



INSTITUTO POLITÉCNICO NACIONAL



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y
TECNOLOGÍAS AVANZADAS

Ingeniería en Sistemas Automotrices

PROYECTO INTEGRADOR

**“SISTEMA CONFIGURACIÓN DE
PERFILES DE CONFORT”**

Desarrollado por los alumnos

Temblador Ramírez Marco Antonio
Varona Hernández Jesús Alejandro

Asesores

Dr. Jaramillo Gómez Juan Antonio

Ciudad de México, a 11 de mayo del 2022



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN
INGENIERÍA Y TECNOLOGÍAS AVANZADAS

UPIITA



SISTEMA CONFIGURACION DE PERFILES DE CONFORT

QUE PARA OBTENER EL TÍTULO DE

“INGENIERO EN SISTEMAS AUTOMOTRICES “

PRESENTAN LOS ALUMNOS:

Jesús Alejandro
Varona Hernández.

Marco Antonio
Temblador Ramírez.

ASESORES:

Dr. Juan Antonio Jaramillo Gómez

SINODALES:

M. en C. Arodí Rafael Carvalho
Domínguez

Dra. Gabriela Vélez Martínez

CIUDAD DE MÉXICO A 11 DE MAYO DEL 2022



Autorización de uso de obra

Instituto Politécnico Nacional

Presente

Bajo protesta de decir verdad el que suscribe Temblador Ramírez Marco Antonio (se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los derechos morales y patrimoniales de la obra titulada Sistema Configuración de Perfiles de Confort, 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 11 de mayo de 2022

Atentamente



Autorización de uso de obra

Instituto Politécnico Nacional

Presente

Bajo protesta de decir verdad el que suscribe Varona Hernández Jesús Alejandro (se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los derechos morales y patrimoniales de la obra titulada Sistema Configuración de Perfiles de Confort, 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 11 de mayo de 2022

Atentamente



I Agradecimientos y dedicatorias

I.1 Agradecimientos

Queremos agradecer en esta sección, en primer lugar, al Instituto Politécnico Nacional, institución que nos brindó la oportunidad de formarnos como profesionales en la carrera de Ingeniería en Sistemas Automotrices, siendo esta una universidad de primer nivel. Agradecemos también las facilidades brindadas, ya fueran instalaciones, actividades, materiales, conocimiento, entre muchas cosas más.

No podemos mencionar a todos y cada uno de los profesores que formaron parte de nuestra educación, pero se deja en claro un enorme aprecio, respeto y admiración. Se les agradece, sobre todo, cada hora que dedicaron para compartirnos sus conocimientos y experiencias.

Agradecemos a nuestros amigos por cada uno de los momentos que pasaron a nuestro lado, siendo momentos de estrés, alegrías, tristezas, decepciones, logros y muchas aventuras. Agradecemos a todos los que en su momento nos estiraron la mano para brindarnos apoyo y ayuda, ya que nunca se permitió dejar a nadie atrás.

Finalmente agradecemos a las personas más importantes en nuestras vidas, las cuales son cada uno de los miembros de nuestras familias: hermanos y padres que estuvieron siempre como pilares sosteniendo este sueño y apoyando en cada uno de los conflictos y dificultades que se nos llegaron a presentar, que no nos dejaron rendirnos nunca y sacrificaron muchas cosas para poder alcanzar este logro.



I.2 Dedicatorias

Quiero dedicar este proyecto a mis padres, que sacrificaron tiempo, bienes, gustos, entre otras cosas, para que este momento pudiera llegar. Son ellos quienes pusieron siempre mis necesidades antes que las de ellos y estuvieron esperando tal vez con las mismas ansias que yo, durante 4 años y medio, ver este momento llegar y estoy seguro de que todas las veces que llegué a sufrir durante este tiempo ellos también lo sintieron, así como también disfrutaron de todas las metas logradas que me trajeron a este punto.

A mi hermano que tal vez sin darse cuenta se convirtió en una de las mayores motivaciones que tuve durante todo este tiempo, ya que decidí volverme un buen ejemplo para que él siguiera mis pasos correctos y mejorara así en los que me equivoqué yo.

A mis amigos que se volvieron como hermanos, sin necesidad de pedírselo me volvieron parte de ellos y jamás dejaron que yo o ninguno de nosotros nos quedáramos atrás.

Por último, a mis tías, tíos y abuelos que me apoyaron en todo momento con lo que pudieron para hacer que tuviera las menores complicaciones posibles.

Jesús Alejandro Varona Hernández

Quiero agradecer en primer lugar a mis padres por todo el apoyo brindado, así mismo, por cada una de las preocupaciones que atravesaron y por los no pocos sacrificios que realizaron durante todos estos años. Agradezco a mi padre por su perseverancia para salir a trabajar todos los días, siempre empeñado en que mis hermanos y yo no sufriéramos la falta de alimento, vivienda y vestimenta ni un solo día. Dentro de sus posibilidades, se aseguró de darme el dinero suficiente para que yo asistiera a la escuela y no cabe duda de que es gracias a él que continué estudiando. Su enseñanza estricta influyó en mí desde pequeño y me volvió, como él, un hombre honesto y disciplinado.

Agradezco encarecidamente a mi madre, quien me levantaba por la madrugada para cerciorarse de que no llegara tarde a mis clases, y quien siempre quería estar segura de que llevara algo de comida a la escuela. Quiero agradecerle por los



Ingeniería en Sistemas Automotrices



desvelos que debió resistir con tal de ayudarme cuando yo tenía proyectos por concluir hasta altas horas de la noche. Persona persistente e incansable, sin su ejemplo, jamás habría logrado ser lo que soy ahora.

Agradezco también a mis hermanos por el apoyo que me proporcionaron, y también, por su paciencia y comprensión cuando llegué de la escuela con demasiado estrés y por lo tanto, no en la mejor disposición. En más de una ocasión, pude acudir a ellos para resolver mis dudas, y no recibí sino ánimo y optimismo de su parte. Aun cuando, debido a las dimensiones de nuestro hogar, vivimos apretados y debemos compartir áreas comunes, me permitieron, sin excepción, realizar mis tareas y actividades en la habitación, sabiendo yo que los estaba privando de comodidad para poder dormir.

Quisiera expresar mi agradecimiento al resto mi familia, por todos sus consejos y respaldo, ya que, si bien no es posible verlos diariamente, en nuestras reuniones me infundieron confianza y ayuda de una u otra forma.

Agradezco a los compañeros que tuve a lo largo de la carrera, por todos los momentos de alegría y adversidad que atravesamos, donde las noches de desvelo no fueron escasas. Sobre todo, agradezco la diversión que me proporcionaron y es que, aunque éramos pocos, encontramos en este pequeño núcleo una amistad sincera.

Por último, quiero agradecer a mis profesores, tanto por los conocimientos que me compartieron, como por el profesionalismo con el cual atendieron mis preguntas, me prodigaron su ayuda o impartieron las clases de las cuales aprendí tanto y que enriquecieron no sólo mi formación académica, sino mi condición de persona.

Marco Antonio Temblador Ramírez



II Tabla de contenido

I	Agradecimientos y dedicatorias	i
I.1	Agradecimientos	i
I.2	Dedicatorias	ii
II	Tabla de contenido	v
III	Resumen	viii
III.1	<i>Abstract</i>	ix
IV	Objetivos	xii
IV.1	Objetivo general	xii
IV.2	Objetivos particulares	xii
1	Introducción	1
2	Antecedentes	4
2.1	Estado del arte	4
2.1.1	¿Qué es Ford MyKey y por qué es tan importante si tienes hijos?	4
2.1.2	¿Cómo usar los asientos de memoria?	5
2.1.3	Pase y encendido	6
2.2	Fundamentos teóricos	8
2.2.1	¿Qué es la tecnología RFID?	8
2.2.2	Sensor de huellas dactilares	8
2.2.3	Pantalla táctil	9
2.2.4	Funcionamiento de un limitador de velocidad	10
2.3	Metodologías	12
2.4	Herramientas de trabajo	13
3	Planteamiento del problema	16
3.1.1	Problemas con las llaves convencionales	16
Robo de vehículos	16	
3.1.2	Conductores sin precaución	16
3.1.3	Asientos y retrovisores movidos	17
3.2	Requerimientos	18



3.3 Propuesta de solución	19
4 Desarrollo de la solución	23
4.1 Módulo de censado	23
4.1.1 Sensor RFID LECTOR MIFARE MFRC522	23
4.1.2 Sensor Lector de huellas dactilares As608.....	23
4.2 Módulo de procesamiento	25
4.2.1 ECU (<i>Engine Control Unit</i>)	25
4.3 Módulo de control	25
4.3.1 Microcontrolador Arduino	25
4.3.2 Servomotores	26
4.3.3 Motorreductores	27
4.4 Módulo de comunicaciones.....	27
4.4.1 OBDII	27
4.4.2 Comunicación Serie Tx y Rx	28
4.5 Módulo de Interfaz hombre-máquina.....	28
4.5.1 Pantalla Táctil Nextion.....	28
4.6 Integración de prototipo o sistema global	29
5 Validación del sistema	31
5.1 Validación del funcionamiento del módulo de censado	31
5.1.1 Sensor RFID	31
5.1.2 Sensor de huellas dactilares	32
5.2 Validación del funcionamiento del módulo de procesamiento	32
5.2.1 ECU	32
5.3 Validación del funcionamiento del módulo de control	33
5.3.1 Microcontrolador Arduino	33
5.3.2 Servomotores	34
5.3.3 Motorreductores	34
5.4 Validación del funcionamiento del módulo de comunicaciones	35
5.4.1 Comunicación Serial (RX y TX)	35
5.5 Validación del funcionamiento del módulo de Interfaz hombre-máquina	36
5.5.1 Pantalla Nextion	36



Ingeniería en Sistemas Automotrices



5.6 Validación Integral del sistema o prototipo	37
6 Conclusiones	39
7 Referencias	41
Anexo A	44



III Resumen

En el presente documento se describe un sistema de configuración de perfil, la configuración comienza con una etapa de seguridad en la que se cuenta con un brazalete o pulsera RFID que la cual será censada por un sensor que estará colocado en la puerta, el cual, al detectar la pulsera nos va a permitir abrir o cerrar los seguros mediante un botón que solo funcione cuando sea leído el brazalete, esto es, en un lapso de 3 segundos para evitar aperturas accidentales.

Este sistema también está inspirado en la parte del confort, es decir, en la necesidad de tener una postura adecuada a la hora de conducir, pero no se trata de sensores que calculen la posición idónea dependiendo de la estatura. Debido a que cada persona tiene una posición con la que puede percibir de una forma correcta el camino, así como los distintos obstáculos que se puedan presentar, este prototipo se encarga de permitir que el usuario ajuste la posición de los asientos y retrovisores mediante una pantalla táctil y una vez encontrada la posición deseada se podrán registrar en la memoria EEPROM de nuestro microcontrolador. Cuando un usuario secundario mueva estos componentes a otra posición, estos se reconfiguran en conjunto a otro sistema, el cual es el encendido del automóvil mediante un sensor de huellas dactilares que encuentra dentro del habitáculo y se encarga de escanear, digitalizar y procesar la impresión de la huella que registre para que nuestro microcontrolador compare estos datos y pueda identificar si está o no registrada, una vez que se hace la comparación nuestro microcontrolador manda un pulso de 2 segundos a un relevador conectado a los 12v de la batería del auto para que se active, este relevador está conectado a nuestro cable de ON, el cual es el encargado del encendido, una vez que el controlador realizó estos pasos procederá a realizar un acomodo automático de los asientos y retrovisores utilizando los datos anteriormente guardados en la memoria de nuestro microcontrolador.

En nuestro prototipo se cuenta con el registro de nuestra huella numero 1 o principal, con la opción de agregar 2 huellas más las cuales son parte de un subsistema de seguridad que consiste en la activación de un limitador de velocidad cuando el auto sea encendido por el usuario numero 2 o 3, esto pensando que el usuario principal



quiera restringir la velocidad para los usuarios secundarios. Nuestro limitador funciona mediante señales enviadas desde la computadora del automóvil a través del OBDII (*On Board Diagnostic*) y así saber en qué momento hacer el corte o límite de manera correcta. El microcontrolador procesa cuál huella está encendiendo el automóvil y activa un relevador que alimenta el limitador, para que, cualquiera de los usuarios cuente con una velocidad limitada y así poder tener su integridad segura como la del automóvil, asimismo, podrá proteger y alargar la vida del motor como la de otros componentes.

Todo podrá ser controlado o modificado mediante una pantalla táctil que nos facilitará el control de los componentes ya que los usuarios pueden visualizar lo que están haciendo mediante la interfaz gráfica, esto nos garantiza un funcionamiento óptimo, así como una mayor comodidad para los usuarios al manejarlo, todo esto en conjunto de un código que se desarrolló para que nuestro microcontrolador mande las órdenes recibidas desde nuestra interfaz.

Las aplicaciones que se le pueden dar a este prototipo son variadas, ya sea para uso particular, para mejorar la seguridad del automóvil y el confort de un usuario, o puede ser para uso de vehículos de transportes que suelen ser utilizados por más de un usuario, como, por ejemplo, personas que cuentan con varios automóviles trabajando en Uber, como también en flotillas de tráileres, combis, camiones, taxis e incluso en vehículos de policía que garantizarían así el uso correcto a las unidades.

III.1 Abstract

This document describes a profile configuration system, the configuration begins with a security stage in which there is an RFID bracelet or bracelet that will be registered by a sensor that will be placed on the door, which, when detecting the bracelet, it will allow us to open or close the locks by means of a button that only works when the bracelet is read, that is, in a period of 3 seconds to avoid accidental opening.

This system is also inspired by the comfort part, that is, the need to have a suitable posture when driving, but it is not about sensors that calculate the ideal position according to height. Because each person has a position with which they can correctly



perceive the truck as well as the different obstacles that may arise, this prototype is responsible for allowing the user to adjust the position of the seats and mirrors through a screen touch and once the desired position is found, it can be registered in the EEPROM memory of our microcontroller. When a secondary user moves these components to another position, they are reconfigured together with another system, which is the ignition of the car by means of a fingerprint sensor which is located inside the passenger compartment and is in charge of scanning, digitizing and printing. of the fingerprint that it registers so that our microcontroller can compare this data and can identify whether or not it is registered, once the comparison is made, our microcontroller sends a pulse of 2 seconds to an optocoupler that is connected to a relay connected to the 12v of the car and this is activated, this relay is connected to our on cable inside the car which is in charge of the ignition, once the controller carried out these steps it will proceed to carry out an automatic adjustment of the seats and mirrors using the previously saved data in the memory of our microcontroller.

In our prototype we are satisfied with the registration of our number 1 or main fingerprint, with the option of adding 2 more fingerprints which are part of a security subsystem which consists of the activation of a speed limiter when the car is started by user number 2 or 3, our limiter works by means of signals sent from the car's computer through OBDII (On Board Diagnostic) and thus know when to cut or limit correctly. The microcontroller processes which fingerprint is starting the car and activates a relay that feeds the limiter, so that any of the users have a limited speed and thus be able to have their safe integrity like that of the car, likewise, they can protect and extend life engine as well as other components.

Everything can be controlled or modified using a touch screen that will facilitate the control of the components since users can visualize what they are doing through the graphical interface, this guarantees us optimal operation, as well as greater comfort for users when handling it.

The applications that can be given to this prototype are varied, either for private use, to improve the safety of the car and the comfort of a user, or it can be for the use of transport vehicles that are usually used by more than one user, as, for example, people who have several cars working in Uber, as well as in fleets of trailers, combis, trucks, taxis and even in police vehicles that would thus guarantee the correct use of the units.



IV Objetivos

IV.1 Objetivo general

Diseñar y construir un sistema capaz de detectar las llaves-brazaletes para el ingreso al vehículo, así como almacenar la configuración de asientos y espejos retrovisores, y tener la posibilidad de limitar la velocidad para algunos usuarios con ayuda del brazaletes de “confianza”.

Incluir un lector de huellas digitales para la lectura de estas y así aumentar la seguridad del vehículo, donde este último puede ser el encargado de generar los perfiles y el control del limitador y los asientos, todo ello mediante una interfaz en una pantalla táctil.

IV.2 Objetivos particulares

- Diseñar y construir un sistema capaz de efectuar la identificación de brazaletes RFID, el cual tenga protección al agua en caso de lluvia debido su natural exposición al aire libre.
- Emplear una tarjeta de desarrollo que interprete los datos recibidos por sensores y mande las señales necesarias hacia la etapa de potencia para la activación de motores.
- Crear una interfaz gráfica sencilla y comprensible para el usuario y una base de datos para que el conductor pueda realizar la configuración de los diferentes brazaletes.
- Llevar a cabo la programación de la tarjeta de desarrollo ya que esta será el cerebro de todo el sistema.

Implementar un sensor de huellas dactilares capaz de distinguir entre los diferentes usuarios.

- Diseñar un código capaz de guardar los datos de posición de los asientos y los retrovisores del automóvil mediante servomotores.



1. Introducción

El ser humano suele siempre buscar el confort en el mundo, y gracias a eso, surgen necesidades que cumplir que se logran mediante trabajos de ingeniería para así poder hacer más fácil la vida o la rutina de las personas en su día a día. En este caso, suele buscarse siempre el confort en la forma de conducir, buscando tener una posición cómoda para visualizar cada parte de nuestro vehículo, también se tiene siempre la necesidad de tener protegidos sus bienes, en este caso su vehículo, desde el control de quien tiene acceso a nuestro automóvil hasta la forma en que lo conducen otras personas.

En este sistema se trabajará con un brazaletes o pulsera RFID para asegurar la entrada al vehículo, en la sección de fundamentos teóricos se podrá obtener información más a detalle sobre este tipo de tecnología. Una vez ubicados dentro del automóvil nos encontraremos con la siguiente parte del sistema, el cual es un sensor de huellas dactilares, su propósito es generar otra etapa de seguridad, pues al escanear la huella dactilar se verificará que esté registrada e identificará qué usuario es para así proceder a la siguiente etapa de seguridad, la cual será la activación de un limitador de velocidad para los usuarios secundarios de la unidad, garantizando así que se conduzca de una manera adecuada. En cuanto al confort del automovilista tratamos el tema de los ajustes de los asientos y retrovisores, los cuales suelen ser utilizados de diferente forma dependiendo de cada tipo de persona. Mediante la memoria del microcontrolador se guardará la configuración del usuario para evitar la incomodidad de hacer un reajuste cuando otra persona usa la unidad. Con este sistema se busca mejorar la seguridad tanto del vehículo como de los usuarios, y también generar más comodidad para ellos, pero además, se obtienen beneficios como lo son el alargar la vida del motor, ya que al limitar la velocidad de los usuarios se les obliga a que hagan un uso correcto al no exigirle de más al motor y se utiliza en óptimas condiciones, por lo que nos permite una mayor duración, lo mismo en todos los sistemas y subsistemas que tienen que ver con el motor como lo son el árbol de levas, bielas, etc., y por lo tanto al mejorar el uso y duración de todos estos componentes logramos obtener un



Ingeniería en Sistemas Automotrices



ahorro en cuanto a combustible y esto permite también generar un cierto impacto ambiental al reducir los gases contaminantes que se puedan emitir.

