ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“SISTEMA DE ADMINISTRACIÓN DE LLAVES VIRTUALES Y GESTIÓN DE PERMISOS PARA DISPOSITIVOS DE CONTROL DE ACCESOS”

PROYECTO INTEGRADOR

Previo a la obtención del Título de:

Ingeniero en Computación

Presentado por:

Juniver Jair Román Macías

Paul Domingo Estrada León

GUAYAQUIL – ECUADOR

AÑO: 2019

AGRADECIMIENTOS

Nuestros más sinceros agradecimientos a Dios por ser el guía durante el trayecto de nuestras vidas brindando apoyo y fortaleza en aquellos momentos de dificultad y de debilidad, a nuestros padres que siempre estuvieron apoyándonos, promoviendo nuestros sueños, por confiar y creer en nuestras expectativas, por los consejos, valores y principios que nos han inculcado.

Agradecemos a los docentes de la Escuela Superior Politécnica del Litoral, por los conocimientos compartidos a lo largo del desarrollo de nuestra carrera, de manera especial, al máster Rodrigo Saraguro tutor de nuestro proyecto, quien ha guiado con su conocimiento, paciencia, y su rectitud como docente.

DEDICATORIA

El presente proyecto se lo dedicamos a nuestros padres, quienes con su apoyo, esfuerzo y amor nos acompañan de alguna u otra forma en nuestros sueños y metas.

A nuestros amigos y personas allegadas que siempre creyeron en nuestras capacidades y estuvieron allí para dar una mano cuando se necesitaba de su ayuda y muchas veces sin ser necesario estaban pendientes.

DECLARACIÓN EXPRESA

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

Juniver Jair
Román Macías

Paul Domingo
Estrada León

-

RESUMEN

La tecnología avanza a pasos agigantados con el objetivo de mejorar las actividades que realizamos en nuestro diario vivir, actividades muy complejas pueden ser resueltas con tecnología que las convierte en más simples. El uso de llaves físicas presenta muchas falencias de seguridad y usabilidad, tales como la fácil replicación y perdida del objeto. En el presente proyecto desarrollamos un sistema computacional que permite la administración de llaves virtuales y gestión de permisos, comparándonos con soluciones existentes con el objetivo de generar un producto de software con facilidad de integración hacia un futuro hardware a desarrollar.

El proyecto se desarrolló mediante la metodología Scrum, y tiene tres pilares fundamentales; servidor (API), administrador web, aplicación móvil (Android) y dispositivo de control de acceso (prototipo en raspberry). Todo esto desarrollando en lenguajes de programación y frameworks de alta escalabilidad como son Django (Python) y Android Studio (Java). Además, consideramos estándares de seguridad de la información como la autenticación, uso de tokens dinámicos y comunicaciones encriptadas.

Como resultados se obtuvo la alta eficiencia del software, tanto bajos tiempos en comunicaciones, como en compatibilidad de tecnologías. El servidor basado en un API REST se adapta a cualquier lenguaje de programación mediante peticiones con respuestas en JSON siempre y cuando repete el formato de las peticiones. En las diferentes pruebas de usuario, el sistema funciona con las condiciones ideales planteadas, sin embargo, en la documentación hemos considerado situaciones en las que podrían afectar a la disponibilidad de la solución.

Palabras Clave: SCRUM, API REST, peticiones, dispositivo de control de acceso, Django, Android Studio.

ABSTRACT

Technology advances by leaps and bounds in order to improve the activities we do in our daily lives, very complex activities can be solved with technology that makes them simpler. The use of physical keys presents many safety and usability flaws, such as the easy replication and loss of the object. In this project we develop a computer system that allows the administration of virtual keys and permission management, comparing us with existing solutions with the objective of generating a software product with ease of integration towards a future hardware to be developed.

The project was developed using the Scrum methodology and has three fundamental pillars; server (API), web administrator, mobile application (Android) and access control device (prototype in raspberry). All this developing in programming languages and high scalability frameworks such as Django (Python) and Android Studio (Java). In addition, we consider information security standards such as authentication, use of dynamic tokens and encrypted communications.

As a result, the high efficiency of the software was obtained, both low communication times and technology compatibility. The server based on a rest API adapts to any programming language through requests with responses in JSON as long as it repeats the format of the requests. In the different user tests, the system works with the ideal conditions raised, however, in the documentation we have considered situations in which they could affect the availability of the solution.

Keywords: *SCRUM, API rest, requests, access control device, Django, Android Studio.*

ÍNDICE GENERAL

ABREVIATURAS

AES	Advanced Encryption Standard
API	Application Programming Interface
REST	Representational State Transfer
SMS	Short Message Service
NFC	Near Field Communication
SDK	Development Kit
IDE	Integrated Development Environment
RSSI	Received Signal Strength Indicator

SIMBOLOGÍA

s	Segundo
m	Metros

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1. INTRODUCCIÓN

En el presente capítulo comenzaremos describiendo la problemática mediante ejemplos y acciones que esta nos lleva a realizar; luego resaltaremos como las aplicaciones de domótica e internet de las cosas (IoT) están cada vez más involucradas en nuestras vidas. En la descripción del problema mostraremos estadísticas de una posible consecuencia de nuestra problemática y las falencias que tiene en su uso cotidiano, luego justificaremos la importancia de nuestro proyecto y estableceremos los objetivos. Por último, realizaremos investigación y análisis acerca de cómo se está manejando el problema alrededor del mundo con las soluciones que ya existen.

La mayoría de las personas han perdido las llaves de sus puertas alguna vez en su vida, lo cual causa frustración, inseguridad y lleva a tomar medidas drásticas como el remplazo físico de la seguridad de la puerta que se lo conoce como cerradura, por el contrario, esto genera gastos adicionales. Pero existe algo que preocupa más en todo esto y es el conocer quien ha ingresado a nuestras puertas; las situaciones aquí pueden ser muchas, como sabemos las llaves físicas se pueden duplicar con facilidad, en caso de pérdida un tercero encuentra las llaves o puede ser que nuestra puerta este abierta, por citar algunos ejemplos.

No es de sorprendernos el hecho de poder apagar un foco estando a kilómetros de distancia del interruptor, o poder controlar a nuestros electrodomésticos desde cualquier lugar que nos encontremos; todo esto es gracias a la domótica y el internet de las cosas. En la actualidad están presentes en la mayoría de los objetos que utilizamos, permitiendo que ciertas tareas se realicen con mayor precisión y rompiendo barreras como la distancia [3].

El presente proyecto se enfoca en el diseño de una solución futurista, tecnológica y amigable para la gestión de llaves virtuales para dispositivos de control de acceso. Puede ser implementado en muchos lugares donde se requiera acceso a un sitio determinado, esto se logra al integrar IoT en dispositivos de control de acceso, de esta manera el usuario podrá interactuar de manera cómoda y segura, accediendo con tan solo una pulsación en su smartphone, además gozar de ciertas funcionalidades extras, como compartir accesos a terceros, revisar registros de accesos, bloquear o desbloquear el dispositivo de manera remota, entre otras.

