



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“DISEÑO E IMPLEMENTACIÓN DE SISTEMA BASADO EN
SENSORES DE BAJO CONSUMO DE ENERGÍA PARA
CONTROL DE ACCESO”**

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN

**ADRIANO XAVIER PINARGOTE VERA
LEONARDO DAVID ERAS DELGADO**

GUAYAQUIL – ECUADOR

AÑO: 2017

AGRADECIMIENTOS

Nuestros más sinceros agradecimientos al personal del laboratorio de prototipado rápido del Centro de Tecnologías e Investigación por brindarnos un lugar para desarrollar el presente proyecto, así también por las herramientas y conocimientos que pudieron aportar.

DEDICATORIA

El presente proyecto lo dedico a mis padres, tutores, amigos y colegas del trabajo, quienes con su apoyo, enseñanza y paciencia lograron hacerme llegar hasta este punto de mi vida.

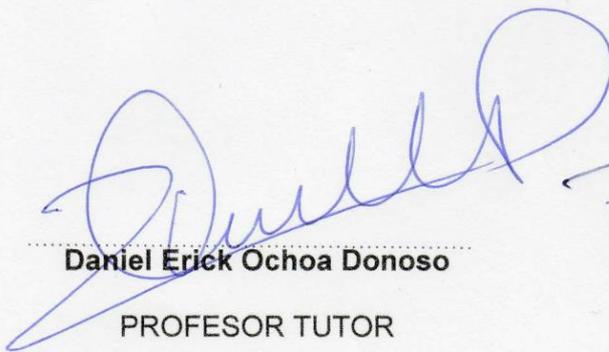
Leonardo Eras Delgado

DEDICATORIA

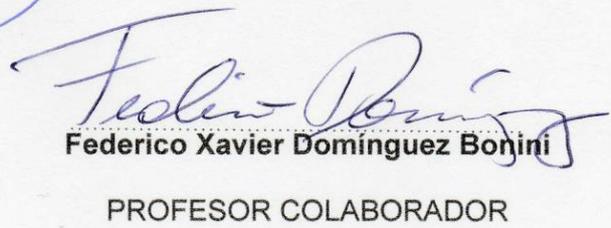
El presente proyecto lo dedico a mis padres que se esforzaron hasta el final por verme culminar mi carrera, a mis hermanos, la familia Pinargote Cruz, la familia Montero Pinargote, y todos mis amigos; gracias a ellos que me apoyaron espiritualmente, emocionalmente y económicamente; y a Génesis Vásquez Zambrano por ser un apoyo incondicional y un pilar en el desarrollo en mi carrera.

Adriano Pinargote Vera

TRIBUNAL DE EVALUACIÓN



Daniel Erick Ochoa Donoso
PROFESOR TUTOR



Federico Xavier Dominguez Bonini
PROFESOR COLABORADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

.....
Adriano Xavier Pinargote Vera

.....
Leonardo David Eras Delgado

RESUMEN

El CTI (Centro de Tecnologías e Investigación) es un centro abierto tanto a estudiantes como profesores de la ESPOL que deseen realizar investigaciones empleando las tecnologías tanto de software, de hardware e impresiones en 3D.

Existen dos áreas principales del edificio: los laboratorios y las oficinas, ambas restringidas por puertas que pueden ser abiertas sólo con tarjetas especiales. Esto es frustrante para trabajadores, debido a que tienen que constantemente revisar las puertas por si hay algún visitante esperando para que les den acceso, interrumpiendo su trabajo.

Por ello se ha propuesto la siguiente solución: timbres inteligentes que permitan alertar a los trabajadores a través de mensajes con su respectiva aplicación para dispositivos móviles Android y iOS. Los timbres deben de funcionar en base a baterías recargables, y mientras no se estén usando, consumir la menor cantidad de energía posible. La conexión entre los timbres y la aplicación emplea el protocolo MQTT, alojado en un servidor CentOS 7. MQTT es un protocolo de mensajes simple que toma en consideración las limitantes en recursos como ancho de banda y batería que los timbres poseen.

La aplicación les permite a los trabajadores, mediante la lectura de un código QR asociado al timbre, suscribirse a éste. Se puede estar suscrito a más de un timbre a la vez. Apenas suscrito podrá recibir notificaciones cuando alguien presione el timbre. El trabajador puede abandonar la suscripción en cualquier momento.

Los resultados demuestran que el proyecto es viable, y que se pueden realizar mejoras significativas en cuanto a seguridad y control a distancia.

ABSTRACT

The Center for Information Technology (CTI) is an open center building for ESPOL students and professors alike who would like to make research by employing software, hardware and 3D printing technologies.

There are two main areas of the building: laboratories and office areas, both restricted by doors which can only be opened with access cards. This is annoying for employees as they constantly must be checking out the doors in case of visitors, interrupting their work, as visitors must wait for someone who has an access card to open the door for them.

In this work, intelligent doorbells are proposed as a possible solution, which can alert workers through notification messages in an app developed both in Android and iOS. The doorbells must work with rechargeable batteries, and while not using them, they must operate with the least amount of energy possible. Connection between doorbells and the mobile application uses MQTT as a protocol, and the broker is allocated in a CentOS 7 server. MQTT is a simple message passing protocol which considers limited resources such as bandwidth and energy in devices.

Workers can subscribe to the doorbells by using the mobile application and by reading a QR code associated with it. One can be subscribed to more than one bell at the time. Once subscribed, the app will receive notifications when someone presses the button of the doorbell. Any worker can cancel their subscription anytime they want.

The results of this work show that the project is viable, and significant improvements regarding security and remote access can be made.

ÍNDICE GENERAL

RESUMEN	I
ABSTRACT	II
ABREVIATURAS.....	V
SIMBOLOGIA.....	VII
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
CAPÍTULO 1	1
1 INTRODUCCIÓN.	1
1.1 Antecedentes	1
1.2 Descripción del problema.....	1
1.3 Objetivo General	1
1.4 Objetivos Específicos	1
1.5 Componentes de la solución	2
1.6 Concepto del problema	3
CAPÍTULO 2	5
2. PHOTON PARTICLE.	5
2.1 Características del Photon	5
2.2 Diagrama de bloques	5
2.3 Consideraciones eléctricas	6
2.4 Definiciones de pines	6
2.5 Códigos de colores del Photon	8
CAPÍTULO 3	9
3. DISEÑO DE LA SOLUCIÓN	9
3.1 Esquema de la solución	9
3.2 Tecnologías de desarrollo utilizadas	10

3.2.1	Ionic Framework	10
3.2.2	AngularJS	11
3.2.3	MQTT	11
3.2.4	Clientes y <i>Brokers</i>	11
3.2.5	Websockets	12
3.2.6	Código QR	12
3.2.7	Particle Web IDE	14
3.3	Librerías y plugins usados.....	15
3.3.1	Cordova Camera plugin	15
3.3.2	Mosquitto	15
3.4	Prototipo.....	15
3.5	Descripción del sistema con diagramas de estados.....	16
3.6	Descripción del sistema con diagrama secuencial.....	19
CAPÍTULO 4		23
4.	DETALLES DE LOS RESULTADOS.....	26
4.1	Resultados de la primera prueba	26
4.2	Resultados de la segunda prueba.....	27
4.3	Resultados de pruebas de rendimiento.....	27
4.4	Ahorro del dispositivo	31
CONCLUSIONES Y RECOMENDACIONES		35

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
CTI	Centro de Tecnologías e Investigación
MQTT	MQ Telemetry Transport
QR	Quick Response
IDE	Integrated Development Environment
MAC	Media Access Control
IoT	Internet of Things
LED	Light Emitting Diode
RAM	Random Access Memory
USB	Universal Series Bus
OS	Operating System
RGB	Red Green Blue
ARM	Advanced RISC Machines
GPIO	General Purpose Input Output
VCD	Voltaje Corriente Directa
VIN	Voltage Input
GND	Ground
SRAM	Static Random-Access Memory
VDD	Voltage Drain Drain
RST	Reset
RX	Receptor
UART	Universal Asynchronous Receiver Transmitter
TX	Transmisor
PWM	Pulse Width Modulation

WKP Wakeup

DAC Digital to Analog Converter

ADC Analog to Digital Converter

M2M Machine to machine

SSL Secure Sockets Layer

HTML(5) HyperText Markup Language (Versión 5)

JS JavaScript

RFID Radio-Frequency Identification

NFC Near Field Communication

SIMBOLOGIA

cm	Centímetro
mm	Milímetro
mV	Milivoltio
mA	Miliamperio
Kb	Kilobits
Mb	Megabits
V	Voltios
MHz	Megahercios
MB	Megabytes
KB	Kilobytes
μ A	Microamperios
Kohm	Kiloohmios
μ F	Microfaradios
Wh	Watts por hora
mAh	Miliamperios por hora
KWh	Kilovatios por hora

ÍNDICE DE FIGURAS

Figura 2.1. Diagrama de bloques del Particle Photon	5
Figura 2.2. Definiciones de pines del Particle Photon	6
Figura 2.3. Ubicación de parches del Particle Photon	7
Figura 2.4. Colores presentados a través del LED	8
Figura 3.1. Esquema de uso del sistema	9
Figura 3.2. Esquema de broker y tópicos	10
Figura 3.3. Primer Prototipo (Photon, 3 LEDs, Botón).....	16
Figura 3.4. Diagrama de estados general	17
Figura 3.5. Diagrama de estados del microcontrolador	18
Figura 3.6 Diagrama secuencial mostrando la interacción entre usuarios con microcontrolador y aplicación móvil.....	19
Figura 3.7. Diseño esquemático del PCB del sistema resaltando los pines del Photon, LEDs y botón de encendido	21
Figura 3.8. Diseño de impresión del PCB del sistema	21
Figura 3.9. Impresión 3D de la caja para el microcontrolador y el PCB	212
Figura 3.10 El visitante pulsa el botón para que alerte a los empleados.....	22
Figura 3.11 El empleado usa la aplicación móvil para escanear el código de la puerta que desea suscribirse.	23
Figura 3.12 Al recibir la notificación en la aplicación móvil, los empleados pueden ignorar o atender la notificación	24
Figura 3.13 Imagen que muestra la notificación en el dispositivo móvil describiendo la puerta y las opciones a elegir.....	25
Figura 4.1. Microcontroladores armados en protoboard. El de la derecha con soporte de batería	26
Figura 4.2. Colocación de prototipo del sistema en la puerta del laboratorio de prototipado ubicado en el CTI para las pruebas de la aplicación	27
Figura 4.3. Resultados de la prueba pedido y cancelación durante 1 hora	30
Figura 4.4. Resultados de la prueba pedido y aceptación durante 1 hora	31

