

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACIÓN WEB PROGRESIVA (PWA), PARA DETERMINAR SI UN AUTOBUS URBANO ESTA EN SU LÍMITE DE CAPACIDAD DE PASAJEROS

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

ANDRÉS DAVID VELASCO YÉPEZ

andres.velasco@epn.edu.ec

JOSSUÉ ANDRÉS DUTÁN SALAS

jossue.dutan@epn.edu.ec

DIRECTOR: Ing. VICENTE ADRIAN EGUEZ SARZOSA, MSc.

adrian.eguez@epn.edu.ec

CODIRECTOR: Ing. CARLOS ESTALESMIT MONTENEGRO ARMAS, MSc.

carlos.montenegro@epn.edu.ec

Quito, enero 2021

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por **Andrés David Velasco Yépez** y **Josué Andrés Dután Salas**, bajo nuestra supervisión.



Ing. Vicente Adrian Eguez Sarzosa, MSc.
DIRECTOR DE PROYECTO



Firmado electrónicamente por:
CARLOS ESTALESMIT
WILLAM MONTENEGRO
ARMAS

Ing. Carlos Estalesmit Montenegro Armas, MSc.
CODIRECTOR DE PROYECTO

DECLARACIÓN

Nosotros, **Andrés David Velasco Yépez** y **Jossué Andrés Dután Salas**, 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.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Andrés David Velasco Yépez



Jossué Andrés Dután Salas

AGRADECIMIENTOS

Agradezco a mis padres por haberme dado todo su apoyo y en sus esfuerzos en darme la mejor educación de calidad posible, como también a mis maestros, a mis compañeros de universidad, colegas de trabajo por brindarme los consejos y experiencias necesarias para llevar este proyecto a cabo. A la universidad en especial por darme la oportunidad de ser un profesional de excelencia.

Andrés David Velasco Yépez

Agradezco a mi mamá, quien me ha apoyado incesantemente con mucho esfuerzo y amor. Agradezco a mi abuelita que me enseñó la importancia de ser constante en cualquier meta que me proponga.

Agradezco a mi papá por todas sus enseñanzas y consejos.

Agradezco a mi pareja por su ayuda, apoyo y compañía.

Agradezco a mi compañero de tesis por su dedicación.

Agradezco a mi tutor por todo lo aprendido y el tiempo obsequiado.

Finalmente, a mis amigos, compañeros y profesores, en general, a todos de quienes he aprendido al cruzar esta travesía.

Jossué Andrés Dután Salas

CONTENIDO

| | |
|---|-----------|
| Resumen | 1 |
| Abstract | 2 |
| 1 INTRODUCCIÓN | 3 |
| 1.1 Planteamiento del Problema | 3 |
| 1.2 Objetivos | 4 |
| 1.2.1 Objetivo General | 4 |
| 1.2.2 Objetivos Específicos | 4 |
| 1.3 Alcance | 4 |
| 1.4 Marco teórico | 5 |
| 1.4.1 Aplicación Web Progresiva (PWA) | 5 |
| 1.4.2 Modelo Vista Controlador | 5 |
| 1.4.3 Metodología de desarrollo de software Scrum | 7 |
| 1.4.4 Internet de las cosas (IoT) | 8 |
| 1.4.5 Herramientas | 8 |
| 1.5 Estructura del Documento | 13 |
| 2 METODOLOGÍA | 14 |
| 2.1 Scrum | 14 |
| 2.1.1 Roles de Scrum | 14 |
| 2.1.2 Product Backlog | 15 |
| 2.1.3 Sprint 0 | 16 |
| 2.1.4 Sprint 1 | 33 |
| 2.1.5 Sprint 2 | 36 |
| 2.1.6 Sprint 3 | 42 |
| 3 RESULTADOS Y DISCUSIÓN | 49 |
| 3.1 Pruebas de Usabilidad | 49 |
| 3.1.1 El cuestionario | 49 |
| 3.1.2 La puntuación | 49 |
| 3.1.3 Resultados | 51 |

| | |
|---|-----------|
| 3.2 Pruebas de Funcionalidad | 52 |
| 3.2.1 Resultados | 52 |
| 4 CONCLUSIONES | 63 |
| 4.1 Conclusiones | 63 |
| 4.2 Recomendaciones | 64 |
| 5 REFERENCIAS BIBLIOGRÁFICAS | 66 |
| 6 ANEXOS | I |
| 6.1 Anexo 1: Representación gráfica de resultados de la encuesta sobre el uso del transporte público | I |
| 6.2 Anexo 2: Diagrama entidad relación de la base de datos | VI |
| 6.3 Anexo 3: Arquitectura de microservicios | VIII |
| 6.4 Anexo 4: Historias de Usuario | IX |
| 6.5 Anexo 5: Sprint Backlog Sprint 0 | XI |
| 6.6 Anexo 6: Sprint Backlog Sprint 1 | XV |
| 6.7 Anexo 7: Documentación detallada del sprint: 1 | .XXXVIII |
| 6.8 Anexo 8: Sprint Backlog Sprint 2 | .XLVIII |
| 6.9 Anexo 9: Sprint Backlog Sprint 3 | LXIV |

RESUMEN

Históricamente, el uso del transporte público en el Distrito Metropolitano de Quito ha presentado notables falencias en el manejo del límite de capacidad de sus unidades de transporte. Se propone desarrollar una aplicación web progresiva (PWA) para determinar si un autobús de transporte público está en su límite de capacidad de pasajeros. El desarrollo incluye una aplicación web de gestión y una aplicación web progresiva para el usuario final, para lo cual se decidió usar el marco de trabajo de Scrum. Se cumplió con el desarrollo de estas aplicaciones. En el sistema web se administra la información que se presenta en la aplicación web progresiva. En la aplicación web progresiva se puede visualizar las paradas, rutas, imágenes del interior del bus así como información del conductor. También es posible guardar paradas a favoritos, encontrar los buses que circulan una determinada ruta en tiempo real y mostrar su velocidad en ese instante. En la encuesta inicial y durante su presentación se demostró el interés de los usuarios en el desarrollo de esta aplicación. Posteriormente, al final del desarrollo se obtuvieron excelentes resultados en la encuesta de usabilidad (SUS), los cuales comprobaron que el correcto funcionamiento del aplicativo.

Palabras clave: pasajeros, transporte, aplicación web, aplicación móvil, desarrollo web, PWA, Scrum, microservicios, websocket, internet de las cosas.

ABSTRACT

Historically, the use of public transport in the Metropolitan District of Quito has had notable shortcomings in managing the capacity limit of its transport units. It is proposed to develop a progressive web application (PWA) to determine whether a public transport bus is at its passenger capacity limit. The development includes a management web application as well as a progressive web application for the end user, for which it was decided to use the Scrum framework. The development of these applications was accomplished. The information presented in the progressive web application is managed in the web system. In the progressive web application it is possible to visualize the bus stops, the routes, images from inside the bus as well as the driver information. It is also possible to save your favorite stops, find buses running on a certain route in real time and show its speed in that instant. In the initial survey and during the presentation of this application, the users demonstrated a lot of interest on the development of this application. Once the development was finished, excellent results were obtained in the usability survey (SUS) which proved the proper functioning of the application.

Keywords: passengers, transport, web application, mobile application, web development, PWA, Scrum, microservices, websocket, internet of things.

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

La movilidad en el Distrito Metropolitano de Quito (DMQ), por medio del transporte terrestre público y privado, es un aspecto fundamental y de uso cotidiano. Sin embargo, persiste en el conocimiento colectivo de los usuarios que el servicio de transporte está lejos de ser de excelente calidad.

En un sondeo realizado en el año 2012, el 38 % de los encuestados indicó que les disgusta usar el transporte público porque las unidades van llenas [1]. Además, en una encuesta realizada en el 2017 por la Agencia Metropolitana de Tránsito (AMT), un 25.89% de los usuarios calificó como deficiente, malo y regular al respeto por el límite de capacidad de la unidad. [2]

Cuando un autobús excede su capacidad de pasajeros genera malestar entre sus usuarios; empujones, maltrato y hacinamiento forman parte del disgusto que sienten los pasajeros en este escenario. El hacinamiento podría provocar daños en la salud y en los aspectos psicológicos de los usuarios, ya que producen sensaciones de cansancio, ansiedad, inseguridad, invasión de privacidad [3]. Además de generar inconvenientes al momento de entrar o salir del vehículo, puesto los usuarios pueden quedar atrapados entre sus puertas o ser víctimas de robo y acoso. El usuario suele determinar cuándo un autobús excede su capacidad de pasajeros de manera empírica al momento de visualizarlo o tomarlo.

Para solucionar esta problemática se propone desarrollar una Aplicación Web Progresiva (PWA) enfocada en un entorno web y móvil, la cual permita conocer al usuario, si la unidad de transporte público que está esperando en su respectiva parada, se encuentra llena de pasajeros. Esta información le permitirá al usuario tomar una decisión: abordar la unidad de transporte público, esperar a otra unidad de transporte o buscar otra alternativa de movilización.

1.2 OBJETIVOS

1.2.1 Objetivo General

Desarrollo de una aplicación web progresiva (PWA) para determinar si la capacidad de pasajeros de un autobús de transporte público está al límite, mediante la implementación del marco de trabajo de Scrum.

1.2.2 Objetivos Específicos

1. Levantar los requerimientos y las necesidades de los usuarios del transporte público al momento de tomar un autobús en una parada.
2. Analizar y diseñar una solución informática que permita a los usuarios decidir si tomar o esperar un automotor del servicio de transporte público en una parada utilizando el marco de trabajo de Scrum.
3. Implementar la solución utilizando una aplicación web progresiva.
4. Proponer recomendaciones sobre el uso del transporte público basadas en las evaluaciones del uso del aplicativo a los usuarios.

1.3 ALCANCE

Para tratar la problemática se propone el desarrollo de una Aplicación Web Progresiva (PWA) que permitirá a los usuarios del transporte público del Distrito Metropolitano de Quito determinar si un autobús está lleno y que permitirá al personal del servicio de transporte gestionar las rutas de viaje. Sin embargo, la aplicación no se lanzará en un ambiente de producción.

