**ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**REINGENIERÍA DE MÓDULOS DE VENTAS, RECAUDOS E INVENTARIO DE APP A PWA CON HERRAMIENTA DEVOPS**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**ANDERSON FABIAN REVELO MORILLO**

@epn.edu.ec

**JOSÉ ISRAEL PÉREZ ESPINOSA**

Jose.perez01@epn.edu.ec

**DIRECTOR: Vicente Egüez S. MSc.**

adrian.eguez@epn.edu.ec

**CODIRECTOR:**

@epn.edu.ec

**Quito, abril 2021**

# **AVAL**

Certificamos que el presente trabajo fue desarrollado por **Anderson Fabian Revelo Morillo** y **José Israel Pérez Espinosa** bajo nuestra supervisión.



|  |  |  |
| --- | --- | --- |
|  |  | **Vicente Egüez S. MSc.**  **DIRECTOR DE PROYECTO** |

# **DECLARACIÓN DE AUTORÍA**

Nosotros, **Anderson Fabian Revelo Morillo** y **José Israel Pérez Espinosa**, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

|  |  |  |
| --- | --- | --- |
| **Anderson Fabian Revelo Morillo** |  | **José Israel Pérez Espinosa** |

# **DEDICATORIA**

**Nombre.**

# **AGRADECIMIENTO**

**Nombre.**

**ÍNDICE DE CONTENIDO**

[**AVAL** I](#_Toc70354412)

[**DECLARACIÓN DE AUTORÍA** II](#_Toc70354413)

[**DEDICATORIA** III](#_Toc70354414)

[**AGRADECIMIENTO** IV](#_Toc70354415)

[**RESUMEN** XIV](#_Toc70354416)

[**ABSTRACT** XV](#_Toc70354417)

[**PRESENTACIÓN** XVI](#_Toc70354418)

[**CAPÍTULO I** 1](#_Toc70354419)

[**1. INTRODUCCIÓN** 1](#_Toc70354420)

[**1.1. ANTECEDENTES** 1](#_Toc70354421)

[**1.2. OBJETIVOS** 2](#_Toc70354422)

[**1.2.1. Objetivo general** 2](#_Toc70354423)

[**1.2.2. Objetivos específicos** 2](#_Toc70354424)

[**1.3. ALCANCE** 2](#_Toc70354425)

[**1.4. MARCO TEÓRICO** 2](#_Toc70354426)

[**1.4.1. Reingeniería de software** 2](#_Toc70354427)

[**CAPÍTULO II** 4](#_Toc70354428)

[**2. REVISIÓN LITERARIA Y FUNDAMENTOS TEÓRICOS** 4](#_Toc70354429)

[**2.1. TRABAJOS RELACIONADOS** 4](#_Toc70354430)

[**2.1.1. Estudios previos** 4](#_Toc70354431)

[**2.1.2. Análisis** 5](#_Toc70354432)

[**2.2. MARCO TEÓRICO** 6](#_Toc70354433)

[**2.2.1. SMART CITY** 6](#_Toc70354434)

[**2.2.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE** 8](#_Toc70354435)

[**2.2.4.1. Marco de trabajo Scrum** 9](#_Toc70354436)

[**2.2.4.1.1. Roles de Scrum** 9](#_Toc70354437)

[**2.2.4.1.2. Artefactos** 9](#_Toc70354438)

[**2.2.4.1.3. Eventos [58]** 10](#_Toc70354439)

[**CAPÍTULO III** 11](#_Toc70354440)

[**3. DISEÑO DE LA ARQUITECTURA DEL PROTOTIPO DE RED LPWAN** 11](#_Toc70354441)

[**3.1. ELEMENTOS DE LA RED LORAWAN** 11](#_Toc70354442)

[**3.2. ARQUITECTURA DEL PROTOTIPO DE RED LPWAN** 11](#_Toc70354443)

[**3.3. HERRAMIENTAS DE DESARROLLO** 11](#_Toc70354444)

[**3.3.1. Lenguajes de programación** 12](#_Toc70354445)

[**3.3.2. Frameworks de desarrollo** 13](#_Toc70354446)

[**3.3.3. Librerías** 14](#_Toc70354447)

[**3.3.4. Servicios de software** 15](#_Toc70354448)

[**3.3.5. Base de datos** 16](#_Toc70354449)

[**3.3.6. Entorno de desarrollo** 16](#_Toc70354450)

[**3.3.7. Control de versiones** 17](#_Toc70354451)

[**3.3.8. Recursos y Diagramación** 17](#_Toc70354452)

[**3.4. APLICACIÓN DE LA METODOLOGÍA DE DESARROLLO** 18](#_Toc70354453)

[**3.4.1. Definición de roles** 18](#_Toc70354454)

[**3.4.2. Definición de historias épicas** 18](#_Toc70354455)

[**3.4.3. Pila de producto (Product Backlog)** 18](#_Toc70354456)

[**3.4.4. Planificación de los Sprints (Release planning)** 20](#_Toc70354457)

[**3.4.4.1. Sprint 1** 21](#_Toc70354458)

[**3.4.4.1.1. Objetivo del Sprint** 21](#_Toc70354459)

[**3.4.4.1.2. Historias de usuario** 21](#_Toc70354460)

[**6. CONCLUSIONES, RECOMENDACIONES Y POSIBLE IMPLEMENTACIÓN** 24](#_Toc70354461)

[**6.1. CONCLUSIONES** 24](#_Toc70354462)

[**6.2. RECOMENDACIONES** 24](#_Toc70354463)

[**6.3. POSIBLE IMPLEMENTACIÓN** 24](#_Toc70354464)

[**REFERENCIAS BIBLIOGRÁFICAS** 25](#_Toc70354465)

[**ANEXOS** 26](#_Toc70354466)

[**Anexo 1.** Diagramas de la tarjeta de desarrollo WiFi LoRa 32 (V2). 26](#_Toc70354467)

[**Anexo 2.** Sprint backlog y Sprint Review 26](#_Toc70354468)

[**Anexo 3.** Código fuente de los nodos detectores de intrusión. 26](#_Toc70354469)

[**Anexo 4.** Códec para decodificar la carga útil del dispositivo. 26](#_Toc70354470)

[**Anexo 5.** Mockups de las aplicaciones web y móvil. 26](#_Toc70354471)

[**Anexo 6.** Manual de usuario. 26](#_Toc70354472)

**ÍNDICE DE FIGURAS**

[**Figura 1.** Ubicación del barrio Divino Niño 3](#_Toc70362096)

**ÍNDICE DE TABLAS**

[**Tabla 3.** Comparación de tecnologías LoRaWAN 7](#_Toc70362105)

[**Tabla 4.** Formato de historias de usuario 9](#_Toc70362106)

