UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD DE CIENCIAS PURAS Y NATURALES

CARRERA DE INFORMÁTICA

TESIS DE GRADO

SISTEMA DE CONTROL DE INGRESOS NO AUTORIZADOS A UN AMBIENTE BASADO EN ANDROID, RASPBERRY PI E INTERNET DE LAS COSAS

Tesis de Grado para obtener el Título de Licenciatura en Informática

Mención Ingeniería de Sistemas Informáticos

POR: GABRIEL CASAS MAMANI

TUTOR METODOLÓGICO: M.Sc. ALDO VALDEZ ALVARADO

ASESOR: PhD. YOHONI CUENCA SARZURI

LA PAZ – BOLIVIA
2016

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

DEDICATORIA

A mis padres Modesto Javier y Yola, por el apoyo incondicional, la confianza, y el esfuerzo que hicieron para darme una profesión, a mis hermanos que gracias a su apoyo se hizo realidad este trabajo. A mis abuelos, su sabiduría y sus enseñanzas fueron valiosas en mi vida.

Gabriel Casas Mamani

AGRADECIMIENTOS

A Dios, por las bendiciones que cada día me da, por brindarme salud, conocimiento, por cuidar y mantener unida a mi familia, por guiarme en el buen camino en momentos importantes de mi formación como persona.

A mi tutor metodológico M.Sc. Aldo Ramiro Valdez Alvarado, por aplicar sus valiosos conocimientos en las correcciones de este trabajo, por ser ejemplo de profesional dedicado a seguir. A mi asesor PhD. Yohoni Cuenca Sarzuri, por su incondicionalidad para brindar apoyo en la redacción y corrección del presente trabajo. De ambos apreciación bastante el ánimo y el apoyo que me brindaron.

A mis queridos padres por demostrar siempre su amor y su cariño, brindando apoyo y orientación en las decisiones de cada integrante de la familia, a mis hermanos por confiar siempre en mí dándome todo su apoyo, a mis tíos y primos de quienes no faltaron palabras de aliento.

A todos, gracias.

RESUMEN

El Internet de las Cosas es el concepto que se refiere a la conexión de objetos habituales a internet a través de dispositivos de hardware. Esta propuesta busca conectar más cosas u objetos que personas a internet, ampliando las posibilidades de negocios y el uso de mejores tecnologías.

Para brindar conectividad a objetos cotidianos se cuenta con diferentes dispositivos de hardware mismos que por sus costos bajos son fácilmente accesibles. Estos dispositivos hacen uso de diferentes protocolos para efectuar su conexión a internet, estos emergen como solución para la comunicación Máquina a Máquina.

Gran parte del Software, Hardware y los Protocolos, que son usados en el Internet de las Cosas, pertenecen a Open-Source, lo que facilita el uso de estas tecnologías.

El presente trabajo muestra la integración de diferentes dispositivos de hardware, empleado diferentes lenguajes de programación, se hace uso de un protocolo de comunicación basado en Internet de las Cosas y se aplica conceptos de Bases de Datos no relacionales, con la finalidad de obtener un sistema que permita controlar y notificar aquellos ingresos no autorizados a un ambiente.

El desarrollo del sistema esta guiado por la refactorización del Modelo en V, mismo que es planteado en base al Método para el Diseño de Sistemas de Control Automático, y la Metodología Mobile-D. El nuevo Modelo en V, pretende ser mucho más eficiente cuando se va a desarrollar sistemas comprendidos por Hardware, Software y aplicaciones móviles.

La integración del sistema resultante es capaz de hacer uso de la tecnología de Identificación por radio frecuencia, mismo que permite la identificación de personas antes de ingresar a un ambiente. La autorización de los ingresos se determina en base a restricciones que se asigna a cada persona registrada en el sistema, estas restricciones están en función del lugar, día, y hora, que se identifica a una persona.

Palabras Clave: Internet de las Cosas, Identificación por Radio Frecuencia, Raspberry Pi, Autorización, Protocolo de Comunicación.

ABSTRACT

The Internet of Things is the concept that refers to the connection of habitual objects to the internet through hardware devices. This proposal seeks to connect more things or objects than people to the Internet, expanding business opportunities and the use of better technologies.

To provide connectivity to everyday objects, there are different hardware devices that are easily accessible because of their low costs. These devices make use of different protocols to make their connection to the Internet, these emerge as a solution for the communication Machine to Machine.

Much of the Software, Hardware and Protocols, which are used in the Internet of Things, belong to Open-Source, which facilitates the use of these technologies.

The present work shows the integration of different hardware devices, using different programming languages, using an Internet-based communication protocol of the Objects and applying non-relational database concepts, in order to obtain a system Which allows control and notification of unauthorized income to an environment.

The development of the system is guided by the refactoring of the Model-V, which is based on the Automatic Control Systems Design Method and the Mobile-D Methodology. The new Model-V aims to be much more efficient when developing systems comprised of hardware, software and mobile applications.

The integration of the resulting system is able to make use of radio frequency identification technology, which allows the identification of people before entering an environment. The authorization of income is determined based on restrictions that is assigned to each person registered in the system, these restrictions are based on the place, day, and time, which identifies a person.

Keywords: Internet of Things, Radio Frequency Identification, Raspberry Pi, Authorization, Communication Protocol.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

MARCO INTRODUCTORIO

1.1. Introducción

El modo en que la tecnología y las telecomunicaciones se expanden, llegando a muchos más lugares, logrando cambiar el modo de comunicación no solo de personas, sino que ahora la comunicación es de los objetos, cada persona poseerá un estimado de siete dispositivos conectados a la red para 2018 a los comunes dispositivos móviles se irán sumando los portables, como las pulseras que hacen seguimiento del ejercicio, los relojes inteligentes y el propio internet de las cosas (Arantxa, 2016) ahora también se ven, vehículos no tripulados, ciudades inteligentes, entre otros.

Muchos dispositivos han sido empleados para fines de comodidad, seguridad, automatización incluso para controlar trabajos que implican riesgos para la integridad del ser humano, todos estos que han sido creados hasta ahora, tienen cualidades diferentes y específicas, como detectar presencia, calor, humedad, pronosticar eventos naturales que pueda causarnos daño, alertar oportunamente de ciertas anomalías en nuestra anatomía como los marcapasos que ahora están conectados a internet para notificar a otras personas y recibir ayuda oportuna.

Se pueden utilizar dispositivos para monitorear, controlar o mejorar el funcionamiento de muchas de las actividades que realizamos habitualmente, y si estos además están siempre conectados a la red mandando datos periódicamente de toda la actividad que realizan, pueden llegar a ser muy aprovechables.

Todo el aprovechamiento que se le da a estos dispositivos, que ya están a nuestro alcance, nos permitirá tener confort, seguridad, información de actividades cotidianas a través de sensores que manden datos, automatización de aparatos, control de la salud entre muchas otras posibilidades.

La seguridad es un punto, que como muchas otras actividades que se llevan a cabo en nuestro cotidiano vivir, tiene un significado importante, más aun si hablamos de proteger bienes a los cuales solamente nosotros o un grupo de personas, tiene el permiso de acceder.

Monitorear contantemente lo que sucede en los ingresos a nuestros bienes (hogar, oficinas, edificios, centros de trabajo, hospitales, colegios, entre otros) nos da la sensación de seguridad que tanto se busca, esto se potencia más cuando recibimos notificación e información periódica a nuestro dispositivo móvil y podemos actuar desde un lugar cercano o lejano para evitar que se produzcan atentados, intrusiones que nos vayan a provocar un perjuicio.

1.2. Antecedentes

En el año 1926, Nicola Tesla, escribió un artículo en la revista Colliers, donde explico: "que se podrá controlar todas las cosas que son partículas y los instrumentos que nos rodean", esto se puede comparar hoy en día con un Smartphone. Con el trabajo de Nikola Tesla se conformaron la base de las comunicaciones inalámbricas y la radio (Ballestín, 2015).

En 1975, ARPAnet comenzó a funcionar como Red, sirviendo como base para unir centros de investigación militares, Universidades, y se trabajó en desarrollar protocolos más avanzados para diferentes tipos de computadoras y cuestiones específicas (Cadenas, 2011) en ese momento ya habían 61 nodos conectados.

En 1999 Kevin Ashton, impartió una conferencia en Procter & Gamble[1] donde habló por primera vez del concepto de Internet de las Cosas, "Las máquinas toman decisiones por sí mismas. Desde objetos como ropa, electrodomésticos o coches, incluso ciudades inteligentes que, asegura, serán más eficientes y sostenibles**.** Más de 3,8 millones de objetos están ya conectados a la nube; en 2020 serán 25.000 millones" (Hernández, 2015**)**.

[1] Protector & Gamble también conocida como (P&G) es una empresa estadounidense multinacional de bines de consumo con localización en el centro de Cincinnati, Ohaio.

En 2005 la agencia de las naciones unidas, Unión Internacional de Telecomunicaciones (ITU) publica el primer estudio sobre el tema, el informe da un vistazo a la siguiente etapa de "siempre en" las comunicaciones como RFID[2] y la computación inteligente promete un mundo de red y dispositivos interconectados que proporcionan contenido e información sea cual sea la ubicación del usuario pertinente (ITU, 2011).

Los dsPIC's[3] son el penúltimo lanzamiento de microchip, comenzando su producción a gran escala a finales de 2004, recibe el nombre de DSP (Procesador Digital de Señales) un circuito integrado que contiene un procesador digital y un conjunto de recursos complementarios capaces de manejar digitalmente las señales del mundo real (Pavón y Cruz 2010).

En 2005, un grupo de estudiantes y profesionales del instituto de Diseño Interactivo Ivrea en Italia, empezó a desarrollar una plataforma de hardware en open source por que las placas del mercado eran demasiado caras para experimentar. Seis años después, Arduino supone una evolución en la forma en que se crea y usa la tecnología (Fernández, 2011).

Arduino es una plataforma física Open-Source basada en una placa electrónica simple, con la lógica de "Hardware y Software fácil de utilizar" además de un entorno de desarrollo para crear software destinado a esta placa. Esta puede ser utilizada para le lectura de diferentes sensores, control de señales que activan o desactivan actuadores electrónicos, entre muchas otras aplicaciones, al ser Arduino una hardware open-source puede ser replicado bajo los mismos esquemas de Arduino, existiendo actualmente diversos modelos: Uno, Leonardo, Mega, Nano, entre otros (Arduino, 2014).

En el año 2009, la fundación RaspberryPi fue fundada en Caldecote, Reino Unido como una asociación caritativa, la fundación surge con el objetivo de desarrollar el uso y entendimiento de los ordenadores en los niños (Dablarui, 2013) la producción en masa comienza en el 2012 con posibilidades a un mayores para abrir más las puertas sobre internet de las cosas, es

[2] Identificación por radiofrecuencia.
[3] Es un nombre genérico que se utiliza para referirse a los controladores digitales de señales (DSC) que ha diseñado Microchip Technology Inc.

capaz de controlar objetos como lo hacía Ardino pero contando con un sistema operativo e interfaz gráfica.

A partir de estas tecnologías se solucionaron cada vez más los problemas cotidianos, automatizándolos y ayudándonos a tener un mejor control de nuestro entorno. Es el caso, de la necesidad de contar con seguridad en lugares donde se desea controlar el ingreso a personas no autorizadas, impulsa a que se diseñen soluciones para controlar este tipo de situaciones.

En épocas no muy lejanas los controles de accesos eran simples puertas, que luego fueron reforzadas por seguros, chapas y por guardias de seguridad. Estos sistemas de seguridad fueron efectivos durante un tiempo, pero empezaron a ser víctimas de personas inescrupulosas, con el objetivo de ingresar a lugares a los que no habían sido autorizados con fines delictivos (Pricer, 2011), las primeras alarmas para evitar estos hechos, emitían ruido o alguna señal perceptible para notificar la intrusión.

Este tipo de sistemas de seguridad permitían alertar de una intrusión, solamente dentro de un perímetro delimitado por lo que debería existir una persona cerca para actuar y evitar la violación de la seguridad, lo cual repercute en casos no favorables, los sistemas más avanzados permiten notificaciones a distancia, informar a centros policiales cercanos, actuadores como los porteros electrónicos que aseguran puertas, sistemas de vigilancia de circuito cerrado que permitan grabar lo que sucede en los ambientes, para poder actuar en caso de algún acto delictivo o algún caso que requiera de intervención para resguardad la seguridad.

En la carrera de Informática de la Universidad Mayor de San Andrés se realizó el trabajo de investigación como tesis de grado, con el tema: “Control inteligente de seguridad en viviendas mediante agentes móviles” en el cual se logra realizar un sistema de control inteligente de seguridad de una vivienda, que se basa en circuitos con la placa Arduino UNO, modelos de sistemas de control con Simulink, agentes móviles y base de datos PostgreSQL (Machaca, 2015).

En la carrera de Informática de la Universidad Mayor de San Andrés se realizó el trabajo de investigación como tesis de grado, con el tema**:** "Sistema demótico para un hogar basado en software y hardware libre" el cual demuestra la posibilidad de implementar un sistema domótica no muy costoso, aplicando software y hardware libre como Arduino y php, que permite automatizar los aparatos eléctricos, electrodomésticos, y la seguridad de un hogar (Chávez, 2015).

En la Escuela superior de ingeniería informática de la universidad Politécnica de Valencia se realizó el trabajo de investigación con el tema: "Introducción de aspectos de seguridad en la internet de las cosas en el ámbito de las viviendas inteligentes", Realizado un diseño e implementación para asignar permisos con los que controlar objetos ofrecidos por la IoT (Ballestín, 2015).

1.3. Planteamiento del problema

La delincuencia y los atentados hacia la propiedad, son un problemas que afectan a toda la sociedad, según la encuesta de victimización y percepción - 2013 sobre "Lugar de los hechos delictivos en nueve ciudades capitales y El Alto - 2013" realizada por el Observatorio Nacional de Seguridad Ciudadana (ONSC), indica que en ciudades capitales el 21.9% y en la ciudad de El Alto el 15,7%, de los hechos delictivos suceden en la puerta de viviendas (ONSC, 2013).

La seguridad en el ingreso a ambientes es un problema también en empresas, instituciones y en lugares donde se debe controlar el mismo a la afluencia de personas, ya que es en estos que se veden tener en cuenta las intrusiones y la concesión de permisos a personas ajenas, con el fin de controlar su autorización.

Una organización que tenga bien instalado y controlado un sistema de control de acceso adquiere más ventajas en materia de seguridad y de productividad, (Ortiz, 2014) un mal control de los accesos provoca perjuicios a los bienes de la organización.

Es por esto que se quiere conseguir realizar un sistema de control para el ingreso a un ambiente, que sea de un coste económico relativamente bajo, ya que el medio local no cuenta aún con tecnología avanzada que es limitada por la falta de presupuesto para el desarrollo tecnológico, se sabe que "Cerca del 58% de las víctimas de robo en la vivienda o el negocio son personas de bajos ingresos económicos, mientras que el 14% pertenecen a estratos de ingresos económicos altos" (Oporto, 2012).

1.3.1. Problema central

Teniendo en cuenta la problemática, y considerando que para resguardar la seguridad de ambientes es necesario controlar los ingresos que se efectúen en diferentes ambientes, se sugiere la siguiente interrogante:

¿Cómo llevar a cabo el control y notificación de algún ingreso no autorizado a un ambiente?

1.3.2. Problemas secundarios

A partir del problema central se identificaron diversos problemas que a continuación se dan a conocer:

- Pocos ambientes cuentan con un control de acceso seguro, lo que provoca ingresos no autorizados.
- La forma de ingreso a un ambiente no está monitoreado constantemente, por lo que las intrusiones escapan del control.
- El uso de mecanismos como porteros electrónicos o chapas, puede ser vulnerado sin ser notificado en tiempo real.
- No contar con información de lo que sucede en los ingresos, provoca desconfianza sobre la seguridad.
- En ambientes con gran afluencia de personas (edificios, oficinas, empresas), controlar quienes ingresan se hace una tarea complicada.

1.4. Definición de objetivos

1.4.1. Objetivo general

Desarrollar un sistema de control de ingresos, que determine aquellos autorizados a un ambiente.

1.4.2. Objetivos específicos

- Identificar un dispositivo de identificación adecuada que permita determinar la autorización en los ingresos.
- Monitorear constantemente los ingresos autorizados a un determinado ambiente, a través de comunicación por internet.
- Identificar el modo de autenticación más confiable para lograr una autorización correcta, y notificar al dispositivo móvil.
- Informar de manera oportuna los ingresos no autorizados.
- Facilitar la tarea en el control de ingresos cuando se va a trabajar con una afluencia de personas.

1.5. Hipótesis

El control de ingresos, utilizando Android, Raspberry Pi e Internet de las Cosas, permite controlar y notificar el ingreso autorizado a un ambiente.

1.5.1. Operalización de variables

Variable dependiente: Control y notificación de ingreso no autorizado a un ambiente.

Variable independiente: Android, RFID, Raspberry Pi, Internet.

Variable interviniente: Recopilación de datos con Internet de las Cosas.

1.6. Justificación

1.6.1. Justificación económica

Un sistema que ofrezca el servicio de controlar los ingresos repercute en una inversión que por lo general tiende a elevarse, ya que varios de estos sistemas requieren de aparatos costosos, software y hardware que realiza tareas específicas, por lo que si se requiere ampliar sus capacidades de control se necesita de una inversión extra.

Los sistemas que cuentan con capacidad de controlar los ingresos a un grupo de personas autorizadas para ingresar a un ambiente restringido, están disponibles en el mercado, pero requieren de un financiamiento costoso.

En el medio local el acceso a la tecnología no se da en su totalidad por los costos elevados de adquisición que representan las nuevas tecnologías.

Es posible acceder a la tecnología, pero a un no se manifiesta como algo habitual, para que se manifieste, se debe priorizar el acceso a diferentes tecnologías sin que el factor económico sea un problema.

El presente trabajo pretende conseguir un sistema de control con un coste económico accesible para los usuarios, que proporcione el control adecuado con un fácil manejo pero sobre todo que proporcione la seguridad en los accesos.

1.6.2. Justificación social

Toda persona necesita sentir seguridad en su entorno social, más aun si se trata de habitar un ambiente propio donde la seguridad es prioridad para gozar de tranquilidad, esta no solo es preocupación de una persona, es aún más importante cuando de proteger a un agrupo de personas se trata (familia, personal, pacientes en hospitales entre otros).

Un medio de acceso, a un grupo de personas autorizadas, garantiza de alguna manera que se pueda evitar intrusiones de personas ajenas a ambientes restringidos, si este medio de acceso

se puede controlar y además interpretar los datos recogidos permite tomar decisiones para crear mecanismos de acceso mucho más seguros, todo gracias a que los datos son confiables ya que son obtenidos de un entorno real.

Que la sociedad cuente con un sistema que tenga estas propiedades, ejerce un nivel de independencia y tranquilidad puesto que el sistema notifica si alguna intrusión o acceso no autorizado sucede, sin importar el medio físico en el que el propietario se encuentre con respecto al sistema.

1.6.3. Justificación científica

El presente trabajo de investigación contempla estudios sobre: electrónica, informática, comunicación vía internet, recolección de datos.

Con el estudio de estos se pretende mostrar cuan potencial y prometedor puede llegar a ser la combinación de tecnologías, la internet de las cosas, consiste en que las cosas tengan conexión a internet en cualquier momento y lugar.

En un sentido más técnico consiste en la integración de sensores y dispositivos en objetos cotidianos que queden conectados a internet a través de redes fijas inalámbricas (ECIXGROUP, 2015).

Se observa que diariamente se recure al uso de la tecnología para solucionar uno o varios problemas, en diferentes situaciones y para diferentes casos particulares o generales, siendo esta una herramienta con un impacto considerable que necesita estudiarse cada día más para poder desarrollar nuevas y mejores aplicaciones.

La combinación de más de una tecnología permiten ampliar más las posibilidades en las aplicaciones desarrolladas, el presente trabajo incursiona en la combinación de tecnologías para poder logar una integración que muestre una de las muchas posibilidades de este tipo de estudio.

1.7. Alcances y Límites

1.7.1. Alcances

El sistema de control propuesto, tendrá la capacidad de identificar los accesos cumpliendo las siguientes funcionalidades:

- El sistema identificara cuando una persona esté cerca del medio de acceso (puerta, portón, garaje, reja) a través del sensor piroeléctrico PIR (Passive Infra Red).
- Se podrá autenticar los accesos autorizados mediante la identificación por radio frecuencia RFID.
- Será posible la gestión de los usuarios que cuenten con el acceso autorizado, que permita conceder acceso a nuevos usuarios o restringir la autorización, a través de Raspberry Pi.
- Con los datos obtenidos se podrá analizar la concurrencia de los usuarios, para proteger más el ambiente en ausencia de ingresos.
- Notificar a los dispositivos Android asociados, si se presenta una intrusión en los accesos, o en su defecto notificar cualquier ingreso.
- Desde el dispositivo Android se podrá bloquear la autenticación de radiofrecuencia, si alguna situación lo requiere, para restringir los ingresos.

1.7.2. Límites

La propuesta contempla casos generales, por lo que cuenta con limitaciones que se describen a continuación:

- Se limita a controlar los accesos por puertas, por lo que el control en ventanas, techos, entre otros escapa del control.
- Los dispositivos asociados deben contar con el sistema operativo Android y acceso a internet.

- Se limita a dos medios de recopilación de datos: sensor piroeléctrico PIR, identificación por radio frecuencia RFID.
- No se contempla la situación en que uno de los dispositivos (dispositivo Android o Raspberry Pi) este sin conexión a internet y no se pueda llegar a cumplir con la notificación correspondiente, en el momento oportuno.

1.8. Aportes

1.8.1. Practico

El aporte del presente trabajo de investigación, contempla información sobre la recopilación continua de datos generada por objetos conectados a internet, el tratado de estos datos y el uso que se les puede dar.

Información sobre el uso de dispositivos para la identificación de personas con el fin de autorizar y negar permisos, que posteriormente dará una base para recrear y mejorar el modo de otorgar y negar permisos a un grupo de personas.

Explicar el medio de comunicación, la manera en la que los datos serán transportados utilizando principios de la Internet de las Cosas, y las consideraciones que se deben tomar al trabajar con envió de información generada por objetos conectados a internet.

Mostrar el modo físico de conexiones entre diferentes sensores y hardware que facilite que los objetos funcionen y transmitan la información captada para su posterior interpretación.

1.8.2. Teórico

Aportar con investigación que explique los antecedentes teóricos sobre el internet de las cosas, las posibilidades que tiene y el potencial que implica tener objetos conectados a internet.

Bases teóricas que expliquen el funcionamiento de las comunicaciones entre objetos y protocolos para que se lleven a cabo la transferencia de datos.

Recopilación de información que permita comprender el funcionamiento de la placa Raspberry Pi, y como es que permite la interacción con el medio a través de sensores.

Investigación teórica sobre los mecanismos de autenticación y las diferentes formas de identificación, el funcionamiento que efectúan, la seguridad que estos proporcionan, con el fin de poder discriminar cuál es la que mejores posibilidades ofrece y la que más seguridad tiene.

1.9. Metodología

Para la investigación y obtención de información respectiva se hace uso de la metodología de investigación Científica, según Del Rio (2011) está orientada a la obtención de nuevos conocimientos y su aplicación para la solución de problemas, el cual contempla las siguientes etapas para seguir con la investigación:

- Concepción de la idea: que puede surgir de inquietudes personales, por encargo, problemas identificados, lecturas, otras investigaciones.
- Planteamiento el problema a investigar y de los objetivos: que es la reformulación de la pregunta inicial después de la exploración.
- El marco teórico y conceptual: que es la revisión documental (información existente y relativa al tema: teorías, hipótesis, objetos de estudio, estudios, entre otros).
- Diseño del modelo de análisis: consta de la metodología, que son las unidades de investigación y los instrumentos de recogida de información, que es la definición de las técnicas o herramientas para la exploración y/o tratamiento de los datos.
- Recogida de los datos: aplicación del instrumento de recogida de información en las unidades de análisis pertinentes.
- Explotación de los datos: es la explotación/análisis de los datos (SPSS[4], transcripciones, vaciados, etc.)

[4] SPSS, es un programa informático muy usado en las ciencias exactas, sociales y aplicadas, además de las empresas de investigación de mercado.

- Análisis de la información: Lectura, interpretación, y selección de los datos más significativos en función de los objetivos, las hipótesis y/o las preguntas de investigación.
- Elaboración del informe: Consiste en exponer el procedimiento seguido, presentación de los resultados.

La metodología de ingeniería de software y hardware será el Modelo en V, es una variación del modelo en cascada que muestra cómo se relacionan las actividades de prueba con el análisis y el diseño (Rojas, 2010), es denominada también de cuatro niveles los cuales son:

- Nivel 1: está orientada al cliente, el inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo, se componen de análisis de requerimientos y especificaciones, se traduce en un documento de requisitos y especificaciones.
- Nivel 2: se dedica a las características funcionales, del sistema propuesto, puede considerarse el sistema como una caja negra, y caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final, se traduce en un documento de análisis funcional.
- Nivel 3: define los componentes de hardware y software del sistema final a cuyo conjunto se denomina arquitectura del sistema.
- Nivel 4: es la fase de implementación en la que se desarrollan los elementos unitarios o módulos del programa.

Las herramientas que se emplearan en el presente trabajo de investigación son: de software como: Python, Java, lenguaje CQL y de hardware como: dispositivo Android, Raspberry Pi, RFID, Conexiones a internet.

CAPÍTULO 2

MARCO TEÓRICO

2.1. Sistema de control automático

El control automático de procesos es una de las disciplinas que se ha desarrollado a una velocidad vertiginosa, dando la base a lo que hoy algunos autores llaman la segunda revolución industrial. El uso intensivo de las técnicas del control automático de procesos tiene como origen la evolución y la tecnificación de las tecnologías de medición y control aplicada al ambiente industrial (Abarca, 2016).

En los años recientes los sistemas de control han asumido, un papel cada vez más importante en el desarrollo de avances en la civilización moderna y la tecnología. Prácticamente, cada aspecto de las actividades de nuestra vida diaria está afectada por algún tipo de sistema de control, estos sistemas se encuentran en gran cantidad en todos los sectores, tales como control de calidad, líneas de embalaje automático, control de máquina herramienta, tecnología espacial, control de seguridad, robótica y muchos otros más. Aun el control de inventarios y los sistemas económicos y sociales se pueden visualizar a través de la teoría de control automático (Kuo, 2016).

Un sistema de control automático es una interconexión de elementos que forman una configuración denominada sistema, de tal manera que el arreglo resultante es capaz de controlarse por sí mismo de manera automática. (Hernández, 2010)

Un sistema de control automático que cuenta con componentes que son susceptibles a ser monitoreados, y capaces de recibir señales externas a manera de entrada, para generar una señal de salida a manera de respuesta, expresa una relación causa y efecto, esto presenta un problema en el control que consiste en seleccionar y ajustar un conjunto especifico de elementos tal que, al interconectarse, el sistema resultante deberá comportarse de una marera especifica.

