UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA:

INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título:

Ingenieros de Sistemas

TEMA:

DESARROLLO DE UNA SOLUCIÓN DE HARDWARE Y SOFTWARE PARA EL CONTROL AUTOMÁTICO DE PUERTAS DE GARAJE.

AUTORES:

CARLOS DAVID POSSO JÁCOME

BRAYAN LEONARDO RAMÍREZ MERINO

TUTOR:

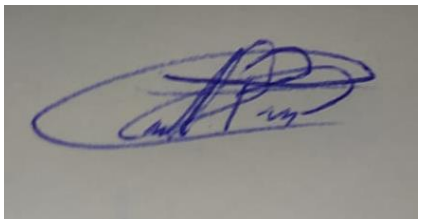
MANUEL RAFAEL JAYA DUCHE

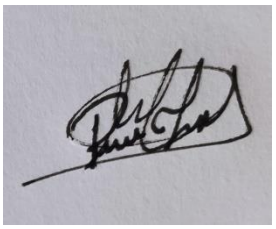
Quito, febrero 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros, POSSO JÁCOME CARLOS DAVID, con documento de identificación N.º 1715709414, y RAMÍREZ MERINO BRAYAN LEONARDO, con documento de identificación N.º 1105535627, 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 grado titulado DESARROLLO DE UNA SOLUCIÓN DE HARDWARE Y SOFTWARE PARA EL CONTROL AUTOMÁTICO DE PUERTAS DE GARAJE. 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 la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.





---------- ----------

Carlos David Posso Jácome

1715709414

Quito, febrero del 2021

Bryan Leonardo Ramírez Merino

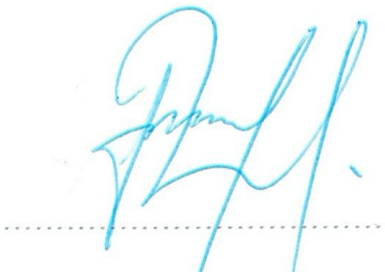
1105535627

Quito, febrero del 2021

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico titulado DESARROLLO DE UNA SOLUCIÓN DE HARDWARE Y SOFTWARE PARA EL CONTROL AUTOMÁTICO DE PUERTAS DE GARAJE. realizado por Carlos David Posso Jácome y Brayan Leonardo Ramírez Merino, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, febrero 2021



Manuel Rafael Jaya Duche

CI: 1710631035

DEDICATORIA

"Se lo dedico a Dios por la oportunidad brindada, a mis padres Amparo y Carlos que, gracias a su amor, esfuerzo y apoyo, me han ayudado a salir adelante y permitido estudiar en esta prestigiosa institución, y a todas las personas que me motivaron a seguir adelante y esforzarme para dar más de mi en momentos difíciles."

Carlos David Posso Jácome

"Primero a Dios por brindarme la oportunidad de vivir, a mis padres Alonso y Margarita a quienes admiro por todo el esfuerzo y sacrificio que han puesto para salir adelante, por su paciencia y apoyo incondicional para darme la oportunidad de estudiar en una prestigiosa institución, a mis hermanos Jorge, Anabel y Valeria quienes me dan su cariño, y me motivan a seguir adelante, a mi abuelita Celsita quien ha sido un pilar fundamental en mi formación, además de transmitirme su sabiduría y consejos para llegar hacer una mejor persona."

Brayan Leonardo Ramírez Merino

AGRADECIMIENTO

"A Dios por la oportunidad que nos da de vivir, a nuestros padres quienes nos dan su amor, su comprensión, su paciencia, su apoyo para poder seguir estudiando, además de ser el motor y la motivación para salir adelante, a nuestros hermanos los cuales nos brindan su apoyo y su cariño, a nuestro tutor de tesis Ing. Manuel Rafael Jaya Duche quien aporta con sus conocimientos para la realización de la tesis y a todos quienes han sido nuestros ingenieros los cuales han aportado para la culminación de nuestros estudios."

Carlos David Posso Jácome

Brayan Leonardo Ramírez Merino

ÍNDICE

ÍNDICE FIGURAS

ÍNDICE TABLAS

Resumen

El presente proyecto tiene como objetivo diseñar una aplicación multiplataforma que controle una puerta de garaje, a través de la construcción de un circuito que permita una conexión Wifi utilizando smartphones.

Para dar solución al proyecto mediante la metodología Proceso Unificado Ágil, se utilizaron en el circuito un NodeMCU y un relé optoacoplador para conectar el motor de la puerta de garaje a la red wifi a la que pertenezca.

El NodeMCU fue programado con varios scripts los cuales permiten la interacción mediante señales eléctricas entre el motor del garaje y la aplicación móvil.

La solución de software fue diseñada en una interfaz amigable para el usuario mediante programación en bloques a través de la plataforma AppInventor y Thunkable, de esta forma se puede acceder a la aplicación en cualquier smartphone con sistema operativo Android o iOS.

Los resultados están generados en base a varias pruebas elaboradas al sistema, se realizó una encuesta de satisfacción para mejorar la aplicación con respecto a la opinión del usuario obteniendo un 90% de satisfacción en el manejo del prototipo. Se probó la aplicación en las diferentes versiones de sistemas operativos de iOS y Android, las versiones más recientes no presentan problemas, pero las versiones más antiguas impiden su instalación o su funcionamiento es defectuoso. Durante un mes se realizaron pruebas de funcionalidad y efectividad con resultados del 96.77% de funcionalidad y 98% de efectividad.

Una vez implementada la solución permite un ahorro económico del 53.125% en comparación a precios en el mercado actual del país, además gracias a la aplicación móvil amigable al usuario y multiplataforma, permite que cualquier usuario con un smartphone sea capaz de interactuar con el sistema.

Abstract

The objective of this project is to design a multiplatform application that controls a garage door, through the construction of a circuit that allows a WiFi connection using smartphones.

To give a solve the project through the Agile Unified Process methodology, a Node MCU and an optocoupler relay were used in the circuit to connect the garage door motor to the Wi-Fi network to which it belongs.

The Node MCU was programmed with various scripts which allow interaction through electrical signals between the garage motor and the mobile application.

The software solution was designed in a user-friendly interface through block programming through the AppInventor and Thunkable platform, in this way the application can be accessed on any smartphone with an Android or iOS operating system.

The results are generated based on several tests made to the system, a satisfaction survey was carried out to improve the application with respect to the user's opinion, obtaining a 90% satisfaction in handling the prototype. The application was tested on the different versions of iOS and Android operating systems, the most recent versions do not present problems, but the older versions prevent its installation or its operation is defective. During a month, tests of functionality and effectiveness were carried out with results of 96.77% functionality and 98% effectiveness.

Once the solution is implemented, it allows an economic saving of 53.125% compared to prices in the current market in the country, in addition, thanks to the user-friendly and multiplatform mobile application, it allows any user with a smartphone to be able to interact with the system.

INTRODUCCIÓN

Antecedentes

Durante los últimos años el avance tecnológico ha dado un salto significativo, en la actualidad la mayoría de equipos electrónicos poseen una conexión a internet. Con ayuda de la tecnología wifi surgen nuevas investigaciones relacionados con domótica o también conocido como el internet de las cosas (IoT). (Hernández Rodríguez & Guzmán Hernández, 2018) Por lo tanto, se buscan soluciones económicas que permitan la interacción en tiempo real a través de la red con objetos cotidianos que son utilizadas a diario para brindar una mejor experiencia al usuario. (Loyola Mendoza, 2018)

Problemática

Según un análisis realizado por la Universidad Central del Ecuador (Andrade Quizhpe, 2019) solo en el 2017 se han reportado 14544 robos a domicilio en la ciudad de Quito, una de las posibles causas es la inseguridad de cerraduras y extravió de controles en puertas de garaje automáticas. Reforzar la seguridad con guardianía y reponer los controles extraviados resulta muy costoso, y aunque existe en la actualidad soluciones que permitan que los garajes sean Smart estas son soluciones integradas, con las cuales los usuarios deberán reemplazar sus motores de puertas garaje, esto genera un gasto muy alto que no todos se pueden permitir.

Justificación

Los usuarios generalmente utilizan sus smartphones para realizar casi cualquier tarea, así como también para controlar dispositivos electrónicos mediante el internet de las cosas, esto brinda una mejor experiencia y una perfecta opción para reducir costos.

En el presente trabajo, se propone desarrollar una solución mediante hardware de baja costo y accesible a cualquier usuario para que las puertas de los garajes sean Smart. El cual se adaptará al motor ya existente y no se realiza una inversión en un motor nuevo, solo se actualizaría para un mejor rendimiento.

Además, se va a desarrollar una aplicación móvil multiplataforma la cual permitirá al usuario final controlar la puerta de su garaje desde la comodidad de su smartphone.

El uso de la aplicación móvil reducirá el costo por perdida de controles de garaje, además, que en una familia podrán usar una sola aplicación en lugar de una serie de controles.

Desarrollo de una solución mediante un prototipo de puertas de garaje que contengan un motor estándar, un NodeMCU el cual se conectara al punto de acceso para brindar conexión.

Realizado el prototipo se va a utilizar AppInventor y Thunkable para el desarrollo de la aplicación móvil multiplataforma. La aplicación móvil permitirá ofrecer una interfaz sencilla y fácil de usar por el usuario, además de generar una solución más factible para que el usuario desde la comodidad de su smartphone controle la puerta de garaje.

Objetivos

Objetivo General

Desarrollar una solución por hardware para la conexión del motor de las puertas de garaje hacia la red y una solución por software con una aplicación móvil multiplataforma para el control automático de las mismas.

Objetivos Específicos

Levantar la información de los requerimientos para el desarrollo de la aplicación móvil multiplataforma.

Analizar la infraestructura y funcionamiento del motor de las puertas automáticas para generar el prototipo.

Comparar herramientas de código libre, con sus principales características para desarrollo de la aplicación móvil y componentes de hardware con sus principales características para elegir el que este acorde a nuestras necesidades

Adaptar un NodeMCU en la infraestructura del prototipo de puertas de garaje para tener una conexión hacia la aplicación móvil.

Desarrollar aplicación móvil multiplataforma que permita la identificación del usuario correspondiente para el control de las puertas de garaje, además de brindar un nivel de seguridad optimo encriptando la contraseña del usuario para ofrecer un mayor grado de confidencialidad.

Optimizar la aplicación realizando una serie de pruebas para la corrección y mejora de esta.

METODOLOGÍA

La metodología del proyecto será el Proceso Unificado Ágil, esta metodología fue desarrollada por Scott Ambler, para describir una aproximación al desarrollo de aplicaciones combinando conceptos propios del proceso unificado tradicional con técnicas ágiles, mejorando la productividad. (Torrecilla, 2020).

El Proceso Unificado Ágil comprende de cuatro fases:

La primera fase se identificará el alcance del proyecto, como es un prototipo identificar la arquitectura para el sistema de las puertas de garaje y la financiación del proyecto.

En la segunda fase se identificará y validará la arquitectura que va a utilizar la aplicación.

En la tercera fase mediante programación permite un desarrollo incremental el cual mediante pasa el tiempo se puede ir agregando nuevas funcionalidades priorizando las necesidades de los desarrolladores.

En la cuarta fase del Proceso Unificado Ágil, una vez que la aplicación móvil se ha desarrollado tendrá que ser validada por una serie de pruebas en diferentes entornos ya que al ser multiplataforma permite que el entorno de la aplicación funcione en diferentes sistemas operativos, además de que las pruebas permiten realizar correcciones y futuras mejoras. Después de realizar estas pruebas se verificará técnicamente que la aplicación puede salir a producción y que los errores provocados en las pruebas fueron corregidos.

CAPÍTULO I
MARCO TEÓRICO

En el primer capítulo se revisará variada información importante entre características, conceptos y definiciones, para entrar en contexto sobre las soluciones presentadas en el proyecto.

1.1 WLAN (Wireless Local Area Network)

Una red de área local inalámbrica (WLAN) es una variante de la red de área local (LAN) la cual permite la conexión de varias computadoras que se encuentran a una distancia corta además de que transmiten señales de alta frecuencia por la cual trasmiten y reciben datos.

Esencialmente WLAN son utilizadas para comunicaciones dentro de un área pequeña que transmitan datos de punto a punto enlazando un conjunto de redes LAN y WLAN. (bibing, 2020)

1.1.1 Elementos

- ✓ Cliente inalámbrico
- ✓ Punto de acceso (AP)
- ✓ El área de cobertura también es esencialmente un Conjunto de servicios básicos (BSS)
- ✓ El servicio de distribución (DS) es la red cableada que proporciona conectividad desde la WLAN al resto del mundo (bibing, 2020)

1.1.2 Características

- ✓ **Movilidad:** transmite información dentro del área de cobertura de la red en tiempo real entre los usuarios.
- ✓ **Facilidad de instalación:** al ser una red inalámbrica facilita la instalación debido a que no dispone de medios alámbricos para su funcionamiento, reduciendo así el tiempo de instalación y de brindar acceso a la red tanto a usuarios permanentes o temporales.
- ✓ **Flexibilidad:** Puede hacer llegar la conexión a lugares por donde sea una dificultad colocar medios cableados los cuales podrían resultar en una red costosa. (bibing, 2020)

1.1.3 Bandas de frecuencia

Las WLAN trabajan en frecuencias que comprenden los 902-908 MHZ, 2400 -24835 GHz y 5725- 5859 GHz. Las bandas que utiliza las WLANS se denominan de uso común ya que pueden ser usadas sin licencia. Al usar estas bandas estarán expuestas a interferencias externas y no tendrán la posibilidad de interactuar con aplicaciones licenciadas. (bibing, 2020)

1.1.4 Seguridad

Las redes WLAN al ser una red inalámbrica tiene un gran problema con la seguridad ya que podrían realizar ataques interceptando el tráfico que se envía por la red inalámbrica si no se toman las medidas de seguridad. (bibing, 2020)

1.1.5 Aplicaciones

- ✓ Escenario residencial.
- ✓ Redes corporativas.
- ✓ Acceso público a internet.
- ✓ Redes libres.
- ✓ Hostpots.
- ✓ Acceso a internet desde medios de transporte públicos.
- ✓ Interconexión de redes de área local.
- ✓ Acceso de banda ancha en entornos rulares.
- ✓ Otros usos corporativos e industriales. (bibing, 2020)

En la tabla 1, se puede apreciar una comparativa entre las distintas características que poseen las redes WLAN y LAN.

Tabla 1. Tabla comparativa entre WLAN y LAN

Aspecto	WLAN	LAN cableada
Velocidad de transmisión	11-54Mbps	100/1000Mbps
Costes de instalación	Bajo	Alto
Movilidad	Si	No
Flexibilidad	Muy alta	Baja
Escalabilidad	Alta	Muy alta
Seguridad	Media	Alta
Demanda	Alta	Muy alta
Configuración e instalación	Fácil	Compleja
Presencia en empresas	Media	Alta
Coste de expansión	Bajo	Alto
Licencia	No regulado	No

Nota: Comparación entre LAN y WLAN cableada

Fuente: (bibing, 2020)

Se observa en la tabla 2, las diferentes tecnologia que pueden ser usadas en una red WLAN con sus caracteristicas más puntuales.

Tabla 2. Tecnología WLAN

Origen	Estándar	Frecuencia	Nivel físico	Velocidad Máxima	Disponibilidad
IEEE	802.11b	2,4 GHz	DSSS	11 Mbps	2001
IEEE	802.11a	5 GHz	OFDM	54 Mbps	2002
IEEE	802.11g	2,4 GHz	OFDM/DSSS	54 Mbps	2002
ETSI	Hiperlan2	5 GHz	OFDM	54 Mbps	2003
IEEE/ETSI	5 GHz Unified Protocol	5 GHz	OFDM	108 Mbps	2003
Bluetooth SIG	Bluetooth	2,4 GHz	OFDM/FHSS	0,721 Mbps	2002

Nota: Estandarización y normalización de la tecnología WLAN

Fuente: (bibing, 2020)

1.2 ESP8266

Anteriormente Arduino comenzó como un simple proyecto utilizado para la educación y makers. Mediante el software libre Arduino IDE se podría programar una serie de aplicaciones. Con el paso del tiempo esta placa fue evolucionando a la cual se le agregaron nuevos módulos y funcionalidades.

