



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“ Sistema de Administración Inteligente de Alumbrado Público ”

TESIS

**QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS
EN INGENIERÍA DE TELECOMUNICACIONES**

**PRESENTA:
ISMAEL MARTIN MARTIN**



MAYO DE 2018, CIUDAD DE MÉXICO

RESUMEN

Este trabajo se enfoca en desarrollar un prototipo de Sistema de gestión inteligente de alumbrado publico, integrando desde la aplicación, la comunicación, la recopilación, almacenamiento y finalmente la integración de una interfaz gráfica, esto con el objetivo de monitorear y controlar luminarios, proporcionando herramientas que permita tener certidumbre en la operación de un sistema de alumbrado publico y que en un futuro permitan implementar herramientas que puedan predecir su comportamiento, que permitirá mejorar la gestión de este tipo de sistema. Este proyecto se desarrolla empleado una tecnología de transmisión inalámbrica de datos, por sus cualidades técnicas llamado LoRa; esta tecnología es de reciente aparición en el mercado, enfocado al Internet de las cosas, que por sus capacidades técnicas, se concluyo implementar en este proyecto. Como primera parte de este trabajo se trata de describir los objetivos y la justificación de este desarrollo. En el alcance se establece los objetivos propuestos, que hacen necesario dividir el proyecto en tres secciones, modularizando estas secciones para ubicar los problemas de manera más rápida. Estas secciones son Módulo Estación, Módulo Central y una Interfaz gráfica.

El Módulo Estación se desarrolla con dos objetivos específicos de control y medición de parámetros de una luminaria. El control operará de manera automática a partir de la cantidad de luminosidad que hay en el medio, pero se condiciona su operación por un parámetro que es definido remotamente por el Modulo Central. La medición se realiza a parámetros de corriente, voltaje y estados, que son reportados a un Módulo Central.

El Módulo Central cumple la función de ser el gestor de este sistema empleado una red en estrella mediante comunicación LoRa, que se encarga de controlar remotamente, recabar y almacenar todos los datos de los Módulos Estación. Se desarrolla en una Raspberry PI 3, y con el demonio del sistemas operativo de tareas programadas se realiza la petición de información de manera periódica a los Módulos Estación; esto permite tener el control de la red en lo relativo de peticiones. Adicionalmente se establece la estampa de tiempo a todas la mediciones obtenidas.

La interfaz gráfica se desarrolla como página web que permite solo acceso de personal autorizado. Este interfaz controla remotamente a los módulos Estación desde la página de inicio, mediante botones que ejecutan los programas de python que envían los comandos de control a los respectivos Módulos Estación. Para la visualización de los datos se emplea el firmware HighChart compatible con php. Además se muestra los datos de los eventos en un tabla, que permite conocer el comportamiento de la luminaria.

Este tipo de tecnología son necesarias para el conocimiento de un sistema como el alumbrado publico para mejorar su gestión, ahorrar energía y su mantenimiento. Este prototipo tiene puntos débiles que se tiene que mejorar, ademas de que se tiene que evaluar como operará en un sistemas real, pero es el primer paso para un propuesta de mercado.

ABSTRACT

This work focuses on developing a prototype Intelligent street lighting management system, integrating from the application, communication, collection, storage and finally the integration of a graphic interface, this with the objective of monitoring and controlling luminaires, tools knowledge that allows to have certainty in the operation of a public lighting system and that in the future, when tools are implemented that can predict their behavior, that allow improving the management of this type of system. This project has been developed as a tool for wireless data transmission, due to its technical characteristics called Lora; This technology is of recent appearance in the market, focused on the Internet of Things, which due to its technical capabilities, can be implemented in this project. The first part of this work is to describe the objectives and justification of this development. In the scope, the proposed objectives are established, which make it necessary to divide the project into three sections, modularizing these sections to locate the problems more quickly. These sections are Station Module, Central Module and a Graphic Interface.

The Station Module is developed with two specific objectives of control and measurement of parameters of a luminaire. The control operates automatically from the amount of luminosity that exists in the medium, but its operation is conditioned by a parameter that is remotely defined by the Central Module. The measurement is made to current, voltage and status parameters, which are reported to a Central Module.

The Central Module fulfills the function of being the manager of this system using a star network through LoRa communication, which is responsible for remotely controlling, collecting and storing all the data of the Station Modules. It is developed in a Raspberry PI 3, and with the daemon of the operating systems of programmed tasks the information request is made periodically to the Station Modules; this allows control of the network in relation to requests. Additionally, the time stamp is established for all the measurements obtained.

The graphic interface is developed as a web page that allows only access by authorized personnel. This interface remotely controls the Station modules from the home page, using buttons that execute the python programs that send the control commands to the respective Station Modules. The HighChart firmware compatible with php is used to display the data. In addition, the data of the events is shown in a table, which allows to know the behavior of the luminaire.

This type of technology is necessary for the knowledge of a system such as public lighting to improve its management, save energy and its maintenance. This prototype has weak points that have to be improved, in addition to the fact that it has to be evaluated how it will operate in real systems, but it is the first step for a market proposal.

TABLA DE CONTENIDO

Resumen	vii
Abstract	viii
Tabla de Contenido	ix
Indice de Figuras	xii
Indice de Tablas	xiv
Capítulo 1 Introducción	1
1.1 Justificación	1
1.2 Planteamiento del proyecto	1
1.3 Objetivo general	2
1.4 Objetivos particulares	2
1.5 Alcance	3
1.6 Metodología	4
1.7 Descripción de la Tesis	6
Capítulo 2. Sistemas inalámbricos de Comunicaciones	7
2.1 Introducción	8
2.2 Lora	10
2.3 Wifi	12
2.4 Wimax	14
2.5 Evaluación de Sistemas inalámbricos de Comunicaciones	15
Capítulo 3 Alumbrado Público	16
3.1 Análisis del alumbrado Público	16
3.2 Tendencias del Alumbrado Público	17
3.3 Caso de la Ciudad de México	18
Capítulo 4 Diseño del sistema	26
4.1 Estructura del Sistema de Administración Inteligente	26
4.2 Diseño del Módulo Estación de medición y control	27
4.2.1 Funciones de medición y control	28
4.2.1.1 Medición de voltaje	29
4.2.1.2 Medición de corriente	31
4.2.1.3 Sensor de Luz	33
4.2.1.4 Recepción de datos e Interpretación de comandos	35
4.2.1.5 Registro de eventos	37
4.2.1.6 Transmisión de datos	39
4.3 Diseño del Módulo Central de Control	41
4.3.1 Procedimiento de Administración Inteligente	41
4.3.2 Sistema de Comunicación	42
4.3.3 Procedimiento de control	44
4.4 Diseño de Interfaz gráfico	47
4.4.1 Administración de base de datos	47
4.4.2 Arquitectura Modelo- Vista-Controlador	47

Capítulo 5 Pruebas	54
5.1 Pruebas de la Estación de Medición y control	54
5.1.1 Verificación de mediciones de voltaje	54
5.1.2 Verificación de mediciones de Corriente	55
5.1.3 Verificación de sensor de luz para control de encendido y apagado de lampara	55
5.1.4 Verificación de registro de eventos	56
5.2 Pruebas del Sistema de comunicación	56
5.3 Pruebas del sistema Integral	59
Capítulo 6 Conclusiones y Trabajos a Futuro.	65
6.1 Conclusiones	65
6.2 Trabajos a Futuro.	65
6.3 Recomendaciones para Implementación de una aplicación similar	66
Apéndice A Dispositivos empleados	67
A.1 Fotorresistencia	67
A.2 Sensor de Voltaje	67
A.3 Sensor de Corriente	68
A.4 Raspberry	69
A.5 Arduino Uno	71
A.6 Diagrama de Modulo Shield Estación	72
A.7 Modulo de Potencia	73
A.8 Antenas Inalambricas DRF1276DM-DORJI	73
A.8.1 Características:	73
A.8.2 Aplicaciones	74
A.8.3 Descripción	74
A.8.4 Funciones de pines	74
A.8.5 Especificaciones Eléctricas	75
A.8.6 Rangos de Valores Máximos	76
A.8.7 Modos de Trabajo	76
A.8.7.1 Modo Estándar	76
A.8.7.2 Modo Conexión en red estrella	76
A.9 DS3231SN	80
A.10 Diagrama de Modulo Shield Central	81
Apéndice B Configuraciones	82
B.1 Configuración de Antenas.	82
B.2 Configuración de Raspberry.	85
B.3 Configuración de conexión wifi vía terminal.	86
B.4 Configuración de puerto I2C y RTC.	87
B.5 Configuración de puerto serie.	89
B.6 Configuración de permisos de puerto serie y GPIOs.	90
B.7 Instalación de servidor LAMP.	90
B.8 Configuración de permisos de superusuario a php.	92
B.9 Configuración de servicio cron	92
Apéndice C Código de proyecto.	94
C.1 Programación de Módulo Estación	94
C.2 Programación de Módulo Central	96
C.2.1 Programa de control remoto de encendido (ON_ID2.py)	96
C.2.2 Programa de Control remoto de Apagado (OFF_ID2.py)	96
C.2.3 Programa de Petición de Parámetros de Medición (LEC_ID2_v2_6.py)	96
C.3 Programación de Interfaz Gráfica	97
C.3.1 Login	97

C.3.1.1 Archivo index.php	97
C.3.1.2 Archivo manejador.php	98
C.3.2 MVC	99
C.3.2.1 Archivo index.phh	99
C.3.2.2 Modelo	99
C.3.2.2.1 Archivo model.php	99
C.3.2.3 Controlador	99
C.3.2.3.1 Archivo controller.php	99
C.3.2.4 Vistas	99
C.3.2.4.1 Archivo template.php	99
C.3.2.4.2 Archivo inicio.php	100
C.3.2.4.3 Archivo consumogeneral.php	100
C.3.2.4.4 Archivo eventos.php	102
C.3.2.4.5 Archivo salir.php	103
C.3.2.4.6 Archivo connection.php	103
Bibliografía	104

INDICE DE FIGURAS

Capítulo 1 Introducción	1
Capítulo 2 Sistemas inalámbricos de Comunicaciones	7
Figura 2.1 Arquitectura de red LoRaWAN.	12
Capítulo 3 Alumbrado Público	16
Figura 3.1 Alumbrado público operando lámparas de Vapor de Mercurio (VM) sin color corregido.	19
Figura 3.2 Alumbrado público operando lámparas de Vapor de Mercurio (VM) con color corregido.	19
Figura 3.3 Alumbrado público operando lámparas de Vapor de Sodio en Alta Presión (VSAP).	20
Figura 3.4 Alumbrado público operando lámparas de Vapor de Sodio en Baja Presión (VSBP).	21
Figura 3.5 Alumbrado público operando lámparas de Aditivos Metálicos (AM).	21
Figura 3.6 Alumbrado público operando módulos con Diodos Emisores de Luz (LEDs).	22
Figura 3.7 Alumbrado público operando lámparas de inducción electromagnética (IND).	22
Figura 3.8 Alumbrado público operando lámparas Aditivos Metálicos Cerámicos (AMC).	23
Figura 3.9 Tendencia de Consumo Energético de las diferentes Tecnologías de luminarios.	24
Figura 3.10 Tendencia de Tiempo de Vida de Distintas Tecnologías de Luminarios	25
Figura 3.11 Índice de Temperatura de Color	25
Capítulo 4 Diseño del sistema	26
Figura 4.1 Estructura de Sistema de Administración Inteligente.	26
Figura 4.2 Objetivos de las Secciones del Proyecto.	27
Figura 4.3 Resumen de integración de Módulo Estación.	27
Figura 4.4 Resumen de Diagramas de estado de Módulo Estación.	28
Figura 4.5 Diagrama de conexión eléctrica del circuito de medición de voltaje.	29
Figura 4.6 Diagrama de Flujo de medición de voltaje.	31
Figura 4.7 Diagrama de Conexión Eléctrica del circuito de Medición de corriente.	32
Figura 4.8 Diagrama de de Flujo de Medición de Corriente	32
Figura 4.9 Diagrama de Conexión Eléctrica de fotoresistencia.	33
Figura 4.10 Diagrama de Flujo de Medición de nivel de luminosidad.	34
Figura 4.11 Diagrama de de Flujo de control de encendido y apagado de lampara.	34
Figura 4.12 Diagrama de conexión eléctrica de antena DRF1276DM y Arduino.	35
Figura 4.13 Diagrama de Flujo para Recepción de datos.	36
Figura 4.14 Diagrama de Conexión Eléctrica de Registro de eventos.	37
Figura 4.15 Diagrama de Flujo de Registro de Eventos.	38
Figura 4.16 Secuencia de Trama de datos	39
Figura 4.17 Diagrama de Flujo de transmisión de datos.	40
Figura 4.18 Resumen de integración de Módulo Central.	41
Figura 4.19 Estructura de trama de datos de peticiones a lamparas.	42
Figura 4.20 Diagrama de conexión eléctrica de antena DRF1276DM y Raspberry.	43
Figura 4.21 Diagrama de conexión eléctrica de modulo DS3231 y Raspberry.	43
Figura 4.22 Diagrama de Flujo de transmisión de datos para apagado de lamparas.	44
Figura 4.23 Diagrama de Flujo de transmisión de datos para habilitación de encendido de lamparas.	45
Figura 4.24 Diagrama de Flujo de Petición de Lectura de Parámetros.	46
Figura 4.25 Diagrama de secuencia de MVC.	48
Figura 4.26 Construcción de proyecto con Arquitectura MVC.	49
Figura 4.27 Diagrama de secuencia de arquitectura MVC.	50
Figura 4.28 Página de Inicio de Interfaz Gráfico.	52
Figura 4.29 Página de Consumo de un lampara en Interfaz Gráfico.	53
Capítulo 5 Pruebas	54
Figura 5.1 Diagrama de pruebas de Transmisión de información.	57

Figura 5.2 Primera cadena de caracteres usada para probar comunicación LoRa.	57
Figura 5.3 Segunda cadena de caracteres usada para probar la comunicación LoRa.	57
Figura 5.4 Diagrama de pruebas de Transmisión y Recepción de información.	58
Figura 5.5 Respuestas obtenidas a nuestra peticiones de información.	58
Figura 5.6 Diagrama de prueba de distancia.	59
Figura 5.7 Prueba de cobertura a una distancia de 523.46 m.	60
Figura 5.8 Diagrama de prueba de tiempos de respuesta.	61
Figura 5.9 Mediciones de tiempo de respuesta.	62
Capítulo 6 Conclusiones y Trabajos a Futuro.	65
Figura 6.1 Diagrama resumen de puntos a revisión en la implementación de un proyecto similar con tecnología LoRa	66
Apéndice A Dispositivos empleados	67
Figura A.1 Fotoresistencia de 2M.	67
Figura A.2 Modulo FZ0430, sensor de voltaje.	67
Figura A.3 Distribución de pines de FZ0430.	68
Figura A.4 Modulo ACS712, Sensor de Corriente.	68
Figura A.5 Distribución de pines de ACS712.	69
Figura A.6 Vista general de Raspberry Pi 3 Modelo.	70
Figura A.7 Vista general de Arduino Uno Rev.3	71
Figura A.8 Circuito General de Módulo de Estación.	72
Figura A.9 Circuito para carga Resistiva.	73
Figura A.10 Módulo DRF1276DM.	73
Figura A.11 Conexiones de Pines en Modo Estándar.	76
Figura A.12 Modulo Nodo en comunicación pausada.	77
Figura A.13 Tiempo de levantamiento (waker time) y Periodo de Pausado (Breath Period).	77
Figura A.14 Modulo Nodo en Comunicación Express	78
Figura A.15 DS3231SN	80
Figura A.16 Circuito General de Módulo Central	81
Apéndice B Configuraciones	82
Figura B.1 Conexión de pines en modo configuración de antena y conexión en DAC02.	84
Figura B. 2 Herramienta de configuración de módem LoRa.	84
Apéndice C Código de proyecto.	94

INDICE DE TABLAS

Capítulo 1 Introducción	1
Capítulo 2 Sistemas inalámbricos de Comunicaciones	7
Tabla 2.1 Frecuencias LoRaWAN para distintas áreas del globo.	12
Tabla 2.2 Tabla comparativa de las tres tecnologías WiMAX, WIFI y LoRa.	15
Capítulo 3 Alumbrado Publico	16
Tabla 3.1 Resumen de la evolución de distintas tecnologías de alumbrado publico	24
Capítulo 4 Diseño del sistema	26
Tabla 4.1 Mediciones de voltaje primario y voltaje secundario de transformador reductor.	30
Tabla 4.2 Tabla se relación de códigos de errores.	37
Tabla 4.3 Descripción del problema probable de los distintos códigos.	37
Tabla 4.4 Ajuste de pines para el cambio de tipo de comunicación	44
Capítulo 5 Pruebas	54
Tabla 5.1 Mediciones de voltaje con medidor PM800 (Schneider Electric) vs Modulo Estación.	54
Tabla 5.2 Mediciones de Corriente con Amperimetro UNI-T UT203 vs Modulo Estación.	55
Tabla 5.3 Tabla de resultados de pruebas a registro de eventos.	56
Table 5.4 Resultados de prueba de distancia.	60
Table 5.5 Tiempos teóricos considerados.	61
Table 5.6 Tiempos medidos de respuesta.	61
Table 5.7 Muestreo de 100 muestras de datos recopilados.	62
Capítulo 6 Conclusiones y Trabajos a Futuro.	65
Apéndice A Dispositivos empleados	67
Tabla A.1 Parámetros de distintas versiones del módulo ACS712.	69
Tabla A.2 Especificaciones técnicas de Arduino.	71
Tabla A.3 Funciones de pines de DRF1276DM.	74
Tabla A.4 Especificaciones eléctricas de DRF1276DM.	75
Tabla A.5 Valores máximos DRF1276D.	76
Tabla A.6 Tiempo de levantamiento, tasa de datos RF y Sensibilidad.	78
Apéndice B Configuraciones	82
Tabla B.1 Parámetros de Configuración de Antenas DRF1276DM.	82
Tabla B.2 Sensibilidad, RF_Factor y RF_BW.	85
Table B.3 Descripción de programación de tareas programadas con cron.	93
Apéndice C Código de proyecto.	94

CAPTULO 1 Introducción

- 1.1 Justificación
- 1.2 Planteamiento del proyecto
- 1.3 Objetivo general
- 1.4 Objetivos particulares
- 1.5 Alcance
- 1.6 Metodología
- 1.7 Descripción de la Tesis

1.1 Justificación

El alumbrado publico para una sociedad es de gran importancia, debido a la percepción de seguridad que genera, lo que facilita que actividades económicas se desarrollen para esta sociedad. El factor mas importante del alumbrado publico para una población es el costo de operación, por ello se requiere un sistema de gestión que permita reducir costos de operación, en rubros como el consumo de energía o la cantidad de personal para la operación. Y que reducción de estos costos no tenga impacto en la operación del alumbrado publico, es la razón e interés en desarrollar un prototipo de sistema de gestión automático de alumbrado publico, que permita reducir el consumo de energía eléctrica y que su gestión pueda ser realizada por un mínimo de personal.

La idea de este trabajo, nace a través de mis primeros años de experiencia profesional como ingeniero electricista, donde me encontré con distintos sistemas de monitoreo a sistemas eléctricos, de distintos fabricantes, con distintos propósitos particulares, pero con un propósito general muy similar, la de monitorear y generar un historial de equipos críticos, esto permitiría identificar deterioros mediante el comportamiento de uso, también se puede generar indices de fallas incluso poder visualizar las condiciones de una falla, pero ademas se puede generar herramientas para el ahorro de energía. Estos sistemas ayudan a los usuarios a prevenir salidas de servicio, conocer mejor el comportamiento del sistema, que tipos de fallas son recurrentes y donde se localizan; todo esto permitirá identificar puntos de oportunidad para el ahorro de energía.

1.2 Planteamiento del proyecto

Como objetivo principal es el desarrollo de un prototipo de sistema de gestión inteligente de alumbrado publico. La columna vertebral del alumbrado publico son las luminarias y para este sistema de gestión debemos obtener parámetros de cada una de las luminarias, por lo que se requiere una herramienta de comunicación que nos permita transmitir y recibir información al que llamaremos módulo estación echando mano de las distintas tecnologías desarrolladas actualmente. Como segunda parte fundamental es el desarrollo de un centro de transmisión y recepción información con cada una de las luminarias al que llamaremos modulo central. Como

ultima parte del proyecto, desarrollar de un interfaz gráfica de gestión que permita el control y monitoreo de cada una de las lamparas.

Este proyecto constará de tres principales secciones, por lo que es necesario describir los principales objetivos de cada sección:

I. Módulo de Estación

- Autonomía en el control de encendido y apagado de luminaria a partir de la cantidad de luz.
- Recepción y transmisión de información.
- Interpretación de comandos recibidos.
- Transmisión de parámetros de la luminaria (encendido, apagado, voltaje de alimentación, corriente y fallas).

II. Módulo Central

- Transmisión y recepción de información a cada una de las luminarias
- Transmisión de comandos de operación hacia las luminarias
- Recabar y almacenar la información obtenida de cada una de las luminarias

III. Interfaz Gráfica de Gestión

- Permitir acceso solo a personal autorizado.
- Controlar remotamente la operación de cada una de las luminarias
- Mostrar gráficamente los parámetros de cada una de las luminarias(estado de encendido, estado de apagado, registro de consumo de energía y fallas)

1.3 Objetivo general

Desarrollar un prototipo de sistema de gestión de alumbrado publico, dividido en tres secciones principales, módulos estación que estarán colocados en cada luminaria, módulo central capaz de enviar y recabar información de cada una de las luminarias y el interfaz gráfico de gestión capaz de enviar comandos a cada una de las luminarias y publicar la información de cada una de ellas.

1.4 Objetivos particulares

Módulo Estación

Los módulos estación estarán colocados en cada una de la luminarias, tendrán el papel de comunicarse con el modulo central, que permitirá la gestión. Esta comunicación debe ser bidireccional.

- Operar de formar autónoma el encendido y apagado de la luminaria dependiendo de la cantidad de luz solar.
- Medir el consumo de Voltaje y Corriente instantáneo.

- Capaz de transmitir y recibir información
- Interpretar la información recibida
- Ejecutar los comandos recibidos (Bloqueo de operación, habilitación de operación y el envío de reporte de estados de la luminaria).
- Poder trabajar con cualquier tecnología de luminaria de la actualidad.

Módulo Central

El propósito de modulo central es establecer comunicación, recabar información y almacenar la información obtenida de cada una de las luminarias.

- Capaz de transmitir y recibir información con cada uno de los Módulos Estación de la red.
- Formular cadena de datos a transmitir dependiendo del Módulo Estación.
- Almacenar los datos recibidos de los módulos estación
- Interaccionar con la interfaz gráfica

Interfaz Gráfica

El objetivo del interfaz gráfica es permitir el control interactivo de las luminarias y mostrar gráficamente los parámetros de las luminarias desde cualquier dispositivos de comunicación (computador, celular, table's, etc.), solo a personal autorizado.

- Control de acceso
- Visualizar estados de las lamparas
- Acceder a información de cada una de las lamparas (ubicación, tecnología, fecha de colocación)
- Visualizar eventos (fallas)
- Reporte de periodos de operación y consumo

1.5 Alcance

Ya que el alcance es considerablemente amplio el alcance se dividirá en tres secciones que son los siguientes: Módulo Estación, Módulo Central e Interfaz Gráfica ya descritos. En este proyecto se empleará la tecnología LoRa, por motivos que se justificará más adelante en este trabajo, para el sistema de comunicación.

Modulo estación:

Implementar modulo que realice las siguientes funciones:

- Medir el voltaje y la corriente eléctrica que consume la luminaria
- Sensar el nivel de luminosidad
- Controlar el dispositivo de comunicación (DRF1276DM).

- Enviar y recibir datos de estado (encendido o apagado).
- Interpretar los datos recibidos del modulo central.
- Apagar o encender la luminaria automáticamente dependiendo del nivel de luminosidad solar.
- Ejecutar comandos de control remoto recibidos del Módulo Central.

La implementación a desarrollar debe permitir una conexión y desconexión rápida.

Módulo Central:

Implementar módulo que realice las siguientes funciones:

- Recibir y almacenar los datos recibidos de cada una de las luminarias.
- Formular las tramas de datos de comandos remotos a enviar a las luminarias.
- Interaccionar con la Interfaz gráfico
- Controlar dispositivo de comunicación LoRa (DRF1276DM).

La implementación a desarrollar debe permitir una conexión y desconexión rápida.

Interfaz Gráfica:

Implementar una interfaz gráfica que permita las siguientes funciones:

- Control remoto interactivo en la operación de luminarias
- Mostrar de gráficamente el comportamiento de operación de luminarias
- Mostrar el estado de las luminarias (encendido o apagado).
- Mostrar detalles de cada una de las luminarias(ubicación, tecnología, fecha de colocación)
- Mostrar eventos en la luminaria (fallas)

1.6 Metodología

La implementación es un trabajo arduo por ello se emplearon materiales evitando afectar a la economía del desarrollador. Seccionando el proyecto como se realizó anteriormente con los tópicos anteriores se describen la metodología a emplear para realizar la implementación de cada una de secciones de este proyecto, así como algunas justificaciones técnicas que llevaron a seleccionar dichas tecnologías respetando los objetivos que se propusieron para la implementación de este proyecto.

Módulo Estación

Esta sección del proyecto será implementada mediante un microcontrolador arduino debido a la accesibilidad que se tiene de este tipo de microcontrolador en el mercado, además de su versatilidad y fácil uso, se adapta de manera muy amigable con el desarrollo de este tipo de prototipos.

Como segundo punto para el sistema de comunicación conociendo que se desarrollará con la tecnología LoRa (Antena DRF1276DM by Dorji), se implementa Módulo para conexión rápida con microcontrolador arduino y antena que en lo sucesivo lo mencionaremos como Shield Estación.

Aprovechando el desarrollo del Shield Estación, en este modulo colocarán los sensores de medición de luz solar y de medición de voltaje y corriente instantáneo, con el objetivo de que este módulo Shield permita una conexión rápida del sistema de comunicación y los sensores que emplearemos en para este prototipo.

Como tercer punto importante es la implementación de un módulo de potencia, con el objetivo de habilitar o inhabilitar la operación las luminarias y poder trabajar con cualquier tecnología de luminarias, este modulo de potencia tendrá el objetivo de solo habilitar o inhabilitar la alimentación de la luminaria.

Módulo Central

Para está sección del proyecto que es empleará una Raspberry Pi 3 B para el módulo central, ya que tiene un puerto uart con el cual se puede comunicar con la antena, ademas de que los programas que tiene por defecto la Raspberry (python en especifico), permite acumular datos, así como también permite construir las tramas con los datos específicos que se requieren para la adecuada comunicación entre los módulos.

Otras de las ventajas de usar de la Raspberry es que permite la instalación de un servidor web que será requerido para probar la implementación de la interfaz gráfica.

Siguiendo con el desarrollo del módulo central se requiere desarrollar modulo de conexión rápida, para la conexión entre la Raspberry y el sistema de comunicación, dando portabilidad a este módulo; en lo siguiente se referirá a este módulo como Módulo Shield Central.

Interfaz Gráfica

Está última sección del proyecto, se empleará al igual que la sección de Módulo Central, una Raspberry. Esto da la posibilidad de implementar dos secciones del proyecto en un equipo, reduciendo así el costo del desarrollo de esté proyecto.

Para la implementación de la interfaz gráfica, con el objetivo de que esta interfaz pueda ser operada desde cualquier dispositivo (Teléfono móvil, Computadora, Laptop) se opta por el desarrollo web resposive. Existen una variedad de programas que pueden ser utilizados para el desarrollo de esta interfaz, pero para el desarrollo de este proyecto no cabe la posibilidad de probar cada uno de ellas y poder explotar sus capacidades; para este prototipo se empleará la configuración LAMP (Linux, Apache, MySQL, PHP), debido al soporte que existe para esta configuración y como trabajos futuros se puede explorar la configuración MEAN (Mongo DB, Express, Angular.js, NODE.js).

Como se menciona con anterioridad, se emplea Apache como servidor web, para poder evaluar nuestro interfaz gráfica, el desarrollo de esta interfaz se realiza principalmente con PHP (lenguaje de desarrollo web) ya que este

programa da la posibilidad de invocar archivos de python (programa empleado para construir las tramas de datos, así como para enviar esta trama de datos). Y MySQL como programa de base de datos para almacenar los datos.

1.7 Descripción de la Tesis

Capítulo 1: En este primer capítulo se presenta una introducción sobre el proyecto a implementar principalmente abarca las cuestiones de justificación, objetivos, alcances del proyecto, así como una breve descripción de manera en que se realizó este proyecto.

Capítulo 2: En el segundo capítulo se trata de explorar las opciones tecnológicas de sistemas de comunicación que se tiene para realizar este tipo de proyectos revisando las ventajas y desventajas de cada una de ellas, y la justificación técnica por la cual se opta por una en particular.

Capítulo 3: Para tercer capítulo se expone a grandes rasgos la evolución que se ha tenido el alumbrado público en el mundo y en México en particular. También se revisan las tendencias tecnológicas que se han desarrollado para el alumbrado público, en particular la telegestión, el enfoque, lo propósitos, las ventajas y desventajas que tiene este tipo de tecnología. Además de revisar los problemas que enfrenta los sistemas de alumbrado público.

Capítulo 4: El cuarto capítulo abarca la descripción, de diseño y la implementación de las diferentes secciones que esta compuesto este proyecto. Seccionando en tres principales partes, módulo estación, módulo central e interfaz gráfico. Como primer paso se establece la arquitectura de este proyecto para continuar con la descripción de cada una de las partes.

Capítulo 5: En este quinto capítulo se realizan diferentes pruebas a cada una de las secciones del proyecto, y por último varias pruebas en conjunto verificando el adecuado funcionamiento en transmisión, recepción y almacenamiento de datos.

Capítulo 6: En el último capítulo se aportan las conclusiones y trabajos futuros posibles para este proyecto, además que se agrega un resumen de los pasos a seguir para generar un proyecto similar.

Anexo A: En este anexo, se describen todos los dispositivos empleados y circuitos implementados para desarrollar este proyecto, esto con el propósito de ser replicado,

Anexo B: Se describe en una primera parte la configuración de las antenas empleado la herramienta de configuración del fabricante, y como segunda parte se describe todas la configuraciones realizadas a la Raspberry empleada para hacerla funcional para este proyecto como módulo central y servidor del interfaz gráfico.

Anexo C: Contiene todo el código desarrollado para este proyecto del Módulo Estación, Módulo Central y la Interfaz gráfica.

Capítulo 2. Sistemas inalámbricos de Comunicaciones

- Sistemas inalámbricos de Comunicaciones
- 2.1 Introducción
- 2.2 Lora
- 2.3 Wifi
- 2.4 Wimax
- 2.5 Evaluación de Sistemas inalámbricos de Comunicaciones

En un sentido amplio y general, entendemos por comunicaciones inalámbricas aquellas comunicaciones entre dispositivos (móviles o no) o entre personas que intercambian información utilizando el espectro electromagnético.

Las comunicaciones inalámbricas a pesar de la publicidad, es un campo con alrededor de cien años, inicio alrededor de 1897 con las demostraciones exitosas de telegrafía inalámbrica de Marconi.

Para 1901 la recepción de radio a través del océano Atlántico se había establecido; por lo tanto el rápido progreso de la tecnología ha estado presente por bastante tiempo. En un intervalo de cien años, varios tipos de sistemas inalámbricos han florecido y otros también han desaparecido.

La transmisión inalámbrica ha pasado desde radio enlaces de punto a punto o redes de transporte de larga distancia terrestre y satelital o a redes de acceso inalámbrico en zonas urbanas para comunicaciones móviles de banda ancha.

La propagación de radio ha sido tradicionalmente un problema difícil de analizar ya que a diferencia de la propagación guiada, experimenta desvanecimientos aleatorios por absorción y multitrayectorias, especialmente en comunicaciones móviles. Además de que una tecnología puede no servir para todos los proyectos de aplicación y volúmenes para Internet de las cosas. Wifi y BTLE son estándares ampliamente adoptados y sirven para aplicaciones relacionados con dispositivos de comunicación personal. La tecnología de telefonía celular es muy adecuado para aplicaciones que se necesitan un alta transferencia de datos y una fuente de alimentación. LPWAN permite un tiempo de vida de la batería por varios años y esta diseñado para sensores y aplicaciones que necesitan enviar pequeñas cantidades de datos a través de largas distancias varias veces por hora de diferentes puntos.

Los factores mas críticos a revisar en un LPWAN son:

- Arquitectura de red.
- Rango de comunicación.
- Tiempo de vida de la batería o baja potencia
- Robustez a interferencias
- Capacidad de Red (máximo numero de nodos en una red)
- Seguridad en Red
- Comunicación una-via vs dos-vias.

- Diferentes servicios de aplicaciones.

Existe mucha actividad en el sector de IoT, la comparación de opción es LPWAN, existen dos principales rubros la comparación técnica y sobre la perspectiva del modelo de negocio. Las redes LPWAN están siendo desarrolladas por que tienen un caso muy fuerte de negocio que apoyarían inmediatamente el desarrollo, y el costo de desarrollar redes en bandas libres con mucho menor capital como lo que ocurre con la actualización del software de 3G. Las preguntas que debemos responder para compara las diferentes tecnologías LPWAN son:

- Flexibilidad para desarrollar una gran variedad de aplicaciones.
- ¿Es un protocolo de comunicación seguro?
- Aspectos técnicos- rango, capacidad, dos vías de comunicación, robustess a la interferencia.
- El costo del desarrollo de redes, el costo de desarrollo de nodos terminales, costo de la batería.
- Ecosistema de proveedores de solución para modelos flexibles de negocio.
- La disponibilidad de productos para asegurar ROI del desarrollo de la red.
- Fortaleza del ecosistema para asegurar la calidad y la longevidad de la solución.

2.1 Introducción

Un sistema inteligente de control de alumbrado público incorpora un Modulo Estación en cada luminaria que le permite reportar valores de funcionamiento, alarmas y eventos asociados a un centro del control.

Las luminarias y controladores se vinculan mediante la tecnología de comunicaciones LORA. Por sus especificaciones, este tipo de comunicación se adapta en forma optima a las características geográficas de las instalación existente del alumbrado publico.

Los controladores y el centro de control, se vincula utilizando conexiones seguras (SSL) sobre Internet con comunicaciones TCP/IP sobre ADSL, 3G o GSM dependiendo de la cobertura telefónica y celular de cada zona o bien por fibra óptica en aquellas zonas en las que se posea infraestructura de comunicaciones.

Los motivos para el control un sistema de alumbrado publico:

- Generar un ahorro en energía: La combinación de LED y sistemas de control permite conseguir un ahorro energético significativo comparación con instalaciones equipadas con fuentes de luz tradicionales.
- Mejora la gestión de la operaciones: Las interrupciones y el cierre de carreteras durante las operaciones de mantenimiento conllevan muchas molestias; a demás perdidas de dinero y tiempo, aumento de las distancias recorridas y dificultades para los residentes.

Las prestaciones de información de un sistema de alumbrado publico permiten supervisar y valorar la instalación de iluminación de forma independiente, y proporciona la información necesaria para gestionar de manera inteligente la red durante toda su vida de servicio.

- Proporciona luz a petición: Cuando los espacios públicos no se usan la iluminación a plena potencia es un desperdicio de energía. Los escenarios de regulación de las prestaciones de luz a petición pueden adaptar la iluminación a las necesidades reales del lugar y al horario.
- Asegurar la fiabilidad: Al controlar cada punto de iluminación, los sistemas de gestión pueden prevenir fallas detectando problemas operativos (lamparas rotas, temperatura de dispositivos, sobretensiones).

Red Intergestionable

- Seguridad
- Eventos: Los eventos son almacenados en una base de datos con una estampa de tiempo exacta y posición geográfica. Los informes pueden ser predefinidos y personalizados para enfocarse en la información más interesante; permitiendo un buena gestión de datos.
- Integración de terceros: Gracias a la tecnología LORA y su almacenamiento de datos flexible en MySQL. Esta flexibilidad aumenta las funcionalidades más allá de la iluminación.
- Situación de emergencia: En caso de problemas de control, el sistema pasa al programa por defecto que asegura que la instalación de iluminación no se apague y siga proporcionando seguridad en el espacio publico.
- Gestión de datos: Toda la reacción del esquema de iluminación es almacenada en una base de datos que permite hacer evaluaciones a largo plazo, como análisis de la energía, la previsión de la vida útil de la lampara, detección de fallas.
- Gestión de alarmas instantáneas: Los eventos son registrados, identificados y localizados en tiempo real. La información puede ser automáticamente enviada a los operarios.
- Compatibilidad absoluta: Se puede trabajar con cualquier tipo de lampara, balastro o driver LED-auxiliar magnético, doble alimentación, reductor, balastro electrónico con o sin interfaz.