Este sistema nos puede traer distintas ventajas en su implementación con sus deferentes subsistemas, en la parte de seguridad tenemos como ventaja el hecho de hacer más seguro nuestro automóvil ya que el sensor RFID solo abre la cerradura del automóvil por 3 segundos para evitar que accidentalmente se abra esta y alguien ajeno pueda entrar a nuestro auto, pasando a la parte del sensor de huellas dactilares se gana una gran seguridad ya que al no ser encendido con una llave previene aún más el riesgo de robo debido a que las huellas son únicas e irrepetibles. Este sensor permite también tener aún más seguridad con nuestro vehículo al poder diferenciar entre las diferentes huellas registradas y poder activar un limitador de velocidad al encender el auto con alguna de las dos huellas secundarias. En cuanto al confort evitamos el realizar un ajuste cada que otra persona utiliza el automóvil ya que al guardar cada perfil la posición deseada solo basta con encender al auto con su huella para que retrovisores y asientos realicen el acomodo automático justo como se guardó en la configuración.



Esta hoja está en blanco intencionalmente



2. Antecedentes

2.1 Estado del arte

2.1.1 ¿Qué es Ford MyKey y por qué es tan importante si tienes hijos?

La empresa de automóviles Ford ha creado un sistema de seguridad que nos da una cierta garantía de que el auto será utilizado de manera correcta y bajos límites establecidos dando así tranquilidad a los padres para que estos puedan permitir el uso de su automóvil a sus hijos ya que este sistema bloquea funciones que para un principiante se pueden considerar distracciones que pueden generar un accidente.[7].

El secreto tras todo el éxito obtenido por parte de este sistema de Ford es mediante el uso de distintas llaves que cuentan con distintas configuraciones, con las cuales su principal funcionamiento es el restringir o limitar la velocidad máxima alcanzada por el auto, así como también el negar la cancelación de asistencia de conducción que intervienen en el desempeño de estos vehículos.[7].

Una gran función que tiene este sistema es que mediante la llave principal se pueden activar funciones a las otras llaves como lo son la asistencia de emergencia con la cual un dispositivo manda una señal de alerta en caso de que ocurra algún percance o accidente para que así pueda llegar la ayuda necesaria y poder dar tranquilidad a los padres.[7].



Ilustración 1. Menú Ford MyKey

2.1.2 ¿Cómo usar los asientos de memoria?

Algunos autos Chevrolet de gama media alta cuenta con un sistema de asientos con memoria los cuales funcionan a través de dos llaves o dispositivos que se utilizan para acceso o encendido del automóvil, con estas se programa y configura un perfil por cada llave en las cuales se registrarán datos como las posiciones de los asientos de los asientos entre otras funciones las cuales serán acomodadas y recordadas al encender y configurar el auto si así lo desean los usuarios.[8].

Al encender el automóvil con cualquiera de las llaves y una vez que el tablero se enciende nos muestra en el centro de este el número de llave que es para saber que perfil se está configurando.[8].



Ilustración 2. Asientos con memoria Chevrolet

A continuación, se muestran una serie de comando que se suelen utilizar en este sistema de Chevrolet para la configuración de asientos en los diferentes perfiles.

¿Cómo configuro mi asiento?

Si está equipado, en el panel de la puerta del conductor, o ubicación similar según el vehículo, una variedad de botones, etiquetados "SET", "1", "2" y un ícono que indica "exit", son específicos para guardar las posiciones de la memoria. Estos botones te permiten configurar y recordar las posiciones del asiento.[8]

1. Antes de guardar, ajusta todas las posiciones disponibles de la memoria, incluyendo el asiento, los espejos y la columna de dirección hidráulica, si está equipado.
2. Coloca el sistema de encendido en la posición "ON", si el vehículo aún no está en marcha.



3. Presiona y suelta el botón "SET" y luego debería sonar un pitido.

4. Mantén presionado inmediatamente el botón con el número que coincide con tu llave/control remoto tipo llavero RKE (ver arriba) hasta que suenen dos pitidos.

5. Se completó la programación. Suelta el botón.

2.1.3 Pase y encendido

Siemens VDO alrededor de 1998 diseñó una alternativa para el acceso al vehículo la cual es el sistema PASE que permite el acceso sin la necesidad de utilizar una llave sobre la cerradura, esto funciona con sensores que localizan la llave con señales de radiofrecuencia donde se envían mensajes programados únicamente para cada llave.[9].

Tiene dos formas de funcionamiento la primera es cuando la llave se encuentra fuera del vehículo, y mediante diferentes algoritmos al agarrar la manija del auto analiza las señales enviadas y recibidas entre la llave y los sensores dentro del auto y permite el acceso abriendo los seguros, a esta función se le suele llamar función *ENTRY*, la segunda función comienza cuando la llave se encuentra dentro del automóvil y se presiona el botón de encendido del auto otra serie de algoritmos procesan y buscan señales con sensores ahora dentro del habitáculo con las que se comunican la llave para permitir el encendido del automóvil únicamente para llaves con el chip autorizado, a esta función normalmente se le conoce como *START*. [9].

Se han creado diferentes sistemas en base a este con actualizaciones o mejoras como lo son la función de abrir las puertas del auto cuando aún no se toca la cerradura, esto mediante sensores que leen y procesan información con lo que determinan que el usuario se está acercando o en el caso contrario, que el usuario ha salido del auto y se está alejando para así cerrar los seguros y activar nuestra alarma.[9].

Gracias a este tipo de sistemas podemos agregar más seguridad a nuestro automóvil el ser chips empleados únicos para cada computadora de cada vehículo, agregando también confort por el hecho de no necesitar una completa interacción con la cerradura del auto y evitando procesos únicamente para hacer más cómoda la experiencia del usuario, la comodidad o confort que se tenga ya dependerá de la



gama de nuestro vehículo debido a que existen sistemas que abren la puerta con la cercanía de la llave evitando así el uso de una manija. [9].

Podemos observar que este es un sistema muy eficiente, cómodo e incluso practico que nos brinda seguridad a nuestro automóvil, pero también podemos observar que debido a su complejidad suele ser complicado agregarle funciones o subsistemas que nos brinden aún más seguridad y aún más comodidad a la hora de utilizar un automóvil, esto nos dice que hay buen rango para mejorar estos sistemas o en su defecto la misma llave convencional. [9].

La historia de la llave ha pasado por diferentes etapas desde la llave convencional hasta las llaves que se autoidentifican solas, el problema con estas es la dificultad de mejorar o agregar subsistemas sin tener que agrandar el tamaño de la llave o dispositivo, ya que al agrandar el tamaño por la necesidad de tener mayor procesamiento le quitamos la practicidad y se vuelve un componente incomodo como para portarse dentro de un bolsillo [9].

No.		Nombre	Características	País	Instituto o compañía	Referencia
1		¿Qué es Ford MyKey y por qué es tan importante si tienes hijos?	Funciones del sistema MyKey	España	Ford	[7]
2		Cómo usar asientos con Memoria	Control de asientos con memoria	Estados Unidos	Chevrolet	[8]
3		Sistema acceso de encendido remoto del automóvil mediante el uso de Smartphone	Control de acceso y encendido	México	CIATEQ	[9]

Tabla 1. Estado del arte



2.2 Fundamentos teóricos

2.2.1 ¿Qué es la tecnología RFID?

La tecnología RFID (*Radio Frequency Identification* o en español identificación por radiofrecuencia), es un sistema de almacenamiento y recuperación de información de manera remota en tiempo real y automática, con el fin de identificar personas, objetos o animales por medio de la utilización de etiquetas, tarjetas o tags RFID mediante la comunicación inalámbrica entre las etiquetas y un lector de estas. Las etiquetas básicamente están compuestas de un chip donde se almacenan datos y de una antena que recibe y envía información, la que permite identificar al objeto al cual están escaneando.[10].

Un sistema RFID consta en su forma básica de etiquetas, antenas, lectores y finalmente el computador, en el que se encuentra algún tipo de aplicación de software y en donde se analiza la información de las etiquetas.[10].

2.2.2 Sensor de huellas dactilares

Un lector óptico funciona con un dispositivo CCD (*Charged Coupled Device*), como el usado en las cámaras digitales, que tienen un arreglo de diodos sensibles a la luz que generan una señal eléctrica en respuesta a fotones de luz. Cada diodo graba un píxel, un pequeño punto que representa la luz que le es reflejada. Colectivamente, la luz y perfiles oscuros forman una imagen de la huella leída. El proceso de lectura comienza cuando usted pone su dedo sobre la ventana del lector, el cual tiene su propia fuente de iluminación, típicamente un arreglo de LEDs, para iluminar las crestas de la huella digital. El CCD genera, de hecho, una imagen invertida del dedo, con áreas más oscuras que representan más luz reflejada (las crestas del dedo) y áreas más claras que representan menos luz reflejada (los valles entre las crestas).[1].

Antes de comparar la información obtenida con la almacenada, el procesador del lector se asegura de que el CCD ha capturado una imagen clara. Checa la oscuridad promedio de los píxeles, o los valores generales en una pequeña muestra, y rechaza la lectura si la imagen general es demasiado oscura o clara. Si la imagen es



rechazada, el lector ajusta el tiempo de exposición – para dejar entrar más o menos luz, e intenta leer la huella de nuevo.[11].

Si el nivel de luz es adecuado, el lector revisa la definición de la imagen (qué tan precisa es la imagen obtenida). El procesador busca varias líneas rectas que se mueven horizontal y verticalmente sobre la imagen, y si esta tiene buena definición, una línea que corre perpendicular a las crestas será hecha de secciones alternantes de pixeles muy claros y oscuros.[11].



Ilustración 3. Demostración de la impresión de huella

2.2.3 Pantalla táctil

Las pantallas táctiles funcionan de diferentes maneras. Algunas sólo pueden detectar un dedo a la vez y se confunden demasiado si se oprimen dos lugares diferentes al mismo tiempo, mientras que otras sí pueden manejar esta acción. A continuación, detallamos las diferentes tecnologías de algunas de ellas:[12]

- Pantallas táctiles resistivas

En términos sencillos, una pantalla táctil resistiva se encuentra compuesta de dos capas de película plástica conductora muy delgadas y transparentes, ubicadas sobre la pantalla propiamente dicha, lo que le permite medir el cambio en la resistencia producido al realizarse la conexión eléctrica debido a la unión de esas capas cuando la presionamos con el dedo o con el lápiz óptico.[12].

Las pantallas táctiles resistivas son mucho más baratas de fabricar que el resto, pero no ofrecen la posibilidad del *multitouch*, es decir, si presionamos con más de un dedo, el dispositivo captará la posición de sólo uno de ellos. Además, estas se descalibran con el paso del tiempo y es necesario recalibrarlas.[12].



- Pantallas táctiles capacitivas

A diferencia de las pantallas táctiles resistivas, las capacitivas se construyen uniendo un panel de vidrio recubierto con un material conductor transparente. Su funcionamiento básico se basa en aprovecharse de la capacidad del cuerpo humano de conducir electricidad, es decir que cuando el usuario toca la superficie de la pantalla activa un campo eléctrico, el cual es registrado e informa al *hardware* del dispositivo, traduciéndolo en comandos.[12].

Al contrario de las pantallas resistivas, una capacitiva ofrece una experiencia de uso mucho más suave, ya que sólo requiere que apoyemos los dedos sin realizar ninguna clase de presión, pero por otra parte no podemos usar un lápiz óptico regular, sino uno especialmente diseñado para estos dispositivos capacitivos, mucho más caros que un lápiz óptico común.[12].

2.2.4 Funcionamiento de un limitador de velocidad

Los primeros limitadores de velocidad surgieron en Estados Unidos como una respuesta práctica a sus carreteras interminables y los estrictos controles de velocidad, con el fin de que los conductores pudieran circular relajados sin necesidad de estar pendientes del velocímetro continuamente.[13].

Aquellos primeros limitadores eran mecánicos, más adelante surgirían los electrónicos, ubicados en el volante o en un lugar próximo. Lo último son los limitadores inteligentes de velocidad, que son los que quiere imponer la UE de forma obligatoria.[13].

Los limitadores que más se ven en la actualidad son electrónicos y se controlan mediante unos mandos ubicados en el volante. Pueden ser activos o pasivos:

- Activos: Aquellos mediante los cuales podemos fijar una velocidad que el coche no va a sobrepasar bajo ningún concepto. En algunos modelos actúan sobre frenos o cambio para evitar que se embale el coche.[13].
- Pasivos: En este caso, se limitan a advertir mediante una señal acústica o visual, que se ha excedido la velocidad. Su función es meramente informativa, ya que no limitan automáticamente la velocidad.[13].

¿Cómo funciona el limitador de velocidad?



Ingeniería en Sistemas Automotrices

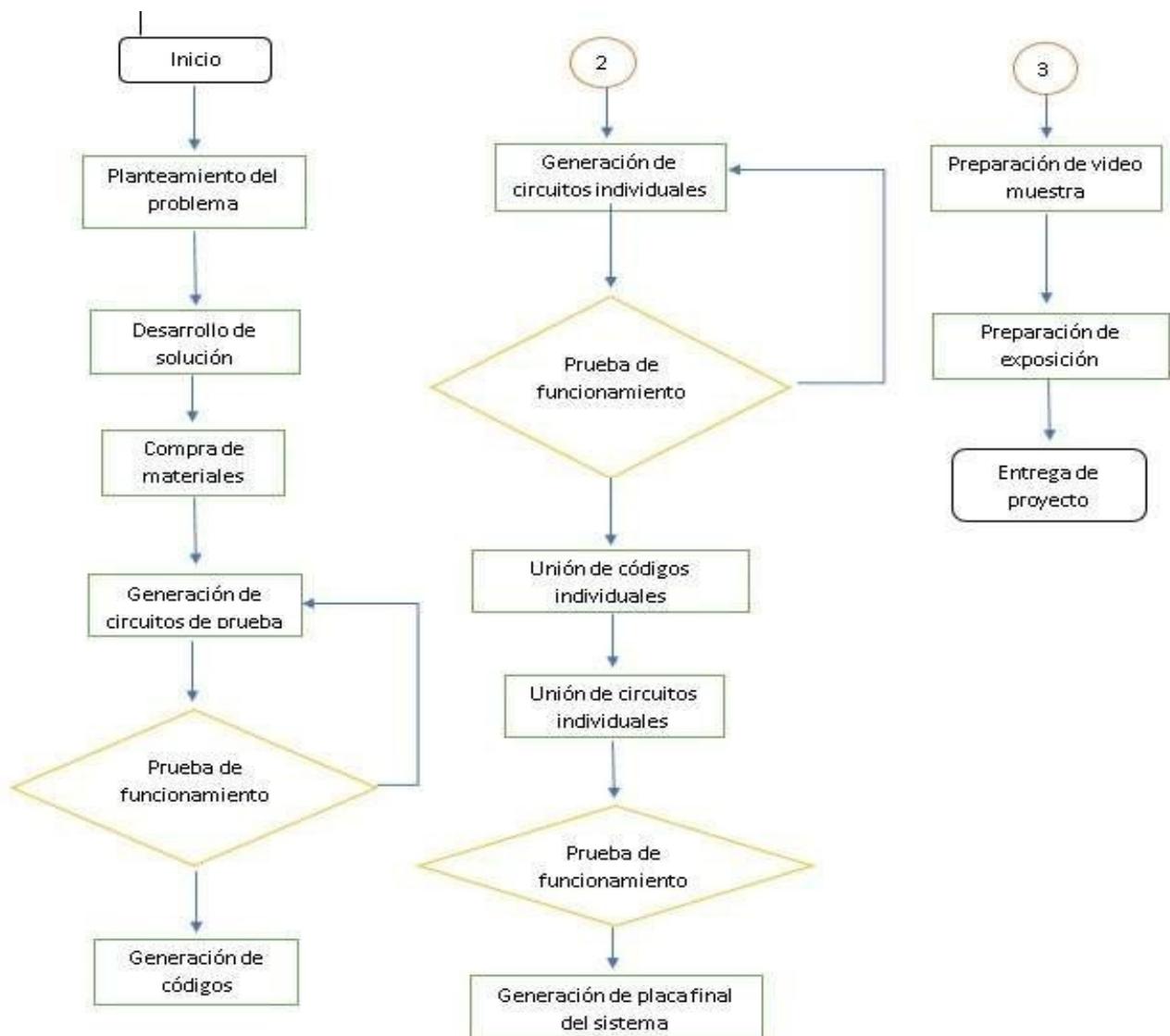


Su funcionamiento es muy sencillo. El conductor fija el límite que no quiere sobrepasar con la ayuda de un comando situado generalmente en la maneta izquierda o en el volante. De esta forma, cuando sobrepase dicha velocidad, el sistema avisará mediante una señal acústica o visual (limitadores pasivos), o controlará el flujo de combustible con que se alimenta el motor con el fin de mantener dicha velocidad máxima, por mucho que el conductor quiera acelerar (limitador activo).[13].



2.3 Metodologías

La metodología es un conjunto de procedimientos y técnicas que se aplican de manera ordenada y sistemática en la realización de un estudio; esta metodología determinará la forma en que se recaban, ordenan y analizan resultados. Para este proyecto integrador se ocupó un diagrama de flujo debido a que el problema a resolver es grande y se necesita llevar un orden correcto del procedimiento, así como una retroalimentación del mismo para futuras correcciones y/o innovaciones.





2.4 Herramientas de trabajo

Las herramientas de trabajo utilizadas en este sistema son:

✦ Arduino

Arduino es una compañía de desarrollo de *software* y *hardware* libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de *hardware* para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real.[14].

Su lenguaje de programación basado en C++ es de fácil comprensión. C++ permite una entrada sencilla a los nuevos programadores y a la vez con una capacidad tan grande, que los programadores más avanzados pueden expresar todo el potencial de su lenguaje y adaptarlo a cualquier situación.[14].

