UNIVERSIDAD PERUANA UNIÓN

FACULTAD DE INGENERIA Y ARQUITECTURA

Escuela Profesional de Ingeniería de Sistemas

Implementación de un sistema automatizado para gestionar la seguridad de accesos en viviendas juliaqueñas mediante aplicativo móvil e internet de las cosas

Tesis para obtener el Título Profesional de ingeniero de Sistemas

Por:
Victor Alexis CCamerccoa Mamani

Asesor:
Jorge Eddy Otazu Luque

Juliaca, agosto de 2019

DECLARACIÓN JURADA DE AUTORÍA DEL INFORME DE TESIS

Jorge Eddy Otazu Luque, de la Facultad de Ingeniería y arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: "**IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA GESTIONAR LA SEGURIDAD DE ACCESOS EN VIVIENDAS JULIAQUEÑAS MEDIANTE APLICATIVO MÓVIL E INTERNET DE LAS COSAS**" constituye la memoria que presenta el Bachiller Victor Alexis Ccamerccoa Mamani para obtener el título de Profesional de Ingeniero de Sistemas, cuya tesis ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en Juliaca, a los 17 días del mes de Julio del año 2021

Ing. Jorge Eddy
Otazu Luque
Asesor

LIMA - PERU

093

ACTA DE SUSTENTACIÓN DE TESIS

En Puno, Juliaca, Villa Chullunquiani, a 24 día(s) del mes de agosto del año 2020 siendo las 09:00 horas, se reunieron en el Salón de Grados y Títulos de la Universidad Peruana Unión, Filial Juliaca, bajo la dirección del Señor Presidente del jurado: Mtro. Fermin Henry Centurión Julca, el secretario: Ing. David Mamani Pari y los demás miembros: Ing. Angel Rosendo Condori Ccaquira y el asesor Ing. Jorge Eddy Olazo Luque con el propósito de administrar el acto académico de sustentación de la tesis titulada: "Implementación de un sistema automatizado para gestionar la seguridad de accesos en viviendas juliaqueñas mediante aplicativo móvil e internet de las cosas" de el(los)/la(las) bachiller/es: a) Victor Alexis Ccamercoa Mamani b) conducente a la obtención del título profesional de Ingeniero de Sistemas
(Nombre del Título Profesional)

con mención en

El Presidente inició el acto académico de sustentación invitando al (los)/a(la)(las) candidato(a)/s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por el(los)/la(las) candidato(a)/s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Candidato (a): Victor Alexis Ccamercoa Mamani

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
Aprobado	16	B	Bueno	Muy bueno

Candidato (b):

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó al(los)/a(la)(las) candidato(a)/s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

Presidente

Secretario

Asesor

Miembro

Miembro

Candidato/a (a)

Candidato/a (b)

AREA TEMÁTICA

Título:

Implementación de un sistema automatizado para gestionar la seguridad de accesos en viviendas juliaqueñas mediante aplicativo móvil e internet de las cosas

- **Línea de Investigación:** Procesos de TI.
- **Área de Conocimiento:** Ingeniería y Tecnología.
- **Sub Área:** Ingeniería Informática.
- **Disciplina:** Robótica y control automático.

Autor del Proyecto: CCAMERCCOA MAMANI, Victor Alexis.

Nombre del Asesor: Ing. OTAZU LUQUE, Jorge Eddy.

DEDICATORIA

Le dedico esto a mi padre espiritual Dios, Tu ayuda ha sido fundamental, has estado conmigo incluso en los momentos más turbulentos. Este trabajo no fue fácil, pero estuviste motivándome y ayudándome hasta donde tus alcances lo permitían.

A mis padres Leonardo y Obdulia por haberme forjado como la persona que soy en la actualidad; muchos de mis logros se los debo a ustedes entre los que se incluye este. Me formaron con reglas y con algunas libertades, pero al final de cuantas, me motivaron constantemente para alcanzar mis anhelos.

Esto es para ustedes.

Finalmente, a mis hermanos por "deshacer sus manos para que YO me hiciera mejor que ayer" y a mis compañeros de estudio quienes compartieron los buenos y malos momentos donde me enseñaron lo que significa una verdadera amistad.

Victor Alexis CCamerccoa Mamani

AGRADECIMIENTO

Mi total Agradecimiento a Dios, por ser la clave que convierte los problemas en Bendiciones y lo inesperado en los Mejores Regalos

A mi asesor el Ing. Jorge Otazu, por su Comprensión y su esfuerzo por cumplir la meta.

A la casa de estudios "Universidad Peruana Unión" especialmente a los docentes de la Escuela Académico profesional de Ingeniería de Sistemas quienes me transmitieron sus conocimientos a lo largo de mi formación profesional.

Victor Alexis CCamerccoa Mamani

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ANEXOS

SIMBOLOS USADOS

- **DB**: Base de datos
- **IOT**: Internet de las cosas
- **INEI**: Instituto Nacional de Estadística
- **APP**: Aplicación
- **V**: Voltios
- **PNP**: policía Nacional del Perú
- **BPS**: bits por segundo
- **SRAM**: estática de acceso aleatorio
- **Kb**: Kilobit
- **GB**: Gygabyte
- **RFID**: identificador de radiofrecuencia
- **WSN**: redes de sensores inalámbricos
- **TFG**: trabajo de fin de grado
- **CNIL**: Comisión nacional de informática y de las libertades
- **AEPD**: Agencia española de proteccion de datos
- **GPS**: Sistema de posicionamiento global
- **NFC**: comunicación de campo cercano
- **UART**: Transmision-receptor asíncrono universal
- **TTL**: Time to live (tiempo de vida)
- **TDD:** Test-Driven-Development (desarrollo basado en pruebas)
- **SC:** escenario de desarrollo
- **RF:** requerimiento funcional
- **RNF:** requerimiento no funcional
- **NH:** número de historia
- **SC:** storycards
- **TC:** Taskcards
- **SGBD:** sistema de gestor de base de datos
- **TFG:** trabajo de fin de grado.

RESUMEN

Juliaca provincia de San Román está situada en el departamento de Puno, reconocida como la perla del capitalismo andino en el Perú, es una zona comercial muy activa la cual al pasar de los años ha incrementado poco a poco la ola delincuencial en los hogares como en sus distintos sectores como informan los medios de comunicación y los periódicos, lo cual se ha convertido en un dolor de cabeza para las autoridades de Juliaca, visto la situación se implementara un modelo de control de acceso para las viviendas, la cual contará de un sistema embebido como de un aplicativo móvil, que permita tener el control de accesos a personas que hagan uso del aplicativo móvil en sus viviendas y poder almacenar en una db "base de datos", la lista de usuarios. El sistema de seguridad contará con un Arduino que es una placa de hardware libre, que será el cerebro del sistema embebido de control de acceso, también tenemos como comunicador entre el hardware y el aplicativo una red bluetooth, por medio de una verificación de contraseña luego seguido el lector de huellas dentro del aplicativo podrá conectarse al hardware e interactuar para luego poder abrir y cerrar la cerradura eléctrica, como modo de incrementar la seguridad se hará de uso la huella dactilar, la cual registro datos biométricos del usuario que permitió acceder a la siguiente actividad, permitiendo de esta manera ingresar a la vista principal de la aplicación solo con personas que registraron sus huellas en el dispositivo móvil, negando a las personas que no estaban registradas. La tarea de conversion de corriente la realizó el módulo relay y un transformador de corriente, esto evito dañar el sistema embebido, la función fue de reducir la corriente de 220v a 12v, esto implico un ahorro de energía en los hogares.

La metodología Mobile-D, se usó debido a su desarrollo especifico en aplicaciones y facilito el despliegue en lo planteado para el desarrollo de la aplicación, se inició con la identificación de los stakeholders o interesados en el proyecto, también se definió el alcance de la aplicación y se delimito el desarrollo de la aplicación para crear una segunda versión mejorando y añadiendo más funcionalidades con nuevas herramientas, luego se comenzó con la configuración del proyecto identificando las herramientas en sus versiones más actuales para tener actualizada la aplicación, para luego ver los requerimientos funcionales y no funcionales donde se vio las características que

debía tener la aplicación y se pasó a la parte de proceso donde se desarrolló los prototipos y también los storyboards y storycards. En la fase de estabilidad se vio la codificación e interfase como también resumen de la documentación para pasar a la fase de pruebas y concluir con el desarrollo de la aplicación de esta manera ayudo la metodologia en el proyecto en su desarrollo y para visualizar la prueba se mostró la funcionalidad en videos elaborados una vez concluida la aplicación.

Palabras clave: Arduino, biométrico, relay, v (voltios), db (base de datos), apk (aplicación), IoT (Internet de las cosas).

ABSTRACT

Juliaca province of San Roman is located in the department of Puno, recognized as the pearl of Andean capitalism in Peru, is a very active commercial area which over the years has gradually increased the wave of crime in homes and in their various sectors as reported by the media and newspapers, which has become a headache for the authorities of Juliaca, given the situation will implement an access control model for homes, which will have an embedded system as a mobile application, which allows access control to people who use the mobile application in their homes and can store in a db "database", the list of users. The security system will have an Arduino that is a free hardware board, which will be the brain of the embedded access control system, we also have as a communicator between the hardware and the application a bluetooth network, through a password verification then followed by the fingerprint reader inside the application can connect to the hardware and interact to then open and close the electric lock, As a way to increase security, the fingerprint will be used, which will register biometric data of the user that allowed access to the next activity, allowing this way to enter the main view of the application only with people who registered their fingerprints on the mobile device, denying people who were not registered. The current conversion task was performed by the relay module and a current transformer, this avoided damaging the embedded system, the function was to reduce the current from 220v to 12v, this implied an energy saving in the homes.

The Mobile-D methodology was used due to its specific development in applications and facilitated the deployment in the proposed for the development of the application, began with the identification of stakeholders or interested in the project, also defined the scope of the application and delimited the development of the application to create a second version improving and adding more features with new tools, Then we started with the configuration of the project, identifying the tools in their most current versions in order to have the application updated. Then we saw the functional and non-functional requirements where we saw the characteristics that the application should have and we went on to the process part where we developed the prototypes and also the storyboards and storycards. In the stability phase, we saw the coding and interface as well as a

summary of the documentation to move on to the testing phase and conclude with the development of the application. In this way, we helped the methodology in the project in its development and to visualize the test, we showed the functionality in videos elaborated once the application was concluded.

Keywords: Arduino, biometrics, relay, v (volts), db (database), apk (application), IoT (Internet of things)

CAPITULO I. El Problema

1.1. Descripción de la situación Problemática:

El incremento de la inseguridad en la que vive Juliaca se muestra en distintas formas, los robos a viviendas es una de tantas que existe, la cual con el pasar del tiempo se ha convertido en algo cotidiano y esto podemos apreciar en la Figura 1, razón por la cual se está empezando a tomar medidas más drásticas una de ellas es la tecnología pero a precios muy elevados a lo que los usuarios no puedan acceder esto referencia Mestanza (2013), en su investigación, ¿Qué sucedería si extraviara la llave o por descuido terminaría en manos ajenas al dueño de la casa? Esto ocasiona un riesgo al patrimonio material y seguridad del dueño y su familia menciona Hernández, Martinez & Méndez (2015).

Según reportes de la PNP identificaron los métodos que usan para ingresar a las viviendas denominándose como "técnicas de apertura" las cuales se manifiestan en su mayoría en cerraduras con llave y traba, uno de los más conocidos según expedientes policiales es el "Bumping" el cual consta de usar una llave distinta a la original y con un suave golpe hacer saltar los pistones de la cerradura, este no es el único método también existe el método del "Resbalón" método clásico usado por los ladrones dedicados al robo de viviendas que solo consta de introducir una tarjeta por el filo de la puerta con el marco la cual solo funciona cuando el dueño no asegura con la traba la cerradura de la puerta, otro método usado para delinquir en las viviendas es el "Ganzuado" donde se manipula manualmente el sistema de la cerradura con alambres fuertes y planos, que termina en un gancho el cual sirve para empujar los pistones de la cerradura para luego abrir e ingresar al domicilio.

El "Magic Key" otro de los métodos usados por los delincuentes el cual se trata de un artilugio que se puede adquirir por medio de internet, similar a la ganzúa y está formado por varias puntas que simulan a la llave original dependiendo el modelo de la cerradura y se pueden adquirir por 300 euros en una tienda virtual china. Por último tenemos el método "Impressioning" que consta de

introducir una llave virgen a la cerradura y darle varios giros para que se marque el patrón de la llave y poder duplicar la llave original como menciona Lozano (2018) en su reporte, de esta manera es cómo actúan los llamados robacasas recurriendo a herramientas ingeniosas para delinquir esto es lo que pone en riesgo los hogares de Juliaca como en otros departamentos del Perú, esto se puede apreciar en la Figura 2. Como también podemos observar en la Figura 3 los actos delictivos en viviendas con vigilancia y sin vigilancia.

Figura 1. Percepción de la inseguridad por tipo de delito en la ciudad de Lima-Perú.
Fuente: Culquichicon (2012), mejorado por mi persona.

Figura 2. Estadística de índice de asaltos a viviendas en la ciudad de Juliaca
Fuente: INEI (2017), mejorado por mi persona

Figura 3. Tasa de robos en viviendas vigiladas vs. viviendas sin vigilancia de personal de vigilancia en Lima-Perú.

Fuente: Culquichicon (2012), mejorado por mi persona.

1.2. Objetivos:

1.2.1. Objetivo General

Implementar un sistema automatizado para gestionar la seguridad de accesos en viviendas juliaqueñas mediante aplicativo móvil e internet de las cosas

1.2.2. Objetivo específico

- Desarrollar el modelo de la arquitectura para gestionar accesos en mejora de la seguridad
- Construir y desarrollar la plataforma móvil y la conexión del hardware con el software
- Realizar pruebas al aplicativo y al hardware para la gestión de accesos en la seguridad

1.3. Justificación

El proyecto surge de la poca seguridad en la gestión de accesos en viviendas, porque el utilizar una llave trae el riesgo de pérdida o hurto, esto conlleva a no tener una gestión o control de las

personas que habitan en nuestros hogares, esto trae por consecuencia tener indirectamente una disminución de la seguridad en nuestras viviendas, entonces para mejorar la seguridad en las viviendas de Juliaca se recurrió a los sistemas de acceso inalámbricos existentes en el mercado, dentro de estos sistemas, incluye la reducción del uso de cables como también reduce el índice de consumo de energía eléctrica, generando de este modo un ahorro y al mismo tiempo que incrementamos la seguridad en los hogares dando accesos a personas pertenecientes al círculo familiar menciona Mestanza (2013). Con esto obtendremos contribuir en la seguridad y la tranquilidad de los residentes y se es de necesidad el proyecto para la protección de bienes materiales adquiridos con mucho esfuerzo Hernández, Martinez & Méndez (2015).

LhamforAccess pretende dejar el concepto de las cerraduras tradicionales por cerraduras electrónicas donde la llave es un Smartphone los cuales facilitaran el ingreso a su hogar, creando de este modo una gestión de accesos para mejorar la seguridad y como respaldo tendremos un acceso biométrico el cual será de respaldo en caso que se deje abierta la aplicación en el dispositivo móvil generando un acceso solo a personas registradas en la aplicación. Las herramientas de comunicación tecnología e información se han convertido muy fundamentales para el intercambio de datos cosa que contribuyó al nacimiento de internet de las cosas herramienta que es de suma necesidad para incrementar la seguridad y guardar datos medio que usaremos para el desarrollo del sistema de gestión de accesos al cual solo podrán ingresar personas registradas en dicho sistema esto conlleva a automatizar nuestros hogares con la finalidad de mejorar la seguridad como señala Cuzme (2015) a la vez que tendremos menos consumo de energía ganaremos un incremento en la seguridad, usando tecnología bluetooth consiguiendo de esta forma una red local rápida que envié datos simples a la cerradura eléctrica creando acceso único por usuario y de esta manera incrementaremos la seguridad en cada vivienda de Juliaca.

1.4. Presuposición filosófica

"Pero el Señor es fiel, y él los fortalecerá y los protegerá del maligno" (2Tesalonisenses3.3)

Este versículo lo aplico en mi vida cotidiana porqué ser fiel al señor me ha mantenido estos 5 años con fuerza y dedicación a mis estudios. Pase por muchos momentos difíciles, pero ahí él

estaba para mi apoyándome día y noche, también aplicare esto en lo que desarrollare para mejorar la seguridad y poder salvaguardar a las personas de gente ajenas. Con mi proyecto mejorar su seguridad en sus hogares y porque no en sus vidas como el señor estuvo haciendo todos estos 5 años de carrera conmigo.

"Póngase toda la armadura de Dios para que Puedan hacerse frente a las artimañas del diablo" (Efesios 6.11)

CAPITULO II. Bases Teóricas de la Investigación

2.1. Revisión de la Literatura

En la investigación realizada por Cuzme (2015), “en su investigación titulada El Internet De Las Cosas Y Las Consideraciones De Seguridad Establecer tubo el objetivo de considerar los mecanismos de seguridad, con la innovación tecnológica del internet de las cosas (iot), se puso como referencia para el desarrollo de la proyecto ver las medidas que debemos tomar en la seguridad dentro de internet de las cosas. Hizo uso de metodología tradicional en cascada para el análisis, diseño, pruebas e implementación, obteniendo como resultado el análisis basado en la comparación de los datos y resultados que se dieron en las entrevistas”.

También menciona acerca del tema el investigador Culquichicon (2012),” ya que tomo como objetivo el diseñar e implementar un sistema que permita la interacción entre dispositivos electrónicos instalados en una vivienda para monitorear y controlar diferentes parámetros de ésta en su investigación titulada domolab: sistema de monitoreo y control remoto de viviendas, también hizo uso de la metodología tradicional en cascada: análisis, diseño, pruebas e implementación para concluir con lo siguiente. Después de analizar el contexto de inseguridad ciudadana en lima metropolitana, se puede concluir que existe la necesidad de mejorar la seguridad de las viviendas en la ciudad. por ello, se puede aprovechar la tecnología disponible para crear un producto que permita satisfacer las necesidades de protección y seguridad de los usuarios”.

Teniendo en cuenta la seguridad también notamos el análisis y diseño de una red domótica según el investigador Calvo (2014), ”observamos que el objetivo tazado fue proponer una solución tecnológica a través de la domótica que permita realizar una mejor gestión de la energía utilizada en la vivienda social, definida como vivienda tipo en este trabajo, favoreciendo el ahorro energético. Hizo uso de la metodología tradicional en

cascada: análisis, diseño, pruebas e implementación para llegar a la solución planteada en este trabajo en cómo se utiliza la electricidad para iluminar, alimentar los aparatos en stand y pueden implicar un ahorro adicional al ya generado por el simple reemplazo de la tecnología de las ampolletas. El reemplazar las ampolletas incandescentes por ampolletas led, contribuyen a reducir el consumo energético hasta en un 90% dependiendo de la calidad de la ampolleta en comparación a las lámparas fluorescentes compactas que son un 80% más eficientes que las ampolletas de incandescencia según informa general electric. Si a este ahorro generado por el recambio de las lámparas incandescentes, halógenas, fluorescentes, o cualquier otra por la tecnología de iluminación led, le sumamos el potencial ahorro que puede generar nuestro sistema de control de iluminación y el control de enchufes haríamos aún más eficiente el uso de la energía. Todo esto es mencionado en su investigación titulada Análisis Y diseño de Una red domótica para viviendas sociales".

También otro punto a tomar es ver las bases de datos, para ver este punto tenemos a la investigadora Solano (2014), "en su investigación análisis de los sistemas de gestión de bases de datos actuales como soporte para las tecnologías de internet de las cosas. Nos muestra en su objetivo de este trabajo de tesis es hacer un análisis de las tecnologías de bases de datos actuales como soporte los sistemas de internet de las cosas. Para esto ella utilizo la metodología tradicional en cascada: análisis, diseño, pruebas e implementación. Concluyendo que todas estas limitaciones conducen a plantearse que la mejor opción Sería crear un nuevo tipo de SGBD especializados para dar soporte a IoT, que sean capaces de dar respuestas a las características de un ambiente dinámico y heterogéneo".

Y por último se vio la privacidad esto podemos apreciar en la investigación Internet de las cosas privacidad y seguridad, realizado por el investigador Castro (2016). " El objetivo de este trabajo es analizar en profundidad todos los cambios que va a suponer la implantación de esta nueva forma de entender la tecnología en general, y con ella la sociedad y la manera de hacer las cosas tanto cotidianas como laborales, para esto utilizo o uso de metodología tradicional en cascada: análisis, Diseño, pruebas e implementación,

para obtener como resultado este TFG se ha ido centrando en la importancia de que a la par de esta evolución, debe tenerse en cuenta una evolución en la seguridad, tanto a nivel de producción como a nivel de usuario. La seguridad y privacidad de nuestros datos e información es uno de los aspectos que más preocupan a la población, así como nuestra integridad física".

2.2. Marco teórico

2.2.1. Fundamentos teóricos

El siguiente capítulo veremos todo respecto a lo requerido para la compresión en el desarrollo e implementación del proyecto, de igual modo se pasará a detallar lo que abarca el tema de internet de las cosas (Iot), se detallara la sección de tecnologías que conforman el diseño de la solución. El modelo a integrarse requerirá las herramientas disponibles en el mercado, y los softwares que son los más adecuados e identificar la métrica de seguridad que existan los manejos de datos que se deben hacer para el funcionamiento correcto del proyecto.

2.3. Marco Teórico

2.3.1. Hogar digital

Según el tesista Culquichicon (2012), muestra un estudio de seguridad en los hogares que están conectados a las tic. Define que el hogar digital es una infraestructura en donde los servicios de entretenimiento, comunicación y gestión de distintos elementos (artefactos, luminarias, chapas eléctricas, puertas), esto es realizado por medio de un conjunto de instalaciones de componentes como hardware y la otra parte por software el principal objetivo de un hogar digital es mejorar la calidad de vida de los hogares peruanos. Eso es un hogar digital, aunque ya existen algunos modelos de casas inteligentes en el Perú.

2.3.2. Domótica

El termino domótica yace desde muchos años atrás con la idea de crear objetos que interactúen por sí solos con un control automatizado y de esta manera poder controlar objetos con un dispositivo móvil, se da referencias acerca de una asociación española de domótica e inmotica, que describe a la domótica como un conjunto de soluciones basadas en tecnologías que permiten optimizar algunos aspectos de una vivienda como (confort, seguridad, ahora de consumo de energía, comunicaciones, entretenimiento, etc.) en otros términos la domótica sería un solución de automatizar y controlar diferentes elementos que componen una vivienda, y estos dispositivos serán capaces de comunicarse entre sí y operar bajo un programa(algoritmo) o configurada previamente por el usuario menciona Culquichicon (2012)

- Mejorar la calidad de vida de las personas que habitan la vivienda.
- Reducción del trabajo doméstico.
- Aumentar el bienestar y la seguridad.
- Racionalización de los consumos de energía, agua potable.

2.3.2.1. *Elementos de un sistema domótica*

Un sistema está compuesto por una capa de sensores, otra capa de controlador, la otra capa por interface y por último una capa de actuadores en la cual cada capa específica determina una función primordial en la domótica indica Calvo (2014) y podemos ver esto en la *Figura 4*, acerca de los componentes de un sistema demótico que puede estar constituido de una cantidad específica de redes, que pueden ser el control como de comunicación que se puede ubicar dentro o fuera la instalación, como ejemplo las redes de acceso a internet, trabajan en conjunto para generar una mejor experiencia servicio de gestión energética y servicio de comunicación, gracias a la integración de servicios de comunicación el sistema demótico y a sus sensores, actuadores, interfaces. Los actuadores pueden ser controlado sin ningún problema desde otros lugares.

Figura 4. Elementos de un sistema demótico

Fuente: Propia

2.3.2.2. *Arquitectura*

Se menciona que los sistemas domóticos se clasifican según su arquitectura el primero es la arquitectura centralizada donde se tiene por entendido que solo uno tiene acceso a un sistema y que nadie más que el con su usuario y contraseña es capaz de entrar en el sistema que si el no da permiso, nadie más puede ingresar al sistema, la segunda arquitectura es la descentralizada. La descentralizada al contrario tiene una acceso independientemente al mismo sistema con los mismo derechos que los que registren en el sistema y por último la tercera arquitectura que es distribuida, en la cual es acceso está abierto a cualquier usuario sin ser restringido por ningún permiso estas son 3 de las bases de la arquitectura en la domótica y según Pérez (2018) una arquitectura similar al de enriques comenta Culquichicon (2012).

2.3.3. ¿Qué es internet de las cosas?

2.3.3.1. *Contexto de internet de las cosas*

Menciona Solano (2014) el termino IoT, se refiriere al entorno en donde todo dispositivos u objetos que se utiliza en nuestra vida cotidiana, están interconectados entre sí, el termino internet de las cosas(IoT) fue acuñado por Kevin Aston y utilizado por primera vez en el año 1999, en el instituto de Massachusetts, se define como una red automática que actúa independientemente

previamente configurado el concepto que se tiene es que los objetos sean capaces de procesar información, comunicación entre ellos y con el medio ambiente y que tome decisiones de manera autónoma es una representación en el mundo virtual de objetos físicos que manda una información entre objetos y de esta manera cada uno de ellos realiza una función específica y de ello tener un control.