Ventajas de un sistema inalámbrico de control de alumbrado publico:

- Eficacia: La red inalámbrica se basa en el protocolo LORA . Se comunica 50 veces más rápido que una red eléctrica de 16 canales de comunicación, cada una con un ancho de banda máximo.
- Fiabilidad: Gracias a la funcionalidad de una red en estrella inalámbrica, la red inalámbrica encuentra el mejor recorrido de comunicación. El sistema incluye las funciones de autoregeneración y normalización. Incluso en los peores escenarios, el índice de transmisión de datos es 10 veces superior en comparación con una comunicación eléctrica sin interferencias.
- Libertad: Una red basada en la tecnología inalámbrica ofrece la completa libertad de incorporar sensores y luminarias. Proporciona una instalación optimizada para el mejor escenario de

regulación.

- Actualización: La red puede actualizarse añadiendo nuevas funciones. También puede ser fácilmente ampliada incorporando nuevos puntos de iluminación, independientemente del suministro eléctrico sin cableado, gracias a la comunicación LORA inalámbrica.
- Accesibilidad: El perfil de regulación puede cambiarse fácilmente conectando simplemente de forma inalámbrica un portátil a una luminaria sin usar ninguna herramienta. La nueva configuración será extendida para todas las luminarias de la red.

2.2 Lora

La tecnología Lora fue desarrollado por Semtech. Lora (Long Range) es una plataforma inalámbrica de baja potencia. Esta tecnología ofrece una mezcla muy completa entre largo alcance, baja potencia de consumo y seguridad en transmisión de datos. Redes publicas y privadas pueden usar esta tecnología para proveer un rango de mayor cobertura, en comparación con las redes de celulares existentes . Permite una fácil conexión a una infraestructura existente, demás ofrece una solución para aplicaciones que operan con servicio de batería.

Lora trabaja en la capa física o utiliza modulación inalámbrica para crear un enlace de largo alcance. Varios sistemas inalámbricos tradicionales emplean la modulación por desplazamiento de frecuencia (FSK, Frequency shifting keing) como enlace físico por que es una modulación muy eficiente dirigido a baja potencia. LoRa se basa en modulación del espectro del ancho de pulso, el cual mantiene las mismas características de baja potencia como la modulación FSK, pero incrementa significativamente el rango de comunicación. El espectro del ancho de pulso ha sido usado en comunicaciones militares y espaciales por décadas debido al rango de distancias de comunicación que pueden ser alcanzados además por su robustez a interferencias. LoRa es la primera implementación de bajo costo para uso comercial. La capacidad de enlace, frecuentemente dado en decibeles, es el factor principal para determinar el alcance de un medio dado. Por lo que con una mínima cantidad de infraestructura se puede realizar a cobertura de una área amplia.

La ventaja de LoRa es en la capacidad de largo alcance. Un radio base que trabaje con la tecnología LoRa puede cubrir una ciudad entera o cientos de kilómetros cuadrados. El rango de alcance depende del medio o las obstrucciones de la localización. Lora tiene una capacidad de enlace mucho mayor que cualquier otra tecnología de comunicación estandarizado.

Con el desarrollo de LoRa, se desarrolló un protocolo llamado LoRaWAN que especifica una manera de transmisión de datos que puede operar en esta tecnología, pero que puede ser sustituido por cualquier otro protocolo, que el desarrollador prefiera o cumpla con sus necesidades, siempre tomando en cuenta que la transmisión de datos se realiza en la capa física.

Lora siendo un tipo de modulación en radiofrecuencia patentado por Semtech, que en conjunto con

LoRaWAN ofrece estas principales ventajas:

- Alta tolerancia a las interferencias .
- Alta sensibilidad para recibir datos (-168dB).
- Geolocalización: Permite GPS-free para aplicaciones de seguimiento de baja potencia.
- Bajo costo: Reduce costos de tres maneras: inversión de infraestructura, costos de operación y sensores en nodos terminales.
- Estandarizado: Adopción de mejora global de interoperabilidad de velocidades y despliegue de redes LoRaWAN y aplicaciones IoT. ¹
- Baja potencia de consumo: Protocolo diseñado específicamente para bajo consumo de energía, extendiendo la vida de la batería hasta por 20 años.
- Largo alcance: Una simple estación provee una profunda penetración en un centro urbano denso, para áreas rurales puede alcanzar una cobertura de 30 millas.²
- Seguridad: Encorriación embebida end-to-end AES128.³
- Alta capacidad: Soporta millones de mensajes por estación, ideal para diversos clientes operadores de redes de servicios públicos.

- Frecuencias de trabajo: 915 Mhz América, 868 Mhz Europa, 433 Mhz Asia.

Todo esto hace a esta tecnología ideal para conexiones a grandes distancias y para redes de IoT, además que se puede utilizar en ciudades inteligentes o lugares con poca cobertura de telefonía móvil o redes privadas de sensores o actuadores, por eso es que nace LoRaWAN.

2.1.1 Protocolo LoRaWAN

LoRaWAN es un protocolo construido específicamente para la tecnología LoRa desarrollada por LoRa Alliance. Emplea un espectro de radio sin licencia en las bandas ISM (Industrial, Scientific and Medical) para permitir un bajo consumo de energía, comunicaciones de una área amplia entre sensores remotos y Puntos de Acceso conectados en una red. Este enfoque de base-estándar para construir una LPWAN (Low Power Wide Area Network) permite una rápida configuración de redes IoT públicas y privadas en cualquier lugar usando hardware y software que es bidireccionalmente seguro, interoperable y móvil, provee una localización precisa. Esta especificación se encuentra disponible para descargar libremente del sitio LoRa Alliance.

1 Estos dispositivos pueden trabajar en distintas frecuencias que no requieren licencia de distintas regiones del mundo, pero requiere de modificar la antena a una adecuada para las distintas frecuencias a la va a trabajar el dispositivo.

2 Este rango de alcance de esta tecnología son para las versiones de mayor potencia, a demás se considera una línea de vista directa.

3 Soportado por el protocolo LoRaWAN.

	Europa	Norteamérica
Banda de Frecuencia	867 -869 Mhz	902-928 Mhz
Canales	10	64+8+8
Ancho de banda /Uplink	125/250 kHz	125/500 kHz
Ancho de banda /Downlink	125 kHz	500 kHz
Potencia TX/Uplink	+14 dBm	+20 dBm
Potencia TX/Downlink	+14dBm	+27 dBm
SF up	7-12	7-10
Tasa de datos	250bps-50kbps	980 bps-21.9kbps

Tabla 2.1.: Frecuencias LoRaWAN para distintas áreas del globo.

2.1.2 Arquitectura de Red

Existen varios redes desarrollados que emplean una arquitectura de malla. Mientras este incremento de rango añade complejidad, reduce la capacidad de la red, y reduce la vida de la batería por la transmisión y recepción de datos de otros nodos que son irrelevantes para ellos. La arquitectura de una red de rango amplio en estrella hace que la mayoría de los sensores preserven la vida de la batería cuando se trabaja con una colectividad de rango amplio.

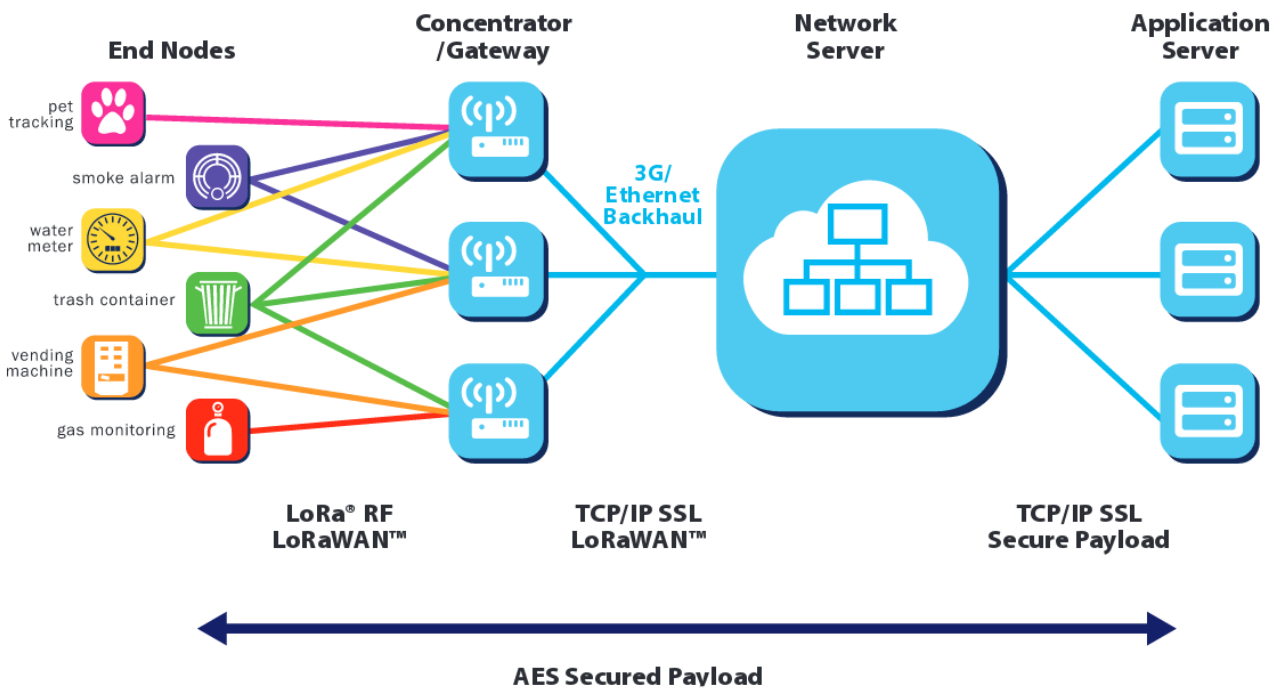


Figura 2.1: Arquitectura de red LoRaWAN.

2.3 Wi-Fi

Una red inalámbrica LAN (WLAN o WIFI) es un sistema diseñado para la transmisión de datos que provee acceso de red, independientemente de su localización, entre dos dispositivos de computo usando señales de radio en lugar de una infraestructura cableada.

En una empresa corporativa, las redes LAN inalámbricas son generalmente implementadas como enlace final entre una red cableada existente y un grupo de equipos cliente, dando a estos usuario acceso inalámbrico a todos los recursos y servicios de la red corporativa del edificio.

La aceptación generalizada de redes WAN depende de la estandarización industrial para asegurar la compatibilidad y fiabilidad de los productos entre los distintos fabricantes.

La especificación 802.11[IEEE std 802.11 (ISO /IEC 8802-11:1999)] como estándar para redes LAN fue ratificado por IEEE en 1997. Esta versión de 802.11 proporcionan una tasa de datos de 1 Mbps y 2 Mbps y un conjunto de métodos de señalización fundamentales y otros servicios.

La principal motivación y beneficio de las redes inalámbricas es el incremento de la movilidad, sin ataduras a las conexiones convencionales, los usuarios de la red pueden moverse casi a cualquier lugar sin restricción de acceso a la red de área local.

Otras ventajas para WLAN incluye el costo efectivo para ubicaciones difíciles de cableado. WLAN liberan a los usuarios de la dependencia de acceso cableado a la red troncal, permitiéndoles en cualquier momento y en cualquier lugar, acceso a la red. Esta libertad de movimiento ofrece numerosas ventajas al usuario para numerosos ambientes de trabajo como:

- Acceso remoto a la información de pacientes para médicos y personal de hospitales.
- Fácil acceso a la red en tiempo real para consultores y auditores.
- Acceso a la base de datos para supervisores, como gerente de línea de producción, auditores de almacén, o ingenieros de construcción.
- Configuración simplificada de red con una mínima participación de MIS para instalaciones temporales, como ferias o salas de conferencias.
- Rápido acceso a las información del cliente para los proveedores, resultando en un mejor servicio y mejora de satisfacción del cliente.
- Acceso independientemente de la ubicación para administradores de la red, para facilitar la solución de problemas.
- Acceso en tiempo real para reuniones de grupos de estudio y enlaces de investigación del estudiantes.

2.3.1 Capa Física

Originalmente se definen tres capas físicas en 802.11 incluido dos técnicas de radio de propagación de espectro. Los estándares basados en radio operan con una banda de 2.4 Ghz. Esta frecuencia es reconocida por la Agencia de Regulación Internacional para operaciones de radio (ITU Radio Regulations).

Las técnicas de propagación de espectro, aunado con el cumplimiento de los requerimientos regulatorios, de incremento de la fiabilidad, de alto rendimiento y la capacidad de varios equipos compartan el espectro sin una cooperación explícita y con mínima interferencia.

El estándar 802.11 de conexiones inalámbricas define tasa de datos de 1 Mbps y 2 Mbps a través de ondas

de radio usando propagación de espectro en intervalos de frecuencia (Frequency hopping spread spectrum, FHSS) o propagación de espectro de secuencia directa (Direct Sequence spread spectrum, DSSS). Estos anteriores su mecanismo de modulación son diferentes por lo que no se interoperan entre ellos.

Empleando una técnica de intervalos de frecuencia, la banda de 2.4 Ghz esta dividido en 75 subcanales de 1 Mhz. El transmisor y el receptor deben de estar de acuerdo con un patrón de intervalo de datos, que son enviados sobre una secuencias de subcanales.

2.3.2 Capa de enlace de datos

La capa de enlace de datos dentro de 802.11 consiste de dos subcapas: Enlace de control lógico (Logical Link Control, LLC) y Control de acceso de medios(Media Access Control, MAC). Este ultimo esta diseñado para soportar múltiples usuarios en un medio compartido. Con el objeto de conseguir que la información fluya, libre de errores.

Wifi es una protocolo que permite una tener una tasa de datos relativamente alta, pero el rango de cobertura es muy baja de unos cuantos metros por lo que para una aplicación de alumbrado publico se encuentra un poco limitado en cuanto al rango de cobertura.

2.4 Wimax

Wimax (Worldwide Interoperability for Microwaves Access) es una tecnología de inalámbrica de banda ancha, el cual soporta accesos inalámbricos de banda ancha punto a multipunto (PMP), que utiliza ondas de radio en las frecuencias de 2.5 a 5.8 Ghz.

Los estándares de conexiones inalámbricas de la IEEE tienen un rango de hasta 30 millas y pueden entregar un ancho de banda de alrededor de 75 Megabits por segundo.

WiMax puede ser usado para redes inalámbricas, como la popular WiFi. WiMax pertenece a la segunda generación de protocolos, que permite una relativa tasa de datos relativamente alta sobre largas distancias, el uso eficiente del ancho de banda y limitación de interferencia a un mínimo posible, en comparación con el protocolo WiFi que su alcance se mide en pies y trabaja sobre distancias cortas, hacen que este protocolo sea opción cuando sea requerido una tasa de datos relativamente alta para largas distancias.

En la actualidad WiMAX, sea ha visto desplazada por distintas tecnologías que han marcado el mercado ya sea por la tasa de datos o por el costo. El objetivo de descripción estas tecnologías es para identificar de cada una de ellas, las posibilidades que ofrecen para el desarrollo de un proyecto como la transmisión de datos para monitoreo y control de alumbrado publico. Ya que se desea realizar un uso eficiente del espectro radioeléctrico y debido a la baja demanda de esta tecnología en el mercado es un riesgo la inversión en esta tecnología que por así decirlo, ya que está a punto de desaparecer.

2.5 Evaluación de Sistemas inalámbricos de Comunicaciones

Es importante realizar una evaluación técnica entre las tres tecnologías. Para observar cual se adapta mejor al desarrollo del proyecto especifica a desarrollar:

	WiMAX	WIFI	LORA
Banda de Frecuencia	2.5 -5.8 Ghz	2.4 Ghz	902-928 Mhz
Canales	1	1	64+8+8
Ancho de banda /Uplink	20 Mhz	20-40 Mhz	125/500 kHz
Ancho de banda /Downlink	20 Mhz	20-40 Mhz	500 kHz
Alcance	2 – 50 Km	50 m	15 Km
Tasa de datos	54 Mbps	11, 54, 300 Mbps	980 bps-21.9kbps

Tabla 2.2: Tabla comparativa de las tres tecnologías WiMAX, WIFI y LoRa.

Se puede apreciar que Wimax y Wifi en cuanto a velocidad de transmisión son mucho mejor que LORA, en cuestión de rango de alcance Wimax es el mejor dentro de estas tres tecnologías, se puede observar que ancho de banda de wifi puede ser de 20 a 40 Mhz, para WiMAX 20 Mhz y para LORA 500 kHz.

En este punto se elegirá la tecnología a emplear para este proyecto, la mejor opción perceptible es WiMAX pero existe un gran problema con esta tecnología, si bien tiene las mejores cualidades, el declive que tiene esta tecnología motiva a analizar su aplicación, además que la instalación de la infraestructura es un poco costosa, principal factor de limitación para la implementación de esta tecnología.

Pasando a la tecnología Wifi, la transferencia de datos es muy alta y se podría enviar y recibir una enorme cantidad de datos, pero el principal factor de inconveniencia es el rango de alcance que tiene esta tecnología.

Finalmente LORA, que es relativamente la tecnología mas joven dentro de las tres, si bien la transferencia de datos es baja en comparación con las dos tecnologías anteriores, ya que nuestro proyecto no requiere de una transferencia de datos muy alta, el rango de alcance es relativamente amplio, y además el costo de los módulos de transmisión y recepción en comparación con las dos tecnologías anteriores es relativamente barato (alrededor de 20 dlls por modulo).

Siendo la tecnología LORA de alguna manera, de las tres tecnologías la que tiene prestaciones mas bajas, es la que mejor se adapta a las necesidades técnicos y económicos de este proyecto.

Capitulo 3 Alumbrado Publico

- 3.1 Análisis del alumbrado Público
- 3.2 Tendencias del Alumbrado Público
- 3.3 Caso de la Ciudad de México

3.1 Análisis del alumbrado Público

El alumbrado publico es una de las infraestructura de gran relevancia para la sociedad, ya que permite extender las actividades cotidianas de las misma sociedad durante las horas de la noche. Además que aumenta la percepción de seguridad, disminuye los accidentes de tránsito, permite el uso de espacios públicos de manera nocturna, mejora el uso de vialidades en horas sin luz solar. Siendo el alumbrado publico un servicio publico municipal en México, esto significa que el municipio queda a su cargo las instalación, operación y mantenimiento de esta infraestructura.

Estudios han mostrado que la oscuridad resulta en gran numero de choques y fatalidades, especialmente los que involucran peatones; las fatalidades de peatones son de 3 a 6.75 veces en plena oscuridad en comparación con luz de solar.

Una de las principales criticas hacia el alumbrado publico es por accidentes por mala operación y ser la causa de contaminación lumínica.

Los primero indicios sobre el alumbrado publico datan del siglo XVI. En Francia, los vecinos estaban obligados a colgar una luz en la puerta de sus casas (1524) no fue hasta 1558 cuando se colocaron faroles en las esquinas de las calles, a la que se organizo un cuerpo de vigilancia nocturna, encargados de encender y apagarlos. Para 1818 fue introducido el gas, para el uso en alumbrado publico, por medio de farolas de gas las cuales se encendían de forma manual, y posteriormente se introdujo el encendido automático de estas farolas.

Las farolas de gas fueron sustituidas por farolas eléctricas(1875), que en un principio la tecnología de arco eléctrico fue el mas empleado. Timisoara, en Rumania, fue la primera ciudad de Europa en contar con alumbrado publico por electricidad (12 de noviembre de 1884). Debido al gran desprendimiento de calor y el rápido desgaste de los electrodos de carbón, las lamparas de arco fueron sustituidas por por lamparas incandescentes para finales del siglo XIX.

Ya en el siglo XX se desarrolló la lampara de vapor de mercurio de alta presión, mejor conocidos como HID (High Intensity Discharge). El desarrollo de la lampara de vapor de sodio cuyo luz es de color ámbar, pero tiene un indice de rendimiento de color mayor, es un fuente puntual, tamaño y el tiempo de vida de esta lampara permitió un mejor diseño de luminarias.

Las lamparas de vapor de sodio actualmente son las lamparas con mayor eficiencia. Pero hoy día se enfrenta a la tecnología led, que promete una mayor eficiencia y un mayor tiempo de vida. En la mayoría de las luminarias el encendido se realiza local mente de forma automática mediante un celda fotoeléctrica o por

programación por reloj astronómico o de manera remota por sistemas de telegestión por medio de tecnologías GSM, de radiofrecuencia, etc.

Numerosos sistemas de control de alumbrado público han sido desarrollados para controlar y reducir el consumo de energía. El rango de control puede involucrar circuitos de alumbrado público o de forma individual, gracias a un sistema de comunicación. Esto puede incluir el envío y recepción de instrucciones ya sea alámbrica o inalámbricamente.

Estos controladores inteligentes pueden ser mecánicos o electrónicos previamente usados para la operación de encendido y apagado de las lámparas. Estos pueden ahorrar energía de distintas maneras entre ellos encendido-apagado, atenuación o dimerización. También varios de estos controladores pueden ser programados a partir de un reloj astronómico vía GPS (Global Positioning System) para un mejor control de horarios de encendido y ahorro de energía. Otra característica de estos sistemas de telegestión además de permitir la gestión de los circuitos de alumbrado público también se puede usar para propósitos de mantenimiento, lo que hace la infraestructura pueda ahorrar tiempo y recursos en gestionar la instalación.

Un sistema de alumbrado público requiere de un continuo mantenimiento, existe de dos tipos el tipo reactivo y preventivo. El mantenimiento reactivo como su nombre lo dice reacciona dependiendo del evento como por ejemplo fallo de lámparas, consiste en sustituir lámparas que han dejado de funcionar. El tiempo preventivo es un reemplazo programado de los componentes, esto se puede realizar en la mayoría de los equipos al momento de cumplir el 85% de su vida útil tienden a perder el índice de color, por lo que la calidad de la iluminación se ve disminuida.

3.2 Tendencias del Alumbrado Público

En la actualidad, para el alumbrado público se emplea con mucha regularidad lámparas de alta intensidad de descarga, por ejemplo las lámparas de sodio de alta presión (luz color ámbar). Sin embargo, una fuente de luz blanca se ha mostrado que permite una mejor visión periférica, además que reduce el tiempo de reacción en un 25%. estableciendo la importancia del alumbrado público. Se continúa describiendo las distintas tecnologías que se disputan el mercado del alumbrado público y sus opciones.

A partir de la entrada de luminarios con tecnología de diodos emisores de luz (LED), se une a un mercado que disputa principalmente tres tecnologías lámparas de inducción de electromagnética, lámparas de aditivos metálicos Cerámicos y lámparas de plasma, la tecnología de vapor de sodio de alta presión debido a su índice de color esta en desuso en el caso de México.

Se ha estimado que los luminarios para alumbrado público con módulo de Diodos Emisores de luz (LEDs), y lámparas de aditivos metálicos cerámicos tendrán un predominio absoluto durante los próximos años, se estima que después de varias décadas solo se considerará el instalar únicamente luminarios para alumbrado público módulos de diodos emisores.

Los sistemas de telegestión conforman también esta tendencia tecnológica, actualmente no existe norma que regule, la construcción o tipo de tecnología de comunicación que se pueda emplear para estos

sistemas, por ello distintos fabricantes han desarrollado sus propias versiones a partir de la tecnología disponible.

La factibilidad de estos que ofrecen estos sistemas de telegestión, son un buen argumento para su implementación, resolviendo alguno de los problemas que más a queja hoy en día a los sistemas de alumbrado público, entre sus características que ofrecen son las siguientes:

- Seguimiento de los consumos en tiempo real y programación del funcionamiento del alumbrado público.
 - Análisis de consumos por día, mes y año, tanto de variables eléctricas como económicas.
 - Verificación de mediciones de ahorro comparativas.
 - Gestión de horarios de encendido y apagado.
- Recepción de alarmas por fallas y pérdidas de servicio.
 - Gestión de alarmas a nivel de cuadro y por fase.
 - Configuración personalizada para activación y des-activación de alarmas.
 - Posibilidad de integración con otros sistemas propios del cliente.
- Gestión de activos con posicionamiento geográfico y los parámetros clave.
 - Inventario de cuadros y luminarias
 - Equipamiento existente
 - Informes de consumo, ahorro, nivel de servicio, inventario, fallas.
- Informes de consumo, ahorro, nivel de servicio, inventario y facturación.
 - Informes de consumo, ahorro, análisis de datos y variables eléctricas
 - cálculo de factura eléctrica
 - informes diarios, mensuales y anuales.

3.3 Caso de la Ciudad de México

La Ciudad de México, ha sido siempre la ciudad más importante de México, y las primeras integraciones de tecnologías para sistemas de alumbrado público, siempre ha sido de las primeras opciones. En la década de 1950, se utilizaban las primeras lámparas con mayor eficacia, mayor flujo luminoso, mayor vida útil y que operaban bajo el principio de descarga eléctrica de alta intensidad con gas de alta presión que incorporaban de forma horizontal los luminarios para alumbrado público vial, que incluían conjuntos ópticos (reflector y un refractor). Iniciando así la implementación para el alumbrado público un largo recorrido tecnológico en la evolución de las distintas tecnologías de lámparas y luminarios que se han utilizado.

Actualmente se están incorporando de forma gradual algunas tecnologías de lámparas con modernos diseños de luminarios para alumbrado público vial, esto con el objeto de reducir el consumo de energía eléctrica y mayor eficiencia energética, tratando de mantener los niveles de iluminación. A partir de esta década la tecnología empleada en México y Estados Unidos para el alumbrado público ha sido prácticamente la misma, pero la difusión de las tecnologías más modernas siempre han iniciado en Estados Unidos.

Anterior a la década de 1950 en la Ciudad de México, las lámparas incandescentes en posición vertical con una potencia de 750 W eran el de mayor uso. Pero para 1950 fueron sustituidas por una tecnología más eficiente basada en descarga eléctrica de alta intensidad y sobre todo instaladas de forma horizontal. Fue en esta década donde se instalaron los primeros luminarios para alumbrado público vial totalmente autobalastados y fabricados en fundición de aluminio inyectada en alta presión, sus características de

diseño y construcción incorporaban un conjunto óptico integrado por un reflector de aluminio y un refractor ovalado, en cuyo interior se operaba mediante un balastro electromagnético y en posición horizontal una lámpara de Vapor de Mercurio sin corrección de color (VmsC) con una potencia de 400 W, la cual generaba un flujo luminoso inicialmente de color blanco frío, con una temperatura de color de 4000 K y una vida útil de 24,000 horas. En México, para 1958 se inicio la incorporación de este tipo de luminarios para el alumbrado público, que terminó de 1968.



Figura 3.1 Alumbrado público operando lámparas de Vapor de Mercurio (VM) sin color corregido.

Para 1964 en México se da inicio a la incorporación de luminarios para alumbrado público vial con lámparas de Vapor de Mercurio con corrección (VMcC) de color con una potencia de 400 W, para sustituir a los anteriores luminarios para alumbrado público vial que operaban lámparas de Vapor de Mercurio sin corrección de color con una potencia de 400 W.



Figura 3.2 Alumbrado público operando lámparas de Vapor de Mercurio (VM) con color corregido.

En nuestro país la instalación de luminarios para alumbrado público vial con lámparas de Vapor de Mercurio con corrección de color con una potencia de 400 W, se dio de 1964 hasta 1982 en las principales ciudades de México.

Para la década de 1970 apareció la lámpara de vapor de sodio en baja presión (VSBP), que en transcurso de esta década sustituyó a la tecnología de Vapor de Mercurio de Alta presión. Las lámparas de Vapor de Sodio en Baja Presión con una potencia de 180 W fueron las de mayor empleo, esto con la finalidad de reducir el consumo de energía eléctrica respecto a las lámparas de Vapor de Sodio de Alta Presión con una potencia de 400 W. Su instalación se centró en áreas donde existían altas concentraciones de neblina o precipitaciones pluviales, en las cuales por sus características monocromáticas de la luz naranja-rojiza que emitían, con una temperatura de color de 1800 K y una vida útil de 18,000 horas, eran las adecuadas; sin embargo por sus limitadas características operacionales no desplazó completamente a la tecnología de Vapor de Mercurio de alta presión.

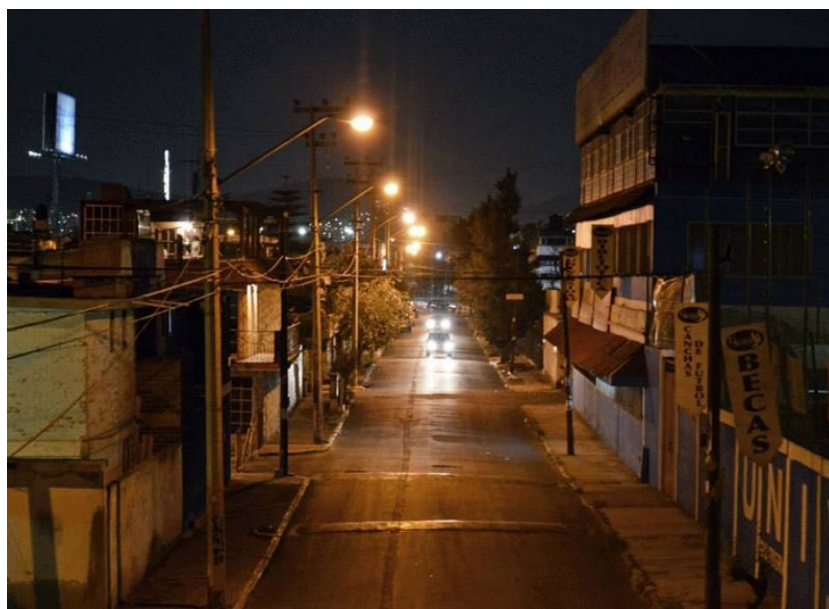


Figura 3.3 Alumbrado público operando lámparas de Vapor de Sodio en Alta Presión (VSAP).

En el año de 1984 se instalan luminarios para alumbrado público vial totalmente autobalastados y fabricados en fundición de aluminio inyectada en alta presión, los cuales por sus características de diseño y construcción incorporan un conjunto óptico independiente integrado por un reflector de aluminio y un refractor ovalado de vidrio prismático o un refractor plano de vidrio claro termotemplado en cuyo interior se opera mediante un balastro electromagnético y en posición horizontal una lámpara de Vapor de Sodio en Alta Presión (VSAP) con una potencia de 250, 150 o 100 W, las cuales generan un flujo luminoso monocromático de color amarillo, con una temperatura de color de 2100 K y una vida útil de 24,000 horas; en las principales ciudades de México fueron instalados de 1984 hasta 2001, aunque hasta la fecha aún continúan operando, a la espera de ser sustituidos.

En la siguiente década de 1990 en la Ciudad de México se da inicio a la incorporación de los luminarios para alumbrado público vial con lámparas de Aditivos Metálicos (AM) con una potencia de 250 o 175 W, que antes se habían instalado en los Estados Unidos, para sustituir a los anteriores luminarios para alumbrado público vial que operaban lámparas de Vapor de Sodio en Alta Presión con una potencia de 250,150 o 100 W. En nuestro país la instalación de luminarios para alumbrado público vial con lámparas de Aditivos



Figura 3.4 Alumbrado público operando lámparas de Vapor de Sodio en Baja Presión (VSBP).

Metálicos con una potencia de 250 o 175 W en las principales ciudades de México, fue desde 1996 hasta 2007, aunque hasta la fecha aún continúan operando.

Al finalizar la primera década del milenio en nuestro país se da inicio a la instalación de luminarios para alumbrado público vial con módulos de Diodos Emisores de Luz (LEDs) con una potencia total de 150,120, 100 u 80 W, para sustituir a los anteriores luminarios para alumbrado público vial que aun operan lámparas



Figura 3.5 Alumbrado público operando lámparas de Aditivos Metálicos (AM).

de Vapor de Sodio en Alta Presión (VSAP) con una potencia de 250,150 o 100W, así como de Aditivos Metálicos con una potencia de 250 o 175 W.



Figura 3.6 Alumbrado público operando módulos con Diodos Emisores de Luz (LEDs).

Durante 2008 en nuestro país se da inicio a la incorporación de los mismos luminarios para alumbrado público vial con lámparas tubulares circulares o rectangulares de Inducción Electromagnética con una potencia de 150, 120, 100 u 80 W, que antes se habían instalado en los Estados Unidos, para sustituir a los anteriores luminarios para alumbrado público vial que aun operan lámparas de Vapor de Sodio en Alta Presión con una potencia de 250, 150 o 100W, así como de Aditivos Metálicos (AM) con una potencia de 250 o 175 W. En nuestro país en algunas ciudades se da inicio a la instalación de luminarios para alumbrado público vial con lámparas tubulares circulares o rectangulares de Inducción Electromagnética con una potencia de 150, 120, 100 u 80 W.



Figura 3.7 Alumbrado público operando lámparas de inducción electromagnética (IND).

Para 2009 en nuestro país se da inicio la incorporación de luminarios para alumbrado público vial con lámparas tubulares de Aditivos Metálicos Cerámicos con una potencia de 140 o 90 W, que antes se habían instalado en los Estados Unidos, para sustituir a los anteriores luminarios para alumbrado público vial que aun operan lámparas de Vapor de Sodio en Alta Presión (VSAP) en versiones de potencia de 250,150 o 100W, así como de Aditivos Metálicos con una potencia de 250 o 175 W.

En nuestro país en algunas ciudades se da inicio a la instalación de luminarios para alumbrado público vial con lámparas tubulares de Aditivos Metálicos Cerámicos (AMC) con una potencia de 140 o 90 W.



Figura 3.8 Alumbrado público operando lámparas Aditivos Metálicos Cerámicos (AMC).

Actualmente en la segunda década dominan el mercado 4 nuevas opciones tecnológicas, que son las lámparas con módulos de Diodos Emisores de Luz (LEDs), Inducción Electromagnética, Aditivos Metálicos Cerámicos y Plasma . El predominio en México de acuerdo con la cantidad actual instalada y por instalar de luminarios para el alumbrado público vial en sus principales ciudades, predominan de acuerdo con el siguiente orden, los que operan con tecnología de Aditivos Metálicos Cerámicos, módulos de Leda y de Inducción Electromagnética .

Si bien los luminarios para alumbrado público vial con módulos de Diodos Emisores de Luz (LEDs) y lámparas de Aditivos Metálicos Cerámicos (AMC) tendrán un predominio absoluto durante los próximos años, se estima que después de varias décadas, solo se considerara el instalar en todas sus ciudades, únicamente luminarios para alumbrado público vial con módulos de Diodos Emisores (LEDs).

En las siguientes décadas por transcurrir, los luminarios para alumbrado público vial con módulos de Diodos Emisores de Luz , sin duda alguna predominarán sobre las lámparas de Inducción Electromagnética (IND) y las lámparas de Plasma (PLA) .Ya que inicialmente en su momento y de una forma determinante, predominaron sobre las lámparas de descarga eléctrica de alta intensidad a través de un gas en alta presión.

En la tabla 1 se puede observar un resumen por las tecnologías que se han empleado en México desde 1950, además de que se puede también ver algunas tendencias de este sector en particular, por ejemplo en

el color se puede ver que se evolución de un color verdoso a uno blanco, esta característica se puede comprobar también con índice de rendimiento de color. Por otro lado se ha incrementado el tiempo de vida de la lamparas, ademas que la potencia de consumo se ha visto drásticamente reducido desde la primer tecnología expuesta hasta la ultima en a un 37.5%. estas observaciones se va a observar mejor en las siguientes gráficas.

