

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingenieros de Sistemas**

**TEMA:
DESARROLLO DE UN SISTEMA QUE DETERMINE EL ESTADO DE UN
PARQUEADERO (LIBRE, OCUPADO) DENTRO DE LA UNIVERSIDAD
POLITÉCNICA SALESIANA, UTILIZANDO BALIZAS Y GEOLOCALIZACIÓN.**

**AUTORES:
JUNIOR VINICIO RAMOS SUAREZ
RODRIGO ALEJANDRO VACA MACIAS**

**TUTOR:
ROBINSON DIMITRI LLERENA PAZ**

Quito, febrero del 2020

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Junior Vinicio Ramos Suarez con documento de identificación N° 2100606504 y Rodrigo Alejandro Vaca Macias, con documento de identificación N° 1721783676, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “DESARROLLO DE UN SISTEMA QUE DETERMINE EL ESTADO DE UN PARQUEADERO (LIBRE, OCUPADO) DENTRO DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, UTILIZANDO BALIZAS Y GEOLOCALIZACIÓN.”, mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
Ramos Suarez
Junior Vinicio
CI: 2100606504



.....
Vaca Macias
Rodrigo Alejandro
CI: 1721783676

Quito, febrero del 2020

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “DESARROLLO DE UN SISTEMA QUE DETERMINE EL ESTADO DE UN PARQUEADERO (LIBRE, OCUPADO) DENTRO DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, UTILIZANDO BALIZAS Y GEOLOCALIZACIÓN.”, realizado por Junior Vinicio Ramos Suarez y Rodrigo Alejandro Vaca Macias, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, febrero del 2020

A handwritten signature in blue ink, appearing to read 'Robinson Dimitri Llerena Paz', with a large, stylized flourish above the name.

Robinson Dimitri Llerena Paz.
CI: 1707106942

Dedicatoria

“Esta tesis está dedicada a mis dos pilares fundamentales, mis padres, Vinicio Ramos y Mirian Suarez quien con su amor y esfuerzo me han inculcado buenos valores para poder cumplir mis metas, gracias a mi madre por tenerme paciencia y por su apoyo en todas mis mocedades; gracias a mi padre por su dureza y consejos que me supieron guiar por un buen camino a lo largo de mi vida.

Gracias a Dios por darme la vida y por sus bendiciones además de todas las buenas oportunidades que me ha brindado, por rodearme de personas que me ayudan y fortalecen cada día más. Una mención especial a mi tutor Robinson Llerena que fue mi guía en el desarrollo de esta tesis. Estoy muy feliz por mi nuevo triunfo, feliz por todas las personas que contribuyeron con su pequeño grano de arena y que confiaron en que podía realizar mi tesis.”

Junior Ramos

“El presente proyecto de titulación lo dedico a mis Padres que son las personas que me han apoyado toda mi vida inculcándome valores y principios, ellos me han dado la fuerza desde el principio de mi carrea para poder luchar todos los días y nunca rendirme hasta alcanzar la meta y haber logrado culminar mis estudios.

Lo dedico a mi esposa Gabriela Reyes quien ha sido un pilar fundamental en mi vida y quien me ha dado fuerzas, ánimo y siempre ha estado ayudándome a salir adelante y no darme por vencido, es una persona vital en mi vida y gracias a su ayuda pude continuar con mi carrera y culminarla de manera satisfactoria.”

Rodrigo Vaca

ÍNDICE

| | |
|---|----|
| INTRODUCCIÓN | 1 |
| Antecedentes | 1 |
| Planteamiento del Problema..... | 3 |
| Justificación de la Investigación | 3 |
| Objetivos de la investigación | 5 |
| Objetivo General | 5 |
| Objetivos Específicos..... | 5 |
| Metodología Aplicada..... | 6 |
| Capítulo 1 | 10 |
| Marco Conceptual | 10 |
| 1. Estudio del hardware y software | 10 |
| 1.1. Estudio del hardware | 10 |
| 1.1.1. Sensores..... | 10 |
| 1.1.1.1. <i>Sensor Ultrasónico HC-SR04</i> | 10 |
| 1.1.1.2. <i>Sensor de fin de carrera</i> | 11 |
| 1.1.1.3. <i>Sensor Inductivo</i> | 11 |
| 1.1.1.4. <i>Sensor Capacitivo</i> | 12 |
| 1.1.1.5. <i>Criterio de selección del sensor</i> | 13 |
| 1.1.2. Controladores | 14 |
| 1.1.2.1. <i>Tarjeta de desarrollo Arduino 2560</i> | 14 |
| 1.1.2.2. <i>Raspberry pi B+</i> | 14 |
| 1.1.2.3. <i>Microcontrolador PIC</i> | 16 |
| 1.1.2.4. <i>Criterios de selección del controlador</i> | 17 |
| 1.1.3. Módulos Bluetooth | 18 |
| 1.1.3.1. <i>HC-05</i> | 18 |
| 1.1.3.2. <i>HC-06</i> | 18 |
| 1.1.3.3. <i>Raspberry pi 3 modulo bluetooth</i> | 19 |
| 1.1.3.3.1. <i>Sistema Operativo de la tarjeta Raspberry pi 3.</i> | 20 |
| 1.1.3.4. <i>Criterios de selección del módulo bluetooth</i> | 21 |
| 1.1.4. Baterías..... | 21 |
| 1.1.4.1. <i>LiPo</i> | 22 |
| 1.1.4.2. <i>LiOn</i> | 23 |
| 1.1.4.3. <i>Fuente de alimentación</i> | 23 |

| | |
|---|----|
| 1.1.4.4. Criterios de selección fuente de alimentación..... | 24 |
| 1.1.5. Balizas FeasyBeacon..... | 24 |
| 1.1.5.1. Principios de funcionamiento..... | 25 |
| 1.1.5.2. Uso práctico..... | 25 |
| 1.2. Estudio del software..... | 26 |
| 1.2.1. HTML5..... | 26 |
| 1.2.2. Python..... | 27 |
| 1.2.3. Android Studio..... | 27 |
| 1.2.4. Sensor GPS neo 6m..... | 29 |
| Capítulo 2..... | 30 |
| Diseño del prototipo..... | 30 |
| 2. Funcionamiento de la aplicación FeasyBeacon..... | 30 |
| 2.1. Instalación Sistema Raspbian en Raspberry Pi..... | 32 |
| 2.2. Esquema de funcionamiento..... | 34 |
| 2.2.1. Opción 1: Servidor Firebase..... | 36 |
| 2.2.2. Opción 2: Servidor Apache..... | 37 |
| 2.3. Diagrama de flujo del sensor..... | 38 |
| 2.4. Diagrama de flujo GPS..... | 39 |
| 2.5. Diagrama de casos de uso UML..... | 40 |
| Capítulo 3..... | 41 |
| Desarrollo de la aplicación y análisis de resultados..... | 41 |
| 3. Creación del servidor Apache..... | 41 |
| 3.1. Diseño e implementación del circuito..... | 45 |
| 3.1.1. Fritzing..... | 47 |
| 3.2. Implementación del circuito..... | 47 |
| 3.3. Prueba de medición del sensor..... | 48 |
| 3.4. Interfaz inicial de la Aplicación web (Esquema funcional)..... | 49 |
| 3.4.1. Página Principal..... | 49 |
| 3.4.2. Información Solicitada..... | 50 |
| 3.4.3. Footer..... | 50 |
| 3.5. Pruebas..... | 51 |
| 3.5.1. Escenarios de pruebas del prototipo..... | 51 |
| 3.5.2. Escenarios de pruebas de la baliza..... | 55 |
| 3.6. Resultados y análisis..... | 58 |
| CONCLUSIONES..... | 62 |

| | |
|----------------------------|----|
| RECOMENDACIONES | 63 |
| GLOSARIO..... | 64 |
| LISTA DE REFERENCIAS | 66 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Planificación del proyecto | 7 |
| Tabla 2. Características de los sensores de proximidad | 13 |
| Tabla 3. Características Raspberry pi3 B+ | 15 |
| Tabla 4. Controladores y sus características | 17 |
| Tabla 5. Módulos bluetooth y sus características | 21 |
| Tabla 6. Características batería de polímero de litio | 22 |
| Tabla 7. Características de las baterías | 24 |
| Tabla 8. Ventajas y desventajas HTML | 26 |
| Tabla 9. Características Python | 27 |
| Tabla 10. Características de Android Studio | 28 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Gráfica porcentaje y fechas de avance del proyecto..... | 8 |
| Figura 2. Sensor ultrasónico para medir distancia. | 11 |
| Figura 3. Representación del sensor final de carrera. | 11 |
| Figura 4. Representación del sensor inductivo..... | 12 |
| Figura 5. Representación del sensor capacitivo. | 12 |
| Figura 6. Dispositivo Arduino mega 2560..... | 14 |
| Figura 7. Raspberry pi3 B+..... | 16 |
| Figura 8. Modulo bluetooth HC-05..... | 18 |
| Figura 9. Módulo HC-06..... | 19 |
| Figura 10. Puertos de conexión raspberry pi3 B+..... | 20 |
| Figura 11. Sistema operativo Noobs y Raspbian. | 20 |
| Figura 12. Baliza, dispositivo para emitir datos a través de bluetooth..... | 25 |
| Figura 13. Interfaz Android Studio. | 28 |
| Figura 14. Modulo GPS NEO 6M..... | 29 |
| Figura 15. Interfaz de la aplicación FeasyBeacon..... | 30 |
| Figura 16. Interfaz para cambiar parámetros físicos y de conexión..... | 31 |
| Figura 17. ISO sistema operativo raspbian. | 32 |
| Figura 18. Despliegue de la instalación de la ISO de raspbian. | 33 |
| Figura 19. Instalación de la ISO en el micro sd con el programa Rufus..... | 33 |
| Figura 20. Esquema de funcionamiento con el servidor firebase. | 36 |
| Figura 21. Esquema de funcionamiento con el servidor apache. | 37 |
| Figura 22. Diagrama de flujo del funcionamiento del sensor. | 38 |
| Figura 23. Diagrama de flujo del funcionamiento del GPS. | 39 |
| Figura 24. Diagrama de casos de uso para el usuario. | 40 |
| Figura 25. Actualización del sistema operativo raspbian..... | 41 |
| Figura 26. Actualización completa del sistema operativo raspbian. | 42 |
| Figura 27. Instalación de apache en la raspberry pi. | 42 |
| Figura 28. Servidor reiniciado para guardar los cambios..... | 43 |
| Figura 29. Instalación de la librería GPIO. | 44 |
| Figura 30. Instalación de la librería GPIO para pines de entrada y salida. | 45 |
| Figura 31. Diagrama esquemático del circuito en el programa fritzing..... | 46 |
| Figura 32. Implementación del circuito en el protoboard y la raspberry. | 47 |
| Figura 33. Prueba del sensor censando un determinado objeto. | 48 |
| Figura 34. Medición del sensor ultrasónico. | 49 |
| Figura 35. Interfaz web de la página principal..... | 49 |
| Figura 36. Retorno del resultado arrojado por los sensores. | 50 |
| Figura 37. Footer de la página web..... | 51 |
| Figura 38. Escenario de prueba 1 del prototipo. | 52 |
| Figura 39. Escenario de prueba 2 del prototipo. | 53 |
| Figura 40. Escenario de prueba 2 del prototipo, distancia de 230 cm..... | 53 |
| Figura 41. Escenario de prueba 3 del prototipo. | 54 |
| Figura 42. Escenario de prueba 1 de la baliza..... | 56 |
| Figura 43. Escenario de prueba 2 de la baliza..... | 57 |
| Figura 44. Escenario de prueba 3 de la baliza..... | 58 |

Resumen

En el presente documento se detalla el diseño de un prototipo que permite conocer por medio de una aplicación web, si un parqueadero se encuentra disponible y su ubicación utilizando balizas, con el objetivo de que se agilice la obtención de un parqueadero libre para los vehículos de los diferentes usuarios, dentro de la Universidad Politécnica Salesiana Campus Sur. En el diseño del prototipo se detalla el funcionamiento de la baliza, que tiene su aplicación propietaria denominada FeasyBeacon y el esquema de funcionamiento del sistema en general, el cual abarca el censado de los parqueaderos, su localización y el servidor de alojamiento de la aplicación web. En la etapa de desarrollo se muestra el esquema funcional de la aplicación web, con la finalidad de que el usuario tenga una mejor visualización de la información entregada por el prototipo. Se realizaron pruebas en varios escenarios, que determinaron la ubicación del prototipo y la baliza para no tener interferencias o daños causados por los mismos usuarios. La aplicación móvil facilita el acceso a los usuarios a los parqueaderos y, además es de fácil asimilación y permite conocer el estado del parqueadero y su localización de manera intuitiva.

Abstract

This document details the design of a prototype that allows you to know through a web application, if a parking is available and its location using beacons, with the aim of speeding up the obtaining of a free parking for the vehicles of different users, within the Salesian Polytechnic University Campus South. The design of the prototype details the operation of the beacon, which has its proprietary application called FeasyBeacon and the operating scheme of the system in general, which encompasses the censusing of the parking units, their location and the hosting the web application. The development stage shows the functional schema of the web application, so that the user has a better visualization of the information delivered by the prototype. Tests were performed in several scenarios, which determined the location of the prototype and the beacon so as not to have interference or damage caused by the same users. The mobile application facilitates the access of the parking lots and also is easy assimilation and allows to know the status of the parking and its location in an intuitive way.

INTRODUCCIÓN

Antecedentes

El presente trabajo investigativo tiene como finalidad integrar nuevas tecnologías en un prototipo que determine si un parqueadero se encuentra libre u ocupado por medio del uso de la tecnología de balizas. Con la masiva utilización de dispositivos móviles se desarrolló una aplicación que será utilizada para que el usuario pueda ingresar de manera sencilla al parqueadero disponible, actualmente no se dispone de un aplicativo que brinde información al usuario sobre la disponibilidad de los parqueaderos y su localización dentro de la Universidad Politécnica Salesiana Campus Sur.

El desarrollo de aplicativos de geolocalización junto con las beacons ha tenido una gran acogida en el ámbito de educación, servicios, transporte, comercio, etc. A continuación, se puntualizan algunos casos de éxito de aplicativos de mayor importancia que se han implementado en los ámbitos antes mencionados.

“Geolocalización Inteligente Implementando Tecnología Beacons En Dispositivos Móviles”, la aplicación desarrollada posee una funcionalidad importante la cual es detectar mediante un sensor bluetooth la mayor cantidad de dispositivos que se ubican en el área de la baliza (beacon). La principal funcionalidad de la aplicación es detectar una baliza cercana al rango del bluetooth, al momento de pulsar un botón esta nos redireccionará al sitio Web, para posteriormente desplegar toda la información que el usuario necesita de la baliza, además se puede visualizar la ubicación.

Con la implementación se desea mejorar el ambiente estudiantil cumpliendo con los siguientes objetivos los cuales son: mejorar la comunicación, mejorar el despliegue de información y la gestión universitaria logrando de esta manera un avance importante en los servicios que posee la universidad brindando a los estudiantes y personal administrativo un alto índice de satisfacción (García, Tello, Mata, & Dorantes, 2018, págs. 38-39).

Por consiguiente, el dispositivo de mayor acción en el proyecto es la baliza (beacon) que interactúa con el smartphone ya que estos dos son los que se conectan entre sí para obtener resultados, donde la baliza tiene el objetivo de transmitir y el sistema ejecuta cada una de las instrucciones que se generaron al momento de ejecutar la aplicación, una de sus principales funciones es recibir información y mostrarla en pantalla, las tareas que la aplicación realiza generan más tareas como por ejemplo: recibir información genera otra tarea la cual es mostrar información y la tarea recibir información genera otra tarea, la cual es recibir petición, estas tareas pueden ejecutarse cuando el dispositivo obtiene la información que es enviada por la baliza (García et al., 2018, pág. 40).

Otra aplicación desarrollada con éxito usando la tecnología beacons es: “Desarrollo de Aplicaciones de “In-door Marketing” utilizando Beacons BLE: Un reporte de Experiencia” (Páez & Bedoya, 2017, pág. 1), este caso de éxito nos da a conocer el cómo las beacons nos permiten determinar la posición de un determinado cliente por su rango bluetooth de baja energía que es mayor al bluetooth estándar que conocemos, además permite orientar al cliente hacia los productos y a las diferentes promociones que son de su interés.

Los principales obstáculos que se dieron al desarrollar la aplicación fueron: poder integrar la información de la tienda y la ubicación de cada una de las balizas en la misma tienda, obtener la información de la distancia de una baliza a un dispositivo móvil, ubicar dentro de un rango aceptable las balizas y los dispositivos móviles y enviar notificaciones cuando el usuario detecta una baliza y se acerque a un producto de su interés (Páez & Bedoya, 2017, pág. 1).

Planteamiento del Problema

La Universidad Politécnica Salesiana, se encuentra en constante crecimiento, debido a esto se ha detectado un inconveniente al momento de encontrar un lugar libre de parqueo dando así una inexistencia de información sobre el estado de los parqueaderos (libre u ocupado) y su localización. Un considerable porcentaje de usuarios poseen vehículos lo que ocasiona congestión a la hora de buscar un estacionamiento libre, sobre todo en las horas pico, es decir, en horas de ingreso a clases.

Los problemas que se pudieron determinar fueron: malestar en docentes, estudiantes y personal administrativo debido a la pérdida de tiempo que se tiene al momento de buscar un parqueadero libre dentro de la Universidad Politécnica Salesiana Campus Sur.

De igual manera se pudo determinar que existen personas que al no encontrar un parqueadero libre, optan por estacionarse en los espacios que son exclusivamente destinados para las personas con discapacidad para evitar atrasos. Un gran porcentaje de estudiantes se estacionan fuera de la Universidad con el riesgo de tener multas de tránsito o sufrir robos.