Pero el complemento más importante de esta placa fue la integración de un módulo WIFI con el que nació el ESP8266, ya que al tener salida a la red permitió que varias aplicaciones se conectaran a una red para de esta manera permitir monitorizar y gestionar los proyectos desde cualquier parte del mundo. (Hardwarelibre, 2020)

1.2.1 Historia

El chip ESP8266 Espressif fue la primera placa creada por una empresa china. La cual se comenzó a hacer muy popular por su bajo costo y las capacidades que ofrecía.

La comunidad de desarrolladores hizo grandes aportes, publicando documentación acerca del ESP8266, así como también firmwares y otros códigos que permitían interactuar con el ESP8266. (CEJA, RENTERIA, RUELAS, & OCHOA, 2020)

1.2.2 Definición

El ESP8266 es una placa de bajo costo con una pila de los protocolos TPC/IP y un microcontrolador. Esta placa permite trabajar con el estándar IEEE 802.11 b/g/n y puede incorporar WPA, WPA2 y WEP como seguridad. (Hardwarelibre, 2020)

1.2.3 Características

- ✓ CPU Tensilica Xtensa L106 32-bit RISC 80Mhz
- ✓ Conversor ADC 10-bit
- ✓ RAM 64 KB i / 96 KB d
- ✓ GPIO 16 pines (no se pueden usar todos, además el GPIO16 está conectado al RTC o Real Time Clock)
- ✓ UART
- ✓ SPI
- ✓ I2C
- ✓ Voltaje 3v y 3.6v
- ✓ Intensidad 80mA
- ✓ Temperatura de operación -40 a 125ºC
- ✓ WiFi IEEE 802.11 b/g/n con soporte IPv4 y protocolos TCP/UDP/HTTP/HTTPS/FTP
- ✓ Consumo 0.0005 a 170 mA según la potencia de señal
- ✓ Modos: Active mode, Sleep Mode, Deep Sleep – Afectan al consumo

(CEJA, RENTERIA, RUELAS, & OCHOA, 2020)

1.2.4 Usos

El módulo ESP8266 al poseer conexión a internet permite realizar una serie de aplicaciones de bajo costo. Entre la más importante se lo puede utilizar para aplicaciones de domótica relacionados con proyectos de internet de las cosas (IoT). (CEJA, RENTERIA, RUELAS, &

OCHOA, 2020) En la figura 1, se puede apreciar la arquitectura de NodeMCU y su chip ESP8266.

Arquitectura

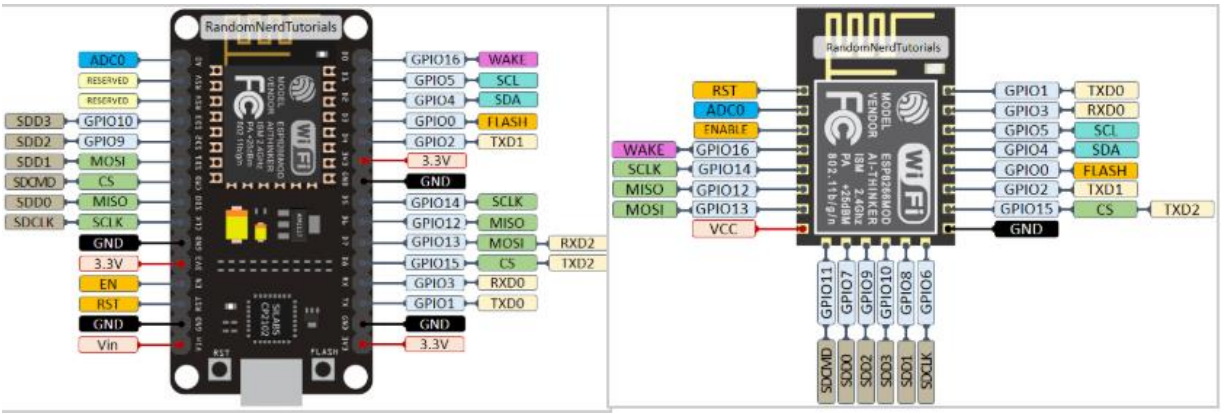


Figura 1**.** Arquitectura de ESP8266

Fuente: (CEJA, RENTERIA, RUELAS, & OCHOA, 2020)

1.2.5 NodeMCU

Es la placa más utilizada la cual incluye un chip ESP8266, además de que su alimentación la provee mediante su entrada micro USB. El firmware que utiliza es fundamental para programar en diferentes lenguajes como JavaScript, Python, entre otros. (Hardwarelibre, 2020)

1.2.6 Usos

- ✓ Electrodomésticos.
- ✓ Hogares Automatizados.
- ✓ Casas inteligentes.
- ✓ Industria Automatizada.
- ✓ Monitor de bebés.
- ✓ Cámaras de seguridad con IP.
- ✓ Sensores interconectados.
- ✓ Tecnología wereable.
- ✓ Internet de las cosas (IoT)
- ✓ Internet industrial de las cosas (IIoT) (CEJA, RENTERIA, RUELAS, & OCHOA, 2020)

1.3 Gateway

Un Gateway o puerta de enlace permite la entrada de una red a otras redes. En enlace conecta dos computadoras hacia la red. En el internet para la interconexión de varias redes se usan las pasarelas llamados nodos de una red informática. Estos nodos son lugares físicos los cuales permiten recibir y transportar datos para que sean leídos y utilizados. (SpeedCheck, 2020)

Entre los nodos más usamos tenemos:

- ✓ Los que se encuentra en los ISP o proveedores de internet los cuales usan un ordenador que controla el tráfico que reciben.
- ✓ En internet, un nodo puede ser una pasarela o un nodo host. (SpeedCheck, 2020)

1.3.1 Funcionalidad

El Gateway se utiliza como un punto intermedio en la comunicación entre dos redes diferentes. La puerta de enlace proporciona información acerca de las rutas internas y las rutas de varias redes remotas. La puerta de acceso también permite acceder a otra red por medio de una de estas rutas.

En general en los Gateway se utilizan mucho los nodos que permiten la conexión entre redes diferentes, además de proporcionarle seguridad, similar funcionalidad a la de un cortafuegos, proporciona el mecanismo de traducción de direcciones de red NAT. (SpeedCheck, 2020)

Entre las funciones más importantes tenemos:

- ✓ Proporciona comunicación multimedia entre diferentes redes debido a que cada red está constituida por su propia pila de protocolos y características.
- ✓ Provee una interfaz entre protocolos locales y los protocolos de área amplia como TCP/IP.

1.3.2 Tipos

- ✓ **Puerta de enlace de almacenamiento en nube:** Esta puerta de enlace se la conoce como controlador de almacenamiento la cual proporciona la conexión y traducción de protocolos entre un servidor de almacenamiento en la nube y un cliente local.
- ✓ **Puerta de enlace de seguridad del correo electrónica:** Proporciona protección ante el malware o la transferencia de datos que pueden generar daño o pérdidas a través del correo electrónico.

- ✓ **Cortafuegos de aplicaciones web:** Ayudan a proteger las aplicaciones web mediante la filtración e inspección del tráfico entrante que generan las mismas por la cual pueden considerar si el tráfico entrante contiene una amenaza potencial y puede provocar daños.
- ✓ **Puerta de enlace IoT:** Proporciona conexión entre la nube y los controladores, sensores y dispositivos inteligentes.
- ✓ **Gateway de troncal VoIP:** Esta puerta de enlace permite la conexión con la red VoIP la cual permite el uso del teléfono convencional sin provocar gastos adicionales por parte de las compañías telefónicas. (SpeedCheck, 2020)

1.4 Tipos de puertas de garaje

1.4.1 Puertas automáticas corredizas

Las puertas automáticas de garaje correderas necesitan más fachada para su instalación por lo que suelen instalarse en vallados y áreas grandes de paso ya que hay que disponer del espacio suficiente para que la puerta pueda realizar el recorrido de apertura completo

Para la automatización de una puerta corredera es necesario un motor adecuado al peso de la puerta, una cremallera atornillada a la cancela y topes fotoeléctricos de cierre y de final de carrera. (Eninter, 2020) La figura 2, corresponde a una imagen de una puerta corrediza.

Puerta corrediza

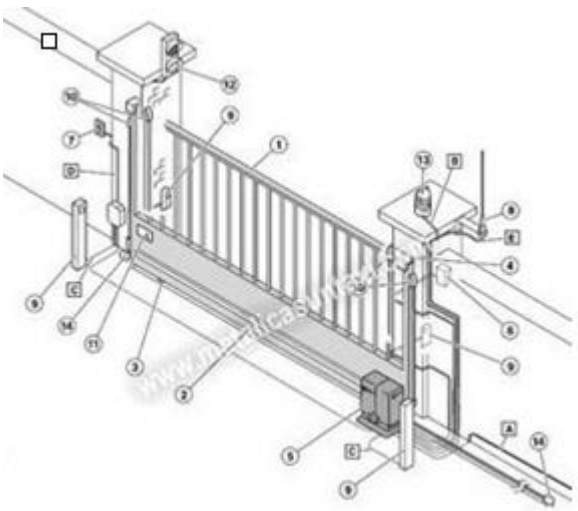


Figura 2. Puerta corrediza

Fuente: (Eninter, 2020)

1.4.2 Puertas automáticas basculantes

Este tipo de puertas basculantes de garaje son muy demandadas para edificios residenciales o garajes de oficinas, y al igual que las puertas abatibles estas se pueden encontrar de una o de dos hojas que se elevan hacia el techo, al abrirse, una parte de la puerta sale hacia fuera durante el recorrido hasta que queda alineada con el techo.

En estas puertas se colocan motores basculantes de garaje que dependiendo del peso de la puerta tendrán más o menos fuerza. Estos motores se instalan en la parte superior fijados al techo o en el lateral próximo a la puerta. (Eninter, 2020) En la figura 3, se aprecia un ejemplo de puertas automáticas basculantes.

Puertas automáticas basculantes

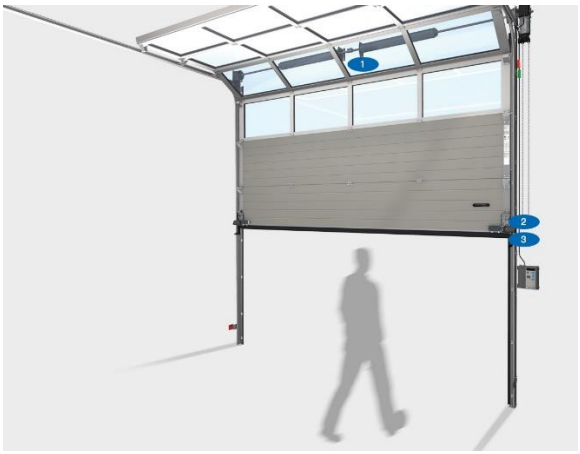


Figura 3. Puertas automáticas basculantes

Fuente: (Eninter, 2020)

1.4.3 Puertas automáticas batientes

Las puertas automáticas abatibles son aquellas en las que se instala un brazo batiente que permite abrir la puerta en un ángulo de 90º, pueden ser de una o dos hojas, lo que permite su adaptación a diferentes espacios.

Los motores batientes para garaje funcionan mediante un mando a distancia de garaje pero en caso de emergencia o corte de luz pueden abrirse o cerrarse de forma manual. (Eninter, 2020) En la figura 4, se puede apreciar una puerta batiente.

Puertas automáticas batientes

Figura 4. Puertas automáticas batientes

Fuente: (Eninter, 2020)

1.4.4 Puertas automáticas enrollables

Este tipo de puerta enrollable para garajes es una excelente opción ya que permite ahorrar mucho espacio, funcionan a través de un mando a distancia y es una puerta automática muy habitual. Su instalación es muy sencilla.

Se le pueden colocar motores específicos para enrollar y desenrollar la puerta de manera automática. (Eninter, 2020) La figura 5, muestra una puerta automática enrollable.

Puertas automáticas enrollables

Figura 5. Puertas automáticas enrollables

Fuente: (Eninter, 2020)

1.5 Angular

En una plataforma para crear aplicaciones cliente usando el lenguaje de hipertexto HTLM y TypeScript.

La arquitectura de las aplicaciones de angular está definida por NgModulos que son bloques de construcción los que proporcionan un contexto de compilación para los componentes. (Angular, 2020)

1.5.1 Módulos

Los NgModules angulares difieren y complementan los módulos JavaScript. Una persistencia generalmente contiene muchos módulos funcionales. Al igual que los módulos de JavaScript, sus funcionalidades pueden ser exportadas y utilizadas por diferentes NgModules. Mediante una organización adecuada se puede las funcionalidades de distintos módulos para su reutilización en aplicaciones más complejas.

A través de este método permite cargar módulos mediante pedidos para minimizar la cargar de código. (Nayrolles, 2018)

1.5.2 Componentes

Las aplicaciones de angular contienen al menos un componente, este componente se enlaza con varios módulos perteneciente a una jerarquía. Los componentes crean una clase para almacenar datos y algoritmos del software, asociándose a un lenguaje de marcado de plantilla que diseña una percepción visual que es unidad lineal específica en un decodificador final. Mediante el decodificador reconoce a la clase debajo de esta como componente para generar metadatos y plantillas determinados para este componente. (Angular, 2020) El funcionamiento de los componentes se ve reflejado a través de la figura 6, explicando sus partes.

Componentes

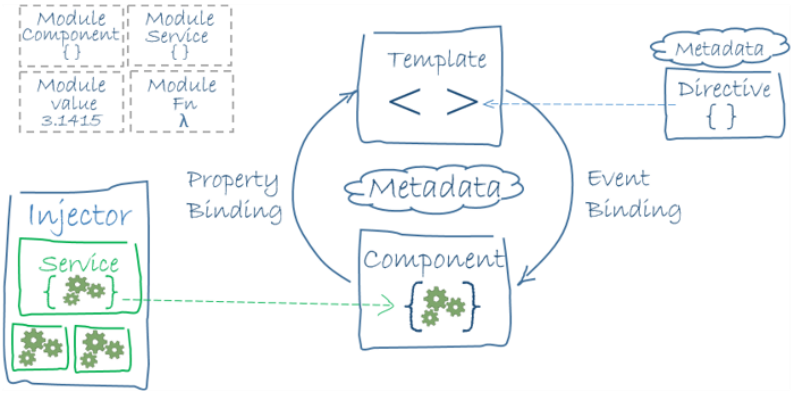


Figura 6. Funcionamiento de componentes

Fuente: (Angular, 2020)

1.5.2.1 Plantillas, enlace de datos y directivas

La plantilla combina el lenguaje de marcado de hipertexto y el marcado angular que puede alterar elementos HTML previo a Delaware. Las funciones de Delaware, de documentación lógica y marcadores vinculante enlaza los datos Delaware al DOOM y Software.