La eliminación de errores y un aumento de la seguridad de los procesos es la contribución importante del uso y aplicación de técnicas de control automático. En este punto es importante destacar que anterior a la aplicación de estas técnicas, era el hombre el que aplicaba sus capacidades de cálculo e incluso su fuerza física para la ejecución del control de un proceso. En la actualidad gracias al desarrollo y aplicación de las técnicas modernas de control, un gran número de tareas y cálculos asociados a la manipulación de las variables han sido delegados a computadoras, controladores y accionamientos especializados para el logro de las aplicaciones del sistema de control automático(Abarca, 2016).

La finalidad de un sistema de control automático es conseguir, mediante las señales externas, un dominio sobre las señales de salida, devueltas como datos, de manera que estas sean expresadas y brinden información del comportamiento del sistema.

Según UPC (2016) el sistema de control automático ideal debe ser capaz de conseguir su objetivo cumpliendo los siguientes requisitos:

- Garantizar la estabilidad y particularmente ser robusto frente a perturbaciones y errores.
- Ser tan eficiente como sea posible, según un criterio preestablecido. Normalmente este criterio consiste en que la acción del control sobre las señales de entrada sea realizable, evitando comportamientos bruscos e irreales.
- Ser fácilmente implementable y cómodo de operar en tiempo real con la ayuda de un ordenador.

Según UPC (2016) los elementos básicos que forman parte de un sistema de control y permiten su manipulación son los siguientes:

- Sensor. Permite conocer los valores de las señales externas, medidas por el sistema.
- Controlador. Utilizando los valores determinados por los sensores y la consigna impuesta, calcula la acción que debe aplicarse para modificar las variables de control en base a cierta estrategia.

- Actuador. Es el mecanismo que ejecuta la acción calculada por el controlador y que modifica las variables de control.

La Figura 2.1 ilustra el esquema del funcionamiento de un sistema de control genérico.

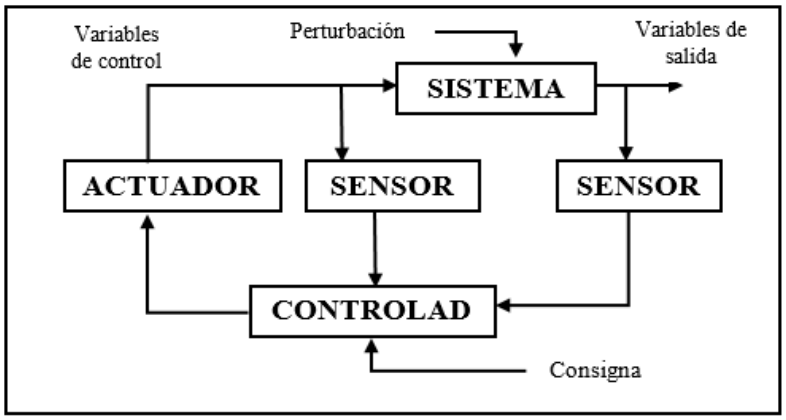


Figura 2.1: Esquema general de un sistema de control
Fuente: (UCP, 2016)

2.2. Metodologías

2.2.1. Método para el diseño de sistemas de control automático

Un buen diseño del sistema de control automático, mediante ingeniería nos permite obtener la configuración, especificaciones, e identificación de parámetros clave para satisfacer una necesidad real. Por lo que es imprescindible contar con un método a seguir para llevar a cabo el mejor diseño posible del sistema.

Según Mírez (2013) para realizar el diseño del sistema seguiremos los siguientes pasos:

- El primer paso en el proceso de diseño consiste en establecer los objetivos del sistema.
- El segundo paso es identificar las variables que deseamos controlar.
- El tercer paso es escribir las especificaciones en función a la precisión que se desea alcanzar. Esta precisión de control requerida conducirá entonces a la identificación de un sensor para medir la variable controlada.

Una vez se tenga claro lo mencionado se procede al primer intento para configurar un sistema que tenga el comportamiento de control deseado.

La configuración del sistema normalmente consiste de un sensor, el proceso bajo control, un actuador, y un controlador. El siguiente paso consiste en identificar un candidato como actuador, el cual dependerá del proceso, por lo que la actuación escogida deberá ser capaz de ajustarse de forma efectiva el comportamiento del proceso (Mírez, 2013).

En general el actuador es un dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o "actuar" otro dispositivo mecánico (Vildósola, 2016) el recibe energía de naturaleza eléctrica y suministra otra energía de diferente naturaleza sometida a la variable de actuación.

El sistema de control puede llegar a ser muy complejo y con frecuencia es no-lineal. En todo caso lo que se requiere es una relación estática entre su entrada (M) y su salida (X) lo que se llamara la curva final de actuación (Páez, 2005).

Para el análisis de los sistemas de actuación es necesario utilizar graficas que representen su funcionamiento, por lo que se presentan interpretaciones del comportamiento de estos actuadores.

En la figura 2.2 se representa la gráfica de un sistema de actuación que se ubica en el primer cuadrante, se observa que existe saturación en ambos extremos y presenta pendiente positiva.

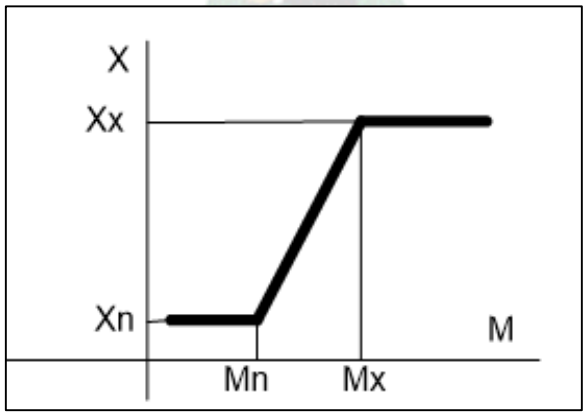


Figura 2.2: Curva de actuación con pendiente positiva
Fuente: (Páez, 2005)

En la figura 2.3 se representa la gráfica de un sistema de actuación que se ubica en el primer cuadrante, se observa que existe saturación en ambos extremos y presenta pendiente negativa.

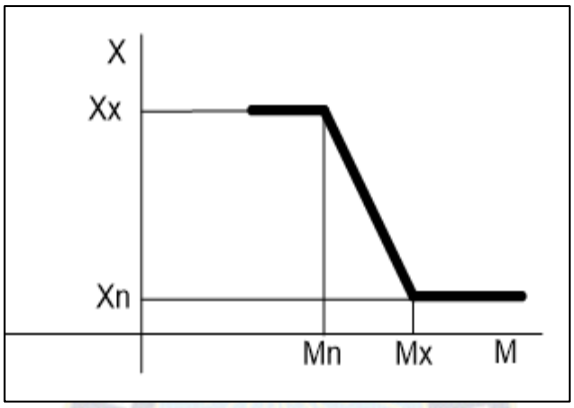


Figura 2.3: Curva de actuación con pendiente negativa
Fuente: (Páez, 2005)

En la figura 2.4 se representa la gráfica de un sistema de actuación que se ubica en el primer y cuarto cuadrante, se observa que existe saturación en ambos extremos y presenta pendiente positiva.

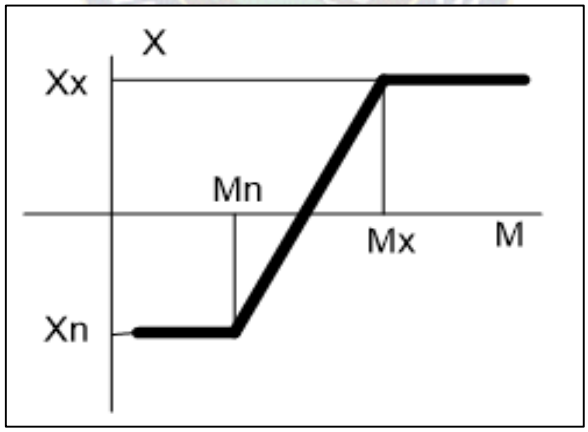


Figura 2.4: Curva de actuación con pendiente positiva
Fuente: (Páez, 2005)

En la figura 2.5 se representa la gráfica de un sistema de actuación que se ubica en el primer y cuarto cuadrante, se observa que existe saturación en ambos extremos y presenta pendiente negativa.

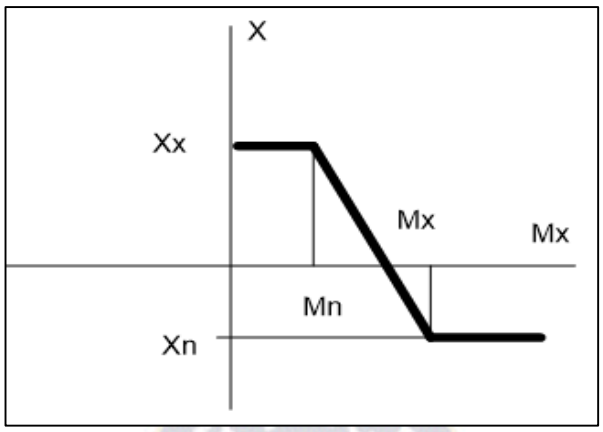


Figura 2.5: Curva de actuación con pendiente negativa
Fuente: (Páez, 2005)

El paso final consiste en seleccionar un controlador que deberá someterse al comportamiento del proceso, el cual recibirá las señales de los sensores, proporcionando así una respuesta que pueda ser interpretada, con los respectivos errores los cuales permitirán ajustar parámetros y conseguir el comportamiento deseado. "Si se puede conseguir el comportamiento deseado ajustando los parámetros se finalizara el diseño y se procederá a documentar los resultados" (Mírez, 2013).

2.2.2. Modelo en V

La metodología de ingeniería de software y hardware será el Modelo en V, es una variación del modelo en cascada que muestra cómo se relacionan las actividades de prueba con el análisis y el diseño (Rojas, 2010).

El modelo está representada en forma de V que se muestra en la figura 2.6, la codificación forma el vértice de la V, con el análisis y el diseño a la izquierda, las pruebas y el mantenimiento a la derecha.

La unión entre las fases de la izquierda y las pruebas de la derecha representan una doble información, por un lado sirve para indicar en qué fase del desarrollo se deben definir las pruebas correspondientes, por otro lado sirve para saber a qué fase de desarrollo hay que volver si se encuentran fallos en las pruebas correspondientes.

Por lo tanto el Modelo en V hace más explícita la parte de las iteraciones y repeticiones de trabajo que están ocultas en el modelo de cascada. Mientras el foco del modelo en cascada se sitúa en los documentos y productos desarrollados, el modelo en V se centra en las actividades de la corrección.

El lado izquierdo de la V representa las necesidades, y la creación de las especificaciones del sistema, el lado derecho de la V representa la integración de todos los pasos y su verificación.

2.2.2.1. Objetivos

- Minimizar los riesgos del proyecto: consiste en mejorar la transparencia del proyecto y control del proyecto, describir los resultados y responsabilidades de cada función. Permite una detección temprana de las desviaciones, riesgos y mejoras de gestión de procesos, reduciendo así los riesgos del proyecto.
- Mejoramiento y garantía de calidad: Es un modelo de procesos estándar, que asegura, obtener los resultados en la calidad deseada, que los resultados obtenidos sean completos y adecuados. Los datos obtenidos pueden ser comparados con otros datos.
- Reducción de los gastos totales durante todo el proyecto: Todo el proceso del ciclo de vida de un sistema lo podemos calcular y controlar mediante la aplicación de procesos estandarizados a si se reduce la dependencia de los proveedores.

2.2.2.2. Niveles del modelo en V

Los cuatro niveles lógicos representados en la figura 2.6 están comprendidas por fases teniendo cada una de ellas una fase correspondiente o paralela de verificación o validación. Esta estructura obedece a que para cada fase del desarrollo debe existir un resultado verificable. En la misma estructura se advierte también que la proximidad entre una fase del desarrollo y su fase de verificación correspondiente va decreciendo a medida que aumenta el nivel dentro de la V, se denominada también de cuatro niveles los cuales se describen a continuación (Martínez, 2012):

- Nivel 1: está orientada al cliente, el inicio y el fin del proyecto constituyen los dos extremos del ciclo, se componen de análisis de requerimientos y especificaciones, se traduce en un documento de requisitos y especificaciones.
- Nivel 2: se dedica a las características funcionales, del sistema propuesto, puede considerarse el sistema como una caja negra, y caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final, se traduce en un documento de análisis funcional.
- Nivel 3: define los componentes de hardware y software del sistema final a cuyo conjunto se denomina arquitectura del sistema.
- Nivel 4: es la fase de implementación en la que se desarrollan los elementos unitarios o módulos del programa.

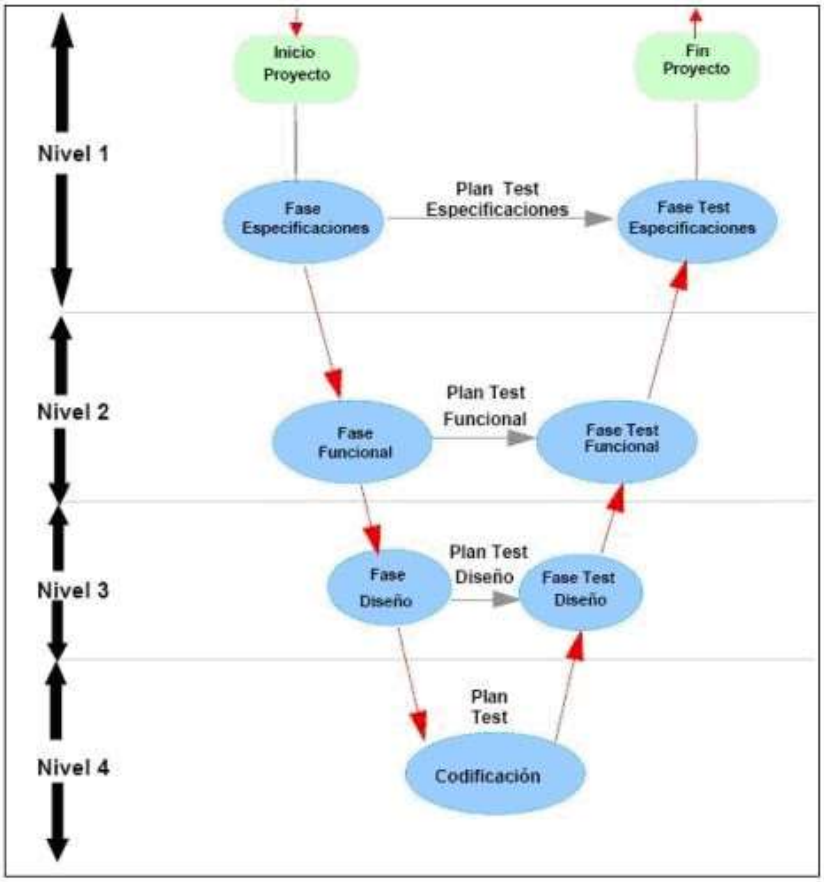


Figura 2.6: Niveles del Modelo en V
Fuente Martínez, 2012

2.2.2.3. Fases del modelo en V

Como se observa en la figura 2.6 el Modelo en V presenta cuatro niveles ya descritos, estos están compuestos por fases los cuales son siete, cada fase tiene un objetivo que debe ser cumplido para lograr el mejor resultado.

- Fase 1. Especificaciones: Se debe definir y documentar los diferentes requisitos del sistema a desarrollar, identificar los valores numéricos más concretos posibles.
  Especificación estructurada utilizando diferentes técnicas de diagramas para modelar el sistema nuevo.
- Fase 2. Funcional: También llamado de alto nivel. Su objetivo es obtener un diseño y visión general del sistema.
  Establecer un conjunto de módulos, desplegando las necesidades obtenidas en la fase de especificaciones, facilitando a la tarea de codificación y los modelos lógicos de los datos físicos.
- Fase 3. Diseño: Consiste en detallar cada módulo de la fase anterior. Cada módulo como resultado de la fase anterior es producida en la herramienta o lenguaje apropiado. Define los componentes de hardware y software del sistema final a cuyo conjunto se denomina arquitectura del sistema.
- Fase 4. Codificación: En esta fase se materializa el diseño en detalle, se implementan, desarrollan los elementos unitarios o módulos del programa.
- Fase 5. Test de Diseño: Se verifica cada módulo desarrollado de forma unitaria, comprobando su funcionamiento adecuado, generando documentos de las pruebas.
  Es la verificación del correcto funcionamiento del hardware y software, con los módulos producidos por las herramientas o lenguajes escogidos.
- Fase 6. Test Funcional: Es la fase de prueba de la funcionalidad en el que se integraran los distintos módulos que forman el sistema. Como en el caso anterior, ha de generarse un documento de pruebas.
  Por una parte, se debe comprobar en todo el sistema tenga el funcionamiento correcto, y por otra, en caso de tratarse con un sistema tolerante a fallos, se debe

verificar que ante la presencia de alguno, persista el funcionamiento correcto. Se comprueba el cumplimento de los requisitos establecidos.

- Fase 7. Test de Especificaciones: Se realizan las últimas pruebas pero sobre un escenario real, en su ubicación final, anotando una vez más las pruebas realizadas y los resultados obtenidos.

  Garantizar el mantenimiento del sistema, corrección de errores detectados en la fase, probando la aplicación del sistema en nuevos entornos.

2.2.2.4. Ventaja y desventaja del Modelo en V

- Ventajas

  Se trata de un proceso ideal, por su robustez, para proyectos pequeños, con equipos de una a cinco personas.

  También es ideal, por su claridad, para los integrantes del equipo que nunca ha programado siguiendo una metodología. Teniendo una mayor comodidad, en la relación entre las etapas de desarrollo y los distintos tipos de pruebas que facilitan la localización de fallos.

- Desventajas

  Cada fase tiene que estar respaldada por su documento correspondiente y test, se habla de una amplia documentación, se debe realizar dos procesos al mismo tiempo, es difícil que el cliente exponga explícitamente todos los requerimientos, por lo que se debe tener paciencia porque se obtendrá el producto al final del ciclo de vida, las pruebas pueden ser caras, a veces, no lo suficiente efectivas.

2.2.3. Mobile-D

El desarrollo de aplicaciones móviles sufre prácticamente los mismos problemas que la gran mayoría de desarrollo de software, aún que hay que tener en cuenta sus características como la corta duración de su desarrollo, la mayoría de proyectos se lleva a cabo por equipos

pequeños que requieren métodos en común para organizar sus tareas ya sea de un modelo ágil o un modelo más estricto y predictivo.

La aproximación de Mobile-D se ha apoyado en muchas otras soluciones bien conocidas y consolidadas: eXtreme Programing (XP), Crystal metodologies y Rational Unified Process (RUP). Los principios de la programación extrema se han reutilizado en lo que se refiere a las prácticas de desarrollo, la metodología Crystal proporcionaron un input muy valioso en términos de la escalabilidad de los métodos y el RUP es la base para el diseño completo del ciclo de vida (Blanco, Camarero, Fumero, Werterski, Rodriguez, 2009).

El ciclo de vida del proyecto se divide en cinco fases: exploración, inicialización, producción, estabilización y prueba del sistema como se observa en la figura 2.7. En general todas las fases contienen tres días de desarrollo destinados a: planificación, trabajo y liberación. Se añadirán días para acciones adicionales en casos particulares, estas fases son descritas para su mejor comprensión (Blanco et al., 2009).

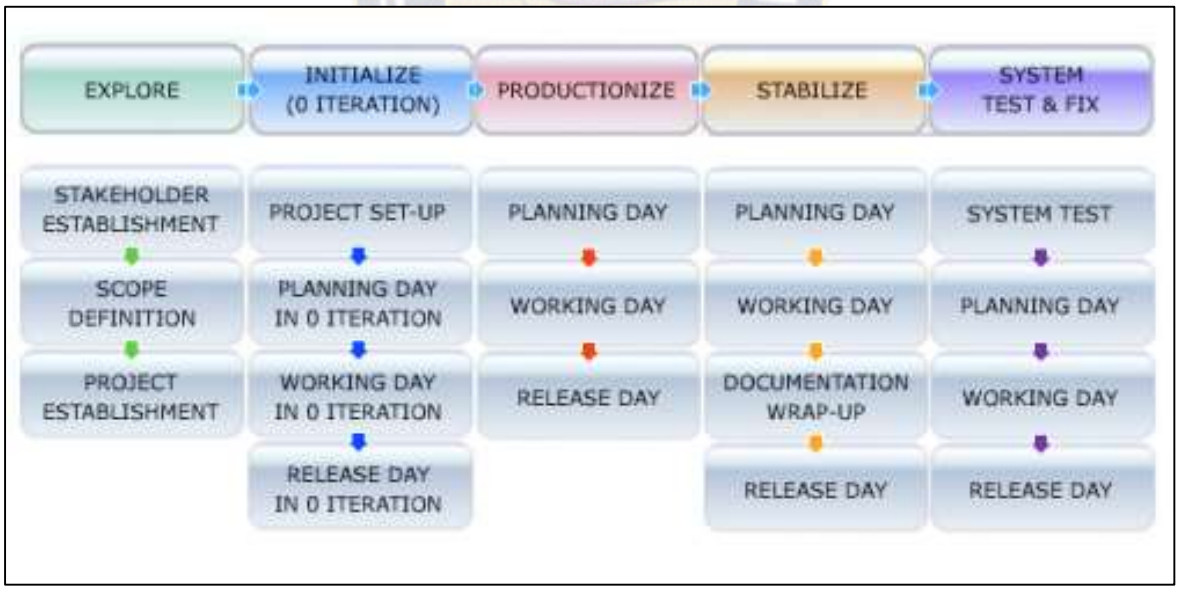


Figura 2.7: Ciclo de desarrollo Mobile-D
Fuente: (Mobile-D, 2016)

La fase de exploración, siendo ligeramente diferente del resto del proceso de producción, se dedica al establecimiento de un plan de proyecto y los conceptos básicos. Por lo tanto, se

puede separar del ciclo principal de desarrollo (aunque no debería obviarse). Los autores de la metodología ponen además especial atención a la participación de los clientes en esta fase.

Durante la fase de inicialización, los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico (incluyendo el entrenamiento del equipo de desarrollo). Los autores de Mobile-D afirman que su contribución al desarrollo ágil se centra fundamentalmente en esta fase, en la investigación de la línea arquitectónica. Esta acción se lleva a cabo durante el día de planificación. Los desarrolladores analizan el conocimiento y los patrones arquitectónicos utilizados en la empresa (extraídos de proyectos anteriores) y los relacionan con el proyecto actual. Se agregan las observaciones, se identifican similitudes y se extraen soluciones viables para su aplicación en el proyecto. Finalmente, la metodología también contempla algunas funcionalidades que se desarrollan en esta fase, durante el día de trabajo.

En la fase de producción se repite la programación de tres días (planificación trabajo-liberación) se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de ante mano (de ahí el nombre de esta técnica de TestDriven Development, TDD). Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.

En la fase de estabilización, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funcione correctamente. Esta será la fase más importante en los proyecto multi-equipo con diferentes subsistemas desarrollados por equipos distintos.

En esta fase, el equipo de desarrolladores realizará tareas similares a las que debían desarrollar en la fase de "producción", aunque en este caso todo el esfuerzo se dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación.

La última fase prueba y reparación del sistema, tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema.

El producto terminado e integrado se prueba con los requisitos del cliente y se eliminan todos los defectos encontrados.

2.3. Sistema Operativo Android

La tecnología se ha visto dominada por el uso de dispositivos móviles, los cuales no solo sirven para comunicarnos, como sucedía años atrás, estos dispositivos ahora nos permiten navegar por internet, comunicarnos, personalizarlos utilizando cientos de miles de aplicaciones disponibles que dan funcionalidades únicas, existen sistemas operativos para cada tipo de dispositivo.

Android es el sistema operativo personalizable y fácil de usar que incluye más de mil millones de dispositivos en todo el mundo, como teléfonos, tabletas, relojes, televisores, coches y otros en los que se utilizara en el futuro (ANDROID, 2016).

Android está basado en Linux, por lo que es de código abierto y cualquiera puede editarlo, es una plataforma creada por Google y por tanto sus servicios son la estrella de Android, aplicaciones como YouTube, Google Maps, Google Mail, y desde luego Google Search, son incluidas en el sistema operativo, lo que le permitió a Google ser omnipresente formando parte de la vida de 8 de cada 10 dueños de teléfonos inteligentes en el mundo (MOBILE, 2015).

2.3.1. Arquitectura de Android

La arquitectura de Android está formado por una pila de software donde se incluye un sistema operativo, middleware y aplicaciones básicas para los usuarios. A esta arquitectura la componen cuatro capas, las cuales están basadas en software libre con el núcleo de Linux, en la figura 2.8 se muestra esta arquitectura.

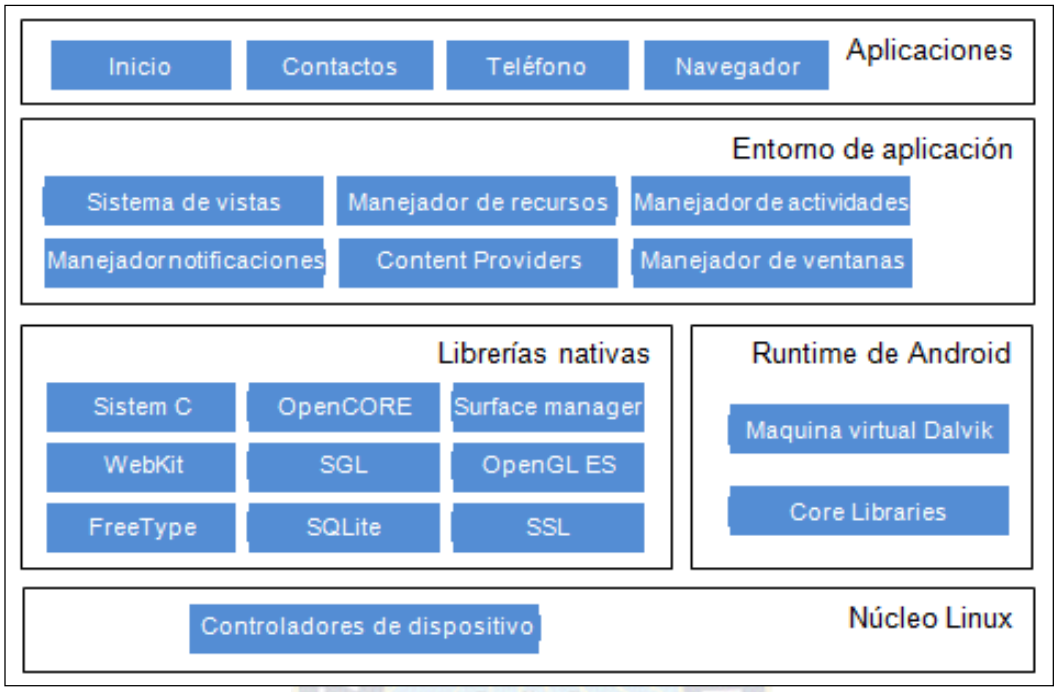


Figura 2.8: Arquitectura de Android
Fuente: (Alvarez, 2013)

2.3.1.1. Núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux en su versión 2.6. Esta capa proporciona servicios esenciales para su correcto funcionamiento, como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para los dispositivos.

Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila, por lo tanto, es la única dependiente del hardware (Alvarez, 2013).

"Una distribución de software basada en Linux que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores".

Android cumple con todas las condiciones de la definición anterior: usa el kernel de Linux, y tiene capas encima con distintos paquetes de software para cumplir con las necesidades de un grupo de usuarios (González, 2014).

2.3.1.2. Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android, no fue posible utilizar una máquina virtual Java estándar. Google tomo la decisión de crear una nueva, la máquina virtual Dalvik, que respondería mejor a estas limitaciones.

Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: que ejecuta ficheros Dalvink ejecutables (.dex) que es un formato optimizado para ahorrar memoria. Además está basado en registros. Cada aplicación corre en su propio proceso Linux con sus propias instancias de la máquina virtual Dalvik, delega al kernel de Linux algunas funciones como threading y el manejo de memoria a bajo nivel (Alvarez, 2013).

2.3.1.3. Librerías Nativas

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son (Alvarez, 2013):

- System C library: una derivación de la librería BSD[5] de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- Media Framework: librería basada en PacketVideo's OpenCORE; soporta codecs de reproducción y grabación de multitud de formatos de audio vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- Surface Manager: maneja el acceso al subsistema de representación gráfica en 2D y 3D.

[5] BSD: Berkeley Software Distribution

- WebKit: soporta un moderno navegador web utilizado en el navegador Android y en la vista webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- SGL: motor de gráficos 2D.
- Librería 3D: implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador harware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- FreeType: fuentes en bitmap y renderizado vectorial.
- SQLite: potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- SSL: proporciona servicios de encriptación Secure Socket Layer.

2.3.1.4. Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones).

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas. Este mismo mecanismo permite a los usuarios reemplazar componentes.

Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK[6] de Android no acaba de ofrecer todo lo disponible para su estándar del entorno de ejecución Java (JRE), pero es compatible con una fracción muy significativa de la misma.

Los servicios más importantes que incluye son (Alvarez, 2013):

- Views: extenso conjunto de vistas, que es la parte visual de los componentes.
- Resource Manager: proporciona acceso a recursos que no son en código.

[6] SDK: Software Development kit.

- Activity Manager: maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- Notification Manager: permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- Content Providers: mecanismo sencillo para acceder a datos de otras aplicaciones como los contactos.

2.3.1.5. Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones funcionaran en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción se puede utilizar el Android NDK[7](Alvarez, 2013).

2.4. Dispositivos de hardware

Implementar un sistema de Internet de las Cosas requiere un componente hardware fuertemente integrado en entorno físico, capaz de interactuar con éste, percibir su estado y transmitir su información; y un componente software adecuado para gestionar la información generada y actuar sobre el hardware anteriormente mencionado (González, 2016).

Por otro lado estos sistemas necesitan componentes que transformen los estímulos que reciben en información útil, estos periféricos son variados, mucho de ellos compatibles con diferentes placas.

[7] NDK: Native Development Kit.

2.4.1. Arduino

Arduino es una plataforma de creación de prototipos de código abierto basado en hardware y software libre, son capaces de leer las entradas como señales a través de sus puertos. Nació en el instituto de Diseño de Interacción de Ivrea como una herramienta fácil para el prototipado rápido, dirigido a estudiantes sin experiencia en electrónica y programación.

Arduino Nano que se muestra en la figura 2.9 es una placa pequeña y completa basada en ATmega328 (Arduino Nano 3.x) o ATmega168 (Arduino Nano 2.x) Tiene la misma funcionalidad del Arduino Duemilanove, pero en un paquete diferente, posee de una sola toma de corriente continua y funciona con un cable USB Mini-B en lugar de una normal. El Arduino nano fue diseñado y está siendo producido por Gravitech (ARDUINO, 2016).

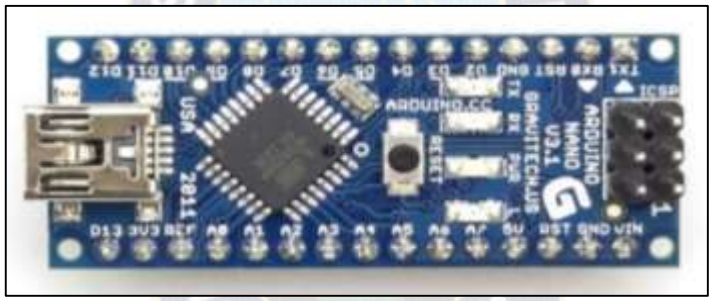


Figura 2.9: Arduino Nano
Fuente: (ARDUINO, 2016)

Arduino nano puede ser alimentado a través de la conexión USB Mini-B, fuente de alimentación no regulada de 6-20V (pin 30), fuente de alimentación regulada 5V (pin 27), el ATmega328 tiene 23 KB de memoria flash para el almacenamiento de código, de los cuales se utilizan 2KB para el gestor de arranque, cuenta con 2KB de SRAM y 1KB de EEPROM.

Cada uno de sus 14 pines pueden utilizarse como entrada o salida, utilizando las funciones pinMode(), digitalWrite() y DigitalRead() cada pin opera a 5V, y puede proporcionar o recibir un máximo de 40mA, tiene una resistencia de pull-up que vine desconectada por defecto, la cual es de 20-50 KOhms (ARDUINO, 2016).

En la tabla 2.1 se muestra las especificaciones de Arduino Nano ATmega328:

Tabla 2.1: Especificaciones Arduino Nano con ATmega328

Características	Especificación
Micro controlador	Atmel ATmega328
Tensión de funcionamiento (nivel lógico)	5V
Voltaje de entrada	7-15 V (recomendado) 6-20 V (limites)
E/S digitales	14 de los cuales 6 proporcionan salida PWM
Memoria flash	32 KB
SRAM	2KB
EEPROM	1KB
Velocidad de reloj	16MHz
Corriente continua para pin I/O	40 mA

Fuente: Fuente: (Arduino, 2016)

2.4.2. Raspberry Pi

La Raspberry Pi el cual se muestra en la figura 2.10 es un pequeño ordenador del tamaño de una tarjeta de crédito que se conecta con un monitor de computadora o también a un televisor, se puede interactuar con el mediante un mouse y teclado tradicional, o de manera remota.

Es un pequeño ordenador que se puede utilizar en proyectos de electrónica a través de sus puertos GPIO (General Purpose Input/Output o Entrada y Salida de uso General), estos puertos son programables por lo que facilita el control de diferentes dispositivos que se le pueden conectar.

Su uso también es similar al de una PC de escritorio tradicional, el sistema operativo raspbian cuenta diversas tareas como hojas de cálculo, procesamiento de texto, navegación por internet, juegos, reproducir videos de alta definición, lenguajes de programación, este ordenador fue creado con el propósito de enseñar a programar en inicios el lenguaje Python (RASPBERRYPI, 2012).

Figura 2.10: Raspberry Pi 2
Fuente: (Upton, 2016)

La tabla 2.2 describe las características más sobresalientes y la comparación entre placas Raspberry Pi, se aprecia en la misma el notable crecimiento que presenta en sus diferentes versiones.

Tabla 2.2: Comparación Raspberry Pi 1, Raspberry Pi 2, Raspberry Pi 3

Características	**Raspberry Pi 1**	**Raspberry Pi 2**	**Raspberry Pi 3**
Procesador	ARM 1176JZF-S a 700MHz	ARM Cortex A7 quad-core a 900MHz.	ARM Cortex-A43 quad-core a 1.2GHz
Procesador grafico	VideoCore IV 520MHz OPENGLES 2.0	VideoCore IV 250MHz OPENGLES 2.0	VideoCore IV 400MHz OPENGLES 2.0
RAM	256MB LPDDR SDRAM 400MHz	1GB LPDDR2 SDRAM 450MHz	¡GB LPDDR2 SDRAM 450MHz
Video	HDMI 1.4 1920x1200	HDMI 1.4 1920x1200	HDMI 1.4 1920x1200
Puertos USB	Uno en el modelo B, dos en el modelo B+	cuatro	Cuatro
Almacenamiento integrado	SD, microSD en el modelo A+	MicroSD	MicroSD
Conexión Red	ninguna	10/100 Ethernal vía hub USB	WiFi 802.11n
Bluetooth	No	No	Bluetooth 4.1
Dimensiones	8,5 x 3,5 cm	8,5 x 3,5 cm	8,5 x 3,5 cm
Precio	29,99 $	34,95$	35$

Fuente: (Baena, 2016)

Raspberry Pi al ser un ordenador con características marcadas por la capacidad de interacción a través de sus puertos GPIO, necesita contar con un Sistema operativo, que estén adaptados para poder controlar los puertos GPIO, muchos de los sistemas operativos que se lanzaron son basados en Linux, también existe aquel basado en Windows 10.

La raspberry pi no solo se utiliza para proyectos de Internet de las Cosas, lo hace especial precisamente la capacidad de personalizarse para el proyecto que se dese, se tiene tantas opciones de proyectos, como sistemas operativos completos, servidores o gestores de contenido, para su entendimiento citamos a algunos que están disponibles:

- Raspbian, es una versión de Linux basado en debían y especialmente desarrollada para Raspberry, es también una de las más populares, siendo el sistema operativo oficial para ese ordenador, pudiendo en el programar dispositivos, levantar un servidor completo, o usarlo como un ordenador convencional.
- Windows IoT Core, Es una versión especial de Windows, que es una plataforma de desarrollo para que los programadores familiarizados más con Windows, experimenten con dispositivos conectados a internet.
- OSMC, es un gestor de contenidos multimedia para usar la Raspberry Pi, es un media center, basado en Kodi.

2.4.3. Módulo detector de presencia PIR

El módulo PIR[8] o pasivo infrarrojo que se muestra en la figura 2.11 es un sensor que reacciona ante determinadas fuentes de energía, tales como el calor del cuerpo humano o animales, es llamado pasivo debido a que no emiten radiaciones sino que las recibe, todo cuerpo que este a una temperatura superior a los 0º Kelvin, emiten radiación infrarroja negativa, esta radiación aumenta si hay un incremento en la temperatura provocada por algún cuerpo cercano, el sensor la percibirá y mandara la señal de detección (Villegas, 2012).

[8] PIR: Passive Infrared

Figura 2.11: Sensor Piroeléctrico PIR
Fuente: (NOVA, 2016)

2.4.4. Módulo RFID

El módulo de Identificación por Radio Frecuencia se usa para referirse a un sistema inalámbrico que usa radiofrecuencia para transmitir información que se encuentra adherida a un objeto, ya sean estas etiquetas, tarjetas u otros medios (Castro, 2015).

Este módulo cuenta con un lector, para el intercambio de información entre este y las etiquetas o tarjeras RFID. A diferencias de las etiquetas con código QR[9], las etiquetas o tarjetas RFID no requieren estar a la vista para que un lector pueda leer la información que contiene, el módulo RFID puede ser conectado tanto a Arduino como a Raspberry Pi, entre otras alternativas, este módulo se muestra en la figura 2.12.

Figura 2.12: Modulo RFID
Fuente: (NOVA, 2016)

2.4.5. Módulos de Conexión a Internet

El módulo inalámbrico ESP8266 ha sido diseñado para que se pueda controlar el microcontrolador conectándolo a una conexión Wi-Fi de manera fácil, eficiente y de bajo

[9] QR: Quick Response Code

costo, soporta la norma 802.11 b/g/n, que es muy utilizado actualmente, y puede funcionar como un punto de acceso o como estación, para él envió o recepción de datos, trabaja a una frecuencia de 2.4GHz (NOVA, 2016), este módulo se muestra en la figura 2.13.

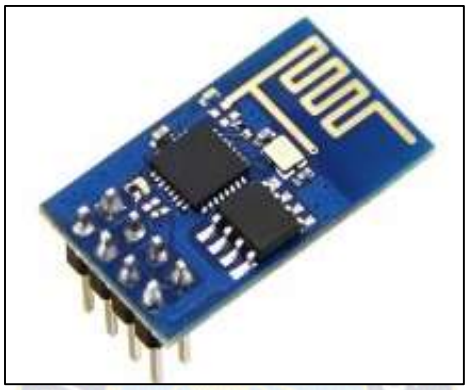


Figura 2.13: Modulo Wi-Fi Esp8266
Fuente: (NOVA, 2016)

2.5. Arquitectura de servicios

2.5.1. WebServices

Las WebServices son basadas XML[10], y en la mayoría de los casos, HTTP[11], los "servicios Web" todavía significan muchas cosas para muchas personas, pero son el intercambio entre sistemas de mensajes basados en SOAP[12]. Estos mensajes se componen de XML, que es un estándar abierto basado en texto accesible por cualquier persona desde cualquier aplicación (IBM, 2011).

Mediante el uso de los servicios web, la aplicación puede publicar su función a mensaje al resto del mundo. Los servicios Web utilizan XML para codificar y decodificar los datos, y SOAP para transportarlos.

Los servicios Web tienen dos tipos de usos (W3SCHOOLS, 2016):

[10] XML: eXtensible Markup Language
[11] HTML: Hipertext Transfer Protocol
[12] SOAP: Simple Object Acces Protocol

- Aplicación con componentes neutralizables; Los WebServices pueden ofrecer aplicación-componentes como: la conversión de monedas, informes del tiempo, o incluso la traducción de idiomas como servicio.
- Conectar el software existente; Los WebServices pueden ayudar a resolver el problema de la interoperabilidad entre diferentes aplicaciones con una forma de vincular los datos, se puede intercambiar datos entre diferentes aplicaciones y diferentes plataformas, cualquier aplicación puede tener un componente de WebServices, se puede crear independientemente del lenguaje de programación.

2.5.2. Servicios Restful

REST define un conjunto de principios de la arquitectura por el cual se puede diseñar servicios web que se centren en los recursos del sistema, incluyendo los estados de estos recursos que se trasfieren a través de HTTP, una amplia gama de clientes escrito en diferentes idiomas si se mide por el número de servicios web que lo utilizan, REST ha surgido en los últimos años solamente como un modelo de servicios Web predominante.

Una de las características clave de un servicio web RESTful es el uso explícito de los métodos HTTP, definido en forma de protocolos en RFC2616[13]. HTTP GET, por ejemplo se define como un método que está destinado a ser utilizado por una aplicación cliente con el fin de recuperar un recurso de producción de datos, para obtener los datos desde un servidor Web, o para ejecutar una consulta con la expectativa de que el servidor Web busca y responda con un conjunto de recursos (Rodriguez, 2008).

Los servicios RESTful es una web API implementado en HTTP bajo principios de REST, es una colección de recursos con cuatro aspectos definidos (Sánchez, 2013):

- El URI (Identificador uniforme de recursos) es base para la web API es como http://example.com/resource.

[13] RFC: Request For Comments: 2616

- El internet media type es soportado por la web API, por lo general para ser consumido con JSON, pero puede ser cualquier otro tipo valido de internet media type, siempre y cuando se trate de un hipertext standard valido.
- El conjunto de las operaciones que soportan la web API utilizan métodos HTTP como por ejemplo: GET, PUT, POST, DELETE.
- El API debe ser iniciado por hipertexto.

A diferencia de las Web Services basados en SOAP no existe un estándar oficial para RESTful web AP. Esto se debe a que REST tiene una arquitectura a diferencia de SOAP que es un protocolo. A pesar de que el resto no es un estándar, una aplicación RESTful puede utilizar estándares como HTTP, URI, XML, entre otros (Sánchez, 2013).

2.5.3. Message Queue Telemetry Transport (MQTT)

Para que la comunicación entre objetos se necesita de protocolos que brinden el servicio, para esto se cuentan con diversas alternativas, el que se va a estudiar es el protocolo MQTT[14] es usado para la comunicación M2M[15]capaces de enviar datos a través de telemetría[16], está orientado a la comunicación de sensores o dispositivos limitados, debido a que puede funcionar en redes que tengan muy poco ancho de banda, latencia alta o redes poco confiables.

La arquitectura MQTT sigue una topología de estrella ilustrada en la figura 2.14, con un nodo central que hace de servidor o "Broker" con una capacidad de hasta 10000 clientes, el bróker es el encargado de gestionar la red y de transmitir los mensajes, los clientes mandan periódicamente un paquete (PINGREQ) y esperan la respuesta de un paquete (PINGRESP) la comunicación puede ser cifrada entre muchas otras opciones (Yébenes, 2015).

[14] MQTT: Message Queue Telemetry Transport
[15] M2M: Machine to machine
[16] Telemetría: Sistema de medición de magnitudes físicas que permite transmitir los datos a un observador lejano.

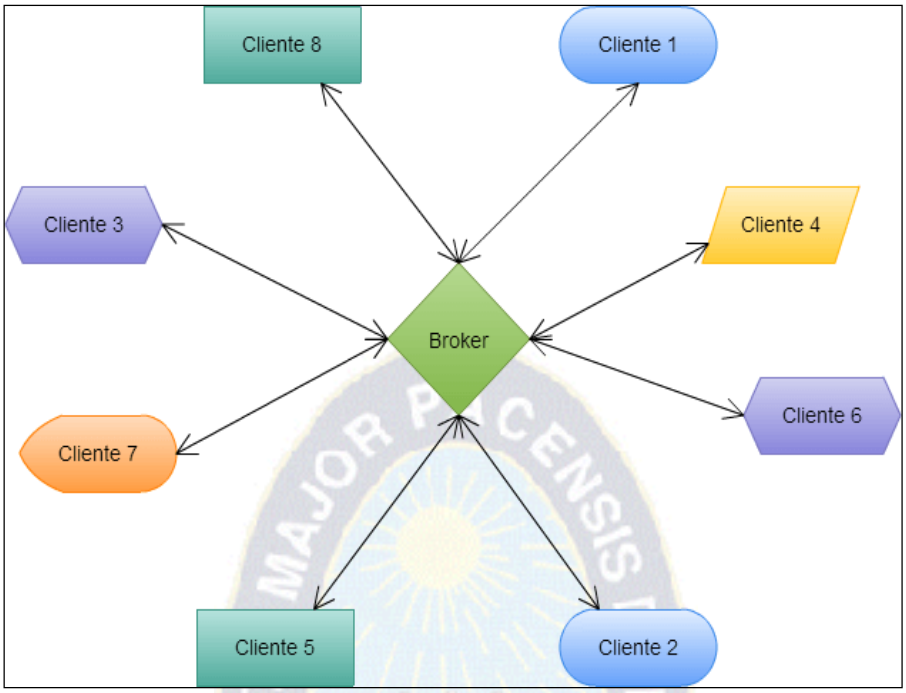


Figura 2.14: Topología MQTT
Fuente: (Yébenes, 2015)

En noviembre de 2011 IBM[17] y Eurotech anuncian su participación conjunta en el grupo de trabajo de la industria M2M de Eclipse, y se dona el código MQTT al proyecto Eclipse Paho, el protocolo también se ha conocido como "WebSphere MQTT", es un producto de IBM que consiste en la implementación de MQTT en manera escalable, actúa directamente con la familia de protocolos WebSphere MQ (MQTT, 2016).

Las especificaciones de este protocolo consta principalmente de tres puntos: Calidad de servicio (QoS), formato de control de paquetes y Control de Paquetes (MQTT, 2016).

2.5.3.1. Calidad de servicio.

Los paquetes son enviados de acuerdo a la calidad QoS definida, MQTT proporciona tres niveles de QoS.

[17] IBM: International Business Machines

- QoS 0: el mensaje es enviado de acuerdo a la disponibilidad de la red. No existe respuesta por el suscriptor y no se realiza un reenvió por el remitente, el mensaje llega al suscriptor ya sea una o varias veces.
- QoS 1: se asegura que el mensaje llegue al suscriptor al menos una vez, el paquete envía un acuse de recibo para corroborar que el mensaje ha sido recibido.
- QoS 2: Se utiliza para prevenir la perdida y la duplicidad de mensajes, en este nivel el consumo general de ancho de banda aumenta.

2.5.3.2. Formato de control de paquetes

El protocolo funciona con intercambio de paquetes de control los cuales se definen de la siguiente forma:

- Cabecera obligatoria: Esta en todos los paquetes MQTT y consta de dos bytes, para definir un paquete se usan los bits del primer byte desde el primer bit hasta el tercer bit en los que se definen las banderas del tipo de paquete que se use, desde el cuarto bit al séptimo bit se indica el tipo de paquete: 0x0: reservado, 0x1: CONNECT, 0x2: CONNACK, 0x3: PUBLISH (QoS), 0x4: PUBACK, 0x5: PUBREC, 0x6: PUBREL, 0x7: PUBCOMP, 0x8: SUBSCRIBE, 0x9: SUBACK, 0xA: UNSUBSCRIBE, 0xB: UNSUBACK, 0xC: PINGREQ, 0xD: PINGRES, 0xE: DISCONECT, 0xF: reservado, el segundo byte se utiliza para la codificación del paquete.
- Cabecera variable: algunos tipos de paquetes contienen una cabecera variable que esta entre la cabecera obligatoria y el cuerpo del paquete, esta consta de dos bytes, su contenido varía dependiendo del tipo de paquete que se encuentra en la cabecera obligatoria.
- Cuerpo del paquete: después de los dos bytes de la cabecera obligatoria y otros dos bytes de la cabecera variable, el resto del paquete consta del cuerpo del paquete donde se encuentran los datos que se transmiten a través del protocolo, los paquetes que requieren de un cuerpo son: CONNECT, SUBSCRIBE, SUBACK,

UNSUBSCRIBE, en el paquete PUBLISH el cuerpo es obligatorio, en los tipos de paquetes restantes el cuerpo es opcional.

- Control de paquete: después de que se efectué la conexión TCP[18] por un cliente hacia el servidor, le primer paquete que se envía al servidor debe ser un paquete CONNECT con la información de la conexión (id cliente, usuario, contraseña y otros), una vez enviado el paquete CONNECT, el servidor responde con un paquete CONNACK que verifica que la conexión se realizó con éxito. Una vez que se realizó la conexión existen once alternativas que el protocolo prevé: PUBLISH; publicar el mensaje, PUBACK; publicación realizada, PUBREC; publicación recibida para QoS 2, PUBREL; publicación liberada para QoS 2, PUBCOMP; publicación completa para QoS 2, SUBSCRIBE; suscripción a topicos, SUBACK verificación de suscripción, UNSUBSCRIBE; finalizar la suscripción, UNSUBACK; confirmación de finalización de suscripción, PINGREQ; solicitar ping, PINGRES; respuesta ping, DISCONECT; desconectar.

2.5.4. Notificaciones push

La tecnología push es una forma de comunicación en la que una aplicación servidora envía un mensaje a un cliente consumidor. Es decir, es un mensaje que un servidor envía a una persona alertándolo de que tiene una información nueva. Lo que caracteriza esta tecnología es que es siempre el servidor el que inicia esta comunicación, aunque el cliente no tenga interés de saber si hay algo nuevo (QODE, 2015)

Las notificaciones push que se muestra en la figura 2.15 permiten él envió de mensajes desde cualquier parte de un sistema a una aplicación móvil tanto si la aplicación está siendo utilizada por el usuario, si está corriendo en segundo plano si todavía no ha sido arrancada o, incluso, si el dispositivo está en reposo (Rodríguez, 2014).

[18] TCP: Transmission Control Protocol

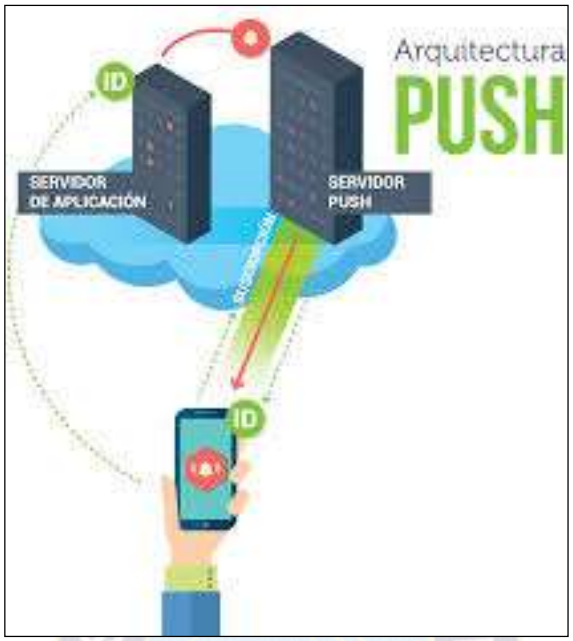


Figura 2.15: Arquitectura Push
Fuente: (Martinez, 2014)

2.5.5. Notificación pull

A diferencia de las notificaciones push, las notificaciones pull se producen cuando el usuario es el que inicia la comunicación. Es el usuario quien elige la información y quien la busca, además de establecer manualmente la frecuencia con la que el servidor envié la notificación.

A través de los sistemas pull el usuario pregunta al servidor por una nueva información, es decir la búsqueda en la red, que generalmente, solemos hacer a través de buscadores tales como Google, Ask, Yahoo, por lo cual, podemos decir que somos nosotros los que elegimos la información que queremos obtener (Hípola, 2012).

Los clientes que usan arquitectura pull, preguntan periódicamente si existen actualizaciones de información, pull significa “jalar” y es eso justamente lo que se está haciendo jalar, información desde el cliente móvil. Este tipo de sistemas es sencillo pero tiene un consumo de recursos elevado del lado del cliente móvil, ya que involucra tener un proceso persistente que continuamente está preguntando al servidor si hay cambios, esta notificación se representa en la figura 2.16 (Martinez, 2016).

Figura 2.16: Arquitectura Pull
Fuente: (Martinez, 2016)

2.6. NoSQL

El termino NoSQL[19] cobija varios conceptos relacionados sobre almacenamiento, gestión de datos y datos voluminosos, es lo que denominan un término "sombrilla" por que abarca varios elementos.

El termino fue acuñado por Calor Strozzi en 1998 y resucitado por Eric Evans en 2009 y el mismo sugirió se llamasen estas bases de datos como Big Data.

Se remontan a la época de las bases de datos de red y jerárquicas y una serie de productos que no eran relacionales los cuales resuelven problemas que no tienen las características similares a los de: amazon.com, Facebook, Youtube, twiter, Netflix, Yahoo, EBay, Hulu, IBM y que en la época en que surgieron no se tenía internet (Díaz, 2013).