✦ Proteus

La característica más importante y que mejor define Proteus es su capacidad de simular el comportamiento de los circuitos electrónicos. Proteus incorpora una amplia librería de dispositivos analógicos y digitales con su correspondiente modelo SPICE que simula su comportamiento. Además, podemos crear nuestros propios dispositivos si disponemos del fichero SPICE correspondiente. De esta manera podemos comprobar el funcionamiento de nuestros diseños.[15].

El módulo de simulación avanzada nos posibilita la simulación de transitorios, frecuencias, transformadas de Fourier, curvas de transferencia, etc. Con él podemos visualizar los datos en gráficos analógicos, digitales y de buses de datos.[15].

Y los diferentes módulos VSM específicos de cada familia de microprocesadores nos permiten observar cómo interactúa el software que se ejecuta en un microcontrolador con cualquier tipo de componente electrónico, tanto digital como analógico, situado en su periferia, reduce considerablemente el número de horas utilizadas en los desarrollos de los equipos. Más de 800 versiones de microprocesadores diferentes se pueden simular en Proteus. Los modelos de microprocesadores simulan completamente los puertos de entrada/salida, las interrupciones, los relojes y



contadores, las *USART's* y todos los periféricos presentes en cada microprocesador soportado.[15].

✦ Nextion Editor

Nextion es una solución de interfaz de máquina humana (HMI) que combina un procesador integrado y una pantalla táctil de memoria con el *software* Nextion Editor para el desarrollo de proyectos HMI GUI.[16].

Con el *software* Nextion Editor, puede desarrollar rápidamente la GUI de HMI arrastrando y soltando componentes (gráficos, texto, botón, control deslizante, etc.) e instrucciones basadas en texto ASCII para codificar cómo interactúan los componentes en el lado de la pantalla.[16].

CARACTERÍSTICAS:

- Resolución 320 x 240.
- RGB 65K fiel a los colores de la vida.
- Pantalla TFT panel resistivo táctil.
- Fácil interfaz de 4 pines a cualquier TTL Host Serial.
- Memoria Flash 4M para Código de usuario, aplicaciones y datos.
- Ranura para tarjeta micro-SD para actualización de firmware.
- Área Visual: 57.6mm x 43.2mm.
- Brillo ajustable: 0 ~ 180 nit, el intervalo de ajuste es de 1%.
- Consumo de energía 5V65mA.



Esta hoja está en blanco intencionalmente



3. Planteamiento del problema

3.1.1 Problemas con las llaves convencionales

Las llaves y los autos han existido en conjunto prácticamente desde sus inicios, la llave fue pensada como sistema de seguridad y como medio de encendido del auto, y ha pasado de ser una pequeña placa de metal hecha para abrir una cerradura, pasando luego a incluir un chip en la parte superior que lee una antena, hasta ser simplemente un llavero con chip que al ser detectada por un sensor activa las funciones del auto sin necesidad de emplear la cerradura. Pero todos estos tipos de llave tienen un inconveniente en común y es la facilidad que tienen de perderse u olvidarse en cualquier situación e incluso puede ser robada, de esta manera, encontramos ya el primer problema.

Robo de vehículos

Un segundo problema son los riesgos que se corren al extraviar la llave o incluso en caso de robo, pues una vez que alguien ajeno a nosotros cuenta con la llave de nuestro automóvil, ya sea una llave normal, de chip o de llavero, le es posible encender y conducir el vehículo con pocas posibilidades de nuestra parte para recuperarlo.

3.1.2 Conductores sin precaución

Un problema común que se tiene con los automóviles es a la hora de que un tercero tenga que conducir nuestro vehículo y no se tiene una manera de supervisar la forma de manejo de este conductor, lo cual nos expone a problemas como el desgaste excesivo de nuestro motor junto con los subsistemas que trabajan en conjunto a este debido una conducción imprudente, o también, se corre el riesgo de tener un percance como un choque o un atropellamiento, lo cual implica problemas incluso legales, así como el hecho de que esté en riesgo la integridad de la persona que conducía la unidad.



Estos problemas pueden verse en flotillas de combis o camiones, ya que muchas personas saben la forma irresponsable de manejar de los choferes del transporte público o conductores de tractocamiones/tráileres que manejan, también, de una forma irresponsable. Al tener vehículos tan grandes y pesados a la hora de tener que frenar de emergencia no les es posible o simplemente con una familia en la que los hijos pidan el vehículo prestado se necesita una forma de mantenerlos seguros.

3.1.3 Asientos y retrovisores movidos

Es bien sabido que para conducir un automóvil es necesario tener una postura correcta que nos permita tener una óptima visualización del camino, los retrovisores, el cofre del auto, entre otras cosas y también se sabe que existen autos con sensores que, calculando de acuerdo con tu estatura, configuran la posición de los asientos y espejos retrovisores en una posición óptima y "cómoda". Pero todas las personas somos diferentes en cuanto a nuestra comodidad y en la mayoría de los casos estos sensores no colocan los componentes de la forma que el usuario considera idónea para la conducción y termina por ser un tanto inservible este sistema para la opinión de mucha gente.

Otro problema parecido es cuando no se cuenta con un sistema de sensores que configuren automáticamente los asientos y espejos, el dueño del vehículo puede ya tener colocado todo de la manera que él considera eficiente, pero al llegar la necesidad de que otro usuario tenga que conducir el vehículo colocara estos componentes de otra forma en la que se sienta cómodo y esto genera malestar e incomodidad para el usuario principal del automóvil.

Con el prototipo de nuestro sistema tratamos de mejorar la seguridad de entrada al vehículo implementando un sensor RFID que será capaz de leer y procesar los datos de pulseras que sustituirán plásticas que aseguran nuestro chip RFID contra el agua para que pueden portarse en todo momento y al ser sujetas a la muñeca se reduce el riesgo de robo o extravió, este sensor enviara los datos a nuestro microcontrolador, mismo que se encargara de la apertura o cierre de seguros. Para el tema de robo de vehículos se utiliza un sensor de huellas dactilares implementado para el arranque del auto, este escanea la huella del usuario para mandar los datos al microcontrolador y



este permita el encendido e inmediatamente comenzará a realizar un ajuste automático de asientos y retrovisores utilizando información sobre posición y ajustes que se podrá guardar en la memoria del micro y todo esto a través de una interfaz que está en nuestra pantalla táctil.

3.2 Requerimientos

Para el desarrollo de este sistema se tienen los siguientes requerimientos para lograr cumplir con los objetivos.

- Un módulo RFID que en este caso trabaje a 13.56 MHz con velocidad de transferencia de datos de 10Mbit/s máximo, para lectura.
- 2 pulseras RFID a prueba de agua que trabaje a 13.56 MHz con ajuste a la muñeca estilo reloj para la comodidad del usuario de solo lectura.
- Un lector de huellas digitales con pines de comunicación Rx y Tx para la conexión con el microcontrolador, con opción de registro de huella y Proceso de ajuste de huellas dactilares.
- Pantalla *Touch* con comunicación serial por Rx y Tx para la conexión con el microcontrolador.
- Limitador de velocidad con conexión por OBDII.
- 4 servomotores para el control de posición electrónico de 20kg.
- 2 moto reductores para el control de posición de los asientos.

3.3 Propuesta de solución

En la siguiente imagen podemos observar la representación de conexión de nuestro sensor RFID con nuestro microcontrolador. La pulsera o brazalete RFID es detectado por nuestro sensor y manda un pulso a un led de color verde que funciona como indicador y de igual manera hace girar el servomotor simulando la apertura de seguros del automóvil durante un intervalo de tiempo de aproximadamente 5 segundos, después lo girara en sentido contrario para representar el cierre de los seguros, en el caso de que el sensor RFID detecte un dispositivo no registrado encenderá únicamente un LED de color rojo indicando así que es el brazalete o pulsera incorrecto.

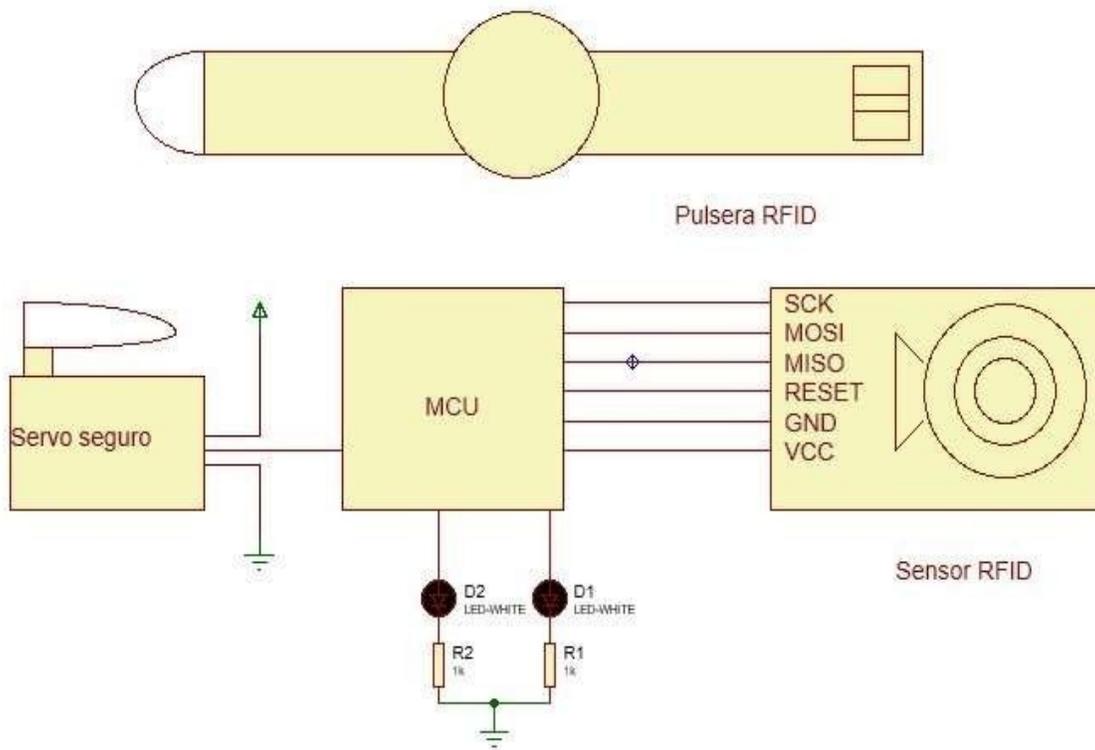


Ilustración 4. Conexión del sensor RFID y el MCU

La ilustración que se encuentra debajo representa la conexión del lector de huellas junto al limitador de velocidad con el microcontrolador, todas estas conexiones nos llevan al siguiente paso el cual es el encendido del automóvil.

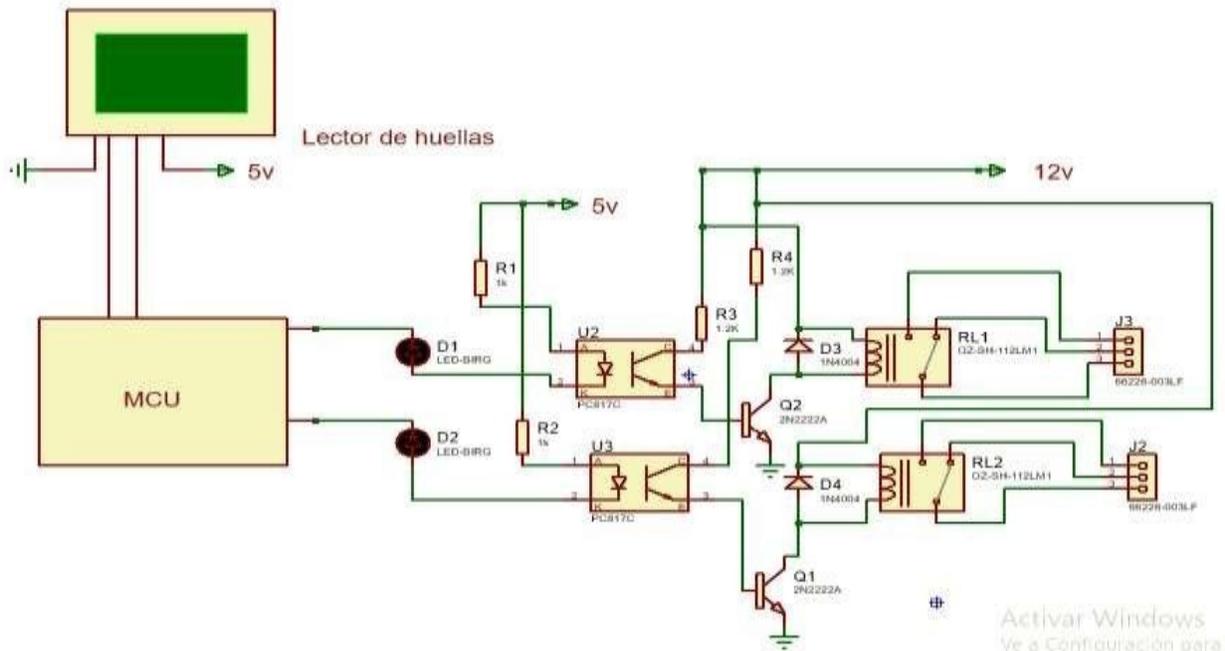


Ilustración 5. Conexión del lector de huellas y el limitador de velocidad con el MCU

La persona una vez dentro del vehículo tendrá que poner su dedo en el detector de huellas digitales, este mandará las señales recibidas al MCU y se detecta si el usuario es principal, secundario o no está registrado. Si la persona es el usuario principal el automóvil encenderá sin ningún problema, de ser el secundario se mandarán señales para que el limitador de velocidad comience su funcionamiento y en caso de que el usuario no este registrado el automóvil no encenderá de ninguna manera.

Para que el sistema sepa que huella digital es la principal o no, debemos de tener un subsistema que nos permita iniciar con todo el proceso de registro de huellas. Esto se lleva a cabo con una pantalla táctil, la cual esta conectada al MCU y programada para su fácil uso dentro del habitáculo del automóvil.

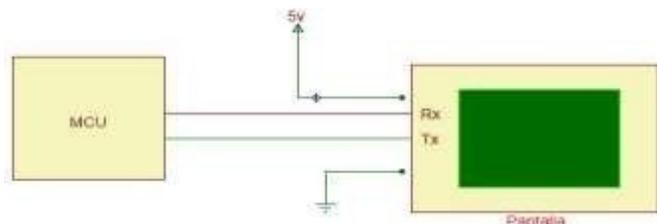


Ilustración 6. Conexión de pantalla táctil con el MCU

Como se ha mencionado anteriormente este no solo es un sistema que aporta en la seguridad del automóvil, si no que también al confort del conductor. ¿De qué manera? al colocar tu dedo en el sensor de huellas y el sistema detecte que eres un usuario correcto, ya sea principal o secundario, no solo encenderá el vehículo y en su caso el limitador de velocidad si no que acomodara los espejos retrovisores y el asiento del conductor conforme sea haya registrado en el sistema. En la imagen de abajo se observa la conexión de 4 servomotores que representan los dos espejos retrovisores del vehículo.

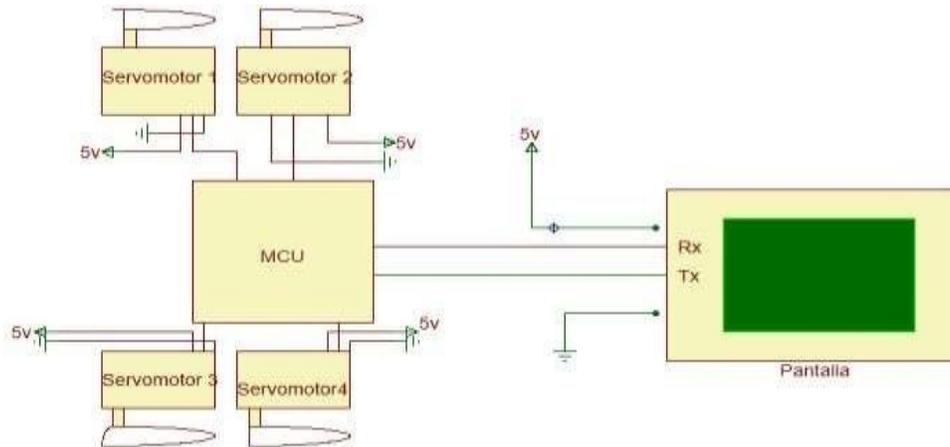


Ilustración 7. Conexión de servomotores y pantalla táctil al MCU

Todo el registro de posición de espejos retrovisores y asiento del conductor se lleva a cabo de igual manera en la pantalla táctil, la cual tiene una interfaz gráfica fácil de comprender para la mayoría de las personas. En la ilustración de abajo observamos la conexión de 2 motorreductores al MCU, los cuales simulan los motores que mueven al asiento de manera horizontal y la inclinación del mismo.

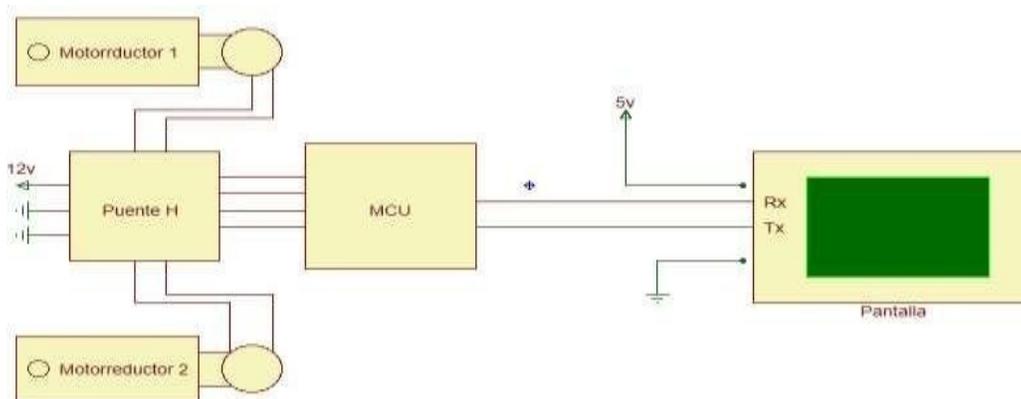


Ilustración 8. Conexión de motorreductores y pantalla táctil al MCU



Esta hoja está en blanco intencionalmente



4. Desarrollo de la solución

4.1 Módulo de censado.

En esta sección encontraremos un pequeño resumen del proceso de cada componente utilizado en el módulo de procesamiento.