Teniendo un entendimiento acerca de la IoT. Son sistemas que actúan en ambientes múltiples como ejemplo tenemos las oficinas, bibliotecas, cocina, dormitorios, duchas, automóviles cabinas de internet, hospitales, municipios, colegios, bancos, etc. En otras palabras no hay lugar alguno donde no se pueda aplicar internet de las cosas ya que cubre un gran campo de aplicación y de este modo nos facilita nuestra vida, Solano (2014) menciona que para realizar o hacer posible que un objeto transmita sus datos es necesario el uso de tecnologías y habilidades como son las RFID (radio Frecuencia Identificación tecnología), dispositivos ubicuos la computación ubicua está relacionado con internet de las cosas en donde los objetos físicos y habilitados por una larga escala de sensores embebidos

Da un resumen acerca de los datos y como se transmiten en internet de las cosas y el concepto acerca la computación ubicua que lleva por concepto en Ingeniera de software y las ciencias de la computación, entendida con la integración de la informática en el entorno de la persona de forma que los ordenadores no se perciban como objetivos diferenciados es algo así como controlar objetos a través de conexiones inalámbricas para la facilidad en nuestra vida el termino de computación ubicua fue acuñada e introducido por primera vez por Mark Weiser en el año 1988, cuando trabajaba para Xerox, y menciona una frase muy curiosa “las tecnologías más profundas son las que desaparecen, se tejen entre ellas mismas sobre el tejido de la vida cotidiana hasta que no se distinguen de esta”.

2.3.3.2. *Estructura de una red IoT*

a. Elementos de internet de las cosas

Según Solano (2014) menciona que internet de las cosas será una realidad siempre y cuando existan tres componentes hardware formado por sensores, actuadores y sistemas de comunicación embebidos, por segundo tenemos el middleware o software que estas compuesto por herramientas de computación para analizar datos y sistemas de almacenamiento y por tercero es la presentación: herramientas de visualización, diseño que puedan ser compatibles con distintas plataforma estos son los tres elementos de las IoT que si no cumpliese no sería una estructura de internet de las cosas.

b. Tecnología identificación de radio frecuencia

Las tecnologías que son usadas para IoT es la (Auto-ID) conocida como identificación automática un ejemplo que se da es la tecnología de identificación de radio frecuencias conocida como (RFID) que son utilizadas para recuperar y almacenar datos de forma remota a través de dispositivos llamados etiquetas o tags RFID, los tags RFID almacenan una antena que es la que trasmite y recibe las señales mediante ondas de radio y un circuito integrado que se encarga de almacenar la información. Los tags pueden ser adheridos a los objetos e incluso implantados en una persona o animal. Los tags RFID si han popularizado en gran manera y su uso más demandado se da en carreteras con peaje, gestión de pasaporte, bibliotecas, atención sanitaria, entre otras, aunque tiene algunos inconvenientes como la parte de seguridad, ya que la etiquetas pueden ser leídas a un distancia por cualquier usuario, aunque este no sea el propietario del elemento como menciona (Solano, 2014)

c. Redes de sensores inalámbricos

Wireless sensor networks (WSN) o redes de sensores inalámbricos es una red inalámbrica de sensores que está compuesta por un conjunto o colección de dispositivos

independientes, distribuidos físicamente, y que tienen la habilidad y capacidad de almacenar y comunicar datos en una red de forma inalámbricas como muestra la *Figura 5*, son colocados en áreas de interés y su usan para el control de eventos, procesos, condiciones, ambientales, etc. Cada sensor actúa como enrutador ya que emite datos por múltiples nodos de entrada conectada a otras redes o internet. En nuestros tiempos la tecnología de sensores inalámbricos ha mejorado por que ya existen sensores inteligentes capaces de procesar, analizar, diseminar una información recopilada del entorno permitiendo que esta tecnología sea más viable los factores más influentes para el éxito radican en las mejoras que ha habido con relación a la eficiencia, el bajo costo y la disponibilidad de servicios etc.

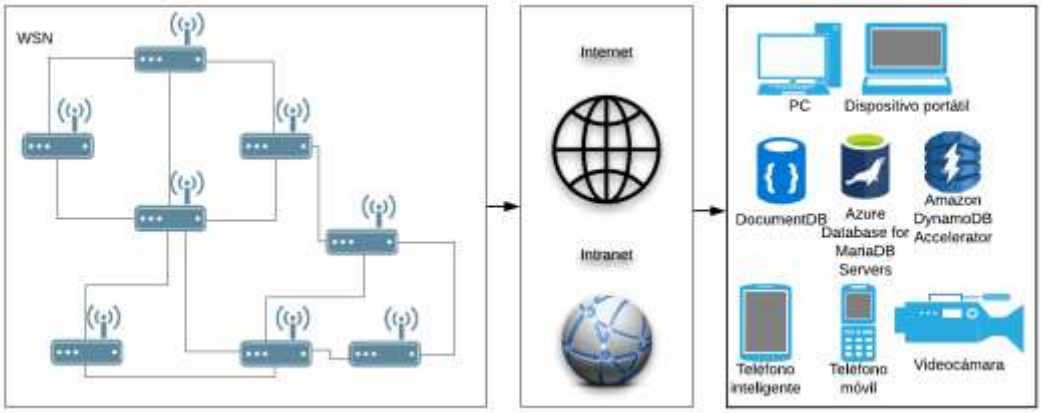


Figura 5. Red de sensores inalámbricos
Fuente: Solano (2014), mejorada por mi persona

2.3.4. Placas de hardware libre

2.3.4.1. *Contexto de placas de hardware libre*

Las placas de hardware libre son muchas en el mercado cuando se refiere a placas de hardware libre tenemos una gran variedad de las cuales pasan todo un proceso de fabricación y modificación con químicos impresiones en tinta laser para crear las pistas, pista de componentes y ya al termino de grabar todas la pistas cubrir el circuito con un químico para que no sea afectada por agua o

algún líquido, al pasar los años han ido mejorando este proceso a nivel que empresas como EUROCIR SA (Barcelona), ELECTONICA FALCON SA (Navarra). CIPSACIRCUITS SA (Barcelona) estas tres empresas son las que lideran el mundo de quemado de placas, posteriormente la compañía como Arduino AG (Arduino), Asus (Tinker Board) y empresa como techbase (Rasberry), hardkernel (Odroid) son los principales distribuidores de placas open source y cada una de ella tiene distintos componentes como muestra la Tabla *1*.

Tabla 1. *Placas de open source y hardware libre*

Herramienta	características
Tinker Board	La placa cuenta con núcleos ARM que incorpora un procesador Rockchip RK3288 cuenta con 2Gb de memoria y doble canal LPDDR3 incorporada con una interface de SD 3.8 también cuenta con una gráficos HD y UHD con salida de audio puerto de Ethernet, 4 puertos USB, pines de Coneccion para iot y cuenta con wifi y bluethoot (©ASUSTeK Computer Inc., 2018)
Odroid	Tiene un procesador Samsung exynos5422 cuenta con 2Gb con LPDDR3 de RAM con puertos USB Genesys GL3521 salida de 3.0 puerto Ethernet y cuenta con una entrada extra de energía a traves de batería más entrada de 5 voltio por Jack (Hardkernel co., 2019).
Rasberry PI	Cuenta con un procesador Broadcom BCM2837B0, Cortex-A53 64-Bit SoC @ 1.4GHz memoria de 1GB LPDDR2 Sdram conectividad 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE Gigabit Ethernet con USB 2.0 (velocidad máxima 300Mbps) 4 × entradas USB 2.0 acceso de pines Gpio de 40 pines y la parte de video y sonido MIPI), HDMI, video compuesto, MicroSDHC, jack 3.5mm y por último la parte de multimedia Códec de video H.264, MPEG-4 (1080p30), OpenGL ES 1.1, 2.0 (Electrónica ElectroPro, 2017).

Arduino	Cuenta con un micro controlador Atmega328P con fuente de alimentación de 5 voltios cuenta con 6 pines analógicos y 6 digitales una memoria flash 32 kb con Sdram 2kb eeprom 1kb y cuenta con una gran cantidad de sensores a utilizar (Arduino, 2019)

Fuente: Propia

Teniendo una variedad de placas open source se da la elección a la placa Arduino por la gran cantidad de sensores que pueden adaptarse a cualquier entorno y de fácil entendimiento en un hogar y la multifuncionalidad que tiene para internet de las cosas.

2.3.5. Sensores

2.3.5.1. *Contexto de sensores*

El tesista Calvo (2014) informa que los sensores son dispositivos responsables de transformar un tipo de magnitud física a una señal eléctrica proporcional a la variable medida, fundamentales en el control del estado de las diversas variedades existentes en una vivienda usadas posteriormente enviarlas al controlador principal o procesarlas previamente y luego transmitirlas. Las variables a evaluar dentro de un hogar son múltiples a continuación se señalan una cuantas que son utilizadas para el control de variables comunes una breve descripción de que son los sensores y dar a conocer algunas categorías de sensores existentes como muestra la Tabla 2.

Salazar (2014) menciona como aportación de un libro de internet de las cosas como versión de prueba acerca de los sensores, conocidos como uno de sus pilares constructivos de internet de las cosas, como sistemas ubicuos pueden implementarse en todas partes también pueden ser implantados bajo la piel humana, en un bolso, prendas de vestir, etc. Algunos pueden tener un tamaño diminuto pero los datos que recogen pueden ser recibidos a cientos de millas de distancia. Complementan los sentidos humanos y se han vuelto indispensables en un gran número de industrias, desde la salud hasta construcción. Los sensores poseen la ventaja clave de poder

anticiparse a las necesidades humanas en base a la información recopilada sobre su entorno. Su inteligencia multiplicada por numerosas redes les permite no sólo informar sobre el entorno, sino también tomar medidas sin la intervención humana nos da a entender que los sensores han llegado al punto de poder controlarlos a una distancia muy grande y poder de esa manera aplicar a cualquier parte y más a un facilitarnos la vida en el día a día de cada familia y hogar.

Tabla 2. *Tipos de sensores*

Herramienta	características
Sensor de luminosidad	Dentro de estos sensores tenemos las fotoceldas la cual no tiene una polaridad y muestra datos acerca la luminosidad.
Sensor de humedad	En los sensores de humedad tiene 2 clasificaciones una de humedad del ambiente y la otra humedad de suelos la cual en ambos casos tienen tiene un pin de señal.
Sensor de presencia	Dentro de estos sensores tenemos el sensor par, y el sensor infrarrojo los cuales detectan objetos y tienen un pin para la señal de salida tienen un pin de señal para recepción de datos.
Sensor de distancia	Este sensor nos ayuda a medir objetos de un punto a otro y según a ello podemos nosotros poder acciones
Sensor de gas	El sensor de gas tiene una gran variedad ya tiene la capacidad de determinar la cantidad gas que está a nuestro alrededor tiene un pin de señal para la recepción de datos
Sensor táctil	El sensor táctil puede darse de tres maneras uno en las pantallas cd, los lectores de huellas y por último el teclado matricial los cuales tiene su alimentación

	el pin de señal y en la lcd tiene un controlador de saturación
Sensores de comunicación	En los sensores de comunicación tenemos el módulo bluethoot, la placa Ethernet Shell, módulo wifi y la placas GSM que son un puente para interconectar con más dispositivos

Fuente: Propia

Dentro de gran variedad de sensores se tiene la selección de sensores de comunicación que sera los más indicada para el proyecto que se ha de desarrollar en la presente investigación.

2.3.6. Actuadores

2.3.6.1. *Contexto de actuadores*

Los actuadores son otra parte muy importante de internet de las cosas motivo por el cual son encargados de realizar acciones gracias a los sensores, en un ambiente puede influir motores, accionadores, referencia Calvo (2014) Un elemento actuador es aquel que permite concretar la operación iniciada por el controlador, se encarga de transformar un tipo de energía en la activación de un proceso cualquiera. El elemento actuador recibe la señal de control y activa por ejemplo un extractor de aire si existe humedad relativa sobre un nivel máximo o activa la electroválvula del riego automático o la electroválvula encargada de controlar el paso del gas o el agua en caso de presentarse alguna fuga, muestra los tipos de actuadores la Tabla *3*.

Tabla 3. *Tipos de actuadores*

Herramienta	características
Actuador hidráulico	Utiliza la presión de algún liquido viscoso para realizar una acción y es empleada cuando necesita de potencia
Actuador neumático	Utiliza la presión de aire comprimido para poder ser utilizado en la mecánica.
Actuador eléctrico	Este actuador es impulsado por una energía alterna o constante conocida como AC y DC.

Fuente Propia

El actuador elegido nos ayudara al desarrollo más rápido del proyecto por motivo de ser al más óptimo para lo que se aplicara siendo este el actuador eléctrico.

2.3.7. Integración

2.3.7.1. *Contexto de la integración*

Según el libro tech pedia versión prueba de los Salazar (2014) menciona acerca el tema de la integración de internet de las cosas La integración de dispositivos inteligentes en los envases, o mejor, en los propios productos permitirá un ahorro de costes significativo y aumentar la simpatía de los productos Económicos. Se continuará con el uso de chips y antenas integrados en sustratos no convencionales como los tejidos textiles y el papel, y el desarrollo de nuevos sustratos, conductores y materiales de unión adecuados para ambientes hostiles y para la eliminación de residuos ecológicamente. La tecnología System-in-Package (SiP) permite la integración flexible y 3D de diferentes elementos como antenas, sensores activos y componentes pasivos en el envase, mejorando el rendimiento y reduciendo el coste de la etiqueta. Incrustaciones RFID con estructura de acoplamiento por tiras se utilizan para conectar el chip del circuito integrado y la antena con el fin de producir una variedad de formas y tamaños de etiquetas, en lugar de montaje directo.

2.3.8. Capas de dispositivos

En la parte de Capa de Dispositivos se encuentra en la base de la arquitectura, es núcleo de todo y es la encargada de múltiples funciones, relacionadas fundamentalmente con el medio físico en el cual se encuentra el dispositivo IoT. La *Figura 6* muestra las principales funciones establecidas para esta capa, las cuales se explican a continuación y depende mucho de las capas de las cuales identificamos, modularizados y al mismo tiempo damos mayor peso al área de internet de las cosas.

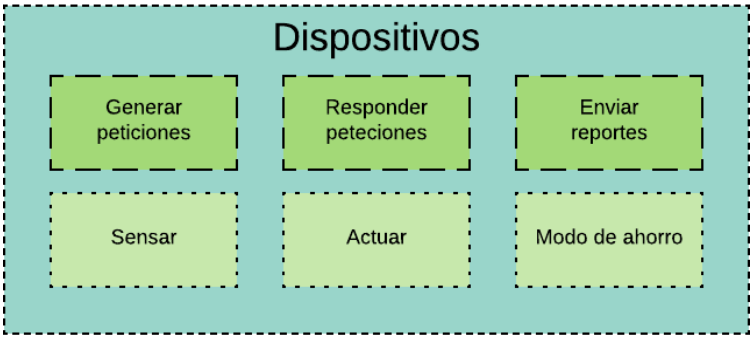


Figura 6. Principales funciones de la capa de dispositivos

Fuente: Architectural proposal for internet of things Benítez (2017)

2.3.8.1. *Seguridad de agregación de datos*

d. Sensar

Permite obtener datos del medio físico del cual podemos sacar datos estadísticos y medirlos estos datos puede ser de temperatura, distancia, presión atmosférica, foto, video, sonido, presencia, gas, etc. De acuerdo a cada dato que emita los sensores puede uno almacenar, y de ello sacar reportes de temperatura de distancia de presión atmosférica, foto, video, presencia, gas, etc. Permite obtener datos del medio físico, estos datos pueden ser, por ejemplo, temperatura, presión atmosférica, un video, o un sonido.

e. Actuar

Se basa en la ejecución de acciones sobre el medio físico, por ejemplo, mover un objeto, apagar el aire acondicionado, cambiar la luz de un semáforo, o abrir el garaje de esta manera que teniendo una forma de como manipular objetos se nos facilita el modo de poder controlar objetos inanimados que lleguen a cumplir una función a través de una dispositivo móvil de controlar de distintos lugares y que el sistema cumpla la función de gravar mover algo abrir o cerrar por ello nos ayuda de mucho la parte de actuar en un sistema domótica.

f. Generar peticiones

Un dispositivo puede contar, como parte de sus funciones, con la posibilidad de realizar peticiones, las cuales pueden ser, por ejemplo, para que otro dispositivo ejecute una acción, solicitar un dato a la nube para actuar en dependencia del resultado, solicitud para entrar en modo de ahorro, etc. También generando petición podemos obtener datos para un registro como de ingreso a un hogar o bloquear un sector desde una distancia y no tengamos que realizar alguna acción para cumplir las peticiones requeridas.

g. Responder peticiones

Se puede dar respuesta a las peticiones que llegan desde otros dispositivos, el Gateway, la nube, una aplicación, etc. Estas peticiones pueden solicitar un dato o una acción, por citar dos ejemplos como al hacer una petición genera un dato ese dato también puede contener la opción de responder como ejemplo sería un cajero, informador virtual que, de instrucción para poder guiar, conducir a nuevas acciones entonces se es de mucha ayuda en ciertos campos.

h. Enviar reportes

Los dispositivos pueden, de manera automática, estar configurados para enviar reportes de datos cuando, por ejemplo, han coleccionado una cantidad de datos determinada, un dato muestreado superó un umbral definido con anterioridad, etc al enviar reportes ya sea de nombres, imágenes, video, código de ello se reporta en una fecha determinada sea para incrementar la economía o saber lo quien ingresa a una casa sin acceso y muchos más.

i. Modo ahorro

Muchos dispositivos IoT necesitan ahorrar energía, por lo cual cuentan con esta función, la cual puede, por ejemplo, ponerlos en modo de suspensión hablando más acerca del modo de ahorro tenemos la parte de economizar la energía en el mundo tenemos muchas cosas que consumen mucha energía y eso indica a que más empresas generen más electricidad y su consumo de creación de hidroeléctricas cada vez son más grande y consumen más recursos y de ello debemos nosotros mejorarla.

2.3.8.2. *Capas a nivel general*

Tabla 4. *Descripción de las capas 7 capas internet de las cosas*

Herramienta	características
Capas de dispositivos	La capa de dispositivos es la parte donde entren los celulares, tablets, televisores, etc. Es la plataforma en otras palabras.
Capas de Gateway	son los encargados de enrutar lo físico con lo digital son importante para enviar como para recibir
Capas de Red	La capa de red es el nivel de datos que se transfieren ya sea paquetes, nombres y contraseñas.

Capa de nube y centro de datos	El nivel o capa de nube y centro de datos da referencia al almacén donde cada dato, paquete es registrado y guardado para generar reportes de ello
Capa de aplicación	Es la plataforma donde se pueden ver los tipos de controles, acceso y manejar de los sensores y actuadores.
Capas de gestión	Es la parte de procesos de como realizara cada acción y esto es la parte de desarrollo de software.
Capas de seguridad	El nivel que necesariamente tiene que tener todo sistema domótico los niveles de acceso que debe tener cada usuario

Fuente: Propia

Para este caso se empleará todos los niveles de capa para tener mayor soporte y más completo el sistema de seguridad de la cual cada nivel es de mucha importancia de la cual es necesario abarcar cada capa a detalle para una arquitectura ideal.

2.3.9. Tipo de datos en internet de las cosas

"Los datos pueden ser de varios tipos, existen datos que generan de forma automática mientras otros datos pueden ser introducidos manualmente de los cuales son necesarios saber qué tipos de datos son los que se están guardando para tomar medidas a partir de este punto se saca encuestas de todo tipo si son de ingresos a páginas esto beneficia a la empresa como a los adquirientes, pero en general lo podemos clasificar de la siguiente manera".

2.3.10. Identificación por radiofrecuencia

"Son los datos recopilados por los tags RFID y provienen de diferentes fuentes, la mayoría de estos datos son generados en tiempo real. cada objeto que forma parte de IoT posee identificador único, que es lo que lo distingue de los demás, tomando en cuenta la enorme cantidad de objetos

que formarán parte de IoT en un futuro, se hace necesario un sistema de direccionamiento que pueda suplir estas necesidades. Uno de los avances notables que ha habido en cuestión de direccionamiento IP, ha sido el surgimiento del protocolo de internet de la versión (IPv6), que se espera pueda reemplazar la versión (IPv4) que está sufriendo el problema de agotamiento de direcciones, sin embargo, IPv6 todavía no está extendido".

2.3.11. Datos sensor

"Las redes de sensores inalámbricos son las vías por la que los datos entran a IoT. Por medio de estos sensores se pueden monitorizar el clima, la temperatura y el ruido. Los sensores inalámbricos y las tecnologías de redes han hecho posible la captura de una elevada cantidad de datos de manera rápida y no solo se puede capturar posiciones con sensores de más nivel que pueden ser muy manejables para sistemas complejos de impresión o fabricación de medidas los sensores son los más adecuados para esos campos"

2.3.12. Datos históricos

Los datos que son almacenados a través de sensores pueden ser requeridos en un futuro para realizar algún análisis o minería de datos y extraer conocimiento de ellos. Las aplicaciones de diseño deberían ser orientadas para que de forma automática analicen qué datos deberían conservarse y cómo hacerlo ya teniendo unos datos históricos se puede aplicar un análisis para la predicción de la mejor del proyecto y su madurez por ello se debe tener datos reales de los cuales se da el incremento de más estudios y mejoras aplicadas a los ámbitos que pueda adaptarse" Solano (2014).

2.3.13. Impacto de internet de las cosas

"Lo primero que se debe observar e imaginar es el gran impacto que se producirá en todos los sectores de actividad. Una red tan grande y con tantos objetos "inteligentes" conectados entre sí necesita unas condiciones necesarias para que se pueda implantar correctamente. Un sector

profesional podría desaparecer, otros en cambio, comenzarán gracias a la nueva era de la tecnología. En este escenario, quedarán los que sepan adaptarse más rápido a estos cambios, así como ocurrió con las tecnologías de la información. Según la firma de analistas Gartner, en todos los sectores tendrá un gran impacto toda la red de dispositivos conectados, desde empresas, autoridades locales, hospitales y consumidores, llegando a unos gastos totales de 260.000 millones en 2020.

Estos datos suenan prometedores en cuanto a la economía, pero, ¿En qué sector se estima que será el que tenga más objetos conectados? El estudio realizado por Gartner que se presentó en la "Gartner Symposium/ITxpo" del 9 de noviembre de 2014 en Barcelona, estima que las aplicaciones para los consumidores, principalmente en el sector de la automoción, serán las que más difusión en los próximos años. Calculan que aproximadamente 13.000 millones de dispositivos conectados pertenecerán al sector de consumo. Como indican en el estudio: "El número de dispositivos inteligentes conectados seguirá creciendo de manera exponencial, ofreciendo a los objetos inteligentes la capacidad de sentir, interpretar, comunicar y negociar, y efectivamente tener una 'voz' digital", según Steve Prentice, vicepresidente de la consultora. "Los CEO deben buscar oportunidades para crear nuevos servicios, escenarios de uso y modelos de negocio basados en este crecimiento". Seguido del sector consumo, afectará también en gran medida al sector industrial los servicios públicos, la fabricación y el transporte serán los mercados clave de esta tecnología.

Las empresas estarán obligadas a buscar un equilibrio entre los datos recogidos y almacenados en la red por los dispositivos y sensores conectados, y el análisis de toda esa cantidad de información con el riesgo de su pérdida o mal uso. Todos los retos que plantea el IoT para la seguridad de toda esa información pondrán a las empresas en la obligación de invertir en seguridad tecnológica a unos niveles que nunca antes se habían pensado. Terminando con el análisis de beneficios, IDC (2015) (International Data Corporation, especialistas en información tecnológica) calcula que el mercado mundial de IoT crecerá hasta llegar a 3.040 millones de dólares hasta el año 2020" Castro (2016).

2.3.14. Impacto en las personas

"Interconexión de personas con personas, de personas con cosas y de cosas con otras cosas. En la misma definición de la tecnología ya se puede observar el impacto que tiene y tendrá en la sociedad IoT. Todo estará interconectado con nosotros, basándose como referencia nuestro teléfono móvil que será la central con la que nos conectaremos al resto del mundo. Y es que todas las aplicaciones que se están desarrollando actualmente ya dan por hecho que el usuario tiene una conexión a internet y se podrá conectar a una red sin problema. Los desarrolladores de aplicaciones y hardware, solo tienen que reinventar lo que ya existe, dotándolo de conexión. Veremos más adelante todos los peligros que esto conlleva, porque existen muchos riesgos en todo el proceso de desarrollo comercialización usuario.

La conexión en todo momento en una red, sin importar dónde ni cuándo, permaneciendo intercambiando datos con todos los destinos muestra de por sí el peligro que puede ocasionar de la resolución de esos problemas dependerá que el impacto hacia las personas sea bueno y útil, o sea un verdadero problema y una violación de todos los derechos de la sociedad" Castro (2016).

2.3.15. Impacto internet de las cosas en las personas

"Interconexión de personas con personas, de personas con cosas y de cosas con otras cosas. En la misma definición de la tecnología ya se puede observar el impacto que tiene y tendrá en la sociedad IoT. Todo estará interconectado con nosotros, basándose como referencia nuestro teléfono móvil que será la central con la que nos conectaremos al resto del mundo. Y es que todas las aplicaciones que se están desarrollando actualmente ya dan por hecho que el usuario tiene una conexión a internet y se podrá conectar a una red sin problema. Los desarrolladores de aplicaciones y hardware, solo tienen que reinventar lo que ya existe, dotándolo de conexión. Veremos más adelante todos los peligros que esto conlleva, porque existen muchos riesgos en todo el proceso de desarrollo-comercialización-usuario. La conexión en todo momento en una red, sin importar dónde ni cuándo, permaneciendo inter- cambiando datos con todos los destinos muestra de por sí el peligro que puede ocasionar. De la resolución de esos problemas dependerá

que el impacto hacia las personas sea bueno y útil, o sea un verdadero problema y una violación de todos los derechos de la sociedad" Castro (2016).