1.1 Descripción del problema

Al momento de extraviar nuestras llaves nos volvemos vulnerables a un sin número situaciones que van en contra de nuestra integridad, uno de los más frecuentes es el robo en viviendas. A nivel de Ecuador se registra una considerable tasa de robos en vivienda, como lo muestra en la Figura 1.1, el INEC (Instituto Nacional de Estadísticas y Censos) registra que, en el periodo de septiembre del 2010 hasta agosto del 2011, 4 de cada 100 hogares fueron víctimas de robos a la vivienda. [1]

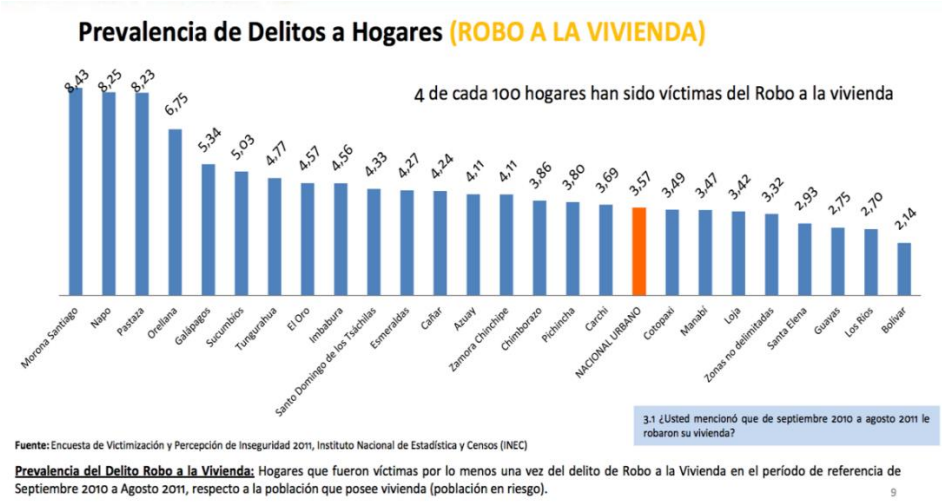


Figura 1.1 Estadísticas de robos en viviendas [1]

Por lo tanto, se debe considerar a nuestras llaves como un elemento personal muy sensible, ya que son las que nos permiten ingresar nuestras residencias, trabajos y cualquier lugar que contenga una cerradura que las utilice. En la actualidad existen varios tipos llaves, las llaves físicas tradicionales que como ya las conocemos son de metal y unas de la más comunes, las llaves magnéticas que son una tarjeta que con una cinta magnética tiene algún tipo de identificación para él acceso y las llaves digitales son claves o implementaciones de la domótica en conjunto con el internet de las cosas.

Todos estos tipos de llaves de cierta forma presentan limitantes que afectan en la integridad de nuestras cerraduras y a su vez estas cerraduras se ven afectadas por que en la actualidad es muy fácil conocer el funcionamiento de esta, esto ha inducido a que las personas desarrollen técnicas para hacerlas aún más vulnerables. Las llaves físicas son endebles: son fáciles de duplicar, toma un par de segundos para que con incluso la captura de una fotografía se pueda imprimir una réplica exacta de la misma; las perdemos con mucha facilidad, y esto incurre a hacer un remplazo de cerradura que la mayoría de las veces no lo hacemos por factores económicos o por no medir los riesgos que esto podría ocasionar.

Los ataques hacia cerraduras que utilizan una llave física son muchos, entre los dos más usados están bumping, que mediante la inserción de una llave falsa y con la ayuda de golpes trata de separar a los pistones de los contrapistones; e impresioning la cual consisten en introducir una lámina en la cerradura y cuando el propietario ingrese la llave a esta quede calcada en dicha lamina de tal forma se obtiene una copia idéntica a la original. El libro también menciona lo fácil que es duplicar una llave magnética, ya que la mayoría de ellas utiliza NFC (Near Field Communication) lo cual es leíble con sensores o incluso con un Smartphone. [5]

La seguridad de una puerta o acceso físico convencional depende de dos componentes: llave y cerradura. La llave de cierta forma representa el permiso para acceder a una cerradura, la cual debe aprobar o negar dichos permisos dependiendo de la legitimidad de estos [7]. El presente proyecto se enfoca en el primer componente el cual es la llave, que en nuestra solución es una abstracción de una llave física convencional que denominamos llave digital virtual, la misma que se puede manipular utilizando tecnología del medio como smartphones e internet.

1.2 Justificación e importancia

El mercado de la domótica en conjunto al internet de las cosas está en auge por lo cual las empresas desarrolladoras de soluciones identifican problemáticas y ven oportunidades de lanzar productos con mejoras a lo existente en el mercado, el uso de llaves físicas posee muchos problemas de interacción y seguridad con el usuario por ende se propone la implementación de una solución que abstraiga la llave física tradicional como la conocemos a un concepto más tecnológico y seguro.

Soluciones a problemáticas de manejo de cerraduras actualmente existen en el mercado internacional (Norte América, Asia y Europa), cada una de ellas resolviendo el problema de diferentes maneras, es decir, el producto propuesto cuenta con diferentes características, funcionalidades y precios. En Ecuador existen muy pocas empresas que desarrollen software para cerraduras inteligentes, por ejemplo, Retail, los cuales manipulan el dispositivo de control de acceso mediante electrónica y no con un adecuado sistema, por lo cual no hay un producto que lidere considerando las limitaciones y problemas existentes en el país. Las soluciones existentes en el mercado quedan obsoletas por la falla de la electricidad o internet, ya que son muy dependientes a estas. Entre soluciones mundiales tenemos como los de: Glue Smart Lock, Klevio,

OpenKey, ZKTeco, Kaadas por citar unos ejemplos, con limitantes como el uso únicamente de internet (conexión de cableada o inalámbrica), dependen de la electricidad local; lo cual atenta contra la disponibilidad el mismo que es un principio de la seguridad.

1.3 Objetivos

1.3.1 Objetivo General

Crear un sistema de administración de llaves virtuales, mediante el desarrollo de un servicio en la nube, un administrador web y una aplicación Android que interactúen entre sí, para la gestión de permisos en dispositivos de control de acceso.

1.3.2 Objetivos Específicos

- Crear un modelo que exponga los componentes del sistema y como estos interactúan entre sí.
- Desarrollar una plataforma web que permita la administración de usuarios, llaves y accesos a dispositivos e implementar una aplicación móvil para la interacción del usuario con el dispositivo de control de acceso, la cual estará comunicada mediante bluetooth.
- Implementar la solución de administración usando lenguaje Python con ayuda del framework Django, y la aplicación Android mediante el software Android Studio.

1.4 Marco Teórico

En estudios previos, se han construido soluciones que implementan Bluetooth, internet, conexiones wifi, SMS, entre otras para establecer

comunicaciones entre el dispositivo (cerradura) y el usuario (aplicación móvil), ciertos integran en sus modelos un servidor en la nube para la gestión de comunicaciones entre los dos extremos y operaciones administrativas, otras de estas proponen incluso arquitectura para la construcción de una cerradura [3]. Muchos de estos estudios se enfocan en soluciones de bajo costo, ya que comercialmente existen muchas opciones, pero casi todas de un costo considerablemente elevado.

