



Universidad Autónoma de Querétaro
Facultad de Ingeniería

Sistema Web para seguridad y control en el Centro Académico de Tecnologías Avanzadas Sustentables, basado en algoritmos de procesamiento y optimización de sistemas embebidos inalámbricos.

Tesis

Que como parte de los requisitos para obtener el grado de Maestro en Ciencias en Mecatrónica

Presenta

Michel Geovanni Mújica Landeros

Dirigido por

Dr. Roque Alfredo Osornio Ríos

Sinodales

San Juan del Río, QRO Enero 2020



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Ciencias Mecatrónica

Sistema Web para seguridad y control en el Centro Académico de Tecnologías Avanzadas Sustentables, basado en algoritmos de procesamiento y optimización de sistemas embebidos inalámbricos.

Tesis

Que como parte de los requisitos para obtener el grado de

Maestro en Ciencias en Mecatrónica

Presenta

Michel Giovanni Mújica Landeros

Dirigido por

Dr. Roque Alfredo Osornio Ríos

Sinodales

Dr. Roque Alfredo Osornio Ríos

Presidente

Dr. Arturo Yosimar Jaen Cuellar

Secretario

Dr. Miguel Trejo Hernández

Vocal

Dr. David Alejandro Elvira Ortiz

Suplente

Dr. Juan José Saucedo Dorantes

Suplente

San Juan del Río, QRO México.

Enero 2020

© 2019 Michel Geovanni Mújica Landeros

Todos los derechos Reservados.

Dirección General de Bibliotecas UAQ

*Esta tesis va dedicada a mis padres Ana e Iván
Mis padrinos Gaspar y Coco
Mis amigos mas cercanos
Esas personas que son una luz en el camino desde lejos
Gracias por cuidarme y darme fuerza en todo momento*

Dirección General de Bibliotecas UAG

Agradecimientos

Inicio agradeciendo a la Universidad Autónoma de Querétaro por abrirme las puertas para realizar mis estudios de posgrado dentro de la institución, al personal del campus San Juan del Río por todo el apoyo y guía.

A mi asesor Doctor Roque, gracias por confiar en este proyecto y por permitirme la libertad creativa para llevar a cabo todas las ideas que surgieron; a mis profesores durante estos dos años, gracias por sus enseñanzas y paciencia.

Agradezco a Conacyt por la beca durante el tiempo que realice mis estudios de posgrado, espero retribuir con este proyecto y futuras aplicaciones, la confianza que se me fue brindada.

A mis padres Ana e Iván, mis Padrinos Gaspar y Coco que han cuidado de mi como si fuera su hijo desde siempre; gracias por creer en mi y apoyarme, incluso en momentos que yo mismo deje de hacerlo, en nuestros momentos más difíciles siempre tienen una palabra de fortaleza y voluntad y es algo que llevaré toda mi vida.

Mis amigos Asael, Carlos, Paco, Eliú, Jorge y los demás que necesitarían su propio capítulo para mencionarlos, gracias por su apoyo, amistad y darme una mano para seguir aprendiendo; ustedes son para mi una segunda familia.

Gracias a esas personas que llegan a la vida para cambiar los paradigmas, para sentir que el apoyo y la comprensión que sobrepasan los kilómetros, gracias por ayudarme a reinventarme, a crear mejores versiones de mi y a partir de ahora explotar toda la experiencia adquirida.

Tabla de contenido

1	Introducción.....	13
1.1	Planteamiento del problema.....	17
1.2	Justificación.....	20
2	Antecedentes.....	22
3	Hipótesis	27
4	Objetivos	28
4.1	Objetivo general	28
4.2	Objetivos Particulares	28
5	Material y Metodología.....	29
5.1	Procesadores y controladores.....	29
5.1.1	Hardware y Software libre	29
5.1.2	Raspberry Pi.....	30
5.1.3	Raspbian	31
5.1.4	Raspberry ZERO.....	33
5.1.5	Arduino	34
5.1.6	NodeMCU	37
5.2	Lenguajes de programación.....	41
5.2.1	Python.....	41
5.2.2	C++.....	42
5.2.3	Ruby & Rails.....	44
5.2.4	IDE.....	46
5.3	Herramientas WEB.....	50
5.3.1	Heroku	50
5.3.2	Base de Datos	53
5.3.3	Inicio de Sesión con Facebook	55
5.3.4	Postman	57
5.4	Protocolos de Comunicación.....	58
5.4.1	Protocolo SPI.....	58
5.4.2	Protocolo I2C.....	60
5.4.3	Protocolo HTTP.....	61
5.4.4	JSON	63
5.4.5	SSH.....	64
5.4.6	Modelo Vista Controlador	65
5.4.7	Programación OTA.....	66
5.5	Hardware.....	69
5.5.1	Router	70

5.5.2	Impresión 3D	72
5.5.3	Cerradura Magnética.....	75
5.5.4	Display Táctil	76
5.5.5	RFID.....	78
5.5.6	Tarjetas SD para Microcontroladores	81
5.5.7	Pantalla OLED.....	82
5.5.8	Fuentes de Alimentación	83
5.5.9	DC-DC	85
5.5.10	Celda Solar.....	86
5.5.11	Baterías Li	87
5.5.12	Cámara Raspberry Pi	90
5.5.13	Sensores Infrarrojo y ambiental	92
5.6	Algoritmo de configuración de Dispositivos IoT	95
5.7	Control de acceso sin conexión a Internet	97
5.8	Control de acceso con Internet.....	98
5.9	Dispositivo de Seguridad Nocturna	100
6	Resultados	102
6.1	Dispositivo de control de acceso sin conexión a Internet	102
6.2	Dispositivo de control de Acceso en CATAS	105
6.3	Identificador de tarjetas RFID.....	112
6.4	Aplicación WEB.....	114
6.5	Sensor ambiental con Celdas solares	117
6.6	Dispositivo de vigilancia nocturna	118
6.7	Tablet Pi	124
7	Conclusiones.....	127
8	Referencias.....	129

Índice de figuras

1-1 Internet of Things (Imagen Libre).....	14
1-2 Proyectos IoT 2018. IoT Analytics GmbH (2019).....	15
1-3 Industria 4.0 (Imagen Libre).....	16
1-4 Sistema Comercial. Mercado Libre (2019).....	18
2-1 Dispositivo Previo. (Elaboración propia).....	26
5-1 Raspberry Pi (Elaboración propia).....	31
5-2 Raspberry Pi Raspbian. Elaboración propia.....	31
5-3 Interfaz PIXEL. Elaboración propia.....	32
5-4 Raspberry ZERO. (Elaboración propia).....	33
5-5 Arduino Uno. (Elaboración propia).....	35
5-6 Placa Uno y Nano. (Elaboración propia).....	36
5-7 ESP8266. (Elaboración propia).....	37
5-8 Formato de pines NodeMCU V1.0. (Promtec).....	39
5-9 NodeMCU. (Elaboración propia).....	39
5-10 NodeMCU prueba Wi-Fi. (Elaboración propia).....	40
5-11 Logo Python. (Python org).....	41
5-12 Python Versión. (Elaboración propia).....	42
5-13 Compilador GCC. (Elaboración propia).....	43
5-14 Versión Ruby. (Elaboración propia).....	44
5-15 Versión Rails. (Elaboración propia).....	45
5-16 Empresas con Plataforma en Rails. (Elaboración propia).....	45
5-17 Arduino IDE. (Elaboración propia).....	47
5-18 IDE ATOM. (Elaboración propia).....	48
5-19 PlatformIO. (Elaboración propia).....	49
5-20 Heroku. (Elaboración propia).....	51
5-21 Heroku Configuración. (Elaboración propia).....	52
5-22 Base de Datos. (Elaboración propia).....	53
5-23 Consulta de Base de Datos. (Elaboración propia).....	55
5-24 Rails con Facebook Developers. (Elaboración propia).....	56
5-25 Creación de cuentas de prueba. (Elaboración propia).....	56
5-26 Cuenta Facebook. (Elaboración propia).....	57
5-27 Postman. (Elaboración propia).....	58
5-28 Protocolo SPI. (Elaboración propia).....	59
5-29 Protocolo I2C. (Elaboración propia).....	60
5-30 Protocolo HTTP. (Elaboración propia).....	61
5-31 Formato JSON. (Elaboración propia).....	63
5-32 SSH. (Elaboración propia).....	64
5-33 Modelo Vista Controlador. (Elaboración propia).....	66
5-34 Conexión OTA en IDE. (Elaboración propia).....	67
5-35 NodeMCU conectado por OTA. (Elaboración propia).....	68
5-36 Seguridad OTA. (Elaboración propia).....	69

5-37 Conexión Router. (Elaboración propia)	70
5-38 Conexión de Internet Interna. (Elaboración propia).....	71
5-39 Router Tequisquiapan. (Elaboración propia)	72
5-40 Router San Juan del Río. (Elaboración propia).....	72
5-41 Impresora 3D. (Elaboración propia)	73
5-42 Diseño propuesto 1. (Elaboración propia).....	74
5-43 Diseño propuesto 2. (Elaboración propia).....	74
5-44 Carcasa Raspberry Zero. (Elaboración propia).....	75
5-45 Cerradura Magnética. (Elaboración propia).....	76
5-46 Pantalla Lcd Táctil. (Elaboración propia).....	77
5-47 TAGs RFID. (Elaboración propia).....	79
5-48 Lector RFID MFRC522. (Elaboración propia).....	80
5-49 Tarjeta microSD y lectora SD para Microcontrolador. (Elaboración propia) ...	81
5-50 Conexión pantalla OLED. (Elaboración propia).....	83
5-51 Conexión OLED-Nano. (Elaboración propia).....	83
5-52 Fuente Radox. (Elaboración propia)	84
5-53 DC-DC. (Elaboración propia)	85
5-54 Celda Solar. (Elaboración propia)	87
5-55 Batería Litio. (Elaboración propia).....	88
5-56 RJMD36316. (Elaboración propia).....	89
5-57 Cámaras Raspberry Pi. (Elaboración propia).....	91
5-58 Sensor PIR. (Elaboración propia).....	92
5-59 Raspberry Zero y PIR. (Elaboración propia).....	93
5-60 Sensor ambiental con NodeMCU. (Elaboración propia)	93
5-61 Algoritmo Usuario-Sistema. (Elaboración propia).....	95
5-62 Control de Usuarios sin Internet. (Elaboración propia)	97
5-63 Control de Accesos con Internet. (Elaboración propia)	99
5-64 Control Nocturno de Acceso. (Elaboración propia)	100
6-1 DNI-9. (Elaboración propia).....	102
6-2 Carcasa y Alimentación DNI-9. (Elaboración propia)	103
6-3 DNI-9 Acoplado. (Elaboración propia).....	104
6-4 DNI-9 Instalado en SJR. (Elaboración propia).....	104
6-5 Pruebas MAJI-2. (Elaboración propia).....	105
6-6 Consumo de Corriente MAJI-2. (Elaboración propia).....	105
6-7 Producción MAJI-2. (Elaboración propia).....	106
6-8 Mensaje API exitoso. (Elaboración propia)	106
6-9 Conexión Eléctrica en CATAS. (Elaboración propia).....	107
6-10 Conexión Interruptor de carrera. (Elaboración propia)	108
6-11 Interruptor instalado en oficina. (Elaboración propia)	108
6-12 MAJI-2 producción. (Elaboración propia)	109
6-13 MAJI-2 instalado en CATAS. (Elaboración propia).....	110
6-14 Acceso con pulsera RFID. (Elaboración propia).....	111

6-15 Identificador de tarjetas. (Elaboración propia).....	112
6-16 Identificador con pulsera. (Elaboración propia).....	113
6-17 APP WEB CATAS. (Elaboración propia)	114
6-18 Menú APP. (Elaboración propia).....	114
6-19 Tabla de Accesos. (Elaboración propia)	115
6-20 Información de Acceso. (Elaboración propia).....	116
6-21 Sensor ambiental. (Elaboración propia).....	117
6-22 Raspberry P y cámara 5MP. (Elaboración propia)	118
6-23 Primer sistema nocturno. (Elaboración propia).....	119
6-24 Versión Producción Sistema Nocturno. (Elaboración propia).....	120
6-25 Sistema de Vigilancia portátil. (Elaboración propia)	121
6-26 Sistema instalado en CATAS. (Elaboración propia)	122
6-27 Foto Nocturna. (Elaboración propia)	123
6-28 Teclado Externo Tablet Pi. (Elaboración propia).....	124
6-29 Raspberry Pi B en Tablet Pi. (Elaboración propia).....	125
6-30 Tablet Pi y Teclado. (Elaboración propia).....	126

Dirección General de Bibliotecas UAG

Abstract

Technology in its different facets expands outside laboratories, it becomes part of our environment, from industrial areas, roads, cities, houses to our pockets with the use of smartphones. Their daily life now allows to generate more comfortable, safe and efficient spaces; Thanks to the internet, communication between different environments is achieved and the user experience immersed in the environment has recently improved. The present work proposes a design that brings together several cutting-edge technologies to generate a security, access and analysis system of the Academic Center for Sustainable Advanced Technologies (CATAS) of the Autonomous University of Querétaro (UAQ) that is monitored through an Internet application, storing the information of the different devices that monitor different needs of the relationship between the owner and the user.

Resumen

La tecnología, en sus diferentes facetas, se expande fuera de los laboratorios; ya sea pasando por caminos, ciudades, o zonas industriales puede llegar incluso a nuestro bolsillo con los teléfonos inteligentes, volviéndose parte de nuestro entorno. Esta cotidianidad actual permite generar espacios mas confortables, seguros y eficientes; junto a esto, la conectividad a internet logra una comunicación entre los diferentes entornos y, más recientemente, mejora la experiencia de usuario para aquellos inmersos en este ecosistema. El presente trabajo plantea un diseño que reúne diversas tecnologías de vanguardia para generar un sistema de seguridad, acceso y análisis del Centro Académico de Tecnologías Avanzadas Sustentables (CATAS) de la Universidad Autónoma de Querétaro (UAQ) el cual es monitoreado mediante una aplicación, almacenando información sobre los diferentes dispositivos que acceden a éste.

Keywords: IoT, RFID, Web, App, Internet of Things, Rails

1 Introducción

Desde la primera década del siglo XXI hablar de la conectividad en el mundo ha sido un tema común gracias a la gran cantidad de avances en diversas áreas de la ingeniería, las telecomunicaciones con el internet tuvieron un gran auge que al paso del tiempo se ha extendido a la gran mayoría de los sectores de la población, facilitando el acceso gracias a los Smartphone; de forma simultanea han surgido mejores dispositivos que facilitan las labores y mejoran la productividad del ser humano.

Los métodos de seguridad, control y automatización de procesos que en el siglo pasado se encontraban exclusivamente en zonas industriales, hoy se trasladan a todos los entornos humanos de manera rápida, gracias a los avances tecnológicos en materia de software y hardware.

La disminución de los costos y gran variedad de microcontroladores surgiendo, permite aumentar la gama de posibilidades; al mismo ritmo una gran variedad de sensores cada vez mas accesibles y eficientes permiten realizar y controlar mayor número de máquinas, conocer las características de nuestro entorno y adaptarlo a nuestras necesidades.

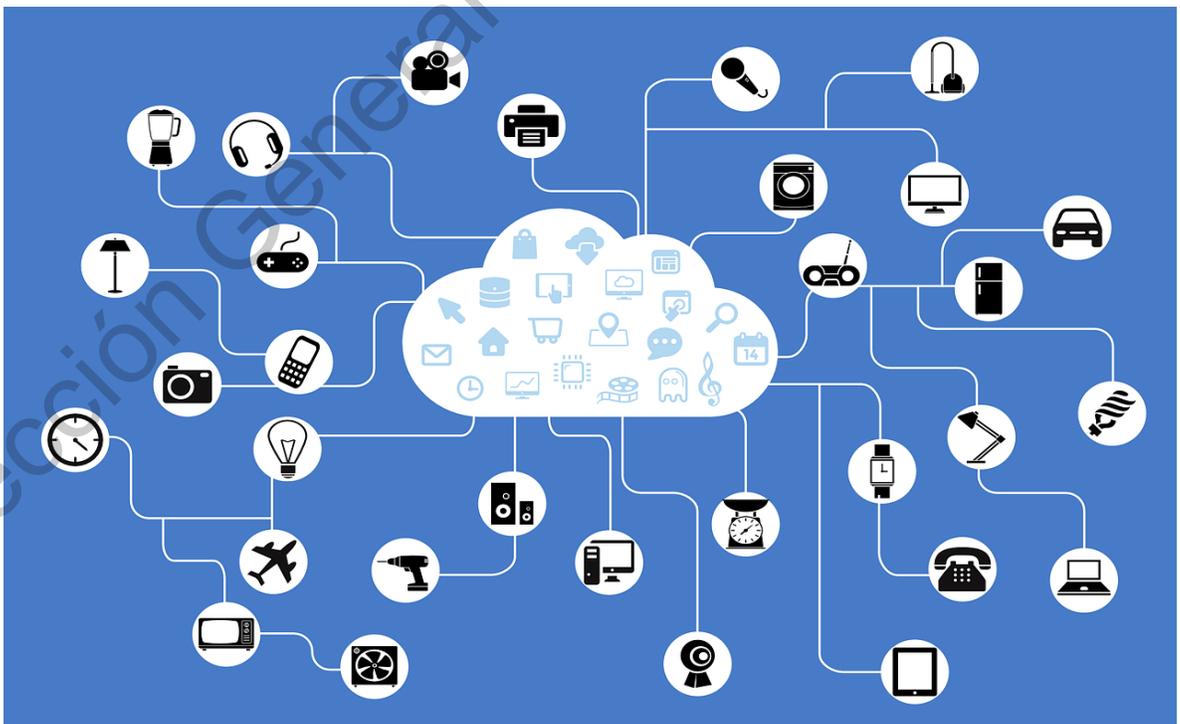
Una de las grandes características consecuentes de estos avances, es permitir el almacenamiento y análisis de datos que provienen de este grupo de sensores, permitiendo medir la interacción del ambiente y las personas con ellos, gracias a esto, se puede conocer de forma cuantitativa cuales son los elementos que cumplen de manera eficiente su trabajo y de la misma manera obtener la retroalimentación del alcance ocurrido.

Todas estas herramientas y procesos van tomando un necesario estándar dentro de la industria y las comunidades de tecnología conocido como Internet of

Things (Internet de las Cosas), generando protocolos de comunicación sugeridos que permiten una mejor interacción entre los dispositivos conectados a través de una red de computadoras. Permitiendo reaccionar y responder de forma cada vez mas rápida y eficiente a las distintas problemáticas y retos que se presentan.

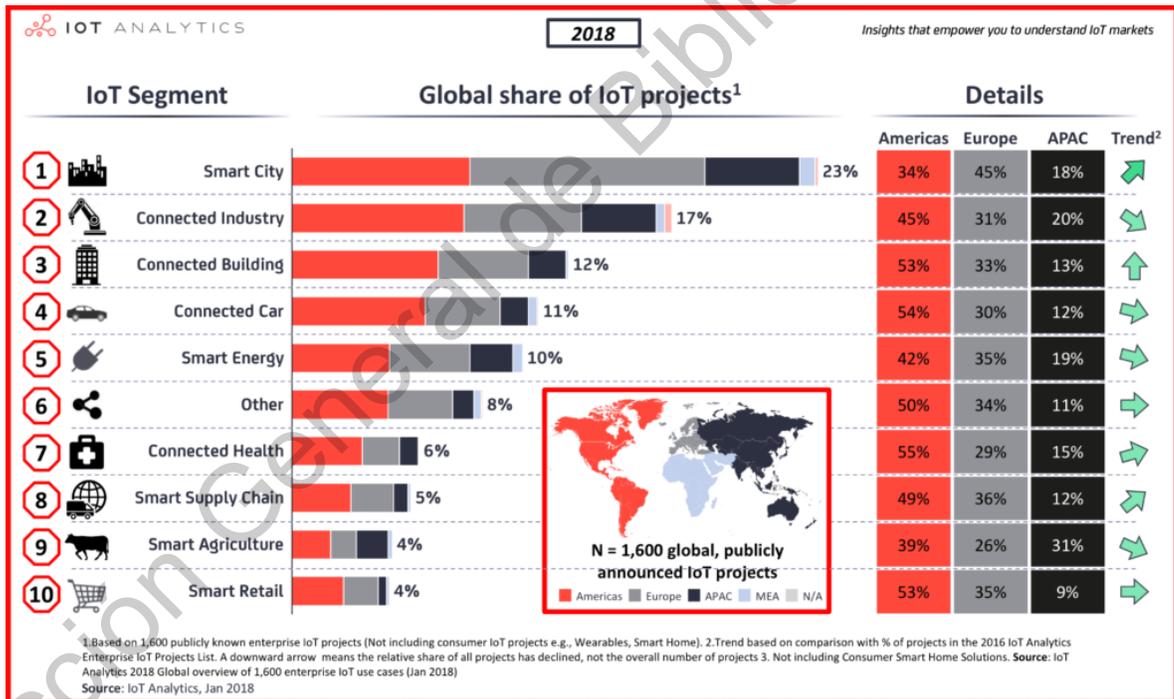
El Internet de las Cosas (IoT) no solo permite la comunicación entre los nuevos dispositivos y herramientas que han surgido con la digitalización de la sociedad, también permite que toda clase de artefactos que han evolucionado con nosotros se permitan entrar estos dispositivos inteligentes, tal como sería un reloj, una mochila, cerraduras o prendas de vestir, por mencionar algunos ejemplos.

En la figura 1-1 se aprecia una representación de diferentes dispositivos cotidianos, herramientas de trabajo y entretenimiento que se encuentran conectadas a una nube de información, la cual procesa y almacena la información de éstos y permite una posible interacción entre ellos de ser necesario.



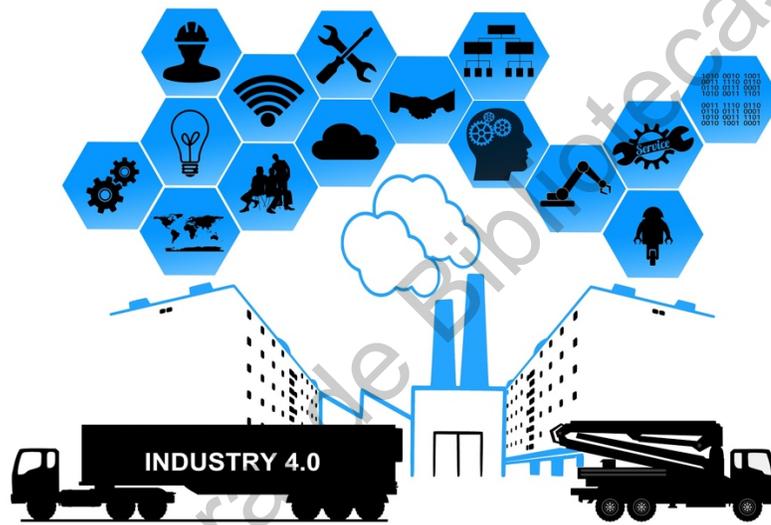
1-1 Internet of Things (Imagen Libre)

Existen empresas dedicadas a medir el uso y el alcance de las herramientas que proporciona el IoT, tal es el caso de IoT Analytics, como se muestra en la figura 1-2, en el año 2018 nuestro continente tiene una tendencia a aplicar la tecnología de la conectividad en las zonas industriales y en las ciudades inteligentes, este último con un porcentaje que se espera aumente en los próximos años, en tercer lugar se encuentra la conectividad de edificios, un paso primordial previo a la proyección de ciudades inteligentes; por lo tanto la implementación de seguridad y análisis de las condiciones del entorno son los primeros tópicos necesarios a cubrir para generar espacios inteligentes.



1-2 Proyectos IoT 2018. IoT Analytics GmbH (2019)

Como un concepto aún no definido en su totalidad se tiene el planteamiento de estar frente a una llamada Revolución 4.0, considerando las máquinas de vapor, la electricidad y la producción en masa, el desarrollo del computo como las tres primeras respectivamente. Dentro de esta propuesta, el IoT, la Inteligencia Artificial (AI), redes sociales y algunos otros servicios juegan un papel fundamental que permita volver esta idea parte de nuestro entorno como lo es ya el IoT.



1-3 Industria 4.0 (Imagen Libre)

1.1 Planteamiento del problema

Existen actualmente en el mercado circuitos y aplicaciones dedicadas a automatizar y controlar edificios departamentales, con sistemas de iluminación, temperatura o seguridad; sin embargo, el propósito de esta Tesis es generar toda la infraestructura física con los sistemas embebidos, sensores y actuadores necesarios así como los algoritmos y aplicaciones diseñadas para el edificio C.A.T.A.S. con la ventaja de ser un sistema personalizado a las necesidades del área y sus usuarios, mejorando el sistema previamente instalado.

En la imagen 1-4 se observa el ejemplo de un sistema comercial, distribuido en nuestro país, diseñado por una empresa China, aunque el sistema puede cubrir parcialmente las necesidades expuestas en esta tesis, presenta algunas problemáticas utilizarlo.

- Nula personalización: el dispositivo cuenta con reglas de programación prediseñadas por el fabricante, por lo cual se necesitaría adaptar el formato de cerraduras y control del edificio al del fabricante.
- Costos: El costo de los componentes se puede reducir considerablemente si se utilizan los propios elementos y se diseña toda una infraestructura personalizable.
- No cuenta con plataforma de internet: Una de las grandes desventajas que ofrece la plataforma que se muestra es la falta de una tarjeta de red, por lo tanto, nula conectividad con internet.



1-4 Sistema Comercial. Mercado Libre (2019)

Parte del proceso de diseño y dentro de las ventajas que ofrece tener un sistema propio sobre un sistema comercial es agregar sistemas y piezas que se adapten a las necesidades de usuarios y el edificio y no viceversa; así como generar dispositivos que puedan agregar varias tecnologías conjuntas que no se encuentran en un solo sistema comercial.

Las cámaras de seguridad que se ofrecen actualmente cuentan con tecnología Wifi o una lente nocturna, sin embargo, pocos sistemas cuentan ahora con ambas tecnologías por lo que se plantea que los diseños propios cuenten con ello.