4.1.1 Sensor RFID LECTOR MIFARE MFRC522

Esta parte del sistema tuvo pocas modificaciones de inicio ya que una vez terminado un circuito básico el cual solo incluía en su primera etapa un Sensor RFID, dos pulseras, un microcontrolador Arduino Mega y una tablilla de pruebas con los cuales se utilizó primeramente un código de prueba, el cual con el monitor serial del software Arduino nos muestra el código que tienen escrito los brazaletes/ pulseras.

Una vez obtenidos estos códigos se desarrolló un programa, el cual al leer un brazalete, tarjeta, llavero o cualquier otro dispositivo RFID que no esté registrado, procederá a encender un led de color rojo que indicará que no es válido el usuario y, cuando lea una de las pulseras, encenderá un led verde y moverá un servomotor que simulará el seguro de una puerta de automóvil.

Una vez que funcionó esta parte del sistema se agregó al código principal para funcionar en conjunto con los demás sistemas.

4.1.2 Sensor Lector de huellas dactilares As608

Para el desarrollo de esta sección del sistema se realizaron pruebas con códigos ejemplo que existen para este sensor, todo esto para analizar y comprender el funcionamiento del sensor. Primeramente, se utilizó un código que permite agregar una huella dactilar a la memoria del sensor (*enroll*) y después se probó un código para borrar la huella (*delete*), los cuales son parte de las librerías.

Una vez comprendidas las funciones que tiene el sensor, se procedió a modificar un código de ejemplo que compara la huella dactilar escaneada por el sensor y, si está



registrada, encenderá un led y moverá la posición de un servomotor, una segunda parte en esta prueba fue el agregar una segunda huella al sensor para así compararla y dependiendo de la huella dactilar que se utilizara, encendería un led de color blanco si es la huella 1 y otro de color naranja si es el usuario 2. Una vez censado las huellas, si estas son las de los usuarios 1 o 2 el monitor serie da la bienvenida y en el caso de que el sensor lea una huella que no está registrada procederá a encender un led de color rojo apagándolo en 3 ocasiones, y de igual manera arrojaría un mensaje en el monitor serie leyéndose que no es correcta la huella. A esto se agregó una prueba extra en la que al escanear la huella 1 se encendería el led de color blanco nuevamente y se vería el mensaje en el monitor serie, en el caso de la huella 2 se activaría el mismo led blanco y un led anaranjado para simular el encendido del limitador, donde también se vería un mensaje de bienvenida y uno que diría “Usted cuenta con una velocidad limitada, limitador activado”.

Para el siguiente paso se procedió a comenzar un código que combine las diferentes funciones que se probaron en la primera parte de este sistema.

Lo primero fue combinar el código en el que se escaneaba la huella con el que podía agregar una huella mediante 4 botones. En este código, el primer botón es el botón de escanear, el cual lo que hace es mandar a una función dentro del código que inicia el proceso de lectura del dedo y en el cuerpo principal del código se hace el análisis de qué órdenes procederá a ejecutar, las cuales seguían siendo las mismas que en la prueba pasada. El segundo botón es el de agregar huella, el cual al ser presionado se dirige a una función dentro del código que inicia el proceso de escanear para agregar una nueva huella, y mediante el monitor serie y el teclado de nuestro computador se envía un número, el cual debe ser entre 2 y 170 que son los ID restantes dentro del sensor de huellas. Una vez que se escribe este número, es leído por el código y asignado a una variable, este proceso se hace dentro de una función para seleccionar el ID, y este procede a registrar la huella y en este caso no se utilizaron los dos botones restantes.

En la segunda parte de este código se procedió a sustituir el proceso de lectura del número en el monitor serie por los dos botones restantes, ahorrando así líneas de código, evitando una función ya que en el sistema solo se planea que se puedan agregar máximo dos huellas más, por lo que el primer botón al ser seleccionado envía el ID 2 a la función de agregar huella y el botón dos envía el ID 3, y en este mismo



código se puede hacer la comparación de estas y proceder a ejecutar las órdenes correspondientes., en el anexo 3 podemos observar un diagrama de conexión.

En la tercera se agregaron 3 botones más y dentro del código se introdujo la función "delete", la cual entra al presionar el quinto botón, después en el monitor serie se puede observar un mensaje que dice que seleccione qué ID desea borrar y eso es mediante el botón número 6 y el botón número 7, que mandan el ID 2 o 3 respectivamente, y una vez seleccionado se borra el usuario.

4.2 Módulo de procesamiento

En este módulo se dará una explicación breve de donde se realiza el procesamiento señales en el automóvil.

4.2.1 ECU (*Engine Control Unit*)

Esta unidad de control procesa las señales que son enviadas de los diferentes sensores con los que cuenta un automóvil como lo es el sensor de velocidad, normalmente ubicado dentro de la transmisión, con este dato se puede calcular el momento preciso para hacer un cambio de velocidad en una caja de transmisión automática, con esta señal del sensor es que funciona el limitador de velocidad ya que va conectado por el puerto OBD-II y el cual genera comunicación con la ECU y así este puede gobernar o limitar el motor ya sea mediante las revoluciones o mediante la velocidad.

4.3 Módulo de control

En este módulo encontraremos una explicación breve de cómo es que se utilizó nuestro microcontrolador Arduino, los Servomotores implementados y los Motorreductores.

4.3.1 Microcontrolador Arduino

Esta sección del sistema es el cerebro, el encargado de procesar los datos que le manden los sensores RFID, sensor de huellas y el ultrasónico para poder ejecutar las órdenes necesarias con los demás componentes. Se comenzó por la elaboración de un código para controlar el módulo RFID en el cual se le introdujeron órdenes sencillas como lo son el encender un led y mover un servomotor mediante una condición.



El siguiente paso fue la elaboración de un código con el que se controlara el sensor de huellas dactilares donde inicialmente contamos con códigos de ejemplo en el que está la opción de agregar una huella, borrar, etc. Cada uno por separado, con lo cual se tomó un código de ejemplo, el cual ejecuta un cierto número de acciones como lo son el mover un servomotor y encender un led, pero se buscó la manera de que el lector no estuviera todo el tiempo encendido ni leyendo para evitar un sobrecalentamiento de este, activándolo solo por un periodo de tiempo para la lectura de la huella. Una vez terminado este código, se procedió a complementarlo haciendo que se pudiera agregar o borrar una huella dentro del mismo código en el que se escanea la huella y se realizaran las acciones, para esto fue necesario la creación de funciones para cada una de estas acciones y se iban probando mediante botones físicos.

Cuando se terminaron las pruebas con botones físicos y todo funcionaba de manera correcta se procedió a incluir la librería de Nextion para poder crear una comunicación de la pantalla con el microcontrolador, para esto se declararon únicamente los objetos que mandaran señales al controlador para que se inicialice la comunicación, esto se hace en el set up del programa y dentro del *void loop* se inicializa un comando que lee los datos mandados desde la pantalla para así procesarlos y ejecutar las órdenes necesarias para que el sistema funcione.

4.3.2 Servomotores

En esta sección de servomotores se trabajó con base a trabajos vistos en otros proyectos como lo son el control de dichos servomotores. Una vez teniendo esta base se procedió a generar un control mediante botones para después sustituirlos con el uso de la pantalla táctil que tiene como ventaja el evitar la necesidad de agregar componentes físicos como botones, lo que lo vuelve más práctico.

Una vez terminado el control de movimiento de estos servomotores se agregó la parte de acomodo automático, utilizando la memoria EEPROM de nuestro microcontrolador, ya que esta memoria no es volátil y tiene alrededor de 100 reescrituras volviéndola muy útil para nuestra necesidad. Una vez almacenado este valor lo que sucederá es que al encender el vehículo con nuestra huella dactilar asignará el valor guardado a la posición de los servomotores para un acomodo



automático, esto es para la parte de retrovisores de nuestro sistema, en el anexo 1 podemos observar un diagrama de conexión.

4.3.3 Motorreductores

En esta sección se tomó como base lo anteriormente visto en la parte de los servomotores, ya que se tomará básicamente el mismo procedimiento para esta sección haciendo un pequeño cambio, el cual es utilizar un contador para cada vez que el botón esté en alto para almacenar este valor en la memoria EEPROM en dos casillas diferentes debido a que son dos motorreductores los que se utilizaron para los asientos, podemos observar en el anexo 2 un diagrama de conexión.

El acomodo automático de estos será de una manera similar a los servomotores, pero en este caso se utilizará una serie de condiciones anidadas para lograr el acomodo de la posición correcta de manera automática en el momento en que se ingrese la huella dactilar.

4.4 Módulo de comunicaciones

A continuación, veremos las formas de comunicación que se implementaron en nuestro prototipo las cuales son nuestro puerto OBDII y el puerto serial de transmisión y recepción.

4.4.1 OBDII

En nuestro sistema necesitaremos la conexión por OBDII (On Board Diagnostics) diagnóstico abordo, el cual nos permite checar en tiempo real datos como lo son las revoluciones del vehículo, velocidad, emisión de gases, entre otras cosas. Lo necesitamos ya que el limitador de velocidad se conecta a través de este sistema.

En nuestro sistema el limitador estará apagado normalmente, y solo se activará cuando el usuario 2 o 3 enciendan el auto, ya que el microcontrolador energiza un relevador que activará este permitiendo así tener asegurado el buen manejo de nuestro automóvil por terceros.

En este apartado se describe el proceso de diseño, análisis, simulación, selección de componentes, construcción física y pruebas para realizar los módulos o subsistemas



de comunicación tanto de datos como de señales de control y de variables potencia del prototipo o sistema.

4.4.2 Comunicación Serie Tx y Rx

La comunicación serie es la manera en la que se pueden transmitir datos tanto al ordenador como con los diferentes tipos de sensores que trabajan con el microcontrolador Arduino. En nuestro caso, es el Mega 2560, y tenemos 3 puertos seriales que en este caso no está conectado al puerto USB.

4.5 Módulo de Interfaz hombre-máquina

En este módulo se dará una explicación de la evolución del funcionamiento de nuestra interfaz hombre-maquina.

4.5.1 Pantalla Táctil Nextion

En esta parte del sistema principalmente se comenzó con el análisis de los componentes o herramientas que nos ofrece el software de Nextion editor, el cual está diseñado para crear interfaces de una manera más rápida, en este software se encuentran herramientas como botones, cuadros de texto, barras progresivas, agregar páginas, etc.

Una vez teniendo la comprensión necesaria de las herramientas para utilizar el software, se precedió a generar una interfaz con un menú de opciones, con lo cual se eliminarían los botones, simplificando el circuito y evitando errores de conexión.

Se comenzó por crear y modificar imágenes para la interfaz, esto se puede hacer con programas de edición, en este caso fue con Paint de la paquetería de Microsoft. En estas imágenes que se necesitan para la interfaz también se crearon versiones con la simulación de un botón presionado para mejor visualización de la interfaz.

Una vez teniendo listas las imágenes que se utilizarían se introdujeron los componentes de botones que ya se sabía que utilizarían como lo son el de *scan*, menú, control de asientos, control de retrovisor, control de velocidad, agregar huella, borrar huella, huella 2, huella 3, así como un botón para regresar a la página principal en cada



página. Después, se fueron agregando cuadros de texto para que los mensajes que se enviaron a través del monitor serie se visualicen en la pantalla Nextion.

Una vez teniendo todos los componentes listos se realizaron simulaciones que puede ejecutar el programa de Nextion editor para verificar que cada componente cumpla con lo que debe realizar, para después cargarlo a la pantalla como se observa en el anexo 4.

Se introdujeron las librerías de Nextion en el código principal de Arduino para poder tener conexión con la pantalla, para esto se tienen que introducir los componentes que harán interacción entre la pantalla y el microcontrolador, así como también deben generarse objetos para los componentes que son botones para ejecutar las órdenes, después en el set up se inicializa la pantalla para que en el *void loop* se pueda introducir un *loop* de Nextion que está leyendo constantemente la información que mande la pantalla al controlador. Por último, se hicieron ajustes en el código principal que ya se tenía para que pueda funcionar en conjunto con la pantalla.

Una vez cargado en la pantalla se conectó otra vez de los puertos de comunicación de Arduino TX y RX para la comunicación entre ambos, y se comenzaron las pruebas de funcionamiento con estos dos en conjunto para corroborar que todo esté en orden y ejecute todo de forma correcta.

4.6 Integración de prototipo o sistema global.

El proceso de la integración del sistema completo fue mediante un análisis para los diferentes tipos alimentaciones que se requieren ya que el sensor RFID trabaja con un voltaje de 3.3v, mientras que el sensor de huellas, la pantalla táctil, el microcontrolador y los servomotores trabajan a 5v. Los motorreductores por otro lado trabajan con un rango de 5 a 9 volts y los relevadores que activan el encendido del auto y el limitador de velocidad trabajan a 12v, por lo que se necesitó de dos reguladores de corriente donde uno solamente nos entrega 5v a la salida, mientras el otro es un módulo regulador variable que utilizamos para alimentar el puente h que controla los motorreductores y así se logró una integración completamente funcional del sistema completo.



Esta hoja está en blanco intencionalmente



5. Validación del sistema

5.1 Validación del funcionamiento del módulo de censado.

En esta sección de validación se describirá de forma breve el proceso de pruebas, errores y mejoras que se fueron realizando a lo largo de este proyecto.

5.1.1 Sensor RFID

El proceso de este sensor contó con varias pruebas, las cuales consistían en intercambiar nuestros dispositivos RFID, los cuales son 1 tarjeta, un llavero, y 2 pulseras. Con estas pruebas obtuvimos una efectividad del 100% en un total de 100 pruebas.

Otro tipo de prueba que se le hizo al sensor fue de distancia, las cuales eran 0.5cm, 1.5cm, 3.0cm, 5.0cm, 8.0cm y 10.0cm y se expresan en la siguiente tabla.

Dispositivo	0.5 cm	1.5 cm	3.0 cm	5.0 cm	8.0 cm	10.0 cm
Llavero	o	o	o	o	X	X
Tarjeta	o	o	o	o	X	X
Pulsera1	o	o	X	X	X	X
Pulsera2	o	o	X	X	X	X

Tabla 2. Pruebas de alcance del sensor RFID

Después de obtener esta tabla de rangos de trabajo se procedió a realizar las simulaciones de trabajo donde notamos una lectura totalmente eficiente del sensor dado que no presento algún error de lectura o de reconocimiento, e hizo las comparaciones necesarias y correctas entre los dispositivos registrados y los que no lo estaban, en este caso solamente se registraron 2 pulseras RFID, y el llavero y la tarjeta fueron utilizados para corroborar que se le niega el acceso a dispositivos no registrados.

Una vez que se terminaron las pruebas se llegó que esta sección del módulo funciona de manera correcta y está lista para introducirse al código final.



5.1.2 Sensor de huellas dactilares

El funcionamiento de este sensor se fue analizando con el paso de los días y conforme se fue evolucionando, primeramente, se notó que el sensor una vez conectado a 5v y al puerto serie de nuestro microcontrolador y cargando el código de prueba incluido en nuestro software de Arduino con las librerías para nuestro sensor nos muestra que hay una comunicación correcta y que se ha identificado el sensor, por lo que se procedió a hacer pruebas para el registro de huellas.

Una vez que se entendió el funcionamiento del sensor procedimos a realizar pruebas con un código de comparación el cual como su nombre lo dice trata de comparar las huellas que están registradas, donde notamos que se hacía una comparación eficiente pero el código no realizaba el trabajo como debía ser por lo que se tuvo que hacer modificación de sintaxis para que este pudiera funcionar bien.

Una vez que el sensor y el controlador realizaban la comparación y se ejecutaban las ordenes de manera correcta se procedió a realizar simulaciones mediante botones con lo que se hacia el control de escanear huellas, así como agregar y borrar con la opción de agregar 2 huellas a parte de la principal y de igual forma estas dos son las únicas que se pueden borrar.

Se dejo esta parte del módulo mientras se desarrollaba la interfaz hombre maquina la cual se explicará más adelante, una vez terminada esta, el siguiente paso fue realizar pruebas en conjunto de el sensor de huellas con la pantalla táctil donde se encuentra nuestra interfaz, en las cuales se notó que tanto el sensor, el microcontrolador y la pantalla realizaron sus órdenes de manera correcta por lo que esta sección se tenía lista para la unión en un código final.

5.2 Validación del funcionamiento del módulo de procesamiento

En la sección de validación de módulo de funcionamiento no se realizaron pruebas, los motivos se explicarán en punto 5.2.1.

5.2.1 ECU

En esta sección no se realizaron pruebas debido a que se trata de la computadora del auto y se considera que esta parte del vehículo funcione al 100%.



5.3 Validación del funcionamiento del módulo de control

En la validación del módulo de control se describirá de forma breve el proceso que se llevó a cabo para lograr un buen funcionamiento con el microcontrolador Arduino, los servomotores implementados y los motorreductores.

5.3.1 Microcontrolador Arduino

Nuestro microcontrolador abarca desde el control de los componentes como servomotores y motorreductores hasta la comunicación con el sensor de huellas dactilares, RFID y nuestra interfaz.

El primer paso en nuestro módulo de control fue el análisis de nuestro sensor de huellas con sus distintas funciones como lo son el agregar, borrar o leer huellas las cuales inicialmente se encuentran separadas y se desarrolló un código capaz de implementar estas 3 funciones en un solo programa, una vez teniendo esta unión se realizaron las distintas pruebas para corroborar que su funcionamiento fuese el indicado.

En la parte de nuestro sensor RFID el proceso de pruebas fue breve ya que se registraron inicialmente los rangos e trabajo de nuestros componentes utilizando un código de pruebas incluido en nuestro software de Arduino el cual es *Read* (lectura) con el cual se alimenta el chip que se encuentra en nuestro dispositivo y nos manda por radio frecuencia el numero registrado en este, una vez obtenidos los números de nuestras pulseras RFID se registraron en el código para que este pudiera tener una base de comparación y así poder permitir o denegar el paso a los usuarios.

La comunicación que se tuvo con nuestra interfaz la cual es una pantalla táctil se realizó haciendo las conexiones con nuestro puerto serial y alimentando nuestra pantalla a 5v, también se realizaron de nueva cuenta las conexiones del sensor de huellas dactilares con sus componentes se prueba los cuales inicialmente eran solo leds que posteriormente se cambiaron por la conexión nuevamente a un led pero agregando un optoacoplador y un relevador para la simulación del encendido del auto una vez teniendo estas conexiones hechas se observó que el funcionamiento de estas dos secciones del prototipo funcionaron de manera correcta.

El código final del proyecto se puede ver en el anexo 5.