1.4 MARCO TEÓRICO

1.4.1 Aplicación Web Progresiva (PWA)

Las PWA (conocidas como Progressive Web Applications en inglés) son aplicaciones web que utilizan tecnologías web modernas para ofrecer una experiencia similar a una aplicación móvil nativa en los dispositivos móviles. Estas aplicaciones necesitan de un servidor para establecer la comunicación pertinente, son accedidas a través de una URL y pueden ser indexadas por los motores de búsqueda [4]

Una aplicación debe cumplir con ciertos requisitos para que sea considerada una aplicación web progresiva [5]:

1. Progresiva: puede, sin ser obligatorio, ser usada en navegadores viejos, pero es funcional en las versiones más recientes de los navegadores.
2. Descubrible: permite que el contenido podrá ser encontrado por los motores de búsqueda.
3. Instalable: permite a los usuarios ver un icono del aplicativo en la pantalla de inicio del dispositivo.
4. Enlazable: se puede compartir con solo enviar la URL.
5. Independiente de la red: puede funcionar offline o con una conexión lenta.
6. Re-engageable: se facilita la reconexión con la aplicación mediante notificaciones cuando haya contenido nuevo disponible.
7. Responsive: puede ser usada en cualquier dispositivo con un navegador.
8. Segura: la conexión entre el usuario y la aplicación está protegida contra terceros que intenten acceder a datos sensibles

1.4.2 Modelo Vista Controlador

Modelo Vista Controlador (MVC) es un patrón de diseño de software que se ha popularizado para construir aplicaciones web, es un patrón universalmente aceptado, a través de

varios lenguajes y marcos de trabajo (frameworks). MVC fue introducido por primera vez en Smalltalk'80 por Krasner y Pope. [6]

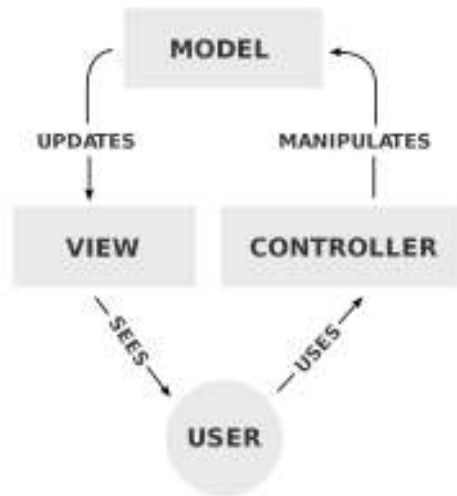


Figura 1.1: Arquitectura MVC

MVC está compuesto de los tres siguientes componentes [6]:

1. **Modelo:** gestiona los datos de la aplicación, contiene los datos de aplicación y la definición e implicación de la lógica de negocio. Un modelo puede ser un objeto único o es una composición de objetos. Además, esta capa permite la comunicación con la base de datos.
2. **Vista:** esta capa es la interfaz mediante la cual el usuario interactúa con la aplicación. La vista tiene la responsabilidad de mostrar los resultados de los datos que contiene el modelo. No obstante, solo muestra los atributos necesarios y oculta los atributos innecesarios.
3. **Controlador:** Un controlador es la conexión entre el usuario y el sistema, se encarga de responder las peticiones del usuario. El controlador maneja tanto el Modelo como la Vista. Controla cómo fluyen los datos en el modelo, lleva a cabo la ejecución que altera el estado del modelo de datos y actualiza la vista tan pronto como los datos se alteran.

1.4.3 Metodología de desarrollo de software Scrum

Scrum es una forma ágil de administrar un proyecto generalmente en el desarrollo de software. A pesar de que SCRUM es percibido en el desarrollo de software como una metodología, en realidad es un marco para gestionar un proceso. [7]

Scrum consta de protocolos o eventos que forman parte del proceso de scrum y varían en cada equipo. [8]

1. Organizar el backlog: revisión y mantenimiento de la lista de tareas del equipo.
2. Planificación del sprint: es una reunión para determinar lo que se hará en el sprint (alcance).
3. Sprint: es el periodo en el que el equipo de trabajo de scrum realiza las actividades para finalizar el incremento y agregarle valor al producto.
4. Daily: reunión diaria breve que responde a las preguntas:
 - ¿Qué hice ayer?
 - ¿Qué haré hoy?
 - ¿He tenido algún problema?
5. Revisión del sprint: reunión para ver el producto demo del incremento.
6. Retrospectiva del sprint: reunión para comentar lo que ha salido bien y lo que se necesita mejorar.

Un equipo de SCRUM está integrado por tres cargos específicos.

1. Scrum Master: es el experto en Scrum del equipo, se enfoca en ayudar al equipo a optimizar su trabajo y se encarga de gestionar las planificaciones y reuniones.
2. Product Owner: es el encargado de conocer y gestionar los requerimientos de los clientes o usuarios y de priorizar el trabajo que debe realizar el equipo.
3. Equipo de Desarrollo: son los miembros del equipo que contribuyen a la realización de las tareas planificadas de cada sprint.

1.4.4 Internet de las cosas (IoT)

Define la tendencia de conectar objetos físicos, como refrigeradores, bombillas, dispositivos inteligentes, entre otros, al internet. Estos dispositivos físicos tienen la característica de recibir y transferir datos a través de la red sin la intervención humana. [9]





Los objetos físicos, sensores, software y tecnologías de interconexión conforman un sistema tradicional de IoT. IoT se complementa de otras tecnologías para ofrecer un mejor servicio al usuario, entre ellas están la Inteligencia Artificial (AI), Cloud Computing y Big Data.

1.4.5 Herramientas

Tabla 1.1: Herramientas


| Nombre | Descripción | Utilizado en |
|--|--|------------------------------------|
|  Angular | Angular es una plataforma de desarrollo para crear aplicaciones web eficientes y sofisticadas de una sola página. [10]. Angular es de código abierto desarrollado en TypeScript por Google. Usa HTML como lenguaje de plantilla y clases o componentes donde se encuentran las variables y métodos. | Aplicación web. |
|  Node.js | Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. [11] | Aplicación web y móvil. |
|  NestJS | NestJS es un framework para crear aplicaciones del lado del servidor Node.js. Está construido y es compatible con TypeScript (también permite a los desarrolladores codificar en JavaScript puro) y combina elementos de OOP (Programación Orientada a Objetos), FP (Programación Funcional) y FRP (Programación Reactiva Funcional). [12] | Aplicación web y aplicación móvil. |

| | | |
|---|---|---|
|  <p>Ionic</p> | <p>Ionic Framework es un kit de herramientas de interfaz de usuario de código abierto para crear aplicaciones móviles y de escritorio con tecnologías web (HTML, CSS y JavaScript). Posee integraciones para frameworks populares como Angular y React . [13]</p> | <p>Aplicación móvil.</p> |
|  <p>Lucidchart</p> | <p>Es una plataforma web que permite crear prototipos de software, diagramas, diseños, entre otros, de una forma colaborativa y en tiempo real.</p> | <p>Diseño de base de datos y arquitectura de la aplicación.</p> |
|  <p>Gitlab</p> | <p>Es una plataforma web de DevOps. Proporciona a sus usuarios el manejo de control de versiones de repositorios Git, herramientas de planificación, de gestión de wikis, de seguimiento de problemas y de integración continua / despliegue continuo. [14]</p> | <p>Aplicación móvil y aplicación web.</p> |
|  <p>Typeorm</p> | <p>TypeORM es un ORM (Object-Relational Mapping) que puede ejecutarse en plataformas NodeJS y puede usarse con TypeScript y JavaScript. Su objetivo es proporcionar características que lo ayuden a desarrollar cualquier tipo de aplicación que utilice bases de datos, desde pequeñas aplicaciones con pocas tablas hasta aplicaciones empresariales a gran escala con múltiples bases de datos. [15]</p> | <p>Aplicación móvil y aplicación web.</p> |
|  <p>MySQL</p> | <p>MySQL, el sistema de gestión de bases de datos SQL de código abierto más popular, es desarrollado, distribuido y respaldado por Oracle Corporation. Las bases de datos MySQL son relacionales y funciona en cliente / servidor o sistemas integrados. [16]</p> | <p>Base de datos.</p> |

| | | |
|--|---|--------------------------------|
|  <p>MongoDB</p> | <p>MongoDB es una base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube. [17]</p> | <p>Base de datos.</p> |
|  <p>Docker</p> | <p>Docker es una tecnología que permite la creación y el uso de contenedores de Linux®. Los contenedores se pueden usar como máquinas virtuales extremadamente livianas y modulares. Además, se obtiene flexibilidad con estos contenedores: se puede crearlos, implementarlos, copiarlos y moverlos de un entorno a otro. [18]</p> | <p>Base de datos.</p> |
|  <p>Webstorm</p> | <p>WebStorm es un IDE que proporciona soporte completo para JavaScript, TypeScript, HTML, CSS, así como para marcos de trabajo como React, Angular y Vue.js sin requerir complementos adicionales.</p> <p>Además de las aplicaciones del lado del cliente, WebStorm sirve de ayuda al desarrollar aplicaciones del lado del servidor con Node.js, aplicaciones móviles con React Native o Cordova y aplicaciones de escritorio con Electron. [19]</p> | <p>Aplicación web y móvil.</p> |
|  <p>Postman</p> | <p>Es una plataforma que se utiliza para el desarrollo de API (Application Programming Interface). Esta plataforma generalmente se la utiliza para hacer peticiones http a determinadas APIs ya sean internas o de terceros y generar una colección de estas mismas con el fin de poder realizar pruebas que validen el comportamiento de estos servicios web [20].</p> | <p>Aplicación web y móvil.</p> |

| | | |
|--|--|--------------------------------|
|  <p>DBEaver</p> | <p>Es una herramienta de administración de bases de datos. Con DBEaver la información se presenta en una especie de hoja de cálculo en la cual se puede usar para manipular la información como también crear informes analíticos basados en los registros de los diferentes almacenamientos de datos y poder expórtalos en un formato determinado [21].</p> | <p>Base de datos.</p> |
|  <p>GitKraken</p> | <p>Es una herramienta que proporciona una elegante interfaz gráfica de usuario para poder trabajar con git y poder llevar el control de los repositorios git de manera simple y rápida. Esta herramienta se integra fácilmente con plataformas de git como los son github, gitlab, etc.</p> | <p>Aplicación web y móvil.</p> |
|  <p>Adobe Experience Design (Adobe XD)</p> | <p>Es una herramienta de edición de gráficos que permite diseñar interfaces gráficas de usuario para aplicaciones web, aplicaciones móviles o aplicaciones de escritorio. Básicamente esta herramienta permite al diseñador realizar prototipos de media o alta fidelidad de la aplicación de manera simple y rápida [22].</p> | <p>Mockups.</p> |
|  <p>Auth0</p> | <p>Es una plataforma que se especializa en "la identidad como servicio"(IDaaS). Auth0 proporciona al usuario un conjunto de APIs a los desarrolladores con el fin de realizar el proceso de autenticación y gestión de usuarios de manera fácil y así poder enfocarse en la lógica del negocio del producto de software en el que estén trabajando.</p> | <p>Aplicación web y móvil.</p> |
|  <p>Node Package Manager</p> | <p>Es el sistema de gestión de paquetes que el entorno de Node.js utiliza por defecto. Principalmente npm se enfoca en ser un repositorio en la nube para publicar proyectos de Node.js y en proporcionar una línea de comandos para interactuar con ya mencionados proyectos [23].</p> | <p>Aplicación web y móvil.</p> |

| | | |
|--|--|---|
|  <p>Open Street Map API</p> | <p>Conjunto de librerías que Open Street Maps proporciona a los desarrolladores para poder integrar sus mapas o consultar o gestionar información geográfica en sus aplicaciones [24].</p> | <p>Aplicación web y móvil.</p> |
|  <p>Raspberry PI</p> | <p>Es un ordenador de placa reducida de bajo coste que está diseñado para facilitar su uso en proyectos multidisciplinarios [25].</p> | <p>Aplicativo IOT.</p> |
|  <p>Man-lab-package</p> | <p>Conjunto de bibliotecas de Javascript y Typescript desarrollados por el equipo de Manticore-labs.</p> | <p>Aplicación web y aplicación móvil.</p> |
|  <p>Bootstrap</p> | <p>Es un conjunto de herramientas de componentes web y estilos de código abierto para el diseño de aplicaciones web.</p> | <p>Aplicación web.</p> |
|  <p>Angular Material</p> | <p>Conjunto de bibliotecas de estilos y componentes web que están basadas en la guía de diseño de "Material Design" para proyectos hechos con el marco de trabajo de Angular.</p> | <p>Aplicación web.</p> |

| | | |
|--|--|--------------------------------|
|  <p>PrimeNG</p> | <p>Librería de componentes gráficos de usuario para aplicaciones web y que es compatible para diversas tecnologías y frameworks como lo son Angular, React y Java.</p> | <p>Aplicación web y móvil.</p> |
|--|--|--------------------------------|

1.5 ESTRUCTURA DEL DOCUMENTO

El presente trabajo está organizado de la siguiente manera. En la sección 2 se describe la metodología utilizada y los avances por sprint del proyecto. En la sección 3, se presentan los resultados y la discusión por cada una de las etapas desarrolladas. Finalmente, en la sección 4 se presentan las conclusiones y las recomendaciones del presente trabajo de titulación.