Las plataformas web actualmente tienen un costo considerable por la necesidad de mantenimiento y funcionalidad del servidor y base de datos, ya que el sistema actual y el edificio cuenta con pocos usuarios, reduce los costos crear una interfaz propia diseñada exclusivamente por y para servicio de la Universidad Autónoma de Querétaro, manteniendo la personalización y escalamiento a medida de las necesidades y problemáticas que surjan con el aumento del uso de los dispositivos inteligentes dentro del campus.

C.A.T.A.S. es un centro tecnológico con diversos tipos de usuarios, por lo cual es necesario llevar un control de acceso de éstos y mantener la mayor autonomía posible en materia de seguridad. El centro cuenta con cerraduras magnéticas instaladas en la mayoría de las puertas que pueden ser integradas con un sistema digital de control de acceso mediante tarjetas con códigos de identificación personal, se necesita desarrollar una base de datos para almacenar los usuarios para almacenar el nivel de acceso que tiene en el edificio.

Se debe buscar la manera de diseñar e implementar sistemas complejos de control y automatización con los dispositivos que se encuentran disponibles para poder competir con el desarrollo urbano de la vida moderna.

Es necesario incrementar la cantidad de edificios inteligentes que se encuentran en nuestra región como los que se encuentran en los países más desarrollados ya que México se encuentra limitado solo a unos cuantos en todo el territorio y se debe incrementar la competitividad tecnológica y social que se pueda ofrecer.

1.2 Justificación

El sistema dedicado al control de entornos inteligentes al implementarse desde sus componentes básicos y diseñado de formada dedicada reduce los costos significativamente.

Mejorar el sistema de seguridad y el de sensores ya implementados dentro del edificio para integrarlos como uno solo, verificar desde su instalación y eficiencia aumenta el confort y control de recursos humanos y materiales dentro de las instalaciones.

Adaptar todo el conjunto de sensores y actuadores a un sistema embebido central para conectarse a una aplicación de Internet genera un grado de autonomía, control, confort y optimización de recursos dentro del edificio.

El uso de herramientas de software libre genera un mayor soporte por parte de una gran cantidad de programadores y desarrolladores, permitiendo generar aplicaciones mas robustas y flexibles sin la necesidad de utilizar herramientas privadas que aumentan el costo y reducen la flexibilidad y conectividad de la plataforma.

El diseñar un sistema de control y seguridad a la medida de las necesidades del edificio ofrece una gran cantidad de ventajas, la reducción de costos al elegir los componentes básicos para su construcción, así como el uso de software libre que no requiere comprar o mantener licencias para su uso; crear un sistema que puede se pueda mejorar o agregar bloques funcionales extras a los previstos en el modelo inicial sin la necesidad de realizar grandes cambios; actualización y seguimiento.

Implementar el ejercicio académico de desarrollar tecnologías propias, genera nuevos retos y oportunidades para futuras generaciones de investigadores y desarrolladores en un área de conocimiento que se encuentra en su curva de innovación.

Generar un edificio inteligente en la región genera un gran beneficio social puesto que marca una transición clave en el alcance y la integración de los sistemas digitales en la vida diaria, así como abre nuevas posibilidades al crecimiento industrial en diferentes escalas y sectores.

El diseñar sistemas propios por parte de la comunidad universitaria para su uso y servicio permite que los mismos alumnos y profesores ataquen las problemáticas que se generan y se mejore el sistema con el paso del tiempo y su crecimiento.

2 Antecedentes

Desde hace algunos años se ha desarrollado un concepto llamado Internet de las cosas (IoT por sus siglas en inglés), la idea como su nombre lo dice es conectar una serie de dispositivos cotidianos a Internet, como lo explica Paul(2017) es una interconexión entre los objetos equipados con una cierta inteligencia usada por los humanos en su día a día.

El concepto se ha aplicado en una gran cantidad de áreas, desde naves industriales a casa habitación, edificios departamentales o comerciales. Los edificios inteligentes surgen de la necesidad energética y de control debido a la enorme cantidad de recursos usados para su funcionamiento, dentro de los trabajos realizados por de la universidad en el área se encuentra Zamudio(2017) quien desarrolló un sistema de adquisición de señales de sensores para confort en el C.A.T.A.S (Centro Académico de Tecnologías Avanzadas y Sustentables) con el cual se dio un primer paso en la medición de diversas condiciones ambientales dentro del edificio en una interfaz receptora; otra adición importante al proyecto la dio Martínez(2017) con el desarrollo del monitoreo y control de acceso en el C.A.T.A.S., de esta manera se agregan elementos de seguridad, muy necesarios dentro del área.

Gracias al aumento que ha tenido el computo en la nube, otras herramientas crecieron y se pudieron convertir en tecnologías viables para el consumo de recursos, transferencia de información y control con la menor perdida de información a través de internet, Biswas (2014), propone una serie de herramientas y retos que desembocan del uso del IoT y el cómputo en la nube, gracias a la productividad y escalabilidad que se muestra con el desarrollo de más aplicaciones con estas tecnologías.

El término de nube data de la década de 1960 y consiste en redes de computadora conectadas para mantener información almacenada disponible vía internet al usuario en cualquier lugar, esto permite utilizar la información adquirida por una red de sensores de forma remota a otro dispositivo, como es el caso de los Smartphone o algún equipo de cómputo.

Bajo la perspectiva de trabajo de Fox(2019), uno de los formatos más actuales de los cuales se tiene referencia, utiliza las diferentes tecnologías del IoT divididas en tres etapas principales, recolección de información, métodos de comunicación y plataforma de cómputo; facilitando la comunicación desde los sistemas embebidos hasta la visualización del usuario, con el propósito de mantener conectada en comunicación de dispositivos toda una región del país determinada.

Las investigaciones actuales mantienen un amplio margen de posibilidades y adiciones al concepto de Internet de las cosas aplicado a edificios inteligentes, como Tavade (2017) utilizó una Raspberry Pi 3 para el Módulo de Control Electrónico (ECM) de un sistema de sensores, guardando las muestras en un servicio de nube, donde se mandó los datos a un PC receptor.

La integración de dispositivos electrónicos a los sistemas IoT debe realizarse de la forma mas natural posible, gracias a una serie de protocolos de comunicación utilizados, cada sistema nuevo que se agrega sigue una serie de reglas que permiten que se adapte de forma sencilla sin alterar demasiado el sistema al que se agrega como Pazos(2015) lo muestra.

Las Raspberry Pi y gran gama de sensores y actuadores son parte del concepto aplicado, Sruthy (2017) utilizó una cámara digital, un sensor infrarrojo (IR) entre otros sensores conectados de forma inalámbrica mediante módulos wifi a una tarjeta Raspberry pi, el cual se implementó en una casa habitación permitiendo a los usuarios conocer y modificar las condiciones en las que se encuentra.

Antes del desarrollo del Internet de las cosas ya existían algunos sistemas de seguridad como las cámaras de vigilancia, renovando este concepto y haciendo uso de las herramientas modernas Patil (2017) con una cámara fotográfica la cual capturaba imágenes bajo ciertas condiciones y comparaba con la imagen anterior, si se detectaba alguna diferencia entre las imágenes se enviaba un mensaje de texto al administrador para avisar movimientos en la habitación permite al cliente recibir en su dispositivo móvil avisos de algún cambio en el estado del lugar observado.

Con el fin de tener un mejor control de los usuarios de un sistema se puede utilizar tarjetas RFID, los cuales funcionan como una llave e identificador de los usuarios de la plataforma como lo muestra la implementación de Kossonon (2017), basado en el uso de tecnología RFID como sistema habilitador de un sistema embebido y una aplicación de IoT (Internet de las cosas); permitió una mejor interfaz para la interacción entre el usuario.

Las tarjetas RFID en el Internet de las cosas también se utilizan para control industrial, en su trabajo Gu (2017) desarrolló un sistema de identificación y control de vehículos con un protocolo de seguridad que permitió reducir la cantidad de errores.

Las aplicaciones que se implementan en el Internet de las cosas son ilimitadas, se pueden aplicar a resolver problemáticas sociales como Al Zamil *et al.* (2017) quienes proponen una metodología para desarrollar una casa habitación inteligente que mejora el rendimiento de aplicaciones de monitoreo de la atención médica para adultos mayores y personas con necesidades especiales.

Mejorar un proceso tan cotidiano dentro de los edificios como lo es buscar un lugar de estacionamiento puede ser optimizado, bajo la propuesta de realizar un estacionamiento inteligente de Rane (2017) con el uso de sensores infrarrojos, cámaras para detección de placas vehiculares y una aplicación móvil para llevar el control de los ingresos en el estacionamiento y avisar a los usuarios de zonas y lugares disponibles.

Durski (2011) describe una aplicación diseñada en Ruby on Rails, uno de los Frameworks mas populares en la actualidad para el desarrollo de aplicaciones web, el trabajo consiste en la revisión y control de un almacén mediante la gestión de base de datos que permite Rails, la parte visual de la aplicación (Front End) se realiza ya con el estándar de HTML 5, lo que mejora la conectividad con nuevos dispositivos inteligentes.

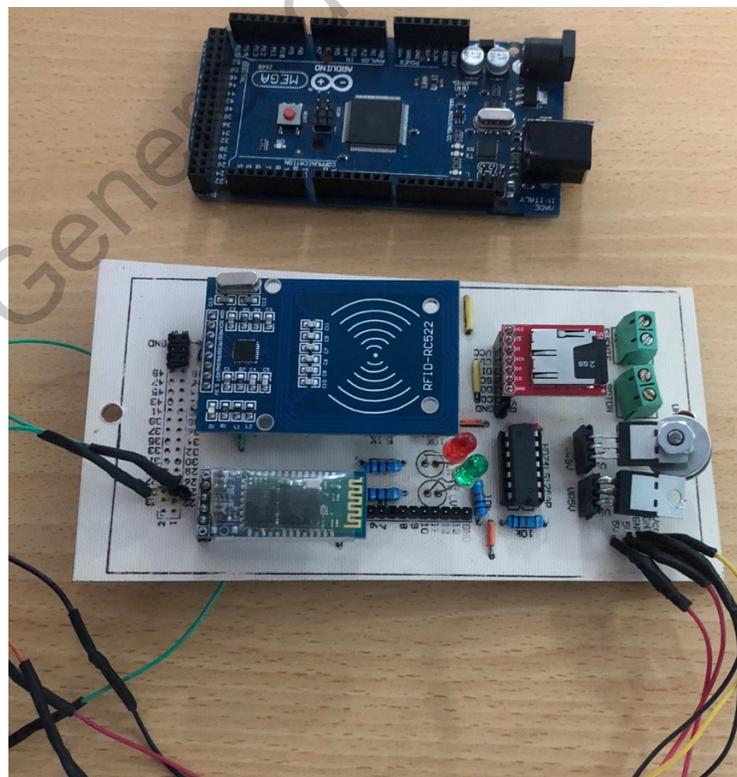
La conectividad de dispositivos con las aplicaciones web o servicios de Internet de forma usual se realizan mediante una API (Interfaz de programación de aplicaciones), que actúa como entrada y sistema moderador entre ambas partes, Garg (2019) realiza este proceso de comunicación, agregando en esta etapa una sección de seguridad y control para la transferencia de información.

Después del análisis de software y hardware implementado en los proyectos de aplicaciones y dispositivos inteligentes, es fundamental conocer el consumo energético que las plataformas consumen y lograr optimizarlo de la mejor manera posible, Salman (2016) diseña una aplicación del IoT enfocado a zonas residenciales, cuidando el uso de los dispositivos para mantener el gasto de potencia eléctrica reducida al mínimo.

El desarrollo realizado en los trabajos anteriores de la Universidad Autónoma de Querétaro es un gran paso en adaptar los conceptos de control, seguridad y automatización a las necesidades del edificio C.A.T.A.S. y como se aprecia en los demás trabajos realizados actualmente alrededor del mundo en esta área, es necesario agregar las comunicaciones entre dispositivos vía internet, mejorar su autonomía y la flexibilidad de control para los administradores y usuarios del edificio.

En la figura 2-1 se muestra un dispositivo previamente instalado dentro del edificio con una serie de sensores y un controlador para activar una cerradura magnética, pero ninguno de los elementos se encontraba ya en funcionamiento.

El dispositivo actual no cuenta con conexión a internet y el consumo energético de la placa del controlador es mucho mayor al necesario para mandar las instrucciones que se necesitan para operar la cerradura; además de una serie de periféricos que no cumplen alguna función.



2-1 Dispositivo Previo. (Elaboración propia)

3 Hipótesis

A partir del desarrollo de algoritmos de procesamiento y comunicación de sistemas embebidos inalámbricos, mejorar la seguridad del C.A.T.A.S. así como la interacción del usuario con el entorno, mediante sensores de condiciones ambientales, cámaras y cerraduras inteligentes, gestionadas mediante dispositivos inteligentes.

Dirección General de Bibliotecas UAO

4 Objetivos

4.1 Objetivo general

Desarrollar un sistema de monitoreo para seguridad con algoritmos de procesamiento y control basado en protocolos de comunicación inalámbrica y aplicación web.

4.2 Objetivos Particulares

- Utilizar el sistema de cerraduras automatizados con tarjetas embebidas otorgando accesos por módulos RFID para un control de acceso de los usuarios mediante tarjetas de personal con diferentes grados de acceso.
- Diseñar aplicación WEB para administradores del edificio montada en Raspberry PI con roles asignables de administración y usuarios para una mejor experiencia de usuario.
- Implementar un sistema de módulos con Sensores, con un sistema dedicado de control que se comuniquen con el usuario/administrador a una terminal con internet para monitoreo de edificio.
- Generar la base de datos de los accesos con la información del sitio al que se ingresa, la hora y la identificación de llave de usuario.
- Agregar elementos de sustentabilidad energética para alimentar los componentes y controladores utilizados en esta tesis.
- Optimizar los procesos de transmisión de información mediante algoritmos y protocolos de comunicación, así como optimizar el consumo energético de los dispositivos electrónicos.

5 Material y Metodología

Se utilizaron diversos instrumentos de software y hardware, así como diferentes protocolos de comunicación que en conjunto generan los diferentes dispositivos que interactúan con el usuario, se describen sus características principales, detalles técnicos, formas básicas y su actividad dentro del diseño de los dispositivos.

5.1 Procesadores y controladores

Se utilizan distintas placas de cómputo y controladores en esta tesis, contando cada una con diferentes versiones para sus diversas aplicaciones, algunas de ellas como sistemas dedicados y otras reprogramables a consideración y necesidad de los usuarios, la mayoría de estas placas electrónicas se basan en el principio del software y el hardware libre.

5.1.1 Hardware y Software libre

El concepto del hardware libre consiste en dispositivos cuyos diagramas y especificaciones son de acceso público, lo cual permite su replicación por parte de cualquier persona o empresa, pudiendo modificar o agregar componentes desde su creación o manteniendo el diseño original, todas funcionales y compatibles con la misma base.

El software libre por su parte son programas informáticos cuyos códigos son de libre acceso para usarlos o modificarlos en distintas aplicaciones. El libre acceso por parte de los desarrolladores iniciales de los proyectos permite que su difusión sea mayor, de tal manera que la comunidad de usuarios no solo se beneficia con su uso si no que se contribuye a su mejora y mantenimiento, desarrollando proyectos y contribuyendo con códigos que permiten aumentar las posibilidades de la plataforma.

También permite una considerable reducción de costos al momento de realizar un proyecto ya que no se necesita utilizar herramientas de hardware y software con licencias usualmente designadas a resolver puntuales problemas industriales.

5.1.2 Raspberry Pi

La Raspberry Pi es una tarjeta de computo creada para facilitar el acceso a la educación informática, fue lanzada en el año 2012 y existen una gran cantidad de variaciones de este modelo, todas las placas, poseen una serie de conectores y periféricos para agregar distintos elementos dependiendo del propósito para el cual se busca.

El diseño general de la Raspberry Pi incluye:

- Un Chipset Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz
- Un procesador gráfico (GPU) VideoCore IV
- Un módulo de 512 MB de memoria RAM
- Un conector de RJ45 conectado a un integrado lan9512 -jzx de SMSC que proporciona conectividad a 10/100 Mbps
- 2 buses USB 2.0
- Una Salida analógica de audio estéreo por Jack de 3.5 mm.
- Salida digital de video + audio HDMI
- Salida analógica de video RCA
- Pines de entrada y salida de propósito general
- Conector de alimentación microUSB
- Lector de tarjetas SD
- Consumo de potencia 3.5 W



5-1 Raspberry Pi (Elaboración propia)

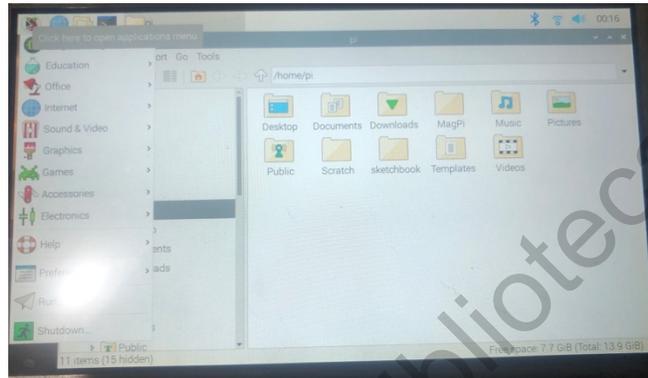
5.1.3 Raspbian

Existen diferentes sistemas operativos que se pueden instalar dentro de la placa, que son provistos por la misma comunidad que gestiona el proyecto Raspberry, la mayoría de éstos se encuentran basados en algún distrito Linux/Unix. Desde el año 2015 las variantes del sistema operativo Raspbian optimizado para los procesadores ARM utilizados en la placa.



5-2 Raspberry Pi Raspbian. Elaboración propia

Raspbian utiliza PIXEL (Pi Improved X-Window Environment, Lightweight) como interfaz gráfica para el usuario, lo que permite un modo de operación más común a la ya acostumbrada forma de los sistemas operativos de Microsoft mucho más comunes entre los usuarios.



5-3 Interfaz PIXEL. Elaboración propia

Las salidas de Pines con las que cuentan las Raspberry son una de las principales características que las distinguen de otras placas de procesamiento, ya que estos permiten agregar además de los periféricos antes mencionados todo tipo de instrumentación que se pueda implementar en algún proyecto con IoT.

El sistema operativo permite un acceso a los pines de manera más natural, utilizando diferentes lenguajes de programación, ventaja que se explicará en otra sección; aunque puede ser utilizando las herramientas de la interfaz gráfica, se puede prescindir de ella para acceder a otras características, otro de los modelos de la raspberry utilizados en el desarrollo de los dispositivos IoT que se beneficia de esta maleabilidad es la Raspberry Zero.

5.1.4 Raspberry ZERO

Los modelos de Raspberry son bastante portátiles y versátiles ya que depende del uso de periféricos conectados dentro del sistema, tales como teclado o monitor, sin embargo existe un modelo de la tarjeta llamado Raspberry ZERO el cual es mucho mas pequeño que las demás tarjetas, de tan solo 65 mm x 30 mm, cuenta con las capacidades de las demás compañeras de la marca, pero con un tamaño y precio mucho menor, considerando que su reducción de tamaño se debe en gran parte a la reducción de periféricos, aun se cuenta con un puerto micro USB y una salida mini HDMI que permite la conexión con un monitor y teclado en caso de ser necesario acceder de forma visual a la tarjeta.



5-4 Raspberry ZERO. (Elaboración propia)

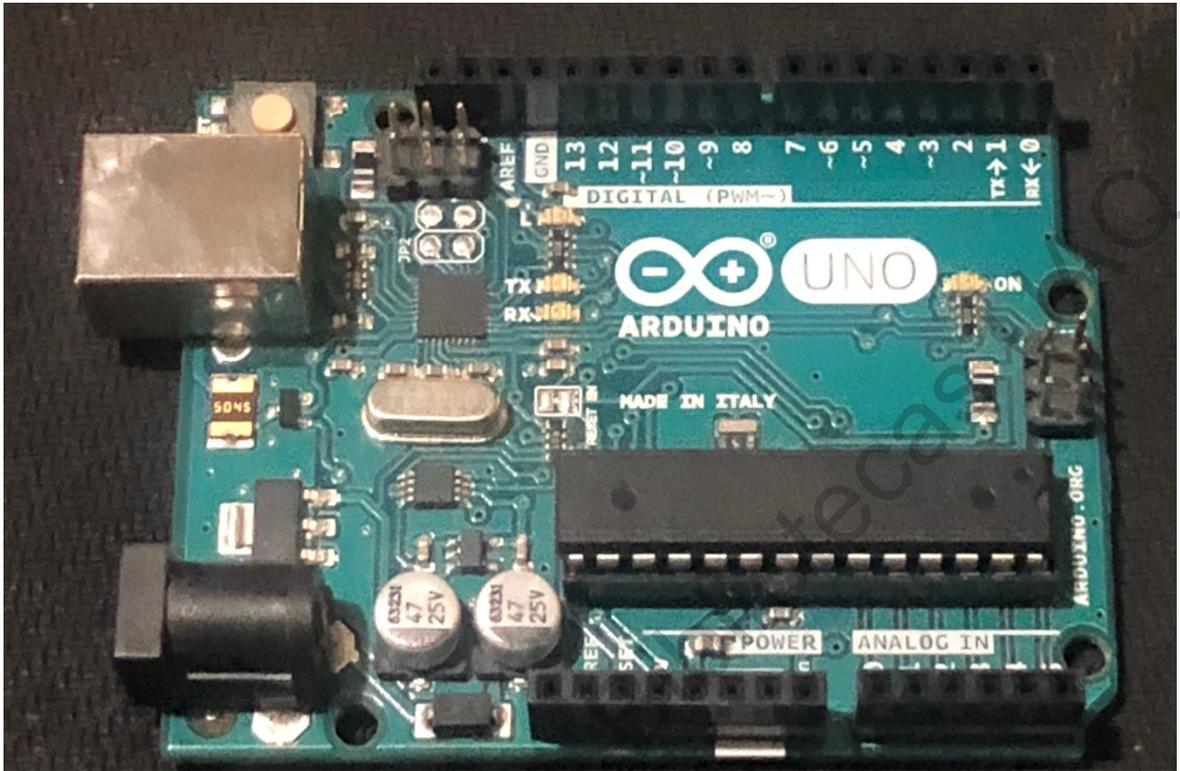
El procesador de la Zero esta reducido en un 40% de su capacidad original con una velocidad de 1 GHz, esta disminución de procesamiento en principio no representa desventaja alguna ya que como se mencionó antes se puede prescindir de la interfaz grafica del Raspbian y acceder únicamente por la línea de comandos a la Raspberry en su mayoría, ayudando así a que el procesamiento de la tarjeta se concentre exclusivamente en las funciones necesarias y no en la interfaz de usuario.

5.1.5 Arduino

Arduino es una placa electrónica de código abierto, basa en hardware y software libre, muy popular en el mundo Maker (Comunidad de desarrolladores de proyectos caseros, de la misma manera que la Raspberry, pero con una mayor cantidad de versiones y variaciones; ya que se enfoca en el uso de diferentes microcontroladores cuya plataforma permite reprogramar de manera sencilla para los fines del usuario.

El proyecto Arduino nace en 2003 en el Instituto de diseño Ivrea en Italia, con la finalidad de tener una alternativa económica para los estudiantes que no podían acceder a las placas de controladores utilizadas en ese momento. Se puede programar en cualquier sistema operativo y sus variantes Windows, GNU/Linux y Mac Os.

El modelo insignia del proyecto es llamado Arduino Uno, es la placa mas replicada de la familia y la que usualmente se utiliza de forma didáctica, en nuestro caso esta placa se utiliza únicamente para prototipos, conceptos y probar módulos de forma aislada.



5-5 Arduino Uno. (Elaboración propia)

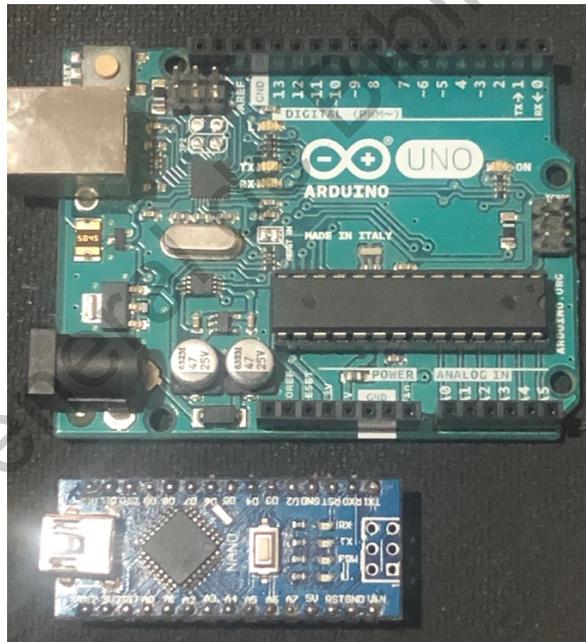
El modelo utilizado en uno de los dispositivos diseñados es el Arduino Nano, básicamente consta de las mismas especificaciones que la placa Uno pero su empaquetad y forma de la placa es mas pequeño, lo que permite una mejor aplicación al desarrollar equipos mas pequeños y prácticos.

Como ya se menciona, el microcontrolador es de hardware libre por lo que pueden existir diferentes variaciones, pero sus especificaciones técnicas mas comunes entre los productos comerciales con el nombre de Arduino nano son los siguientes.

- Microcontrolador ATmega328
- Arquitectura AVR
- Voltaje de operación 5v
- Memoria Flash 32 KB (2KB usados en bootloader)
- Velocidad de reloj 16 MHz

- Pines analógicos 8
- EEPROM 1KB
- Corriente Continua en Pines 40mA
- Voltaje de entrada 5-12V
- Pines digitales 12
- Tamaño 18x45 mm
- Peso 7 gr.

En la siguiente imagen se puede apreciar una comparativa entre el diseño de la Placa Uno y la Nano.