2.3.16. Impacto en las consumiciones de recurso

"La preocupación que existe hoy en día por el estado de salud de la tierra, las energías renovables, y el consumo de recursos a nivel en el que lo estamos haciendo, ayudado por el cambio climático y el aumento masivo de la población, ha hecho que los avances tecnológicos se centren en ayudar a frenar el proceso lo máximo posible la tecnología de Internet de las Cosas parece estar diseñada especialmente para esta tarea y es uno de los campos más prometedores. El uso de sensores puede controlar las diferentes variables necesarias para el mantenimiento de estos recursos en uno de los siguientes apartados comentaremos en más profundidad las diferentes aplicaciones derivadas del IoT en el ámbito de las energías renovables" Castro (2016).

2.3.17. Seguridad

2.3.17.1. *Contexto de la seguridad*

“Las aplicaciones son diversas al igual que en confort o gestión energética, pasando desde el control de gases peligrosos dentro de la vivienda, o inflamables, como gas, que, al haber una fuga, el sistema una vez detectado el hecho se encargará de cerrar la electroválvula de la red de gas, o la electroválvula de la red de agua en caso de presentarse una rotura en la red. Detectar humo. Es posible identificar un posible foco de incendio antes de que se materialice al detectar una cierta concentración de humo en la vivienda, tomando como medida la activación del sistema de incendios en la habitación alertada. Alarmas de intrusos. Con un sistema domótica es posible alertar al usuario de la presencia de una persona ajena a la vivienda y generar una señal local y remota del suceso. Monitoreo. Gracias a la integración de los sistemas de comunicación, acceder a la red domótica no es un problema, siempre que se tenga acceso a internet, es posible monitorear el estado de la vivienda a través de los indicadores del sistema y/o de cámaras destinadas para ello. Además brinda mayores servicios de accesibilidad para personas que lo requieran prescindiendo del típico interruptor de luz o apertura/cierre de persianas entre otras cosas beneficiando a individuos con algún grado” Calvo (2014).

2.3.17.2. *Seguridad en la trasmisión de datos*

Se procederá entonces a analizar detalladamente los posibles agujeros (o canales) de los que se puede sacar información en la vida útil de los dispositivos de IoT. Se empezará con la que parece la vía más obvia en cuanto a posibles ataques se refiere. La transmisión de datos entre los dispositivos conectados a IoT, puesto que estos dispositivos están enfocados a estar continuamente transmitiendo información entre ellos o hacia la nube. Es fácil reconocer que es en este aspecto donde debemos centrarnos en defendernos respecto a posibles ataques ya que los sistemas distribuidos emplean numerosos canales de distribución, ya sea inalámbricos, por cable, etc. Todas estas vías de información, sobre todo las que son inalámbricas y públicas, son susceptibles de sufrir ataques.

Es por eso que estos dispositivos necesitan garantizar un nivel de seguridad mínimo en cuanto a la integridad, protección y encriptación de sus comunicaciones ya que, si no se proporcionan estos niveles de seguridad, no será complicado que un atacante pueda acceder a esa información intercambiada. La información puede contener tanto datos privados como datos personales, incluso información acerca de los dispositivos que puedan permitir el control del mismo.

Es necesario por tanto un buen sistema de cifrado de datos para evitar los ataques de tipo Man in Té Middle, en el que el atacante se encuentra en el medio de los dos extremos de la comunicación y simplemente interfiere todos los mensajes de la misma sin que los comunicadores se den cuenta de que están siendo interceptados y posiblemente alterados. Siendo esto último altamente peligroso debido a que, si la información que se modifica actúa directamente sobre el dispositivo y su actividad, puede causar un gran problema, siendo incluso peligroso para la vida de las personas.

Vamos a poner un escenario en el que se va a poder apreciar las consecuencias de un posible ataque en la comunicación entre un dispositivo y su conexión con los datos a la nube. Supongamos que hemos comprado una moto de último modelo, que incorpora entre otras cosas un sistema inteligente conectado a la red que ofrece servicios ofrecidos por el fabricante, facilitándonos información como la velocidad, posición, tiempo medio, estado general de la moto, etc. Supongamos también que la comunicación entre nuestro vehículo e internet es poco segura y tiene un nivel bajo de encriptación. Un atacante podría acceder a la comunicación y obtener valores como la posición del vehículo, sabiendo en todo momento la ruta que seguimos, dónde vivimos, donde trabajamos etc. También podría conocer el estado del motor, nivel de líquidos el otro factor importante, es que, si este atacante puede modificar la información en la comunicación y pudiera dar las mismas órdenes que damos nosotros, podría hacer que se encendieran las luces, cambiar de marcha, acelerar, frenar, dependiendo de las funcionalidades que permita hacer el dispositivo de IoT Castro (2016).

j. Seguridad en el software

> “Se pasará ahora a otro de los canales sobre los que se pueden realizar más ataques. Es el aprovechamiento de las debilidades en el software en los dispositivos IoT cuando

se crea un dispositivo y se le añade un sistema operativo, normalmente se utilizan versiones simplificadas de los sistemas operativos de uso común como pueden ser Windows, Linux, etc. De esta forma los costes de fabricación de estos dispositivos se reducen considerablemente. Esta práctica, evidentemente supone un riesgo a la seguridad ya que cuando se detecta una vulnerabilidad en estos sistemas operativos se explotan en todos los dispositivos que lo tengan instalado.

Otra característica que se obtiene al desarrollar el dispositivo abaratando costes en su desarrollo, es la interfaz web. Estos, al ser de pequeño tamaño, y no disponer la mayoría de una pantalla, tienen que ser controlados a través de internet. Si estas interfaces para acceder a los dispositivos y configurarlos, manipularlos, etc, no se protegen debidamente, cuando se produzca algún ataque y accedan intrusos, podrán acceder a todos los dispositivos que usen esa interfaz.

Una última característica que tienen en común la mayoría de dispositivos de IoT es su acceso a los servicios en la nube. Si en estas plataformas de servicio no se llevan a cabo tareas de mantenimiento, actualización y protección, puede ser una vía muy peligrosa para poder acceder a toda la información y poder tomar el control de los dispositivos algunos dispositivos como Smartphone o televisiones inteligentes, pueden acceder a repositorios de aplicaciones para añadir nuevas funcionalidades. Ésta puede ser la puerta de entrada a aplicaciones maliciosas que puedan tomar el control de los dispositivos, explotar vulnerabilidades, obtener información confidencial, o incluso descargando más aplicaciones maliciosas sin nuestro consentimiento por tanto, de igual manera, los responsables de estos servicios en la nube y repositorios tienen la misma responsabilidad de mantener correctamente sus servicios para permitir que los dispositivos accedan a ellos sin ningún problema para su seguridad" Castro (2016).

k. Seguridad en la configuración y funcionalidad

"Muchas veces, el principal problema en cuanto a seguridad se refiere, radica principalmente en la configuración (por defecto, y posibles opciones configurables) y la

funcionalidad del propio dispositivo muchos fabricantes, a la hora de establecer la configuración por defecto del dispositivo, eligen unas opciones que posiblemente el usuario no va a utilizar nunca, o bien, que por estar activadas y permitir el uso de esa funcionalidad avanzada, el usuario no tenga suficientes conocimientos y la emplee mal, produciendo una brecha de seguridad y posible acceso para intrusos de nuevo, entra en juego el papel de fabricantes, que son los responsables directos de esta posible vía de acceso de intrusos. Los distribuidores deben de ser conscientes, y adoptar una política de configuración segura, permitiendo además a los usuarios poder configurar el dispositivo acorde a sus necesidades, sin ser un sistema rígido, marcado con las opciones de fábrica.

Se pondrá un ejemplo para poder entender mejor este problema cuando adquirimos un Smartphone, éste viene con unas configuraciones por defecto generales del dispositivo, y además viene con un apartado de configuración para redes. Esta configuración por defecto puede ser peligrosa. Si viene activado para que el dispositivo esté conectado permanentemente a internet, y un usuario inexperto no es consciente de esto, puede ser un problema ya que el dispositivo estará continuamente conectado a internet, descargando y transfiriendo datos, posiblemente sin el conocimiento del usuario" Castro (2016).

l. Seguridad en el hardware

"Para este caso, esta posible brecha de seguridad ya existe antes de IoT. Los problemas debidos a la mala política de seguridad para el hardware son las menos frecuentes, pero lamentablemente son las que producen problemas más difíciles de solucionar obviamente, los problemas de seguridad en el hardware se producen sin la necesidad de estar conectados a internet, pero puesto que los dispositivos de IoT también cuentan con este problema, se ha incluido en esta lista de posibles brechas de seguridad.

Los ataques contra el hardware se producen, sobre todo, cuando el dispositivo tiene una gran seguridad a nivel de software, cuando se encuentran en puntos aislados de la

red, o cuando están bien protegidas para un acceso a través de internet es entonces cuando se producen estos ataques, que suelen estar dirigidos a los componentes conectados a la red eléctrica la diferencia de estos ataques es que para realizarlos es necesario un equipamiento especializado para producirlos. Según el equipo que del que se disponga se podrán realizar diferentes tipos de ataques, desde ingeniería inversa a monitorización de interfaces los ataques más habituales son los accesos a la información tanto volátil y no volátil como memoria RAM y discos duros. Si podemos acceder a la memoria no volátil, es posible acceder a claves, información de acceso, etc. Si podemos acceder a la memoria no volátil, se puede acceder a toda la información guardada.

Hay dos posibles protecciones para este tipo de ataques el primero de ellos es garantizar que, si el dispositivo es manipulado, automáticamente se destruya la información que contenta. La segunda es un buen sistema de cifrado de información, de modo que, si finalmente acceden a los datos, les sea imposible descifrarlos un punto a tener en cuenta en este apartado es el borrado de información cuando un dato se borra de un dispositivo no volátil, se modifican las tablas de asignación de archivos en el sistema de ficheros, haciendo que el espacio que ocupaba el dato está disponible para otro nuevo dato, pero sin realizar un borrado efectivo existen herramientas que, sin un mínimo de pasadas de borrado, es posible recuperar la información que se encontraba en el disco" Castro (2016).

m. Seguridad de usuarios

"Se va a terminar con una vía de acceso que no está relacionada directamente con los dispositivos IoT, sino con los usuarios. En muchos casos, si todos los puntos anteriores como el software. Amenazas hardware, comunicaciones y configuraciones están perfectamente protegidos y construidos, un mal uso del usuario puede poner en serio peligro toda la información del dispositivo es probablemente el principal motivo por el que se producen ataques sobre los dispositivos ya que, como parte de la cadena, los usuarios somos el eslabón más débil y podemos cometer errores el principal ataque se basa en la llamada ingeniería social. Básicamente se centra en aprovechar los errores

humanos para comprometer la seguridad de los dispositivos mediante la confusión y el engaño.

Debido a que la mayoría de los accesos a las plataformas privadas en internet son a través de unas credenciales, la ingeniería social intenta acceder a ellas mediante estafas a través de correos electrónicos, sitios web falsos, o suplantación de identidad estos ataques están dirigidos a grandes empresas y clientes potenciales para poder obtener un mayor beneficio es necesario crear una cultura de seguridad para los usuarios y concienciarlos sobre los problemas que pueden producir si no ponen atención a sus actividades mientras utilizan los dispositivos ya que como hemos comentado anteriormente, no sirve de nada que un dispositivo esté perfectamente protegido, si nuestra clave de seguridad la apuntamos en nuestra red social preferida. De esto se hablará más adelante en el punto 4.6 del TFG" Castro (2016).

2.3.17.3. *Proteccion y privacidad Recomendaciones*

"Como se viene contando en todo este TFG, de todos los desafíos con los que se encuentra el desarrollo de IOT, la protección sobre la privacidad de los usuarios es el que genera mayor preocupación en los consumidores. Por ello, las autoridades europeas de protección de datos, destacando entre ellas la francesa (CNIL) y la española (AEPD) realizaron el primer dictamen en 2014 sobre las recomendaciones que debemos de seguir.

El grupo de trabajo de la Unión Europea del artículo 29 de protección de datos (G29) ha dado una serie de recomendaciones sobre tres tipos de sistemas de IOT los weareables computing (tecnología para llevar puesta), los dispositivos que registran actividades de las personas y su estilo de vida y la domótica entre las recomendaciones más importantes destacamos: "para que esta tecnología tenga éxito y de frutos es necesario, en palabras del G29, que los usuarios del Internet de las Cosas puedan permanecer siempre en control de sus datos y deben saber claramente a utilizar los mismos. Deben dar su consentimiento expreso tras recibir la información de forma clara y transparente. Esta claridad es otra de las bases del éxito de estas tecnologías".

Los retos que según el G29 resaltan para superar son: Controlar que los datos y la información que cae en manos de terceros. Es importante que el consumidor sepa en todo momento quien obtiene esos datos y qué hace con ellos, por tanto, el consentimiento es libre por parte del usuario el que se usen o no sus datos. Otro aspecto importante que controlar es que los datos se usen para el fin que fueron recogidos. El G29 también cree que es importante que las aplicaciones y dispositivos se puedan usar de forma anónima. La Agencia Española de Protección de Datos habla al respecto del dictamen del G29 la información personal sólo puede ser recogida para unos fines determinados, explícitos y legítimos. Este principio permite a los usuarios conocer cómo y con qué fines se están utilizando sus datos y decidir en consecuencia. Además, los datos recogidos deben limitarse a los estrictamente necesarios para la finalidad definida previamente. Los datos que son innecesarios para tal fin no deben ser recogidos y almacenados por si acaso o porque podrían ser útiles más adelante igual de importantes que las recomendaciones generales sobre la privacidad y seguridad del G29, son las recomendaciones dirigidas a otros sectores específicos" Castro (2016).

2.3.18. Android

Android es el sistema más predominante usado por la mayoría de marcas de Smartphone empezó a revolucionar con su primer sistema operativo donuts 1.6 en la cual gracias a ello tuvimos toda la información del mundo en nuestras manos del mismo modo sentó la bases para cualquier dispositivo al mismo tiempo creo la tienda virtual Android marker, pasados un tiempo innovo su primer modelo con eclair 2.1 su segundo modelo la cual trajo mejores navegación GPS, la personalización de la pantalla y los mensajes de voz todo un bum para android luego de un tiempo innovo más el modelo esto indico que seguiría mejorando sus sistemas operativos para lo cual presento android 2.2 froyo que trajo las siguientes características acciones con voz, se transformó en hostpost que sería capaces de dar internet a otros equipos además incorporo un compilador jit a dalvik y esto mejor en 5 veces más el rendimiento y android no se quedó ahí innovo más y presento android 2.3 gingerbread y trajo un api para los juegos y de esta manera sea mejor y alcance un nuevo nivel, trajo ademas NFC que transfiera datos de un dispositivo móvil a otro con solo juntarlos y un control de batería, luego sorprendió aún más con android 3.0 honeycomb esta marco el inicio de una nueva era por que trajo compatibilidad con tablets

también la parte de barras del sistema o gadgets para un mejor control y la configuración rápida fue su punto de crecimiento en adelante.

Android 4.0 ice cream sandwich creo otro concepto presentado por su versión anterior ya que conto con pantalla personalizable un control de consumo de datos y android beam que a traves de NFC podría compartir datos de un dispositivo a otro con solo juntarlos de ahí salió la versión de android 4.1 jelly beam inicio con google now asistente que te indicaba el clima, las notificación en pantalla y el cambio de cuenta para mejoro la versión 4 y la 4.1 fue mejorada en la versión 4.4 kit kat que tenía comandos de voz de google con solo decir "ok google" a traves de su micrófono del celular también trajo un diseño envolvente y la marcada inteligente de llamada que reconocía contactos con solo digitar unos cuantos números para la versión 5.0 de android llamada lollipop trajo consigo en su diseño materia desing la cual trajo modelo muy agradables a los ojos de los clientes y adaptable a muchas pantallas tables, relojes, tv autorradios, etc. También trajo la notificación en la pantalla de bloqueo en la cual podíamos saber quién no hablaba sin desbloquear el celular y la versión 6.0 llamada marshmallow trajo en si la multiactividad en el equipo los permisos para aplicaciones y una batería inteligente y más duradera en si así siguió android innovando hasta el día de hoy después del 6 0 vino el 7.0 llamado nougat, la versión 8.0 oreo y android 9.0 android pie la última en el mercado y todo esto gracias a la empresa google y que está basado en Linux y de este modo vemos que android por ser libre hay mayor facilidad de desarrollo Astrium (2014).

2.3.19. Aplicaciones

Las aplicaciones son complementos de un dispositivo móvil ya que gracias a estas llamadas aplicaciones es que podemos encontrar distintos tipos de funciones que traen, como puede ser de guardar archivos, para mejorar o retocar fotos, mejorar videos , recetas, reproductores de música y video, juegos en gran variedad, para rutinas de ejercicio, salud, redes sociales, correos electrónicos, aplicaciones de office calendario y muchos más la aplicaciones son los encargados de dar el soporte a la parte física en un sistema domótica en al cual es el dispositivo que a traves de la aplicaciones puede indicar acciones a objetos y se menciona las 3 aplicaciones como se muestra en la Tabla *5*.

Tabla 5. *Tipos de aplicaciones*

Herramienta	características
Aplicaciones nativas	Utiliza recursos como de sistema o hardware, publicada en tiendas de distribuciones mayoría no ceben estar conectadas a red
Aplicaciones web	Pueden ser utilizadas por cualquier dispositivo sin importar el sistema operativo puede requerir de un costo por el desarrollo ninguna aprobación para publicarla.
Aplicaciones hibridas	Son las más adecuada ya que son publicadas y más de un dispositivo puede usarla como se mencionó es multiplataforma.

Fuente: propia

Dentro de las tres aplicaciones la más adecuada es las aplicaciones nativas para poner en funcionamiento pronto del sistema de seguridad y a medida el proyecto tome madures puede optarse por las aplicaciones hibridas.

2.3.20. Ide de desarrollo

Los IDEs son los gestores o programas que ayudan a codificar o construir un aplicativo ya sea hibrido web o nativo cabe resaltar que estos programas tienen herramientas que ayudan a facilitar el desarrollo de aplicativos de las cuales para el mundo de la programación en aplicativos móviles tenemos android studio con las herramientas de codificación construcción y pruebas del aplicativo tanto con un Smartphone o virtualmente en la misma computadora también, eclipse que necesita más librerías para alcanzar a android studio, de ahí las hibridas que son Xamarin form, Ionic, phongap y muchos más para dar un ejemplo los 3 más conocidos y uno nuevo como muestra la Tabla *6*.

Tabla 6. *IDES de desarrollo*

Herramienta	características
Android studio	La más comercial por la cantidad de herramientas y libre para el desarrollo de aplicativos
Xamarin	Xamarin otra buena herramienta para desarrollo de aplicaciones, pero hibridad para dos sistemas operativos
Titanium Appacelerator	Otro ide de desarrollo hibrido que genera un solo aplicativo para dos sistemas operativos
Eclipse	Ide de desarrollo nativo de android studio y netamente java

Fuente: Propia

Se opta por la más comercial. Por los conocimientos básicos y la cantidad de material de respaldo para su mejor desarrollo y la cantidad de herramientas que posee es el motivo principal para la elección del ide de desarrollo.

2.3.21. Lenguajes de programación

Los lenguajes de programación son muy amplios y cada uno de ellos tiene un resalte por encima de los demás el lenguaje de desarrollo o de programación no ayuda a darle seguridad forma y funcionalidad a un proyecto, sistema, aplicativo, etc. La mayoría de los lenguajes son pesados y más complicadas para uno java en la actualidad hay mucha demanda de desarrolladores, pero es complejo el procedimiento de crear funciones, modificar reportes, eliminar por otro lado hay lenguajes de programación fáciles y rápidos de seguir la lógica como Python que es un lenguaje que puede uno en menos de 10 min levantar una plataforma con funciones, el famoso crud en tampoco tiempo. Hay lenguajes de programación más complejos como c, c# y o c++ entonces se tiene una gran gama de lenguajes de programación para ello se da los 4 más usado como muestra la Tabla 7.

Tabla 7. Lenguajes de programación

Herramienta	características
Java	Lenguaje de programación más usado y engorroso para la programación de aplicativos
Python	Lenguaje más fácil y fácil de dar funciones añadir estilos.
C#	Lenguaje pesado para el desarrollo de aplicaciones y usado para Aplicaciones hibridas
JavaScript	Lenguaje derivado del mimo java con la diferencia de que es más fácil desarrollo de aplicaciones con JavaScript
Dart	Lenguaje ordenado pero flexible para programación y es el más misterioso de los últimos.
Kotlin	Otro de los lenguajes nuevos es Kotlin y es 100% interoperable con java.

Fuente: Propia

La mejor opción para su uso es que se le dará es android estudio ya sea pesado engorroso al trabajarlo bien con sus estilos funciones actividades alertas y reportes resulta ser la mejor opción por el momento como el proyecto necesita tener más madurez puede a lo largo de la construcción que se opte por una nueva herramienta hibrida.

2.3.22. Bases de datos

Las bases de datos o gestores son los encargados de almacenar unas cantidades inmensa de datos lo cuales son necesarios siempre en un sistema en un aplicativo o para solo almacenar datos entonces las bases de datos (DB) es de mucha importancia saber que cuentan con consultas, modificaciones alteraciones eliminaciones de tablas genera reportes por nombres apellidos direcciones y muchos campos más dependiendo donde se aplicara su uso en el sistema

de seguridad habrá el control de quien ingresa o no por ello será quien sepa quién ingresa dentro de los cuales existen dos tipo relacionales y no relacionales como muestra la Tabla *8*.

Tabla 8. *Tipos de bases de datos*

Herramienta	características
Bases de datos relaciona	Las bases de datos relacional son las más conocidas la que solo son para pequeños sistemas
Bases de datos no relacional	Las bases de datos no relación son las que son las mejores ya que soportan cargas grandes y no afecta al sistema

Fuente: Propia

Se usará la base de datos no relación por el tiempo de respuesta y para posteriormente aplicar big data al generarse grandes cantidades de datos y concurrencia de usuarios.

2.3.22.1. *Firebase*

Firebase unas una base de datos no relacional entre ellos tiene muchos servicios como respuestas en tiempo real y para el uso que se pretende dar se adapta pero teniendo en cuenta que una vez que el sistema crezca los datos guardar serán grandes y puede ser que pierda la seguridad en la cual se da por mejores opciones Firebase como tal nos ayudara a tener un control de datos minuciosamente de quien entre y quien sale entonces con ello se llevara el control más estricto en la seguridad de accesos también mencionar que podremos implementar módulos de mensajería para la comunicación en tiempo real.

Figura 7. Firebase y la conectividad con dispositivos.

Fuente: Firebase recuperado de http://programaenlinea.net/firebase-plataforma-desarrollo-google/, mejorado por mi persona.

2.3.23. Estándares

2.3.23.1. *Contexto de los estándares*

"Los autores Salazar (2014) del libro tech pedia de versión prueba en la página 28 menciona acerca de los estándares que se manejan en las IoT Los dispositivos IoT son muy diversos y miden diferentes parámetros y con diferentes convenciones y unidades de medida. Aunque los protocolos en propiedad siguen compitiendo, es probable que los estándares de código abierto serán una de las formas de obtener estos datos para interpolar. Claramente, los estándares abiertos son la herramienta clave para el éxito de las tecnologías de comunicación inalámbrica y, en general, para cualquier tipo de comunicación de máquina a máquina. Sin embargo, se ha reconocido como un elemento importante para el despliegue de aplicaciones IoT la necesidad de una configuración más rápida de las normas de interoperabilidad. Es necesario aclarar los requisitos para una identificación global única, denominación y DNS. Un reto que debe abordarse en el futuro es la falta de convergencia en la definición de modelos de referencia comunes, arquitecturas de referencia para las futuras redes, la Internet del Futuro y la IoT, y la integración de sistemas y redes heredadas.

CAPITULO III. Materiales y Métodos

3.1. Descripción del lugar de ejecución

El lugar donde se realizará el proyecto será en el distrito de San Miguel provincia de San Román del departamento de Puno. Será en la vivienda de la familia Ccamerccoa donde cuenta con 1 puertas externa y 3 puertas interiores y sera instalada en uno de los dormitorios será donde se instalare para realizar las pruebas respectivas. Ya que el dormitorio cuenta con solo con cerraduras con llave de esta manera evaluaremos este punto para determinar a comparación de los demás cuartos y la puerta principal. y se veremos los beneficios juntamente con la gestión de accesos que se manejará a comparación con las demás.

3.2. Materiales

Tabla 9. *Herramientas de desarrollo del proyecto (Hardware).*

Materiales (Hardware)	**Características**
CPU	• Procesador i7. • Memoria RAM 8GB. • Disco duro 1TB • Tarjeta gráfica Nvidia 1660 súper
Laptop Lenovo	• Procesador i7 • Memoria RAM 16GB • Disco duro 1TB
Samsung A50	• Android versión 10.0 Q.