ÍNDICE DE TABLAS

Tabla 2.1. Significados de estados del LED del Photon [15].....	9
Tabla 3.1. Posibles códigos de estado respuesta	24
Tabla 3.2. Comparación de tecnologías QR, RFID y NFC.	26
Tabla 3.3. Historia de usuario visitante	35
Tabla 3.5. Historia de empleado respuesta a notificación	37
Tabla 4.1. Resultados de la prueba de tiempos de encendido del dispositivo	41
Tabla 4.2. Mediciones de la prueba pedido y cancelación durante 1 hora	42
Tabla 4.3. Resultados de la prueba pedido y cancelación durante 1 hora	43
Tabla 4.4. Consumo de batería	45
Tabla 4.5. Relación de número de cargas de un dispositivo en comparación con el consumo en dólares	46
Tabla 4.6. Relación de cantidad de dispositivos en carga continua por consumo anual en dólares	46

CAPÍTULO 1

1 INTRODUCCIÓN.

1.1 Antecedentes

Usualmente en edificios con seguridad interna los visitantes no tienen acceso por puertas con acceso restringido y tienen que esperar a que la persona a la que visitan les abra la puerta. Esto causa molestias al visitante y usualmente se resuelve usando un timbre el cual al ser presionado emite un ruido que puede perturbar a todos los usuarios del edificio.

1.2 Descripción del problema

Es necesario implementar un portero o timbre inteligente capaz de alertar al personal de trabajo encargado de algún área específica del edificio que hay una persona esperando ser atendida en una puerta que cuenta con seguridad. Esta alerta debe de ser lo más discreta posible. El portero debe de implementar controladores usados en tecnologías IoT, de manera que a través de un protocolo de transmisión de datos pueda alertar al personal, además debe contar con un módulo de batería recargable, y que sea una solución que permita ahorrar energía a largo plazo.

1.3 Objetivo General

Desarrollo e implementación de un timbre inteligente y su aplicación acompañante, la cual sea capaz de notificar a los usuarios de un edificio cuando un visitante les está esperando.

1.4 Objetivos Específicos

- Emplear tecnologías de IoT para desarrollar soluciones innovadoras de bajo costo y consumo energético.

- Implementar herramientas de desarrollo multiplataforma para generar aplicaciones para dispositivos móviles.

1.5 Componentes de la solución

Controladores. - Los Photon [2] son dispositivos microcontroladores capaces de ser reprogramables y conectarse a internet, manejados en lenguaje y colaboración Open source, al igual que los Arduinos, usan la transmisión serial de datos, lo que los hace compatibles con la mayoría de lenguajes de programación utilizados en la actualidad. El protocolo admitido para redes inalámbricas es 802.11 b/g/n, usa una memoria RAM de 128 Kb y 1 Mb de Flash para almacenar código reprogramable.

Batería recargable. - Uno de los requerimientos del sistema es que cuente con baterías recargables, el laboratorio de Prototipado Rápido del Centro de Tecnologías de Información, espacio que se nos ha facilitado para realizar las pruebas de desarrollo, cuenta con diferentes tipos de baterías recargables de Li-ion para el sistema descrito, dado que es una aplicación de IoT, debido a los métodos empleados con los controladores, su fuente de alimentación compatible con entradas USB, voltajes de 3.3 V a 5 V y su capacidad de extremo bajo consumo, hay un ahorro significativo comparado a un sistema conectado a la corriente eléctrica. Otro requerimiento es informar a los usuarios el nivel de carga de la batería [4].

Aplicación móvil. - Las aplicaciones móviles orientadas a notificar y hacer acciones simples a alto nivel para el usuario nos permiten simplificar el uso de los controles de las puertas, por lo tanto, el sistema debe contar con el desarrollo de una app, en Android y iOS que permita suscribirse a través de códigos QR a una puerta en la cual el usuario desea recibir una alerta para notificar que alguien está pidiendo que se abra.

Servidor. - Para controlar el manejo de alertas y suscripciones se propone usar la variante del tradicional modelo de cliente-servidor, y usar publish-subscribe a través de un servidor que emplee el protocolo MQTT [5].

Manejo de Retroalimentación del Dispositivo. - Para la retroalimentación se ha pedido que el dispositivo pueda informar al usuario el estado en el que se encuentra su llamada, para ello tendremos diferentes estados del requerimiento, por lo que se deben usar LEDs que indiquen el estado del requerimiento pendiente, aceptado o ignorado.

1.6 Concepto del problema

Para definir el concepto actual que resume los requerimientos en el listado de ideas proporcionados por el tutor, tenemos tres procesos importantes para el sensor de control de acceso: suscripción, llamada, respuesta (aceptación, pendiente y rechazo), y un proceso de diagnóstico, los cuatro procesos serán descritos a continuación en un lenguaje de alto nivel:

1. Suscripción. - Los usuarios que deseen hacerse cargo de atender una o varias puertas, debe suscribirse a través de un código QR que estará ubicado al lado de la puerta, a través de la aplicación móvil del sistema, empleando la cámara. Su MAC (Identificador único del dispositivo) es almacenada en el servidor, de tal forma que quede registrado en esa puerta en específico. Así será notificado cada vez que alguien necesite entrar y toque el timbre. Es necesario que el suscriptor se encuentre en el edificio, para esto se verifica a través de las redes Wi-Fi que el dispositivo está escuchando (término que se refiere a las redes inalámbricas que se encuentran en el rango de alcance), de no estar en el rango, la alerta no es mostrada en el dispositivo móvil del usuario.

2. Llamada. - Cuando una persona quiere tener acceso a través de una puerta que cuenta con seguridad, procede a accionar el botón del dispositivo que envía un mensaje al servidor. El servidor envía una alerta a las direcciones IP con acceso WIFI del edificio, para que sean enviadas a las MACs suscritas con la puerta que fue pulsada. De esta forma sólo reciben alertas los usuarios que se encuentren en las redes Wi-Fi que el servidor contiene registradas o que al menos los dispositivos móviles se encuentren dentro del rango donde son detectadas por el dispositivo móvil.

3. Respuesta. - Cuando el usuario es alertado por la aplicación tiene dos opciones: aceptar o ignorar la alerta. El dispositivo inteligente tiene que retroalimentar a la persona que está esperando en la puerta:
 - i. Aceptación. - Respuesta atendida por el usuario: El dispositivo debe dar a conocer a la persona que su alerta fue atendida por un usuario, para esto se debe dar un mensaje de aprobación, a través de LEDs (color azul o verde).
 - ii. Pendiente. - Espera de Respuesta: El dispositivo debe dar a conocer a la persona que su alerta está esperando ser atendida por un usuario, para

esto se debe dar un mensaje que indique que hay un tiempo de espera, a través de LEDs (color amarillo).

- iii. Ignorar. - Respuesta no atendida por ningún usuario: El dispositivo debe dar a conocer a la persona que su alerta no fue atendida por usuario alguno, para esto se debe dar un mensaje de ignorado, a través de LEDs (color rojo).

4. Diagnóstico. - Cuando el personal de trabajo requiera el estado actual de la batería, el dispositivo inteligente muestra a través de un LED, si la batería aún está con carga o si es necesario reemplazarla (color verde a color rojo en una escala de un LED RGB).

CAPÍTULO 2

2. PHOTON PARTICLE.

Como herramienta que se utilizará de IoT, tenemos al Particle Photon [6], cuyas prestaciones descritas a continuación cumplen con el uso requerido:

2.1 Características del Photon

- Módulo de Wi-Fi de partículas PØ
- Chip Wi-Fi Broadcom BCM43362
- Wi-Fi 802.11 b / g / n
- STM32F205RGY6 120 Mhz ARM Cortex M3
- 1 MB flash, 128 KB RAM
- LED de estado RGB incorporado, capaz de controlar LED externos.
- 18 GPIO de señal mixta.
- Diseño programable a través de código abierto (Open Source)
- Sistema operativo en tiempo real.

2.2 Diagrama de bloques

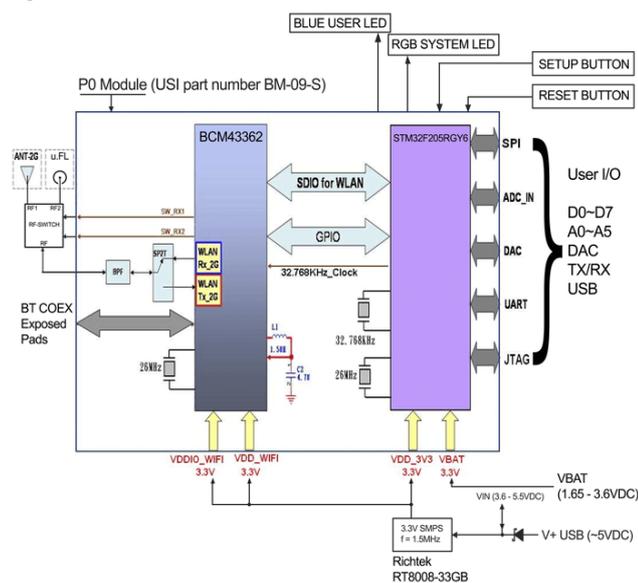


Figura 2.1. Diagrama de bloques del Particle Photon [6]

2.3 Consideraciones eléctricas

Se puede alimentar a través de un conector Micro-USB B a través del pin de V in. Regulado entre 3.6 VCD y 5.5 VCD. Cuando se alimenta a través del puerto USB, V in emitirá un voltaje de aproximadamente de 4.8 VCD. Cuando se usa como salida, la carga máxima en VIN es 1A. 3V3 también se puede usar como salida, pero tiene una sobrecarga de 100 mA disponible.