5-6 Placa Uno y Nano. (Elaboración propia)

5.1.6 NodeMCU

NodeMCU es el nombre que se le da a la placa de desarrollo de código libre basado en el ESP8266 y al firmware originalmente utilizado para trabajar con la placa. El ESP8266 es un módulo de wifi producido por el fabricante Espressif Systems en Shanghai. Su primer modelo llegó en el año 2014 con la versión ESP-01 para conectarse a redes Wifi y realizar conexiones sencillas de TCP/IP.



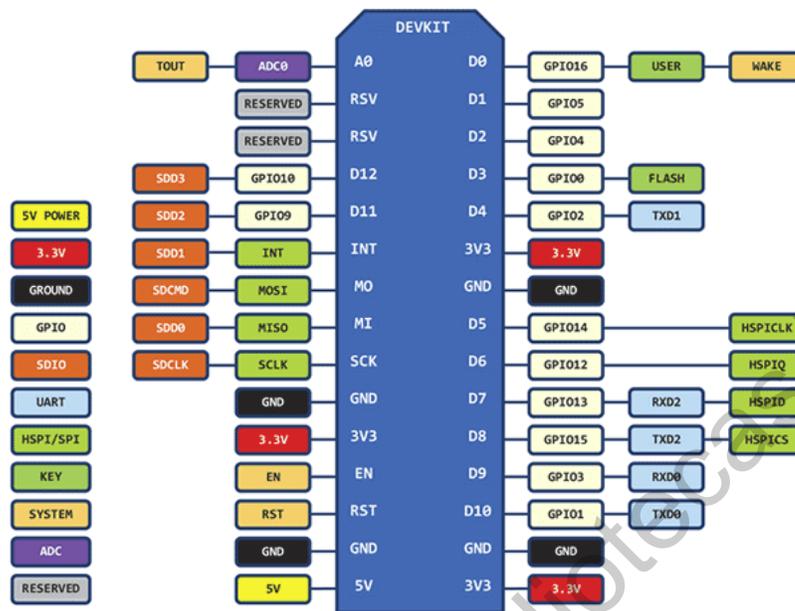
5-7 ESP8266. (Elaboración propia)

Con el aumento de la comunidad de software libre utilizando esta herramienta surgió mayor cantidad de información de la misma y se empezaron a desarrollar placas de trabajo con este controlador dentro de ellas, las placas agregan pines, periféricos y sistemas de alimentación según sus propósitos, pero toda placa NodeMCU lleva el ESP8266 dentro de la misma, dentro del módulo Wi-Fi encontramos el microcontrolador Tensilica L 106 el cual cubre la función de recibir las instrucciones del módulo Wifi que a su vez se conectan a los periféricos de las distintas placas.

Debido a que es una placa de desarrollo bastante reciente y de hardware libre, existen aún algunas confusiones en los nombres de las versiones y sus diferencias dependiendo del fabricante que llegan a ser muy sutiles. Aunque de manera general cuenta con los siguientes elementos.

- Puerto micro USB y conversor serie USB
- Alimentación a través de USB
- Pines de conexión
- LED indicador y botón de Reinicio integrados
- Microcontrolador a 32 bits
- 64 KiB RAM
- FLASH 512KiB a 4MiB
- Interfaz SPI, I2C
- UART
- ADC 10 bits
- Wi-Fi 2.4 GHz soporta WPA/WPA2

El modelo que se utiliza en esta tesis es conocido como la segunda generación V1.0/V2, tiene algunos pines digitales mas por lo que se acopló de manera precisa a las necesidades de este proyecto, y su tamaño estándar permite que se ajusta a las placas de prototipo y a los modelos de tarjetas de desarrollo prefabricadas, ocupando 8 hileras y dejando una de cada lado disponible para conexiones. En la siguiente imagen se definen de manera esquemática los pines de conexión para alimentación y agregar otros elementos de sensores o actuadores.



5-8 Formato de pines NodeMCU V1.0. (Promtec)

De manera física en la imagen 5-9 se ve el circuito y las etiquetas colocadas al momento de realizar los primeros prototipos y conocer la forma de salida de los pines.



5-9 NodeMCU. (Elaboración propia)

Para probar el módulo de Wifi de la placa, sus conexiones básicas y la salida serial de conexión de la placa a un ordenador, se realizó la conexión desde el NodeMCU a un servidor global que permite conocer la fecha y hora exacta en diferentes zonas del mundo. Obteniendo el resultado favorable que se observa en la imagen 5-10, los datos se obtienen y son enviados al puerto serial de la computadora, comparándose con los que se tienen dentro de ella.



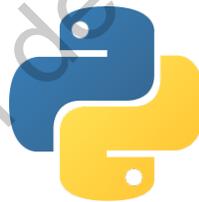
5-10 NodeMCU prueba Wi-Fi. (Elaboración propia)

5.2 Lenguajes de programación

Las distintas herramientas de computo y procesamiento mostradas en la sección 5.1 requieren instrucciones para trabajar, expresadas de una manera que la máquina pueda entender, en esta sección se describen algunos de los lenguajes de programación utilizados para este propósito, así como los IDE utilizados para escribir y/o compilar estos lenguajes de ser necesarios se especifica.

5.2.1 Python

Apareció de manera formal en 1991 y fue diseñado por Guido Van Rossum, actualmente es administrado por la Python Software Fundación posee una Licencia de código abierto y a partir de su versión 2.1.1 es compatible con la Licencia Pública general de GNU

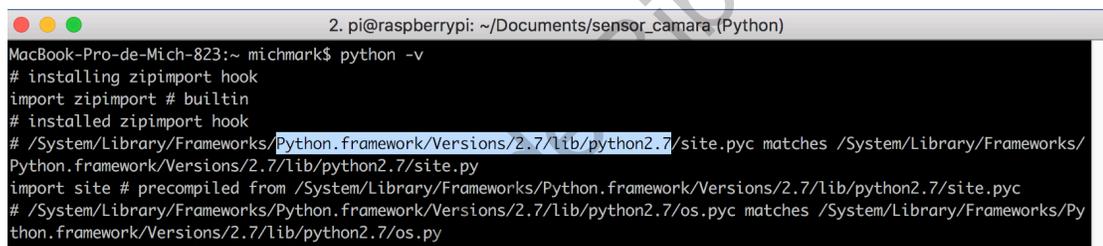


5-11 Logo Python. (Python org)

Python es un lenguaje de scripting independiente del sistema operativo y orientado a objetos, puede ser implementado en una gran cantidad de áreas, desde aplicaciones de internet, servidores, cálculo numérico, procesamiento y adquisición de señales y una gran cantidad funciones más, gracias a la gran cantidad de librerías e interfaces que pueden ser agregadas de una forma sencilla. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

Las versiones actuales de Python son 2, la 2.7 y la 3.7 al momento del desarrollo de este documento, cada una de ellas con pequeñas diferencias, sin embargo, para lograr una conectividad mas ideal con las otras herramientas aquí utilizadas se utiliza alguna de las versiones disponibles mediante el gestor de paquetes provenientes de la rama 2.7.

Todo el desarrollo de este proyecto se llevo acabo en plataformas UNIX, por lo tanto, ya tienen una versión precargada de este lenguaje para su uso interno, si se desea realizar el uso y verificación dentro de alguna plataforma Microsoft se puede consultar cualquier guía de la comunidad de Python en línea. En la siguiente imagen se hace una toma de pantalla en un sistema Unix para verificar que la versión de Python instalada es la correspondiente al propósito del proyecto.



```
2. pi@raspberrypi: ~/Documents/sensor_camara (Python)
MacBook-Pro-de-Mich-823:~ michmark$ python -v
# installing zipimport hook
import zipimport # builtin
# installed zipimport hook
# /System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site.pyc matches /System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site.py
import site # precompiled from /System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site.pyc
# /System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/os.pyc matches /System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/os.py
```

5-12 Python Versión. (Elaboración propia)

5.2.2 C++

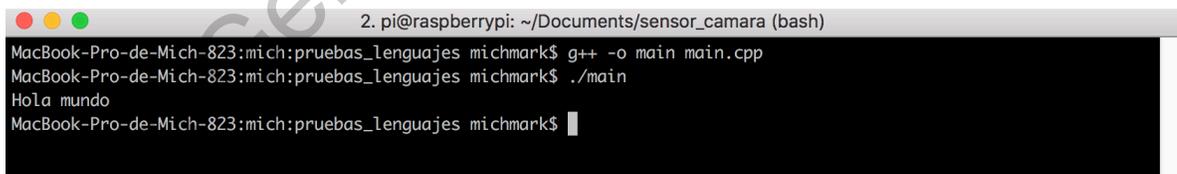
Diseñado en 1979 por Bjarne Stroustrup, se buscaba extender dentro del lenguaje C la manipulación de objetos, se han ido agregando además de eso, paradigmas de programación estructurada, bajo la norma ISO/IEC 14882.2017 actualmente, su nombre se deriva de una expresión dentro del lenguaje para referirse a un incremento de C. Sus características principales son las siguientes.

- Sintaxis heredada de C.
- Programación Orientada a Objetos.
- Agrupación de Instrucciones.
- Lenguaje didáctico, muchos lenguajes derivan de este.

- Portátil, permite el uso de diferentes sistemas operativos.
- Lenguaje de Alto nivel.
- Consta de palabras reservadas.
- Tipos de datos.
- Manejo de librerías.
- Herencia.

Como se muestra, también es un lenguaje multiplataforma lo que permite que un mismo proyecto pueda ser construido, compilado o depurado en diferentes terminales, la redacción de las sentencias puede ser realizado en algún IDE que permita la escritura y compilación del lenguaje dentro del mismo entorno o generar las sentencias de lenguaje y posteriormente ser compiladas.

El compilador que se utiliza para generar los comandos máquina desde el lenguaje C++ en este proyecto es GCC, es parte de la GNU Compiler Collection, es el compilador mas utilizado por los sistemas UNIX, en la imagen 5-13 se observa un ejemplo clásico



```
2. pi@raspberrypi: ~/Documents/sensor_camara (bash)
MacBook-Pro-de-Mich-823:mich:pruebas_lenguajes michmark$ g++ -o main main.cpp
MacBook-Pro-de-Mich-823:mich:pruebas_lenguajes michmark$ ./main
Hola mundo
MacBook-Pro-de-Mich-823:mich:pruebas_lenguajes michmark$
```

5-13 Compilador GCC. (Elaboración propia)

5.2.3 Ruby & Rails

Ruby es un lenguaje de programación interpretado y orientado a objetos, fue desarrollado por el programador Yukihiro “Matz” Matsumoto en 1993, presentado en 1995, su descripción al escribir el lenguaje es parecida a Python y de la misma manera cuenta con una licencia de Software Libre, el lenguaje de rige bajo la filosofía de mejorar la productividad y la diversión del desarrollador, al momento de la redacción de este documento las versiones de la rama 2.6 son la última estable que se encuentra usando para producción de proyectos.

A terminal window screenshot showing the command 'ruby -v' being executed on a MacBook. The output displays the Ruby version as 2.6.3p62, along with the date (2019-04-16), revision number (67580), and architecture (x86_64-darwin17). The terminal title bar indicates the user is '2. pi@raspberrypi' in a directory '~/Documents/sensor_camara' using a 'bash' shell.

```
2. pi@raspberrypi: ~/Documents/sensor_camara (bash)
MacBook-Pro-de-Mich-823:~ michmark$ ruby -v
ruby 2.6.3p62 (2019-04-16 revision 67580) [x86_64-darwin17]
MacBook-Pro-de-Mich-823:~ michmark$
```

5-14 Versión Ruby. (Elaboración propia)

Rails es un Framework de aplicaciones web de código abierto escrito en lenguaje Ruby, sigue el paradigma MVC (Modelo, Vista, Controlador), intenta minimizar la labor del desarrollador y concentrar la mayor cantidad de herramientas con menor configuración que permita realizar proyecto de una gran escala productiva y de mantenimiento, utiliza una escala de librerías llamadas Gemas que permiten agregar funcionalidades y es su formato oficial de Bibliotecas.

Permite escribir un buen código evitando que se repitan fragmentos de código de forma innecesaria y favoreciendo la convención antes que la configuración, en este momento las ramas de la versión 5.2 son las que se encuentran en funcionamiento.

```
2. pi@raspberrypi: ~/Documents/sensor_camara (bash)
MacBook-Pro-de-Mich-823:~ michmark$ rails -v
Rails 5.2.3
MacBook-Pro-de-Mich-823:~ michmark$
```

5-15 Versión Rails. (Elaboración propia)

La filosofía de Rails se basa en “No te repitas”, de manera que los componentes que deben estar conectados no necesitan que se escriban líneas de código de más para ello y evitar la redundancia programática, manteniendo la menor cantidad de bucles posibles.

Una de las razones para elegir estas herramientas de computo sobre muchas otras existentes dentro de las opciones libres o usadas dentro de la academia, es su alcance, Rails es una herramienta utilizada por un gran número de empresas que hoy encabezan la lista de servicios tecnológicos en todo el mundo



5-16 Empresas con Plataforma en Rails. (Elaboración propia)

5.2.4 IDE

Un IDE (Entorno de Desarrollo Integrado) no es un lenguaje de programación sin embargo se anexa en esta sección pues es la herramienta que se utiliza para escribir los lenguajes de programación utilizados en esta tesis. Un IDE puede constar de un editor de código, herramientas de construcción, compilador, un depurador, estas y otras funciones propias de cada interfaz están diseñadas para mejorar la productividad del programador, un mismo IDE puede ser usado para distintos lenguajes y control de tarjetas como es el caso de los que se muestran en este apartado.

5.2.4.1 *Arduino IDE*

Con el proyecto Arduino también fue lanzada una plataforma para escribir código y conectarse con las tarjetas de la comunidad, esta herramienta también es multiplataforma por lo cual es muy versátil, aunque cuenta únicamente con los elementos más básicos de un IDE, tener una versión estable en los principales sistemas operativos, es una de las grandes ventajas que posteriormente ayudaron a una mejor interacción dentro del proyecto.

Dentro de esta plataforma podemos programar las tarjetas NodeMCU y Arduino en una pequeña variante de lenguaje Java, mandando por default dos funciones principales al controlador, las funciones setup y loop. En la figura 5-17 se muestra la forma básica al momento de agregar funciones con esta plataforma a los microcontroladores.



```
Arduino  Archivo  Editar  Programa  Herramientas  Ayuda
sketch_nov02b Arduino 1.8.7
sketch_nov02b
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
Arduino Nano, ATmega328P (Old Bootloader) en /dev/cu.wchusbserial143420
```

5-17 Arduino IDE. (Elaboración propia)

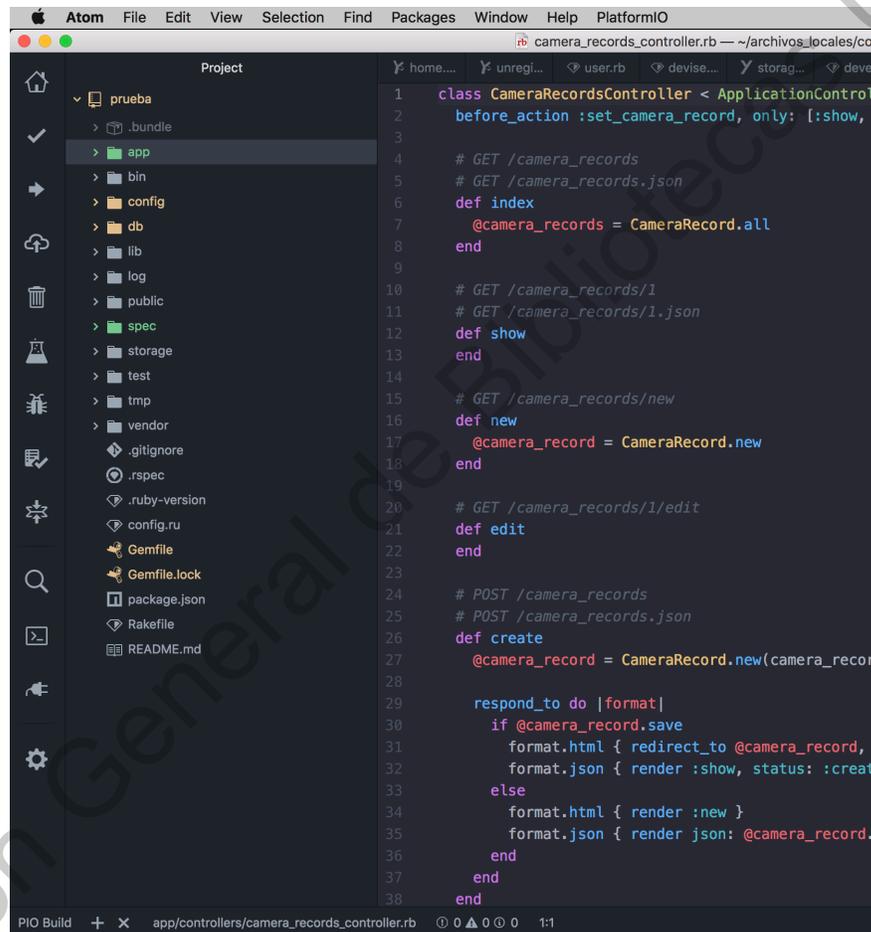
La función setup se utiliza para agregar las sentencias que se repetirán únicamente una vez al momento que el microcontrolador ejecute sus funciones y la función Loop es el ciclo que permite almacenar las instrucciones que serán repetidas de forma permanente mientras el controlador este encendido o sean activadas mediante alguna señal o acción externa.

5.2.4.2 Atom

Es un editor para lenguajes de programación, es de código abierto y es compatible con macOS, Linux y Windows, tiene soporte para plugins de Node.js y control de versiones de Git integrado desarrolladas por GitHub; fue construido utilizando tecnologías web, la mayoría de los paquetes que maneja son de software libre y tiene una gran cantidad de lenguajes que pueden trabajar en su entorno.

Su lanzamiento inicial se da el 26 de febrero de 2014 y su versión 1.0 es liberada el 25 de junio de 2015, a partir de la versión 1.5.1 soporta lenguajes como

C, C++, C#, Java, JavaScript, COBOL, CoffeeScript, Go, HTML, Less, Make, Objective-C, Perl, PHP, Python, Ruby, Ruby & Rails, SQL, XML, YAML, muchos de los mencionados son utilizados dentro del proyecto de la aplicación web integrados dentro de Rails, para cubrir diferentes acciones ya sea de internet, base de datos o manejo de sentencias.



```
1 class CameraRecordsController < ApplicationController
2   before_action :set_camera_record, only: [:show,
3
4   # GET /camera_records
5   # GET /camera_records.json
6   def index
7     @camera_records = CameraRecord.all
8   end
9
10  # GET /camera_records/1
11  # GET /camera_records/1.json
12  def show
13  end
14
15  # GET /camera_records/new
16  def new
17    @camera_record = CameraRecord.new
18  end
19
20  # GET /camera_records/1/edit
21  def edit
22  end
23
24  # POST /camera_records
25  # POST /camera_records.json
26  def create
27    @camera_record = CameraRecord.new(camera_record_params)
28
29    respond_to do |format|
30      if @camera_record.save
31        format.html { redirect_to @camera_record, :status => :created }
32        format.json { render :show, status: :created, location: @camera_record }
33      else
34        format.html { render :new }
35        format.json { render json: @camera_record.errors, status: :unprocessable_entity }
36      end
37    end
38  end
39 end
```

5-18 IDE ATOM. (Elaboración propia)

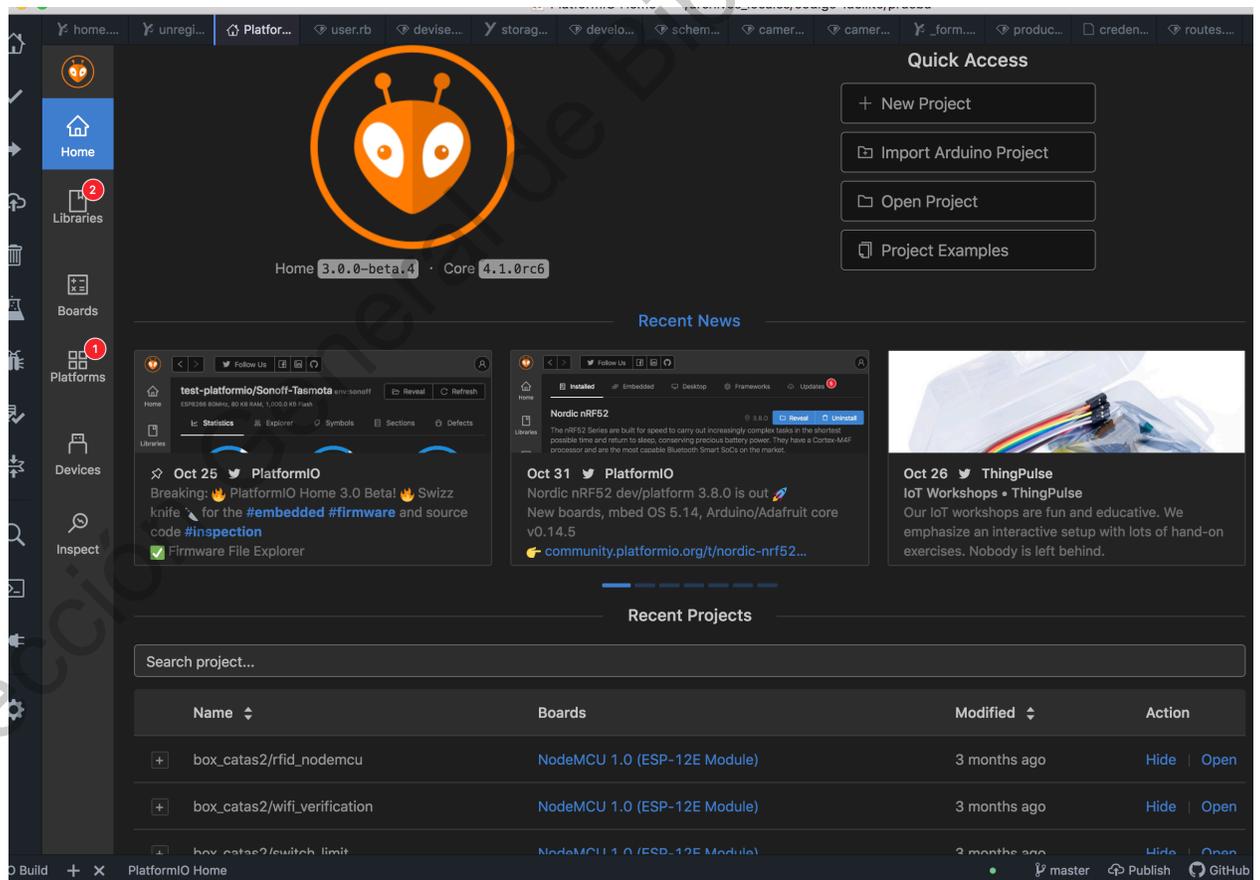
5.2.4.3 Platform IO

Es un ecosistema de desarrollo de código abierto, especialmente diseñado para proyectos de IoT, esta escrito en Python y trata de ser una plataforma de

comunicación para gestionar cualquier tipo de tarjeta de desarrollo, se encarga de manejar todas las librerías disponibles y ayuda en la gestión de proyectos de hardware entre diferentes programadores.

Esta aplicación de consola se puede usar en conjunto con algún editor que permita agregar la herramienta como lo puede ser Atom, Eclipse, Emacs, NetBeans, Sublime Text, VIM, Visual Studio, dentro de esta tesis se realizó la gestión con Atom.

Una vez instalados todos los paquetes necesarios entre estos dos elementos contamos por default con una enorme cantidad de tarjetas de microcontroladores, entre las cuales ya estaban el Arduino Nano y el NodeMCU, sin la necesidad de agregar mas tarjetas.



5-19 PlatformIO. (Elaboración propia)

En la figura 5-19 se aprecia la interfaz realizada al gestionar la aplicación de PlatformIO con Atom, la razón de utilizar este segundo IDE durante la etapa de producción de dispositivos se debe a su fácil conectividad y manejo de tarjetas y paqueterías utilizadas para proyectos de IoT, ya que contenía la mayoría de las librerías que fueron utilizadas por los sensores y actuadores de este proyecto.

La programación de los componentes en esta plataforma se realiza en C++ a diferencia del IDE de Arduino que se usaba principalmente Java, aunque los dos lenguajes son muy diferentes, dentro de sus sentencias básicas, para programar microcontroladores, los cambios no presentan una mayor complicación para un desarrollador con un poco más de mínima experiencia.

5.3 Herramientas WEB

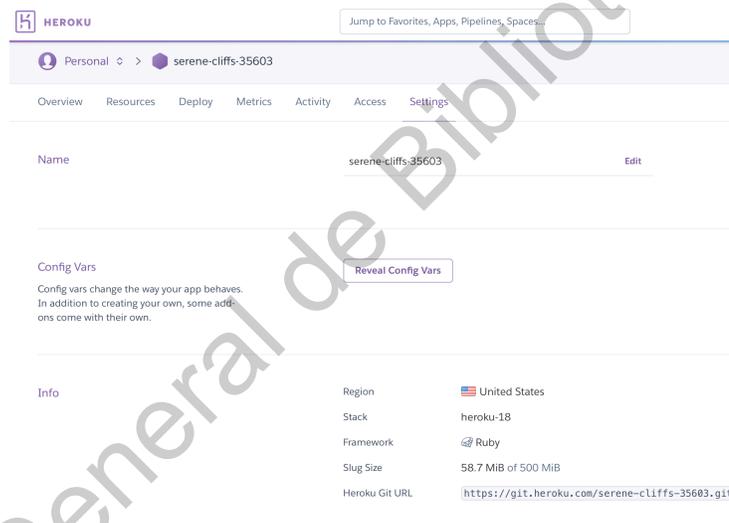
Cuando se realiza un proyecto de IoT, se requiere una serie de herramientas mayores para gestionar, probar y mantener la parte que se encuentra en la nube, a parte de programar la aplicación se necesita acceder a una serie de herramientas web que no son de código libre como los lenguajes y placas de desarrollo previamente analizados, pero cuentan con versiones gratuitas o de prueba para manejar el sistema.