	• 64 GB memoria interno. • Seguridad Knox versión 3.5 • Knox api nivel 31
Huawei p30 pro	• Emui versión 10.0.0, Android versión 10.0 Q. • 256 GB de almacenamiento interno
Samsung Galaxy s9 plus	• Android versión 10.0 Q. • 64 GB memoria interno. • Seguridad Knox version3.4.1 • Knox api nivel 30
Arduino uno R3	• Rango 5 a 12 voltios • Microcontrolador Atmega328 • Memoria flash de 32KB (0.5KB para el bootloader) • SRAM 2kb • 6 entradas analógicas • 14 pines digitales
Modulo bluetooth	• Rango 3.3v a 5 voltios • Chip BC417143

	• Velocidad de transmisión de transmisión 1200bps
Cerradura electrónica Yale	• Rango con 12 voltios con conversor de 12 a 220voltios
Transformador de corriente	• Converso de energía de 220 a 12 voltios
Módulo relay	• 5 voltios DC
Cables	• Conectores elaborados específicamente para las conexiones

Fuente: Elaboración Propia.

Tabla 10. *Herramientas utilizadas en el proyecto (Software).*

Herramientas (Software)	Versión	Valor
IDE Android Studio	v.3.6.3	Desarrollado por Google Licencia Apache 2.0.
Samsung A50	Exynos 9610	Emulador.
SDK	v.29.0.0	Herramienta de uso por default para desarrollo de aplicación móviles.
Equipo móvil	v.9.0	Equipo físico disponible en modo desarrollador.
Plataforma versión	soporte API 18 Android 7.0 en adelante.	Recomendado mínimo Api 28 en adelante.
Librerías Externas	Actualizada 2020	Librerías con actualización
Kit de desarrollo Java (JDK8)	Java SE 8	Compatibilidad en el desarrollo.
Android SDK Tools	29.0.3	Licencia abierta para desarrollo móvil nativo.
Java	v.8	Código Abierto
Firebase (Authentication)	Actualizada 2020	Licencia gratuita limitada
Github	Actualizada 2020	Licencia gratuita ilimitada
Excel, World y Power point.	2016	Licencia Terceros.
Mendeley	2019	Open Source
Lucidchart		Licencia gratuita limitada
Adobe XD	30.0.12.14	Licencia Estudiante
Coreldraw	2020	Licencia gratuita limitada 30 días
Arduino IDE	2007	Open Source
Gif Animator	1.0.0.1.1	Open Source

Fuente: Elaboración Propia.

Las herramientas seleccionadas se hicieron de uso por la gran diversidad de desarrollo y por ser herramientas open source y por variedad de herramientas que posee

Herramientas (Software)	Versión	Valor
IDE Android Studio	v.3.6.3	Desarrollado por Google Licencia Apache 2.0.
Samsung A50	Exynos 9610	Emulador.
SDK	v.29.0.0	Herramienta de uso por default para desarrollo de aplicación móviles.
Equipo móvil	v.9.0	Equipo físico disponible en modo desarrollador.
Plataforma versión	soporte API 18 Android 7.0 en adelante.	Recomendado mínimo Api 28 en adelante.
Librerías Externas	Actualizada 2020	Librerías con actualización
Kit de desarrollo Java (JDK8)	Java SE 8	Compatibilidad en el desarrollo.
Android SDK Tools	29.0.3	Licencia abierta para desarrollo móvil nativo.
Java	v.8	Código Abierto
Firebase (Authentication)	Actualizada 2020	Licencia gratuita limitada
Github	Actualizada 2020	Licencia gratuita ilimitada
Excel, World y Power point.	2016	Licencia Terceros.
Mendeley	2019	Open Source
Lucidchart		Licencia gratuita limitada
Adobe XD	30.0.12.14	Licencia Estudiante
Coreldraw	2020	Licencia gratuita limitada 30 días

Arduino IDE	2007	Open Source
Gif Animator	1.0.0.1.1	Open Source

Tabla *10*.

3.3. Metodología

3.3.1. Tipo de Investigación

La investigación propositiva parte de la estructura fáctica donde se requiere de 3 componentes básicos y esto a la vez lo vemos de la siguiente manera tiene la unión de las teorías existentes que están relacionados al hecho particular y al hecho singular que es materia de mi investigación pero que no concluye solamente comparando o relacionando la teoría con el hecho, sino que va más allá se pretende dar una solución pretende dar una iniciativa a la cual se le denomina propuesta en lo que resume es que la investigación propositiva es aquella que va a ser la mescla de las teorías existentes sobre un hecho particular identificado para que se pueda desarrollar una propuesta para que se evalué y en su mejor caso los hogares lo pueden implementar y no debe ser cualquier propuesta

Teoría + Hecho + Solución = Investigación Propositiva

3.3.2. Arquitectura de Investigación

3.3.2.1. *Prueba aplicativa funcional*

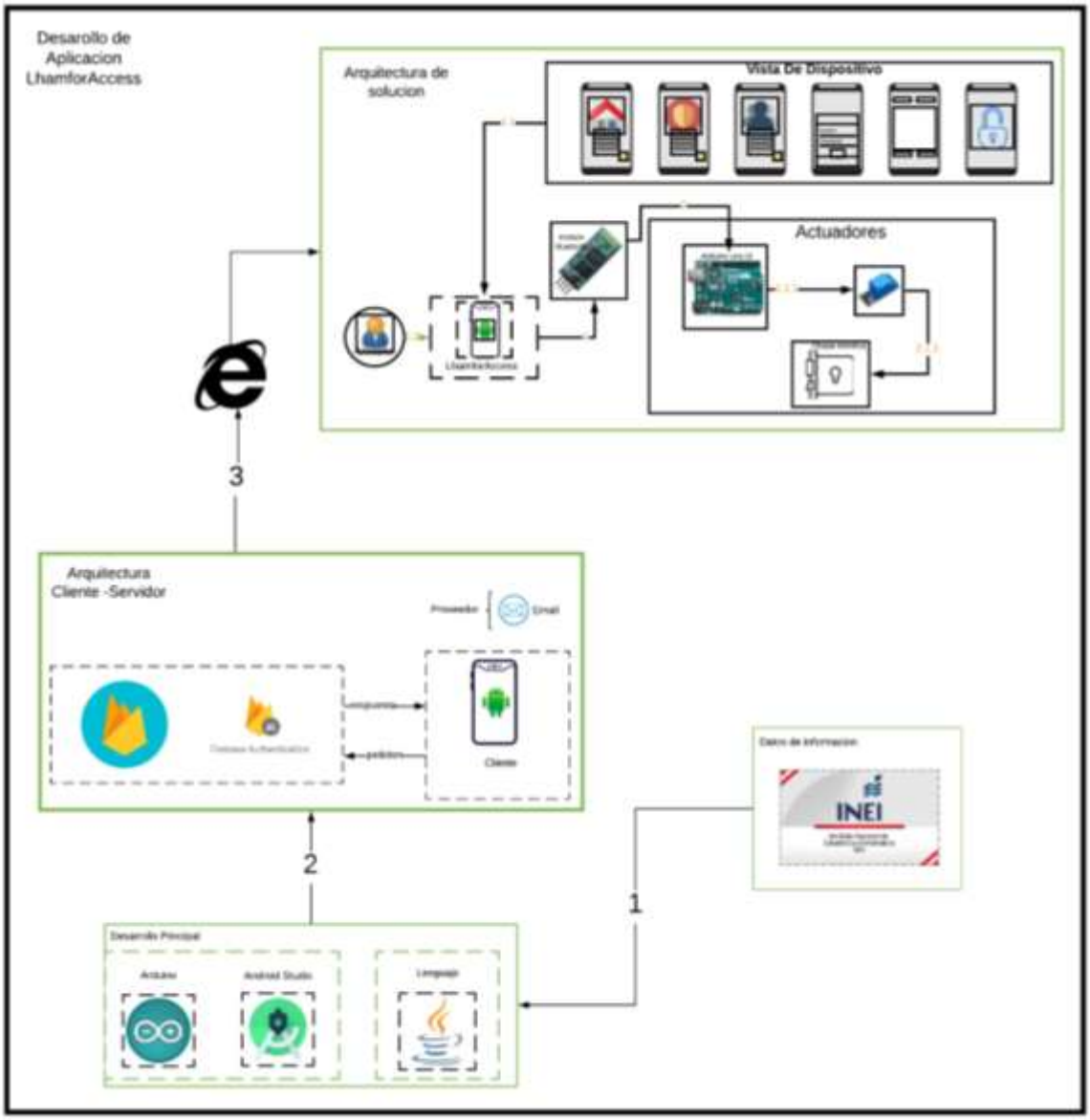


Figura 8. Arquitectura de Investigación.

Fuente: Elaboración Propia.

3.3.2.2. *Arquitectura de Solución*

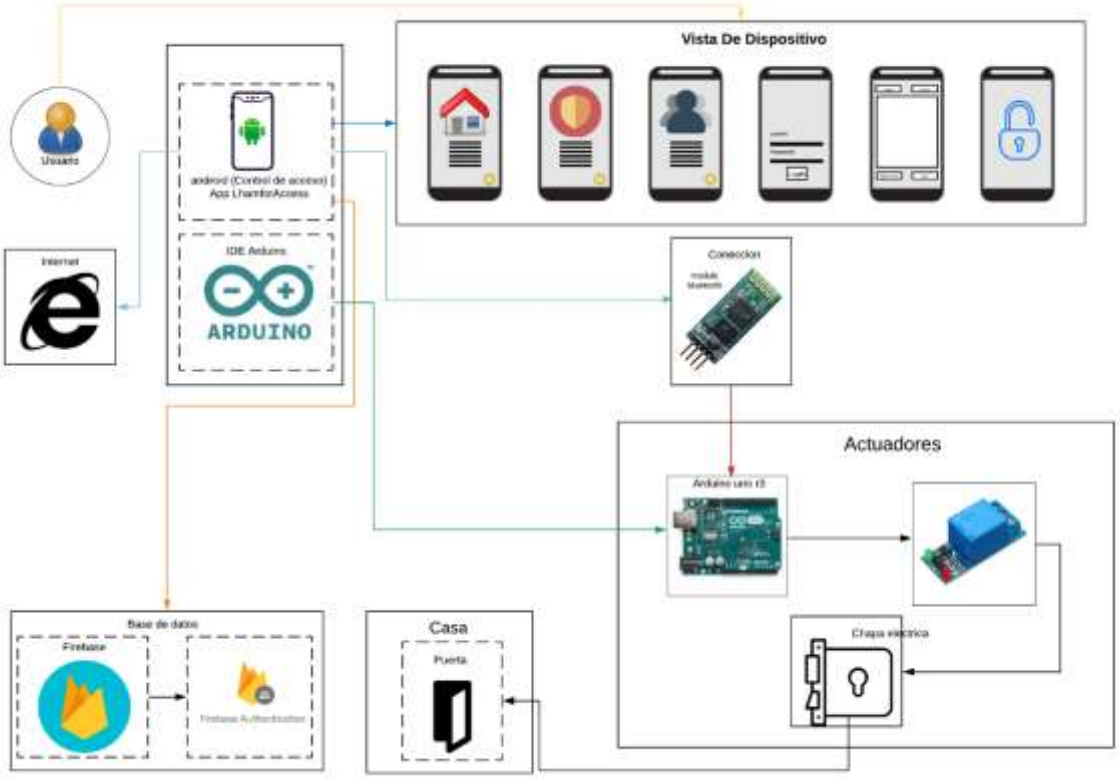


Figura 9. Arquitectura de Solución

Fuente: Elaboración Propia.

3.3.3. Metodología de la Investigación (Aplicación)

Se hizo de uso la metodología Mobile-D en cada una de sus fases para satisfacer los requerimientos de los usuarios en un periodo reducido, para incrementar la seguridad de la vivienda donde se desarrolló la investigación, con la metodología pretendemos desarrollar ciclos muy rápidos en equipos reducidos según guerrero (2015). Para luego obtener una herramienta rápida segura e intuitiva de usar, y no requiere de una gran cantidad de documentación solo lo necesario para el desarrollo de la aplicación, como principal documentación cuenta con los

storyboards y storycards y con respecto a las pruebas puede verse en pruebas de interfaz y pruebas funcionales

Figura 10. Mobile-D y sus fases

Fuente: Velandia (2014)

- **Exploración:** en esta fase se encarga de la parte planificativa e identificación de los requisitos del proyecto donde se tendrá la visión completa del alcance del proyecto también todas las funcionalidades

- **Inicialización:** la fase de inicialización es la satisfacción en la próxima fase del proyecto que divide en 4 etapas, la puesta en marcha del proyecto, la planificación inicial, el día de la prueba y día de salida.

- **Producción:** en la fase de producción, se debe implementar la programación de los tres días, interactivamente hasta montar la aplicación e implementar las funcionalidades que se desea obtener. Aquí requeriremos el desarrollo dirigido por pruebas Test-Driven-Development, para la verificación del correcto funcionamiento

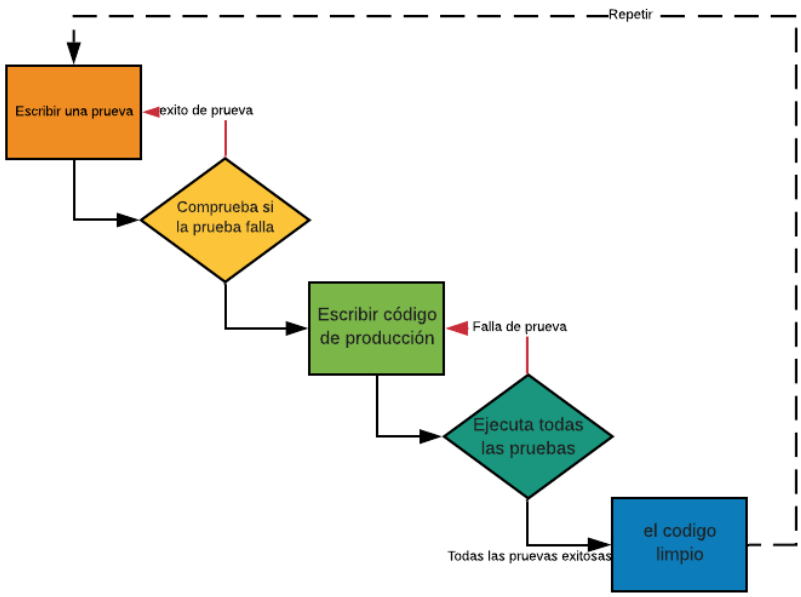


Figura 11. Test-Driven Development modelo de Pruebas de software

Fuente: (guerrero, 2015), mejorado por mi persona

- **Estabilización:** En esta fase se dará las últimas acciones de la integración y verificar el funcionamiento completo del sistema. en toda la metodología esta fase actual es la más importante de todas porque nos ayuda a asegurar la estabilidad del desarrollo, se puede tomar en cuenta incluir en esta fase la parte de producción de documentos.

- **Pruebas:** En esta fase por si es de cierre por que en esta etapa se hace las pruebas de la aplicación una vez culminada por la cual se debe hacer todas la pruebas posibles y necesarias para tener una versión estable y final, en esta etapa si existiera algún error deberá darse solución en el acto, pero ojo deberemos tener en cuenta que no deberemos reconstruir todo el aplicativo desde cero si no solucionar dentro de lo desarrollado ya que esto significaría romper el ciclo (guerrero, 2015).

3.3.4. Metodología de la Investigación (Hardware)

En la metodología waterfall se desempeña en forma cascada, ya que cada tarea que se realice a lo largo del proyecto respete el orden de primero a ultimo y no como las metodologías agiles que toman actividades paralelas a las demás, pero tiene una facilidad que puede ayudarse de metodologías agiles para cubrir proyectos que no esten a su capacidad original de esta manera se vuelve más robusto esta metodologia entre ella tenemos:

- Flexibilidad ante los posibles cambios en la planificación
- Colaboración entre los diferentes interesados de todos los proyectos
- Gestión de recursos
- Gestión de la disponibilidad del equipo en todos los proyectos en los que cada persona participe
- Imputación de horas y costes
- Seguimiento a tiempo real del avance del proyecto

Con esas características sumadas a la metodologia llega a tener un mayor peso y una facilidad de aplicación proyectos (Sinnaps, 2019).

- **Planificación del proyecto:** En esta fase inicial del proyecto se verá el planeamiento del proyecto a nivel del hardware, donde decidimos a que es lo que se quiere conseguir y al mismo tiempo hasta donde se debe desarrollar el proyecto

- **Requerimientos:** Definiremos puntualmente lo que se requerirá para la parte hardware del proyecto lo cuáles serán los materiales especificados en la página (p.50-51) pero en especial los requerimientos en la metodologia serán basados más a la parte de funcionalidad, en lo que el sistema hardware realizar.

- **Análisis:** En esta fase se analizará todas las partes involucradas en el desarrollo de hardware, de cómo la seguridad actual y la moderna y analizaremos también los puntos fuertes del proyecto y lograr tener algo bien definido para pasar a la siguiente fase.

- **Diseño:** En esta fase pasaremos al diseño de la arquitectura a nivel de hardware ya que esta será nuestra automatización y el diseño es muy importante para la codificación y el sistema sea completamente funcional.

- **Codificar:** La fase de codificación es la parte donde codificaremos a la placa Arduino dándole el funcionamiento del sistema diseñado en la fase anterior.

- **Pruebas:** La parte de pruebas del sistema la cual podríamos incorpora la evaluación mediante (TDD) la cual nos ayudaría mucho para garantizar el buen funcionamiento del sistema a un 80% o 90% de certidumbre de que el proyecto cumpla con lo requerido.

- **Liberación**: Parte final de liberación del proyecto donde ya se hizo la prueba se hizo el diseño la codificación para su próxima liberación final del proyecto donde veremos su funcionamiento final del producto.

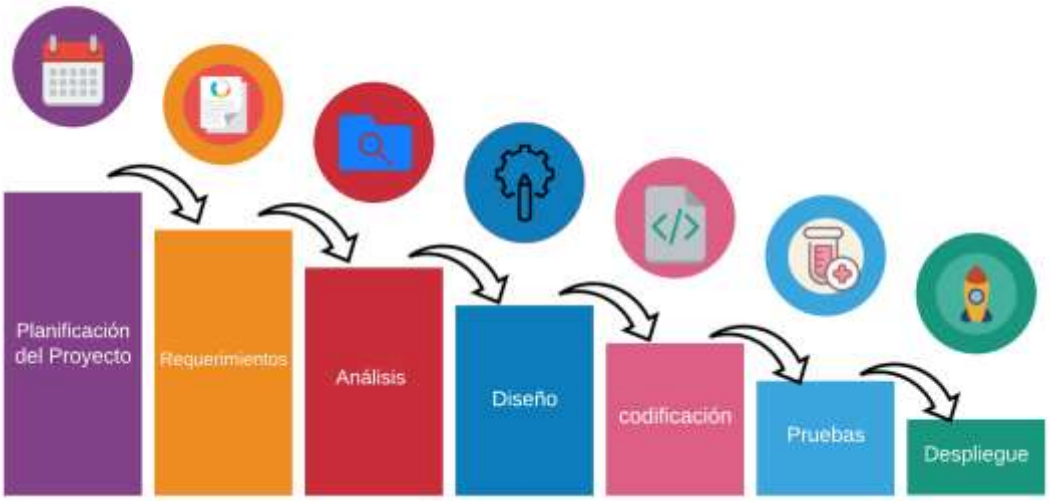


Figura 12. Modelo cascada de metodología de waterfall

Fuente: https://vmsportfolio.blogspot.com/2018/07/metodologia-agile-los-problemas-que.html, mejorado por mi persona

CAPITULO IV. Proceso de Desarrollo-software

En la investigación se hizo de uso la metodología Mobile-D que es de uso para desarrollo de aplicaciones razón por la cual se desarrolló una estructura personalizada al de la original y también en la parte del plan de trabajo.

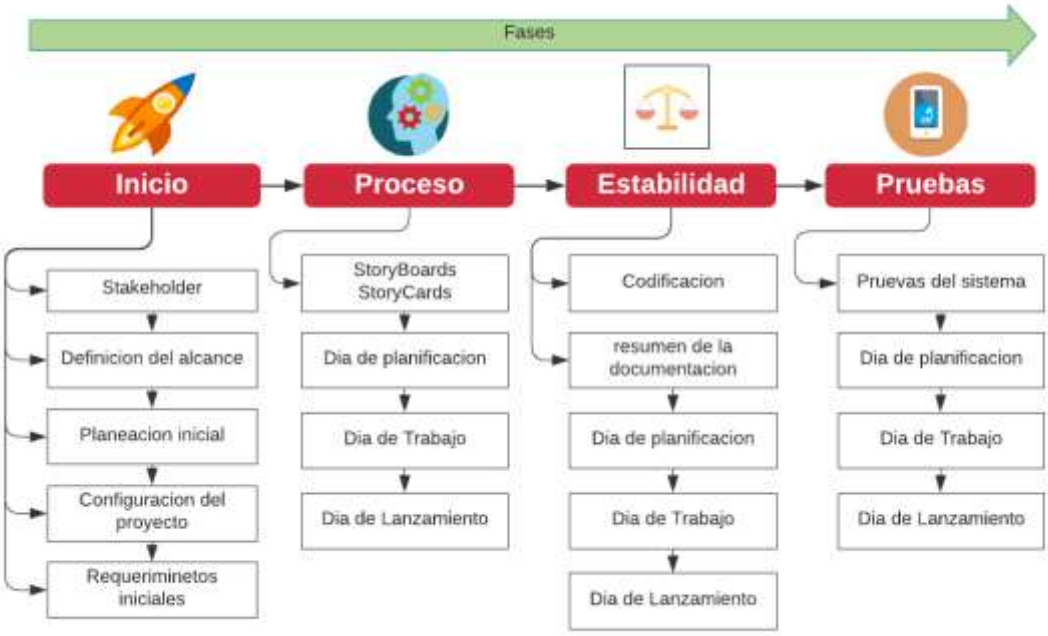


Figura 13. Personalizado Metodología Mobile-D

Fuente: Elaboración Propia.

En la *Figura 13* podemos apreciar que la fase de exploración e iniciación para reducir la fase

4.1. Primera Fase: Inicio

4.1.1. Recopilación de información.

Se indago en revistas locales como en las estadísticas del instituto nacional de estadísticas e informática arrojando datos que sorprenden acerca de la delincuencia que dentro de los años 2010 hasta el 2015 hubo un descenso en los asaltos a mano armada a viviendas en Juliaca pero desde el año 2016 hasta el 2017 incremento esta suma dando a lugar una preocupación los moradores de Juliaca, identifica también que los robos a casa se sitúa en la segunda posición con 64.80%, vemos también esto en viviendas con vigilancia en el Perú son también afectadas viendo estos datos es que se concluye que hay un problema que está afectando a la ciudad de Juliaca, mediante esta información se dio inicio al desarrollo de la aplicación.

4.1.2. Establecimiento de Stakeholders

definió a los siguientes involucrados.

- Líder del Proyecto: encargado de solucionar el problema identificado, para este caso el investigador.

- Usuario: Personas que habitan en la vivienda con dirección en el jr. luna s/n Barr. buenos aires M.ª D lt.22, donde se realizará el proyecto.

La idea nació de la problemática existente de hurtos a casas la cual lleva por nombre LhamforAccess como prueba para incrementar la seguridad de las viviendas.

4.1.3. Definición del alcance

Lo que pretendemos abarcar es tener un control de acceso para incrementar la seguridad en la vivienda y de esta manera restringir a los usuarios no registrados.

4.1.3.1. *Dependencias*

Los requisitos a tener en cuenta son:

- Manejo base de dispositivos móviles.
- Aplicación intuitiva.
- señal móvil o wifi.
- Idioma único español.
- El dispositivo móvil debe contar mínimo con huella digital o reconocimiento facial.
- Aplicación soporta las versiones de android 7 lollipop hasta la versión android 9 Pie.
- Librerías de android studio **Anexo B**.

4.1.3.2. *Restricciones*

Las limitaciones encontradas son

- La aplicación registrara y se autentificara solo con internet.
- La aplicación solo podrá ser instalada en dispositivos android.
- La aplicación requerirá un administrador para eliminar a los usuarios que ya no quieran que tenga acceso.

- El usuario no podrá abrir una puerta si no se encuentra a 2 metros de distancia.

- La aplicación no funcionara correctamente si no tiene lector de huella.

4.1.4. Inicio de la planeación

Se desarrollará los pre-requisitos necesarios para el proyecto, como la arquitectura de solución mostrado en la *Figura 9*.

- El nombre de la aplicación es LhamforAccess.

- debe contarse con un Android versión 5.0 API 21 como mínimo y como máximo android 9 api 29.

El escenario se tiene de la siguiente forma:

- el usuario debe contar con un dispositivo móvil con sistema operativo android y que sea usuario de la aplicación.

- El usuario luego de autenticarse con su correo y su password deberá autenticarse como método de seguridad con su huella o reconocimiento fácil de su dispositivo

- Debe contar con internet al momento de su logue y registro de usuario ya que la red que maneja lo resto es una local.

- Crea una alerta siempre y cuando no tenga internet al momento del logue.

4.1.5. Configuración del proyecto

Aquí se dio la configuración de las siguientes actividades de ambiente de desarrollo para aplicaciones Android los cuales se menciona a continuación.

- Instalación de kit de desarrollo jdk 8.1.
- Instalación del IDE Android Studio y SDK por default.
- Instalación de dependencias Requeridas.
- Instalación del emulador Canary y equipo móvil físico Capacitación en los cursos de desarrollo Java para Android.