Periodo de instalación	Tecnológica	Potencia [Watts]	Indice Ten Color [K]	Tiempo de Vida [hrs]	Abreviatura
1958-1965	Lampara de vapor de Mercurio sin corrección de color	400	4000	24000	VMsC
1964-1982	Lampara de vapor de Mercurio con corrección de color	400	4000	24000	VMcC
1973-1985	Lampara de Vapor de sodio de alta presión	400	2100	24000	VSAP
1979-1984	Lampara de Vapor de sodio de baja presión	180	1800	18000	VSBP
1980-2001	Lampara de Vapor de sodio de alta presión auto balastrado	250,150,100	2100	24000	VSAP-auto
1996-2007	Lampara de Aditivos metálicos	250-175	4000	20000	AM
2008-actual	Lampara de inducción electromagnética	150, 120, 100 80	5000	80000-100000	IND
2009-actual	Lampara tubular de aditivos metalices ceramicos	140, 90	2800	30000	AMC
2010-actual	Lampara con diodos Emisores de Luz.	150, 120, 100, 80	6000	50000	LED

Tabla 3.1 Resumen de la evolución de distintas tecnologías de alumbrado publico

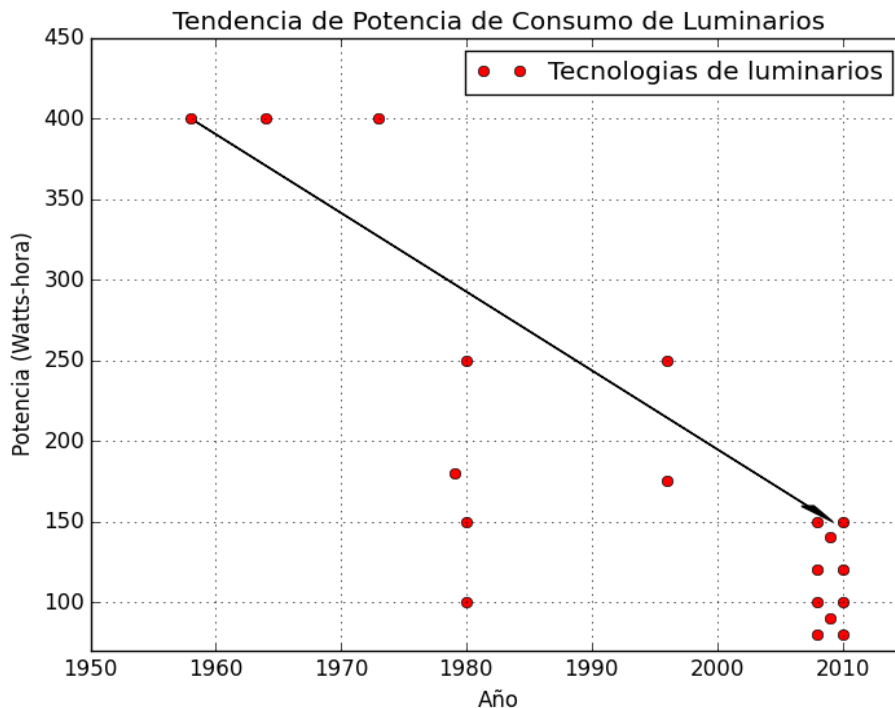


Figura 3.9 Tendencia de Consumo Energético de las diferentes Tecnologías de luminarios.

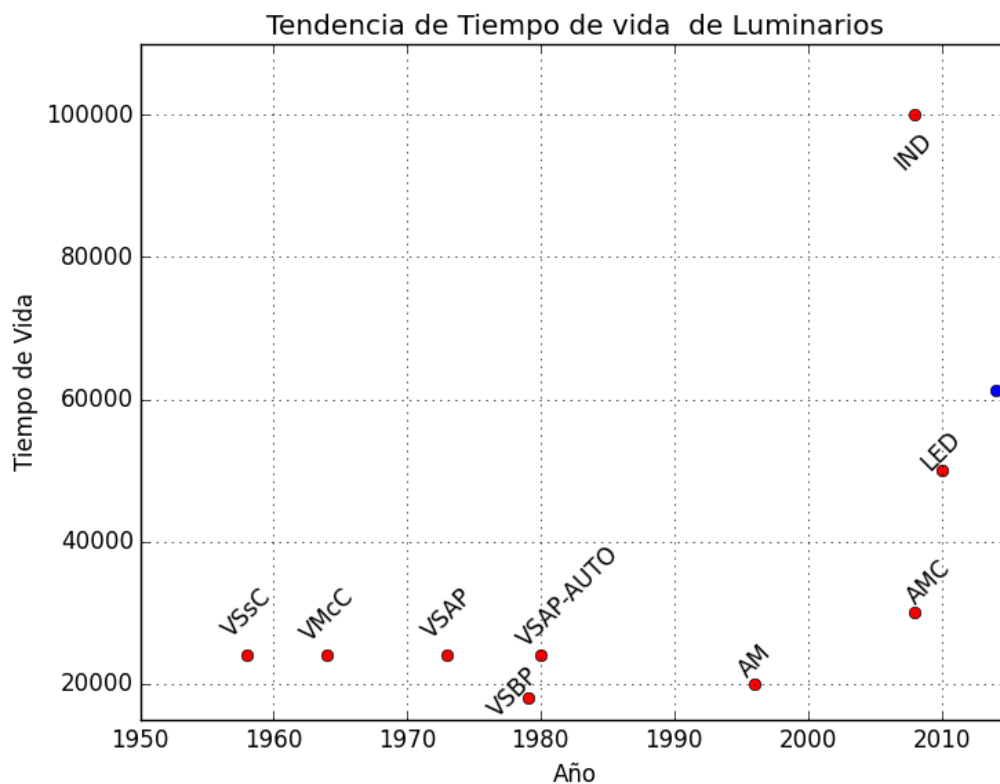


Figura 3.10 Tendencia de Tiempo de Vida de Distintas Tecnologías de Luminarios

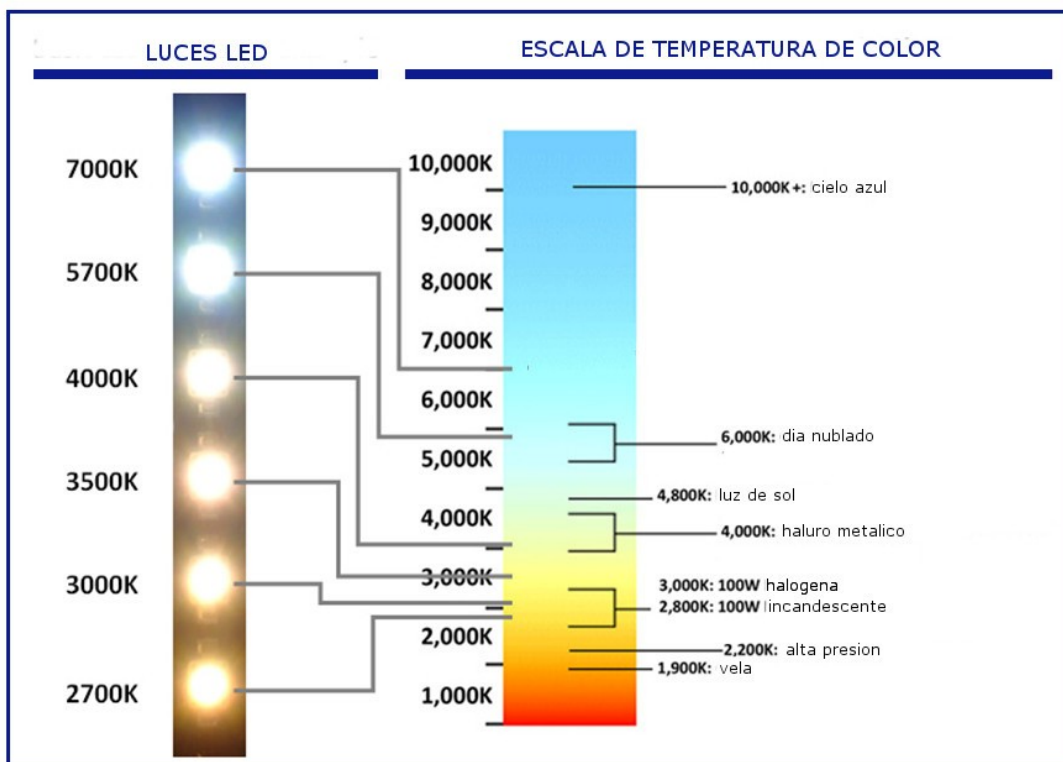


Figura 3.11 Indice de Temperatura de Color

Capítulo 4 Diseño del sistema

- 4.1 Estructura del Sistema de Administración Inteligente
- 4.2 Diseño de la estación de medición y control
 - 4.2.1 Funciones de medición y control
 - 4.2.1.1 Medición de voltaje
 - 4.2.1.2 Medición de corriente
 - 4.2.1.3 Sensor de Luz
 - 4.2.1.4 Recepción de datos e Interpretación de comandos
 - 4.2.1.5 Registro de eventos
 - 4.2.1.6 Transmisión de datos
 - 4.3 Diseño de la Estación Central de Control
 - 4.3.1 Procedimiento de Administración Inteligente
 - 4.3.2 Sistema de Comunicación
 - 4.3.3 Procedimiento de control
 - 4.4 Diseño de Interfaz gráfico
 - 4.4.1 Administración de base de datos
 - 4.4.2 Arquitectura Modelo- Vista-Controlador

4.1 Estructura del Sistema de Administración Inteligente

En el capítulo 1, en su sección de planteamiento del proyecto se describe de forma breve este proyecto, retomaremos nuevamente esta descripción para realizar una descripción más profunda.

Este proyecto consta de tres principales secciones que son las siguientes y se pueden observar en la Figura 1:

- I. Modulo Estación
- II. Modulo Central
- III. Interfaz Gráfica de Gestión

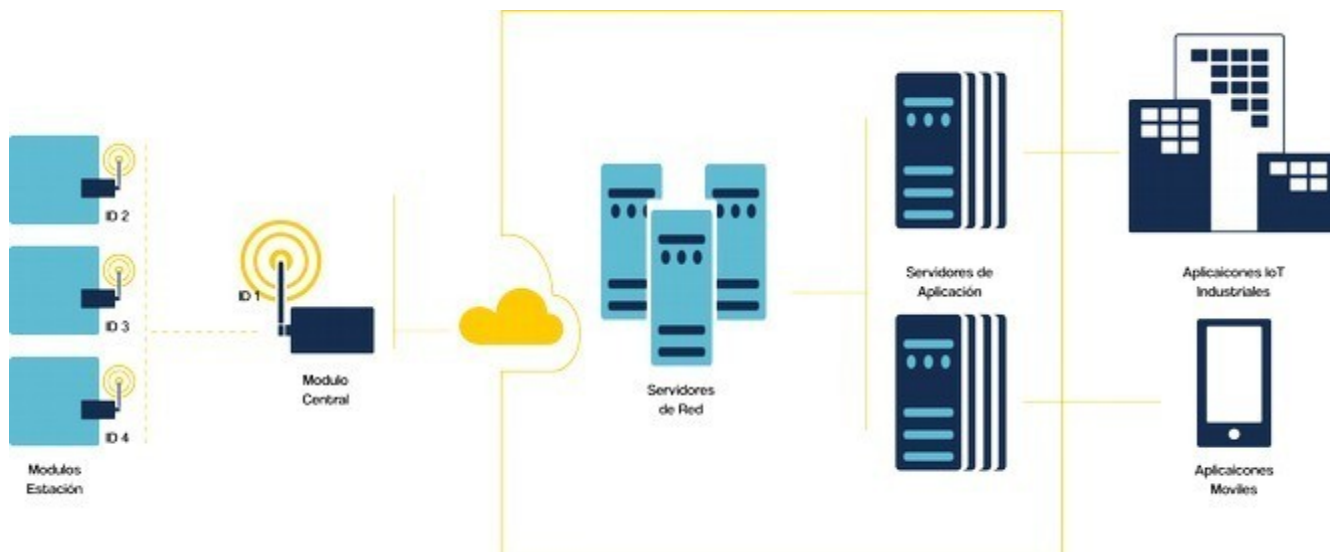


Figura 4.1 Estructura de Sistema de Administración Inteligente.

Esta división es debido a que cada una cumplen con funciones específicas que permiten que el sistema opere de manera correcta. Estas funciones específicas, se describen en la figura 2.

Las dos primera secciones son módulos que están compuestos por una parte física y otra por software. La ultima sección, este servidor puede estar montado en un servidor dentro de una red local o puede ser un servidor ubicado en la nube.

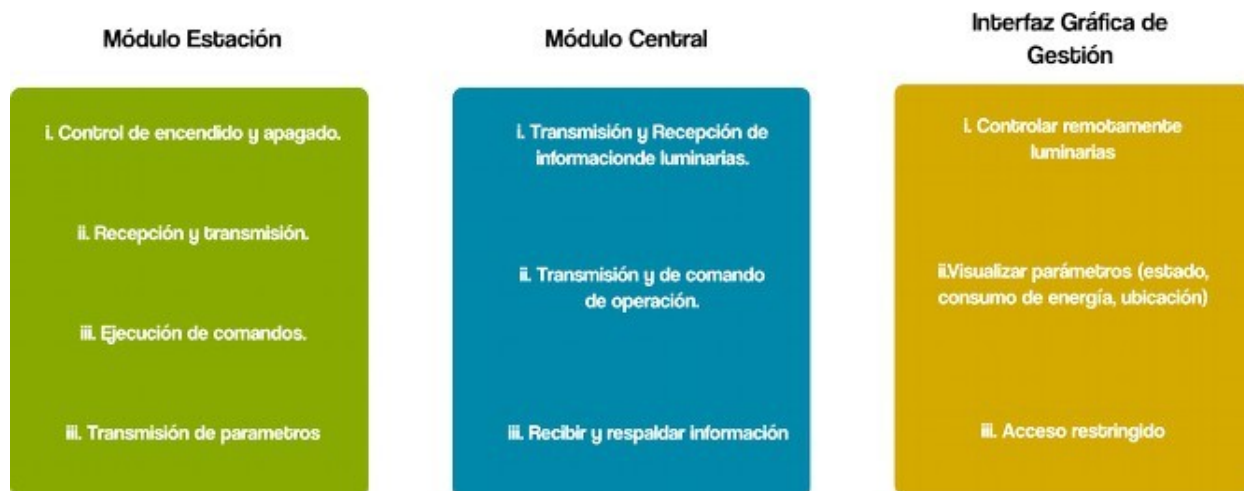


Figura 4.2 Objetivos de las Secciones del Proyecto.

4.2 Diseño de la estación de medición y control

En la estructura de diseño lora, también son conocidos como nodos, ya que son puntos de conexión final para aplicaciones de tipo IoT (Internet of things/ Internet de las cosas). Para este trabajo estos Módulos Estación se les nombrará como módulos Estación o módulos de Aplicación.

Estos módulos de Estación su desarrollo se dividen en dos principales secciones:

- Software: que consta en el diseño y desarrollo de programación para adquirir todos los parámetros que se necesitan en este proyecto (Corriente, voltaje, cantidad de luz, eventos), controlar el encendido y apagado las lamparas, así como transmitir y recibir información a través del modulo Lora, e interpretar los comandos enviados por el modulo central.
- Hardware: esta sección contemple la implementación de todos los módulos para sensar y modulo de transmisión y recepción de datos, ademas el desarrollo del modulo de potencia para controlar las lamparas.



Figura 4.3 Resumen de integración de Módulo Estación.

4.2.1 Funciones de medición y control

Las funciones de medición y control, describe como funciona el algoritmo para la medición de los parámetros, para que finalmente formar las base para desarrollar el software que realizará esta función. Para ello de las funciones que deben se deben incorporar son:

- Medición de voltaje
- Medición de corriente
- Sensor de Luz
- Transmisión de datos
- Recepción de datos
- Interpretación de comandos
- Registro de eventos

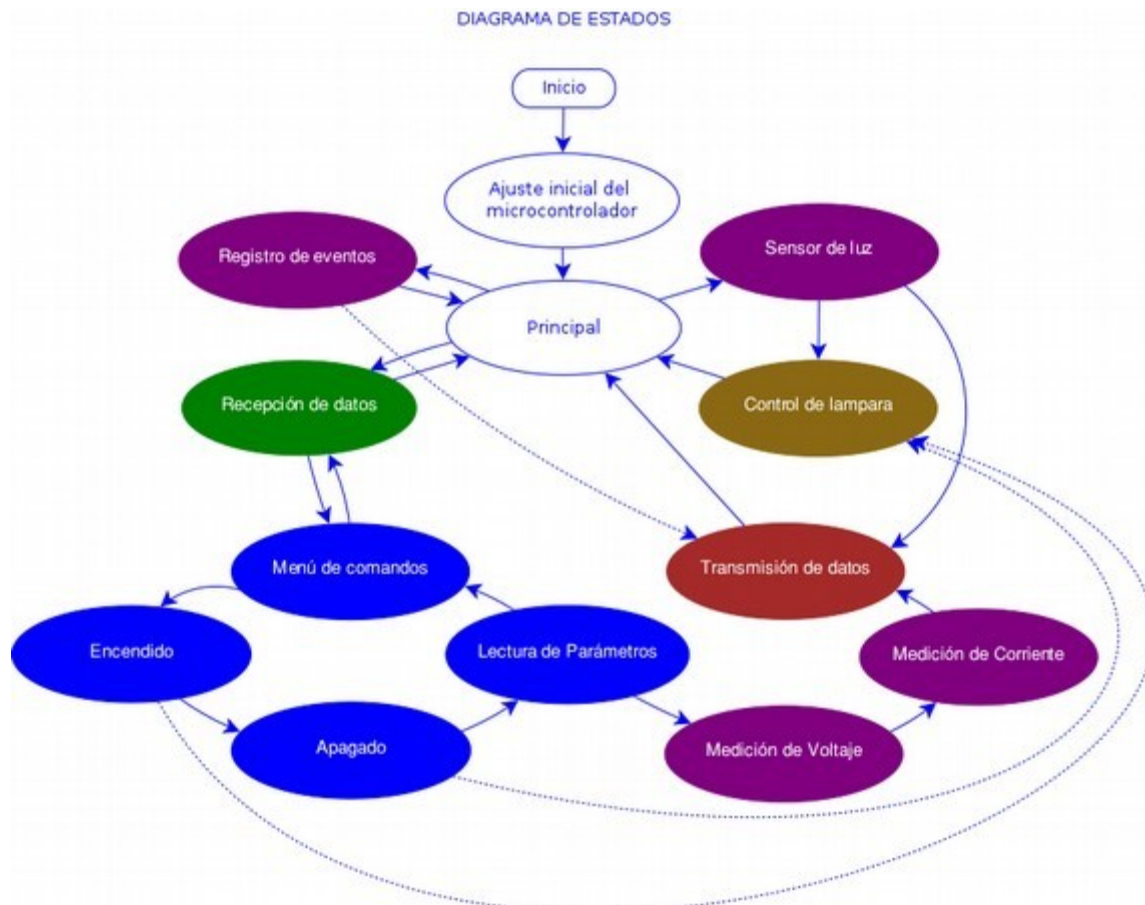


Figura 4.4 Resumen de diagramas de estado de Módulo Estación.

Los métodos para obtener mediciones pueden ser variados, dependiendo de la tecnología del dispositivo utilizado. El proceso de adquisición de datos, para este proyecto se realiza mediante, conversión analógica a

digital, que consiste en tomar una señal analógica y compararla con un valor de referencia fijo y conocido, en base a esto se obtiene un valor numérico el cual podría ser visto como un conjunto de unos y ceros o mejor dicho un valor digital. En este caso se monitoreará 4 principales parámetros voltaje, corriente de operación de la luminaria, el parámetro que mide el nivel de luz del ambiente y por ultimo determinar si la luminaria esta encendida o apagada.

4.2.1.1 Medición de voltaje

La medición de voltaje es un parámetro importante ya que permite conocer si existe suministro eléctrico para la operación de la luminaria.

Para este diseño no se necesita graficar esta señal, y obtener lo valores de la onda, solo se requiere el valor rms del voltaje. Por ello se tomarán una cierta cantidad de muestras, en un tiempo determinado (500 ms) y se realizará un promedio a partir de la datos obtenidos en ese periodo.

El modulo estación, esta basado en arduino uno, y como primer paso se reduce el nivel tensión de 120 a 12 volts mediante un transformador y como segundo paso se pasa esta tensión a un divisor del tensión, que se conecta con una de las entradas analógicas del microcontrolador y mediante software realiza la estimación del valor del voltaje. Para ello se utiliza el modulo FZ0435 que consta de dos resistencia en serie de 30kΩ y 7.5 kΩ para un comodidad de uso, no obstante este divisor de tensión se puede sustituir por uno integrado a la placa base. El circuito de medición de voltaje se realiza en la figura 6.

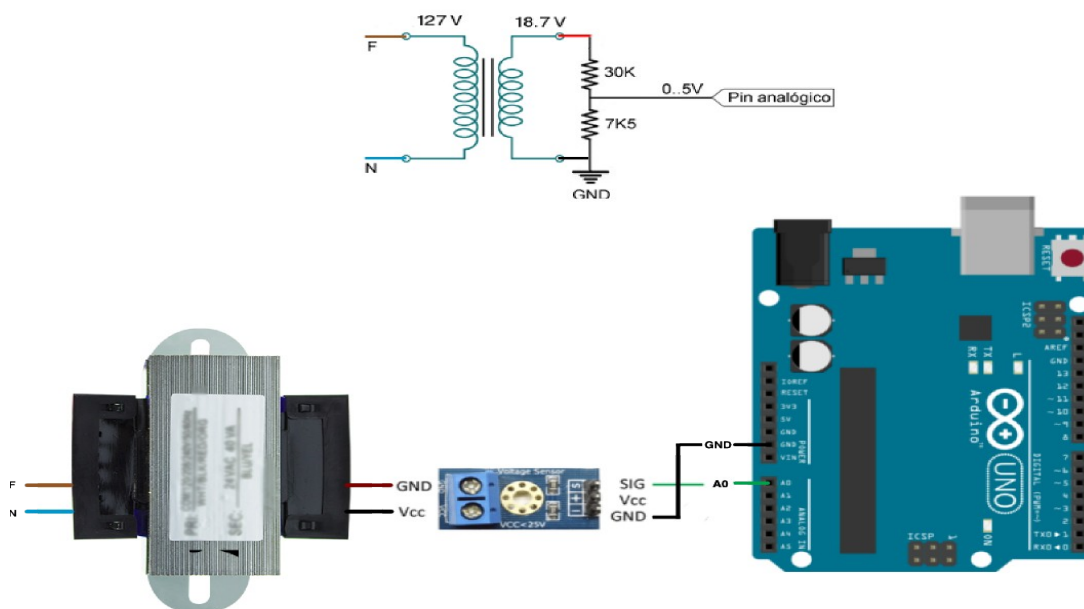


Figura 4.5 Diagrama de conexión eléctrica del circuito de medición de voltaje.

Para realizar una adecuada conversión de unidades se tiene que considerar lo siguiente, en el primer caso se tiene que considerar la relación de transformación del transformador que se esta empleado en este caso, las

mediciones que se realizo al trasformador reductor se muestran en la tabla 1.

	Medición 1	Medición 2	Medición3	Promedio
Voltaje Primario	122.7	122.6	123.4	122.9
Voltaje Secundario	13.56	13.66	13.64	13.62
Relación	9.0486	8.9751	9.0469	9.0235

Tabla 4.1 Mediciones de voltaje primario y voltaje secundario de transformador reductor.

Seguido ya que existe un divisor de voltaje, se tiene las siguiente consideración. Tomanos el voltaje promedio secundario del transformador seguido se determinada la caída de voltaje en cada una de las resistencia para determinar la relación en el divisor de voltaje.

$$v_{eq} = v_1 + v_2$$

$$v_{eq} = ir_1 + ir_2$$

$$i = \frac{v_{eq}}{r_1 + r_2} = \frac{13.62 \text{ volts}}{30.0 \text{ k}\Omega + 7.5 \text{ k}\Omega} = \frac{13.62 \text{ volts}}{37.5 \text{ k}\Omega} = 0.3632 \text{ miliamperes}$$

$$v_1 = ir_1 = (0.3632 \text{ miliamperes})(30 \text{ k}\Omega) = 10.8959 \text{ volts}$$

$$v_2 = ir_2 = (0.3632 \text{ miliamperes})(7.5 \text{ k}\Omega) = 2.7239 \text{ volts}$$

$$relacion_{divisor} = 2.7239 \text{ volts} / 13.62 \text{ volts} = 5.0001$$

Por lo tanto el ajuste que se tiene que hacer con el valor de medición obtenida para obtener la relación adecuada en un primer paso multiplicar por 5.0001 y posterior por 9.0235. Después que se tiene la señal, que emula el nivel de voltaje a una entrada analógica del microcontrolador, se continua mediante software el calculo de este siguiendo el diagrama 1.

Hay que tomar en cuenta que esta función se ejecuta solo cuando se le solicita, y no se ejecuta de forma automática, se realiza cuando hay una petición de lectura de parámetros, se emplea para llenar los parámetros requeridos para la trama de información a enviar.

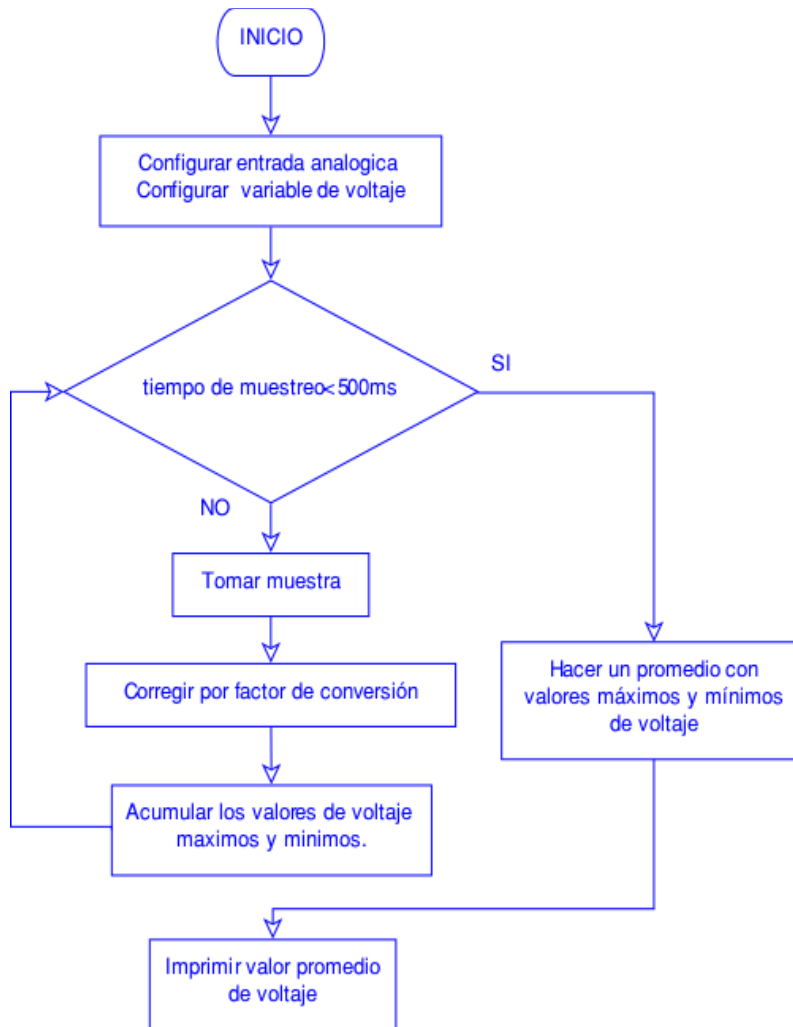


Figura 4.6 Diagrama de flujo de medición de voltaje.

4.2.1.2 Medición de corriente

La medición de corriente, permitirá conocer el consumo que la luminaria tienen durante el tiempo que este encendido, este valor de corriente, al igual que el parámetro de voltaje, se mide por un determinado tiempo (500 ms), pero a diferencia del nivel de voltaje, este va almacenando los valores de corriente medidos, solo los valores de mas altos y los valores mas bajos. Donde al termino de este periodo se realiza un promedio del valor mayor y menor.

El modulo estación, esta basado en arduino uno, y para la medición de corriente se emplea un modulo para arduino para medir corriente (ACS712), este modulo funciona a partir del principio de efecto Hall, lo que lleva a convertir corriente a un cierto nivel de voltaje. El circuito de medición de corriente es como se muestra en el la figura 6.

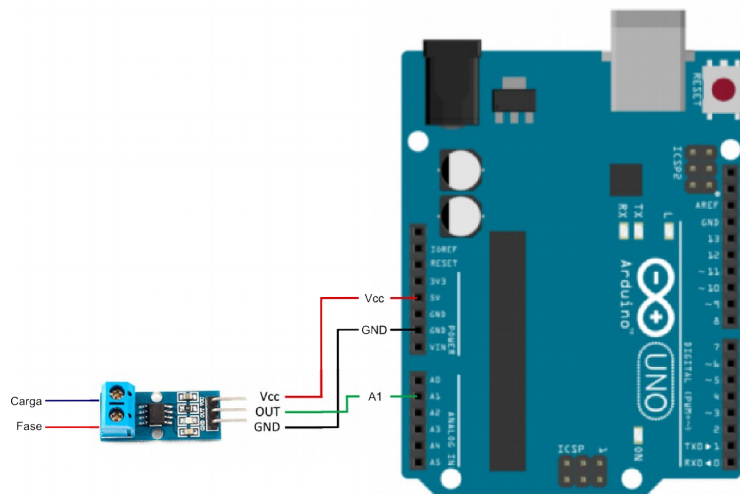


Figura 4.7 Diagrama de Conexión Eléctrica del circuito de Medición de corriente.

Para obtener el valor de corriente mediante software se sigue el diagrama de flujo que se muestra a continuación.

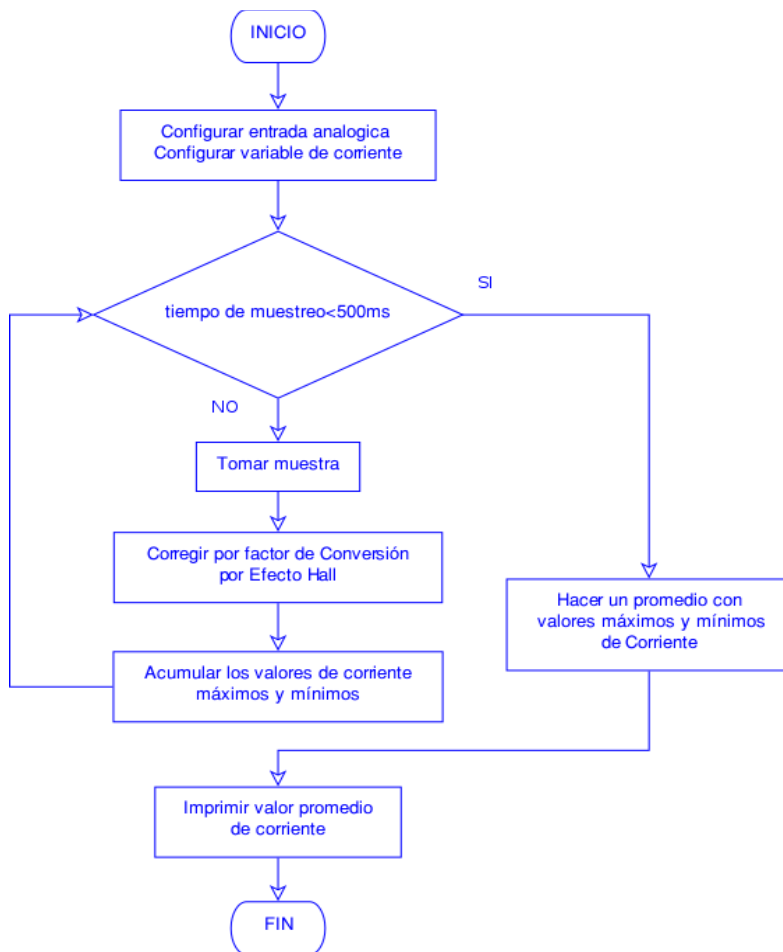


Figura 4.8 Diagrama de de flujo de Medición de Corriente

4.2.1.3 Sensor de Luz

Para medir el nivel de luz en el ambiente se emplea una fotoresistencia, cuya resistencia varia en función de la luz recibida, referida a $2M\Omega$. Para obtener la señal que determina el nivel de luz en el ambiente esta fotoresistencia se conecta a través de un divisor de voltaje. Esta señal se puede usar para medir el nivel de luz a través de una de las entradas analógicas del microcontrolador , en el siguiente diagrama se muestra como se encuentra la conexión eléctrica para adquirir esta señal.

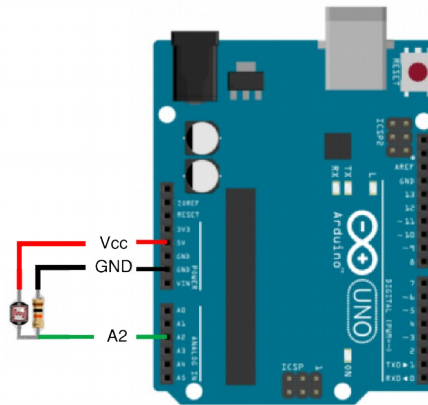


Figura 4.9 Diagrama de Conexión Eléctrica de fotoresistencia.

Es bien conocido que el tiempo de respuesta de una fotoresistencia es limitada, pero es adecuada para este tipo de aplicaciones que no se requiere de mucha precisión, por esta misma razón el juste de umbral para el encendido y apagado de la lampara se debe realizar manualmente conforme al comportamiento de la fotoresistencia.

Para determinar el nivel de luz del ambiente, ya que se sabe que los valores que se pueden adquirir esta señal van de 0-1024 , a prueba y error se determina el umbral de la cantidad de luz adecuada a la que la luminaria debe operar. Se debe considerar el comportamiento de la fotoresistencia que menor nivel de luminosidad mayor resistencia, por lo que mayor caída de voltaje en la fotoresistencia, y mayor el valor de la señal. A mayor cantidad de nivel de luminosidad, menor resistencia, por lo que menor caída de voltaje, y menor el valor de la señal. Considerando este comportamiento se obtuvo que el valor optimo de umbral es de 200.

En el diagrama 3 se describe como se estima el nivel de luz solar. Y como es que habilita el encendido o apagado de la luminaria.

Programa Principal de control de Luminaria

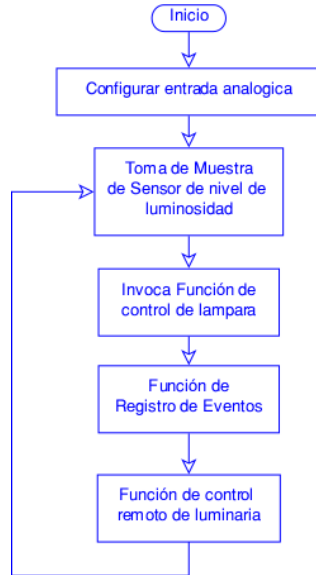


Figura 4.10 Diagrama de Flujo de Medición de nivel de luminosidad.

Función de Control de Lampara

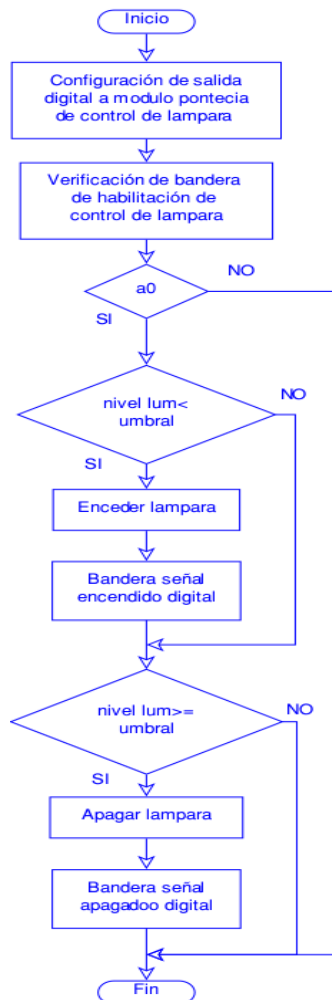


Figura 4.11 Diagrama de de flujo de control de encendido y apagado de lampara.

4.2.1.4 Recepción de datos e Interpretación de comandos

Para la recepción de datos, se considera en dos secciones principales, la primera sección es considerar la antena, la propia antena requiere de una configuración propia para ser aplicable para un modulo estación con parámetros muy específicos. Esta configuración es descrita en el anexo B en la sección de configuración de antena DRF1276DM.

Siendo que la antena haya sido configurado adecuadamente para modulo estación, se continua con la conexión al microcontrolador, el valor de voltaje de operación del microcontrolador es de 5 Volts, y el voltaje de operación de la antena es especificado en 3 volts, por ello se requiere un convertidor de niveles lógicos, para una adecuada operación eléctrica, La siguiente figura describe la conexión eléctrica de la antena.