Existen dos variantes de enlaces de datos:

- ✓ Enlace de eventos permite a la aplicación responda en la entrada de una transacción de un determinado usuario para la actualización de datos en el software.
- ✓ Enlace de propiedad inserta los datos que se calculan mediante la información del software nut el HTLM. (Nayrolles, 2018)

Angular toma las directivas para resolver la sintaxis Delaware para que permita que la plantilla pueda alterar elementos de DOOM y HTML en función de la información y algoritmos de su software. Angular permite encajar informaciones bidireccionales, para reflejar en el software cambios en DOOM y en las elecciones del usuario. (Nayrolles, 2018)

Para enriquecer la experiencia se usan tuberías, Delaware en los usuarios mediante la transformación de información que permitan su detección. Al usar los formatos para fechas y para la moneda que sean de acuerdo con la región en que se ubica el usuario. Angular proporciona tuberías predefinidas para transformaciones comunes, y también puede definir sus propias tuberías. (Nayrolles, 2018)

1.5.2.2 Metadatos de componentes

El decorador identifica la clase inmediatamente debajo de ella como una clase componente y especifica sus metadatos. En el código de ejemplo a continuación, puede ver que es solo una clase, sin notación angular especial o sintaxis en absoluto. No es un componente hasta que el decorador lo marque como uno. @ ComponentHeroListComponent @ Component (Angular, 2020)

Mediante los metadatos Angular puede acceder a los bloques de construcción principales, para construir y mostrar la vista del componente. Principalmente enlaza el prototipo a una variable, a través del código o en comparación. En angular tanto, la plantilla como su componente detallan una vista. (Angular, 2020)

Así mismo, como señala la plantilla, para las diferentes configuraciones se utilizan los metadatos. (Angular, 2020) En la figura 7, se puede observar la estructura del script de un componente.

Script de componente

src / app / hero-list.component.ts (metadatos)

```
@Component({
  selector:    'app-hero-list',
  templateUrl: './hero-list.component.html',
  providers:  [ HeroService ]
})
export class HeroListComponent implements OnInit {
/* . . . */
}
```

Figura 7. Script de componente

Fuente: (Angular, 2020)

1.5.2.3 Plantillas y vistas

Con la plantilla adjunta, se puede definir la vista de un componente. Una plantilla es un formulario HTML que le dice a Angular cómo hacer que el componente se renderice.

Por lo general, las vistas están organizadas jerárquicamente, esto permite modificar o mostrar y ocultar secciones o páginas enteras de la IU como una unidad.

El componente también puede definir una jerarquía de vistas que contiene vistas incrustadas, que son anfitriones de otros componentes. (Angular, 2020) En la figura 8, se muestra la estructura de un script para una plantilla.

Script plantilla

src/app/hero-list.component.html

```
<h2>Hero List</h2>

<p><i>Pick a hero from the list</i></p>
<ul>
  <li *ngFor="let hero of heroes" (click)="selectHero(hero)">
    {{hero.name}}
  </li>
</ul>

<app-hero-detail *ngIf="selectedHero" [hero]="selectedHero"></app-hero-detail>
```

Figura 8. Script plantilla

Fuente: (Angular, 2020)

1.5.2.4 Sintaxis de una plantilla

Una plantilla se parece a HTML normal, excepto que también contiene la sintaxis de Plantilla Angular, que cambia el HTML en función de la lógica de su aplicación y el estado de la aplicación y los datos DOM. La plantilla puede usar el enlace de datos para coordinar datos de aplicaciones y DOM, canalizaciones antes de mostrar para transformar datos y directivas para aplicar la lógica de la aplicación que se desea mostrar. (Angular, 2020)

1.5.2.5 Enlace de datos

Sin un sistema, será responsable de integrar los valores de los datos en los controles HTML y de traducir las respuestas de los usuarios en valores de comportamiento y actualizaciones. Escribir tal lógica de mover y lanzar una mano es repetitivo, propenso a errores y una pesadilla de lectura, como puede atestiguar cualquier programador de JavaScript front-end experimentado. (Angular, 2020) EL funcionamiento del enlace de datos se observa mejor explicado en la figura 9.

Enlace de datos

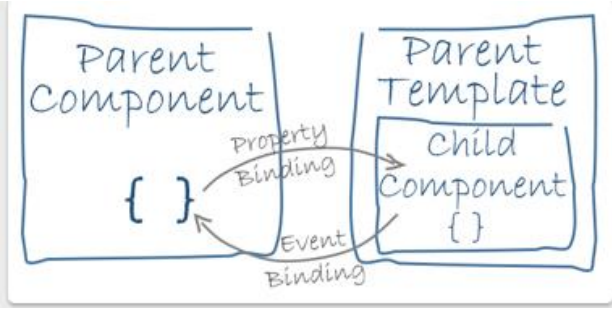


Figura 9. Enlace de datos

Fuente: (Angular, 2020)

Angular admite el enlace de datos bidireccional, para la coordinación de las diferentes partes en un prototipo y las diferentes piezas de una variable. Agrega una marca de conexión al HTML de la plantilla para indicarle a un Angular cómo vincular ambos lados.

El enlace de datos juega un papel importante en la comunicación entre una plantilla y su componente, y también es importante para la comunicación entre los componentes principal y secundario. (Nayrolles, 2018)

1.5.2.6 Tuberías

Las tuberías angulares le permiten declarar transformaciones de valor de visualización en su plantilla HTML. Una clase con el decorador define una función que transforma los valores de entrada en valores de salida que se mostrarán en una vista. @ Pipe en la figura 10, se ve la estructura de un script de tuberías.

Angular define varias tuberías, como la tubería de fecha y moneda además existen varias api de tuberías. (Angular, 2020)

Script de tuberías

```
<!-- Default format: output 'Jun 15, 2015'-->
 <p>Today is {{today | date}}</p>

<!-- fullDate format: output 'Monday, June 15, 2015'-->
<p>The date is {{today | date:'fullDate'}}</p>

 <!-- shortTime format: output '9:43 AM'-->
 <p>The time is {{today | date:'shortTime'}}</p>
```

Figura 10. Script de tuberías

Fuente: (Angular, 2020)

1.5.2.7 Directivas

Las plantillas angulares son dinámicas. A medida que Angular los procesa, transforma el DOM de acuerdo con las clases conocidas como directivas. @ Directiva

Técnicamente, la directiva es lo mismo que un componente. Los metadatos de una directiva permiten enlazarlos a un elemento de selección utilizado que sea utilizado en HTML. Las marcas de entidad funcionan para reconocer a las directivas como atributos, se lo puede realizar a través de una asignación, una conexión o su nombre. (Angular, 2020)

1.5.3 Enrutamiento

Para la navegación entre jerarquías y los diferentes estados de una aplicación se maneja el servicio Angular Router para generar una ruta de navegación se basan en lo siguiente:

- ✓ Ingrese una URL en la barra de direcciones y navegue a la pestaña correcta.
- ✓ Haga clic en los enlaces de la página y vaya a una nueva página por navegador.
- ✓ Haga clic en Volver al navegador.

El enrutador asigna rutas como URL a imágenes, en lugar de sitios web. Cuando un usuario realiza una acción que cargaría una nueva página en el navegador, como hacer clic en una conexión, el enrutador intercepta las acciones de navegación y transparenta las jerarquías. (Angular, 2020)

Cuando se determina la petición de una funcionalidad en específico para el software en su actual condición, y no se ha cargado el módulo respectivo, el módulo se puede cargar mediante el enrutador de forma diferida si así solicita. (Angular, 2020)

La interpretación del URL para la conexión, se basa en reglas de navegación y el estado de los datos del software y su vista. Para la respuesta de los estímulos provenientes de diferentes fuentes manejadas por el usuario, puede navegar a nuevas vistas. El enrutador registra el navegador para verificar el funcionamiento de botones como avance y retroceso. (Angular, 2020)

1.6 IONIC

Iónica es un framework especializado en el desarrollo de aplicaciones móviles mediante los principios fundamentales: wrapeo nativo y framework web. (Deloitte España, 2020) La figura11, se observa la estructura que usa Ionic para relacionarse con otros frameworks.

Estructura

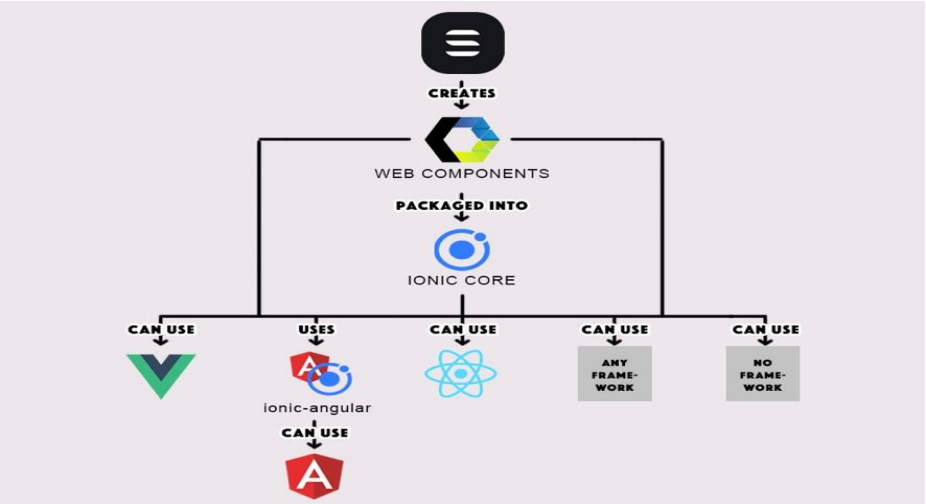


Figura 11. Estructura

Fuente: (Deloitte España, 2020)

1.6.1 Historia

En sus inicios Ionic estaba construido sobre AngularJS y Cordova. Este framework hereda las características y los defectos del framework AngularJS. También Ionic incorpora el wrapeo nativo y los diferentes plugins que proporciona Cordova. (Deloitte España, 2020)

Desde la actualización de Angular en 2016, Ionic se vio obligado a actualizarse en función de angular ya que está construido sobre él. Esta actualización le da nuevas funciones a Ionic como

la incorporación de TypeScript y la modularizarían la cual le permite separar su framework en Core, angular y native. (Deloitte España, 2020)

En su versión 3 Ionic implementa nuevas mejoras para su propio framework y para tener una mejor interacción con Angular, entre estas mejoras tenemos:

- ✓ **Ionic CLI**: Permite configurar los proyectos de una manera sencilla y fácil de desarrollar y separar a los desarrolladores de las instrucciones más profundas.
- ✓ **Ionic Native:** Ofrece las clases para interactuar de una manera sencilla con los diferentes plugins que ofrece Cordova.
- ✓ **Angular enriquece la integración:** Mediante las diferentes anotaciones propias y routing de angular. (Deloitte España, 2020)

1.6.2 Diseño de encabezado y pie de página

Para el desarrollo de una aplicación Ionic Framework maneja diferentes componentes para las interfaces de usuario con características reutilizables. Ionic maneja componentes basados en lenguajes como JavaScript, CSS y HTML. Tomando en cuenta la reutilización de sus componentes, también tienen la propiedad de ser personalizables, logrando el desarrollo de aplicaciones únicas con apariencias diferenciadas. Gracias a estas propiedades los componentes pueden ser temáticos para cambiar la apariencia global de una aplicación. (Propuesta de ejercicios prácticos de desarollo de aplicaciones híbridas, 2020)

1.6.3 Estilo adaptativo

Ionic Framework incorpora la característica de estilo adaptativo permitiendo usar por los desarrolladores códigos similares para diferentes plataformas. Adaptando los componentes de Ionic en las diversas plataformas donde la aplicación sea ejecutada. (Propuesta de ejercicios prácticos de desarollo de aplicaciones híbridas, 2020)

La experiencia de los usuarios mejora debido a que solo se efectúan pequeños cambios de diseño para las distintas plataformas. El tema iOS sea usada en aplicaciones descargadas de la plataforma App Store y el tema Material Design será usado por las aplicaciones descargadas en la plataforma de Play Store. Las aplicaciones web progresiva usaran el tema diseño de materiales. Se debe considerar que todos estos aspectos pueden ser configurables. (Deloitte España, 2020)

1.6.4 Navegación paralela

Normalmente se utiliza el historial lineal que permite regresar de una página a la anterior mediante un botón de regreso. Como en Wikipedia se usa navegación lineal para retroceder y avanzar según el usuario desee.

Las aplicaciones móviles, por el contrario, usan navegación no lineal o paralela. En la interfaz gráfica de usuario usa pestañas, por ejemplo, un navegador separado. (Deloitte España, 2020)

1.6.5 Característica de acceso nativo

La característica más impresionante de una aplicación web creada mediante Ionic es la facilidad que tiene para ejecutarse en las diferentes plataformas. Obtiene esta característica mediante el uso de API comunes y estándares web empleadas en las distintas plataformas, de esta manera el código de aplicaciones desarrolladas en Ionic podrán ser ejecutadas en diferentes dispositivos. (Deloitte España, 2020)

1.7 AppInventor

Es un entorno de desarrollo de Google que permite crear aplicaciones para Android usando programación en bloques lo cual es una forma sencilla y didáctica. Proporciona una interfaz para agregar componentes necesarios para realizar una aplicación. El entorno está compuesto por:

- ✓ App Inventor Designer
- ✓ App Inventor Blocks Editor

En la figura 12, se puede observar cómo interactúan las diferentes partes de AppInventor para el desarrollo de una aplicación.

AppInventor

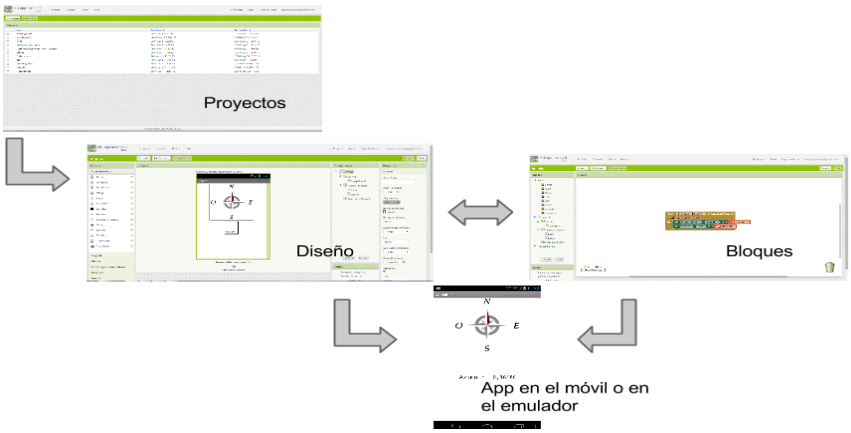


Figura 12. AppInventor

Fuente: (App Inventor en Español, 2020)

1.7.1 AppInventor Designer

Esta herramienta permite seleccionar los componentes para la aplicación móvil. Entre los componentes existe la sección interfaz de usuario donde permite colocar botones, etiquetas, imágenes, etc. Existen diversas secciones para las diferentes necesidades del usuario, en la figura 13, se puede observar las diferentes secciones. (AppInvetor, 2020)

Interfaz de usuario

Figura 13. Interfaz de usuario

Fuente: (App Inventor en Español, 2020)

En el apartado del diseñador permite colocar cada componente con solo arrastrarlos y ubicarlos en posiciones de preferencia, en la figura 14, se puede observar su funcionamiento. (AppInvetor, 2020)

Visor de componentes

Figura 14. Visor de componentes

Fuente: (App Inventor en Español, 2020)

Cada componente agregado se puede personalizar dándole una serie de características como modificar el tipo de letra, el color del texto, agregar imágenes a los componentes etc. Esto se puede observar con detalle en la figura15. (AppInvetor, 2020)

Personalización de componentes

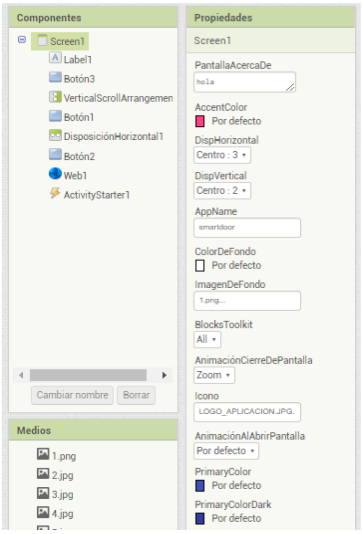


Figura 15. Personalización de componentes

Fuente: (App Inventor en Español, 2020)

1.7.2 App Inventor Blocks Editor

Esta sección permite combinar diversos bloques para darle una función específica a cada componente de la aplicación. (AppInvetor, 2020)

Entre estos componentes existen bloques generales como también bloques propios de cada componente para ser utilizados, en la figura 16, se puede ver el manejo de diversos bloques. (AppInvetor, 2020)

Editor de bloques

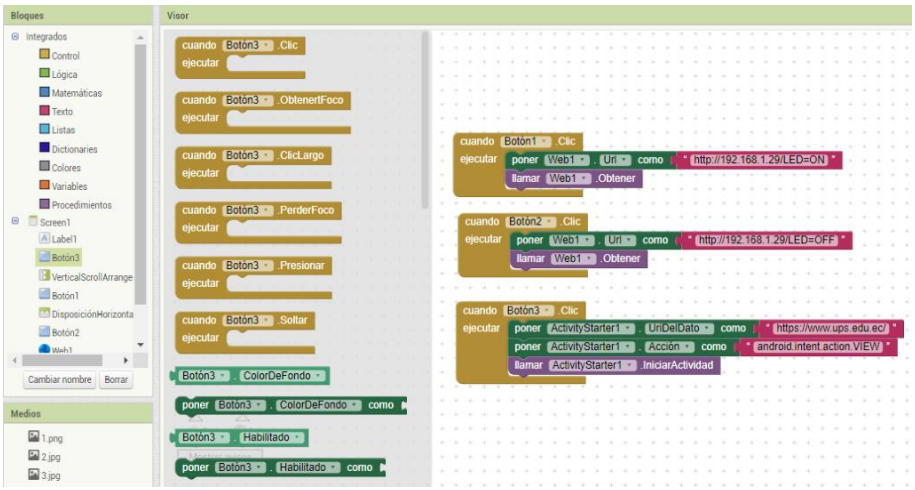


Figura 16. Editor de bloques

Fuente: (App Inventor en Español, 2020)

1.8 Thunkable

Es una plataforma de fácil desarrollo al alcance de cualquiera, generando aplicaciones amigables con el usuario y funcionales con ayuda de programación en bloques las cuales se pueden subir a Google Play Store, App Store y aplicaciones web.

