**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](#_Toc70526855)

[**DECLARACIÓN DE AUTORÍA** II](#_Toc70526856)

[**DEDICATORIA** III](#_Toc70526857)

[**AGRADECIMIENTO** IV](#_Toc70526858)

[**RESUMEN** IX](#_Toc70526859)

[**ABSTRACT** X](#_Toc70526860)

[**PRESENTACIÓN** XI](#_Toc70526861)

[**CAPÍTULO I** 12](#_Toc70526862)

[1. INTRODUCCIÓN 12](#_Toc70526863)

[1.1. ANTECEDENTES 12](#_Toc70526864)

[1.2. OBJETIVOS 13](#_Toc70526865)

[1.3. ALCANCE 13](#_Toc70526866)

[1.4. MARCO TEÓRICO 13](#_Toc70526867)

[1.5. METODOLOGÍA 21](#_Toc70526868)

[**CAPÍTULO II** 25](#_Toc70526869)

[2. REVISIÓN LITERARIA Y FUNDAMENTOS TEÓRICOS 25](#_Toc70526870)

[2.1. TRABAJOS RELACIONADOS 25](#_Toc70526871)

[**2.1.1. Estudios previos** 25](#_Toc70526872)

[**2.1.2. Análisis** 26](#_Toc70526873)

[2.2. MARCO TEÓRICO 27](#_Toc70526874)

[**2.2.1. SMART CITY** 27](#_Toc70526875)

[**2.2.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE** 27](#_Toc70526876)

[**2.2.4.1. Marco de trabajo Scrum** 28](#_Toc70526877)

[**2.2.4.1.1. Roles de Scrum** 28](#_Toc70526878)

[**2.2.4.1.2. Artefactos** 28](#_Toc70526879)

[**2.2.4.1.3. Eventos [58]** 29](#_Toc70526880)

[**CAPÍTULO III** 30](#_Toc70526881)

[3. DISEÑO DE LA ARQUITECTURA DEL PROTOTIPO DE RED LPWAN 30](#_Toc70526882)

[3.1. ELEMENTOS DE LA RED LORAWAN 30](#_Toc70526883)

[3.2. ARQUITECTURA DEL PROTOTIPO DE RED LPWAN 30](#_Toc70526884)

[3.3. HERRAMIENTAS DE DESARROLLO 30](#_Toc70526885)

[**3.3.1. Lenguajes de programación** 31](#_Toc70526886)

[**3.3.2. Frameworks de desarrollo** 32](#_Toc70526887)

[**3.3.3. Librerías** 33](#_Toc70526888)

[**3.3.4. Servicios de software** 34](#_Toc70526889)

[**3.3.5. Base de datos** 35](#_Toc70526890)

[**3.3.6. Entorno de desarrollo** 35](#_Toc70526891)

[**3.3.7. Control de versiones** 36](#_Toc70526892)

[**3.3.8. Recursos y Diagramación** 36](#_Toc70526893)

[3.4. APLICACIÓN DE LA METODOLOGÍA DE DESARROLLO 37](#_Toc70526894)

[**3.4.1. Definición de roles** 37](#_Toc70526895)

[**3.4.2. Definición de historias épicas** 37](#_Toc70526896)

[**3.4.3. Pila de producto (Product Backlog)** 37](#_Toc70526897)

[**3.4.4. Planificación de los Sprints (Release planning)** 40](#_Toc70526898)

[**3.4.4.1. Sprint 1** 40](#_Toc70526899)

[**3.4.4.1.1. Objetivo del Sprint** 40](#_Toc70526900)

[**3.4.4.1.2. Historias de usuario** 40](#_Toc70526901)

[6. CONCLUSIONES, RECOMENDACIONES Y POSIBLE IMPLEMENTACIÓN 43](#_Toc70526902)

[6.1. CONCLUSIONES 43](#_Toc70526903)

[6.2. RECOMENDACIONES 43](#_Toc70526904)

[6.3. POSIBLE IMPLEMENTACIÓN 43](#_Toc70526905)