[**Tabla 9**. Descripción de lenguajes de programación 12](#_Toc70362107)

[**Tabla 10**. Descripción de frameworks de desarrollo 13](#_Toc70362108)

[**Tabla 11.** Descripción de librerías de implementación 14](#_Toc70362109)

[**Tabla 12.** Descripción de servicios de software 15](#_Toc70362110)

[**Tabla 13.** Descripción de la Base de datos MySQL 16](#_Toc70362111)

[**Tabla 14**. Descripción de entornos de desarrollo 16](#_Toc70362112)

[**Tabla 15**. Descripción de herramientas de control de versiones 17](#_Toc70362113)

[**Tabla 16**. Descripción de herramientas adicionales 17](#_Toc70362114)

[**Tabla 17.** Detalle de historias épicas 18](#_Toc70362115)

[**Tabla 18.** Historia de usuario HE01-01 18](#_Toc70362116)

[**Tabla 19.** Historia de usuario HE03-03 19](#_Toc70362117)

[**Tabla 20.** Historia de usuario HE03-02 19](#_Toc70362118)

[**Tabla 21.** Historia de usuario HE01-02 19](#_Toc70362119)

[**Tabla 22.** Historia de usuario HE02-01 19](#_Toc70362120)

[**Tabla 23.** Historia de usuario HE03-01 20](#_Toc70362121)

[**Tabla 24.** Historia de usuario HE02-02 20](#_Toc70362122)

[**Tabla 25.** Historia de usuario HE03-04 20](#_Toc70362123)

[**Tabla 26.** Historia de usuario HE03-05 20](#_Toc70362124)

[**Tabla 27**. Release plannig 21](#_Toc70362125)

[**Tabla 28**. Historia de usuario HE01-01 para el Sprint 1 21](#_Toc70362126)

[**Tabla 29.** Historia de usuario HE01-02 para el Sprint 1 22](#_Toc70362127)

# **RESUMEN**

La constante expansión de las redes inalámbricas y el crecimiento de la conectividad de los dispositivos inteligentes ha motivado la búsqueda de nuevas soluciones de comunicación que permitan contrarrestar problemas sociales como la inseguridad ciudadana, siendo el robo en las viviendas uno de los más relevantes. Una solución para este tipo de problemas es automatizar el hogar usando sensores que permitan detectar intrusiones en los inmuebles.

Este trabajo se centra en el desarrollo de un prototipo de red LPWAN, que mediante el uso de las tecnologías LoRa y LoRaWAN combinadas con MQTT y una API REST, notifican actos de intrusión en una vivienda tanto al personal de seguridad de la zona residencial como al propietario del inmueble. El personal de seguridad utiliza una aplicación web para gestionar la autenticación de usuarios y supervisar, dentro de un mapa en tiempo real, el estado de los nodos de cada residencia. Por otra parte, el propietario del inmueble puede manejar la activación y desactivación de los nodos a través de una aplicación móvil. Las notificaciones push se han habilitado cada vez que se produce una intrusión o se desconecta un nodo. Los elementos hardware del prototipo trabajan de acuerdo con la especificación 1.0.2 del protocolo LoRaWAN en las bandas ISM 902\_928.

Los resultados obtenidos han demostrado que la solución propuesta funciona adecuadamente en entornos urbanos cumpliendo con las características de una red LPWAN, ya que proporciona comunicación de largo alcance y notificaciones inmediatas.

**Palabras clave:** LPWAN, LoRaWAN, LoRa, seguridad de la vivienda, detección de intrusos.

# **ABSTRACT**

Wireless networks constant expansion and the growth of smart devices connectivity has motivated the search of new communication solutions to provide benefits for social problems like citizen insecurity where home robbery is one of the most relevant. A solution this type of problem is home automation, by using sensors that can detect intrusions at a house.

This work focuses on the development of a prototype of an LPWAN network by using LoRa and LoRaWAN technologies combined with MQTT and an API REST to generate notification alerts of intrusion into a home for the security personal of residential area and the house owner. The security personal uses web application for manage user authentication and monitor, within a real-time map, the status of the nodes of each home. Otherwise, the house owner uses a mobile application for handle activation and deactivation of the nodes. Push notifications has been enabled whenever an intrusion occurs, or a node is disconnected. The elements hardware of the prototype work according to the specification 1.0.2 of the LoRaWAN protocol in the ISM 902\_928 bands. The results obtained have shown that the solution proposed works appropriately in urban environments complying with the characteristics of an LPWAN network, due to provides communication of long range and immediate notifications

**Keywords:** LPWAN, LoRaWAN, LoRa, Home Security, Intrusion detection

# **PRESENTACIÓN**

El presente documento se encuentra estructurado de la siguiente manera:

El Capítulo I presenta el problema que la solución propuesta pretende contrarrestar, los objetivos del proyecto, su alcance, la metodología de trabajo utilizada para su desarrollo y una breve descripción de la zona residencial en donde el prototipo fue puesto a prueba.

El Capítulo II define los fundamentos teóricos del trabajo realizado, así como una breve revisión literaria relacionada al tema de seguridad del hogar y a la tecnología LoRa.

El Capítulo III presenta una descripción detallada de los componentes del prototipo de red LPWAN junto con el diseño de su arquitectura, el software utilizado para el desarrollo del proyecto y la aplicación del marco de trabajo ágil SCRUM.

En el Capítulo IV se presenta la implementación de la solución propuesta, detallando su desarrollo y haciendo uso de los temas expuestos en el Capítulo III.

El Capítulo V muestra las distintas pruebas efectuadas al prototipo y los resultados obtenidos que permitieron determinar la factibilidad de la solución.

Para finalizar, el Capítulo VI presenta las conclusiones y recomendaciones obtenidas al haber finalizado este trabajo, así como una posible implementación del mismo.