5.3.1 Heroku

Es una herramienta utilizada en el desarrollo de aplicaciones, para mantener concordancia dentro de la infraestructura, administrador de servidor, base de datos y seguridad entre otras cosas. Se encuentra dentro de las llamadas PaaS (Plataforma de Servicio), pues dentro de ellas puedes realizar todas las actividades antes mencionadas y resolver el despliegue de la aplicación.

Se funda en el año 2007 por Lindenbaum, Wiggins y Herry con el objetivo de soportar aplicaciones desarrolladas en lenguaje Ruby, pero posteriormente se expandió a otros lenguajes de programación como Java, Node.js, Scala y Python.

Heroku permite manejar los servidores, configuraciones, escalamiento y administración, los principales usuarios de Heroku son programadores que desean enfocarse en el desarrollo y utilizar la plataforma para minimizar la gestión de la infraestructura, en una etapa temprana dentro del desarrollo de la aplicación, ayuda a reducir el número de operaciones.



5-20 Heroku. (Elaboración propia)

Existen dos configuraciones primordiales dentro del uso de Heroku, una versión gratuita que es la que se utiliza dentro de esta tesis, la cual consta de las herramientas básicas y un servidor que entra en modo de sueño en caso de que el tráfico de la página web sea nulo. Así como una versión de un costo aproximado de 7 dólares que permite que el servidor no entre en modo de sueño y la capacidad de la base de datos es mucho mayor.

Los Dynos son la base de su modelo de arquitectura, están basadas en contenedores Linux que son las que proveen la capacidad de computo, cada uno de ellos esta aislado del resto, por lo que a mayor cantidad de Dynos mayor es la velocidad de respuesta de la aplicación con el usuario.

En la figura 5-21 se aprecia la actividad realizada en la aplicación creada dentro de esta tesis, se muestra la información del desarrollador, el costo del mantenimiento del servidor y el uso de Dynos, la base de datos utilizada en el modo de producción y los cambios desarrollados por el programador a lo largo de la actividad. Algunos datos de la aplicación son reservados para mantener la seguridad de la información procesada.

Se entrega una URL para su acceso llamada serene-cliffs, la cual es generada de forma aleatoria al generar la aplicación, en caso de que sea requerido un nombre no aleatorio, entra dentro de los servicios que tienen un costo.

The screenshot displays the Heroku dashboard for the application 'serene-cliffs-35603'. The interface includes a navigation bar with tabs for Overview, Resources, Deploy, Metrics, Activity, Access, and Settings. The main content area is divided into several sections:

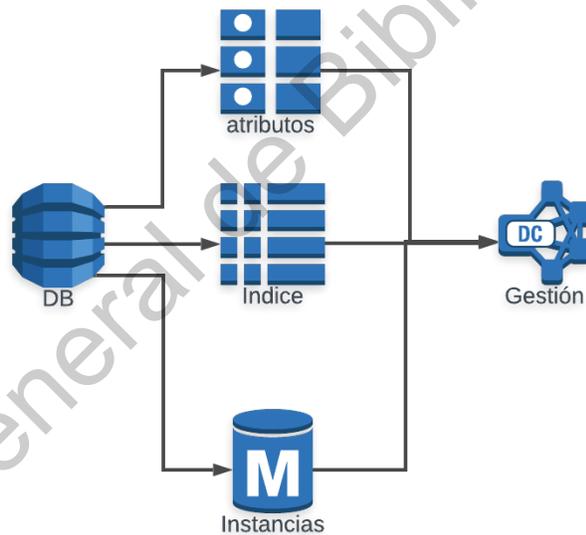
- Installed add-ons:** Shows 'Heroku Postgres' (Hobby Dev) with a cost of \$0.00/month. The instance ID is 'postgres-cylindrical-29207'.
- Dyno formation:** Shows 'This app is using free dynos' and the command 'web bin/rails server -p \$PORT -e \$RAILS_ENV' with a status of 'ON'.
- Collaborator activity:** Lists 'mujicalanderos.m@gmail.com' with '2 deploys'.
- Latest activity:** A list of recent events:
 - Deployment of version v7 (fb52d4a1) on Jun 1 at 11:17 PM.
 - Build succeeded on Jun 1 at 11:16 PM.
 - Deployment of version v6 (2fc6ed6b) on Jun 1 at 5:49 PM.
 - Provisioning completed for 'postgres-cylindrical-29207' on Jun 1 at 5:49 PM.
 - Attach DATABASE (@ref:postgres-cylindrical-29207) on Jun 1 at 5:49 PM.

5-21 Heroku Configuración. (Elaboración propia)

5.3.2 Base de Datos

Una base de datos es la manera en que podemos almacenar y organizar grandes cantidades de información para encontrar y extraer fácilmente, se define como un conjunto de información agrupada o estructurada.

Desde el punto de vista informático que atañe al proyecto se usa como una serie de tablas que contienen información que es enviada desde algún dispositivo hacía la aplicación en internet. Contando con una serie de tablas en las cuales se despliega y almacena la información según sus características y atributos.



5-22 Base de Datos. (Elaboración propia)

Toda base de datos debe contar con las siguientes características para mantener la integridad de la información y su manejo.

- Independencia lógica de datos.
- Redundancia mínima.
- Acceso concurrente de múltiples usuarios.
- Integridad de datos
- Consultas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación.

5.3.2.1 SQLite

Durante la etapa de pruebas de la aplicación de Heroku, se utilizó SQLite para gestionar la base de datos, es una herramienta de software libre que permite trabajar de forma eficaz en sistemas con pocos recursos computacionales, ya sea por su tamaño o por la capacidad de la infraestructura disponible. Rails utiliza esta base de datos por default dentro de sus gemas para el prototipo de la aplicación y se mantuvo la misma hasta el momento de mandar la aplicación a Heroku para su implementación. Algunas de sus características distintivas son las siguientes.

- La base de datos completa se encuentra en un solo archivo.
- Funciona en memoria.
- No utiliza dependencias externas.
- Librerías de acceso a diferentes lenguajes de programación.
- Diferentes formatos para longitudes de texto.
- Soporta funciones SQL definidas por el usuario.
- Su código fuente es público.

5.3.2.2 PostgreSQL

Sistema de gestión de base de datos relacional y orientada a objetos, es de código abierto manejado de forma libre por la comunidad PGDG (PostgreSQL Global Development Group), fue lanzada en 1996 y desarrollada por Michael Stonebreaker.

Utiliza el estándar de SQL que se encuentra en la mayoría de base de datos relacionales que al momento están en uso comercial, su característica principal es la Alta Congruencia, esto permite que mientras un proceso modifica una tabla otra instrucción puede acceder a la misma sin bloqueos.

En la figura 5-23 se ve un ejemplo de una consulta a la base de datos de la aplicación cuando ya encuentra en producción en Heroku, la consulta se realiza en lenguaje de Ruby e inmediatamente Rails realiza el cambio a una consulta de la base de datos en SQL.

```
>> User.first
User Load (0.4ms) SELECT "users".* FROM "users" ORDER BY "users"."id" ASC LIMIT ? [{"LIMIT", 1}]
=> #<User id: 1, email: "mujicalanderos.m@gmail.com", username: "", name: "Mich Mark", last_name: nil, bio: nil, uid: "10214501490790368", provider: "facebook", created_at: "2019-05-23 01:38:15", updated_at: "2019-05-27 20:33:12">
>> User.second
User Load (0.3ms) SELECT "users".* FROM "users" ORDER BY "users"."id" ASC LIMIT ? OFFSET ? [{"LIMIT", 1}, {"OFFSET", 1}]
=> #<User id: 2, email: "ulhtcvskla_1558586919@tbnw.net", username: "", name: "Susan Alcfiachccbb Huiberg", last_name: nil, bio: nil, uid: "110371880203274", provider: "facebook", created_at: "2019-05-23 05:20:44", updated_at: "2019-05-29 20:00:54">
>> User.third
User Load (0.2ms) SELECT "users".* FROM "users" ORDER BY "users"."id" ASC LIMIT ? OFFSET ? [{"LIMIT", 1}, {"OFFSET", 2}]
=> #<User id: 3, email: "michel@test.com", username: "", name: nil, last_name: nil, bio: nil, uid: nil, provider: nil, created_at: "2019-05-24 01:49:07", updated_at: "2019-05-27 21:43:28">
>> User.fourth
```

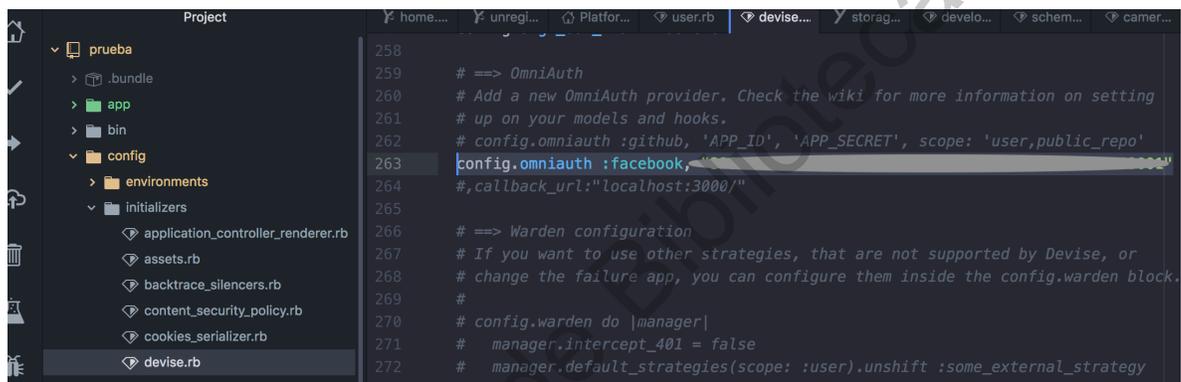
5-23 Consulta de Base de Datos. (Elaboración propia)

5.3.3 Inicio de Sesión con Facebook

Muchas de las aplicaciones web y páginas de internet con las que convivimos actualmente, permiten que el usuario acceda a su cuenta o pueda crear una a partir de una dirección de correo electrónico o con alguna red social, Rails utiliza una gema llamada 'omniauth-facebook' que permite la creación de un usuario nuevo

mediante la cuenta de Facebook a través de la relación con la aplicación de Facebook Platform.

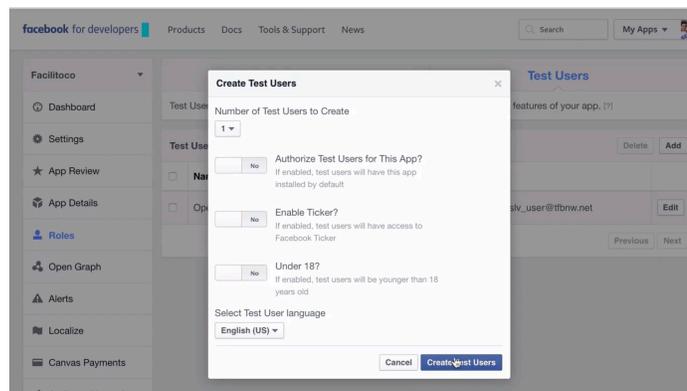
En la imagen 5-24 se muestra la forma en que se hace la llamada de la función de Facebook con la gema omniauth, por razones de seguridad se mantienen ocultas las demás configuraciones y su ingreso a la base de datos, ya que la prueba se realiza con datos reales.



```
258
259 # ==> OmniAuth
260 # Add a new OmniAuth provider. Check the wiki for more information on setting
261 # up on your models and hooks.
262 # config.omniauth :github, 'APP_ID', 'APP_SECRET', scope: 'user,public_repo'
263 config.omniauth :facebook, 'APP_ID', 'APP_SECRET', scope: 'user,public_repo'
264 #, callback_url: "localhost:3000/"
265
266 # ==> Warden configuration
267 # If you want to use other strategies, that are not supported by Devise, or
268 # change the failure app, you can configure them inside the config.warden block.
269 #
270 # config.warden do |manager|
271 #   manager.intercept_401 = false
272 #   manager.default_strategies(scope: :user).unshift :some_external_strategy
```

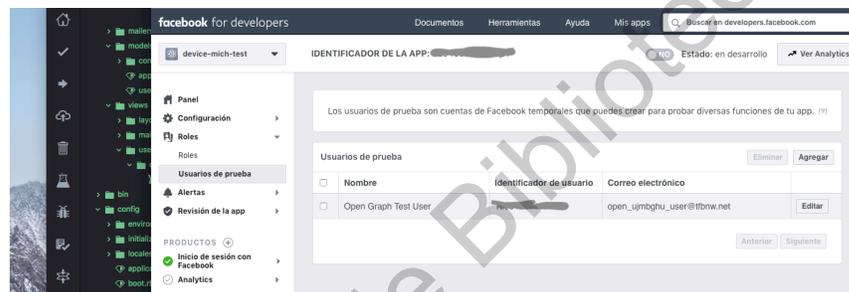
5-24 Rails con Facebook Developers. (Elaboración propia)

Entre las funciones que ofrece Facebook después de vincularse a una cuenta real, ofrece la posibilidad de generar usuarios dentro de la aplicación de la red social, para probar cuentas entrantes al servidor web sin necesidad de generar cuentas de correo electrónico, bots o algún tipo de cuentas falsas.



5-25 Creación de cuentas de prueba. (Elaboración propia)

Los usuarios de testeo y el manejo de cuentas solamente se utilizaron en la fase local de pruebas y desarrollo de la aplicación, ya que en la etapa de producción de proyecto se elimino esta biblioteca de la aplicación con el fin de garantizar información fidedigna, únicamente dispositivos físicos reales instalados dentro de la institución pueden conectar a la aplicación que actualmente gestiona al edificio CATAS.

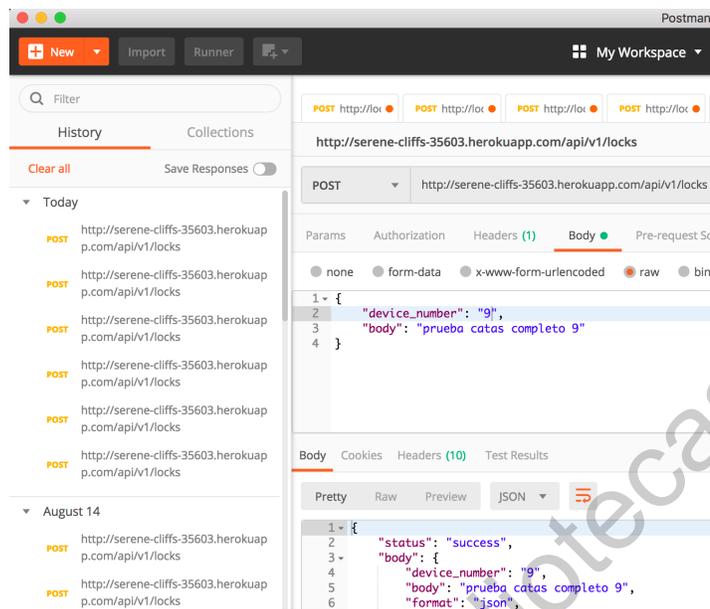


5-26 Cuenta Facebook. (Elaboración propia)

5.3.4 Postman

Es una herramienta diseñada para probar servicios Web, fue fundada en el 2014 con el propósito de poder probar, desarrollar y documentar Apis de manera sencilla, en la siguiente imagen vemos una prueba de conexión exitosa simulando un dispositivo enviando información a la aplicación.

En la sección izquierda se muestran los datos de las pruebas realizadas, el envió de una cadena de texto a la dirección URL de la aplicación en Heroku y la respuesta exitosa por parte de esta.



5-27 Postman. (Elaboración propia)

5.4 Protocolos de Comunicación

Cada una de las herramientas que se han explicado mantienen un propósito particular dentro de los dispositivos IoT como componentes aislados; es necesario establecer una comunicación entre los diferentes elementos de hardware y software para transmitir las instrucciones o recopilar información.

Este es uno de los puntos donde la optimización de sistemas juega un papel importante, ya que elegir el correcto protocolo de información permitirá transmitir los datos de un punto a otro de los elementos, con el mejor consumo energético posible, velocidad y la mejor resolución de las cadenas de datos.

5.4.1 Protocolo SPI

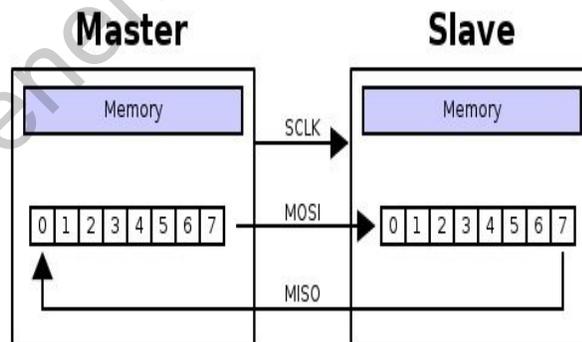
SPI es un acrónimo para Interfaz Serial Periférica, nace a principios de 1980 cuando Motorola lo empieza a introducir, es muy utilizado debido a su relativa

simplicidad, alta velocidad de transmisión y la gran cantidad de microcontroladores, sensores y pantallas que le utilizan.

Funciona de manera síncrono, es decir tiene un reloj maestro en alguno de los dispositivos que se conecta a este, para dirigir el envío y la secuencia de datos; al utilizar varias líneas de comunicación se permite ser full dúplex, lo que permite la comunicación bidireccional.

Se define un maestro (Master) y esclavo (Slave) o esclavos, entre los cuales se realiza la comunicación, uno de los dispositivos diseñados en esta tesis utiliza este protocolo de comunicación con varios esclavos; en la práctica para su correcto funcionamiento, los esclavos deben estar aislados entre ellos para evitar interferencias, ya que utilizan la misma línea de comunicación de forma conmutada.

En la figura 5-28 se muestra la relación Maestro-Esclavo, en la cual el microcontrolador es el maestro y los sensores de los dispositivos actúan como esclavos.



5-28 Protocolo SPI. (Elaboración propia)

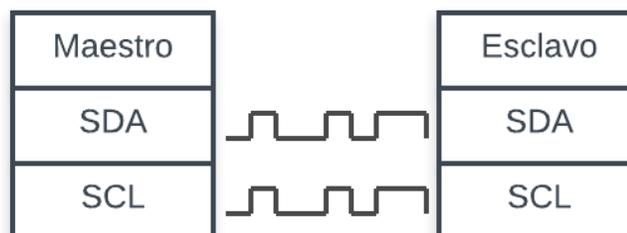
El utilizar este protocolo permite conectar diferentes dispositivos a un solo microcontrolador con una menor cantidad de líneas de datos, con esto se logra utilizar placas con menos pines a los previamente instalados, reducir costos y tamaños. Las cuatro líneas utilizadas para conectar el maestro y los esclavos cumplen las siguientes funciones.

- MOSI (Maestro Salida Esclavo Entrada): Línea que lleva los bits del maestro al esclavo.
- MISO (Maestro Entrada Esclavo Salida): Línea que lleva los bits del esclavo al maestro.
- CLK (Reloj): Línea que genera el maestro, encargada de la señal de reloj que sincroniza el envío y manejo de datos.
- SS (Selección de Esclavo): Habilita o deshabilita los esclavos conectados a un Maestro, de esta manera se evita recibir información del sensor equivocado en caso de que un agente externo lo habilite.

5.4.2 Protocolo I2C

El I2C significa Circuito Inter-Integrado, fue desarrollado por Phillips en la década de 1980, toma algunos de los elementos de la comunicación SPI y UART; sin embargo, este solamente utiliza dos líneas de datos a diferencia del SPI, de igual manera es serial y síncrono.

- SDA: Datos Seriales, es la vía de comunicación entre maestro y esclavos.
- SCL: Reloj Serial, línea de la señal de reloj para mantener la sincronización.

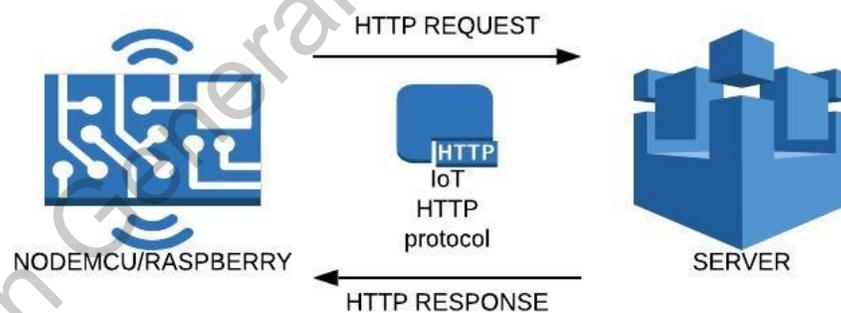


5-29 Protocolo I2C. (Elaboración propia)

Teniendo similitudes con el otro protocolo, cada uno tiene ventajas y desventajas, el protocolo SPI es más rápido y el I2C tiene un menor consumo de recursos en hardware y energético, esta ventaja es utilizada para conectar por este último pantallas que utilizan menos energía y permite tener más pines disponibles.

5.4.3 Protocolo HTTP

Es el acrónimo para el Protocolo de Transferencia de Hipertexto, permite la transmisión de información a través de Internet, su versión 1.1 fue lanzada en 1999, definiendo la sintaxis y semántica que los clientes y servidores utilizan; los mensajes enviados por el cliente que usualmente es un Navegador Web se llaman Peticiones y los mensajes enviados por el servidor se llaman respuestas.



5-30 Protocolo HTTP. (Elaboración propia)

Como su nombre lo dice está diseñado para transmitir texto, pero también se pueden enviar imágenes y vídeos mediante formularios de datos, el HTTP es un protocolo de la capa de aplicación y se debe transmitir mediante algún formato que

permita la seguridad o encriptar los datos contenidos en las peticiones o formularios. Algunas de sus características son las siguientes.

- Es sencillo, está pensado para ser leído y desarrollado fácilmente, reduciendo la depuración de errores.
- Extensible, con la implementación de cabeceras HTTP, el protocolo se puede ampliar y desarrollar funciones propias por el programador, únicamente se requiere que tanto cliente como servidor lean y escriban bajo el mismo paradigma.
- Con sesiones sin estados: al no tener estados no guarda ningún dato entre peticiones, el uso de cookies permite mantener la información de la sesión en que se encuentra el navegador, permitiendo guardar los datos de la comunicación.

Existen métodos de petición predefinidos a los que se llegan a referir como verbos; cada uno de estos indica la acción que se busca efectuar.

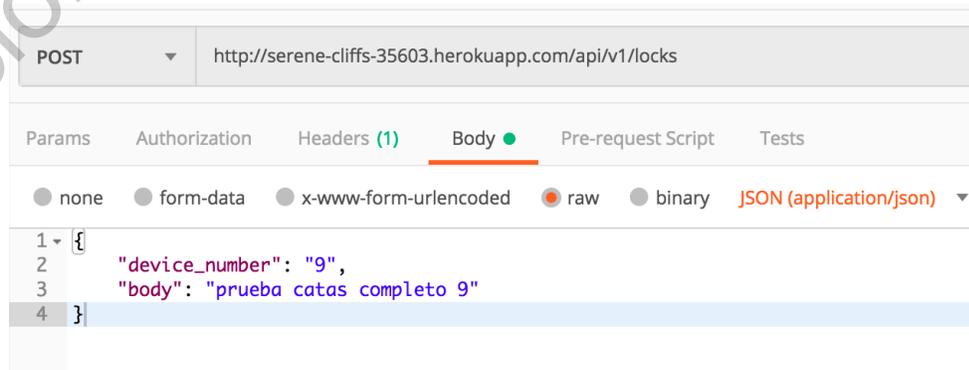
- GET: Solicita un recurso específico, su única función es recuperar datos mediante la petición.
- POST: Envía los datos que fueron pedidos o designados como recurso específico.
- HEAD: Realiza una solicitud similar a la de GET, pero únicamente recibe la cabecera, no el cuerpo, resulta efectivo cuando únicamente se buscan los encabezados y no trasladar toda la información.

5.4.4 JSON

Acrónimo para Notación de Objeto de JavaScript, se trataba de un subconjunto de notación para JavaScript, pero como a sido ampliamente adoptado ya se considera un lenguaje de marcado independiente. Una de las ventajas que ofrece es la facilidad de leerlo con un analizador sintáctico, de esta manera se puede acceder a distintos tramos de la información con menor cantidad de esfuerzo de programación.

Los tipos de datos que se pueden mandar en un formato JSON son variados mas allá de cadenas de caracteres, lo que constituye una mejor comunicación del sistema que envía con la aplicación que recibe la información, entre los formatos aceptados se encuentran.

- Números enteros, decimales, positivos o negativos.
- Cadenas de caracteres.
- Null, representa un valor nulo.
- Array son listas ordenadas que pueden contener varias cadenas o números.
- Objetos son pares de datos que envían grupos de información o instrucciones. Es el formato que se utiliza para la transmisión de datos desde los dispositivos en CATAS a la aplicación web.



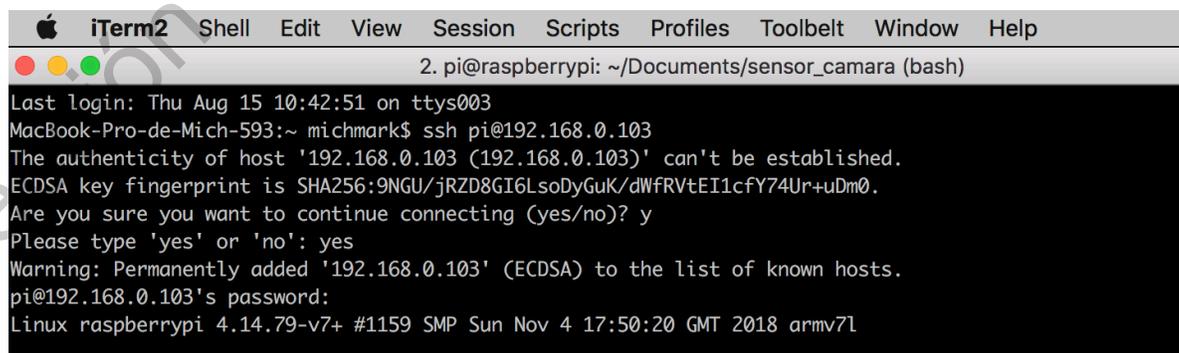
5-31 Formato JSON. (Elaboración propia)

Dentro de la aplicación de Rails, se genera una API (Interfaz de Comunicación de Aplicaciones) la cual recibe la información de las placas de microcontroladores mediante una petición HTTP, el texto dentro de la respuesta es enviado en formato JSON y se recibe únicamente si el analizador de textos concuerda que es la semántica correcta preprogramada, de esta manera se asegura un mayor filtro de seguridad en la información enviada por Internet.