[**REFERENCIAS BIBLIOGRÁFICAS** 44](#_Toc70526906)

[**ANEXOS** 46](#_Toc70526907)

[Anexo 1. Diagramas de la tarjeta de desarrollo WiFi LoRa 32 (V2). 46](#_Toc70526908)

[Anexo 2. Sprint backlog y Sprint Review 46](#_Toc70526909)

[Anexo 3. Código fuente de los nodos detectores de intrusión. 46](#_Toc70526910)

[Anexo 4. Códec para decodificar la carga útil del dispositivo. 46](#_Toc70526911)

[Anexo 5. Mockups de las aplicaciones web y móvil. 46](#_Toc70526912)

[Anexo 6. Manual de usuario. 46](#_Toc70526913)

**ÍNDICE DE FIGURAS**

[**Figura 1.** El proceso de reingeniería Fuente [5] 16](#_Toc70526808)

[**Figura 2.** Esquema deScrum Fuente [6] 18](#_Toc70526809)

[**Figura 3.** DevOps y el ciclo de vida de una aplicación Fuente [9] 20](#_Toc70526810)

**ÍNDICE DE TABLAS**

[**Tabla 1.** Formato de historias de usuario 18](#_Toc70526914)

[**Tabla 4.** Formato de historias de usuario 28](#_Toc70526915)

[**Tabla 5**. Descripción de lenguajes de programación 31](#_Toc70526916)

[**Tabla 6**. Descripción de frameworks de desarrollo 32](#_Toc70526917)

[**Tabla 7.** Descripción de librerías de implementación 33](#_Toc70526918)

[**Tabla 8.** Descripción de servicios de software 34](#_Toc70526919)

[**Tabla 9.** Descripción de la Base de datos MySQL 35](#_Toc70526920)

[**Tabla 10**. Descripción de entornos de desarrollo 35](#_Toc70526921)

[**Tabla 11**. Descripción de herramientas de control de versiones 36](#_Toc70526922)

[**Tabla 12**. Descripción de herramientas adicionales 36](#_Toc70526923)

[**Tabla 13.** Detalle de historias épicas 37](#_Toc70526924)

[**Tabla 14.** Historia de usuario HE01-01 37](#_Toc70526925)

[**Tabla 15.** Historia de usuario HE03-03 38](#_Toc70526926)

[**Tabla 16.** Historia de usuario HE03-02 38](#_Toc70526927)

[**Tabla 17.** Historia de usuario HE01-02 38](#_Toc70526928)

[**Tabla 18.** Historia de usuario HE02-01 38](#_Toc70526929)

[**Tabla 19.** Historia de usuario HE03-01 39](#_Toc70526930)

[**Tabla 20.** Historia de usuario HE02-02 39](#_Toc70526931)

[**Tabla 21.** Historia de usuario HE03-04 39](#_Toc70526932)

[**Tabla 22.** Historia de usuario HE03-05 39](#_Toc70526933)

[**Tabla 23**. Release plannig 40](#_Toc70526934)

[**Tabla 24**. Historia de usuario HE01-01 para el Sprint 1 40](#_Toc70526935)

[**Tabla 25.** Historia de usuario HE01-02 para el Sprint 1 41](#_Toc70526936)

# **RESUMEN**

**Palabras clave:**

# **ABSTRACT**

**Keywords:**

# **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. Aplicación móvil

Una aplicación móvil, es una aplicación informática desarrollada para que funcione en un dispositivo móvil inteligente, tablet u otro. Este tipo de aplicaciones se encuentran en tiendas online, por medio de las cuales son accedidas por los usuarios que deseen usarlas [10].

Existen varias formas de desarrollar una aplicación móvil. Cada una de ellas presentan diferentes características y limitaciones. Entre Los tipos de aplicaciones móviles que se conocen son: nativas, web e híbridas [11].