5.3.2 Servomotores

El funcionamiento de estos componentes inicio principalmente con botones que simulaban la parte de la interfaz teniendo 2 botones por cada servomotor los cuales nos permiten el controlar el movimiento de 0 a 180° y de 180° a 0°.

Una vez que se obtuvo el código que nos permitió este control de movimiento se procedió a agregar 4 botones más y nuevamente el sensor de huellas con lo que se realizaron pruebas simulando con estos botones el control de nuestra interfaz que nos permite guardar posición, seleccionar en que usuario guardar (de los 3 disponibles) y con el sensor de huellas realizar la prueba de seguridad la cual consiste en que si la huella no es la indicada o no está registrada el valor o la posición no se guardara pero en el caso de que se coloque una huella que si está registrada podemos guardar el valor de la posición en espacio de la memoria EEPROM de nuestro microcontrolador, notando así que esta sección del sistema realiza su función de manera correcta simulando así de buena forma el acomodo automático de retrovisores.

Cuando se desarrolló este código por completo y este cumplió con su funcionamiento se procedió a agregarse a el código en conjunto.

El código final del proyecto se puede ver en el anexo 5.

5.3.3 Motorreductores

Los motorreductores tuvieron un proceso similar el de los servomotores ya que se utilizó la misma configuración de conexión para los botones y con el sensor de huellas dactilares, lo que cambio en esta sección fue el uso de contadores que registran cada señal mandada por los botones para tener así un control de su posición mediante estos, en este caso el dato que se registra en otras secciones de nuestra memoria EEPROM son los contadores, para que una vez que se simulo un encendido del auto con las diferentes huellas se realizó un ajuste correcto de forma automática durante las diferentes pruebas las cuales fueron:

- Pruebas de encendido con la huella 1: En esta prueba notamos que al a ver echo un cambio de posición comparado con la posición guardada en la huella 1 se hace una comparación se coloca de forma en que coinciden ambas posiciones.



- Prueba de encendido con huella 1 con la misma posición: En esta prueba se simula el encendido del auto con la huella 1 cuando el contador se encuentra en el mismo valor que el registrado en la huella 1 con lo que notamos que no hubo movimiento en los motorreductores demostrando así un funcionamiento correcto
- Prueba de encendido con la huella 2 con posición diferente: Esta prueba es muy similar a la primera en esta sección ya que se simuló el encendido del auto con la huella dos teniendo una configuración de posición diferente a la que se encuentra guardada en la memoria del usuario 2 donde se observó una comparación entre estas de forma correcta y un acomodo adecuado.
- Prueba de encendido con la huella 2 con la misma posición: Se realizó exactamente los mismos pasos que con la huella 1 notando una comparación correcta y no se realizó movimiento por parte de los motorreductores lo que nos indica que el ajuste automático funciona de forma correcta.

Estas pruebas se realizaron de igual forma con la huella 3 llevando a cabo el mismo procedimiento y obteniendo los resultados esperados.

5.4 Validación del funcionamiento del módulo de comunicaciones

En el apartado del módulo de validación de comunicaciones se tiene de igual forma que en los módulos anteriores una explicación breve de su proceso, fallas y correcciones.

5.4.1 Comunicación Serial (RX y TX)

En esta sección, durante todas y cada una de las pruebas con las diferentes secciones del sistema no se registró ninguna falla, lo que nos dice que nuestro microcontrolador logró una comunicación correcta con cada uno de nuestros sensores, lo que nos da una efectividad del 100% en la comunicación.

En este apartado se puede describir el proceso de pruebas y validación realizadas en el módulo de comunicación, así como los rediseños, ajustes, modificaciones, hechas para garantizar el mejor desempeño del subsistema o módulo.



5.5 Validación del funcionamiento del módulo de Interfaz hombre-máquina

En este módulo se describirán algunas pruebas realizadas para el desarrollo de nuestra interfaz.

5.5.1 Pantalla Nextion

Para nuestra interfaz se utilizó una pantalla táctil como se vio anteriormente, donde las primeras pruebas se realizaron dentro del simulador que está incluido en nuestro software de Nextion editor con el cual podemos observar que las ordenes indicadas para nuestra pantalla se cumplen de forma correcta refiriéndonos en primer lugar a los objetos que están programados para hacer un cambio de página o realizar una representación de carga o avance mediante barras de progreso etc. Y en segundo lugar a los datos que se envían por medio del puerto serie desde nuestra pantalla táctil hasta nuestro microcontrolador, donde se observó que todo trabaja como debe de ser.

Una vez terminadas las simulaciones para corroborar el funcionamiento de cada uno de los objetos que se utilizaron en nuestra interfaz se procedió a realizar pruebas montando principalmente el sensor de huellas dactilares para corroborar que los datos mandados y recibidos sean los correctos y nuestro microcontrolador realice los trabajos que se tengan registrados en cada línea de código, con esta pruebas se observó que se cumplía con el funcionamiento del subsistema o modulo por lo que el siguiente paso fue realizar la unión final de nuestro código y nuestro circuito.



5.6 Validación Integral del sistema o prototipo.

Durante las pruebas de este sistema ya en completo funcionamiento se encontraron con fallas o errores principalmente de programación, detalles que afectaban el funcionamiento, como lo es que las señales que manda la pantalla táctil no estaban siendo correctamente interpretadas por un error en el comando de lectura de dichas señales. Una vez resuelto este problema se encontró otro “problema”, el cual era que al simular el encendido del vehículo con la segunda y tercera huella, los motorreductores giraban en un intento de acomodo automático errado debido a que no se habían guardado valores en la memoria que le corresponde a estos dos usuarios, pero una vez que se le registró un valor se logró un funcionamiento correcto de manera completa



Esta hoja está en blanco intencionalmente



6. Conclusiones

En la sección del circuito se pudo observar durante cada una de las pruebas que no hubo fallas como podrían ser el que algunos componentes se calienten, o que los sensores no generen las señales adecuadas por lo que se considera que todos y cada uno de los componentes utilizados en este proyecto fueron seleccionados correctamente debido a que cumplieron con su función como se tenía planeado.

Podemos observar que la metodología aplicada a nuestro proyecto fue la correcta y de mucha ayuda ya que al seguir los pasos planeados de esta favoreció a que se llegara de manera correcta al objetivo planteado, gracias a la secuencia planteada como lo fue el probar subsección por subsección ayuda a poder comprender mejor el funcionamiento de cada una de estas para así poder avanzar de una manera progresiva pero totalmente funcional. La parte lógica de este proyecto se encontraron diversos conflictos que requirieron constantes reprogramaciones a nuestro microcontrolador, pero gracias a que cada sección se provo de manera individual junto con sus debidos códigos y se llevaron a cabo las evoluciones de estas secciones paso a paso se lograron percibir fallas o errores de código que a simple vista podrían no afectar el funcionamiento del proyecto, pero una vez corregidas le dan una mejor presentación como uso.

Con este proyecto llegamos a la conclusión de que con instrumentos que individualmente podrían considerarse fáciles de utilizar pueden llegar a formar parte de un sistema totalmente rentable y funcional.

Se pensó en la comodidad y el confort que se puede llegar a tener al obtener el producto final, que el conductor no tenga la necesidad de acomodar retrovisores y asientos cada que se cambie de usuario, tener una base de datos con los diferentes conductores de cada automóvil y contar con un poco de seguridad en cuanto a límites de velocidad, tener en cuenta la integridad y la vida de los conductores tanto del conductor como de pasajeros y de los demás automovilistas que se pueden encontrar en el camino.



Esta hoja está en blanco intencionalmente



7. Referencias

- [1] **Erjavec, Jack and Arias, Jeff, Hybrid,,** Electric and Fuel-Cell Vehicles» Delmar Cengage Learning, p. 336, USA, 2006.
- [2] **Frank, R.,** «Sensing in the Ultimately Safe Vehicle» Society of Automotive Engineers (SAE), USA, 2004.
- [3] **Pallas, Ramón,** «Sensores y Acondicionadores de Señal» Ed. Alfaomega-Marcombo), 4a Edición, p. 480, USA, 2004.
- [4] **Prosser, S.J.** «Automotive Sensors: past, present and future» en Journal of Physics ,P. 6, USA, 2007.
- [5] **Barrett Steven, Pack Daniel,** « Atmel AVR Microcontroller Primer » Alfaomega, USA, 2008.
- [6] **Galeano Gustavo,** « Programación de sistemas Embebidos en C» Ed. Alfaomega, p. 521, México, 2009.
- [7] **m,** «¿Qué es Ford MyKey y por qué es tan importante para tus hijos?» 2020. [En línea]. Available: <https://hipertextual.com/presentado-por/ford/ford-mykey>
- [8] **Chevrolet** «Cómo usar los asientos con memoria.» 2019. [En línea]. Available [https://es.chevrolet.com/how-to/memory-seats:](https://es.chevrolet.com/how-to/memory-seats)
- [9] **F.J. Ruvalcaba Moya,** "Sistema de acceso y encendido remoto del automóvil mediante el uso de smartphone", Tesis de maestría, CIATEQ, Zapopan, Jalisco, México, 2018. [En línea]. Available: <https://ciateq.repositorioinstitucional.mx/jspui/bitstream/1020/280/1/RuvalcabaMoyaFranciscoJ%20MSIM%202018.pdf>
- [10] **R.A. Báez Llancafil; J.L. Chacano Nuñez,** "S" "Tecnología RFID para el reconocimiento y asignación de pallets y rollos de cartulina en bodega", Tesis de licenciatura, Facultad de 30 Ciencias de la Ingeniería, Universidad Austral de Chile, Valdivia, Chile, 2013. [En línea]. Available: <http://cybertesis.uach.cl/tesis/uach/2013/bmfcib142t/doc/bmfcib142t.pdf>



- [11] **R**, "¿Cómo funciona una pantalla táctil?",2015. [En línea]. Available: <https://www.unocero.com/noticias/como-funciona-una-pantalla-tactil/>
- [12] "Qué es el limitador de velocidad y cómo funciona.",2020. [En línea]. Available: <https://www.motor.mapfre.es/accesorios/noticias-accesorios/limitador-velocidad/>
- [13] **Arduino**, "¿Que es Arduino?",2010. [En línea]. Available: <https://arduino.cl/que-es-arduino/#:~:text=Arduino%20es%20una%20plataforma%20de,una%20serie%20de%20pin e s%20hembra.&text=Por%20lo%20tanto%2C%20la%20placa,determinado%20dise%C3%B1 o%20de%20circuitos%20ADa%20interna.>
- [14] **Hubor**, "Hubor - Proteus",2015. [En línea]. Available: <https://www.huborproteus.com/proteus-pcb/prospice.html>
- [15] **Hubor**, "Hubor - Proteus",2015. [En línea]. Available: <https://www.huborproteus.com/proteus-pcb/prospice.html>
- [16] **ITEAD STUDIO All Right Reserved**, "What's Nextion",2011. [En línea]. Available: <https://nextion.tech/>

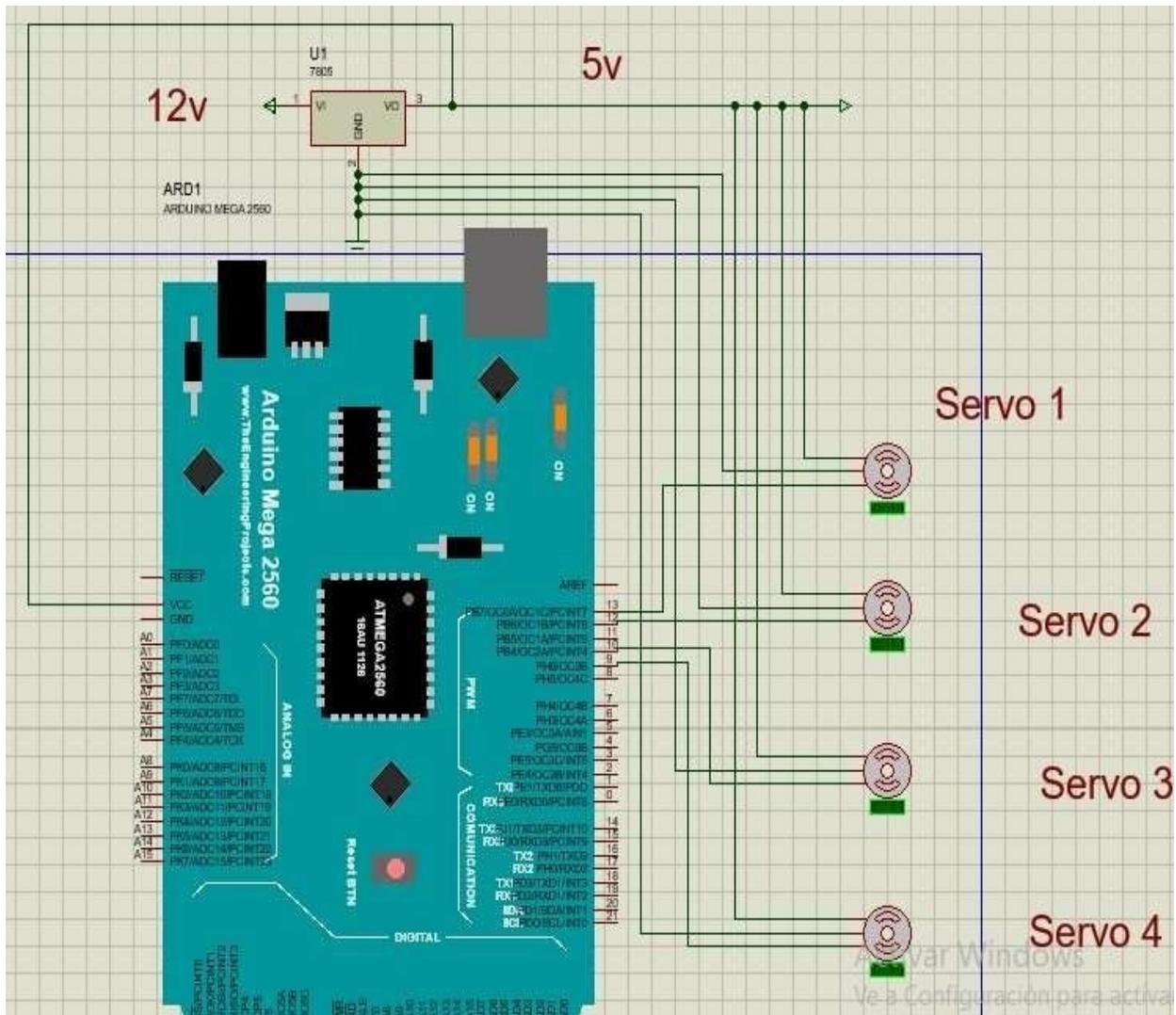


Esta hoja está en blanco intencionalmente

Anexo A

Anexo 1

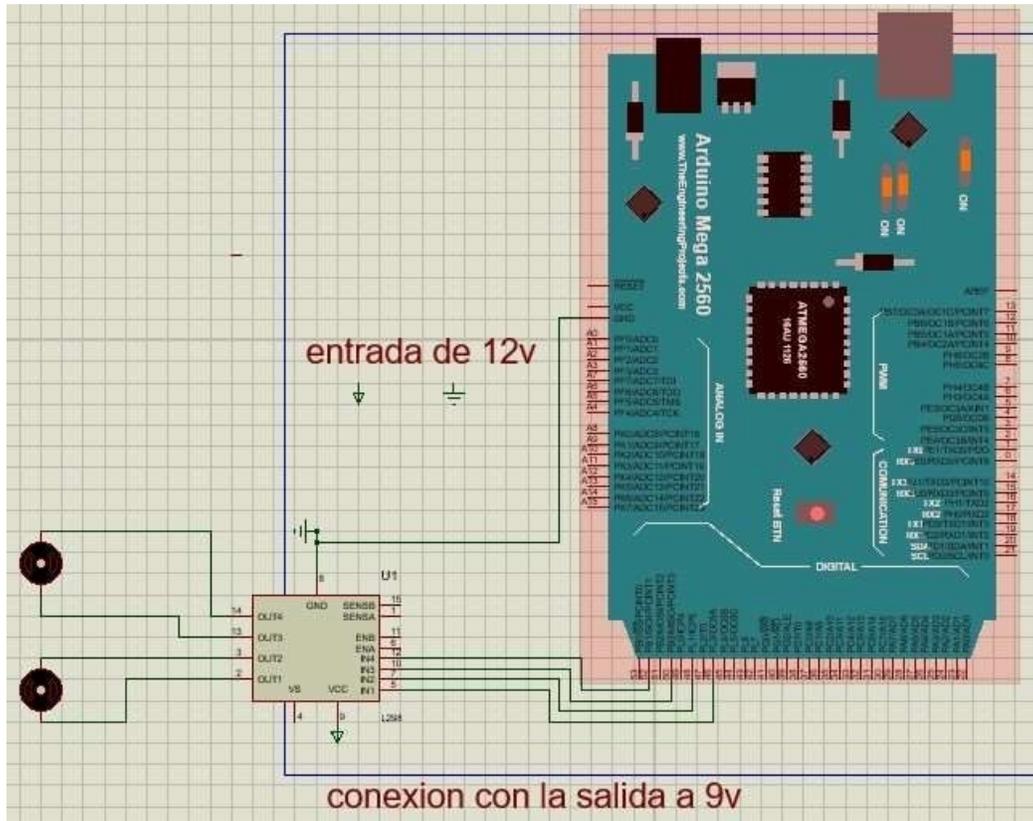
Diagrama de conexión de servomotores.





