2
}.addOnFailureListener {
    Toast.makeText(this, "fallo conexión 2",
Toast.LENGTH_SHORT).show()
}

}
    botonReabastecerTratamiento3!!.setOnClickListener {
        //Obteniendo pastillas actuales de firebase y calculando
pastillas

databaseReab.child("Tratamiento3").get().addOnSuccessListener {
```



```
        val x3 =(it.child("numPast").value)
        NumPastillasreab= 15 - (x3.toString().toInt())
        //Desplegando pastillas a ingresar
        PastillasCantidad.text=NumPastillasreab.toString()
        numTrataReab=3
    }.addOnFailureListener {
        Toast.makeText(this, "fallo conexión 3",
        Toast.LENGTH_SHORT).show()
    }
}
botonReabastecerTratamiento4!!.setOnClickListener {
    //Obteniendo pastillas actuales de firebase y calculando
    pastillas
    databaseReab.child("Tratamiento4").get().addOnSuccessListener {
        val x4 =(it.child("numPast").value)
        NumPastillasreab= 15 - (x4.toString().toInt())
        //Desplegando pastillas a ingresar
        PastillasCantidad.text=NumPastillasreab.toString()
        numTrataReab=4
    }.addOnFailureListener {
        Toast.makeText(this, "fallo conexión 4",
        Toast.LENGTH_SHORT).show()
    }
}
botonIngresarMedicamento!!.setOnClickListener {
    databaseReab2=FirebaseDatabase.getInstance().getReference("BanderasCon
    trol")
    databaseReab2.child("IngreNTrat").setValue(numTrataReab).addOnSuccessL
    istener {
        Toast.makeText(this, "ingrese pastillas",
        Toast.LENGTH_SHORT).show()
        val intentToAlarmas: Intent = Intent(this,
        AlarmasSCTr::class.java)
        intentToAlarmas.putExtra("revolverPos", numTrataReab)
        //startActivity(intentToAlarmas)
    }.addOnFailureListener {
        Toast.makeText(this, "Fallo Control",
        Toast.LENGTH_SHORT).show() }
    }
    bSalirReabTratamientos!!.setOnClickListener {
        val ReabToMainIntent: Intent =Intent(this,
        MainActivity::class.java)
        PastillasCantidad.text = null
        startActivity(ReabToMainIntent)
    }
}
}
```

Ventana de seleccion de tipo de pastilla

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```



```
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".SeleccionarTipoDePastilla">

<ImageButton
    android:id="@+id/ImagenPastilla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:contentDescription="PastillaRedonda"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.164"
    app:srcCompat="@mipmap/ic_launcher" />

<TextView
    android:id="@+id/Tableta"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tableta"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.507" />

<ImageButton
    android:id="@+id/ImagenTableta"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:contentDescription="Tableta"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.41"
    app:srcCompat="@mipmap/ic_launcher" />

<TextView
    android:id="@+id/Pastilla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pastilla Redonda"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.293" />

<ImageButton
    android:id="@+id/ImagenCapsula"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```




```
android:contentDescription="Capsula"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.65"
app:srcCompat="@mipmap/ic_launcher" />

<TextView
    android:id="@+id/Capsula"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Capsula"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.727" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Actividad correspondiente de kotlin

```
package com.example.apptt2

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.ImageButton
import android.widget.Toast
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase

class SeleccionarTipoDePastilla : AppCompatActivity() {
    var botonPastilla : ImageButton ?= null
    var botonTableta : ImageButton ?= null
    var botonCapsula : ImageButton ?= null
    private lateinit var databasesp:DatabaseReference//referencia para
base de datos firebase

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_seleccionar_tipo_de_pastilla)

        //Vinculando Botones
        botonPastilla= findViewById(R.id.ImagenPastilla)
        botonTableta= findViewById(R.id.ImagenTableta)
        botonCapsula= findViewById(R.id.ImagenCapsula)
        //creando variables para verificar disponibilidad
        var tp1Counter=0
        var tp2Counter=0
        var tp3Counter=0
        //variable intent para cambio de actividad
        val inTratIntent: Intent =
Intent(this, IngresarNuevoTratamientoActive::class.java)
```



```
//consultando disponibilidad de contenedores en firebase
databasesp=FirebaseDatabase.getInstance().getReference("BanderasControl")

databasesp.child("DispContenedores").get().addOnSuccessListener {
    tp1Counter=(it.child("c1Disp").value).toString().toInt()
    tp2Counter=(it.child("c2Disp").value).toString().toInt()
    tp3Counter=(it.child("c3Disp").value).toString().toInt()
}.addOnFailureListener {
    Toast.makeText(this, "mala conexión, intente mas tarde",
Toast.LENGTH_SHORT).show()
}