4.1.6. Requerimientos

4.1.6.1. *Requerimiento Funcional*

Tabla 11. *Requerimiento funcional de la aplicación*

Código	**Función**	**Descripción**
RF001	Acceso de Usuario.	El usuario estará Autenticado siempre y cuando se le dé un usuario master para crear nuevos usuarios en Firebase con conexión a internet.
RF002	Registro de huella en el teléfono con pin.	Debe registrar datos biométricos en el teléfono del usuario

RF003	Registro de reconocimiento facial con pin.	El usuario de contar con esta opción deberá registrar en su teléfono.
RF004	Redes bluetooth.	Usuario debe poder identificar y ver las redes bluetooth de su domicilio y conectarse para realizar su función apertura la puerta posteriormente.
RF005	Añadir de usuario	El usuario fijara mediante la aplicación respuestas de alertas en puntos críticos clave.
RF006	Cerrar sesión Usuario	El usuario debe poder cerrar sesión una vez terminado la actividad de abrir y cerra su domicilio.
RF007	Información	El aplicativo debe contar con una pequeña introducción de que va el aplicativo.

Fuente: Elaboración propia

4.1.6.2. *Requerimientos no Funcionales*

Tabla 12. *Requerimiento no funcional de la aplicación.*

Código	**Función**	**Descripción**

RNF001	logueo	Tiempo de respuesta 3s en espera.
RNF002	registro	Tiempo de respuesta 3s en espera
RNF003	Seguridad.	Será de dos pasos uno con correo y password y el segundo paso con huella previamente registrada en el dispositivo móvil
RNF004	Disponibilidad	• Estará disponible solo en datos y/o Wifi. • Por el momento no estará subido al Google play store. Por qué está en su fase de desarrollo y la prueba de efectividad en el problema
RNF005	Sistema multiusuario	Sera de uso por todos los usuarios registrados en la apk.

Fuente: Elaboración propia

4.2. Segunda Fase: Proceso

En esta fase a continuación de elaborar un esquema de uso diagrama de navegación comúnmente llamado Storyboards del usuario, y dentro de esto esta los Storycards y los Taskcards para mostrar la escena con el día de trabajo y el deployment

4.2.1. Test-Driven-Development

4.2.1.1. *Esquema de Navegación*

De desarrollo con el propósito de que el usuario pueda familiarizarse con el aplicativo como muestra en la *Figura 14* .

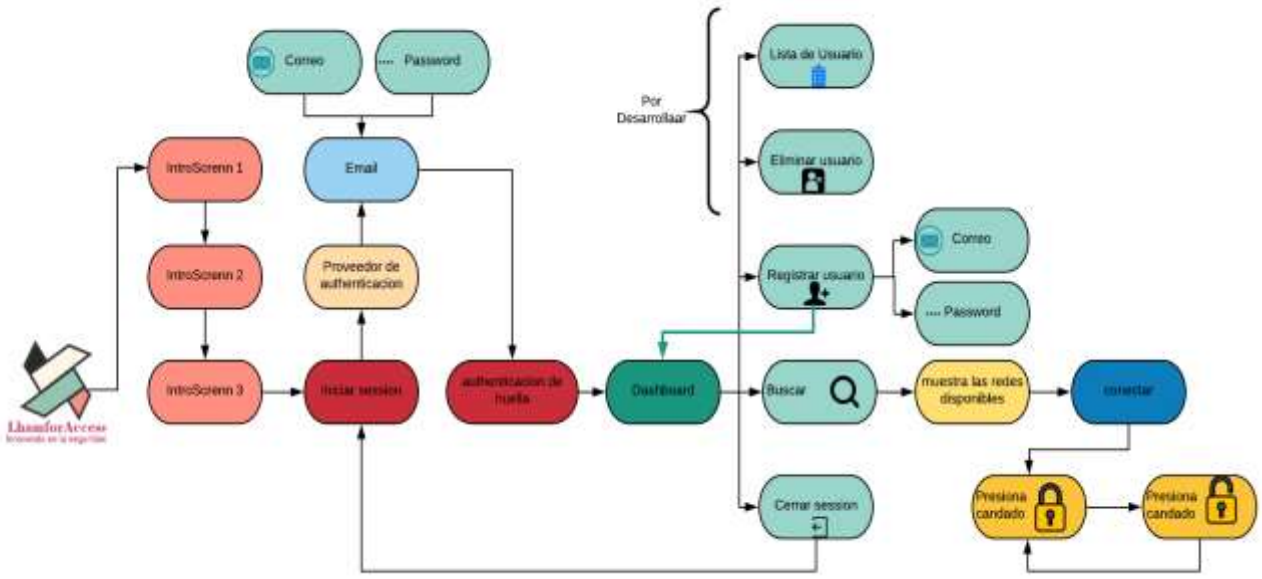


Figura 14. Storyboards o Esquema de Navegación.

Fuente: Elaboración propia.

4.2.1.2. *Diseño de Prototipos*

Partiendo como punto de inicio de los requerimientos funcionales se determinó los prototipos en base al esquema de navegación para luego ajustar los storycards y así determinar el desarrollo como en el esquema mostrada en la *Figura 14*. Para el desarrollo

de los prototipos hubo muchas opciones, pero por su sencillez y gran facilidad de uso y su magnífica presentación se hizo de uso de la herramienta adobe Xd.

Figura 15. Herramienta de prototipo

Fuente: Elaboración propia

- Pantalla Introducción

  Los tres mockups son la parte inicial del aplicativo donde da detalles de que como funciona la aplicación y esto solo se mostrará por primera vez al ser instalado mas no se repartirá por segunda vez parte de la codificación se muestra en el **Anexo C.**

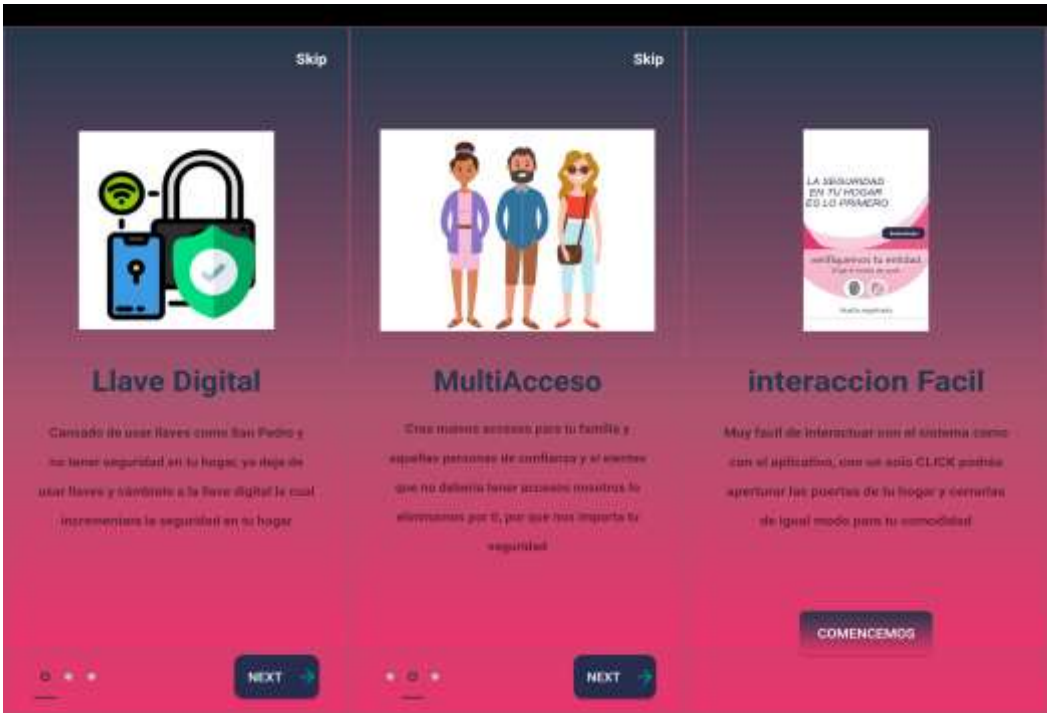


Figura 16. Pantalla de bienvenida

Fuente: Elaboración propia

- Pantalla Dialog alert

En esta pantalla se verá un mensaje de alerta siempre y cuando no cuente con internet una vez activado la red este mensaje desaparecerá refrescando en el botón inténtelo de nuevo y para hacerlo más llamativo tendrá una pequeña animación y la parte de codificación se muestra en el **Anexo D**.

Figura 17. Alerta de no tener internet

Fuente: Elaboración propia

- Pantalla Iniciar Sesión

La pantalla principal donde se le proporcionara un usuario master para asi poder ingresar a la aplicación, pero para ingresar deberá llenar el correo y el password para que de esta manera se active el botón inicio de sesión y se muestra en el

Anexo E, y vez acceda dentro de la aplicación cree nuevos usuarios.

Figura 18. Pantalla del logueo

Fuente: Elaboración propia

- Pantalla Autenticación

En esta pantalla concentra la verificación por huella o por reconocimiento faciales cuales deben registrarse previamente en el dispositivo móvil el cual brindara más seguridad y podemos ver la codificación en el

Anexo F.

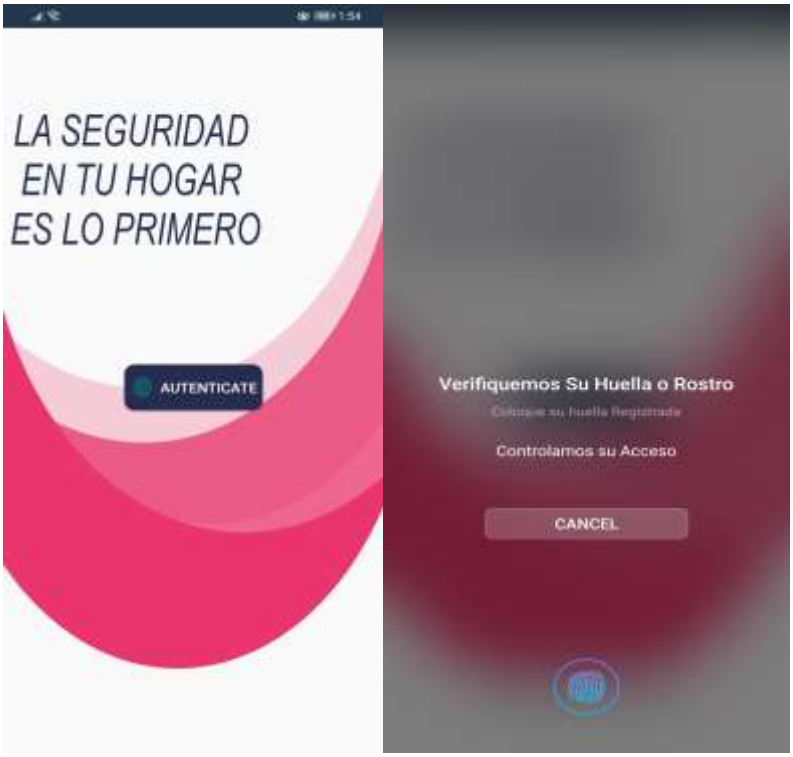


Figura 19. Autentificación por huella o por reconocimiento facial

Fuente: Elaboración propia

- Pantalla Dashboard

En la que encontraremos 4 botones uno de ellos nos redirigirá aun activity nuevo donde podremos registrar un nuevo usuario y está a la vez nos redirigirá al Dashboard como podemos ver en la *Figura 21*. Pantalla Registrar nuevo usuario *Figura 20*. Pantalla Dashboard el segundo botón nos permitirá cerrar sesión donde nos redirigirá al activity de logueo y el tercer botón llamado búsqueda nos permitirá mostrar las redes disponibles de bluetooth como se muestra en la *Figura 22*. Pantalla de búsqueda de redes, por último, el cuarto botón de conectar esperará que seleccionemos una red y pueda conectarse y nos dirigirá a la funcionalidad de abrir o cerrar la puerta como se codifico en el

Anexo G y el

Controlling.java

```
176        @Override
177        protected void onResume() {
178            if (mBTSocket == null || !mIsBluetoothConnected) {
179                new ConnectBT().execute();
180            }
181            Log.d(TAG,  msg: "Resumen");
182            super.onResume();
183        }
184
185        @Override
186        protected void onStop() {
187            Log.d(TAG,  msg: "Detener");
188            super.onStop();
189        }
190
191        @Override
192        protected void onSaveInstanceState(Bundle outState) {
193            super.onSaveInstanceState(outState);
194        }
195
196        @SuppressLint("StaticFieldLeak")
197        private class ConnectBT extends AsyncTask<Void, Void, Void> {
198            private boolean mConnectSuccessful = true;
199
```

Controlling.java

```
200         @Override
201         protected void onPreExecute() {
202
203             progressDialog = ProgressDialog.show(
204                     context: Controlling.this,  title: "Espere un momento",  message: "Conectando...");
205         }
206
207         @Override
208         protected Void doInBackground(Void... devices) {
209             try {
210                 if (mBTSocket == null || !mIsBluetoothConnected) {
211                     mBTSocket = mDevice.createInsecureRfcommSocketToServiceRecord(mDeviceUUID);
212                     BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
213                     mBTSocket.connect();
214                 }
215             } catch (IOException e) {
216                 mConnectSuccessful = false;
217             }
218             return null;
219         }
220
221         @Override
222         protected void onPostExecute(Void result) {
223             super.onPostExecute(result);
224
```

Controlling.java

```
221         @Override
222         protected void onPostExecute(Void result) {
223             super.onPostExecute(result);
224
225             if (!mConnectSuccessful) {
226                 Toast.makeText(getApplicationContext(),  text: "No se pudo conectar al dispositivo. "
227                         + "Por favor, encienda su hardware.", Toast.LENGTH_LONG).show();
228                 finish();
229             } else {
230                 msg( s: "Conectado al dispositivo");
231                 mIsBluetoothConnected = true;
232                 mReadThread = new ReadInput();
233             }
234             progressDialog.dismiss();
235         }
236
237     }
238     @Override
239     protected void onDestroy() { super.onDestroy(); }
242 }
```

Anexo H muestra el registro de usuario a nivel de codificación.

Figura 20. Pantalla Dashboard

Fuente: Elaboración propia

Figura 21. Pantalla Registrar nuevo usuario

Fuente: Elaboración propia

Figura 22. Pantalla de búsqueda de redes

Fuente: Elaboración propia

- Pantalla Funcionalidad de apertura

Dará la funcionalidad mediante la red bluetooth para la apertura la cerradura eléctrica de esta manera también cerrarla cuenta de dos vistas donde una de ellas esta con el candado cerrado y el otro con el candado abierto como muestran en la *Figura 23* y la *Figura 24*.

Figura 23. Cerradura cerrada

Fuente: Elaboración propia

Figura 24. Cerradura abierta

Fuente: Elaboración propia

4.2.1.3. *Storycards*

Tomando como punto de partida los requerimientos funcionales se determinaron los guiones de desarrollo se hizo de uso por la aceptación de la metodologia. El cual se tomará como referencia para el desarrollo

Tabla 13. *Storycards Introducción a la aplicación*

/ ID	**Tipo**	**Dificultad**	**Esfuerzo**		**Prioridad**
			Estimado	Gastado	
SC001	Nuevo	baja	5h	4h	**Baja**
Nombre:					
Introducción a la aplicación					
Descripción:					
Dará una descripción breve del sistema en general					
Éxito: **información**	• Dara detalles principales de la aplicación con animaciones.				
Fallido: **versión**	• No permitirá la animación a versiones anteriores a android 5 lollipop				

Fuente: Elaboración propia.

Tabla 14. *Storycards Iniciar Sesión.*

/ ID	**Tipo**	**Dificultad**	**Esfuerzo**		**Prioridad**
			Estimado	Gastado	
SC002	Nuevo	media	10h	8h	**alta**
Nombre:					

Inicio de sesión

Descripción:

Para la persona que adquiera la aplicación deberá solicitar un usuario master para asi poder crear nuevos usuarios para su domicilio.

Éxito: **Ingreso o logueo**	• Se le proporciona un usuario master para su logueo • Se ha registrado el usuario mediante Firebase (Authentication) mediante el proveedor email. • El botón de inicio de sesión se activará solo en caso de ser rellenados los campos de correo y password.
Fallido: **Datos no registrados o internet**	• El aplicativo no ingresara con solo tener un correo y una contraseña, debe estar registrado en Firebase Authentication. • Solo requerirá de internet en caso de creación del usuario y loguearse al sistema. • Mandara una alerta de dialogo siempre y cuando no cuente con internet y quiera accedes sin internet.

Fuente: Elaboración propia.

Tabla 15. *Storycards Autenticaron biométrico*

/ ID	**Tipo**	**Dificultad**	**Esfuerzo**		**Prioridad**
			Estimado	Gastado	
SC003	Nuevo	alta	20h	24h	**alta**

Nombre:

Autenticación biométrica

Descripción:

El usuario para ser uso del aplicativo deberá registrar su huella o rostro.

	• El usuario antes del uso de la aplicación registrara su huella seguido de un pin de respaldo.
Éxito: huella o rostro registrado	• El usuario antes del uso de la aplicación registrara su rostro como segunda opción junto al pin ya ingresado de la huella. • El usuario una vez ingresado con la contraseña se verificará su identidad con una lectura biométrica sea de rostro o huella para que le permita interactuar con el sistema.
Fallido: sin registro	• El aplicativo invalidara si el usuario no tiene ningún dato biométrico registrado previamente.

Fuente: Elaboración propia.

Tabla 16. *Storycards Registro de usuario*

/ ID	**Tipo**	**Dificultad**	**Esfuerzo**		**Prioridad**
			Estimado	Gastado	
SC004	Nuevo	media	10h	7h	**Media**
Nombre:					
Registro de usuario					
Descripción:					
Una vez obtenido el usuario master este podrá crear nuevos usuarios en el sistema para su actividad con el sistema individualmente					
	• El sistema debe permitir el registro de un nuevo usuario.				
Éxito: Registro de usuario	• El usuario debe contener mínimo las siguientes características una mayúscula una minúscula seguido del carácter @ y también debe tener un .com al final para ser válido.				

	• El password debe contener las siguientes características un numero una mayúscula una minúscula y mínimo debe tener una cantidad mayor al de 6 dígitos. • El sistema no validara ningún correo de no tener internet activado. • El sistema no registrara correo ni contraseñas que no tengan las características ya mencionadas.
Fallido: **sin registro**	

Fuente: Elaboración propia.

Tabla 17. *Storycards de buscar y conectar a una red bluetooth*

/ ID	**Tipo**	**Dificultad**	**Esfuerzo**		**Prioridad**
			Estimado	Gastado	
SC005	Nuevo	media	15h	15h	**Alta**
Nombre: **Buscar y conectar a aun red bluetooth**					
Descripción: **Buscará una red bluetooth para poder conectarse sabiendo de antemano la contraseña de este dispositivo.**					
Éxito: **Buscar y conectar**			• El usuario deberá buscar la red bluethoot que le pertenezca para asi ingresar a su domicilio. • La conexión deberá tener máximo un rango de 2 metros de distancia.		

Fallido: conectar	• Si el propietario quisiera abrir su domicilio mayor un a distancia de dos metros el dispositivo bluetooth no encontrará su red y no podrá abrir su puerta.

Fuente: Elaboración propia.

4.2.1.4. *Task Card*

Tomando como punto de partida los requerimientos funcionales se determinaron los guiones de desarrollo se hizo de uso por la aceptación de la metodologia. El cual se tomará como referencia para el desarrollo.

Tabla 18. *Storycards 002- TaskCard 001. implementar Firebase Auth.*

/ ID	**Tipo**	**Dificultad**		**Esfuerzo**		
		Antes	Después	Estimado	Gastado	**Confianza**
SC002 TC001	–Nuevo	medio	fácil	8 h.	5 h.	Mucha confianza (4)
Descripción						
Configuración de Firebase en el proyecto						
Fecha		Tipo de Estado		Comentarios		
20/08/2019		Día de Planificación		Se inicio de acuerdo al cronograma previamente elaborado		
20/08/2019		Día de Trabajo		Se desarrollo la parte de configuración del proyecto		
20/08/2019		Dia de lanzamiento		No hubo errores al momento de iniciar la Coneccion de Firebase y android studio		

Fuente: Elaboración propia

Tabla 19**.** *Storycards 003- TaskCard 002. Datos biométricos*

/ ID	**Tipo**	**Dificultad**		**Esfuerzo**		
		Antes	Después	Estimado	Gastado	**Confianza**
SC003	**–**Nuevo	medio	alta	8 h.	12 h.	Mucha confianza (4)
TC002						
Descripción						
Registro de datos biométricos						
Fecha		Tipo de Estado		Comentarios		
17/08/2019		Día de Planificación		Se inicio de acuerdo al cronograma previamente elaborado		
17/08/2019		Día de Trabajo		Se desarrollo de que en la pantalla se logre ver la lectura de huella		
19/08/2019		Dia de lanzamiento		No hubo errores al momento de envió de datos peri si de versión de SDK, pero se solucionó a tiempo		

Fuente: Elaboración propia

Tabla 20. *Storycards 004- TaskCard 003. implementar Registro de usuario cierre de sesión.*

/ ID	**Tipo**	**Dificultad**		**Esfuerzo**		
		Antes	Después	Estimado	Gastado	**Confianza**
SC004	**–**Nuevo	medio	medio	8 h.	12 h.	Media confianza (2)
TC003						
Descripción						
Registro de usuarios por la base de datos solo usuario master						
Fecha		Tipo de Estado		Comentarios		
21/08/2019		Día de Planificación		Se inicio de acuerdo al cronograma previamente elaborado		

21/08/2019	Día de Trabajo	Se desarrollo el registro de usuarios y el de cerrar sesión
23/08/2019	Dia de lanzamiento	No hubo errores al momento de envió de datos

Fuente: Elaboración propia

Tabla 21. *Storycards 004- TaskCard 004 conexión del bluetooth.*

/ ID	**Tipo**	**Dificultad**		**Esfuerzo**		
		Antes	Después	Estimado	Gastado	**Confianza**
SC005	**–**Nuevo	difícil	medio	10 h.	9 h.	Media confianza (2)
TC004						
Descripción						
Coneccion de bluethoot						
Fecha		Tipo de Estado		Comentarios		
26/08/2019		Día de Planificación		Se inicio de acuerdo al cronograma previamente elaborado		
26/08/2019		Día de Trabajo		Se desarrollo la búsqueda y activación de la red bluetooth más la conexión a la red		
27/08/2019		Dia de lanzamiento		No hubo errores al momento de la Coneccion excepto por aquellos dispositivos que no son hardware		

Fuente: Elaboración propia

4.3. Tercera Fase: Estabilidad.

Ya llegados a esta fase se integrará la parte funcional, se hizo las mejoras como también el arreglo de los errores para posteriormente mejorar la funcionalidad de la aplicación esta fase apoyara a mantener un equilibrio en la aplicación y los errores sean

mínimos e imperceptibles para el usuario y con esto se consigue la mejor a futuro. Nuestro principal guía fue los Storycards para su mejor desempeño se hizo necesario de las herramientas que se ve en la

Herramientas (Software)	Versión	Valor
IDE Android Studio	v.3.6.3	Desarrollado por Google Licencia Apache 2.0.
Samsung A50	Exynos 9610	Emulador.
SDK	v.29.0.0	Herramienta de uso por default para desarrollo de aplicación móviles.
Equipo móvil	v.9.0	Equipo físico disponible en modo desarrollador.
Plataforma versión	soporte API 18 Android 7.0 en adelante.	Recomendado mínimo Api 28 en adelante.
Librerías Externas	Actualizada 2020	Librerías con actualización
Kit de desarrollo Java (JDK8)	Java SE 8	Compatibilidad en el desarrollo.
Android SDK Tools	29.0.3	Licencia abierta para desarrollo móvil nativo.
Java	v.8	Código Abierto
Firebase (Authentication)	Actualizada 2020	Licencia gratuita limitada
Github	Actualizada 2020	Licencia gratuita ilimitada
Excel, World y Power point.	2016	Licencia Terceros.
Mendeley	2019	Open Source
Lucidchart		Licencia gratuita limitada
Adobe XD	30.0.12.14	Licencia Estudiante

Coreldraw	2020	Licencia gratuita limitada 30 días
Arduino IDE	2007	Open Source
Gif Animator	1.0.0.1.1	Open Source

Tabla *10*. Herramientas utilizadas en el proyecto (Software).

4.3.1. Estructura de las clases java para su desarrollo.

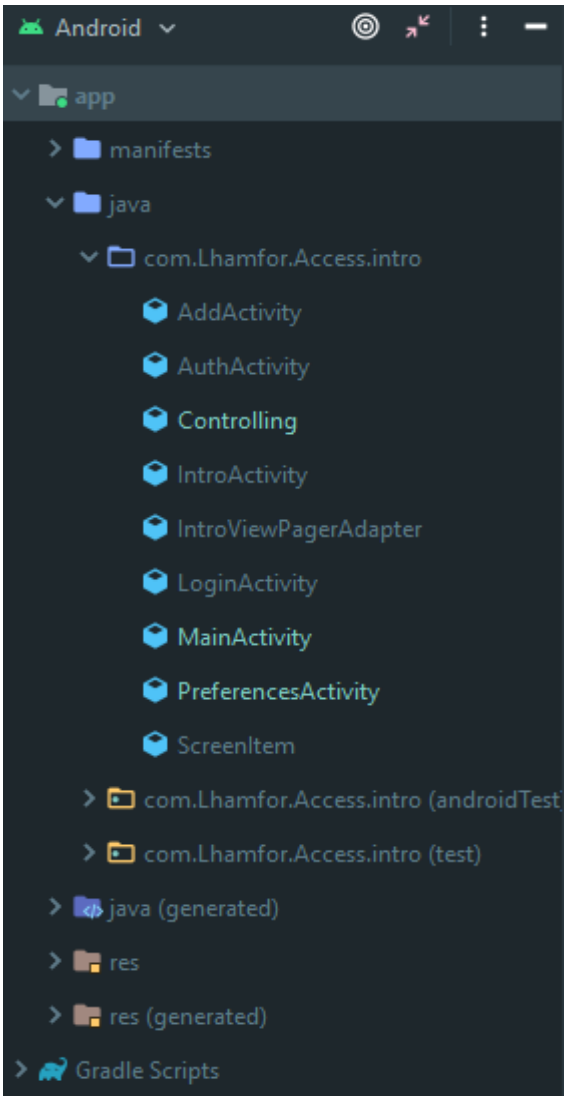


Figura 25. Estructura Android Studio.
Fuente: Elaboración Propia.