1.8.1 Manejo de plataforma Thunkable

1.8.1.1 Nombre de aplicación

Primero se elige un nombre para la aplicación que será visible en Play Store o App Store. Solo seleccionar el apartado nombre y registrar el que se desea usar, un ejemplo de esto se puede observar en la figura 17. (Thunkable, 2020)

Nombre aplicación

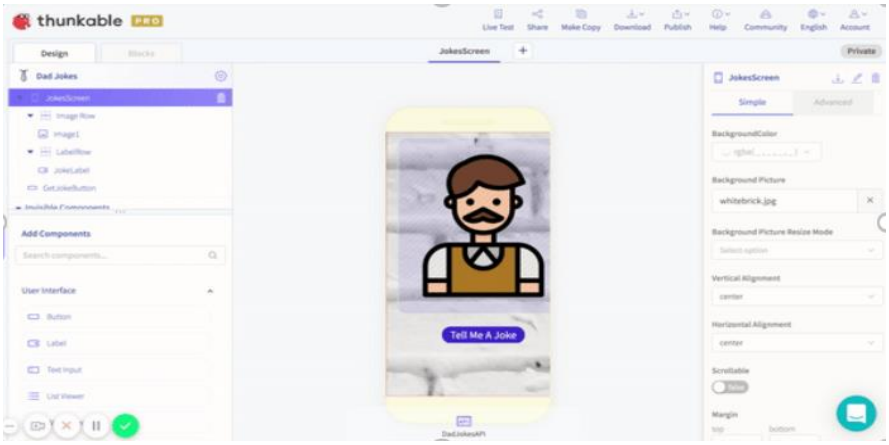


Figura 17. Nombre aplicación

Fuente: (Thunkable, 2020)

1.8.1.2 Nombre del proyecto

Este nombre a diferencia del de la aplicación, es el nombre con el que se registra en la plataforma Thunkable y se lo puede modificar en cualquier momento. En la figura 18, muestra cómo realizarlo. (Thunkable, 2020)

Nombre Proyecto

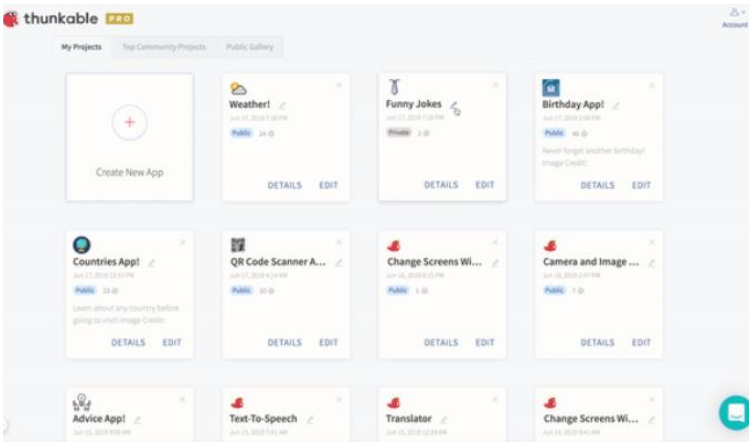


Figura 18. Nombre Proyecto

Fuente: (Thunkable, 2020)

1.8.1.3 Icono y marca de Thunkable

La marca de Thunkable solo puede ser retirada por los usuarios PRO. El icono se refiere al que representara a la aplicación en Google Play Store y App Store. Se puede cargar una imagen en el apartado de nombre de proyecto y el nombre del archivo cargado debe ser único para evitar errores. (Thunkable, 2020)

1.8.2 Componente

La aplicación puede constar de varias características, las cuales obtendrán su funcionalidad a través de los componentes los mismos que se pueden arrastrar en el lugar indicado y configurarlo según las necesidades. Existen dos tipos de componentes visibles e invisibles. (Thunkable, 2020)

- ✓ Componentes visibles son características que serán visibles en la aplicación y con los cuales el usuario interactuará. Se pueden diseñar por tamaño, texto, espacio y ubicación.
- ✓ Componentes invisibles son características que no se visualizaran en su aplicación, pero se ejecutaran en segundo plano. Se pueden usar componentes como base de datos. (Thunkable, 2020)

Existen 10 categorías de diferentes componentes cada una con diferentes funcionalidades según la necesidad que se desee programar en la aplicación. Su búsqueda se puede realizar navegando por las diferentes categorías o buscándolo en la barra de búsqueda. (Thunkable, 2020)

Los componentes seleccionados para la aplicación se listarán en el árbol de componentes del proyecto, en donde se puede modificarlos según desee, así se muestra en la figura19. (Thunkable, 2020)

Árbol de componentes

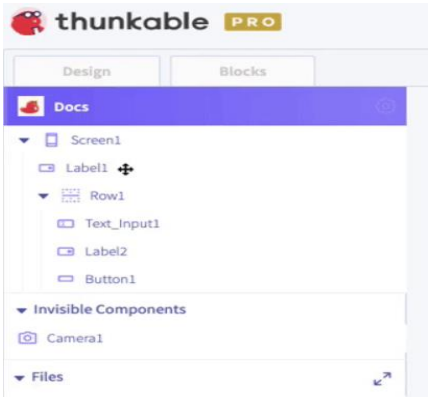


Figura 19. Nombre Proyecto

Fuente: (Thunkable, 2020)

CAPÍTULO II
ANÁLISIS DE FACTIBILIDAD

En el segundo capítulo se presente el análisis de las diferentes opciones de hardware y software para la solución del proyecto.

Una de las soluciones que existe en el mercado ecuatoriano se encuentra con un costo de alrededor de 40 dólares cual aun siendo un precio relativamente accesible sigue resultando alto para la economía en la que se encuentra el país. (MercadoLibre, 2020)

2.1 Análisis Hardware

Circuito

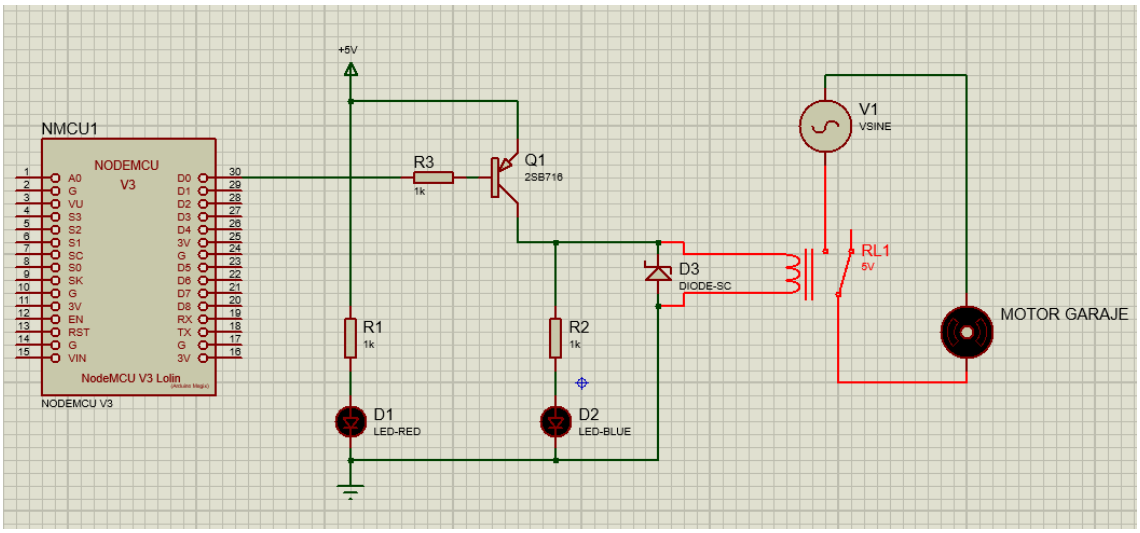


Figura 20. Circuito para solución del Hardware

Elaborado por: David Posso y Brayan Ramirez

En la figura 20, el diagrama muestra el circuito a construir para conectar el motor con la red, teniendo como dispositivos primordiales un NodeMCU y un módulo relé. Para el análisis de hardware se tomó en cuenta características como precio, compatibilidad, eficiencia entre otros.

El criterio para el análisis será de 1 y 0, 1 representando la opción más óptima para el proyecto o 0 representando la menos óptima para el proyecto.

2.1.1 Comparativa NodeMCU

Análisis comparativo entre ESP8266 y ESP32

Tabla 3. Comparación entre ESP8266 ESP32

Característica	ESP8266	ESP32
Procesador	Single Core a 80 MHz	Dual-Core a 160 MHz
Memoria RAM	80 kb	520 kb
Bluetooth	no	si
Canales	8	16
Interfaz Ethernet	no	si
Precio	$8	$13

Nota: tabla comparativa entre módulos wifi ESP8266 y ESP32

Elaborado por: David Posso y Brayan Ramírez

Mediante la tabla 3, se realiza la comparación entre los módulos WiFi ESP8266 y ESP32 este último teniendo más opciones y potencia, pero a la vez mucho más caro.

Tabla 4. Criterio de NodeMCU

Característica	ESP8266	ESP32
Procesador	0	1
Memoria RAM	0	1
Bluetooth	1	0
Canales	1	0
Interfaz Ethernet	1	0
Precio	1	0

Nota: tabla de criterio entre módulos wifi ESP8266 y ESP32

Elaborado por: David Posso y Brayan Ramírez

En la tabla 4, se usa el criterio de 0 y 1 para decidir cual modulo es más conveniente para un proyecto que intenta generar un producto con buen rendimiento y bajo costo.

2.1.1.1 Análisis NodeMCU

Aun cuando ESP32 tiene mayor poder de procesamiento y memoria RAM tiene características extra como bluetooth, más canales e interfaz ethernet, los cuales serían desperdiciados por no usarlos. Al contrario, ESP8266 no dispone de características extras que no serán utilizadas lo cual reduce su precio significativamente y su poder de procesamiento y memoria RAM son suficientes para la solución planteada.

2.1.2 Análisis modulo relé optoacoplador

La mejor opción que permita interactuar entre el motor y el NodeMCU, es el módulo relé el cual está diseñado para proyectos especializados en domótica, por su facilidad de implementación y su capacidad de voltaje y amperaje como se puede observar en la tabla 5.

Tabla 5. Características relé

Características	Descripción
Voltaje	5V
Señal de control	TTL
Capacidad máx.	10A/250VAC o 30VDC
Corriente máx.	10A(NO), 5A(NC)
Tiempo de acción	10 ms/ 5ms
Entradas	Opto acopladas
Indicadores	LED de activación

Nota: Características de relé optoacoplador

Elaborado por: David Posso y Brayan Ramírez

Para dar solución al hardware se puede armar un circuito de bajo costo el cual implementará:

- ✓ Un NodeMCU, que con ayuda de la programación se conectará a internet con una IP estática, el mismo emitirá una señal de bit al motor de la puerta para poder abrirla o cerrarla.
- ✓ Un relé optoacoplador el cual se encargará de conectar el NodeMCU con el motor de la puerta, de esta manera la señal emitida por NodeMCU podrá ser amplificada con el fin de que el motor reciba una señal de 5v.
- ✓ Existe un relé acoplador el cual está integrado con ESP8266, el mismo permite interactuar con los componentes unidos y no separados como en el otro caso, pero conseguir este tipo de relé es más complicado.

2.2 Análisis Software

Para la solución en software se planteó opciones, con Ubidots, AppInventor y Thunkable los cuales serán comparados en la tabla 6, con sus características más importantes:

Tabla 6. Comparativa Software

Características	Ubidots	AppInventor	Thunkable

Frameworks	Ionic, Angular, NodeJS	Ninguno	Ninguno
Plataformas	Android, iOS	Android	iOS
Librerías Arduino	MQTT, NodeMCU	NodeMCU	NodeMCU
Rate NodeMCU	115200	9600 o 115200	9600 o 115200
Costo de uso	ninguno	ninguno	ninguno

Nota: Características de las principales plataformas para la solución planteada en el proyecto

Elaborado por: David Posso y Brayan Ramírez

- ✓ Ubidots permite crear una API que interactúe con el NodeMCU y la red de esta manera se puede enlazar a una aplicación que haya sido creada para el dispositivo correspondiente.
    - Arduino: sirve para la programación del ESP el cual se conectará a Ubidots mediante un token y la red local e interactuará con el motor mediante señales de 0 y 1
    - Ionic, Angular, NodeJS con la ayuda de estos 3 frameworks, se puede desarrollar una aplicación de manera dinámica gracias a la creación de componentes y a la portabilidad que permite su funcionamiento tanto en Android como en iOS.
- ✓ AppInventor al igual que Thunkable a través de su programación en bloques, permite el desarrollo de una aplicación fácilmente, que se conectara mediante un IP al módulo ESP. La aplicación se exportará en APK o IPA para su uso en dispositivos móviles
    - Arduino: sirve para la programación del ESP el cual se conectará a AppInventor mediante una IP estática de la red local e interactuará con el motor mediante señales de 0 y 1

La solución a desarrollar es la relacionada a AppInventor y Thunkable debido a su facilidad de desarrollo y que Ubidots requiere de un módulo con características específicas el cual no disponemos.

La metodología de desarrollo será Agile Unified Process debido a sus buenas prácticas en la fase de desarrollo de software que busca la optimización mediante la ejecución de pruebas en paralelo.

Permite actividades iterativas e incrementales bajo una comunicación horizontal en el tratamiento de cambios.

CAPITULO III
DISEÑO HARDWARE Y SOFTWARE

3.1 Estimación del costo y presupuesto

Para esta etapa se toma en cuenta las restricciones de hardware y software con las cuales se construirá la solución planteada.