5.4.5 SSH

Es un protocolo de administración remota que permite al programador controlar servidores remotos a través de internet, con un formato de autenticación. Los equipos basados en UNIX pueden usar directamente la conexión SSH en sus terminales, los equipos con Windows requieren utilizar algún cliente externo que les permita la comunicación.

Este protocolo permite la comunicación remota entre las diferentes Raspberry utilizadas en el proyecto y la computadora principal utilizada para desarrollar todos los códigos. En la imagen 5-32 se observa la forma de ingresar desde un equipo UNIX a una de las Raspberry.



```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
2. pi@raspberrypi: ~/Documents/sensor_camara (bash)
Last login: Thu Aug 15 10:42:51 on ttys003
MacBook-Pro-de-Mich-593:~ michmark$ ssh pi@192.168.0.103
The authenticity of host '192.168.0.103 (192.168.0.103)' can't be established.
ECDSA key fingerprint is SHA256:9NGU/jRZD8GI6LsoDyGuK/dWfRVtEI1cfY74Ur+uDm0.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.0.103' (ECDSA) to the list of known hosts.
pi@192.168.0.103's password:
Linux raspberrypi 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l
```

5-32 SSH. (Elaboración propia)

Ambos equipos deben estar conectados a la misma red Wifi para evitar el acceso de usuarios no deseados y se debe conocer la IP del equipo al que se va a ingresar, la Raspberry también debe ser configurada previamente de forma manual habilitando los puertos.

El acceso de este protocolo no permite el acceso a la interfaz gráfica, pero si permite el acceso a todas las carpetas de documentos, y a realizar modificaciones dentro de ellos, por lo tanto, se pueden compilar y ejecutar programas que se encuentren en la Raspberry para activar módulos conectados a ella.

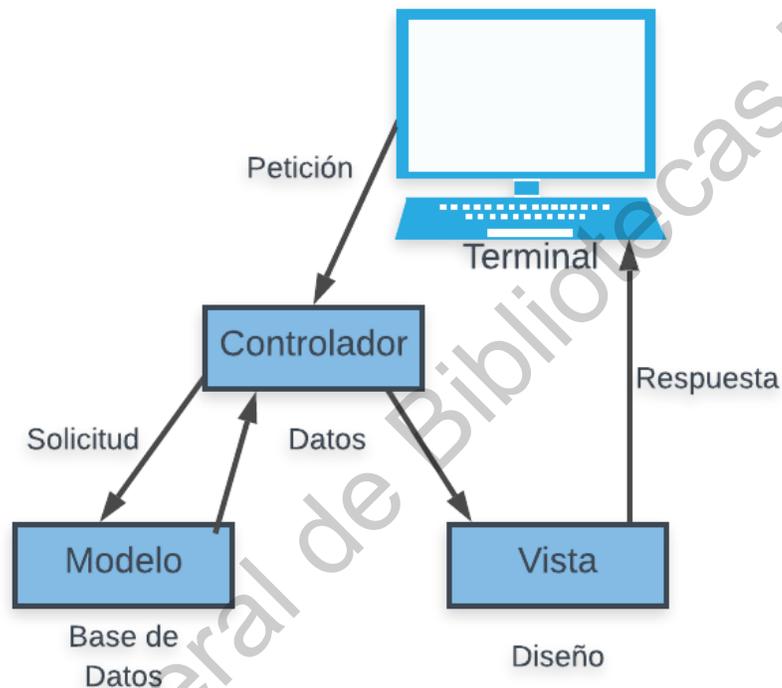
5.4.6 Modelo Vista Controlador

El MVC es un patrón de arquitectura de Software que separa los datos, la lógica, la representación visual dentro de una aplicación y los módulos de comunicaciones. Se basa en la idea de reutilización de código y separación de conceptos; filosofía fundamental en Ruby & Rails.

Se definen los componentes de este patrón de la siguiente manera.

- Modelo: Es la capa donde trabajan los datos, se encuentran la base de datos en formato de tablas de la aplicación, aquí se realizan las consultas para acceder a la información. Usualmente las consultas no se realizan directamente en SQL si no en el lenguaje del Framework en nuestro caso Ruby.
- Vista: Contienen el código de la interfaz a la que accede el usuario durante la visualización y la forma en que se expresan las salidas, de manera general aquí se encuentra el lenguaje HTML y todas las herramientas de personalización gráfica.

- Controlador: Su responsabilidad es manipular los datos del modelo y actualizar la información de las vistas, sirve de vínculo entre ambas y responde a las interacciones que llegan desde la API para nuestra aplicación.



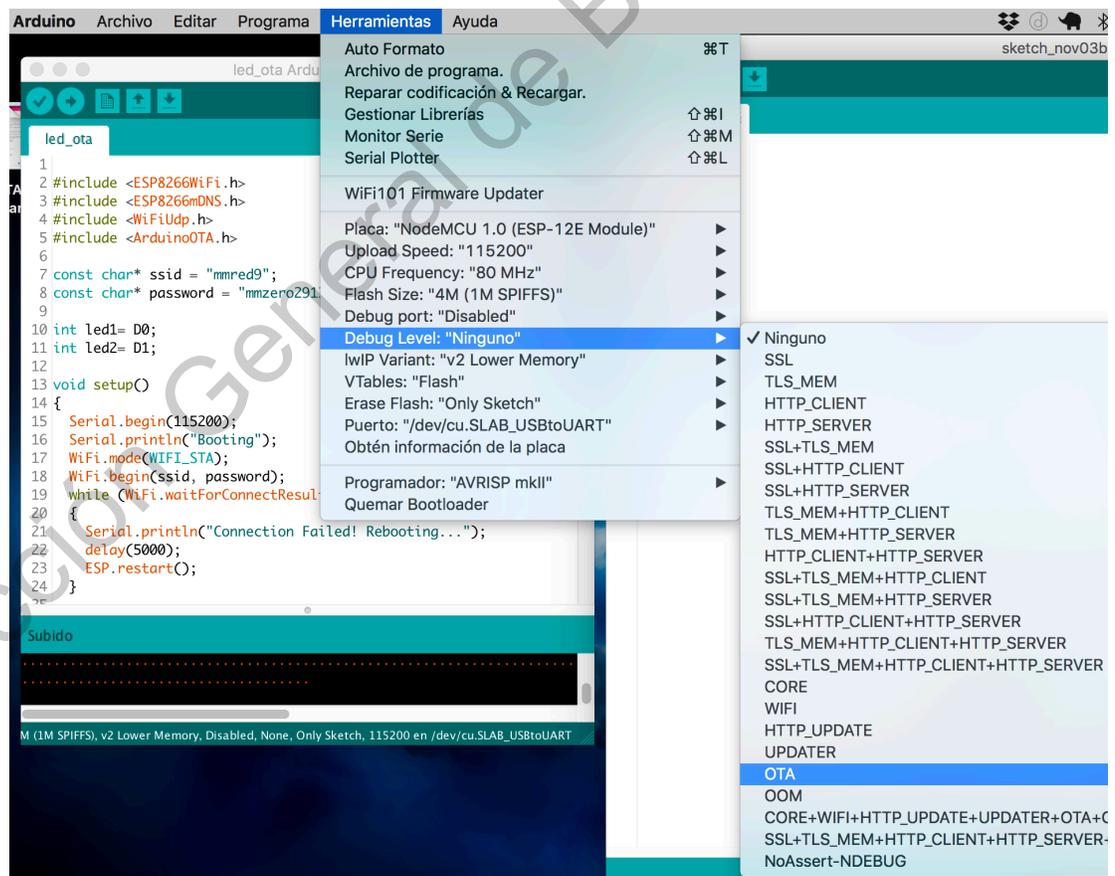
5-33 Modelo Vista Controlador. (Elaboración propia)

5.4.7 Programación OTA

OTA es la descripción de un método de programación de modo inalámbrico viene de las siglas Over The Air (En el aire), en el caso de esta tesis, los microcontroladores con conexión a internet se pueden reprogramar vía Wifi, mediante la conexión a internet que utilizan para transmitir sus datos.

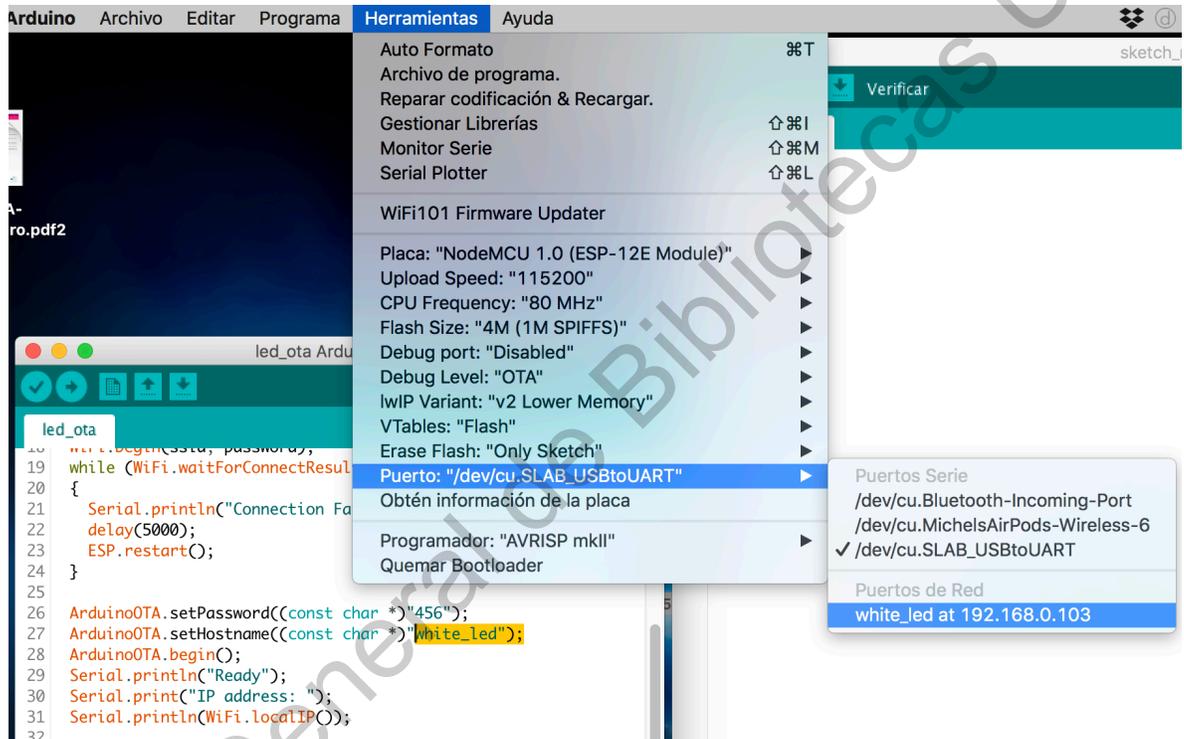
Los microcontroladores son programados y reprogramados mediante un cable micro USB, conectados a un equipo de cómputo, con el compilador adecuado, durante la fase de pruebas este aspecto no presenta mayor inconveniente; cuando los dispositivos ya se encuentran instalados y en funcionamiento, retirarlos para hacer modificaciones resulta bastante complicado, ya que por seguridad se busca que su acceso no sea sencillo.

Los NodeMCU utilizan su módulo de Wifi para enviar datos a la aplicación y también se puede utilizar para reprogramar, esta función se puede realizar con ambos IDE, se muestra en la figura la selección de la Herramienta OTA en el IDE de Arduino, de esta manera se habilita la búsqueda de dispositivos con esta conexión habilitada.



5-34 Conexión OTA en IDE. (Elaboración propia)

La librería utilizada “ArduinoOTA.h” es funcional con cualquier microcontrolador que utilice un modulo Wifi de la familia ESP, no es exclusivo de la familia de placas Arduino, al programar inicialmente una placa con esta librería, permite su posterior programación de manera inalámbrica, mientras se cumplan los protocolos de seguridad previamente establecidos.



5-35 NodeMCU conectado por OTA. (Elaboración propia)

Una vez identificado el dispositivo con el nombre que en su primera programación se asignó como en esta prueba “White_led” y conociendo la IP se puede enviar un nuevo programa para reemplazar al anterior, agregar o eliminar elementos.

A parte de necesitar estar conectados a la misma red Wifi, el NodeMCU solo puede ser reprogramado conociendo la contraseña previamente asignada en la

programación inicial. De esta manera se agrega un filtro mas de seguridad a la modificación de las cerraduras y protección de los usuarios que las utilizan.



5-36 Seguridad OTA. (Elaboración propia)

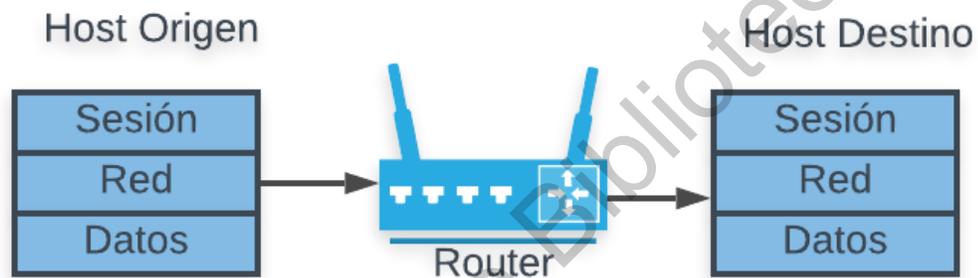
5.5 Hardware

En esta sección se describen:

- Los elementos electrónicos como sensores y actuadores que realizan las tareas o envían la información recabada a las placas de microcontroladores o algún equipo computacional.
- Las diferentes fuentes de alimentación utilizadas de la manera más optima posible en cada uno de los dispositivos buscando cubrir las necesidades mínimas de potencia y a su vez para mejorar el consumo energético, utilizando el generador eléctrico correcto.
- Los equipos de red locales y el maquinado de piezas creadas a la medida de las tarjetas y su aplicación.

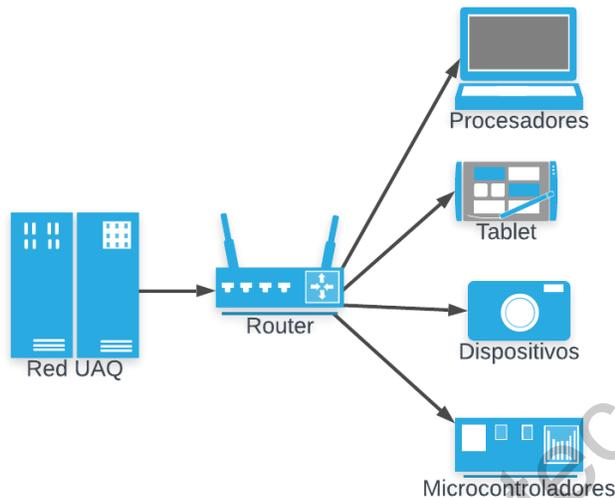
5.5.1 Router

También conocido como enrutador es un dispositivo que permite la conexión entre varios equipos de computo mediante una red de internet, ya sea de manera física con una conexión Ethernet o vía Wifi. Puede crear una subred dentro de una red local o de Internet para reducir la cantidad de dispositivos que tienen acceso a ella con algún propósito específico.



5-37 Conexión Router. (Elaboración propia)

La UAQ (Universidad Autónoma de Querétaro) cuenta con una infraestructura de Internet en cada uno de sus campus, no se considera óptimo utilizar la misma red a la que tienen acceso maestros, alumnos, personal administrativo y otros usuarios, por lo que se planteó la opción de utilizar una subred creada por un Router, utilizando este Router los dispositivos conectados a esta red quedan aislados de los demás elementos de la Red UAQ manteniendo la seguridad y con un bajo tráfico de datos.



5-38 Conexión de Internet Interna. (Elaboración propia)

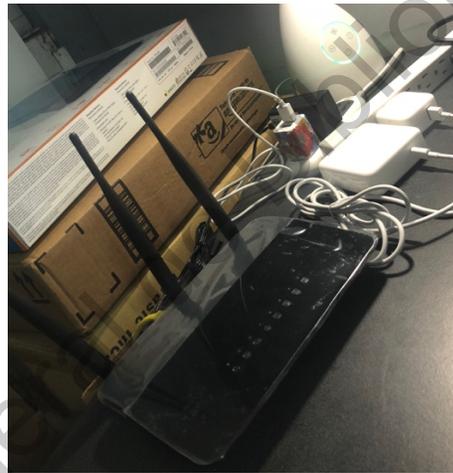
En el campus de Tequisquiapan donde se encuentra el edificio CATAS de la UAQ, se instaló un Router DIR-809 de la compañía D-Link con las siguientes características.

- Doble banda 2.4 y 5 GHz.
- Soporte para IPv6.
- Modo repetidor.
- 5 puertos de Ethernet.
- 3 antenas con alcance de 200 metros.
- Botón de configuración.

Durante la fase de pruebas se instaló otro dispositivo de las mismas características en el campus San Juan del Río en el edificio de posgrado, para mantener una congruencia en la conectividad de los dispositivos durante su etapa de prototipo y su etapa de producción y uso. A la fecha de redacción de este documento el Router de Tequisquiapan se encuentra con el Servidor de red en el almacén dedicado al mismo.



5-39 Router Tequisquiapan. (Elaboración propia)

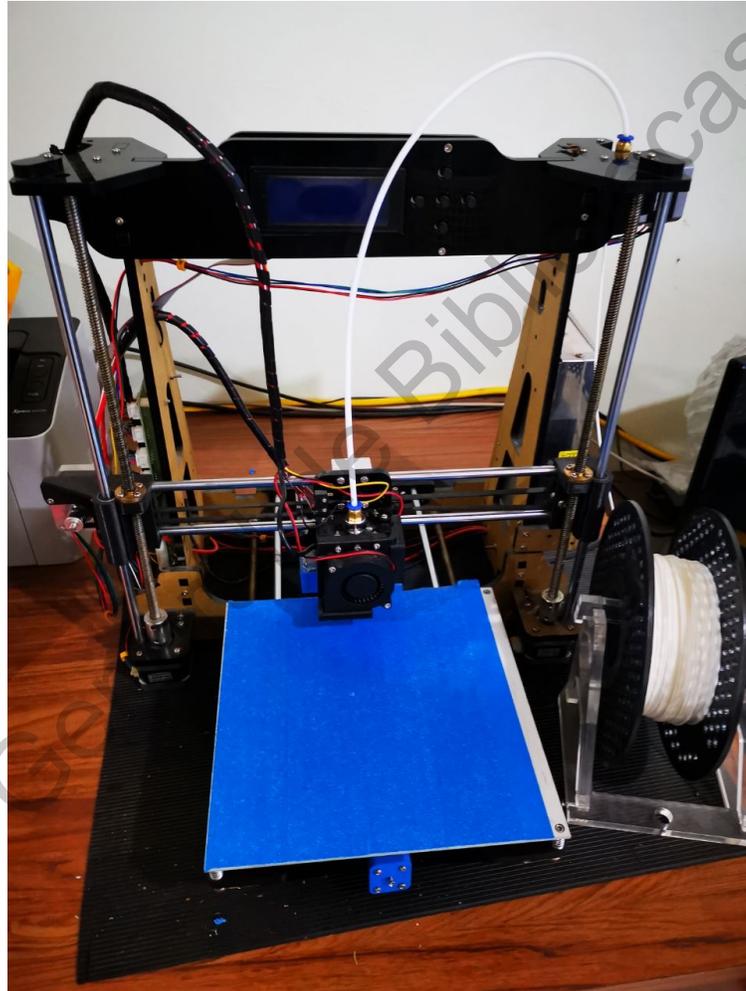


5-40 Router San Juan del Río. (Elaboración propia)

5.5.2 Impresión 3D

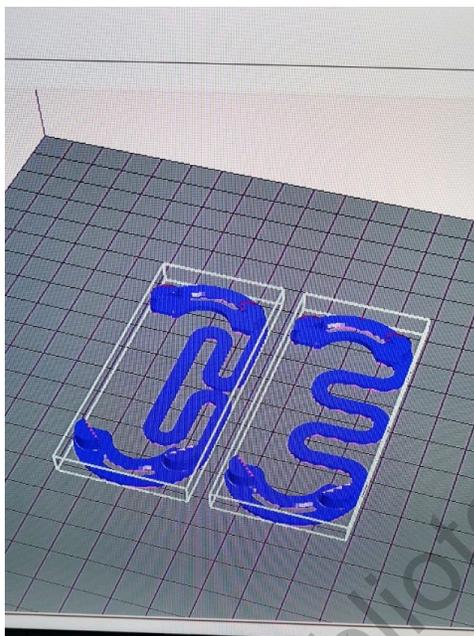
La impresión 3D es un grupo de tecnologías de fabricación para crear objetos tridimensionales a partir de un filamento plástico distribuido a manera de capas. Se utilizan para la fabricación de piezas de bajo costo y personalizadas por el desarrollador, desde el año 2003 las impresoras aumentaron su venta y reducción de costos, gracias a la cultura Maker alrededor del mundo que causó que cada vez más personas quisieran realizar sus propios proyectos y construir sus propias piezas en casa.

Se utilizó la impresora P802EA de la compañía Tronxy que se muestra en la imagen, para diseñar las carcasas de las Raspberry en su modelo final, para proteger el equipo y reducir el espacio y darle un acabado personalizado con los componentes que se iban a utilizar.

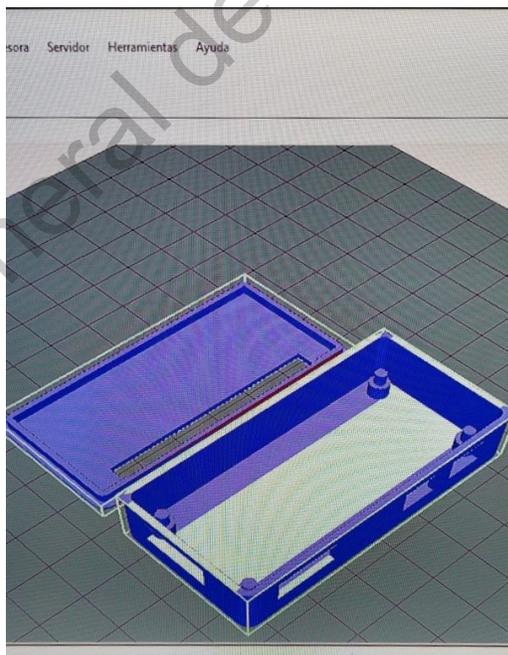


5-41 Impresora 3D. (Elaboración propia)

Se buscaron múltiples diseños que pudieran cumplir con los propósitos necesarios, puesto que todos los puertos y periféricos de la Raspberry serían utilizados en alguna etapa desde su configuración o su fase de trabajo.



5-42 Diseño propuesto 1. (Elaboración propia)



5-43 Diseño propuesto 2. (Elaboración propia)

El resultado impreso fue una carcasa con elementos de ambos modelos, para dejar libres los periféricos y del menor tamaño posible, para crear un dispositivo portátil y sutil.



5-44 Carcasa Raspberry Zero. (Elaboración propia)

5.5.3 Cerradura Magnética

Las cerraduras son una invariante pieza de cualquier sistema de seguridad, son fáciles de colocar y son la forma más básica de limitar el acceso a diferentes áreas de un edificio. En el caso de las cerraduras electromagnéticas son dispositivos que cuentan con dos principales piezas, por un lado, el electroimán, y por el otro lado una lámina metálica llamada pieza móvil o pieza polar. El electroimán se coloca en el marco de la puerta, trabaja como imán en la medida que circule corriente por su bobina y cierra la puerta; al dejar de recibir corriente eléctrica permite la apertura de la puerta.



5-45 Cerradura Magnética. (Elaboración propia)

La Cerradura Magnética utilizada en esta tesis es de la compañía Asekuro y cuenta con las siguientes características técnicas.

- Voltaje de Entrada 12 V/24V.
- Corriente de entrada con 12 V es 0.5 A.
- Switch de múltiples estados NA y NO (Normalmente abierto y normalmente cerrado).
- Jumper Selector de Voltaje (predefinido en 12 V).
- Bracket para su instalación en L, ZL, I y U.

5.5.4 Display Táctil

La Raspberry Pi modelo B tiene un tamaño aproximado de 9cm. x 5cm, lo que permite que sea un dispositivo portátil, aunque para acceder a su interfaz gráfica requiere un monitor, lo que causa que la convierte en un instrumento de escritorio, para mejorar aún mas su versatilidad se utiliza con ella una pantalla táctil Lcd de 7 pulgadas que permite utilizar su interfaz gráfica y crear un dispositivo portátil.



5-46 Pantalla Lcd Táctil. (Elaboración propia)

La pantalla cuenta con las siguientes características técnicas que permiten su compatibilidad con el sistema operativo de la Raspberry con ajustes mínimos dentro del código de configuración.

- Marca Waveshare.
- Voltaje de entrada 5 a 7 v.
- Entrada micro USB de alimentación.
- Salida HDMI.
- IPS de 7 pulgadas.
- Resolución 1024 x 600.
- Control capacitivo táctil.
- Compatible con Raspberry Pi.

- Compatible con Raspbian, W10 y Ubuntu
- Control de luz de Fondo.

5.5.5 RFID

RFID es el acrónimo para Identificación por Radio Frecuencia, el propósito de este sistema es acceder a la información de un dispositivo mediante la proximidad de este. Su propósito dentro de los sistemas de seguridad se basa en la identificación de objetos o personas mediante etiquetas de radio frecuencia.

Las etiquetas RFID constan de un pequeño dispositivo con una antena en forma de bobina en el interior, la cual permite enviar o recibir información al estar en contacto con el campo magnético de otra antena con las mismas características, pero alimentado con corriente eléctrica.

El RFID puede operar en 4 bandas de frecuencia.