4.3.1.1. *Proceso de Codificación*

Teniendo como punto de partida las Task Card de una fase anterior mostrare la codificación que se realizó para el funcionamiento de la aplicación.

- Introducción a la aplicación (IntroActivity).
- Login de Bienvenida (LoginActivity).
- Autenticación de biométrica (AuthActivity).
- Pantalla principal del aplicativo (MainActivity).
- Registrar nuevo usuario (AddActivity).
- Clase específica para controlar la conexión a red bluetooth (Controlling).
- Listar las redes bluetooth y que la actividad Controlling pueda hacer uso de ellas (IntroViewPagerAdapter).
- Creación de los métodos getters an setters (ScreenItem).
- Nos ayudara a ver la imagen en orden junto a las instrucciones con la actividad principal (PreferenceActivity).

4.4. Cuarta Fase: Pruebas.

Ya por si la cuarta fase de pruebas evaluara nuestro desarrollo a lo largo de las 3 primeras fases evaluaremos el aplicativo en cuanto a su diseño y funcionalidad serán partes fundamentales para las pruebas

4.4.1. Pruebas de interface

Se realizaron en emuladores como en dispositivos móviles reales y podemos ver en la Tabla *22*, en dichos equipos se muestra de la misma manera sin deformaciones y corren perfectamente en las tres interfaces.

Tabla 22. *Equipos probados.*

Teléfono	**Características**
Samsung	• A50 Android Q 10, 4GB RAM 128GB memoria interno, • S9 plus Android Pie 9 actualizado Android Q 10, 6GB RAM 128GB memoria interna.
Huawei	• P30 Pro Android Oreo 8.1actualizado Android Pie 9 actualizado Android Q 10, 8GB RAM y 256GB memoria interno. • P10 Android 7 Nogout 4 GB RAM 32GB memoria interno

Fuente: Elaboración propia

4.4.2. Pruebas de aceptación

Tabla 23. *Prueba de Aceptación 1.- RF001*

Pruebas de Aceptación	
NH-001	Nombre de la Historia: Login.
Condición:	Validación de correo y password • **El sistema pedirá autenticarse con su correo (usuario) y password.** • **Una vez rellenado los campos será activará el botón de iniciar sesión.**
Entrada:	El usuario rellenara los campos usuario y password.
Resultado esperado:	Y esto activara el botón de iniciar sesión para ingresar a la siguiente actividad siempre y cuando este registrado como usuario master o el usuario master cree nuevos accesos.
Evaluación de la prueba:	Prueba Exitosa.

Fuente: Elaboración propia

Figura 26. Login de la aplicación

Fuente: Elaboración Propia.

Tabla 24. *Prueba de Aceptación 2.- RF002- RF003*

Pruebas de Aceptación	
NH-002	Nombre de la Historia: Acceso biométrico.
Condición:	Verificación de huella y reconocimiento facial • **Una vez logueado en la aplicación como medida de respaldo requerirá que se autentifique y requerirá tener Coneccion a internet.**
Entrada:	El usuario presiona en el botón Autentícate.
Resultado esperado:	Desplegara una pantalla flotante y que le indique el método de acceso que requiere.
Evaluación de la prueba:	Prueba Exitosa.

Fuente: Elaboración propia

Figura 27. Método de autenticación

Fuente: Elaboración propia

Tabla 25. *Prueba de Aceptación 3.- RF004*

Pruebas de Aceptación	
NH-003	Nombre de la Historia: Bluetooth.
Condición:	Búsqueda de redes bluetooth
	• **Ya pasado de la actividad de autentificación podremos buscar las redes bluetooth para luego conectarse al hardware.**
Entrada:	El usuario presiona en el botón buscar y posteriormente a conectarse una vez encontrado la red bluetooth.
Resultado esperado:	Mostar las redes disponibles y mandarle a un nuevo activity cunado presione conectar
Evaluación de la prueba:	Prueba Exitosa excepto por aquellas redes que no sean de hardware.

Fuente: Elaboración propia

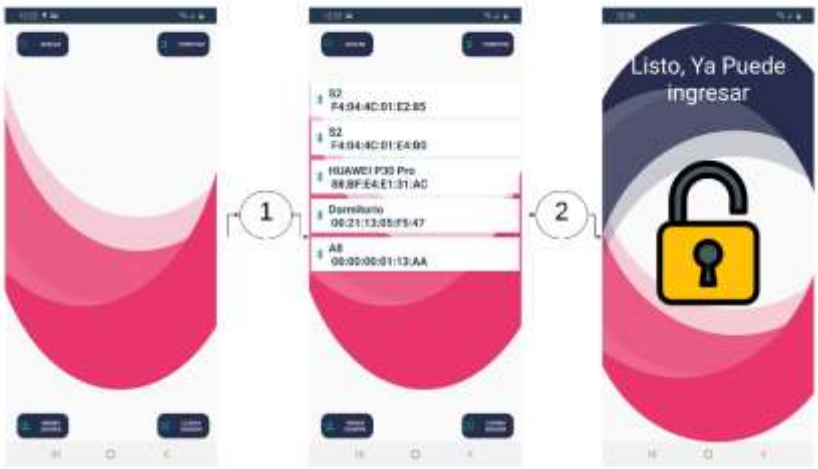


Figura 28. Búsqueda y conexión a la red bluetooth

Fuente: Elaboración propia

Tabla 26. *Prueba de Aceptación 4.- RF005*

Pruebas de Aceptación	
NH-004	Nombre de la Historia: Agregar nuevos usuarios.
Condición:	Ingresa a un nuevo activity y registra usuarios • **Debe estar logueado el usuario master para crear nuevos usuarios en la red y requerirá tener Coneccion a internet.**
Entrada:	El usuario presiona en el botón añadir usuario.
Resultado esperado:	Ingresara a la vista de add user y creara nuevos usuarios y luego pondrá registrar y devolverla a la vista principal.
Evaluación de la prueba:	Prueba Exitosa.

Fuente: Elaboración propia

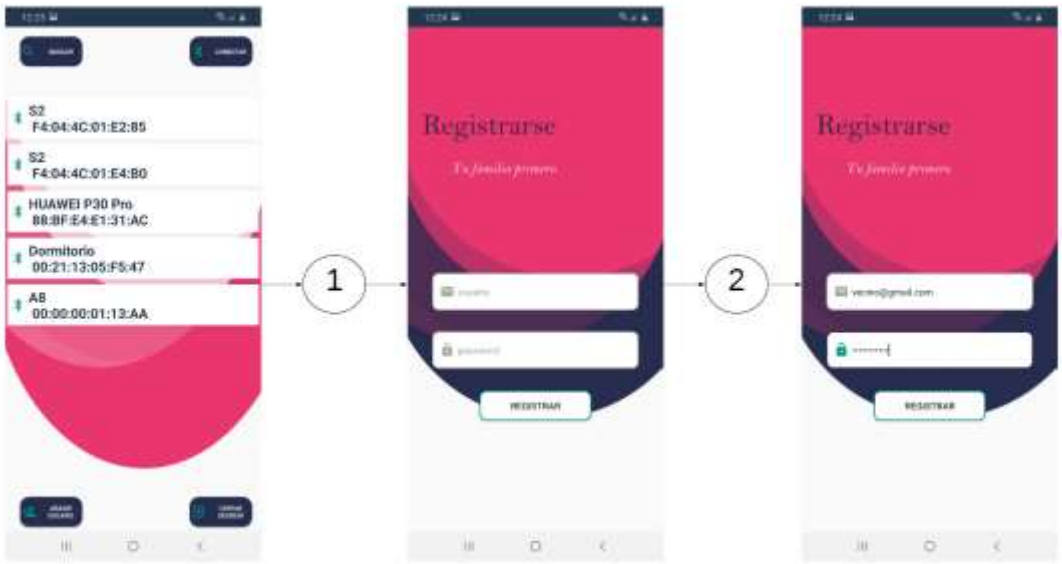


Figura 29. Add usuario

Fuente: Elaboración propia

Tabla 27. *Prueba de Aceptación 5.- RF006*

Pruebas de Aceptación	
NH-006	Nombre de la Historia: Salir de la aplicación.
Condición:	Salida del aplicativo. • **El usuario deberá terminar todas las acciones que realice para de esta manera salir dl aplicativo y previniendo el riesgo a la seguridad del hogar.**
Entrada:	El usuario presiona el botón de cerrar sesión.
Resultado esperado:	se redireccionará a la actividad de Login comenzando el siclo de loguearse con su usuario.
Evaluación de la prueba:	Prueba Exitosa.

Fuente: Elaboración propia

Figura 30. Salir de la aplicación

Fuente: Elaboración propia

Tabla 28. *Prueba de Aceptación 6.- RF007*

Pruebas de Aceptación	
NH-006	Nombre de la Historia: Intro de la aplicación.
Condición:	Ingreso la información de la aplicación. • **Descripción: El ingreso de pantalla solo se realizará por primera vez.** • **Mostrará información general de la aplicación por solo una vez.**
Entrada:	El usuario presiona en el botón next.

Resultado esperado:	Pasará de una visita a la siguiente mostrándole la segunda información y al final mostrará el Login una vez acabado las tres vistas
Evaluación de la prueba:	Prueba Exitosa.

Fuente: Elaboración propia

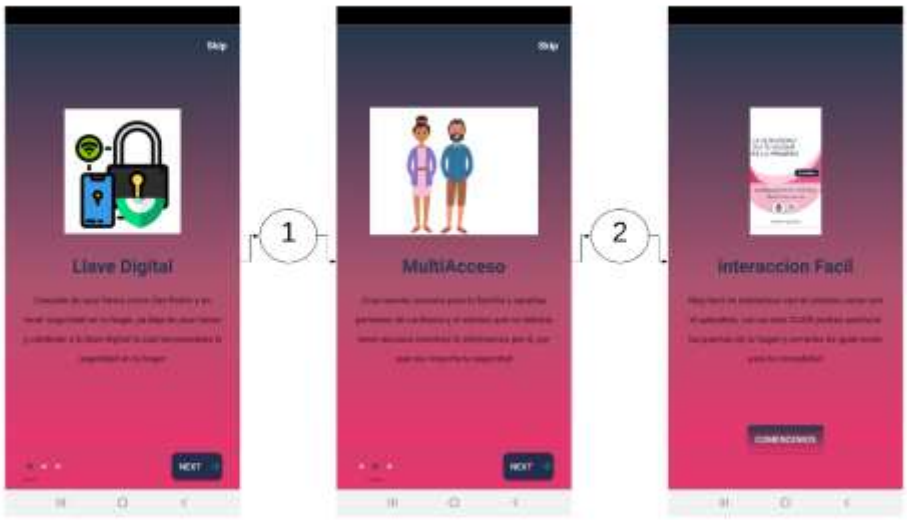


Figura 31. Introducción.

Fuente: Elaboración Propia.

4.4.3. Análisis de resultados

Tabla 29. *Resultados.*

Numero	Actividad	Usuario	Observaciones
N001	Introducción a la aplicación	Usuario	Mejorar en la forma del dinamismo de la presentación.

N002	Login	Usuario	La aplicación responde en 3 segundos.
N003	Autenticación	Usuario	La aplicación responde en menos de 1 segundo dependiendo de la calidad del teléfono.
N004	Búsqueda de la red bluetooth	Usuario	La búsqueda tarda 1 segundo.
N005	Conectarse a la red bluetooth	Usuario	La búsqueda tarda 1 segundo.
N006	Registro de usuario	Usuario	El registro cuando envía los datos tarda un promedio de 2 minutos a 2 minutos y 30 milisegundos.

Fuente: Elaboración propia

n. N001- Introducción a la aplicación

- Resultado video link: *https://youtu.be/mpgGdFngnh0*
- Codificación **Anexo C.** IntroActivity.

o. N002- Login

- Resultado video link: *https://youtu.be/4ZmYPDAO--Y*

Codificación

- **Anexo E.** Login o inicio de sesión

p. N003- Autenticación

- Resultado video link: *https://youtu.be/XBVocEPDTQc*

Codificación

Anexo F. Autenticación

- **Anexo G.** Principal actividad de la aplicación

q. N004- Búsqueda de la red bluetooth

- Resultado video link: *https://youtu.be/xuaxZ4ybYHU*

Codificación

- **Anexo G.** Principal actividad de la aplicación

r. N005- Conectarse a la red bluetooth

- Resultado video link: *https://youtu.be/-oFp-dLCdtQ*

Codificación

- **Anexo G.** Principal actividad de la aplicación

s. N006 - Registro de usuario

- Resultado video link: *https://youtu.be/vSq7ReCCKyo*

Codificación

Controlling.java

```
176     @Override
177     protected void onResume() {
178         if (mBTSocket == null || !mIsBluetoothConnected) {
179             new ConnectBT().execute();
180         }
181         Log.d(TAG,  msg: "Resumen");
182         super.onResume();
183     }
184
185     @Override
186     protected void onStop() {
187         Log.d(TAG,  msg: "Detener");
188         super.onStop();
189     }
190
191     @Override
192     protected void onSaveInstanceState(Bundle outState) {
193         super.onSaveInstanceState(outState);
194     }
195
196     @SuppressLint("StaticFieldLeak")
197     private class ConnectBT extends AsyncTask<Void, Void, Void> {
198         private boolean mConnectSuccessful = true;
199
```

Controlling.java

```
200         @Override
201         protected void onPreExecute() {
202
203             progressDialog = ProgressDialog.show(
204                     context: Controlling.this,  title: "Espere un momento",  message: "Conectando...");
205         }
206
207         @Override
208         protected Void doInBackground(Void... devices) {
209             try {
210                 if (mBTSocket == null || !mIsBluetoothConnected) {
211                     mBTSocket = mDevice.createInsecureRfcommSocketToServiceRecord(mDeviceUUID);
212                     BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
213                     mBTSocket.connect();
214                 }
215             } catch (IOException e) {
216                 mConnectSuccessful = false;
217             }
218             return null;
219         }
220
221         @Override
222         protected void onPostExecute(Void result) {
223             super.onPostExecute(result);
224
```

Controlling.java

```
221         @Override
222         protected void onPostExecute(Void result) {
223             super.onPostExecute(result);
224
225             if (!mConnectSuccessful) {
226                 Toast.makeText(getApplicationContext(),  text: "No se pudo conectar al dispositivo. "
227                         + "Por favor, encienda su hardware.", Toast.LENGTH_LONG).show();
228                 finish();
229             } else {
230                 msg( s: "Conectado al dispositivo");
231                 mIsBluetoothConnected = true;
232                 mReadThread = new ReadInput();
233             }
234             progressDialog.dismiss();
235         }
236
237     }
238     @Override
239     protected void onDestroy() { super.onDestroy(); }
242 }
```

- **Anexo H.** Registro de usuarios

CAPITULO V. Proceso de Desarrollo-hardware

Para desarrollar esta segunda parte nos enfocaremos en la parte de hardware donde describiremos a detalle los códigos requerimientos necesario para el desarrollo de este capítulo la cual utilizaremos la metodologia waterfall por su trabajo en modo cascada en base a esto se determina lo siguiente para lo cual modificamos la metodologia para el desarrollo del proyecto.

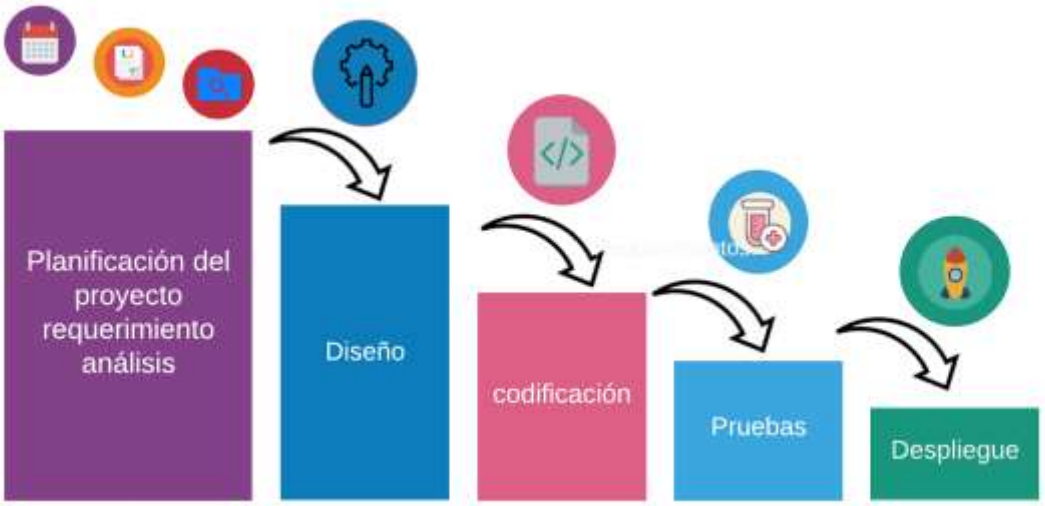


Figura 32. Metodologia Waterfall modificada.

Fuente: Elaboración propia

5.1. Planificación del proyecto

5.1.1. Análisis

5.1.1.1. *Dependencias*

- Manejo de las herramientas electrónicas para el desarrollo.

- Familiarización con el sistema embebido para un mejor uso
- El usuario debe tener en cuenta las conexiones que tenga para no dañar el sistema embebido.

5.1.1.2. *Restricciones*

- No funciona a grandes distancias
- No envía datos de cierre ni apertura de la cerradura
- El cable no son profesionales aún.
- El sistema debe estar siempre en constante alimentación de energía eléctrica.

5.1.1.3. *Configuración del proyecto*

- Arduino ide
- Java versión 8.1
- Librería para reconocimiento de los sensores
- Instalar los drivers de USB

5.1.2. Requerimientos no funcionales

Tabla 30. *Requerimientos no funcionales*

Código	Función	Descripción

RNF001	Conexión	El dispositivo bluetooth hc-06 deberá estar disponible para la conexión del disponible para la aplicación.
RNF002	Activación del relay	El relay deberá mandar la acción de abrir y controlar la corriente que pasa por el Arduino

Fuente: Elaboración propia

5.2. Diseño del sistema embebido

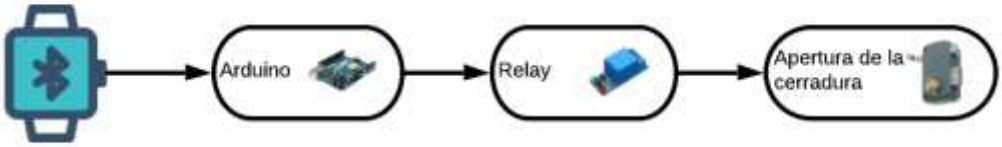


Figura 33. Arquitectura para el desarrollo del sistema embebido.
Fuente: Elaboración propia

5.2.1. Funcionalidad

- **Módulo bluetooth**: trabajará como esclavo ya que recibirá peticiones de conexión del dispositivo móvil.

- **Arduino:** será el cerebro del sistema embebido el que dará la información y funcionalidad a todo el sistema tanto de corriente como de instrucciones.

- **Transformador:** viene cuando se adquiere la cerradura eléctrica ya que ayudara a la conversion de energía para la cerradura

- **El módulo relay:** velara por la corriente de la placa y la chapa esta regulara par que no llegue a quemas.

5.3. Codificación

Tabla 31. *TaskCard 001. Activación y cambio del nombre y la contraseña del módulo bluetooth*

/ ID	**Tipo**	**Dificultad**		**Esfuerzo**		
		Antes	Después	Estimado	Gastado	**Confianza**
TC001	Nuevo	medio	medio	10 h.	9 h.	Mucha confianza (4)
Descripción						
Coneccion de bluethoot						
Fecha		Tipo de Estado		Comentarios		
04/09/2019		Día de Planificación		Se inicio de acuerdo al cronograma previamente elaborado		
05/09/2019		Día de Trabajo		Se desarrollo, método basado en ejemplo para cambiar el nombre al dispositivo como se ve en el **Anexo A** y sea más fácil de identificar y la codificación del funcionamiento		
05/09/2019		Dia de lanzamiento		No hubo errores		

Fuente: elaboracion propia

Tabla 32. TaskCard 00.

/ ID	**Tipo**	**Dificultad**		**Esfuerzo**		
		Antes	Después	Estimado	Gastado	**Confianza**

TC002	Nuevo	medio	medio	10 h.	9 h.	Mucha confianza (4)

Descripción

Coneccion del módulo relay

Fecha	Tipo de Estado	Comentarios
06/09/2019	Día de Planificación	Se inicio de acuerdo al cronograma previamente elaborado
8/09/2019	Día de Trabajo	Se desarrollo la activación del relay cuándo mande una petición el aplicativo a traves del módulo bluetooth.
10/09/2019	Dia de lanzamiento	No hubo errores

Fuente: elaboracion propia

5.3.1. Codificación en el ide de Arduino

5.3.1.1. *Void Setup*

Conexion_bluetooth §

```
#include <SoftwareSerial.h>
SoftwareSerial alexis(2, 3);

long int password1 = 92;//abrir
long int password2 = 551;// cerrar

int relay = 8;
long int data;

void setup(){

pinMode(relay, OUTPUT);
digitalWrite(relay, LOW);
delay(100);
alexis.begin(9600);

}

```

Figura 34. Declaración de variables y librerías más el método setup

Fuente: elaboracion propia

5.3.1.2. *Void Loop*

```
19 void loop(){
20
21   while(alexis.available()==0) ;
22
23  if(alexis.available()>0) {
24   data = alexis.parseInt();
25  }
26  delay(100);
27  if (data == password1){
28     digitalWrite(relay,HIGH);
29 }
30  if( data == password2){
31      digitalWrite(relay,LOW);
32  }
33 }
```

Figura 35. Tiene la parte repetitiva más la condicional donde escribimos si va en alto o bajo

Fuente: elaboracion propia

5.4. Pruebas

Tabla 33. *Prueba de Aceptación-RNF001*

Caso de prueba	
CP-001	Nombre de la Historia: Coneccion Bluetooth.
Condición:	Deberá estar activado.

- **El sistema embebido estará activo para la conexión con el aplicativo y poder interactuar hardware con software.**

Entrada: red	El módulo bluetooth estará siempre para el dispositivo
Resultado esperado:	Que esté disponible para los dispositivos android

Evaluación de la prueba:	Prueba Exitosa.

Fuente: Elaboración propia

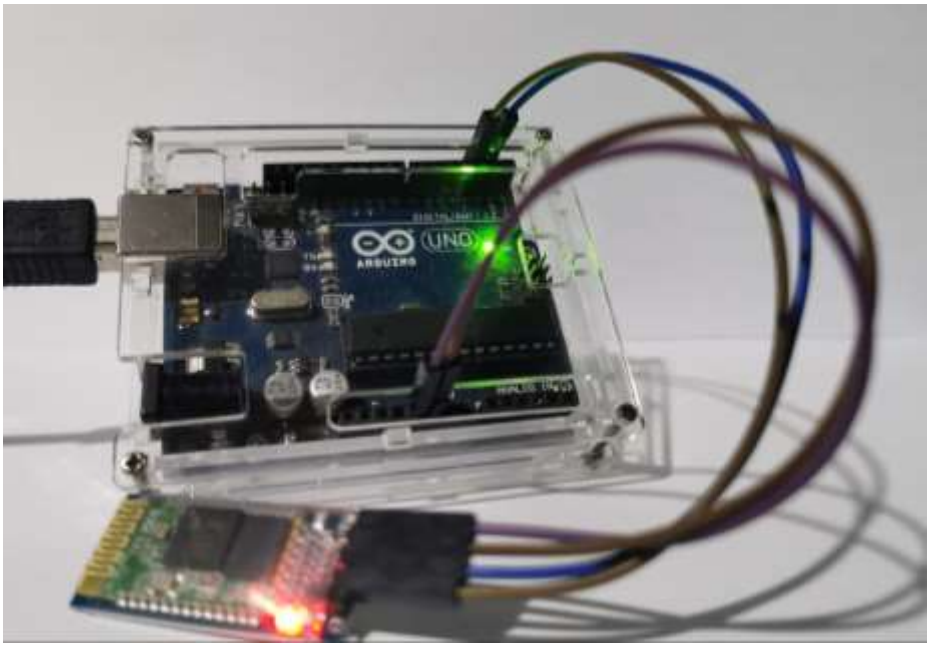


Figura 36. Arduino en funcionamiento con el módulo bluetooth.