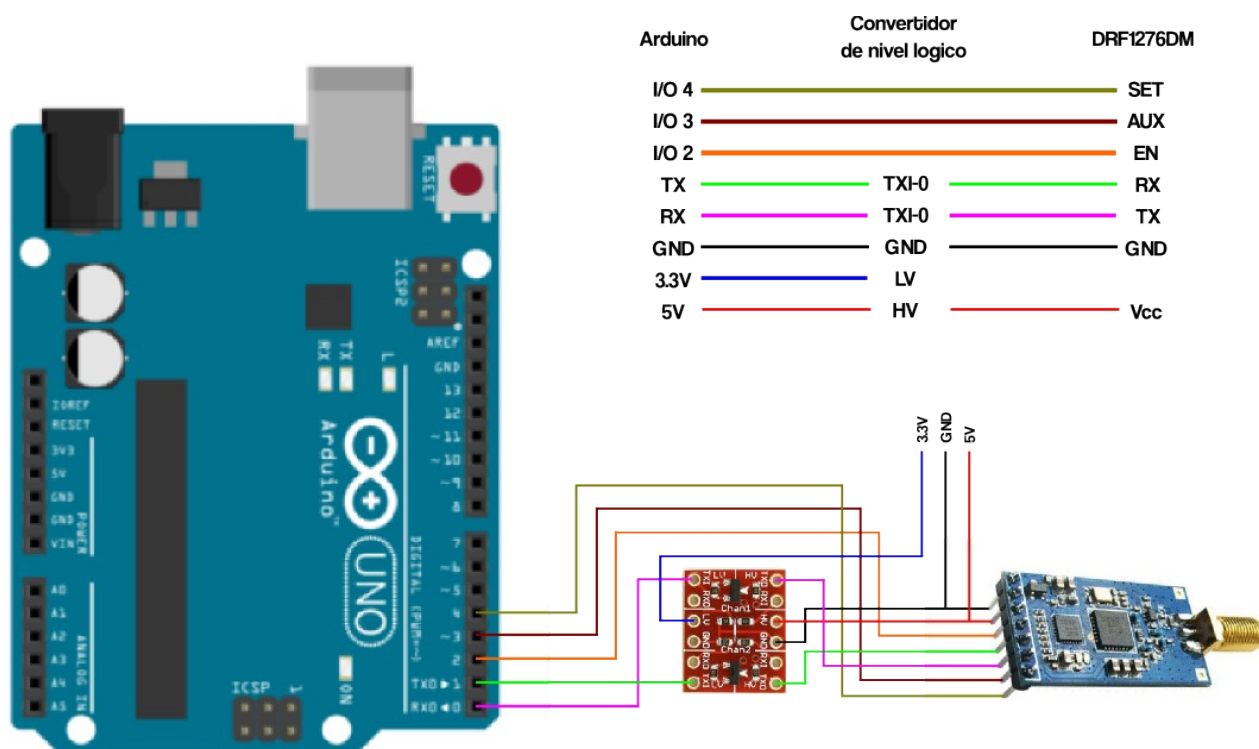


Figura 4.12 Diagrama de conexión eléctrica de antena DRF1276DM y Arduino.

Se puede observar que la antena se encuentra conectado al puerto uart del microcontrolador. Por lo que para recibir los datos tenemos que habilitar el puerto serial del microcontrolador.

La interpretación de la comandos que son enviados remotamente, son recibidos por medio del puerto UART, para ello se sigue el siguiente diagrama de flujo.

Función de Control Remoto Luminaria

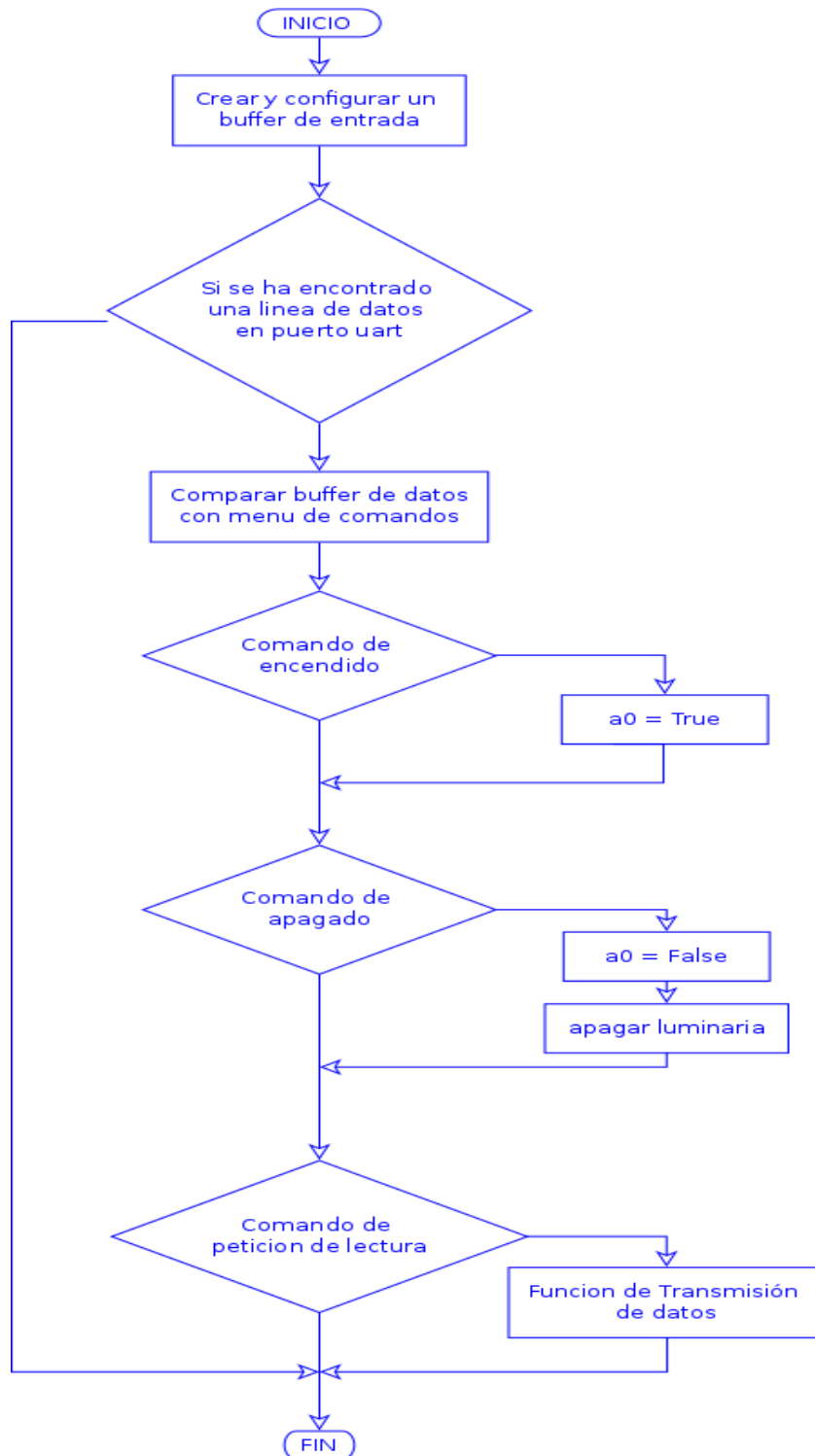


Figura 4.13 Diagrama de flujo para Recepción de datos.

4.2.1.5 Registro de eventos

El registro de eventos, esta función tienen principal de verificar que la luminaria opere correctamente, y para ello ocupa un sensor que verifica que la luminaria está encendida. Comparando el estado de la luminaria encendida y además comparando que el estado del control de la luminaria también se encuentre activada.

Esta verificación se realiza igualmente mediante una fotoresistencia, la cual en el siguiente diagrama se describe su conexión eléctrica.

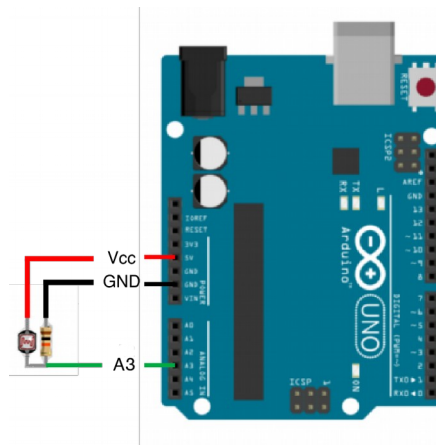


Figura 4.14 Diagrama de Conexión Eléctrica de Registro de eventos.

Para la elaboración del control debido a fallas en la tabla 2, se contemplan los siguientes errores que se interpretan con sus respectivos códigos.

	Bandera de control Remoto	Estado	Verificador de Fallas		Voltaje [V]		Condición	Código
1	ON	ON	ON	√	127	√	√	220
2	ON	ON	OFF	x	127	√	x	330
3	ON	ON	OFF	x	0	x	x	440
4	ON	OFF	ON	x	127	√	x	550
5	ON	OFF	OFF	√	127	√	√	220
6	ON	OFF	OFF	√	0	x	x	660
7	OFF	-	ON	-	-	-	x	770

Tabla 4.2 Tabla de relación de códigos de errores.

Código	Descripción de problema
220	Operación Adecuada
330	Falla de lampara no enciende, CUIDADO hay tensión de alimentación a lampara.
440	Falla de lampara no enciende, No hay tensión de alimentación a lampara.
550	Falla de lampara esta encendida no se tiene control, CUIDADO hay tensión de alimentación a lampara.
660	Falla No hay tensión de alimentación a lampara.
770	Falla de Control Remoto.

Tabla 4.3 Descripción del problema probable de los distintos códigos.

Función de verificación de Fallas

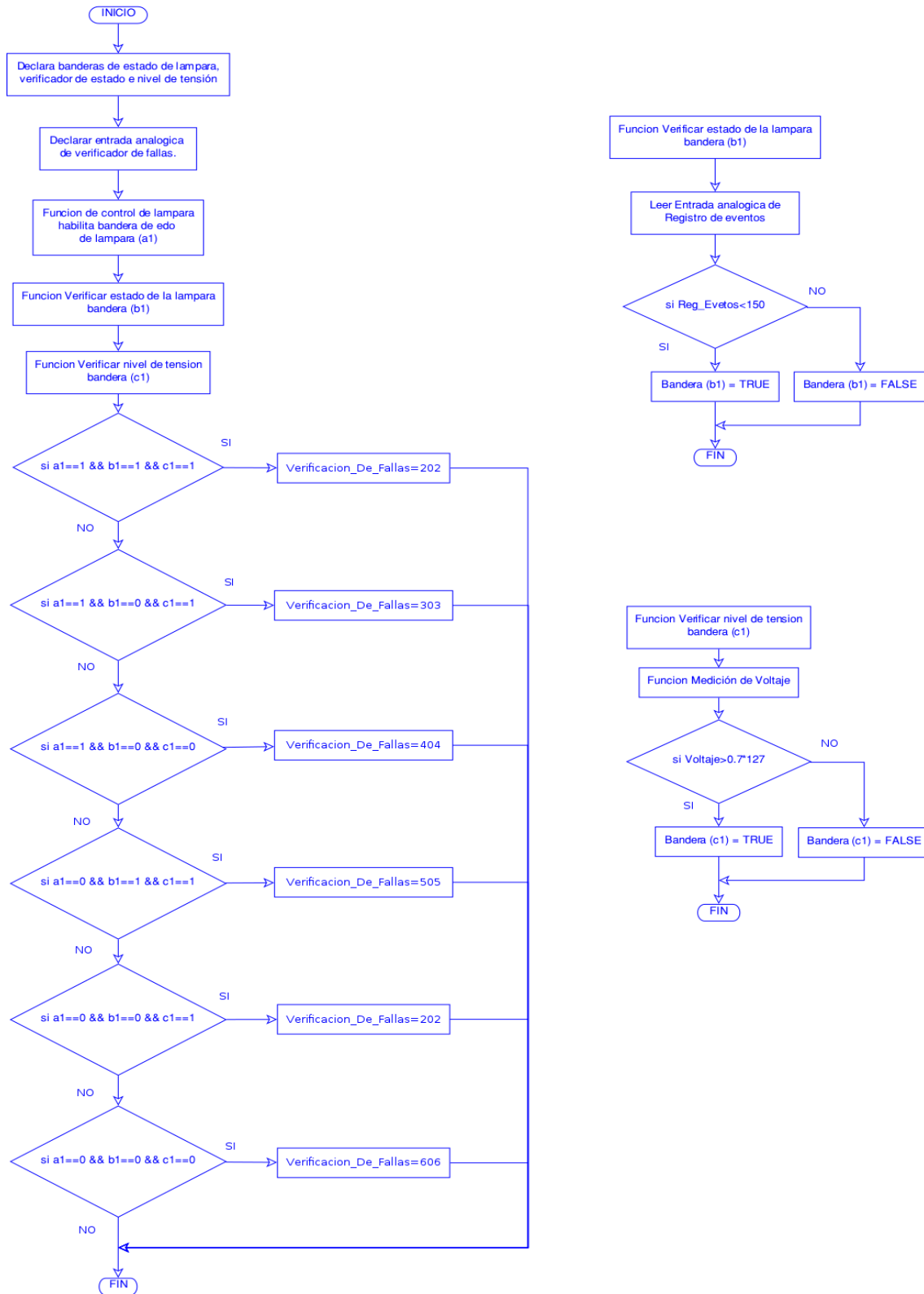


Figura 4.15 Diagrama de flujo de Registro de Eventos.

En el este diagrama de flujo, se describe como opera mediante software, el desarrollo de esta función.

4.2.1.6 Transmisión de datos

La transmisión de datos consta de dos etapas la primera es la obtención de los diferentes parámetros que se va a enviar, por lo que previo a que la transmisión de datos se realice, se deben de ejecutar las funciones de medición corriente y voltaje, nivel de luminosidad y verificación de operación adecuada de luminaria por el registro de eventos.

Ya con los parámetros obtenidos se debe conjuntar estos parámetros en una trama de bytes; siendo este un prototipo, se emplea una trama propia para enviar información¹, por lo que no sigue regulación alguna. LoraWAN emplea tipo de encapsulación AES Secured Payload.

La figura 12 muestra el patrón de la trama de bytes para la transmisión de datos:

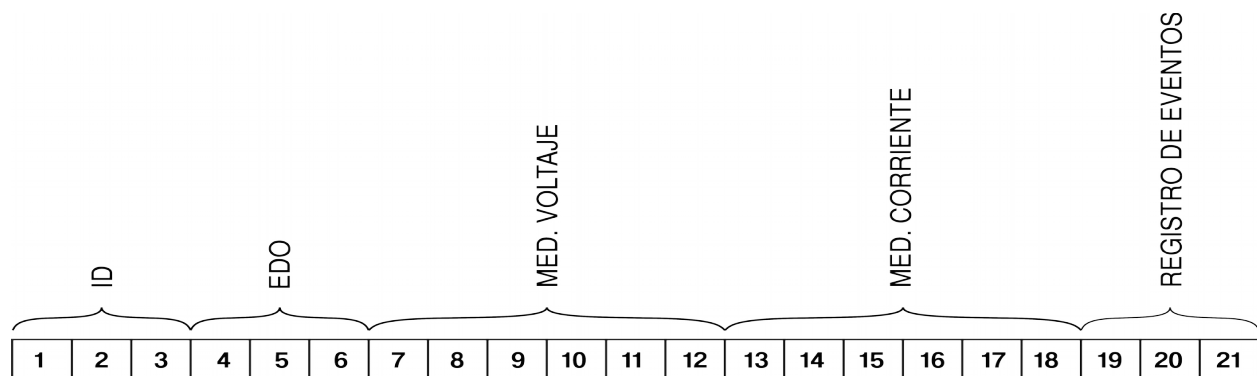


Figura 4.16 Secuencia de trama de datos

Como ultima operación esta trama de datos se enviá al puerto uart del microcontrolador, el cual esta conectado eléctricamente con la antena, como se describió anteriormente en la recepción de datos².

A continuación el el diagrama de flujo de la figura describe la funcionalidad de la función de transmisión de datos.

¹ La razón de proponer una trama propia es para reducir el numero de bytes a transmitir, reduciendo el tiempo de transmisión.

² Nota: Para cerrar la trama de datos, es recomendable enviar un retorno de carro, para poder identificar al momento recibir los datos, el final de la trama.

FUNCIÓN DE TRANSMISIÓN DE DATOS

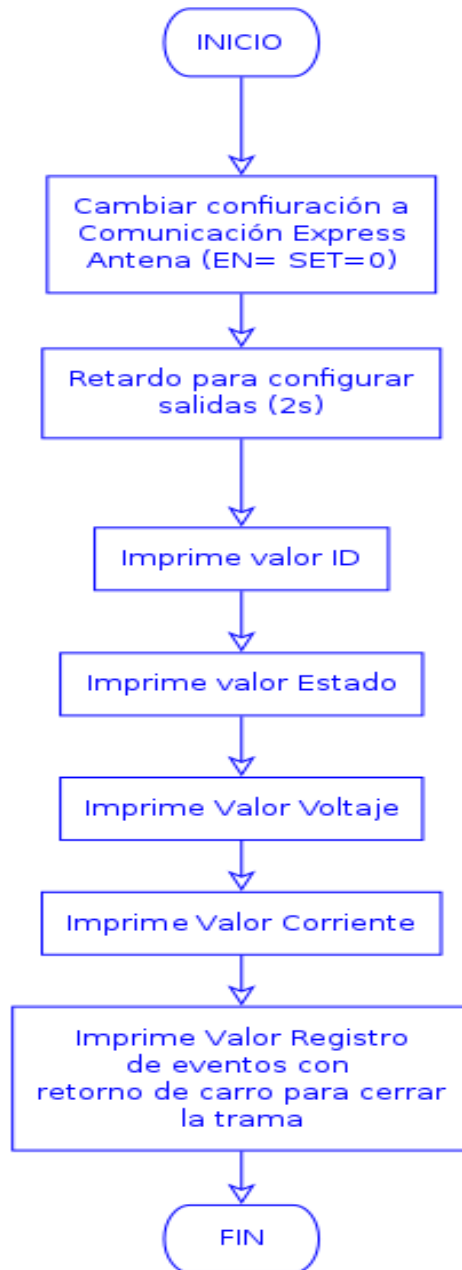


Figura 4.17 Diagrama de Flujo de transmisión de datos.

4.3 Diseño de la Estación Central de Control

La estación Central de control, es el dispositivo mas importante del sistema ya que recaba toda las información de los distintos Módulos de Estación de medición y control. Esta estación tiene como objetivo principal enviar y recibir información a cada uno de los dispositivos en las luminarias, dándole la capacidad de poder monitorear y controlar estos dispositivos de manera remota mediante la tecnología LoRa.

Este Modulo de Central al igual que el Modulo de estación, su desarrollo se dividen en dos principales secciones:

- **Software:** Consta en el diseño y desarrollo de programación para adquirir todos los parámetros que se necesitan en este proyecto (Corriente, voltaje, eventos) de cada Modulo de Estación y Almacenar todos estados de manera ordenada en una base de datos.
- **Hardware:** Se contempla la implementación de un Modulo Central, esto con la conexión eléctrica de la antena DFR1276DM (Dorji). La importancia de tener un servicio NTP es importante para determinar el momento de la toma de la muestra o la imlementción de un reloj externo tambien puede ser posible esto con el fin de no perder la sincronía de tiempo para el registro de muestras de cada una de las luminarias.

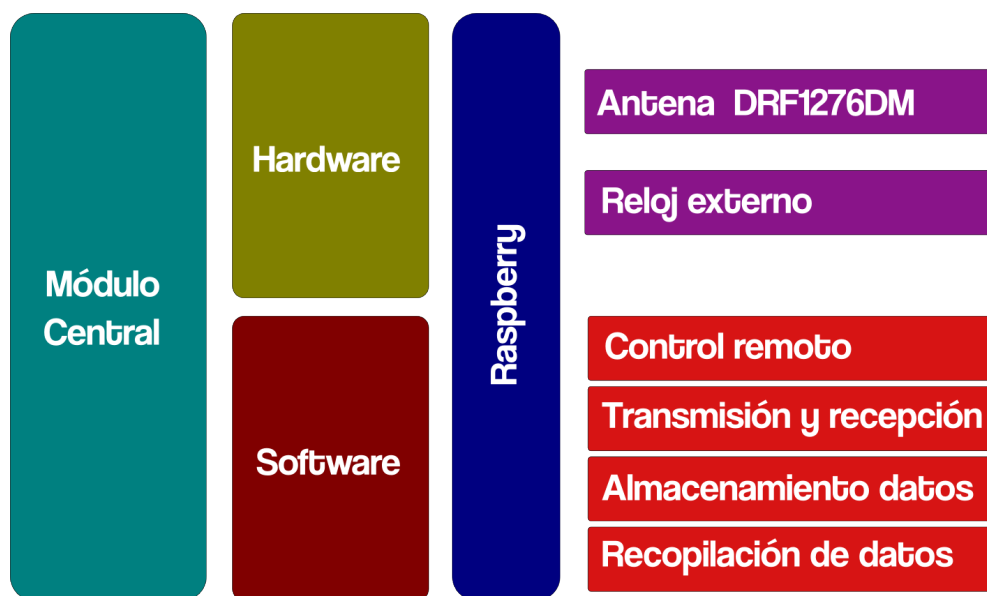


Figura 4.18 Resumen de integración de Módulo Central.

4.3.1 Procedimiento de Administración Inteligente

El procedimiento de administración, va a ser de la manera de como se realizará el control remoto de habilitación de encendido o apagado de las luminarias, la recopilación de parámetros de luminarios y su almacenamiento. Para cumplir estos objetivos se tiene que ejecutar las siguientes operaciones:

- Control remoto de habilitación encendido de lampara (simple o Broadcast).

La petición de habilitación de encendido de lampara, se realiza bajo demanda, esto significa que es

necesario ejecutar un programa por parte del personal para que esta solicitud se realice, como estado de default todas las lamparas estarán habilitadas para poder encender y apagar dependiendo el nivel de luminosidad solar.

- Control remoto de apagado de lampara (simple o Broadcast).

La petición apagado al igual que la petición de habilitación de encendido, requiere el personal realice la petición, mediante la solicitud para que esta petición se realice.

- Petición solicitud de datos (simple o Broadcast)

Para esta petición de solicitud de datos, esta se realiza de manera periódica, y de manera automática empleando la herramienta de tareas programables (cron) del propio sistema operativo de la Raspberry, la cual permite ejecutar este programa de petición de datos de manera periodica; para esta aplicación se emplea periodos de 15 minutos entre una muestra a otra, siendo este el periodo de muestreo de la compañía suministradora.

Con esta breve descripción de cada una de las operaciones, podemos observar que no siempre se requiere de un programa que este siempre en modo ejecución, y que pueden operar a base de peticiones y este ultimo funciona cada cierto periodo gracias a un servicio de raspbian (cron), es factible realizar cada una en una programa específico para la petición requerida¹.

4.3.2 Sistema de Comunicación

El sistema de comunicación cumple la función de transmitir y recibir datos hacia y de los módulos Estación. Este sistema esta compuesto principalmente en la conexión eléctrica del puerto uart de la raspberry y la antena con tecnología LoRa. Que se muestra en la siguiente figura.

Siendo la estampa de tiempo un parámetro importante, ya que la Raspberry tiene el problema de no retener el ajuste de tiempo, es necesario que este proyecto pueda optar entre dos opciones, habilitar un servicio NTP (Network Time Protocol) por parte de la Raspberry para actualizar la estampa de tiempo y en otro caso se puede agregar un modulo DS3231 (RTC, Real Time Clock) la incluye una pila de alimentación independiente. Se opta por esta segunda opción ya que en algún momento se puede perder la conexión a Internet por lo que podemos perder el servicio NTP. En la figura 3 y 4 se describe su conexión eléctrica con la raspberry con el RTC y la antena DRF1276DM.

Con el propósito de reducir el numero de bytes a transmitir se desarrollo una trama propia para la transmisión de peticiones, el arreglo de esta transmisión de datos es describe en la figura14.

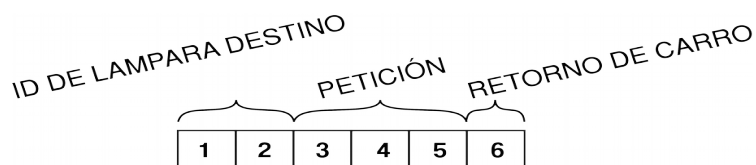


Figura 4.19 Estructura de trama de datos de peticiones a lamparas.

¹ Sí cada tipo de petición es un programa específico, para un número amplio de peticiones sería un número amplio de programas, con lo que se generaría posteriormente un problema de manejo de programas de peticiones, este problema se dejará para trabajos a futuro.

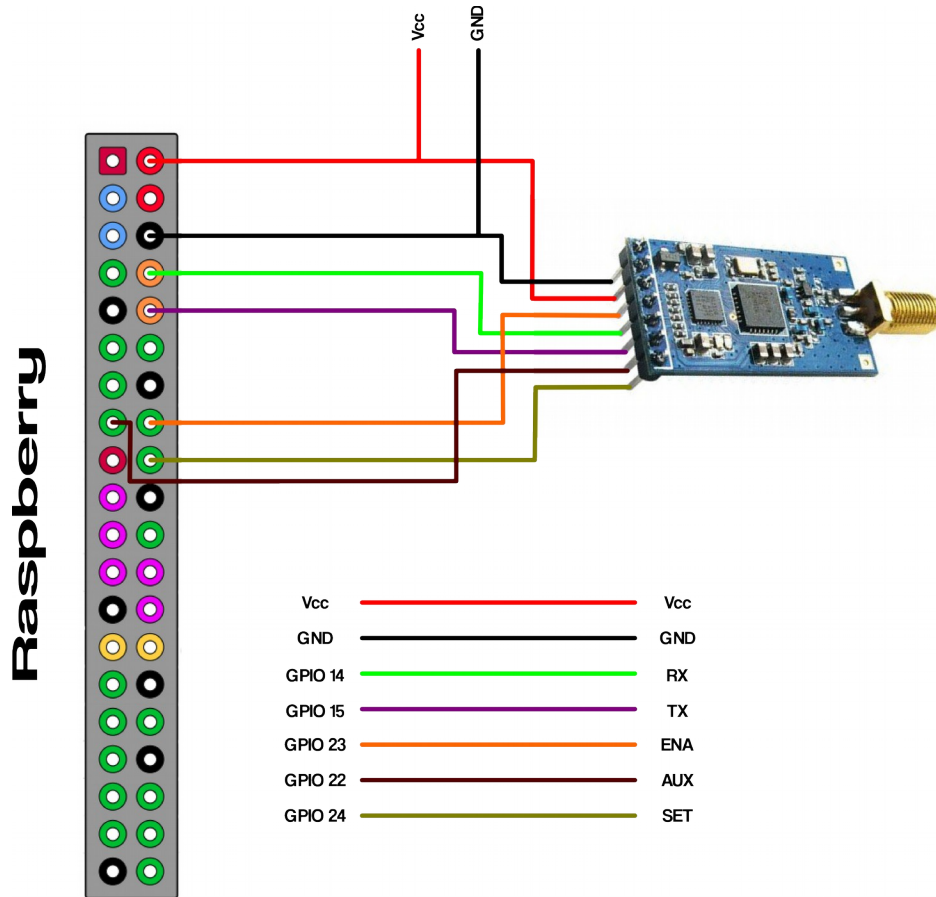


Figura 4.20 Diagrama de conexión eléctrica de antena DRF1276DM y Raspberry.

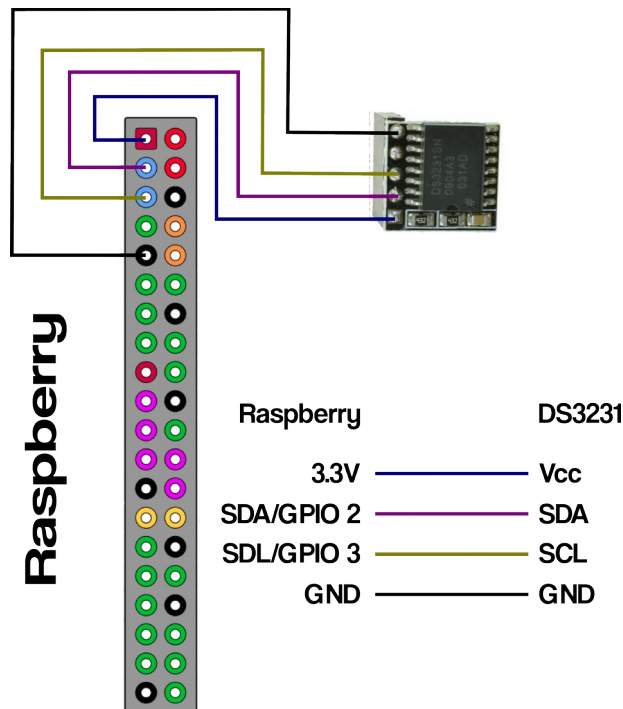


Figura 4.21 Diagrama de conexión eléctrica de modulo DS3231 y Raspberry.

4.3.3 Procedimiento de control

Hay un problema que acarrea el uso de esta versión (DRF1276DM) de antena, que se tiene que considerar al momento de realizar una recepción de datos; esta versión de dispositivos tiene dos formas de trabajo, comunicación express (continua) y una comunicación intermitente o pausada. Se emplea la comunicación intermitente para este proyecto con el propósito de reducir el consumo energético por parte de la antena. El problema que aparece en esta versión de antenas cuando se realiza una petición de lectura de datos al momento de realizar la transformación de datos para la petición se realiza con la configuración en comunicación pausada, pero al momento de la recepción de datos por parte del modulo central se tiene que cambiar de forma de comunicación de intermitente o pausada a express, para poder recibir los datos; esto solamente en el caso de la petición de lectura de datos que es el que requiere recibir datos. Para realizar el cambio de tipo de comunicación se realiza mediante el cambio lógico de los pines EN y SET de la antena DRF 12756DM que se describe en la siguiente tabla.

Tipo de Comunicación	Modulo Central		Modulo Estación	
	EN	SET	EN	SET
Express	0	0	0	0
Pausada	0	1	1	1

Tabla 4.5. Ajuste de pines para el cambio de tipo de comunicación

Como se menciona al principio del diseño del Modulo central, ya que cada tipo de petición se realizará con un programa específico ya que cada uno construye una trama donde se coloca la petición.

La primera petición a analizar es la de apagado remoto que se describe en el siguiente diagrama.



Figura 4.22 Diagrama de Flujo de transmisión de datos para apagado de lamparas .

Para este caso en particular hay que recordar como se tiene que construir la trama los primero 2 bytes son del ID de la lampara a la que se le esta enviado la petición, seguido de tres bytes en el cual va colocado la petición que es una combinación de tres bytes para este caso es "OFF" y para cerrar la trama y la antena identifique que es el fin e la trama se coloca en un byte el byte que representa un retorno de carro("0D") en código ascii. Completo la trama es enviada a través del puerto serie.²

Para la segunda petición de habilitación de encendido, es muy similar a la petición de apagado solo que la combinación de la petición se coloca la combinación "ON". Para generar la petición se sigue el siguiente diagrama de flujo.



Figura 4.23 Diagrama de Flujo de transmisión de datos para habilitación de encendido de lamparas.

Como tercera petición es la que corresponde a la adquisición de datos es decir enviar una petición para que un Módulo estación en específico transmita sus parámetros de medición. Para la recepción de datos se requiere un cambio de tipo de comunicación de breath a express. Ya que conocemos la distribución de la trama se distribuye byte por byte al parámetro correspondiente de la información enviada por el modulo estación, posterior a esto se guardan todos estos parámetros en un arreglo correspondiente que posteriormente estos arreglos se guardaran en una de las tablas de una base de datos de tipo MySQL. Esto con el objeto de poder acceder a esta información de una fácil manera, gracias a las librerías compatibles entre python y MySQL se puede lograr esto, además de que proporciona cierta seguridad ya que en para acceder a una base de datos, es requerido una usuario y contraseña.

Otras ventajas de usar una base de datos es que se acceder a estos datos desde distintos programas de desarrollo web, esto da facilidad de implementar una interfaz gráfica a partir de la información que se

² La implementación del programa para el envío de peticiones se realiza en pyton.

obtiene de una petición de lectura de parámetros.

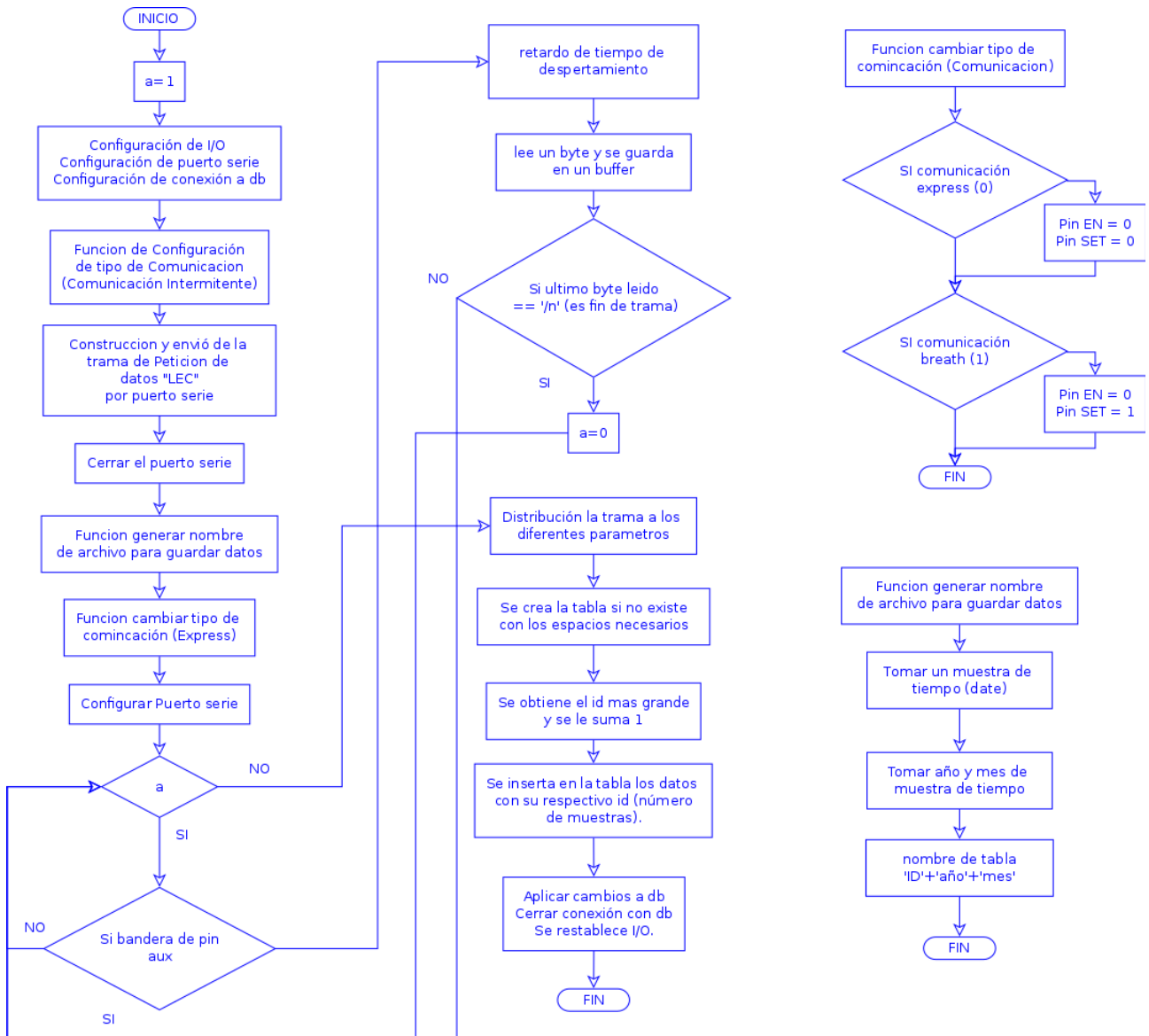


Figura 4.24 Diagrama de Flujo de Petición de Lectura de Parámetros.

4.4Diseño de Interfaz gráfico

4.4.1 Administración de base de datos

Siendo una base de datos un conjunto de de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Hoy día existen programas denominados sistemas gestores de base de datos que permiten agrupar, almacenar, y permiten acceder de manera rápida y estructurada a estos datos. En este proyecto se emplea el programa MySQL para generar las base de datos ,como apoyo para una gestión gráfica de las bases de datos se utiliza PhpMyAdmin.

MySQL es un manejador de base de datos de código libre permite almacenar y organizar la información que se requiere para poder acceder a la interfaz gráfica y al manejo de la información que es obtenida de los distintos módulos de Estación. El fácil acceso a los datos desde programas de desarrollo web fue el principal factor para seleccionar esta forma de almacenar los datos.

Después de la instalación de MySQL que se describe en el anexo B para acceder al gestor de base de datos gráfico, es necesario ir abrir un buscador web. Y en el área de direcciones escribir los siguiente:

http://ip_de_raspberry/phpmyadmin

Se mostrará un menú para autenticar, en este punto se utiliza como usuario root y también se emplea la contraseña establecida durante la instalación para ingresar.