Comercialmente existen soluciones para la gestión de cerraduras utilizando domótica en conjunto al internet de las cosas, en la tabla 1 se muestra tres soluciones comerciales. La tabla 1 muestra los productos Klevio, Danalock V3 y Nest x Yale, en las cuales refleja cuatro características importantes: tipo de comunicación que la llave digital virtual utiliza para conectarse con la cerradura, si incluye o no una llave física tradicional para el acceso alternativo, usa fuente de poder en caso de fallo de la red eléctrica local, el usuario puede compartir accesos a quien él quiera y por último un registro de las acciones realizadas.

Producto	**Comunicación con cerradura**	**Llave física**	**Fuente de poder externa**	**Compartir accesos**	**Registro de actividad**
Klevio	Wifi - 4G	Si	No	No	Si
Kaadas K9	Bluetooth	SI	No	No	Si
Kevo Contemporary	Wifi	Si	No	Si	Si
Samsung SHP-DR708	Bluetooth	Si	No	Si	Si
Digital Keys LDK400	NFC	SI	Si	SI	Si

Tabla 1.1.1 Comparativa de soluciones comerciales

De la tabla 1.1 podemos inferir que ninguna solución comercial es totalmente a prueba de fallos, por ejemplo: el producto de Klevio no utiliza batería, esto quiere decir que en caso de fallo del tendido eléctrico local esta queda inoperable y no se puede compartir la llave, sería de poca usabilidad que un único usuario deba ser el que autorice abrir una cerradura donde ingresan múltiples personas; Por otro lado, el producto de Danalock V3 y Nest x Yale si tienen redundancia en el sistema eléctrico siendo a prueba de fallas pero su conexión depende del Wifi el cual su disponibilidad también depende del tendido eléctrico del hogar. Y, por último, está el producto de Digital Keys, que utiliza tecnología NFC para establecer comunicación con la cerradura. Es de esta forma que las soluciones presentadas en la tabla 1.1 crean un círculo de dependencias que en casos particulares como los presentados su funcionalidad falla.

La mayoría de las soluciones centralizan enfoques en la cerradura como un objeto que sea capaz de controlarlo todo, el estudio titulado “Door-automation system using bluetooth-based android for mobile phone” propone la implementación de una llave bluetooth para abrir una cerradura realizando un prototipo con solenoide, Arduino y aplicación Android, donde la única comunicación que existe entre el smartphone con la cerradura vía bluetooth [4]. La seguridad es uno de los temas más difíciles de controlar en soluciones con domótica e internet de las cosas, ya que por más minimalistas que parezcan muchas veces la información que extraen o las acciones que estas realicen pueden afectarnos directa o indirectamente. Todo esto se lo puede controlar utilizando un gestor de permisos, mediante la implementación de un backend que controle y autorice al hardware realizar acciones si y solo si la orden es legítimamente válida según el conjunto de normas y reglas establecidas al usuario [2].

En el presente estudio nos enfocaremos a cambiar la forma que operan actualmente las soluciones comerciales y con retroalimentación de estudios anteriores mejorar la seguridad del usuario. Por ende, esta solución estará compuesta de tres componentes principales: La nube, la aplicación móvil y la cerradura. La nube, es el backend que estará encargado de gestionar todas las funciones administrativas necesarias para mantener operatividad y seguridad entre los dos extremos (usuario y cerradura).

La aplicación móvil, que es la herramienta que permite al usuario interactuar con el sistema estará desarrollada para la plataforma Android con el fin de alcanzar la mayor tasa de compatibilidad posible con los smartphones de los usuarios en el mercado. La cerradura será un dispositivo creado a partir de un raspberry pi porque permite fácil integración de código para conexión con la nube, junto a un módulo de bluetooth agregado que será el puente de la conexión entre el smartphone del usuario y el dispositivo de cerradura, y por último un conjunto de Leds que expondrán el estatus momentáneo del dispositivo.

CAPÍTULO 2

2. METODOLOGÍA

En la presente sección se detallará de lo general hacia lo más específico la metodología de planeación y desarrollo del proyecto con la necesidad de dar la mejor alternativa de solución al problema. Además, describiremos el diseño de la solución: definición de las tecnologías, metodología de desarrollo de software, esquema general del producto, backend, aplicación móvil, comunicaciones, seguridades y futuros proyectos.

2.1 Definición de tecnologías

El presente proyecto se desarrolla con tecnologías de código abierto y con lenguajes de programación que permitan la escalabilidad de software para futuras mejoras e implementaciones. A continuación, se listan las tecnologías a utilizar, así como una breve descripción de su utilidad y justificación de su elección.

- Ubuntu 18.04, es en sistema operativo de código abierto con distribución Linux, muy usado en servidores y computadores ya que cuenta con una alta compatibilidad con la mayoría de los hardware y procesadores del mercado.
- Django 2.2.2, es un framework basado en Python y utilizado en programación web, además de esto brinda servicios de backend al contar con librerías que lo convierten en un servicio API REST.
- Android Studio, es el entorno de desarrollo nativo del sistema operativo de smartphones Android, el cual facilita en gran parte la comunicación con el hardware.

- PostgreSQL, es un sistema de base de datos relacional, de código abierto y orientado a objetos, compatible con gran mayoría de lenguajes de programación.

Para fines de demostración, la solución de software del presente proyecto será probada en un prototipo desarrollado en raspberry, el cual realizará funciones simples como el encendido/apagado de luces leds al realizar operaciones con la solución de software.

2.2 Metodología de desarrollo de software

Las metodologías de desarrollo de software ágiles son orientadas a que el producto tenga las especificaciones que desea el cliente, por ende, el proyecto se ejecuta bajo el marco de trabajo para el desarrollo ágil de software Scrum. Se eligió Scrum ya que es ideal para trabajo en equipo y presentación de entregables funcionales en periodos cortos (1 a 4 semanas), cada iteración se presentará un entregable funcional que se denomina Sprint, los sprints constan de un conjunto de historias de usuario las cuales se le denomina Sprint Backlog, las historias de usuario son requerimientos que el cliente definió en el proceso de levantamiento de requerimientos forman parte del Product Backlog.

Las historias de usuario son escritas de tal forma que las entienda cualquier persona y que describan el requerimiento del cliente, estas poseen atributos como son: responsable de identificarla y desarrollarla (integrante del equipo de desarrollo de software), esfuerzo determinado en puntos scrum (1 punto scrum equivale a 8 horas laborables de un programador, los integrantes del equipo de desarrollo asignan valores de la serie de fibonacci), prioridad de ejecución (los valores de prioridad que van de decena en decena desde el 10 hasta el 50, siendo 10 de menor prioridad y 50 la de mayor prioridad), fecha que fue escrita, numero único

de identificación, titulo (debe ser corto de tal forma que facilite la identificación de la historia de usuario).

Con toda la información organizada se puede realizar el proceso de Sprint Planning Meeting, que consiste en determinar el número de Sprints que existirá tomando en cuenta la prioridad y esfuerzo de cada historia de usuario, el resultado de esta operación es el Sprint Backlog. Los roles que la metodología scrum define son los siguientes:

- Project Owner, escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.
- Master Scrum, vela por que el equipo cumpla con lo planificado.
- Development team Member, equipo de desarrollo de software.