Opciones en el desarrollo de una aplicación móvil:

* Aplicaciones móviles nativas.
* Aplicaciones móviles nativas multiplataforma
* Aplicaciones móviles híbridas
* Aplicaciones Web Progresivas

1.4.1.1 Aplicaciones móviles nativas

Las aplicaciones móviles nativas están escritas en el lenguaje de programación y los marcos proporcionados por el propietario de la plataforma y se ejecutan directamente en el sistema operativo del dispositivo, como iOS y Android.

1.4.1.2 Aplicaciones móviles nativas multiplataforma

Las aplicaciones móviles nativas multiplataforma se pueden escribir en una variedad de marcos y lenguajes de programación diferentes, pero se compilan en una aplicación nativa que se ejecuta directamente en el sistema operativo del dispositivo.

1.4.1.3 Aplicaciones móviles híbridas

Las aplicaciones móviles híbridas se crean con tecnologías web estándar, como JavaScript, CSS y HTML5, y se incluyen como paquetes de instalación de aplicaciones. A diferencia de las aplicaciones nativas, las aplicaciones híbridas funcionan en un 'contenedor web' que proporciona un tiempo de ejecución del navegador y un puente para las API de dispositivos nativos a través de Apache Cordova.

1.4.1.2 Aplicaciones Web Progresivas (PWA)

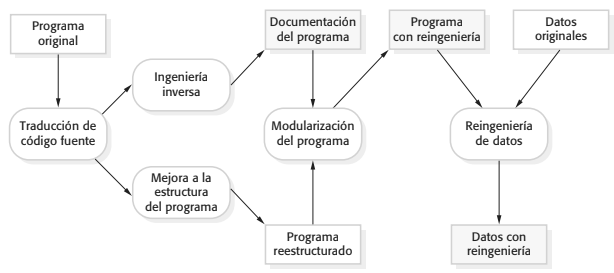
Las aplicaciones web progresivas son aplicaciones web que utilizan APIs y funciones emergentes del navegador web junto a una estrategia tradicional de mejora progresiva para ofrecer una aplicación nativa. Las aplicaciones web progresivas son un patrón de diseño útil, aunque no son un estándar formalizado. Se puede pensar que PWA es similar a AJAX u otros patrones similares que abarcan un conjunto de atributos de aplicación, incluido el uso de tecnologías y técnicas webs específicas.

Para poder llamar PWA a una aplicación web, técnicamente hablando debe tener las siguientes características: Contexto seguro (HTTPS), uno o más Servicio Workers y un archivo de manifiesto [1].

* **Contexto seguro (HTTPS):** La aplicación web se debe servir a través de una red segura. Ser un sitio seguro no solo es una buena práctica, sino que también establece tu aplicación web como un sitio confiable, especialmente si los usuarios necesitan realizar transacciones seguras. La mayoría de las funciones relacionadas con una PWA, como la geolocalización e incluso los servicios *workers*, solamente están disponibles cuando la aplicación se ha cargado mediante HTTPS.
* ***Service workers*:** Un service worker es un *script* que permite interceptar y controlar cómo un navegador web maneja tus solicitudes de red y el almacenamiento en caché de activos. Con los servicios worker, los desarrolladores web pueden crear páginas web rápidas y fiables junto con experiencias fuera de línea.
* **El archivo manifest:** Un archivo JSON que controla cómo se muestra tu aplicación al usuario y garantiza que las aplicaciones web progresivas sean detectables. Describe el nombre de la aplicación, la URL de inicio, los iconos y todos los demás detalles necesarios para transformar el sitio web en un formato similar al de una aplicación.

1.4.2. 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]

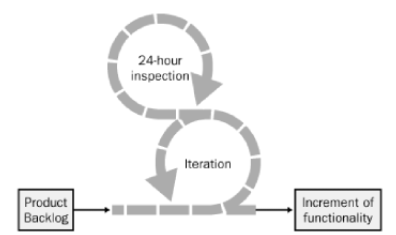
La figura 1.1 es un modelo general del proceso de reingeniería. La entrada al proceso es un sistema heredado, en tanto que la salida es una versión mejorada y reestructurada del mismo programa. Las actividades en este proceso de reingeniería son las siguientes:

* **Traducción del código fuente:** Con una herramienta de traducción, el programa se convierte de un lenguaje de programación antiguo a una versión más moderna del mismo lenguaje, o a un lenguaje diferente.
* **Ingeniería inversa:** El programa se analiza y se extrae información de él. Esto ayuda a documentar su organización y funcionalidad. De nuevo, este proceso es, por lo general, completamente automatizado.
* **Mejoramiento de la estructura del programa:** La estructura de control del programase analiza y modifica para facilitar su lectura y comprensión, lo cual suele estar parcialmente automatizado, pero se requiere regularmente alguna intervención manual.
* **Modularización del programa:** Las partes relacionadas del programa se agrupan y, donde es adecuado, se elimina la redundancia. En algunos casos, esta etapa implicará refactorización arquitectónica (por ejemplo, un sistema que use muchos almacenes de datos diferentes puede refactorizarse para usar un solo depósito). Éste es un proceso manual.
* **Reingeniería de datos:** Los datos procesados por el programa cambian para reflejar cambios al programa. Esto puede significar la redefinición de los esquemas de bases de datos y convertir las bases de datos existentes a la nueva estructura. Por lo general, hay que limpiar los datos. Esto implica encontrar y corregir errores, eliminar registros duplicados, etcétera. Hay herramientas disponibles para auxiliar en la reingeniería de datos.

Según Somerville [5] indica que para realizar una reingeniería no es necesario realizar todos los procesos especificados anteriormente.

1.4.3. SCRUM

Scrum es un marco de trabajo para dirigir problemas complejos, que implementa una estructura iterativa, incremental, en la que, al inicio de una iteración el equipo revisa las tareas a realizarse, y selecciona lo que se convertirá al final del sprint en un incremento de la funcionalidad [6].



**Figura 2.** Esquema deScrum  
  
Fuente [6]

Un equipo de trabajo de SCRUM está conformado por tres roles:

* ***Product Owner*:** determina los requisitos iniciales generales y el plan de las entregas.
* ***Scrum Master*:** responsable de enseñar SCRUM y asegurarse de que todos sigan las reglas y prácticas.
* ***Development Team*:** responsable de desarrollar la funcionalidad y del éxito de cada iteración y del proyecto.

Scrum cuenta con los siguientes artefactos:

* ***Product Backlog* (PB):** Los requerimientos para el sistema son listados en el *Product Backlog*, que es, una lista de historias de usuario (HU). Siendo una historia de usuario una descripción corta y esquemática de las funcionalidades que debe incorporar el software. Para la presente solución se ha propuesto el formato mostrado en la Tabla 1. para la recopilación de historias de usuario.

El *product Owner* es el responsable del contenido, priorización y disponibilidad del *Product Backlog*.

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

**Tabla 1.** 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*:** Define las tareas del *Product Backlog* que se realizarán en el Sprint, además se ven reflejados los avances en cada tarea.
* **Incremento:** es el producto completo que se va logrando Sprint a 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.

Eventos de SCRUM:

* ***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.

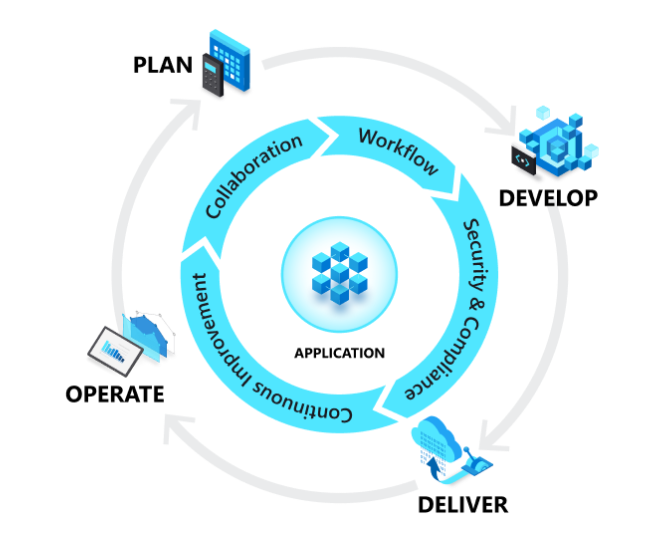
1.4.4. GitLab

1.4.5. DevOps

DevOps es un conjunto de prácticas que trabaja para automatizar e integrar los procesos entre el desarrollo de software y los equipos de TI, para que puedan construir, probar y lanzar software de manera más rápida y confiable.