La base de datos de la próxima generación, se caracterizan por ser: no relacional, distribuido, de código abierto, y escalable horizontalmente. La intención original ha sido crear bases de

[19] Not Only SQL

datos modernas a escala Web, el movimiento comenzó a principios de 2009 y está creciendo rápidamente. A menudo se aplican características tales como: esquema libre, el apoyo de replicación facial, simple API, eventualmente consistente, enormes cantidades de datos y muchos más (NoSQL, 2016).

Los sistemas NoSQL intentan atacar problemas de alta escalabilidad, tolerancia a grandes volumen de datos, eficiencia y respuesta rápida, proponiendo una estructura de almacenamiento más versátil, aunque sea acostó de perder ciertas funcionalidades como las transacciones que engloban operaciones en más de una colección de datos, o la incapacidad de ejecutar el producto cartesiano de dos tablas, teniendo que recurrir a la desnormalización de datos (Paramio, 2011).

2.7. Base de datos Cassandra.

Apache Cassandra es una base de datos NoSQL de código abierto masivamente escalable, es perfecta para gestionar grandes cantidades de datos, estructurados, semi-estructurados, y no estructurados, a través de múltiples centros de datos y la nube.

Cassandra proporciona una disponibilidad continua, escalabilidad lineal, y la simplicidad operativa a través de muchos servidores, de conveniencia sin ningún punto único de fallo, junto con un potente modelo de datos dinámico diseñado para una máxima flexibilidad y tiempo de respuesta rápido.

Cassandra proporciona una escalabilidad lineal, lo que significa que la capacidad se puede añadir fácilmente, simplemente mediante la adición de nuevos nodos en línea. Por ejemplo, si 2 nodos pueden manejar 100.000 transacciones por segundo, 4 nodos apoyaran 200.000 trans/seg y 8 nodos abordaran 400.000 trans/seg (DATASTAX, 2016) la distribución de nodos se muestra en la figura 2.17.

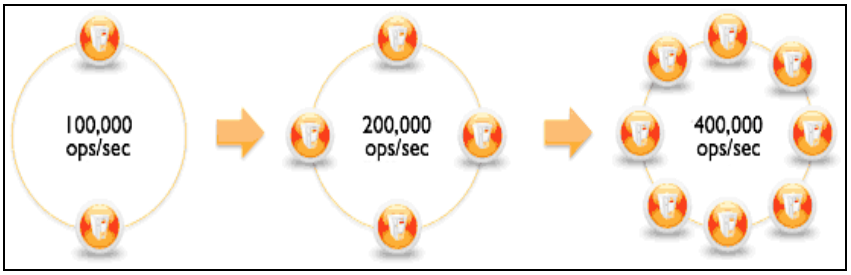


Figura 2.17: Transacciones según nodos
Fuente: (DATASTAX, 2016)

Cassandra utiliza un lenguaje que es CQL[20] es la interfaz principal que es similar a SQL[21], estas dos comparten la misma idea abstracta de una estructura construida en base de columnas y filas. La principal diferencia es que a diferencia de SQL, Cassandra no soporta la unión o las sub consultas. En su lugar Cassandra hace hincapié en la desnormalizacion a través de CQL que tiene la característica de colecciones y agrupaciones especificadas en niveles de esquemas.

El modelo de datos de Cassandra está compuesto por:

- Clúster: que permite la distribución en múltiples maquinas operando juntas y presentándose como una única entidad al usuario final, así la estructura más externa es el clúster.
- Espacio clave: un clúster es un contenedor para un espacio clave, que es el más extenso, similar a un contenedor de tablas en el modelo relacional.
- Familia de columnas: es un contenedor para una ordenada colección de registros, los cuales forman una ordenada colección de columnas. Otra diferencia es que en el modelo relacional, las tablas contienen únicamente columnas y registros, mientras que en una familia de columnas puede haber columnas relacionadas, como super-columnas comunes.

La representación de espacio clave, súper columnas y columnas se muestran en la figura 2.18

[20] CQL: Cassandra Query Language
[21] SQL: Structured Query Language

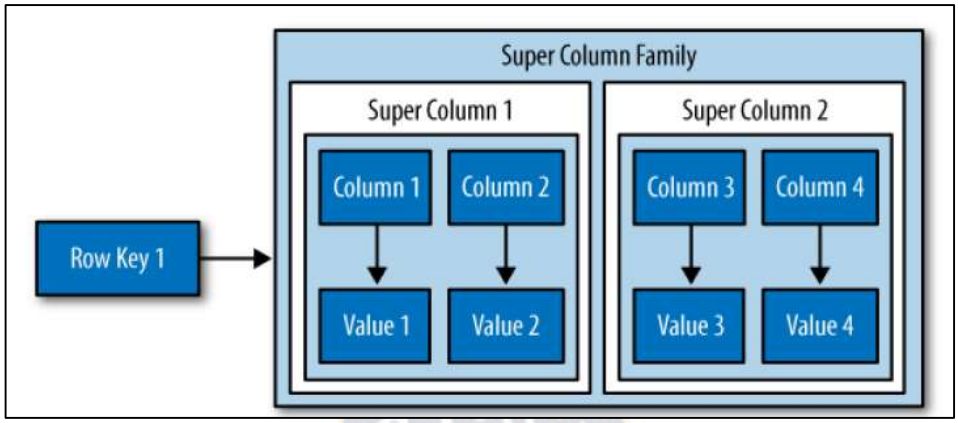


Figura 2.18: Modelo de datos de cassandra
Fuente: (EUGOMAN, 2016)

2.8. Internet de las Cosas

En 1999 Kevin Ashton inventa el término "Internet of Things". "El IoT es el mundo en el que cada objeto tiene una identidad virtual propia y capacidad potencial para integrarse e interactuar de manera independiente en la red con cualquier otro individuo ya sea una maquina (M2M) o un humano". (Romorico, 2014)

El acceso a internet se hace cada vez más notable, un reporte de Ericsson, predice que el llamado Internet de las Cosas será una completa realidad en dos años, puesto que los objetos inteligentes con conexión a internet serán muchos más que los celulares, y que para 2021, 28 mil millones de dispositivos estarán conectados a la red, de los que 16 mil millones serán parte de la IoT (Nuñez, 2016).

2.8.1. Impacto del Internet de las Cosas

Internet de las cosas abarca un concepto muy grande por lo que se estima que todos los ingresos que genere no serán de los productos vendidos como ocurre en un mercado tradicional, sino más bien, será de los servicios brindados, ya que se genera mucha información a partir de los sensores y dispositivos (Asay, 2014).

En el estudio realizado por Vision Mobile se muestra el crecimiento de desarrolladores para IoT desde el 2014, con una estimación hasta el 2020 ilustrada en la figura 2.19.

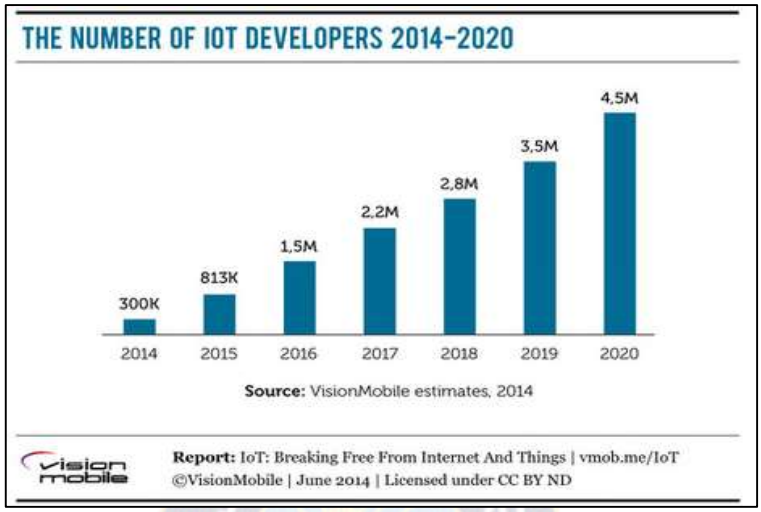


Figura 2.19: Número de desarrolladores para internet de las cosas
Fuente: (Asay, 2014)

Estos datos, con su estimación hasta el 2020, muestran el crecimiento considerable en la cantidad de desarrolladores, para Internet de las Cosas, lo que supone la creación de nuevas y mejores soluciones que contribuyan a su desarrollo, en diferentes lugares y para distintas situaciones.

2.8.2. Aplicaciones del Internet de las Cosas

El gran alcance de IoT hace que elaborar un análisis profundo de los ámbitos de aplicación terminaría siendo demasiado extenso, por lo que se presenta una descripción comprensible que de una idea general sobre las capacidades de esta tecnología, los ámbitos que se consideran son:

- Medio ambiente.

  Según Bliznakoff (2014) dentro de este ámbito se pretende incluir todas aquellas aplicaciones centradas principalmente en redes de sensores destinadas a la protección y la salud, no solo del ser humano sino también de nuestro planeta como ejemplos destacados tenemos:

  - Control de polución de ríos/mares, comprobar el estado del agua y medir el pH.

- Protección de Fauna Salvaje, mediante collares de localización es posible conocer la ubicación de animales.
- Prevención de catástrofes, una de las aplicaciones con mayor impacto para el ser humano en cuanto a evitar muertes seria la alerta temprana de catástrofes naturales.

- Ciudades inteligentes.

  Las redes de banda ancha, las cosas conectadas y los datos abiertos ayudaran a impulsar la competitividad, la sostenibilidad y la habitabilidad. Transformarán drásticamente la experiencia urbana de los viajeros y de los habitantes de las ciudades, según Business Intelligence (BI), las áreas urbanizadas generan 4.1 TB de datos al día por kilómetro cuadrado en 2016 (Jadoul, 2015).

- Salud.

  Existen en el mercado muchos wearables[22] en el ámbito de la salud, este sector se considera pionero en este tipo de dispositivos, con desarrollos para medir la presión arterial o las actividades físicas, ahora se pueden ir más allá, como medir en tiempo real el nivel de glucosa y lanzar una alarma cuando los niveles estén fuera de los límites establecidos. Los sensores que se utiliza en el ámbito de salud incluyen: electrocardiogramas, presión sanguínea, oximetros, entre otros sensores especializados (Teruel, 2015).

2.8.3. Internet de las Cosas en la seguridad de ingresos

Las aplicaciones de IoT son extremadamente variadas, la seguridad está contemplada entre ellas, las soluciones que se puede dar al servicio de seguridad, en particular al control de ingresos, cuenta con diferentes herramientas, sensores y dispositivos que trabajan en conjunto para brindar soluciones mucho más eficientes.

[22] Wareable, hace referencia al conjunto de aparatos y dispositivo electrónicos que se incorporan en alguna parte de nuestro cuerpo.

Muchos de estos dispositivos están disponibles para trabajar con ellos y brindar soluciones a problemas de seguridad en ingresos, entre ellas tenemos la identificación dactilar, facial, ocular, cada una de estas utiliza una técnica diferente de identificación, la que se estudiara es la RFID, este medio permite identificar personas portadoras de una tarjeta o etiqueta de radio frecuencia.

Todos estos dispositivos brindan identificaciones seguras unas más que otras, por lo que obtener el mayor beneficio de ellas, depende mucho del modo en que se vaya a trabajar con estos, la manera de aprovechar el potencial de la identificación de personas en los ingresos, es poder interpretar los datos de identificación que estos dispositivos generan, con el fin de interpretar los mismos. El verdadero valor de IoT no está netamente en la conexión entre dispositivos, sino en el valor de los datos que serán capaces de recoger, en la capacidad de analizarlos, rentabilizarlos y extraer información valiosa que dé lugar a nuevas fuentes de ingresos, reducción de costos o a dar mejor experiencia a los destinatarios de esta tecnología. (Carrasco, 2016)

CAPÍTULO 3

MARCO APLICATIVO

3.1. Introducción

En este capítulo se desarrollara el prototipo del Sistema de Control de Ingresos no Autorizados a un Ambiente, haciendo el buen uso de las metodologías descritas (ver sección 2.2). El capítulo muestra la aplicación de las fases de cada una de las metodologías, orientado a utilizar aquellas que puedan mejorar y dar mayor eficiencia del modelo en V aplicada a sistemas que funcionen de manera conjunta con software y hardware, la composición de estas se usara para orientar el objetivo de la investigación y así lograr el mejor funcionamiento del prototipo.

3.1.1. Refactorización del Modelo en V

La refactorización del Modelo en V, tomando pasos y fases del Método para el Diseño de Sistemas de Control Automático y la Metodología Mobile–D, permitirá que esta se adecue mejor al tipo de investigación propuesto. Como parte de la refactorización se reformula ciertos acápites descritos en las Fases del Modelo en V.

Fase 1. Especificaciones: Como indica la primera fase del Modelo en V, se debe definir los diferentes requisitos a los que el sistema debe responder. Para reforzar estos requisitos se aplica los pasos primero y segundo del Método para el Diseño de Sistemas de Control Automático los cuales indican que los mismos se deben definir contemplando los objetivos del sistema, utilizando diferentes técnicas de diagramas, también se debe definir en esta fase la variable que se desea controlar con el sistema propuesto. Por último se debe contemplar el uso que el cliente dará al sistema, por lo que se da especial atención a su participación, siendo esta última parte de la fase de exploración de la Metodología Mobile-D

Fase 2. Funcional: En esta Fase se debe obtener un diseño y visión general del sistema. Establecer un conjunto de módulos, desplegando las necesidades obtenidas en la fase de

especificaciones, facilitando a la tarea de codificación y los modelos lógicos de los datos físicos.

Fase 3. Diseño: Manteniendo lo descrito en el esta fase del Modelo en V se debe definir los componentes de hardware y software para desarrollar el sistema final a cuyo conjunto se denomina arquitectura el sistema. Ya que en esta fase se deben describir los componentes especialmente de hardware, es apropiado hacer uso del tercer paso del Método para el Diseño de Sistemas de Control Automático el cual se refiere a la identificación de un sensor que deberá medir la variable controlada y así alcanzar una precisión aceptable.

Fase 4. Codificación: El Modelo en V indica que en esta fase se materializa el diseño de la fase anterior desarrollando los elementos que conforman al sistema, denominando así a las clases o módulos. La fase de producción de la metodología Mobile-D sugiere planificar la iteración de trabajo en términos de requisitos y tareas a realizar, con el fin de lograr eficiencia al desarrollar estos elementos. Esta fase debe adecuarse al cuarto paso mencionado en el Método para el Diseño de Sistemas de Control Automático, el mismo indica que cada uno de estos elementos deberá someterse al comportamiento del proceso, integrando el funcionamiento del software con los distintos dispositivos de hardware para así logar el comportamiento deseado.

Fase 5. Test de Diseño: Se verifica cada clase o modulo desarrollado de forma unitaria comprobando su funcionamiento adecuado, generando documentos de pruebas. Se realiza la verificación del correcto funcionamiento del hardware y software, con las clases o módulos producidos por las herramientas o lenguajes escogidos.

Fase 6. Test Funcional: En esta fase del modelo en V se debe comprobar que todo el sistema tenga el funcionamiento adecuado. Ante la presencia de errores que se puedan detectar al momento de comprobar el sistema, la metodología Mobile-D da flexibilidad en la corrección de estos errores, llevando acabo las últimas acciones de integración para asegurar que el sistema completo funcione correctamente.

Se comprueba el cumplimiento de los requisitos establecidos en la primera fase.

Fase 7. Test de Especificaciones: Se realizan las últimas pruebas, sobre un escenario real en su ubicación final, anotando una vez más las pruebas realizadas y los resultados obtenidos. Se debe garantizar el mantenimiento del sistema, con la finalidad de probarlo en nuevos entornos.

3.2. Fase 1. Especificaciones

El sistema, está conformado por tres elementos los cuales son vitales para que el control se lleve a cabo, con estos se asegura que se vaya a cumplir el objetivo propuesto. Estos elementos son descritos para su mejor comprensión en la tabla 3.1:

Tabla 3.1: Partes del Sistema de Control de Ingresos no Autorizados a un Ambiente

N.-	ELEMENTO	FUNCIÓN
1	Sistema identificación	Se encarga de recibir señales externas de identificación, para que se procesen los mismos y así mandar órdenes al actuador. Este está constantemente conectado a internet y es el encargado de autorizar ingresos.
2	Sistema del actuador	Su función es ejecutar las órdenes en forma de acciones físicas, resultantes de los datos que procesa el sistema receptor.
3	Aplicación móvil	Este está conectado a internet su función es recibir las notificaciones que el sistema genera cuando se detecte un acceso no autorizado, entre otras.

Fuente: (Elaboración Propia).

Para el funcionamiento conjunto de estas y haciendo uso del Modelo en V, se inicia describiendo las especificaciones del sistema, para esto se debe distinguir entre dos posibles entidades que intervienen con el sistema en primer lugar está la persona, a quien se le debe identificar mediante RFID, a esta también se le debe notificar los intentos de acceso no autorizados a algún ambiente, en segundo lugar el usuario, quien es encargado de registrar personas, otorgar permisos entre otras funciones administrativas.

Teniendo en claro lo mencionado las especificaciones del funcionamiento del sistema son:

- Diseñar un sistema capaz de identificar, mediante RFID aquellos accesos que cuenten con las autorizaciones correspondientes.
- Definir el modelo de datos que permita almacenar aquellos generados en el concepto de IoT (ver sección 2.8.3).
- Definir los roles administrativos que tendrán los usuarios sobre el sistema.
- Diseñar la configuración adecuada entre la Raspberry Pi y el actuador.
- Diseñar la aplicación móvil, que deberá recibir notificaciones emitidas por la Raspberry Pi al identificar un intento de ingreso no autorizado.

Como parte de la metodología para esta fase, se identifica la variable que deseamos controlar. Para objetos de la investigación la variable que se desea controlar es el ingreso a un ambiente, controlando con esto si existe o no la autorización correspondiente.

3.2.1. Historias de Usuario

Con el fin de cumplir con efectividad los objetivos planteados es necesario contar con un conjunto de historias de usuario para la especificación de requisitos, estos mantendrán en claro lo que se desea realizar durante todo el desarrollo de la investigación, siguiendo del Modelo en V.

El modelo de datos es importante para el respaldo de la información la historia de usuario de esta se muestra en la tabla 3.2.

Tabla 3.2: Diseño inicial del modelo de datos

Historia de Usuario	
ID: # 1	**Usuario:** Desarrollador
Nombre: Diseño inicial del modelo de datos	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Baja
Programador Responsable: Gabriel Casas Mamani	
Descripción Diseño inicial del modelo de datos, el cual contendrá el registro de los usuarios, personas, permisos, con los que trabajara el sistema.	
Observación El modelo de datos se lo diseña en términos de NoSQL	

Fuente: (Elaboración Propia)

Para viabilizar la funcionalidad del prototipo se debe crear una historia de usuario, mostrada en la tabla 3.3, esta debe contemplar la selección de los componentes finales del mismo, tanto de software como de hardware.

Tabla 3.3: Identificación de software y hardware

Historia de Usuario	
ID: # 2	**Usuario:** Desarrollador
Nombre: Identificación de software y hardware	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Baja
Programador Responsable: Gabriel Casas Mamani	
Descripción Identificar los dispositivos de hardware que conformara el producto final para cada uno de ellos se escoge el software adecuado que le brindara funcionalidad.	
Observación El software debe ser compatible con el lenguaje python.	

Fuente: (Elaboración Propia)

El sistema alojado en la Raspberry Pi, que hará de servidor debe ser capaz de identificar las tarjetas RFID, para obtener el código de identificación, esta historia de usuario se observa en la tabla 3.4.

Tabla 3.4: Diseño del sistema servidor

Historia de Usuario	
ID: # 3	**Usuario:** Desarrollador
Nombre: Descripción del sistema servidor	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Baja
Programador Responsable: Gabriel Casas Mamani	
Descripción El sistema servidor alojado en la Raspberry Pi, contara con clases, drivers, que permitan la identificación de tarjetas RFID a través del módulo RC522	
Observación Escoger los driver, librerías adecuadas para efectivizar la conexión entre el módulo RC522 y la Raspberry Pi	

Fuente: (Elaboración Propia)

El programa que valide los permisos en base a las restricciones alojadas en la base de datos, es parte importante del servidor, pues con esta se otorgara o negara la autorización de ingreso, esta historia de usuario es descrita en la tabla 3.5.

Tabla 3.5: Programa de validación de permisos

Historia de Usuario	
ID: # 4	**Usuario:** Desarrollador
Nombre: Programa de validación de permisos	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Baja
Programador Responsable: Gabriel Casas Mamani	
Descripción El programa que valide los permisos deberá contemplar el tipo de restricción por: Lugar, Día, Hora, con el fin de contar con una respuesta positiva o negativa, resultante del análisis de las restricciones, y poder otorgar o negar la autorización de ingreso.	
Observación Considerar varios días, muchos lugares, o un intervalo de horas	

Fuente: (Elaboración Propia)

La comunicación entre el sistema de Control de Ingresos no Autorizados y el dispositivo móvil es esencial para efectivizar la notificación, al detectar un intento de ingreso no autorizado, por lo que la historia de usuario para esta se describe en la tabla 3.6.

Tabla 3.6: Medio de comunicación

Historia de Usuario	
ID: # 5	**Usuario:** Desarrollador
Nombre: Establecer el medio de comunicación	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Baja
Programador Responsable: Gabriel Casas Mamani	
Descripción Se establece el medio de comunicación para notificar al dispositivo móvil, esta comunicación debe adecuarse al concepto de Internet de las Cosas	
Observación Utilizar un protocolo de comunicación	

Fuente: (Elaboración Propia)

Atreves del medio de comunicación, también se deben poder enviar comandos desde el dispositivo móvil al sistema servidor, esta historia de usuario de aprecia en la tabla 3.7.

Tabla 3.7: Recepción de comandos en el sistema servidor

Historia de Usuario	
ID: # 6	**Usuario:**
Nombre: Recepción de comandos en el sistema servidor	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Baja
Programador Responsable: Gabriel Casas Mamani	
Descripción El sistema servidor debe ser capaz de decepcionar comandos procedentes del dispositivo móvil, para ejecutar acciones como el bloqueo de la lectura de tarjetas, autorización de acceso mediante una contraseña.	
Observación Los comandos no deben ser complejos, se recomienda usar palabras entendibles y fáciles de recordar.	

Fuente: (Elaboración Propia)

Para controlar las autorizaciones, el sistema trabaja con restricciones, mismos que deben adecuarse a las necesidades de quien desea controlar los ingresos, la historia de usuario para este caso se la describe en la tabla 3.8.

Tabla 3.8: Definición de restricciones

Historia de Usuario	
ID: # 7	**Nombre:** Definición de restricciones
Usuario: Usuario del Sistema	**Categoría:** Sistema servidor
Descripción Como usuario, requiero establecer mis propias restricciones para controlar los ingresos que el sistema vaya a controlar.	

Fuente: (Elaboración Propia)

El usuario final será notificado en su dispositivo móvil cuando el sistema registre un intento de ingreso no autorizado, para ello se crea la historia de usuario en la tabla 3.9.

Tabla 3.9: Notificación al dispositivo móvil

Historia de Usuario	
ID: # 8	**Nombre:** Notificación al Dispositivo móvil
Usuario: Usuario del Sistema	**Categoría:** Dispositivo móvil
Descripción Como usuario, me interesa recibir notificaciones de parte del sistema, siempre que se haya detectado un ingreso no autorizado, para luego poder tomar una acción de bloqueo.	

Fuente: (Elaboración Propia)

El usuario final pobra acceder a la parte administrativa del sistema esta historia de usuario se describe en la tabla 3.10.

Tabla 3.10: Administración del sistema

Historia de Usuario	
ID: # 9	**Nombre:** Administración del sistema
Usuario: Usuario del Sistema	**Categoría:** Sistema Servidor
Descripción Como usuario deseo poder acceder a la parte administrativa, contemplando métodos de seguridad, para realizar acciones como agregar nuevas restricciones, agregar nuevos usuarios, eliminar usuarios, disponer del estado de activación o desactivación de tarjetas.	

Fuente: (Elaboración Propia)

3.2.2. Diagramas para el modelado del sistema

Siguiendo la fase uno del Modelo en V, se adoptaran los siguientes diagramas que forman parte de un modelo UML: casos de uso, que representa algún proceso de un usuario o actor con el sistema.

Diagrama de secuencias, que representa la secuencia de mensajes entre componentes, subsistemas o actores, este diagrama se adecua perfectamente para describir el funcionamiento que se muestra en la tabla 3.1.

3.2.2.1. Diagrama de Casos de Uso

La figura 3.1 Muestra el diagrama de casos de uso empleado para describir las acciones que ejerce un usuario administrador sobre el sistema, el cual es encargado de otorgar o cambiar permisos de ingreso a diferentes personas, con el criterio de distintos ambientes, días determinados o diferentes horas, este diagrama sigue lo descrito en la historia de usuario número nueve.

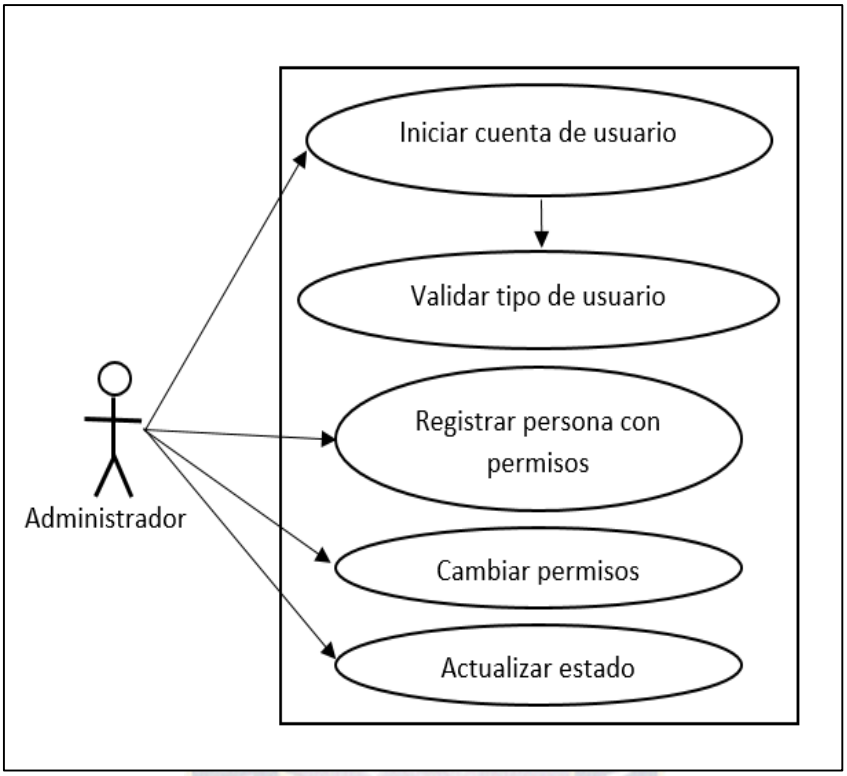


Figura 3.1: Diagrama de casos de uso para la administración de permisos.
Fuente: (Elaboración propia).