Es recomendable crear otro usuario, que es el que se usa para manejar los datos de este proyecto. Para ello se tiene ingresa en la sección SQL e ingresa el siguiente código:

```
CREATE USER 'nombre_usuario'@'localhost' IDENTIFIED BY 'tu_contraseña';  
GRANT ALL PRIVILEGES ON *.* TO 'nombre_usuario'@'localhost';  
FLUSH PRIVILEGES;
```

La primera linea crea el usuario y define la contraseña, la segunda linea otorga privilegios al usuario creado y la tercera linea refresca todos los privilegios.

Se sale de la pagina para volver a ingresar como el usuario que se acaba de crear. El siguiente paso es crear una base de datos, que se realiza con la siguiente linea en SQL:

```
CREATE DATABASE nombre_base_de_datos;
```

Con esto se tiene configurado completamente la base de datos, el ingreso de datos se dará desde el archivo de petición de lectura de parámetros y el acceso a los datos se de desde la programación de el interfaz gráfico. Como nota adicional es necesario crear tablas donde se almacenaran los datos pero no se realiza este paso aquí ya que esto se realiza desde el archivo de petición de lectura de datos del Modulo Central.

4.4.2 Arquitectura Modelo- Vista-Controlador

En el diseño del interfaz gráfico se busco varias opciones de como generarlo, y se opto por desarrollar una pagina web, por que desde cualquier dispositivo se puede acceder al sistema, esto siempre y cuando se encuentre autorizado para ingresar. Siendo este un prototipo, la interfaz gráfica se implementa en una red

local, para efectos de escalabilidad se puede se puede colocar en Internet; donde podrá tener un mayor alcance dispositivos para administrar.

La elección de una página web fue definida, pero la selección del lenguaje de programación fue un poco difícil de seleccionar entre una variedad que el mercado ofrece hoy en día. Se seleccionó implementarlo en php ya que además se ser compatible con el sistema operativo, php es capaz de ejecutar archivos de python, formatos en el que están escritos los programa de control remoto del modulo central, además que el soporte que existe de este lenguaje fue fundamental para este implementación.

El diseño de esta interfaz se realiza usando una arquitectura MVC (Modelo Vista Controlador), esta arquitectura de software, separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (Modelo, Vista , Controlador). Esta arquitectura es frecuente en aplicaciones, en donde la Vista es una pagina HTML que maneja datos dinámicos. El Modelo es el Sistema de gestión de base de datos y la lógica de negocio. El controlador es el responsable de recibir los eventos de entrada desde la Vista. Un modelo puede tener diversas vistas, cada una con su correspondiente controlador.

Modelo: Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

Vista: Este presenta el Modelo, usualmente la interfaz gráfica de usuario. La vista es la capa de la aplicación que ve el usuario en un formato adecuado para interactuar. Las vistas son responsables de recibir datos del modelo y los muestra al usuario.

Controlador: Es la capa que controla todo lo que puede realizar la aplicación. Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Está compuesto por acciones que se representan con funciones en una clase. El controlador es responsable de recibir eventos de entrada, contiene reglas de gestión de eventos que pueden significar peticiones a vistas

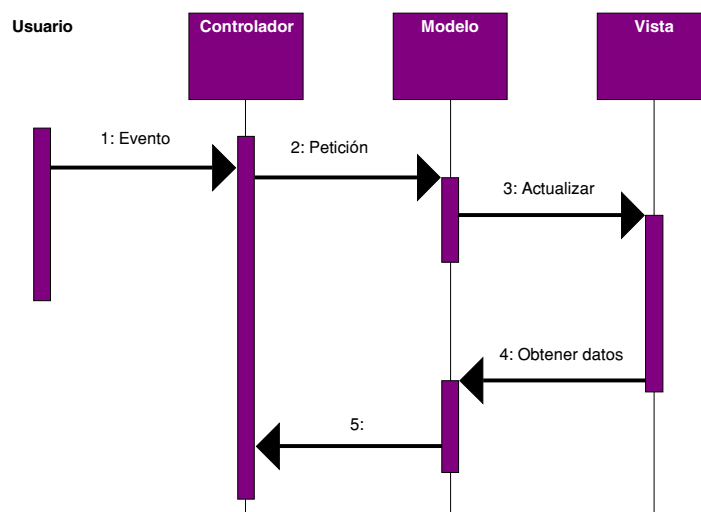


Figura 4.25 Diagrama de secuencia de MVC.

La secuencia de la arquitectura MVC, sigue los pasos siguientes:

- Usuario Introduce el evento

- El Controlador recibe el evento y lo traduce en una petición al Modelo
- El modelo llama a la vista para su actualización
- Para cumplir con la actualización la Vista puede solicitar datos al modelo
- El Controlador recibe el control.

Ventajas y Desventajas

La popularidad de este diseño se debe a que es mucho más fácil organizar aplicaciones grandes.

Las ventajas

- Separación entre interfaz, lógica de negocio y de presentación, que además provoca parte de las ventajas siguientes.
- Sencillez para crear distintas representaciones de los mismos datos.
- Facilidad para la realización de pruebas unitarias de los componentes.
- Reutilización de los componentes.
- Simplicidad en el mantenimiento de los sistemas.
- Facilidad para desarrollar prototipos rápidos.
- Los desarrollos suelen ser mas escalables.

Desventajas:

- Tener que ceñirse a una estructura predefinida, lo que a veces puede incrementar la complejidad del sistema. Hay problemas que son difícil de resolver respetando el patrón.
- La curva de aprendizaje para los nuevos desarrolladores es estima mayor que la de modelos más simples como webforms.
- La distribución de componentes obliga a crear y mantener un mayor numero de ficheros.

El diseño de la interfaz gráfica es el siguiente:

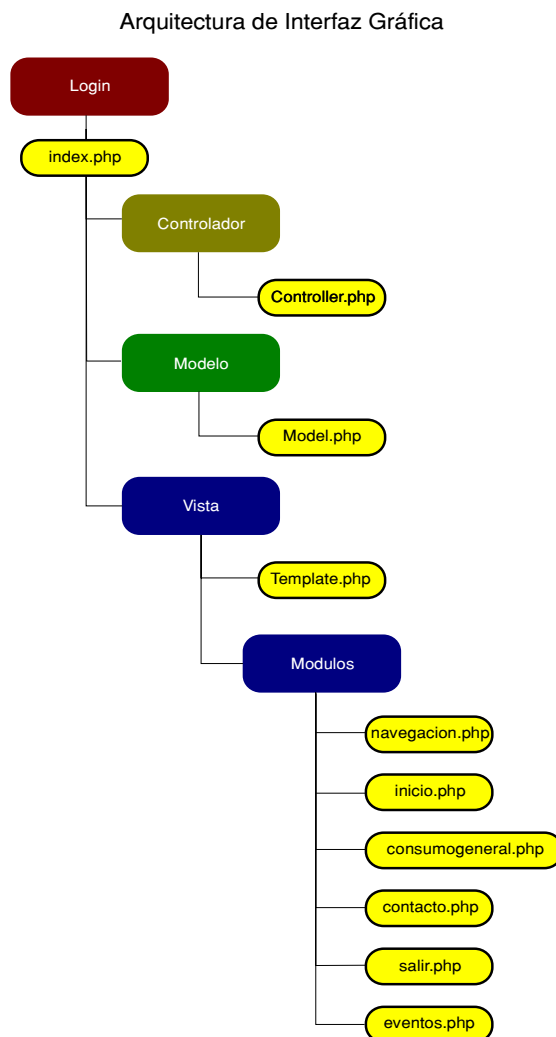


Figura 4.26 Construcción de proyecto con Arquitectura MVC.

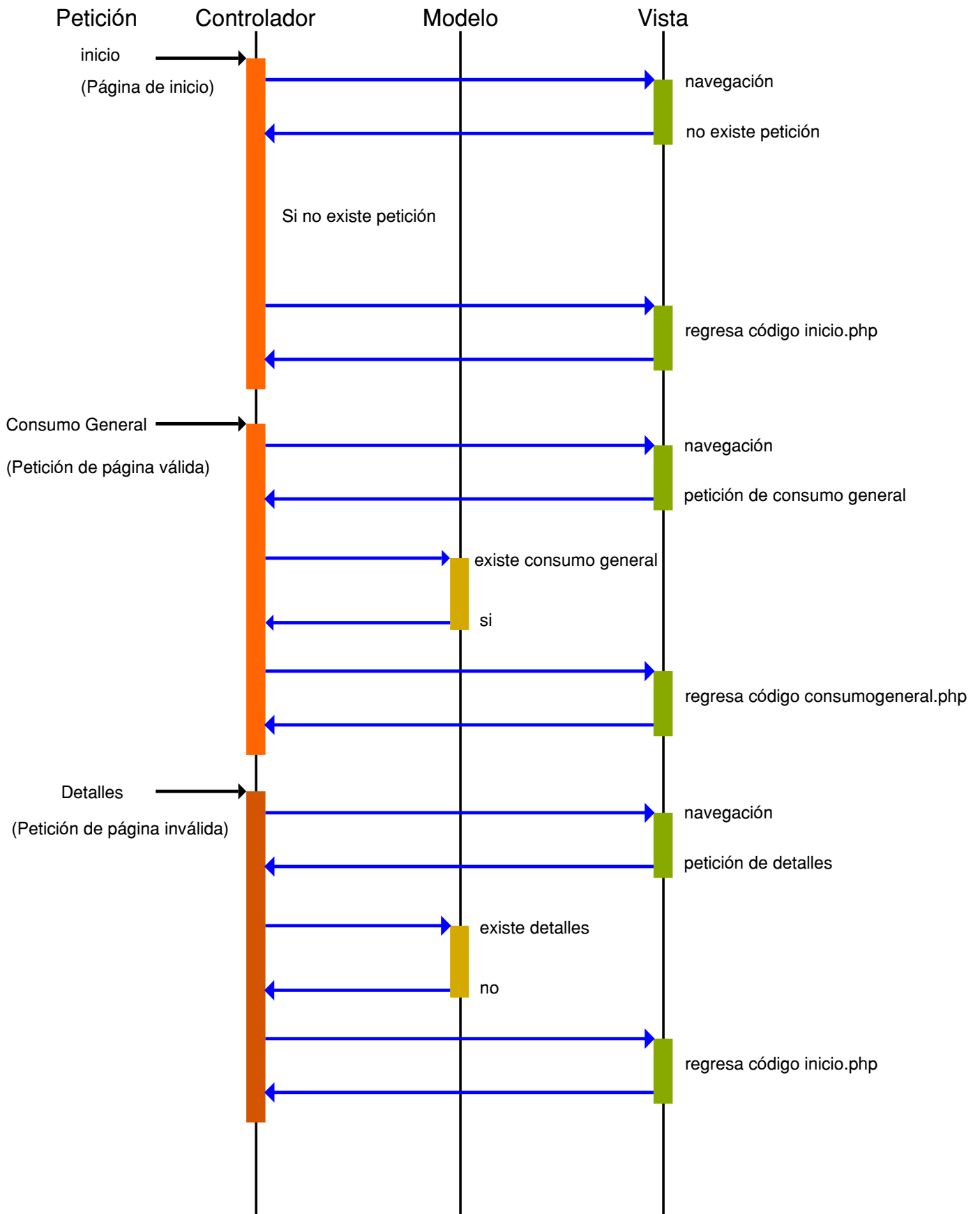


Figura 4.27 Diagrama de secuencia de arquitectura MVC.

La figura 3, describe el funcionamiento de la arquitectura MVC de este proyecto , se puede observar existen tres diferentes peticiones que son generadas por el usuario petición de inicio, petición valida y petición invalida.

Antes de describir el funcionamiento de esta arquitectura se debe mencionar que se para entrar en esta interfaz de control, se inicia entrando a la pagina de login que funciona de la siguiente manera:

- Se solicita el usuario y contraseña para acceder.
- Se compara estos datos con datos de base de datos.
- Sin son validos ingresan, si no son validos se retorna a pagina de login.

La arquitectura MVC funciona de la siguiente manera para el primer caso:

- Cuando se acredita el ingreso se abre una pagina index.php que invoca al controlador y al modelo.
- Como primer paso el controlador invoca al template que es el administra todo el código para generar el interfaz gráfico, en este primer paso se enviá el código del menú de navegación y se pregunta si existe alguna petición por parte del usuario,
- Siendo que es una petición de inicio, se enviá el código correspondiente con el inicio del interfaz (inicio.php).
- De esta manera se completa el código a mostrá al usuario, el controlador se encarga de ello.

En el segundo caso, si existe una petición valida, la arquitectura opera de esta manera:

- En caso de que exista una petición por parte de un usuario, el controlador invoca al template, en este primer paso se enviá el código del menú de navegación y se pregunta si existe alguna petición por parte del usuario, Para esta petición válida se hace una petición al controlador que muestre esa sección del usuario solicita.
- El controlador invoca al modelo, para verificar si existe la petición solicitada por el usuario, en este caso que es válido. El modelo habilita la controlador a invocar en el template a la pagina solicita.
- Se solicita al template, presentar la sección de código solicitada, para incorporarlo a la programación de la interfaz gráfica.
- De esta manera se completa el código a mostrar al usuario, el controlador se encarga de ello.

En el tercer caso si existe una petición invalida, la arquitectura opera de esta manera:

- En caso de que exista una petición por parte de un usuario, el controlador invoca al template, en este primer paso se enviá el código del menú de navegación y se pregunta si existe alguna petición por parte del usuario; Para esta petición inválida se hace una petición al controlador que muestre esa sección solicitada.

- El controlador invoca al modelo, para verificar si existe la petición solicitada por el usuario, en este caso que es inválido. El modelo envía información al controlador para invocar al template que muestre la página de inicio.
- Se solicita al template presentar el código de inicio de la interfaz gráfica.
- De esta manera se completa el código a mostrar al usuario, el controlador se encarga de integrar todo el código.

Las peticiones validas son aquellas que están contenidas en la carpeta de módulos, que son las diferentes páginas que puede visualizar el usuario.

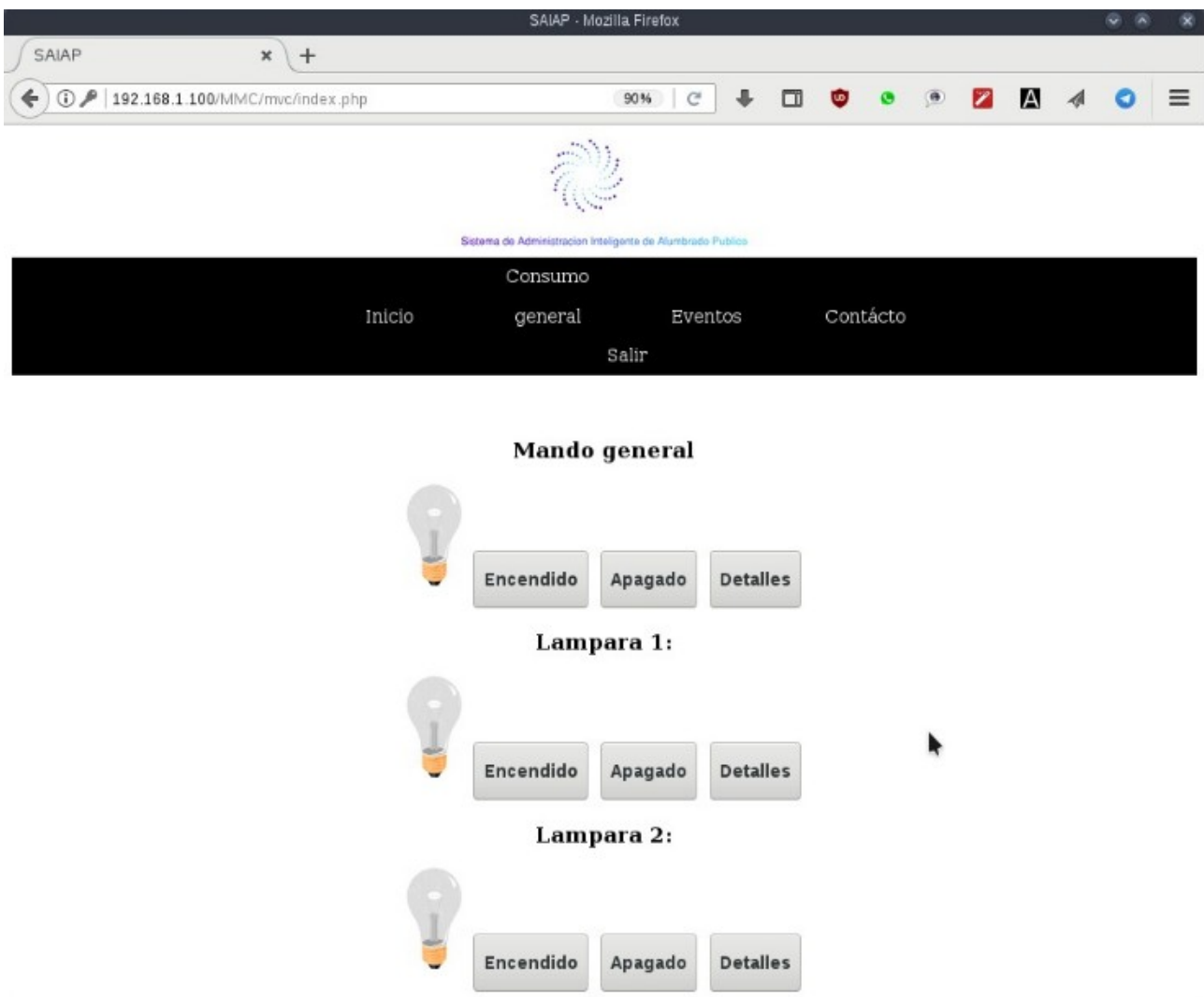


Figura 4.28 Página de Inicio de Interfaz Gráfica.

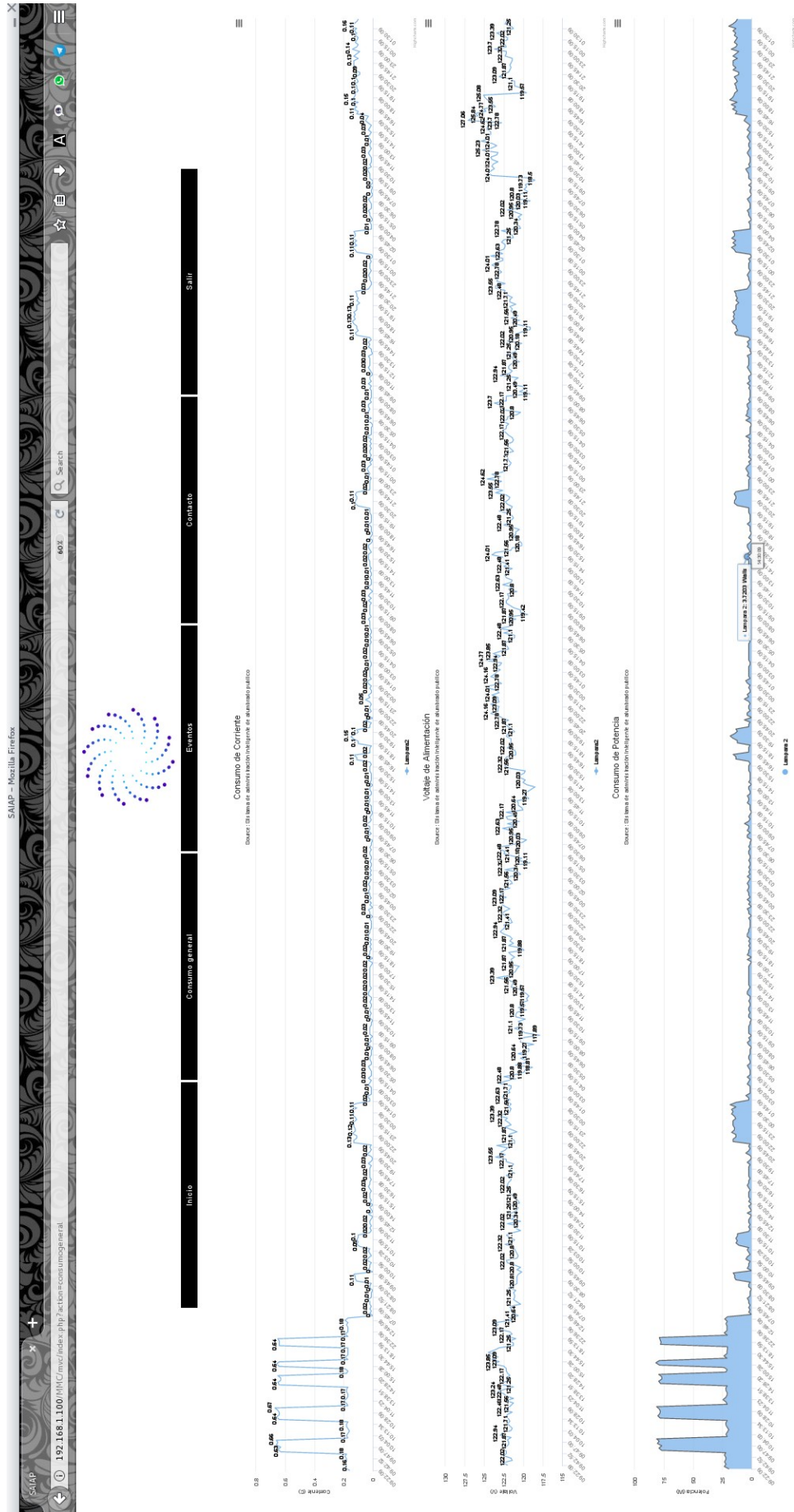


Figura 4.29 Página de Consumo de un lampara en Interfaz Gráfica.

Capitulo 5 Pruebas

- Pruebas de la Estación de Medición y control
 - Verificación de mediciones de voltaje
 - Verificación de mediciones de Corriente
 - Verificación de sensor de luz para control de encendido y apagado de lampara
 - Verificación de registro de eventos
- Pruebas del Sistema de comunicación
- Pruebas del sistema Integral

Durante la etapa de desarrollo de este proyecto fue necesario probar de manera individual, algunos elementos que requerían de cierta atención, entre estos elemento se encuentran:

- Sistema Medición
- Transmisión y recepción de datos (Modulo Estación)
- Transmisión y recepción de datos (Modulo Central)
- Almacenamiento de datos

En un resumen general estos son los siguientes puntos que hay que probar de manera individual, y verificar su correcta operación:

- Pruebas de la Estación de Medición y control
- Pruebas del Sistema de comunicación
- Pruebas del sistema Integral

5.1 Pruebas de la Estación de Medición y control

En este primer punto a verificar la correcta medición de nivel de tensión comparado con otros equipos de medición y determinara la exactitud de nuestra medición de voltaje y corriente.

Con respecto al los sensores de luminosidad se verifica su adecuada operación mediante pruebas de control de luminosidad manual.

5.1.1 Verificación de mediciones de voltaje

No. de Medición	Medición de Voltaje con PM800 [V]	Medición de Voltaje con Modulo Estación [V]	Diferencia [V]	% Desviación
1	124	131.04	7.04	5.68
2	125	130.28	5.28	4.22
3	124	129.8	5.8	4.68
4	124	130.12	6.12	4.94
5	124	129.6	5.6	4.52
6	124	128.44	4.44	3.58

Tabla 5.1. Mediciones de voltaje con medidor PM800 (Schneider Electric) vs Modulo Estación.

En la tabla 1 se puede observar el comportamiento de las mediciones realizadas, con dos equipos empleado como referencia el equipo de medición PM800 (Schneider Electric) vs nuestro dispositivo desarrollado (Modulo Estación). De seis muestras que se tomaron se puede observar que solo en la primera muestra se supera el 5 % de desviación con respecto al medidor de referencia, si se aplica la norma NETA ATS – 2009 entra nuestro Dispositivo podría ser aprobado ya que cumple con una desviación máxima de 5% en la

medición de voltaje.

A pesar de cumplir con la desviación máxima como trabajo futuro queda pendiente realizar una revisión en la metodología de obtención de estos datos, con el objeto de poder reducir esta desviación en las mediciones de voltaje.

5.1.2 Verificación de mediciones de Corriente

A continuación se realiza la comparaciones de mediciones de corriente, tomando como referencia un amperímetro UNI-T UT203 versus mediciones del Modulo Estación.

Como descripción del desarrollo, las dos primeras muestras se realizaron en vacío, esto significa sin que la lampara estuviera encendida.

Para las muestras 3, 4 y 5 se realizaron con una lampara fluorescente encendida.

Para las muestras 6, 7 y 8 se realizaron con una lampara incandescente encendida.

No. de Medición	Medición de Corriente con Amperímetro [A]	Medición de Corriente con Modulo Estación [A]	Diferencia [A]	% Desviación	Tipo de lampara
1	0.09	0.19	0.1	111.11	Vacío
2	0.09	0.19	0.1	111.11	
3	0.23	0.49	0.26	113.04	Lampara Fluorescente
4	0.22	0.51	0.29	131.82	
5	0.22	0.51	0.29	131.82	
6	0.59	0.66	0.07	11.86	Lampara Incandescente
7	0.59	0.64	0.05	8.47	
8	0.6	0.63	0.03	5.00	

Tabla 5.2. Mediciones de Corriente con Amperímetro UNI-T UT203 vs Modulo Estación.

La tabla 2 muestra los resultados de las distintas mediciones de corriente, como se menciona se trabajo evaluando tres opciones para medir distintas.

En el primer caso cuando se evaluó en vacío sin que la lampara este encendida se tiene una desviación del 110% por lo que para esta medición se puede concluir que la medición realizada en este punto no es fiable. En el segundo caso se evaluó con un lampara fluorescente encendida se tiene una desviación del 130 % en promedio. En el tercer y ultimo caso se observa una desviación no mayor al 12%. Ya que como primera prueba habrá que examinarla metodología con la que se realizan las mediciones ya que las desviaciones son muy irregulares y desproporcionadas. Como conclusión las mediciones son incorrectas es necesario buscar otra metodología para realizar este tipo de mediciones.

5.1.3 Verificación de sensor de luz para control de encendido y apagado de lampara

Para realizar este tipo de pruebas es necesario probar a distintos niveles de luminosidad, por ello para realizar este ejercicio ya que es mas fácil iniciaremos con un habitación en completa oscuridad y mediante una lampara, se incide luz en el sensor de luminosidad¹, al cambiar la distancia entre la lampara y el sensor

¹ Es probable que en una habitación completamente oscura la lampara automáticamente se encienda la lampara hay que procurar que esta luz de

se puede modificar la cantidad de luminosidad que incide en el sensor, con ello verificar que opera de manera adecuada.

La señal que proviene del sensor es una señal analógica que se convierte a una señal digital, la cual tiene un máximo de 1024 partes; dicho esto, de manera empírica se ha observado que la operación correcta del Modulo es con un umbral de 200 partes. Por lo que nuestra referencia de operación será este valor para el encendido de la lampara.

También es importante señalar que se probó el control remoto, la forma mas visible de observar esta prueba es probar cuando la lampara esta encendida, así se puede observar el encendido o apagado de la lampara, que sucede en este caso, el control remoto habilita o deshabilita el control automático de encendido o apagado de la lampara; para el caso de que la lampara este encendida y el control remoto mande a deshabilitar el control automático, la lampara también se apaga. Para el caso de que esta prueba se realizo cuando la lampara estaba apagada y el control remoto manda a deshabilitar el control automático, no se alcanza a percibir ningún cambio pero se deshabilita el control automático de la lampara.

Verificación de sensor de luz para control de encendido y apagado de lampara

5.1.4 Verificación de registro de eventos

Para la verificación de detección de fallas en la lampara, se verifico todas las alarmas y combinaciones posibles de la fallas que se contemplaron en el diseño, por ello se empleara la tabla contemplada en el diseño de registro de eventos.

No de prueba	Bandera de control Remoto	Estado	Verificador de Fallas		Voltaje [V]		Condición	Código	Prueba
1	ON	ON	ON	√	127	√	Correcto	220	OK
2	ON	ON	OFF	x	127	√	Falla	330	OK
3	ON	ON	OFF	x	0	x	Falla	440	OK
4	ON	OFF	ON	x	127	√	Falla	550	OK
5	ON	OFF	OFF	√	127	√	Correcto	220	OK
6	ON	OFF	OFF	√	0	x	Falla	660	OK
7	OFF	ON	-	-	-	-	Falla	770	OK

Tabla 5.3: Tabla de resultados de pruebas a registro de eventos.

Con estos resultados, se verifica que la operación del verificador de fallas esta operando de manera adecuada respondiendo a las fallas contempladas.

5.2 Pruebas del Sistema de comunicación

En esta segunda sección se evaluó el sistema de comunicación en este caso la transmisión y recepción de datos con la tecnología LoRa. Por lo que para este punto se realiza dos tipos de pruebas el primero la transmisión de datos y el segundo la recepción de datos. La primera prueba de verificación de transmisión de datos consistió el transmitir una cadena de caracteres desde el modulo central y poderlas recibir en otro extremo en otra antena en un modulo estación.

lampara no incida en el sensor de luminosidad.

Como segunda prueba fue verificar en este caso la respuesta a una petición de información al Modulo Estación, por lo que se envía información de la petición y posterior a ello se espera la respuesta del modulo estación, esta respuesta ya esta programada y conocemos su estructura por lo que se tiene que verificar que cumpla con esto.

Prueba de Transmisión y Recepción de datos

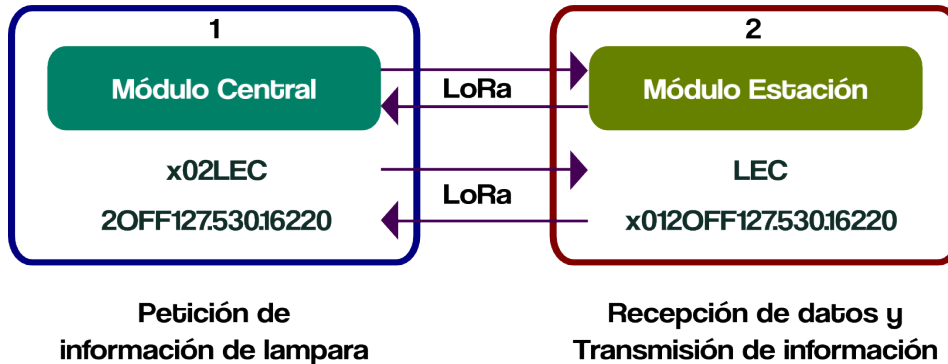


Figura 5.4.: Diagrama de pruebas de Transmisión y Recepción de información.

```

GPIO.setup(23, GPIO.OUT) #ajusta SET
LEC_ID2_v1_5.py:14: RuntimeWarning: This channel is already in use, continuing
s.
GPIO.setup(24, GPIO.OUT) #ajusta SET
["2018-4-19", "6:42", "2", "Inh", "131.04", "00.20", "26.208", "770"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:42", "2", "Inh", "130.28", "00.20", "26.056", "770"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:46", "2", "00H", "131.04", "00.49", "64.2096", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:47", "2", "00H", "130.43", "00.51", "66.5193", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:48", "2", "00H", "131.50", "00.55", "72.325", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:48", "2", "00H", "129.97", "00.58", "75.3826", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:48", "2", "00H", "130.73", "00.62", "81.0526", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:48", "2", "00H", "129.97", "00.60", "77.982", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:49", "2", "00H", "130.28", "00.59", "76.8652", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:49", "2", "00H", "131.50", "00.59", "77.585", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:51", "2", "00H", "129.51", "00.66", "85.4766", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:51", "2", "00H", "129.82", "00.64", "83.0848", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:54", "2", "00H", "130.43", "00.63", "82.1709", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:56", "2", "00H", "130.12", "00.51", "66.3612", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:57", "2", "00H", "129.66", "00.63", "81.6858", "220"]
pi@MC:~$ sudo python LEC_ID2_v1_5.py
["2018-4-19", "6:59", "2", "Inh", "129.51", "00.19", "24.6069", "770"]
pi@MC:~$

```

Figura 5.5.: Respuestas obtenidas a nuestra peticiones de información.

Al final de esta prueba se pudo obtener los datos que se esperaban en la estructura programada, id del modulo estación, estado, consumo de corriente, voltaje de alimentación, potencia de consumo instantáneo y un identificador de eventos.

5.3 Pruebas del sistema Integral

Para probar el sistema de comunicación, se plantearon realizar tres tipos de pruebas:

- Una prueba de distancia probar una distancia considerable, para observar si el sistema opera de manera correcta, con los parámetros que se han configurado las antenas que en este caso se probó a una distancia mayor de 500 metros.
- La segunda prueba que se contempló es medir el tiempo de respuesta que tarda en responder a los módulos de estación a un comando del modulo central, para ello se considera que las antenas están configuradas para trabajar en modo dormido.
- Como tercera prueba es realizar un numero determinado de peticiones (100) y determinar el porcentaje de tramas que tienen perdida de información y así tener una un referencia de la tasa de fiabilidad de transmisión de información del sistema de comunicación..

La primera prueba sobre una prueba de distancia, esta prueba se realizó en la localidad de El Dadho, Chilcuautla, Hidalgo, considerando que es una comunidad rural el espectro radioeléctrico esta considerablemente mas libre que en la ciudad de México. Se realizó en esta localidad ya es donde se contaba con dos sitios que podían proveer alimentación eléctrica a una distancia aproximada de 523.46 metros. Como dato a considerar, como se menciona en el anexo B en el apartado de la configuración de las antenas, el parámetro que determina de alguna manera el alcance de la antena es el factor de propagación (Spread Factor , SF) , que en esta versión de antena no se puede modificar directamente, para la ello se configura con los parámetros RF_Factor y RF_BW (Factor de propagación LoRa y Ancho de banda LoRa)² configurados en 2048 chips y 125 Khz respectivamente, además de que la potencia de salida también es importante para lograr un alcance considerable que se define por Power (Potencia) que es configurado en 7 que significa que se ajusta en la potencia de salida máxima de 20 dBm.

El procedimiento de la prueba fue el siguiente, mediante la interfaz gráfica se hace la solicitud de encendido de apagado de lampara, y que a su vez el modulo central envía en comando respectivo para que el modulo de estación ejecute el comando³.

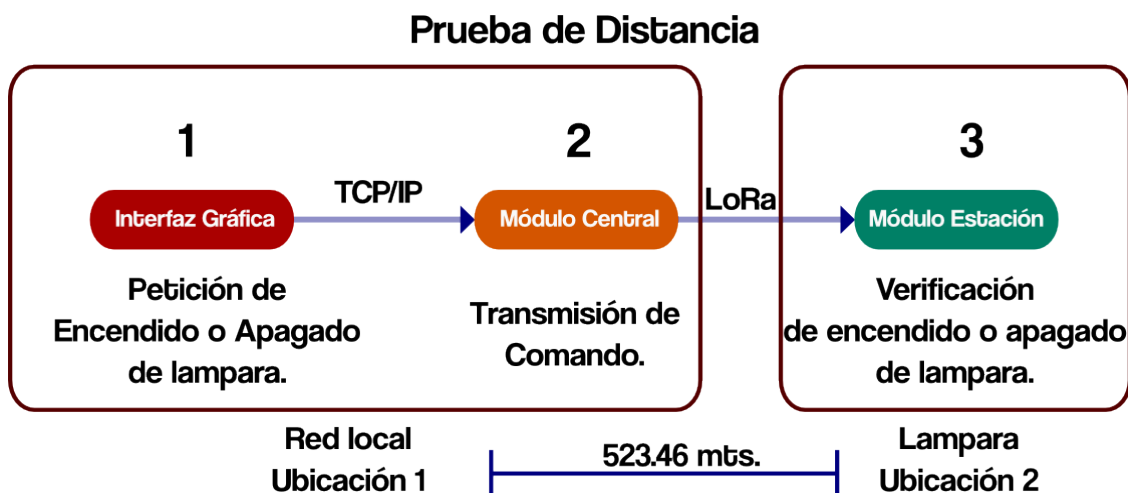


Figura 5.6.: Diagrama de prueba de distancia.

² Un Factor propagación LoRa (RF_Factor) alto significa una alta sensibilidad pero un tiempo de transmisión en el aire mayor. Un ancho de banda lora (RF_BW) significa una baja sensibilidad, es recomendado 125K.

³ En esta prueba se forz a que por default la lampara se encienda, cubriendo la fotoresistencia que mide la cantidad de luz, y poder verificar visualmente el encendido y apagado de la lampara.



Figura 5.7. Prueba de cobertura a una distancia de 523.46 m.

Esta prueba de distancia se puede realizo con dos diferentes distancias en el primer caso fue de 20 metros y la prueba mas larga fue de 523.46 metros, con casi linea de vista directa. Se realizo 10 peticiones y se muestra en la siguiente tabla.