2 METODOLOGÍA

2.1 SCRUM

El desarrollo de este proyecto incluye una la aplicación móvil y una aplicación web. En la aplicación web que es en donde se va a gestionar toda la información que el usuario va a visualizar en la aplicación móvil. Se decidió usar el marco de trabajo de Scrum. Este marco de trabajo se centra en la entrega continua de software con el máximo valor posible al cliente de una manera iterativa e incremental.

2.1.1 Roles de Scrum

2.1.1.1 Dueño del Producto

El dueño de producto es el representante del cliente en este caso es la empresa “Manticore Labs”, se definió al Magíster Adrian Eguez como el “Product Owner” del proyecto.

2.1.1.2 Scrum Master

El Scrum Master tiene la responsabilidad de que se entiendan y se pongan en práctica los valores y reglas del marco de Scrum, de tal forma que, se definió como Scrum Master al ingeniero Cristian Lara el cual es el director de proyectos de “Manticore Labs”.

2.1.1.3 Equipo de Desarrollo

Los equipos de desarrollo son los responsables de llevar a cabo todas las tareas necesarias para generar incrementos del producto valor, en consecuencia, se definió como equipo de

desarrollo a los señores:

- Dután Jossué.
- Velasco Andrés.

2.1.1.4 Esquema roles Scrum

Recapitulando, el equipo de desarrollo se encuentra definido de la siguiente forma:

| Rol Scrum | Responsable |
|------------------|-------------------------------|
| Product Owner | Adrian Eguez |
| Scrum Master | Cristian Lara |
| Development Team | Andrés Velasco y Jossué Dután |

Tabla 2.1: Roles de Scrum

2.1.2 Product Backlog

El product backlog del proyecto se encuentra conformado por las historias de usuario que pueden ser consultadas en su totalidad en el 6.4 Anexo 4.

Nomenclaturas de la tabla:

- F: Funcional
- N: No Funcional
- M: Aplicación Móvil
- W: Aplicación Web
- IOT: Internet de las cosas
- B: Tarea de backend.
- 001: Número de historia.

2.1.3 Sprint 0

2.1.3.1 Sprint Planning: 0

Los objetivos de este sprint son:

- Realizar la encuesta y analizar sus resultados con el fin de poder determinar las funcionalidades que tendrá el aplicativo móvil.
- Diseñar la arquitectura del sistema y su arquitectura de datos.
- Maquetar las interfaces de usuario del aplicativo (Mockups).
- Levantar los respectivos ambientes de desarrollo: “backend“, “frontend“ y el entorno de desarrollo para el aplicativo móvil.

2.1.3.2 Sprint Backlog: 0

Para el sprint cero se consideraron las historias de usuario descritas en el 6.5 Anexo 5

2.1.3.3 Ejecución del Sprint

Encuesta Descriptiva

Para la captación de requerimientos del producto software, se decidió realizar una encuesta del tipo descriptiva con respuestas cerradas por cada pregunta. Con el fin de reflejar las condiciones o conductas presentes en el transporte público del distrito metropolitano de Quito que causan la problemática presente en este ámbito.

Para definir la estructura de la encuesta nos hemos basado en varias preguntas de la encuesta realizada por la Agencia Metropolitana de Transito (AMT) [26] en la cual tiene entre sus principales objetivos:

- Evaluar el mejoramiento de la calidad del servicio del transporte público en el DMQ,

- Conocer la opinión sobre el trato que perciben los usuarios por parte de los conductores y ayudantes.

- Conocer la percepción de los usuarios sobre el transporte de pasajeros respetando el aforo máximo de viajeros y la conducción con puertas cerradas.

- Estimar el tiempo de espera para tomar un autobús.

- Conocer la percepción de los usuarios sobre el respeto de los límites de velocidad establecidos dentro del área del Distrito Metropolitano de Quito.

Contenido de la Encuesta

El contenido de la encuesta tanto como preguntas como respuestas se las puede encontrar en el 6.1 Anexo 1.

Realización de la encuesta

La encuesta se la realizó digitalmente a través de la plataforma de formularios de Google y se distribuyó su enlace en las redes sociales Facebook y WhatsApp.

Resultados de la encuesta

Los resultados completos de la encuesta realizada se los puede encontrar en el 6.1 Anexo 1.



Figura 2.1: Gráfico pastel: resultados pregunta 8

¿Consideras útil conocer cuáles unidades de transporte te permitirán llegar a tu parada de destino?

183 respuestas



Figura 2.2: Gráfico pastel: resultados pregunta 10

¿Consideras conveniente incluir la información del conductor que opera la unidad de transporte en la aplicación? (Al igual como lo hacen las aplicaciones de Uber o Cabify)

182 respuestas

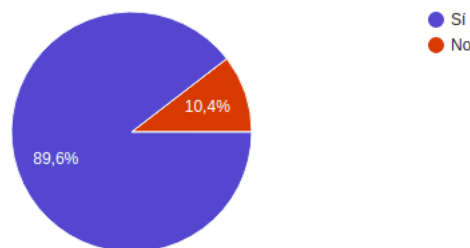


Figura 2.3: Gráfico pastel: resultados pregunta 13

Conclusiones de la encuesta

En base a los resultados de la encuesta se decidió tomar como prioritarias las siguientes funcionalidades:

- Visualizar paradas y rutas en un mapa del Distrito Metropolitano de Quito.
- Visualizar la información de una determinada unidad de transporte, esta información implica lo siguiente:
 - Ver el interior de la unidad de transporte.
 - Ver la velocidad con la circula la unidad de transporte.
 - El nombre del conductor que se encuentra operando la unidad de transporte.
- Realizar un seguimiento de la posición geográfica de una unidad de transporte en específico.

En resumen, el aplicativo móvil va a permitir al usuario final realizar el seguimiento de una unidad de transporte en tiempo real mostrando la información del vehículo como lo es su interior, la velocidad con la circula y además la información del conductor.

Utilización de Submódulos

Los submódulos en el contexto de este proyecto son pequeños sistemas que cumplen con una funcionalidad específica, por ejemplo, en este proyecto tenemos el submódulo de empresa el cual tiene toda la lógica para gestionar una empresa como lo son la gestión edificios, establecimientos, empleados, vehículos de la empresa, etc. El objetivo de usar submódulos es el de “no volver a inventar la rueda”.

Para el proyecto utilizamos principalmente el ya mencionado submódulo de empresa y el submódulo de roles a parte de otros submódulos que se irán explicando en los “sprints” futuros. De igual importancia, estos submódulos que se utilizan en el proyecto fueron diseñados y desarrollados por el equipo de “Manticore-Labs” con el fin de ser reutilizados en todos los proyectos de la organización.

Diseño del diagrama entidad relación de la base de datos

Para el diseño lógico de la base de datos del sistema se empleó el diagrama entidad relación. Cabe recalcar que en el diagrama no se incluyeron las entidades pertenecientes a los submódulos que se agregaron al proyecto. Sin embargo, en el diagrama se puso una referencia hacia esas entidades de los otros submódulos.

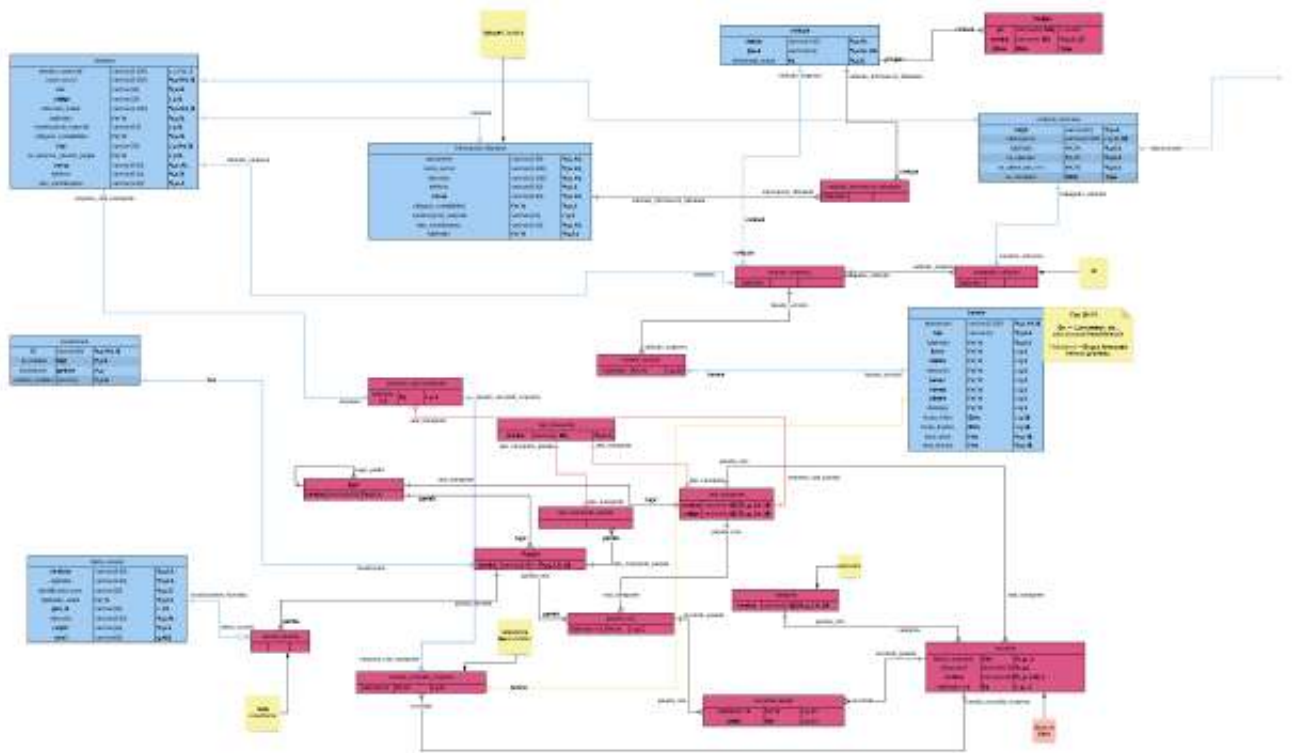


Figura 2.4: Diagrama entidad-relación de la base de datos sprint 0

Las entidades marcadas con color azul son las entidades correspondientes a los submódulos externos que se agregaron al proyecto, mientras que las entidades de color púrpura corresponden a las entidades del proyecto en desarrollo (submódulo de transportes). El diagrama de la base de datos a mayor detalle se lo encontrará en el 6.2 Anexo 2.