- Baja frecuencia 125-134 Hz
- Alta frecuencia 13.56 MHz
- Ultra Alta frecuencia 868 GHz
- Microondas 2,45 GHz

Las tarjetas y lectores mas comúnmente utilizados de forma comercial son lo de 13.56 MHz y son los que se utilizan en esta tesis, ya que son mas seguros que las tarjetas de baja frecuencia y su relación costo beneficio es mayor que las tarjetas de mas altas frecuencias.

Las antenas RFID se colocan en pequeños TAGs que pueden venir en diferentes modelos, como los que se aprecian en la figura, las más comunes son las tarjetas y los llaveros, también se aprecian las calcomanías y pulseras, todas ellas cuentan con las mismas características y se puede utilizar el modelo que se considere mas practico para cada ocasión.

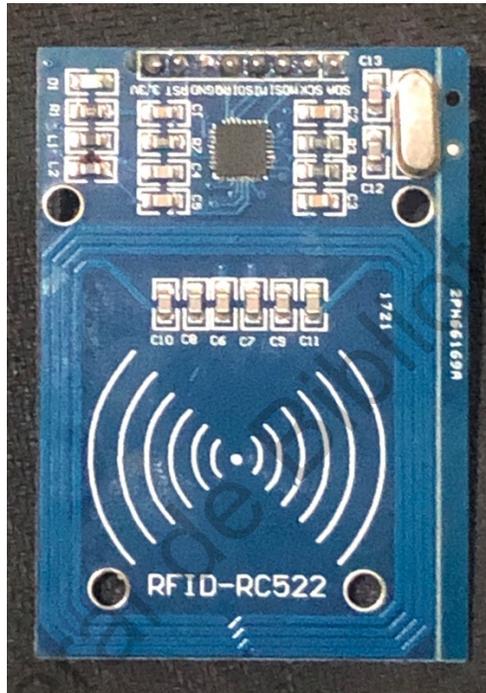


5-47 TAGs RFID. (Elaboración propia)

Los TAGs contienen por default un ID que consta de una cadena de datos de 4 valores hexadecimales concatenados cada uno con un valor entre 0 y 255, por lo tanto, no existen dos tarjetas en el mundo con el mismo identificador, por lo que pueden funcionar como llaves de seguridad mediante su identificador.

Las versiones comerciales de los TAGs cuentan con una pequeña memoria de 1Kb, dividida en 16 grupos de 64 bytes para almacenar alguna información extra si se desea.

Existen sistemas comerciales que pueden realizar la lectura de los TAGs, para tener un mejor control del sistema se utiliza el lector MFRC522, el cual utiliza la tecnología MIFARE, el cual es uno de los estándares mas utilizados en el campo de las tarjetas RFID.



5-48 Lector RFID MFRC522. (Elaboración propia)

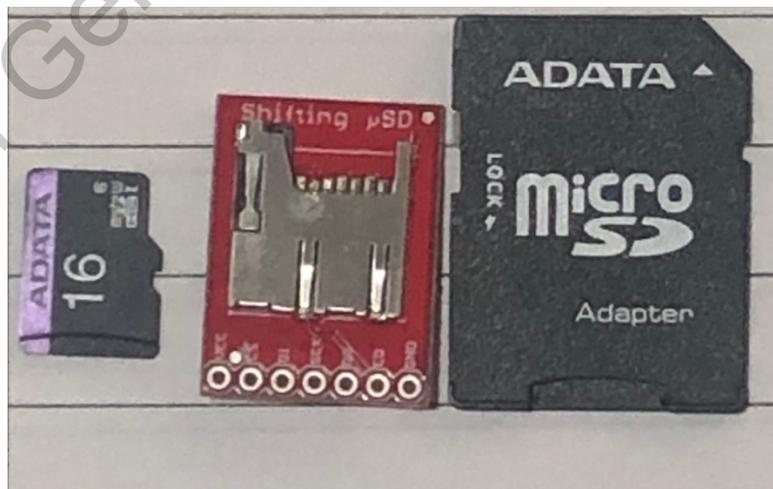
- Cuenta con conexión SPI, I2C y UART.
- Es compatible con los microcontroladores y sus tarjetas de desarrollo presentadas en esta tesis.
- Opera con la frecuencia de 13.56MHz
- Alimentación de 3.3 V
- Consumo de 10 a 26 mA.

Su voltaje de operación es muy pequeño por lo cual su consumo energético puede ser suministrado por cualquiera de las tarjetas, se utiliza el protocolo SPI para su comunicación por la velocidad a la cual son manejados los datos y cuenta con mayor cantidad de documentación.

Se utiliza la librería MFRC522 compatible con Arduino y el NodeMCU, por motivos de seguridad se utiliza únicamente el ID de las tarjetas para su manejo y no se escribe información personal de los usuarios dentro de ellas, de esta manera se utiliza esta tecnología como llave de acceso dentro del edificio CATAS y a su vez sirve como control de registro para la aplicación web.

5.5.6 Tarjetas SD para Microcontroladores

SD es un dispositivo en formato de tarjeta de memoria, se utiliza ampliamente en dispositivos portátiles como celulares, cámaras y reproductores de música, su pequeño tamaño y gran capacidad es un agregado bastante llamativo, para almacenar volúmenes de información cuando los dispositivos no estén conectados a internet.



5-49 Tarjeta microSD y lectora SD para Microcontrolador. (Elaboración propia)

La lectora de SD se conecta al microcontrolador utilizando de nueva cuenta el protocolo SPI, recordando que es el mismo protocolo que utilizan las lectoras de RFID, con los ajustes de instrumentación adecuados se pueden utilizar ambos sistemas en el mismo microcontrolador con los mismos pines de conexión.

Algunos de los dispositivos no están conectados a internet, esto debido a la configuración que se diseñó para los mismos, como elementos aislados, de tal manera que toda la información recabada por los mismos es almacenada en tarjetas SD, para su posterior análisis y almacenamiento.

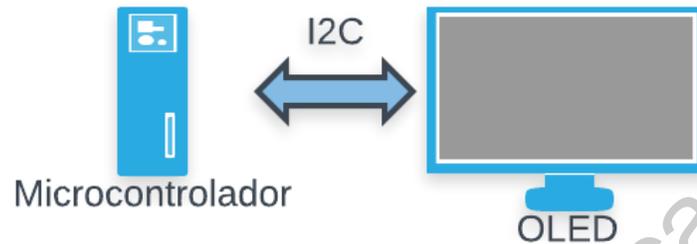
5.5.7 Pantalla OLED

OLED viene de Diodo Emisor de Luz Orgánico, las pantallas constan de leds con una capa emisiva de compuesto orgánico que al tener paso de corriente eléctrica emite luz; por lo tanto, su consumo energético es mucho menor a la mayoría de las pantallas utilizadas en los microcontroladores.

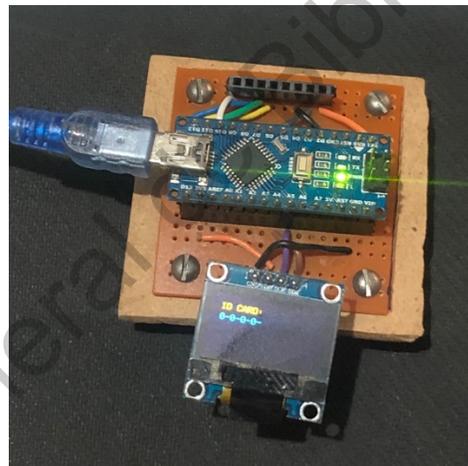
Para estos microcontroladores están diseñadas pantallas OLED especiales de 0.96 pulgadas

- Controlador SDD1306
- Comunicación SPI o I2C
- Tamaño de 25mm x 14 mm
- Resolución de 128 x 64
- Alimentación de 3.3 o 5 v

Ya que no se requieren altas velocidades y se busca aprovechar su bajo consumo energético se optó por una pantalla OLED con el protocolo de comunicación I2C



5-50 Conexión pantalla OLED. (Elaboración propia)



5-51 Conexión OLED-Nano. (Elaboración propia)

5.5.8 Fuentes de Alimentación

Todos los dispositivos que se construyeron durante la tesis utilizan un rango de alimentación entre los 3.3 V y 12 V de corriente directa, al igual que la mayoría de los dispositivos electrónicos del mercado, es muy importante mantener el grado de tensión necesario para evitar picos de voltaje o caídas bruscas que puedan dañar el funcionamiento de los equipos.

Se utiliza una fuente conmutada Radox como la de la imagen para alimentar las cerraduras y sus dispositivos de control, para asegurar una alimentación constante y regulada.



5-52 Fuente Radox. (Elaboración propia)

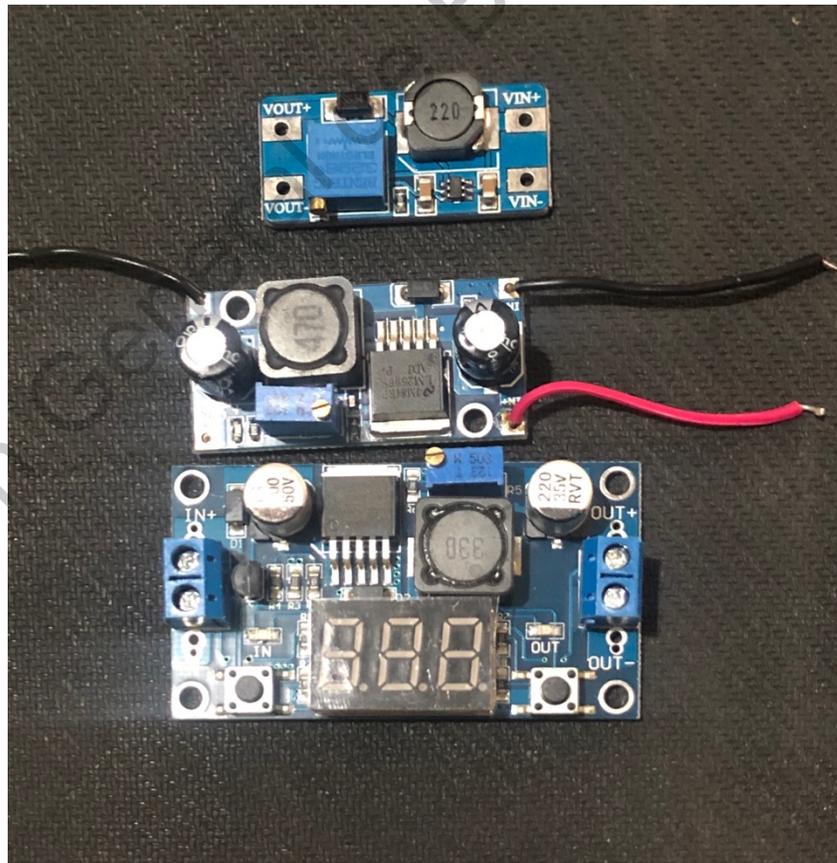
Especificaciones Técnicas

- Voltaje de entrada de 110 a 220 V AC
- Salida de 12 V (regulada de 10 a 14 V)
- Corriente de salida de 3 A
- Potencia de salida 36 Watts
- Carcasa de Aluminio con disipador
- Protección de sobrecarga
- Temperatura de trabajo de 0 A 40° C

5.5.9 DC-DC

En algunas partes del proyecto, los dispositivos requerían algún voltaje específico diferente al entregado por las fuentes de alimentación o que no empataba con la salida de los microcontroladores, gracias a los DC-DC es posible ajustar y regular tanto para elevar la tensión como para disminuirla.

Los DC-DC como en la imagen 5-53 permiten gracias a su tamaño ser parte del proceso de alimentación de dispositivos sin aumentar en gran medida el espacio disponible para ellos y entregar de manera exacta el suministro energético o realizar un acoplamiento.



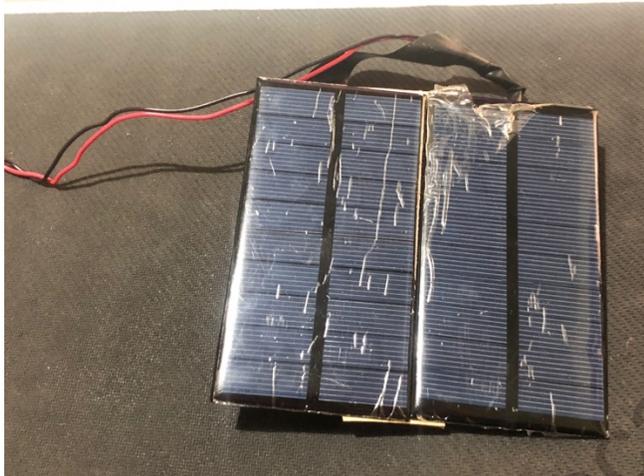
5-53 DC-DC. (Elaboración propia)

- El MT3608 es el primer DC-DC que se encuentra en la imagen, este permite elevar la tensión mientras el consumo energético no supere los 2 A, se utiliza para regular la salida del circuito de baterías con la entrada del microcontrolador para garantizar un voltaje de entre 5V y 7V desde una salida de 3.7 a 4.2 V
- El LM2506 realiza el propósito contrario, reduce la tensión de entrada desde una fuente de alimentación al valor correcto necesario por los sensores o microcontroladores, podemos ver dos versiones de diferente tamaño, una con display de exhibición de datos y la otra sin ella, el despliegue de datos aumenta el tamaño por lo que si no es necesario como parte del sistema se debe de omitir, se utiliza en las fuentes de 12 V para reducir el voltaje al valor de entrada de los microcontroladores y sensores.

5.5.10 Celda Solar

Dispositivo que capta la energía de la radiación solar para utilizarla ya sea en forma de colectores solares o paneles fotovoltaicos, para generar energía en forma de calor o electricidad respectivamente.

Los paneles fotovoltaicos están formados por celdas que convierten la luz en electricidad, los materiales de los que están fabricadas suelen ser silicio de cristalino o arseniuro de galio, pueden producir aproximadamente de 50 a 150 W/m² dependiendo de la cantidad de luz solar y la eficiencia de la celda.



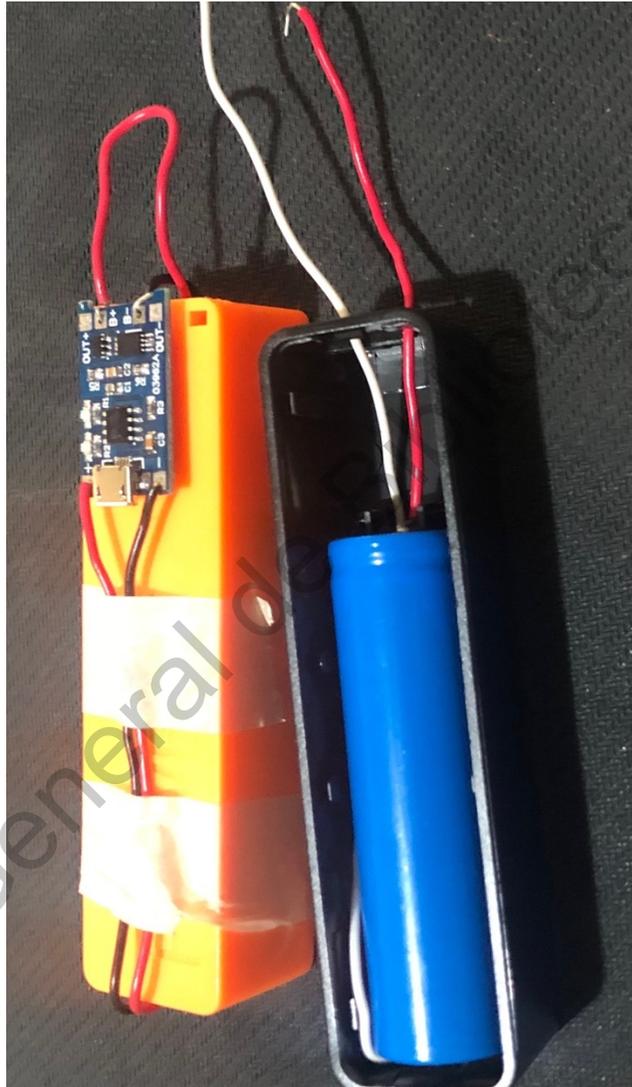
5-54 Celda Solar. (Elaboración propia)

Las celdas solares han incrementado su uso en la última década como una de las alternativas de energía limpia junto con los generadores eólicos, como parte de la filosofía del campus y su filosofía de Tecnología sustentable se trabaja en el uso de estas energías alternativas. Dentro del campus se cuenta con paneles solares de gran tamaño para alimentar baterías de reserva, se utiliza ese concepto a baja escala con celdas grupos de celdas como en la figura de 5V y 0.2 A de tal manera que puedan alimentar dispositivos que no dependan de la red eléctrica.

5.5.11 Baterías Li

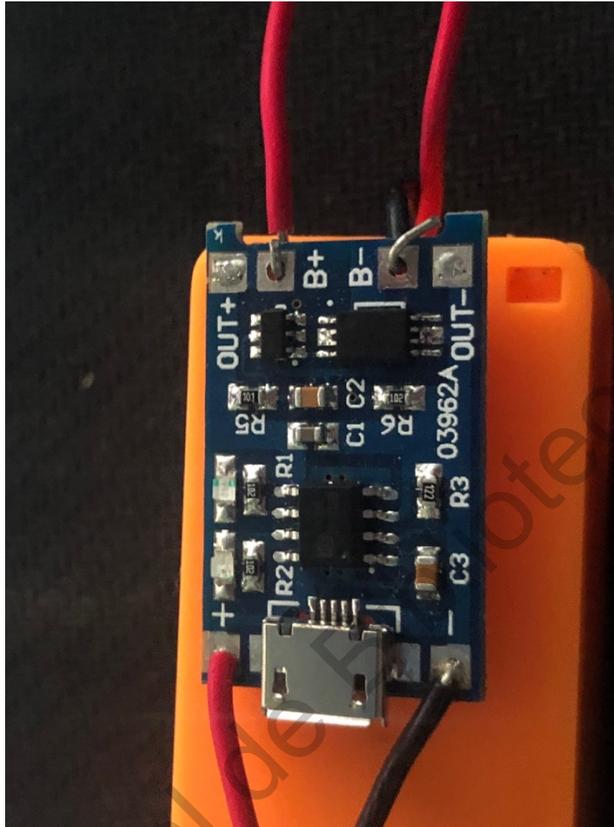
Son el acumulador de energía mas extendido en el mundo, ya que su uso comercial es bastante diverso en toda clase de equipos electrónicos, existen diferentes formas de baterías de Ion Litio dependiendo de la aplicación que se le de, existen en formas rectangulares y formas tubulares.

La capacidad de cada batería utilizada en esta tesis es de 3.7 V son de forma tubular y fueron recicladas de otros dispositivos de carga, se utilizan con el fin de almacenar la energía generada por las celdas solares.



5-55 Batería Litio. (Elaboración propia)

Entre la celda solar y la batería debe existir un dispositivo que regule la cantidad de energía que entra a la batería ya que esta puede ser dañada por una constante carga sin protección, se utiliza el circuito de carga RJMD36316, específicamente diseñado para cargar baterías de Litio con una entrada de 5v.



5-56 RJMD36316. (Elaboración propia)

El dispositivo cuenta con dos posibles entradas, micro USB y pines positivo y negativo dependiendo del tipo de entrada, una vez que la batería se encuentra completamente cargada es desconectada de forma automática del circuito para su protección; incluye dos salidas una para la batería (B+ y B-) y otra salida para el dispositivo que se va a alimentar, de esta manera mientras la celda solar se encuentra en funcionamiento ella misma alimenta al microcontrolador y carga la celda, al desactivar la celda solar por falta de luz, entra la batería a alimentar el sistema.

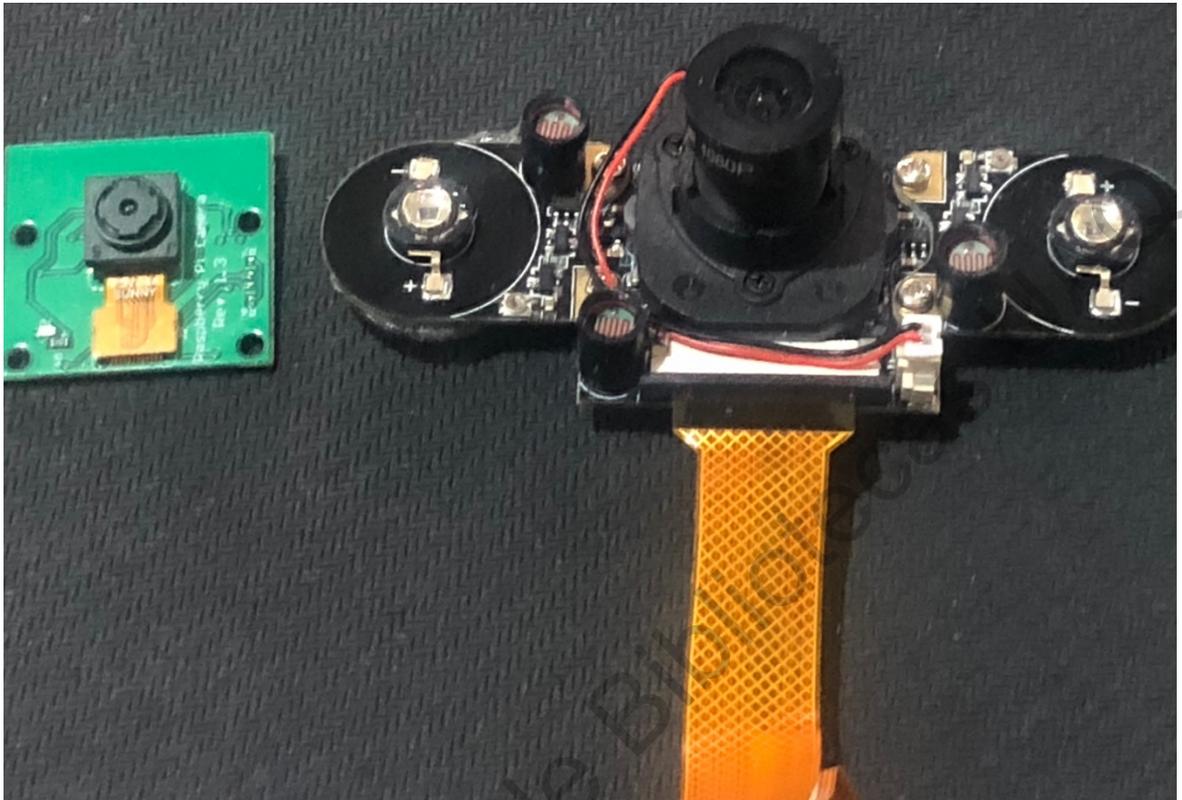
Ya que la batería genera 3.7 V y los microcontroladores utilizados trabajan a un voltaje mínimo de 5V se utiliza un DC-DC de subida para generar el incremento de voltaje, reduciendo en la misma proporción la cantidad de corriente.

5.5.12 Cámara Raspberry Pi

El enfoque de las medidas de seguridad siempre lleva consigo la vigilancia, la Raspberry cuenta con varios módulos de cámaras para diferentes propósitos, compatibles con todos sus modelos hasta el día de hoy, cuenta con un puerto dedicado y múltiple documentación para agregarlo a un proyecto de Internet de las cosas.

En la imagen se pueden apreciar las dos cámaras utilizadas en desarrollo y producción, la cámara de la izquierda es la cámara V1.3 de la compañía OEM

- Resolución de 5 MP.
- Video en 1080p a 30fps.
- Sensor OmniVision OV5647.
- Campo de visión horizontal de 54 grados.
- Campo de visión vertical de 41 grados.



5-57 Cámaras Raspberry Pi. (Elaboración propia)

La cámara de la derecha es una versión nocturna de la cámara regular, es igualmente compatible con todos los modelos de Raspberry que existen hoy, es compatible con leds infrarrojos como los que se muestran, lo que permite tomar imágenes de buena calidad sin la necesidad de luz visible.

Ambas cámaras cuentan con la misma configuración básica y para tomar videos o fotos en condiciones regulares de iluminación no presentan diferencias al momento de configurarse, la documentación de la comunidad de Raspberry provee librerías en Python para la activación y control de la cámara.

5.5.13 Sensores Infrarrojo y ambiental

Los sensores PIR (Infrarrojo Pasivo) es un módulo capaz de medir la radiación infrarroja de los cuerpos, enviando una señal digital si detecta un cambio en la intensidad del campo que recibe por causa del movimiento.

Es un sensor bastante sencillo y versátil ya que dentro del mismo se pueden realizar ajustes de manera manual sin necesidad de configuraciones por parte de un procesador, aunque también es posible realizarlo de esta manera.



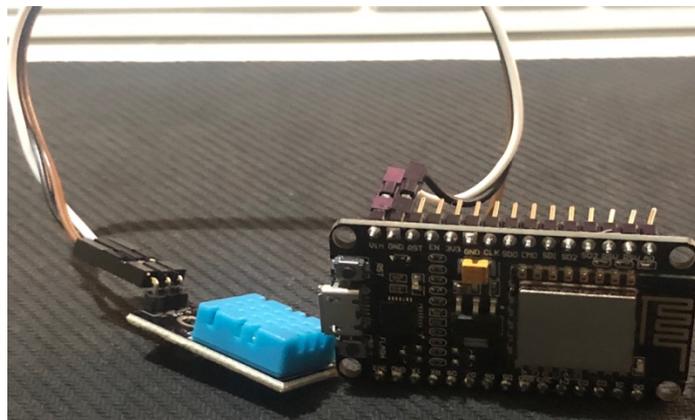
5-58 Sensor PIR. (Elaboración propia)

Dentro del diseño de dispositivos se trabaja junto con la Raspberry y su cámara para generar un dispositivo versátil sencillo y compacto que puede tomar imágenes y videos según la configuración, usualmente estos sensores PIR se utilizan con los pines digitales de las placas de microcontroladores, aquí es posible trabajar con la placa de computo Raspberry gracias a su tira de pines digitales que permite el acoplamiento a este tipo de sensores.



5-59 Raspberry Zero y PIR. (Elaboración propia)

El sensor de condiciones ambientales que se utiliza es el DHT11, que permite medir la humedad relativa y la temperatura utilizando un sensor capacitivo y un termistor respectivamente, el envío de datos al procesador es mediante un solo hilo de información lo que permite un bajo consumo recursos energéticos y de hardware.