Los guardias que se encuentran a la entrada del parqueadero no poseen conocimiento sobre si existen lugares libres, por tal motivo no tienen un control de si se debe seguir dando paso a los vehículos para entrar a los parqueaderos, por dicha razón los parqueaderos suelen colapsar lo que causa congestión y molestia en los usuarios.

Justificación de la Investigación

Con el uso de la tecnología que hay en el mercado se desarrolló un prototipo para censar el estado de un parqueadero (libre ocupado), además se utilizó la tecnología beacons, para la transmisión de información hacia los dispositivos cercanos. El objetivo de desarrollo del prototipo es para dotar de un servicio que no posee la universidad, el cual es conocer si un

parqueadero está disponible o no y su localización, además de reducir el tiempo de búsqueda de este.

Se desarrollará una aplicación móvil que indique al usuario donde se encuentra libre un estacionamiento mediante el uso de balizas, de esta manera se evitará atrasos y congestión en los parqueaderos. Actualmente el proceso de búsqueda de un parqueadero se lo realiza visualmente recorriendo y buscando alrededor de toda la universidad lo cual causa desventajas ya que no se encuentra fácilmente por lo que se pierde tiempo y genera malestar en los usuarios causando atrasos y congestión en horas pico.

Al desarrollar dicho prototipo se obtendrá un beneficio para sus usuarios al integrar las nuevas tecnologías beacons con los diferentes sensores que existen.

Kappta, sistema reconocido en España y que ha sido un éxito y de los primeros en lograr combinar las beacons con la tecnología BLE con el objetivo de detectar mediante un dispositivo móvil que detecta cuando un usuario se encuentra cerca de un establecimiento en donde se ha colocado una beacon con información de ofertas de productos, permitiéndole a cada uno de los posibles usuarios acceder a la información de la baliza. Además, estos dispositivos son utilizados para el envío de ofertas a varios usuarios que se encuentren dentro del rango de la baliza.

Con la misma idea de Kappta, la Universidad de Oklahoma usa las beacons para enviar información a sus estudiantes, las beacons están ubicados en los sitios más frecuentados por los estudiantes, como la biblioteca, el campus y salas de exposiciones. La aplicación, además, es capaz de identificar el perfil del usuario, por lo que solo le envía notificaciones que son relevantes para él. Los perfiles que se identifican en la aplicación de la Universidad de Oklahoma son: estudiantes, profesores y visitantes (Kappta, 2016).

También uno de los mayores casos de éxito se da en España, “Aeropuertos de Madrid y Barcelona”, implementaron el uso de beacons para enviar información a los pasajeros de manera sencilla, pero para recibir dicha información debe tener instalada la app del aeropuerto y encendida la conexión bluetooth. Así, el pasajero puede conocer cada uno de los datos sobre su vuelo, tiempos en los cuales debe pasar por los controles, documentación necesaria para el embarque, hora del embarque, alertas sobre los parqueaderos donde se encuentra su vehículo en el parqueadero o incluso ofertas de los distintos productos comerciales dentro del aeropuerto (Using Beacons, 2016).

Estos ejemplos muestran el enorme potencial que tienen las beacons por eso se propone desarrollar una aplicación móvil que ayudara a conocer si el parqueadero esta libres y su localización. El desarrollo de dicha aplicación brindara grandes beneficios a los usuarios (estudiantes, personal docente, personal administrativo y sus visitantes) de la Universidad Politécnica Salesiana.

Objetivos de la investigación

Objetivo General

Desarrollar un prototipo que determine si un parqueadero se encuentra disponible y cuál es su ubicación física, dentro de la Universidad Politécnica Salesiana Campus Sur utilizando balizas.

Objetivos Específicos

Análisis de las ventajas que ofrecen las balizas hoy en día como tecnología de bluetooth y aprovechar sus usos.

Examinar las ventajas que ofrecen los diferentes tipos de sensores aplicables a las necesidades del proyecto y aprovechar sus usos.

Diseñar un aplicativo móvil para reducir el tiempo de búsqueda de un parqueadero disponible, en la Universidad Politécnica Salesiana Campus Sur.

Realizar pruebas de sensores y baliza para determinar si el funcionamiento de éstos es óptimo para nuestro proyecto.

Efectuar pruebas del prototipo en el parqueadero de la Universidad y verificar su correcto funcionamiento.

Metodología Aplicada

Para el desarrollo del prototipo se realizó un estudio de los parqueaderos para determinar el óptimo para el proyecto. También, se realizó una planificación para ver los avances continuos del proyecto. La metodología XP tiene como objetivo tener una retroalimentación continúa con el usuario y potenciar las relaciones del equipo de trabajo como clave para llegar al éxito, consta de 4 fases fundamentales que se detallarán a continuación:

Fase de planificación: Para lograr cumplir con esta fase conjuntamente con el docente tutor del proyecto, se realizó una estimación de los tiempos en los cuales se lograría completar las diferentes fases del proyecto, de esta manera se establecieron objetivos y se desarrolló un plan de trabajo donde constan actividades que se realizarán y la duración de cada una de las actividades propuestas.

Como podemos observar en la Tabla 1., se tomaron en cuenta cuatro objetivos principales (entregable, fecha, porcentaje de avance y tiempo de actividad) que ayudarían a optimizar el desarrollo y finalización del proyecto de titulación.

Para poder desarrollar el proyecto de manera adecuada, ordenada y llegar a la correcta finalización en los tiempos establecidos se propone un esquema que consta de entregables, fecha, porcentaje de avance en el cual se plantean posibles porcentajes de avance que van a ayudar a visualizar si el proyecto se lo está desarrollando de manera efectiva, por última se planteó el tiempo de actividad, que es el tiempo que tomaría la realización eficiente de cada actividad.

Como se observa en la Tabla 1., se logró llegar al objetivo de manera eficiente y en las fechas establecidas como se puede observar en la Figura 1., en la cual se observa cual fue el porcentaje de avance en cada una de las fechas propuestas.

Tabla 1. Planificación del proyecto

| No Actividad | Entregable | Fecha | Porcentaje de Avance | Tiempo Actividad |
|--------------|---|--------|----------------------|------------------|
| 1 | Inicio del proyecto de titulación | 07-oct | 2% | |
| 2 | Documento del lugar para la ubicación de la baliza y parqueadero seleccionado | 08-oct | 5% | 4 horas |
| 3 | Documento con el estudio del hardware para el desarrollo del proyecto. | 14-oct | 10% | 15 horas |
| 4 | Documento con el estudio del software para el desarrollo del proyecto. | 21-oct | 15% | 15 horas |
| 5 | Entrega del módulo raspberry pi en funcionamiento con los módulos | 04-nov | 25% | 35 horas |
| 6 | Entrega de la versión configurada de la baliza | 15-nov | 35% | 25 horas |
| 7 | Entrega de la Aplicación móvil. | 25-nov | 50% | 65 horas |
| 8 | Entrega del prototipo funcional. | 05-dic | 65% | 40 horas |
| 9 | Implementación de geolocalización dentro la aplicación Móvil | 13-dic | 70% | 25 horas |
| 10 | Pruebas realizadas a la aplicación móvil. | 27-dic | 80% | 20 horas |
| 11 | Pruebas del funcionamiento del prototipo | 03-ene | 90% | 20 horas |
| 12 | Entrega del sistema con su funcionamiento completo | 06-ene | 100% | 20 horas |

Nota: Tabla de planificación del proyecto con el porcentaje de avance y el tiempo de cada actividad.
Elaborado por: Junior Ramos y Rodrigo Vaca

Grafica porcentaje de avance del proyecto



Figura 1. Gráfica porcentaje y fechas de avance del proyecto.
Elaborado por: Junior Ramos y Rodrigo Vaca

Fase de diseño: Para lograr el cumplimiento de esta fase se realizaron reuniones con el tutor con la finalidad de presentar avances del diseño y revisar el funcionamiento del prototipo, de esta manera se logró determinar las fallas y posteriormente corregirlas, optimizando el tiempo y los recursos utilizados cumpliendo de manera eficaz con lo deseado por el usuario.

Fase de Desarrollo: Se realiza una proyección del prototipo creado Figura 31., mediante el programa fritzing los programadores pueden interactuar simultáneamente con la herramienta ofreciendo un debate sobre cuál es la mejor solución que va a asegurar calidad en el desarrollo y también se podrá determinar posibles inconvenientes de compatibilidad, diseño, etc.

Fase de pruebas: En esta fase tenemos dos tipos de pruebas las unitarias y las pruebas de aceptación.

En las pruebas unitarias se realizaron pruebas de estrés a cada módulo determinando que cada uno de los módulos son capaces de funcionar bajo cualquier escenario aumentando la calidad del prototipo.

Para las pruebas de aceptación se realizaron diferentes tipos de pruebas para cada usuario como por ejemplo colocar la baliza en un rango más lejano, diferenciar con colores cuando un parqueadero se encuentre libre (verde) u ocupado (rojo).

Capítulo 1

Marco Conceptual

1. Estudio del hardware y software

1.1. Estudio del hardware

El estudio del hardware se divide en dispositivos encargados de monitorear el estado del parqueadero y procesar dicha información, estos son: sensores, controladores y la baliza.

1.1.1. Sensores

Debido a que se busca monitorear la presencia de vehículos en el estacionamiento, el tipo de sensores a utilizarse deben ser los que detecten presencia o distancia, entre los cuales se tienen los siguientes tipos.

1.1.1.1. Sensor Ultrasónico HC-SR04

Es uno de los sensores más utilizados en el mundo de Arduino, debido a que no solo es idóneo para detectar objetos, sino que también es capaz de detectar la distancia en la que este objeto se encuentra con una elevada precisión, una de las ventajas del sensor ultrasónico es que los colores negros o la luz solar no le afectan al momento de realizar el censado. “El sensor ultrasónico envía una onda ultrasónica a través del disparador o trigger, rebota contra el objeto y el receptor detecta la onda” (Hernández, 2019).

Sensor ultrasónico



Figura 2. Sensor ultrasónico para medir distancia.
Elaborado por: Junior Ramos y Rodrigo Vaca

1.1.1.2.Sensor de fin de carrera

Es un pulsador que se puede utilizar normalmente abierto (la corriente no pasa hasta que se presiona el pulsador) o normalmente cerrado (la corriente pasa hasta que se presiona el pulsador) y se utilizan para detectar objetos móviles en un área determinada. “Un final de carrera se lo coloca, justamente, al final de un desplazamiento mecánico, y antes que se active el último interruptor que es la parada de emergencia” (Electrónica, 2015).

Sensor final de carrera



Figura 3. Representación del sensor final de carrera.
Fuente: (Electrónica, 2015)

1.1.1.3.Sensor Inductivo

“Tienen la capacidad de detectar objetos metálicos en áreas de búsqueda generalmente muy pequeñas. El diámetro del sensor es el factor decisivo para la distancia de conmutación, que con frecuencia es de solo unos milímetros” (Fernandez, 2005). La principal funcionalidad de

este sensor es detectar mediante ondas magnéticas detectar materiales metálicos. Es así que podemos conocer si un material férreo se acerca al sensor. Se utiliza mucho en industrias y en el ámbito educativo.



1.1.1.4.Sensor Capacitivo

Son capaces de detectar cualquier tipo de material como metal, plástico, agua o cualquier líquido o sólido, a diferencia de la Figura 3., que solo puede detectar materiales férreos. Además, estos sensores son digitales por lo cual los resultados que arrojen serán evaluados a través de 0 y 1.



Una vez definido los 4 tipos de sensores de presencia, en base a sus especificaciones técnicas, existencia en el mercado ecuatoriano, costos y compatibilidad con la tarjeta de desarrollo Arduino o raspberry, se elaboró la siguiente tabla comparativa.

Tabla 2. Características de los sensores de proximidad

| Características | Ultrasónico | Fin de Carrera | Inductivo | Capacitivo |
|---|---|---|---------------------|---|
| Modelo | HC-SR04 | End Stop Cnc Ramps V1.4 3d Arduino | LJ12A3-4-Z/BX | LJC18A3-B-Z/BX |
| Marca | Genérica | Genérica, diseñada por MakerBot.com | HENGWEI ELECTRONICS | HENGWEI ELECTRONICS |
| Voltaje de alimentación (VCC) | 5 VDC | Hasta 125 VAC | 6-36 VDC | 6-36 VDC |
| Tipo de salida | TTL | Normalmente abierta / Normalmente Cerrada | NPN | NPN |
| Voltaje de salida | Proporcional a la distancia del objeto | VCC | VCC | VCC |
| Corriente de funcionamiento | 15 mA | Al ser mecánico, no necesita ser alimentado | 300 mA | 300 mA |
| Corriente máxima | No se especifica | 1 A | No se especifica | No se especifica |
| Distancia de detección | 2 cm – 4m | Debe haber contacto | 4mm +-10% | 1-10 mm |
| Tipo de objetos detectados | Metálicos y no metálicos capaces de reflejar las ondas ultrasónicas | Sólidos | Metálicos | Metálicos y no metálicos (plástico, vidrio, agua, aceite) |
| Costo aproximado a la fecha 21/10/2019 | \$3 | \$3 | \$ 12 | \$13,50 |

Nota: Tabla comparativa entre los 4 posibles sensores de proximidad a utilizar.
Elaborado por: Junior Ramos y Rodrigo Vaca

1.1.1.5. Criterio de selección del sensor

Analizada la Tabla 2., la característica más relevante para escoger el dispositivo es la distancia de detección, ya que se busca tener un sensor el cual debe tener la capacidad de detectar al vehículo sin estar muy cerca a éste, por lo que se ha tomado la decisión de utilizar en el proyecto el sensor ultrasónico HC-SR04 que cuenta con una distancia de detección de hasta 4 metros

versus el resto de sensores cuya distancia máxima de detección fue de 1cm, además el sensor ultrasónico y el fin de corriente son los más baratos de los 4 modelos analizados.

1.1.2. Controladores

El controlador será el encargado de recibir la señal por parte del sensor y conectarse con la baliza y la aplicación mediante bluetooth, por lo que se van a definir los posibles controladores que podrían utilizarse en base a sus especificaciones técnicas, existencia en el mercado ecuatoriano, costos y compatibilidad el sensor ultrasónico.

1.1.2.1. Tarjeta de desarrollo Arduino 2560

Arduino nos ayuda al desarrollo de proyectos de orden educativo, proyectos que nos permiten la programación de microcontroladores junto con la electrónica, además podemos comunicar esos pequeños dispositivos que tienen un lenguaje de programación de bajo nivel con lenguaje de alto nivel como es html5 entre otros. “Arduino es una tarjeta de desarrollo de hardware libre construida con el microcontrolador Atmega 2560” (Veloso, 2018).



1.1.2.2. Raspberry pi B+

Es un ordenador de placa simple, su costo es bajo, diseñada inicialmente para ser utilizado en el ámbito educativo en escuelas y universidades. Ofrece la posibilidad de integrar un software

de código abierto, Raspbian, una versión adaptada de Debian es su sistema operativo oficial y el que se usa para la implementación del prototipo y para que el acceso a la aplicación web desde el celular en el presente proyecto (Agelet Aumedes, 2019).

Características:

Tabla 3. Características Raspberry pi3 B+

| Características | Raspberry pi 3 model b+ |
|---------------------------------|---|
| Procesador | Broadcom bcm2837b0, cORTEX-a53 (armV8) 64-BIT sOc |
| Frecuencia de reloj | 1,4 ghZ |
| GPU | VideoCore IV |
| Memoria | 1gb lpddr2 sdram |
| Conectividad inalámbrica | 2.4ghZ / 5ghZ ieee 802.11.B/G/N/AC Bluetooth 4.2, ble |
| Conectividad de red | Gigabit Ethernet over USB 2.0 (300 Mbps de máximo teórico) |
| Puertos | GPIO 40 pines HDMI 4 x USB 2.0 CSI DSI Micro SD USB |

Nota: Características principales de la raspberry pi3 B+.
Fuente: (Pastor, 2018)

Raspberry pi3 B+



Figura 7. Raspberry pi3 B+.
Elaborado por: Junior Ramos y Rodrigo Vaca

1.1.2.3. Microcontrolador PIC

Son una familia de microcontrolador desarrollados por la empresa Microchip. Estos microcontroladores son muy utilizados por los profesionales y estudiantes que utilizan estos microcontroladores para realizar sus proyectos. Esto se debe a su coste inferior al de resto de competidores, gran velocidad de funcionamiento, reducido juego de instrucciones, muchas herramientas de software libres y gran variedad de modelos (Martínez, 2019).

El microcontrolador PIC precisa un circuito de alimentación de 5 V, carece de un oscilador interno y el terminal MCLR se encuentre en alto durante la ejecución del programa (a 0 V se produce un reset) (Imas, 2017).

A continuación, se realiza una comparativa entre los controladores antes mencionados tomando en cuenta los criterios que necesitamos para nuestro proyecto, además debemos conocer cuál de ellos nos ofrece alguna ventaja en su costo, facilidad de programación, compatibilidad o su software de programación. Una vez definidos vemos a continuación las características principales de cada uno de ellos.

Tabla 4. Controladores y sus características

| Características | Arduino Mega | Raspberry Pi 3 | PIC |
|---|--------------------------------------|---------------------------|--|
| Modelo | Mega 2560 | Raspberry Pi 3 B+ | 16f877a |
| Marca | Arduino | Raspberry | Microchip |
| Voltaje de alimentación | 7-12 VDC | 5 VDC | 2-5.5 VDC |
| Voltaje de trabajo | 5 VDC | 5 VDC | 3.3 – 5 VDC |
| Tipo de salida | Digital, PWM | Digital | Digital, PWM |
| Corriente de funcionamiento | 40 mA por cada Pin de I/O | 16 mA por cada pin GPIO | 25 mA por cada Pin de I/O |
| Corriente máxima | 200 mA en total para todos los pines | 50 mA para todos los GPIO | 200 mA para todos los pines en grupos de 3 puertos |
| Pines digitales I/O | 54 | 48 | 33 |
| Bluetooth embebido | No | Si (Bluetooth 4.2) | No |
| Método de Programación | USB | USB | Dispositivo Programador extra |
| Software de programación | IDE Arduino | Scratch 2.0, Thonny IDE | PIC C, MPLab |
| Costo aproximado a la fecha 21/10/2019 | \$18 | \$80 | \$5 |

Nota: Tabla comparativa entre los 3 controladores Arduino, raspberry y pic.
Elaborado por: Junior Ramos y Rodrigo Vaca

1.1.2.4. Criterios de selección del controlador

Examinada la Tabla 4., la principal característica para elegir el dispositivo es la forma en cómo se va a comunicar con la baliza ya que es nuestra prioridad en el proyecto. Es por eso que se seleccionó el controlador raspberry pi y continuación, se detallan algunas características por su selección.