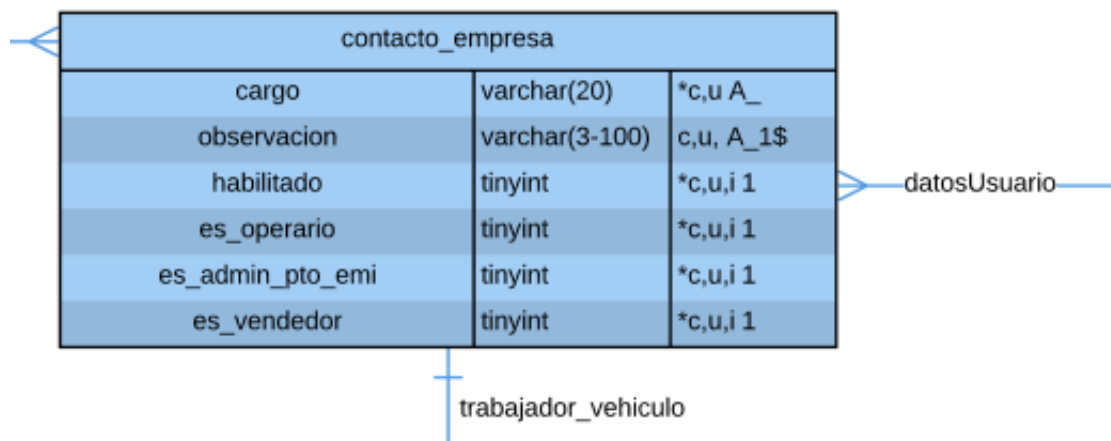


Figura 2.5: Entidad de un submódulo externo

Nomenclatura

Se estableció una nomenclatura que permite especificar las validaciones que tendrá cada atributo o campo de una entidad.

| Símbolo | Descripción |
|---------|---------------------------------------|
| *c | Campo obligatorio al crear |
| c | Campo opcional al crear |
| u | EL Campo se puede editar |
| i | El campo es índice |
| A | El campo acepta letras |
| _ | El campo acepta espacios |
| \$ | El campo acepta caracteres especiales |
| 1 | El campo acepta números |

Tabla 2.2: Nomenclatura para el diseño de la base de datos

Diseño de la arquitectura del sistema

Para el diseño de la arquitectura consideraron las siguientes capas:

- ❑ **Capa del cliente:** En esta capa consta la parte del cliente y es donde estarán las interfaces de usuario para los aplicativos tanto para la parte web y móvil. También en esta capa estarán los dispositivos IOT dado que al igual que las aplicaciones descritas anteriormente también se conectan a la nube.

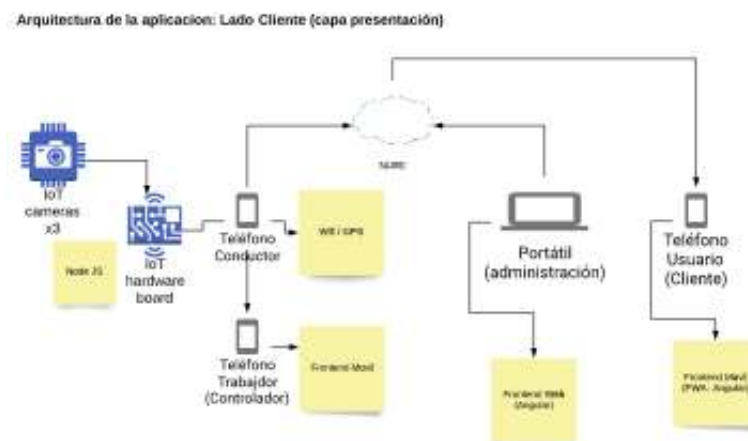


Figura 2.6: Capa del cliente

- ❑ **Capa de seguridad:** En esta capa están las políticas de seguridad que protegerán a los controladores de nuestro servidor backend de la capa del negocio.

- ❑ **Capa del negocio:** En esta capa se encuentra el servidor backend o servidor web, aquí estará toda la lógica del negocio, este servidor servirá de intermediario para acceder a los datos y consumir los servicios de otros servidores como lo son los micro-servicios de google-cloud y auth0.

- ❑ **Capa de datos:** En esta capa estarán nuestros servidores de bases de datos. El servidor de base de datos relacional (MySQL) y el servidor de base de datos no relacional (MongoDB). El servidor de bases de datos NoSQL se lo utiliza para guardar información geográfica para las paradas y recorridos.

- ❑ **Capa de IA:** Esta capa es opcional y está fuera del alcance del proyecto, por lo tanto, se la desarrollará a futuro. En esta capa funcionarán diversas tecnologías de inteligencia artificial que harán uso de la información que los usuarios del aplicativo móvil envían. Esta información es el resultado de la encuesta (2.12) que los usuarios llenan de forma opcional cuando ven la información de una unidad de transporta.

Arquitectura de la aplicación: Nube

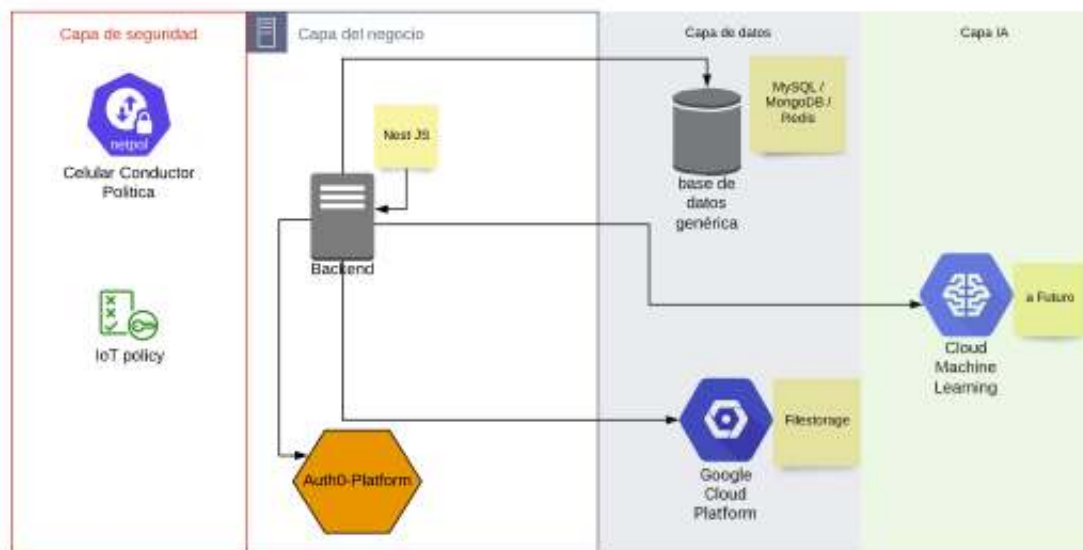


Figura 2.7: La nube del sistema

Se creó el diagrama de navegación del aplicativo web como guía para crear las pantallas de gestión del sistema. El aplicativo web tiene la finalidad de ser una herramienta para que los transportistas puedan administrar el sistema y determinar los datos que serán mostrados al usuario final en el aplicativo móvil.

El diagrama (Figura 2.8) consta de cuatro niveles, en el nivel superior se encuentran las opciones del menú principal, de las cuales a su vez nacen las pantallas hijas. Cada recuadro representa una pantalla en el aplicativo web, a excepción de los recuadros con líneas entrecortadas pues estos son modales con alguna funcionalidad relevante.

A continuación, se detalla una breve descripción del propósito de cada pantalla:

- Gestión de empresa: permite administrar las empresas registradas en el sistema.
- Asignación de rutas de transporte: asignar las rutas de transporte del sistema a una empresa.
- Gestión de horarios recorridos: establecer los horarios y los recorridos que van a tener cada una de las rutas de transporte.
- Asignación de vehículos: asignar los vehículos del sistema a una empresa.
- Asignación de trabajadores: determinar cuáles trabajadores de la empresa trabajarán en un vehículo de la empresa seleccionado.
- Gestión de horarios de servicio: gestionar los horarios de servicio disponibles para un vehículo de la empresa.
- Asignación de rutas: asignar rutas de transporte de la empresa a un vehículo de la empresa.
- Gestión de trabajadores: administrar los trabajadores de la empresa.
- Gestión de datos de contacto: manejar los datos personales de cada trabajador.
- Gestión de cargos: gestionar los cargos que tendrán los empleados de la empresa.
- Gestión de vehículos: crear y modificar los vehículos del sistema
- Gestión de dueños de vehículos: registrar los dueños de los vehículos del sistema.
- Gestión de lugares: creación de las ciudades del sistema.

- ❑ Gestión de paradas: administrar y colocar en el mapa las paradas de una ciudad.
- ❑ Asignar tipos de transporte: designar los tipos de transporte vehicular que tiene una parada.
- ❑ Gestión de rutas de transporte: administrar las rutas de transporte de una ciudad.
- ❑ Gestión de recorridos: fijar en el mapa el recorrido de una ruta.
- ❑ Asignación de paradas: determinar cuáles paradas pertenecen a una ruta o recorrido.
- ❑ Gestión de tipos de transporte: establecer los tipos de transporte que se usarán en el sistema.
- ❑ Gestión de categorías: definir las categorías que se usarán en las paradas y rutas.

La funcionalidad a mayor detalle se definirá posteriormente en historias de usuario.

Diseño de la arquitectura del sistema: Interacción con microservicios y protocolos

Para la interacción con los protocolos como MQTT y Websockets además de los servicios de GCP (Google Cloud Platform) se ha propuesto la arquitectura disponible en el 6.3 Anexo 3.

Este diagrama, muestra en detalle la interacción entre las capas de la arquitectura a través de los diferentes protocolos y servicios cuya implementación será detallada en los “sprints” posteriores.

Diseño de interfaces gráficas de usuario del aplicativo móvil

Para el diseño de interfaces de usuario para la aplicación móvil se apoyó en el uso de pestañas en la parte inferior de la pantalla (tabs) para ir a los principales módulos de la aplicación los cuales son:

- ❑ Mapa General de las paradas: En esta pantalla se presenta un mapa con todas las paradas registradas en el sistema, como se muestra en la figura 2.9.
- ❑ Menú de rutas de una parada: Esta pantalla aparece cuando el usuario selecciona una parada y entonces podrá ver las rutas que pasan por la misma y con la respectiva cooperativa a cargo de la ruta. Otra funcionalidad de esta pantalla es donde el usuario puede agregar la parada en cuestión a sus paradas favoritas.

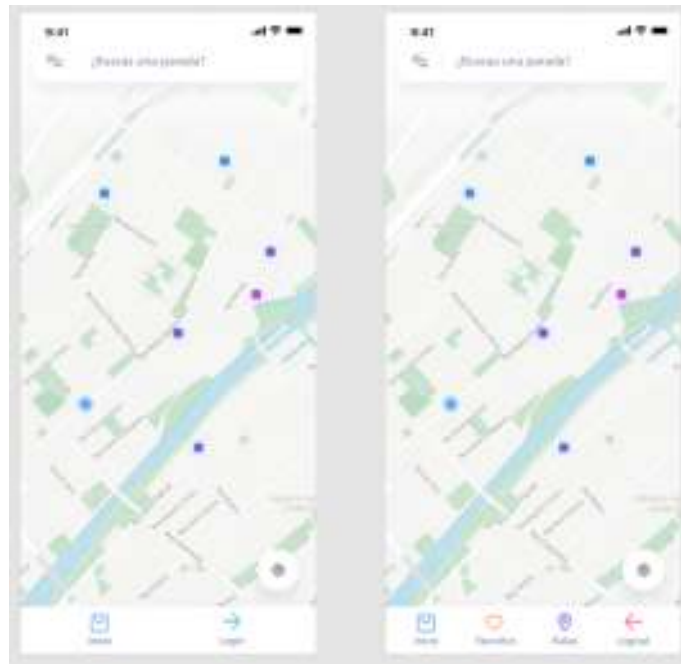


Figura 2.9: Pantalla del mapa de paradas

- ❑ Seguimiento de una ruta determinada: Esta pantalla aparece cuando el usuario selecciona una ruta y entonces podrá ver todas las unidades de transporte que circulan dentro de ella.