5-60 Sensor ambiental con NodeMCU. (Elaboración propia)

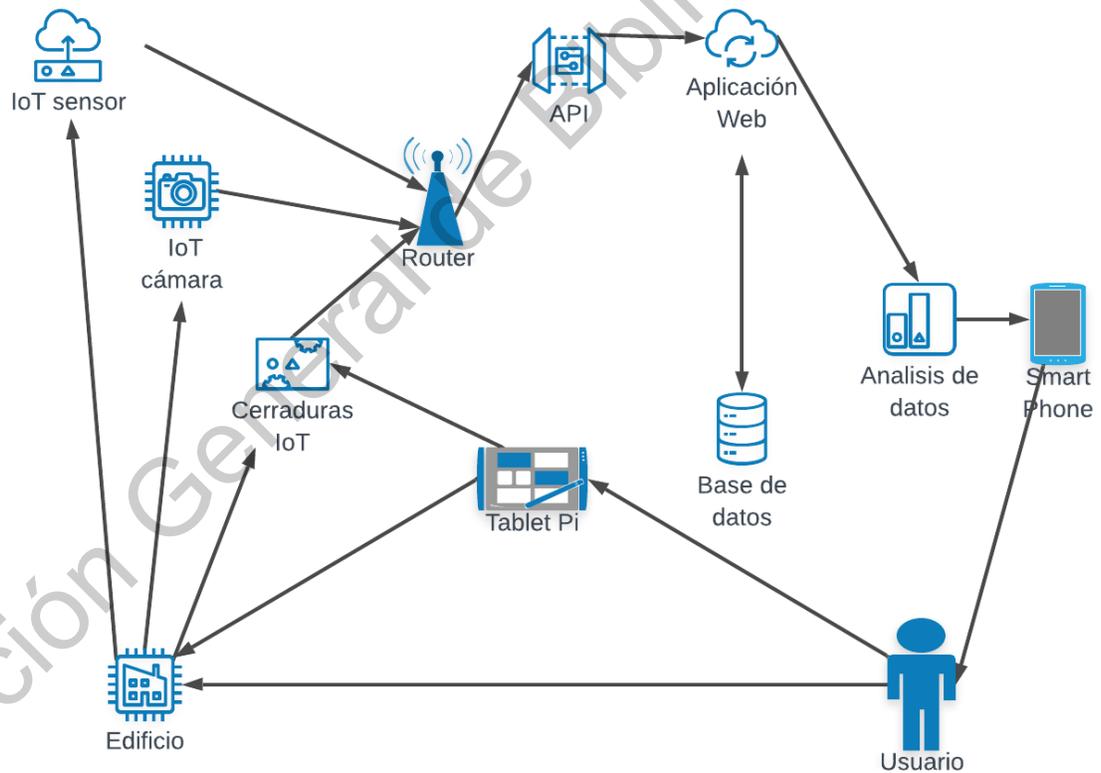
Sus características técnicas son.

- Voltaje de operación de 3-5 V
- Rango de temperatura de 0 a 50 °C
- Precisión de temperatura de $\pm 2^{\circ}\text{C}$
- Resolución de temperatura de 0.1°C
- Rango de humedad de 20% a 90% de RH
- Precisión de humedad de 5% de RH
- Resolución de humedad de 1% RH
- Tiempo de sensado 1 segundo
- Interfaz digital: Single-BUS
- Tamaño 16*12*5 mm (puede variar según modelo)

Debido a la simplicidad y bajo consumo energético de este sensor es ideal para trabajar el concepto de dispositivos sustentables, utilizando un microcontrolador de bajo consumo como el NodeMCU con conexión a internet y el sistema alimentado con la configuración propuesta de baterías y celdas solares.

5.6 Algoritmo de configuración de Dispositivos IoT

Se debe establecer una correcta conexión entre los dispositivos y la aplicación, así como la experiencia de usuario al entrar en contacto con el edificio debe ser lo más simple posible, de tal manera que el ambiente cuente con armonía que mejore la interacción del personal con un entorno inteligente. En la figura 5-61 se plantea el algoritmo general de conectividad entre los dispositivos diseñados, el edificio y los usuarios.



5-61 Algoritmo Usuario-Sistema. (Elaboración propia)

Dentro de las empresas y comunidad de desarrolladores de IoT no existe un estándar de conexión o formato establecido, por lo que existe una gran libertad al momento de decidir como se plantea la interacción y en que orden se envía o procesa la información. Por lo que el formato de solución e implementación es exclusivo para este diseño de edificio inteligente.

Se puede apreciar que una vez iniciada la interacción se genera un ciclo, pero eligiendo un punto de partida que sería natural en un ambiente real se realizaría de la siguiente manera.

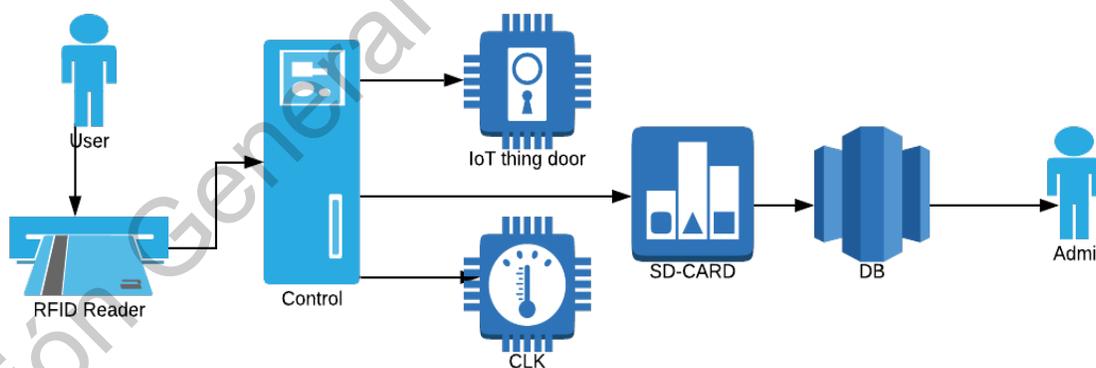
- El usuario tiene acceso a las instalaciones del edificio y los dispositivos conectados dentro.
- Dentro del edificio se encuentran los sensores ambientales, las cámaras y los accesos a los salones, oficinas y museo mediante las puertas electrónicas.
- Cuando uno o más de los dispositivos entran en contacto con el usuario dentro del edificio envían sus datos al Router con la conexión a internet de la Universidad, los dispositivos funcionan incluso sin la conexión a internet, únicamente que el proceso de información terminaría en este punto.
- Si la conexión de internet esta establecida con éxito se transfieren los datos con el protocolo HTTP a la API, utilizando el formato JSON.
- La API obtiene la información y si la entrada de datos es correcta y entran a la aplicación web donde son almacenados en la base de datos o actualizados.
- Se despliega la información en la vista correspondiente en formato de tablas
- La visualización y análisis se puede acceder mediante un dispositivo conectado a internet con un navegador web, ya sean computadoras o Smartphone.
- El usuario nuevamente entra en la interacción al recibir la información en su dispositivo y puede realizar los cambios que considere pertinentes con la

Tablet Pi, ya sea agregar o restringir accesos de las cerraduras vía OTA o cambiar la configuración de imagen o video de la cámara de la Raspberry mediante una conexión SSH.

En las siguientes secciones se describe más a detalle la configuración de cada uno de los dispositivos que interactúan dentro de este algoritmo.

5.7 Control de acceso sin conexión a Internet

Uno de los primeros dispositivos diseñados para esta tesis y probar la efectividad del control de usuarios mediante las tarjetas RFID fue construido con el propósito de conocer la hora de entrada y salida de profesores y conferencistas dentro de los salones del edificio “F” en el campus San Juan del Río de la UAQ.



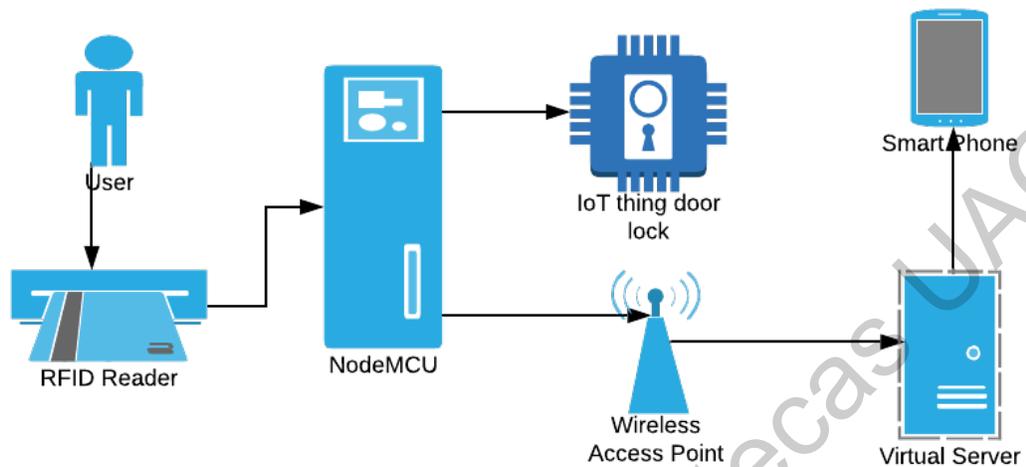
5-62 Control de Usuarios sin Internet. (Elaboración propia)

Consiste en un usuario que ingresa su tarjeta en el dispositivo en su hora de entrada y la tarjeta permanece dentro del mismo hasta la hora de salida, con el fin de contabilizar el tiempo que se estuvo en esa área. Su funcionamiento general es el siguiente.

- El usuario ingresa su tarjeta RFID personal dentro del sistema de control y permanece ahí hasta que la persona desea abandonar la instalación o terminar su registro.
- El control consiste en un microcontrolador sin conexión a internet y un sensor óptico que registra la llegada de la tarjeta.
- En caso de existir una cerradura, esta es manipulada por el control.
- Ya que no se cuenta con conexión a Internet, se utiliza un modulo DS3231 como reloj externo, lo que permite programar la hora y fecha dentro del sistema, estos datos son copiados del equipo que programa el dispositivo y tiene un retraso no superior a los 5 segundos. Tiene un sistema de alimentación propio para evitar perdida de información.
- La información del ID y del reloj interno es enviada a la tarjeta SD conectada mediante SPI al controlador.
- Esta información es enviada de forma manual a una base de datos en un equipo de computo por un administrador local cada cierto tiempo para prevenir un desborde de datos.

5.8 Control de acceso con Internet

Como parte de los dispositivos inteligentes los cuales deben tener acceso a internet, están las cerraduras y accesos utilizados en el campus de Tequisquiapan de la Universidad Autónoma de Querétaro, previamente a esto se debe tener la infraestructura necesaria de conexión fija de internet con una banda de 2.4GHz disponible en el Router.

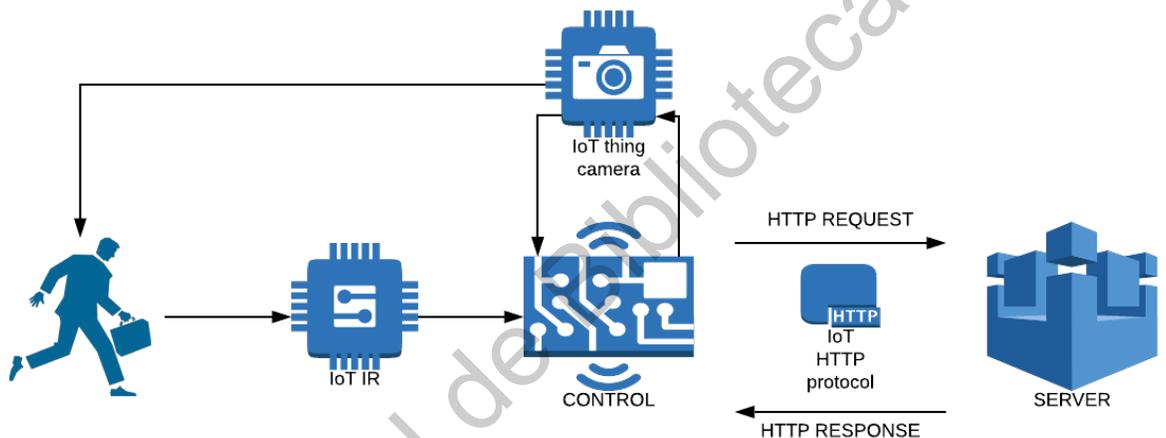


5-63 Control de Accesos con Internet. (Elaboración propia)

- El usuario pasa su tarjeta frente al lector RFID dentro del dispositivo, la conexión tarda unos segundos.
- La información de la tarjeta se envía al microcontrolador y se compara con la lista de ID que almacena dentro y tienen acceso al área.
- Si la información del identificador es correcta, se activa una etapa de potencia que permite la manipulación de la cerradura magnética por un instante determinado, permitiendo al usuario abrir la puerta.
- Simultáneamente si el dispositivo cuenta con acceso a Internet constante, envía los datos de la tarjeta y su localización dentro del edificio en forma de etiqueta.
- La información es almacenada dentro del servidor y procesada por el mismo.
- El despliegue de la información se presenta en un navegador web en los dispositivos de computo que tengan acceso a la URL de la aplicación.

5.9 Dispositivo de Seguridad Nocturna

Con la Raspberry Zero, el sensor PIR y la cámara Nocturna en conjunto se construye un dispositivo para almacenar la información de usuarios que se presenten el edificio fuera de las horas regulares de trabajo.



5-64 Control Nocturno de Acceso. (Elaboración propia)

- La Raspberry es preprogramada al finalizar la jornada laboral del edificio o cuenta con un cronómetro interno para que se active de forma automática el código correspondiente de la función de la cámara que se desea.
- Dentro de la Raspberry se encuentran diferentes códigos de Python que permiten realizar diferentes funciones, como Tomar fotos o videos cada cierta cantidad de tiempo, así como la revisión continua del sensor infrarrojo.
- Si es activada la función de PIR, cada vez que un usuario se mueva dentro del alcance de 180 grados del sensor, una señal es enviada a la Raspberry y activa la cámara según sea el caso en formato Fotografía o Video.

- El sensor y la cámara pueden trabajar en completa oscuridad ya que utilizan la radiación infrarroja de los cuerpos o de las luces led para detectar cambios en el ambiente.
- La información de las fotos o videos es almacenada dentro de la cámara y es transferida vía Internet a algún otro dispositivo, la aplicación web o un cliente de correo electrónico.

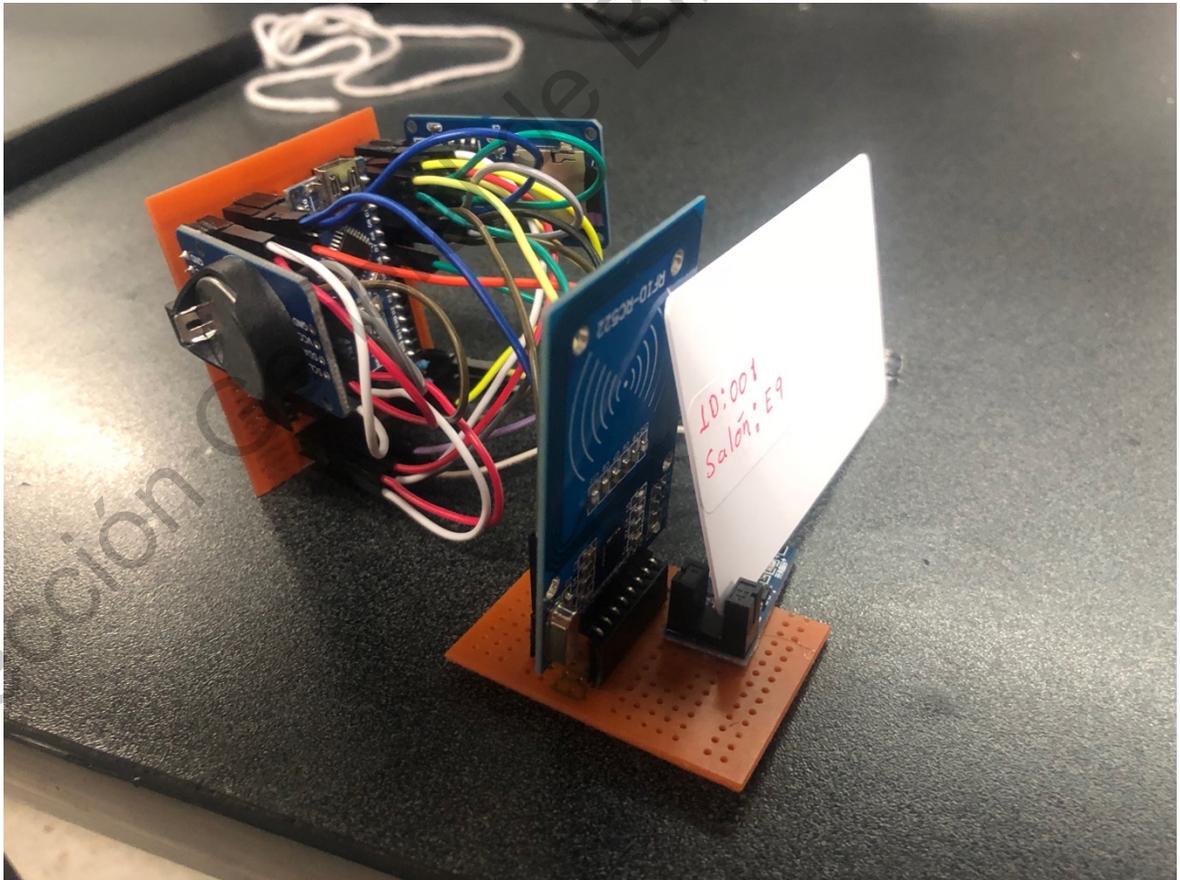
Dirección General de Bibliotecas UAG

6 Resultados

En esta sección se muestran las imágenes y aplicaciones de los dispositivos descritos en la sección de Metodología, así como los elementos que se agregaron al pasar de la etapa de prueba a la de producción.

6.1 Dispositivo de control de acceso sin conexión a Internet

En la figura 6-1 se puede muestra el primer diseño terminado del dispositivo descrito en la sección 5.7, utilizando el nombre DNI-9 para referirse a el, se muestra la conexión entre los módulos RFID, SD, reloj y el sensor óptico que sostiene la tarjeta se encarga de iniciar el conteo del tiempo que la tarjeta esta conectada.



6-1 DNI-9. (Elaboración propia)

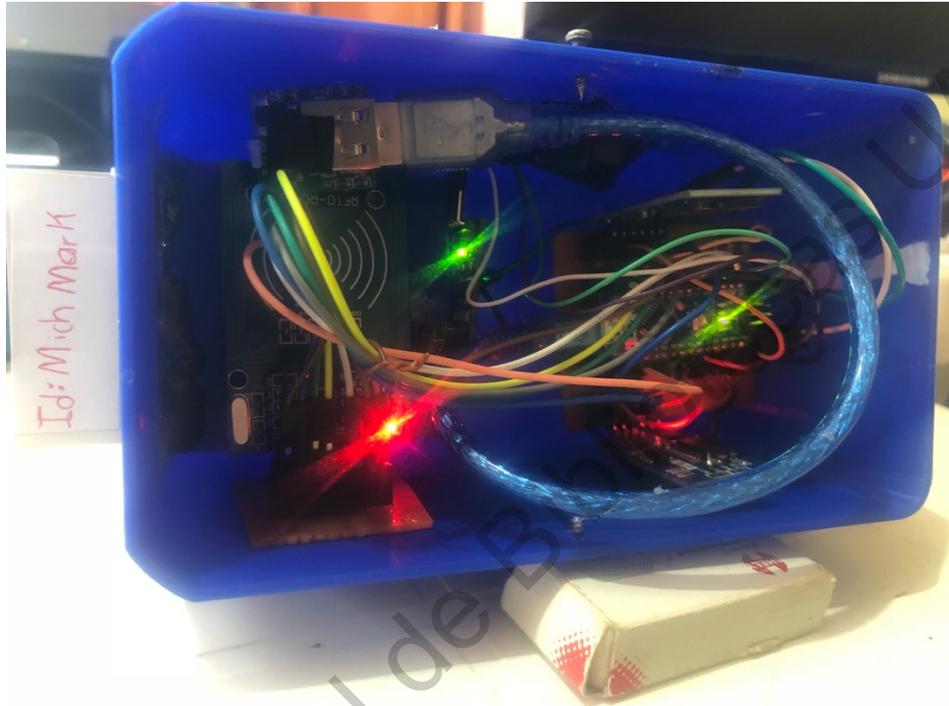
En la figura 6-2, se agrega una etapa de acoplamiento al sistema con una fuente regulada de 120 V AC a 5 V DC, con el propósito de utilizar la entrada micro USB del Arduino Nano, el cual alimenta los demás componentes.

Se muestra la carcasa diseñada para contener el sistema dentro de los salones y asegurar su protección.



6-2 Carcasa y Alimentación DNI-9. (Elaboración propia)

En las figuras 6-3 y 6-4 se observa el dispositivo DNI-9 terminado e instalado para iniciar su interacción con el usuario.



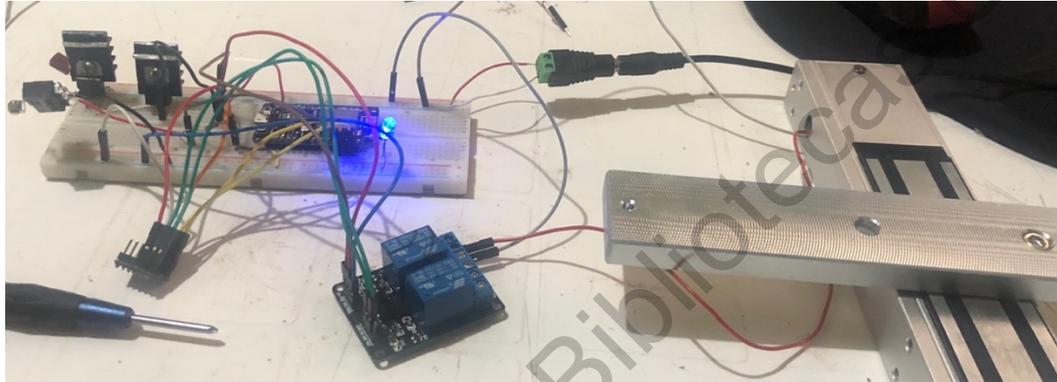
6-3 DNI-9 Acoplado. (Elaboración propia)



6-4 DNI-9 Instalado en SJR. (Elaboración propia)

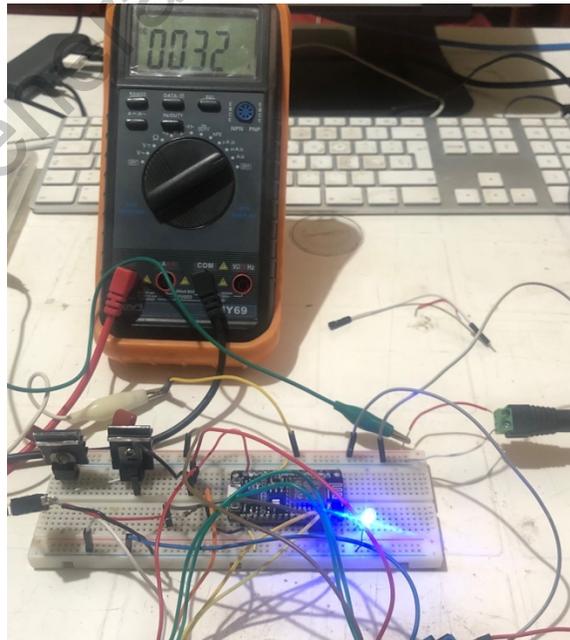
6.2 Dispositivo de control de Acceso en CATAS

En la figura 6-5 se muestra el diseño del dispositivo de control de cerraduras con el NodeMCU, al que se refiere como MAJI-2 en una placa de pruebas para verificar el funcionamiento y acoplamiento de los componentes, se utiliza una fuente de alimentación de 12 V regulada y de manera interna se realiza la distribución de energía mediante reguladores lineales y acopladores lógicos.



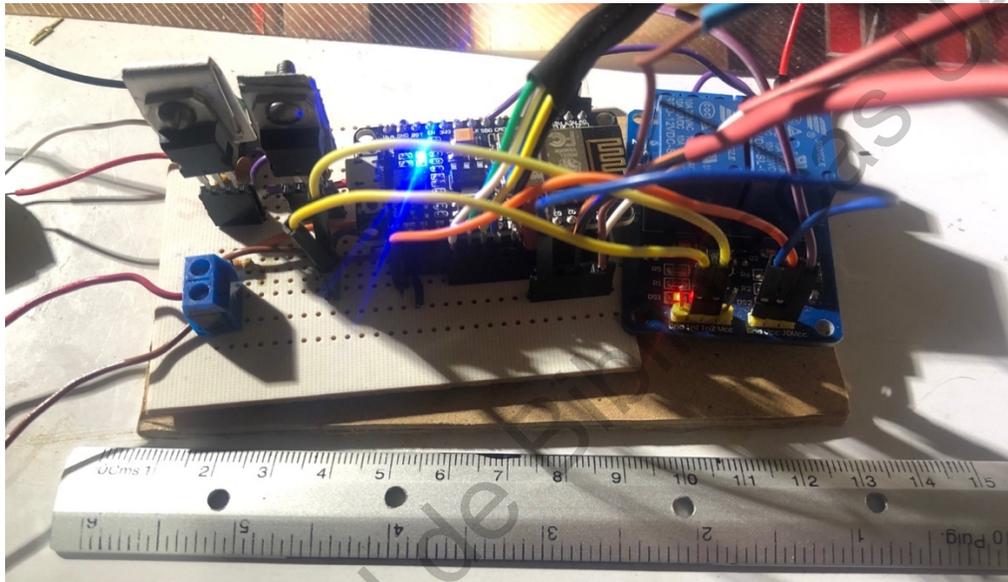
6-5 Pruebas MAJI-2. (Elaboración propia)

Durante la etapa en la placa de pruebas se realizó el cálculo de la corriente que consume el dispositivo como se muestra en la imagen 6-6 es menor a 350mA .