# **CAPÍTULO I**

## **1. INTRODUCCIÓN**

### **1.1. ANTECEDENTES**

El término aplicación web progresiva (PWA) hace referencia a la creación de aplicaciones flexibles y adaptables mediante el uso exclusivo de tecnologías web [1]. Las PWA proporcionan la misma experiencia tanto en ordenadores de escritorio como en dispositivos móviles. Evolucionan desde páginas en las pestañas del navegador hasta aplicaciones inmersivas de alto nivel [2], lo que permite que sean multiplataforma [3]. Una de las principales ventajas de las PWA, además de ser multiplataforma, es no tener que pasar por todo el proceso de descarga, instalación y concesión de permisos en una tienda de aplicaciones, por el que tiene que pasar una aplicación nativa o híbrida, evitando por completo la burocracia de publicación de aplicativos de las tiendas. Una vez que un usuario muestra interés en una PWA, el buscador automáticamente sugerirá añadir un atajo de esta a la pantalla de inicio, el mismo que no se diferenciará al de una aplicación nativa [3]. Otro inconveniente de usar aplicaciones nativas con respecto a una PWA es que necesitan la decisión del usuario para tener instalada la versión actualizada, mientras una PWA siempre se tiene acceso a la versión más actualizada de la aplicación. Manticore Labs es una empresa de desarrollo de software, que se encuentra ubicada en Quito, cuenta con una cartera de 14 clientes entre ellos PetroEcuador, Uribe Schwarzkopf,Diners Club, etc. Manticore Labs ha desarrollado una aplicación híbrida, la cual entre sus funcionalidades, permite gestionar el inventario, ventas y recaudos de una empresa; esta aplicación se despliega a producción a través de las tiendas de aplicaciones tradicionales, Play Store de Google y App Store de Apple, lo que dificulta su actualización, debido al proceso que se lleva a cabo en estas tiendas de aplicaciones para aprobar su publicación, tomando entre 2 y 3 semanas, desde la solicitud de actualización hasta que los cambios en el aplicativo se ven reflejados en las tiendas. Además, Soporte de Google [4] indica que, para publicar la aplicación en la tienda de Play Store, se debe de pagar una cantidad de dinero.

El presente proyecto busca dar solución a esta problemática, mediante la reingeniería de esta aplicación a una PWA, la cual no necesita de una tienda de aplicaciones para ser instalada o actualizada en dispositivos.

### **1.2. OBJETIVOS**

#### **1.2.1. Objetivo general**

Aplicar la reingeniería de software en módulos de ventas, recaudos e inventario del aplicativo móvil de la empresa Manticore Labs a PWA mediante Scrum y Gitlab como herramienta DevOps.

#### **1.2.2. Objetivos específicos**

* Definir el alcance de la reingeniería de los módulos de ventas, recaudos e inventario.
* Establecer los requerimientos específicos a implementarse en la aplicación PWA.
* Rediseñar la base de datos y la arquitectura.
* Iniciar los flujos de integración y desarrollo continuo e implementarlos usando Gitlab.
* Implementar los requerimientos del aplicativo en la aplicación PWA.
* Probar usabilidad de la aplicación PWA.

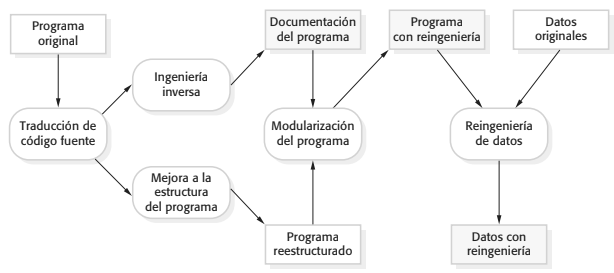
### **1.3. ALCANCE**

Por definir.

### **1.4. MARCO TEÓRICO**

**1.4.1. Reingeniería de software**

La reingeniería es el proceso que implica el análisis y alteración de un sistema software existente para reconstituirlo en una nueva forma. El proceso típicamente engloba una combinación de procesos como se muestra en la figura 1.1.



**Figura 1.** El proceso de reingeniería  
Fuente [5]

# **CAPÍTULO II**

## **2. REVISIÓN LITERARIA Y FUNDAMENTOS TEÓRICOS**

### **2.1. TRABAJOS RELACIONADOS**

Este apartado presenta una breve revisión de soluciones relacionadas tanto a seguridad para el hogar como a las tecnologías LoRa y LoRaWAN dentro del contexto de hogares inteligentes.

#### **2.1.1. Estudios previos**

La solución propuesta en [6], implementa un sistema de seguridad basado en las tecnologías Wi-Fi y GSM. Aquí se utiliza dos tipos de sensores que permiten detectar algún movimiento o incendio dentro del hogar. Para notificar dicho evento cada uno se conecta a una placa NodeMCU, que mediante Wi-Fi transporta el mensaje a un Raspberry Pi (master node). El envío de la notificación la realiza el módulo GSM que también se encuentra conectado al Raspberry. La notificación puede ser a través de un correo electrónico, un SMS o una llamada al teléfono de los usuarios previamente establecidos, es decir, al propietario del inmueble, a los bomberos, a la policía, etc. El usuario también tiene la posibilidad de acceder a los datos de los nodos y a la cámara del sistema a través de internet colocando la IP del Raspberry.

Por otro lado, el trabajo denominado Design and Implementation of a New Smart Home Control System Based on Internet of Things [7], describe el uso de una red Zigbee combinada con tecnología de red 3G / 4G para el monitoreo remoto del hogar. El sistema tiene varias aplicaciones que permiten garantizar la seguridad familiar dentro de la vivienda, entre ellas está la videovigilancia, implementando una cámara de video que permite al usuario monitorear el inmueble a través de su teléfono móvil o un aplicativo web.

Las soluciones [8] y [9] se enfocan en el uso de Bluetooth. En [8], se lo utiliza para establecer conexión con una PC, la cual actúa como un módulo de control principal, y que junto con Internet notifica, mediante SMS o correo electrónico, al dueño de la propiedad si alguien forcejea la puerta o ingresa a su vivienda. Para detectar dichas acciones se emplea dos tipos de sensores (vibración y movimiento). Otras funciones del sistema son: bloquear la puerta, encender una cámara y activar la alarma antirrobo. Ahora bien, en [9] se describe un sistema de hogar inteligente que utiliza servicios web basados en Bluetooth y REST como capa interoperable. Este sistema tiene características como autenticación de usuario mediante de una aplicación móvil, conectividad Bluetooth e Internet, control automatizado de electrodomésticos, sistema de seguridad y vigilancia contra incendios e intrusos. Este último incorpora dispositivos que sirven para detectar un incendio o intrusión en el inmueble, si ha ocurrido alguno de estos dos eventos, el sistema activa una sirena y notifica al usuario por correo electrónico.

En cuanto a LoRa y LoRaWAN, se han encontrado soluciones que utilizan estas tecnologías para detectar fugas de gas y controlar la iluminación de una vivienda. Por ejemplo, en [10] se presenta una arquitectura LoRaWAN que utiliza un sensor MQ2 junto con una placa TTGO para detectar y notificar al gateway la presencia de gas LPG en el entorno. La comunicación entre la placa y el gateway se realiza mediante LoRa. Cabe destacar que en este proyecto se propone implementar a futuro un Cluster Controller, que se encargará de gestionar la información de los sensores para que los guardias de seguridad puedan monitorearlos. Otro trabajo [11], implementa una red LoRaWAN que gestiona un termostato y las luces de una vivienda. En esta solución se implementó un gateway utilizando una Raspberry Pi3 B y un shield LoRaGo PORT. El encendido y apagado tanto de las luces como del termostato se lo puede realizar mediante un aplicativo web, móvil o un asistente de voz. Además, se utiliza MQTT para publicar los mensajes en el servidor de aplicaciones usado por LoRaWAN.