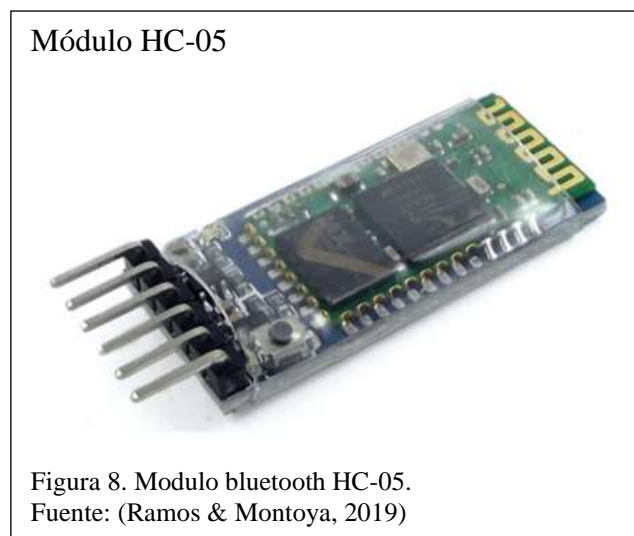
- La Raspberry a diferencia de Arduino cuenta con un módulo Wifi y Ethernet integrado, con el cual se podrá enviar a la aplicación web el estado del parqueadero
- A diferencia de Arduino, la Raspberry puede alojar un servidor
- Raspberry utilizará un mensaje para enviar una url que redirija al usuario hacia una aplicación web donde podrá consultar el estado del parqueadero

- Al no ser de código abierto, no se puede cambiar el funcionamiento de la baliza y su aplicación queda limitada a enviar un único mensaje vía bluetooth
- La baliza solamente se comporta como emisor de un mensaje vía bluetooth (hasta 10 dispositivos), por lo tanto, no se puede conectar directamente con Arduino

1.1.3. Módulos Bluetooth

1.1.3.1.HC-05

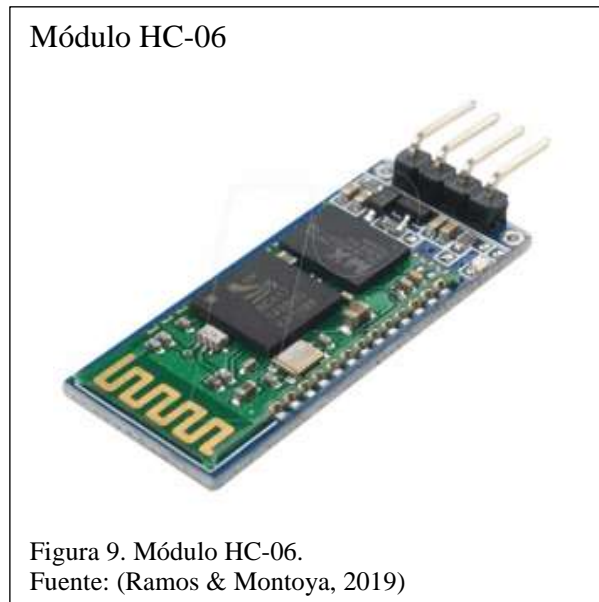
Es un módulo Bluetooth es utilizado hoy en día en una amplia gama de proyectos o trabajos donde necesites una conexión inalámbrica confiable y sencilla de utilizar. Una de las características fundamentales del HC-05 es su transferencia de datos, ya que se lo configura en modo Maestro-Esclavo, es decir que tiene la capacidad de establecer conexiones mediante una computadora o celular, también establece una comunicación con varios dispositivos bluetooth cercanos (Gonzalez & Velásquez, 2019).



1.1.3.2.HC-06

Es uno de los sensores más adquiridos en el mercado, esto se debe a su bajo costo y facilidad de uso. Algunas de sus características más importantes son: el costo, por lo mencionado anteriormente, al ser popular el módulo HC-06 posee la ventaja de ser un dispositivo potencialmente vendido y por ende con un costo bajo de venta, lo que hace atractivo para

realizar cualquier tipo de proyecto universitarios o trabajos. La segunda ventaja del módulo HC-06, es su facilidad de adquisición en cualquier mercado local, a diferencia de otros dispositivos que al no ser conocidos no tienen buena notoriedad en el mercado local (Ramos & Montoya, 2019).



1.1.3.3. Raspberry pi 3 modulo bluetooth

Sus características más importantes son:

- “Cargador de 1 A para la RaspBerry Pi 1; 1.5 A para la Raspberry Pi 2 y 2.5 A para la Raspberry Pi 3. Acepta cualquier cargador de Android con Micro-USB” (Macías Olives, 2018, pág. 20)
- “Se puede conectar una pantalla HDMI o una pantalla con RCA. - Se puede conectar Teclado y Ratón a través de sus puertos” (Macías Olives, 2018, pág. 20)
- “Incluye un conector para Cable Ethernet o adaptador Wifi USB. En la Raspberry Pi 3 viene incorporado el módulo Wifi” (Macías Olives, 2018, pág. 21)
- “Dispone de una ranura para tarjeta SD o MicroSD mayor a 8Gb se recomienda de clase 10 en adelante” (Macías Olives, 2018, pág. 21)

Componentes de la placa raspberry pi3 B+

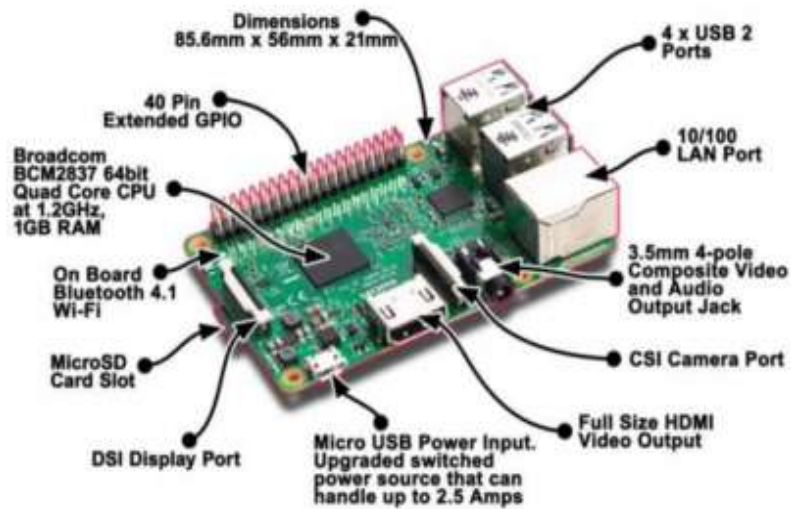


Figura 10. Puertos de conexión raspberry pi3 B+.
Fuente: (Macías Olives, 2018)

1.1.3.3.1. Sistema Operativo de la tarjeta Raspberry pi 3.

“Una parte importante de la Raspberry pi3 es su Sistema Operativo; el mismo que se instalará en la tarjeta SD y con la facilidad de que al cambiar el Sistema Operativo simplemente se cambie de tarjeta” (Macías Olives, 2018, pág. 21).

Sistema operativo raspberry Pi3



Figura 11. Sistema operativo Noobs y Raspbian.
Fuente: (Macías Olives, 2018)

A continuación, tenemos una tabla comparativa entre las características más importantes de cada uno de los módulos antes descritos.

Tabla 5. Módulos bluetooth y sus características

| Características | Hc 05 | Hc 06 | Bluetooth Raspberry |
|---|-----------------------------|-----------------------------|----------------------------|
| Funcionalidad | Maestro Esclavo | Esclavo | Esclavo |
| Compatibilidad | Desde v2.0 | Desde v2.0 | Desde v4.0 |
| Frecuencia | 2.4Ghz | 2.4Ghz | 2.4Ghz y 5Ghz |
| Sensibilidad | Aprox 4dbm -+0.3 | Aprox 4dbm-+0.3 | Aprox 4dbm-+0.3 |
| Distancia de radiación | 10 m en condiciones optimas | 10 m en condiciones optimas | 7 m en condiciones optimas |
| Puerto Serie | si | no | no |
| Autenticación y Seguridad | si | si | si |
| BLE | si | si | si |
| Compatibilidad | Android 6 +, IOS 5+ | Android 6 +, IOS 5+ | Android 6 +, IOS 5+ |
| Voltaje de alimentación | 3.3 a 5 Vdc | 3.3 a 5 Vdc | 5Vdc |
| Costo aproximado a la fecha 21/10/2019 | \$6 | \$6.50 | \$80 |

Nota: Comparativa entre módulos bluetooth hc-05, hc-06 y bluetooth raspberry.
Elaborado por: Junior Ramos y Rodrigo Vaca

1.1.3.4. Criterios de selección del módulo bluetooth

Debido a que el controlador seleccionado es raspberry pi3 B+, se descarta automáticamente utilizar el módulo bluetooth HC-05 y HC-06, en su lugar se utilizará el bluetooth de la raspberry pi, debido a que el funcionamiento de la baliza no permite conexión con Arduino o pic, además raspberry permite montar un servidor en la misma y cuenta con conexión bluetooth 4.2, BLE, lo que permite enviar una cadena de caracteres (mensaje aceptado por la baliza), es decir que es capaz de recibir y transmitir datos y es una característica necesaria para este proyecto, el cual debe recibir datos de la baliza y enviar datos del sensor sobre el estado del parqueadero.

1.1.4. Baterías

Para poder desarrollar un proyecto electrónico se debe ampliar los conocimientos sobre los elementos que van a integrar el proyecto final, que, al ponerlo en funcionamiento, deberá realizar o cumplir con los objetivos propuestos al inicio del mismo. Las baterías o fuentes de energía cumplen un papel fundamental para el funcionamiento del proyecto y para esto hay

varios tipos de baterías o fuentes de energía que ayudarán, entre ellos tenemos LiPo, LiOn y fuentes de alimentación.

1.1.4.1.LiPo

“Tiene la capacidad de almacenar altas densidades de energía gracias a su componente químico, el Polímero de iones de Litio” (TBem, 2017, pág. 1).

Brindan energía en una mayor cantidad de tiempo (2 a 3 años), una de sus fortalezas es que tiene un peso relativamente pequeño y una pequeña dimensión y es por eso que puede brindar energía a nuestro proyecto final por una mayor cantidad de tiempo. Es por este motivo que muchos la utilizan para proyectos electrónicos, robótica, etc.

A continuación se detallan sus características principales:

Tabla 6. Características batería de polímero de litio

| Características | Descripción |
|---------------------------------|----------------------|
| Capacidad Mínima | 5000mAh |
| Configuración | 3S1P/11.1V/3Celdas |
| Constante de descarga | 25C |
| Pico de descarga (10sec) | 35C |
| Peso | 370g |
| Dimensiones | 146x50x25 mm |
| Plug de carga | JST-XH |
| Plug de descarga | 4mm Bullet-connector |

Nota: Características batería de polímero de litio.

Fuente: (TBem, 2017)

1.1.4.2.LiOn

Se trata de una pila recargable que se encuentra conformada por dos o más celdas en las que se encuentran los iones de litio, el modo de funcionamiento es mediante la combinación de celdas se produce energía la cual se la va a utilizar para proveer de corriente a cualquier dispositivo que conectemos (Ordoñez, 2016).

Este tipo de baterías son muy utilizadas debido en el ámbito electrónico por estudiantes y profesores ya que es muy fácil de utilizar y en el mercado ecuatoriano tiene un costo bajo y un gran beneficio al momento de utilizarlo en el desarrollo de proyectos que necesiten de baterías para su funcionamiento.

1.1.4.3.Fuente de alimentación

Las fuentes de alimentación tienen una gran cantidad de usos en proyectos, las podemos clasificar en fuentes de alimentación reguladas y fuentes de alimentación no reguladas. Las fuentes no reguladas no son muy recomendables debido a su gran tamaño y peso, son sencillas pero su voltaje no se mantiene muy estable y afectará al prototipo. Las fuentes reguladas sin embargo son aquellas que mantienen su voltaje estable de manera estable.

Tabla 7. Características de las baterías

| Características | LiPo | LiOn | Fuente Alimentación |
|---|-------------------------------|---|--|
| Vida útil | Mayor 500 ciclos de carga | Menor a 300 ciclos de carga | Mayor de 1mil horas de uso |
| Otros | Tecnología popularmente usada | Pierde capacidad de carga debido al clima ambiental | Diseñado para uso continuo y extenuante |
| Comportamiento de capacidad de carga | Lineal | Lineal | Lineal |
| Voltaje nominal | 3,7 V por celda | 1,2 V por celda | Fuente conmutada desde 2,4 V hasta 30V dependiendo de la tecnología. |
| Voltaje de entrada | No | No | 120vca, o 5 Vdc |
| Costo aproximado a la fecha 21/10/2019 | \$20 | | \$6 |

Nota: Tabla comparativa entre batería LiPo, LiOn y fuente de alimentación.
Elaborado por: Junior Ramos y Rodrigo Vaca

1.1.4.4. Criterios de selección fuente de alimentación

Analizada la Tabla 7., se optó por utilizar en el proyecto un PowerBank que es una batería portátil recargable que es utilizada para cargar dispositivos electrónicos median una salida USB. Se seleccionó dicha batería por su bajo costo y tamaño lo que es útil para nuestro proyecto.

1.1.5. Balizas FeasyBeacon

Es una tecnología de proximidad bluetooth de baja energía, es decir, tiene mayor rango de detección. Permite enviar información a diferentes dispositivos móviles a través de bluetooth. La aplicación propietaria de la baliza feasybeacon funciona tanto en IOS como en Android, es compatible con Eddy Stone (url). La distancia de detección es de 400m y debido a que es pequeña y ligera es perfecta para instalaciones estables como en tiendas, museos, estadios, restaurantes, etc.

Baliza



Figura 12. Baliza, dispositivo para emitir datos a través de bluetooth.
Elaborado por: Junior Ramos y Rodrigo Vaca

1.1.5.1.Principios de funcionamiento

La tecnología Beacons se basa en transmisión mediante Bluetooth de baja energía es decir que la transmisión tendrá un rango de 400 a 500 metros. Esta tecnología únicamente envía información, en el siguiente proyecto de titulación la baliza envía una cadena de caracteres encapsulada en una URL. Las balizas únicamente se activan cuando un dispositivo las detecta y se le pide enviar la información (Ryte, 2019).

1.1.5.2.Uso práctico

Los posibles usos de las balizas son:

Compras móviles: Se usan en tiendas para brindar información de cada uno de los productos que están en ofertas o rebajas, incluso para conocer los precios de dichos productos. En centros comerciales de Europa se usa muy a menudo este tipo de publicidad en las compras para atraer más clientes.

Pago móvil: “En el punto de venta, los usuarios pueden realizar pagos sin efectivo con muy poca interacción. Todo lo que tienen que hacer es estar cerca de un terminal con el dispositivo final” (Ryte, 2019, pág. 1).

Navegación: Se utiliza para obtener una descripción del lugar en donde se encuentra el usuario, es decir cuando viaja a una nueva ciudad se obtiene información de cada lugar de destino (nombre, historia, productos y datos relevantes), así mismo se aplica en aeropuertos, estadios, edificios.

Contenido relevante para el contexto: “Dependiendo del escenario de aplicación, se pueden utilizar beacons y un dispositivo receptor para mostrar ciertos contenidos relevantes, que también se pueden vincular a la compra móvil, la navegación o la domótica” (Ryte, 2019, pág. 1).

1.2. Estudio del software

El estudio de software contempla una gran diversidad plataformas que se usan para poder programar aplicaciones Android, en el presente proyecto usaremos la aplicación propia de la baliza que es FeasyBeacon como se detalla en el capítulo 3, por otro lado, existen lenguajes de programación para desplegar la información sobre el estado de los parqueaderos.

1.2.1. HTML5

Es un estándar que se utiliza para el diseño de páginas web, que sirven para visualizar información ya sea fotos, videos o tablas desde cualquier smartphone.

Tabla 8. Ventajas y desventajas HTML

| Lenguaje | Ventajas | Desventajas |
|----------|--|---|
| HTML | <ul style="list-style-type: none"> -Permite describir hipertexto. -Texto en forma estructurada. -Cuenta con editor de páginas web o WYSIWYG. -Cuenta con archivos pequeños. -Cuenta con despliegue rápido. -Su lenguaje es fácil de aprender -Se ejecuta en todo tipo de navegadores web. | <ul style="list-style-type: none"> -Lenguaje estático -Su interpretación en cada navegador web puede ser diferente. -Almacena muchas etiquetas que son consideradas como basura. -Dificultad de corrección -Diseño lento -Etiquetas limitadas |

Nota: Tabla de ventajas y desventajas de HTML.
Elaborado por: Junior Ramos y Rodrigo Vaca

1.2.2. Python

Es un lenguaje de programación de alto nivel que ha alcanzado una gran popularidad, ya que al ser un lenguaje simple, versátil y fácil de desarrollar. Es un lenguaje bastante utilizado para la interpretación de diferentes dispositivos electrónicos y sensores. Este lenguaje ofrece la capacidad de crear todo tipo de programas desde páginas web hasta programas complejos, además es multiplataforma y tiene una sintaxis muy clara al momento de su programación (Palacios, 2015, pág. 75).