El consumo de corriente promedio típico es de 80 mA con 5 V @ VIN in con el sensor Wi-Fi activo. La corriente de reposo es típicamente de 80 µA. Es importante usar un cable de calidad para minimizar las caídas (corriente x resistencia = voltaje). La fuente de alimentación debe ser suficiente para generar 1 A de corriente para proporcionar una cantidad adecuada de sobrecarga de corriente.

Cuando se enciende el Photon desde cables largos en USB y VIN, se debe tener cuidado para proteger contra variaciones de voltaje dañinos. Para evitar picos de voltaje, se sugiere mantener el cableado de entrada lo más corto posible. Si los cables largos son inevitables, se recomienda agregar un diodo zener de 5,1 V o un dispositivo de supresión de transitorios similar de VIN a GND.

2.4 Definiciones de pines

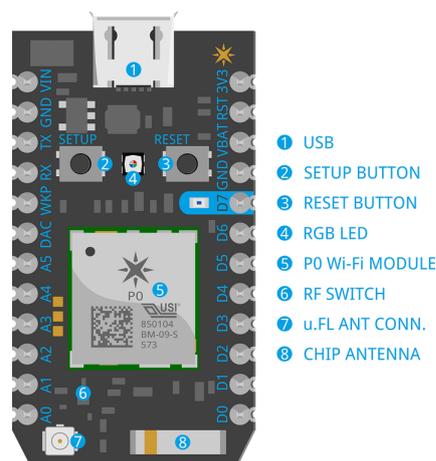


Figura 2.2. Definiciones de pines del Particle Photon [6]

- VIN: Este pin puede ser empleado tanto como entrada como salida. Como entrada puede proveer entre 3.6 y 5.5 VCD al Photon. Cuando el Photon es cargado vía USB, este pin provee voltaje de salida de aproximadamente 4.8 VCD, con una carga máxima de 1A.

- RST: Entrada reset. Contiene un resistor entre RST y 3.3 V de tipo pull-up de 1 Kohm, además de un capacitor de 0.1 μ F entre RST y GND.
- VBAT: Suministro al RTC interno, al SRAM y a los registros de respaldo cuando no hay 3.3 V presentes (1.65 a 3.6 VCD).
- 3V3: Este pin es la salida del regulador interno y se encuentra conectado al VDD del módulo Wi-Fi. Cuando se suministra de poder al Photon vía VIN o a través del puerto USB, este pin provee una salida de 3.3 VCD. Este pin puede ser usado además para alimentar al Photon de forma directa (entrada máxima de 3.3 VCD). Cuando se usa como salida, la carga máxima es de 100 mA.
- RX: Usado generalmente como UART RX, pero puede ser empleado como GPIO o PWM digital.
- TX Usado generalmente como UART TX, pero puede ser empleado como GPIO o PWM digital.
- WKP: Pin de reanimación, despierta al dispositivo del estado de espera/dormido. Este pin puede ser empleado como GPIO o ADC. Se la llama A7 cuando es empleada como ADC.
- DAC: Salida análoga a digital de 12 bits. Puede ser empleado como GPIO.
- A0~A7: Entradas análogas a digitales. A6 y A7 son representados como DAC y WKP respectivamente.
- D0~D7: Pins de entrada digital solamente. D0~D3 pueden ser empleados como salida de tipo PWM.

El dispositivo soporta el uso simultáneo de Bluetooth y otras señales de radio externas a través de tres parches de oro cercanos al pin A3. Estos parches son de 9 por 9 mm, distanciados entre sí a 12 mm.

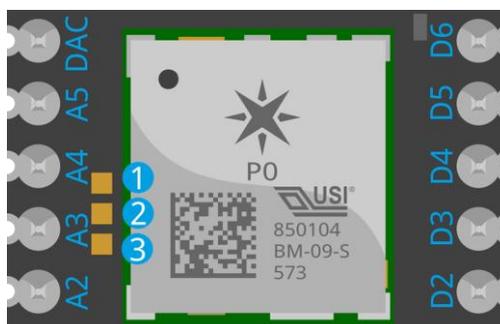


Figura 2.3. Ubicación de parches del Particle Photon [6]

Cuando se tienen dos señales de radio usando las mismas frecuencias de banda para el sistema, por ejemplo, Wi-Fi y Bluetooth, una interfaz puede ser usada para coordinar la actividad de transmisión, de tal forma se asegura el mejor rendimiento reduciendo la interferencia y conflictos de las dos señales. Los parches pueden ser usados para esta tarea.

2.5 Códigos de colores del Photon

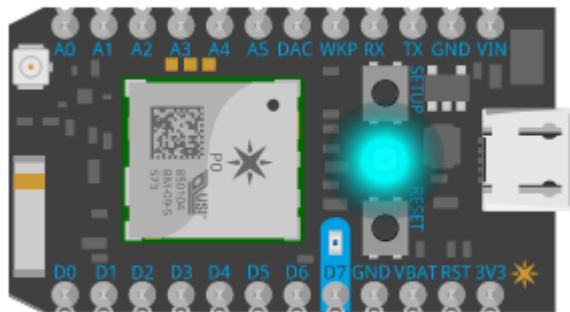


Figura 2.4. Colores presentados a través del LED [15]

El Photon presenta un LED entre los botones “Setup” y “Reset” que indican el estado actual del microcontrolador a través de colores y pulsaciones cada uno con su significado [15].

CAPÍTULO 3

3. DISEÑO DE LA SOLUCIÓN

3.1 Esquema de la solución

Para el desarrollo de la solución se ha elaborado el siguiente esquema de uso:

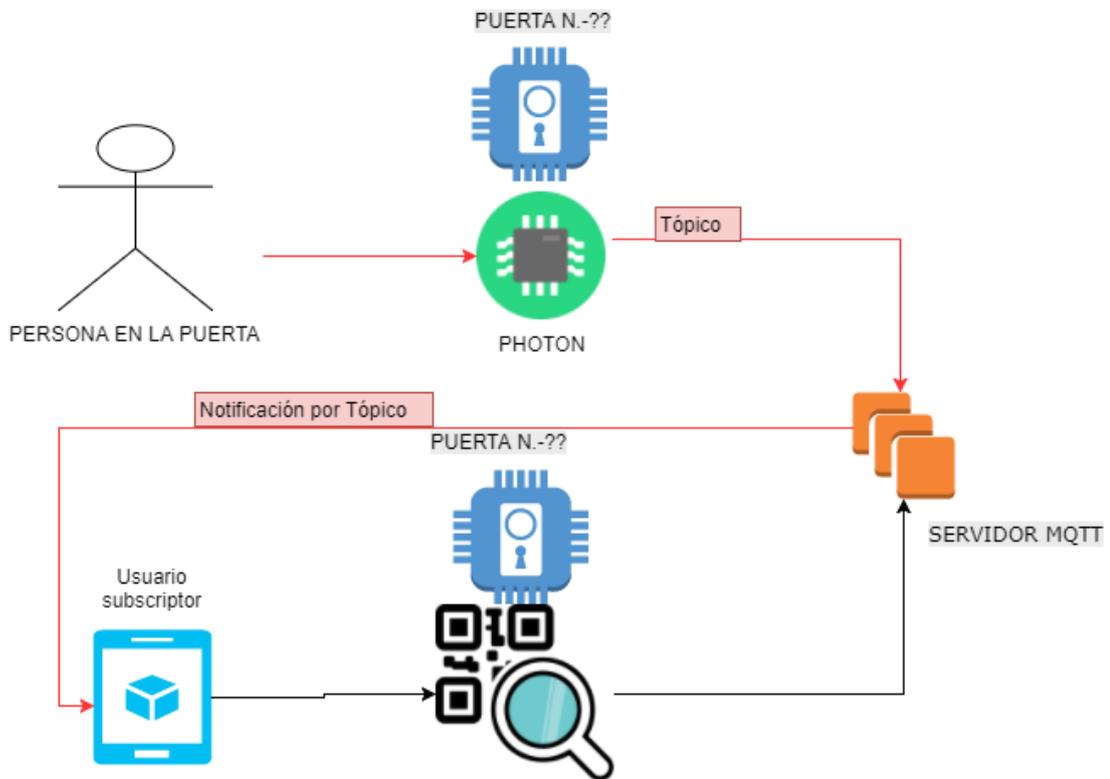


Figura 3.1. Esquema de uso del sistema

Los componentes del sistema, mostrados en la figura 3.1 son un servidor, la aplicación y un microcontrolador (por puerta).

El servidor funciona como conexión entre todos los dispositivos, a través de conceptos MQTT como *brokers* y *tópicos*. La conexión entre un dispositivo móvil y el microcontrolador es representada por un tópico. Un dispositivo móvil puede estar suscrito a más de un tópico al mismo tiempo.