El término DevOps se formó combinando las palabras "desarrollo" y "operaciones" y significa un cambio cultural que cierra la brecha entre los equipos de desarrollo y operación, que históricamente funcionaron en silos [8].

DevOps influye en el ciclo de vida de la aplicación a lo largo de sus fases de planificación, desarrollo, entrega y operación. Cada fase depende de las demás y las fases no son específicas de un rol. En una verdadera cultura DevOps, cada rol está involucrado en cada fase hasta cierto punto.



**Figura 3.** DevOps y el ciclo de vida de una aplicación  
  
Fuente [9]

Fases de DevOps [9]:

* **Planificar:** En la fase de planificación, los equipos de DevOps idean, definen y describen características y capacidades de las aplicaciones y sistemas que están construyendo. Realizan un seguimiento del progreso en niveles bajos y altos de granularidad, desde tareas de un solo producto hasta tareas que abarcan carteras de varios productos. La creación de trabajos pendientes, el seguimiento de errores, la gestión del desarrollo de software ágil con Scrum, y la visualización del progreso con tableros son algunas de las formas en que los equipos de DevOps planifican con agilidad y visibilidad.
* **Desarrollar:** La fase de desarrollo incluye todos los aspectos de la codificación: redacción, prueba, revisión e integración del código por parte de los miembros del equipo, así como la construcción de ese código en artefactos de construcción que se pueden implementar en varios entornos.
* **Lanzar:** El lanzamiento es el proceso de implementación de aplicaciones en entornos de producción de una manera consistente y confiable. La fase de lanzamiento también incluye la implementación y configuración de la infraestructura conforma esos entornos.
* **Operar:** La fase de operación implica el mantenimiento, la supervisión y la resolución de problemas de aplicaciones en entornos de producción. Al adoptar las prácticas de DevOps, los equipos trabajan para garantizar la confiabilidad del sistema, la alta disponibilidad y el objetivo de cero tiempos de inactividad a la vez que refuerzan la seguridad y la gobernanza. Los equipos de DevOps buscan identificar problemas antes de que afecten la experiencia del cliente y mitigar los problemas rápidamente cuando ocurren.

Prácticas de DevOps[9]:

* ***Continuous integration and continuous delivery* (CI/CD):** La integración continua es una práctica de desarrollo de software en la que los desarrolladores combinan los cambios de código con frecuencia en la rama de código principal. La integración continua emplea pruebas automatizadas, que se ejecutan cada vez que se confirma un nuevo código para que el código en la rama principal sea siempre estable.
* ***Version control*:** El control de versiones es la práctica de administrar el código en versiones, realizar un seguimiento de las revisiones y el historial de cambios para que el código sea fácil de revisar y recuperar. Esta práctica generalmente se implementa utilizando sistemas de control de versiones como Git, que permiten que varios desarrolladores colaboren en la creación de código.
* ***Agile software development*:** Es un enfoque de desarrollo de software que enfatiza la colaboración en equipo, los comentarios de los clientes y usuarios y una alta adaptabilidad al cambio a través de ciclos de lanzamiento cortos. Los equipos que utilizan metodologías ágiles brindan cambios y mejoras continuos a los clientes, recopilan sus comentarios y luego aprenden y se ajustan en función de los deseos y necesidades del cliente.
* ***Infrastructure as code*:** La infraestructura como código define los recursos y topologías del sistema de una manera descriptiva que permite a los equipos administrar esos recursos como lo harían con el código. Esas definiciones también se pueden almacenar y versionar en sistemas de control de versiones, donde se pueden revisar y revertir, nuevamente como código.
* ***Configuration management*:** La gestión de la configuración se refiere a la gestión del estado de los recursos en un sistema que incluye servidores, máquinas virtuales y bases de datos. Mediante el uso de herramientas de gestión de la configuración, los equipos pueden implementar cambios de forma controlada y sistemática, reduciendo los riesgos de modificar la configuración del sistema.
* ***Continuous monitoring*:** El monitoreo continuo significa tener una visibilidad completa y en tiempo real del rendimiento y el estado de toda la pila de aplicaciones, desde la infraestructura subyacente que ejecuta la aplicación hasta los componentes de software de nivel superior. Esta visibilidad consiste en la recopilación de telemetría y metadatos, así como en la configuración de alertas para condiciones predefinidas que requieren la atención de un operador.