A continuación, se muestran algunas características del lenguaje.

Tabla 9. Características Python

| Lenguaje | Características | Fortalezas | Debilidades | Opinión |
|----------|--|---|-------------------------------------|---|
| Python | -Crea cualquier programa o sitio web. -No requiere compilación. | -Código abierto -Permite resolver cualquier tipo de problemas. -Lenguaje Multiplataforma, POO y portable. | -Interpretación de lenguajes lenta. | -Se utiliza para crear páginas web dinámicas debido a que es multiplataforma. |

Nota: Tabla características de Python.
Fuente: (Desarrollo Web, 2015)

1.2.3. Android Studio

Hoy en día Android Studio es denominado un entorno de desarrollo basado en lenguaje de programación de JAVA donde proporciona la función a los usuarios de programar aplicaciones móviles de toda índole para Sistemas Operativos Android. Esta plataforma posee una amplia gama de herramientas de edición de códigos, las diferentes herramientas para desarrolladores de IntelliJ IDEA, donde para ejecutar el código el software cuenta con emuladores de dispositivos Android con el fin de que los programadores puedan verificar la aplicación desarrollada y conocer los errores que se pueden presentar durante la compilación (Rosas Ganán, 2019, pág. 21).

Android Studio



Figura 13. Interfaz Android Studio.
Fuente: (Caicedo & Camargo, 2019)

A continuación, mostramos algunas características de Android Studio.

Tabla 10. Características de Android Studio

| Características | Android Studio |
|--------------------------------|---|
| Visualización | Línea de comandos |
| Programación | Comandos de programación tradicional |
| Accesibilidad | Acceso desde el ordenador donde se desarrolla la aplicación |
| Compatibilidad | Aplicación Android |
| Versión recomendada | Versión 3.5 |
| Plataformas compatibles | Windows x32-x64, Linux |
| Procesamiento | Tiempo Real |
| Otros | Usa altos recursos de procesamiento. |

Nota: Tabla de características de Android Studio.
Fuente: Junior Ramos y Rodrigo Vaca

1.2.4. Sensor GPS neo 6m

Es un módulo GPS muy popular, rentable y de alto rendimiento con una antena de parche de cerámica, un chip de memoria incorporado y un batería de respaldo que se puede integrar convenientemente con una amplia gama de microcontroladores. También tiene una alta sensibilidad para aplicaciones en interiores. Además, hay una batería recargable compatible con MS621FE para respaldo y EEPROM para almacenar ajustes de configuración. El módulo funciona bien con una entrada de CC (HAREENDRAN, 2019).

Modulo GPS NEO 6M



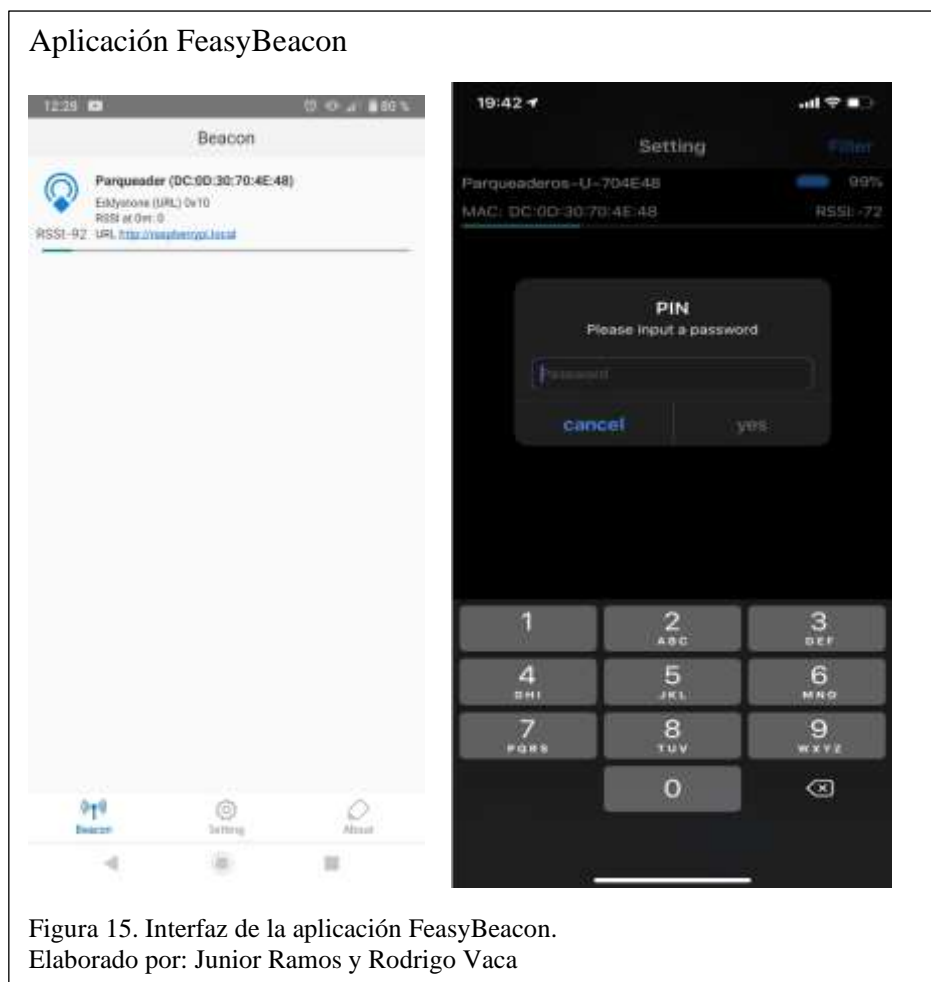
Figura 14. Modulo GPS NEO 6M.
Elaborado por: Junior Ramos y Rodrigo Vaca

Capítulo 2

Diseño del prototipo

2. Funcionamiento de la aplicación FeasyBeacon

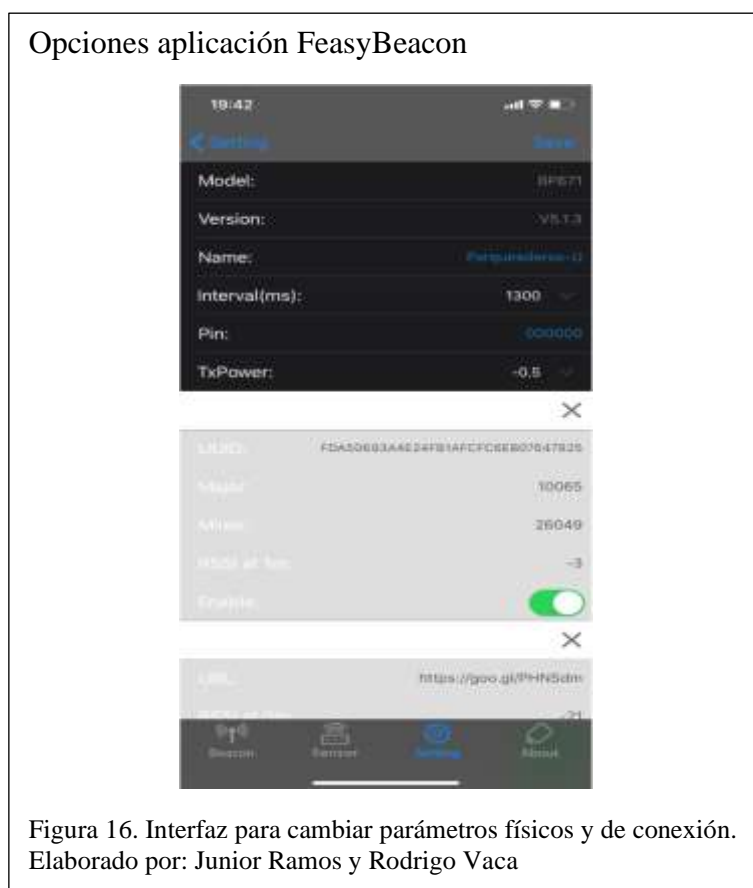
La baliza que se utilizará en el proyecto a desarrollar es de la tecnología FeasyBeacon, el cuál es un dispositivo emisor que mediante bluetooth de baja energía nos va a permite enviar información en forma de carácter hacia varios dispositivos en forma de broadcast a quienes tengan instalado la aplicación FeasyBeacon App en sus smartphones.



En las siguientes imágenes podemos observar cual es el comportamiento de la aplicación FeasyBeacon. Como tal hay 4 opciones con las cuales se podrá interactuar con la aplicación y la baliza. Para fines de pruebas se procedió a usar las siguientes opciones:

- Beacon
- Setting

La opción Beacon indica cuantas balizas de la marca FeasyBacon se encuentran dentro del rango de censado de la aplicación. En la imagen de la izquierda podemos observar el nombre Parquaderos - UPS, es como fue configurada la baliza para fines de prueba y conexión. En la imagen de la derecha se procede a configurar la baliza desde la opción Setting, aquí inicialmente se deberá incluir la clave por defecto “000000”. Una vez dentro se podrá variar parámetros físicos y de conexión de la baliza como se observa en la Figura 16.



Los principales parámetros que se muestran son:

Nombre del dispositivo, modelo, versión de firmware, intervalo de actualización, potencia de transmisión en db. La baliza fue diseñada para trabajar bajo parámetros de envío de cadena de

caracteres, en este caso se utilizará el apartado de URL. Si todos los parámetros están correctamente configurados, se procederá a grabar la configuración con el botón SAVE.

2.1.Instalación Sistema Raspbian en Raspberry Pi

1. Primero se descarga la imagen del sistema operativo Raspbian

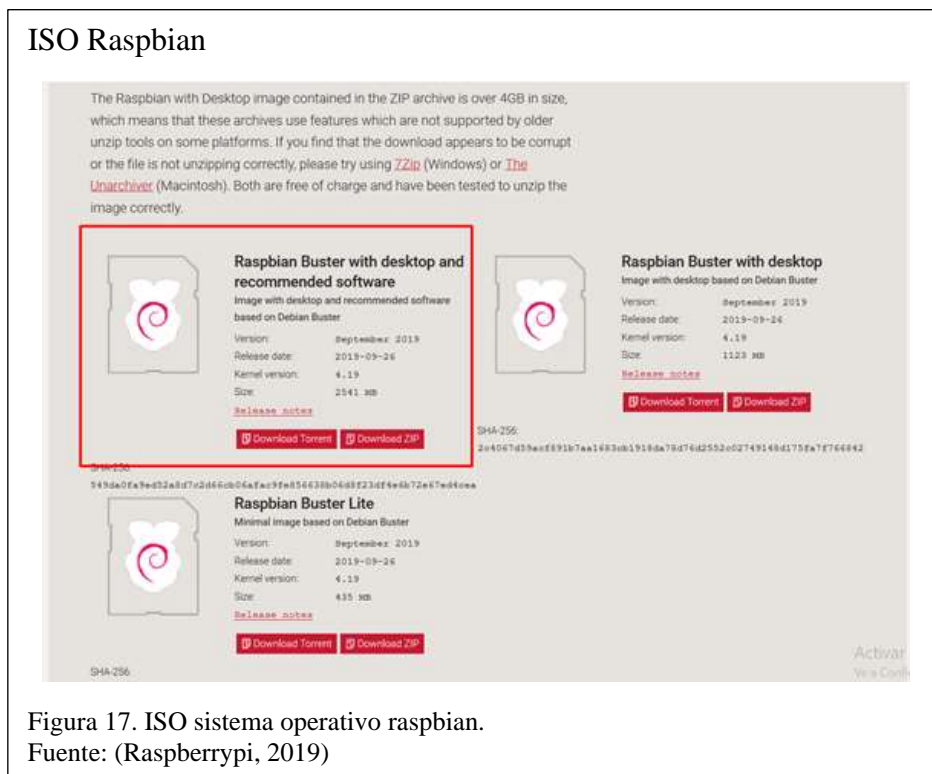


Figura 17. ISO sistema operativo raspbian.
Fuente: (Raspberrypi, 2019)

2. Se Verificará las reléase notes de la instalación, y verificamos que posea la compatibilidad con la versión de la raspberry pi 3 versión B+.

Instalación ISO raspbian

```
* Raspberry Pi firmware 5db8e4e1c63178e200d6fbae23ed4a9bf4656658
2018-03-13:
* Raspberry Pi 3 B+ support
* WiFi is disabled until wireless regulatory domain is set (Pi 3 B+ only)
  - The domain can be done through 'Raspberry Pi Configuration' (rc_gui),
    'raspi-config' or by setting 'country=' to an appropriate ISO 3166
    alpha2 country code in /etc/wpa_supplicant/wpa_supplicant.conf.
* Default wireless regulatory domain is now unset
* Added support to desktop for different screen sizes and resolutions,
  including multiple preset options in Appearance Settings and pixel doubling
  option in Raspberry Pi Configuration
* Version 2.1.16 of Thonny included
* Version 29.0.0.113 of Adobe PepperFlash player included
* Version 1.2.post1 of Pygame Zero included
* Bluetooth plugin now supports connection to Bluetooth LE HID devices
* Network plugin now indicates 5G-compatible APs
* Latest changes to BlueZ ALSA service merged
  - service now started on CLI boot as well as GUI boot
* Latest changes to dhcpcd networking plugin merged
* Improved support for running on pi-top devices
* Small design changes to PiX theme and icons
* Bug fix - hide spurious window resize handles
* Bug fix - Scratch 2 remote GPIO state block now works correctly
* Updated WiFi Firmware
  - brcmfmac43455-sdio 7.45.154
  - brcmfmac43430-sdio 7.45.98.38
```

Figura 18. Despliegue de la instalación de la ISO de raspbian.
Elaborado por: Junior Ramos y Rodrigo Vaca

3. Se procede al link de descarga en formato Zip, y procedemos a instalar la imagen en el micro sd con el programa Rufus, que nos ayudara a crear un USB booteable. Una vez realizada la configuración, se tiene la raspberry pi lista para su uso.

Instalación de la ISO en el micro sd



Figura 19. Instalación de la ISO en el micro sd con el programa Rufus.
Elaborado por: Junior Ramos y Rodrigo Vaca

2.2. Esquema de funcionamiento

A continuación, se detalla cuáles son las opciones para el levantamiento del servidor, se deberá tomar en cuenta que el proyecto es un prototipo para la Universidad y se elegirá la opción más económica y que no implique mucha complejidad y tiempo.

Para la implementación del proyecto se plantean 2 soluciones las cuales serán detalladas a continuación determinado también cada una de sus limitantes. Al final elegimos la mejor opción tanto para el nosotros como diseñadores y para los usuarios. Se debe tener en cuenta el funcionamiento del prototipo, se necesita almacenar la información que sea obtenida a través del sensor y la Raspberry Pi 3 y es por eso por lo que podemos utilizar el servidor firebase o el servidor apache.

Opción 1: La primera solución es utilizando el Servidor Firebase para acceder a través de internet a la aplicación web desde cualquier parte del mundo. Si el servidor es firebase los datos quedarán almacenados en la nube de firebase y quedarán accesibles desde cualquier red wifi o plan de datos y únicamente se necesitará una condición para poder acceder a dicho datos:

- a. **Conexión a baliza:** El celular debe conectarse a la baliza a través de la aplicación FeasyBeacon.

Opción 2: Segunda solución es utilizando el Servidor Apache (implementado en la Raspberry) que para acceder a la aplicación web se lo debe hacer mediante una red local (LAN) de la universidad. Si el servidor es apache, los datos serán vistos en tiempo real en el servidor de la Raspberry, con la limitante de que solo se podrá acceder si tenemos dos condiciones:

- a. **Conexión a la misma red:** La Raspberry Pi y el celular del usuario en donde está instalada la aplicación deben estar conectados a la misma red wifi (Universidad Politécnica Salesiana).
- b. **Conexión a la baliza:** El celular debe conectarse a la baliza a través de la aplicación FeasyBeacon.

Limitantes que considerar: Si el celular del usuario está en una red wifi diferente, o con su propio plan de datos no habrá posibilidad de ver los datos del servidor debido a que no se encontraría dentro de la misma red.

Tomando en cuenta las dos opciones mencionadas se plantean 2 interrogantes:

1. ¿Es necesario que el proyecto requiera conectarse a la red privada de la Salesiana?

Si este es el caso los datos de censado del parqueadero serán locales y accesibles únicamente bajo la condición de estar conectados tanto la raspberry pi como el usuario a la red wifi de la salesiana.

Si se da este punto, se podrá implementar el servidor bajo Apache.

2. ¿Si el proyecto requiere que se tenga acceso a internet a través de cualquier red?

Si este es el caso los datos de censado del parqueadero serán accesibles al estar conectados a cualquier red wifi o plan de datos celular.

Si se da este punto, se podrá implementar el servidor bajo firebase, ya que firebase es un servidor alojado en la nube con la capacidad de integrar datos desde cualquier parte del mundo.

Consideraciones: La aplicación está diseñada para los usuarios de la UPS, siendo ese el caso al momento de sacar las credenciales necesarias para obtener un puesto en el parqueadero cada semestre uno de los nuevos requisitos sería que al momento de obtener el cupo se instale la aplicación móvil en su celular.