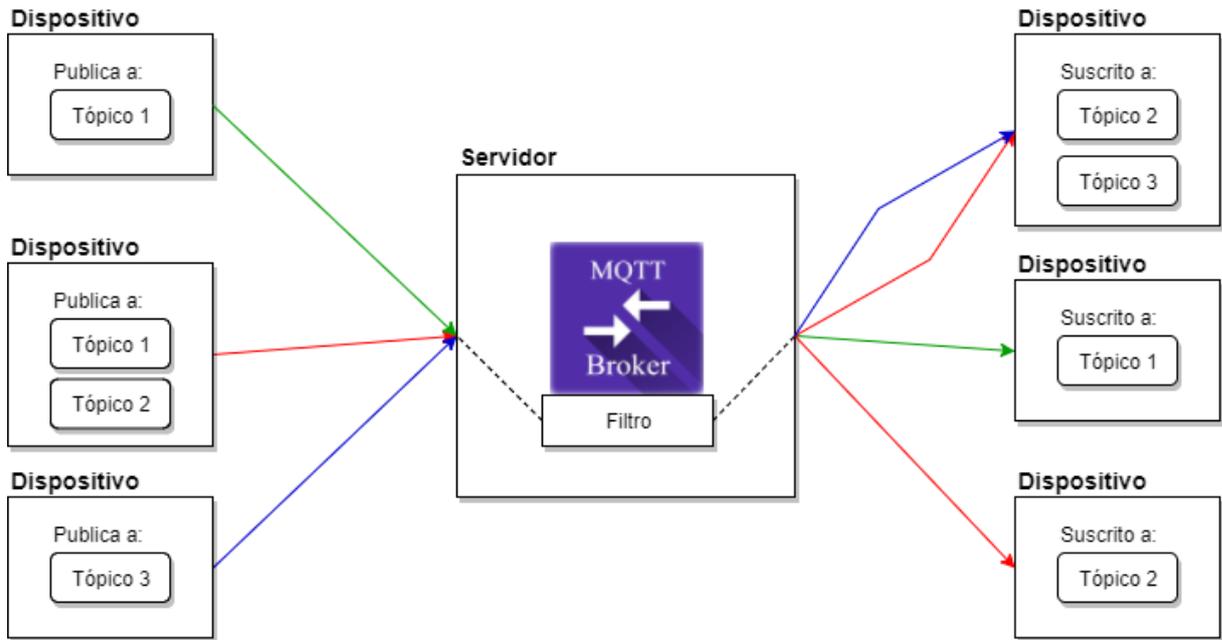


Figura 3.2. Esquema de *broker* y tópicos

Todos los mensajes publicados por algún dispositivo pasan a través del *broker* representado dentro del servidor de acuerdo a la figura 3.2, y desde allí son dirigidos a los dispositivos suscritos. Como elemento en común está el tópico. Un tópico es una cadena de caracteres que el *broker* emplea como filtros para los mensajes y enviarlos a los clientes correctos. Una explicación más extensa del *broker* se encuentra en la siguiente página.

Para el desarrollo de la aplicación móvil se usará Ionic Framework, que es una herramienta que permite desarrollo tanto en Android como en IOS al emplear librerías que pueden ser ejecutadas para ambos sistemas operativos.

El microcontrolador emplea el lenguaje de programación C++, además de las librerías para manejo de los protocolos MQTT y SSL.

3.2 Tecnologías de desarrollo utilizadas

3.2.1 Ionic Framework

Ionic es un framework open source que facilita la creación de aplicaciones multiplataforma, lo que significa que no tienen necesidad de volver a ser escritas en diferente código, se programa en HTML5 y JS, y al tener soporte para librerías proporcionadas por la comunidad puede

brindar control sobre hardware y software del dispositivo móvil Android o iOS, junto con los debidos protocolos para conexión a Internet [7].

3.2.2 AngularJS

Angular se integra con Ionic Framework para proporcionar vistas a elementos dinámicos y estáticos que se usan en las aplicaciones, a través de Lenguaje Javascript [8]. La razón por la cual fue elegido, es por su integración con servidores y protocolos para paso de mensajes y objetos, a través de JavaScript manejado por promesas (Representa un valor que puede estar disponible en el presente, futuro o nunca, es usado para computaciones asíncronas) que garantiza que la aplicación móvil sea asíncrona y fácil de codificar, debido a que se puede depurar en tiempo real, para analizar las respuestas obtenidas del servidor, de esta forma se garantiza que si la promesa extrae una respuesta, siempre volverá a obtenerla, además se pueden programar manejos de errores para evitar que la aplicación quede “congelada” (es decir, que se quede en un fallo en el cual no se pueda realizar ninguna acción debido a que no se puede navegar en ella), se cierre o quede en un loop (repetición).

3.2.3 MQTT

MQTT o *MQ Telemetry Transport* es un protocolo de tipo publish/subscribe, sencillo y ligero diseñado para equipos que trabajan con redes de alta latencia y poca disponibilidad. Emplea pocos recursos de ancho de banda e intenta proveer confiabilidad a la red, con un grado de aseguramiento que dependerá del nivel de seguridad que el usuario desee otorgarle para la llegada de mensajes [9]. Estos principios lo hacen ideal para las tecnologías emergentes como M2M e IoT que tienen limitantes en ancho de banda y poder de batería.

3.2.4 Clientes y *Brokers*

En terminología MQTT, un *broker* es la implementación del protocolo, que tiene las siguientes tareas:

- Recibe todos los mensajes.
- Filtra los mensajes por tópico.
- Envía los mensajes a los clientes suscritos.
- Mantiene las sesiones de clientes conectados.
- Autentica y autoriza a los clientes.

Los clientes son dispositivos que cuentan con alguna librería que implementa el protocolo MQTT y se conecta a un *broker* a través de la red. Un cliente sólo tiene la capacidad de publicar, suscribirse o ambos al mismo tiempo. Mayoritariamente, los clientes, son los dispositivos con restricciones de recursos (memoria, procesamiento, uso del ancho de banda, etc.), debido a que no pueden realizar las operaciones descritas anteriormente por el *broker*.

Los clientes no se conectan entre sí, las conexiones siempre son entre clientes y *brokers*. La conexión es inicializada a través de un mensaje especial CONNECT desde el cliente al *broker*. El *broker* envía una respuesta CONNACK y un código de estado. Según el código de estado se establece la conexión o se la rechaza. Una vez que la conexión es establecida, el *broker* la mantendrá hasta que el cliente se lo pida o pierda la conexión con este [10].

Cualquier cliente puede abandonar la conexión con el *broker* en cualquier momento que desee. Puede enviar un mensaje especial conocido como last will (testamento) al abandonar la conexión, y de esta forma avisar a los demás clientes que se retira de la red. Esto es importante si otros clientes dependen del que abandona para realizar sus tareas.

3.2.5 Websockets

WebSockets es un protocolo para la comunicación entre clientes y servidores, y provee una comunicación de tipo full-duplex sobre una sola conexión TCP. Permite la transferencia de datos en tiempo real desde y hacia el servidor. Para el proyecto, el servidor será un browser generado por Ionic, a través del browser se podrán manejar las interacciones con las herramientas como lector de códigos QR y manejo de suscripciones a las puertas.

3.2.6 Código QR

Un código QR, o *Quick Response Code*, es un código de barras de dos dimensiones, capaz de almacenar información de tipo alfanumérica, numérica, binaria y en kanji [11]. Un código QR puede almacenar desde varias decenas hasta varios cientos de veces la cantidad de información que un código de barras.

	Código QR	RFID	NFC
--	------------------	-------------	------------

Costo	Gratuito	Desde \$0.07 (no metales, pasivo). Desde \$1.00 (metales, pasivo) Desde \$15 (activo).	Desde \$0.07
Compatibilidad con dispositivos móviles	Compatible con dispositivos móviles con cámaras. Mínimo de 3 megapíxeles. En algunos casos necesita una aplicación para leer.	Requiere lector.	Requiere lector. Dispositivos Android desde la versión 4.0 poseen aplicación que emplea NFC para transferencia de fotos, contactos y direcciones. Apple, desde iPhone 6.
Rango	Depende del tamaño, nivel de detalle de la imagen y la iluminación sobre la misma.	Pasivos: máximo hasta 24 m. Activos: máximo hasta 450 m.	Inferior a 20 cm.
Desventajas	Puede ser difícil de leer, depende de la intensidad de luz. Imposible de leer con daños superiores al 30%.	Interferencia entre señales de dispositivos. Interferencia con metales y agua.	Imposible de leer si el chip es dañado (caso pasivo) o si no posee energía (caso activo).

Tabla 3.2. Comparación de tecnologías QR, RFID y NFC

QR no es la única solución posible para el manejo de suscripciones a las puertas. Existen diferentes posibles tecnologías como RFID (*Radio-Frequency Identification*) y NFC (*Near-Field Communication*), las cuales son comparadas contra QR en la tabla 1.

QR es la tecnología que se usará como parte de la solución al problema, ya que como primer punto a favor no tiene costos asociados. Otra ventaja es que no es necesario el uso de energía para alimentar el código generado, lo que no ocurre con chips activos RFID o NFC.

El tamaño del código QR depende de los siguientes factores:

- Distancia entre el dispositivo capturador y el código. Esto determina el detalle capturado por la cámara.
- Densidad, o nivel de detalle del código. Mientras más datos se empleen en la imagen, mayor el nivel de detalle.

El tamaño mínimo de un código QR es de 2 cm, tamaños inferiores a ese disminuye la legibilidad en el escaneo realizado a través de la cámara en la mayoría de los dispositivos [12].

3.2.7 Particle Web IDE

Con Particle Web IDE (WEB Integrated Development Environment), conocido como Particle Build, se puede programar un módulo controlador a través de la conexión a redes inalámbricas, así como también actualizar código, firmware, depurar programas para prototipado rápido en dispositivos IOT, de manera rápida y sin instalar un software en el computador, solo se necesita conexión internet para el procedimiento de configuración, es usado para flashear el código que se ejecuta en el firmware del microcontrolador IOT a través de internet, de forma que no necesita ser removido de su posición, ni de su configuración de Hardware, ésto brinda una ventaja cuando el dispositivo se encuentra en estado comercial y no pueda desarmarse.

3.3 Librerías y plugins usados

3.3.1 Cordova Camera plugin

Este plugin permite tener control de la cámara del dispositivo móvil (Android o iOS) [13], sea para tomar fotos, usar algoritmos de reconocimiento de imagen, grabar videos, incluir realidad aumentada, etc. Se usará este plugin para capturar los códigos QR y extraer la información que se encuentra codificada en el código analizado, de esa forma se obtiene la suscripción a una puerta del edificio.

3.3.2 Mosquitto

Mosquitto es una librería que implementa las versiones 3.1 y 3.1.1 del protocolo MQTT, el cual es un método ligero en cantidad de contenido de paquetes, en comparación a protocolos como HTTP, para paso de mensajes [14]. Debido a las capacidades y limitaciones que el microcontrolador posee, como la memoria RAM, almacenamiento interno, módulos de conexión inalámbrica y número de pines; se ha elegido este protocolo para el envío y recepción de mensajes.

El protocolo MQTT permite métodos de encriptación a través de la red como SSL. SSL incrementa considerablemente la cabecera de los mensajes a través de la red, incrementando el tamaño total del paquete en el paso de mensajes.