#### **2.1.2. Análisis**

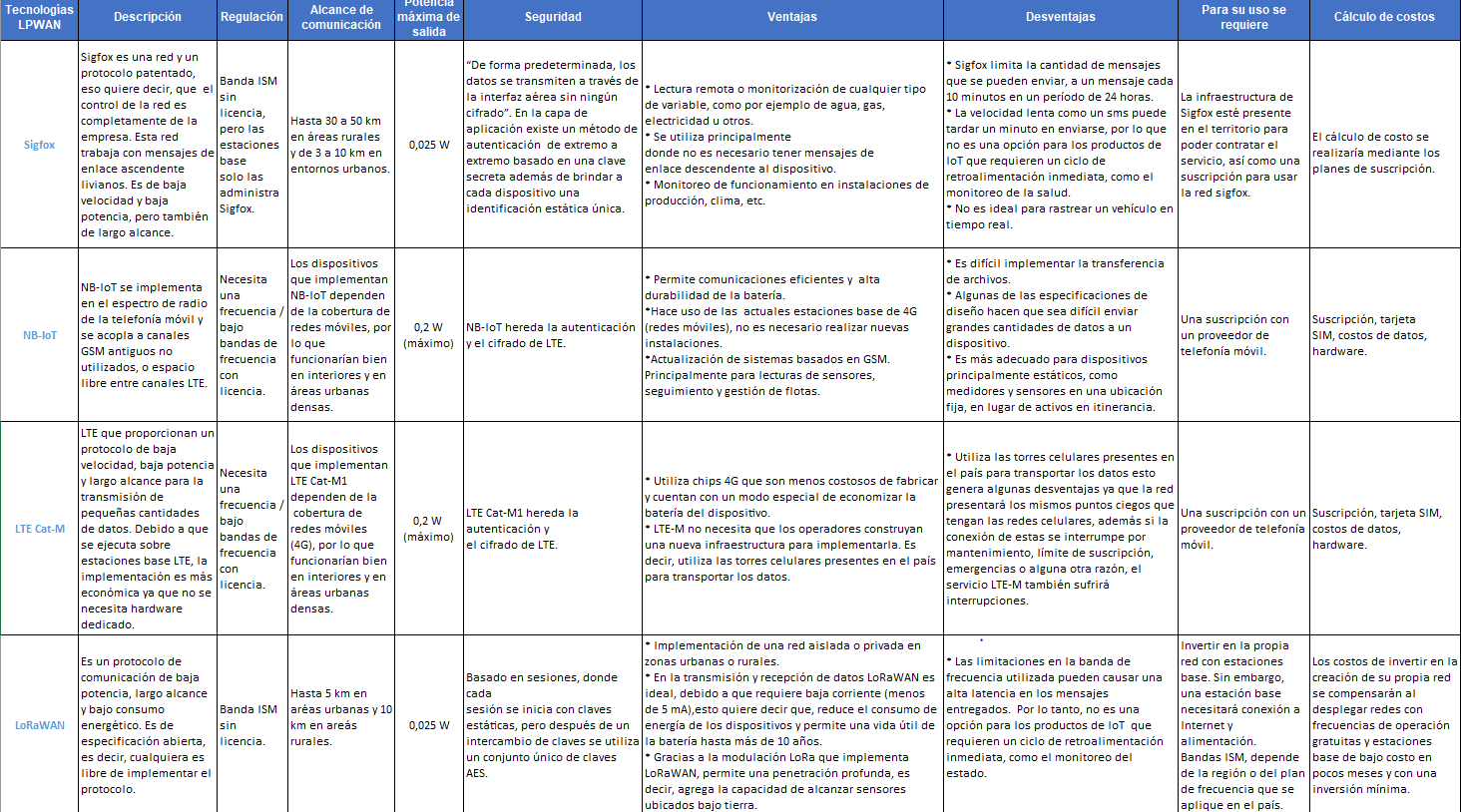
Los sistemas revisados anteriormente proponen varias formas de implementar una solución con respecto a la seguridad del hogar, enfatizando la detección de intrusos. Sin embargo, algunos inconvenientes que podrían llegar a presentar se relacionan con la tecnología utilizada para la comunicación y notificación de eventos detectados por los sensores, la usabilidad y la tranquilidad que brindan a los usuarios. Por ejemplo, en [6–9] utilizan Wi-Fi, Zigbee y Bluetooth como medio de comunicación entre nodos y elementos centrales, estos últimos, debido a la tecnología de corto alcance, tienen que ser ubicados en una zona específica, sin ningún tipo de obstáculo para que puedan establecer conexiones óptimas con el resto de los componentes.

Por otro lado, en [6,8] se especifica el uso de SMS como medio de notificación al propietario del inmueble. Implementar este tipo de tecnología podría llegar a ser costosa dependiendo de los mensajes que se envíen. Ahora bien, la solución [6] y [10] no presentan una interfaz gráfica que permita al usuario gestionar el estado de los nodos conectados al sistema.

Es importante señalar que ninguna solución cuenta con un aplicativo que brinde al personal encargado de la seguridad la posibilidad de monitorear el estado de los sensores que posee cada vivienda, ni notificarles directamente cuando una intrusión ocurra dentro de la misma, por lo que no se asegura que alguien acudirá al sitio a verificar lo sucedido. Como último punto, en las soluciones [10] y [11] se puede evidenciar que la tecnología LoRa junto con LoRAWAN se adaptan fácilmente a escenarios relacionados al contexto de hogares inteligentes.

### **2.2. MARCO TEÓRICO**

#### **2.2.1. SMART CITY**



**Tabla 3.** Comparación de tecnologías LoRaWAN

Fuente [52]

#### **2.2.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE**

Una metodología de desarrollo de software hace referencia a un marco de trabajo o framework cuya finalidad es la de optimizar, estructurar, planificar y controlar el proceso y desarrollo de un proyecto software [53].

Utilizar una metodología de desarrollo de software de forma correcta, permite al equipo de trabajo mejores estimaciones, comunicación continua con el cliente, entrega de sistemas estables, entendimiento claro de las tareas que tienen que realizar e identificación de dificultades o problemas que retrasan la finalización del proyecto en cuestión, garantizando elevar los estándares y el aseguramiento de éxito de un proyecto software [54]. Por otro lado, si no se hace uso de una metodología de desarrollo de software el resultado final del producto será impredecible y no se podrá controlar el avance del proyecto [55].