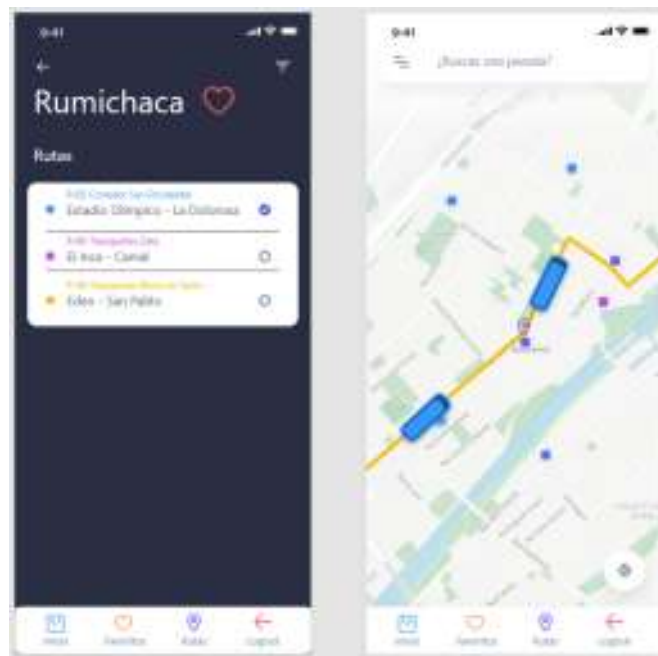


Figura 2.10: Menú de rutas y seguimiento de ruta

- ❑ Seguimiento de una unidad de transporte: Esta pantalla aparece cuando el usuario selecciona una unidad de transporte de la pantalla de Seguimiento de ruta. En ella se

podrá hacer el seguimiento individual de la unidad de transporte en cuestión.



Figura 2.11: Seguimiento de unidad de transporte

- ❑ Detalle de una unidad de transporte: Esta pantalla se muestra cuando el usuario selecciona la unidad de transporte que se haya siguiendo.



Figura 2.12: Detalle de una unidad de transporte

Como se puede ver en la figura 2.12, en la pantalla se muestra lo siguiente:

- Fotografías del interior de la unidad de transporte.

- Velocidad con la que circula la unidad de transporte.
 - Una encuesta que pregunta al usuario si la unida de transporte se halla llena de pasajeros.
- Paradas Favoritas del usuario: En esta pantalla se muestra cuando el usuario presiona el botón 'Favoritos'. Como se puede ver en la figura 2.13, la pantalla muestra las paradas que el usuario ha marcado como favoritos y a través de esta pantalla se puede realizar de manera rápida el seguimiento previamente explicado.



Figura 2.13: Paradas favoritas

Ambiente de desarrollo para el servidor web

Para levantar el ambiente de desarrollo para el 'backend', se estableció el repositorio con el nombre 'submódulo-transportes-backend' en la plataforma de gitlab como se muestra en la figura 2.14.

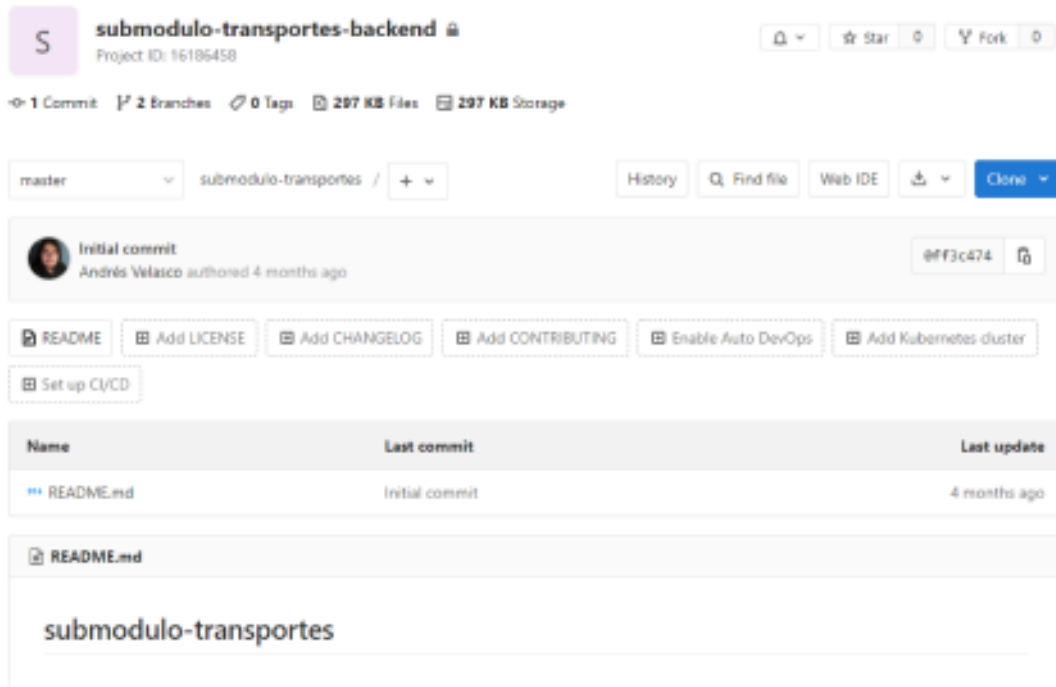


Figura 2.14: Repositorio submódulo-transportes-backend: rama Master

Posteriormente desde el proyecto base, que se muestra a continuación, se creó una rama llamada 'transportes'.

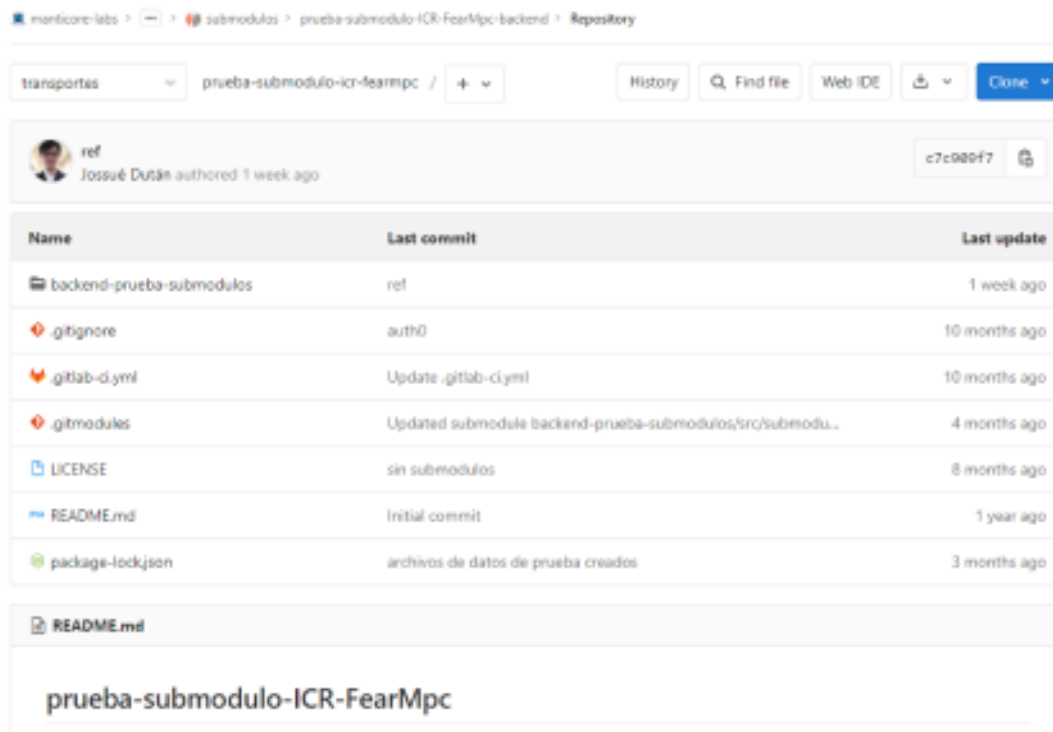


Figura 2.15: Proyecto base: backend

Dentro de este proyecto se incorporaron los siguientes submódulos:

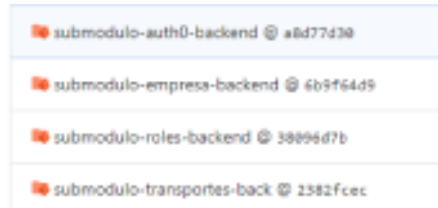


Figura 2.16: Submódulos: backend

Ambiente de desarrollo para el 'frontend'

Para levantar el ambiente de desarrollo para el 'frontend', se creó el repositorio con el nombre 'submódulo-transportes-frontend' en la plataforma de gitlab.

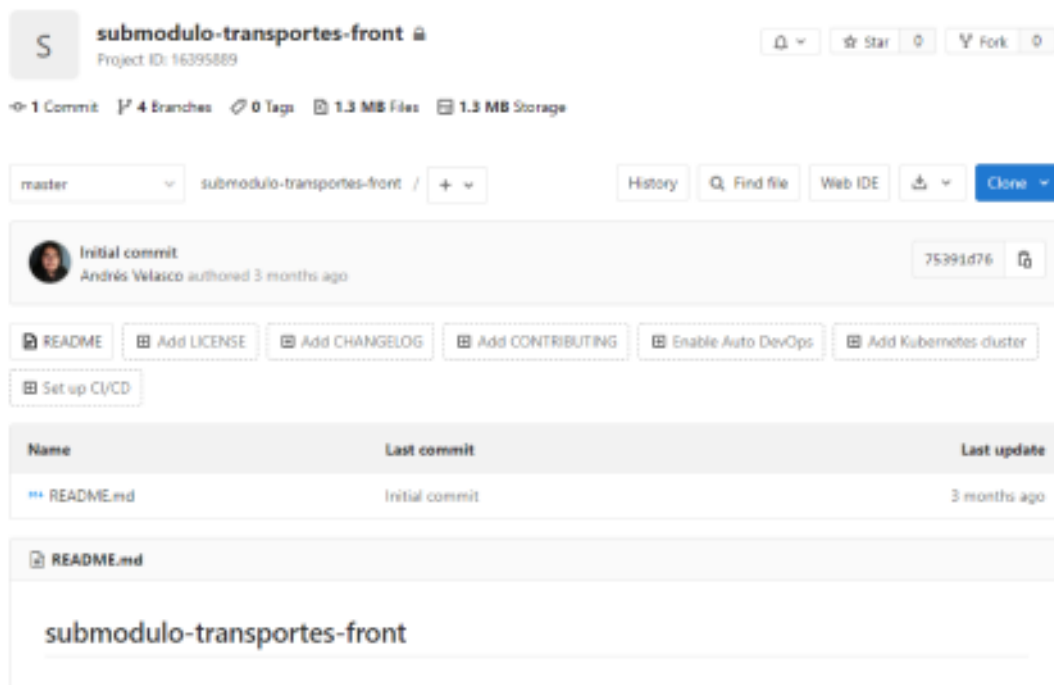


Figura 2.17: Repositorio submódulo-transportes-frontend: rama Master

Posteriormente desde el proyecto base, que se muestra a continuación, se instauró una rama llamada 'transportes'.