3.4 Prototipo

El primer prototipo del sistema se describe a continuación, tiene los siguientes componentes:

- 1 Particle Photon, el microcontrolador.
- 3 LEDs para indicar el estado del pedido: rojo, verde y amarillo.
- 1 botón para enviar el aviso al servidor.
- 4 resistencias en total, 3 (220 Ohms) para los LEDs y 1 (10 Kohms) para el botón.
- 7 cables macho-macho.

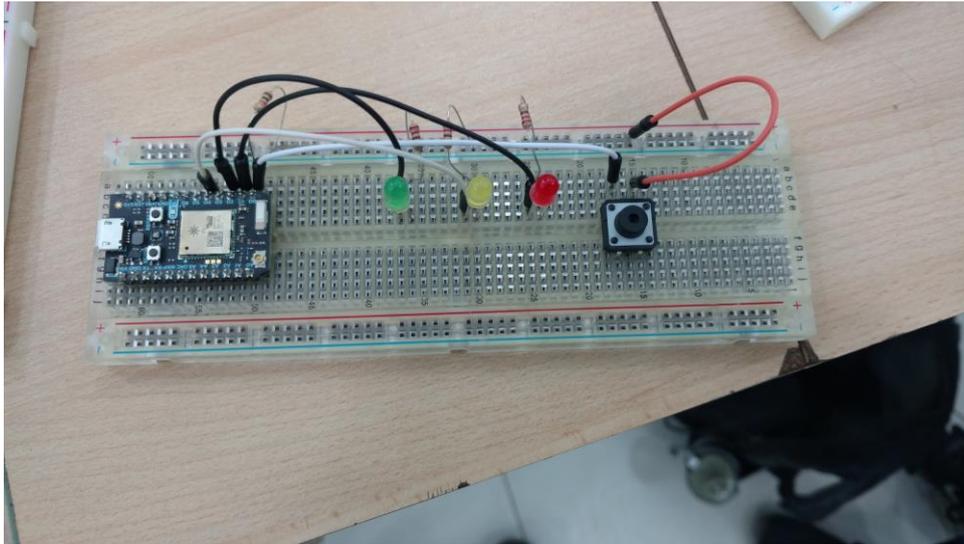


Figura 3.3. Primer Prototipo (Photon, 3 LEDs, Botón)

3.5 Descripción del sistema con diagramas de estados

Para realizar la conexión a través de protocolo MQTT desde la aplicación móvil, es necesario el uso de WebSockets. Se usa la librería `mqttjs.js` que recibe como parámetros la dirección y puerto del WebSocket donde se encuentra alojado el servidor *broker*, luego se elige un canal al cual se suscribirá y uno para la publicación de mensajes.

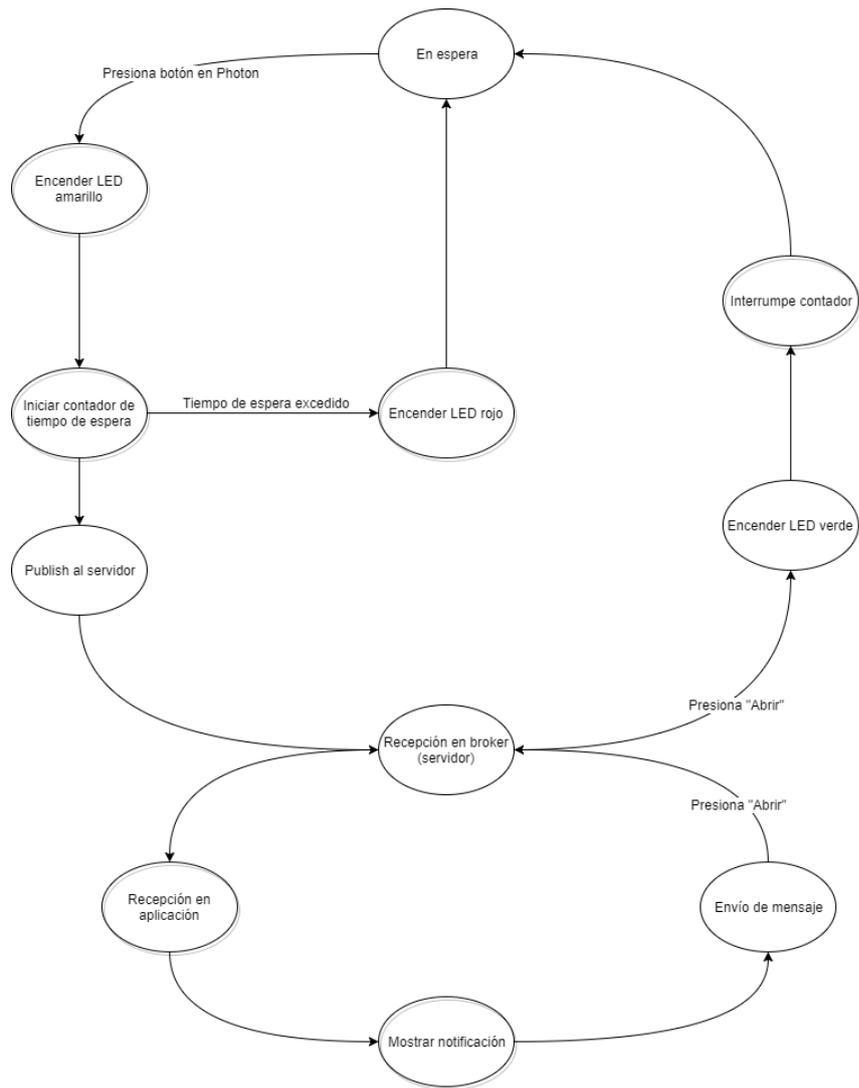


Figura 3.4. Diagrama de estados general

La figura 3.3 muestra el diagrama de estados de la solución en general. El tiempo de espera al cual se refiere el estado “Iniciar contador de tiempo de espera” se calculará realizando pruebas con trabajadores del CTI.

Una vez llegado el mensaje desde el microcontrolador a la aplicación en el dispositivo móvil, se genera una notificación con dos botones en la barra de notificaciones (Android o iOS). El primer botón indica al usuario que hay alguien esperando en la puerta y permite aceptar la notificación, si alguien va a ir a abrirla, enciende el LED verde e interrumpe el tiempo de espera y la notificación es eliminada en los otros dispositivos móviles donde haya sido recibida. El segundo botón de la notificación permite eliminar la notificación en el dispositivo móvil, esto no interrumpe el tiempo de espera.

La figura 4.2 muestra la interacción en tiempo entre usuarios, tanto visitante como usuario con aplicación móvil. El recuadro *Alt: Respuesta del usuario con aplicación* muestra las opciones que se tienen al recibir la notificación en la aplicación móvil. Estos son los tiempos que maneja el microcontrolador:

- Un minuto de espera desde presionar el botón (inicio de parpadeo de LED amarillo) hasta que el requerimiento sea cancelado (se enciende el LED rojo).
- Un minuto desde que se recibe el mensaje de que alguien va a abrir la puerta, en ese tiempo el LED verde va a parpadear (cancela el tiempo de espera para cancelación, apaga el LED amarillo).
- Luego de cinco minutos de inactividad, el microcontrolador se apagará. Se se encenderá al presionar el botón. Una vez conectado a la red y a la nube de Particle, iniciará el contador de tiempo para cancelación de requerimiento, el parpadeo del LED amarillo y publicará al servidor el mensaje “Botón presionado”.

Para evitar la llegada de muchos mensajes desde el microcontrolador a la aplicación móvil, se ha deshabilitado el envío desde que se presiona el botón hasta que ocurra una de dos cosas: o termina el tiempo de espera (máximo un minuto sin atender el pedido), o acaba el parpadeo del LED verde luego de un minuto de aceptar ir a abrir la puerta (máximo dos minutos de espera si se cuenta con el tiempo anterior).

3.7 Elaboración del PCB e Impresión 3D

Para el esquema de elaboración del PCB (Printed Circuit Board) se utilizó el programa Eagle de AutoDesk, que permite elaborar el diseño y exportar a un formato de impresión, para ser impreso en la ProtoMat Plotter S63.

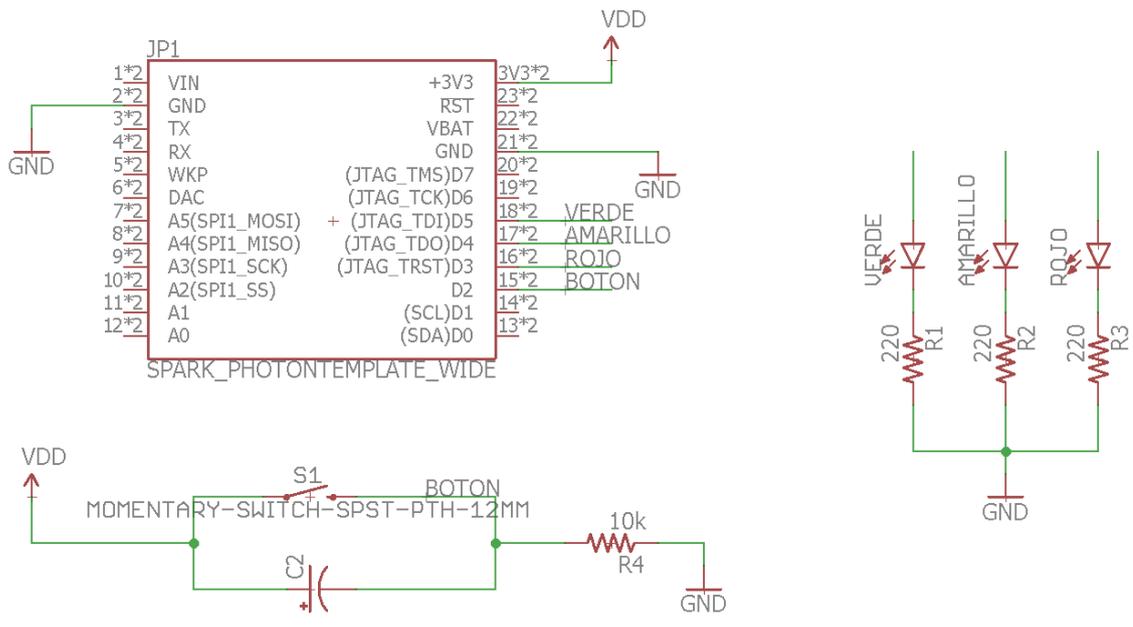


Figura 3.7. Diseño esquemático del PCB del sistema resaltando los pines del Photon, LEDs y botón de encendido