//Eventos Onclick para cambiar a activiti de configuración y
activar banderas
botonPastilla!!.setOnClickListener {
    //Enviando bandera a base de datos firecloud
    tp1Counter+=1
    if(tp1Counter>1){
        Toast.makeText(this, "contenedor ocupado, lo siento",
Toast.LENGTH_SHORT).show()
    }else{

databasesp.child("DispContenedores").child("c1Disp").setValue(1).addOn
SuccessListener {
        inTratIntent.putExtra("revolverPos",1)
        //iniciando la actividad correspondiente
        startActivity(inTratIntent)}.addOnFailureListener
{
        Toast.makeText(this, "intente mastarde",
Toast.LENGTH_SHORT).show()
    }
}

botonTableta!!.setOnClickListener {
    //Enviando bandera a base de datos firecloud
    tp2Counter+=1
    if(tp2Counter>1){
        Toast.makeText(this, "contenedor ocupado, lo siento",
Toast.LENGTH_SHORT).show()
    }else{

databasesp.child("DispContenedores").child("c2Disp").setValue(1).addOn
SuccessListener {
        inTratIntent.putExtra("revolverPos",2)
        //iniciando la actividad correspondiente
        startActivity(inTratIntent)}.addOnFailureListener
{
        Toast.makeText(this, "intente mas tarde",
Toast.LENGTH_SHORT).show()
    }
}

botonCapsula!!.setOnClickListener {
```



```
//Enviando bandera a base de datos firecloud
tp3Counter+=1
if(tp2Counter>2){
    Toast.makeText(this, "contenedores ocupados, lo
siento", Toast.LENGTH_SHORT).show()
}else{

databasesp.child("DispContenedores").child("c3Disp").setValue(tp3Count
er).addOnSuccessListener {
    inTratIntent.putExtra("revolverPos",tp3Counter+2)
    //iniciando la actividad correspondiente
    startActivity(inTratIntent)}.addOnFailureListener
{
    Toast.makeText(this, "intente mas tarde",
Toast.LENGTH_SHORT).show()
    }
}
}
}
```

Actividades de Kotlin adicionales

```
package com.example.apptt2

import android.app.AlarmManager
import android.app.PendingIntent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.content.Intent
import android.os.SystemClock
import android.widget.Toast
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase

class AlarmasSCTr:AppCompatActivity() {
    private lateinit var databaseAlramas:DatabaseReference//Referencia
a base de datos
    private lateinit var
databaseAlramasB:DatabaseReference//Referencia a base de datos
    private lateinit var alarmmanager:AlarmManager//Servicio de
alarmas de andrioid
    //Variables de broadcast de las alarmas
    private lateinit var pendingIntentA1: PendingIntent
    private lateinit var pendingIntentA2: PendingIntent
    private lateinit var pendingIntentA3: PendingIntent
    private lateinit var pendingIntentA4: PendingIntent

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val stAlarma=intent
        val setearAlarmas=stAlarma.getIntExtra("revolverPos",0)
        val rechoras=intent
        val recHrs=rechoras.getIntExtra("modHrs",0)
        var tiempoRepetir=8
        val intent1=Intent(this,AlarmReciver::class.java)
        val intent2=Intent(this,AlarmReciver2::class.java)
```



```
val intent3=Intent(this,AlarmReciver3::class.java)
val intent4=Intent(this,AlarmReciver4::class.java)
var borradorAlarma=0
//intents para actividades y ejecucion de alarmas
val intentToMainA=Intent(this,FinIngresoPastillas::class.java)
val
intentAlmToBorrar=Intent(this,BorrarTratamientoActive::class.java)
val
intentAlmToNTr=Intent(this,IngresarNuevoTratamientoActive::class.java)
val
intentAlmToMod=Intent(this,ModificarTratamientoActive::class.java)
pendingIntentA1= PendingIntent.getBroadcast(this,0,intent1,0)
pendingIntentA2= PendingIntent.getBroadcast(this,0,intent2,0)
pendingIntentA3= PendingIntent.getBroadcast(this,0,intent3,0)
pendingIntentA4= PendingIntent.getBroadcast(this,0,intent4,0)
//Conectando a base de datos

databaseAlramas=FirebaseDatabase.getInstance().getReference("Tratamien
tos")

databaseAlramasB=FirebaseDatabase.getInstance().getReference("Banderas
Control")
//Obteniendo servicios de alarmas de sistema
alarmmanager=getSystemService(ALARM_SERVICE) as AlarmManager

when(setearAlarmas){
    1->{

databaseAlramas.child("Tratamiento1").get().addOnSuccessListener {
    tiempoRepetir=it.child("PeriodoIng").value as Int

    alarmmanager.setRepeating(
        AlarmManager.ELAPSED_REALTIME_WAKEUP,
        SystemClock.elapsedRealtime()+60000,
        (tiempoRepetir*60*60*100).toLong(),
        pendingIntentA1
    )
    startActivity(intentToMainA)
}.addOnFailureListener {
    Toast.makeText(applicationContext, "fallo
alarma1", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToNTr)
}
}
2->{

databaseAlramas.child("Tratamiento2").get().addOnSuccessListener {
    tiempoRepetir=it.child("PeriodoIng").value as Int

    alarmmanager.setRepeating(
        AlarmManager.ELAPSED_REALTIME_WAKEUP,
        SystemClock.elapsedRealtime()+120000,
        (tiempoRepetir*60*60*100).toLong(),
        pendingIntentA2
    )
    startActivity(intentToMainA)
}.addOnFailureListener {
    Toast.makeText(applicationContext, "fallo
```