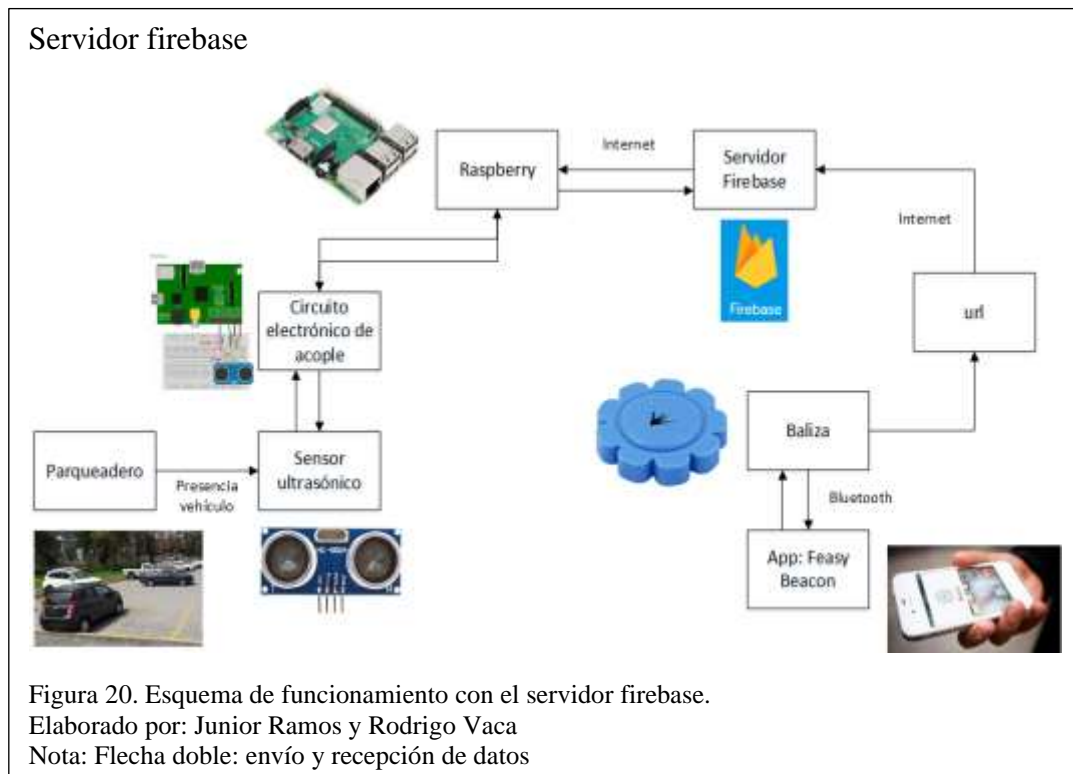
Si un usuario que no es parte de la UPS ingresa como invitado al parqueadero va a tener que conectarse tanto a la baliza como a la Red wifi de la institución.

Nota: Para poder controlar de forma la raspberry pi se usa el programa VNC Viewer, que nos permite controlar de forma remota un ordenador a través de un cliente multiplataforma. Primero

o descargamos de nuestro navegador y lo instalamos podemos acceder a nuestra raspberry y controlarla.

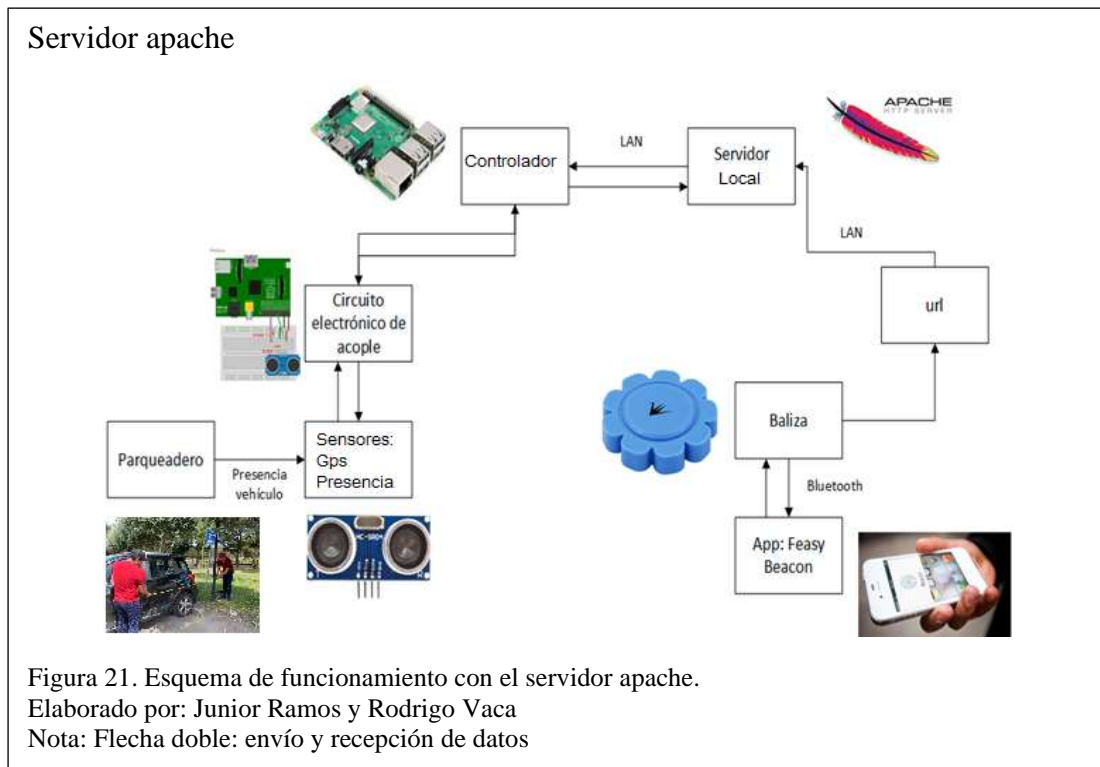
Los esquemas de conexión de ambas opciones se detallan a continuación:

2.2.1. Opción 1: Servidor Firebase



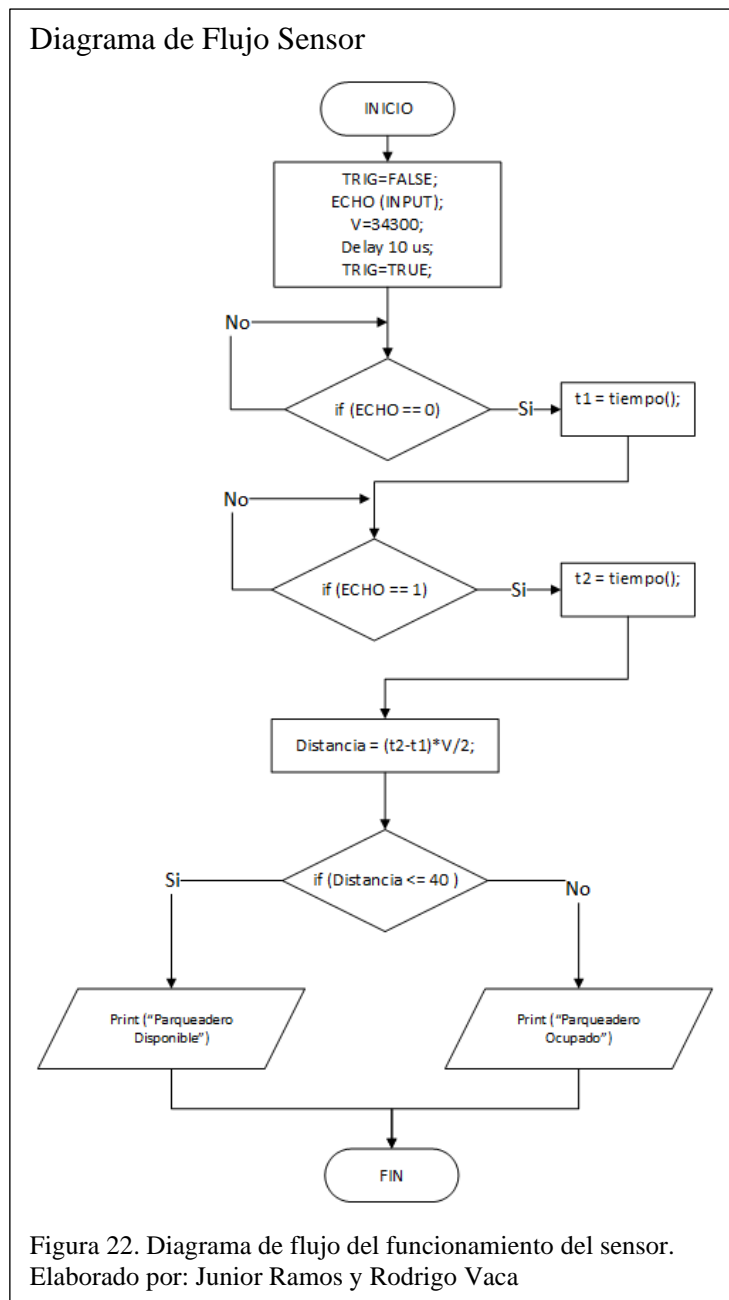
La Figura 20., muestra el funcionamiento del prototipo con el servidor firebase, el funcionamiento es el siguiente, el sensor ultrasónico detecta la presencia de un vehículo en el parqueadero y mediante del circuito envía los datos a la raspberry y a través de una conexión a internet dichos datos son recibidos por el servidor firebase que los aloja en la nube. Cuando el usuario abre la aplicación en su celular se conecta a la baliza y accede al URL que se despliega para poder acceder al servidor que contiene los datos y es así como se determinará si se dispone o no de un parqueadero disponible.

2.2.2. Opción 2: Servidor Apache



La Figura 21., muestra el funcionamiento del prototipo con el servidor apache, y el funcionamiento es casi el mismo al anterior, cuando el sensor ultrasónico recibe los datos los envía a la raspberry, pero ahora la conexión no es a través de internet, los datos se visualizan en tiempo real en la raspberry que es donde se montará el servidor. Para tener acceso a los datos tanto el usuario como la raspberry se deben conectar a la red de la universidad, se accederá a los datos en el servidor mediante la URL que nos proporcione la baliza.

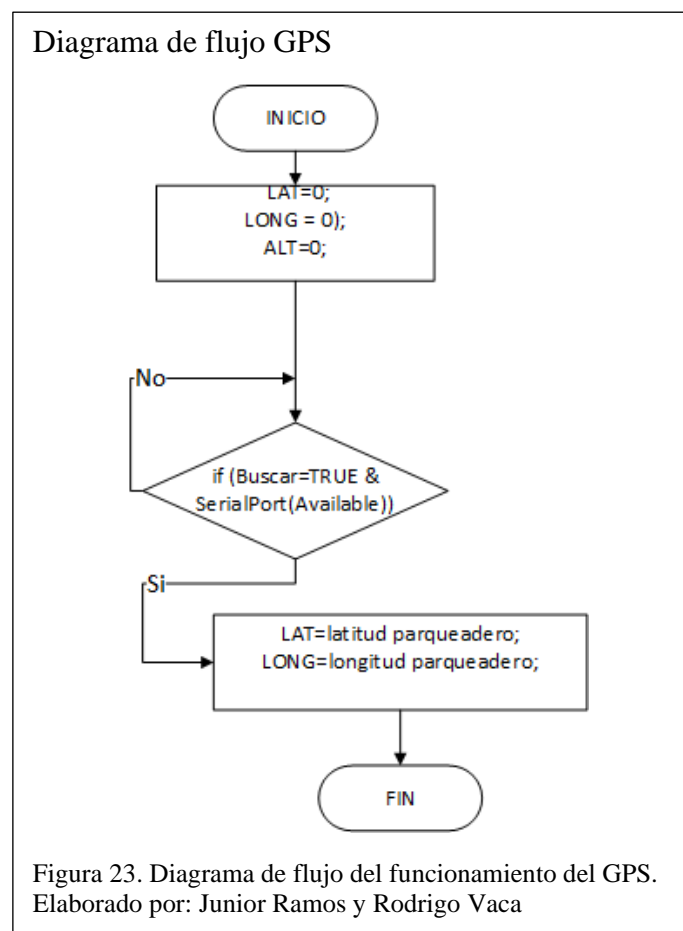
2.3. Diagrama de flujo del sensor



Primero se desactiva el trigger del sensor (encargado de disparar la señal ultrasónica), luego se espera un tiempo de 10 microsegundos (necesarios para estabilizar la señal); posteriormente se activará logrando el disparo de la señal ultrasónica, finalmente mientras no se detecte señal, es decir, el pin ECHO sea igual a cero (0) se tomará el tiempo del proceso, siendo este el tiempo inicial t1.

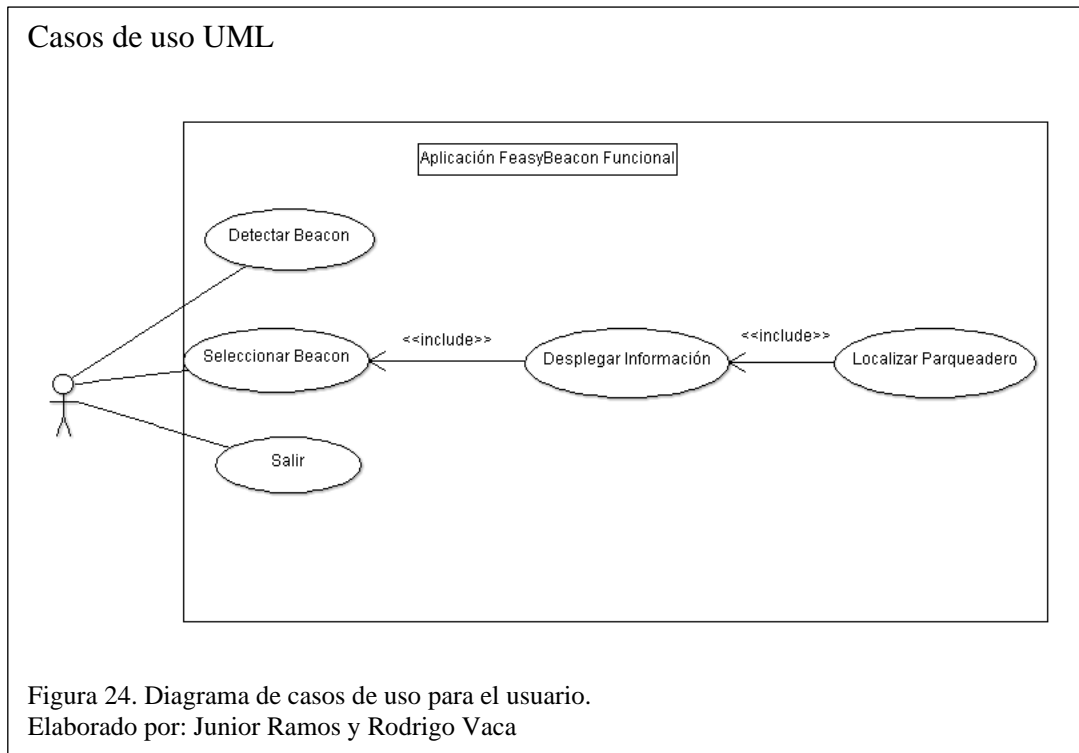
Después cuando detecte el rebote de la señal, es decir, el pin ECHO cambie a uno (1) se volverá a tomar el tiempo del proceso, siendo el tiempo final t_2 . Una vez tomados estos tiempos se aplicará la fórmula para calcular la distancia hacia el objeto donde rebotó. Por último, para la detección del parqueadero disponible se va a tomar una distancia de 40 cm, si la distancia medida por el sensor es menor o igual a ese valor, se asumirá que el espacio está ocupado.

2.4. Diagrama de flujo GPS



El módulo GPS NEO-6M, tiene sus valores en cero al iniciar (latitud, longitud y altitud), pero al momento de ponerlo en funcionamiento nuestra variable buscar tendrá un valor de verdadero y el puerto serial disponible, si no se cumple esta condición se repetirá el proceso y buscare nuevamente, pero si se cumple esta función el módulo GPS arrojará la latitud y longitud que fue leída en ese momento.

2.5. Diagrama de casos de uso UML



El usuario abre la aplicación FeasyBeacon para poder detectar la baliza. Selecciona la baliza encontrada y despliega la información del estado del parqueadero y de forma continua se despliega su localización. Finalmente, el usuario puede realizar otra búsqueda para desplegar la información actualizada o salir de la aplicación.

Capítulo 3

Desarrollo de la aplicación y análisis de resultados

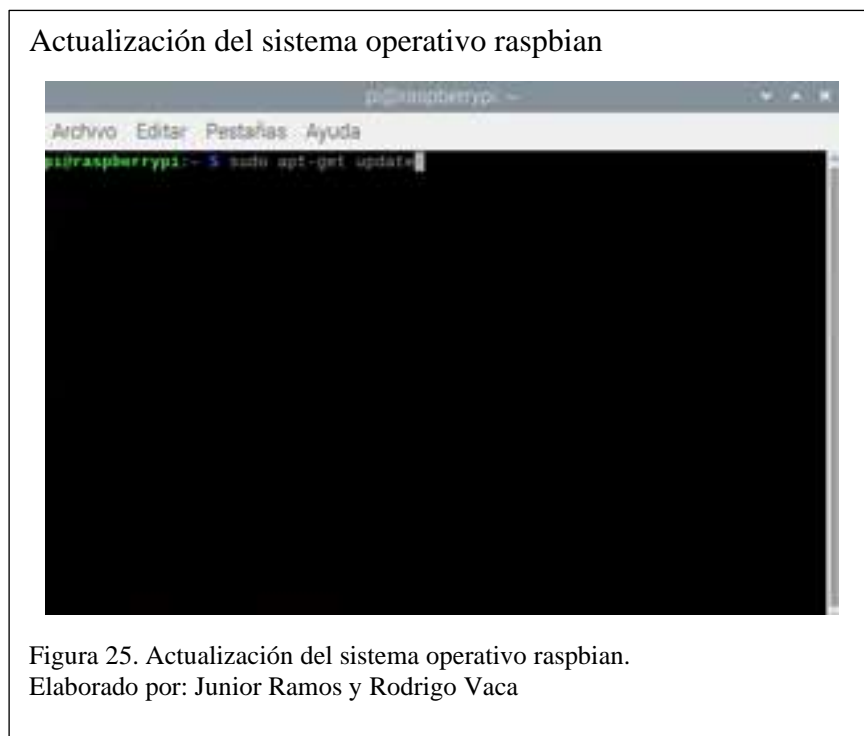
3. Creación del servidor Apache

Para que el usuario pueda hacer uso de la aplicación web, se utilizará un servidor local montado sobre la raspberry pi, para lo cual se utilizará Apache.