Como parte de la documentación del diagrama de casos de uso se menciona a los actores que son parte de esta: tenemos al administrador y al sistema, se describe cada caso de uso de la figura 3.1, se describe en la tabla 3.11 el primer caso de uso que es la acción de inicio de sesión en el sistema por parte del administrador.

Tabla 3.11: Caso de uso, Iniciar cuenta de usuario.

Caso de uso	Iniciar cuenta de usuario.
Descripción	El usuario ingresa mediante un nombre y una contraseña, con el fin de administrar lo que le compete.
Precondición	Debe estar registrado en el sistema, con algún nivel de usuario.
Actores	Usuario, Sistema.
Condición de fracaso	Se niega el acceso, por intentos repetitivos.
Condición de éxito	Acceso correcto al sistema.

Fuente: (Elaboración propia).

Se describe en la tabla 3.12, el segundo caso de uso, el cual trata de la validación que el sistema efectúa para dar acceso al administrador.

Tabla 3.12: Caso de uso, Validar tipo de usuario.

Caso de uso	Validar tipo de usuario.
Descripción	EL sistema realiza la validación el nivel asignado de usuario.
Precondición	Haber iniciado con una cuenta correcta.
Actores	Sistema.
Condición de fracaso	No cargar los roles de usuario.
Condición de éxito	Cargar los roles que le corresponda al usuario.

Fuente: (Elaboración propia).

Al finalizar la validación el administrador dispone de ciertas acciones sobre el sistema, en la tabla 3.13 se describe el primero de ellos que consiste en el registro de nuevas personas con alguno de los permisos de ingreso.

Tabla 3.13: Caso de uso, Registrar persona con permisos.

Caso de uso	Registrar persona con permisos.
Descripción	El usuario podrá crear nuevas personas concediendo permisos de ingreso, siguiendo el criterio de lugar, día, hora.
Precondición	Tener el rol correspondiente para realizar estas acciones.
Actores	Usuario, Sistema.
Condición de fracaso	Declinar la creación de los registros de persona con permisos de ingreso.
Condición de éxito	Registrar a la persona con los permisos concedidos.

Fuente: (Elaboración propia).

En la tabla 3.14 se describe la segunda acción que el administrador puede realizar sobre el sistema, el mismo consiste en el cambio de los permisos que una persona pasee en ese momento.

Tabla 3.14: Caso de uso, Cambiar permisos.

Caso de uso	Cambiar permisos.
Descripción	El usuario podrá cambiar los permisos de ingreso a personas.
Precondición	Tener el rol correspondiente para realizar estas acciones.
Actores	Usuario, sistema.
Condición de fracaso	Mantener los permisos anteriores.
Condición de éxito	Registrar los nuevos permisos de ingreso.

Fuente: (Elaboración propia).

En la tabla 3.15, se describe la tercera acción que el administrador puede realizar, el mismo consiste en actualizar el estado de la identificación por radiofrecuencia pudiendo tomar los valores de activo o bloqueado.

Tabla 3.15: Caso de uso, Actualizar estado.

Caso de uso	Actualizar estado.
Descripción	El usuario podrá asignar un nuevo estado de la tarjeta RFID, los cuales son: Activo, Bloqueado.
Precondición	Tener el rol correspondiente para realizar estas acciones.
Actores	Usuario, Sistema
Condición de fracaso	Mantener el estado anterior.
Condición de éxito	Registrar el nuevo estado asignado.

Fuente: (Elaboración propia).

Siguiendo la historia de usuario número cuatro, la figura 3.2 muestra el diagrama de casos de uso empleado para describir la intervención que tiene el sistema entre una persona y el ingreso a un ambiente operado por un actuador. El sistema verificara los permisos que posee dicha persona, con el fin de validar si está en la hora correcta, el día permitido y en el lugar al que fue autorizado, si el sistema responde de forma afirmativa a todas estas restricciones se manda una señal al sistema de actuación.

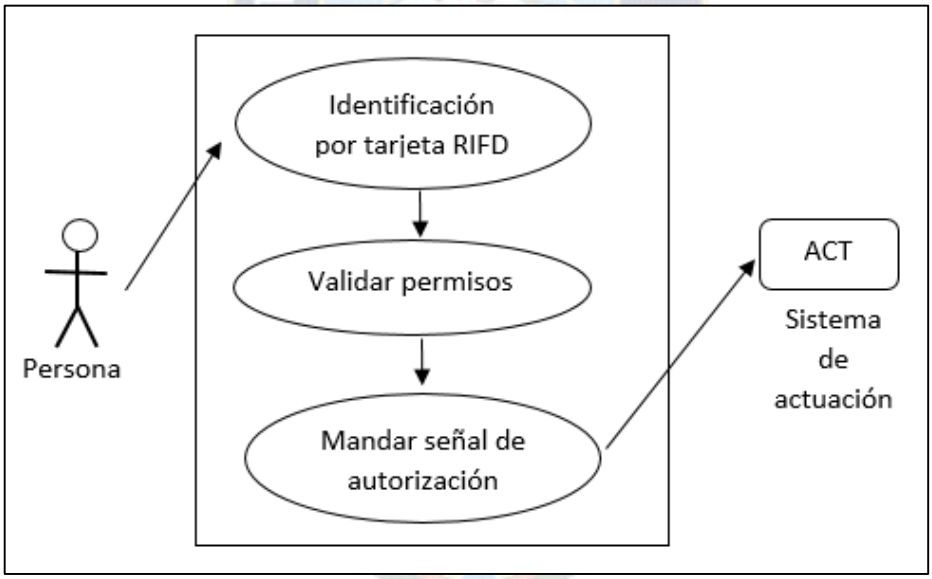


Figura 3.2: Diagrama de casos de uso para el ingreso autorizado de una persona
Fuente: (Elaboración Propia)

Para la documentación de este caso de uso se menciona a los actores de este diagrama, los cuales son la persona a ser identificada, el sistema de verificación de permisos, y el sistema de actuación.

En la tabla 3.16 se describe el primer caso de uso que consiste en la identificación por RFID.

Tabla 3.16: Caso de uso, Identificación por RFID.

Caso de uso	Identificación por RFID.
Descripción	EL sistema realiza la identificación correspondiente a alguna persona.
Precondición	Estar registrado con algún permiso.
Actores	Persona, Sistema.
Condición de fracaso	Rechazar y notificar el intento de ingreso como no autorizado.
Condición de éxito	Proceder a validar los permisos que posee la persona.

Fuente: (Elaboración propia).

En la tabla 3.17, se describe el segundo caso de uso que consiste en la intervención del sistema para efectuar la validación de permisos que tiene la persona.

Tabla 3.17: Caso de uso, Validar permisos.

Caso de uso	Validar permisos.
Descripción	EL sistema realiza la validación de los permisos que se concedió a la persona.
Precondición	Haber realizado la identificación por RFID.
Actores	Sistema.
Condición de fracaso	Registrar y notificar el intento de ingreso como no autorizado.
Condición de éxito	Responder a la validación de manera positiva.

Fuente: (Elaboración propia).

En la tabla 3.18, se describe el tercer caso de uso que consiste en la autorización del ingreso, siempre que se verificaran los permisos y se cuenta con una respuesta positiva.

Tabla 3.18: Caso de uso, Mandar señal de autorización.

Caso de uso	Mandar señal de autorización.
Descripción	EL sistema efectúa él envió de una señal para accionar el actuador.
Precondición	Haber obtenido una respuesta positiva de la validación.
Actores	Sistema, Actuador.
Condición de fracaso	Preservar el estado de no acceso al ambiente.
Condición de éxito	Proceder a mandar la señal para accionar el actuador.

Fuente: (Elaboración propia)

3.2.2.2. Diagrama de Secuencias

El diagrama de secuencias describe de mejor manera el funcionamiento del sistema, en este intervienen la persona, el sistema receptor que es el encargado de realizar la identificación por radiofrecuencia, la base de datos en Cassandra donde se encuentran los permisos, el sistema de actuación que acciona el ingreso, y la aplicación móvil encargado de recibir notificaciones que el sistema genera.

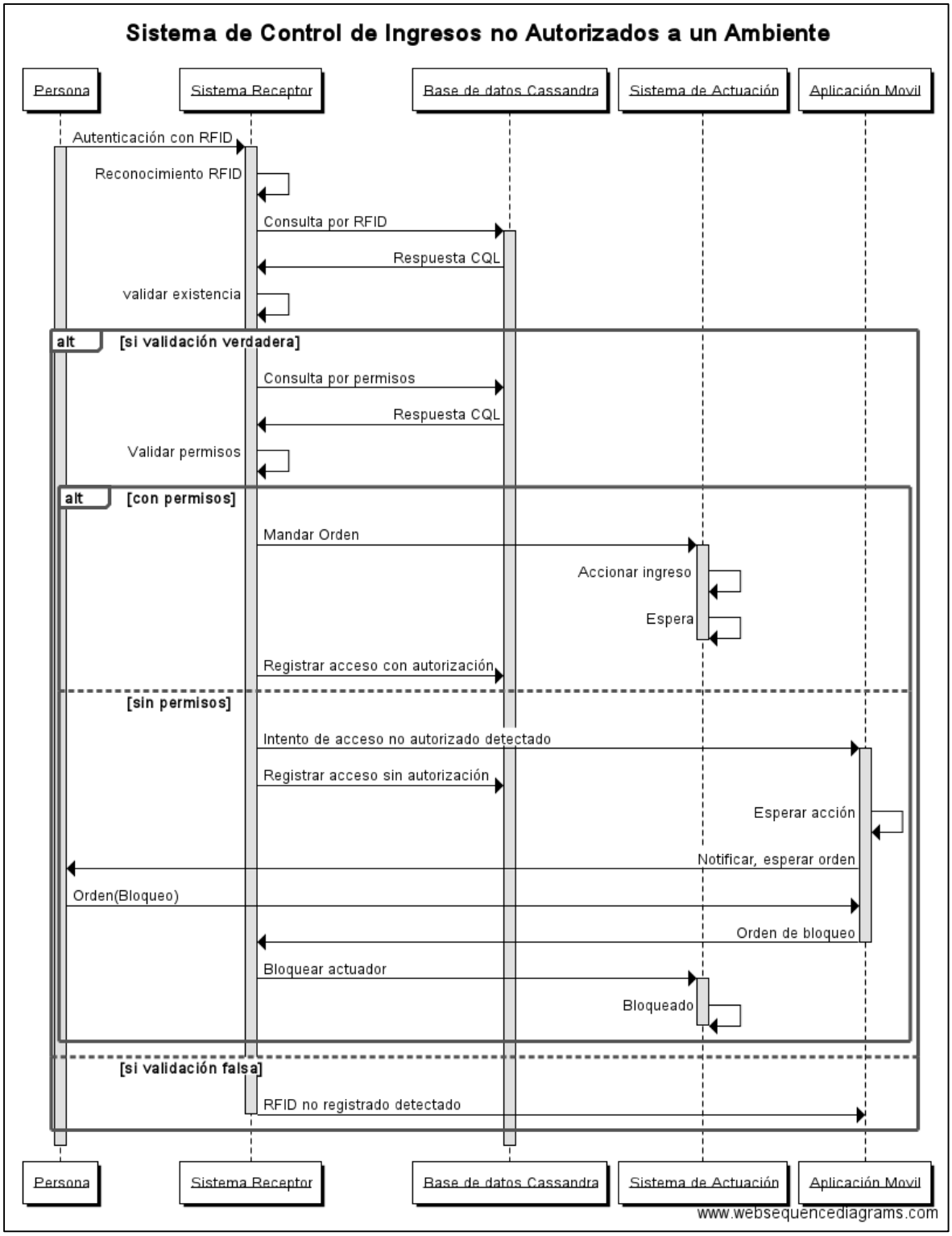


Figura 3.3: Diagrama de Secuencias del sistema de control
Fuente: (Elaboración Propia)

3.3. Fase 2. Funcional

Partiendo de las especificaciones de la fase uno, se establece el diagrama de clases que representa al sistema, también el modelo de datos no relacional en Cassandra.

3.3.1. Diagrama de clases

Siguiendo la historia de usuario número tres, se muestra en la figura 3.4 el diagrama que describe las clases que componen al sistema y la relación entre las mismas, dando a conocer las funciones con las que se contara.

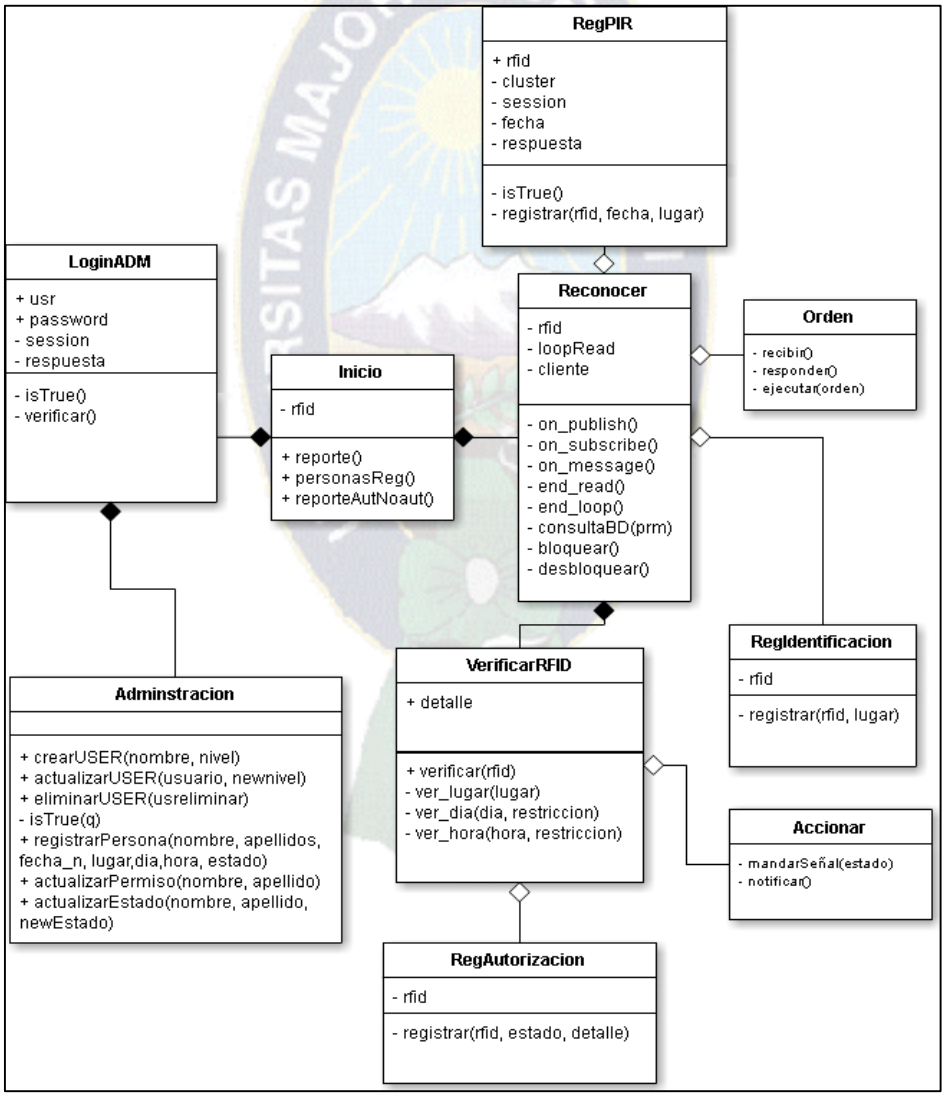


Figura 3.4: Diagrama de clases del sistema de control
Fuente: (Elaboración Propia)

3.3.2. Modelo de datos en Cassandra

Debido a que se generaran datos con demasiada frecuencia por parte del sensor PIR y la Identificación por Radio Frecuencia, y tomando en cuenta que la cantidad de los mismos tiende a crecer bastante con el tiempo, se adopta este modelo de datos debido a que es más tolerante a fallos por la eficiencia en el registro y la respuesta a las consultas.

Haciendo uso de la historia de usuario número uno se inicia con el diseño de la base de datos, para lo cual se debe definir los requerimientos a los que deberá responder:

- Obtener los registros de ingresos a algún ambiente.
- Obtener las horas en los que las personas se aproximan a los ingresos.
- Listar los permisos concedidos a personas.
- Listar las horas, días y lugares de ingresos de una persona.
- Obtener aquellos ingresos con y sin autorización.
- Registrar usuarios de administración con sus niveles y roles.

En la figura 3.5 se muestra el modelado de datos para la tabla que registra la identificación de una persona por medio de la tarjeta RFID, al realizar un ingreso a algún ambiente.

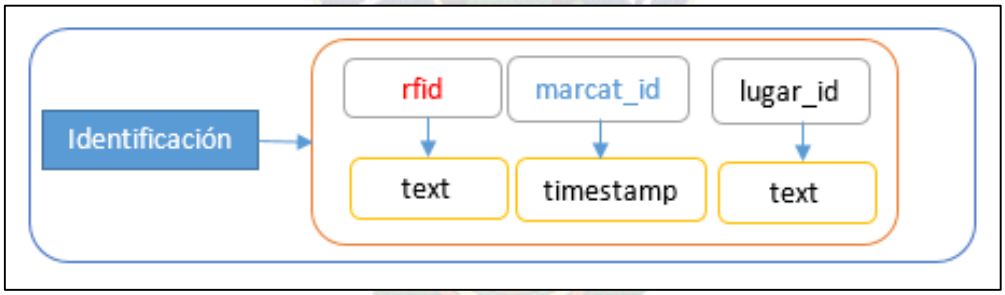


Figura 3.5: Modelado de identificación
Fuente (Elaboración Propia)

En la figura 3.6 se muestra el modelado de datos para la tabla presencia, que registra la aproximación suficiente de una persona a algún ingreso, para realizar su identificación.

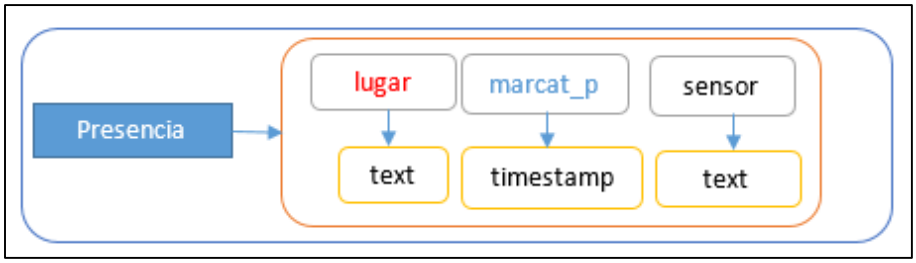


Figura 3.6: Modelado de presencia
Fuente (Elaboración Propia)

En la figura 3.7 se muestra el modelado de datos para la tabla que registra los datos necesarios para identificar a una persona incluyendo el ID de la tarjeta de radio frecuencia con los permisos concedidos, para realizar los ingresos.

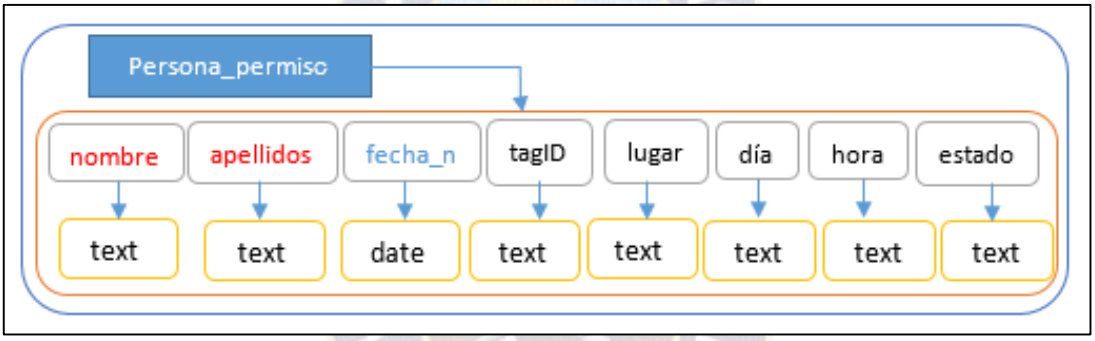


Figura 3.7: Modelado de personas con permisos
Fuente (Elaboración Propia)

En la figura 3.8 se muestra el modelado de datos para la tabla que registra, la existencia o no de autorización de una persona, al intentar realizar un ingreso a un ambiente.

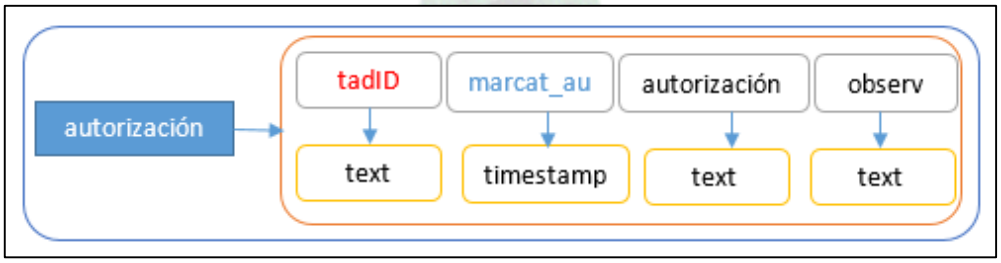


Figura 3.8: Modelado de autorización
Fuente (Elaboración Propia)

En la figura 3.9 se muestra el modelado de datos para la tabla que registra los usuarios que pueden acceder a la parte administrativa (ver figura 3.1) del sistema restringidos por un nivel, y roles asignados.

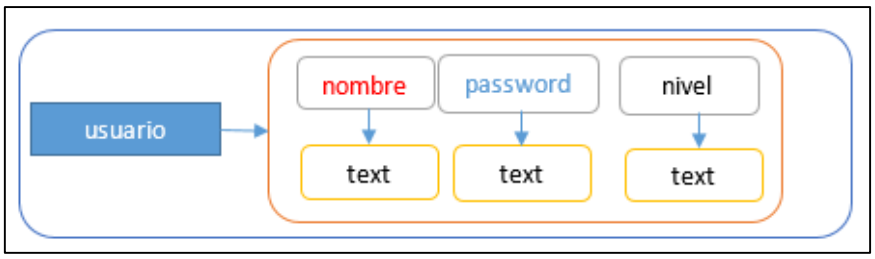


Figura 3.9: Modelado de usuario
Fuente (Elaboración Propia)

En la figura 3.10 se muestra el modelado de datos para la tabla que registra los roles o permisos con los que cuentan los administradores del sistema, según el nivel asignado.

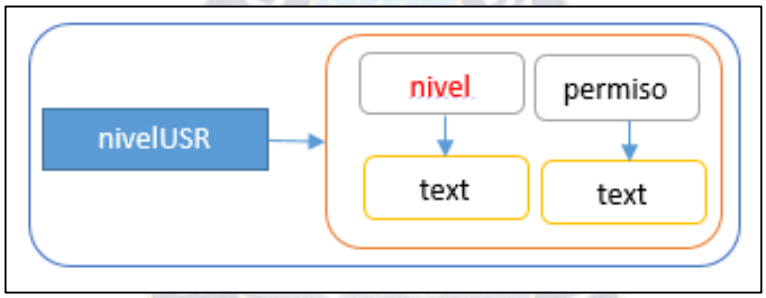


Figura 3.10: Modelado de nivel de usuario
Fuente (Elaboración Propia)

3.3.3. Roles de usuario

El sistema contempla un lado de administración, en el cual los usuarios que cuenten con el acceso, pueden realizar acciones de registro, eliminación y modificación sobre la base de datos en Cassandra.

Está claro que si se va a conceder acceso a los datos contenidos en la base de datos, se debe restringir cualquier alteración inadecuada en la misma, para proteger la información que se puede obtener a partir de estos datos.

En la base de datos Cassandra se contempla el registro de niveles de usuarios, mismos que están encargados de restringir a los usuarios que tienen la autorización de realizar modificaciones con los roles administrativos correspondientes sobre el sistema, se identificaron tres niveles iniciales para cada tipo de usuario, cada uno de ellos tienen diferentes roles, que se describen en la tabla 3.19.

Tabla 3.19: Definición de roles por niveles de usuarios

Nivel	Descripción de roles
NSU	Nivel se Súper Usuario: Este nivel es el más alto considerado, en este nivel se pueden se tienen los siguientes roles, creación de nuevos usuarios, eliminar usuarios, registrar personas, cambiar permisos, actualizar estados.
NU1	Nivel de Usuario 1: Este nivel es el de nivel medio que tiene acciones más restringidas que el NSU, en este nivel se tiene los siguientes roles, registrar personas, cambiar permisos, actualizar estados.
NU2	Nivel de Usuario 2: Este nivel es el más bajo y está restringido a solo dos roles, el registro de personas, actualizar estados.

Fuente: (Elaboración Propia)

3.4. Fase 3. Diseño

En la fase de diseño se identificaran tanto los componentes de software como los de hardware que se emplearan en el prototipo, esta identificación se la debe realizar tomando en cuenta lo descrito en la historia de usuario número dos.

En esta fase el Modelo en V, se refiere a la precisión que se desea alcanzar para medir la variable controlada, en ese sentido se deben identificar los sensores adecuados que deben ser capaces de medir dicha variable y logar la precisión aceptable para que el sistema cumpla el propósito planteado.

3.4.1. Identificación de componentes de hardware

Estos componentes son esenciales, para que el software que se desarrollara en esta investigación pueda actuar sobre ellos y les de la funcionalidad que el Sistema de Control de Ingresos no Autorizados a un Ambiente necesita, cada uno de ellos cuenta con una función diferente que se describen en la tabla 3.20:

Tabla 3.20: Descripción de componentes de Hardware

Componente	Descripción
Raspberry Pi	Función de servidor para alojar y poner en marcha al sistema.
Módulo RFID	Proporciona la identificación de radio frecuencia que es enviada al sistema, para su interpretación.
PIR	Sensor de identificación de presencia, cada dato será enviado al sistema alojado en la Raspberry Pi para su interpretación.
RELÉ	Este dispositivo es el que transforma las señales transmitidas por la Raspberry Pi, en acción física para activar el circuito del actuador.
Dispositivo Móvil	Receptor de las notificaciones generadas por el sistema al detectar un acceso no autorizado.

Fuente: (Elaboración Propia)

3.4.2. Identificación de componentes de software

Cada uno de los dispositivos de hardware mencionados en la tabla 3.19, forman parte del Sistema de Control de Ingresos no Autorizados a un Ambiente, para que los mismos tengan el funcionamiento correcto, es necesario hacer uso de librerías, drivers y programas determinados, estas también son necesarias para desarrollar el software que verificara las restricciones, por tanto son descritos en la tabla 3.21:

Tabla 3.21: Descripción de componentes de Software.