```
alarma2", Toast.LENGTH_SHORT).show()
        startActivity(intentAlmToNTr)
    }
}
3->{

databaseAlramas.child("Tratamiento3").get().addOnSuccessListener {
    tiempoRepetir=it.child("PeriodoIng").value as Int

    alarmmanager.setRepeating(
        AlarmManager.ELAPSED_REALTIME_WAKEUP,
        SystemClock.elapsedRealtime()+180000,
        (tiempoRepetir*60*60*100).toLong(),
        pendingIntentA3
    )
    startActivity(intentToMainA)
}.addOnFailureListener {
    Toast.makeText(applicationContext, "fallo
alarma3", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToNTr)
}
}
4->{

databaseAlramas.child("Tratamiento4").get().addOnSuccessListener {
    tiempoRepetir=it.child("PeriodoIng").value as Int

    alarmmanager.setRepeating(
        AlarmManager.ELAPSED_REALTIME_WAKEUP,
        SystemClock.elapsedRealtime()+240000,
        (tiempoRepetir*60*60*100).toLong(),
        pendingIntentA4
    )
    startActivity(intentToMainA)
}.addOnFailureListener {
    Toast.makeText(applicationContext, "fallo
alarma4", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToNTr)
}
}
else->{
    // Toast.makeText(applicationContext, "no alarma",
Toast.LENGTH_SHORT).show()
    startActivity(intentToMainA)
}
}

databaseAlramasB.get().addOnSuccessListener {
    borradorAlarma=it.child("Borrar").value as Int
    when (borradorAlarma) {
        1->{
            //pendingIntentA1=
PendingIntent.getBroadcast(this,0,intent1,0)
            alarmmanager.cancel(pendingIntentA1)
            Toast.makeText(applicationContext, "alarma1
borrada", Toast.LENGTH_SHORT).show()
            startActivity(intentAlmToBorrar)
        }
    }
}
```



```
2->{
    //pendingIntentA2=
PendingIntent.getBroadcast(this,0,intent2,0)
    alarmmanager.cancel(pendingIntentA2)
    Toast.makeText(applicationContext, "alarma2
borrada", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToBorrar)
}
3->{
    //pendingIntentA3=
PendingIntent.getBroadcast(this,0,intent3,0)
    alarmmanager.cancel(pendingIntentA3)
    Toast.makeText(applicationContext, "alarma3
borrada", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToBorrar)
}
4->{
    //pendingIntentA4=
PendingIntent.getBroadcast(this,0,intent4,0)
    alarmmanager.cancel(pendingIntentA4)
    Toast.makeText(applicationContext, "alarma4
borrada", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToBorrar)
}
else->{
    Toast.makeText(applicationContext, "alarmas
vigentes", Toast.LENGTH_SHORT).show()
}
}
}.addOnFailureListener {
    Toast.makeText(applicationContext, "falló conexión B",
Toast.LENGTH_SHORT).show()
}

//reagendando alarmas
when(recHrs){
1->{

databaseAlramas.child("Tratamiento1").get().addOnSuccessListener {
    tiempoRepetir=it.child("PeriodoIng").value as Int
    //pendingIntentA1=
PendingIntent.getBroadcast(this,0,intent1,0)
    alarmmanager.setRepeating(
        AlarmManager.ELAPSED_REALTIME_WAKEUP,
SystemClock.elapsedRealtime()+(tiempoRepetir*60*60*1000).toLong(),
        (tiempoRepetir*60*60*1000).toLong(),
        pendingIntentA1
    )
    startActivity(intentToMainA)
}.addOnFailureListener {
    Toast.makeText(applicationContext, "fallo
alarma1", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToMod)
}
}
}
2->{
```



```
databaseAlramas.child("Tratamiento2").get().addOnSuccessListener {
    tiempoRepetir=it.child("PeriodoIng").value as Int

    alarmmanager.setRepeating(
        AlarmManager.ELAPSED_REALTIME_WAKEUP,
        SystemClock.elapsedRealtime()+(tiempoRepetir*60*60*1000).toLong(),
        (tiempoRepetir*60*60*1000).toLong(),
        pendingIntentA2
    )
    startActivity(intentToMainA)
}.addOnFailureListener {
    Toast.makeText(applicationContext, "fallo
alarma22", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToMod)
}