No. de Prueba	Petición	Distancia [mts]	Verificación	Condición
1	Apagado	523.46	Encendido	Incorrecto
2	Apagado	523.46	Apagado	Correcto
3	Encendido	523.46	Encendido	Correcto
4	Apagado	523.46	Apagado	Correcto
5	Encendido	523.46	Encendido	Correcto
6	Apagado	20.00	Apagado	Correcto
7	Encendido	20.00	Encendido	Correcto
8	Apagado	20.00	Apagado	Correcto
9	Encendido	20.00	Encendido	Correcto
10	Apagado	20.00	Apagado	Correcto

Table 5.4. Resultados de prueba de distancia.

Como segunda prueba en esta sección se mide el tiempo de respuesta desde el momento de la petición desde el modulo central, hasta la obtención de la respuesta sea registrada en la base de datos, no se puede obtener el tiempo en el que se imprime en la interfaz gráfica ya que la interfaz gráfica muestra estos datos a petición del usuario y no depende del tiempo de muestreo. Por ello ha que contemplar los tiempos que toma cada dispositivo en operar; las antenas como se configuraron el tiempo de wake-up desde 2 segundos ademas de que hay que contemplar el tiempos transmisión, recepción y decodificación de las peticiones.

Para el tiempo de transmisión contemplaremos la velocidad teórica proporcionada por el ajuste de los

parámetros que es de 537 bps (ajustes RF_factor =2048, RF_BW=125); el numero de bytes a transmitir corresponde a 1 byte de ID, 3 bytes a estado, 6 a voltaje, 4 a corriente, y 3 de eventos. Y dan un total de 18 bytes. Por lo que aritmeticamente da un tiempo de transmisión teórico de 0.2681 segundos

Prueba de tiempo de respuesta a petición de información

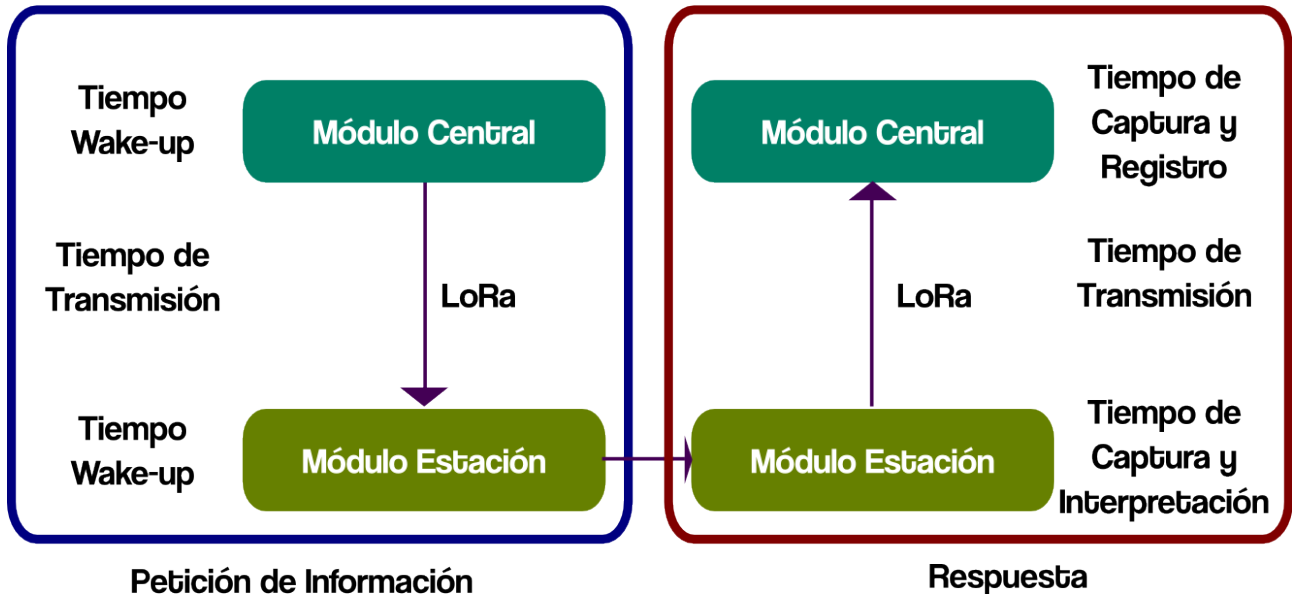


Figura 5.8.: Diagrama de prueba de tiempos de respuesta.

Tiempos	Tiempo teórico aproximado [seg]
Wake-up de modulo central	2
Tiempo de transmisión	0.2681
Wake-up de modulo estación	2
Tiempo de captura e interpretación	2
Tiempo de transmisión	0.2681
Tiempo de captura y registro	Desconocido
Total Aproximado	6.5363

Table 5.5. Tiempos teóricos considerados.

Número de prueba	Tiempo medidos de respuesta [seg]
1	6.649475
2	6.799377
3	6.641472
4	6.674455
5	6.653128
6	6.663602
Total promedio	6.68

Table 5.6. Tiempos medidos de respuesta.

El proceso de Medición se realizo con una función propia de python que se coloco al principio donde se

solicita la petición y el segundo punto se coloca después de cerrar la conexión a la base de dato.

```

pi@MMC: ~/M_Central
[13]
pi@MMC:~/M_Central $ python LEC_ID2_v2_6.py
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1050, "Table 'test1' already exists")
  result = self._query(query)
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1265, "Data truncated for column 'Fecha' at row 1")
  result = self._query(query)
6652.661 ms
pi@MMC:~/M_Central $ ^C
pi@MMC:~/M_Central $ ^C
pi@MMC:~/M_Central $ date
mié may 9 08:21:12 CDT 2018
pi@MMC:~/M_Central $ date
mié may 9 08:21:15 CDT 2018
pi@MMC:~/M_Central $ python LEC_ID2_v2_6.py
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1050, "Table 'test1' already exists")
  result = self._query(query)
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1265, "Data truncated for column 'Fecha' at row 1")
  result = self._query(query)
6799.377 ms
pi@MMC:~/M_Central $ python LEC_ID2_v2_6.py
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1050, "Table 'test1' already exists")
  result = self._query(query)
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1265, "Data truncated for column 'Fecha' at row 1")
  result = self._query(query)
6641.472 ms
pi@MMC:~/M_Central $ date
mié may 9 08:22:05 CDT 2018
pi@MMC:~/M_Central $ python LEC_ID2_v2_6.py
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1050, "Table 'test1' already exists")
  result = self._query(query)
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1265, "Data truncated for column 'Fecha' at row 1")
  result = self._query(query)
6674.455 ms
pi@MMC:~/M_Central $ python LEC_ID2_v2_6.py
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1050, "Table 'test1' already exists")
  result = self._query(query)
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1265, "Data truncated for column 'Fecha' at row 1")
  result = self._query(query)
6653.128 ms
pi@MMC:~/M_Central $ python LEC_ID2_v2_6.py
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1050, "Table 'test1' already exists")
  result = self._query(query)
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1265, "Data truncated for column 'Fecha' at row 1")
  result = self._query(query)
6663.602 ms
pi@MMC:~/M_Central $ date
mié may 9 08:25:28 CDT 2018
pi@MMC:~/M_Central $ python LEC_ID2_v2_6.py
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1050, "Table 'test1' already exists")
  result = self._query(query)
/usr/local/lib/python2.7/dist-packages/pymysql/cursors.py:165: Warning: (1265, "Data truncated for column 'Fecha' at row 1")
  result = self._query(query)
6649.475 ms
pi@MMC:~/M_Central $ █

```

Figura 5.9.: Mediciones de tiempo de respuesta.

Como tercera y ultima prueba es verificar la perdida de paquetes o que un cierta muestra de muestras para ello echaremos mano de la base de datos que tiene la captura de estos datos y verificaremos si alguno es un dato erróneo.

N. Muestra	Fecha y Hora	Identificador	Estado	Voltaje	Corriente	Potencia	Evento
1	2018-05-03 09:22:09	2	Inh	122.02	0.16	19.5232	777
2	2018-05-03 09:36:08	2	OFF	122.63	0.19	23.2997	555
3	2018-05-03 09:39:27	2	OFF	121.41	0.18	21.8538	555
4	2018-05-03 09:40:29	2	OFF	121.56	0.19	23.0964	555
5	2018-05-03 09:41:47	2	OFF	122.48	0.18	22.0464	555
6	2018-05-03 09:42:52	2	OFF	122.17	0.18	21.9906	555
7	2018-05-03 09:43:27	2	OFF	122.17	0.19	23.2123	555
8	2018-05-03 09:43:58	2	OFF	122.63	0.18	22.0734	555
9	2018-05-03 09:44:49	2	OON	122.17	0.63	76.9671	222
10	2018-05-03 09:45:34	2	OON	121.87	0.65	79.2155	222
11	2018-05-03 09:47:00	2	OON	121.87	0.63	76.7781	222
12	2018-05-03 09:47:34	2	OON	122.02	0.66	80.5332	222
13	2018-05-03 09:48:16	2	OON	121.56	0.65	79.014	222

14	2018-05-03 09:51:28	2	OON	122.94	0.66	81.1404	222
15	2018-05-03 09:55:07	2	OFF	122.63	0.17	20.8471	555
16	2018-05-03 10:04:03	2	OFF	121.56	0.18	21.8808	555
17	2018-05-03 10:05:29	2	OFF	121.41	0.17	20.6397	555
18	2018-05-03 10:06:07	2	OFF	121.71	0.16	19.4736	555
19	2018-05-03 10:08:30	2	OFF	122.02	0.17	20.7434	555
20	2018-05-03 10:11:56	2	OFF	121.71	0.18	21.9078	555
21	2018-05-03 10:13:34	2	OFF	121.87	0.18	21.9366	555
22	2018-05-03 10:15:26	2	OFF	122.17	0.17	20.7689	555
23	2018-05-03 10:21:52	2	OON	122.02	0.64	78.0928	222
24	2018-05-03 10:24:27	2	OON	121.41	0.66	80.1306	222
25	2018-05-03 10:26:04	2	OON	122.48	0.64	78.3872	222
26	2018-05-03 10:28:09	2	OON	121.87	0.64	77.9968	222
27	2018-05-03 10:34:48	2	OON	121.56	0.65	79.014	222
28	2018-05-03 10:36:08	2	OON	121.25	0.67	81.2375	222
29	2018-05-03 10:37:14	2	OFF	122.17	0.17	20.7689	555
30	2018-05-03 10:50:50	2	OFF	121.25	0.18	21.825	555
31	2018-05-03 11:04:23	2	OFF	122.17	0.2	24.434	555
32	2018-05-03 11:04:41	2	OFF	122.48	0.18	22.0464	555
33	2018-05-03 11:04:55	2	OFF	123.24	0.17	20.9508	555
34	2018-05-03 11:06:21	2	OFF	122.32	0.17	20.7944	555
35	2018-05-03 13:10:07	2	OFF	121.25	0.18	21.825	555
36	2018-05-03 13:38:51	2	OFF	122.63	0.18	22.0734	555
37	2018-05-03 13:48:27	2	OFF	122.48	0.17	20.8216	555
38	2018-05-03 14:23:33	2	OON	121.71	0.64	77.8944	222
39	2018-05-03 14:23:58	2	OON	122.17	0.65	79.4105	222
40	2018-05-03 14:27:07	2	OON	121.71	0.63	76.6773	222
41	2018-05-03 14:28:20	2	OON	122.02	0.63	76.8726	222
42	2018-05-03 14:28:32	2	OON	122.17	0.65	79.4105	222
43	2018-05-03 14:32:28	2	Inh	122.17	0.18	21.9906	777
44	2018-05-03 14:33:13	2	Inh	122.48	0.17	20.8216	777
45	2018-05-03 14:53:39	2	Inh	123.85	0.19	23.5315	777
46	2018-05-03 15:00:26	2	OON	123.09	0.64	78.7776	222
47	2018-05-03 15:09:14	2	OON	124.01	0.66	81.8466	222
48	2018-05-03 15:29:04	2	OON	123.24	0.64	78.8736	222
49	2018-05-03 15:39:28	2	Inh	123.7	0.17	21.029	777
50	2018-05-03 15:43:31	2	Inh	124.47	0.19	23.6493	222
51	2018-05-03 15:44:30	2	Inh	123.7	0.17	21.029	222
52	2018-05-03 17:44:08	2	Inh	123.09	0.17	20.9253	777
53	2018-05-03 17:49:04	2	Inh	121.25	0.17	20.6125	777
54	2018-05-03 17:53:28	2	Inh	121.25	0.17	20.6125	222
55	2018-05-03 18:04:13	2	OON	121.71	0.64	77.8944	222
56	2018-05-03 18:13:59	2	OON	120.95	0.64	77.408	222
57	2018-05-03 18:34:22	2	OON	122.17	0.64	78.1888	222
58	2018-05-03 18:59:55	2	OON	121.41	0.65	78.9165	222
59	2018-05-03 19:11:43	2	Inh	121.71	0.17	20.6907	222
60	2018-05-03 19:50:42	2	Inh	123.09	0.19	23.3871	222

61	2018-05-03 22:36:06	2	Inh	123.55	0.17	21.0035	222
62	2018-05-03 22:43:49	2	Inh	123.24	0.17	20.9508	222
63	2018-05-03 23:12:40	2	Inh	123.55	0.17	21.0035	222
64	2018-05-04 12:44:52	2	Inh	121.41	0.18	21.8538	777
65	2018-05-04 12:45:25	2	Inh	121.71	0.17	20.6907	777
66	2018-05-04 12:46:08	2	Inh	120.64	0.19	22.9216	777
67	2018-05-04 12:48:34	2	Inh	121.25	0.17	20.6125	777
68	2018-05-09 07:00:09	2	Inh	121.56	0	0	777
69	2018-05-09 07:15:09	2	Inh	121.56	0.02	2.4312	777
70	2018-05-09 07:30:09	2	OFF	121.71	0.01	1.2171	555
71	2018-05-09 07:45:09	2	OFF	120.34	0.02	2.4068	555
72	2018-05-09 08:00:09	2	OFF	120.8	0.01	1.208	222
73	2018-05-09 08:15:09	2	OFF	121.25	0.02	2.425	222
74	2018-05-09 08:20:52	2	OFF	121.1	0.02	2.422	222
75	2018-05-09 08:21:27	2	OFF	121.1	0.03	3.633	222
76	2018-05-09 08:21:52	2	OFF	121.25	0.01	1.2125	222
77	2018-05-09 08:22:16	2	OFF	120.95	0.01	1.2095	222
78	2018-05-09 08:22:37	2	OFF	121.87	0.01	1.2187	222
79	2018-05-09 08:23:01	2	OFF	121.25	0	0	222
80	2018-05-09 08:25:39	2	OFF	120.8	0.02	2.416	222
81	2018-05-09 08:30:09	2	OFF	120.8	0.02	2.416	222
82	2018-05-09 08:45:09	2	OFF	120.8	0.01	1.208	222
83	2018-05-09 09:00:09	2	OON	120.8	0.11	13.288	222
84	2018-05-09 09:15:09	2	OON	120.64	0.12	14.4768	222
85	2018-05-09 09:30:09	2	OON	120.18	0.12	14.4216	222
86	2018-05-09 09:45:09	2	OON	120.8	0.13	15.704	222
87	2018-05-09 09:56:30	2	OFF	120.34	0	0	222
88	2018-05-09 09:57:09	2	OFF	121.1	0.03	3.633	222
89	2018-05-09 09:59:58	2	OFF	122.02	0.02	2.4404	222
90	2018-05-09 10:00:09	2	OFF	122.17	0.02	2.4434	222
91	2018-05-09 10:00:56	2	OFF	120.8	0.01	1.208	222
92	2018-05-09 10:01:12	2	OFF	120.64	0.04	4.8256	222
93	2018-05-09 10:01:32	2	OFF	120.8	0.03	3.624	222
94	2018-05-09 10:02:06	2	OFF	120.95	0.01	1.2095	222
95	2018-05-09 10:03:03	2	OFF	121.1	0.02	2.422	222
96	2018-05-09 10:03:28	2	OFF	121.25	0.03	3.6375	222
97	2018-05-09 10:04:20	2	OFF	122.32	0.02	2.4464	222
98	2018-05-09 10:05:01	2	OON	121.41	0.09	10.9269	222
99	2018-05-09 10:05:36	2	OON	121.1	0.11	13.321	222
100	2018-05-09 10:05:52	2	OON	121.71	0.1	12.171	222

Table 5.7. Muestreo de 100 muestras de datos recopilados.

Des esta muestra de datos que se estuvieron transmitiendo no se presenta ninguna perdida de datos por que la tasa de perdida de datos puede ser mas pequeña tendría que revisarse en otra muestra mas grande para obtener un dato fiable ero con esto nos percatamos que la perdida de datos es menor a 1%.

Capitulo 6 Conclusiones y Trabajos a Futuro.

6.1 Conclusiones

Para cerrar la descripción de este proyecto se comenta las siguientes conclusiones, como se pudo observar este proyecto consta de tres principales secciones; ya que esta enfocado en la tecnología de Internet de las cosas, la comunicación es impredecible pero cada una de las secciones es tan importante como la otra, y cada una requiere un trabajo meticuloso, para emplear las mejores metodologías para las distintas aplicaciones y secciones de un proyecto como este. Y de esta manera poder aplicar cada una de estas secciones a otras aplicaciones. Pero hay que recordar que dependiendo de la aplicación va a ser el tipo de tecnología mas adecuada.

Otro punto que se puede observar es que las opciones de medición realizadas con arduino, no son muy exactas, por ello se recomienda analizar el uso de otras metodologías más precisa, para la mediciones de voltaje y corriente. Además de hay que determinar la periodicidad de muestreo de mediciones.

Otro punto a abordar es que este prototipo fue desarrollado con módulos estándar que se puede adquirir en el mercado, la reducción del circuito eléctrico es posible, tanto en el microcontrolador, como en su sección de control y comunicación para el modulo de Estación. Para la modulo Central es posible conectarlo a una computadora y mediante un puerto serial, adquirir los datos de los distintos módulos estación. Esto se puede lograr con un adaptador PL2303HXD.

Como conclusión de la sección de comunicación si bien es cierto que existe un protocolo de transmisión de datos llamado LoRaWAN, este protocolo esta en desarrollo y no ha llegado a una versión final por lo que esta en constante cambio, por ello es recomendable desarrollar tu propio trama de comunicación de datos, para una aplicación privada, en un futuro es susceptible a que este protocolo sea regulado y es posible que se reduzca la longitud de la trama de datos. Ya que se tiene una tendencia de limitar el numero de bytes a transmitir además de limitar el numero de peticiones por día.

Durante el desarrollo del interfaz gráfico, se observo que php requiere de la intervención de un usuario para poder actualizar los datos y realmente no existe una percepción de operación en tiempo real. Otra alternativa que se conoció fue la plataforma Node.js, express, Mongo DB y Angular, que por características permite la percepción de un control en tiempo real, ya que estas herramientas parten de la base de javascript. Y parece mas adecuado para el desarrollo de una aplicación de este tipo. Y quedará como trabajo a futuro el desarrollo de una aplicación en javascript usando esta tecnología.

6.2 Trabajos a Futuro.

- Para este tipo de aplicaciones tiene un sinnúmero de posibilidades, concentrándonos en el alumbrado publico, y primer punto que queda como trabajo futuro, reducir las conexiones eléctricas a un circuito mas pequeño.
- Queda como trabajo futuro considerar que metodologías medición,son los mas adecuado para este tipo de aplicaciones.
- Otro de los puntos importantes es determinar si es necesario implementar el control de la iluminación ya que existe una infinidad de opciones en mercado actualmente que realiza el control de luminosidad. En caso de implementarla seleccionar la mejor opción.
- Si bien es cierto que el uso del un dispositivo GPS, contrapone al poco consumo de energía para la transmisión de datos. Es necesario para poder conocer la ubicación de cada una de las luminarias.

- Este tipo de aplicaciones como se menciona anteriormente ofrece muchas posibilidades, entre ellas, ya que las luminarias están distribuidas por toda una ciudad, se pueden agregar medidores de calidad del aire, sensores temperatura, humedad , etc. y poder tener un mayor conocimiento de nuestro entorno.
- Trabajando con las antenas DRF1276DM de Dorji, se observo que en ocasiones es necesario realizar una configuración local de una antena, existe la posibilidad de poder desarrollar un librería que permita cambiar la configuración de una antena en tiempo real.
- Como se dijo en la sección de conclusiones, es factible desarrollar una aplicación en Node.js y javascript para la un percepción de control en tiempo real.
- Con esta cantidad de datos puede ser posible desarrollar una aplicación que permita eficiente en uso del alumbrado publico.
- Por ultimo no hay que recordar que la seguridad es importante y es necesario para este tipo de aplicaciones en el que se puede controla dispositivos y datos del comportamiento de un sistema almacenaran datos .

6.3 Resumen para Implementación de una aplicación similar

Como ultima tarea de este ejercicio de desarrollo de una implementación es obligado hacer un resumen de pasos de un desarrollador debe seguir para implementar una aplicación similar:



Figura 6.1.: Diagrama de resumen de puntos a revisión en la implementación de un proyecto similar con tecnología LoRa.

Apéndice A Dispositivos empleados

- A.1 Fotorresistencia
- A.2 Sensor de Voltaje
- A.3 Sensor de Corriente
- A.4 Raspberry
- A.5 Arduino Uno
- A.6 Diagrama de Modulo Shield Estación
- A.7 Modulo de Potencia
- A.8 Antenas Inalambricas DRF1276DM-DORJI
- A.9 DS3231SN
- A.10 Diagrama de Modulo Shield Central

A.1 Fotorresistencia

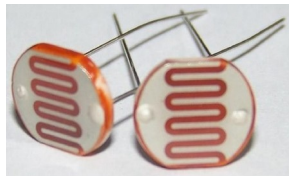


Figura A.1: Fotorresistencia de 2M.

El LDR (Light Dependet Resistor) o fotorresistencia es un componente electrónico que varia su resistencia en función de la luz que incide sobre sus superficie. Cuanto mayor es la intensidad de la luz que incide en la superficie de la fotorresistencia, menor será su resistencia y cuanto menos luz incida mayor será su resistencia. Los valores típicos varían entre 1 M o más, en la oscuridad y 100Ω con luz brillante.

Como características principales se puede decir que tiene una disipación máxima de 50m W a 1W. Soporta un voltaje máximo de 600V. A demás que el tiempo de respuesta típico de un LDR esta en el orden de una décima de segundo.

El uso de este dispositivo provee de ventajas como el bajo costo, el fácil empleo y la lata relación resistencia luz-oscuridad. Pero también tiene desventajas como una respuesta espectral estrecha, efecto histerisis, respuesta lenta en materiales estables, estabilidad por temperatura baja para los materiales más rápidos.

A.2 Sensor de Voltaje (FZ0430)

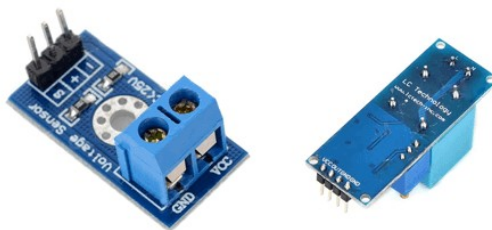


Figura A.2 Modulo FZ0430, sensor de voltaje.

El FZ0430 es módulo comercial que permite medir tensiones de hasta 25V de forma sencilla con un procesador como arduino. Consiste en un simple divisor de tensión con resistencia de $30 k\Omega$ y $7.5 k\Omega$, lo

que supone que la tensión percibida tras el módulo sea dividida por un factor de $5 \left(\frac{7.5}{30+7.5} \right)$.

La tensión máxima que se puede medir es de 25v para un procesador de tensión de alimentación Vcc de

5v, y 16.5v para un procesador de Vcc 3.3v. Superar esta tensión en la entrada de FZ0430 dará el pin analógico de arduino en este caso. En caso de emplear una carga de de 25v, la corriente que atraviesa el divisor es de 0.7mA, y las pérdidas del divisor alcanza 16.67mW.

Ya que este módulo consta de un simple arreglo de divisor de voltaje puede ser fácilmente sustituido, donde se puede ajustar los valores del divisor para una mejor precisión en la aplicación que se esta desarrollando. Pero para nuestra aplicación fue empleada este modulo por que ademas de ser un módulo estándar para arduino, la construcción de la misma y los bornes que facilitan su conexión fueron las razones por lo que empleo este módulo. Además de que se puede conseguir fácilmente en cualquier tienda de electrónica. Por otro lado la conexión eléctrica se realiza de manera muy sencilla mediante sus terminales disponibles; conectando GND y SIG del FZ0430 a GND y a una entrada analógica de Arduino, respectivamente.



Figura A.3: Distribución de pines de FZ0430.

A.3 Sensor de Corriente (ACS712)



Figura A.4: Modulo ACS712, Sensor de Corriente.

El módulo ACS712 es un sensor de corriente tanto alterna como continua, permite medir la intensidad de corriente que atraviesa un conductor. Este módulo ACS712 se puede emplear con un procesador como arduino para medir la intensidad de corriente o la potencia consumida por la carga.

Este módulo consiste en un sensor de efecto hall de precisión y bajo offset junto con un canal de conducción localizado cerca de la superficie del integrado cuando la corriente fluye por el canal de cobre genera un campo magnético que es detectado por el sensor de efecto Hall y es convertido en tensión. Esta tensión de salida es proporcional a la corriente, y altamente independiente a la temperatura.

Este sensor viene calibrado de fabrica, pero para una medición de precisión hace falta un ajuste de la calibración. Además de que la resistencia de este módulo es muy baja, 1.2 mΩ lo que se traduce en pequeñas pérdidas.

Existen varios modelos para distintos rangos de consumo de 5, 20, 30 amperes; ya que las lamparas empleadas en este proyecto no superan los 60 Watts, empleo un modelo de 5 amperes. El sensor soporta corrientes hasta 5x el rango de medición del sensor. Sobre pasar este limite dañará el módulo.

La mayor desventaja del módulo ACS 712 es que es un sensor intrusivo, lo que significa que es necesario

cortar el conductor para insertarlo. Es una gran desventaja frente a otros sensores no intrusivos. Este módulo se puede adquirir en cualquier tienda de electrónica y se pueden encontrar en las distintas versiones de ACS712 5A, 20A y 30 A preparados para conectar de forma sencilla a un procesador como arduino.

ACS712 proporciona una salida de tensión proporcional a la intensidad de corriente que atraviesa el sensor al se identifica como OUT en el modulo. Para hacer funcionar este modulo es necesario una alimentación de 5 Volts, por lo que es necesario una conexión de GND y Vcc a GND y 5V, respectivamente. La salida del sensor es necesario su conexiona a una entrada analógica.



Figura A.5: Distribución de pines de ACS712.

La tensión de salida del ACS712 es proporcional a la intensidad que atraviesa al sensor. Se incluye un offset de 2.5 V de forma que la referencia está centrada, lo que permite medir intensidades positivas y negativas.

$$V = 2.5 + K * I \Rightarrow I = (V - 2.5) / K$$

El factor de proporcionalidad y la resolución dependen den modelos del sensor para rango de 5, 20 y 30 A.

Max Intensidad [A]	Sensibilidad [mV/A]	Tensión salida [V]	Resolución [mA]
± 5	185	1.575 - 3.425	26
± 20	100	0.5 – 4.5	49
± 30	66	0.52 – 4.48	74

Tabla A.1. de parámetros de las distintas versiones del módulo ACS712.

A.4 Raspberry Pi 3 Modelo B

La Raspberry Pi es una plataforma que ofrece características y capacidades para funcionar como módulo para desarrollar prototipos de proyectos. La elección del empleo de este dispositivo para el desarrollo del modulo central fue por las características propias de la Raspberry, a demás de la fácil adquisición, el bajo costo que conlleva y gran comunidad de existe sobre este dispositivo.

Una de sus principales desventajas es que, si bien este dispositivo tiene implementado dos puertos uart, uno de tipo PL011 (PrimeCell UART) y un mini UART. El primero se encuentra conectado al modulo de bluetooth y el segundo se encuentra disponible para su empleo para realizar proyectos, pero este puerto UART, su velocidad en baudios esta vinculada a la frecuencia central de la VPU GPU VC4. Esto significa que como el gobernador de la frecuencia VPU varía la frecuencia del núcleo, por lo tanto también la velocidad en baudios del puerto UART esto hace que el miniUART sea limitado en el estado predeterminado. Esto hace necesario para no tener el problema de la variación de la velocidad, de deshabilitar el modulo de bluetooth para hacerlo disponible para los diferentes proyectos en este caso para

el modulo central, procedimiento que se encuentra disponible gracias a la comunidad, que es descrito en el anexo B .

Las especificaciones que favorecen el uso la Raspberry son las siguientes:

- Procesador: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- Memoria: 1GB LPDDR2 SDRAM
- Conectividad:
 - 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
 - Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps), Full-size HDMI
 - 4 USB 2.0 ports
- Accesibilidad: Extended 40-pin GPIO header
- Video y sonido:
 - 1 full size HDMI
 - MIPI DSI display port
 - MIPI CSI camera port
 - 4 pole stereo output and composite video port
- Multimedia: H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
- Uso de SD card: Formato Micro SD para cargar sistema operativo y almacenamiento de datos.
- Alimentación:
 - 5 V/2.5 A DC via conector micro USB
 - 5 V DC via header GPIO
 - Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
- Temperatura de Operación: 0 - 50 °C
- Vida útil de producción: La Raspberry Pi 3 Model B+ se mantendrá en producción hasta por lo menos Enero de 2023.

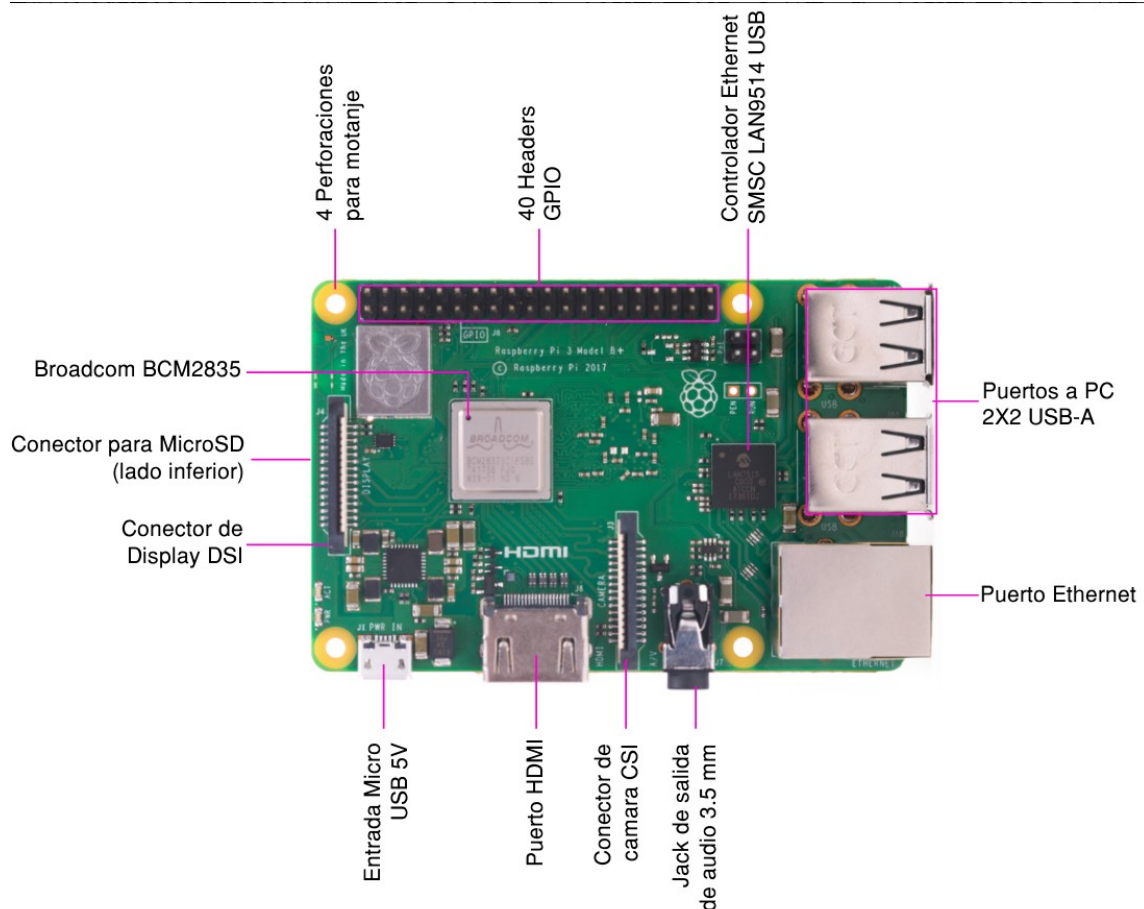


Figura A.6: Vista general de Raspberry Pi 3 Modelo.

A.5 Arduino Uno

Arduino Uno es una placa de microcontrolador basada en Atmega 328P. Tiene 14 pines digitales de entrada/salida (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 Mhz, una conexión USB, una conector de alimentación, un header ICSP y un boton de reset. Contiene todo lo necesario para soportar el microcontrolador; un rápida conexión a una computadora se logra con un cable USB. La siguiente tabla describe algunas de sus especificaciones tecnicas:

Microcontrolador	ATmega328P
Voltaje de operación	5V
Voltaje de Entrada (recomendado)	7-12V
Voltaje de Entrada (limite)	6-20V
Pines Digitales I/O	14 (6 provistos de salidas PWM)
Pines PWM-Digital I/O	6
Pines de Entrada Analogica	6
DC Corriente por I/O Pin	20 mA
DC Corriente por 3.3V Pin	50 mA
Memoria Flash	32 KB (ATmega328P), 0.5 KB usado para bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Frecuencia de reloj	16 MHz
LED_BUILTIN	13
Longitud	68.6 mm
Ancho	53.4 mm
Peso	25 g

Tabla A.2. Especificaciones técnicas de Arduino.

La selección de este dispositivo ademas de su bajo costo, por el gran soporte y fácil adquisición, se vuelve una herramienta fiable de trabajo para desarrollar este tipo de prototipos. Ademas de que su curva de aprendizaje es muy rápida.

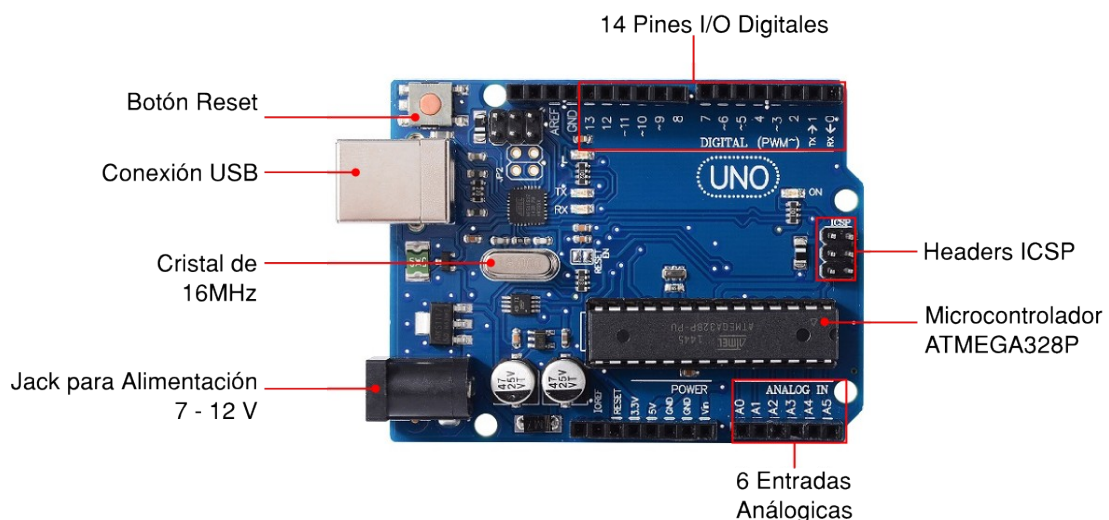


Figura A.7: Vista general de Arduino Uno Rev.3

A.7 Modulo de Potencia

Para el modulo de Potencia del Módulo de Estación, se analizaron los distintos métodos para lograr el control de la lampara que se encuentra a un nivel de tensión relativamente alto. Ya que la salida de control de arduino es de encendido y apagado, por fácil implementación. Se optó por implementar el control por medio de MOC 3011 y BTA 16. Implementado el circuito sugerido en la hoja de datos del propio MOC3011 para carga resistiva.