6-6 Consumo de Corriente MAJI-2. (Elaboración propia)

En la figura 6-7 se muestra el dispositivo terminado para el control de acceso, con la etapa de acoplamiento lógico, alimentación y etapa de potencia, resulta un dispositivo de aproximadamente 13 cm. Por lo cual es un 30% mas pequeño que el diseño anterior.



6-7 Producción MAJI-2. (Elaboración propia)

En la imagen 6-8 se muestra una prueba que se realizó al dispositivo antes de instalarlo para verificar la respuesta de la API al envío de datos del dispositivo.

```
Waiting for connection
WiFi connected
IP address:
192.168.1.116
{
  "number": "123",
  "room": "1"
}
200
{"message": "access success to room 1"}
```

6-8 Mensaje API exitoso. (Elaboración propia)

Se realizaron las conexiones eléctricas dentro del edificio CATAS, sacando una línea de alimentación de 120V AC para conectar la fuente y posteriormente el dispositivo.



6-9 Conexión Eléctrica en CATAS. (Elaboración propia)

La cerradura solo puede ser abierta mediante la llave de acceso RFID desde afuera de las instalaciones, por dentro se instalaron interruptores de carrera, que permiten desactivar por unos instantes la cerradura para poder abrir la puerta por dentro, esta señal es enviada al NodeMCU como una señal digital mediante un acoplamiento lógico.

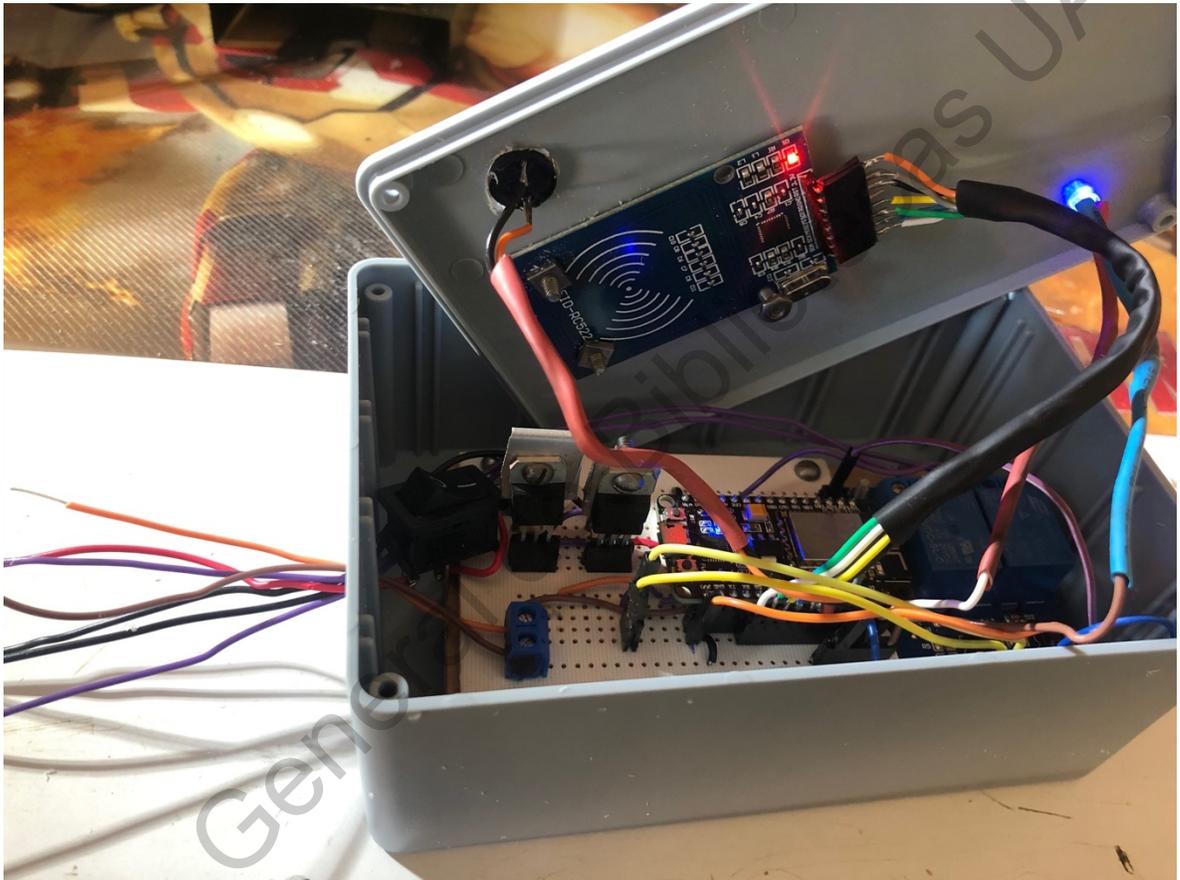


6-10 Conexión Interruptor de carrera. (Elaboración propia)



6-11 Interruptor instalado en oficina. (Elaboración propia)

En la imagen 6-12 se observa el dispositivo terminado para su instalación, para comodidad del usuario se agregaron un led azul y un zumbador como indicadores físicos cuando la cerradura es activada o desactivada mediante el interruptor o la llave RFID.



6-12 MAJI-2 producción. (Elaboración propia)

En la imagen 6-13 se muestra la cerradura instalada a fuera de la oficina principal de CATAS para regular su acceso



6-13 MAJI-2 instalado en CATAS. (Elaboración propia)

La figura 6-14 muestra un usuario accediendo a la oficina principal de CATAS con una pulsera RFID

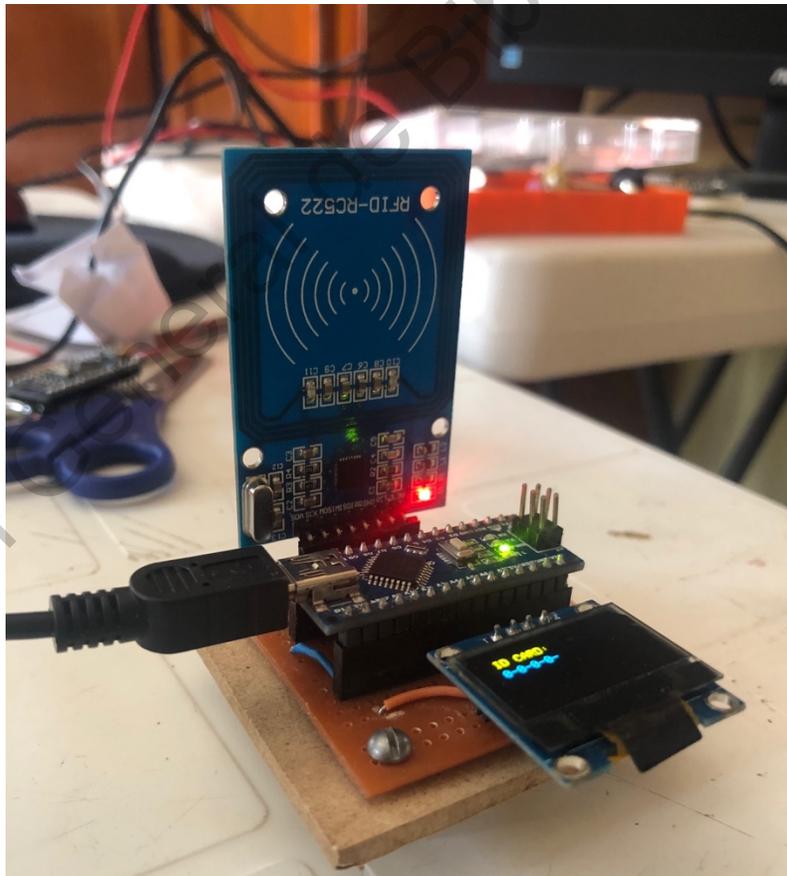


6-14 Acceso con pulsera RFID. (Elaboración propia)

6.3 Identificador de tarjetas RFID

El dispositivo descrito en esta sección es una extensión necesaria de los dispositivos descritos en la sección 6-2 y 6-3, consiste en un identificador de tarjetas RFID, permite a un administrador leer el número hexadecimal identificador de cada TAG, antes de ser entregadas a un usuario, si se desea agregar a la lista de llaves que tienen acceso o removerla.

Es un dispositivo sencillo que consta de una lectora de RFID, un Arduino Nano para tener un tamaño reducido y una pantalla OLED conectada por I2C al Arduino, está diseñado de una manera compacta como se ve en la imagen 6-15 para estar en el escritorio del administrador.



6-15 Identificador de tarjetas. (Elaboración propia)

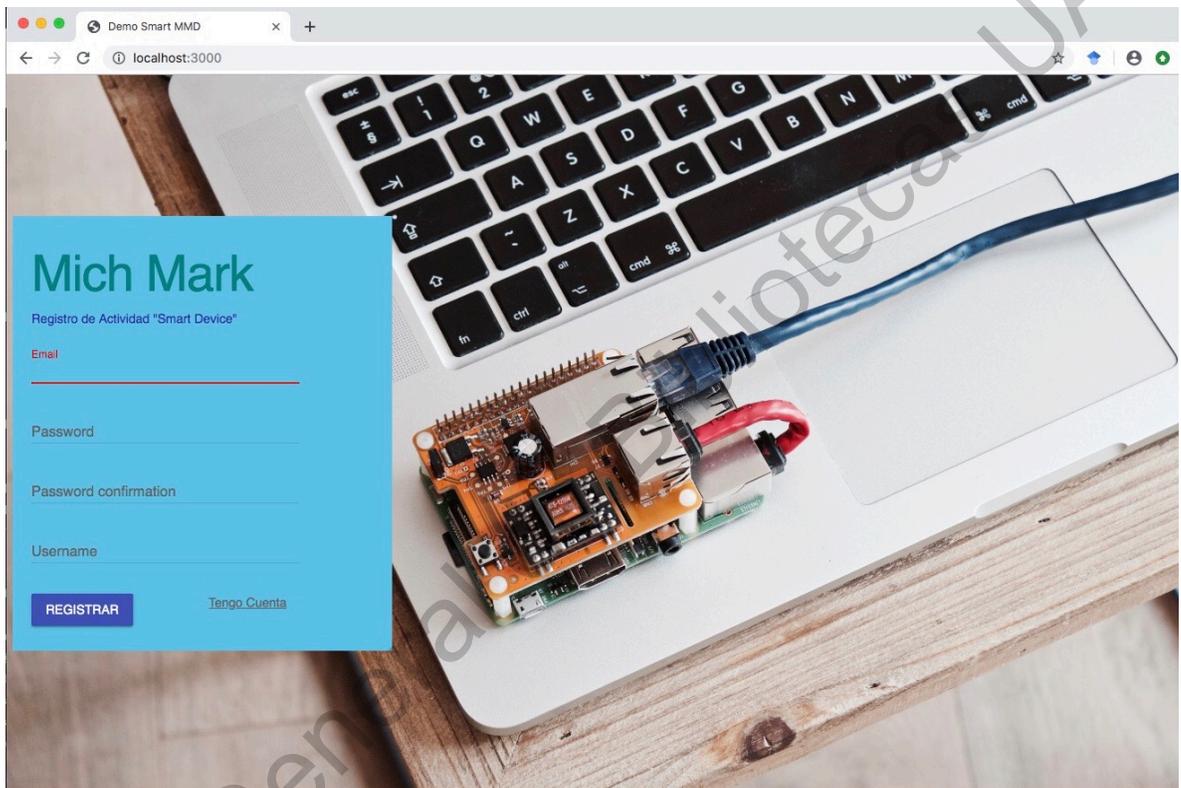
En la figura 6-16 se muestra el identificador funcionando, recibe la información del ID, del TAG dentro de la pulsera, esta pulsera funciona como control general dentro de las instalaciones, por motivos de seguridad el numero no es mostrado.



6-16 Identificador con pulsera. (Elaboración propia)

6.4 Aplicación WEB

En la figura 6-17 se muestra la interfaz de inicio de la aplicación web, de la aplicación de control en el edificio CATAS, requiere un acceso de usuario que necesita llenarse. En la imagen 6-18 se muestra el menú de acceso después del registro.



6-17 APP WEB CATAS. (Elaboración propia)



6-18 Menú APP. (Elaboración propia)

En la figura 6-19 se muestran algunos de los accesos registrados, los usuarios mich123 y cerradura1 son registros generados en la etapa de pruebas, y los usuarios museo y oficina son los primeros espacios en tener el sistema funcionando de tiempo completo.

Las llaves 1 a 5 mostradas en la parte izquierda son los TAGs entregados a los usuarios y administradores designados para tener total acceso al edificio, la llave azul es la pulsera con el TAG maestro.

Dispositivos CATAS		
Body	User	
prueba catas completo 9	cerradura1@gmail.com	Show Edit Destroy
key blue	museo@catas	Show Edit Destroy
key blue	oficina@catas	Show Edit Destroy
key 1	oficina@catas	Show Edit Destroy
key 2	museo@catas	Show Edit Destroy
key 5	museo@catas	Show Edit Destroy
key 2	oficina@catas	Show Edit Destroy
key 2	oficina@catas	Show Edit Destroy
key 3	oficina@catas	Show Edit Destroy
usuario 15324822142	mich123@test.com	Show Edit Destroy

6-19 Tabla de Accesos. (Elaboración propia)

Si se entra al link mostrado al lado derecho de la tabla, en el menú de mostrar se despliega una pantalla como la que aparece en la imagen 6-20, el cuerpo contiene la llave de acceso, la segunda línea muestra la zona del edificio donde ocurrió el acceso y finalmente la tercera línea muestra la hora en la que sucede la entrada, el horario que se muestra en la aplicación es el UTC, por lo tanto esta 6 horas adelantado del horario de nuestro país.



6-20 Información de Acceso. (Elaboración propia)

6.5 Sensor ambiental con Celdas solares

La construcción del sensor ambiental es sencilla, es alimentado por el par de celdas solares conectadas en paralelo para aumentar la corriente que puede ser proporcionada, la energía se regula mediante el circuito que administra la batería de Litio, el sensor ambiental se encuentra fuera de la caja para no tener interferencia del calor que disipan los elementos electrónicos.



6-21 Sensor ambiental. (Elaboración propia)

6.6 Dispositivo de vigilancia nocturna

El primer diseño de pruebas de cámara con Raspberry se realizó con unas tarjetas modelo B y la cámara V1.3 de 5MP, el sistema cumplía de forma parcial el propósito, ya que las fotografías nocturnas no se podían ver y el sistema resultaba demasiado sobrado y costoso, por lo que se optó por utilizar una Raspberry con menor consumo.



6-22 Raspberry P y cámara 5MP. (Elaboración propia)

La figura 6-23 muestra el primer dispositivo diseñado con la cámara nocturna y la Raspberry Zero, en una de las cajas de los dispositivos RFID, después de hacer la prueba de concepto se procedió a mejorar el diseño para reducir el tamaño.



6-23 Primer sistema nocturno. (Elaboración propia)

Una vez que se termina el diseño de la carcasa, se reduce el tamaño del dispositivo, se vuelve una versión portátil, con acceso a los periféricos para la configuración.



6-24 Versión Producción Sistema Nocturno. (Elaboración propia)

El utilizar la Raspberry Zero reduce la cantidad de energía necesaria para alimentar el sistema, por lo tanto, puede trabajar de forma portátil con un banco de batería para celular, la cual le brinda una autonomía de 6 a 10 horas y el tamaño fue reducido considerablemente menor que el original.



6-25 Sistema de Vigilancia portátil. (Elaboración propia)

En la figura 6-26 se muestra una imagen del dispositivo instalado en el edificio CATAS, se observa que la ausencia de conexiones eléctricas permite que sea colocado en la zona que se considere mas conveniente.



6-26 Sistema instalado en CATAS. (Elaboración propia)

En la imagen 6-27 se aprecia una fotografía tomada por el sistema de vigilancia nocturno en un ambiente de prueba diseñado, para percibir la cantidad de detalles de la imagen. Se pueden apreciar los detalles de los objetos colocados en una habitación con total oscuridad.



6-27 Foto Nocturna. (Elaboración propia)

6.7 Tablet Pi

La Tablet Pi es un dispositivo diseñado para llevar el control y ver la información de los dispositivos diseñados durante esta tesis, esta computadora hecha con una Raspberry y la pantalla LCD Táctil descrita en el capítulo anterior, utiliza un teclado inalámbrico como el que se observa en la imagen, en caso de requerirlo.



6-28 Teclado Externo Tablet Pi. (Elaboración propia)

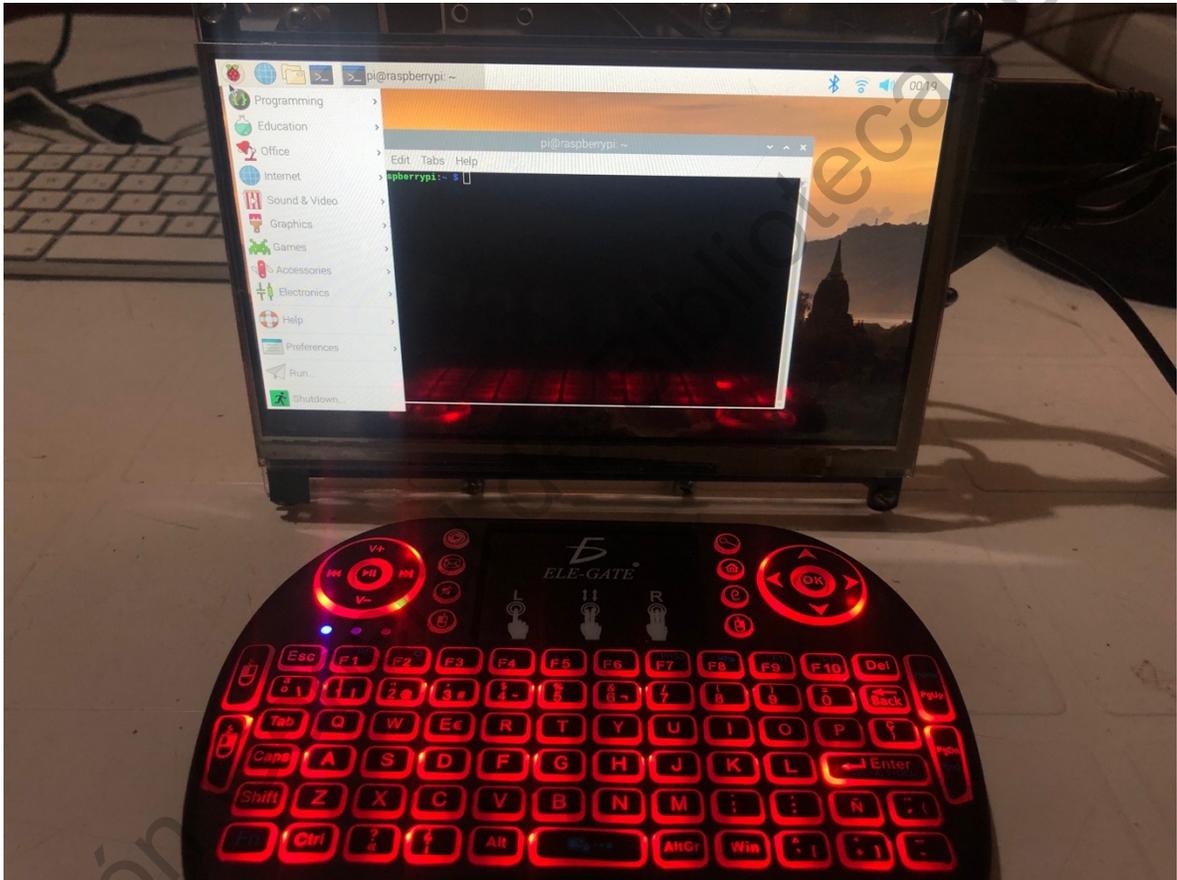
En la imagen 6-29 se muestra la Raspberry conectada a la pantalla LCD mediante un cable HDMI y un cable micro USB para la alimentación.



6-29 Raspberry Pi B en Tablet Pi. (Elaboración propia)

El diseño rustico de la Tablet Pi no es impedimento para tener versatilidad, puede ser colocada en forma horizontal o vertical sobre una superficie plana, en la imagen 6-30 se observa la terminal de la Tablet y el menú de aplicaciones disponibles.

- Este dispositivo se encarga de revisar la aplicación web mediante el navegador.
- Reprogramar las cerraduras con el protocolo OTA.
- Acceder a la cámara mediante SSH.
- Cubrir funciones de equipo de escritorio de forma móvil.



6-30 Tablet Pi y Teclado. (Elaboración propia)

7 Conclusiones

El desarrollo de esta tesis fue un reto académico que permitió trabajar con áreas de desarrollo de nuestra década, los entornos inteligentes son cada vez mas comunes y la interacción mas natural con los usuarios es un tópico que requiere su propia dedicación.

Es importante construir las bases para que futuros trabajos y desarrollos tecnológicos surjan de las aulas de las universidades de México, produciendo en la medida de lo posible nuestra propia tecnología.

Todos los equipos diseñados e instalados en el transcurso de este trabajo fueron diseñado bajo un sistema de bloques lo que permite mejorarlos o adaptarlos a otros trabajos con una cantidad mínima de modificaciones, espero sea una semilla plantada dentro de una tierra fértil llena de nuevas ideas.

Utilizando los demás equipos instalados en el edificio CATAS es posible mejorar la sustentabilidad energética que presentan los diseños actuales, generando autonomía cada vez mas marcada de las clásicas fuentes energéticas.

Los dispositivos tienen una relativa autonomía y se mantendrán funcionando sin necesidad de un programador que los gestione de manera directa, sin embargo, se debe considerar tener un mantenimiento ideal para la plataforma y servicio técnico de rutina para todos los dispositivos.

Algunos elementos de funcionamiento no fueron explicados o incluidos dentro del presente documento por motivos de seguridad, ya que es necesario mantener los accesos remotos a los dispositivos o la aplicación únicamente en mano de los administradores del edificio, ellos determinaran si se desea ingresar nuevos desarrolladores al código fuente de la aplicación o de los dispositivos.

Se deja abierta la posibilidad a probar estos avances en otros campus de la Universidad Autónoma de Querétaro, y a diseñar nuevos dispositivos bajo los algoritmos y premisas aquí planteados para atacar los nuevos retos que se presenten.

Dirección General de Bibliotecas UAQ

8 Referencias

IoT Analytics GmbH (2019) IoT Project List: Database Of Enterprise IoT Projects 2018. Recuperado de: <https://iot-analytics.com/product/list-of-1600-enterprise-iot-projects-2018/>

Al Zamil, M. G., Rawashdeh, M., Samarah, S., Hossain, M. S., Alnusair, A., & Rahman, S. M. M. (2018). An Annotation Technique for In-Home Smart Monitoring Environments. *IEEE Access*, 6, 1471-1479.

Bächle, M., & Kirchberg, P. (2007). Ruby on rails. *IEEE software*, 24(6).

Biswas, A. R., & Giaffreda, R. (2014, March). IoT and cloud convergence: Opportunities and challenges. In *2014 IEEE World Forum on Internet of Things (WF-IoT)* (pp. 375-376). IEEE.

Durski, K., Murlewski, J., Makowski, D., & Sakowicz, B. (2011, February). Warehouse management system in Ruby on Rails framework on cloud computing architecture. In *2011 11th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)* (pp. 366-369). IEEE.

Fox, J., Donnellan, A., & Doumen, L. (2019, April). The deployment of an IoT network infrastructure, as a localised regional service. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (pp. 319-324). IEEE.

Garg, H., & Dave, M. (2019, April). Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)* (pp. 1-6). IEEE.

Gu, C. (2018). Fast Discrepancy Identification for RFID-Enabled IoT Networks. *IEEE Access*, 6, 6194-6204.

Kossonon, B. E., & Ya, W. H. (2017). IOT based smart restaurant system using RFID.

Mados, B., Hurtuk, J., Chovancová, E., Fecil'ák, P., & Bajkó, D. (2017, November). Downsizing of web server design using Raspberry Pi 3 single board computer platform. In *Informatics, 2017 IEEE 14th International Scientific Conference on* (pp. 238-242). IEEE.

Martínez Luna, A. (2017). *Desarrollo de sistema de monitoreo y control de acceso para la seguridad del CATAS*.

Pazos, N., Müller, M., Aeberli, M., & Ouerhani, N. (2015, December). ConnectOpen-automatic integration of IoT devices. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (pp. 640-644). IEEE.

Patil, N., Ambatkar, S., & Kakde, S. (2017, April). IoT based smart surveillance security system using raspberry Pi. In *Communication and Signal Processing (ICCSP), 2017 International Conference on* (pp. 0344-0348). IEEE.

Paul, P. V., & Saraswathi, R. (2017, March). The Internet of Things—A comprehensive survey. In *Computation of Power, Energy Information and Commuincation (ICCPEIC), 2017 International Conference on* (pp. 421-426). IEEE.

Rane, S., Dubey, A., & Parida, T. (2017, July). Design of IoT based intelligent parking system using image processing algorithms. In *Computing Methodologies and Communication (ICCMC), 2017 International Conference on* (pp. 1049-1053). IEEE.

Redondo, J. M., & Ortin, F. (2015). A comprehensive evaluation of common python implementations. *IEEE Software*, 32(4), 76-84.

Salman, L., Salman, S., Jahangirian, S., Abraham, M., German, F., Blair, C., & Krenz, P. (2016, December). Energy efficient IoT-based smart home. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)* (pp. 526-529). IEEE.

Sruthy, S., & George, S. N. (2017, August). WiFi enabled home security surveillance system using Raspberry Pi and IoT module. In *Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2017 IEEE International Conference on*(pp. 1-6). IEEE.

Tavade, T., & Nasikkar, P. (2017, March). Raspberry Pi: Data logging IOT device. In *Power and Embedded Drive Control (ICPEDC), 2017 International Conference on* (pp. 275-279). IEEE.

Zamudio Ramírez, I. (2016). *Desarrollo de un sistema de adquisición de señales provenientes de sensores para confort y calidad energética en CATAS* (Doctoral dissertation).