}
3->{

databaseAlramas.child("Tratamiento3").get().addOnSuccessListener {
    tiempoRepetir=it.child("PeriodoIng").value as Int

    alarmmanager.setRepeating(
        AlarmManager.ELAPSED_REALTIME_WAKEUP,
        SystemClock.elapsedRealtime()+(tiempoRepetir*60*60*1000).toLong(),
        (tiempoRepetir*60*60*1000).toLong(),
        pendingIntentA3
    )
    startActivity(intentToMainA)
}.addOnFailureListener {
    Toast.makeText(applicationContext, "fallo
alarma33", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToMod)
}

}
4->{

databaseAlramas.child("Tratamiento4").get().addOnSuccessListener {
    tiempoRepetir=it.child("PeriodoIng").value as Int

    alarmmanager.setRepeating(
        AlarmManager.ELAPSED_REALTIME_WAKEUP,
        SystemClock.elapsedRealtime()+(tiempoRepetir*60*60*1000).toLong(),
        (tiempoRepetir*60*60*1000).toLong(),
        pendingIntentA4
    )
    startActivity(intentToMainA)
}.addOnFailureListener {
    Toast.makeText(applicationContext, "fallo
alarma44", Toast.LENGTH_SHORT).show()
    startActivity(intentAlmToMod)
}

}
}
```




```
}  
}
```

```
package com.example.apptt2  
  
import android.app.NotificationManager  
import android.app.PendingIntent  
import android.content.BroadcastReceiver  
import android.content.Context  
import android.content.Intent  
import android.widget.Toast  
import androidx.core.app.NotificationCompat  
import androidx.core.app.NotificationManagerCompat  
import com.google.firebase.database.DatabaseReference  
import com.google.firebase.database.FirebaseDatabase  
  
class AlarmReciver:BroadcastReceiver() {  
    private lateinit var databaseRecl:DatabaseReference  
    override fun onReceive(context: Context?, intent: Intent?) {  
        var nombreTR1 = "Tratameinto 1"  
        //var npast1=0  
        //Intent para actividad de notificacion  
        val  
reciverToNotiIntent=Intent (context,OmisionNoti::class.java)  
  
        intent!!.flags=Intent.FLAG_ACTIVITY_NEW_TASK or  
Intent.FLAG_ACTIVITY_CLEAR_TASK  
        val  
penIntent=PendingIntent.getActivity(context,0,reciverToNotiIntent,0)  
        //Cosultando nombre del tratamiento  
  
databaseRecl=FirebaseDatabase.getInstance().getReference("Tratamientos  
")  
        databaseRecl.child("Tratamiento1").get().addOnSuccessListener  
{  
            nombreTR1=(it.child("nombreTr").value).toString()  
            reciverToNotiIntent.putExtra("nombreTratam",nombreTR1)  
            //npast1=it.child("numPast").value as Int  
        }.addOnFailureListener {  
            Toast.makeText(context, "mala conexión alarmal",  
Toast.LENGTH_SHORT).show()  
        }  
        //Construyendo la notificación de tratamiento 1  
        val builder=NotificationCompat.Builder(context!,"ECareCh")  
            .setSmallIcon(R.drawable.ic_launcher_background)  
            .setContentTitle("Elite Care")  
            .setContentText("Ingiere tu pastilla del $nombreTR1")  
            .setAutoCancel(true)  
            .setDefaults(NotificationCompat.DEFAULT_ALL)  
            .setPriority(NotificationCompat.PRIORITY_HIGH)  
            .setContentIntent(penIntent)  
        val  
notificationManager=NotificationManagerCompat.from(context)  
        notificationManager.notify(123,builder.build())  
        //Actualizando número de pastillas preguntar a edgar si lo  
hará el.
```



```
//npast1--1  
  
//databaseRec1.child("Tratamiento1").child("numPast").setValue(npast1  
    }  
}
```

```
package com.example.apptt2  
  
data class Tratamientos(val DuracionD:Int?=null, val  
PeriodoIng:Int?=null, val nombreTr:String?=null, val  
numPast:Int?=null, val permanente:Boolean?=null) {  
}
```