En el Anexo 2.1 se enmarca la documentación correspondiente que demanda esta metodología, así como las responsabilidades de cada integrante del equipo.

2.3 Esquema general del producto

La solución de cuenta con cuatro componentes que interactúan entre sí para lograr el objetivo, la Figura 2.1 muestra esquemáticamente los componentes (Servicio en la nube, Administrador web, Aplicación móvil y Dispositivo de control de acceso). Aspectos muy importantes por considerar son la interacción que existirán entre sí y las herramientas que serán implementados.

Los medios de interacción entre los componentes serán el internet y la tecnología Bluetooth, como se observa en la Figura 2.1; el administrador web se comunica con el servicio en la nube por medio de internet, al igual lo hacen la aplicación móvil y el dispositivo de control de acceso. Con esto podemos darnos cuenta de que el servicio en la nube es nuestro punto central que se encargara del control total de los demás componentes.

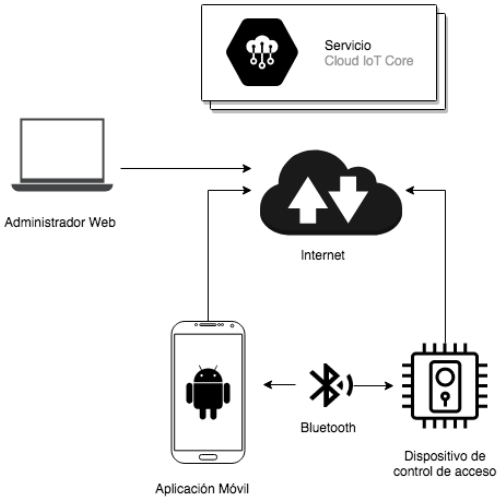


Figura 2.1 Esquema del proyecto

El componente que hemos llamado dispositivo de control de acceso es una abstracción de cualquier elemento de seguridad que conceda o niegue permisos ya sean cerraduras, controles de parking, cajas fuertes, por citar unos ejemplos; en la presente fase del proyecto el dispositivo de control de acceso se limitara a ser un prototipo desarrollado en raspberry el cual permita demostrar las funcionalidades del software desarrollado, que encienda luces leds de diferentes colores dependiendo de las acciones ordenadas por la interacción entre componentes.

2.4 Backend

El modelo de entidades del sistema dispone de cuatro tablas: Usuario, Dispositivo, Llave y Registro. En la entidad Usuario se registrarán los datos necesarios de los usuarios del sistema. La entidad Dispositivo servirá para inscribir y tener registro de a quien le pertenece dicho dispositivo, junto a un registro del estado del mismo, es decir, si se encuentra bloqueado o desbloqueado. En cuanto a la entidad Llave tiene como propósito mantener constancia de las llaves virtuales que los usuarios generen, las cuales estarán ligadas a un dispositivo en especial.

Y, por último, la entidad Registro servirá para llevar un registro de los eventos ocurridos entre usuarios y dispositivos. Las relaciones entre estas entidades están diagramadas en la Figura 2.2

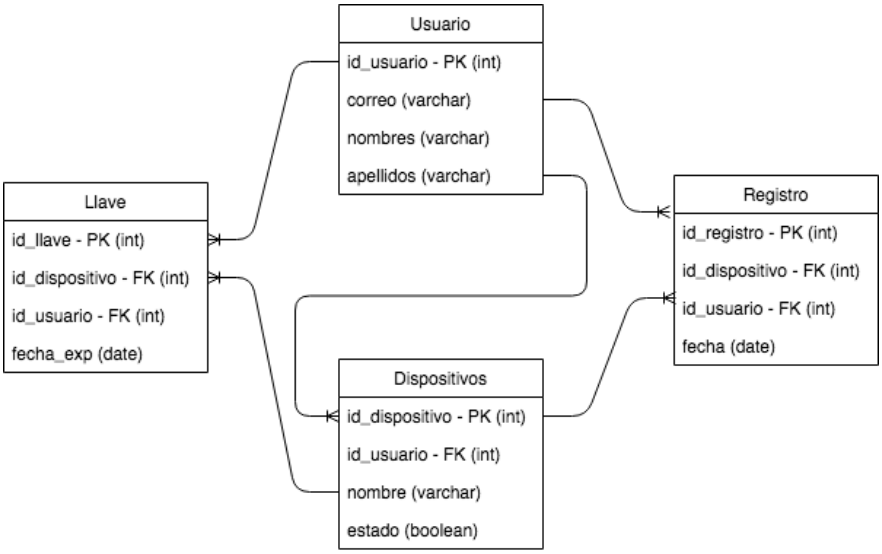


Figura 2.2 Modelo Entidad Relación

Entre las entidades Usuario y Dispositivo, se encuentra una relación de una a muchas, la que permitirá al usuario poder registrar y tener acceso a distintos dispositivos a la vez, como se observa en la Figura 2.3.

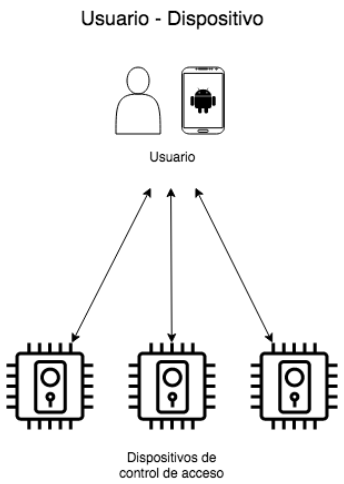


Figura 2.3 Relación Usuario - Dispositivo

Entre las entidades Usuario, Dispositivo y Llave, Usuario - Llave cuentan con una relación una a muchas así mismo como Dispositivo – Llave, la que permite la creación de una llave única por cada dispositivo de control de acceso y por cada usuario, como se observa en la Figura 2.4.

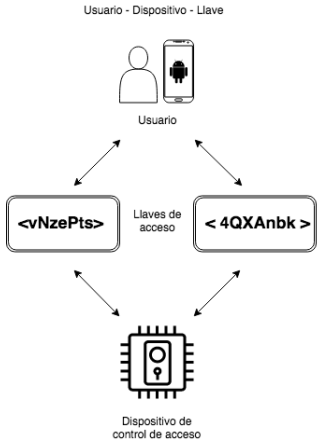


Figura 2.4 Relación Usuario – Dispositivo – Llave

Entre las entidades Usuario, Dispositivo y Registro, es necesaria una relación una a muchas entre Usuario – Registro, como también en Dispositivo – Registro, como se muestra en la Figura 2.5. Esto permitirá trazabilidad en los eventos ocurridos entre usuarios y dispositivos.

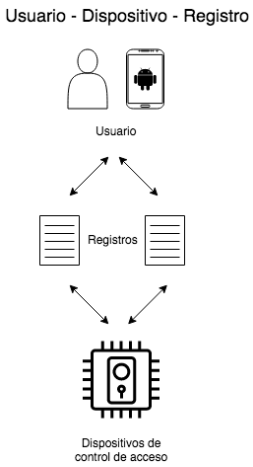


Figura 2.5 Relación Usuario – Dispositivo – Registro

En la tabla 2.1 se describe el API REST utilizado para las comunicaciones de los distintos componentes con el servidor.