Componente	Descripción
Python	Lenguaje de programación que es base para soportar todas las librerías y drivers, que el sistema necesita.
RPi.GPIO	Librería necesaria para poder controlar los puertos GPIO de la Raspberry Pi por medio del lenguaje Python.
Apache Cassandra	Útil para soportar la gran cantidad de datos generados, valiosa por su disponibilidad y escalabilidad sin comprometer el rendimiento.
cassandra-driver	Necesario para crear instancias de cluster, el cual nos permite conectarnos a la base de datos de cassandra y hacer consultas a través del lenguaje Python.
MFR22	Librería que permite que el módulo RFID-RC522 se integre con la Raspberry Pi, para efectuar la identificación de las tarjetas de radio frecuencia.

SPI-Py	Librería que permite manipular la interfaz SPI de la Raspberry Pi, y realizar conexiones de este tipo.
Mosquitto	Librería más completa de clientes MQTT para Python.
Android Studio	Esencial para producir la aplicación móvil que recibirá las notificaciones que genere el sistema.
Libretita Paho	Cliente para Java/JMV, que tiene los protocolos MQTT necesarios para efectuar la comunicación desde el lenguaje Java.

Fuente (Elaboración Propia)

3.4.3. Arquitectura del prototipo

Descritos los componentes de software y hardware que se emplea en el Sistema de control de Ingresos no Autorizados a un Ambiente, en la figura 3.11 se representa el conjunto que conforma el producto final, el Modelo en V llama a esta, arquitectura del sistema.

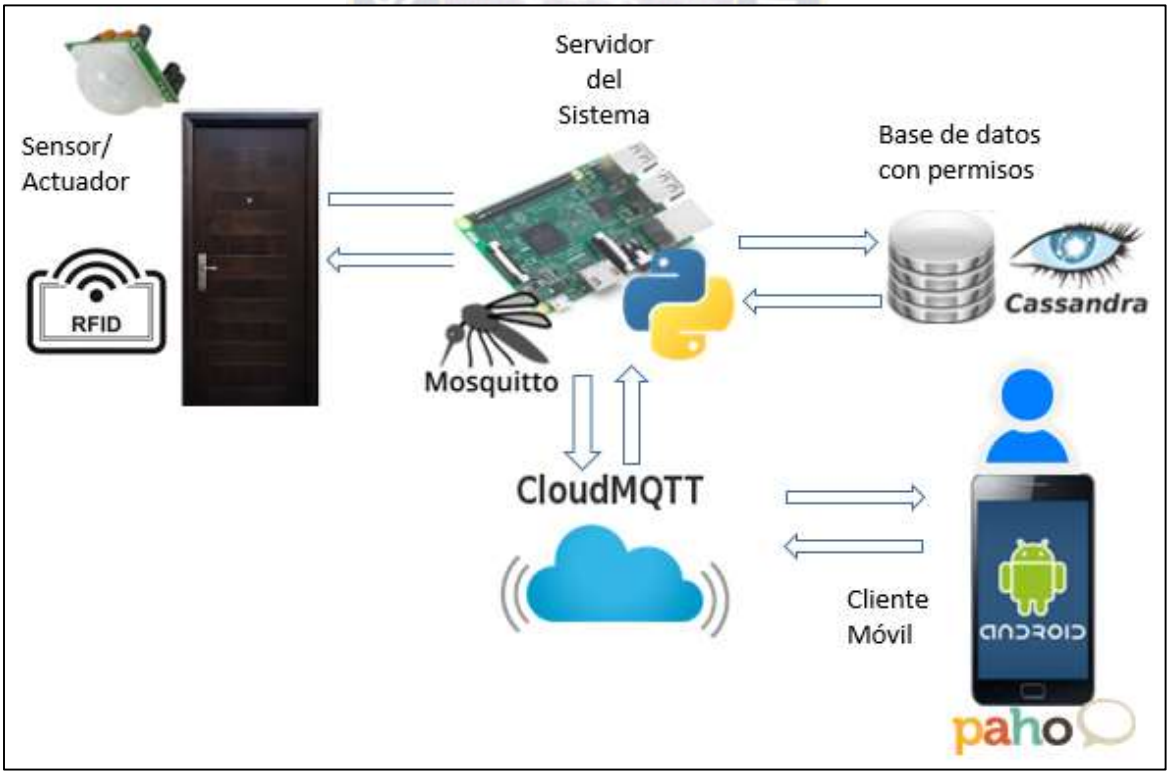


Figura 3.11: Arquitectura Funcional del prototipo
Fuente:(Elaboración Propia)

El hardware esencial para integrar el funcionamiento del sistema es la Raspberry Pi, en la misma se aloja la base de datos Cassadra, el sistema de identificación, validación, y administración de permisos y mediante ella se transmiten los mensajes en forma de notificaciones al dispositivo móvil.

En la figura 3.12 se muestra la manera adecuada de conexión entre la interfaz SPI de la Raspberry Pi y el Modulo RFID, la misma debe estar debidamente configurada para lograr la comunicación entre ambos, esta configuración se la efectúa en la parte de codificación del Modelo en V, ya que esta fase es netamente de diseño, el esquema resultante es:

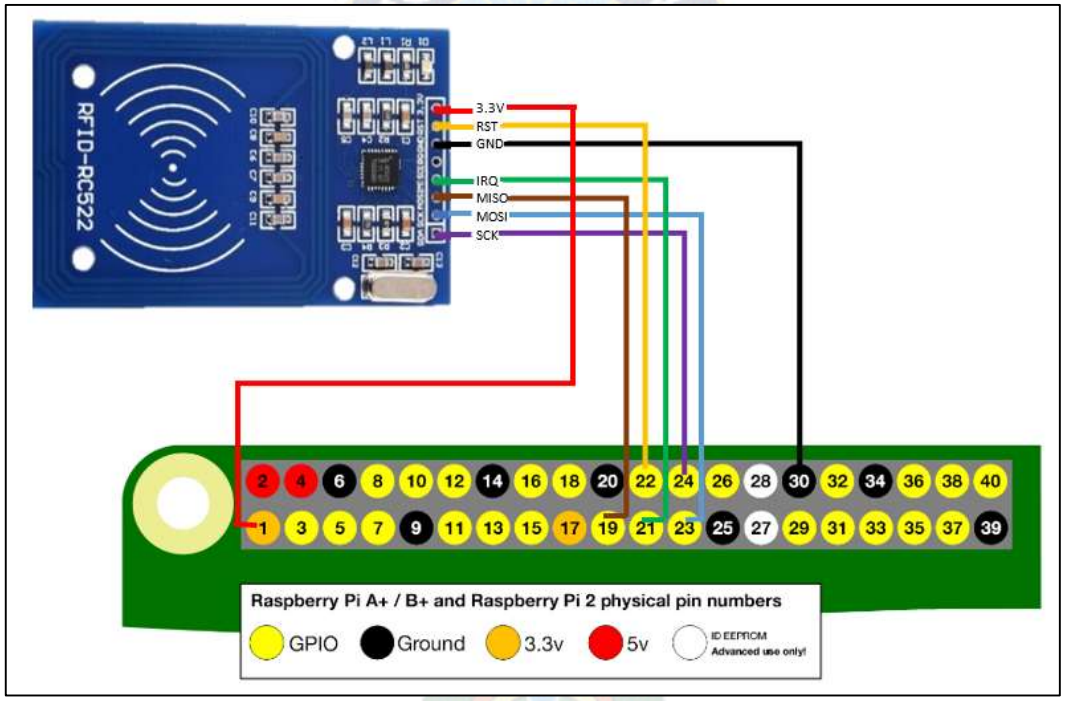


Figura 3.12: Conexión entre la Raspberry Pi y el Modulo RFID
Fuente: (Elaboración Propia)

En la figura 3.13 se aprecia la conexión de la Rapberry Pi con el módulo PIR, mismo que permite la identificación de presencia en cercanías de los ingresos que son controlados por el sistema, el sensor responde emitiendo señales lógicas tras detectar la un cuerpo que posea un temperatura superior a los 0°K, la cual es interpretada como radiación infrarroja negativa.

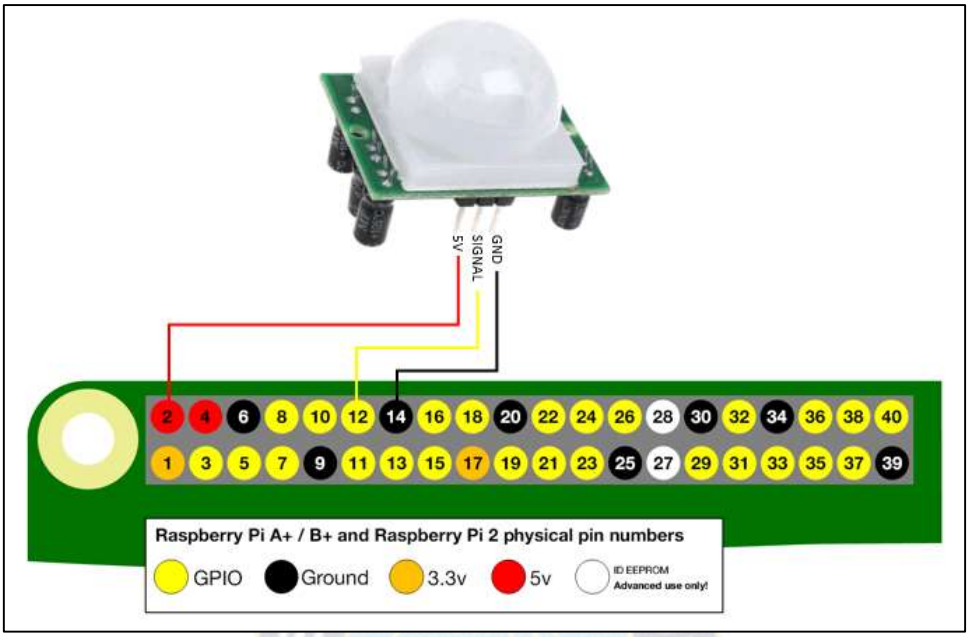


Figura 3.13: Conexión entre la Raspberry Pi y el Módulo PIR
Fuente: (Elaboración Propia)

En la figura 3.14 se aprecia la conexión de la Rapberry Pi con el RELÉ, que permite la recepción y transformación de señales emitidas después de validar un ingreso autorizado, para acciona un circuito en el actuador que está compuesta por una chapa eléctrica y la alimentación de la misma.

Para proteger el RELÉ, el circuito del actuador, y la conexión a la Raspberry Pi, se necesita de una conexión previa con los siguientes componentes electrónicos mostrados en la tabla 3.22:

Tabla 3.22: Componentes para el RELÉ

Componente	Descripción
R1	Resistencia de 1k con capacidad de 1/4w
D1	Diodo 1N4001
TR1	Transistor BD137

Fuente: (Elaboración Propia)

Descritos los componentes se presenta, en la figura 3.14 la conexión final entre la Raspberry Pi, el RELÉ, y la chapa que funciona como el actuador.

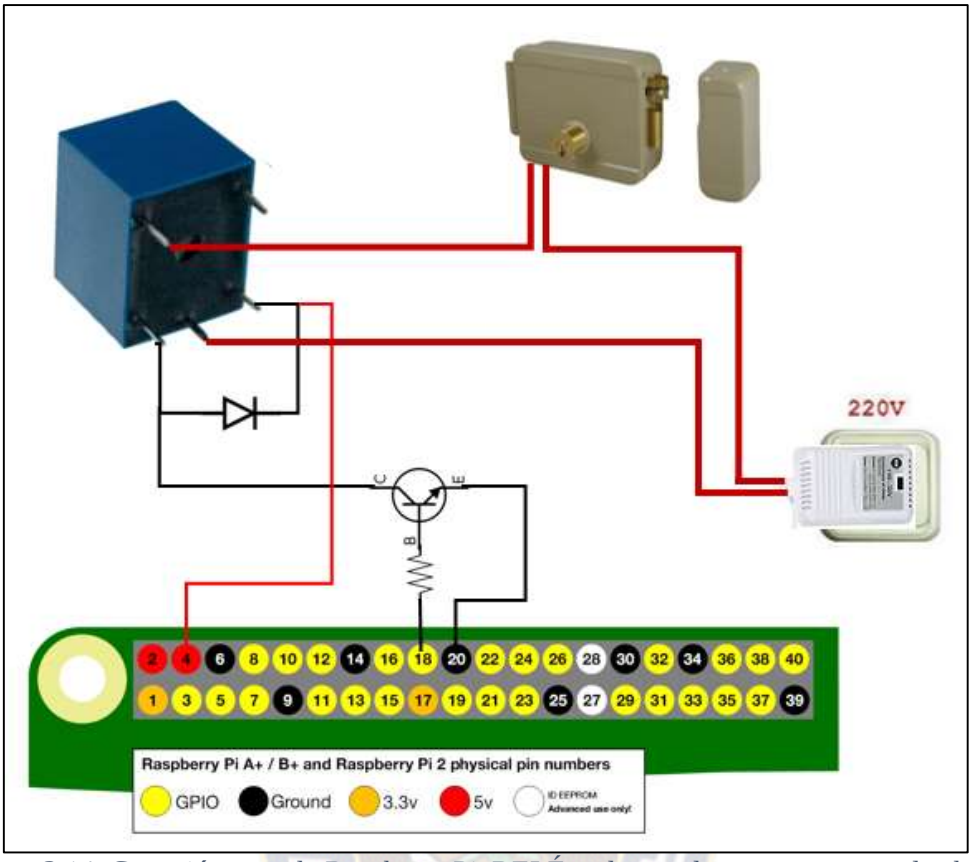


Figura 3.14: Conexión entre la Raspberry Pi, RELÉ y el actuador compuesto por la chapa
Fuente: (Elaboración Propia)

3.5. Fase 4. Codificación

En esta fase se lleva a cabo la programación del sistema aplicando los componentes de software mencionados en la tabla 3.21, la misma deberá adecuarse al funcionamiento de cada componente de hardware, materializando la funcionalidad que se describió en los diagramas de las fases anteriores, para esto se elabora un plan de trabajo en términos de las especificaciones de la primera fase que aún deben ser realizadas:

- Diseñar un sistema capaz de identificar, mediante RFID aquellos accesos que cuenten con las autorizaciones correspondientes.
- Diseñar la aplicación móvil, que deberá recibir notificaciones emitidas por la Raspberry Pi al identificar un intento de ingreso no autorizado.

3.5.1. Sistema de identificación

La primera especificación a tratar es la identificación por radiofrecuencia, que es la parte principal del sistema de Control de Ingresos no Autorizados a un Ambiente, mediante esta se obtiene el código de identificación asignado a cada persona sometida al control y se verifica los permisos que tiene comparando con las restricciones que existe en la base de datos, para cumplir la especificación se establece el siguiente plan de trabajo de la tabla 3.23:

Tabla 3.23: Plan de trabajo codificación del sistema

Actividad	Para	Responsable
Habilitar la interfaz ISP de la Raspberry Pi	Utilizar comunicación por interfaz ISP	Desarrollador
Instalar el controlador ISP-Py y MFRC522-python	Integrar Modulo RFID y Raspberry Pi mediante python	Desarrollador
Elaborar el algoritmo para obtener el código de identificación que poseen las tarjetas RFID	Obtener y verificar el código de cada tarjeta RFID	Desarrollador
Establecer las restricciones que se deberán considerar para otorgar permisos	Para poder conceder permisos en base a las restricciones.	Desarrollador Interesado del control

Fuente: (Elaboración Propia)

Para habilitar la interfaz ISP de la Raspberry Pi se debe editar el archivo de configuración ubicado en la dirección */boot/config.txt*, en él se debe agregar las siguientes líneas que muestra la figura 3.15:

```
device_tree_param = = spi en
dtoverlay = spi-bcm2708
```

Figura 3.15: Instrucciones para habilitar interfaz ISP
Fuente: (Elaboración Propia)

También se debe eliminar de la lista negra la comunicación de la tarjera con la interfaz ISP dándole así funcionalidad a la misma, la lista negra de la Raspberry Pi se encuentra en la siguiente dirección */etc/modprobe.d/raspi-blacklist.conf* en la misma se debe comentar la línea *blacklist spi-bcm2708* para que la configuración se efectivice, se reinicia el sistema operativo.

Los controladores ISP-Py y MFRC522-python se los puede encontrar en los repositorios libres de GitHub, para poder usarlos se debe hacer una clonación de los mismos desde la terminal.

Los comandos de la figura 3.16, son para la obtención e instalación del controlador ISP-Py, los mismos se los debe introducir en una terminal:

```
$ sudo apt-get install python-dev
$ git clone https://github.com/lthiery/SPI-Py.git
$ cd SPI-Py
$ sudo python setup.py install
```

Figura 3.16: Instalación de controladores ISP-Py
Fuente: (Elaboración Propia)

Los comandos de la figura 3.17, son para la obtención e instalación del controlador MFRF522, los mismos se los debe introducir en una terminal:

```
$ git clone https://github.com/mxgxw/MFRF522-python.git
$ cd MFRC522-python
```

Figura 3.17: Instalación de controladores y MFRC522-python
Fuente: (Elaboración Propia)

El código en python de la figura 3.18 es para la lectura de tarjetas RFID, el mismo contempla lo siguiente: la lectura constante de alguna tarjeta RFID, que el estado de lectura sea verificable durante todo el programa, la obtención del código de identificación que contiene la tarjeta, la orden para que se detenga la lectura de las tarjetas RFID.

```
#!/usr/bin/env python
# -*- coding: utf8 -*-
import RPi.GPIO as GPIO
import MFRC522
import signal
continue_reading = True
def end_read(signal,frame):
    global continue_reading
    print "Ctrl+C, detectado, Deteniendo la lectura."
    continue_reading = False
    GPIO.cleanup()
signal.signal(signal.SIGINT, end_read)
MIFAREReader = MFRC522.MFRC522()
print "Bienvenido al sistema de control de accesos por RFID "
print "Presione Ctrl-C para detener"
while continue_reading:
    (status,TagType) =
MIFAREReader.MFRC522_Request(MIFAREReader.PICC_REQIDL)
    if status == MIFAREReader.MI_OK:
        print "Tarjeta Detectada"
    (status,uid) = MIFAREReader.MFRC522_Anticoll()
    if status == MIFAREReader.MI_OK:
        print "Card read UID:
"+str(uid[0])+","+str(uid[1])+","+str(uid[2])+","+str(uid[3])
        key = [0xFF,0xFF,0xFF,0xFF,0xFF,0xFF]
        MIFAREReader.MFRC522_SelectTag(uid)
        status =
MIFAREReader.MFRC522_Auth(MIFAREReader.PICC_AUTHENT1A, 8, key, uid)
        if status == MIFAREReader.MI_OK:
            MIFAREReader.MFRC522_Read(8)
            MIFAREReader.MFRC522_StopCrypto1()
        else:
            print "error de autenticación"
```

Figura 3.18: Código de Identificación por RFID
Fuente: (Elaboración, Basado en Git)

En el plan de trabajo se contempla la descripción de los permisos que se deben considerar para otorgar los ingresos con plena autorización, para lo cual ya se diseñó el modelo de datos, pero no se mencionó aun el modo en que los permisos son verificados por el sistema, en esta fase se describe el código funcional que realiza esta verificación.

Los permisos son concedidos a personas registradas con ciertas restricciones siguiendo el criterio de hora correcta, día permitido, y lugar al que fue autorizado, de este modo el sistema analiza las restricciones que estén en el formato establecido tomando los siguientes casos de la tabla 3.24:

Tabla 3.24: Casos de restricciones, para otorgar permisos

	Caso 1	**Caso 2**	**Caso 3**
Lugar	Sala	Sala, Ingreso, recamara1, recamara2…	*
Día	mon	mon, sat, wed,…	*
Hora	8:30	8:30 – 12:00	*

Fuente: (Elaboración Propia)

Para el caso uno de la restricción hora, se debe establece un límite adecuado para que se restrinja la hora establecida, se permite el ingreso hasta la media noche, por lo que el sistema considera el intervalo 8:30-00:00.

En el caso tres se hace uso del símbolo ***** para representar la ausencia de restricciones, algunas personas podrán contar con el privilegio de tener con acceso total a todos los lugares, sin restricción de días, y sin que la hora de ingreso sea controlada.

Para verificar estas restricciones mismas que se le imponen a cada persona registrada en el sistema, se emplean consultas a la base de datos desde el lenguaje python, la respuesta de estas consultas es analizada por los métodos que responderán de marea afirmativa o negativa a cada restricción, el código de verificación se aprecia en la figura 3.19.

```
import sys, regAutorizacion
from datetime import datetime, time
from cassandra.cluster import Cluster
detalle="detalle:"

def verificar(rfid):
    regresar=False
    global detalle
    f= "%a-%H:%M"  # Asigna un formato de ejemplo1
    hoy = datetime.today()  # Asigna fecha-hora
    diahora = hoy.strftime(f)  # Aplica formato ejemplo1
    print diahora  # Muestra fecha-hora según ejemplo1
    restriccion=diahora.split('-')
    #print restriccion
    cluster = Cluster()
    session= cluster.connect('control_ingresos')
    q="select lugar_permiso,dia_permiso,hora_permiso from persona_permiso"
    "where tagid_permiso='"+rfid+"' and estado='activo' allow filtering"
    res=session.execute(q)
    if res:
        for r in res:
            datos= r.lugar_permiso+"_"+r.dia_permiso+"_"+r.hora_permiso
        dat=datos.split('_')
        if ver_lugar(dat[0])==ver_dia(dat[1], restriccion)==
        ver_hora(dat[2],restriccion)==True:
            #permiso consecido
            detalle=detalle+" CON PERMISO."
            regresar= True
            regAutorizacion.registrar(rfid, "autorizado", detalle)
        else:
            #Sin permiso suficiente
            detalle=detalle+" SIN PERMISO"
            regresar= False
            regAutorizacion.registrar(rfid, "no autorizado", detalle)
        print detalle
        detalle=""
    else:
        print "erro!! El RFID no existe o esta bloqueado"
        regAutorizacion.registrar(rfid, "El RFID no existente o bloqueado")
    return regresar
```

Figura 3.19: Verificación de restricciones
Fuente: (Elaboración Propia)

A este código se le envía como argumento el RFID que es capturado por el código de la figura 3.18, en el proceso de verificación de las restricciones, se hace uso de métodos que a continuación de describirán.

En la figura 3.20 se observa el código que verifica, el lugar donde se realiza la identificación por RFID, este lugar debe coincidir con el que se encuentra registrado en la base de datos

para cada persona. El lugar de verificación está contenido en un archivo único asignado a cada ingreso.

```
def ver_lugar(lugar):
    global detalle
    archivo = open("ubicacion", "r")
    u = archivo.readline()
    if lugar==u:
        detalle+=" lugar correcto,"
        #print "lugar correcto"
        return True
    elif lugar=="*":
        detalle+=" acceso a todos los ambientes,"
        return True
    else:
        detalle=detalle+" lugar no autorizado,"
        return False
```

Figura 3.20: Código de verificación del lugar
Fuente: (Elaboración Propia)

En la figura 3.21 se observa el código que verifica, el día en el que se realiza la identificación por RFID, este día debe coincidir con el que se encuentra registrado en la base de datos para cada persona.

```
def ver_dia(dia, res):
    global detalle
    #res tiene dia hora actual: dia=res[0] hora:res[1]
    if res[0].lower() in dia.split(','):
        detalle=detalle+" dia correcto,"
        #print "dia correcto"
        return True
    elif dia=="*":
        detalle+=" acceso todos los dias,"
        return True
    else:
        detalle=detalle+" dia no autorizado,"
        #print "dia no autorizado"
        return False
```

Figura 3.21: Código de verificación del día
Fuente: (Elaboración Propia)

En la figura 3.22 se observa el código que verifica, la hora en el que se realiza la identificación por RFID, esta hora debe coincidir con el que se encuentra registrado en la base de datos para cada persona.

```
def ver_hora(hora,res):
    #res tiene dia hora actual: dia=res[0] hora:res[1]
    global detalle
    h=hora.split('-')
    if len(h)==2:
        if res[1]>=h[0] and res[1]<=h[1]:
            detalle=detalle+" hora correcta,"
            return True
        else:
            detalle=detalle+" fuera de hora,"
            return False
    elif res[1]>=h[0] and res[1]<="00:00":
        detalle=detalle+" hora correcta,"
        return True
    elif hora[0]=="*":
        detalle=detalle+" acceso a cualquier hora,"
        return True
    else:
        detalle=detalle+" fuera de hora,"
        return False
```

Figura 3.22: Código de verificación de la hora
Fuente: (Elaboración Propia)

3.5.2. Aplicación para recibir notificaciones

La segunda especificación restante a tratar es la aplicación móvil, que deberá recibir notificaciones emitidas por la Raspberry Pi al identificar un intento de ingreso no autorizado, como la aplicación se limita a recibir notificaciones y enviar mensajes al servidor, en la tabla 3.25 se plantea el siguiente plan de trabajo:

Tabla 3.25: Plan de trabajo para la aplicación móvil

Actividad	**Para**	**Responsable**
Identificar el medio de comunicación en términos de IoT	La aplicación se comunique con el servidor de identificación usando IoT	Desarrollador
Identificar la librería adecuada, para la aplicación	La aplicación pueda usar el protocolo adecuado	Desarrollador
Mostrar código que permite la comunicación	Contar con una base para efectivizar la comunicación por el protocolo MQTT	Desarrollador

Fuente: (Elaboración Propia)

La comunicación se basara en el protocolo MQTT, que es usado para la comunicación machine-to-machine (M2M) de IoT, existen librerías para diferentes lenguajes, por lo que su uso es adecuado para el tipo de sistema propuesto.

Se hará uso del servicio Paho de eclipse, Android utiliza maven como gestor de dependencias, por medio de esta se pueden importar las mismas a través de un servidor en internet, la ubicación de cada dependencia que se vaya a utilizar se la debe definir en el sistema de compilación Gradle.

En cada proyecto de Android existen dos archivos de Gradle, la primera es del proyecto y la segunda de los módulos contenidos en ese proyecto, en la figura 3.23 se muestra como añadir el servicio de paho utilizando maven desde el servidor de eclipse.

```
buildscript {
    repositories {
      maven {
          url "https://repo.eclipse.org/content/repositories/paho-releases/"
      }
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.1.2'
    }
}
```

Figura 3.23: Código para agregar servicio paho a proyecto Android
Fuente: (Elaboración Propia)

En la figura 3.24 se muestra como acceder a las dependencias que se encuentran en el repositorio añadido.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
}
```

Figura 3.24: Código de acceso a las dependencias paho
Fuente: (Elaboración Propia)

Cuando se haya agregado el servicio se procede a declarar el mismo en el archivo *AndroidManifest.xml* añadiendo el código de la figura 3.25.