```
import android.app.PendingIntent  
import android.content.BroadcastReceiver  
import android.content.Context  
import android.content.Intent  
import android.widget.Toast  
import androidx.core.app.NotificationCompat  
import androidx.core.app.NotificationManagerCompat  
import com.google.firebase.database.DatabaseReference  
import com.google.firebase.database.FirebaseDatabase  
  
class AlarmReciver2:BroadcastReceiver() {  
    private lateinit var databaseRec2:DatabaseReference  
    override fun onReceive(context: Context?, intent: Intent?) {  
        var nombreTR2="Tratamiento 2"  
        //var npast2=0  
        //Intent para actividad de notificacion  
        val  
reciverToNotiIntent2=Intent(context, OmissionNoti::class.java)  
        intent!!.flags=Intent.FLAG_ACTIVITY_NEW_TASK or  
Intent.FLAG_ACTIVITY_CLEAR_TASK  
        val penIntent2=  
PendingIntent.getActivity(context, 0, reciverToNotiIntent2, 0)  
        //Cosultando nombre del tratamiento  
        databaseRec2=  
FirebaseDatabase.getInstance().getReference("Tratamientos")  
        databaseRec2.child("Tratamiento2").get().addOnSuccessListener  
{  
            nombreTR2 =(it.child("nombreTr").value).toString()  
            //npast2=it.child("numPast").value as Int  
        }.addOnFailureListener {  
            Toast.makeText(context, "mala conexión alarma2",  
Toast.LENGTH_SHORT).show()  
        }  
        val builder= NotificationCompat.Builder(context!!, "ECareCh")  
            .setSmallIcon(R.drawable.ic_launcher_background)  
            .setContentTitle("Elite Care")  
            .setContentText("Ingerir pastilla de $nombreTR2")  
            .setAutoCancel(true)  
            .setDefaults(NotificationCompat.DEFAULT_ALL)
```



```
.setPriority(NotificationCompat.PRIORITY_HIGH)
.setContentView (penIntent2)

val notificationManager=
NotificationManagerCompat.from(context)
notificationManager.notify(126,builder.build())
//Actualizando número de pastillas preguntar a edgar si lo
hará el.
//npast2-=1

//databaseRec1.child("Tratamiento2").child("numPast").setValue(npast2)
}
}
```

```
package com.example.apptt2

import android.app.PendingIntent
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.widget.Toast
import androidx.core.app.NotificationCompat
import androidx.core.app.NotificationManagerCompat
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase

class AlarmReceiver3:BroadcastReceiver() {
    private lateinit var databaseRec3: DatabaseReference
    override fun onReceive(context: Context?, intent: Intent?) {
        var nombreTR3="Tratamiento 3"
        //var npast2=0
        //Intent para actividad de notificacion
        val
reciverToNotiIntent3=Intent(context,OmisionNoti::class.java)
        intent!!.flags=Intent.FLAG_ACTIVITY_NEW_TASK or
Intent.FLAG_ACTIVITY_CLEAR_TASK
        val penIntent3=
PendingIntent.getActivity(context,0,reciverToNotiIntent3,0)
        //Consultando nombre del tratamiento
        databaseRec3=
FirebaseDatabase.getInstance().getReference("Tratamientos")
        databaseRec3.child("Tratamiento3").get().addOnSuccessListener
{
            nombreTR3 =(it.child("nombreTr").value).toString()
            //npast3=it.child("numPast").value as Int
        }.addOnFailureListener {
            Toast.makeText(context, "mala conexión alarma3",
Toast.LENGTH_SHORT).show()
        }
        val builder= NotificationCompat.Builder(context!,"ECareCh")
        .setSmallIcon(R.drawable.ic_launcher_background)
        .setContentTitle("Elite Care")
        .setContentText("Ingerir pastilla de $nombreTR3")
        .setAutoCancel(true)
        .setDefaults(NotificationCompat.DEFAULT_ALL)
        .setPriority(NotificationCompat.PRIORITY_HIGH)
```



```
.setContentIntent (penIntent3)

    val notificationManager=
NotificationManagerCompat.from(context)
    notificationManager.notify(125,builder.build())
    //Actualizando número de pastillas preguntar a edgar si lo
    hará el.
    //npast3-=1

//databaseRec1.child("Tratamiento3").child("numPast").setValue(npast3)
    }
}
```

```
package com.example.apptt2

import android.app.PendingIntent
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.widget.Toast
import androidx.core.app.NotificationCompat
import androidx.core.app.NotificationManagerCompat
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase

class AlarmReciver4:BroadcastReceiver() {
    private lateinit var databaseRec4: DatabaseReference
    override fun onReceive(context: Context?, intent: Intent?) {
        var nombreTR4="Tratamiento 4"
        //var npast2=0
        //Intent para actividad de notificacion
        val
reciverToNotiIntent4=Intent(context,OmisionNoti::class.java)
        intent!!.flags=Intent.FLAG_ACTIVITY_NEW_TASK or
Intent.FLAG_ACTIVITY_CLEAR_TASK
        val penIntent4=
PendingIntent.getActivity(context,0,reciverToNotiIntent4,0)
        //Consultando nombre del tratamiento
        databaseRec4=
FirebaseDatabase.getInstance().getReference("Tratamientos")
        databaseRec4.child("Tratamiento3").get().addOnSuccessListener
{
            nombreTR4 =(it.child("nombreTr").value).toString()
            //npast3=it.child("numPast").value as Int
        }.addOnFailureListener {
            Toast.makeText(context, "mala conexión alarma3",
Toast.LENGTH_SHORT).show()
        }
        val builder= NotificationCompat.Builder(context!,"ECareCh")
            .setSmallIcon(R.drawable.ic_launcher_background)
            .setContentTitle("Elite Care")
            .setContentText("Ingerir pastilla de $nombreTR4")
            .setAutoCancel(true)
            .setDefaults(NotificationCompat.DEFAULT_ALL)
            .setPriority(NotificationCompat.PRIORITY_HIGH)
            .setContentIntent (penIntent4)
```