Fuente: elaboración propia

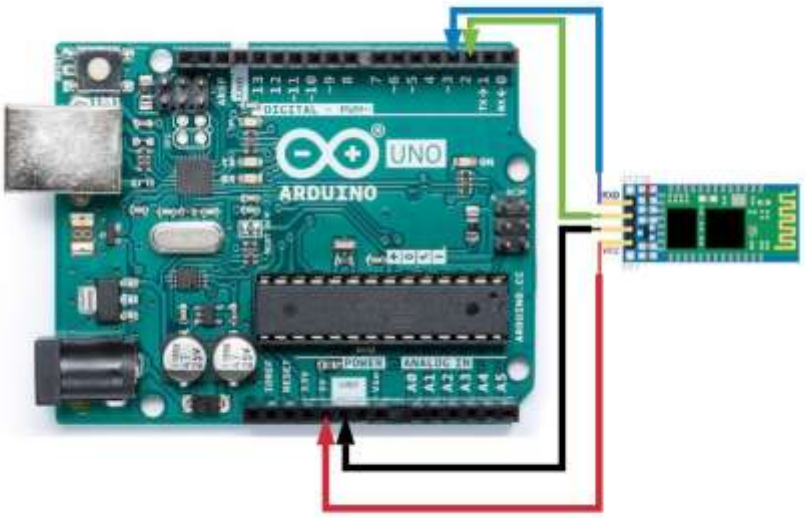


Figura 37. Arduino con el módulo bluetooth diagrama del circuito.

Fuente: elaboración propia

Tabla 34. *Prueba de Aceptación-RNF002*

Caso de prueba	
CP-002	Nombre de la Historia: Relay
Condición:	Regulará la corriente.
	• **el módulo relay esta principalmente diseñado para controlar las altas y bajas de corrientes.**
Entrada: modulo	Regulador a 12 voltios
Resultado esperado:	El funcionamiento correcto de la corriente para no dañar al circuito
Evaluación de la prueba:	Prueba Exitosa.

Fuente: Elaboración propia

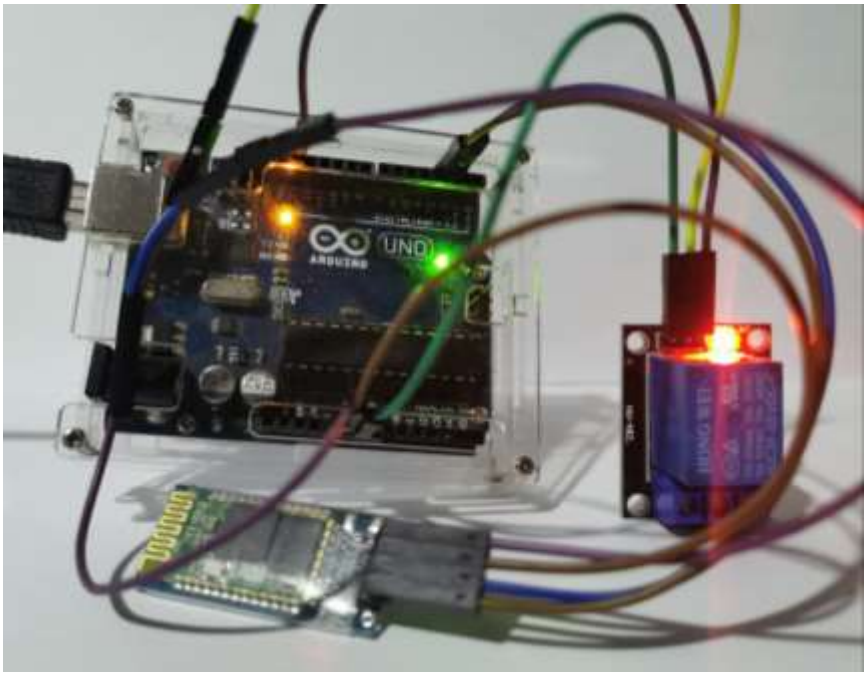


Figura 38. Arduino en funcionamiento con el módulo relay

Fuente. Elaboracion propia

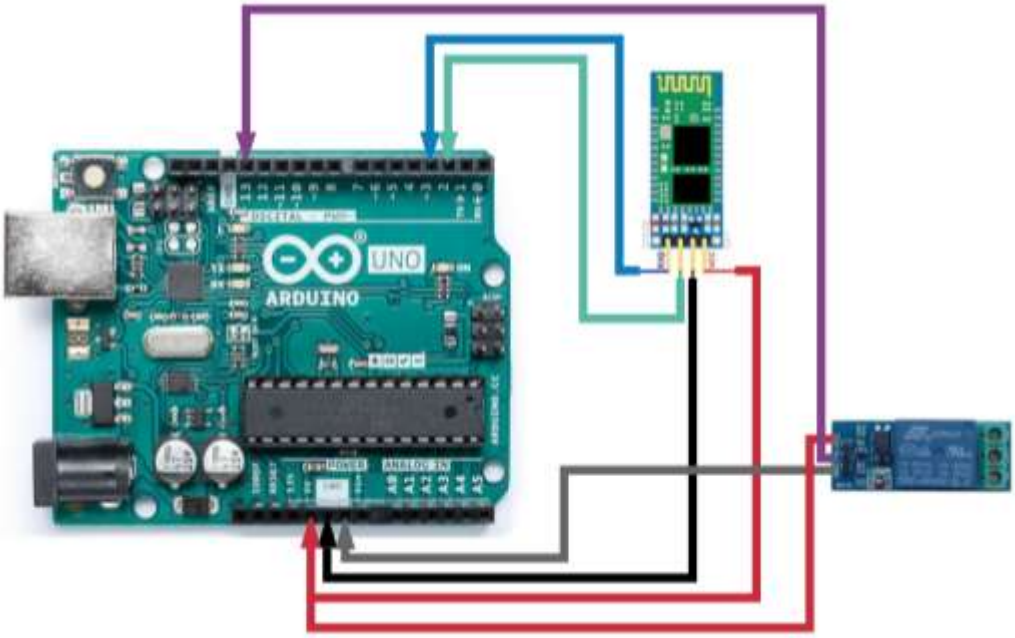


Figura 39. Arduino con el módulo relay y bluetooth diagrama del circuito

Fuente. Elaboración propia

Tabla 35. *Pruebas al sistema embebido*

Numero	Actividad	Usuario	Observaciones
N001	Conectar al módulo bluethoot	Usuario	Red con un rango de dos metros.
N002	Activación del relay	Usuario	El tiempo de respuesta durara 4 s.

Fuente: Elaboración propia

5.5. Despliegue

En esta fase logramos integrar la parte de hardware con la aplicación para revisar los últimos detalles y durante las pruebas hechas en la parte de la aplicación obtuvimos los resultados esperados y se desplegara haciendo la instalación y funcionamiento del sistema embebido junto a la aplicación.

t. N001- Despliegue

- Resultado video link: *https://youtu.be/gD12RehVjCA*
- Codificación **Anexo I.** Integración de hardware con el software

Figura 40. Instalación de la cerradura eléctrica

Fuente: Elaboración propia

Figura 41. Final de la instalación del sistema embebido

Fuente: Elaboración propia

CAPITULO VI. Resultados

6.1. Resultado 1 para el objetivo específico 1

Uno de los objetivos específicos fue desarrollo el modelo de la arquitectura para gestionar accesos en mejora de la seguridad, esto conlleva a realizarse la estructura primaria para el proyecto guiándonos de las referencias en el mercado internacional y como la idea propuesta incrementaría la seguridad en los hogares partimos por la recopilación de información acerca de que el desarrollo sea de conocimiento del investigador, en esto apoyó los requerimientos funcionales como no funcionales seguidas de nuestra arquitectura de solución y para el diseño de este objetivo y llegarlo a cumplirlos hubo muchas herramientas que fueron de apoyo y con ello se llegó a obtener un mejor resultado como se esperaba entre la parte de la aplicación y el hardware, y por último el índice de aprobación para nuestro objetivo específico para el aplicativo móvil y el hardware se puede ver en la *Figura 42*. También se puede apreciar en la *Figura 9* donde la elaboración de esta arquitectura trajo un buen resultado al momento de la implementación empezando por la aplicación y la funcionalidad empezando por la introducción luego se muestra la parte del logueo donde se ingresa datos registrados en Firebase para posteriormente ingresar a la siguiente actividad , parte como desarrollador y usuario ayudo para generar un 70% para verificar y validar esta información muestro como respaldo la Tabla *37* y **¡Error! No se encuentra el origen de la referencia.** de satisfacer el primer resultado fue desarrollar los storyboards y el esquema de navegación como podemos ver en la *Figura 14*, donde se aprecia el modo de uso de la aplicacion.

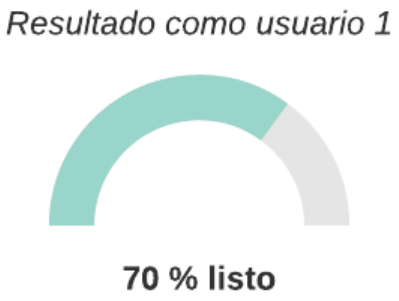


Figura 42. **N**ivel de experiencia con la aplicación y el hardware de aprobación como usuario

Fuente: Elaboración propia

6.2. Resultado 2 para el objetivo específico 2

El segundo objetivo a cumplir en el proyecto fue de Construir y desarrollar la plataforma móvil y la conexión del hardware con el software para lo cual fue de ayuda la práctica constante de la réplica de códigos para de este modo entender el funcionamiento del código desarrollado tanto en la parte de la aplicación como en el hardware, también se ganó la experiencia necesaria para darle al sistema una funcionalidad en al cual pueda ser de utilidad al usuario y de su entera compresión la conexión al hardware nos trajo un poco de problemas pero entre datos que existen en la red pudo ser solucionado a tiempo aunque se buscaron constantemente la mejor para darle simplicidad y una funcionalidad múltiple con esto nos referimos a dispositivos hardware como muestra la *Figura 41* y también vemos para evidenciar esto la codificación en la *Figura 34* y la *Figura 35*. Pero como usuario de la aplicación se dio un nivel de aprobación como se ve en la *Figura 47* que es respaldado con la **¡Error! No se encuentra el origen de la referencia.** que muestra las fechas de accesos de los usuarios también muestro el código completo respecto a la aplicación en el **Anexo M** y la dirección en donde se encuentra es en el link https://github.com/ALEXIS2ES/LhamfordAccess-base.git.

Se obtuvo como opinión de usuario y desarrollador bajo el desarrollo que se muestra en la *Figura 46* y parte del resultado obtenido se muestra desde el **Anexo B** hasta el **Anexo I** que muestra la integración a nivel de hardware, ya que tenemos un diseño sencillo y funcional también podemos apreciar esto en la *Figura 40*, donde se muestra la instalación en la puerta y su funcionamiento sin fallas ya que debemos considerar que la puerta que se muestra es de referencia. Y el link *https://youtu.be/gD12RehVjCA* contiene o muestra el funcionamiento del hardware y la aplicación.

Tabla 36. *Pruebas de usuario por fechas*

Numero	**Fecha**	**Fecha**	**Fecha**	**Fecha**	**Fecha**	**Fecha**	**Fecha**
Usu01	22/12/19	30/12/19	05/01/19	08/01/20	18/01/20	01/02/20	07/02/20
Usu02	15/12/19	21/12/19	13/01/19	15/01/20	10/01/20	07/02/20	13/02/20
Usu03	Octubre	Nov-Dic	Ene-Feb	Mar-Abr	Mayo	Junio	Julio
Usu04	20/11/19	22/11/19	15/12/19	29/12/19	17/01/19	28/01/19	10/02/19
Usu05	03/11/19	12/11/19	10/12/19	12/01/20	15/01/20	20/02/20	28/02/20
Usu06	29/10/19	20/11/19	23/12/19	04/01/20	28/01/20	14/02/20	03/03/20
Usu07	12/11/19	01/11/19	07/11/19	20/12/19	30/01/19	25/01/19	19/03/19

Fuente: Elaboración propia

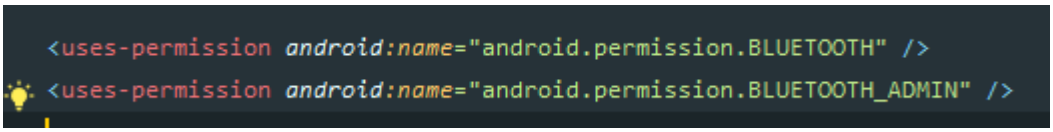


Figura 43. Configuración de variables bluethoot

Fuente: Elaboración propia

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    switch (requestCode) {
        case BT_ENABLE_REQUEST:
            if (resultCode == RESULT_OK) {
                msg( str: "conexion exitoso");
                new SearchDevices().execute();
            } else {
                msg( str: "no pudo conectarse");
            }
            break;
        case SETTINGS:
            SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences( context: this);
            String uuid = prefs.getString( key: "prefUuid",  defValue: "Null");
            mDeviceUUID = UUID.fromString(uuid);
            Log.d(TAG,  msg: "UUID: " + uuid);
            String bufSize = prefs.getString( key: "prefTextBuffer",  defValue: "Null");
            mBufferSize = Integer.parseInt(bufSize);
            break;
        default:
            break;
    }
    super.onActivityResult(requestCode, resultCode, data);
}
```

Figura 44. Coneccion de hardware y la aplicación

Fuente: Elaboración propia

```
Abrir_Cerrar.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if(isChecked){
            try {
                mBTSocket.getOutputStream().write(abrir.getBytes());
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        else{
            try {
                mBTSocket.getOutputStream().write(cerrar.getBytes());
            } catch (IOException e) {
                e.printStackTrace();
            }

        }
    }
});
```

Figura 45. Envió de datos al módulo bluetooth y este a su vez al módulo relay

Fuente: Elaboración propia

```
@Override
protected void onPostExecute(Void result) {
    super.onPostExecute(result);

    if (!mConnectSuccessful) {
        Toast.makeText(getApplicationContext(),  text: "No se pudo conectar al dispositivo. " +
                "Por favor, encienda su hardware.", Toast.LENGTH_LONG).show();
        finish();
    } else {
        msg( s: "Conectado al dispositivo");
        mIsBluetoothConnected = true;
        mReadThread = new ReadInput();
    }

    progressDialog.dismiss();
}
```

Figura 46. Código de Coneccion el módulo Bluetooth

Fuente: Elaboración propia

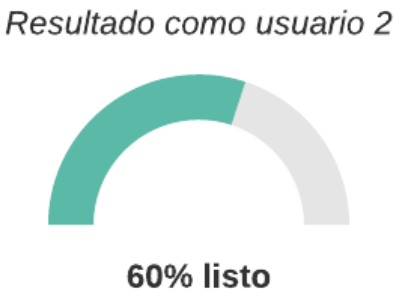


Figura 47. **N**ivel de experiencia con la aplicación y el hardware de aprobación como usuario

Fuente: Elaboración propia

6.3. Resultado 3 para el objetivo específico 3

Realizar pruebas al aplicativo y al hardware para la gestión de accesos en la seguridad ya completada la parte de construcción del aplicativo y el hardware se hizo la instalación para hacer la prueba y ver si tiene alguna falla el sistema ya pasado el mes no tuvimos fallas en el sistema y la aplicación registraba el día que el usuario ingresaba actualizando de esta manera la lista de usuarios solamente se actualizaba se podía registrar usuarios las fechas con normalidad ya que la base de datos daba la funcionalidad

por defecto, se utilizó como autentificador la opción email en la base de datos ya que si se hubiese dado acceso de google cualquier usuario que tuviese el correo podría ingresar normalmente al aplicativo sin necesidad de registro y dando como resultado como un punto de error para la aplicación en fin los resultados de la pruebas elaboradas dieron un buen resultado y el sistema no tubo fallas en el aplicativo cuando se hizo la migración del aplicativo a una nueva versión para que de esta manera pueda adaptarse a los distintos dispositivos modernos. Y tenga más herramientas de desarrollo, la aceptación que se tuvo como usuario se muestra en la *Figura 48* y esto también se puede apreciar en la Tabla *29* para la aplicación y la tabla Tabla *35* para el sistema embebido.

Como ultimo resultado en nuestro tercer objetivo específico se muestra la *Figura 16*, *Figura 17*, *Figura 18*, *Figura 19*, *Figura 20*, *Figura 21*, *Figura 22*, *Figura 23* y en la *Figura 24* como parte de respaldar el resultado mostrándonos cada etapa del aplicativo en su etapa o fase final de la cuales se hicieron pruebas como en la Tabla *29* donde nos muestra los resultados y observaciones de la aplicación y también podemos ves esto en *Figura 40*. Instalación de la cerradura eléctrica Donde vemos la instalación y en el siguiente link *https://youtu.be/gD12RehVjCA* vemos el funcionamiento como prueba para respaldar la *Figura 48* tenemos la Tabla *37*, siempre viéndolo como desarrollador y usuario.

Tabla 37. *Pruebas de usuario*

Numero	**Nombre**	**Prueba1**	**Prueva2**	**Prueba3**	**Prueva4**	**Prueba5**	**Prueba 6**
Usu01	Frank	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa
	CCamerccoa	80%	81%	85%	90%	90%	98%
Usu02	Michell	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa
	CCamerccoa	78%	84%	88%	94%	96%	98%
Usu03	Alexis	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa
	CCamerccoa	90%	93%	95%	97%	99%	100%

Usu04	Roger CCamerccoa	Exitosa 79%	Exitosa 82%	Exitosa 85%	Exitosa 89%	Exitosa 91%	Exitosa 98%
Usu05	Obdulia Mamani	Exitosa 50%	Exitosa 65%	Exitosa 75%	Exitosa 80%	Exitosa 85%	Exitosa 90%
Usu06	Leonardo CCamerccoa	Exitosa 55%	Exitosa 65%	Exitosa 75%	Exitosa 85%	Exitosa 90%	Exitosa 95%
Usu07	Haysen CCamerccoa	Exitosa 65%	Exitosa 75%	Exitosa 85%	Exitosa 89%	Exitosa 90%	Exitosa 95%

Fuente: Elaboración propia

Figura 48. **N**ivel de experiencia con la aplicación y el hardware de aprobación como usuario

Fuente: Elaboración propia

CAPITULO VII. Conclusiones y Recomendaciones

La conclusión obtenida está en base principalmente a los objetivos trazados en el proyecto en primer lugar por las herramientas utilizadas en el proyecto se adaptaron de buena manera en el desarrollo, uno comenzando en la metodologia cuando se decidió utilizarla hubo las modificaciones a esta y con ello conseguimos mostrar todo lo que se quería obtener en el proyecto, ya que su enfoque se dio en el desarrollo de la aplicación, tomando de guía los requerimientos funcionales y no funcionales para posteriormente crear las storyboard donde definimos el diseño de la aplicacion con esto conseguimos tener una interfaz agradable para el usuario también fue de utilidad los Storycards y las Taskcards. Para la verificación del desarrollo planteado resultase exitoso, evaluamos mediante las pruebas de aceptación todo ello ayudo a concluir en que la aplicación satisface en incrementar la seguridad de la vivienda donde se implementó el sistema embebido más la aplicación.

También se determinó el tiempo en la que trabaja el aplicativo con el hardware es un total 1 minuto con la práctica y un total de 4 minutos sin práctica. interactuar con el sistema en los primeros usos el usuario estará tomando un estimado de 3 minutos a 4. Cómo también se vio la mejora que puede implementarse para una segunda versión del aplicativo, se dio las alertas necesarias en la aplicación. Pal caso la alerta de la conexión a la red en la parte de inicio de sesión para luego verificar si la persona que ingrese al sistema también sea válida sus datos biométricos, y de esta manera confirmar que la persona que ingrese al aplicativo esté registrada en el dispositivo móvil como en la base de datos.

En la pantalla de conexión donde se muestra la red bluetooth el usuario deberá seleccionar la ser configurada, ya que el módulo tiene una contraseña distinta a la que viene por default de esta manera tendremos varias redes con distintas contraseñas para la seguridad del usuario y también se vieron las modificaciones en la parte de hardware y de crear un sistema robusto para ver todo y como ha de verse el aplicativo en dispositivos físicos y no virtuales y ver su desempeño total y analizar los puntos fuertes de la aplicación con esto contribuimos a incrementar la seguridad

en cada vivienda teniendo un control de accesos a personas registradas, sera de apoyo total por que en la actualidad todo el mundo porta un dispositivo móvil de gama media y esto es más que suficiente para que el aplicativo sea instalado se concluye que la aplicación es de apoyo en la seguridad. Como parte del resultado pongo un video que muestra el proyecto desarrollado y su funcionamiento podemos ver el resultado obtenido en el siguiente link *https://youtu.be/BVt5VhrmACU*.

7.1. Recomendaciones

Como dato final la investigación realizada es el punto de partida de buscar nuevas herramientas para mejorar la seguridad en la vivienda no solo controlando el acceso si no implementando sensores de movimiento que captan cualquier movimiento inusual por la noches activando alarmas y sensores dactilares para agilizar el ingreso también crear accesos a distancia más largas es una de las mejoras que se pueda dar a la investigación otro punto seria elaborar sistemas domóticos que se acomoden al bolsillo de las personas pero que sean completos en funcionalidad una vez conseguida esto podemos analizar los resultados que se obtengas que sistemas domóticos son los más solicitados o crear nuevos controles de accesos ya que no es el único modo de controlar accesos mediante internet de las cosas si no que tenemos una amplia gama de métodos para reflejar esto en una base de datos y convertirlas en una ente inteligente.

- Como recomendación de parte del investigador utilizar marcos cerrados en la fabricación de puertas o al momento de adquirirlas y no las que tiene abertura ya que pondría en juego la seguridad de su hogar.

- Mejorar la red bluetooth migrándola a una red wifi para aumentar la distancia de apertura y poder controlar más sensores.

- Crear una nueva cerradura eléctrica que se adapte a todo tipo de puerta y convirtiéndola a una que tenga lector biométrico.

LISTA DE REFERENCIAS

©ASUSTeK Computer Inc. (2018). *Tinker Board | Tarjeta de Desarrollo | ASUS*. https://www.asus.com/latin/Single-Board-Computer/Tinker-Board/

Arduino. (2019). *Arduino: tecnologia para todos*. https://doi.org/10.1007/SpringerReference_5244

Astrium. (2014). *Android*. https://www.android.com/intl/es_es/

Benítez. (2017). *Architectural Proposal for Internet of Things*. *November*. https://www.researchgate.net/publication/320353907

Calvo. (2014). *Anàlisis Y Diseño De Una Red Domòtica para Viviendas Sociales*. 118. http://cybertesis.uach.cl/tesis/uach/2014/bmfcic169a/doc/bmfcic169a.pdf

Castro. (2016). "Internet de las cosas. Privacidad y Seguridad,. *Nternet de Las Cosas. Privacidad y Seguridad*, 94. http://sinbad2.ujaen.es/sites/default/files/publications/Memoria_0.pdf

Culquichicon. (2012). *DOMOLAB: SISTEMA DE MONITOREO Y CONTROL REMOTO DE VIVIENDAS*. 2273393. http://repositorio.puce.edu.ec/bitstream/handle/22000/8492/INTERNET DE LAS COSAS TESIS Y CONSIDERACIONES DE SEGURIDAD - FINAL.pdf?sequence=1&isAllowed=y

Cuzme. (2015). El Internet De Las Cosas Y Las Consideraciones De Seguridad. *Pontificia Universidad Católica Del Ecuador*, 179. https://doi.org/10.1017/CBO9781107415324.004

Electrónica ElectroPro. (2017). *Raspberry Pi 3 Modelo B+ con WiFi Bluetooth y 1GB de Memoria RAM*. https://electropro.pe/index.php?route=product/product&product_id=866

guerrero. (2015). *Metodología Mobile-D: Para desarrollos de aplicaciones móviles - Blog de Manuel Guerrero*. http://manuelguerrero.blogspot.es/1446543763/metodologia-mobile-d-para-desarrollos-de-aplicaciones-moviles/

Hardkernel co., L. (2019). *Products – ODROID*. https://www.hardkernel.com/product/page/7/

INEI. (2017). *Robo en la vivienda*. https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1519/cap03.pdf

Perez, bustos & beron & henriques. (2018). *ANÁLISIS SISTEMÁTICO DE LA SEGURIDAD EN INTERNET OF THINGS*. *2*.

Salazar, S. (2014). Internet de las cosas. *Internet de Las Cosass*, 24. https://doi.org/10.1016/j.cirp.2011.03.145

Sinnaps. (2019). *Metodología Waterfall: ¿qué es y cómo combinarla con Agile? | Sinnaps*. https://www.sinnaps.com/blog-gestion-proyectos/metodologia-waterfall

Solano. (2014). *Análisis de los sistemas de gestión de bases de datos actuales como soporte para internet a las tecnologías de internet de las cosas*. 92. https://riunet.upv.es/handle/10251/43325

Velandia. (2014). *SISTEMA MÓVIL DE SONDEO PREVENTIVO DE VEHÍCULOS CON SOPORTE OBDII PARA MEJORAR LA VIDA ÚTIL DEL AUTOMOTOR*.

ANEXOS

Anexo A. cambio de contraseña y password del módulo bluetooth

Anexo B. Dependencia de android studio para el desarrollo.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'

    implementation 'com.google.android.material:material:1.3.0-alpha01'

    implementation 'com.google.firebase:firebase-core:17.4.3'
    implementation 'com.google.firebase:firebase-auth:19.3.1'

    implementation "androidx.biometric:biometric:1.0.1"

    implementation 'pl.droidsonroids.gif:android-gif-drawable:1.2.8'
```