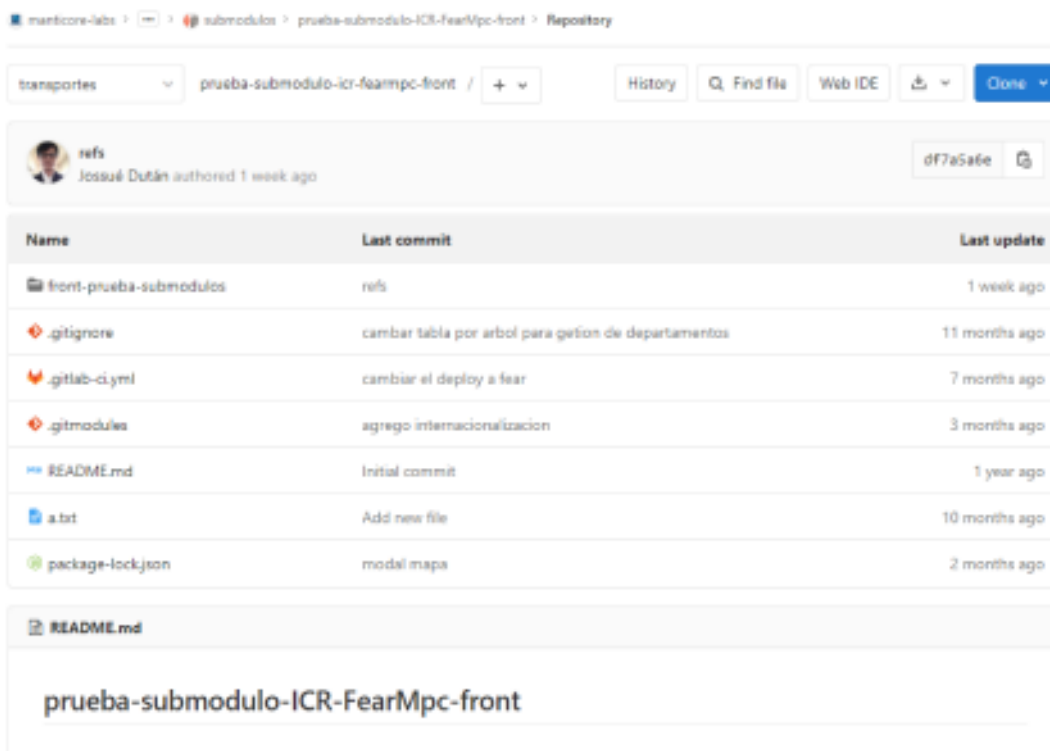


Figura 2.18: Proyecto base: frontend

Dentro de este proyecto se agregaron los siguientes submódulos:

| Name |
|---|
| - |
| submodulo-articulos-front @ d537e53 |
| submodulo-cargas-masivas-front @ 394c6e07 |
| submodulo-empresa-front @ e7d834e3 |
| submodulo-front-comun @ e8e5c7b2 |
| submodulo-internacionalizacion-front @ dbe56f84 |
| submodulo-inventario-front @ 22a4df33 |
| submodulo-menu-lateral-front @ cbfa8692 |
| submodulo-pedido-front @ 7864e05f |
| submodulo-roles-frontend @ 5509c88c |
| submodulo-transportes-front @ 11e9ccce |
| submodulo-vendedor-front @ 575e2304 |

Figura 2.19: Submódulos: frontend

Ambiente de desarrollo para la aplicación móvil

Para levantar el ambiente de desarrollo para la aplicación móvil, se hizo el repositorio con

el nombre 'transportes-movil' en la plataforma de gitlab.

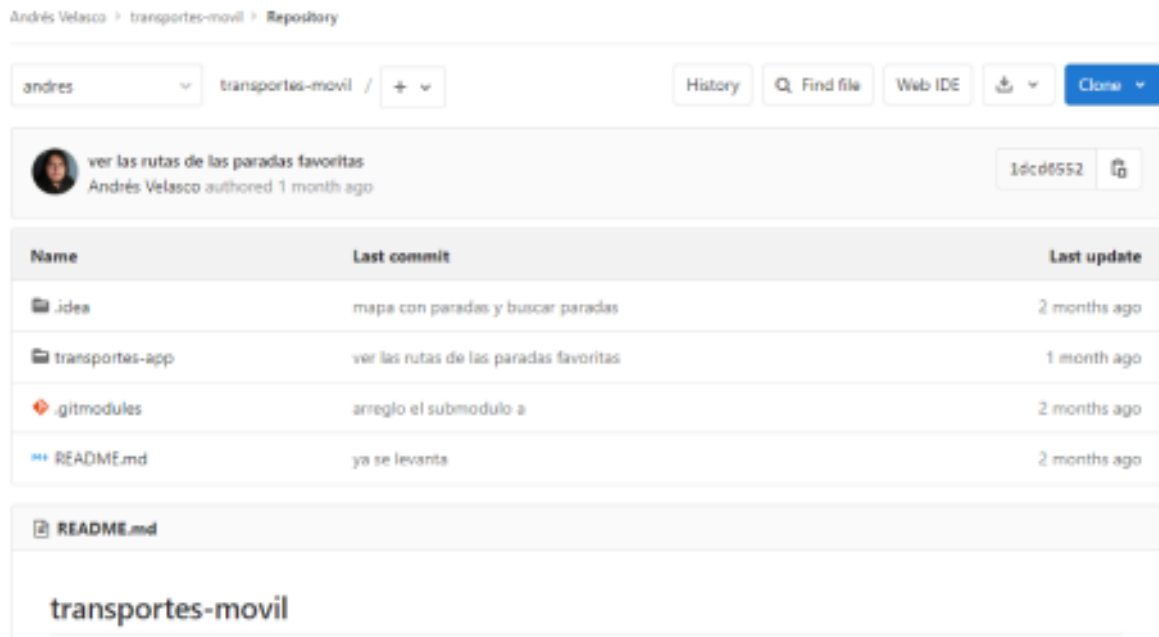


Figura 2.20: Repositorio transportes-movil

Dentro de este proyecto se anexó el submódulo de auth0-movil:



Figura 2.21: Submódulo auth0-movil

2.1.3.4 Sprint Review: 0

| Código | Descripción | Cumple |
|---------|---|--------|
| HN000 | Realizar encuesta | SI |
| HN001 | Diseño del modelo lógico de la base de datos | SI |
| HNWM002 | Diseño de interfaces gráficas de usuario (Mockups) | SI |
| HN003 | Diseño de la arquitectura del sistema | SI |
| HN004 | Levantar el ambiente de desarrollo para el servidor web (backend) | SI |

| | | |
|-------|---|----|
| HN005 | Levantar el ambiente de desarrollo para el frontend | SI |
| HN006 | Levantar el ambiente de desarrollo para el aplicativo móvil | SI |

Tabla 2.3: Sprint Review 0

2.1.3.5 Sprint Retrospective: 0

¿Qué salió bien en la iteración?

Durante la creación de los repositorios, la utilización de los submódulos de git permite que el código sea mas reutilizable, además de poder versionarlo dado que se pueden crear ramas dentro de los submódulos.

¿Qué se puede mejorar?

El diseño de la pantalla donde se puede mostrar la información detallada de la unidad de transporte debería mostrar la información del respectivo conductor. Todavía no se la integro a los diseños hasta verificar si es legalmente factible poner dicha información.

2.1.4 Sprint 1

2.1.4.1 Sprint Planning: 1

Los objetivos de este sprint son:

- Levantar el servidor web (backend).
- Crear la API REST de todas las entidades del diagrama de base datos descrito previamente.

2.1.4.2 Sprint Backlog: 1

Para el sprint 1 se consideraron las historias de usuario descritas en el 6.6 Anexo 6

2.1.4.3 Ejecución del Sprint

Dentro de este sprint se implemento el servidor web y la API REST de todas las entidades.

Este proceso abordo las siguientes actividades:

- Implementación de las entidades del diagrama entidad relación.
- Implementación de los DTO's (Mecanismos de validación de datos) para actualizar y crear registros en la base datos.
- Implementación de servicios (interacción con la capa de datos).
- Desarrollo de los controladores.
- Creación de datos de prueba.
- Pruebas de la API REST (Estas pruebas se llevaron a cabo con la herramienta POST-MAN).

La documentación detallada de este sprint se la puede encontrar en el 6.7 Anexo 7. El cual aborda, de manera más técnica, cómo se implementaron las entidades, los mecanismos de validación, los servicios y cómo se crearon los datos de prueba para las distintas entidades del diagrama entidad-relación.

2.1.4.4 Sprint Review: 1

| Código | Descripción | Cumple |
|--------|--|--------|
| HNB007 | Implementar la entidad vehículo en el Backend | SI |
| HNB008 | Implementar la entidad contacto-empresa en el Backend | SI |
| HNB009 | Implementar la entidad trabajador-vehículo en el Backend | SI |
| HNB010 | Implementar la entidad vehículo-informacion-tributaria en el Backend | SI |
| HNB011 | Implementar la entidad vehículo-empresa en el Backend | SI |
| HNB012 | Implementar la entidad informacion-tributaria en el Backend | SI |
| HNB013 | Implementar la entidad empresa en el Backend | SI |

| | | |
|--------|--|----|
| HNB014 | Implementar la entidad horario en el Backend | SI |
| HNB015 | Implementar la entidad horario-servicio en el Backend | SI |
| HNB016 | Implementar la entidad empresa-ruta-transporte en el Backend | SI |
| HNB017 | Implementar la entidad tipo-transporte en el Backend | SI |
| HNB018 | Implementar la entidad parada en el Backend | SI |
| HNB019 | Implementar la entidad localización en el Backend | SI |
| HNB020 | Implementar la entidad lugar en el Backend | SI |
| HNB021 | Implementar la entidad datos-usuario en el Backend | SI |
| HNB022 | Implementar la entidad parada-favorita en el Backend | SI |
| HNB023 | Implementar la entidad tipo-transporte-parada en el Backend | SI |
| HNB024 | Implementar la entidad ruta-transporte en el Backend | SI |
| HNB025 | Implementar la entidad horario-recorrido-empresa en el Backend | SI |
| HNB026 | Implementar la entidad parada-ruta en el Backend | SI |
| HNB027 | Implementar la entidad categoría en el Backend | SI |
| HNB028 | Implementar la entidad recorrido-parada en el Backend | SI |
| HNB029 | Implementar la entidad recorrido en el Backend | SI |

Tabla 2.4: Sprint Review: 1

2.1.4.5 Sprint Retrospective: 1

¿Qué salió bien en la iteración?

Con el uso de las bibliotecas proporcionadas por 'Manticore-labs' se pudo implementar toda la API-REST básica para el aplicativo.

¿Qué se puede mejorar?

Se puede mejorar la API-REST implementando métodos personalizados para poder listar registros con localizaciones geográficas. Por ejemplo, listar las paradas que estén dentro de un radio determinado en base a unas coordenadas proporcionadas, en definitiva, se trata de listar las paradas más cercanas al usuario.

2.1.5 Sprint 2

2.1.5.1 Sprint Planning: 2

Los objetivos de este sprint son:

- ❑ Desarrollar el 'frontend' de la aplicación de gestión web.
- ❑ Implementar en el 'frontend' la gestión de las entidades del modelo entidad relación.

2.1.5.2 Sprint Backlog: 2

Para el sprint 2 se consideraron las historias de usuario descritas en el 6.8 Anexo 8

2.1.5.3 Ejecución del Sprint

En total se realizaron 14 gestiones correspondientes a cada una de las historias de usuario planificadas para este sprint. A continuación se describirán las gestiones más importantes.

Gestión de paradas

Se implementó la ruta en el frontend para poder gestionar paradas para un determinado lugar como se evidencia en los gráficos siguientes:



Figura 2.22: Ruta de Gestión de Paradas

La funcionalidad de asignar tipo de transporte permite agregar tipos de transporte (alimentador, bus-tipo, etc.) a la parada. Dado que suelen existir paradas por donde se detienen unidades de transporte del tipo alimentador, bus-tipo, entre otros.

22 de mayo de 2020 21:52

PERFIL

LOGOUT

Menu principal / Gestion Lugar / Gestion Paradas

CREAR PARADA

Llene los campos necesarios.

Nombre * Localizacion *

Ingrese el nombre del lugar. Seleccione desde el mapa la localizacion de la parada.