```
        val notificationManager=  
NotificationManagerCompat.from(context)  
        notificationManager.notify(124,builder.build())  
        //Actualizando número de pastillas preguntar a edgar si lo  
hará el.  
        //npast4-=1  
  
//databaseRec1.child("Tratamiento4").child("numPast").setValue(npast4)  
    }  
}
```

```
package com.example.apptt2  
  
import android.app.PendingIntent  
import android.content.BroadcastReceiver  
import android.content.Context  
import android.content.Intent  
import androidx.core.app.NotificationCompat  
import androidx.core.app.NotificationManagerCompat  
  
class AlarmReciver5:BroadcastReceiver() {  
    override fun onReceive(context: Context?, intent: Intent?) {  
        val intTra= intent?.getStringExtra("nombreTr")  
        val reci2Noti=Intent(context,OmisionNoti::class.java)  
        intent!!.flags=Intent.FLAG_ACTIVITY_NEW_TASK or  
Intent.FLAG_ACTIVITY_CLEAR_TASK  
        val penIntent5=  
PendingIntent.getActivity(context,0,reci2Noti,0)  
        //Construyendo la notificación de tratamiento 1  
        val builder= NotificationCompat.Builder(context!,"ECareCh2")  
            .setSmallIcon(R.drawable.ic_launcher_background)  
            .setContentTitle("Elite Care")  
            .setContentText("Omitiste la dosis de $intTra")  
            .setAutoCancel(true)  
            .setDefaults(NotificationCompat.DEFAULT_ALL)  
            .setPriority(NotificationCompat.PRIORITY_HIGH)  
            .setContentIntent(penIntent5)  
        val notificationManager=  
NotificationManagerCompat.from(context)  
        notificationManager.notify(101,builder.build())  
    }  
}
```



13. Anexos

Presupuesto de componentes y materiales:

Nombre	Descripción	Proveedor	Precio unitario	Cantidad	Precio total
Solenoide push-pull Rantec 5V	Actuador lineal eléctrico	UNIT ELECTRONICS	\$140.0	3	\$420.0
Raspberry Pi3 b+	Tarjeta de Procesamiento de datos	330 ohms	\$1040.0	1	\$1040.0
Arduino MEGA 2560	Tarjeta de control	Arduino	\$792.88	1	\$792.88
IRFZ44N	Transistor MOSFET	EDESUR ELECTRONICS	\$35.00	8	\$280.0
4N25	Optoacoplador	ELECTRÓNICA QUERETARO	\$35.00	3	\$105
PM55L-048	Motor a pasos bipolar	H-AVR	\$141.50	1	\$141.50
28BYJ-48	Motor a pasos unipolar	SANDOROBOTICS	\$55.0	1	\$55.0
A4988	Driver para motor a pasos bipolar	ABBASTER EAGLETEK	\$40.00	1	\$40.0
UNL2003	Driver para motor a pasos unipolar	330 ohms	\$40.0	1	\$40.0
Pantalla Waveshare	Pantalla táctil de 5'	Waveshare	\$995.5	1	\$995.5
XKC-Y26-NPN	Electroválvula de acción directa 2/2	Genebre	No disponible	1	\$1200.0



				(\$1200.0 aprox.)	