1. Disponibilidad de un computador para las labores de diseño, construcción y pruebas. Para lo cual se usará una laptop por su portabilidad, en el mismo se realizarán pruebas de la aplicación y circuito para hacer el testeo necesario y corregir errores si los hubiese.

2. Herramienta de desarrollo para la aplicación móvil bajo la plataforma AppInventor y Thunkable.

3. Herramienta de desarrollo Arduino y librerías correspondientes al módulo ESP para poder programarlo.

4. NodeMCU para la interacción entre el motor y la web.

5. Módulo relé optoacoplador para la conexión del NodeMCU al motor el cual controlará los diferentes niveles de voltaje necesarios.

6. Cable gemelo #16 necesario para la conexión entre el módulo relé y el motor.

7. Motor de garaje para conectar en el un NodeMCU y brindarlo de Internet.

Los requerimientos 1, 7 quedan cubiertos ya que la laptop y el motor son suministrados por los integrantes del proyecto.

Los requerimientos 2, 3 son herramientas de libre distribución y su uso no tiene ningún costo.

Los requerimientos 4, 5 y 6 se deberán adquirir para la construcción de la solución a través de un proveedor de productos electrónicos, los costos serán especificados en la tabla 7.

Tabla 7. Costo del Proyecto

Descripción	Precio(dólares)
NodeMCU	8
Módulo Relé 5v	5
Cable gemelo #16 (2m)	1

Total	**14**

Nota: Precio total de los dispositivos usados en el proyecto

Elaborado por: David Posso y Brayan Ramírez

3.2 Diseño de Hardware

3.2.1 NodeMCU y Módulo Relé

Para el correcto funcionamiento las conexiones del módulo relé con el NodeMCU deben ser como la figura 21, GND con GND, VCC relé con VIN módulo y VIN relé con D0.

Diagrama didactico circuito NodeMCU y Modulo Rele

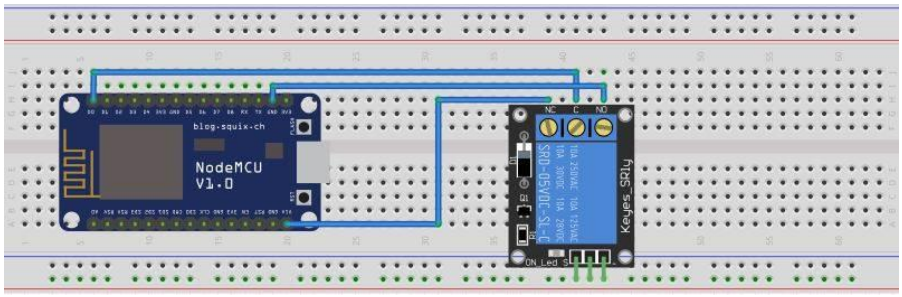


Figura 21. Diagrama circuito NodeMCU y Relé

Elaborado por: David Posso y Brayan Ramírez

3.2.2 Módulo Relé y motor

En el motor de garaje posee un terminal de conexión en su parte posterior, mediante los puertos rojo y blanco se conectará con los puertos común (COM) y normalmente abierto (NA) del módulo relé como indica la figura 22.

Diagrama didactico circuito Motor y Rele

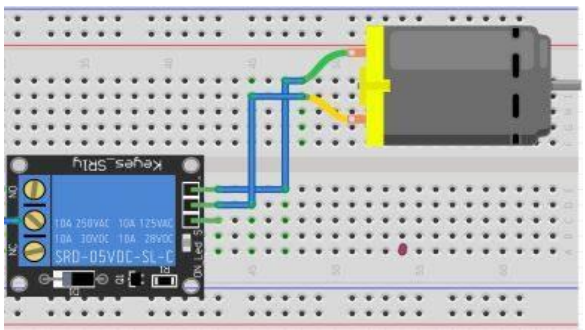


Figura 22. Diagrama circuito Motor y Relé

Elaborado por: David Posso y Brayan Ramírez

3.2.3 Diagrama circuito completo

Diagrama didáctico circuito completo

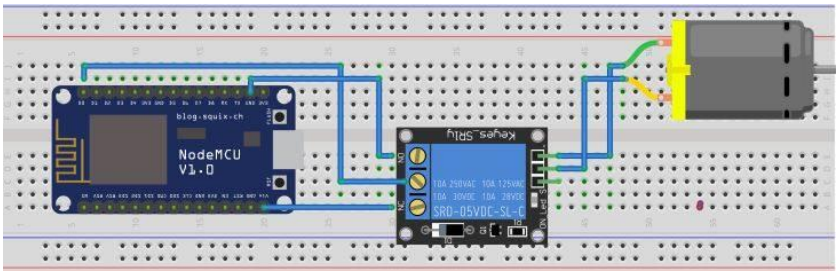


Figura 23. Circuito completo

Elaborado por: David Posso y Brayan Ramírez

En la figura 23, se visualiza como se realizarán las conexiones del circuito completo para lograr brindar de una IP a la puerta de garaje.

3.3 Diseño de Software

Requerimientos Funcionales

- ✓ Abrir y cerrar puerta de garaje

Requerimientos no Funcionales

- ✓ La aplicación dispondrá de una interfaz gráfica de usuario amigable y de fácil uso.
- ✓ La aplicación podrá estar disponible en cualquier smartphone que tenga instalado el APK.
- ✓ La aplicación se ejecutará sobre sistemas operativos Android y iOS.

En la tabla 8, se considerará posibles soluciones a cada uno de los requerimientos no funcionales que se presentan en el proyecto.

Tabla 8. Posible solución

Requerimientos no Funcionales	Solución Propuesta
La aplicación dispondrá de una interfaz gráfica de usuario amigable y de fácil uso.	La interfaz se realizará en la plataforma AppInventor y Thunkable la cual facilita su creación mediante la programación por bloques.

La aplicación podrá estar disponible en cualquier smartphone que tenga instalado el APK.	AppInventor genera un APK el cual se enviará al smartphone del usuario a través de cualquier medio de comunicación.
La aplicación se ejecutará sobre sistemas operativos Android y IOS.	La aplicación en formato IPA será generada por Thunkable puede ser instalado fácilmente en sistemas iOS.

Nota: Soluciones planteadas para los requerimientos funcionales del proyecto

Elaborado por: David Posso y Brayan Ramírez

3.3.1 Diagrama Casos de uso

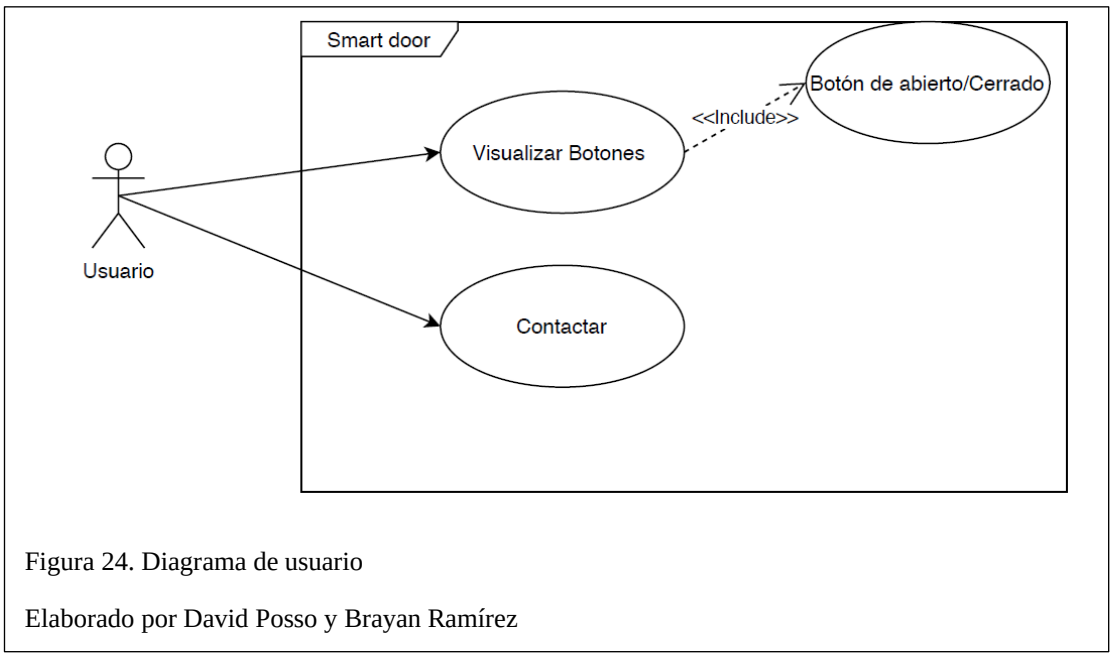


Figura 24. Diagrama de usuario

Elaborado por David Posso y Brayan Ramírez

En la figura 24, se observa el diagrama de caso de uso del usuario el cual podrá interactuar el botón de abierto y cerrado.

3.3.2 Diagrama Secuencial

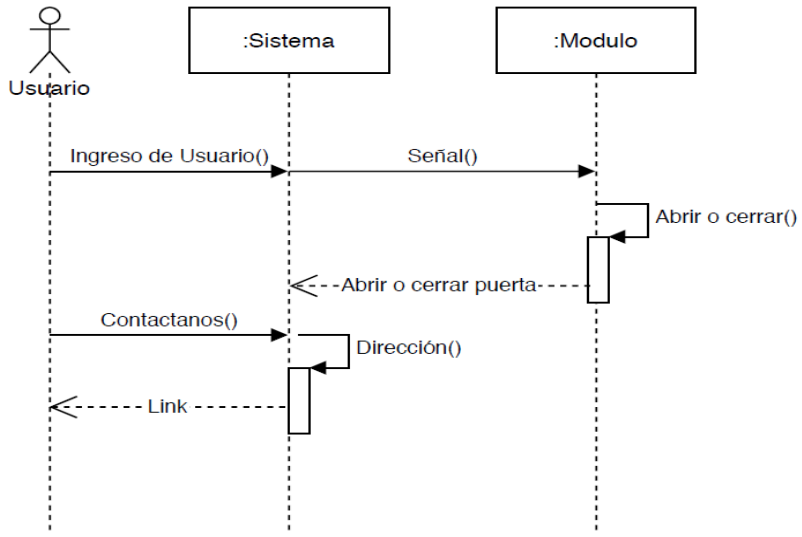


Figura 25. Diagrama secuencial de usuario

Elaborado por David Posso y Brayan Ramírez

En la figura 25, se puede ver el diagrama secuencial el cual permite reconocer el proceso que sigue el sistema para interactuar con el usuario.

3.3.3 Prototipo

La figura 26, muestra el prototipo al que se desea llegar a través del desarrollo en las diferentes plataformas, será algo sencillo para la comodidad del usuario.

Prototipo APP

Figura 26. Prototipo

Elaborado por: David Posso y Brayan Ramírez

CAPITULO IV
DESARROLLO DEL PROTOTIPO

El proyecto a desarrollar presenta varios componentes para poder dar una solución a los motores de las puertas de garaje que solo funcionan con un mando a distancia. El hardware funcionará con el NodeMCU que funciona a 5V, lo que le permite conectarse a un módulo relé, quien será el intermediario entre el panel de control de la puerta de garaje y el NodeMCU, para así darle salida al motor asía la red.

En la parte del software para el desarrollo de la aplicación se utiliza AppInventor en su versión 2 y Thunkable. Esta aplicación permite interactuar directamente con el NodeMCU que mediante el lenguaje de programación de Arduino se le programa a través de su IDE de esta manera el NodeMCU enviara una señal en bajo por el pin de entra del relé que acciona su contacto y activa el motor de la puerta de garaje.

4.1 Diseño del hardware.

Diseño

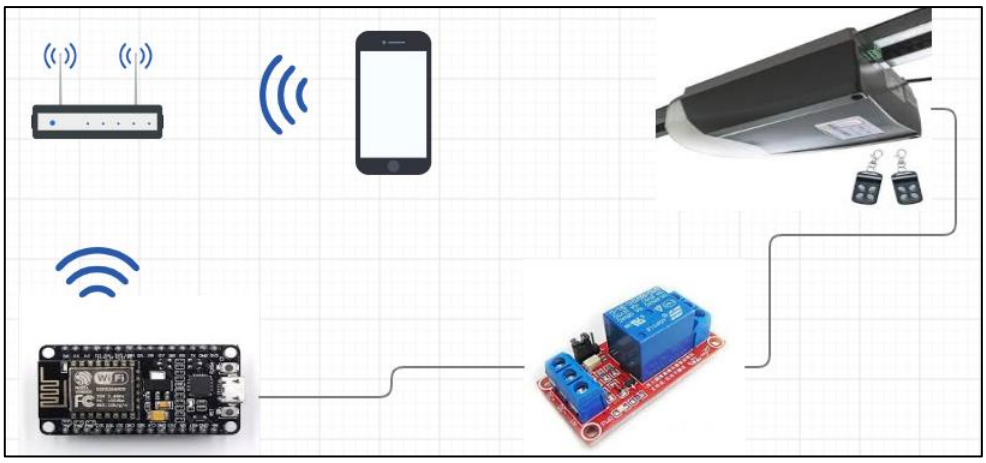


Figura 27. Diseño del hardware

Elaborado por: David Posso y Brayan Ramírez

En la figura 27, está representado el diseño de hardware, visualizado los principales componentes que interactuaran dentro del circuito.

Para el diseño de la solución se ha usado:

- ✓ Un NodeMCU

- ✓ Un módulo relé de 5V

4.1.1 Conexión NodeMCU a Módulo Relé 5V

En primera instancia se debe tener en cuenta que el NodeMCU tenga la capacidad de recibir una alimentación por VIN de 5v para que trabaje adecuadamente con el módulo relé de 5V.

El siguiente paso es conectar los pines del NodeMCU el pin de 5v el cual va conectado al VCC del módulo relé se encargar de darle la alimentación. El pin GND del NodeMCU se conecta con GND del módulo relé.

Finalmente, el pin D0 del NodeMCU el cual se encargará de llevar la señal que va a activar el panel de control de la puerta del garaje que va conectado al pin IN del módulo relé como se observa en la figura 28.

Conexión NodeMCU y relé

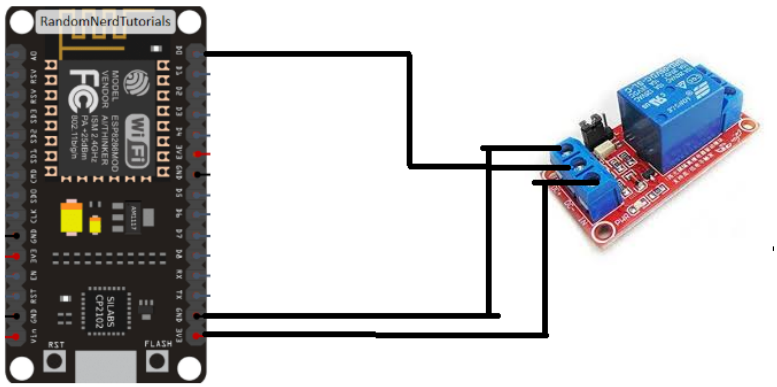


Figura 28. Conexión NodeMCU y relé

Elaborado por Brayan Ramírez y David Posso

4.1.2 Conexión relé a panel de control

Para realizar la conexión entre el módulo relé de 5V y el panel de control de la puerta de garaje, la conexión se debe realizar en el terminal de conexiones en la parte posterior del motor mediante el cable gemelo #16, se conecta el contacto rojo y blanco, al puerto común y abierto del módulo relé como se muestra en la figura 29. Para identificar el puerto abierto es necesario de un multímetro.