```
<service android:name="org.eclipse.paho.android.service.MqttService"/>
```

Figura 3.25: Código para declarar el servicio paho
Fuente: (Elaboración Propia)

En el mismo *AndroidManifest.xml* se debe dar permisos de trabajo a la aplicación, el código para los permisos se muestra en la figura 3.26.

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Figura 3.26: Código de permisos
Fuente: (Elaboración Propia)

Ya se tiene el servicio paho configurado y funcionando en el proyecto, se muestra en la figura 3.26 el código que realiza la conexión al servidor que transporta mensajes por el protocolo MQTT.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    String clientId = MqttClient.generateClientId();
    MqttAndroidClient client= new MqttAndroidClient(this.getApplicationContext(),
            "m12.cloudmqtt.com:1949",clientId);
    String topic = "GPIO";
    String payload = "hola desde android";
    byte[] encodedPayload = new byte[0];
    try {
        IMqttToken token = client.connect();
        token.setActionCallback(new IMqttActionListener() {
            public static final String TAG = "";
            @Override
            public void onSuccess(IMqttToken asyncActionToken) {
                Log.d(TAG, "onSuccess");
            }
            @Override
            public void onFailure(IMqttToken asyncActionToken, Throwable exception) {
                Log.d(TAG, "onFailure");
            }
        });
        encodedPayload = payload.getBytes("UTF-8");
        MqttMessage message = new MqttMessage(encodedPayload);
        client.publish(topic, message);
        publicar(client);
    } catch (UnsupportedEncodingException  | MqttException e) {
        e.printStackTrace();
    }
}
```

Figura 3.27: Código de conexión al servidor por medio de MQTT
Fuente: (Elaboración Propia)

En la figura 3.28 se puede apreciar la aplicación móvil conectada a internet y recibiendo notificaciones de ingresos autorizados como no autorizados.

Figura 3.28: Aplicación Móvil
Fuente: (Elaboración Propia)

3.5.3. Aplicación de una plataforma para pruebas

La comunicación entre el dispositivo móvil y el Sistema de Control de Ingresos no Autorizados a un Ambiente, se realiza a través de una plataforma que presta servicio bajo el protocolo MQTT llamada CloudMQTT.

CloudMQTT es la plataforma adecuada para llevar a cabo las pruebas de conectividad que el sistema necesita, para efectivizar las notificaciones.

La plataforma nos proporciona acceso a un servidor a través de un puerto con usuario, contraseña, en la tabla se muestra los datos proporcionados por la plataforma para realizar conexiones con el servidor de CloudMQTT.

Tabla 3.26: Datos para conexión a CloudMQTT

Servidor	m12.cloudmqtt.com
Usuario	tqkdfthm
Password	0NNzyRa7-Rvk
Puerto	19494
SSL Port	29494

Fuente: (Elaboración Propia)

Esta plataforma permite crear tópicos mediante estos un usuario conectado puede llevar a cabo suscripciones y publicaciones, para las pruebas se crea el tópico GPIO, RFID, y los usuarios que existen dentro de este tópico son:

Tabla 3. 27: Usuarios de prueba

Usuario	Tópico
usr01	GPIO
usr02	GPIO
usr03	RFID
usr04	RFID

Fuente: (Elaboración Propia)

3.6. Fase 5. Test de Diseño

En el test de diseño se documenta las verificaciones funcionales de forma individual, que se realizaron tanto al software como al hardware descrito en la fase de diseño, esta verificación contempla la plena funcionalidad tanto de las conexiones de hardware, como también del programa desarrollado.

En la Figura 3.29 se observa la conexión física entre el Módulo RFID y la Raspberry Pi, siguiendo el diagrama de la figura 3.12, para realizar las pruebas respectivas.

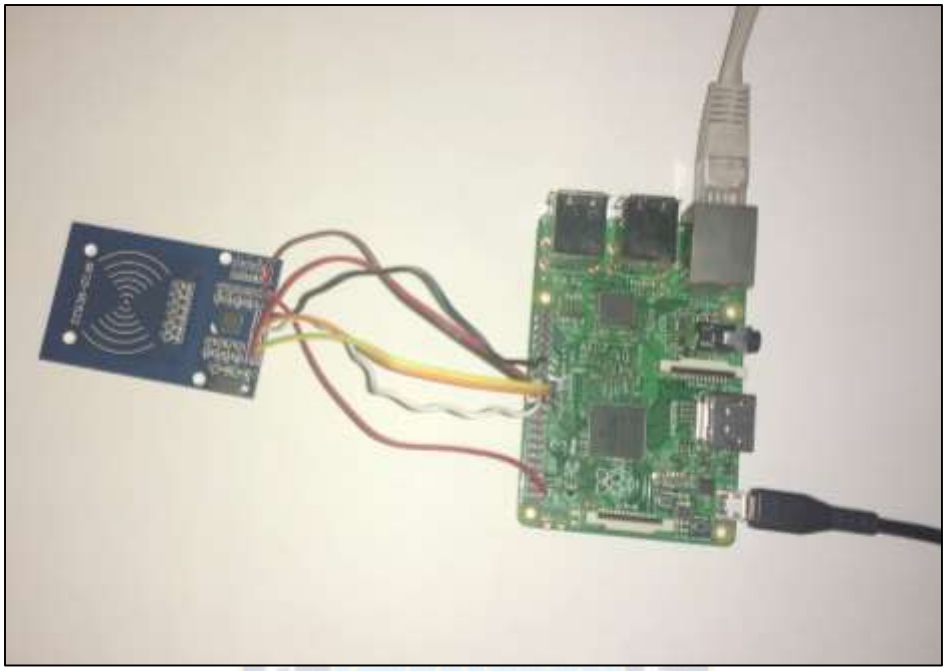


Figura 3.29: Conexión Física de El Modulo RFID y la Raspberry Pi
Fuente: (Elaboración Propia)

Los resultados que se obtuvieron de las pruebas en esta conexión, se describen en la tabla 3.28, como parte esencial de la prueba se identifica aquellos errores que puedan surgir al efectuar la conexión.

Tabla 3.28: Pruebas de conexión entre el Modulo RFID y Raspberry Pi

Documento de Prueba Uno	
Dispositivos	Módulo RFID, Raspberry Pi
Objetivo	Realizar la identificación de las tarjetas RFID, obteniendo el código único de cada una de ellas.
Observación	La conexión respondió de forma favorable al sistema de identificación
Errores	La obtención del código único de cada tarjeta RFID, necesita conversión binaria a hexadecimal

Fuente: (Elaboración Propia)

En la Figura 3.30 se observa la conexión física entre el sensor de presencia PIR y la Raspberry Pi, siguiendo el diagrama de la figura 3.13, para realizar las pruebas respectivas.

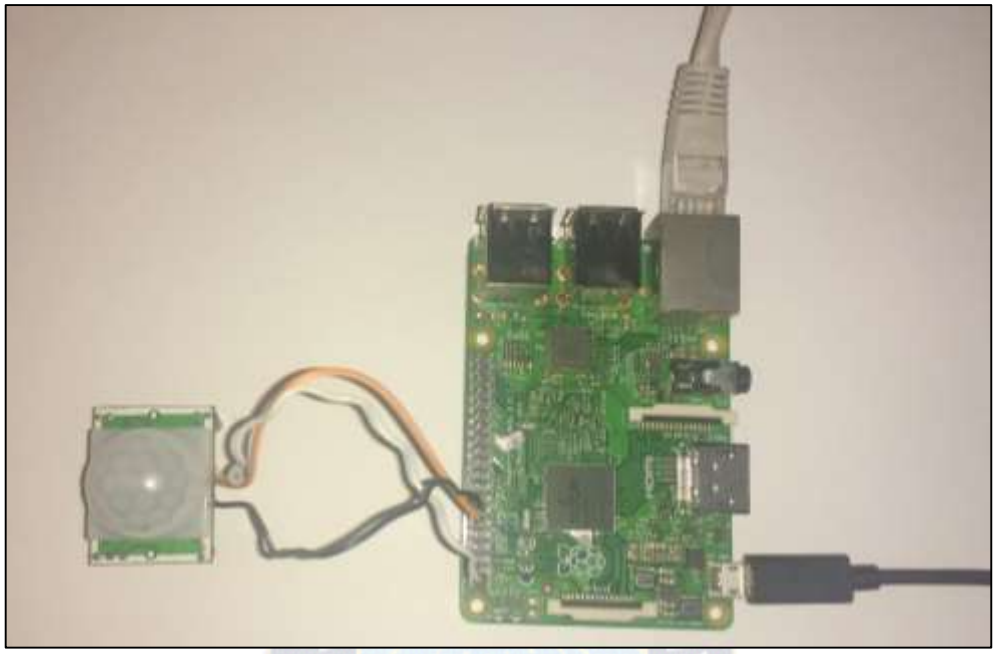


Figura 3.30: Conexión Física del sensor de presencia PIR y la Raspberry Pi
Fuente: (Elaboración Propia)

Los resultados que se obtuvieron de las pruebas en esta conexión, se describen en la tabla 3.29, previamente se tenía conocimiento que el sensor necesitaría de algún tipo de calibración, por lo que no se realiza esta prueba esperando incertidumbre.

Tabla 3.29: Prueba de conexión entre el sensor PIR y la Raspberry Pi

Documento de Prueba dos	
Dispositivos	Sensor de presencia PIR, Raspberry Pi
Objetivo	Realizar la identificación de presencia en cercanías de los ingresos que están sometidas al control
Observación	La conexión respondió correctamente, pero se necesita una calibración para que la detección sea optima
Errores	El sensor necesita de calibración, en sus dos resistencias variables, la primera establece el tiempo que se mantendrá activa la salida del sensor, la segunda permite establecer la distancia de detección que puede variar de 3-7 m.

Fuente: (Elaboración Propia)

En la Figura 3.31 se observa la conexión física entre el RELÉ la Raspberry Pi, siguiendo el diagrama de la figura 3.14, para realizar las pruebas respectivas.

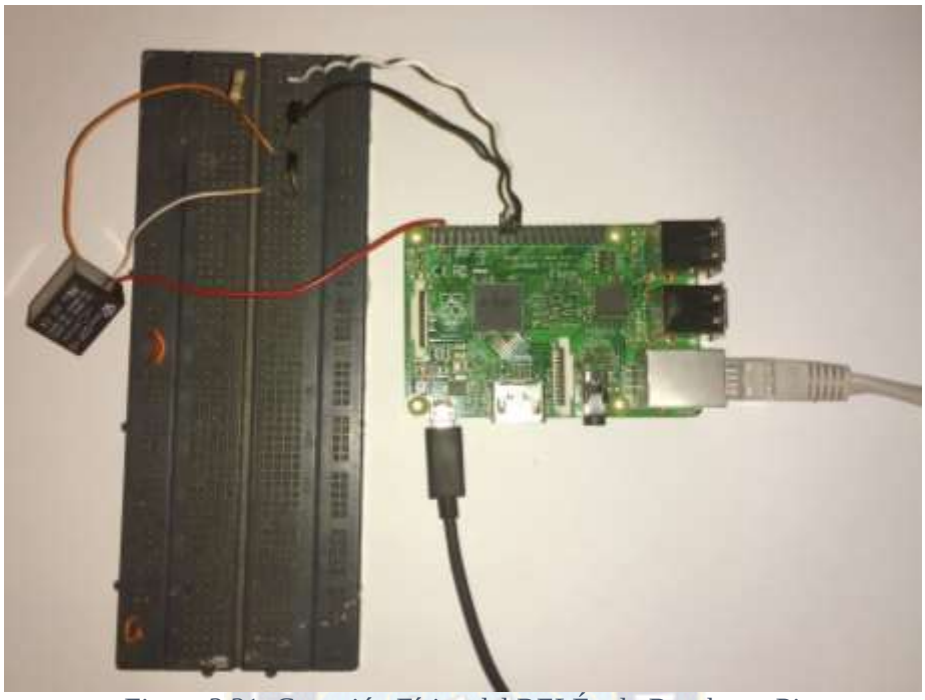


Figura 3.31: Conexión Física del RELÉ y la Raspberry Pi
Fuente: (Elaboración Propia)

La prueba contempla el funcionamiento individual del RELÉ sin accionar todavía ningún circuito, los resultados que se obtuvieron, se describen en la tabla 3.30,

Tabla 3.30: Prueba de conexión entre el sensor PIR y la Raspberry Pi

Documento de Prueba tres	
Dispositivos	RELÉ, Raspberry Pi
Objetivo	Probar el funcionamiento electromagnético del RELÉ, a partir de señales pequeñas emitidas por la Raspberry Pi
Observación	La conexión entre los componentes que el RELÉ necesita se realizó sin mayor problema, toda la configuración respondió de forma favorable.
Errores	Se debe escoger entre dos estados posibles del relé, los cuales son: normalmente abierto, normalmente cerrado. Por conveniencia se usara normalmente cerrado

Fuente: (Elaboración Propia)

La prueba al software en desarrollado se las realiza de forma unitaria, esperando corregir errores en el proceso, estas correcciones individuales garantizaran que al integrar el sistema final, no presente mayores errores.

En la figura 3.32 se muestra el resultado del código encargado verificar las restricciones que tiene cada persona al ser identificada por medo de la tarjeta RFID, para posteriormente conceder o negar el permiso de ingreso.

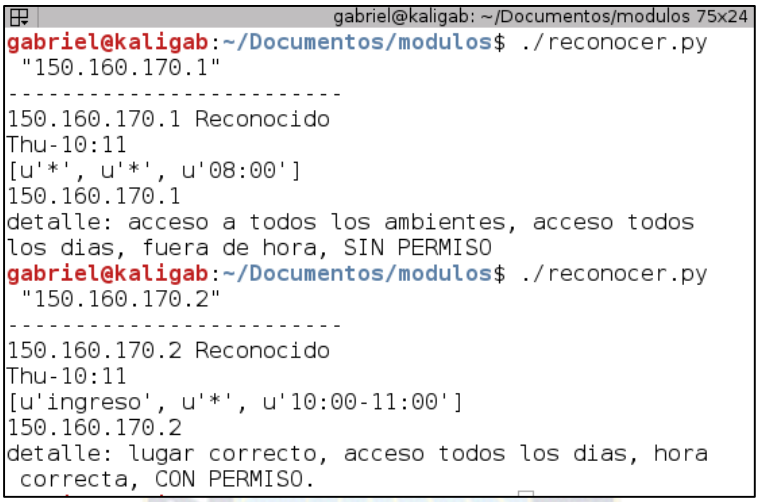


Figura 3.32: Resultado de verificación de restricciones
Fuente: (Elaboración Propia)

En la tabla 3.31 se muestra los resultados obtenidos de la verificación del código que está encargado de analizar las restricciones, en cada persona identificada.

Tabla 3.31: Prueba de verificación de permisos

Documento de Prueba cuatro	
Módulos	verificarRFID.py, reconocer.py
Objetivo	Verificar las restricciones que cada persona tiene.
Observación	El código realiza la verificación con los parámetros necesarios, y en base a ellos concede o niega el permiso de ingreso.
Errores	Se detectó el error de hora y fecha, en el servidor, se debe sincronizar correctamente para subsanarlo.

Fuente: (Elaboración Propia)

En la figura 3.33 se muestra los resultados obtenidos de la verificación del código que está encargado de registrar la autorización, después de realizar la verificación de las restricciones, el criterio del parámetro es: 0 para no autorizado, 1 para autorizado.

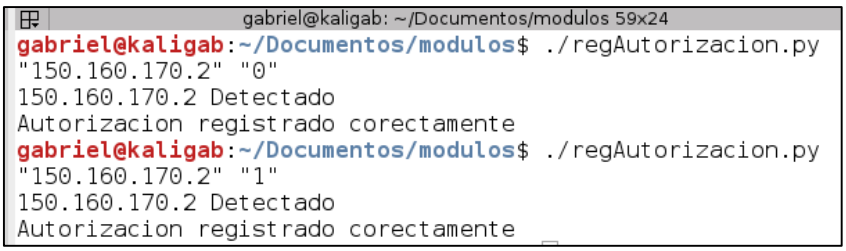


Figura 3.33: Resultado del registro de autorización
Fuente: (Elaboración Propia)

En la tabla 3.32 se muestra los resultados obtenidos de la verificación del código que realiza el registro de autorizaciones.

Tabla 3.32: Prueba de registro de autorizaciones

Documento de Prueba cinco	
Módulos	regAutorizacion.py
Objetivo	Registrar de forma correcta las autorizaciones que se efectué después de realizar la verificación de restricciones.
Observación	El código realiza el registro de las autorizaciones de forma adecuada.
Errores	Sin errores.

Fuente: (Elaboración Propia)

En la figura 3.34 se muestra los resultados obtenidos de la verificación del código que está encargado de registrar la identificación de cada tarjeta RFID.

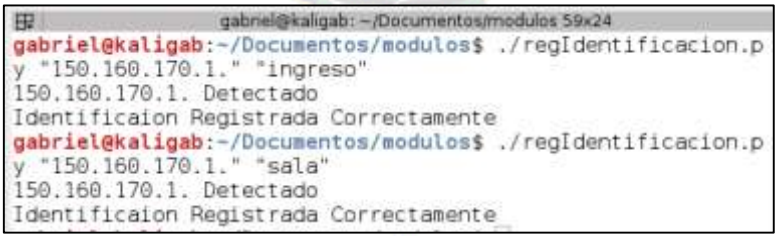


Figura 3.34: Resultado del registro de identificación
Fuente: (Elaboración Propia)

En la tabla 3.33 se muestra los resultados obtenidos de la verificación del código que realiza el registro de identificación de tarjetas RFID.

Tabla 3.33: Prueba de registro de identificación

Documento de Prueba seis	
Módulos	regIdentificacion.py
Objetivo	Registrar de forma correcta las identificaciones de cada tarjeta RFID que usa el sistema
Observación	El código realiza el registro de las Identificaciones de forma adecuada.
Errores	Sin errores.

Fuente: (Elaboración Propia)

En la figura 3.35 se muestra los resultados obtenidos de la verificación del código que está encargado de validar el acceso correcto a la parte administrativa del sistema.

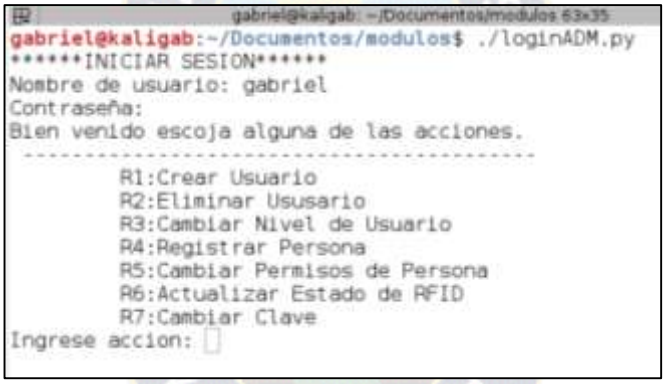


Figura 3.35: Resultado código para acceso administrativo
Fuente: (Elaboración Propia)

En la tabla 3.34 se muestra los resultados obtenidos de la verificación del código que permite el acceso a la parte administrativa del sistema.

Tabla 3.34: Prueba de acceso a la administración del sistema

Documento de Prueba seis	
Módulos	loginADM.py
Objetivo	Conceder acceso a la parte administrativa del sistema, con verificaciones pertinentes de usuario y contraseña
Observación	El código realiza la verificación adecuada protegiendo la contraseña
Errores	Sin errores.

Fuente: (Elaboración Propia)

3.7. Fase 6. Test Funcional

En esta fase se asegura que la integración de cada unidad verificada en la fase anterior, preste el funcionamiento adecuado, tras cada integración se espera errores que se pueden subsanar, gracias a la flexibilidad que nos brinda el Modelo en V se verificara además si se cumplen los objetivos que en un principio se planteó para la investigación.

El sistema ya integrado es capaz de responder a los objetivos, pudiendo identificar cada ingreso, con la verificación de las restricciones, y notificando aquellos ingresos que no cuentan con autorización. En la figura 3.36 se muestra la funcionalidad del sistema tras su integración como conjunto final.

Figura 3.36: Resultado código de funcionalidad final del sistema
Fuente: (Elaboración Propia)

En la tabla se muestra el cumplimiento de cada objetivo, con la integración completa del sistema tanto en software como en hardware.

Tabla 3.35: Cumplimiento de objetivos con el sistema

Objetivo	Software	Hardware	Respuesta del sistema
Identificar un medio de autenticación adecuada para los accesos, relacionando diferentes dispositivos	-	Raspberry Pi Módulo RFID	El sistema responde adecuada mente al módulo RFID, con las librerías usadas.
Monitorear constantemente los ingresos autorizados a un determinado ambiente, a través de comunicación por internet	Reconocer.py	Raspberry Pi Módulo RFID Sensor PIR	El sistema es capaz conectarse a internet para informar de los ingresos que el software identifica.
Identificar el medio de autenticación más confiable para lograr una autenticación segura y notificar al dispositivo móvil asociado.	Reconocer.py verificarRFID.py	Raspberry Pi	El sistema transmite mensajes por el Protocolo MQTT en forma de notificaciones al dispositivo móvil
Informar de manera oportuna los ingresos no autorizados, para identificar situaciones que estén en contra de la seguridad	verificarRFID.py	Raspberry Pi	El sistema notifica oportunamente la detección de un ingreso no autorizado y permite la recepción de comandos para ejecutar bloqueos de identificación
Facilitar la tarea en el control de ingresos cuando se van a trabajar con una afluencia de personas grande	Sistema integrado	Raspberry Pi Módulo RFID Sensor PIR	El sistema es capaz de identificar muchos usuarios registrados previamente.

Fuente: (Elaboración Propia)

3.8. Fase 7. Test de Especificaciones

En esta fase se debe instalar el sistema completo en una ubicaron final, para hacer pruebas se instalara todo lo necesario en el ingreso a un departamento, propiamente dicho a una puerta, en términos de IoT este será el objeto que se conectara a internet.

El la figura 3.37 se muestra la parte posterior donde está ubicado la Raspberry Pi integrando el sistema de identificación en su ambiente final para conectar la puerta de ingreso a internet.

Figura 3.37: Instalación del sistema en su ubicación final
Fuente: (Elaboración Propia)

En la figura 3.38 se muestra la parte anterior de la ubicación del sistema de identificación en su ambiente final, en esta se posiciona el módulo RFID.

Figura 3.38: Instalación del módulo RFID en su ubicación final
Fuente: (Elaboración Propia)

En la figura 3.39 se muestra el resultado final de la instalación del Sistema de Control de Ingresos a un Ambiente Basado en Android, Raspberry Pi e Internet de las Cosas, para ello se debe contar con acceso a internet cercana a la puerta de prueba, con suministro de energía eléctrica para alimentar la Raspberry Pi y todos los dispositivos conectados al mismo.

Figura 3.39: Sistema de Control de Ingresos no autorizados a un Ambiente
Fuente: (Elaboración Propia)

CAPÍTULO 4

PRUEBA DE HIPÓTESIS

4.1. Introducción

En el presente capitulo tiene como objetivo demostrar que la hipótesis planteada en el primer capítulo cumple con lo mencionado, para esto se emplea el uso de un estadístico de prueba.

La prueba se realizara en base al análisis de los datos obtenidos en las pruebas de la fase de Test Funcional del Modelo en V. Para demostrar la validez de la hipótesis, se realiza la prueba estadística usando Chi-Cuadrado para determinar su aceptación o su rechazo.

4.2. Prueba de hipótesis

Para la prueba de hipótesis se recopilan datos del ingreso a un ambiente, con siete personas que están sometidas al control con restricciones de ingreso.

Para la demostración:

Se determina la hipótesis nula:

> H_o = El control de ingresos, utilizando Android, Raspberry Pi e Internet de las Cosas, **no** permite controlar y notificar el ingreso autorizado a un ambiente.

Se determina la hipótesis alternativa:

> H_a = El control de ingresos, utilizando Android, Raspberry Pi e Internet de las Cosas, permite controlar y notificar el ingreso autorizado a un ambiente.

Se necesita definir el nivel de significancia, que se interpreta como el error que se comete al rechazar la hipótesis nula H_o.

Cuando se rechaza la hipótesis nula $H_{o,}$ no es seguro 100% de que sea cierto, se asume la probabilidad de error p.

Si p < 0,05 se RECHAZA la hipótesis

Si p > 0,05 se ACEPTA la hipótesis

La relación entre el valor de la Chi-Cuadrado y p de error es iguala la relación inversa, es decir, a mayor valor de Chi-Cuadrado, menor valor de p.

Para fines de la prueba se determina el nivel de significancia α=0,05.

El estadístico de prueba que se empleara es la Chi-Cuadro, mismo que nos permitirá determinar la validez de la hipótesis utilizando la formula siguiente:

$$x^2_{calc} = \sum \frac{(f_0 - f_e)^2}{f_e}$$

Siendo:

f_0= Frecuencia del valor observado.

f_e= Frecuencia del valor esperado.

La información que el sistema interpreto del control de ingresos en base a las restricciones de lugar, día y hora, y el dato proporcionado por la tarjeta RFID, se muestra en la figura 4.1

```
cqlsh:control_ingreso> select * from autorizacion ;

 tagid_au      | marcat_au                       | autorizacion
---------------+---------------------------------+--------------
 150.160.170.7 | 2016-10-08 21:08:55.000000+0000 | no autorizado
 150.160.170.5 | 2016-10-08 21:08:38.000000+0000 |    autorizado
 150.160.170.3 | 2016-10-08 21:07:45.000000+0000 | no autorizado
 150.160.170.1 | 2016-10-08 21:05:47.000000+0000 | no autorizado
  202,53,94,49 | 2016-10-16 02:30:09.000000+0000 | no autorizado
  202,53,94,49 | 2016-10-16 02:30:11.000000+0000 | no autorizado
  202,53,94,49 | 2016-10-16 03:30:17.000000+0000 | no autorizado
  202,53,94,49 | 2016-10-16 03:30:23.000000+0000 | no autorizado
  202,53,94,49 | 2016-10-16 03:30:56.000000+0000 | no autorizado
  202,53,94,49 | 2016-10-16 03:32:32.000000+0000 |    autorizado
  202,53,94,49 | 2016-10-16 04:39:51.000000+0000 |    autorizado
```

Figura 4.1: Información de autorización
Fuente: (Elaboración Propia)

La información también proporciona las observaciones de los ingresos efectuados, mostrado en la figura 4.2

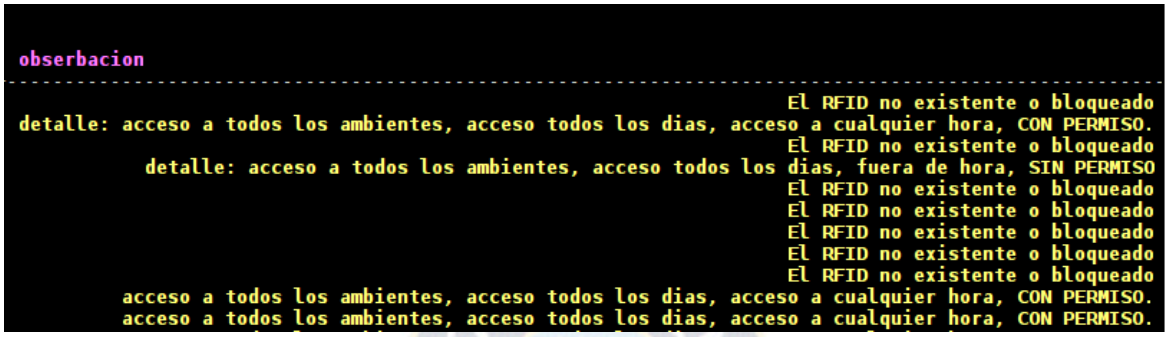


Figura 4.2: Observaciones de autorizaciones
Fuente: (Elaboración Propia)

De esta información se excluye aquellas generadas por las restricciones marcadas con * pues están representan la ausencia de restricción por lo que la autorización al ingreso es plena, el resto proporciona los datos necesarios para la identificación de autorizaciones por lugar, día y hora.