DHT11	Sensor de temperatura y humedad	TORRES2109	\$49.99	1	\$49.99
ACTECK INTEGRA 5R500	Fuente de poder	ddtech	\$428.99	1	\$428.99
Buzzer	Minibuzzer de 5V	Geek Factory	\$10.00	1	\$10.0
WINSINN USB	Ventilador enfriador 5V	Amazon	\$323.54	Por definir en TT2	\$970.62
				Se propone 3 de por ahora	
Pi3OV5647	Cámara para Raspberry Pi3	UCTRONICS	\$184.16	1	\$184.16
XKC-Y26	Sensor de nivel de agua capacitivo	CAIHUIMALLMX1	\$506.50	2	\$1013.00
Filamento PVC	Filamento de PVC para impresión 3D	3D JAKE	\$1019.98	2	\$2039.96

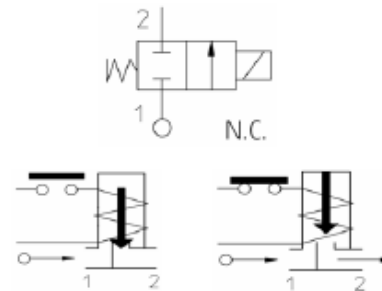
TOTAL: \$9856.10

Todos los precios de la tabla de presupuesto están en pesos mexicanos.

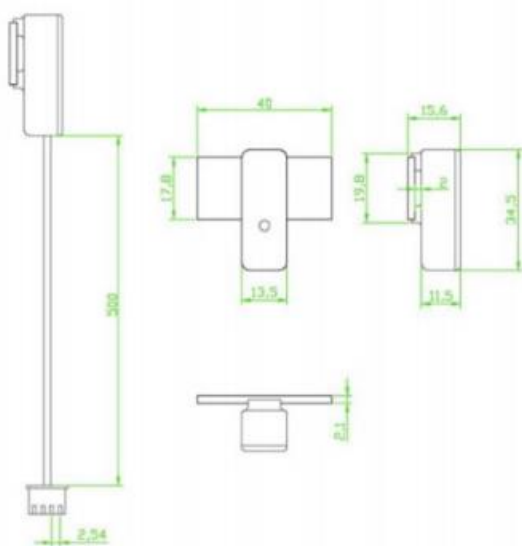
Art.: 4030

Electroválvula 2/2 Vías N.C. Acción Directa / Solenoid Valve 2/2 Ways N.C. Direct Acting

Características	Features
1. Electroválvula de acción directa 2/2 vías.	1. Direct acting solenoid valve 2/2 ways.
2. Normalmente cerrada N.C.	2. Normally closed N.C.
3. No requiere una presión mínima para su funcionamiento.	3. Minimum operational pressure is not required.
4. Máxima presión admisible (PS): 40 bar	4. Maximum allowable pressure (PS): 40 bar
5. Temperatura de trabajo: -10°C a +140°C.	5. Working temperature: -10°C to +140°C.
6. Cierre mediante juntas y diafragma en FKM (Vitón).	6. Sealing by FKM (Viton) gaskets and diaphragm.
7. Cuerpo de latón CW617N, según UNE-EN 12165	7. Body made in brass CW617N, acc/ UNE-EN 12165
8. Conexiones de rosca gas (BSP) según ISO 228/1	8. Gas (BSP) threaded connections acc/ ISO 228/1
9. Bobinas disponibles (ver tabla).	9. Coils available (see table).
10. Para aplicaciones generales en el campo de la automatización.	10. For general applications on automation field.
11. Adaptada para la interceptación de fluidos compatibles con los materiales en que está construida.	11. Adapted for the interception of fluids compatible with the construction materials.



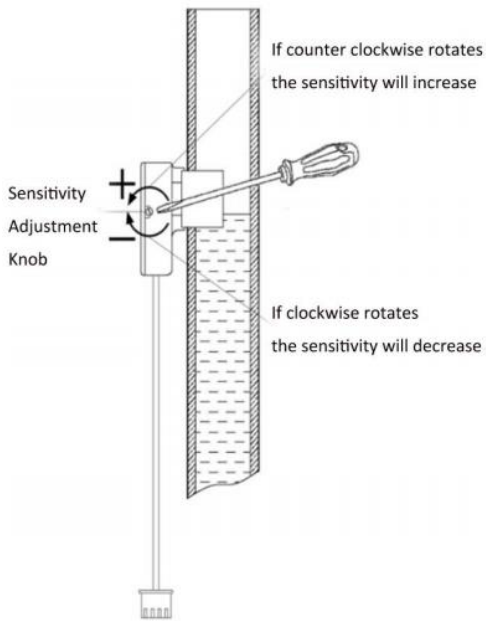
www.amazon.com/shops/icstation



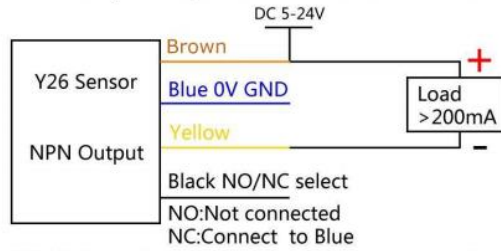
Product Introduction:

1. Model: XKC-Y26-NPN
2. Input Voltage: DC 5-24V (Suggest DC 5-12V)
3. Power Consumption: 5mA
4. Output Voltage (high level): Vin
5. Output Voltage (low level): 0V
6. Output Current: 1-100mA
7. Response Time: 500ms
8. Operating Ambient Temperature: 0-85 Celsius
9. Sensor Sensitivity: tube outer diameter (D:mm); sensor max tube wall (L:mm)
 $D \geq 100; 20 \pm 2$
 $100 > D \geq 80; 15 \pm 2$
 $80 > D \geq 60; 12 \pm 1.5$
 $60 > D \geq 40; 7 \pm 1.0$
 $40 > D \geq 30; 5 \pm 1.0$
 $30 > D \geq 20; 3 \pm 1.0$
 $20 > D \geq 10; 1.5 \pm 0.5$
10. Applicable Tube Diameter Range: $\geq 10\text{mm}$