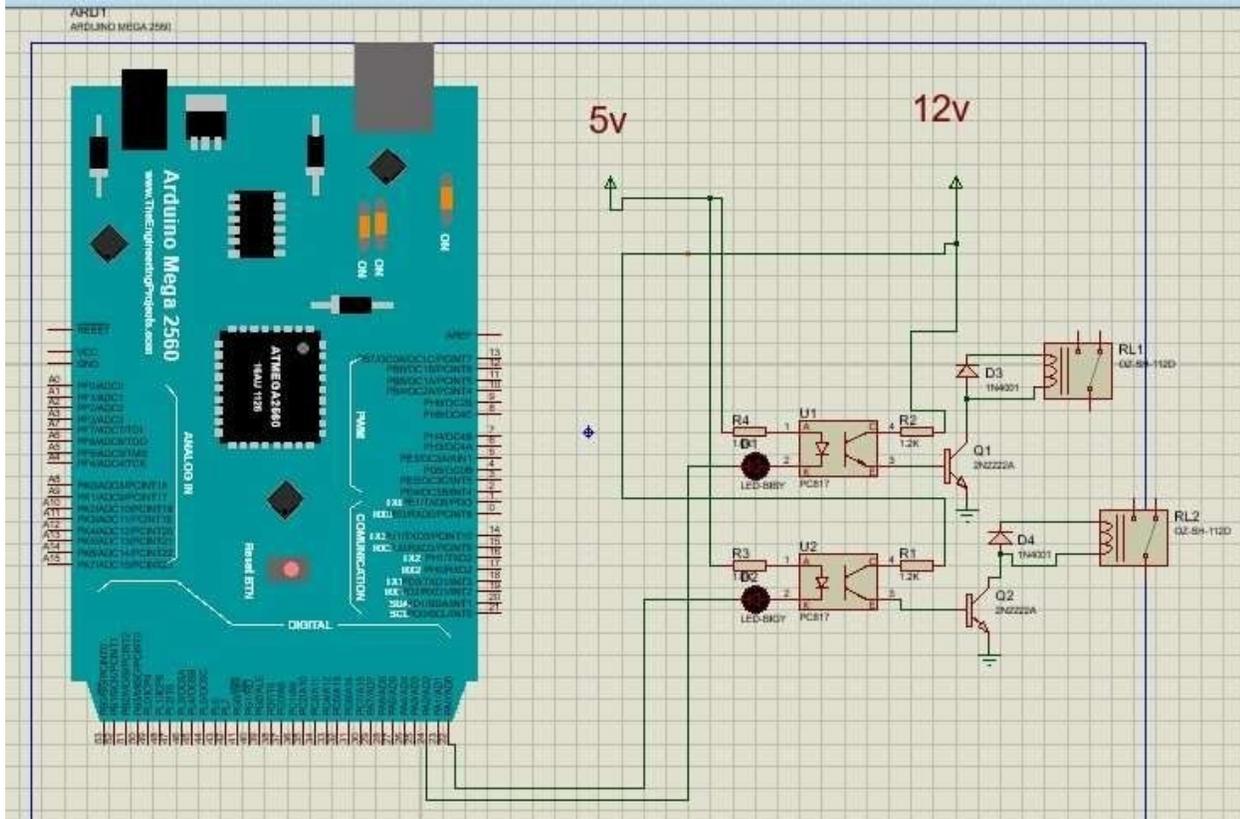
Anexo 2

Diagrama de conexión de motorreductores



Anexo 3

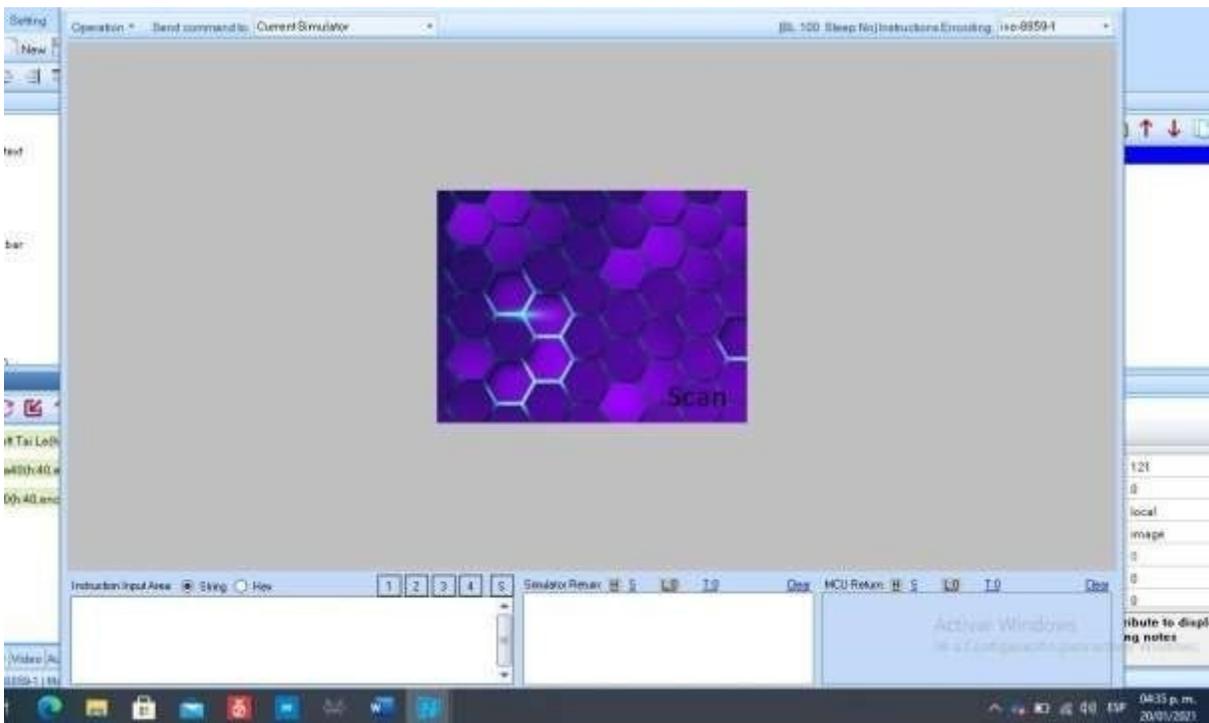
Diagrama de conexión del sistema de encendido por huella digital





Anexo 4

Representación de simulación en Nextion editor



Anexo 5

Código final

```
#include"Nextion.h"  
  
#if (defined(__AVR__) || defined(ESP8266)) && !defined(__AVR_ATmega2560__)  
#endif  
  
#include<Servo.h>  
  
#include <EEPROM.h>  
  
#include <SPI.h> // Libreria SPI  
  
#include <MFRC522.h> // Libreria MFRC522  
  
#include <Adafruit_Fingerprint.h> // Libreria para el Sensor de huella dactilar  
  
SoftwareSerial mySerial(2, 3); // Crear Serial para Sensor Rx, TX del Arduino
```



```
#define mySerial Serial1

#define RST_PIN 5 // Pin de reset

#define SS_PIN 53 // Pin de slave select

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial); // Crea el objeto Finger
comunicacion pin 18 , 19

MFRC522 mfrc522(SS_PIN, RST_PIN); // Objeto mfrc522 enviando pines de slave select
y reset

byte LecturaUID[4]; // Array para almacenar el UID leído byte
Usuario1[4]= {0x7E, 0x2B, 0x5D, 0x3C}; // NUMERO DEL USUARIO 1
byte Usuario2[4]= {0x46, 0x28, 0x5D, 0x3C}; // NUMERO DEL USUARIO 2

//-----Declaracion de variables-----

int j; int
s;

const int led =11; // led conectado al pin 11

int del=27; int
Del,be1,be2,bd1,bd2;

uint8_t id;

int f;

Servo S1;// servo de retrovisor izquierdo arriba y abajo
Servo S2;// servo de retrovisor izquierdo derecha izquierda
Servo S3;// servo de retrovisor derecho arriba y abajo Servo S4;// servo
de retrovisor derecho derecha izquierda int motor1=53;//motor
```



```
derecho arriba abajo int motor2=51;// motor derecho adelante
atras int motor3=49;// motor izquierdo arriba abajo int
motor4=47;//motor izquierdo adelante atras int i;// valor para asientos
automaticos int c=0;//valor para acomodo automatico int
cont1=0,cont2=0,cont1m,cont2m,posicion1,posicion2; int
mr1,mr2,mr21,mr22,mr31,mr32;// memorias de posicion perfiles 1, 2, 3 int
pos1=0,pos2=0,pos3=0,pos4=0;// posiciones servos int sa;// variable
para memoria e servos int sm;// variable para memoria se
motorreductores int b1=23; int b2=25; int b3=27; int b4=29; int bm=35; int
id1=43; int id2=45; int Bo1; int Bo2; int Bo3;

int Bo4;

int Bm,m1,m2,m3,m4,m21,m22,m23,m24,m31,m32,m33,m34;

int i1,i2;

bool scanning = false;

int counter = 0; int

id_ad_counter = 0; bool

id_ad = false; uint8_t

num = 1; bool

id_selected = false; bool

first_read = false; bool

main_user = false; bool

add_new_id = false; bool

door_locked = true; int
```



```
main_user_ID = 1;
```

```
int NH;
```

```
//-----Se declaran los objetos para la pantalla-----  
-----
```

```
NexButton Asiento_Ar = NexButton(3, 1, "Asiento_Ar");  
NexButton Asiento_Ab = NexButton(3, 2, "Asiento_Ab");  
NexButton Asiento_Ad = NexButton(3, 3, "Asiento_Ad");  
NexButton Asiento_At = NexButton(3, 4, "Asiento_At");  
NexButton RAr_Isq = NexButton(5, 1, "RAr_Isq");  
NexButton RAb_Isq = NexButton(5, 2, "RAb_Isq");  
NexButton RIzq_Isq = NexButton(5, 3, "RIzq_Isq");  
NexButton RDer_Isq = NexButton(5, 4, "RDer_Isq");  
NexButton RAr_Der = NexButton(6, 1, "RAr_Der");  
NexButton RAb_Der = NexButton(6, 2, "RAb_Der");  
NexButton RIzq_Der = NexButton(6, 3, "RIzq_Der");  
NexButton RDer_Der = NexButton(6, 4, "RDer_Der");  
NexButton Scan_Vel = NexButton(7, 2, "Scan_Vel");  
NexButton Menos_Vel = NexButton(8, 1, "Menos_Vel");  
NexButton Mas_Vel = NexButton(8, 2, "Mas_Vel");  
NexButton Scan_H = NexButton(9, 1, "Scan_H");  
NexButton Huella_2 = NexButton(10, 1, "Huella_2");  
NexButton Huella_3 = NexButton(10, 3, "Huella_3");  
NexButton C_Asientos = NexButton(2, 1, "C_Asientos");  
NexButton C_Retrovisores = NexButton(2, 2, "C_Retrovisores");  
NexButton C_Velocidad = NexButton(2, 3, "C_Velocidad");
```



```
NexButton C_Agrgar_H = NexButton(2, 4, "C_Agrgar_H");  
  
NexButton C_Borrar_h = NexButton(2, 5, "C_Borrar_h");  
  
NexButton BotScan = NexButton(0,3,"BotScan");  
  
NexButton BHuella_2 = NexButton(13, 1, "Huella_2");  
NexButton BHuella_3 = NexButton(13, 2, "Huella_3");  
NexButton Guardar = NexButton(4, 4, "Guardar");  
NexButton guardarm = NexButton(3,6, "guardarm");  
NexButton Basientos = NexButton(3,5, "Basientos");  
  
// Medicion  
NexGauge GVel = NexGauge(8,4,"GVel");  
  
// TXT  
NexText Bienvenido = NexText(0, 4, "Bienvenido");  
NexText Vel_Lim = NexText(0, 5, "Vel_Lim");  
NexText ScanTxt = NexText(9, 2, "ScanTxt");  
NexText Agregado2 = NexText(11, 2, "Agregado2");  
NexText Agregado3 = NexText(12, 2, "Agregado3");  
NexText ScanB = NexText(15, 2, "ScanB");  
NexText t2 = NexText(10, 6, "t2");  
NexText t0 = NexText(13, 3, "t0");  
NexText DelTex = NexText(14, 2, "ScanB");  
NexText Tguardado = NexText(4, 0, "Tguardado");  
NexText texasientos = NexText (3,7,"texasientos");  
NexText texto = NexText (15,1,"texto");  
  
//PAge
```



```
NexPage page10 = NexPage (10,0,"page10");
```

```
NexPage page13 = NexPage (13,0,"page13");
```

```
NexPage page7 = NexPage (7,0,"page7");
```

```
NexPage page1 = NexPage(1,0,"page1");
```

```
NexPage page2 = NexPage(2,0,"page2");
```

```
//-----Registro de objetos a leer-----
```

```
NexTouch *nex_listen_list[] =
```

```
{
```

```
&C_Asientos,
```

```
&C_Retrovisores,
```

```
&C_Velocidad,
```

```
&C_Agrgar_H,
```

```
&C_Borrar_h,
```

```
&Asiento_Ar,
```

```
&Asiento_Ab,
```

```
&Asiento_Ad,
```

```
&Asiento_At,
```

```
&RAr_Isq,
```

```
&RAb_Izq,
```

```
&RIzq_Izq,
```

```
&RDer_Izq,
```

```
&RAr_Der,
```

```
&RAb_Der,
```

```
&RIzq_Der,
```



&RDer_Der,

&Scan_Vel,

&Menos_Vel,

&Mas_Vel,

&Scan_H,

&Huella_2,

&Huella_3,

&BotScan, &BHuell_2,

&BHuell_3,

&t2,

&t0,

&DelTex,

&Guardar,

&guardarm,

&Basientos,

NULL

};

void Encender_Auto();

void Mal_Registro();

void Limitador(); void

uno(); void doss();

void tres();

// funciones de el sensor de huellas

uint8_t getFingerprintEnroll(); void



```
getFingerprintID(); uint8_t
```

```
deleteFingerprint(uint8_t id);
```

```
//funciones para la pantalla
```

```
void Scan(void*ptr); void
```

```
Agregar_(void*ptr); void
```

```
Borrar (void *ptr); void
```

```
Dos(void *ptr); void
```

```
Tres(void *ptr); void BDos
```

```
(void *ptr); void
```

```
BTres(void *ptr); void
```

```
AsientoAr (void *ptr); void
```

```
AsientoAb (void *ptr); void
```

```
AsientoAd (void *ptr); void
```

```
AsientoAt (void *ptr); void
```

```
RArIsq (void *ptr); void
```

```
RAblzq (void *ptr); void
```

```
Rlzqlzq (void *ptr); void
```

```
RDerlzq (void *ptr); void
```

```
RArDer (void *ptr); void
```

```
RAbDer (void *ptr); void
```

```
RlzqDer (void *ptr); void
```

```
RDerDer (void *ptr); void
```



```
save (void *ptr);  
void mguardar (void *ptr);  
void save2 (void *ptr);  
  
//-----inicia setup----- void setup()  
{  
  //Se declaran los pines para los servos  
  S1.attach(9);  
  S2.attach(10);  
  S3.attach(12);  
  S4.attach(13);  
  
  pinMode(24, OUTPUT);  
  pinMode(22, OUTPUT); pinMode  
(26, OUTPUT); pinMode(28,  
OUTPUT); pinMode(30, OUTPUT);  
  pinMode (32, OUTPUT);  
  pinMode (34, OUTPUT);  
  pinMode (36, OUTPUT);  
  pinMode (38, OUTPUT);  
  pinMode(b1,INPUT);  
  pinMode(b2,INPUT);  
  pinMode(b3,INPUT);  
  pinMode(b4,INPUT);
```



```
pinMode(31,OUTPUT);

pinMode(33,OUTPUT);

pinMode(37,OUTPUT);

pinMode(39,OUTPUT);

pinMode(41,OUTPUT);

pinMode(motor1,OUTPUT);

pinMode(motor2,OUTPUT);

pinMode(motor3,OUTPUT);

pinMode(motor4,OUTPUT);

SPI.begin();    // inicializa bus SPI

mfr522.PCD_Init();    // inicializa modulo lector RFID

Serial.println("Sistema Activado");    // Muestra texto Listo

//-----Se inicializa la pantalla Nextion-----

- nexInit();

//-----Se declaran los objetos que seran llamados
cuando sean detectados-----

BotScan.attachPop(Scan);

C_Agregar_H.attachPop(Agregar_);

C_Borrar_h.attachPop(Borrar);

Huella_2.attachPop(Dos); Huella_3.attachPop(Tres);

BHuell_2.attachPop(BDos);

BHuell_3.attachPop(BTres);

Asiento_Ar.attachPop(AsientoAr);
```



```
Asiento_Ab.attachPop(AsientoAb);  
Asiento_Ad.attachPop(AsientoAd);  
Asiento_At.attachPop(AsientoAt);  
RAr_Isq.attachPop(RArlsq);  
RAb_Isq.attachPop(RAbIsq);  
RIzq_Isq.attachPop(RIzqIsq);  
RDer_Isq.attachPop(RDerIsq);  
RAr_Der.attachPop(RArDer);  
RAb_Der.attachPop(RAbDer);  
RIzq_Der.attachPop(RIzqDer);  
RDer_Der.attachPop(RDerDer); Guardar.attachPop(save);  
guardarm.attachPop(mguardar);  
Basientos.attachPop(save2);  
  
Serial.begin(9600);  
  
while (!Serial); // Yun/Leo/Micro/Zero/...  
delay(100);  
Serial.println("Sistema de Perfiles y confort");  
  
// set the data rate for the sensor serial port  finger.begin(57600); //  
inicializa comunicacion con sensor a 57600 Baudios  delay(5); if  
(finger.verifyPassword()) {
```



```
Serial.println("Detectado un sensor de
huella!");  for (j=0;j<5;j++)
{
  Serial.println(".");
}
} else {
  Serial.println("No hay comunicacion con el sensor de huella");
Serial.println("Revise las conexiones");  for (j=0;j<5;j++)
{
  Serial.println(".");
}
while (1) { delay(1); }

}

finger.getTemplateCount();

Serial.print("El sensor contiene "); Serial.print(finger.templateCount); Serial.println("
plantillas");

Serial.println("Esperando por una huella valida...");
for (j=0;j<5;j++)
{
  Serial.println(".");
}
}
```



```
//-----Inicia void loop-----  
  
----- void loop()  
  
{  
  cont1=EEPROM.read(14);  
  cont2=EEPROM.read(15); nexLoop(nex_listen_list);  
  
  //-----seccion del sensor RFID-----  
  -  
  
  if (! mfrc522.PICC_IsNewCardPresent()) // si no hay una tarjeta presente  
  return;      // retorna al loop esperando por una tarjeta  
  
  if (! mfrc522.PICC_ReadCardSerial()) // si no puede obtener datos de la tarjeta  
  return;      // retorna al loop esperando por otra tarjeta  
  
  Serial.print("UID:"); // muestra texto UID:  
  for (byte i = 0; i < 4; i++) { // bucle recorre de a un byte por vez el UID  
  if (mfrc522.uid.uidByte[i] < 0x10){ // si el byte leído es menor a 0x10  
    Serial.print(" 0"); // imprime espacio en blanco y numero cero  
  } else{  
  // sino  
    Serial.print(" "); // imprime un espacio en blanco  
  }  
  Serial.print(mfrc522.uid.uidByte[i], HEX); // imprime el byte del UID leído en  
  hexadecimal  
  LecturaUID[i]=mfrc522.uid.uidByte[i]; // almacena en array el byte del UID  
  leído
```



```
}
```

```
Serial.print("\t"); // imprime un espacio de tabulacion
```

```
if(comparaUID(LecturaUID, Usuario1)){ // llama a funcion comparaUID con Usuario1
```

```
Serial.println("Bienvenido usuario 1"); // si retorna verdadero muestra texto bienvenida
```

```
abrirPuerta(); // funcion para abrir la puerta
```

```
}
```

```
else if(comparaUID(LecturaUID, Usuario2)){ // llama a funcion comparaUID con Usuario2
```

```
Serial.println("Bienvenido Usuario 2"); // si retorna verdadero muestra texto bienvenida
```

```
abrirPuerta(); // funcion para abrir la puerta
```

```
} else { // si
```

```
retorna falso
```

```
Serial.println("No Registrado"); // muestra texto equivalente a acceso denegado
```

```
Mal_Registro(); // funcion sonido de targeta no identificada
```

```
}
```

```
mfr522.PICC_HaltA(); // detiene comunicacion con tarjeta //-----
```

```
-----seccion del sensor de huellas-----
```

```
NH=finger.templateCount; // lectura de huellas para entrar al if de registro principal
```

```
if (NH==0) // En caso de no haber huella principal entrara al if para registrar la huella principal
```