Estos datos son esenciales para utilizar el estadístico de prueba, los valores observados se muestran en la tabla 4.1.

Tabla 4.1: Valores Observados

	Autorizado	No Autorizado	**Total**
Lugar	30	6	**36**
Día	32	14	**46**
Hora	3	9	**12**
Total	**65**	**29**	**94**

Fuente: (Elaboración Propia)

A partir de la tabla de valores observados, se procede a calcular los valores esperados, mismos que serán utilizados conjuntamente en el estadístico de prueba, el resultado se muestra en la tabla 4.2.

$\frac{65*36}{94} = 24.89$ $\frac{65*46}{94} = 31.81$

$\frac{65*12}{94} = 8.29$ $\frac{29*36}{94} = 11.11$

$\frac{29*46}{94} = 14.19$ $\frac{29*12}{94} = 3.70$

Tabla 4.2: Valores Esperados

	Autorizado	No Autorizado	**Total**
Lugar	24.89	11.11	**36**
Día	31.81	14.19	**46**
Hora	8.29	3.70	**12**
Total	**65**	**29**	**94**

Fuente: (Elaboración Propia)

Se observa que los totales de los valores esperados son iguales a los totales de los valores observados, por lo que se puede proceder a calcular el valor de la Chi-Cuadrado, empleando el estadístico de prueba:

$$x^2_{calc} = \frac{(30-24-89)^2}{24.89} + \frac{(6-11.11)^2}{11.11} + \frac{(32-31.81)^2}{31.81} + \frac{(14-14.19)^2}{14.19} + \frac{(3-8.19)^2}{8.19} + \frac{(9-3.70)^2}{3.70}$$

$$x^2_{calc} = \mathbf{14.37}$$

Para comprobar el resultado obtenido del estadístico de prueba, se debe obtener el Chi-Cuadrado crítico, para ello es necesario determinar los grados de libertad haciendo uso de la siguiente formula:

$$Gl = (N^0\ de\ filas - 1) * (N^0\ d\ columnas - 1)$$

$$Gl = (2-1) * (3-1)$$

$$Gl = 2$$

El valor crítico de la tabla Chi-Cuadrado, con grado de libertad=2 y nivel de significancia=0.05, es de **5.99**

Se debe cumplir la desigualdad: Chi-Cuadrado calculado es menor que Chi-Cuadrado crítico, para que la hipótesis planteada sea aceptada, si el estadístico toma un valor mayor a calor crítico, entonces se rechazara la hipótesis nula.

$$x_{calc}^2 \leq valor\ critico$$

$$14.47 \leq 5.99$$

4.3. Toma de decisión

El resultado del estadístico haciendo uso de los datos arroja un resultado de **14.47**, este es mayor que el valor crítico obtenido de la tabla Chi-Cuadrado que es de **5.99**.

La relación entre el valor de Chi-Cuadrado calculado y p de error, es inversa, como el valor calculado es de **14,47** es mayor que **5,99**, entonces el valor de p es menor a **5%**, basado en el nivel de significancia determinado.

Por tanto se rechaza la hipótesis nula: "El control de ingresos, utilizando Android, Raspberry Pi e Internet de las Cosas, **no** permite controlar y notificar el ingreso autorizado a un ambiente"

Se acepta la hipótesis alternativa: "El control de ingresos, utilizando Android, Raspberry Pi e Internet de las Cosas, permite controlar y notificar el ingreso autorizado a un ambiente".

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

El trabajo de investigación incursiona en la implementación de un sistema que integre diferentes tipos de tecnologías, en todo el desarrollo del sistema se logra integrar componentes de hardware como el identificador RFID, sensor de presencia, con la Raspberry Pi, conformando así la parte física del sistema.

En cuanto a software se logra integrar el lenguaje python con todas las librerías necesarias, con el lenguaje de consultas de Cassandra CQL, como parte del servidor, y el Lenguaje Java como parte del cliente, todo conectado a internet por medio del protocolo de comunicación MQTT.

Para el correcto desarrollo del sistema se empleó la refactorización del Modelo en V, mismo que está basada en el Método para el Diseño de Sistemas de Control Automático, y la Metodología Mobile-D. El nuevo Modelo en V, pretende ser mucho más eficientes, cuando se adiciona al sistema el uso una aplicación móvil.

5.2. Estado de los Objetivos

5.2.1. Estado del Objetivo General

Se cumple el objetivo general plateado "Desarrollar un sistema de control de ingresos, que determine aquellos autorizados a un ambiente" en el capítulo tercero se llevó acabo todo el desarrollo del sistema con sus diferentes pruebas, siguiendo el Modelo en V, cumpliendo así con la funcionalidad del mismo.

5.2.2. Estado de los objetivos específicos

Para cada objetivo planteado se describe las conclusiones tras el desarrollo del sistema:

- Objetivo 1: “Identificar un dispositivo de identificación adecuada que permita determinar la autorización en los ingresos”, entre los dispositivos de hardware utilizados se emplea el módulo RFID, que es el más adecuado y económico para llevar a cabo la identificación de los ingresos.
- Objetivo 2: “Monitorear constantemente los ingresos autorizados a un determinado ambiente, a través de comunicación por internet”, la aplicación móvil recibe notificaciones, por medio de las cuales se puede monitorear los ingresos que se efectúan los ambientes controlados, esto se puede observas en la figura 3.28.
- Objetivo 3: “Identificar el modo de autenticación más confiable para lograr una autorización correcta, y notificar al dispositivo móvil”, el sistema identifica un ingreso autorizado, analizando las restricciones por Lugar, Día, y hora, almacenados en la base de datos Cassandra representados en la figura 3.7, el código la figura 3.19 verifica las restricciones y se emite una respuesta positiva o negativa de autorización.
- Objetivo 4: “Informar de manera oportuna los ingresos no autorizados”, tanto el sistema de control como la aplicación móvil, están conectados a internet por medio el protocolo MQTT, por lo que se las notificaciones generadas son oportunas.
- Objetivo 5: “Facilitar la tarea en el control de ingresos cuando se va a trabajar con una afluencia de personas”, la base de datos Cassandra brinda tolerancia a gran cantidad de datos por lo que el registro de los datos generados por el control, serán tolerados, por su parte el sistema de identificación está diseñado para reconocer tarjetas RFID de manera recurrente.

Se consideró adicionar una parte administrativa al sistema, el ingreso al mismo es controlado por un inicio se sesión y se clasifica a usuarios por niveles de acceso con roles diferentes, la finalidad es que se pueda registrar nuevas personas, cambiar permisos de ingreso, disponer del estado de las tarjetas tomando valores de bloqueada o activo.

5.2.3. Estado de la hipótesis

La hipótesis del presente trabajo sostiene que “El control de ingresos, utilizando Android, Raspberry Pi e Internet de las Cosas, permite controlar y notificar el ingreso autorizado a un ambiente”, en el capítulo cuarto de puede verificar la demostración de la hipótesis, mismo que respalda el control que el sistema realiza.

5.3. Recomendaciones

El presente trabajo de investigación se concluyó con la única funcionalidad de controlar los ingresos no autorizado a un ambiente, para esto se conectó una puerta a internet por medio de la Raspberry Pi, los datos generados por este control son bastante valiosos por lo que se recomienda implementar un módulo de analítica de grandes cantidades de datos, para que los mismos tengan la interpretación adecuada.

Se recomienda ampliar las bondades de IoT para este tipo de controles, ya que no solo se podría emplear RFID, sensor de presencia, sino muchos otros tipos de sensores que mejores los controles y el modo de acceso.

5.4. Recomendaciones para futuros trabajos

Para futuros trabajos se recomienda el empleo de diferentes mecanismos de identificación como la biométrica, la facial o la de retina, estos prometen ser mucho más eficientes en cuanto a la identificación única de cada individuo.

Se recomienda hacer uso de sistemas expertos para contar un sistema de control mucho más automático e independiente.

Es un hecho que la conectividad de objetos a internet es tendencia en IoT, y que los datos trasportados por esta conectividad brindan información valiosa, por lo que se recomienda el uso de nuevos protocolos de comunicación, que garanticen la seguridad de los datos.

BIBLIOGRAFÍA

Abarca P (2016). *ABC Del Control Automático*. Encuentro Eléctrico 2016, Chile

Alvarez E. (2013). *Arquitectura de Android.* Recuperado de: http://androideric. blogspot.com/2013/02/13-arquitectura-de-android.html [Acceso: septiembre 2016]

ANDROID. (2016). *Android*. Recuperado de: https://www.android.com [Acceso: septiembre 2016]

Arantxa A. (2016). *Ejemplos de Dispositivos Conectados a Internet de las Cosas.* Recuperado de: http://www.muycomputerpro.com/2014/05/16/ejemplos-internet-de-las-cosas [Acceso: mayo 2016]

Arlandy M. (2014). *Configurando Notificaciones Push para desarrollos Android con Google Cloud Messaging.* Recuperado de: https://www.adictosaltrabajo.com/tutoriales/configurando-notificaciones-push-android/ [Acceso: septiembre 2016]

Asay A. (2014). *The Internet Of Things Will Need Millions of Developers*. Recuperado de: http://readwrite.com/2014/06/27/internet-of-things-developers-jobs-opportunity, [Acceso: mayo 2016]

Asían A. (2016). *Ejemplos de dispositivos conectados a Internet de las Cosas.* Recuperado de: http://www.muycomputerpro.com/2014/05/16/ejemplos-internet-de-las-cosas, [Acceso: abril 2016]

Baena M. (2016). *Comparamos todos los modelos de Raspberry*, Recuperado de: http://www.gadgetos.com/noticias/comparamos-todos-modelos-raspberry/, [Acceso: junio 2016]

Ballestín A. (2015). *Introducción de aspectos de seguridad en la Internet de las cosas en el ámbito de las viviendas inteligentes*, Recuperado de: Universidad Politécnica de Valencia. https://riunet.upv.es/handle/10251/59695, [Acceso: abril 2016]

Blanco P., Camarero J., Fumero A., Werterski A., Rodriguez P. (2009). *Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone*. Universidad Politecnica de Madrid. Madrid.

Bliznakoff D. (2014). *IoT: Tecnologías: usos, tendencias y desarrollo futuro*, Recuperado de: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/40044/6/dbliznakoffTFM0115memoria.pdf, [Acceso: mayo 2016]

Cadenas R. (2011). *Perfil de Usuarios en Cibercafés del Municipio Valera del Estado Trujillo,* Universidad del Zulia.

Carrasco F. (2016). *La seguridad un reto clave para el despegue de IoT*, Recuperado de: http://www.corporateit.cl/index.php/2016/07/08/la-seguridad-es-un-reto-clave-para-el-despegue-de-iot/ [Acceso: agosto 2016]

Castro L. (2015). *¿Qué es RFID?*, Recuperado de: http://aprenderinternet.about.com/od/Glosario/g/Que-Es-RFID.htm, [Acceso: mayo 2016]

Chávez J. (2015). *Sistema Domótico para un Hogar Basado en Software y Harware Libre.* Universidad Mayor de San Andrés.

Dablarui (2013). *Raspberry Pi*. Recuperado de: http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/, [Acceso: abril 2016]

DATASTAX (2016). *Acerca de Apache Cassandra*. Recuperado de: http://docs.datastax.com/en/cassandra/3.x/cassandra/cassandraAbout.html [Acceso: septiembre 2016]

Del Rio O. (2011). *El Proceso de Investigación: Etapas y Planificación de la investigación en comunicación*. Recuperado de: https://www.academia.edu/2443422/, [Acceso: mayo 2016]

Díaz W. (2013). *Bases de Datos NoSQL.* Recuperado de: http://basesdedatosnosql.blogspot.com/ [Acceso: 2016]

ECIXGROUP (2015). *Una Aproximación a algunos elementos de Internet de las Cosas*. Recuperado de: http://ecixgroup.com/el-grupo/una-aproximacion-algunos-elementos-de-internet-de-las-cosas/, [Acceso: mayo 2016]

EUGOMAN. (2016). *Modelo de Datos de Cassandra*. Recuperado de: https://sites.google.com/site/uegoman/modelo-de-datos-de-cassandra [Acceso: septiembre 2016]

Fernández L. (2011). *El Documental Arduino narra la historia del open source en el mundo físico*. Recuperado de: http://www.spri.eus/euskadinnova/es/innovacion-social/noticias/documental-arduino-narra-historia-open-source-mundo-fisico/7285.aspx, [Acceso: mayo 2016]

Gaucha O. y Gamarra O. (2014). *El Brazo Robótico de Josney*, Recuperado de: http://med.se-todo.com/istoriya/27318/index.html, [Acceso: abril 2016]

Gonzales G. (2014). *¿Qué significa que Android esté basado en Linux?* Recuperado de: http://blogthinkbig.com/android-esta-basado-en-linux/ [Acceso: septiembre 2016]

Gonzáles L. (2016). *Hardware del Internet de las Cosas.* Recuperado de: http://equipo.altran.es/hardware-iot-internet-de-las-cosas/ [Acceso: septiembre 2016]

Hernández A. (2015). *Kevin Ashton, el padre del Internet de las Cosas*. Recuperado de: http://www.finanzas.com/xl-semanal/magazine/20151004/kevin-ashton-padre-internet-8912.html, [Acceso: abril 2016]

Hernández R. (2010). *Introducción a los Sistemas de Control*. Instituto Tecnológico de Aguascalientes. México.

Hípola P. (2012). *¿Hasta dónde tecnología Pull y Push?* Recuperado de: https://aprendizajeubicuo.wordpress.com/2012/04/23/hasta-donde-tecnologia-pull-o-push/ [Acceso: septiembre 2016]

IBM. (2011). *Comprender las especificaciones de los servicios web, Parte 1: SOAP* Recuperado de: https://www.ibm.com/developerworks/ssa/webservices/tutorials/ ws-understand-web-services1/#ibm-pcon [Acceso: septiembre 2016]

NOVA (2016). *Importadora Nova*, Recuperado de: http://www.importadoranova.com/11-electronica, [Acceso: mayo 2016]

Jadoul M. (2015). *Internet de las Cosas en la Salud un mercado en crecimiento.* Recuperado de: http://edriel.com/internet-de-las-cosas-en-la-salud/, [Acceso: mayo 2016]

JAVA (2016). *Java Messaging Service (JMS)*. Recuperado de: https://docs.oracle.com/ cd/E19509-01/820-5892/ref_jms/index.html [Acceso: septiembre 2016]

Kuo B. (2016). *Sistemas De Control Automatico*. Departament of Electrical and Computer Engineering University of Illinois at Urbana-Champaing. Mexico.

Machaca B. (2015). *Control Inteligente de Seguridad en Viviendas Mediante Agentes Móviles*. Universidad Mayor de San Andrés. Bolivia.

Martinez C. (2016). *Arquitectura Pull y Push en Aplicaciones Móviles*. Recuperado de: http://sg.com.mx/revista/48/arquitectura-pull-y-push-aplicaciones-moviles# [Acceso: septiembre 2016]

Martinez L. (2012). *Modelo en V*. Recuperado de: http://softwareverde.blogspot.com /2012/09/modelo-en-v.html [Acceso: septiembre 2016]

Mateos M. (2013). *Tecnología Push*. Recuperado de: http://www.genbeta.com/a-fondo /tecnologia-push-asi-funciona [Acceso: septiembre 2016]

MICROSFT. (2016). *Publish/Subscribe*. Recuperado de: https://msdn.microsoft.com/en-us/library/ff649664.aspx [Acceso: septiembre 2016]

Mírez J. (2013). *Diseño de Sistemas de Control*. Recuperado de: https://jmirezcontrol. wordpress.com/2013/08/28/c048-diseno-de-sistemas-de-control/ [Acceso: septiembre 2016]

MOBILE. (2015) *¿Cómo Funciona Android?* Recuperado de: http://comofuncionaque.com /que-es-android/ [Acceso: septiembre 2016]

MOBILE-D (2016). *Ciclo de Desarrollo Mobile-D*. Recuperado de: http://agile.vtt.fi /mobiled.html [Acceso: 2016]

MQTT. (2016). *Mqtt*. Recuperado de: http://mqtt.org/ [Acceso: septiembre 2016]

NOSQL. (2016). *No Only SQL*. Recuperado de: http://nosql-database.org/ [Acceso: 2016]

Núñez A. (2016). *En 2018 habrán más objetos conectados a la red que Smartphones,* Recuperado de: http://www.sinembargo.mx/01-06-2016/3049436, [Acceso: mayo 2016]

ONSC (2013). *Lugar de los hechos delictivos en nueve ciudades capitales y El Alto – 2013.* Observatorio Nacional De Seguridad Ciudadana. Recuperado de: http://www.onsc.gob.bo/estadisticas_es.html, [Acceso: mayo 2016]

Oporto H. (2012). *Inseguridad Ciudadana y Criminalidad en Bolivia*. Recuperado de: http://henryoporto.blogspot.com/2013/04/inseguridad-ciudadana-y-criminalidad-en.html, [Acceso: mayo 2016]

Ortiz E. (2014). *Sistema de Control de Acceso para Unidades Militares.* Universidad Militar Nueva Granada. Bogota.

Pavón C. Cruz O. (2010). *Historia, Manejo y aplicación de los Controladores digitales de señales dcPIC*. Facultad de Electrónica y Computación Universidad Nacional de Nicaragua. Nicaragua.

Perez J. y Gardey A. (2013). *Definición de Notificación.* Recuperado de: http://definicion.de /notificacion/ [Acceso: septiembre 2016]

Pramio C. (2011). *El concepto NoSQL, o cómo almacenar tus datos en una base de datos no relacional.* Recuperado de: http://www.genbetadev.com/bases-de-datos/el-concepto-nosql-o-como-almacenar-tus-datos-en-una-base-de-datos-no-relacional [Acceso: 2016]

Pricer K. (2011). *Historia de los Sistemas Biométricos para el Control de Accesos*, Recuperado de: http://www.articuloz.com/seguridad-articulos/historia-de-los-sistemas-biometricos-para-el-control-de-acceso-4423152.html, [Acceso: abril 2016]

QODE (2015). *¿Qué son las notificaciones push?* Recuperado de: http://qode.pro/blog/que-son-las-notificaciones-push/ [Acceso: septiembre 2016]

RASPBERRYPI (2012). *Raspberry Pi*. Recuperado de: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/, [Acceso: junio 2016]

Rodriguez A. (2008). *RESTful Web Services: The basics*. IBM Corporation.

Rojas M. (2010). *Ciclo de Vida - Modelo en V*, Recuperado de: http://spanishpmo.com/index.php/ciclos-de-vida-modelo-en-v/, [Acceso: mayo 2016]

Romorico A. (2014). *El Internet de las Cosas: Una mirada a la actividad global de la UIT*. Recuperado de: http://www.jentel.mx/component/k2/item/1936-el-internet-de-las-cosas-una-mirada-a-la-actividad-global-de-la-uit.html, [Acceso: mayo 2016]

Sánchez J. (2013). *RESTful Web Services*, Recuperado: http://jc.pe/2013/04/18/restful-web-services/, [Acceso: junio 2016]

Teruel A. (2015). *Internet de las Cosas en la Salud un mercado en crecimiento*. Recuperado de: http://edriel.com/internet-de-las-cosas-en-la-salud/, [Acceso: mayo 2016]

UIT (2015). *The Internet of Things report in the press*, Unión Internacional de Telecomunicaciones. Recuperado de: http://www.itu.int/osg/spu/publications/internetofthings/index.html, [Acceso: abril 2016]

UPC (2016). *Sistemas de Control*. Recuperado de: https://upcommons.upc.edu/bitstream/handle/2099.1/3330/34059-5.pdf?sequence=5 [Acceso: septiembre 2016]

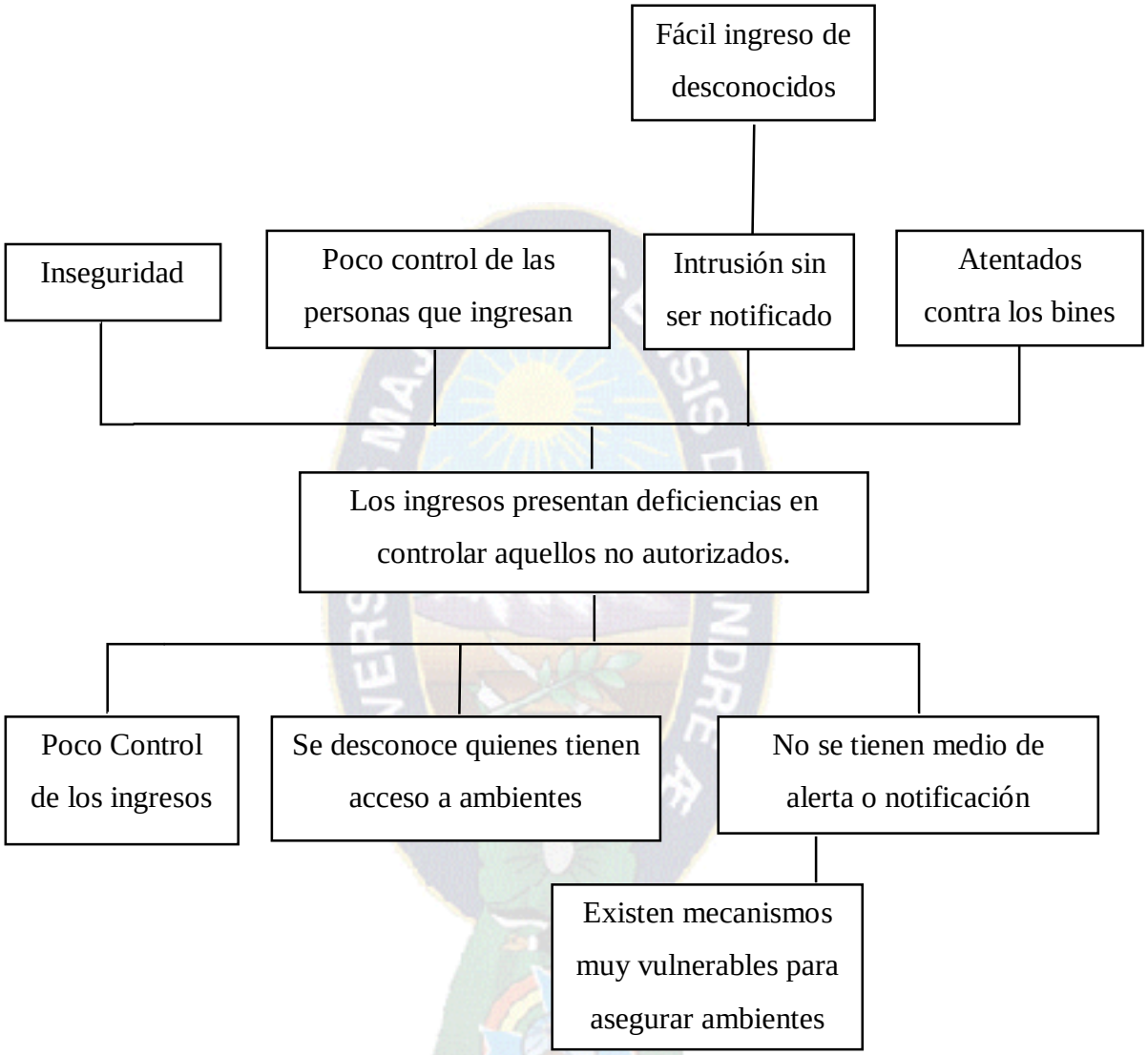
Upton E. (2016). *Raspberry Pi 3 on sale now at $35*, Recuperado de: https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/, [Acceso: junio 2016]

Vildósola E. (2016). *Actuadores.* Soltex Chile S.A. Chile.

Yébenes J. (2013). *¿Qué es MQTT?*, Recuperado de: https://geekytheory.com/que-es-mqtt/, [Acceso: junio 2016]

ANEXOS

ANEXO A – ÁRBOL DE PROBLEMAS



ANEXO B – ÁRBOL DE OBJETIVOS

ANEXO C – ESPECIFICACIONES TECNICAS DE LOS DISPOSITIVOS

Módulo RFID

Modelo	RFID-RC522
Corriente de operación	13-26mA a 3.3V
Corriente de stand by	10-13mA a 3.3V
Corriente de sleep-mode	<80uA
Corriente máxima	30mA
Frecuencia de operación	13.56Mhz
Distancia de lectura	0 a 60mm
Protocolo de comunicación	SPI
Velocidad de datos máxima	10Mbit/s
Dimensiones	40 x 60 mm
Temperatura de operación	-20 a 80ºC
Humedad de operación	5%-95%
Máxima velocidad de SPI	10Mbit/s

Sensor de Presencia PIR

Modelo	PIR HC-SR501
controlador	PIR BISS0001
Rango de detección	3 m a 7 m, ajustable mediante trimmer (Sx)
Lente	fresnel de 19 zonas, ángulo < 100º
Salida activa alta	3.3 V
Tiempo	salida configurable mediante trimmer (Tx)
Redisparo	configurable mediante jumper de soldadura
Temperatura	-20 a 80ºC
Consumo de corriente en reposo	< 50 μA
Voltaje de alimentación	4.5 VDC a 20 VDC

Raspberry Pi 2

SoC	Broadcom BCM2836
CPU	ARM11 ARMv7 ARM Cortex-A7 4 núcleos @ 900 MHz
Overclocking	Sí, hasta arm_freq=1000 sdram_freq=500 core_freq=500 over_voltage=2 de forma segura
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
RAM	1 GB LPDDR2 SDRAM 450 MHz
Usb 2.0	4
Salidas de Vídeo	HDMI 1.4 @ 1920x1200 píxeles
Almacenamiento	microSD
Ethernet	Sí, 10/100 Mbps
Tamaño	85,60x56,5 mm
Peso	45 g
Consumo	5v, 900mA, aunque depende de la carga de trabajo de los 4 cores
Precio	35 dólares

DOCUMENTACIÓN