11. Liquid Error: $\pm 1.5\text{mm}$
12. Humidity: 5%-100%
13. Material: ABS
14. Waterproof Performance: IP65

www.amazon.com/shops/icstation



Forward/NO Output: Black Hanging;
 Reverse/NC Output: Black connect to Blue cable.



NPN Output: Received an induction signal and output a low level.

NO: The sensor senses the liquid level and outputs a low level.

NC: The sensor does not sense the liquid level and outputs a low level.

NOTE:

1. The Load can be Relay, Buzzer, Speaker, LED indicator;
2. The maximum output current is 200mA;
3. Pay attention to the polarity of the load and power supply.

Minebea

PM55L-048



PM Type
Motors

■ 参考特性 Reference Characteristics

モータサイズ Motor Size	PM55L-048	
ステップ数 Number of Steps per Rotation	48 (7.5°/Step)	
駆動方式 Drive Method	2-2相 2-2 PHASE	
駆動方式 Drive Circuit	ユニポーラ定電圧 UNIPOLAR CONST. VOLT.	バイポーラチョッパー BIPOLAR CHOPPER
駆動電圧 Drive Voltage	24 [V]	24 [V]
電流/相 Current / Phase		800 [mA]
巻線抵抗/相 Coil Resistance / Phase	30 [Ω]	5.5 [Ω]
駆動 IC Drive IC	25C3346	8D463877EFV
マグネット材質 Magnet Material	フェライトプラスチック (MSPL)	
	Ferrite plastic magnet (MSPL)	
	極異方性フェライト磁鉄 (MS50)	
	Polar anisotropy ferrite sintered magnet (MS50)	
絶縁抵抗値 Insulation Resistance	100M [Ω] MIN	
	絶縁耐圧 Dielectric Strength	
	AC 500 [V] 1 [min]	
絶縁種 Class of Insulation	CLASS E (U-lead線を除く except lead wire)	
動作温度範囲 Operating Temp.	-10 [°C] - 50 [°C]	
保存温度範囲 Storage Temp.	-30 [°C] - 80 [°C]	
動作湿度範囲 Operating Hum.	20 [%] RH - 90 [%] RH	

■ 用途 Applications

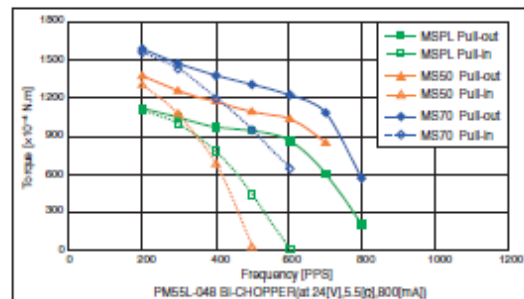
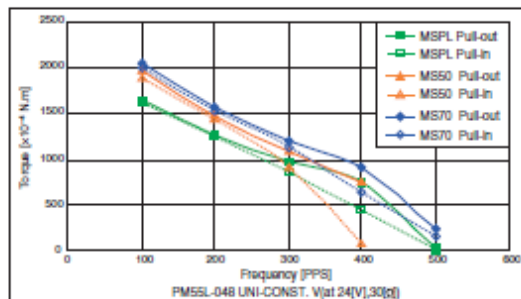
OA 機器：プリンタ、スキャナ
 OA Equipment : Printers / Scanners

産業機器：流量制御バルブ
 Industrial equipment : Flow control valves

玩具：スロットマシン
 Toys : Slot machines

HA 機器：ミシン
 Home automation appliances : Sewing machines

■ トルク特性 Torque Characteristics



このトルク値は一般参考値です。ご使用の際は放熱条件やデューティにより温度上昇が異なりますので、実機での熱条件を考慮してモータを選定下さい。
 These torque values are reference only. Heat radiation conditions and temperature rise effect by duty are different on each equipment,
 therefore please select motors after considering the actual equipment.

28BYJ-48 – 5V Stepper Motor

The 28BYJ-48 is a small stepper motor suitable for a large range of applications.



Rated voltage :	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stride Angle	5.625°/64
Frequency	100Hz
DC resistance	50Ω±7%(25°C)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz)
Self-positioning Torque	>34.3mN.m
Friction torque	600-1200 gf.cm
Pull in torque	300 gf.cm
Insulated resistance	>10MΩ(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K(120Hz)
Noise	<35dB(120Hz, No load, 10cm)
Model	28BYJ-48 – 5V