Llamada	Descripción
createUsuario	Crea un usuario
readUsuario	Retorna los datos de un usuario
updateUsuario	Actualiza los datos de un usuario
deleteUsuario	Elimina el usuario
createDispositivo	Crea un dispositivo
readDispositivo	Retorna los datos de un dispositivo
updateDispositivo	Actualiza los datos de un dispositivo
deleteDispositivo	Elimina el dispositivo
createLlave	Crea una llave de acceso, con opción a ser: sin fecha de expiración, con fecha de expiración y cantidad de usos
readLlave	Retorna los datos de una llave de acceso
updateLlave	Actualiza los datos de una llave de acceso
deleteLlave	Elimina la llave de acceso
createReporte	Crea un reporte
readReporte	Retorna los datos de un reporte
login	Inicia sesión a un usuario y retorna un token de sesión
logout	Cierra la sesión de un usuario
autenticarLlaveAcceso	Autentica la llave de acceso

Tabla 2.1 Descripción del API REST

2.5 Aplicación Móvil

Con 10 años en el mercado, Android el sistema operativo de Google para dispositivos móviles está instalado en alrededor del 90% de los smartphones como se observa en la Figura 2.6 [10].

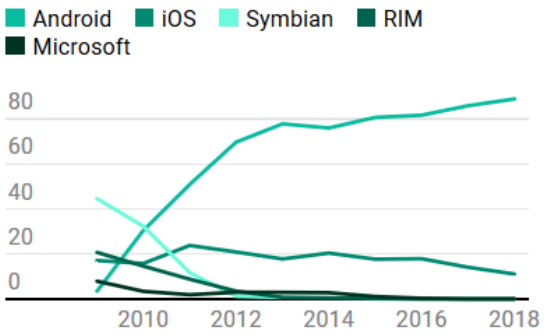


Figura 2.6 Evolución de la cuota de mercado de Android (2009-2018)

Android cuenta con una larga lista de versiones, a medida que pasa el tiempo realiza mejoras en su sistema operativo y los apoda con un nombre, en el Figura 2.7 se muestra las versiones y su porcentaje de uso a nivel mundial. [11]

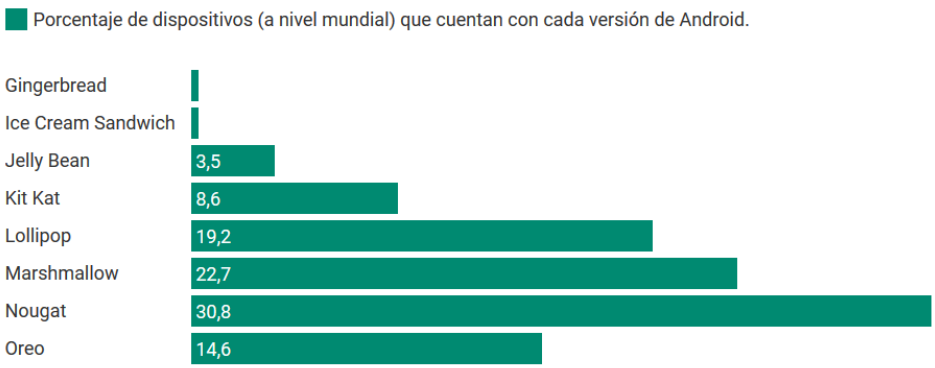


Figura 2.7 Distribución de Android en septiembre de 2018

A pesar de que cada versión mejora el sistema operativo, su compatibilidad de aplicaciones es la misma, gracias a que su Kernel es

basado en Linux y otros programas de código abierto. En la actualidad Nougat es el más instalado en dispositivos, pero esto cambia a medida que los fabricantes lanzan las actualizaciones para sus dispositivos con capas de personalización.

El aplicativo móvil del presente proyecto se desarrolla en Android Studio el cual es la herramienta oficial para el desarrollo de aplicaciones para Android y cuenta con varias de herramientas como: analizador de APK, emulador, Gradle plugin, herramientas SDK, IDE, editor de diseño visual, estilos rápidos, fiables y productivos. Lo más importante es el SDK, el cual es un conjunto de APIs facilitadas por Google para el desarrollo, integración y emulación de la aplicación.

2.6 Comunicaciones

En la actualidad las comunicaciones hacen posible la interacción entre dispositivos esto con el fin de beneficiarnos de cierta forma, por ejemplo, que un smartphone se pueda comunicar con una cerradura. La comunicación se compone de los siguientes elementos: emisor, receptor, mensaje, canal y código; los cuales son siempre validos en cualquier situación, desde la comunicación interpersonal, hasta la comunicación de dispositivos electrónicos. A continuación, se definirán los roles que ocuparán cada componente en el proceso de la comunicación y de igual forma la definición de las tecnologías a usar como canales y códigos.

- Emisor/Receptor: componentes de la solución, los cuales son: dispositivo de control de acceso, aplicación móvil, backend y administrador web.
- Mensaje: contenido que enviaremos por medio de un canal.
- Canal: medio por el un mensaje viaja desde el emisor hacia el receptor, en el presente proyecto se usará: Bluetooth e Internet.

- Código: JSON y contenido encriptado que se detallara más adelante en la definición de las seguridades.

La tecnología Bluetooth es una de las más usadas en redes inalámbricas de área personal (WPAN), es capaz de transmitir datos en la banda de 2.4 GHz. Frente a las otras tecnologías inalámbricas se destaca por su facilidad de comunicación con dispositivos móviles, uso de protocolos universales y permite sincronización. A la fecha la versión más actual es el Bluetooth v5.1, que presentas mejoras en dispositivos con GPS para una ubicación más exacta. El protocolo utilizado en la solución es Link Management Protocol (LMP), el cual establece un enlace de radio entre dos dispositivos mediante bluetooth y además que facilita la integración ya que esta implementado en los controladores bluetooth; esto nos permite abstraernos sin preocuparse por los drivers del hardware.

Otra tecnología de comunicación es la Internet, el cual es un conjunto de redes de comunicación interconectadas de forma descentralizada, utiliza protocolos TCP/IP que homogeniza a la red de forma mundial. Gracias a esto podremos comunicarnos con nuestro backend en la nube por medio de internet. En el desarrollo de la solución se presentan varios tipos de comunicación entre lo que hemos denominado sus componentes (dispositivo de control de acceso, aplicación móvil, backend y administrador web). Las relaciones de los componentes son las siguientes:

- Dispositivo de control de acceso – Aplicativo Móvil. Este proceso es el paso de una orden desde la aplicación móvil hacia el dispositivo de control de acceso por medio de Bluetooth, se enviará un código encriptado definido en el apartado de seguridades.
- Dispositivo de control de acceso – Backend. Estos dos componentes estarán en continua sincronización siempre y cuando sea posible ya

que son dependientes de la internet, esto quiere decir que el medio de comunicación será requerimientos a servidor por parte del dispositivo del control de acceso. De esta forma el servidor y el dispositivo de control de acceso siempre estarán actualizados y serán conocedores de los permisos otorgados.