Anexo C. IntroActivity.

```
public class IntroActivity extends AppCompatActivity {

    private ViewPager screenPager;
    IntroViewPagerAdapter introViewPagerAdapter ;
    TabLayout tabIndicator;
    Button btnNext;
    int position = 0 ;
    Button btnGetStarted;
    Animation btnAnim ;
    TextView tvSkip;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);
        if (restorePrefData()) {

            Intent loginActivity = new Intent(getApplicationContext(),LoginActivity.class );
            startActivity(loginActivity);
            finish();
        }

        setContentView(R.layout.activity_intro);

        getSupportActionBar();

        btnNext = findViewById(R.id.btn_next);
        btnGetStarted = findViewById(R.id.btn_get_started);
        tabIndicator = findViewById(R.id.tab_indicator);
        btnAnim = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.button_animation);
        tvSkip = findViewById(R.id.tv_skip);
```

```
final List<ScreenItem> mList = new ArrayList<>();
mList.add(new ScreenItem( title: "Llave Digital", description: "Cansado de usar llaves co
mList.add(new ScreenItem( title: "MultiAcceso", description: "Crea nuevos accesos para tu
mList.add(new ScreenItem( title: "Facil Interaccion", description: "Muy facil de interact

screenPager =findViewById(R.id.screen_viewpager);
introViewPagerAdapter = new IntroViewPagerAdapter( mContext: this,mList);
screenPager.setAdapter(introViewPagerAdapter);


tabIndicator.setupWithViewPager(screenPager);

btnNext.setOnClickListener((v) → {

        position = screenPager.getCurrentItem();
        if (position < mList.size()) {

            position++;
            screenPager.setCurrentItem(position);
        }
        if (position == mList.size()-1) {

            loaddLastScreen();
        }
});

tabIndicator.addOnTabSelectedListener(new TabLayout.BaseOnTabSelectedListener() {
    @Override
    public void onTabSelected(TabLayout.Tab tab) {

        if (tab.getPosition() == mList.size()-1) {

            loaddLastScreen();
        }
    }
    @Override
    public void onTabUnselected(TabLayout.Tab tab) {

    }
```

```
        }
        @Override
        public void onTabReselected(TabLayout.Tab tab) {

        }
    });

    btnGetStarted.setOnClickListener((v) -> {

            Intent loginActivity = new Intent(getApplicationContext(),LoginActivity.class );
            startActivity(loginActivity);
            savePrefsData();
            finish();
    });

    tvSkip.setOnClickListener((v) -> {

            screenPager.setCurrentItem(mList.size());
    });
}

private boolean restorePrefData() {

    SharedPreferences pref = getApplicationContext().getSharedPreferences( name: "myPrefs",MODE_PRIVATE);
    Boolean isIntroActivityOpnendBefore = pref.getBoolean( key: "isIntroOpnend", defValue: false);
    return  isIntroActivityOpnendBefore;
}

private void savePrefsData() {

    SharedPreferences pref = getApplicationContext().getSharedPreferences( name: "myPrefs",MODE_PRIVATE);
    SharedPreferences.Editor editor = pref.edit();
    editor.putBoolean("isIntroOpnend",true);
    editor.commit();
}
    private void loaddLastScreen() {

        btnNext.setVisibility(View.INVISIBLE);
        btnGetStarted.setVisibility(View.VISIBLE);
        tvSkip.setVisibility(View.INVISIBLE);
        tabIndicator.setVisibility(View.INVISIBLE);

        btnGetStarted.setAnimation(btnAnim);

    }
}
```

Anexo D. Alert Dialog

```
//Alerta de dialogo de conneccion a internet
ConnectivityManager connectivityManager = (ConnectivityManager)
        getApplicationContext().getSystemService(Context.CONNECTIVITY_SERVICE);

NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo();

if (networkInfo == null || !networkInfo.isConnected()|| !networkInfo.isAvailable()){


    Dialog dialog = new Dialog( context: this);

    dialog.setContentView(R.layout.alert_dialog);

    dialog.setCanceledOnTouchOutside(false);

    dialog.getWindow().setLayout(WindowManager.LayoutParams.WRAP_CONTENT,
            WindowManager.LayoutParams.WRAP_CONTENT);

    dialog.getWindow().setBackgroundDrawable(new ColorDrawable(TRANSPARENT));

    dialog.getWindow().getAttributes().windowAnimations =
            android.R.style.Animation_Dialog;


    Button btTryAgain = dialog.findViewById(R.id.bt_try_again);

    btTryAgain.setOnClickListener((v) → {

            recreate();
    });

    dialog.show();
}
```

Anexo E. Login o inicio de sesión

```
public class LoginActivity extends AppCompatActivity {

    //variables globales
    private static final String TAG = " ";
    EditText mCorreo, mClave;
    Button minicioSession;

    //variables de faceboo
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        //instanciamos la vista
        mCorreo = (EditText) findViewById(R.id.txt_correo);
        mClave = (EditText) findViewById(R.id.txt_password);
        minicioSession =  (Button) findViewById(R.id.btn_iniciosession);

        //doy la funcion del boton en desactivado si no estan completados
        mCorreo.addTextChangedListener(loginTextWatcher);
        mClave.addTextChangedListener(loginTextWatcher);

        // instancio las variables de firebase
        mAuth = FirebaseAuth.getInstance();

        //doy funcion al botton inicio de session
        minicioSession.setOnClickListener((v) → {
                String email, password;
                email = mCorreo.getText().toString();
                password = mClave.getText().toString();
                mAuth.signInWithEmailAndPassword(email, password)
                        .addOnCompleteListener( activity: LoginActivity.this, (task) → {
                                if (task.isSuccessful()) {
```

```
                            // Sign in success, update UI with the signed-in user's information
                            Log.d(TAG,  msg: "signInWithEmail:success");
                            Intent intent = new Intent( packageContext: LoginActivity.this, AuthActivity.class);
                            startActivity(intent);
                            Toast.makeText( context: LoginActivity.this,  text: "Verifiquemos que eres tu.",
                                    Toast.LENGTH_SHORT).show();
                            FirebaseUser user = mAuth.getCurrentUser();
                            updateUI(user);
                        } else {
                            // If sign in fails, display a message to the user.
                            Log.w(TAG,  msg: "signInWithEmail:failure", task.getException());
                            Intent intent = new Intent( packageContext: LoginActivity.this, LoginActivity.class);
                            startActivity(intent);
                            Toast.makeText( context: LoginActivity.this,  text: "Error al iniciar session.",
                                    Toast.LENGTH_SHORT).show();
                            updateUI( user: null);
                        }
                    });
        });
    //activar el button  de inicion siempre este completado
    private TextWatcher loginTextWatcher = new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
            String usernameInput = mCorreo.getText().toString().trim();
            String passwordInput = mClave.getText().toString().trim();

            minicioSession .setEnabled(!usernameInput.isEmpty() && !passwordInput.isEmpty());

        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    };

    //este metodo sirve para verificar si el usuario existe en la base de datos
    @Override
    protected void onStart() {
        super.onStart();
        FirebaseUser currentUser = mAuth.getCurrentUser();
        updateUI(currentUser);
    }

    //metodo para validar la informacion del usuario e ingresar a una nueva activity
    private void updateUI(FirebaseUser user) {
        if (user != null)
        {
            Intent intent = new Intent( packageContext: LoginActivity.this, AuthActivity.class);
            startActivity(intent);
            finish();
        }
    }
```

Anexo F. Autenticación

```java
public class AuthActivity extends AppCompatActivity {

    @RequiresApi(api = Build.VERSION_CODES.P)

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_auth);

        final Executor executor = Executors.newSingleThreadExecutor();

        final BiometricPrompt biometricPrompt = new BiometricPrompt.Builder( context: this)
                .setTitle("Verifiquemos Su Huella o Rostro")
                .setSubtitle("Coloque su huella Registrada")
                .setDescription("Controlamos su Acceso")
                .setNegativeButton( text: "Cancel", executor, new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                    }
                }).build();

        Button authenticate = findViewById(R.id.fingerprint);

        final AuthActivity activity = this;

        authenticate.setOnClickListener((v) -> {
                biometricPrompt.authenticate(new CancellationSignal(), executor, onAuthenticationSucceeded(result) -> {
                        activity.runOnUiThread(() -> {
                                Intent next = new Intent( packageContext: AuthActivity.this, MainActivity.class);
                                startActivity(next);
                                /*Toast.makeText(MainActivity.this, "Authenticated", Toast.LENGTH_LONG).show();*/
                        });

                });
        });
    }
}
```

Anexo G. Principal actividad de la aplicación

Controlling.java

```
1   package com.Lhamfor.Access.intro;
2
3   import ...
21
22  public class Controlling extends Activity {
23
24      private static final String TAG = " ";
25      private int mMaxChars = 50000;
26      private UUID mDeviceUUID;
27      private BluetoothSocket mBTSocket;
28      private ReadInput mReadThread = null;
29
30      private boolean mIsUserInitiatedDisconnect = false;
31      private boolean mIsBluetoothConnected = false;
32
33      private BluetoothDevice mDevice;
34
35      final static String abrir="92";//abrir
36      final static String cerrar="551";//cerrar
37
38      private ProgressDialog progressDialog;
39      ToggleButton Abrir_Cerrar;
```

Controlling.java

```
42      @Override
43      protected void onCreate(Bundle savedInstanceState) {
44          super.onCreate(savedInstanceState);
45          setContentView(R.layout.activity_controlling);
46
47          /*Activityhelper.initialize(this);*/
48          Abrir_Cerrar=(ToggleButton)findViewById(R.id.abrir_cerrar);
49
50          Intent intent = getIntent();
51          Bundle b = intent.getExtras();
52          mDevice = b.getParcelable(MainActivity.DEVICE_EXTRA);
53          mDeviceUUID = UUID.fromString(b.getString(MainActivity.DEVICE_UUID));
54          mMaxChars = b.getInt(MainActivity.BUFFER_SIZE);
55
56          Log.d(TAG,  msg: "Listo");
57
```

Controlling.java

```
57
58          Abrir_Cerrar.setOnCheckedChangeListener((buttonView, isChecked) -> {
61                  if(isChecked){
62                      try {
63                          mBTSocket.getOutputStream().write(abrir.getBytes());
64                      } catch (IOException e) {
65                          e.printStackTrace();
66                      }
67                  }
68                  else{
69                      try {
70                          mBTSocket.getOutputStream().write(cerrar.getBytes());
71                      } catch (IOException e) {
72                          e.printStackTrace();
73                      }
74
75                  }
76          });
78      }
```


Controlling.java
private class ReadInput implements Runnable {
private boolean bStop = false;
private Thread a;
public ReadInput() {
a = new Thread(target: this, name: "Entrada");
a.start();
}
public boolean isRunning() { return a.isAlive(); }
@Override
public void run() {
InputStream inputStream;
try {
inputStream = mBTSocket.getInputStream();
while (!bStop) {
byte[] buffer = new byte[256];
if (inputStream.available() > 0) {
inputStream.read(buffer);
int i = 0;
Controlling.java
for (i = 0; i < buffer.length
&& buffer[i] != 0; i++) {
}
final String
= new String(
buffer, offset: 0, i);
}
Thread.sleep(millis: 500);
}
} catch (IOException e) {
e.printStackTrace();
} catch (InterruptedException e) {
e.printStackTrace();
}
}
public void stop() { bStop = true; }
}


Controlling.java
private class DisConnectBT extends AsyncTask<Void, Void, Void> {
@Override
protected void onPreExecute() {
}
@Override
protected Void doInBackground(Void... params) {
if (mReadThread != null) {
mReadThread.stop();
while (mReadThread.isRunning())
;
mReadThread = null;
}
try {
mBTSocket.close();
} catch (IOException e) {
e.printStackTrace();
}
return null;
}

Controlling.java

```
152         @Override
153         protected void onPostExecute(Void result) {
154             super.onPostExecute(result);
155             mIsBluetoothConnected = false;
156             if (mIsUserInitiatedDisconnect) {
157                 finish();
158             }
159         }
160     }
161
162     private void msg(String s) {
163         Toast.makeText(getApplicationContext(), s, Toast.LENGTH_SHORT).show();
164     }
165
166     @Override
167     protected void onPause() {
168         if (mBTSocket != null && mIsBluetoothConnected) {
169             new DisConnectBT().execute();
170         }
171         Log.d(TAG,  msg: "Pausa");
172         super.onPause();
173     }
```

Controlling.java

```
176        @Override
177        protected void onResume() {
178            if (mBTSocket == null || !mIsBluetoothConnected) {
179                new ConnectBT().execute();
180            }
181            Log.d(TAG,  msg: "Resumen");
182            super.onResume();
183        }
184
185        @Override
186        protected void onStop() {
187            Log.d(TAG,  msg: "Detener");
188            super.onStop();
189        }
190
191        @Override
192        protected void onSaveInstanceState(Bundle outState) {
193            super.onSaveInstanceState(outState);
194        }
195
196        @SuppressLint("StaticFieldLeak")
197        private class ConnectBT extends AsyncTask<Void, Void, Void> {
198            private boolean mConnectSuccessful = true;
199
```

Controlling.java

```
200         @Override
201         protected void onPreExecute() {
202
203             progressDialog = ProgressDialog.show(
204                     context: Controlling.this,  title: "Espere un momento",  message: "Conectando...");
205         }
206
207         @Override
208         protected Void doInBackground(Void... devices) {
209             try {
210                 if (mBTSocket == null || !mIsBluetoothConnected) {
211                     mBTSocket = mDevice.createInsecureRfcommSocketToServiceRecord(mDeviceUUID);
212                     BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
213                     mBTSocket.connect();
214                 }
215             } catch (IOException e) {
216                 mConnectSuccessful = false;
217             }
218             return null;
219         }
220
221         @Override
222         protected void onPostExecute(Void result) {
223             super.onPostExecute(result);
224
```

Controlling.java

```
221         @Override
222         protected void onPostExecute(Void result) {
223             super.onPostExecute(result);
224
225             if (!mConnectSuccessful) {
226                 Toast.makeText(getApplicationContext(),  text: "No se pudo conectar al dispositivo. "
227                         + "Por favor, encienda su hardware.", Toast.LENGTH_LONG).show();
228                 finish();
229             } else {
230                 msg( s: "Conectado al dispositivo");
231                 mIsBluetoothConnected = true;
232                 mReadThread = new ReadInput();
233             }
234             progressDialog.dismiss();
235         }
236
237     }
238     @Override
239     protected void onDestroy() { super.onDestroy(); }
242 }
```

Anexo H. Registro de usuarios

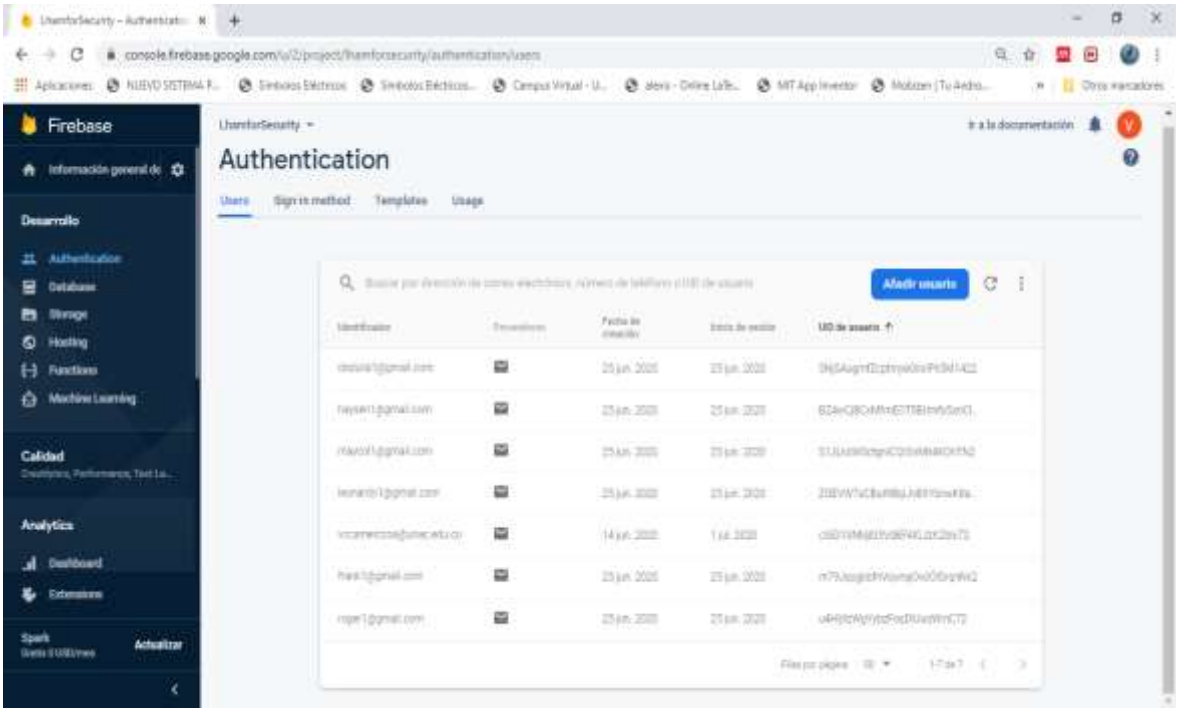
```
public class AddActivity extends AppCompatActivity {

    // variables globales
    private static final String TAG = " ";
    EditText mCorreo, mClave;
    Button mRegister;

    //variables de firebase
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add);

        mCorreo = (EditText) findViewById(R.id.txt_correo);
        mClave = (EditText) findViewById(R.id.txt_password);
        mRegister =  (Button) findViewById(R.id.btn_register);

        // instanmcia la varible de firebase
        mAuth = FirebaseAuth.getInstance();

        //damos una accion al botton mregister
        mRegister.setOnClickListener((v) → {
                String email, password;
                email = mCorreo.getText().toString();
                password = mClave.getText().toString();

                //patron para validar el email
                Pattern patternEmail = Pattern
                        .compile("^[ A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@"
                                + "[A-Za-z0-9-]+(\\.[A-Za-z]{2,})$");
                Matcher matherEmail = patternEmail.matcher(email);

                //patron para validar contraseña segura
                Pattern patternClave = Pattern
                        .compile("^(?=\\w*\\d)(?=\\w*[a-z])\\S{6,}$");
                Matcher matherClave = patternClave.matcher(password);
```

```
// validacion si email y password si no estan con los validadores
if (TextUtils.isEmpty(email))
{
    Toast.makeText( context: AddActivity.this,  text: "El Campo correo no puede estar vacio.",
            Toast.LENGTH_SHORT).show();
    return;
}
// validacion de caracteres con esther para email
if (!matherEmail.find())
{
    Toast.makeText( context: AddActivity.this,  text: "Ingrese Un Correo Valido.",
            Toast.LENGTH_SHORT).show();
    return;
}
// validacion de password
if (TextUtils.isEmpty(password))
{
    Toast.makeText( context: AddActivity.this,  text: "El Campo password no puede estar vacio.",
            Toast.LENGTH_SHORT).show();
    return;
}
// validacion de caracteres con esther para clave
if (!matherClave.find())
{
    Toast.makeText( context: AddActivity.this,  text: "Ingrese Un Password Segura(1 mayus, 1 minus, 1 num, 6 caract minimo).",
            Toast.LENGTH_SHORT).show();
    return;
}

// creacion del usuario
mAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener( activity: AddActivity.this, (task) -> {
            if (task.isSuccessful()) {
                // Sign in success, update UI with the signed-in user's information
                Log.d(TAG,  msg: "createUserWithEmail:success");
                Intent intent = new Intent( packageContext: AddActivity.this, MainActivity.class);
                startActivity(intent);
                Toast.makeText( context: AddActivity.this,  text: "Registro Exitoso.",
                        Toast.LENGTH_SHORT).show();
```

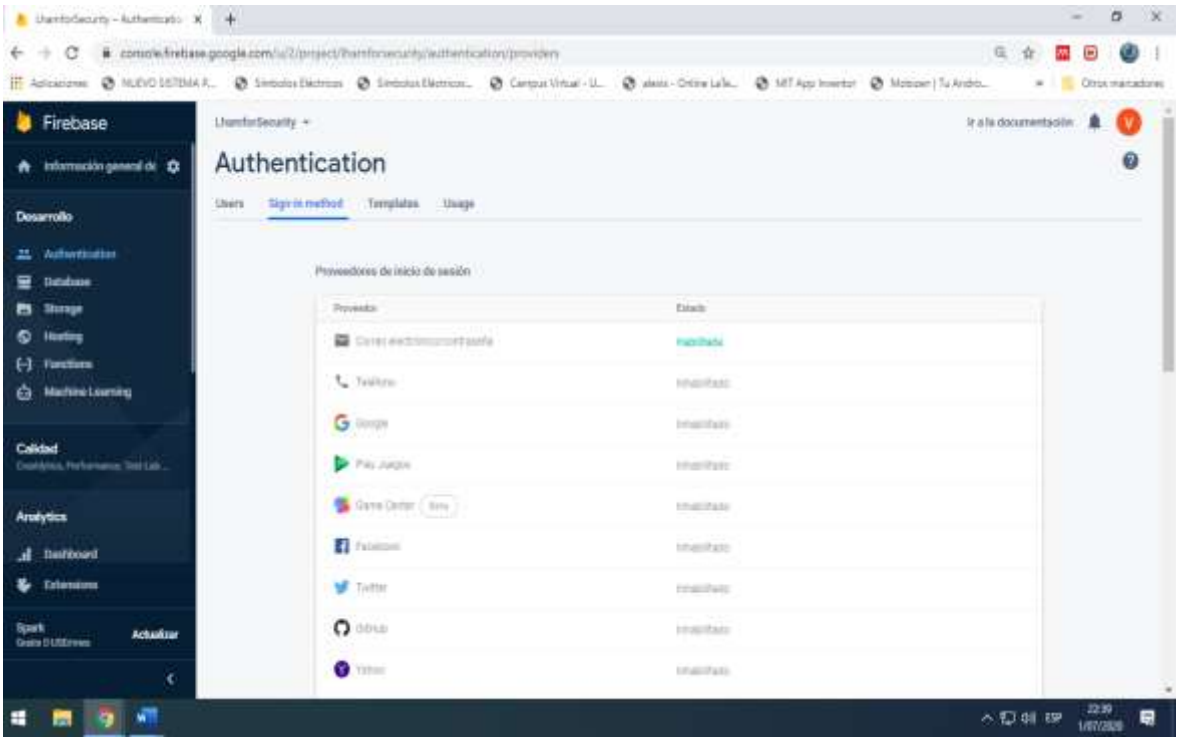
```
                            FirebaseUser user = mAuth.getCurrentUser();
                            updateUI(user);
                        } else {
                            // If sign in fails, display a message to the user.
                            Toast.makeText( context: AddActivity.this,  text: "Error a Registar.",
                                    Toast.LENGTH_SHORT).show();
                            Log.w(TAG,  msg: "createUserWithEmail:failure", task.getException());
                            Toast.makeText( context: AddActivity.this,  text: "Authentication falied.",
                                    Toast.LENGTH_SHORT).show();
                            updateUI( user: null);
                        }
                    });
        });
    }

    private void updateUI(FirebaseUser user) {
        if (user != null)
        {
            Intent intent = new Intent( packageContext: AddActivity.this, MainActivity.class);
            startActivity(intent);
            finish();
        }
    }
}
```

Anexo I. Integración de hardware con el software

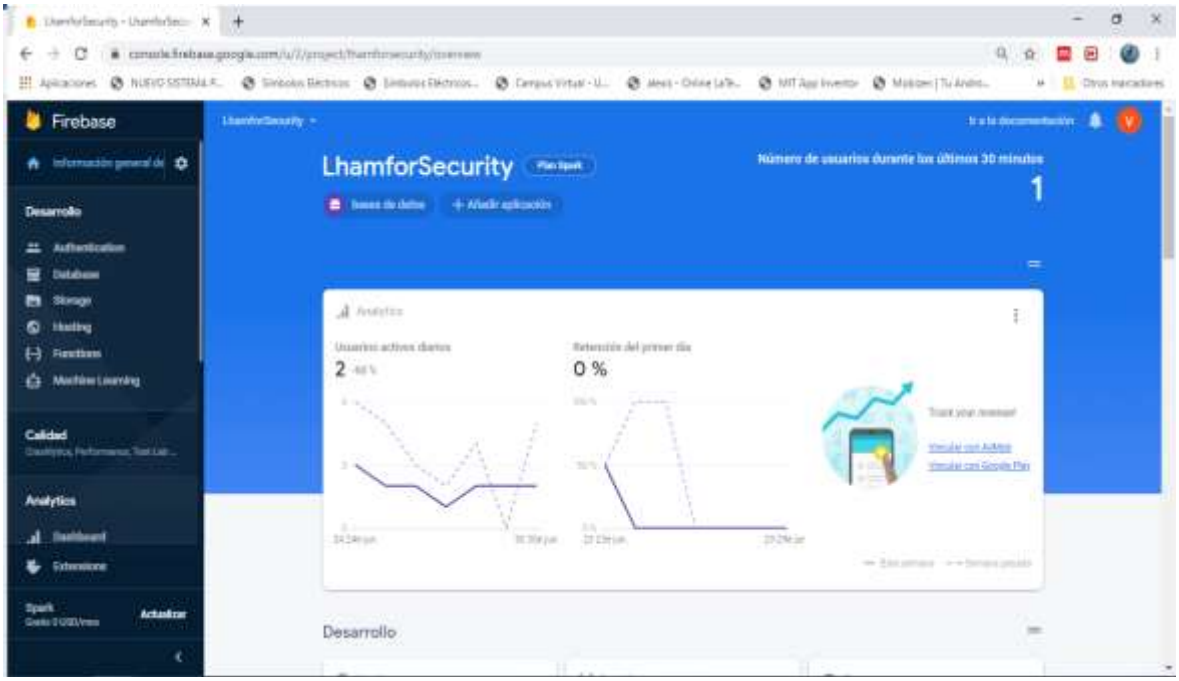
Anexo J. Usuarios en Firebase Authentication



Anexo K. Método utilizado para el registro de usuarios



Anexo L. Candidad de usuarios diarios en la base de datos



Anexo M. Codificación completa del proyecto