### 1.5. METODOLOGÍA

Para la realización de este proyecto se usó las siguientes fases: inicio, planificación, implementación, revisión, y lanzamiento. Las fases descritas permitieron que la reingeniería del aplicativo móvil se desarrolle de manera ordenada, clara y eficiente.

**Fase de inicio**

En esta fase se definió el alcance de la reingeniería en los módulos de ventas, recaudos e inventario, para ello se revisó la documentación de los módulos. Luego se seleccionó los requerimientos a implementarse, los mismos que formaron parte del Backlog que fueron evaluados mediante su la complejidad. Además, se realizará la reestructuración del diseño del aplicativo para que finalmente poner en marcha los flujos de integración y desarrollo en Gitlab. No se realizará la ingeniería inversa pues, se contó con la documentación necesaria de la aplicación actual. Las siguientes fases se repitieron en cada Sprint, y los cuales tuvieron una duración de 2 semanas.

**Fase de planificación**

Los requerimientos fueron recogidos y formaron parte del Sprint Backlog.

**Fase de implementación**

Se realizó la re-codificación de funcionalidades del aplicativo móvil para que se comporte como una PWA. Para el control de las actividades se realizó reuniones a mitad del sprint. Las reuniones sirvieron para evaluar los avances y tomar medidas correctivas

**Fase de revisión**

Por cada sprint finalizado se realizó la presentación en una reunión *Sprint Review* del avance funcional a la parte interesada. El equipo SCRUM también realizó una reunión de retrospectiva.

**Fase de lanzamiento**

En base al nivel de aceptación se levantó en producción el aplicativo con las nuevas funcionalidades. Los últimos Sprints fueron tomados para efectuar las pruebas de usabilidad del aplicativo.

# **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**

#### **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 5**. 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 6**. 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 7.** 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 8.** 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 9.** 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 10**. 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 11**. 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 12**. 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 13.** 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 14.** 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 15.** 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 16.** 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 17.** 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 18.** 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 19.** 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 20.** 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 21.** 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 22.** 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 23**. 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 24**. 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 25.** 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. [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.

[6] Schwaber, K., 2004. Agile project management with Scrum. Redmond, Wash.: Microsoft Press.

[7] GitLab. 2021. What is GitLab?. [online] Available at: <https://about.gitlab.com/what-is-gitlab/> [Accessed 28 April 2021].

[8] Atlassian. 2021. What is DevOps? | Atlassian. [online] Available at: <https://www.atlassian.com/devops> [Accessed 28 April 2021].

[9] Azure.microsoft.com. 2021. What is DevOps? DevOps Explained | Microsoft Azure. [online] Available at: <https://azure.microsoft.com/en-us/overview/what-is-devops/> [Accessed 28 April 2021].

[10] M. García, «USOS Y TIPOS DE APLICACIONES MÓVILES», 2011.

[11] V. Javier Cuello, Diseñando apps para móviles, español, Primera edición. Barcelona: TugaMOVIL, 2013.

[12] Amazon Web Services, Inc. 2021. Mobile Application Development. [online] Available at: <https://aws.amazon.com/es/mobile/mobile-application-development/> [Accessed 28 April 2021].

[13] Amazon Web Services, Inc. 2021. Mobile Application Development. [online] Available at: <https://aws.amazon.com/es/mobile/mobile-application-development/> [Accessed 28 April 2021].

# **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.