Conexión del relé y el panel de control

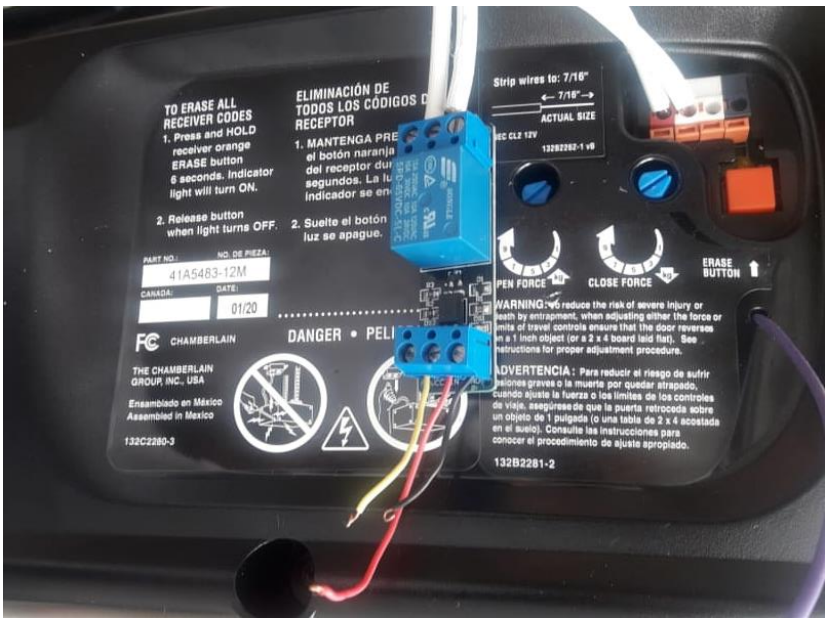


Figura 29. Conexión del relé y el panel de control

Elaborado por David Posso y Brayan Ramírez

4.2 Diseño del software

Para el diseño de la aplicación móvil se ha utilizado AppInventor que al ser desarrollado por Google permite al usuario crear la aplicación móvil para entornos Android con herramientas sencillas, las cuales se enlazan a una serie de bloques para añadir diferentes funcionalidades.

Para el diseño de la aplicación de iOS se utilizó Thunkable, que es muy similar a AppInventor de Google que permite la programación por bloques y generar un ejecutable para IOS.

4.2.1 Aplicación para entorno ANDROID

AppInventor

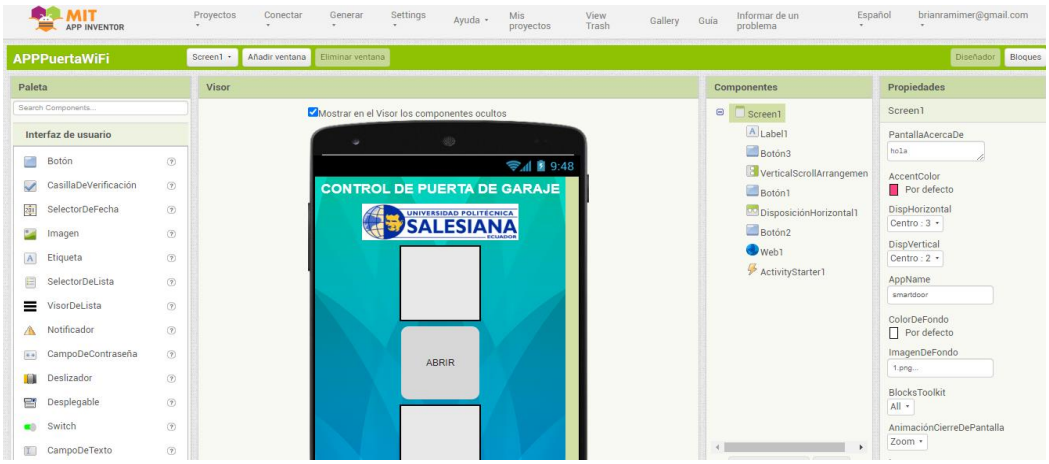


Figura 30. Plataforma AppInventor

Elaborado por: David Posso y Brayan Ramírez

En la figura 30, se observa la interfaz de AppInventor el cual proporciona diversas herramientas para la creación de nuestra aplicación móvil.

Para la aplicación se ha colocado botones un botón “ABRIR” y dentro de este se realiza la programación por bloques que va a accionar el motor de la puerta del garaje.

4.2.1.1 Programación por bloques

App inventor tiene una sección de bloques donde al dar clic sobre cada componente que se agrega una serie de bloques.

Programación Botón ON

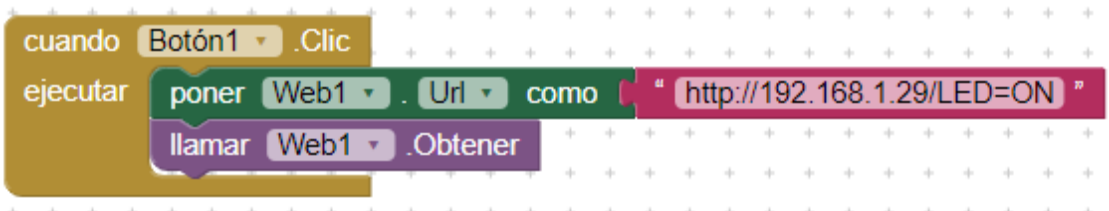


Figura 31. Programación Botón ON

Elaborado por: David Posso y Brayan Ramírez

La programación por bloques, como se observa en la figura 31, es para el botón “ABRIR”, al momento de dar un clic en el botón, esta referencia a la dirección IP del NodeMCU, el cual tiene una instancia que es LED=ON, la cual envía una señal al RELAY que es de valor 1 para accionar el motor de la puerta de garaje.

Esta instancia hace referencia a la programación realizada en el NodeMCU como se observa en la figura 32.

Programación de Arduino

```
client.println("<a href=\"/LED=ON\"\"><button>Turn On </button>");
```

Figura 32. Programación de Arduino botón ON

Elaborado por: David Posso y Brayan Ramírez

4.2.1.2 APK Android

Una vez realizado el diseño y la programación de las funcionalidades de los botones, AppInventor permite generar un ejecutable .apk que es la extensión para que Android instale en dispositivos móviles. Este APK se puede descargar en un computador o generando un código QR y escaneándolo desde un smartphone el cual se redirigirá a un enlace para descargar la aplicación. Estas opciones se pueden visualizar en la figura 35.

Extensión APK

Figura 33. Generar archivo con extensión APK

Elaborado por David Posso y Brayan Ramírez

4.2.1.3 Instalación del APK en el smartphone

La manera más fácil de instalar la nueva aplicación es mediante la aplicación de app inventor la cual permite escanear un código QR que da desde la página de app inventor. En la figura 34, se observar la opción para instalar la app mediante QR.

Código QR

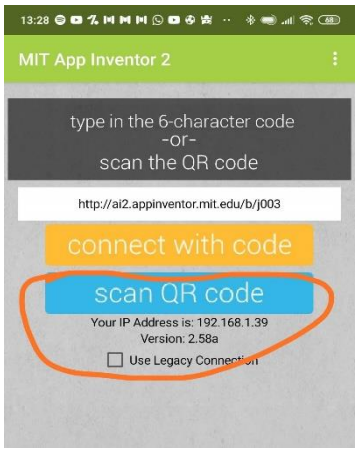


Figura 34. Escanear código QR

Elaborado por: David Posso y Brayan Ramírez

Al escanear este código QR se dirige directamente al navegador de smartphones para descargar la aplicación. En la figura 35, se puede ver la descarga en proceso.

Descarga APK

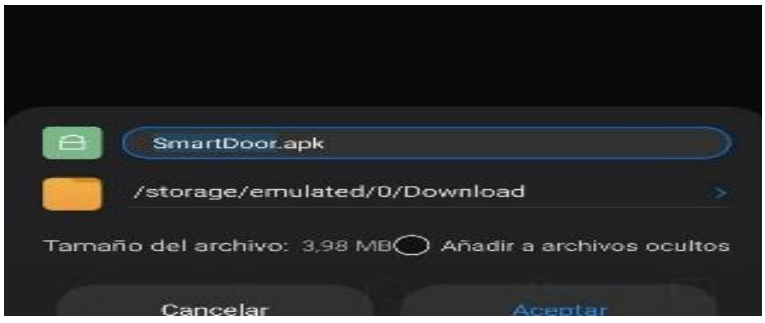


Figura 35. Descarga de archivo APK

Elaborado por: David Posso y Brayan Ramírez

Una vez descargada el APK de la aplicación se procede a instalarla en un smartphone Android. El icono que se puede ver en la figura 36, también es personalizable en AppInventor

Instalación APK

Figura 36. Instalación APK

Elaborado por: David Posso y Brayan Ramírez

Instalada la aplicación se abre la misma y está lista para utilizarla la cual ya interactúa directamente con el sistema de la puerta del motor construido anteriormente. La interfaz se puede observar en la figura 37.

Ejecución

Figura 37. Ejecución aplicación móvil

Elaborado por: David Posso y Brayan Ramírez

4.2.2 Aplicación para entorno iOS

Para el desarrollo de la app para iOS se utilizó Thunkable el cual tiene mucha similitud con el AppInventor de Google como se observa en la figura 38.

Plataforma Thunkable

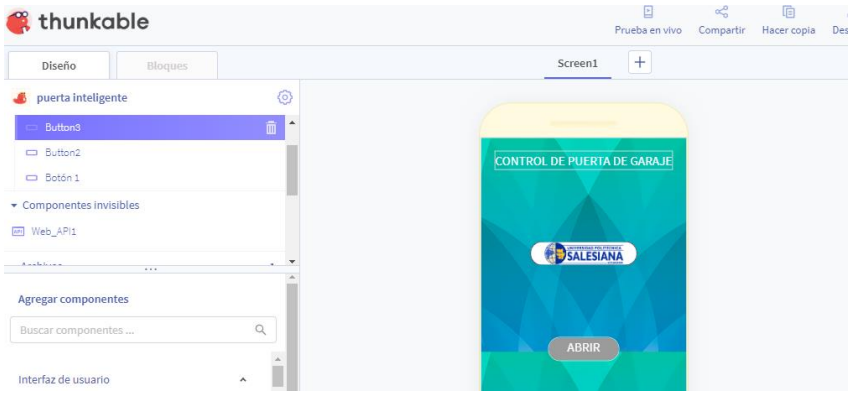


Figura 38. Plataforma Thunkable

Elaborado por: David Posso y Brayan Ramírez

Thunkable a diferencia de app inventor proporciona un entorno multiplataforma ya que al momento de generar la aplicación se la puede hacer tanto para iOS como para Android.

4.2.2.1 Programación por bloques Thunkable

Al igual que AppInventor Thunkable permite realizar programación por bloques y cada componente de la aplicación tiene una serie de bloques que al unirlos generarán una acción. La programación en Thunkable se puede ver en la figura 39.

Programación en Thunkable

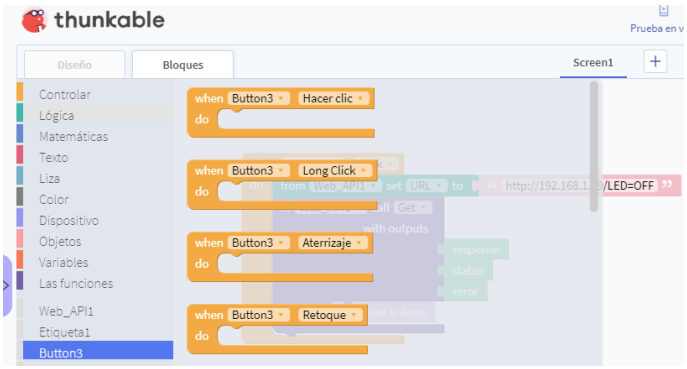


Figura 39. Programación en Thunkable

Elaborado por: David Posso y Brayan Ramírez

Para la aplicación se ha colocado un botón “ABRIR”, mediante la programación de bloques activara el motor de la puerta del garaje.

En la figura 40, está la serie bloques que se necesita para que mediante el botón de abrir se pueda accionar el relé el cual recibe la señal LED=ON.

Programación Botón

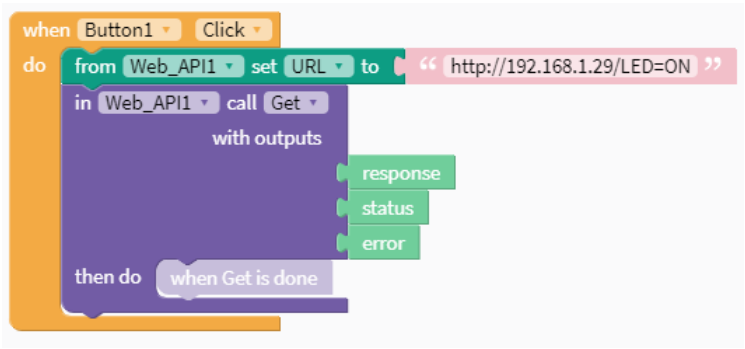


Figura 40. Programación Botón

Elaborado por: David Posso y Brayan Ramírez

4.2.2.2 IPA iOS

En Thunkable para generar el instalador para iOS, el cual debe ser una extensión IPA, se debe realizar mediante la opción Download iOS App como se observa en la figura 41.

Download

Figura 41. Download IPA

Elaborado por: David Posso y Brayan Ramírez

Se genera un enlace el cual es enviado a un correo electrónico, para descargar la aplicación para iOS. En la figura 42, muestra donde se debe ingresar el correo para recibir la app.

Enviar a email

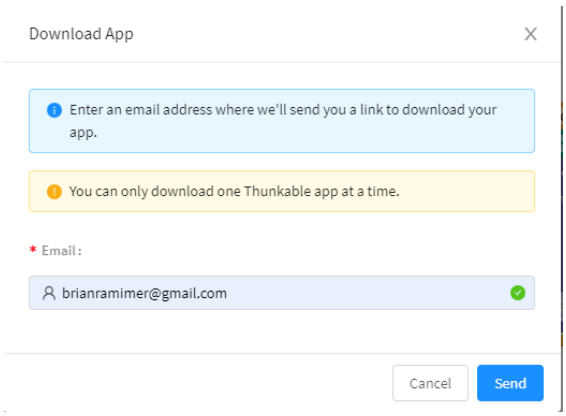


Figura 42. Enviar a email archivo IPA

Elaborado por: David Posso y Brayan Ramírez

CAPÍTULO V
PRUEBAS Y RESULTADOS

En el capítulo quinto se presenta las diferentes pruebas realizadas en el proyecto, al igual que sus resultados y diversas correcciones para presentar la solución que mejor se adapte a las necesidades del usuario.

5.1 Prueba de conexión en el circuito

5.1.1 Prueba conexión circuito 1

Tabla 9. Prueba conexión circuito 1

PRUEBA CONEXIÓN CIRCUITO 1	
REQUISITOS	✓ NodeMCU 3.5 V ✓ Relé 5V ✓ Motor
PROCEDIMIENTO	✓ Se realiza la conexión NodeMCU relé. ✓ Se conecta el relé al panel de control.
RESULTADO	Existe una falla de conexión entre el NodeMCU ya que el relé necesita un voltaje de 5V para transmitir la señal que activará el panel de control del motor.

Nota: Se obtiene el resultado de la prueba con un NodeMCU de 3.5V

Elaborado por: Brayan Ramírez y David Posso

En la tabla 9, se detalla la primera prueba del circuito se encontró un problema no tomado en cuenta al momento de comprar los diferentes componentes, existe una diferencia de voltajes entre los componentes del circuito.

5.1.2 Prueba conexión circuito 2

Tabla 10. Prueba conexión circuito 2

PRUEBA CONEXIÓN CIRCUITO 2	
REQUISITOS	✓ NodeMCU 5 V ✓ Relé 5V ✓ Motor
PROCEDIMIENTO	✓ Se realiza la conexión NodeMCU relé. ✓ Se conecta el relé al panel de control del motor.

RESULTADO	La conexión es exitosa ya que este NodeMCU posee el voltaje necesario para trasmitirlo al relé y que se active el motor.