- Backend – Aplicativo Móvil. Al igual que el dispositivo de control, este también establecerá comunicación con el backend por medio del internet, basada en requerimientos y llamadas https.
- Backend – Administrador Web. Como se ha mencionado con anterioridad el administrador web se encuentra dentro del backend, por ende, entre ellos se comunican directamente por medio de estructura de datos como por ejemplo el JSON.

2.7 Seguridad

Los componentes del sistema necesitan estar en constante comunicación para funcionar de manera adecuada, y dentro de un correcto funcionamiento se tiene que garantizar la seguridad de este, por lo que es fundamental que, las comunicaciones dentro del sistema sean seguras.

Las comunicaciones que el sistema realizara son a través de internet y bluetooth, como lo muestra la Figura 2.8. En el caso de internet, componentes como: el administrador web (usuario), smartphone (usuario), y el dispositivo de control de acceso consumirán llamadas de un API REST alojado en el servidor usando el protocolo https. Y en el caso de bluetooth componentes como: smartphone (usuario), y el dispositivo de control de acceso compartirán datos encriptados por medio de una conexión bluetooth.

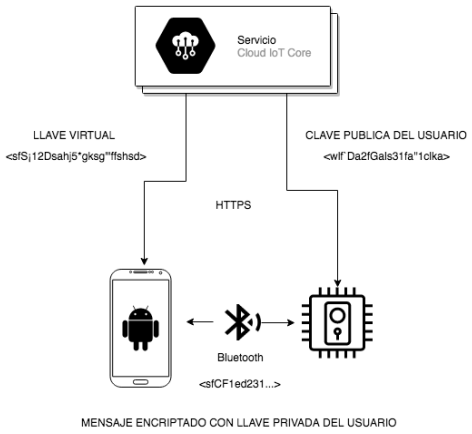


Figura 2.8 Esquema de Seguridades

Aparte de garantizar seguridad en las comunicaciones, se tiene que asegurar los datos sensibles guardados en el servidor. Para esto, los datos sensibles del servidor estarán encriptados usando el algoritmo iterativo PBKDF2, que a diferencia de algoritmos como Bcrypt o Scrypt, este requiere menos recursos computacionales (memoria y CPU), manteniendo robustez y resistencia a ataques de fuerza bruta y ataques de diccionario [9]. Cuando el smartphone se comunique con el dispositivo de control de acceso, su comunicación estará cifrada por medio de AES. AES es elegido por su habilidad de poder ser implementado con facilidad en varias plataformas, optimizado para consumir pocos recursos y producir encriptación y desencriptación instantánea y, por último, por su simplicidad, pero efectividad [12].

El smartphone tendrá una clave privada para encriptar el mensaje que enviara al dispositivo de control de acceso, y el dispositivo por otro lado, tendrá la clave pública del smartphone para desencriptar el mensaje.

Es necesario asegurar la solución contra de situaciones no ideales, donde su fuente de poder, o comunicaciones se vean comprometidas. El dispositivo de control de acceso es el componente más frágil e importante del sistema, el cual siempre debe estar encendido y conectado a internet.

No es posible asegurar que el dispositivo siempre tendrá una conexión eléctrica, puesto a que se debe considerar casos en donde esta se puede perder y es en ese momento donde el dispositivo deberá tener una fuente de poder de respaldo, como una batería.

En cuanto a la conexión a internet, esta conexión tiene probabilidades mucho más altas de perderse. En el caso de un dispositivo de control de acceso sin conexión a internet, este no podrá autenticar usuarios el cual es un grave problema. Es necesario que el dispositivo tenga un plan de respaldo para este escenario, por lo que el dispositivo contará con una llave física de emergencia, ideal para este caso.

CAPÍTULO 3

3. DESARROLLO DE LA SOLUCIÓN

En este capítulo se describirá el proceso de desarrollo, así como las funcionalidades y características a detalle de cada componente de nuestro sistema.

3.1 Administrador web

El administrador web es una instancia del servidor, desarrolladas en el mismo framework y lenguajes de programación, permite al usuario realizar todas las operaciones propuestas en el capítulo anterior, excepto la interacción bluetooth. En la tabla 3.1 se muestran las especificaciones y funcionalidades del administrador web del sistema.

ADMINISTRADOR WEB	
Especificaciones	
Lenguaje de programación	Python3.7 - Django2.2.3
Funcionalidad	
Servicios de autenticidad - Desbloqueo de dispositivo de manera remota - Gestión llaves - Gestión de dispositivos - Lectura de registros	

Tabla 3.1 Especificaciones del administrador web

3.2 Aplicación móvil

La aplicación móvil fue desarrollada en Android Studio, y con compatibilidad única con dispositivos Android. La tabla 3.2 muestra las especificaciones, tipos de conectividad y funcionalidad de la aplicación móvil del sistema.

APLICACIÓN MÓVIL	
Especificaciones	
Plataforma	Android
Conectividad	
Conectividad con servidor	Internet
Conectividad con dispositivo	Bluetooth
Funcionalidad	
Servicios de autenticidad - Desbloqueo de dispositivo de manera local usando bluetooth - Gestión llaves - Gestión de dispositivos - Lectura de registros	

Tabla 3.2 Especificaciones de la aplicación móvil

3.3 Servidor

El servidor web es un API REST, que maneja las sesiones con autenticación de tokens dinámicos, tiene endpoints para la comunicación entre el aplicativo móvil y el servidor además del prototipo de dispositivo de control de acceso y el servidor, están en constante comunicación. La tabla 3.3 detalla la funcionalidad de los registros de actividad en el Sistema. En donde se detalla como estos se comportan al momento de: Crearlos, lectura y edición.

REGISTROS DE ACTIVIDAD	
Creación	El dispositivo de control de acceso crea un registro (a través del API REST) por cada desbloqueo del mismo
Lectura	La lectura de los registros se realiza a través de la aplicación móvil y administrador web
Edición	No se puede alterar, ni eliminar los registros

Tabla 3.3 Pruebas de funcionalidad de los registros de actividad.

En la tabla 3.4 se especifican las funcionalidades acerca de las llaves de control de acceso del sistema, donde se especifican: La cantidad de llaves digitales que se pueden compartir, accesos ilimitados, fechas de validez, veces que una llave se puede usar y el revoco de llaves.

LLAVES VIRTUALES	
Cantidad de llaves virtuales para compartir	Ilimitada
Acceso Ilimitado	Se puede compartir acceso ilimitado a un dispositivo
Fechas	Se puede fijar un rango de fechas con horas y minutos para la validez de una llave
Cantidad de usos	Se puede crear una llave de un solo uso o de usos ilimitados
Revoco de llaves	Se puede revocar la llave para un dispositivo de un usuario

Tabla 3.4 Pruebas de funcionalidad de llaves digitales.

3.4 Prototipo

El prototipo esta implementado en un raspberry pi que corre un script en Python que interactúa directamente con el servidor en la nube y mediante bluetooth con la aplicación móvil en los dispositivos Android. Con finalidad de demostración posee leds que encienden dependiendo de las órdenes y validaciones correspondientes que debe realizar el sistema en el proceso que simula el control de acceso.