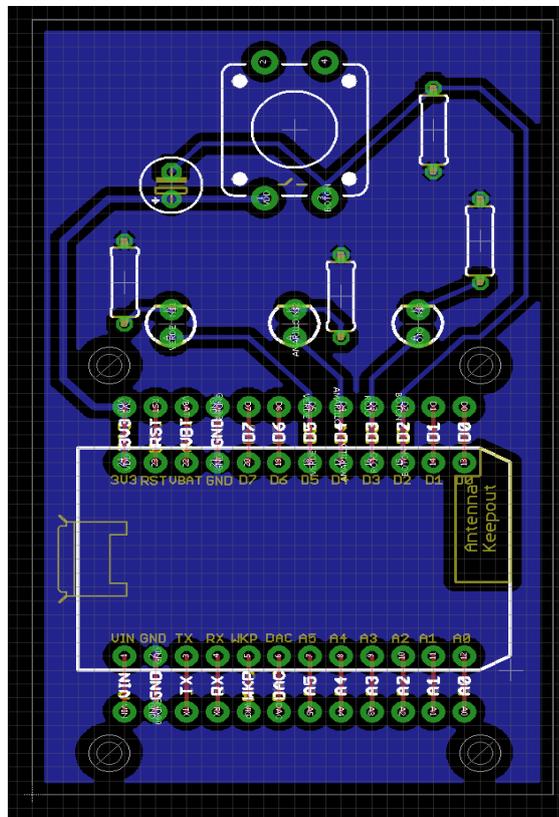


Figura 3.8. Diseño de impresión del PCB del sistema

Para el diseño de la caja se usó el programa Sketch-Up 2017 de Trimble.



Figura 3.9 Impresión 3D de la caja para el microcontrolador y el PCB.

3.8 Descripción del sistema con historias de usuarios

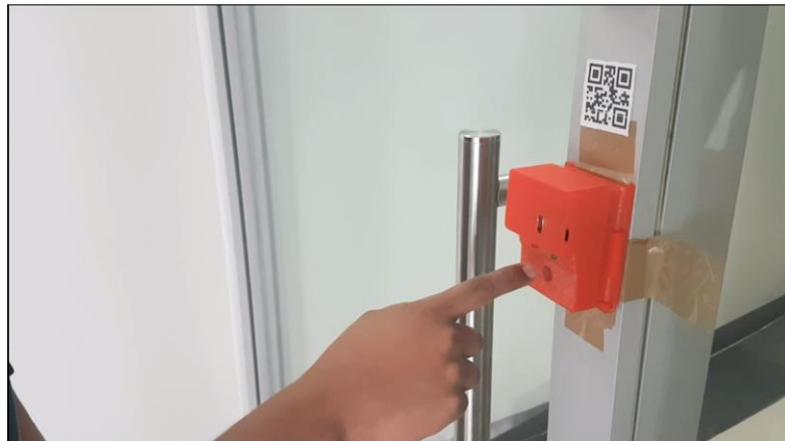


Figura 3.10 El visitante pulsa el botón para que alerte a los empleados

Historia de usuario	
Número: 1	Nombre: Visitante presiona botón
Usuario: Visitante	
Precondiciones:	
<ul style="list-style-type: none"> • Servidor <i>broker</i> debe de estar funcionando. 	
Descripción: Una vez que llega a la puerta, el visitante puede presionar el botón del microcontrolador colocado cerca de la puerta. En ese momento, el LED amarillo empezará a parpadear durante un minuto entero. Se envía la petición desde el microcontrolador al	

broker, el cual lo redirigirá a todos los dispositivos móviles suscritos a dicha puerta. Presionar el botón durante el proceso, no realiza acción alguna.

Observaciones:

- Si el requerimiento es atendido, el LED amarillo se apagará y empezará a parpadear el LED verde, indicando que algún empleado está comprometido a ir y abrirle la puerta. Luego de un minuto de parpadeo, el LED verde también se apagará, luego de esto, un nuevo visitante podrá presionar el botón para iniciar un nuevo requerimiento
- Si nadie atiende el requerimiento, ya sea ignorando la notificación o presionando el botón “Ignorar” de la notificación, el LED amarillo terminará su periodo de parpadeo e inmediatamente el LED rojo se encenderá durante 5 segundos. Luego de esto, el visitante podrá presionar el botón otra vez para iniciar un nuevo requerimiento.

Tabla 3.3. Historia de usuario visitante

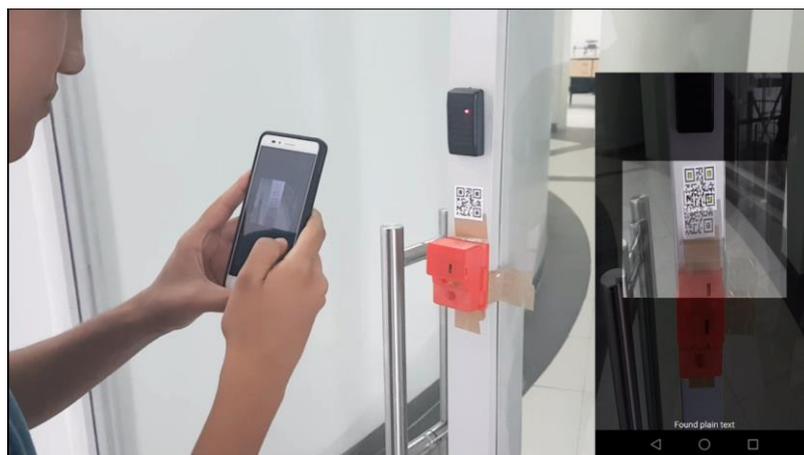


Figura 3.11 El empleado usa la aplicación móvil para escanear el código de la puerta que desea suscribirse.

Historia de usuario	
Número: 2	Nombre: Suscripción a la puerta
Usuario: Empleado	
Precondiciones: <ul style="list-style-type: none"> • La aplicación móvil debe de ser descargada e instalada en el dispositivo móvil. • Servidor <i>broker</i> debe de estar funcionando. 	
Descripción: El empleado puede suscribirse a una puerta, a través del código QR asociado. La ventana inicial de la aplicación presenta el mensaje que el empleado no tiene asociada puerta alguna, con un enlace a la sección indicando “suscribirse a una puerta”. Esto lo	

redirige a la sección en la cual podrá presionar el botón para escanear un código QR. Una vez escaneado, debe de regresar a la ventana principal y actualizar la lista de puertas asociadas, una vez realizado, podrá recibir notificaciones desde dicha puerta.

Observaciones:

- El empleado puede suscribirse a una puerta seleccionando el botón “Suscribirse a una puerta” en la parte superior izquierda de la aplicación.
- En caso de no encontrar códigos QR para escanear, se puede cancelar la operación.
- Es necesario actualizar la lista de puertas suscritas, caso contrario no se reciben notificaciones.

Tabla 3.4. Historia de empleado suscripción a puerta



Figura 3.12 Al recibir la notificación en la aplicación móvil, los empleados pueden ignorar o atender la notificación

Historia de usuario	
Número: 3	Nombre: Respuesta a notificación
Usuario: Empleado	
Precondiciones: <ul style="list-style-type: none"> • Empleado debe de estar suscrito a la puerta descrita, a través de la aplicación móvil. • Servidor <i>broker</i> debe de estar funcionando. • La aplicación no ha sido descartada de memoria. 	
Descripción: Una vez presionado el botón manejado por el microcontrolador, éste publica un mensaje hacia el <i>broker</i> , el cual lo redirigirá a todos los dispositivos móviles suscritos a dicha puerta. Una vez que la notificación llega al dispositivo móvil con la aplicación instalada, el empleado tiene cuatro opciones:	

<ol style="list-style-type: none"> 1. Presiona el botón “Iré a abrir”, y se debe de comprometer a ir a abrir la puerta que generó la notificación. 2. Presiona el botón “Ignorar”, en este caso, la notificación desaparecerá. 3. Ignorar la notificación, la cual desaparecerá luego de un minuto. 4. Presionar la notificación: en este caso, la aplicación pasará a primer plano, mostrando el mensaje “Visitante esperado en puerta #” mostrando dos botones “Iré a abrir” e “Ignorar” con las mismas operaciones descritas en los puntos 1 y 2 respectivamente.
<p>Observaciones:</p> <ul style="list-style-type: none"> • Que un empleado presione el botón “Iré a abrir”, tanto del punto 1 como del punto 4, ocasiona que el microcontrolador envíe un mensaje de tipo ACK hacia los dispositivos suscritos para que eliminen la notificación, así se minimiza el conflicto con otros empleados intentando responder positivamente al requerimiento. • Cuando el tiempo de espera termina del lado del microcontrolador, se envía un mensaje de tipo <i>request time out</i> hacia los dispositivos suscritos, para que la notificación sea eliminada. De esta forma los empleados no pueden atender requerimientos anteriores. • La eliminación de la notificación es por puerta, es decir, la puerta 1 elimina sólo las notificaciones que generó. Notificaciones de otras puertas no se verán afectadas.