Nota: Se obtiene el resultado de la prueba con un NodeMCU de 5V

Elaborado por: Brayan Ramírez y David Posso

En la tabla 10, se detalla la segunda prueba del circuito con componentes que trabajen con el mismo voltaje, la prueba es exitosa.

5.2 Prueba de conexión con Ubidots

5.2.1 Prueba 1

Tabla 11. Prueba 1

PRUEBA 1	
REQUISITOS	✓ NodeMCU 5V a rate 9600 ✓ Página de Ubidots
PROCEDIMIENTO	Se compila en el IDE de Arduino el código para la conexión de NodeMCU con la página de Ubidots.
RESULTADO	La conexión falla debido a que para trabajar con la librería de Ubidots es necesario que el NodeMCU trabaje con un rate de 115200 y con un voltaje de 5V. Entonces el NodeMCU utilizado solo cumple con una de estas condiciones.

Nota: Se obtiene el resultado de la prueba al realizar una conexión del NodeMCU con la plataforma Ubidots.

Elaborado por: Brayan Ramírez y David Posso

En la tabla 11, se detalla la primera prueba relacionada con la interacción entre el IDE Arduino y Ubidots, se presenta un problema debido a la velocidad con la que trabajan las librerías de Ubidots. Para solucionarlo se podía comprar nuevos componentes o utilizar un IDE diferente que reemplace a Ubidots.

5.3 Pruebas de entorno web

Se utilizó una página para comprobar la funcionalidad desde la conexión con el NodeMCU.

Tabla 12. Prueba de interfaz de usuario

PRUEBA DE INTERFAZ DE USUARIO	
REQUISITOS	✓ Plataforma web

	✓ Registro de usuario ✓ Sistema de puerta de garaje
PROCEDIMIENTO	✓ Se inicia la plataforma web ✓ El usuario se registra para poder acceder al sistema. ✓ Una vez en el sistema el usuario interactúa con la interfaz para el control de la puerta de garaje.
RESULTADO	La interfaz al ser intuitiva le permite al usuario utilizar la plataforma de una manera sencilla.

Nota: Se obtiene el resultado de la prueba de la interfaz de usuario brindándole un uso fácil e interactivo al usuario.

Elaborado por: Brayan Ramírez y David Posso

En la tabla 12, se detalla la prueba de interfaz, realizada en una página web, para verificar el funcionamiento del circuito.

Inicio aplicación web

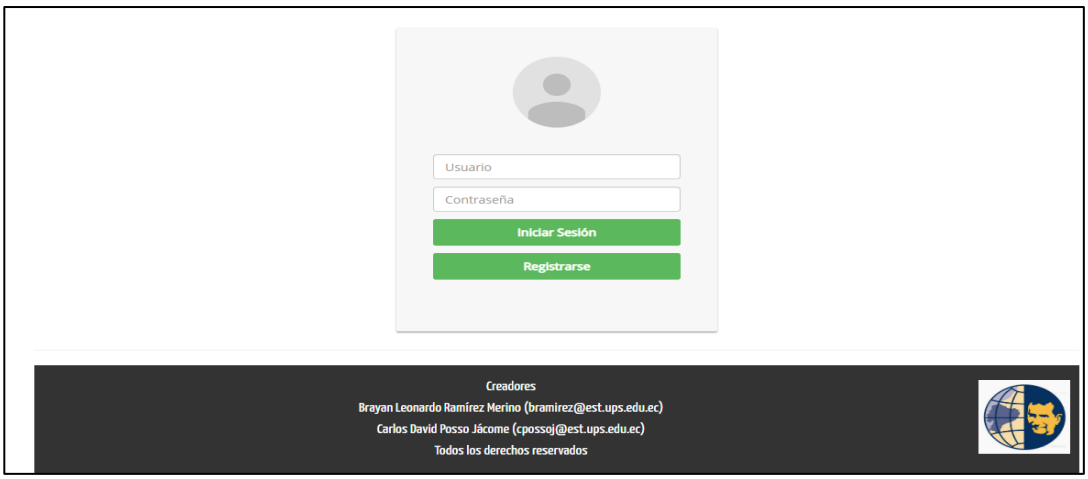


Figura 43. Pantalla de inicio aplicación Web

Elaborado por: Brayan Ramírez y David Posso

En la figura 43, se observa el inicio de la aplicación que requiere de identificación al usuario para poder utilizarla.

Tabla 13. Aplicación web desde smartphone

ACCESO A LA APLICACIÓN WEB DESDE UN SMARTPHONE	
REQUISITOS	✓ Smartphone ✓ Sistema de puerta de garaje

PROCEDIMIENTO	✓ El usuario se conecta a la aplicación web desde su smartphone. ✓ El usuario se registra e ingresa al sistema con sus credenciales.
RESULTADO	El aplicativo web funciona correctamente desde un smartphone.

Nota: Se obtiene el resultado de la prueba utilizando un smartphone para ingresar al aplicativo web dando un funcionamiento correcto.

Elaborado por: Brayan Ramírez y David Posso

En la tabla 13, la aplicación web es probada mediante un smartphone para comprobar la conexión con el circuito. No se generaron problemas en esta prueba.

Interfaz aplicación

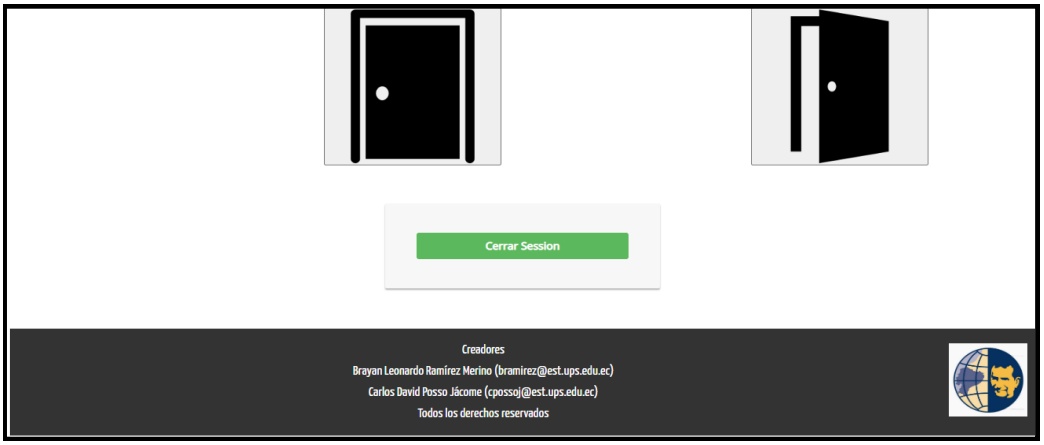


Figura 44. Interfaz para abrir y cerrar puerta de garaje

Elaborado por: Brayan Ramírez y David Posso

En la figura 44, se puede observar la interfaz web de la aplicación que se visualiza en un ordenador, a diferencia de la interfaz que estaría disponible en un smartphone como se observa en la figura 45.

Página web smartphone

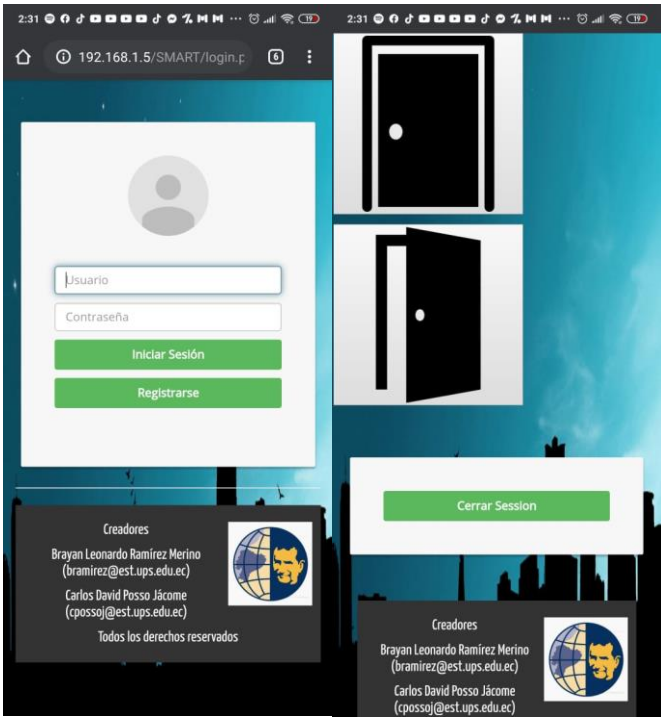


Figura 45. Página web sobre smartphone

Elaborado por: Brayan Ramírez y David Posso

Tabla 14. Prueba satisfacción

CRITERIO	PERSONAS	Observación
MUY SATISFACTORIO	0	
SATISFACTORIO	1	
INDIFERENTE	1	
POCO SATISFACTORIO	5	Ingreso al login y el navegador
NADA SATISFACTORIO	3	Ingreso al login y el navegador

Nota: Resultados de la encuesta realizada a 10 personas.

Elaborado por: Brayan Ramírez y David Posso

En la tabla 14, se detalla los resultados de la prueba de satisfacción realizada a los usuarios que utilizan el garaje mediante la aplicación móvil.

Encuesta de satisfacción

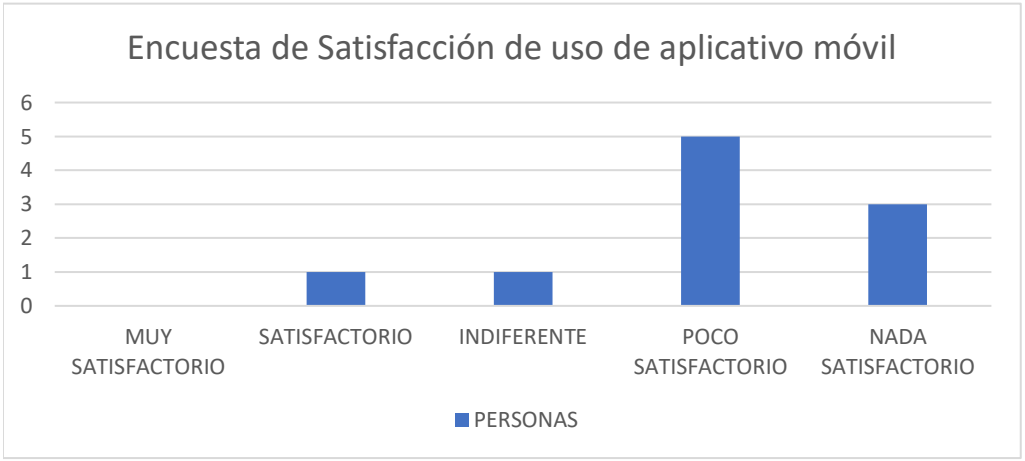


Figura 46. Encuesta de Satisfacción

Elaborado por: David Posso y Brayan Ramírez

En la figura 46, se muestra la gráfica con los resultados de la encuesta de satisfacción, la cual demuestra que el 80% de personas están poco satisfechas o nada satisfechas al usar la aplicación móvil debido al uso de un login para manipular la app.

Tabla 15.. Prueba de satisfaccion

PRUEBA DE SATISFACCIÓN	
REQUISITOS	✓ Smartphone ✓ Sistema de puerta de garaje
PROCEDIMIENTO	✓ El usuario se conecta a la aplicación web desde su smartphone. ✓ El usuario se registra e ingresa al sistema con sus credenciales. ✓ Se realiza un encuentra a 10 personas sobre la satisfacción del aplicativo web
RESULTADO	Los resultados de acuerdo con la figura 48 el 80% de los usuarios encuestados están poco o nada satisfechos con el aplicativo web ya que genera incomodidad ir al navegador del celular e ingresar o buscar cada vez el sitio web cuando se quiere utilizarlo.

Nota: Se obtiene el resultado de la prueba de satisfacción del usuario en el cual indica que el ingresar al aplicativo web desde el smartphone le resulta incómodo.

Elaborado por: Brayan Ramírez y David Posso

En la tabla 15, se detalla el procedimiento realizado para efectuar la encuesta de satisfacción y los resultados obtenidos.

5.4 Prueba con la aplicación móvil en Android y iOS

Tabla 16. Prueba Interfaz

PRUEBA DE INTERFAZ DE USUARIO EN LA APLICACIÓN MÓVIL	
REQUISITOS	✓ Aplicación en Android o iOS ✓ Sistema de puerta de garaje
PROCEDIMIENTO	✓ Instalamos la aplicación en nuestro móvil Android o iOS ✓ Conectar el smartphone a la red. ✓ Ejecutar la aplicación móvil.
RESULTADO	La aplicación se conecta satisfactoriamente al sistema de la puerta de garaje y brinda una interfaz sencilla para el usuario.

Nota: Se obtiene el resultado de la prueba de la interfaz de usuario de la aplicación móvil

Elaborado por: Brayan Ramírez y David Posso

En la tabla 16, se detalla la prueba realizada a la aplicación móvil desarrollada en AppInventor y Thunkable para comprobar la conexión con el circuito, no presentan problemas.

Tabla 17. Prueba Satisfacción Criterio

CRITERIO	PERSONAS
MUY SATISFACTORIO	6
SATISFACTORIO	2
INDIFERENTE	2
POCO SATISFACTORIO	0
NADA SATISFACTORIO	0

Nota: Para esta prueba se pidió a 10 personas usar el aplicativo móvil.

Elaborado por: Brayan Ramírez y David Posso

En la tabla 17, se muestra el criterio de los diferentes encuestados que probaron la aplicación móvil sin tener que pasar por un login.

Encuesta Satisfacción aplicativo móvil

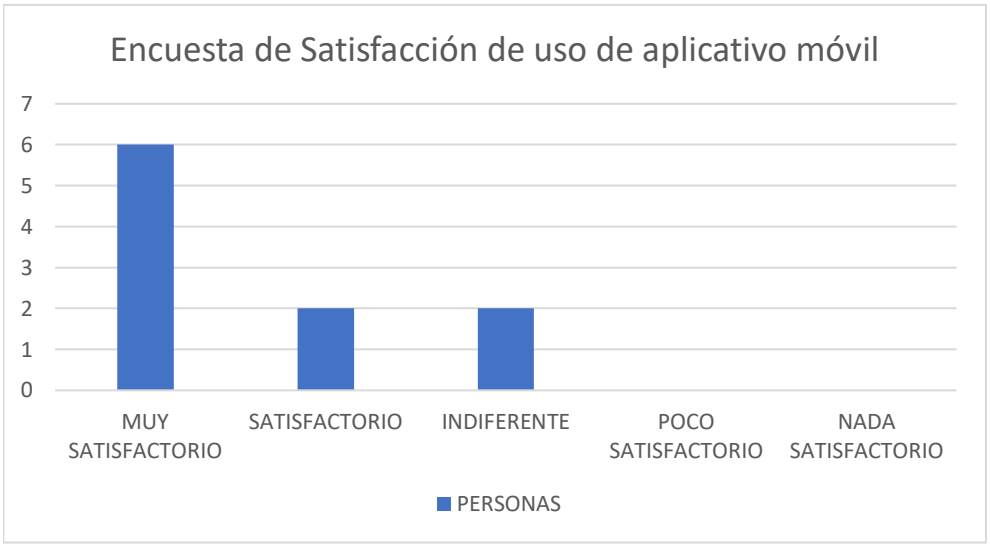


Figura 47. Encuesta aplicativo móvil

Elaborado por: David Posso y Brayan Ramírez

En la figura 47, muestra que un 80% de los encuestados están muy satisfechos o satisfechos con la aplicación móvil para controlar el garaje.