En la actualidad, se debe de tomar en cuenta qué metodología de desarrollo es la más adecuada para obtener un proyecto software de calidad. Existen algunas características a considerar como: la velocidad de finalización, el tamaño del sistema y el nivel de colaboración e interacción entre los miembros del equipo de desarrollo del software. Es por eso que, dentro de las metodologías de desarrollo, existen dos enfoques principales que son: las tradicionales y ágiles [56].

La metodología de desarrollo tradicional trabaja de manera lineal, rigurosa y disciplinada, esto quiere decir, que no se adapta adecuadamente a los cambios donde los requisitos pueden variar y se centra especialmente en planificar todo el trabajo a realizar mediante una documentación exhaustiva. Una vez que está todo detallado comienza el ciclo de desarrollo del proyecto software. Por otro lado, la metodología de desarrollo ágil surge para solventar los problemas que ocasiona la implementación de una metodología tradicional debido a que se centra especialmente en ser iterativa e incremental, adaptable y flexible a los cambios donde los requisitos no pueden predecirse o pueden variar y presentar constante comunicación entre cliente y el equipo de desarrollo durante el proceso [53]**.**

La metodología ágil se enfoca en la construcción de un proyecto satisfactorio en el menor tiempo posible, es por eso que se convierte en la más usada en el mundo del software. Dentro de la metodología ágil existen diferentes marcos de trabajos, tal es el caso de Scrum, el cual es un marco de trabajo ágil que permite construir y obtener un producto útil en poco tiempo, para que el cliente pueda probarlo en situaciones reales de consumo, permitiendo la mejora continua [53]. Por tal motivo, este marco de trabajo es adecuado para efectuar el desarrollo de la solución propuesta, misma que busca entregar un prototipo útil que se encuentra sujeto a cambios y pruebas constantes, permitiendo entregas continuas con el fin de atender mejor las necesidades y deseos del usuario final.

##### **2.2.4.1. Marco de trabajo Scrum**

Scrum es un marco de trabajo ágil que se enfoca especialmente en desarrollar, entregar y mantener de manera productiva productos complejos con el mayor valor posible. Este marco de trabajo se compone del equipo Scrum junto a los roles, eventos, artefactos y reglas.

###### **2.2.4.1.1. Roles de Scrum**

El equipo de trabajo Scrum define tres tipos de roles [57]:

* **Product Owner:** es la persona que va a estar en contacto con el cliente y el asegura que el equipo de trabajo comprenda de manera clara y concisa lo que se va a desarrollar.
* **Scrum Master:** es la persona que coordina al equipo de trabajo y el que guía el uso correcto de estándares Scrum en el proceso de desarrollo.
* **Development Team:** son las personas con habilidades especializadas para el desarrollo, que se encargarán de realizar las tareas definidas por el Product Owner.

###### **2.2.4.1.2. Artefactos**

Son herramientas que sirven para la implementación del proyecto software. Siendo estos los siguientes [57]:

* **Product Backlog (PB):** es una lista creada mediante las Historias de Usuario (H.U.) por el Product Owner, el cual la ordena por prioridades. Siendo una historia de usuario una descripción corta y esquemática de las funcionalidades que debe incorporar el proyecto software. Para la presente solución se ha propuesto el siguiente formato para la recopilación de historias de usuario, Tabla 4.

|  |  |
| --- | --- |
| **Historia de usuario** | **Identificación** |
| **Título:** | |
| **Descripción:** | |
| **Prioridad:** | |

**Tabla 4.** Formato de historias de usuario

El formato consta de los siguientes campos:

* + **Identificación:** es el código que identifica a una historia de usuario.
  + **Título:** es una descripción global del requerimiento del cliente.
  + **Descripción:** es una pequeña descripción de la funcionalidad que se desea realizar.
  + **Prioridad:** esta característica indica el nivel de importancia de la historia de usuario. Puede contener tres niveles: alta, media o baja.
* **Sprint Backlog:** es la descomposición de una H.U del P.B en tareas más pequeñas para alojarse en el Scrum board donde se va a ir reflejando el avance de cada tarea.
* **Incremento:** es el producto completo que vamos logrando Sprint por Sprint. Tomando en cuenta que los Sprints son ciclos de trabajo con una duración fija que va desde una a cuatro semanas, con el fin de obtener un producto final, utilizable y potencialmente liberable.

###### **2.2.4.1.3. Eventos [58]**

Son actividades que se realizan dentro del ciclo principal de Scrum llamado Sprint. Entre estos se tiene:

* **Sprint Planning:** es una reunión donde participa todo el equipo Scrum, con la finalidad de detallar, aclarar y delegar las características del producto a desarrollar.
* **Daily Scrum:** es una reunión que se lleva a cabo todos los días durante el Sprint y que tiene una duración de 15 minutos, con el fin de comprobar el progreso o inconvenientes que se tiene al realizar cada tarea asignada a los miembros del equipo de desarrollo.
* **Sprint Review:** es una reunión que dirige el Scrum Master, en la cual asisten el equipo de desarrollo junto a las partes interesadas, con fin de revisar y analizar si se logró cumplir con el objetivo del Sprint durante el trabajo realizado.
* **Sprint Retrospective:** es una reunión que se lleva a cabo con todo el equipo scrum, con el fin de conocer e inspeccionar aquello que produjo problemas durante el Sprint, de modo que se permita proponer mejoras para iniciar con un nuevo Sprint.

# **CAPÍTULO III**

## **3. DISEÑO DE LA ARQUITECTURA DEL PROTOTIPO DE RED LPWAN**

### **3.1. ELEMENTOS DE LA RED LORAWAN**

En base a la arquitectura que debe cumplir una red LoRaWAN, se ha indagado y establecido los elementos necesarios a utilizar para el desarrollo del prototipo de red propuesto, siendo estos los siguientes:

### **3.2. ARQUITECTURA DEL PROTOTIPO DE RED LPWAN**

### **3.3. HERRAMIENTAS DE DESARROLLO**

Para la implementación de las aplicaciones web y móvil del sistema de detección de intrusos se usaron las herramientas presentadas en las Tablas 9-16.