CAPÍTULO 4

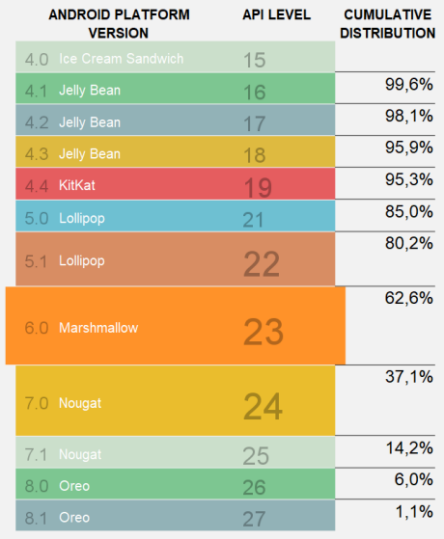
4. ANÁLISIS DE RESULTADOS

En el presente capitulo analizaremos los resultados del desarrollo del proyecto, por medio de comparaciones en el mercado y la aceptación del usuario final como consumidor referenciándonos en la compatibilidad. Concluiremos el capítulo recomendando futuros proyectos.

4.1 Análisis de soporte y compatibilidad

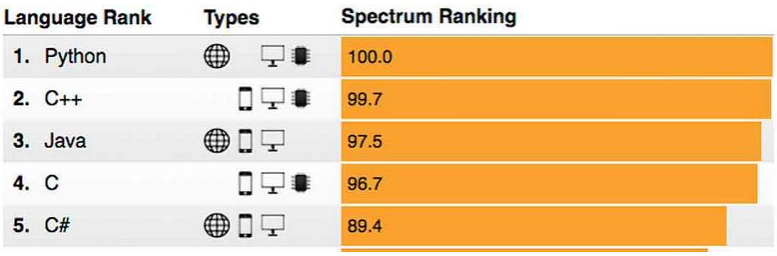
La compatibilidad es un aspecto muy importante a la hora de lanzar una solución de software al mercado, es por aquello que en base a las tecnologías usadas y lenguajes de programación mostramos el grado de compatibilidad de la solución. La aplicación móvil fue desarrollada en el API 23 de Android, el cual se ejecuta en el 62,6% de todos los smartphones con sistema operativo Android como está ilustrado en la Figura 3.1. Aunque la cifra de compatibilidad no es alta, se debe considerar la rápida evolución del sistema operativo Android en periodos de tiempo cortos, por lo cual en poco tiempo este porcentaje crecerá.

El servidor que fue desarrollado en Python mediante el framework orientado al desarrollo web Django, el cual integra un Administrador Web y un servicio API REST. Python es uno de los lenguajes top según la IEEE Spectrum, la cual posesiona a Python como el lenguaje en el puesto #1 para el desarrollo de soluciones web, escritorio y de sistemas embebidos. El script que incorpora el prototipo también fue desarrollado en Python, el cual establece comunicación con el servidor y el dispositivo de control de acceso.



ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,6%
4.2 Jelly Bean	17	98,1%
4.3 Jelly Bean	18	95,9%
4.4 KitKat	19	95,3%
5.0 Lollipop	21	85,0%
5.1 Lollipop	22	80,2%
6.0 Marshmallow	23	62,6%
7.0 Nougat	24	37,1%
7.1 Nougat	25	14,2%
8.0 Oreo	26	6,0%
8.1 Oreo	27	1,1%

Figura 4.1 Compatibilidad API Android



Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C++		99.7
3. Java		97.5
4. C		96.7
5. C#		89.4

Figura 4.1 Top lenguajes de programación 2018

4.2 Solución

La solución presentada ha tomado las fortalezas de la competencia y ha tratado de fusionarlas de la manera más sencilla, con la finalidad de entregar un producto robusto y completo que no abrume con muchas opciones al usuario, y sea fácil de usar. En la Tabla 4.1 se presenta un

resumen final de las funcionalidades integradas en el sistema en comparación con los productos de alta gama del mercado.

FUNCIONALIDAD															
	Aplicación móvil	Administrador Web	NFC	Bluetooth	Compartir llaves	Acceso indefinido	Llaves con fecha	Llaves de un solo uso	Revocar llaves	Acceso remoto	Logs	Funcionalidad de timbre	Intercomunicador	Solicitar acceso	Platfr.	
															IOS	Android
Samsung SHP-DR708	x		x	x	x		x	x		x	x	x				x
Digital Keys LDK400	x	x	x		x		x			x	x			x	x	x
Klevio	x				x		x		x	x		x	x		x	x
Kaadas K9	x			x											x	x
Kevo Contemporary	x			x	x	x			x		x				x	x
Solución Propuesta	x	x		x	x	x	x	x	x	x	x					x

Tabla 4.1 Comparación de funcionalidad frente a la competencia.

4.3 Resultados

La tabla 4.2 muestra las pruebas de latencia realizadas en el servidor (Instancia de Digital Ocean) donde se realizó 1000 requests por minuto. El propósito de esta prueba es de comprobar el tiempo de ejecución del API REST.

LATENCIA DEL SERVIDOR	
CREATE	344 ms
READ	333 ms
UPDATE	354 ms
DELETE	307 ms

Tabla 4.2 Pruebas de latencia

La tabla 4.3 muestra las pruebas realizadas sobre la conectividad de bluetooth de la aplicación móvil con el dispositivo de control de acceso implementado en un Raspberry pi 3 y probado en un ambiente de tipo puertas adentro (departamento).

BLUETOOTH	
Conexión	
Conexión automática	Se conecta automáticamente con dispositivos previamente enlazados
Latencia	6.63 ms
Alcance (Puertas Adentro)	
< 1m	RSSI = 0
1m - 5m	RSSI = - 4
5m - 10m	RSSI = - 12
10m - 20m	RSSI = - 21

Tabla 4.3 Pruebas de funcionalidad de bluetooth.

4.4 Futuros Trabajos

En la primera fase de este proyecto se obtiene el software que controlara los distintos tipos de dispositivos de control de acceso, como pueden implementados en casilleros, torniquetes, puertas, parqueos, entre otros. Por lo tanto, se proyecta:

- Diseñar e implementar el hardware del prototipo en una placa de circuitos impresos, con el fin de ahorrar mayores recursos e incrementar el tipo de aplicaciones.
- Desarrollar una aplicación móvil para el sistema operativo IOS, ya que es el segundo sistema operativo móvil más usado en smartphones, y brindar completa accesibilidad a todos los usuarios. [13]
- Diseñar e implementar sistemas tolerantes a fallos que no hagan vulnerable el sistema de llaves virtuales, tanto su software como su

hardware. Como también reforzar la seguridad mediante la exploración de las vulnerabilidades.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- El modelo de interacción de los componentes del sistema propuesto previamente interactúa de la forma esperada, su compatibilidad es total, permitiendo que la comunicación entre los mismos sea eficaz. El modelo es escalable, es decir, si en futuros trabajos se desea alterar la estructura con la agregación de más módulos, se facilitará su integración.
- La plataforma web que administra los principales actores del sistema, permite al usuario o administrador gestionar los distintos componentes, como sus dispositivos, los accesos de dichos dispositivos, manipulación remota del dispositivo (desbloquearlo), ver reportes de actividades, por destacar los más importantes, al igual que la aplicación móvil que a través de una interfaz amigable con el usuario realiza las mismas funciones que el administrador de forma más sencilla gracias a la implementación de un servicio de API REST en nuestro servidor.
- El producto desarrollado es un producto de software en una versión beta, pensado en continuar desarrollándose sobre sus bases en futuros proyectos, ya que la solución desarrollada en su totalidad propone un uso más eficiente y mejorado de las llaves físicas como las conocemos hoy en día, haciéndolas más seguras, accesibles y amigables con los usuarios finales.