Tabla 18. Prueba satisfacción

PRUEBA DE SATISFACCIÓN	
REQUISITOS	- Aplicación en Android - Sistema de puerta de garaje
PROCEDIMIENTO	-Instalamos la aplicación en nuestro móvil Android. - Conectar el smartphone a la red. - Ejecutar la aplicación móvil.
RESULTADO	El resultado según la figura 49 el 80% de las personas encuestadas les es más fácil usar una aplicación móvil sencilla y sin ningún login, ya que el smartphone al proporcionar sus propias medidas de seguridad genera incomodidad en el cliente como se puede observar en la figura 48

Nota: Prueba de usabilidad de 10 usuarios

Elaborado por: David Posso y Brayan Ramírez

En la tabla 19, se detalla el procedimiento a realizar para efectuar la segunda prueba de satisfacción.

5.5 Pruebas de compatibilidad

En estas pruebas lo que se hizo es instalar la aplicación en diferentes versiones de Android e IOS.

Tabla 19. Compatibilidad con Android

VERSION DE ANDROID	RESULTADO
ANDROID 10	La aplicación se ejecutó correctamente
ANDROID 9	La aplicación se ejecutó correctamente
ANDROID 8	La aplicación se ejecutó correctamente
ANDROID 7	La aplicación se ejecutó correctamente
ANDROID 6	La aplicación se ejecuta correctamente
ANDROID 5	Al abrir la aplicación no muestra la interfaz de usuario correctamente.
ANDROID 4	No permite la instalación de la aplicación móvil

Nota: Compatibilidad con diferentes versiones de Android

Elaborado por: David Posso y Brayan Ramírez

En la tabla 19, se muestran las pruebas de compatibilidad con las diferentes versiones de Android, los resultados muestran que la aplicación funciona correctamente desde la versión 6 del sistema Android.

Tabla 20. Compatibilidad en iOS

VERSION DE ANDROID	RESULTADO
iOS 13	La aplicación se ejecutó correctamente
iOS 12	La aplicación se ejecutó correctamente
iOS 10	La aplicación se ejecutó correctamente
iOS 8	La aplicación se ejecutó correctamente
iOS 3	No permita instalar la aplicación móvil.

Nota: Compatibilidad con diferentes versiones de iOS

Elaborado por: David Posso y Brayan Ramírez

En la tabla 20, se muestran las pruebas de compatibilidad con las diferentes versiones de iOS, los resultados muestran que la aplicación funciona correctamente desde la versión 8 del sistema iOS.

5.6 Resultado de pruebas de efectividad

Una vez realizada las pruebas anteriores, se procede a efectuar una verificación cuantitativa de la solución planteada y de la aplicación móvil en base a una serie de criterios.

Tabla 21. Cumplimiento de los criterios

PORCENTAJE DEL CUMPLIMIENTO DE LOS CRITERIOS	
98-100	CUMPLIMIENTO EXCELENTE
65-97	CUMPLIMIENTO MUY BUENO
41-64	CUMPLIMIENTO BUENO
21-40	CUMPLIMIENTO REGULAR
Menor 20	INACEPTABLE

Nota: Porcentaje de cumplimiento para evaluar pruebas de efectividad

Elaborado por: David Posso y Brayan Ramírez

En la tabla 21, se detallan los criterios a usar para comprobar la efectividad de la aplicación desarrollada con porcentajes en la escala del 0 a 100.

Criterios

- ✓ Pruebas de la conexión a la red mediando la plataforma web.
- ✓ Pruebas de la conexión NodeMCU a una a la red.
- ✓ Prueba de funcionamiento del circuito del sistema de puerta de garaje.
- ✓ Prueba de conexión aplicación Android con a la red.
- ✓ Prueba de conexión aplicación IOS con a la red.

Tabla 22. Criterios

CRITERIOS	%	OBSERVACIONES
Pruebas de la conexión con una IP privada mediando la plataforma web	90	El sistema funciona, pero presenta un error en la conexión al iniciar al mismo tiempo con la aplicación móvil.
Pruebas de la conexión NodeMCU a una IP privada	100	La conexión NodeMCU es satisfactoria
Prueba de funcionamiento del	100	La combinación de cada componente brinda un funcionamiento óptimo con el entorno web y la aplicación móvil.

circuito del sistema de puerta de garaje		
Prueba de conexión aplicación Android con IP privada	100	La conexión de la aplicación Android funciona correctamente.
Prueba de conexión aplicación iOS con IP privada	100	La conexión de la aplicación iOS funciona correctamente.
SUMA	490	
TOTAL	98	

Nota: Porcentaje de cumplimiento según los criterios evaluados

Elaborado por: David Posso y Brayan Ramírez

Como se puede observar en la tabla 22, anterior la aplicación web presenta un pequeño error de conexión al iniciarse al mismo tiempo que la aplicación móvil por lo que su porcentaje de cumplimiento es menor.

$$500 * 100$$

$$499 \quad X \quad = 98.00\%$$

$$100\% - 98.00\% = 2\%$$

Tabla 23. Porcentaje de proyecto cumplido

PORCENTAJE DEL CUMPLIMIENTO DEL PROYECTO	
98-100	SE HA LOGRADO EL OBJETIVO DEL PROYECTO
65-97	SE HA LOGRADO EL OBJETIVO, PERO PUEDE MEJORAR
41-64	SE DEBE MEJORAR PARA LOGRAR EL OBJETIVO
21-40	BUSCAR MEJORES Y NUEVAS ALTERNATIVAS PARE REPLANTEAR LA SOLUCÓN
Menor 20	NO SE ALCANZÓ EL OBJETIVO Y INACEPTABLE

Nota: Porcentaje del proyecto cumplido según el criterio evaluado

Elaborado por: David Posso y Brayan Ramírez

De acuerdo a la tabla 23, el margen de error del presente proyecto es de un 2%, por lo que existe una efectividad del 98% lo que hace viable a la solución planteada.

5.7 Rentabilidad del Sistema

En la tabla 24, se realiza una comparación entre el precio promedio en el mercado actual ecuatoriano y el costo total invertido en el proyecto. En la tabla 7, se encuentran los costos del circuito a este precio se le aumenta el costo de mano de obra el cual será el 25% adicional del costo del circuito tomando en cuenta que la instalación se la puede realizar en un tiempo máximo de 1 hora, lo cual equivale a 3.50 dólares por hora. Y un 25% adicional del costo del circuito será considerado como ganancia neta. Sumando los costos adicionales el proyecto sería valorado en 21 dólares. (Los precios encontrados en el mercado no incluyen el valor de la instalación.)

Tabla 24. Comparativa precios

Nombre	Descripción	Precio (dólares)
Circuito del proyecto	NodeMCU, Módulo Relé optoacoplador de 5V, cable gemelo #16, mano de obra y ganancia.	21
Modulo del mercado en la actualidad	Modulo inteligente WiFi Tiene dos relés	44,80

Nota: Comparación de precio promedio del mercado con respecto al costo del circuito en el proyecto

Elaborado por: David Posso y Brayan Ramírez

5.7.1 Cálculo de rentabilidad del sistema

44,80(valor mercado) – 21(costo solución) = 23,80(Diferencia de precios)

23,80 * 100 / 44,80 = 53.125% Porcentaje de ahorro con circuito del proyecto

Existe un beneficio económico del 53.125% con respecto al valor en el mercado de dispositivos similares a la solución del proyecto.

5.8 Pruebas de funcionalidad

Durante el mes de agosto se realizó un candelario en Excel para monitorizar el funcionamiento de la aplicación como se puede ver en la figura 48, se tuvo un problema externo que afecto a la funcionabilidad del producto debido a un fallo eléctrico, que impidió su conexión a la red y su normal funcionamiento.

Calendario funcionalidad del prototipo

Figura 48. Calendario de funcionalidad

Elaborado por: David Posso y Brayan Ramírez

En la figura 49, se muestran los resultados de la prueba de funcionalidad de la aplicación móvil con un 96.77% de correcta funcionabilidad y un 3.23% de no funcionabilidad debido a problemas externos.

Resultados de Funcionabilidad

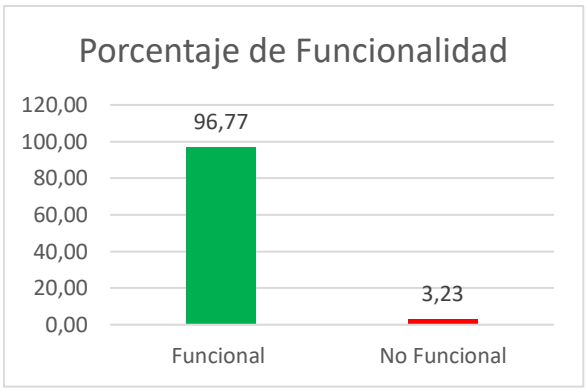


Figura 49. Grafica de funcionalidad

Elaborado por: David Posso y Brayan Ramírez

5.9 Pros y Contras de la aplicación móvil

Aun tomando en cuenta el beneficio que genera el no utilizar controles físicos, como lo son el ahorro en controles nuevos y la inseguridad que producía extraviar uno de los controles físicos, también se consideraron distintos puntos de vista de pros y contras de la aplicación. En la tabla 25, se detallan los pros y contras los cuales son; el beneficio en seguridad que generan los sistemas operativos tanto como Android y iOS en los smartphone, permitiendo bloquear el celular en caso de perdida y la única forma de volver a utilizarlo sería restaurándolo de fábrica perdiendo toda la información en el proceso, tomando este mismo punto se puede decir que la aplicación será personal para cada usuario que necesite de la aplicación, al funcionar localmente

se evita ciberataques a través de internet debido a que no usa puertos del Router para funcionar, y por último la aplicación no envía información relevante, solo envía señales de 0 y 1 para su funcionamiento. En contra la aplicación debe volver a ser instalada mediante un archivo en formato APK o IPA en caso de pérdida del dispositivo.

Tabla 25. Pros y Contras de la aplicación

PROS	CONTRAS
La aplicación es personal debido a que es instalada en móviles cuyo uso es individual para cada usuario.	Perdida del dispositivo móvil sería necesario volver a instalar la aplicación mediante un archivo APK o IPA.
La aplicación no comparte datos solamente envía señales para comunicarse con el motor.	
Al funcionar localmente es más segura debido a que no se usa ningún puerto por donde recibir un ciberataque.	
La aplicación instalada en Android, gracias al sistema operativo proporciona defensa por capas lo cual evita la instrucción y proporciona seguridad al usuario.	
La aplicación instalada en iOS, gracias al sistema operativo ofrece diferentes niveles de protección garantizando a las apps estar exentos de software desconocido y malicioso para evitar su manipulación.	

Nota: Análisis de los pros y contras en la seguridad de la aplicación móvil.

Elaborado por: David Posso y Brayan Ramírez

CONCLUSIONES

- Mediante la elaboración del prototipo se consigue la reducción de costos de alrededor del 53.125% con respecto al valor de productos con funciones similares en el mercado actual.
- Después de realizar las pruebas de satisfacción 80% de los usuarios, presentaron quejas sobre el manejo del login como método de autenticación, debido que los smartphones ya poseen sus propias funcionalidades de seguridad. Por lo tanto, se rediseño la aplicación retirando la función de login para conseguir como resultados en la siguiente prueba satisfacción que 90% de los usuarios están conformes con la aplicación.
- Al terminar con los diseños, configuración he implementación del sistema de la puerta de garaje, se condujo una serie de pruebas para obtener 98% de efectividad, con solo 2% de error en efectividad debido a un conflicto con la aplicación web y la aplicación móvil y 96.77% de funcionalidad contra un 3.23 % de no funcionalidad debido a un fallo eléctrico de acuerdo a los resultados obtenidos.
- La elaboración del prototipo permite obtener varios beneficios en seguridad los cuales son; conexión local para evitar ciberataques desde la red global, funcionamiento únicamente con señales sin necesidad de información relevante del usuario, seguridad implementada por los sistemas operativos iOS y Android que restringen el acceso no autorizado al dispositivo, y suprimir el uso de controles físicos los cuales son de fácil perdida o hurto con los cuales se pueden efectuar robos a domicilios.

RECOMENDACIONES

- Es recomendable antes de cualquier conexión tomar en cuenta el voltaje mínimo necesario con el que trabaja cada dispositivo electrónico para evitar inconvenientes en su ejecución.
- Para futuras investigaciones relacionadas con motores similares se recomienda usar ESP8266 integrado al módulo relé acoplador de 5v, los cuales siendo un solo dispositivo mejora su portabilidad y reduce su costo.
- Se recomienda trabajar con dispositivos que cumplan las características necesarias para la plataforma Ubidots, debido a sus robustas características que permite generar sistemas más complejos.
- Se recomienda el desarrollo en AppInventor y Thunkable plataformas gratuitas para reducir costos del proyecto.

LISTA DE REFERENCIAS

Angular. (20 de Julio de 2020). *Angular*. Obtenido de Angular: https://angular.io/guide/architecture

App Inventor en Español. (20 de Julio de 2020). *App Inventor en Español*. Obtenido de App Inventor en Español: https://sites.google.com/site/appinventormegusta/presentaciones-google-docs

AppInvetor. (20 de Julio de 2020). *The MIT App Inventor Library: Documentation & Support*. Obtenido de The MIT App Inventor Library: Documentation & Support: https://appinventor.mit.edu/explore/library

bibing. (20 de Julio de 2020). *REDES INALÁMBRICAS DE*. Obtenido de REDES INALÁMBRICAS DE: http://bibing.us.es/proyectos/abreproy/11761/fichero

CEJA, J., RENTERIA, R., RUELAS, R., & OCHOA, G. (20 de Julio de 2020). *Módulo ESP8266 y sus aplicaciones en el internet de las cosas* . Obtenido de Módulo ESP8266 y sus aplicaciones en el internet de las cosas : https://www.ecorfan.org/republicofperu/research_journals/Revista_de_Ingenieria_Electrica/vol1num2/ECORFAN_Revista_de_Ingenier%C3%ADa_El%C3%A9ctrica_VI_N2_3.pdf

Deloitte España. (20 de Julio de 2020). *Ionic: historia de uno de los principales frameworks visuales*. Obtenido de Ionic: historia de uno de los principales frameworks visuales: https://www2.deloitte.com/es/es/pages/technology/articles/Ionic-principales-framework-visuales.html

Eninter. (20 de Julio de 2020). *TIPOS DE PUERTAS DE GARAJE*. Obtenido de TIPOS DE PUERTAS DE GARAJE: https://www.eninter.com/blog/tipos-puertas-garaje/

Hardwarelibre. (20 de Julio de 2020). *ESP8266: el módulo WIFI para Arduino.* Obtenido de ESP8266: el módulo WIFI para Arduino: https://www.hwlibre.com/esp8266/

Hernández Rodríguez, A., & Guzmán Hernández, D. (2018). Prototipo de cerradura inteligente usando aplicación en android. *Recursos electronicos*.

Loyola Mendoza, A. A. (2018). Sistema Domótico con Aplicación Móvil en Android para mejorar el control de la energía y acceso a puertas en un hogar. *Universidad Cesar Vallejo*.

MercadoLibre. (2020). *mercado libre*. Obtenido de https://articulo.mercadolibre.com.ec/MEC-425874788-motor-puerta-de-garaje-abre-la-puerta-desde-el-celular-wifi-_JM?quantity=1#position=1&type=item&tracking_id=b1b7db59-07e6-49a7-830f-7675018f7075

Nayrolles, M. (2018). *Angular Design Patterns.* BIRMINGHAM - MUMBAI: Packt Publishing, Limited.

Propuesta de ejercicios practicos de desarollo de aplicaciones hibridas. (20 de Julio de 2020). Obtenido de Propuesta de ejercicios practicos de desarollo de aplicaciones hibridas: http://oa.upm.es/49346/1/PFC_CESAR_GONZALEZ_FERNANDEZ.pdf

SpeedCheck. (20 de Julio de 2020). *Gateway*. Obtenido de Gateway: https://www.speedcheck.org/es/wiki/gateway/

Thunkable. (20 de Julio de 2020). *Thunkable*. Obtenido de Thunkable: https://docs.thunkable.com/