Tabla 3.5. Historia de empleado respuesta a notificación

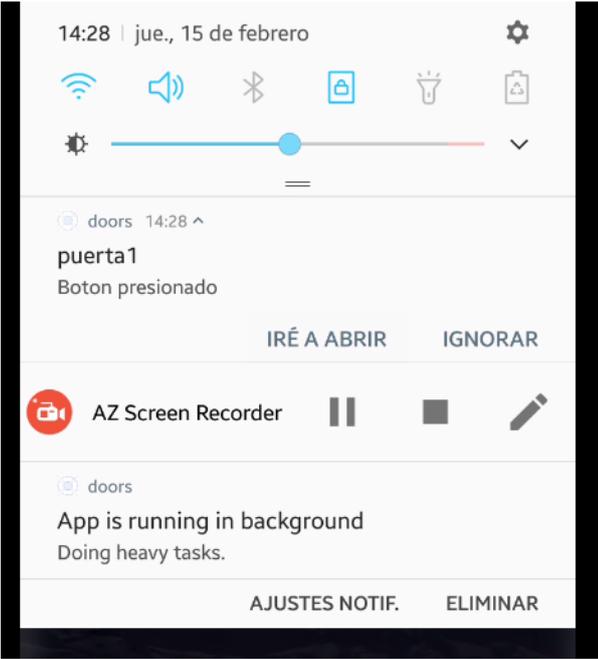


Figura 3.13 Imagen que muestra la notificación en el dispositivo móvil describiendo la puerta y las opciones a elegir.

CAPÍTULO 4

4. DETALLES DE LOS RESULTADOS

4.1 Resultados de la primera prueba

Luego de la primera prueba realizada con el microcontrolador, se llegó a la conclusión de deshabilitar la publicación del mensaje “Botón presionado” cada vez que se presiona el botón y reemplazar esta acción por requerimiento, es decir, el mensaje sólo se publica al presionar el botón y luego entra en estado de espera, en el que no hay interacción sino hasta que termina el tiempo de espera o se decide aceptar e ir a abrir desde la aplicación móvil.

Otro detalle por corregir es la eliminación de notificaciones en los dispositivos móviles suscritos a una puerta apenas uno de los suscritos decide aceptar la notificación e ir a abrir, de esta forma se previene que los demás respondan una notificación que ya ha sido aceptada. Se requiere eliminar la notificación de todos los dispositivos apenas expire el tiempo de espera por los mismos motivos.

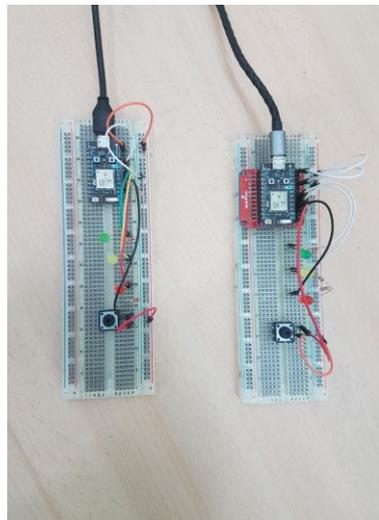


Figura 4.1. Microcontroladores armados en protoboard. El de la derecha con soporte de batería

La figura 4.1 muestra dos microcontroladores armados, uno con soporte de batería de tipo litio-polímero o ión-litio desde 3.7 a máximo 4.2 voltios nominales de una sola célula.

La opción ignorar debe de bloquear la llegada de mensajes nuevos de una puerta específica, durante un periodo de tiempo determinado por el usuario del dispositivo móvil, en caso de que no desee ser molestado por las notificaciones de esa puerta.

4.2 Resultados de la segunda prueba

Luego de la segunda prueba se tomó la información del dispositivo que llama a la notificación, de esta forma se puede tener una identificación para cada notificación y así eliminarla cuando nadie atiende la puerta, luego del tiempo de espera, o cuando algún dispositivo con la aplicación la atiende, en ese caso debe de desaparecer para los demás.



Figura 4.2. Colocación de prototipo del sistema en la puerta del laboratorio de prototipado ubicado en el CTI para las pruebas de la aplicación

4.3 Resultados de pruebas de rendimiento

Para las pruebas de rendimiento, un factor importante es el tiempo que el dispositivo demora en encender al ser pulsado por el botón, este proceso involucra encender el microcontrolador, establecer conexión con Wi-Fi, y al cloud de Particle, para proceder a enviar la notificación y esperar una respuesta para la activación de los LEDs.

Para esto se generaron dos series de pruebas, la diferencia entre ambas es una corrección que se realizó para asegurar la conexión al Wi-Fi, la segunda serie considera esto último. Se tomó el tiempo desde el encendido hasta que el dispositivo complete su conexión con el cloud Particle, lo cual se puede ver, según la documentación del Photon, a través del LED interno que toma un color cian. Los tiempos de las pruebas se muestran en la tabla 4.3:

Prueba	Tiempo (en segundos)	Tiempo 2 (en segundos)
1	10.13	9.53
2	7.91	7.81
3	14.02	7.99
4	14.82	8.93
5	11.25	7.92
6	13.68	18.87
7	8.26	10.06
8	14.01	14.5
9	9.67	11.76
10	9.42	9.81
11	8	8.03
12	9.71	12.69
13	14.16	14.98
14	14.6	15.24
15	10.49	12.62
16	13.83	9.09
17	9.4	11.04
18	14.8	13.48
19	11.16	8.03
20	7.78	9.69
21	14.66	7.95
22	13.4	13.95
23	9.13	14.52
24	8.09	9.88
25	9.11	13.78
26	8.46	12.8
27	No conexión	10.7
28	27.92	9.29
29	9.51	11.78
30	19	9.97
Promedio	11.94	11.223

Tabla 4.1. Resultados de la prueba de tiempos de encendido del dispositivo

Los resultados el encendido del microcontrolador en promedio mejora un poco menos de un segundo en el encendido, pero de esta forma se garantiza la conexión, al no presentarse instancias en las que no pierde conexión con la nube Photon. Otro factor importante es el de la duración de la batería, se consideran 2 estados generales del microcontrolador:

- Procesando un pedido.
- En modo de ahorro.

Se define procesar un pedido como al conjunto de eventos desde que un visitante presiona el botón hasta que termine el LED verde de parpadear o que se encienda el LED rojo de cancelación.

Se han diseñado dos pruebas para el pedido, ambas durante un periodo de una hora: en la primera, se realiza el pedido (parpadeo de LED amarillo), seguido de su cancelación (encendido de LED rojo). En la segunda, se realiza el pedido (parpadeo de LED amarillo), seguido de su aceptación (parpadeo de LED verde).

Antes de empezar las pruebas, se toman los primeros 15 datos (5 minutos) del nivel de batería, al finalizar las pruebas se realiza lo mismo para promediar los datos. La tabla 4.4 muestra los resultados obtenidos de la primera prueba, la tabla 4.5, de la segunda.

Medición	Inicio	Fin
1	84.485	74.055
2	83.848	74.711
3	83.848	74.055
4	83.848	73.398
5	83.195	72.742
6	83.195	73.398
7	83.195	74.055
8	83.848	72.742
9	83.195	74.711
10	83.195	74.711
11	83.195	72.742
12	83.195	72.742
13	83.520	73.672
14	83.195	73.640
15	83.385	73.551
Promedio	83.490	73.662

Tabla 4.2. Mediciones de la prueba pedido y cancelación durante 1 hora

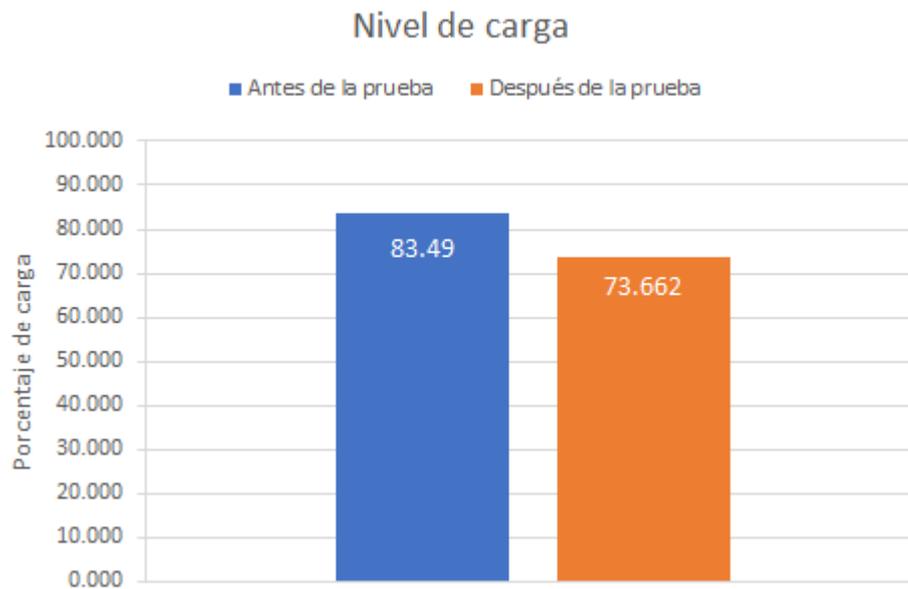


Figura 4.3. Resultados de la prueba pedido y cancelación durante 1 hora

La figura 4.3 compara los promedios de nivel de la carga de batería medidos antes y luego de las pruebas. Existe un gasto de 11.771% de carga durante 1 hora. Calculando, la batería duraría un aproximado de 8 horas con 29 minutos.

Medición	Inicio	Fin
1	39.383	33.410
2	39.383	33.410
3	40.824	33.410
4	40.824	33.410
5	39.383	33.410
6	40.824	33.410
7	42.414	31.691
8	40.824	31.691
9	42.414	31.691
10	39.383	33.410
11	40.824	31.691
12	40.824	33.410
13	40.824	33.410
14	42.414	31.691
15	42.414	33.410
Promedio	40.864	32.837

Tabla 4.3. Resultados de la prueba pedido y aceptación durante 1 hora

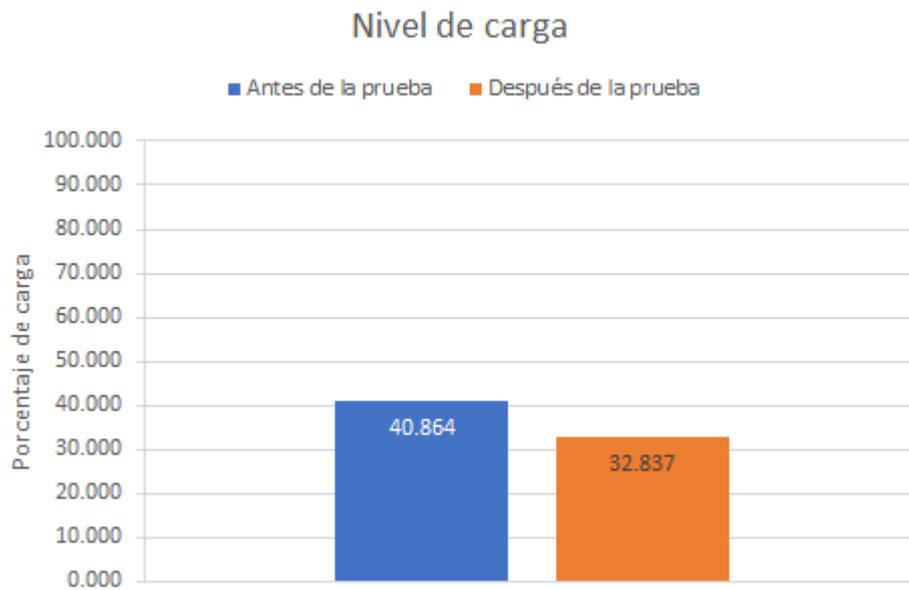


Figura 4.4. Resultados de la prueba pedido y aceptación durante 1 hora

La figura 4.4 compara los promedios de nivel de la carga de batería medidos antes y luego de las pruebas. Existe un gasto de 19.642% de carga durante 1 hora. Calculando, la batería duraría un aproximado de 5 horas con 5 minutos.