#### **3.3.1. Lenguajes de programación**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Aplicado en** | **Logo** |
| **HTML5** | Es un lenguaje de marcado de hipertexto (HTML, por sus siglas en inglés) es el componente más básico de un sitio web y permite estructurar y organizar el contenido del sitio web. HTML5 es la última versión que brinda nuevas características para que los sitios o aplicaciones web sean más diversos y de gran alcance [67]. | Aplicación web y móvil | Logo Html Html5 - Free image on Pixabay |
| **CSS** | Es un lenguaje de hojas de estilo en cascada (CSS, por sus siglas en inglés) usado para describir la presentación de un documento escrito en HTML o XML. Permite diseñar páginas web, por ejemplo, para modificar el color, tamaño y estilo de fuente y otras características [68]. | Aplicación web y móvil | Logo Css Css3 - Imagen gratis en Pixabay |
| **JavaScript** | Es un lenguaje de programación, dinámico, sencillo, ligero e interpretado, que admite estilos orientados a objetos y programación funcional. Hoy en día ha crecido de manera acelerada y es un lenguaje que se puede usar en distintos entornos, tales como Node.js, Apache CouchDB, entre otros [69]. | Aplicación web y móvil | javascript-shield-logo - Cantabria TIC |
| **TypeScript** | Es un lenguaje de programación de código abierto, que se basa en JavaScript, permite agregar definiciones de tipo estáticos, dando como resultado una mejor documentación y validez para que el código funcione correctamente.  El código TypeScript se puede transpilar en código JavaScript por medio del compilador de TypeScript o Babel, el nuevo código JavaScript que surge es limpio y simple que se puede ejecutar en un navegador, servidor Node.js o en aplicaciones, donde se ejecute JavaScript. [70]. | Aplicación web y móvil | Icono Archivo, tipo de, manuscrito, oficial Gratis de vscode |
| **C** | Es un lenguaje de programación estructurado, que proporciona gran flexibilidad de programación, pero muy baja comprobación de incorreciones, esto quiere decir, que es responsabilidad del programador revisar la sintaxis de programación o acciones que otros lenguajes realizan por sí mismos [71]. | Aplicación Nodo LoRa | clogo - Desafio Latam |

**Tabla 9**. Descripción de lenguajes de programación

#### **3.3.2. Frameworks de desarrollo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Aplicado en** | **Logo** |
| **Angular** | Es un framework de diseño de aplicaciones web y creación de aplicaciones de una sola página (SPA, por sus siglas en inglés), gratuito y de código abierto [72]. Usa como lenguaje de programación TypeScript, conjunto de JavaScript/ECMAScript que facilita mucho el desarrollo [73]**.** | Aplicación web | Angular (framework) - Wikipedia, la enciclopedia libre |
| **IONIC** | Es un framework gratuito y de código abierto, que ofrece una biblioteca de herramientas y componentes optimizados para dispositivos móviles y con el objetivo de crear aplicaciones web y nativas multiplataforma rápidas, interactivas y de alta calidad [74]. | Aplicación móvil | 5 Tips for building Ionic Framework Apps - Kumulos |
| **Sails.js** | Es un framework utilizado para la creación de aplicaciones modernas relacionadas con Node.js y que trabaja con el patrón de arquitectura de software, modelo vista controlador (MVC, por sus siglas en ingles). Sails.js permite la integración con diferentes bases de datos proporcionando una capa de acceso a datos simple. De igual forma, permite la integración con herramientas de vanguardia como los WebSocket y es compatible a nivel de interfaz con: Angular, React, Ios, Android y Windows Phone [75]. | Aplicación web y móvil | Logo resources | Sails.js |
| **Node JS** | Es un entorno ideado para la ejecución de JavaScript y para la creación de aplicaciones network escalables, construido con el motor de JavaScript V8 de Chrome y orientado a eventos asíncronos [76]. | Aplicación web y móvil | Node JS logo vector |

**Tabla 10**. Descripción de frameworks de desarrollo

#### **3.3.3. Librerías**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Utilizado** | **Logo** |
| **Leaflet** | Es una biblioteca JavaScript de código abierto, que sirve para desarrollar mapas eficientes e interactivos para todas las plataformas de escritorio y móviles. Brinda a los desarrolladores una API bien documentada de fácil uso, con funciones de mapeo y código fuente simple y legible [77]. | Aplicación web | Leaflet - a JavaScript library for interactive maps |
| **MOSCA** | Es una librería desarrollada en JavaScript, de código abierto para Node.js basado en el protocolo MQTT. Puede ser empleado como aplicación independiente o embebido en diferentes proyectos Node.js. Sirve para la comunicación entre dispositivos IOT mediante eventos que permiten suscribirse a tópicos o realizar publicaciones [78]. | Aplicación web | GitHub - moscajs/mosca: MQTT broker as a module |
| **EEPROM** | Memoria de solo lectura, borrable eléctricamente (EEPROM, por sus siglas en inglés) permite leer y escribir bytes de la EEPROM de Arduino, es decir, sirve para gestionar el uso de memorias en las placas, mediante los métodos update, get, put, read y write. Esta librería forma parte del IDE de Arduino, así que no hay necesidad de descargarla [79]. | Red LORAWAN | Using Eeprom In Arduino To Store Data - Arduino Uno, HD Png Download ,  Transparent Png Image - PNGitem |

**Tabla 11.** Descripción de librerías de implementación

#### **3.3.4. Servicios de software**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Aplicado en** | **Logo** |
| **AWS** | AWS (Amazon Web Service), es una plataforma en la nube que brinda servicios de integración de datos a nivel mundial, ofrece tecnologías de infraestructura de cómputo, almacenamiento y base de datos. De igual forma, brinda tecnologías innovadoras como inteligencia artificial, análisis de datos e internet de las cosas. AWS permite la integración en la nube a este tipo de tecnologías con aplicaciones existentes de manera rápida, fácil y rentable [80]. | Red LORAWAN | Amazon Web Services - Wikipedia, la enciclopedia libre |
| **ChirpStack** | Es un servidor LoRaWAN de código abierto, que proporciona los componentes para la creación de redes LPWAN basadas en el protocolo LoRaWAN, incorpora una interfaz amigable con el usuario fácil de usar, para administrar dispositivos y APIs Grpc y REST para integrar con infraestructuras externas [81]. | Red LORAWAN | ChirpStack open-source LoRaWAN<sup>®</sup> Network Server |
| **OneSignal** | Es un servicio de envío de notificaciones push por medio de diferentes plataformas que estén previamente registrados. OneSignal, posee una documentación API REST fácil de entender, para que los desarrolladores realicen integraciones sin ningún inconveniente [82]. | Aplicación web y móvil | Meet OneSignal at RustConf. We're hiring! |
| **Docker** | Es una aplicación que sirve para la creación y uso compartido de contenedores ligeros y portables con el objetivo de ejecutarse en las máquinas MacOS y Windows [83]. | Aplicación web y móvil | aldakur.net |
| **Putty** | Es un software de código abierto, que sirve para conectarse a servidores remotos para administrarlos, por medio de una sesión que se realiza por el cliente SSH y Telnet [84]. | Red LORAWAN | PuTTY libera una actualización de seguridad que corrige 8 graves errores. —  Una al Día |