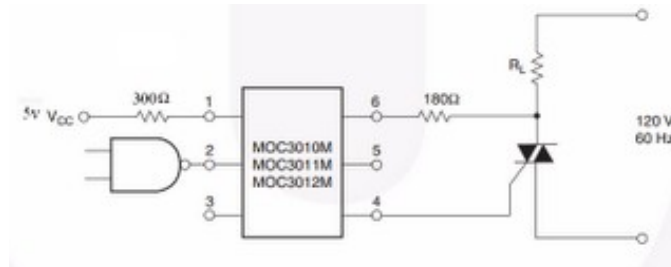


Figura A.9: Circuito para carga Resistiva.

El MOC es un optoacoplador que esta fabricado con un diodo infrarrojo emisor de arseniuro de galio acoplado a un switch de silicio activado por luz; este switch funciona como un triac que activa triacs de potencia y que de alguna manera separa el circuito de baja tensión del de alta tensión(120 V), y que el arduino no puede controlar directamente al TRIAC. El MOC tiene dos configuraciones para dos tipos de cargas resistiva e inductiva. Es importante recordar que el MOC 3011 no tiene detención de punto cero, por lo que si en algún punto se cambiar la metodología de control será necesario cambiar de MOC.

Las especificaciones electricas del MOC 3011 son las siguientes:

- Dispositivo acoplado ópticamente
- .Alto aislamiento: 7500 V pico.
- Salida del dispositivo diseñado para: 115 Vac.
- 6 pin DIP.
- Compatible con MOC3009, MOC3010, MOC3011, y MOC3012.
- Rango de temperatura: -40 a +100 grados centigrados
- Potencia máxima de disipación: 300mW

A.8 Antenas Inalambricas DRF1276DM-DORJI

Radio Módem de datos SX1278 de Largo Alcance LORA.

A.8.1 Características:

- Espectro de frecuencia LoRa
- Banda de frecuencia ISM 868/915 Mhz
- Sensibilidad de recepción -138 dBm
- Potencia de salida Máxima 20 dBm
- Puerto serial activo
- Wireless activo
- Capacidad de conexión en estrella
- Voltaje de alimentación 3.4~5.5 V

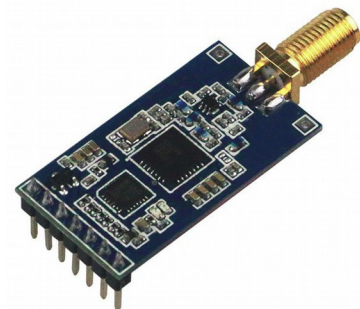


Figura A.10: Módulo DRF1276DM.

A.8.2 Aplicaciones

- Automatización residencial
- Alarma de seguridad
- Telemetría
- Lector automático de medidores.
- Acceso sin contacto
- Registrador de datos Wireless
- Control Remoto de Motores
- Red de sensores Wireless

A.8.3 Descripción

DRF1276DM es un tipo de radio módem de baja tasa de datos de largo alcance, basado en SX1276 de Semtech, DRF1276DM es un modulo de transceptor sub-1 Ghz de bajo costo para operaciones ISM(Industrial Scientific Medical) sin licencia y bandas LPRD. La Modulación/demodulación en espectro de frecuencia, la operación multi-canal, la alta eficiencia de ancho de banda y un rendimiento de anti-bloqueo hace que los módulos DRF1276DM sean fácil para realizar enlaces wireless robustos y fiables.

El modulo puede trabajar en dos diferentes modos: El modo Estándar y el modo Red en Estrella. En el modo Estándar DRF1276DM actúa como un radio módem de datos transparente, el cual se comunica con el host en el formato de datos presentado sin la necesidad de codificación o decodificación. En el Modo de red en estrella, un modulo deberá ser configurado como modulo central y los demás módulos deberán ser configurados como módulos nodo. La comunicación entre el modulo central y módulos nodos es bidireccional pero los módulos nodos no pueden comunicarse con cualquier otro.

A.8.4 Funciones de pines

PIN	DIP-A	Función	Descripción
1	GND	Tierra	Tierra (0V)
2	VCC	Voltaje de Entrada	Voltaje de Entrada
3	EN	Entrada	Pin de habilitación (Enable Pin)
4	RXD	Entrada	RXD: Entrada UART, nivel TTL
5	TXD	Salida	TXD: Salida UART, nivel TTL
6	AUX	Salida	Indicación I/O de datos del interfaz UART
7	SET	Entrada	Pin de Control, Efectivo en modo nodo/central

Tabla A.3. Funciones de pines de DRF1276DM.

Nota:

- Para versiones de firmware < V2.0, el pin SET es reservado sin una función en especifica. Para firmware versión >= V2.0, el nivel lógico del pin SET es usado para cambiar entre comunicación express(velocidad normal de comunicación) y comunicación pausada en modo central/nodo. En el estado de suspensión, el pin SET debe ser conectado a un valor lógico alto o flotante.
- Si el módulo es conectado a un microcontrolador, el modulo enviará una cadena de caracteres al microcontrolador cuando sea energizado

9600 N 8 1 DRF127xDM V2.7

Los últimos caracteres alfanuméricos V2.7 indican que el firmware es 2.7. Si los usuarios conectan un adaptador USB DAC02 a una computadora y ejecutar una herramienta serial como Advanced Serial Port Monitor, el modulo también enviará un cadena de caracteres a la herramienta tan pronto como sea conectado en el adaptador USB.

- Con el fin de configurar los parámetros, el pin EN debe ser conectado a un nivel lógico bajo en cualquier modo. En el modo estándar, el pin EN también debe ser conectado a un nivel lógico bajo para permitir a los módulos trabajar en rendimiento pleno. Si EN es conectado a nivel lógico alto en este modo, el RFIC y MCU de los módulos estarán en modo suspensión profunda y solo podrán ser habilitados para ser ajustados con este PIN con un valor lógico bajo. En modo central el Pin EN también necesita estar en un nivel lógico bajo así como en el modo estándar. En modo nodo, el modulo monitoreará la señal wireless en un intervalo de tiempo y se cerrará el interfaz UART para ahorrar energía cuando el pin EN es ajustado a un nivel lógico alto. Si el modulo en modo nodo necesita enviar datos al modulo central, los usuarios pueden ajustar EN a un nivel lógico bajo y el modulo nodo habilitará el interfaz UART para recibir datos del microcontrolador.

A.8.5 Especificaciones Eléctricas

Símbolo	Parámetro	Min.	Típico	Max.	Unidades
VCC	Voltaje de Alimentación	3.4		5.5	V
Temp	Rango de temperatura de Operación	-40	25	80	°C
RH	Rango Humedad Relativa	10		90	%
Freq	Rango de Frecuencia	868		930	MHz
Mod	Tipo de Modulación		FS		
I _R	Corriente de recepción en modo Estandar Modo Transmisión @20dBm Modo Dormido		15.2 75 3.9		mA mA μA
P _{out}	Potencia Máxima de Salida @860 MHz @900 Mhz @920 Mhz @930 MHz		19 19 18.5 18.2	20	dBm
Sen	Sensibilidad de Recepción			-138	dBm
DR _{IN}	Tasa de Datos del Puerto UART	1.2	9.6	57.6	Kbps
T _R	Tiempo de levantamiento Inalámbrico	2		10	s
Z _{ANT}	Impedancia de Antena		50		Ohms

Tabla A.4. Especificaciones eléctricas de DRF1276DM.

A.8.6 Rangos de Valores Máximos

Símbolo	Parámetro	Min.	Max.	Unidades
VCC	Voltaje de Alimentación	-0.3	5.5	V
VI	Voltaje de Entrada	-0.3	3.3	V
VO	Voltaje de Salida	-0.3	3.3	V
TST	Temperatura de Almacenamiento	-55	125	°C

Tabla A.5. Valores máximos DRF1276D.

A.8.7 Modos de Trabajo

A.8.7.1 Modo Estándar

El modo estándar es también llamado modo transparente en el cual el modulo de recibe o transmite datos al host a través de un puerto serial (UART) en el formato de datos presentado y los usuarios no necesitan tener cuidado con el procesamiento interno de datos del modulo. En este modo el pin EN debe ser ajustado a un nivel lógico bajo en caso contrario el modulo entrará en modo suspensión profunda. El pin SET no funciona en este modo por lo que los usuarios pueden ajustarlo en nivel lógico alto.

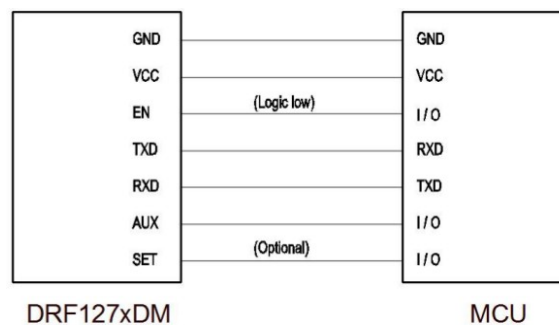


Figura A.11: Conexiones de Pines en Modo Estándar.

El pin AUX de DRF1276DM dará la indicación sobre la entrada y salida de datos en el interfaz AURT con 2 ms de antelación por lo que los usuarios pueden usar esta característica para informar al host sobre los datos. Esto es opcional para la aplicación.

A.8.7.2 Modo Conexión en red estrella

En este modo un modulo DRF1276DM debe ser ajustado como modulo central y los otros módulos deben ser módulos nodos para un red en estrella. Para el modulo central, trabajará a pleno rendimiento por lo que su consumo de energía es similar en el modo estándar y el pin EN debe ser conectado a un valor lógico bajo. El nivel lógico del pin SET para de modulo central debe ser el mismo que el modulo nodo.

El pin SET es usado en modo central/nodo para intercambiar la forma de comunicación (comunicación rápida o comunicación pausada) del modulo nodo. Cuando el pin EN y SET del modulo nodo son ajustados a un nivel lógico alto o flotante el modulo nodo trabajará en forma de comunicación pausada. Permitirá un

intervalo de tiempo y gastará un corto tiempo (para habilitar el parámetro del temporizador) para monitorear la señal wireless por lo que el modulo central necesita enviar un preámbulo largo(>retardo de comunicación), en orden para asegurar el modulo nodo para recibir exitosamente los datos.

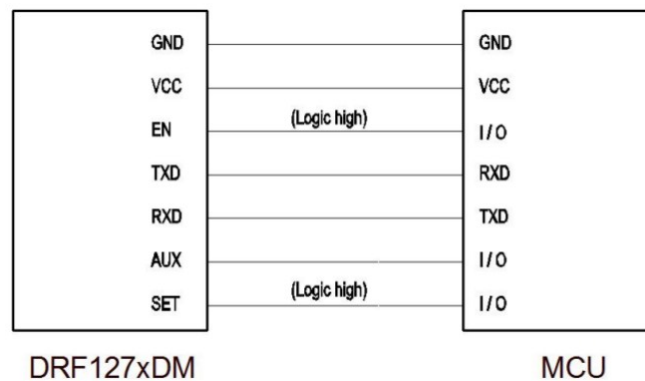


Figura A.12: Modulo Nodo en comunicación pausada.

El parámetro temporizador de levantamiento es introducido en la versión 2.7 en orden de proveer una mayor flexibilidad en el consumo de energía para comunicación pausada. El tiempo de levantamiento se refiere a la longitud de tiempo para que el modulo nodo detecte la señal wireless cuando el modulo nodo es levantado en el periodo de retardo. Un tiempo corto significa menos energía consumida pero es necesario el modulo central para enviar datos con alta relación de transferencia de datos en RF. En otras palabras un periodo de retardo corto significa menos consumo de energía pero requiere una distancia corta para una alta tasa de transferencia de datos.

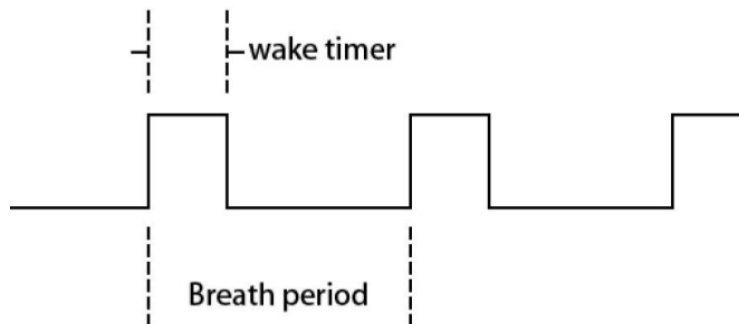


Figura A.13: Tiempo de levantamiento (waker time) y Periodo de Pausado (Breath Period).

Con el fin de reducir la complejidad del calculo de tiempo de levantamiento con la taza de transferencia de datos, el modulo nodo seleccionará automáticamente el parámetro RF_BW y RF_Factor disponible de acuerdo con el parámetro temporizador de levantamiento. Hay dos parámetros RF_BW y RF_Factor no funcionan en este modo y los usuarios solo necesitan ajustar el parámetro de tiempo de levantamiento acuerdo a la aplicación.

Tiempo de levantamiento	Tasa de datos RF	Sensibilidad
64 ms	5.47 kbps	-123 dBm
32 ms	3.125 kbps	-136 dBm
16 ms	1.758 kbps	-129 dBm
8 ms	0.977 kbps	-132 dBm
4 ms	0.537 kbps	-135 dBm
2 ms	0.293 kbps	-138 dBm

Tabla A.6. Tiempo de levantamiento, tasa de datos RF y Sensibilidad.

Para el modulo nodo, si el pin EN y SET del modulo son ajustados a un nivel lógico bajo, el modulo trabajará a pleno rendimiento por lo que el modulo central no necesitará enviar un preámbulo largo. Esto por que es llamado comunicación rápida. La desventaja de esta manera es que el modulo nodo consumirá mucho mas corriente. No hay forma en la cual el nivel lógico del pin SET del modulo central y modulo nodo deba se configurado para el mismo caso el modulo central enviará una longitud errónea de preámbulo, el cual resulta en en perdida de datos. *Por favor notar que el cambio del nivel lógico del pin SET debe realizarse después de ajustar a un nivel lógico bajo el pin EN.*

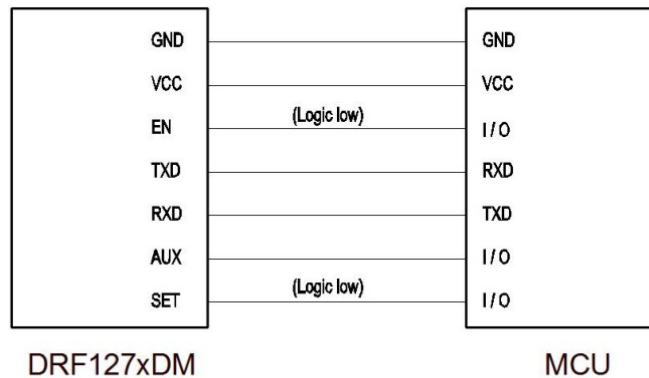


Figura A.14: Modulo Nodo en Comunicación Express

En modo de red en estrella los primero dos bytes del paquete de datos el cual el modulo central envía al modulo nodo debe ser el nodo_ID de la tarjeta modulo nodo. Cuando el modulo nodo detecta la señal efectiva wireless del modulo central, revisará los dos primeros bytes del paquete de datos y los compara con los propios. Acorde al valor de la información del nodo_ID, en el paquete de datos, el modulo nodo procesará de diferentes maneras. Asumiendo el modulo central envía una cadena de datos : 0x00 0x01 0x02 0x03 0x04 0x05 (los primeros dos bytes deben ser el nodo ID de la tarjeta).

- Si el nodo ID del modulo nodo es 0x00 0x00, cuando recibe el paquete de datos, transferirá la cadena completa [0x00 0x01 0x02 0x03 0x04 0x05]al host (microcontrolador o PC) y no filtrará la información ID del paquete de datos.
- Si el nodo ID del modulo nodo es 0x00 0x01, cuando recibe el paquete de datos, primero comparará los dos primeros bytes del paquete de datos y sabrá que los datos son para ese modulo. El modulo filtrará los primeros dos bytes de información ID y transferirá los datos reales de la

cadena [0x02 0x03 0x04 0x05] al host.

- Si el nodo ID del modulo nodo esta entre 0x00 0x02 y 0xFF 0xFF, cuando sea recibidos por el modulo el paquete de datos, encontrará que ID es diferente del propio, por lo que filtrará completamente el paquete de datos, y el host no obtendrá ningún dato.

El valor ID en el paquete de datos enviado al modulo central puede ser 0x00 0x00 ~ 0xFF 0xFF, donde 0xFF 0xFF es un valor ID especial el cual puede funcionar como comando de broadcast. Cuando todos los módulos detectan la información ID en el paquete de datos como 0xFF 0xFF, no realizarán ninguna comparación con su propia ID y enviará los dato reales automáticamente. Esta característica es muy útil para aplicaciones recolección de datos de medidores.

Por ejemplo, cuando el modulo central en un data logger quiere recolectar datos de medición, pueden ser enviados en una cadena de datos especial (definido por el usuario), con la información del nodo_ID 0xFF 0xFF, los módulos nodo en los medidores podrán ser habilitado con una forma de comunicación pausada. Cuando el host en los módulos nodo detectan la cadena especial sabrán que el data logger está recolectando datos de medición y podrán cambiar de una forma de comunicación pausada a una de comunicación rápida por medio del control de los pines EN y SET y permitirá que modulo nodo trabaje a rendimiento pleno para la transmisión de datos. El data logger puede recolectar los datos rápidamente y no necesita esperar por un retardo de tiempo por cada modulo conectado.

A.9 DS3231SN RTC (Real Time Clock)



Figura A.15: DS3231SN

Mientras una Raspberry se encuentra conectado a la red es capaz de ajustar correctamente su reloj mediante un servicio NTP. Sin una conexión a un sistema de tiempo y de fecha, estos datos serán casi seguro que sean erróneos. Para el caso de este proyecto es un problema ya se requiere una estampa de tiempo.

Para resolver este problema se emplea un modulo RTC (Real Time Clock) o Modulo de Reloj de Tiempo Real. Es usado con una batería para mantener el tiempo si se apaga la Raspberry; y cuando se reinicia la Raspberry puede ajustar su reloj interno usando el tiempo mantenido por el RTC. DS3231 incorpora medición y compensación de la temperatura garantizando una precisión de al menos 2ppm, lo que equivale a un desfase máximo de 172 ms/día o un segundo cada 6 días.

La comunicación se realiza a través del bus I2C, por lo que es sencillo obtener los datos medidos. La tensión de alimentación es de 2.3 a 5.5 V. Frecuentemente incorpora una pequeña EEPROM AT24C32 que puede ser empleada para almacenar registros y mediciones. Además tiene incluido una batería CR 2032 para mantener el dispositivo en hora al retirar la alimentación.

El bajo costo de DS3231, la relativa alta exactitud que puede mantener horas, minutos y segundos, así también días, meses y años de información. También la compensación automática por años bisiestos y por meses con 31 días.

Estas características que ofrece DS3231, fueron la razones para colocarlo dentro del este proyecto para tener una referencia de tiempo cuando se perdía la conexión a Internet, de manera precisa para el modulo central .

A.10 Diagrama de Modulo Shield Central

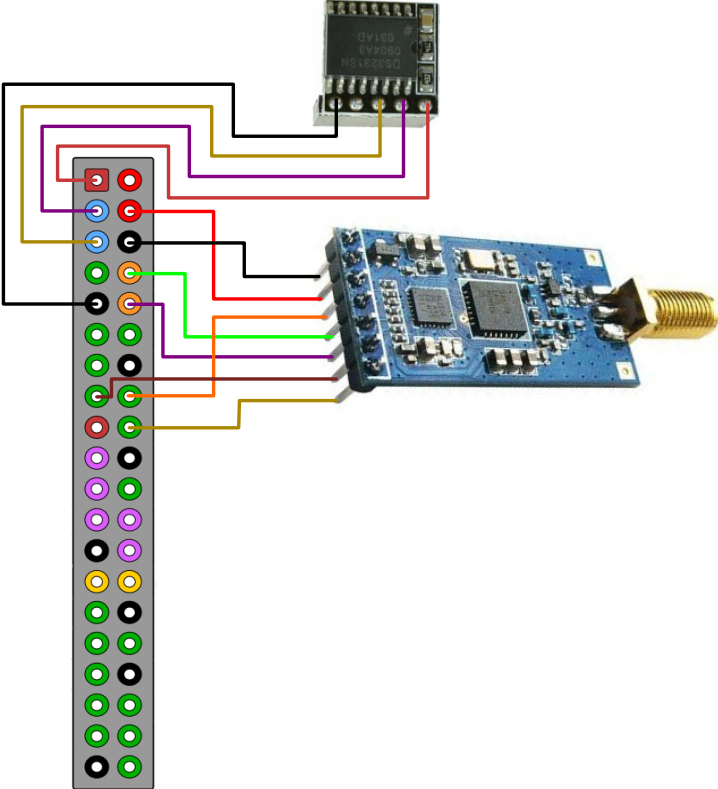


Figura A.16: Circuito General de Módulo Central

Apéndice B Configuraciones

- B.1 Configuración de Antenas.
- B.2 Configuración de Raspberry.
- B.3 Configuración de conexión wifi vía terminal.
- B.4 Configuración de puerto I2C y RTC.
- B.5 Configuración de puerto serie.
- B.6 Configuración de permisos de puerto serie y GPIOs.
- B.7 Instalación de servidor LAMP.
- B.8 Configuración de permisos de superusuario a php.
- B.9 Configuración de cron.

B.1 Configuración de Antenas.

En este punto se describe como se realiza la configuración de las antenas DRF1276DM, Dorji fabricante de estas antenas, proporciona dos maneras de configurar este tipo de antenas. La primera forma de configuración de las antenas es por medio de una herramienta de configuración gráfica desarrollado por este fabricante. La segunda forma de configuración es por medio en envío de código hexadecimal desde cualquier microcontrolador hacia la antena a través del puerto UART de la antena; esta segunda forma de configuración no se describe en este trabajo.

El primer paso para la configuración es generar una tabla como la siguiente:

	Central	Nodo 2	Nodo 3	Nodo 4	Unidades
Parámetros de Radio					
RF Frequency	915	915	915	915	Mhz
RF mode	Central	Nodo	Nodo	Nodo	Nodo
RF_Factor	2048	2048	2048	2048	Chips
RF_BW	125	125	125	125	kHz
Parámetros de Red					
Node ID	1	2	3	4	
Breath	2	2	2	2	Seg
Power	7	7	7	7	
Net ID	1	1	1	1	
Wake Timer	32	32	32	32	ms
Parámetros de Puerto Serial					
Baud Rate	9600	9600	9600	9600	bps
Parity	No	No	No	No	

Tabla B.1: Parámetros de Configuración de Antenas DRF1276DM.

Como se puede ver en la tabla existen tres tipos de parámetros a configurar parámetros de radio, parámetros de red y parámetros del Puerto Serial.

Para los parámetros de Radio, se tiene que configurar:

- RF Frequency: Que indica la frecuencia central de la portadora RF.
- RF mode: Modo Estándar, Para conexión en estrella, Modo central, modo nodo.
- RF_Factor: Factor de Propagación Lora. Un valor alto significa alta sensibilidad pero un tiempo de transmisión sobre el aire largo. Solo puede ser modificado por usuarios en modo estándar.
- RB_BW: Ancho de banda Lora. Un valor alto significa baja sensibilidad. Valor recomendado: 125 K. Solo puede ser modificado por usuarios en modo estándar.

Para los parámetros de red se tienen que configurar:

- Node ID: Solo es usado por módulos nodo: 0~65535.
- Breath: Es el periodo de levantamiento de modulo nodo. Solo disponible para firmware superiores a 2.0.
- Power: Es usado para ajustar la potencia de salida de DRF1276D. Existen 7 niveles de potencia. El nivel 7 significa la potencia máxima de salida---20dBm y 0 la menor potencia de salida.
- NetID: Solo los módulos con el mismo Network ID(0~255) pueden comunicarse entre ellos. Puede evitar interferencias con módulos desconocidos.
- Waker Timer: El tiempo de detección de señales inalámbricas durante el periodo de breath, solo disponible para firmware 2.7 o superior.

Para los parámetros de Puerto serial se tiene que configurar:

- Baud Rate: Define la tasa de datos entre DRF1276DM y el host.
- Parity: Define la paridad entre DRF1276DM y el host.

Con esta tabla se puede realizar la configuración de las antenas. Pero antes de proceder con la explicación de la configuración de usando el software del fabricante; el usuario puede inserta la antena a una tarjeta convertidor UART a USB (DAC02, en este caso) para conectarlos a una computadora y ejecutar la herramienta de configuración gráfica. *En el modo de configuración, el pin EN debe ser conectado a GND y entonces la herramienta de configuración o los comandos podrán configurar de manera efectiva¹*. Nota: la condición para que dos o mas módulos se comuniquen es que la frecuencia, RF_Factor, RF_BW y Net ID debe ser los mismos.

¹En el modo de configuración, el pin EN debe ser conectado a GND y entonces la herramienta de configuración o los comandos podrán configurar de manera efectiva.

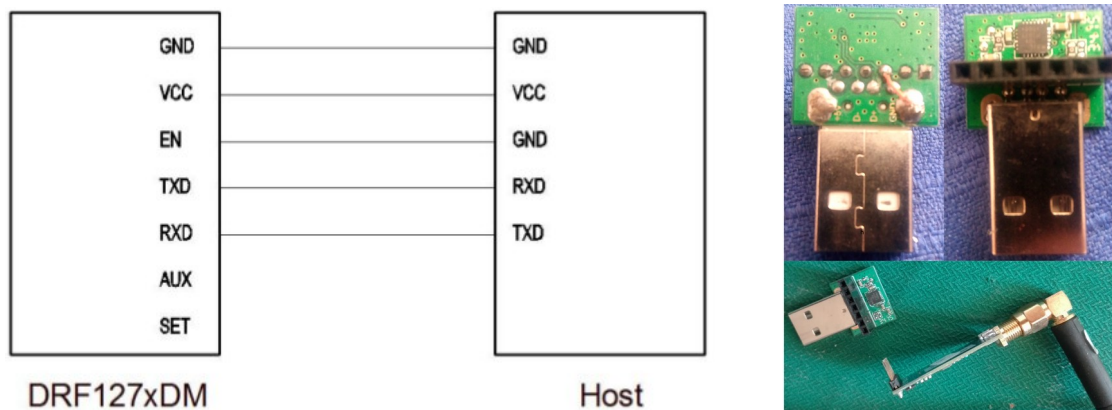


Figura B.1: Conexión de pines en modo configuración de antena(izquierda) y conexión en DAC02(derecha).

En la figura 1 se muestra la conexión de los pines al momento de realizar su configuración, es importante tener en claro que el pin EN se tiene que aterrizar para poder programar las antenas. En nuestro caso para poder aterrizar el pin EN, se realizo una puente entre GND y el hembra correspondiente al pin EN en el conector DAC02, que se puede apreciar en la figura 1 de lado derecho , de esta manera este convertidor se empleo para configurar todas las antenas, solo montando las antenas y ejecutando la herramienta de configuración. El puerto UART correspondiente aparecerá en el interfaz.

Se puede dar clic en el botón OPEN para activar el puerto UART, y ahora se puede dar clic en el botón Read All para leer los parámetros de default.

Ahora se puede cambiar a los parámetros de la tabla 1 y después dar clic en Write All para escribir los nuevos parámetros en el modulo. Para confirmar que los datos se han escrito correctamente, el interfaz gráfico responde Successful, en otro caso no se han escrito los parámetros modificados.

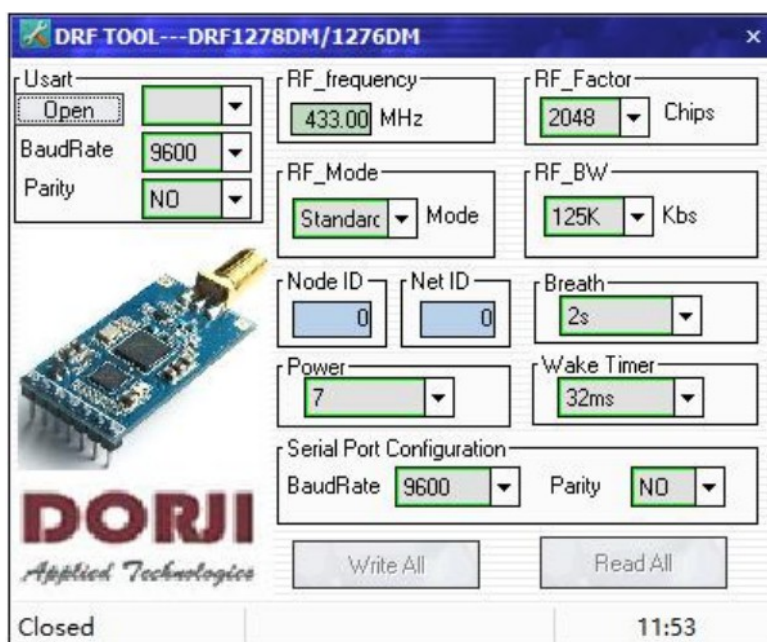


Figura B. 2: Herramienta de configuración de módem LoRa.

RF_Factor [Chips]	RF_BW [Khz]	RF data rate [bps]	Sensibilidad [dBm]
4096	125	293	-138
2048	125	537	-135
4096	250	586	-135
1024	125	977	-132
2048	250	1074	-132
4096	500	1172	-132
512	125	1758	-129
1024	250	1954	-129
2048	500	2148	-129
256	125	3125	-126
512	250	3516	-126
1024	500	3908	-126
128	125	5470	-123
256	250	6250	-123
512	500	7032	123
128	250	10940	-120
256	500	12500	120
128	500	21880	117

Tabla B.2: Sensibilidad, RF_Factor y RF_BW.

B.2 Configuración de Raspberry.

El módulo central de este proyecto esta construido con una raspberry por lo que es importante aprender conoce con se habilita para trabajar con ella.

Como primer paso hay que descargar la ultima versión del sistema operativo de la raspbian, es recomendable la versión lite². Este sistema operativo se descarga en un archivo de tipo ISO. El siguiente paso es montar este sistema operativo en una tarjeta MicroSD de un tamaño no menor a 8 G, existen varias manera para hacer esta operación dependiendo del sistema operativo que se este empleando por lo que hay que realizar la metodología correspondiente.

En este caso se esta trabajando con una sistema operativo Linux y hay que realizar el siguiente procedimiento para montar el sistema operativo raspbian a una tarjeta MicroSD:

- Es necesario ejecutar el comando `lsblk` para ver cuales son los dispositivos que están conectados a tu computadora .
- Si tu computadora tiene una ranura para memorias MicroSD, inserta la tarjeta. En caso contrario utiliza un lector de tarjeta SD y conectarlo a tu computadora.
- Se ejecuta nuevamente el comando `lsblk`. El nuevo dispositivo que aparece es la tarjeta SD. El

² La versión lite, si bien esta versión no incluye interfaz gráfica, su forma de trabajar es con la línea de comando; reduce a un mínimo el número de procesos que este sistema operativo requiere para operar y a demás ocupa un menor espacio en la tarjeta MicroSD, por lo que tiene un mejor rendimiento.

nombre del dispositivo servirá para el siguiente paso.

- Ahora es momento de desmontarlo con el siguiente comando:

```
umount /dev/sdX1
```

Es necesario remplazar `sdX1` con el nombre de la tarjeta SD.

- Para copiar la imagen iso a la tarjeta SD, se realiza con el siguiente comando:

```
dd bs=4M if=2018-04-18-rasbian-stretch.img of=/dev/sdX conv=fsync
```

`4M` hace referencia al ajuste de tamaño del bloque; `2018-04-18-rasbian-stretch.img` es necesario sustituirlo por el nombre de la imagen que se descargó; En `/dev/sdX` se debe sustituir `sdX` por el nombre de la tarjeta SD sin el número, esta acción toma varios minutos.

Ya montado el sistema operativo raspbian en la tarjeta MicroSD. Esta se inserta en la ranura para es tipo de tarjetas en la raspberry.

Para continuar con la configuración de la raspberry, se tiene que conectar una fuente de 5 vcc al conector microusb; con esto se encenderá la raspberry. La primera vez que se inicie la Raspberry Pi 3, aparecerá una pantalla azul donde se pueden realizar las primeras configuraciones.

La primera configuración será respecto al idioma. En el menú de la pantalla azul, se selecciona la opción numero cuatro:

```
4 Localisation Options Set up language and regional settings to match your location
```

en el siguiente menú seleccionar la primera opción:

```
I1 Change Locale Set up Language and regional settings to match your location
```

el siguiente paso es deshabilitar cualquier idioma seleccionado, verificando que no exista ningún idioma seleccionado, se selecciona `es_MX.UTF-8 UTF-8`.

También es necesario configurar la zona horaria al igual que en la selección anterior se selecciona la opción numero 4 del menú principal:

```
4 Localisation Options Set up language and regional settings to match your location
```

en el siguiente menú, seleccionar la segunda opción:

```
I2 Change Timezone Set up timezone to match your location
```

seguido se selecciona el área geográfica: `America` (nuestro caso) y zona horaria: `Ciudad de México` (nuestro caso).

Al realizar estos cambios se reiniciará la raspberry, para poder ingresar a nuestro usuario, se utiliza los siguientes datos:

```
Nombre de usuario: pi
```

```
Contraseña: raspberry
```

es recomendable cambiar lo mas pronto posible la contraseña del equipo, con la primera opción de menú principal de configuración, esto después de ejecutar el comando `sudo raspi-config`:

```
1 Change User Password Change password for the default user (pi)
```

B.3 Configuración de conexión wifi vía terminal.

Ya se esta usando un sistema operativo en la raspberry en versión lite , no se cuenta con una interfaz gráfica por lo que es necesario configurar la conexión wifi vía terminal.

Para escanear las redes inalámbricas existentes se usa el comando: `sudo iwlist wlan0 scan`. Esto muestra una lista de todas las redes inalámbricas existentes, con información adicional de cada una de las redes.

Para añadir la red que se desea conectar, se abre el archivo de configuración `wpa-supPLICANT`, con el siguiente comando:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

y al final del archivo se agrega lo siguiente:

```
network{
    ssid="nombre_de_la_red"
    psk="contraseña_de_red"
}
```

Para guardar el archivo se preciona `ctrl+x`, despues `Y`, finalmente se oprime `Enter`.

Para obtener la ip se logra con el siguiente comando:

```
ip addr show
```

dato con el que se puede conectar remotamente via `ssh`.

Después de tener una conexión a internet es recomendable ejecutar los siguientes comando para actualizar la lista de paquetes desde sus fuentes para este sistema operativo y para actualizar los paquetes instalados.

```
sudo apt-get update
sudo apt-get upgrade
```

B.4 Configuración de puerto I2C y RTC.

La Raspberry es capaz de actualizar su fecha y hora utilizando un servicio de NTP (Network Time Protocol), por medio de internet, pero cuando no existe una conexión a Internet el tiempo en la raspberry es por lo regular erróneo, ya que este dispositivo no es capaz de guardar estos valores, para conservar estos valores se emplea hardware adicional como es el caso de un DS3231. DS3231 es un modulo de reloj de tiempo real o RTC, es un circuito alimentado por una pequeña pila que ayuda a mantener estos valores y a seguir contando cuando las raspberry se apaga. Pero usar este modulo se necesita activar el puerto `i2c` de la raspberry que es el medio de comunicarse. Esto se logra con los siguientes pasos:

Para activar puerto `i2c`:

- En la terminal ejecutamos el comando: `sudo raspi-config`
- Mostrará una pantalla azul con un menú. En el menú será necesario seleccionar la opción:

```
5 Interfacing Options  Configure connections to peripherals
```
- Seguido seleccione la opción:

```
p5 I2C Enable/Disable automatic loading of I2C kernel module
```
- Dar clic en seleccionar y aceptar hasta terminar la configuración.

Para que la Raspberry reconozca los cambios es necesario reiniciar.

En la instalación del reloj en la Raspberry, se tiene que seguir los siguientes pasos:

- Será necesario modificar el archivo `sudo nano /etc/modules`, colocando en la parte final las siguientes líneas.

```
snd-bcm2835
i2c-bcm2835
i2c-dev
rtc-ds1307
i2c-bcm2708
```

- Para guardar los cambios, hacer clic en `ctrl +x` y presionar `y + Enter`.
- En este punto será necesario conectar el módulo DS3231 como se describió en el anexo A.
- Enciende la Raspberry, y verifica que el puerto i2c está habilitado con `lsmod | grep i2c_` que mostrará lo siguiente: `i2c_bcm2708`, lo que significa que el módulo está operando adecuadamente.
- Para probar la interfaz para explorar a todas las posiciones i2c esclavo se usa: `sudo i2cdetect -y 1`, que para este punto mostrará una tabla con las posiciones i2c entre ellas del RTC (en la posición 0x68).