Según las especificaciones del microcontrolador (capítulo 2, sección 3), la corriente de reposo es típicamente de 80 μA . Para medir la corriente del sistema en modo ahorro, se ha conectado una resistencia de 1.2 Ohms en serie, medido el voltaje en paralelo a esa resistencia con un osciloscopio y luego calculado la corriente dando 79-80 μA como medición.

4.4 Ahorro del dispositivo

Para poder determinar cuánto ahorra a nivel de consumo eléctrico y gasto económico, se necesita contrastar con un timbre casero que cumpla con la misma función pero que se encuentre conectado a la red eléctrica, para este ejemplo en particular, un timbre de pared genérico de 10W/h consume mensualmente la cantidad de 1.11 Dólares [16].

Si el dispositivo se encuentra sin uso, consumiendo solo 80 μA de corriente de reposo, calculamos el tiempo dividiendo la carga eléctrica de la batería, en nuestro ejemplo es 2000mAh, para la corriente de reposo, da como resultado 25000 horas que se resumen en 1042 días aproximadamente, lo que significa 34 meses y medio para que la batería necesite ser cargada, es decir que sin uso el microcontrolador dura casi 3 años (2,85 años exactamente) con la misma batería.

Al analizar el sistema propuesto a través del consumo de porcentaje de energía de la batería se obtuvieron los siguientes resultados:

Batería	2000
Porcentaje de consumo por uso	0.77280
Usos hasta agotarse (Días)	129.3995859
Usos por Día en un mes	4.313319531

Tabla 4.4. Consumo de batería

La tabla 4.4 se interpreta de esta forma, en una batería de 2000mAh, si se desea conocer el consumo de un visitante por día, se calcula cuánto porcentaje consume la corriente de reposo por día, es decir 80 μ A en un día, este valor es 0.00192 μ Ah representado en porcentaje de la batería en 2000mAh, es de 0.096% de la batería. Con estos valores podemos recalculamos la tabla anterior asumiendo un uso por día.

El porcentaje de consumo de la batería por cada uso tiene un aproximado de 0,68% más el consumo de reposo que es 0.096% por día, éstos resultados nos permiten determinar la cantidad de usos que se le puede dar al sistema antes que la batería se agote, cuyo valor es aprox. 129 veces, lo que nos indica que duraría 129 días a un uso por día, éste valor puede ser reducido a nivel mensual, dándonos una cantidad cerca de 4 visitantes al día que usen el sistema, para que su recarga sea mensual, siguiendo la convención de que es una batería de 2000 mAh, podemos realizar una conversión a Wh, dado que el voltaje medido es 3.7 V, la multiplicación de estos valores nos da 7.4 Wh que es el consumo del sistema.

Trabajando con dólares, en Ecuador la tarifa actual de Febrero, 2018 es 0,10 centavos de dólar/KWh, de forma que al multiplicar estos valores tenemos que el consumo mensual del dispositivo es 0,00074 centavos de dólar, a partir de ese precio se puede extender a la siguiente tabla:

Número de Cargas	Consumo (Centavos de dólar)
1	0.00074
2	0.00148
3	0.00222
4	0.00296
5	0.0037
6	0.00444
7	0.00518
8	0.00592
9	0.00666
10	0.0074
20	0.0148
30	0.0222
50	0.037
100	0.074
1000	0.74
1500	1.11

Tabla 4.5. Relación de número de cargas de un dispositivo en comparación con el consumo en dólares

Se observa entonces que el punto de igualación de consumo mensual entre el timbre eléctrico genérico y el sistema propuesto es cuando realizamos 1500 cargas en el mes, dado que es una cantidad absolutamente aberrante, se establece como referencia que se cargue 30 veces por mes al dispositivo, una vez cada día, a parte que se obtienen 147 usos por día, el consumo sólo llegaría a ser dos centavos al mes, lo que equivale al 0,018% del consumo del timbre genérico, si se contempla a un periodo anual, sosteniendo en carga continua a más de un dispositivo:

Número de Dispositivos	Consumo Anual (Centavos de dólar)
1	0.2701
2	0.5402
3	0.8103
4	1.0804
5	1.3505
6	1.6206
7	1.8907
8	2.1608
9	2.4309
10	2.701
20	5.402
30	8.103
50	13.505
100	27.01

Tabla 4.6. Relación de cantidad de dispositivos en carga continúa por consumo anual en dólares

En la tabla encontramos la relación que existe entre el dispositivo con su consumo anual en dinero, si se instalan 4 dispositivos en un edificio, anualmente estando en carga continua (es decir conectados a la red eléctrica directamente) consumen menos energía de la que consume un timbre genérico en tan solo un mes.

CONCLUSIONES Y RECOMENDACIONES

El uso de tecnologías IoT promueve nuevas soluciones y reinenciones de bajo consumo, si son complementadas con herramientas de desarrollo web y móvil llegan a tener un valor de uso importante debido al amplio campo de aplicaciones donde pueden ser usados, en nuestro caso, dispositivos de prototipo que son usados para alertar en tiempo real a un dispositivo móvil, que es el centro de esta investigación y desarrollo, demostrando que a un bajo costo de materiales dedicados a investigación se puede lograr un alcance comercial del producto.

Recomendamos leer las especificaciones completas de los dispositivos IoT que se utilicen en un proyecto, debido a limitaciones de hardware que puedan tener, y verificar si el dispositivo cumple con los requerimientos del objetivo del proyecto.

Se concluye que el sistema en el cual el dispositivo es usado a nivel de consumo y ahorro de energía es una buena alternativa comparada a los timbres genéricos comercializados actualmente.

Como trabajo futuro queda planteada la posibilidad de instalar una cámara de baja resolución (VGA) que capture una foto cuando el botón del dispositivo IoT es pulsado, de manera que la notificación recibida alerte en los dispositivos móviles adjuntando una foto, de esa forma, los usuarios suscritos a la puerta pueden identificar a la(s) persona(s) que está(n) esperando ser atendida(s).

BIBLIOGRAFÍA

- [1] S. Gupta, N. Mudgal, and R. Mehta, "Analytical study of IoT as emerging need of the modern era," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 233–235.
- [2] Particle. (2016). *Particle Photon (Wi-Fi)*. [En línea]. Disponible en: <https://www.particle.io/products/hardware/photon-wifi-dev-kit>
- [3] M. Jubelag. (2015). *SketchUp for 3D printing: a tutorial for beginners*. [En línea]. Disponible en: <https://all3dp.com/sketchup-3d-printing-tutorial/>
- [4] Particle docs. (2012, Ene. 30). *Particle: Connect your Internet of Things (IoT) devices*. [En línea]. Disponible en: <https://docs.particle.io/reference/firmware/electron/#fuelgauge>
- [5] HiveMQ. (2015, Oct. 13). *MQTT Essentials part 2: Publish & Subscribe*. [En línea] Disponible en: <https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe>
- [6] Particle Photon Datasheets. (2016). *Particle Photon (Wi-Fi)*. [En línea]. Disponible en: [https://docs.particle.io/datasheets/photon-\(wifi\)/photon-datasheet/](https://docs.particle.io/datasheets/photon-(wifi)/photon-datasheet/)
- [7] Ionic (2013, Oct. 2). *Ionic framework*. [En línea]. Disponible en: <https://ionicframework.com>
- [8] Google. (2010, Feb. 15). *AngularJS*. [En línea]. Disponible en: <https://angularjs.org/>
- [9] MQTT. (2017, Nov. 17). *MQTT Frequently Asked Questions*. [En línea]. Disponible en: <http://mqtt.org/faq>
- [10] HiveMQ. (2015, Oct. 13). *MQTT essentials: part 1 - Introducing MQTT*. [En línea]. Disponible en: <https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment>
- [11] Denso Wave. (2017, Nov. 18). *What is a QR code?* [En línea]. Disponible en: <http://www.qrcode.com/en/about/>
- [12] Tap Walk. (2017, Sep. 24). *Minimum practical size of a QR Code*. [En línea]. Disponible en : <http://tapwalk.com/minimum-practical-size-of-a-qr-code/>
- [11] Phonegap. (2012, Abr. 16). *Phonegap API Documentation*. [En línea]. Disponible en: http://docs.phonegap.com/en/edge/cordova_camera_camera.md.html
- [13] Eclipse. (2006, Jun. 15). *Mosquitto MQTT*. [En línea]. Disponible en: <https://mosquitto.org/>
- [14] MQTT. (2017). *MQTT Quality of Service* [En línea]. Disponible en: <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels>

- [15] Particle Photon device modes. (2016). *Particle Photon*. [En línea]. Disponible en: <https://docs.particle.io/guide/getting-started/modes/photon/>
- [16] Electrocalculator. (2014) *¿Cuánto gasta un aparato eléctrico? ¿Cuánta energía consume?* [En línea]. Disponible en: <http://www.electrocalculator.com/>