{

```
Serial.println("No Existen Huellas Registradas por favor introduce cual sera la huella principal"); id=1; for (j=0;j<5;j++)
```

```
{
```

```
Serial.println(".");
```

```
}
```

```
while (! getFingerprintEnroll() );
```

```
}
```

```
}
```

```
//-----funcion abrir puerta-----
```

```
void abrirPuerta() { digitalWrite(28, HIGH); // Abrir la cerradura delay(1000);
```

```
digitalWrite(28, LOW); // cierra la cerradura
```

```
}
```

```
//-----funcion de comparacion de id RFID-----
```

```
boolean comparaUID(byte lectura[],byte usuario[]) // funcion comparaUID
```

```
{
```

```
for (byte i=0; i < 4; i++){ // bucle recorre de a un byte por vez el UID
```

```
if(lectura[i] != usuario[i]) // si byte de UID leido es distinto a usuario
```

```
return(false); // retorna falso
```

```
}
```

```
return(true); // si los 4 bytes coinciden retorna verdadero
```

```
}
```

```
//-----scan-----
```

```
void Scan(void *ptr)
```



```
{ s=1;
Serial.println("Scan.....");
while(counter<15)
{
  getFingerprintID();
delay(100); counter++;
}
counter=0;
if(f==1) {
  Serial.println("No Existe");
  Bienvenido.setText("No existe");
}
s=0;
}
//-----usuario 1-----
void uno()
{
if(f==0) {
  Serial.print("Finger.fingerID=");
  Serial.println(finger.fingerID);

  Serial.println("BIENVENIDO Alejandro Varona");
  Bienvenido.setText("BIENVENIDO Alejandro Varona");
```



```
Serial.println("Usted cuanta con velocidad normal");
```

```
Vel_Lim.setText("Usted cuanta con velocidad Normal");
```

```
for (j=0;j<5;j++)
```

```
{
```

```
Serial.println(".");
```

```
}
```

```
Encender_Auto();
```

```
S1.write(EEPROM.read(0));
```

```
S2.write(EEPROM.read(1));
```

```
S3.write(EEPROM.read(2));
```

```
S4.write(EEPROM.read(3));
```

```
Serial.print("S1=");
```

```
Serial.println(EEPROM.read(0));
```

```
Serial.print("S2=");
```

```
Serial.println(EEPROM.read(1));
```

```
Serial.print("S3=");
```

```
Serial.println(EEPROM.read(2));
```

```
Serial.print("S4=");
```

```
Serial.println(EEPROM.read(3));
```

```
c=0;
```

```
Serial.println(".");
```

```
Serial.print("mr1:");
```

```
Serial.println(EEPROM.read(12));
```

```
Serial.print("mr2:");
```



```
Serial.println(EEPROM.read(13));  
  
Serial.print("posicion1:");  
  
Serial.println(EEPROM.read(14));  
  
Serial.print("Posicion2:");  
  
Serial.println(EEPROM.read(15));  
  
mr1=EEPROM.read(12); mr2=EEPROM.read(13);  
  
posicion1=EEPROM.read(14);  
posicion2=EEPROM.read(15);  
  
  
if (posicion1<mr1)  
{  
  for(i=0;i<=200;i++)  
  {  
    digitalWrite(motor1,HIGH);  
digitalWrite(motor2,LOW);  posicion1=posicion1+1;  
    Serial.println(".");  
    Serial.print("Posicion1=");  
Serial.print(posicion1);  if  
(posicion1==mr1)  
  {  
    digitalWrite(motor1,LOW);  
digitalWrite(motor2,LOW);  
cont1=posicion1; c=1;  
i=201;
```



```
Serial.println("Posicion lista");
```

```
    }  
  }  
}  
if (posicion1>mr1)  
{  
  if (c==0)  
  {  
    for (i=0;i<=200;i++)  
    {  
  
      digitalWrite(motor1,LOW);  
digitalWrite(motor2,HIGH);   posicion1=posicion1-1;  
      Serial.println(".");  
      Serial.print("Posicion1=");  
Serial.print(posicion1);   if  
(mr1==posicion1)  
      {  
        digitalWrite(motor1,LOW);  
digitalWrite(motor2,LOW);  
cont1=posicion1;   posicion1=mr1;  
        i=201;  
        Serial.println("Posicion lista");  
      }  
    }  
  }  
}
```



```
    }  
  }  
}  
c=0; if  
(posicion2<mr2)  
{  
  for (i=0;i<=200;i++)  
  {  
    posicion2=posicion2+1;  
    Serial.println(".");  
    Serial.print("Posicion2=");  
Serial.print(posicion2);  
digitalWrite(motor4,HIGH);  
digitalWrite(motor3,LOW); if  
(mr2==posicion2)  
  {  
    digitalWrite(motor4,LOW);  
digitalWrite(motor3,LOW);  
cont2=posicion2; c=1;  
i=201;  
    Serial.println("Posicion lista");  
  }  
}  
}
```



```
if (posicion2>mr2)
{
  if (c==0)
  {
    for (i=0;i<=200;i++)
    {
      posicion2=posicion2-1;
      digitalWrite(motor3,LOW);
      digitalWrite(motor4,HIGH);

      Serial.println(".");
      Serial.print("Posicion2=");
      Serial.print(posicion2);

      if (mr2==posicion2)
      {
        digitalWrite(motor4,LOW);
        digitalWrite(motor3,LOW);
        cont2=posicion2;
        posicion2=mr2;
        i=201;
        Serial.println("Posicion lista");
      }
    }
  }
}
```



}

```
EEPROM.write(14,cont1);
```

```
EEPROM.write(15,cont2); delay
```

```
(3000); page1.show();
```

```
}
```

```
f=0;
```

```
finger.fingerID==0;
```

```
}
```

```
//-----usuario 2-----
```

```
void doss()
```

```
{ if(f==0)
```

```
{
```

```
Serial.println("BIENVENIDO Usuario 2");
```

```
Bienvenido.setText("Bienvenido Usuario 2");
```

```
Encender_Auto();
```

```
Limitador();
```

```
S1.write(EEPROM.read(4));
```

```
S2.write(EEPROM.read(5));
```

```
S3.write(EEPROM.read(6));
```

```
S4.write(EEPROM.read(7));
```

```
Serial.print("S1=");
```

```
Serial.println(EEPROM.read(4));
```



```
Serial.print("S2=");  
  
Serial.println(EEPROM.read(5));  
  
Serial.print("S3=");  
  
Serial.println(EEPROM.read(6));  
  
Serial.print("S4=");  
  
Serial.println(EEPROM.read(7));  
  
  
c=0;  
  
Serial.println(".");  
  
Serial.print("mr21:");  
  
Serial.println(EEPROM.read(16));  
  
Serial.print("mr22");  
  
Serial.println(EEPROM.read(17));  
  
Serial.print("posicion1:");  
  
Serial.println(EEPROM.read(14));  
  
Serial.print("Posicion2:");  
  
Serial.println(EEPROM.read(15));  
  
mr21=EEPROM.read(16);  
mr22=EEPROM.read(17);  
posicion1=EEPROM.read(14);  
posicion2=EEPROM.read(15); if  
(posicion1<mr21)  
{  
  for(i=0;i<=200;i++)  
  {
```



```
digitalWrite(motor1,HIGH); digitalWrite(motor2,LOW); posicion1=posicion1+1;
Serial.println(".");
Serial.print("Posicion1=");
Serial.print(posicion1); if
(posicion1==mr21)
{
digitalWrite(motor1,LOW);
digitalWrite(motor2,LOW);
cont1=posicion1; c=1;
i=201;
Serial.println("Posicion lista");

}
}
}
if (posicion1>mr21)
{
if (c==0)
{
for (i=0;i<=200;i++)
{

digitalWrite(motor1,LOW);
digitalWrite(motor2,HIGH); posicion1=posicion1-1;
```



```
Serial.println(".");  
Serial.print("Posicion1=");  
Serial.print(posicion1);  if  
(mr21==posicion1)  
{  
    digitalWrite(motor1,LOW);  
digitalWrite(motor2,LOW);  
cont1=posicion1;  posicion1=mr21;  
    i=201;  
    Serial.println("Posicion lista");  
}  
}  
}  
}  
c=0;  if  
(posicion2<mr22)  
{  
    for (i=0;i<=200;i++)  
    {  
        posicion2=posicion2+1;  Serial.println(".");  
        Serial.print("Posicion2=");  
Serial.print(posicion2);  
digitalWrite(motor4,HIGH);
```



```
digitalWrite(motor3,LOW);  if
(mr22==posicion2)
{
    digitalWrite(motor4,LOW);
digitalWrite(motor3,LOW);
cont2=posicion2;  c=1;
i=201;
    Serial.println("Posicion lista");
}
}
}
if (posicion2>mr22)
{
    if (c==0)
    {
        for (i=0;i<=200;i++)
        {
            posicion2=posicion2-1;
digitalWrite(motor3,LOW);
digitalWrite(motor4,HIGH);

            Serial.println(".");
            Serial.print("Posicion2=");
Serial.print(posicion2);  if
(mr22==posicion2)
```



{

```
digitalWrite(motor4,LOW);
```

```
digitalWrite(motor3,LOW);
```

```
cont2=posicion2;  posicion2=mr22;
```

```
i=201;
```

```
Serial.println("Posicion lista");
```

```
}
```

```
}
```

```
}
```

```
} for
```

```
(j=0;j<5;j++)
```

```
{
```

```
Serial.println(".");
```

```
}
```

```
EEPROM.write(14,cont1);
```

```
EEPROM.write(15,cont2);
```

```
delay (3000);
```

```
page1.show();
```

```
}
```

```
f=0;
```

```
finger.fingerID==0;
```

```
}
```



//-----usuario 3-----

```
----- void tress()
{

if(f==0) {

Serial.println("BIENVENIDO Usuario 3");
  Bienvenido.setText("Bienvenido Usuario 3");
  Encender_Auto();
  Limitador();
S1.write(EEPROM.read(8));
S2.write(EEPROM.read(9));
S3.write(EEPROM.read(10));
S4.write(EEPROM.read(11));
Serial.print("S1=");
Serial.println(EEPROM.read(8));
Serial.print("S2=");
Serial.println(EEPROM.read(9));
Serial.print("S3=");
Serial.println(EEPROM.read(10));
Serial.print("S4=");
Serial.println(EEPROM.read(11));
c=0;
Serial.println(".");
Serial.print("mr31:");
```



```
Serial.println(EEPROM.read(18));  
  
Serial.print("mr32:");  
  
Serial.println(EEPROM.read(19));  
  
Serial.print("posicion1:");  
  
Serial.println(EEPROM.read(14));  
  
Serial.print("Posicion2:");  
  
Serial.println(EEPROM.read(15));  
  
mr1=EEPROM.read(18);  
  
mr2=EEPROM.read(19);  
  
posicion1=EEPROM.read(14);  
posicion2=EEPROM.read(15); if  
(posicion1<mr31)  
{  
  for(i=0;i<=200;i++)  
  {  
  
    digitalWrite(motor1,HIGH);  
digitalWrite(motor2,LOW);  posicion1=posicion1+1;  
  
    Serial.println(".");  
    Serial.print("Posicion1=");  
Serial.print(posicion1);  if  
(posicion1==mr31)  
{
```



```
digitalWrite(motor1,LOW);
digitalWrite(motor2,LOW);
cont1=posicion1; c=1;
i=201;
  Serial.println("Posicion lista");

  }
}
}
if (posicion1>mr31)
{
  if (c==0)
  {
    for (i=0;i<=200;i++)
    {
      digitalWrite(motor1,LOW);
digitalWrite(motor2,HIGH);   posicion1=posicion1-1;

      Serial.println(".");
      Serial.print("Posicion1=");
Serial.print(posicion1);   if
(mr31==posicion1)
    {
```



```
digitalWrite(motor1,LOW);
digitalWrite(motor2,LOW);
cont1=posicion1;  posicion1=mr31;
  i=201;
  Serial.println("Posicion lista");
}
}
}
}
c=0;  if
(posicion2<mr32)
{
  for (i=0;i<=200;i++)
  {
    posicion2=posicion2+1;
    Serial.println(".");
    Serial.print("Posicion2=");  Serial.print(posicion2);  digitalWrite(motor4,HIGH);
digitalWrite(motor3,LOW);  if (mr32==posicion2)
  {
    digitalWrite(motor4,LOW);
digitalWrite(motor3,LOW);
cont2=posicion2;  c=1;
i=201;
```



```
Serial.println("Posicion lista");

}

}

}

if (posicion2>mr32)
{
  if (c==0)
  {
    for (i=0;i<=200;i++)
    {
      posicion2=posicion2-1;
digitalWrite(motor3,LOW);
digitalWrite(motor4,HIGH);
      Serial.println(".");   Serial.print("Posicion2=");   Serial.print(posicion2);   if
(mr32==posicion2)
      {
        digitalWrite(motor4,LOW);
digitalWrite(motor3,LOW);
cont2=posicion2;   posicion2=mr32;
        i=201;
        Serial.println("Posicion lista");
      }
    }
  }
}
```



```
    } for  
(j=0;j<5;j++)  
{  
    Serial.println(".");  
}  
EEPROM.write(14,cont1);  
EEPROM.write(15,cont2);  
delay (3000);    page1.show();  
}  
  
f=0;  
finger.fingerID==0;  
}  
  
//-----funcion agregar-----  
--  
void Agregar_(void *ptr)  
{  
    digitalWrite(28,HIGH);  
  
    Serial.println("-----Se leera la huella principal para autorizar-----");  
    ScanTxt.setText("Se leera la huella principal para autorizar");  
    while(counter<15)  
    {
```



```
getFingerprintID();
delay(100); counter++;
}
counter=0;

if(main_user==true)
{
    page10.show();

    Serial.println ("-----Selecciona el numero entre 2 y 3 para guardar la huella");
    t2.setText("Selecione la huella que desea agregar");   while (counter<=10)
    {
        Serial.println(".");
        delay(20);
        Serial.println("");
        //digitalWrite(32,HIGH);
        delay(500);

        counter++;

    }
    counter=0;

}else {
```



page2.show();

```
}  
main_user=false;  
}  
//-----Funcion borrar-----  
-----  
void Borrar(void *ptr)  
{  
digitalWrite(34,HIGH);  
  
Serial.println("-----Se leera la huella principal para autorizar-----");  
ScanTxt.setText("Se leera la Huella principal para autorizar");  
  
while(counter<15)  
{  
    getFingerprintID();  
delay(100);  
  
counter++;  
}  
counter=0;  
  
if(main_user==true)
```



```
{
  page13.show();
  Serial.println ("-----Selecciona el numero entre 2 y 3 para borrar una huella");
  t0.setText("Seleccione la huella que desea borrar");  while (counter<=20)
  {
    Serial.println(".");
    delay(20);
    Serial.println("");
    //digitalWrite(32,HIGH);
    delay(500);

    counter++;

  }
  counter=0;

  }else {
    page2.show();

  }
  digitalWrite(34,LOW);

}
//-----boton huella 2-----
```



```
void Dos(void *ptr)
{
    digitalWrite(30,HIGH);
    delay(1000); digitalWrite(30,LOW);
    be1=HIGH; if (be1==HIGH)
    {
        Serial.println("Usted selecciono al usuario 2");
        Agregado2.setText("Usted selecciono la huella 2");
        id=2;
    }
    Serial.println("Generandoel id #");
    Serial.println(id);

    while (! getFingerprintEnroll() );

    Agregado2.setText("Huella 2 Registrada");
    digitalWrite(28,LOW);
}
//-----boton huella 3----- void
Tres(void *ptr)
{
    digitalWrite(32,HIGH);
    delay(1000);
```



```
digitalWrite(32,LOW);  
be2=HIGH; if(be2==HIGH)  
  
{  
    Serial.println("Usted selecciono al usuario 3");  
Agregado2.setText("Usted selecciono la huella 3");  
    id=3;  
}  
    Serial.println("Generando el id #");  
Serial.println(id);  
  
while (! getFingerprintEnroll() );  
  
Agregado2.setText("Huella 2 Registrada");  
digitalWrite(28,LOW);  
}  
//-----Boton borrar 2-----  
void BDos (void *ptr)  
{  
    digitalWrite(36,HIGH);  
    delay(1000);  
    digitalWrite(36,LOW);  
    bd1=1;    if  
(bd1==HIGH)
```