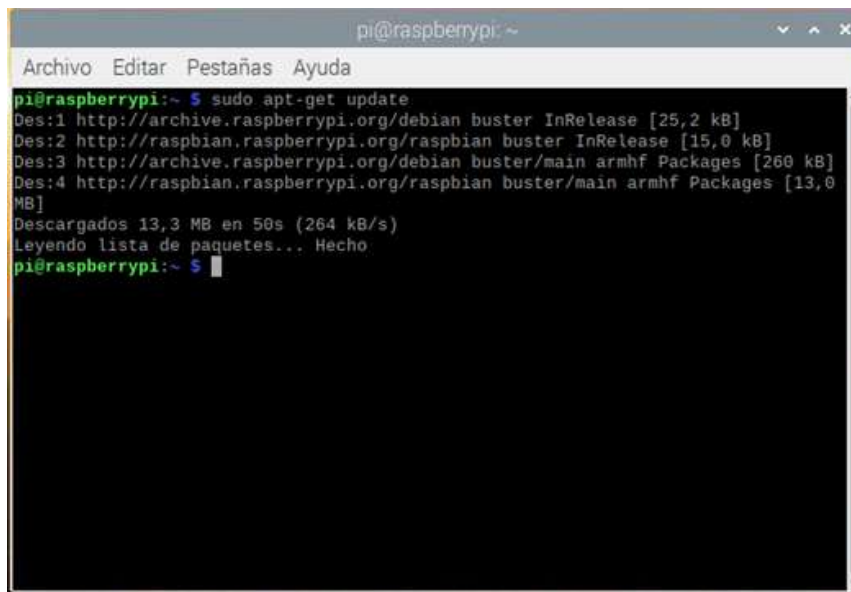
Dentro de Raspbian (Sistema Operativo de la raspberry), antes de montar el servidor Apache se deben actualizar todos los repositorios (para evitar problemas de compatibilidad generados por versiones antiguas).

Para ello se abre la línea de comandos y se digita:

```
$ sudo apt-get update
```



Sistema operativo raspbian actualizado



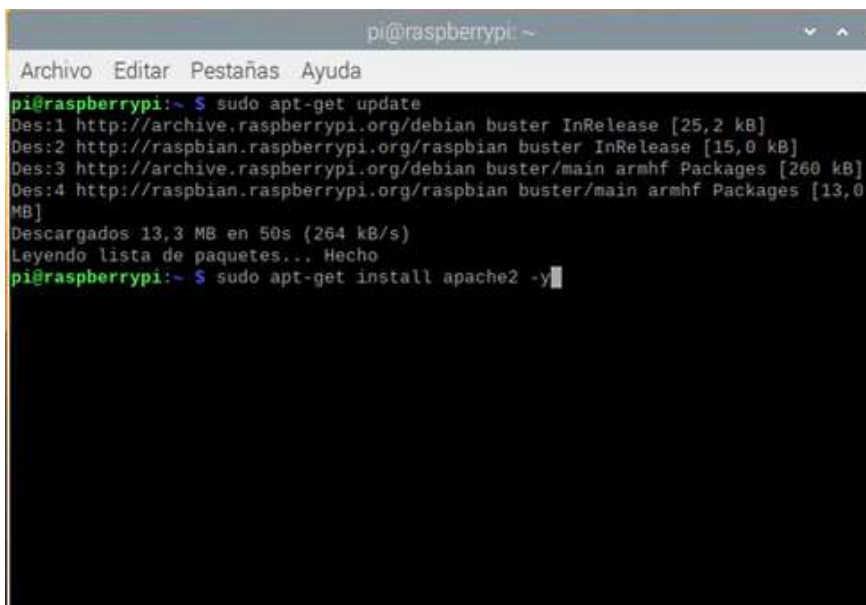
```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo apt-get update  
Des:1 http://archive.raspberrypi.org/debian buster InRelease [25,2 kB]  
Des:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]  
Des:3 http://archive.raspberrypi.org/debian buster/main armhf Packages [260 kB]  
Des:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0  
MB]  
Descargados 13,3 MB en 50s (264 kB/s)  
Leyendo lista de paquetes... Hecho  
pi@raspberrypi:~$
```

Figura 26. Actualización completa del sistema operativo raspbian.
Elaborado por: Junior Ramos y Rodrigo Vaca

Una vez actualizados los repositorios se puede instalar Apache, a continuación, en la misma línea de comandos se digita:

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

Instalación del servidor apache



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo apt-get update  
Des:1 http://archive.raspberrypi.org/debian buster InRelease [25,2 kB]  
Des:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]  
Des:3 http://archive.raspberrypi.org/debian buster/main armhf Packages [260 kB]  
Des:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0  
MB]  
Descargados 13,3 MB en 50s (264 kB/s)  
Leyendo lista de paquetes... Hecho  
pi@raspberrypi:~$ sudo apt-get install apache2 -y
```

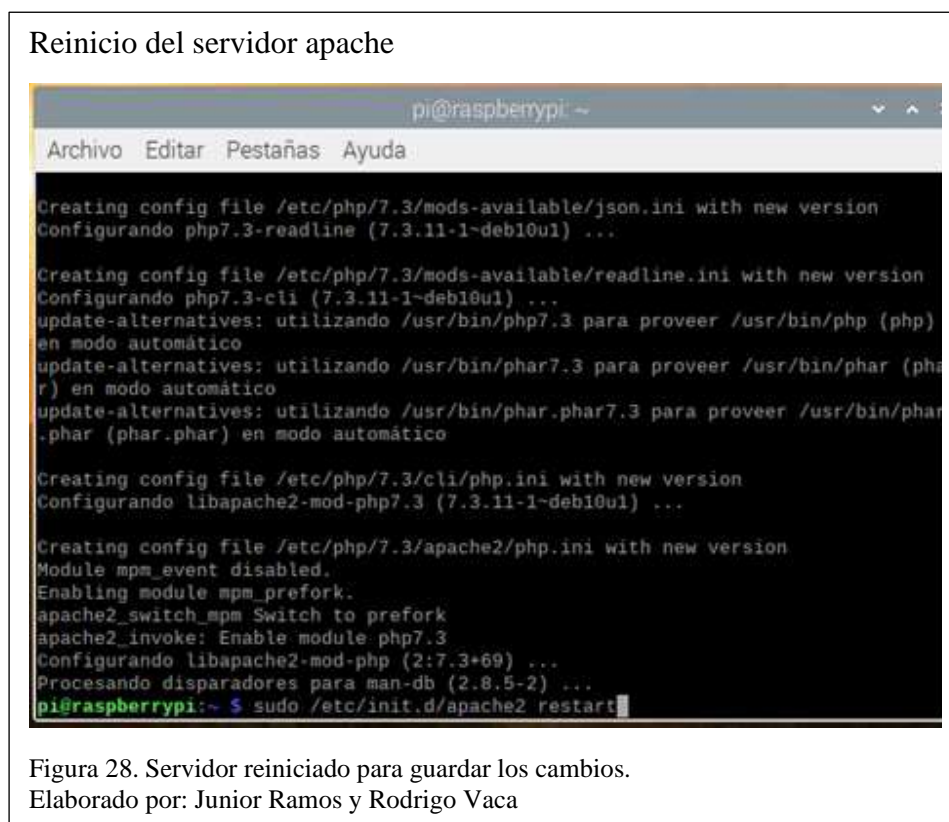
Figura 27. Instalación de apache en la raspberry pi.
Elaborado por: Junior Ramos y Rodrigo Vaca

Luego para conectar Python con la interfaz web, se debe instalar la librería de php, nuevamente en la línea de comandos se digita:

```
$ sudo apt-get install libapache2-mod-php
```

Ahora para que se efectúen correctamente los cambios se debe reiniciar el servidor, nuevamente en la línea de comandos se digita:

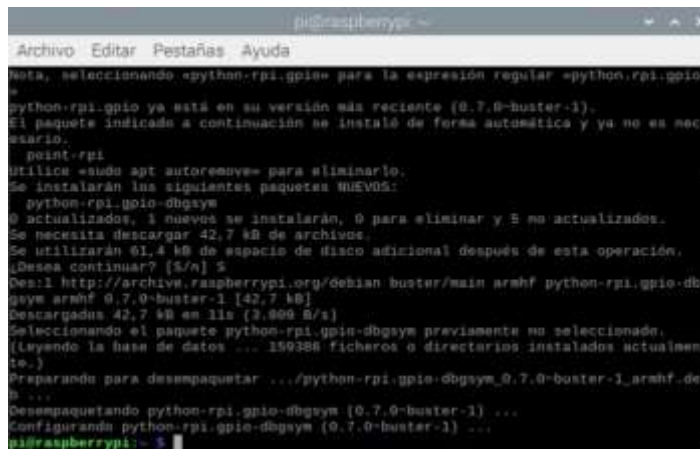
```
$ sudo /etc/init.d/apache2 restart
```



Luego para conectar físicamente Python con la raspberry pi es necesario instalar la librería GPIO (para controlar los pines digitales de entrada y salida). Para esto se digita los siguientes comandos:

```
$ sudo apt-get install python.rpi.gpio
```

Instalación librería GPIO



```
pi@raspberrypi ~$ sudo apt-get install python.rpi.gpio
Nota, seleccionando «python-rpi.gpio» para la expresión regular «python.rpi.gpio»
python-rpi.gpio ya está en su versión más reciente (0.7.0-buster-1).
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
python-rpi
  Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes NUEVOS:
python-rpi.gpio-dbgSYM
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 42,7 kB de archivos.
Se utilizarán 61,4 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [Y/n] Y
Des:1 http://archive.raspberrypi.org/debian buster/main armhf python-rpi.gpio-dbgSYM armhf 0.7.0-buster-1 [42,7 kB]
Descargados 42,7 kB en 11s (3,909 B/s)
Seleccionando el paquete python-rpi.gpio-dbgSYM previamente no seleccionado.
(Leyendo la base de datos ... 159388 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../python-rpi.gpio-dbgSYM_0.7.0-buster-1_armhf.de
0 ...
Desempaquetando python-rpi.gpio-dbgSYM (0.7.0-buster-1) ...
Configurando python-rpi.gpio-dbgSYM (0.7.0-buster-1) ...
pi@raspberrypi:~$
```

Figura 29. Instalación de la librería GPIO.
Elaborado por: Junior Ramos y Rodrigo Vaca

Una vez que se ha instalado el servidor Apache, la aplicación web se encontrará en la dirección /var/www, es decir se alberga en la carpeta www que viene por defecto; y para utilizar todo su contenido sin problemas se le deben dar permisos especiales. Se debe utilizar el comando:

```
$ sudo chmod 775 /var/www/
```

El comando chmod significa “cambio de modo” y 775, significa que se está dando todos los permisos para crear el archivo y todos los permisos para el grupo del usuario principal y para otros usuarios se les está dando permiso de lectura y ejecución del archivo.

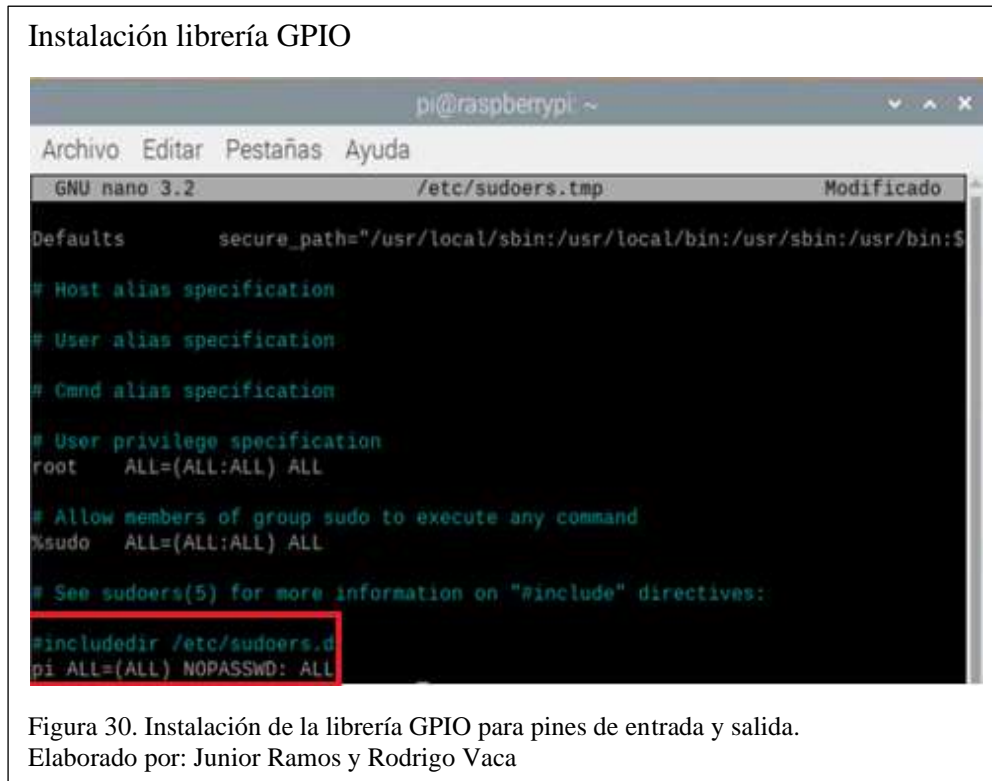
El sistema operativo Raspbian para su acceso tiene por defecto al usuario “**pi**” y la contraseña “**raspberry**”, estos valores no se van a modificar.

Ahora se debe dar permisos al usuario “pi” (que es la propia raspberry) de ejecutar comandos con sudo. Es importante porque php va a hacer la ejecución de Python con privilegios. Para ello en la línea de comandos se digita:

```
$ sudo visudo
```

Lo que despliega el siguiente código, se debe ir al final del código y se debe agregar esta nueva línea.

```
pi ALL=(ALL) NOPASSWD: ALL
```



Luego se debe salir presionando las teclas “ctrl” + “x” y se deben guardar los cambios. Una vez hecho esto los archivos del proyecto se deben guardar en la carpeta **www**.

3.1. Diseño e implementación del circuito

Para el diseño del circuito, primero se deben efectuar las conexiones del sensor y la raspberry con ayuda de un protoboard con el objetivo de medir la distancia de censado de nuestro sensor ultrasónico en el prototipo. El diagrama esquemático es el siguiente y fue realizado con el programa Fritzing.

Diagrama esquemático del circuito

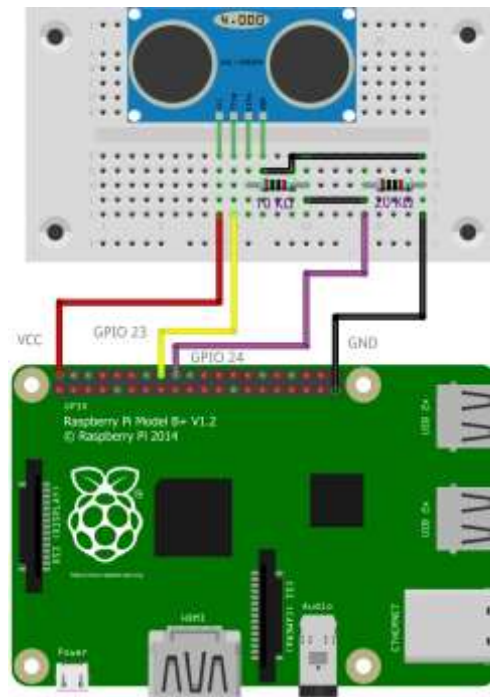


Figura 31. Diagrama esquemático del circuito en el programa fritzing.
Elaborado por: Junior Ramos y Rodrigo Vaca

En el presente circuito tenemos un sensor HCSR04 el cual posee 4 pines, uno funciona para el voltaje de entrada, el segundo es el hecho que sirve para captar la información de retorno, el tercer pin es el trigger que funciona como disparador de la onda mecánica del sensor y el cuarto pin es la tierra. Para realizar la comunicación del sensor a la Raspberry se lo hace con conexiones desde los pines hacia la GPIO (General Port Input/Output) los cuales admiten hasta 3.3 V por este motivo utilizamos un divisor de tensión para poder obtener el voltaje correcto y que nuestra Raspberry no se quemé.

Se utiliza los principios de movimiento rectilíneo uniforme debido a que la velocidad es constante utiliza la velocidad del sonido, utilizando la siguiente fórmula $d=V/t$. El sensor tiene una visión de 120 grados mediante el trigger el sensor dispara las ondas mecánicas al chocar con un objeto la señal retorna hacia el eco y del eco se envía la información hacia la Raspberry.

3.1.1. Fritzing

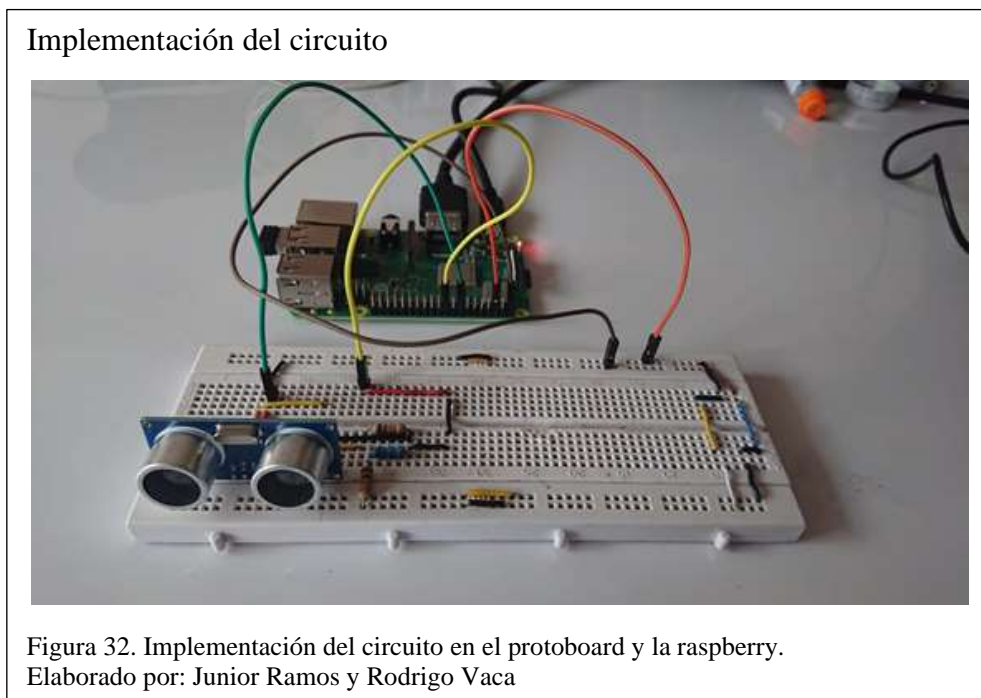
Es un software que funciona automatizando diferentes diseños electrónicos para los diferentes diseñadores de circuitos electrónicos que vean la necesidad de realizar un prototipo más adecuado. El objetivo principal de esta herramienta es brindar las herramientas necesarias para facilitar la documentación ordenada y el mejor entendimiento de un proyecto. Fritzing posee una interfaz de usuario amigable e intuitivo lo que ayuda a agilizar el trabajo (Blazetic, 2011).

3.2. Implementación del circuito

Para la implementación del circuito se utilizó los siguientes materiales:

- Raspberry pi3 b+
- hc-sr04 ultrasónico
- Resistencias
- Protoboard
- Baliza FeasyBeacon
- Sensor GPS Neo 6M
- Batería PowerBank
- Celular con Wifi y Bluetooth

El montaje del circuito se muestra a continuación:



3.3.Prueba de medición del sensor

Prueba del sensor

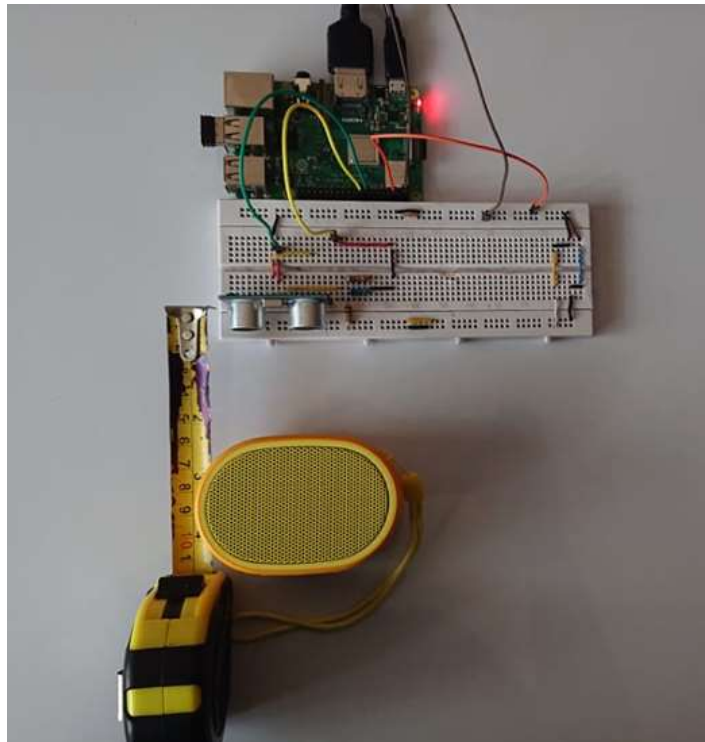


Figura 33. Prueba del sensor censando un determinado objeto.
Elaborado por: Junior Ramos y Rodrigo Vaca

Como se puede observar con el flexómetro la distancia aproximada del parlante hasta el sensor es de 5.5 cm. Para determinar que se encuentre funcionando correctamente se ejecuta el programa “ultrasonido” en Python (ubicado en la carpeta `cd /var/www/rpi-ultrasonido`), el cual muestra una distancia de 5.37 cm, comprobando así que la medición funciona.

Para ello entramos primero en la carpeta con el comando:

```
$ cd /var/www/rpi-ultrasonido
```

Una vez dentro para ejecutar el programa de Python se digita el comando:

```
$ sudo Python python/ultrasonido.py
```

Resultado del sensor

```
pi@raspberrypi:/$ cd /var/www/rpi-ultrasonido
pi@raspberrypi:/var/www/rpi-ultrasonido$ sudo python python/ultrasonido.py
Medicion de la distancia en curso
Espere que el sensor se estabilice
Distancia: 5.37 cm
pi@raspberrypi:/var/www/rpi-ultrasonido$
```

Figura 34. Medición del sensor ultrasónico.
Elaborado por: Junior Ramos y Rodrigo Vaca

Una vez que ya se ha podido medir distancia se va a utilizar este valor para determinar la presencia del vehículo. La distancia se determinará más adelante dependiendo del lugar en donde se ubique el dispositivo.

3.4. Interfaz inicial de la Aplicación web (Esquema funcional)

La siguiente es la interfaz de la página web vista desde un dispositivo móvil, el modelo consta de 3 partes:

3.4.1. Página Principal

Página principal

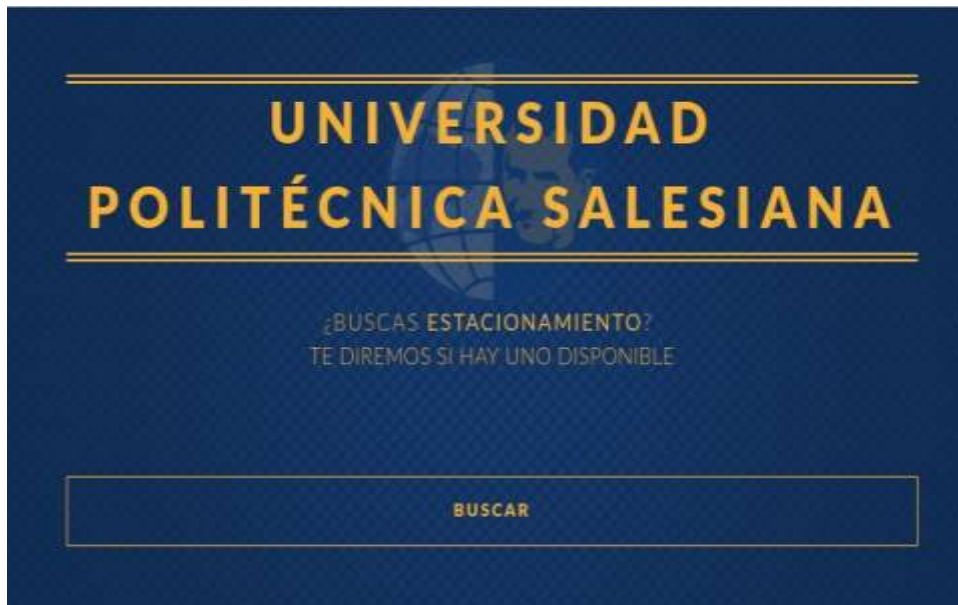


Figura 35. Interfaz web de la página principal.
Elaborado por: Junior Ramos y Rodrigo Vaca

Es la pantalla inicial que se muestra al usuario cuando ingresa a la página web desde nuestro celular.

Aquí se muestra:

- **Cabecera:** Nombre de la Universidad
- **Detalle:** Información simple de lo que hace el programa
- **Botón “Buscar”:** El botón que usará el usuario para encontrar un parqueo disponible.

3.4.2. Información Solicitada

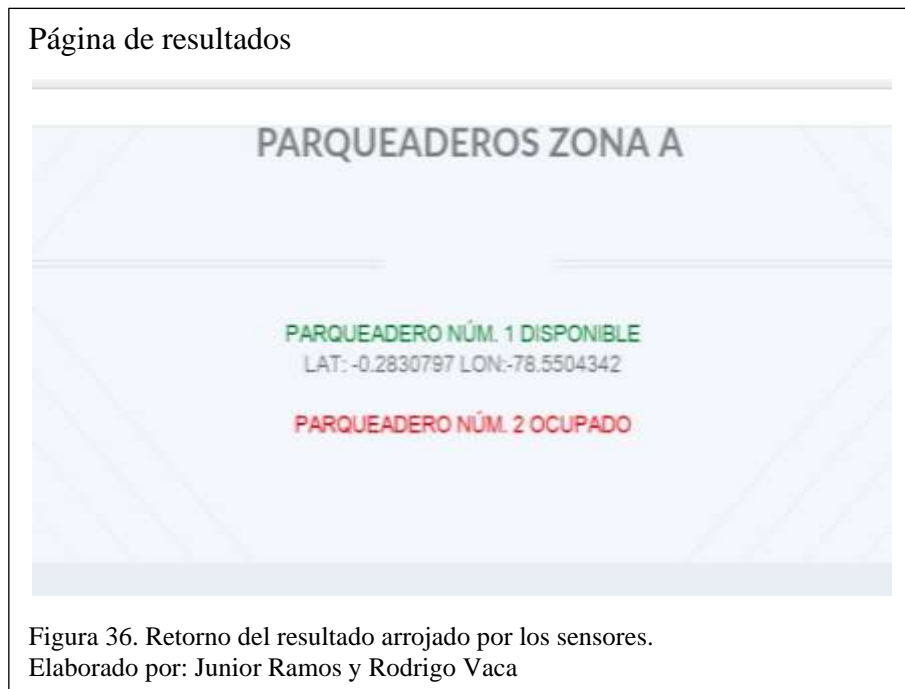


Figura 36. Retorno del resultado arrojado por los sensores.
Elaborado por: Junior Ramos y Rodrigo Vaca

Muestra la información que regresa de los sensores, indicando si existe o no existe un parqueo disponible, además tiene la opción de volver a buscar nuevamente.

3.4.3. Footer

Muestra el nombre del proyecto, nombre de la universidad, y un pequeño perfil de los estudiantes, además de un resumen de la funcionalidad del programa.

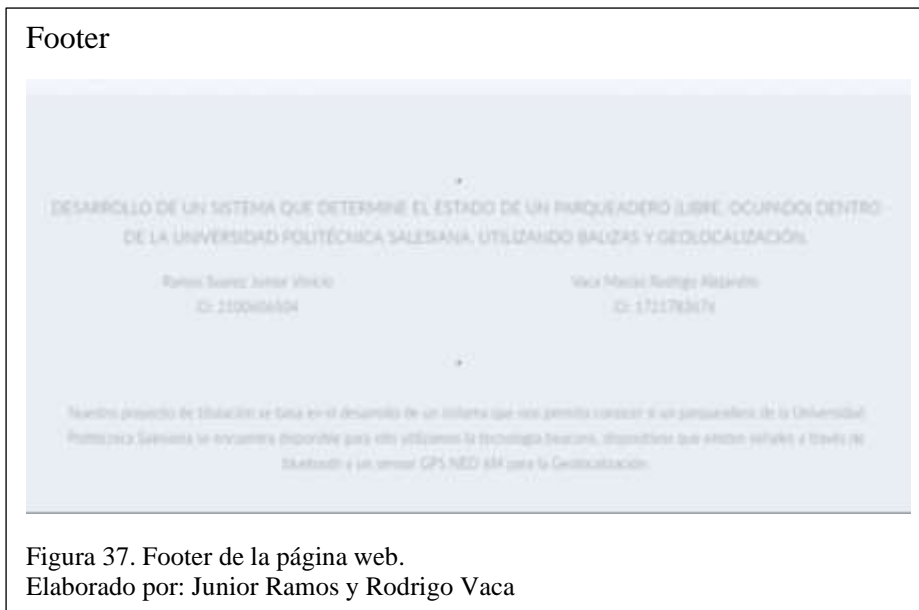


Figura 37. Footer de la página web.
Elaborado por: Junior Ramos y Rodrigo Vaca

3.5.Pruebas

3.5.1. Escenarios de pruebas del prototipo

Se planteó diferentes escenarios de pruebas con diferentes distancias en este caso se realizó con tres escenarios, el escenario de prueba número uno es con una distancia de 15 cm, que medirá desde el punto donde será ubicado el prototipo (parte trasera) hacia adelante como se observa en la Figura 38.

El escenario de prueba número dos es con una distancia de 230cm, de igual manera el punto inicial de medición será el mismo pero el final será en la parte central del vehículo como se observa en la Figura 39. y 40.

El escenario de prueba número tres es con el prototipo apuntando hacia el piso ya no al frente, el punto inicial sería el mismo pero el final sería el piso.

Escenario de prueba 1:

Nombre: Distancia de 15cm.

Descripción: Se puso en marcha el prototipo programado con una distancia de 15cm de referencia hacia el vehículo desde el punto inicial como se observa en la Figura 38., detectando

la presencia del vehículo y desplegando los datos a los usuarios mediante la baliza como se puede observar en la Figura 36.



Podemos observar que la distancia máxima para detectar el vehículo es su parte posterior, siendo este el caso debemos saber que no todos los vehículos se parquearan a esa misma distancia desde el punto inicial. El primer escenario de prueba se logró de manera exitosa.

Escenario de prueba 2:

Nombre: Distancia de 230 cm.

Descripción: Se puso en marcha el prototipo programado a una distancia mayor a la del escenario de prueba 1, en este caso fue de 230 cm de referencia hacia el vehículo desde el punto inicial como se observa en las Figuras 39. y 40., detectando la presencia del vehículo y desplegando los datos a los usuarios a través de la baliza como se observa en la Figura 36.

Escenario de prueba 2



Figura 39. Escenario de prueba 2 del prototipo.
Elaborado por: Junior Ramos y Rodrigo Vaca

Escenario de prueba 2 a 230 cm



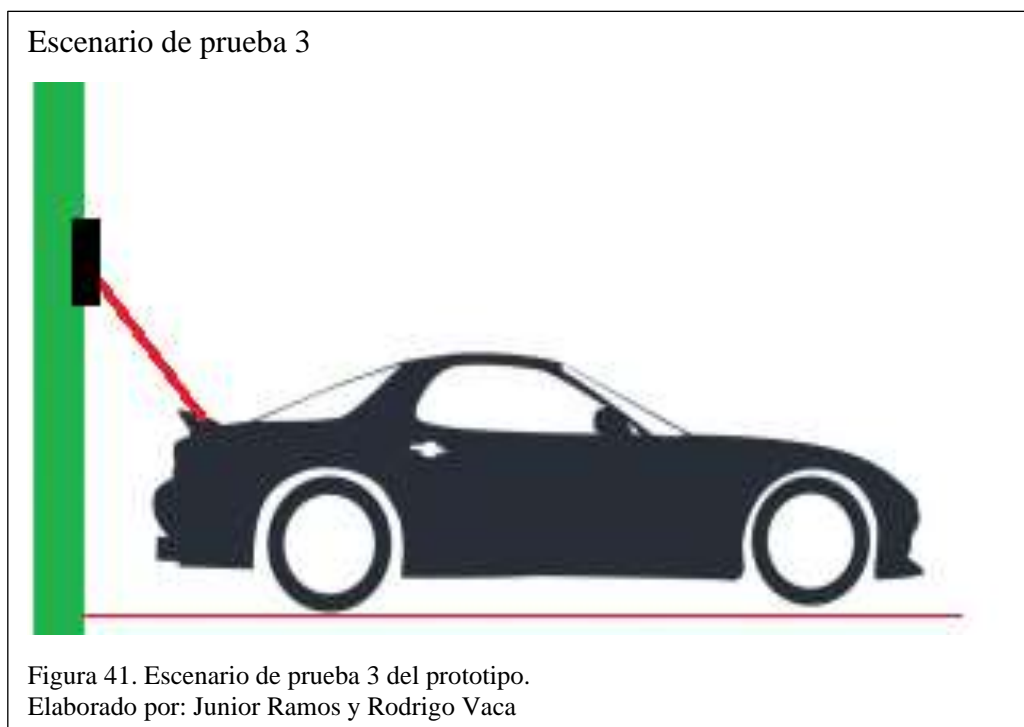
Figura 40. Escenario de prueba 2 del prototipo, distancia de 230 cm.
Elaborado por: Junior Ramos y Rodrigo Vaca

Ahora podemos observar en la Figura 40., que la distancia máxima para determinar la presencia de un vehículo es de 230 cm, en este escenario el margen de error sería mínimo ya que siempre va a detectar la presencia de un vehículo dado que el rango de medición es mayor al del escenario de prueba 1. El escenario número dos de prueba se logró de manera exitosa.

Escenario de prueba 3:

Nombre: Vista al piso

Descripción: Se puso en marcha el prototipo apuntando hacia el piso, se lo programó con una distancia de 80 cm de referencia hacia el vehículo desde el punto inicial como se observa en la Figura 41., detectando la presencia del vehículo y desplegando los datos a los usuarios a través de la baliza como se observa en la Figura 36.



En el caso de prueba 3 debemos tomar en cuenta que los vehículos son de diferentes medidas y no nos permiten determinar una distancia aproximada para el censado. El escenario de prueba número tres se logró de manera exitosa.

3.5.2. Escenarios de pruebas de la baliza

Estos escenarios nos permitirán seleccionar la ubicación más adecuada para la baliza, el escenario de prueba número uno para la ubicación de la baliza es el poste cercano a la entrada como podemos observar en la Figura 42.

El escenario de prueba número dos para la ubicación de la baliza es la caceta del guardia como se observa en la Figura 43.

El escenario de prueba número tres para la ubicación de la baliza sería debajo del puente que está cercano a la entrada como se observa en la Figura 44.

Escenario de prueba 1:

Nombre: Poste

Descripción: Se ejecutó la aplicación FeasyBeacon ubicando la baliza en el poste como se observa en la Figura 42., debemos tomar en consideración que la aplicación debe ser abierta antes o durante el ingreso a los parqueaderos de la UPS.

Escenario de prueba 1 baliza



Figura 42. Escenario de prueba 1 de la baliza.
Elaborado por: Junior Ramos y Rodrigo Vaca

Escenario de prueba 2:

Nombre: Caceta del guardia

Descripción: Se ejecutó la aplicación FeasyBeacon ubicando la baliza en la caceta del guardia como se observa en la Figura 43., pero se determinó que el guardia no se responsabiliza por la pérdida de la misma y no es permitida su ubicación en ese lugar.

Escenario de prueba 2 baliza



Figura 43. Escenario de prueba 2 de la baliza.
Elaborado por: Junior Ramos y Rodrigo Vaca

Escenario de prueba 3:

Nombre: Puente

Descripción: Se ejecutó la aplicación FeasyBeacon ubicando la baliza en el puente cercano a la entrada como se observa en la Figura 44., pero tomando en cuenta el rango de la baliza, la interferencia y que la aplicación debe ser abierta antes o durante la entrada, esta ubicación no es muy adecuada para la baliza.

Escenario de prueba 3 baliza



Figura 44. Escenario de prueba 3 de la baliza.
Elaborado por: Junior Ramos y Rodrigo Vaca

3.6. Resultados y análisis

Se seleccionó el mejor escenario para que el prototipo tenga un mayor desempeño junto con todos los componentes que lo conforman, además de la baliza que es el componente principal

Implementación y descripción del escenario posible, prototipo

Nombre: Escenario 2, distancia de 230cm

Se implementó el escenario de prueba número dos, el cual consiste en ubicar el prototipo a 230cm de distancia como se observa en las Figuras 39. y 40. El escenario 2 es el óptimo para el funcionamiento del prototipo debido a la ubicación de la playa de estacionamientos que nos brinda la posibilidad de ubicar el prototipo de manera segura. Por consiguiente, si el sensor está ubicado a una distancia de 230 cm desde su ubicación, nos permite tener una precisión del 99% para la detección de un vehículo, ya que la medida máxima que detecta el sensor es la mitad del vehículo como se observa en la Figura 39.

Considerando que el módulo se va a instalar a la intemperie y estará sujeto a polvo, lluvia y daños externos se consideró los parámetros de diseño: IP 69, protección contra polvo, agua y altas temperaturas, IK10, protección contra golpes externos que es la principal causa de daño del módulo.

El sensor ultrasónico tiene una sensibilidad de 1 a 3%, es decir cada 5 segundos está censando para detectar la presencia de un vehículo en el parqueadero, si dentro de los 5 segundos alguien se parqueo durante 3 segundos y salió por qué no le gusto sería un caso aislado, es decir el grado de efectividad de un total de 10 casos en los cuales 2 nos ocurrió el caso aislado nos daría un grado de efectividad del sensor del 80%.

El módulo GPS presenta una gran precisión y es muy fácil de utilizar, el sensor se comunica con nuestro prototipo a través de sus puertos seriales y pines Rx, Tx. El GPS tiene una sensibilidad de captura de -148dBm, sensibilidad de rastreo de -161 dBm y su máxima velocidad es de 515 m/s. El módulo nos brinda una muy buena precisión para detectar la ubicación del parqueadero, pero debemos calibrarlo unos minutos ya que es un modelo que es usado con fines educativos y sus recursos son limitados.

Descripción de escenarios no posibles

Los escenarios 1 y 3 no son posibles porque en el escenario 1, los vehículos no siempre se van a parquear a una distancia de 30 cm desde la ubicación del módulo como se observa en la Figura 38., es decir no se tiene un alto grado de precisión de un vehículo debido a la poca distancia para detectar la presencia del mismo, además no todos los usuarios van a parquear su vehículo a una distancia tan corta de 30 cm. El escenario 3 que nos ubica el sensor en un Angulo de 45° apuntando al piso como se observa en la Figura 41., no brinda un alto grado de detección de vehículos debido a que no todos los vehículos tienen el mismo tamaño o modelo, lo cual no permite tener una distancia promedio desde el punto inicial donde se ubicó el prototipo hasta el punto final (detección del vehículo).

Implementación y descripción del escenario posible, baliza

Nombre: Escenario 1, poste

El mejor escenario en el que se pudo trabajar de manera óptima con la baliza es el primer escenario llamado “poste” como se observa en la Figura 42., en el cual se obtuvo excelentes resultados ya que la baliza se encontraba en un punto alto y aislado, en el cual no va a ser detectado por las personas y no va a poder ser movido ni manipulado, esta ubicación es estratégica ya que se encuentra libre y sin obstáculos. Así se logra evitar las interferencias y vamos a poder tener la potencia máxima del dispositivo que transmite la señal por medio de decibeles, lo que nos ayuda a lograr abarcar una mayor área de cobertura para su funcionamiento, también se encuentra cerca de la entrada al parqueadero por lo que los usuarios van a poder abrir la aplicación FeasyBeacon y desplegar la información antes y durante la entrada al parqueadero.

La baliza al estar ubicado en un poste va a requerir métodos de protección para evitar daños como resultado de la ubicación seleccionada, por lo que se va a utilizar una carcasa con la protección IP 69 que es una protección contra agua e IK10 que consiste en proteger al dispositivo de golpes y polvo.

La ubicación seleccionada cuenta con una cámara de seguridad lo que nos va a ayudar a evitar que la baliza sea robada o exista una pérdida de la misma y de igual manera como se encuentra cerca de la caceta del guardia es un lugar muy seguro donde no se van a tener inconvenientes.