TERMINAL PLAYON MARIN S-N

EDITAR

ASIGNAR TIPO DE TRANSPORTE

VER TIPOS DE TRANSPORTE

DESHABILITAR

Figura 2.23: Formulario para crear una parada

Gestión de rutas de transporte

Se implementó la pantalla en el aplicativo web para poder gestionar las rutas de transporte para un determinado lugar (ciudad). Esta ruta permite al usuario seleccionar una ruta y gestionar sus paradas y gestionar sus recorridos, como se evidencia en el gráfico siguiente:

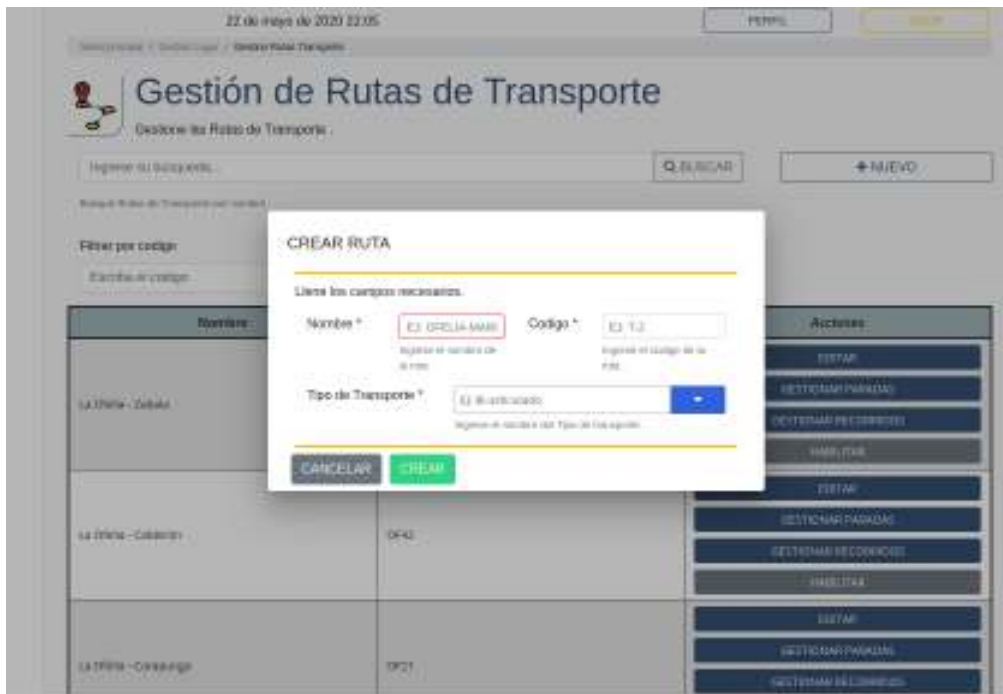


Figura 2.24: Formulario para crear una ruta de transporte

Gestión de paradas para una ruta de transporte

Se implementó la ruta en el frontend para poder gestionar las paradas para una ruta de transporte como se evidencia en el gráfico siguiente:



Figura 2.25: Asignar paradas para una ruta de transporte

Gestión de recorridos

Para la aplicación web se desarrolló la pantalla para gestionar los recorridos de una ruta de transporte. Cabe recalcar que los recorridos son las diferentes trayectorias que puede tener una ruta de transporte tanto para viajes de ida como de vuelta. La funcionalidad de esta pantalla se la puede evidenciar en los siguientes gráficos.



Figura 2.26: Formulario de creación de un recorrido

Para cada recorrido es necesario dibujar la trayectoria del recorrido, para ello se traza la misma en un mapa.



Figura 2.27: Editar la trayectoria (Información geográfica) de un recorrido

Gestión de paradas para un recorrido

No todas las paradas de una ruta de transporte van a estar en la trayectoria de un recorrido, por lo tanto, las paradas que se encontrarán incluidas en esta trayectoria tienen que ser asignadas manualmente al recorrido, razón por la cual se desarrolló la ruta de gestión de paradas para un recorrido. Esta funcionalidad se evidencia en el siguiente gráfico.



Figura 2.28: Asignación de paradas para un recorrido

Es necesario aclarar que las paradas que se le muestran al usuario para asignar son paradas que fueron agregadas a la ruta de transporte a la que pertenece el recorrido en cuestión, también solo se listan paradas de la ruta que están habilitadas.

2.1.5.4 Sprint Review: 2

| Código | Descripción | Cumple |
|--------|---|--------|
| HFW030 | Como usuario administrador, yo puedo gestionar empresas | SI |

| | | |
|--------|--|----|
| HFW031 | Como usuario administrador, yo puedo gestionar vehículos | SI |
| HFW032 | Como usuario administrador, yo puedo asignar vehículos a una determinada empresa | SI |
| HFW033 | Como usuario administrador, yo puedo asignar dueños a los vehículos | SI |
| HFW034 | Como usuario administrador, yo puedo gestionar horarios para los Recorridos | SI |
| HFW035 | Como usuario administrador, yo puedo gestionar horarios para los Vehículos | SI |
| HFW036 | Como usuario administrador, yo puedo gestionar lugares | SI |
| HFW037 | Como usuario administrador, yo puedo gestionar paradas para un lugar determinado | SI |
| HFW038 | Como usuario administrador, yo puedo gestionar rutas de transporte | SI |
| HFW039 | Como usuario administrador, yo puedo gestionar recorridos para una ruta de transporte | SI |
| HFW040 | Como usuario administrador, yo puedo gestionar Paradas-Ruta de un recorrido específico | SI |
| HFW041 | Como usuario administrador, yo puedo gestionar paradas de una ruta de transporte | SI |
| HFW042 | Como usuario administrador, yo puedo gestionar categorías | NO |
| HFW043 | Como usuario administrador, yo puedo gestionar tipos de transporte | NO |

Tabla 2.5: Sprint Review 2

2.1.5.5 Sprint Retrospective: 2

¿Qué salió bien en la iteración?

Durante el desarrollo de este sprint se pudo implementar con éxito los componentes para dibujar la trayectoria de un recorrido en un mapa al igual de poder registrar y editar la localización de una parada desde la selección de su ubicación geográfica a través de un mapa dentro del formulario.

¿Qué se puede mejorar?

La asignación de tipos de transporte para una parada solamente asigna una parada a la vez, lo que puede hacer al respecto es reemplazar el componente actual por un componente más sofisticado como el componente que se utiliza para asignar paradas a un recorrido. También se encontraron unos defectos visuales en la gestión de categorías y tipos de transporte razón por la cual en el sprint review anterior se los etiqueto como no cumplido.

2.1.6 Sprint 3

2.1.6.1 Sprint Planning: 3

Los objetivos de este sprint son:

- Desarrollar el aplicativo móvil.

- Implementar el programa que manejará la lógica del dispositivo IOT (“Raspberry PI”).

2.1.6.2 Sprint Backlog: 3

Para el sprint 3 se consideraron las historias de usuario descritas en el 6.9 Anexo 9

2.1.6.3 Ejecución del Sprint

Lo que se detalla a continuación corresponde al flujo del usuario final en el aplicativo móvil, desde el registro e ingreso al sistema del usuario, hasta el seguimiento detallado de una unidad de transporte.

Registro de usuarios e ingreso al sistema

Se implementó con éxito los flujos para permitir a los usuarios registrarse en el sistema e ingresar al mismo.



Figura 2.29: Registro de usuario

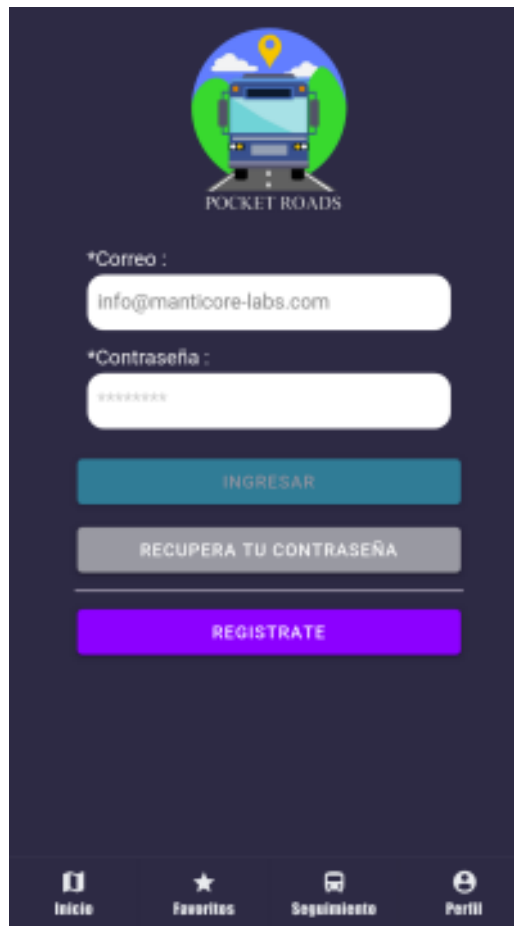


Figura 2.30: Ingreso al sistema

Visualizar paradas y consultar rutas

Se desarrolló la funcionalidad con el objetivo de buscar paradas a través del mapa mediante el uso de la librería de mapas geográfico llamado "open-layers". Al seleccionar una parada, se muestran todas las rutas de transporte que pasan por la misma.

La lógica detrás de la funcionalidad consiste primero en recuperar todas las paradas habilitadas en el sistema en un rango de 5 kilómetros desde la posición geográfica del usuario, si existe algún problema en recuperar la ubicación del usuario se tomara en cuenta las coordenadas geográficas de la parada "El Trébol" por ser una parada y ubicación geográfica central.

Cuando dichas paradas son recuperadas se consultan sus ubicaciones geográficas almacenadas en la base de datos de MongoDB y se combinan respectivamente los datos geográficos de la parada con su información. Cabe recalcar que este procedimiento se lo realiza en el servidor web a través de una determina petición HTTP. Esta información es renderizada

en el mapa. Posteriormente a través de las funcionalidades de “open-layers” cada parada dibujada en el mapa emitirá su información al ser seleccionada y este evento es escuchado y se muestra la pantalla de “rutas que pasan por la parada” y se consultan todas las rutas que tengan la parada emitida.