- Para instalar el DS3231 en la Raspberry, se tiene que modificar el archivo `/etc/modprobe.d/raspi-blacklist.conf`, para hacerlo se ejecuta el siguiente comando:

```
sudo sed -i 's/blacklist i2c-bcm2708/#blacklist i2c-bcm2708/' /etc/modprobe.d/raspi-blacklist.conf
```

- Para cargar el módulo `i2c-bcm2708`:

```
sudo modprobe i2c-bcm2708
```

- Para notificar al sistema que se agrega un nuevo dispositivo en la posición i2c 0x68:

```
echo ds3231 0x68 | sudo tee /sys/class/i2c-adapter/i2c-1/new_device
```

- Reinicia la Raspberry y ejecuta el siguiente comando:

```
sudo i2cdetect -y 1
```

se mostrará UU lo que significa que el reloj ha sido agregado correctamente.

- La fecha y hora que tiene el módulo RTC se puede ver con:

```
sudo hwclock
```

- Para obtener la hora correcta, se puede obtener desde internet con el servicio NTP, con el siguiente comando:

```
sudo ntpd -q
```

- Obtenidos los datos correctos de fecha y hora, es necesario cargarlos en el RTC DS3231, con:

```
sudo hwclock -w
```

- Finalmente para asegurarse que el RTC sea quien esté asignando el tiempo en la Raspberry desde que esta se inicia se debe modificar el archivo `sudo nano /etc/rc.local`, donde hay que agregar las siguientes líneas antes de la línea de `exit 0`:

```
echo ds3231 0x68>/sys/class/i2c-adapter/i2c-1/new_device
hwclock -s
```

seguido hacer clic en `Ctrl +x`, después presionar `Y + Enter`. Con esto se agrega el reloj DS3231.

B.5 Configuración de puerto serie.

Por default el puerto serie Raspberry Pi es configurado para el uso de un modulo de bluetooth, y el puerto miniUART, varia su reloj, como se menciona en el Anexa A.

Como primer paso hay que deshabilitar el bluetooth. Esto se consigue con los siguientes pasos:

- Se tiene que modificar el archivo `/boot/config.txt` para deshabilitar el bluetooth de la Raspberry Pi 3, con el siguiente comando se abre el archivo.

```
sudo nano /boot/config.txt
```

- Al final del archivo se agregar la siguiente linea:

```
dtoverlay=pi3-disable-bt
```

- A continuación se debe oprimir: `Ctrl+x`
- Entonces aparecerá una notificación preguntando si desea guardar los cambios, deberá oprimir que si.
- Para finalizar este paso se tiene que reiniciar la raspberry con el siguiente comando.

```
sudo reboot
```

Para habilitar el puerto serie para usarlo, se debe realizar es lo siguiente:

- Hay que editar el archivo `/etc/inittab`. Este archivo tiene el mando de habilitar el login prompt y se necesita deshabilitarlo. Para editar este archivo hay que ejecutar el siguiente comando:

```
sudo nano /etc/inittab
```

- Seguido hay que moverse al final del archivo. Se podrá ver una linea similar a lo siguiente:

```
T0:23:respawn:/sbin/getty/ -L ttyAMA0 115200 vt100
```

- Para deshabilitarlo hay que añadir un carácter `#` al inicio.

```
#T0:23:respawn:/sbin/getty/ -L ttyAMA0 115200 vt100
```

- por ultimo se tiene que guardar y cerrar el archivo con: `Ctrl+x`
- Aparecerá una notificación preguntando si desea guardar, deberá oprimir que si.

Seguido hay que modificar otro archivo, por lo que se usará el siguiente comando:

```
sudo nano /boot/cmdline.txt
```

- Este archivo contendrá algo parecido a lo siguiente:

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200  
console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline  
rootwait
```

- Es necesario remover todo lo referido a `ttyAMA0`. El archivo ahora se verá algo parecido a esto:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4  
elevator=deadline rootwait
```

- A continuación se debe oprimir: `Ctrl+x`
- Entonces aparecerá una notificación preguntando si desea guardar los cambios, deberá oprimir que si.

Finalmente para que el dispositivo aplique todos los cambio es necesario reiniciar la raspberry con el

siguiente comando:

```
sudo reboot
```

B.6 Configuración de permisos de puerto serie y GPIOs.

Para poder utilizar los servicios de puerto serie y los gpios de la Raspberry Pi 3 B, si necesidad de utilizar el comando sudo, es necesario darle a nuestro usuario cierto permisos. Para ello se realiza el siguiente procedimiento:

- hay que tener en cuenta que la configuración va a depender del nombre de usuario, en nuestro caso nuestro usuario *pi*. Este dato lo conocemos desde que entramos a la terminal de raspberry. Pero también lo podemos ver el prompt como sigue:

```
nombre_de_usuario@nombre_de_maquina
```

Para esta versión de raspberry hay grupos que ya están debidamente habilitados para usar estos servicios, lo que hay que hacer es agregar a nuestro usuario a estos grupos `gpio` y `dialout` para gpios y puerto serie respectivamente, con los siguientes comandos con estructura.

```
sudo adduser nombre_de_usuario gpio
sudo adduser nombre_de_usuario dialout
```

nuestro caso:

```
sudo adduser pi gpio
sudo adduser pi dialout
```

- Para aplicar los cambios es necesario reiniciar la maquina con el siguiente comando:

```
sudo reboot
```

- Finalmente para verificar que nuestro usuario ya pertenecen a estos grupos aplicamos el siguiente comando debemos verificar que aparezcan:

```
id
```

- y aparecerá lo siguiente, buscamos los grupos.

```
uid=1000(pi) gid=1000(pi) grupos=1000(pi),20(dialout),44(video),997(gpio)
```

- Listo podremos usar los servicios de gpios y puerto serie sin usar sudo.

B.7 Instalación de LAMP.

LAMP, es un acrónimo de Linux, Apache, MySQL y PHP. Instalar estos programas nos permite tener a nuestra disposición un servidor web personal con el que se puede desarrollar paginas web basados en php en una raspberry. Linux hace referencia al sistema operativo (raspbian), Apache hace referencia al servidor web a emplear, MySQL es un programa de base de datos y PHP para desarrollar paginas web en la Raspberry Pi 3.

Para su instalación, se siguen estos pasos:

- Como primer paso se tiene que crear y dar permisos al grupo que usa apache por defecto, con los siguientes comandos:

```
sudo addgroup www-data
sudo usermod -a -G www-data www-data
```

- Es necesario instalar el paquete `apache2`, esto se realiza con el siguiente comando:

```
sudo apt-get install apache2 -y
```

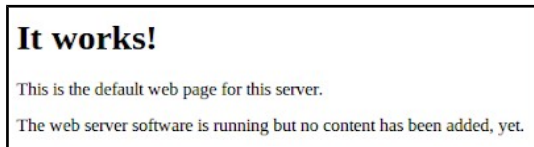
Para probar que el servidor web esta funcionando bien. Abrimos un buscador web y se escribe en en la dirección lo siguiente:

http://ip_local_de_raspberry

En nuestro caso:

<http://192.168.1.100>

y aparecerá la siguiente notificación:



Esto significa que el servidor web esta trabajando bien.

- Para instalar PHP en la raspberry se ejecuta el siguiente comando:

```
sudo apt-get install php5 libapache5-mod-php5
```

PHP ayuda a ejecutar dinámicamente archivos de python. Para crear un archivo php se tiene que mover al directorio `/var/www/html` y crear un archivo `index.php` sustituyendo en que existe.

- Es necesario reiniciar Apache, con el siguiente comando:

```
sudo /etc/init.d/apache2 restart
```

- Para instalar MySQL y PHPMyAdmin es necesario activar una interfaz de loopback, ya que se no se hace, dará error al instalar MySQL; esto con el siguiente comando:

```
sudo ifup lo
```

ahora para instalar ejecutamos el siguiente comando:

```
sudo apt-get install mysql-server mysql-client php5-mysql phpmyadmin
```

Al instalarse aparecerá una pantalla solicitando una contraseña de seguridad. Tras esto se realiza lo siguiente:

```
sudo nano /etc/php5/apache2/php.ini
```

Se abrirá un archivo donde se tiene que agregar lo siguiente, después de la línea "Dynamics Extensions":

```
extension=mysql.so
```

se guarda el archivo. Y se escribe los siguientes comandos:

```
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf.d/phpmyadmin.conf
```

```
sudo /etc/init.d/apache2 reload
```

Para hacer funcionar phpmyadmin se agrega al final del archivo `/etc/apache2/apache2.conf` la instrucción:

```
Include /etc/init.d/apache2/apache2.conf
```

Por ultimo se reinicia el servidor web con:

```
/etc/init.d/apache2 restart
```

B.8 Configuración de permisos de superusuario a php.

Al momento de instalar php se crea un usuario y grupo llamando `www.data` y `www.data`. PHP usará este usuario y grupo para invocar sus archivos de paginas web. Php tiene que ejecutar archivos que no le pertenecen a este usuario y por defecto rasbian no permite que los usuarios ejecuten archivos que le pertenezcan por ello hay que darle permisos de superusuario al usuario `www.data` para que pueda ejecutar los archivos de ejecución programados.

Esto se realiza con los siguientes pasos:

- Ejecutar el siguiente comando:

```
sudo visudo
```

Este comando abrirá el archivo `/etc/sudoers`, archivo que define los usuarios con permisos de superusuario.

- Después de esto nos abrirá un archivo, deberá moverse a la parte final, oprimir la tecla `a` y colocar lo siguiente:

```
www-data ALL=(root) NOPASSWD:ALL
```

- Seguido debe oprimir la tecla `ESC`, escribir lo siguiente:

```
:wq
```

y presionar la tecla `Enter` para guardar los cambios.

- Para aplicar los cambios es necesario reiniciar la maquina con el siguiente comando:

```
sudo reboot
```

B.9 Configuración de cron.

El servicio de cron en rasbian es un servicio para la ejecución de tareas programadas, para este proyecto se programa ejecutar un petición de información cada 15 minutos al modulo de estación desde el modulo central, esto se logra con cron. El archivo que hay que modificar se llama `crontab`, que se logra de la siguiente manera:

- Para modificar `crontab` se ejecuta el siguiente comando:

```
crontab -e
```

- Después de esto se abrirá un archivo, deberá moverse a la parte final, oprimir la tecla `a` y colocar lo siguiente :

```
0,15,30,45 * * * * cd M_Central/ && python LEC_ID2_v1_6.py && cd
```

- Seguido debe oprimir la tecla `ESC` y escribir lo siguiente:

```
:wq
```

- y presionar la tecla `Enter` para guardar los cambios.

Para para detener el servicio de cron tiene que hacer lo siguiente:

- Para modificar `crontab` se ejecuta el siguiente comando:

```
crontab -e
```

- Después de esto nos abrirá un archivo, deberá moverse a la parte final, oprimir la tecla `a` y colocar un carácter `#` al inicio de la linea que se modifico para habilitar cron, debe aparecer como lo

siguiente :

```
#0,15,30,45 * * * * cd M_Central/ && python LEC_ID2_v1_6.py && cd
```

- Seguido debe oprimir la tecla ESC, escribir los siguiente:

```
:wq
```

y presiona la tecla Enter para guardar los cambios.

Que es lo que se está haciendo, al momento se agregar la linea de programación de cron:

Linea de programación de cron	Significado
0,15,30,45	Esto significa en que minutos se tiene que realizar la tarea programada. Para este caso en los minutos 0, 15, 30 y 45 se ejecutará la tarea programada.
*	Esto significa a que horas se tiene que realizar la tarea programada. Para este caso * significa que a cualquier hora se ejecutará la tarea programada.
*	Esto significa en que días se tiene que realizar la tarea programada. Para este caso * significa que en cualquier día se ejecutará la tarea programada.
*	Esto significa en que mes se tiene que realizar la tarea programada. Para este caso * significa que en cualquier mes se ejecutará la tarea programada.
*	Esto significa en que año se tiene que realizar la tarea programada. Para este caso * significa que en cualquier año se ejecutará la tarea programada.
cd M_Central	Esta parte significa que se esta cambiando de directorio, para esta caso se entra al directorio M_Central donde esta guardado el archivo de ejecución.
python LEC_ID2_v1_6.py	Con este comando se ejecuta el archivo de ejecución con python, para este caso nuestro archivo se llama LEC_ID2_v1_6.py
cd	Por ultimo para salir del directorio M_Central, se usa este comando para retornar una nuestro directorio base.

Table B.3: Descripción de programación de tareas programadas con cron.

Para conectar ejecutar los comandos uno después de otro se emplea && .

Por ultimo cuando se detiene el servicio de cron, agregamos el carácter #, esto significa que se esta comentando esa linea, por lo que queda sin efecto de programación.

Apéndice C Código de proyecto.

C.1 Programación de Módulo Estación

```
#define SerialBufferSize 10

void Comunicacion(int a);
boolean CheckSerial();
void menuCommands();
void ControlLamp();

float get_corriente();
float get_voltaje();
int get_events();

//////////////////////////////////// Se declaran las diferentes Peticiones //////////////////////////////////////
char ON[SerialBufferSize] = "ON";
char OFF[SerialBufferSize] = "OFF";
char LEC[SerialBufferSize] = "LEC";

//////////////////////////////////// Se inicializa Buffer para guardar datos //////////////////////////////////////
char inputBuffer[SerialBufferSize]; // Se declara bufer
int serialIndex = 0; // Se inicializa indice de recepcion de datos

int ID = 2; //ID de dispositivo

int a0 = 1; //Bandera para habilitar aplicacion
int a1; //Bandera para estado de la lampara ON/OFF
int umbral = 200; //Umbral de luz
//int light = 1; //Fotoresistencia Entrada
float lec_light; //parametro de cantidad de luz
int ledpin = 7; //Salida a relevador

float Voltaje;
float Corriente;

//////////////////////////////////// Se declaran las entradas analogicas a utilizar //////////////////////////////////////
#define light A0 //Sensor Fotoresistencia Entrada
#define eventos A1 //Sensor Registro de eventos Entrada
#define Med_Corriente A2 //Sensor de Medicion de Corriente
#define Med_Voltaje A3 //Sensor de Medicion de Voltaje
////////////////////////////////////Funcion de ajustes////////////////////////////////////
void setup() {
    pinMode(ledpin, OUTPUT); //Salida a relevador
    pinMode(4, OUTPUT); //ajusta SET
    pinMode(2, OUTPUT); //ajusta EN
    pinMode(3, INPUT); //ajusta AUX

    Serial.begin(9600);
    int a = 1; //a=0 Comunicacion express // a=1 Comunicacion with breath
    Comunicacion(a);
}
////////////////////////////////////Funcion primaria de programa////////////////////////////////////
void loop() {
    menuCommands();
    lec_light = analogRead(light);
    if (a0) ControlLamp();
}
////////////////////////////////////Funcion control automatico de lampara////////////////////////////////////
void ControlLamp() {
    if (lec_light < umbral) {
        digitalWrite(ledpin, HIGH);
        a1 = 1;
    }
    if (lec_light >= umbral) {
        digitalWrite(ledpin, LOW);
        a1 = 0;
    }
}
////////////////////////////////////Menu de comandos de Modulo Central////////////////////////////////////
void menuCommands() {
    int i = 0;
    if (CheckSerial()) {
        //Serial.print(inputBuffer);
        if (inputBuffer[1] == ON[1] && inputBuffer[2] == ON[2]) {
            a0 = 1;
        }
        if (inputBuffer[1] == OFF[1] && inputBuffer[2] == OFF[2] && inputBuffer[3] == OFF[3]) {
            a0 = 0;
            digitalWrite(ledpin, LOW);
        }
        if (inputBuffer[1] == LEC[1] && inputBuffer[2] == LEC[2] && inputBuffer[3] == LEC[3]) {
            Voltaje= get_voltaje();
            Corriente= get_corriente();
            int Evento=get_events();
            Comunicacion(0);
            while (i < 2) {
                delay(2000);
                Serial.print(ID);
                if (a0 == 0) Serial.print("Inh");
                else {
                    if (a1 == 1) Serial.print("OON");
                    if (a1 == 0) Serial.print("OFF");
                }
            }
        }
    }
}
```


C.2 Programación de Módulo Central

Estos programas se encuentran en una carpeta de la raíz del usuario pi en la Raspberry Pi 3, llamada M_Central.

C.2.1 Programa de control remoto de encendido (ON_ID2.py)

```
#!/usr/bin/env python

import serial
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.OUT)#ajusta EN
GPIO.setup(24, GPIO.OUT)#ajusta SET
GPIO.setup(22, GPIO.IN, pull_up_down = GPIO.PUD_UP) #ajusta AUX
ser=serial.Serial(
    port='/dev/ttyAMA0',
    baudrate=9600
)

def comunicacion(b):
    #b=0 # comunicacion express #b=1 comunicacion breath
    if b==0:
        GPIO.output(23,False)
        time.sleep(1)
        GPIO.output(24,False)
    if b==1:
        GPIO.output(23,False)
        time.sleep(1)
        GPIO.output(24,True)

comunicacion(1)
ser.write('\x00'+'\x02'+'\x0N'+'\x0D')
print('ON APLICACION 2')
ser.close()
GPIO.cleanup()
```

C.2.2 Programa de Control remoto de Apagado (OFF_ID2.py)

```
#!/usr/bin/env python

import serial
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.OUT)#ajusta EN
GPIO.setup(24, GPIO.OUT)#ajusta SET
GPIO.setup(22, GPIO.IN, pull_up_down = GPIO.PUD_UP) #ajusta AUX
ser=serial.Serial(
    port='/dev/ttyAMA0',
    baudrate=9600
)

def comunicacion(b):
    #b=0 # comunicacion express #b=1 comunicacion breath
    if b==0:
        GPIO.output(23,False)
        time.sleep(1)
        GPIO.output(24,False)
    if b==1:
        GPIO.output(23,False)
        time.sleep(1)
        GPIO.output(24,True)

comunicacion(1)
ser.write('\x00'+'\x02'+'\x0F'+'\x0D')
print('OFF APLICACION 2')
ser.close()
GPIO.cleanup()
```

C.2.3 Programa de Petición de Parámetros de Medición (LEC_ID2_v2_6.py)

```
#!/usr/bin/env python

import serial
import RPi.GPIO as GPIO
import time
from datetime import datetime
import datetime
import json
import pymysql
conn= pymysql.connect(host="localhost", user="MMC", passwd="password123", db="MMC")
myCursor = conn.cursor()

a=1
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.OUT) #ajusta EN
GPIO.setup(24, GPIO.OUT) #ajusta SET
GPIO.setup(22, GPIO.IN, pull_up_down = GPIO.PUD_UP) #ajusta AUX
ser=serial.Serial('/dev/ttyAMA0',baudrate=9600)
dataframe_2_save=[]
```

```

#b=0 # comunicacion express #b=1 comunicacion breath
def comunicacion(b):
    if b==0:
        GPIO.output(23,False)
        time.sleep(1)
        GPIO.output(24,False)
    if b==1:
        GPIO.output(23,False)
        time.sleep(1)
        GPIO.output(24,True)
#define estampa de tiempo
def stamp_time():
    t=datetime.datetime.now()
    anio=t.year
    mes=t.month
    nombre_tabla="ID_02"+"_"+str(anio)+"_"+str(mes)
    return nombre_tabla

comunicacion(1)
time1=time.time()
ser.write('\x00'+'\x02'+'\x00'+'\x0D')
ser.close()

comunicacion(0)
ser=serial.Serial("/dev/ttyAMA0",baudrate=9600, timeout=5)
data=[]

while a:
    if GPIO.input(22):
        time.sleep(2.0/1000.0)
        r=ser.read()
        data.append(r)
        if (r=='\n' ):
            a = 0

ID=data[0]
Edo=data[1]+data[2]+data[3]
Edo.strip()
Voltage=data[4]
Current=data[10]
Event=data[14]
for i in range(16):
    if (i>4 and i<=9):
        Voltage=Voltage+data[i]
    Voltage.strip()
    #Current=data[10]
    if (i>9 and i<=13):
        Current=Current+data[i]
    Current.strip()
    #Event=data[14]
    if (i>13 and i<=16):
        Event=Event+data[i]
    Event.strip()

now=datetime.datetime.now()
datestamp=str(now.year)+"-"+str(now.month)+"-"+str(now.day)
datestamp.strip()
hourstamp=str(now.hour)+"."+str(now.minute)
hourstamp.strip()

Power=str(float(Voltage)*float(Current))

nombre_tabla=stamp_time() ##nombre de tabla para MySQL

#####PREGUNTAR LA EXISTENCIA DE LA TABLA
myCursor.execute("CREATE TABLE IF NOT EXISTS test1 (ID int(10) NOT NULL AUTO_INCREMENT, Fecha DATE, Hora TIME, Identificador INT(2), Estado VARCHAR(5), Voltaje float, Corriente float, Potencia float, Evento INT(3), PRIMARY KEY(ID));")
myCursor.execute("SELECT MAX(id) FROM test;")
a=myCursor.fetchall()
Id= int(a[0][0])+1
myCursor.execute("INSERT INTO test (ID, Fecha, Hora, Identificador, Estado, Voltaje, Corriente, Potencia, Evento) VALUES(%s, NOW(), NOW(), %s, %s, %s, %s, %s, %s);", (Id, ID, Edo, Voltage, Current, Power, Event))

conn.commit()
conn.close()

time2=time.time()
print '%0.3f ms' % ((time2-time1)*1000)
ser.close()
GPIO.cleanup()

```

C.3 Programación de Interfaz Gráfica

C.3.1 Login

C.3.1.1 Archivo index.php

```

<<?php
include('login.php');
if(isset($_SESSION['Usuario'])){
    header("location: profile.php");
}
?>

<!DOCTYPE html>

```

```

<html>
<head>
  <meta charset="utf-8">
  <title>SAIAP</title>
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>
<body>
<div class="container">
  <div class="row">
    <div class="col-md-5 col-md-offset-3">
      <div class="page-header text-center">
        <h1>Sistema de Administración Inteligente de Alumbrado Público</h1>
      </div>
      <form class="form-signin" method="post" action="manejador.php">
        <h2 class="form-signin-heading">Please sign in</h2>
        <label for="user" class="sr-only">Usuario</label>
        <input type="text" name="user" class="form-control" placeholder="Nombre de Usuario" required autofocus>
        <label for="password" class="sr-only">Contraseña</label>
        <input type="password" name="password" class="form-control" placeholder="Contraseña" required>
        <input type="submit" name="submit" class=" btn btn-lg btn-primary btn-block" value="Login">
      </form>
    </div>
  </div>
</div>
</body>
</html>

```

C.3.1.2 Archivo manejador.php

```

<?php
/*
session_start();
$connection =mysqli_connect("localhost","MMC","password123","MMC");

if(isset($_POST['user'])){
  $uname = $_POST['user'];
  $password = $_POST['password'];
  $sql = "SELECT * FROM login WHERE Usuario=$uname AND Contraseña=$password LIMIT 1";
  $result = mysqli_query($connection, $sql);
  $row = mysqli_num_rows($result);
  $x=5;
  echo $x;
}
*/
/*
if (isset($_POST['submit'])) {

    if ( (isset($_POST['user'])) || (isset($_POST['password'])) ){
        $nombre = $_POST['user'];
        $pass = $_POST['password'];

        $sql = "SELECT * FROM login WHERE Usuario=$nombre AND Contraseña=$pass LIMIT 1";
        $result = mysqli_query($connection, $sql);
        if (mysqli_num_rows($result)==1) {
            //crear nuestra sesion
            $_SESSION['username'] = $nombre;
            header("location: mvc/index.php");

        }else{
            header("location: index.php?res=incorrecto");
        }
    }else{
        header("location: index.php");
    }
}

*/

if (isset($_POST['submit'])) {
  $username = "carlos";
  $password = "123";
  if ( (isset($_POST['user'])) || (isset($_POST['password'])) ){
    $nombre = $_POST['user'];
    $pass = $_POST['password'];

    if (($nombre == $username) && ($pass == $password)) {
      //crear nuestra sesion
      $_SESSION['username'] = $nombre;
      header("location: mvc/index.php");

    }else{
      header("location: index.php?res=incorrecto");
    }
  }else{
    header("location: index.php");
  }
}

?>

```

C.3.2 MVC

C.3.2.1 Archivo index.php

```
<?php
require_once "controllers/controller.php";
require_once "models/model.php";

$mvc =new MvcController();
$mvc -> plantilla();
?>
```

C.3.2.2 Modelo

C.3.2.2.1 Archivo model.php

```
<?php
/**
 *
 */
class EnlacesPaginas{

    public function enlacesPaginasModel($enlacesModel) {
        # code...
        if($enlacesModel == "consumogeneral" ||
            $enlacesModel == "eventos" ||
            $enlacesModel == "eventoslamp1" ||
            $enlacesModel == "contacto" ||
            $enlacesModel == "salir")
        {
            $module = "views/modules/".$enlacesModel.".php";
        }
        elseif ($enlacesPaginasModel= "inicio") {
            # code...
            $module = "views/modules/inicio.php";
        }
        else {
            # code...
            $module = "views/modules/inicio.php";
        }
        return $module;
    }
}
?>
```

C.3.2.3 Controlador

C.3.2.3.1 Archivo controller.php

```
<?php
class MvcController{
    #LLAMADA DE PLANTILLA
    #-----
    public function plantilla()
    {
        # code...
        include "views/template.php";
    }
    #INTERACCION DEL USUARIO
    #-----
    public function enlacesPaginasController() {
        # code...
        if (isset($_GET["action"])){
            $enlacesController = $_GET["action"];
            #echo $enlacesController;
        }
        else{
            $enlacesController = "inicio";
        }
        $respuesta = enlacesPaginas::enlacesPaginasModel($enlacesController);
        include $respuesta;
    }
}
?>
```

C.3.2.4 Vistas

C.3.2.4.1 Archivo template.php

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title> SAIAP </title>
    <?php include "estilo.css"; ?>
</head>
<body>
    <header>
        
    </header>

    <?php include "modules/navegacion.php"; ?>
    <section>
        <?php
            $mvc =new MvcController();
            $mvc -> enlacesPaginasController();
        ?>
    </section>
</body>
```

```
</html>
```

C.3.2.4.2 Archivo inicio.php

```
<br>
<head>
<link href="site.css" rel="stylesheet">
</div>
</head>

<?php
if (isset($_POST['LightON']))
{
exec("sudo python /home/pi/M_Central/ON_ID2.py");
}
if (isset($_POST['LightOFF']))
{
exec("sudo python /home/pi/M_Central/OFF_ID2.py");
}
if (isset($_POST['Light1ON']))
{
exec("sudo python /home/pi/M_Central/ON_ID4.py");
}
if (isset($_POST['Light1OFF']))
{
exec("sudo python /home/pi/M_Central/OFF_ID4.py");
}
if (isset($_POST['Lig']))
{
exec("sudo python /home/pi/M_Central/ON_BCAST.py");
}
if (isset($_POST['Lig0']))
{
exec("sudo python /home/pi/M_Central/OFF_BCAST.py");
}
?>

<br>
<body>
<div align="center">
<form method="post">

<h3>Mando general</h3>
<button id="search" class="btn" name="Lig"><h3>Encendido</h3></button>&nbsp;
<button id="search" class="btn" name="Lig0"><h3>Apagado</h3></button>&nbsp;
<button id="search" class="btn" name="Details"><h3>Detalles</h3></button>

<h3>Lampara 1:</h3>
&nbsp;
<br>
<button id="search" onclick="document.getElementById('myImage1').src='pic_bulbon.gif'" class="btn"
name="LightON"><h3>Encendido</h3></button>&nbsp;
<button id="search" onclick="document.getElementById('myImage1').src='pic_bulboff.gif'" class="btn"
name="LightOFF"><h3>Apagado</h3></button>&nbsp;
<a href="index.php?action=Detalles"><button id="search" class="btn" name="Details"><h3>Detalles</h3></button></a>

<h3>Lampara 2:</h3>
&nbsp;
<br>
<button id="search" onclick="document.getElementById('myImage2').src='pic_bulbon.gif'" class="btn"
name="Light1ON"><h3>Encendido</h3></button>&nbsp;
<button id="search" onclick="document.getElementById('myImage2').src='pic_bulboff.gif'" class="btn"
name="Light1OFF"><h3>Apagado</h3></button>&nbsp;
<button id="search" class="btn" name="Details"><h3>Detalles</h3></button>

</form>
</div>
<section>
</section>
</body>
```

C.3.2.4.3 Archivo consumogeneral.php

```
<?php
require_once("connection.php");
?>

<script src="views/modules/code/highcharts.js"></script>
<script src="views/modules/code/modules/exporting.js"></script>
<script src="views/modules/code/modules/export-data.js"></script>
<div id="container" style="min-width: 310px; height: 400px; margin: 0 auto"></div>
<div id="container2" style="min-width: 310px; height: 400px; margin: 0 auto"></div>
<div id="container3" style="min-width: 310px; height: 400px; margin: 0 auto"></div>
<script type="text/javascript">
Highcharts.chart('container', {
  chart: {
    type: 'line'
  },
  title: {
    text: 'Consumo de Corriente'
  },
  subtitle: {
    text: 'Source: Sistama de administración inteligente de alumbrado publico'
  },
  xAxis: {
    categories: [
<?php
```



```

        $sql = "SELECT * FROM test;";
        $result = mysqli_query($connection, $sql);
        while ($registros=mysqli_fetch_array($result)) {
            ?>
                '<?php echo $registros["Hora"] ?>',
            <?php
        }
        ?>
    ]
},
yAxis: {
    title: {
        text: 'Corriente (C)'
    }
},
plotOptions: {
    line: {
        dataLabels: {
            enabled: true
        },
        enableMouseTracking: false
    }
},
series: [{
    name: 'Lampara2',
    data: [
        <?php
        $sql = "SELECT * FROM test;";
        $result = mysqli_query($connection, $sql);
        while ($registros=mysqli_fetch_array($result)) {
            ?>
                <?php echo $registros["Corriente"] ?>,
            <?php
        }
        ?>]
    ]
});
</script>
<script type="text/javascript">
Highcharts.chart('container2', {
    chart: {
        type: 'line'
    },
    title: {
        text: 'Voltaje de Alimentación'
    },
    subtitle: {
        text: 'Source: Sistema de administración inteligente de alumbrado publico'
    },
    xAxis: {
        categories: [
            <?php
                $sql = "SELECT * FROM test;";
                $result = mysqli_query($connection, $sql);
                while ($registros=mysqli_fetch_array($result)) {
                    ?>
                        '<?php echo $registros["Hora"] ?>',
                    <?php
                }
                ?>
            ]
        },
    yAxis: {
        title: {
            text: 'Voltaje (V)'
        }
    },
    plotOptions: {
        line: {
            dataLabels: {
                enabled: true
            },
            enableMouseTracking: false
        }
    },
    series: [{
        name: 'Lampara2',
        data: [
            <?php
            $sql = "SELECT * FROM test;";
            $result = mysqli_query($connection, $sql);
            while ($registros=mysqli_fetch_array($result)) {
                ?>
                    <?php echo $registros["Voltaje"] ?>,
                <?php
            }
            ?>]
        ]
    }
});
</script>
<script type="text/javascript">
Highcharts.chart('container3', {
    chart: {
        type: 'area'
    },
    title: {
        text: 'Consumo de Potencia'
    }
});

```

```

    },
    subtitle: {
        text: 'Source: Sistema de administración inteligente de alumbrado publico'
    },
    xAxis: {
        categories: [
            <?php
            $sql = "SELECT * FROM test;";
            $result = mysqli_query($connection, $sql);
            while ($registros=mysqli_fetch_array($result)) {
                ?>
                '<?php echo $registros["Hora"] ?>',
            }
            ?>
        ],
        tickmarkPlacement: 'on',
        title: {
            enabled: false
        }
    },
    yAxis: {
        title: {
            text: 'Potencia (W)'
        },
        labels: {
            formatter: function () {
                return this.value ;
            }
        }
    },
    tooltip: {
        split: true,
        valueSuffix: ' Watts'
    },
    plotOptions: {
        area: {
            stacking: 'normal',
            lineColor: '#666666',
            lineWidth: 1,
            marker: {
                lineWidth: 1,
                lineColor: '#666666'
            }
        }
    },
    series: [{
        name: 'Lampara 2',
        data: [
            <?php
            $sql = "SELECT * FROM test;";
            $result = mysqli_query($connection, $sql);
            while ($registros=mysqli_fetch_array($result)) {
                ?>
                '<?php echo $registros["Potencia"] ?>',
            }
            ?>
        ]
    }
    ]
});
</script>

```

C.3.2.4.4 Archivo eventos.php

```

<?php
// CONEXION A LA BASE DE DATOS
require_once("connection.php");

//CONSULTA
//$solicitud = "SELECT * FROM test ORDER BY Evento DESC";
$solicitud = "SELECT * FROM test";

$resultado = mysqli_query($connection, $solicitud);

// Titulo de la tabla
echo "<h3>Tabla de Registro de Eventos</h3>";

// VACIADO DE DATOS
echo "<table align='center'><tr><td>Fecha y Hora</td><td>Estado</td><td>Identificador</td><td>Evento</td></tr>";

while ($fila = mysqli_fetch_array($resultado)) {
    # code...
    //echo $fila['Nombre'];

    echo "<tr>";
    echo "<td>". $fila['Fecha']. " ". $fila['Hora']. "</td>";
    echo "<td>". $fila['Estado']. "</td>";
    echo "<td>". $fila['Identificador']. "</td>";
    echo "<td>". $fila['Evento']. "</td>";
    echo "</tr>";
}
echo "</table>"

?>

```

C.3.2.4.5 Archivo salir.php

```
<?php
session_start();
session_destroy();
header("location: ../index.php");
?>
```

C.3.2.4.6 Archivo connection.php

```
<?php
$connection =mysqli_connect("localhost","MMC","password123","MMC");
?>
```

BIBLIOGRAFÍA

- [1] Concentrador de medidores digitales para sistema de mediciones remotas. José Guadalupe Fuentes Velázquez. 2016. IPN.
- [2] LoRaWAN What is it? A technical overview of LoRa and LoRaWAN. LoRa Alliance. Noviembre 2015.
- [3] Wireless Networking: Introduction to bluetooth and wifi.
- [4] <https://medium.com/beelan/haciendo-iot-con.lora-capitulo-1-qué-es-lora-y-lorawan-8c08d44208e8>
- [5] <http://www.luxgi.com/blog/wp-content/uploads/2015/08/escala-temperatura-de-color-bombillas-led.jpg>
- [6] <http://www.iluminet.com/historia-alumbrado-publico-estados-unidos-mexico/>
- [7] <http://parquesalegres.org/biblioteca/blog/importancia-del-alumbrado-publico/>
- [8] Automate the boring stuff with python, Practical Programming for total beginners. Al Sweigert. no starch press. 2015.
- [9] Programming PHP. .Kevin Tatroe, Peter MacIntyre and Rasmus Lerdorf. O'reilly. 2013.
- [10] Linux Command Line and Shell Scripting Bible. Richard Blum.Wiley. 2008
- [11] SX1276/77/78/79 -137MHz Low Power Long Range Transceiver, Datasheet Wireless, Senssind & Timing. Semtech. Agosto 2016.
- [12] <https://raspberrypi.stackexchange.com/questions/45570/how-do-i-make-serial-work-on-the-raspberry-pi3>
- [13] <https://raspberrypi.stackexchange.com/questions/40105/access-gpio-pins-without-root-no-access-to-dev-mem-try-running-as-root>
- [14] <http://thepiguy.altervista.org/art004/>
- [15] <https://blog.desdelinux.net/como-instalar-lamp-en-ubuntu/https://blog.desdelinux.net/como-instalar-lamp-en-ubuntu/>
- [16] <http://cpham.perso.univ-pau.fr/LORA/RPIgateway.html>
- [17] DRF1276DM LORA Long Range SX1278 Data Radio Modem. Dorji Applied Technologies. Mayo 2017
- [18] <https://pymysql.readthedocs.io/en/latest/>
- [19] Testing Data Radio Modem with Serial Port Tool. Dorji Applied Technologies. Mayo 2016.
- [20] Cisco Solution for LoRaWAN. Cisco. 2017
- [21] SX1272/3/6/7/8: LoRa Modem. Low Energy Consumption Design . Semtech. Julio 2013.
- [22] <https://www.raspberrypi.org/documentation/installation/installing-images/linux.md>
- [23] <https://blog.desdelinux.net/como-conectarse-a-una-red-wifi-usando-el-terminal/>
- [24] <https://forum.arduino.cc/>