En este lugar nuestra baliza va a tener un porcentaje de señal y alcance del 99% ya que no existen edificios u objetos cercanos que puedan realizar alguna interferencia por los motivos descritos este lugar permitió realizar con éxito las pruebas.

Se realizó el servidor localmente para mitigar posibles ataques cibernéticos con información sensible de la universidad por lo que únicamente se puede visualizar la información si el usuario se encuentra dentro de la LAN de la Universidad Politécnica Salesiana Campus Sur.

Se determina que mediante el protocolo propietario de Linux de Hostname se logra generar una URL confiable, válida y reconocible por la baliza de marca FeasyBeacon para lograr desplegar la información sobre los parqueaderos.

Descripción de escenario no posible.

El segundo escenario descrito anteriormente llamado “Caseta del Guardia” como se observa en la Figura 43., en donde se obtuvo dificultades debido a que el guardia no se responsabiliza de la pérdida del dispositivos, además al no encontrarse en un lugar alto puede sufrir diversos daños y se puede obtener muchas interferencias debido a que si se la ubica en ese lugar podría estar constantemente moviéndose lo que disminuiría considerablemente la calidad de señal y deterioraría la integridad del dispositivo ya que se encuentra expuesto a la constante manipulación lo que afectaría su funcionamiento tanto en hardware y software.

CONCLUSIONES

- Las balizas brindan un método de transmisión de datos vía inalámbrica y por ende el dispositivo tiene facilidades de movilidad y ubicación a comparación con los sistemas de cableado
- En el envío y recepción de ondas el tiempo de retardo es vital, debido a que una buena lectura de las ondas permitirá mejores resultados. El sensor ultrasónico ofrece dichos resultados, ya que su tiempo de retardo en cada envío y recepción de ondas es de 10us evitando falsos positivos
- El promedio de búsqueda de un parqueadero usando el método visual es de 7 minutos y usando la aplicación desarrollada el promedio de búsqueda es de 3 minutos, reduciendo así el tiempo de búsqueda y ahorrándole al usuario 4 minutos
- En base a los escenarios estudiados y alcance de las balizas se determinó que para su implementación se podrían utilizar 3 balizas adicionales, una en cada bloque, para brindar una cobertura total del campus y garantizar así que todos los usuarios puedan consultar la disponibilidad de los parqueaderos en cualquier momento
- Con el diseño de la aplicación se logró facilitar al usuario el acceso a los parqueaderos disponibles
- La usabilidad del aplicativo móvil es de fácil asimilación para el usuario y le permitirá conocer el estado del parqueadero de manera intuitiva

RECOMENDACIONES

- Si se desea hacer la aplicación para más parqueaderos debemos considerar un sensor para cada parqueadero formando una red tipo mesh entre todos los controladores, además el aumento de sensores lo que no afectará el funcionamiento del prototipo
- Se recomienda que el servidor en donde van a estar alojados los datos sea un servidor local para evitar ataques y fugas de información
- Se recomienda generar una red categoría C de direccionamiento (su primer segmento se encuentra entre el 192 y el 223 ambos incluidos) probado para que se pueda gestionar de una mejor manera los accesos al parqueadero
- Determinar las condiciones de protección que debe cumplir el encapsulamiento o carcasa del hardware del prototipo como son IK 10, brinda protección contra golpes e IP 69, protección contra polvo, agua y altas temperaturas; para asegurar un óptimo funcionamiento y el mayor tiempo de vida posible

GLOSARIO

Balizas (beacons): Son dispositivos que permiten la comunicación con otros dispositivos móviles a través de bluetooth

App: Aplicación móvil

BLE: (Bluetooth Low Energy), estándar de tecnología de comunicación inalámbrica

GPIO: General Purpose Input Output (Entrada Salida de Propósito General)

Tarjeta SD: Tarjeta de memoria para dispositivos portátiles, SD (Seguro Digital)

mAh: Miliamperio hora, unidad de carga eléctrica

GHz: Gigahercio

Vdc: Voltaje de alimentación

Exo-térmica: “Produce energía que se aprovecha cuando se ha agotado la batería” (Ordoñez, 2016)

USB: Bus universal en serie que proveen alimentación eléctrica a dispositivos electrónicos

Booteable: Es una USB de arranque

PowerBank: Batería para cargar dispositivos móviles

FeasyBeacon: Aplicación propia de la baliza compatible con bluetooth, además muestra información de balizas cercanas

HTML: “Es un estándar para definir la estructura y el contenido de una página Web” (Diego, 2019)

GPS: Sistema de posicionamiento global

Url: (localizador de recursos uniforme), dirección de internet que muestra información a un usuario

Ups: Universidad Politécnica Salesiana

dB: Decibelios, unidad de medida entre tensión y potencia eléctrica

Rufus: Aplicación para crear USB de arranque

LAN: Red de área local

GPIO (General Purpose Input Output): Pines genéricos (entrada/salida) que puede ser controlado por el usuario cuando se ejecuta

LISTA DE REFERENCIAS

- Agelet Aumedes, D. (17 de Enero de 2019). *Diseño y construcción de un prototipo para el control de presencia con Raspberry Pi*. Obtenido de Universidad Politécnica de Cataluña: <https://upcommons.upc.edu/handle/2117/130920>
- Blazetic, G. O. (2011). *FRITZING PRIMEROS PASOS*. Obtenido de Fritzing.org: <http://fritzing.org/media/uploads/learning/translations/Fritzing-PrimerosPasos.pdf>
- Caicedo, C. C., & Camargo, Q. L. (22 de Mayo de 2019). *Prototipo de aplicación móvil en android para optimizar el uso del tiempo en el reporte de visitas en el proyecto de atención a la primera infancia en condición de vulnerabilidad del Municipio del Tambo - Cauca*. Obtenido de Universidad Distrital Francisco Jose de Caldas: <http://hdl.handle.net/11349/15790>
- Desarrollo Web. (02 de Febrero de 2015). *Desarrollo Web*. Obtenido de Tabla comparativa de los lenguajes de programación: <http://desarrollowebbydesarrolloweb.blogspot.com/2015/02/tabla-comparativa-de-los-lenguajes-de.html>
- Diego, P. J. (20 de Enero de 2019). *Qué es HTML5*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-html5/>
- Electrónica. (25 de Enero de 2015). *paletosdelaelectronica wordpress*. Obtenido de Paletos de la Electrónica: <https://paletosdelaelectronica.wordpress.com/2015/01/25/interruptores-fin-de-carrera-i/>
- Fernandez, A. G. (Octubre de 2005). *Sensores Magnéticos e Inductivos*. Obtenido de Universidad Autonoma del Estado de Hidalgo: [http://dgsa.uaeh.edu.mx:8080/bibliotecadigital/bitstream/handle/231104/169/Sensores %20magneticos.pdf?sequence=1](http://dgsa.uaeh.edu.mx:8080/bibliotecadigital/bitstream/handle/231104/169/Sensores%20magneticos.pdf?sequence=1)
- García, Tello, Mata, & Dorantes. (Junio de 2018). *GEOLocalización Inteligente Implementando Tecnología Beacons en Dispositivos Móviles*. Obtenido de Academia Tamaulipeca de Investigación Científica y Tecnológica, A.C.: <http://www.atictac.org.mx/pdf/Revista%20Num%201%20Vol%202%20ene-jun%202018.pdf#page=46>
- Gonzalez, H., & Velásquez, S. .. (29 de Mayo de 2019). *CONTROL Y ADQUISICIÓN DE VARIABLES PARA UN ENTORNO DE TRABAJO*. Obtenido de Universidad Nacional Experimental Politécnica Antonio José de Sucre: http://scholar.googleusercontent.com/scholar?q=cache:4Lhtgv3VW1MJ:scholar.google.com/+m%C3%B3dulo+bluetooth+hc-05+que+es&hl=es&lr=lang_es&as_sdt=0,5&as_ylo=2018
- HAREENDRAN, T. (04 de Abril de 2019). *Módulo GPS NEO-6M - Una introducción*. Obtenido de ElectroSchematics: <https://www.electroschematics.com/neo-6m-gps-module/>

- Hernández, L. d. (2019). *Sensor ultrasónico, mideindo la distancia en un sistema de aparcamiento*. Obtenido de Programar Facil: <https://programarfacil.com/blog/arduino-blog/sensor-ultrasonico-arduino-medir-distancia/>
- Imas, G. J. (27 de Junio de 2017). *Proyectos electrónicos con microcontrolador PIC16F877A*. Obtenido de Universidad Pública de Navarra: <https://academica-e.unavarra.es/bitstream/handle/2454/24597/Proyectos%20electr%2B%C2%A6nicos%20con%20microcontrolador%20PIC16F877A.pdf?sequence=1&isAllowed=y>
- Kappta. (14 de Diciembre de 2016). *Beacons en la Universidad de Oklahoma*. Obtenido de Blog kappta: <https://blog.kappta.com/beacons-en-la-universidad-de-oklahoma/>
- Macías Olives, J. J. (Junio de 2018). *DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE CONTROL Y MONITOREO DE PROCESOS CON SISTEMAS EMBEBIDOS ARDUINO Y RASPBERRY PI PARA PYMES*. Obtenido de ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO: <http://dspace.esPOCH.edu.ec/handle/123456789/9213>
- Martínez, Á. J. (06 de Junio de 2019). *Diseño e implementación de una placa entrenadora/programadora para microcontroladores PIC*. Obtenido de UNIVERSIDAD POLITÉCNICA DE CARTAGENA: <http://hdl.handle.net/10317/7900>
- Ordoñez, J. A. (19 de Octubre de 2016). *Baterías de Litio/Litio-Ion/Li-Ion...Que es?* Obtenido de Bateriasdelitio.net: <https://www.bateriasdelitio.net/?p=6>
- Páez, & Bedoya. (17 de Enero de 2017). *Desarrollo de Aplicaciones de “In-door Marketing” utilizando Beacons BLE: Un reporte de Experiencia*. Obtenido de Universidad Piloto de Colombia: <http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/1119/00003472.pdf?sequence=1>
- Palacios, G. M. (2015). *MODELAMIENTO DE UN SISTEMA DE GESTIÓN PARA LA SOCIEDAD HOTELERA IRPE. S.A.C. – TUMBES, 2015*. Obtenido de UNIVERSIDAD CATÓLICA LOS ÁNGELES DECHIMBOTE: http://repositorio.uladech.edu.pe/bitstream/handle/123456789/296/PALACIOS_GONZAGA_MAXIMO_JUNIOR_MODELAMIENTO_SISTEMA_HOTELERIA.pdf?sequence=1
- Pastor, J. (25 de Abril de 2018). *Raspberry Pi 3 Model B+, análisis: más potencia y mejor WiFi para un miniPC que sigue asombrando*. Obtenido de xataka: <https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando>
- Ramos, M. R., & Montoya, L. L. (2019). *CONTROL ELECTRÓNICO PARA ENCENDIDO/BLOQUEO Y SEGURIDAD DE UN VEHÍCULO*. Obtenido de UNIVERSIDAD TECNOLÓGICA ISRAEL : <https://repositorio.uisrael.edu.ec/handle/47000/1905>
- Raspberrypi. (2019). *Raspbian*. Obtenido de raspberrypi.org: <https://www.raspberrypi.org/downloads/raspbian/>

- Rosas Ganán, F. M. (Abril de 2019). *Análisis y desarrollo de una aplicación móvil android para el ingreso y control de actividades de visitas a clientes de la Compañía I.I.A.S.A ubicada en la ciudad de Guayaquil*. Obtenido de Universidad de Guayaquil: <http://repositorio.ug.edu.ec/handle/redug/40115>
- Ryte. (2019). *Beacon*. Obtenido de Ryte Wiki: <https://es.ryte.com/wiki/Beacon>
- TBem. (03 de Mayo de 2017). *¿Cómo usar una LiPo?* Obtenido de teslabem: <https://teslabem.com/blog/como-usar-y-cuidar-las-baterias-lipo/>
- Using Beacons. (25 de Mayo de 2016). *Tecnología beacon en los aeropuertos de Madrid y Barcelona*. Obtenido de Using Beacons: <http://www.usingbeacons.com/tecnologia-beacon-en-los-aeropuertos-de-madrid-y-barcelona/>
- Veloso, C. (19 de Junio de 2018). *ARDUINO MEGA 2560 CARACTERÍSTICAS*. Obtenido de Electrónica analógica y digital: <https://www.electrontools.com/Home/WP/2018/06/19/arduino-mega-2560-caracteristicas/>
- Wendling, M. (2010). *Sensores*. Obtenido de Universidad Estatal Paulista: <https://paulocrgomes.com.br/wp-content/uploads/2018/11/sensores.pdf>