**Tabla 12.** Descripción de servicios de software

#### **3.3.5. Base de datos**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Aplicado en** | **Logo** |
| **MySQL** | Es un gestor de bases de datos relacional de código abierto, permitiendo así disponer de esta herramienta para la realización de pequeños proyectos o prototipos de manera fiable, ordenada y estandarizada. MySQL se basa en la arquitectura cliente-servidor permitiendo la comunicación entre sí, para crear y administrar bases de datos basados en el modelo relacional [85]. | Aplicación web y móvil | mysql-logo - Red Code' |

**Tabla 13.** Descripción de la Base de datos MySQL

#### **3.3.6. Entorno de desarrollo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Aplicado en** | **Logo** |
| **Visual Studio Code** | Es un editor de texto o código fuente muy ligero y potente disponible para ejecutarse en los principales sistemas operativos Windows, MacOS y Linux. Posee varias extensiones para el desarrollo diferentes lenguajes de programación como: C ++, C #, Java, Python, PHP, Go) e incorpora soporte para JavaScript, TypeScript y Node.js [86]. | Aplicación web y móvil | Logo Visual Studio Code - Vector PNG |
| **Arduino IDE** | Es un software de código abierto escrito en Java y basado en Processing, sirve para facilitar la escritura de código y su carga en la placa Arduino. Está disponible para ejecutarse en los principales sistemas operativos Windows, MacOS X y Linux. Este software puede ser usado con cualquier placa Arduino [87]. | Red LORAWAN | Arduino - Wikipedia, la enciclopedia libre |

**Tabla 14**. Descripción de entornos de desarrollo

#### **3.3.7. Control de versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Aplicado en** | **Logo** |
| **Git** | Es un software de código abierto y gratuito, que permite realizar el control de versiones distribuido de código fuente, de igual manera está diseñado para manejar las versiones de proyectos pequeños y grandes, con velocidad y eficiencia [88]. | Aplicación web y móvil | Archivo:Git-logo.svg - Wikipedia, la enciclopedia libre |
| **GitHub** | Es un repositorio online que sirve para alojar y gestionar proyectos, permite el trabajo colaborativo mediante el sistema de control de versiones Git [89]. | Aplicación web y móvil | GitHub Logos and Usage · GitHub |
| **GitHub Desktop** | Es una aplicación de escritorio desarrollado por GitHub que permite administrar proyectos, realizar seguimiento mediante un historial y realizar cambios de versiones de manera gráfica sin hacer uso de la línea de comandos que brinda Git [90]**.** | Aplicación web y móvil | File:Github-desktop-logo-symbol.svg - Wikimedia Commons |

**Tabla 15**. Descripción de herramientas de control de versiones

#### **3.3.8. Recursos y Diagramación**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Aplicado en** | **Logo** |
| **Lucidchart** | Es un software web que permite al usuario bosquejar y compartir diagramas de flujo profesional de forma colaborativa en tiempo real. Además, permite la elaboración de organigramas, pizarras, mapas, entre otros [91]. | Creación de ilustraciones | Lucidchart |
| **Flaticon** | Es un sitio web donde se recopila y se aloja iconos en formato PNG, SVG, EPS, PSD y Base64, para poder usar en proyectos de desarrollo o ilustraciones [92]. | Aplicación web y móvil | Iconos vectoriales gratis - SVG, PSD, PNG, EPS y fuente de iconos - Miles  de iconos gratis. |
| **Adobe Illustrator** | Es un software de diseño vectorial que permite a los diseñadores e ilustradores crear logotipos, iconos, gráficos y redimensionar ilustraciones y vallas publicitarias para pantallas celulares y sitios web, sin afectar su nitidez y características [93]. | Aplicación web y móvil | Descargar Adobe Illustrator Portable - Brogramador |

**Tabla 16**. Descripción de herramientas adicionales

### **3.4. APLICACIÓN DE LA METODOLOGÍA DE DESARROLLO**

#### **3.4.1. Definición de roles**

En base a lo descrito dentro del marco de trabajo Scrum se establecieron los roles necesarios para la implementación del prototipo, siendo el equipo de desarrollo los autores el presente proyecto, tomando en cuenta que todo lo realizado en estas secciones se lo hizo bajo la supervisión de su director y codirector.

#### **3.4.2. Definición de historias épicas**

La Tabla 17, muestra las historias épicas consideradas para el desarrollo del proyecto propuesto.

|  |  |  |
| --- | --- | --- |
| **Código** | **Título** | **Prioridad** |
| **HE01** | Implementación y comunicación de los componentes LoRa y LoRaWAN. | Alta |
| **HE02** | Desarrollo de un backend para la integración con los componentes LoRaWAN. | Alta |
| **HE03** | Desarrollo de la aplicación web y móvil que permita recibir y consumir los servicios desde una API RESTful. | Media |

**Tabla 17.** Detalle de historias épicas

#### **3.4.3. Pila de producto (Product Backlog)**

A continuación, se presenta el listado de historias de usuario en las Tablas 18-26 necesarios para la implementación del proyecto.

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE01-01** |
| **Título:** Notificación de intrusiones en una vivienda. | |
| **Descripción:** Como elemento detector de intrusos, necesito enviar mensajes de los eventos capturados por el sensor magnético y de movimiento utilizando la tecnología LoRaWAN, para que la infraestructura de backend pueda notificar dicha acción a los usuarios suscritos al sistema**.** | |
| **Prioridad:** Alta | |

**Tabla 18.** Historia de usuario HE01-01

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE03-03** |
| **Título**: Monitoreo del estado de los nodos de una vivienda. | |
| **Descripción:** Como administrador del sistema, necesito visualizar un mapa en tiempo real que presente el estado de los nodos de cada vivienda, para que el personal de seguridad pueda tomar las acciones necesarias dependiendo de estos estados. | |
| **Prioridad:** Media | |

**Tabla 19.** Historia de usuario HE03-03

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE03-02** |
| **Título:** Gestión del acceso a la aplicación web y móvil. | |
| **Descripción:** Como módulo de control de acceso, necesito autenticar a través de un login las credenciales ingresadas por el usuario, para permitirle hacer uso de las funciones de la aplicación web o móvil. | |
| **Prioridad:** Media | |

**Tabla 20.** Historia de usuario HE03-02

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE01-02** |
| **Título:** Configuración de los componentes de la red LoRaWAN. | |
| **Descripción:** Como administrador de red, necesito llevar a cabo diferentes configuraciones basadas en el estándar LoRaWAN y en regulaciones establecidas en el país referentes al uso de bandas libres, para que los elementos dentro de la red puedan comunicarse entre sí. | |
| **Prioridad:** Alta | |

**Tabla 21.** Historia de usuario HE01-02

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE02-01** |
| **Título:** Procesamiento de información. | |
| **Descripción:** Como backend de aplicación,necesito capturar, interpretar y procesar los datos obtenidos tanto del aplicativo web y móvil como de los elementos de la red, para gestionarlos adecuadamente. | |
| **Prioridad:** Alta | |