{

```
Serial.println("Usted selecciono al usuario 2");
```

```
DelTex.setText("Selecciono la huella 2");
```

```
id=2;
```

```
}
```

```
Serial.print("Deleting ID #");
```

```
Serial.println(id);
```

```
deleteFingerprint(id);
```

```
DelTex.setText("Usted borro la Huella ");
```

```
}
```

```
//-----boton Borrar 3-----
```

```
void BTres(void *ptr)
```

```
{
```

```
digitalWrite(38,HIGH);
```

```
delay(1000);
```

```
digitalWrite(38,LOW);
```

```
bd2=1; if(bd2==HIGH)
```

```
{
```

```
Serial.println("Usted selecciono al usuario 3");
```

```
DelTex.setText("Selecciono la huella 3");
```

```
digitalWrite(38,HIGH); delay(1000);
```

```
digitalWrite(38,LOW); id=3;
```

```
}
```



```
Serial.print("Deleting ID #");
```

```
Serial.println(id);
```

```
deleteFingerprint(id);
```

```
DelTex.setText("Usted borro la Huella ");
```

```
}
```

```
//-----funcion asiento arriba-----
```

```
---
```

```
void AsientoAr (void *ptr)
```

```
{
```

```
digitalWrite(motor1,HIGH);
```

```
digitalWrite(motor2,LOW); cont1=cont1+1;
```

```
Serial.print("Cont1:");
```

```
Serial.println(cont1);
```

```
delay(100);
```

```
digitalWrite(motor1,LOW);
```

```
digitalWrite(motor2,LOW);
```

```
}
```

```
//-----Funcion asiento abajo-----
```

```
-----
```

```
void AsientoAb (void *ptr)
```

```
{
```

```
Serial.print("Cont1:");
```

```
Serial.println(cont1);
```



```
digitalWrite(motor2,HIGH);

digitalWrite(motor1,LOW);  delay(100);

digitalWrite(motor1,LOW);

digitalWrite(motor2,LOW);

cont1=cont1-1;

}

//-----funcion asiento adelante-----
-
-----

void AsientoAd (void *ptr)
{
  Serial.print("Cont2:");
  Serial.println(cont2);
  digitalWrite(motor4,LOW);
  digitalWrite(motor3,HIGH);
  delay(100);  digitalWrite(motor3,LOW);
  digitalWrite(motor4,LOW);
  cont2=cont2+1;
}

//-----funcion asiento atras-----
-----

void AsientoAt (void *ptr)
{
  Serial.print("Cont2:");
  Serial.println(cont2);
```



```
digitalWrite(motor4,HIGH);
digitalWrite(motor3,LOW);
cont2=cont2-1;
delay(100);
digitalWrite(motor3,LOW);
digitalWrite(motor4,LOW);
}
//-----funcion retrovisor izquierdo arriva-----
-----
void RArlsq (void *ptr)
{

    pos1+=15;
    S1.write(pos1);
    delay (100);
    Serial.print("pos1=");
    Serial.print(pos1);    if
    (pos1==180)

    {

        digitalWrite(31,HIGH);
        delay (100);
        digitalWrite(31,LOW);
    }else
```



```
{
  digitalWrite(31,LOW);
}
}
//-----funcion retrovisor izquierdo abajo-----
-----
void RAblzq (void *ptr)
{

  pos1-=15;
  S1.write(pos1);
  delay(100);
  Serial.print("pos1=");
  Serial.print(pos1);   if
  (pos1==0)
  {
    digitalWrite(31,HIGH); delay
    (100);
    digitalWrite(31,LOW);

  }
else
  {
    digitalWrite(31,LOW);
  }
}
```



}

//-----funcion retrovisor izquierdo izquierda-----

void Rlzlzq (void *ptr)

{

pos2+=15;

S2.write(pos2);

delay(100);

Serial.print("pos2=");

Serial.print(pos2); if

(pos2==180)

{

digitalWrite(33,HIGH);

delay (100);

digitalWrite(33,LOW);

}else

{

digitalWrite(33,LOW);

}

}

//-----funcion retrovisor izquierdo derecha-----

void RDerlzq (void *ptr)



```
{  
  pos2-=15;  
  S2.write(pos2);  
  delay(100);  
  Serial.print("pos2=");  
  Serial.print(pos2);  if  
  (pos2==0)  
  {  
    digitalWrite(33,HIGH);  
    delay (100);  
    digitalWrite(33,LOW);  
  }  
  else  
  {  
    digitalWrite(33,LOW);  
  }  
}  
  
//-----funcion retrovisor derecho arriba-----  
-  
-----  
  
void RArDer (void *ptr)  
{  
  pos3+=15;  
  S3.write(pos3);  
  delay(100);  
}
```



```
Serial.print("pos3=");
Serial.print(pos3);    if
(pos3==180)
{
    digitalWrite(33,HIGH);
delay (100);
digitalWrite(33,LOW);
}
else
{
    digitalWrite(33,LOW);
}
}
//-----funcion retrovisor derecho abajo-----
-----
void RAbDer (void *ptr)
{
    pos3-=15;
S3.write(pos3);
delay(100);
    Serial.print("pos3=");
Serial.print(pos3);    if
(pos3==0)
{
```



```
digitalWrite(33,HIGH);    delay
(100);    digitalWrite(33,LOW);
}
```

```
else
```

```
{
  digitalWrite(33,LOW);
}
```

```
}
```

```
//-----Funcion retrovisor derecho izquierda-----
-----
```

```
void RlZqDer (void *ptr)
```

```
{
```

```
  pos4+=15;
```

```
  S4.write(pos4);
```

```
  delay(100);
```

```
  Serial.print("pos4=");
```

```
Serial.print(pos4);    if
```

```
(pos4==180)
```

```
{
```

```
  digitalWrite(33,HIGH);
```

```
delay (100);
```

```
digitalWrite(33,LOW);
```

```
}
```

```
else
```



```
{
  digitalWrite(33,LOW);
}
}

//-----Funcioon retrovisor derecho izquierda-----
-----

void RDerDer (void *ptr)
{
  pos4-=15;
  S4.write(pos4);
  delay(100);
  Serial.print("pos4=");
  Serial.print(pos4);   if
  (pos4==0)

  {
    digitalWrite(33,HIGH);
  delay (100);
  digitalWrite(33,LOW);
  }
else
  {
    digitalWrite(33,LOW);
  }
}
```



```
void save( void *ptr)
```

```
{  
  sa=1;  
  digitalWrite(37,HIGH);  
  delay(500);  
  digitalWrite(37,LOW);  
  
  Serial.println("Scan.....");  
  while(counter<15)  
  {  
    getFingerprintID();  
    delay(100); counter++;  
  }  
  counter=0;  
  if(f==1) {  
    Serial.println("No Existe");  
    Tguardado.setText("No existe");  
  }  
  sa=0;  
  
  f=0;  
  finger.fingerID==0;  
  page2.show();  
}
```



```
//-----Funcion guardar posicion  
asientos-----  
-----
```

```
void mguardar(void *ptr)  
{  
  Serial.print("counter="); Serial.print(counter); sm=1;  
  texto.setText("Coloque el dedo para escanear el perfil");  
  Serial.println("Scan.....");  
  texto.setText("Scan.....");  
  while(counter<15)  
  {  
    getFingerprintID();  
    delay(100); counter++;  
  }  
  counter=0;  
  if(f==1) {  
    Serial.println("No Existe");  
  }  
  sm=0; delay  
(500);  
  
  f=0;  
  finger.fingerID==0;  
}
```



```
//-----funcion para guardar el valor
de posicion cuando no se desea guardar----- void save2 (void *ptr)
{
  EEPROM.write(14,cont1);
  EEPROM.write(15,cont2);
}
//-----guardar posicion de servos en usuario 1-----
-
-----
void ser1 ()
{ if(f==0)
{
  Serial.println("Guardado en perfil 1");
  Tguardado.setText("Guardado en perfil 1");
  for (j=0;j<5;j++)
  {
    Serial.println(".");
  }
  m1=pos1;
  m2=pos2; m3=pos3;
  m4=pos4;
  EEPROM.write(0,m1);
  EEPROM.write(1,m2);
```



```
EEPROM.write(2,m3);  
EEPROM.write(3,m4); delay  
(3000);  
  
}  
  
}  
  
//-----memoria servos usuario 2-----  
-----  
void ser2()  
{  
  
if(f==0) {  
  
    Serial.println("Guardado en perfil 2");  
Tguardado.setText("Guardado en perfil 2");  
    for (j=0;j<5;j++)  
    {  
        Serial.println(".");  
    }  
    m21=pos1;  
m22=pos2; m23=pos3;  
m24=pos4;
```



```
EEPROM.write(4,m21);  
  
EEPROM.write(5,m22);  
  
EEPROM.write(6,m23);  
  
EEPROM.write(7,m24); delay  
(3000);  
  
}  
  
}  
  
//-----memria de servos para usuario 3-----  
-----  
  
void ser3()  
{  
if(f==0) {  
    Serial.println("Guardado en perfil 3");  
    Tguardado.setText("Guardado en perfil 3");  
  
    for (j=0;j<5;j++)  
    {  
        Serial.println(".");  
    }  
  
    m31=pos1;  
m32=pos2; m33=pos3;  
m34=pos4;  
  
    EEPROM.write(8,m31);  
    EEPROM.write(9,m32);
```



```
EEPROM.write(10,m33);
EEPROM.write(11,m34);
delay (3000);
}
}
//-----funcion para memoria motorreductores usuario 1-----
-
-----
void more1()
{
  if(f==0)
  {
    Serial.println("Guardado en perfil 1");
    texto.setText("Guardado en perfil 1");

    for (j=0;j<5;j++)
    {
      Serial.println(".");
    }

    EEPROM.write(12,cont1);
    EEPROM.write(13,cont2);
    Serial.println( EEPROM.read(12));
    Serial.println( EEPROM.read(13));
```



```
EEPROM.write(14,cont1);

EEPROM.write(15,cont2);

Serial.print("\t");

delay (3000);

    page2.show();
}

}

//-----funcion memoria motorreductores usuario 2-----
-----

void more2()
{

if(f==0) {
    Serial.println("Guardado en perfil 2");
    texto.setText("Guardado en perfil 2");
    for (j=0;j<5;j++)
    {
        Serial.println(".");
    }
    EEPROM.write(16,cont1);
    EEPROM.write(17,cont2);
    Serial.println( EEPROM.read(16));
```



```
Serial.println( EEPROM.read(17));

EEPROM.write(14,cont1);
EEPROM.write(15,cont2);
Serial.print("\t");
delay (3000);
}

page2.show();
}

//-----funcion memoria motorreductores usuario 3-----
-----
void more3()
{ if(f==0)
{
Serial.println("Guardado en perfil 3");
texto.setText("Guardado en perfil 3");

for (j=0;j<5;j++)
{
Serial.println(".");
}

EEPROM.write(18,cont1);
EEPROM.write(19,cont2);
Serial.println( EEPROM.read(18));
Serial.println( EEPROM.read(19));
```



```
EEPROM.write(14,cont1);

EEPROM.write(15,cont2);

Serial.print("\t");
    delay (3000);
page2.show();

}
}

//-----Funcion para encender el auto-----
-
-----

void Encender_Auto() {
    Serial.println(" AUTORIZADA *** ");
digitalWrite(24, HIGH); // Encender el auto  for
(j=0;j<5;j++)
    {
        Serial.println(".");
    }
// delay(5000);

//digitalWrite(24, LOW); // Fin del pulso para encender el auto

}

//-----Funcion para activar el limitador-----
-----
```



```
void Limitador()
```

```
{  
  Serial.println("Usted cuanta con velocidad Limitada");  
  Serial.print("Limitador activado");  
  Vel_Lim.setText("Limitador activado");  
  digitalWrite(26,HIGH); for (j=0;j<5;j++)  
  {  
    Serial.println(".");  
  }  
  delay (3000);  
}  
//-----Funcion de mal registro-----  
-----  
void Mal_Registro() { // Activa el Buzzer 2 veces por tarjeta no autorizada  
  Serial.println("Huella no identificada");  
  Bienvenido.setText("Huella no identificada");  
  digitalWrite(22, LOW); delay(1000);  
  digitalWrite(22, LOW); delay(1000);  
  digitalWrite(22, HIGH); delay(1000);  
  digitalWrite(22, LOW); delay(1000);  
  digitalWrite(22, HIGH);  
  
  delay(1000);  
  digitalWrite(22, LOW);  
  delay(1000);
```



```
digitalWrite(22,  
HIGH); delay(1000);  
digitalWrite(22, LOW);  
delay(1000); for  
(j=0;j<5;j++)  
{  
  Serial.println(".");  
}  
f=1;  
}  
//-----Esta funcion lee la huella -----  
void getFingerprintID()  
{Serial.print(sa); uint8_t p =  
finger.getImage(); switch (p)  
{  
  case FINGERPRINT_OK:  
    break;  case FINGERPRINT_NOFINGER: return  
p;  case FINGERPRINT_PACKETRECEIVEERR: return  
p;  case FINGERPRINT_IMAGEFAIL: return p;  
  
  default: return p;  
}  
// OK success!
```



p =

```
finger.image2Tz(); switch (p)
{
  case FINGERPRINT_OK: break;      case
FINGERPRINT_IMAGEMESS: return p;   case
FINGERPRINT_PACKETRECEIVEERR: return p;
case FINGERPRINT_FEATUREFAIL: return p;   case
FINGERPRINT_INVALIDIMAGE: return p;
default: return p;
}
// OK converted!

p = finger.fingerFastSearch();

if (p == FINGERPRINT_OK)
{
  scanning = false;

  //-----manda a la activacion del vehiculo-----
  if (s==1){

    digitalWrite(41,HIGH);
    delay(200);  digitalWrite(41,LOW);
    if(finger.fingerID == 1)
    {
```



```
Serial.println("Se activo
main_user");    main_user = true;
id_ad = false;    uno();

}
if (finger.fingerID==2)
{
doss();
}
if (finger.fingerID==3)
{
tress();
}
}
/* if(finger.fingerID == 1)
{
Serial.println("Se activo main_user");
main_user = true;

id_ad = false;
}*/  if
(finger.fingerID!=1)
{
Serial.println("Se desactivo el main_user");
main_user = false;
```



```
    }  
    //-----manda a la memoria de servos de cada usuario -----  
----- if (sa==1){  
digitalWrite(39,HIGH);  
delay(200);  
digitalWrite(39,LOW);  
if(finger.fingerID == 1)  
    {  
ser1 ();  
  
    }  
    if (finger.fingerID==2)  
    {  
ser2(); }  
  
    if (finger.fingerID==3)  
    {  
ser3();  
    }  
    }  
    //-----manda a guardar en memoria de cada usuario-----  
if (sm==1)  
    {
```



```
digitalWrite(37,HIGH);  
delay(200); digitalWrite(37,LOW);  
if(finger.fingerID == 1)  
{  
  more1();  
  
}  
if (finger.fingerID==2)  
{  
  more2();  
}  
if (finger.fingerID==3)  
{  
  more3();  
}  
}  
  
f=0;  
Serial.print(" ID: #");  
Serial.print(finger.fingerID);  
  
Serial.print("%: ");  
  
Serial.print(finger.confidence);  
delay(500); for (j=0;j<5;j++)
```



```
{  
  Serial.println(".");  
}  
  
} //end finger OK  
  
else if(p == FINGERPRINT_NOTFOUND)  
{  
  scanning = false;  
  id_ad = false; counter  
= 0; MaI_Registro();  
  
  add_new_id = false;  
  main_user = false;  
  
  return p;  
  
} //end finger error  
} // returns -1 if failed, otherwise returns ID #
```



```
int getFingerprintIDez() {
uint8_t p = finger.getImage(); if (p
!= FINGERPRINT_OK) return -1;

p = finger.image2Tz(); if (p !=
FINGERPRINT_OK) return -1;

p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) {
Mal_Registro(); return -1;
}
// Si hay coincidencias de huella Serial.print("ID #"); Serial.print(finger.fingerID);
Serial.print(" coincidencia del "); Serial.println(finger.confidence);
return finger.fingerID;
}

//-----Funcion para seleccionr numero de huella-----
-----

uint8_t readnumber(void) {
uint8_t num = 0;

while (num == 0) { while
(! Serial.available()); num
= Serial.parseInt();
}
}
```



```
        return num;
    }
//-----funcion para borrar huella-----
-----
uint8_t deleteFingerprint(uint8_t id) {
uint8_t p = -1;

    p = finger.deleteModel(id);

    if (p == FINGERPRINT_OK) {
        Serial.println("Deleted!");
    } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
Serial.println("Communication error");    return p;
    } else if (p == FINGERPRINT_BADLOCATION) {
        Serial.println("Could not delete in that location");
return p;
    } else if (p == FINGERPRINT_FLASHERR) {
Serial.println("Error writing to flash");    return
p;
    } else {
        Serial.print("Unknown error: 0x"); Serial.println(p, HEX);
return p;
    }
}
//-----Funcion para agregar una huella-----
```



```
uint8_t getFingerprintEnroll() {  
  
    int p = -1;  
    Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);  
    while (p != FINGERPRINT_OK) {    p = finger.getImage();    switch  
(p) {  
  
        case FINGERPRINT_OK:  
            Serial.println("Image taken");  
            break;  
  
        case FINGERPRINT_NOFINGER:  
            Serial.println(".");  
            break;  
  
        case FINGERPRINT_PACKETRECEIVEERR:  
            Serial.println("Communication error");    break;  
  
        case FINGERPRINT_IMAGEFAIL:  
            Serial.println("Imaging error");  
            break;    default:  
            Serial.println("Unknown error");  
            break;  
    }  
}  
  
// OK success!
```



```
p = finger.image2Tz(1);
switch (p) {  case
FINGERPRINT_OK:
    Serial.println("Image converted");    break;
    case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");    return
p;
    case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");    return
p;
    case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
return p;
    case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
return p;  default:
    Serial.println("Unknown error");
return p;
}

Serial.println("Remove finger");
delay(2000);  p = 0;
```



```
while (p !=
FINGERPRINT_NOFINGER) {  p =
finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {  p =
finger.getImage();  switch (p) {  case
FINGERPRINT_OK:
    Serial.println("Image taken");
break;
    case FINGERPRINT_NOFINGER:
    Serial.print(".");
break;
    case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");  break;
    case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
break;  default:
    Serial.println("Unknown error");
break;
}
}
```



```
// OK success!

p = finger.image2Tz(2);
switch (p) {  case
FINGERPRINT_OK:
    Serial.println("Image converted");
break;
    case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");    return
p;
    case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");    return
p;
    case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
return p;
    case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
return p;  default:
    Serial.println("Unknown error");
return p;
}

// OK converted!
```



```
Serial.print("Creating model for #"); Serial.println(id);
```

```
p = finger.createModel();  
if (p == FINGERPRINT_OK) {  
    Serial.println("Prints matched!");  
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {  
Serial.println("Communication error");    return p;  
} else if (p == FINGERPRINT_ENROLLMISMATCH) {  
Serial.println("Fingerprints did not match");    return  
p;  
} else {  
    Serial.println("Unknown error");  
return p;  
}
```

```
Serial.print("ID "); Serial.println(id);  
p = finger.storeModel(id); if (p  
== FINGERPRINT_OK) {  
    Serial.println("Stored!");  
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {  
Serial.println("Communication error");    return p;  
  
} else if (p == FINGERPRINT_BADLOCATION) {  
Serial.println("Could not store in that location");  
return p;
```



```
    } else if (p ==  
FINGERPRINT_FLASHERR) {  
Serial.println("Error writing to flash"); return  
p;  
    } else {  
        Serial.println("Unknown error");  
return p;  
    }  
  
return true;  
}
```



Esta hoja está en blanco intencionalmente