Recomendaciones

- Se recomienda que la aplicación móvil se migre utilizando un framework hibrido basado en JavaScript o C# ya que esto permitiría que la aplicación sea multiplataforma y corra en los distintos sistemas operativos que tenga soporte el framework.
- Se recomienda separar los módulos del backend para crear un sistema distribuido para mejorar el throughput, ya que a gran escala se podrán

requerir más instancias del software en producción, de esta forma mejoraríamos la disponibilidad como base de la seguridad de la información.

- Integrar ciertas funcionalidades que presenta la competencia tales como: intercomunicador, timbre, acceso biométrico y doble autenticación para permisos de controles más restringidos, como cerraduras de cajas fuertes. Con la finalidad de darle mayor valor al producto en el mercado.

BIBLIOGRAFÍA

[1] INEC Ecuador. Ecuador en cifras. Recuperado el, 25, 2010.

[2] NH Ismail, Zarina Tukiran, NN Shamsuddin, and EIS Saadon. Android-based home door locks application via bluetooth for disabled people. In 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014) , pages 227–231. IEEE, 2014.

[3] Jeong-ile Jeong. A study on the iot based smart door lock system. In Information Science and Applications (ICISA) 2016, pages 1307–1318. Springer, 2016.

[4] Lia Kamelia, S.R. Alfin Noorhassan, W. S. Mada Sanjaya, and W.S. Edi Mulyana. Door-automation system using bluetooth-based android for mobile phone. 9:1759–1762, 01 2014.

[5] Deviant Ollam. Keys to the kingdom: impressioning, privilege escalation, bumping, and other key-based attacks against physical locks. Elsevier, 2012.

[6] Sharon Panth, Mahesh Jivani, et al. Home automation system (has) using android for mobile phone. International Journal of Electronics and Computer Science Engineering (IJECSE), 3(1):1–11, 2013.

[7] Yong Tae Park, Pranesh Sthapit, and Jae Young Pyun. Smart digital door lock for the home automation. In TENCON 2009-2009 IEEE Region 10 Conference, pages 1–6. IEEE, 2009.

[8] Ming Yan and Hao Shi. Smart living using bluetooth-based android smartphone. International journal of wireless & mobile networks, 5(1):65, 2013.

[9] Ertaul, L., Kaur, M., & Gudise, V. A. K. R. (2016). Implementation and Performance Analysis of PBKDF2, Bcrypt, Scrypt Algorithms. In *Proceedings of the International Conference on Wireless Networks (ICWN)* (p. 66). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[10] Yoo, S., Ryu, H. R., Yeon, H., Kwon, T., & Jang, Y. (2019). Visual Analytics and Visualization for Android Security Risk. Journal of Computer Languages.

[11] Shah, N., & Modi, N. (2019). Enhancing Security of Android-Based Smart Devices: Preventive Approach. In *Information and Communication Technology for Intelligent Systems* (pp. 589-597). Springer, Singapore.

[12] Daemen, J., & Rijmen, V. (2013). The design of Rijndael: AES-the advanced encryption standard. Springer Science & Business Media.

[13] StatCounter, «StatCounter,» Julio 2019. [En línea]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide.

[14] Tendrl, «https://github.com/Tendrl/documentation/wiki/Best-Practices-for-Response-Times-and-Latency,» 7 Noviembre 2017. [En línea].

ANEXOS

HISTORIAS DE USUARIO

No. 1	**Prioridad:** 10
Titulo:	Crear cuenta
Descripción:	**Como** usuario **Quiero** registrarme en la aplicación **Para** poder usar el servicio
Responsable:	
Esfuerzo: 5	**Fecha:** 10/06/2019

No. 2	**Prioridad:** 50
Titulo:	Iniciar sesión
Descripción:	**Como** usuario y administrador **Quiero** iniciar sesión **Para** para controlar mis llaves virtuales
Responsable:	
Esfuerzo: 5	**Fecha:** 10/06/2019

No. 3	**Prioridad:** 40
Titulo:	Compartir llaves
Descripción:	**Como** usuario **Quiero** compartir llaves virtuales **Para** que puedan usarlas terceros
Responsable:	
Esfuerzo: 8	**Fecha:** 10/06/2019

No. 4	**Prioridad:** 20
Titulo:	Temporizar llaves
Descripción:	**Como** usuario **Quiero** que mis llaves virtuales sirvan por un tiempo especificado **Para** dejar de compartir a terceros no autorizados fuera de tiempo
Responsable:	
Esfuerzo: 8	**Fecha:** 10/06/2019

No. 5	**Prioridad:** 10
Titulo:	Registrar dispositivos
Descripción:	**Como** usuario **Quiero** registrar dispositivos de control de acceso **Para** poder utilizarlos con la aplicación
Responsable:	
Esfuerzo: 8	**Fecha:** 10/06/2019

No. 6	**Prioridad:** 30
Titulo:	Abrir dispositivo de control de acceso
Descripción:	**Como** usuario **Quiero** abrir mi dispositivo de control de acceso por medio de una aplicación móvil **Para** facilitar el acceso
Responsable:	
Esfuerzo: 5	**Fecha:** 12/06/2019

No. 7	**Prioridad:** 20
Titulo:	Registros
Descripción:	**Como** usuario

	Quiero ver mis registros de quien abrió el dispositivo de control de acceso **Para** tener control del acceso	
Responsable:		
Esfuerzo: 8		**Fecha:** 12/06/2019

No. 8		**Prioridad:** 40
Titulo:	Revocar Permisos	
Descripción:	**Como** usuario **Quiero** revocar permisos de mis llaves **Para** tener total control de mis dispositivos	
Responsable:		
Esfuerzo: 8		**Fecha:** 12/06/2019

No. 9		**Prioridad:** 50
Titulo:	Ingresar dispositivos	
Descripción:	**Como** administrador **Quiero** registrar el número de serie de mis dispositivos que fabrico **Para** que mediante el código de serie puedan vincularse con el servidor	
Responsable:		
Esfuerzo: 8		**Fecha:** 12/06/2019

No. 10		**Prioridad:** 30
Titulo:	Abrir por http	
Descripción:	**Como** usuario	

	Quiero abrir mi dispositivo de control de acceso desde el servidor **Para** abrir cuando no tenga dispositivo celular	
Responsable:		
Esfuerzo: 8		**Fecha:** 12/06/2019