**Tabla 22.** Historia de usuario HE02-01

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE03-01** |
| **Título:** Gestión de la información de un usuario final. | |
| **Descripción:** Como administrador del sistema, necesito realizar operaciones CRUD en la aplicación web, para gestionar la información de cada usuario final registrado en el sistema. | |
| **Prioridad:** Media | |

**Tabla 23.** Historia de usuario HE03-01

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE02-02** |
| **Título:** Envío de avisos de desconexión y de actividad sospechosa. | |
| **Descripción:** Como backend de aplicación, necesito enviar notificaciones referentes a la desconexión del nodo o a la detección de intrusos tanto a la aplicación web como al dispositivo móvil, para que el personal de seguridad pueda acudir en ayuda. | |
| **Prioridad:** Alta | |

**Tabla 24.** Historia de usuario HE02-02

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE03-04** |
| **Título:** Gestión del estado de los nodos. | |
| **Descripción:** Como usuario final, necesito activar o desactivar los nodos a través del aplicativo móvil**,** para que el sensor vinculado a la vivienda emita mensajes únicamente si existe alguna intrusión en el inmueble. | |
| **Prioridad:** Media | |

**Tabla 25.** Historia de usuario HE03-04

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE03-05** |
| **Título:** Registro de alertas de detección de intrusos. | |
| **Descripción:** Como administrador del sistema, necesito tener un registro, dentro del aplicativo web, de las notificaciones emitidas al detectarse intrusos en las viviendas registradas en el sistema, para constatar que las alertas fueron recibidas por el personal de seguridad. | |
| **Prioridad:** Baja | |

**Tabla 26.** Historia de usuario HE03-05

#### **3.4.4. Planificación de los Sprints (Release planning)**

Mediante el release plannig presentado en la Tabla 27, se especifica historias de usuario que serán desarrolladas dentro de cada sprint.

|  |  |  |  |
| --- | --- | --- | --- |
| **RELEASE PLANNING** | | | |
| **SPRINT 1** | **SPRINT 2** | **SPRINT 3** | **SPRINT 4** |
| HE01-01 | HE02-01 | HE03-01 | HE03-04 |
| HE01-02 | HE02-02 | HE03-02 | HE03-05 |
|  |  | HE03-03 |  |

**Tabla 27**. Release plannig

Cabe señalar que para ponderar el esfuerzo de cada historia de usuario se utilizó la técnica Plannig Poker, la cual consistió en la participación activa de los miembros del equipo Scrum. Esta técnica hace uso de una baraja modificada donde las cartas están enumeradas bajo la serie de Fibonacci (0, 1, 2, 3, 5, 8, 13). Se hicieron rondas con estas para determinar la estimación de cada historia de usuario.

##### **3.4.4.1. Sprint 1**

###### **3.4.4.1.1. Objetivo del Sprint**

Conectar los dispositivos LoRaWAN mediante LoRa y TCP/IP con el afán de comunicar actos de intrusión al servidor de ChirpStack.

###### **3.4.4.1.2. Historias de usuario**

El primer sprint se desarrolló en base a las historias de usuario presentadas en las Tablas 28 y 29, las cuales muestran el esfuerzo y los criterios de aceptación relacionados a cada una de ellas.

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE01-01** |
| **Título:** Notificación de intrusiones en una vivienda. | |
| **Descripción:** Como elemento detector de intrusos, necesito enviar mensajes de los eventos capturados por el sensor magnético y de movimiento utilizando la tecnología LoRaWAN, para que la infraestructura de backend pueda notificar dicha acción a los usuarios suscritos al sistema**.** | |
| **Prioridad:** Alta | **Esfuerzo:** 13 |
| **Criterios de Aceptación:** | |
| * Cuando los nodos detecten un acto de intrusión, el mensaje tiene que llegar con un payload de 1 al servidor de aplicaciones LoRaWAN de ChirpStack | |

**Tabla 28**. Historia de usuario HE01-01 para el Sprint 1

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **HE01-02** |
| **Título:** Configuración de los componentes de la red LoRaWAN. | |
| **Descripción:** Como administrador de red, necesito llevar a cabo diferentes configuraciones basadas en el estándar LoRaWAN y en regulaciones establecidas en el país referentes al uso de bandas libres, para que los elementos dentro de la red puedan comunicarse entre sí. | |
| **Prioridad:** Alta | **Esfuerzo:** 8 |
| **Criterios de Aceptación:** | |
| * Al enviar un paquete mediante la comunicación LoRa este debe utilizar el plan de frecuencias ISM 902\_928 * Los paquetes de enlace ascendente únicamente deben utilizar SF del 7 al 10 y el Data Rate del 0 al 4. | |

**Tabla 29.** Historia de usuario HE01-02 para el Sprint 1

**CAPÍTULO VI**

## **6. CONCLUSIONES, RECOMENDACIONES Y POSIBLE IMPLEMENTACIÓN**

### **6.1. CONCLUSIONES**

### **6.2. RECOMENDACIONES**

### **6.3. POSIBLE IMPLEMENTACIÓN**

# **REFERENCIAS BIBLIOGRÁFICAS**

[1] "Introduction to progressive web apps", MDN Web Docs, 2020. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Progressive\_web\_apps/Introduction. [Accessed: 26- jun- 2020].

[2] A. Osmani, "Getting Started with Progressive Web Apps  |  Google Developers", Google Developers, 2020. [Online]. Available: https://developers.google.com/web/updates/2015/12/getting-started-pwa. [Accessed: 10- Jun- 2020].

[3] P. LePage, "Tu primera Progressive Web App  |  Web Fundamentals  |  Google Developers", Google Developers, 2020. [Online]. Available: https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/?hl=es. [Accessed: 10- Jun- 2020].

[3] T. Ater, Building Progressive Web Apps, 1st ed. Sebastopol: O'Reilly, 2020

[4] How to use the Play Console - Play Console Help", Support.google.com, 2020. [Online]. Available: https://support.google.com/googleplay/android-developer/answer/6112435. [Accessed: 09- Jul- 2020].

[5] Sommerville, I., 2011. *Ingeniería de Software*. México: Pearson Educación̤.

# **ANEXOS**

## **Anexo 1.** Diagramas de la tarjeta de desarrollo WiFi LoRa 32 (V2).

## **Anexo 2.** Sprint backlog y Sprint Review

## **Anexo 3.** Código fuente de los nodos detectores de intrusión.

## **Anexo 4.** Códec para decodificar la carga útil del dispositivo.

## **Anexo 5.** Mockups de las aplicaciones web y móvil.

## **Anexo 6.** Manual de usuario.