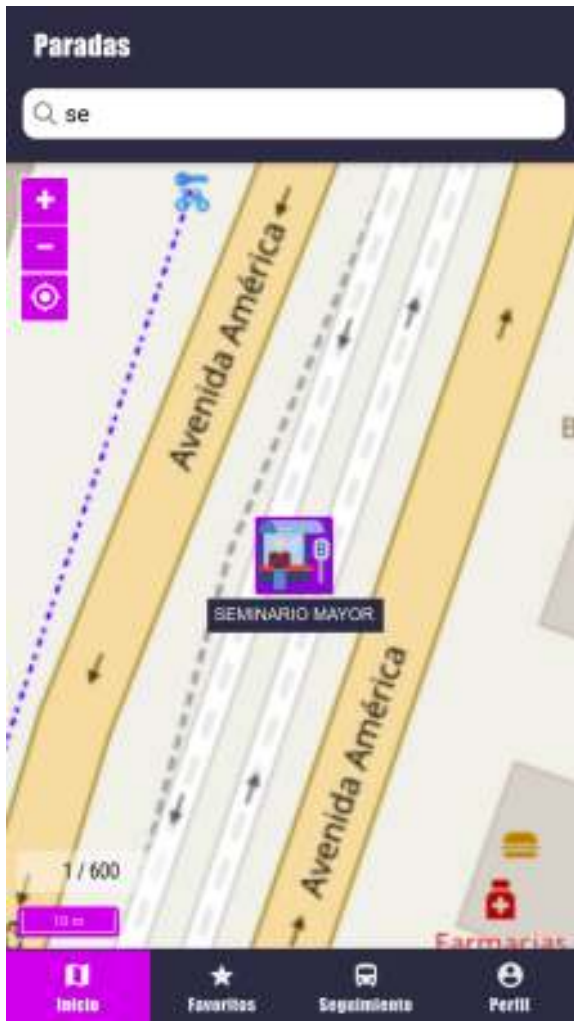


Figura 2.31: Mapa de paradas

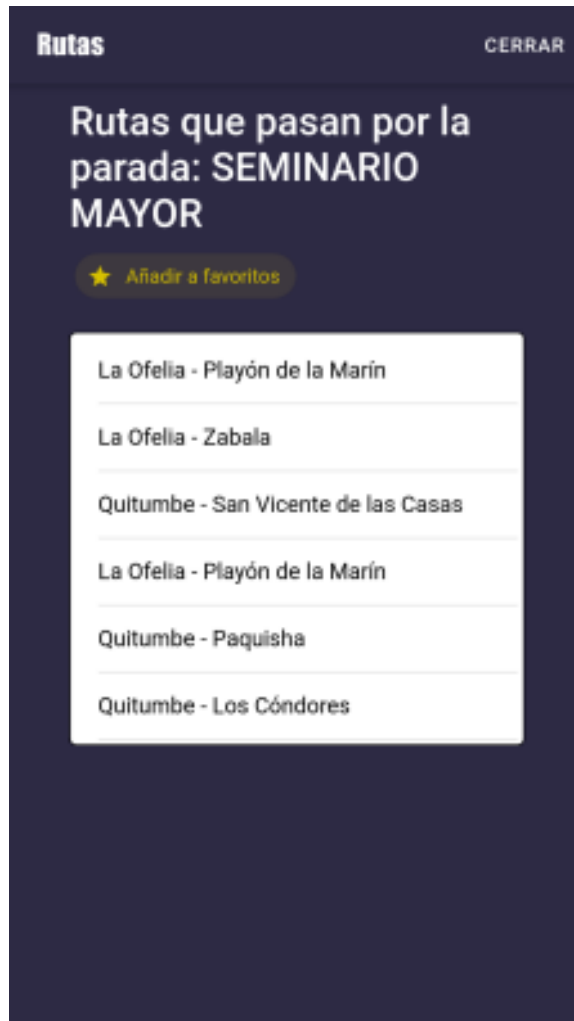


Figura 2.32: Rutas que pasan por la parada

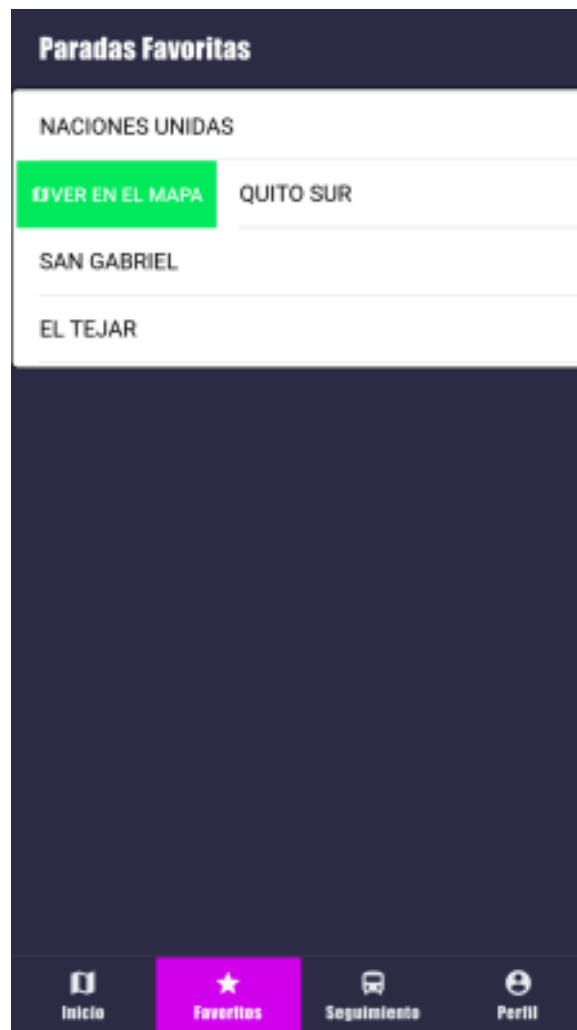


Figura 2.33: Paradas favoritas del usuario

Gestionar paradas favoritas

Una vez que el usuario inicie sesión en el sistema se cargaran las paradas que seleccionó como favoritas en la pantalla de “paradas-favoritas”. En dicha pantalla las paradas favoritas cuentan con las acciones de ubicar la parada en el mapa y eliminar la parada de sus favoritas.

Seguimiento de la ruta

Cuando el usuario selecciona una ruta, se consultan todos los recorridos que tenga esa ruta que estén disponibles según la fecha y hora en que el usuario haga la consulta. Los recorridos consultados se muestran en un menú. Al seleccionar un recorrido, este es dibujado en el mapa y posteriormente se mostrará en el mismo mapa la posición de cada unidad de transporte asignado a dicho recorrido.

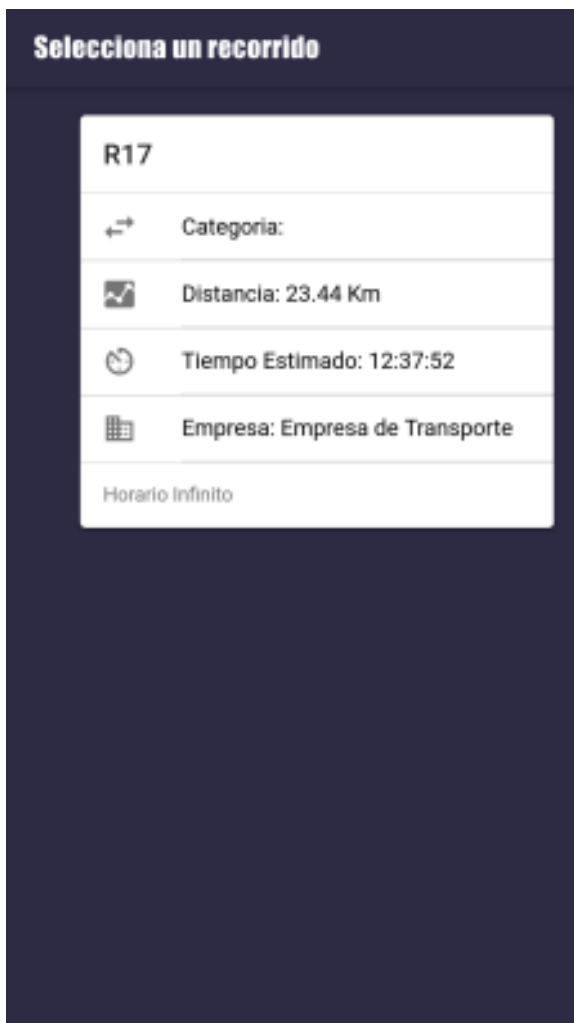


Figura 2.34: Menú de recorridos



Figura 2.35: Seguimiento de la ruta

Para obtener la ubicación en tiempo real de unidad de transporte, se implementó la geolocalización en segundo plano a través del propio aplicativo móvil, cabe aclarar que esta funcionalidad es exclusiva para usuarios con el rol de trabajador.

Ver información de unidad de transporte

Dentro del mapa de seguimiento de la ruta al seleccionar una unidad de transporte se muestra una pantalla en donde se puede observar toda la información de dicha unidad como lo son las fotografías del interior del vehículo con el fin de comprobar si dicho medio de transporte está lleno de pasajeros o no. También se muestra la velocidad con la que circula dicha unidad de transporte una pequeña información del conductor del vehículo.



Figura 2.36: Información de la unidad de transporte

Las fotografías del interior del vehículo son tomadas desde el dispositivo "IOT" (Raspberry PI). Este dispositivo toma las fotografías dentro de un intervalo de tiempo determinado y las codifica en "base64" para ser enviadas a través del protocolo "MQTT" al servidor. El servidor recibe las imágenes y las descodifica. Posteriormente las imágenes son guardadas en "Google Cloud Storage" en un "bucket" determinado y el enlace de cada imagen guardada es almacenado la base datos.

2.1.6.4 Sprint Review: 3

| Código | Descripción | Cumple |
|---------------|--|---------------|
| HFM044 | Como usuario, yo puedo registrarme en el sistema | SI |
| HFM045 | Como usuario registrado, yo puedo iniciar y cerrar sesión | SI |
| HFM046 | Como usuario, yo puedo visualizar paradas en un mapa | SI |
| HFM047 | Como usuario, yo puedo buscar paradas | SI |
| HFM048 | Como usuario, yo puedo ver las rutas de una parada determinada | SI |
| HFM049 | Como usuario, yo puedo agregar una parada a mis paradas favoritas | SI |
| HNIOT051 | Permitir que el "Raspberry PI" tome fotografías y las envíe al servidor web central en un periodo de tiempo determinado | SI |
| HFM052 | Como usuario registrado, yo puedo visualizar en un mapa las unidades de transporte de una ruta de transporte en específico | SI |
| HFM053 | Como usuario registrado, yo puedo visualizar la información de una determinada unidad de transporte | SI |
| HFM054 | Como usuario trabajador registrado, yo puedo iniciar un recorrido | SI |

Tabla 2.6: Sprint Review 3

2.1.6.5 Sprint Retrospective: 3

¿Qué salió bien en la iteración?

Durante el desarrollo de este sprint se pudo implementar con éxito todos los flujos para determinar si una determinada unidad de transporte está en su aforo máximo de pasajeros a través de la visualización de las fotografías del interior del vehículo respectivo.

¿Qué se puede mejorar?

La pantalla de información de una unidad de transporte se puede mostrar también una fotografía del conductor del vehículo respectivo. También se puede mejorar la calidad de las fotografías e incluso incluir una marca de agua en las mismas y censurar los rostros de las personas que aparezcan en dichas imágenes.

3 RESULTADOS Y DISCUSIÓN

3.1 PRUEBAS DE USABILIDAD

La usabilidad es la propiedad que refleja la sencillez con la que se usa el sistema. [27]

Para evaluar la usabilidad del aplicativo móvil se usó el cuestionario de Sistema de Escalas de Usabilidad o SUS (System Usability Scale) por sus siglas en inglés. SUS es una herramienta confiable y rápida para medir la usabilidad. Consiste en un cuestionario de 10 ítems con cinco opciones de respuesta en escala de muy de acuerdo a muy en desacuerdo y permite evaluar una amplia variedad de productos y servicios, incluidos sitios web y aplicaciones. [28]

3.1.1 El cuestionario

Los 10 ítems que conforman el cuestionario SUS y se les pide a los participantes que califiquen con una de cinco respuestas se muestran en la Tabla 3.1.

3.1.2 La puntuación

Para medir los puntos obtenidos con el cuestionario se toman las siguientes consideraciones[29]:

- Para los elementos impares restar uno de la respuesta del usuario
- Para los elementos pares restar de 5 el valor de la respuesta del usuario
- Sumar las respuestas convertidas de cada usuario y multiplicar el total de esa suma por 2.5, de modo que se convierte de una escala de 0 a 40 a una escala de 0 a 100

| ítem | Totalmente desacuer- do | | | | Totalmente de acuerdo |
|--|-------------------------------|---|---|---|--------------------------|
| 1. Creo que me gustaría usar esta aplicación con frecuencia. | 1 | 2 | 3 | 4 | 5 |
| 2. Encontré la aplicación innecesariamente compleja. | 1 | 2 | 3 | 4 | 5 |
| 3. Considero que la aplicación es fácil de usar. | 1 | 2 | 3 | 4 | 5 |
| 4. Creo que necesitaría el apoyo de un técnico para usar esta aplicación. | 1 | 2 | 3 | 4 | 5 |
| 5. Encontré que las funciones de esta aplicación están bien integradas. | 1 | 2 | 3 | 4 | 5 |
| 6. Pienso que había demasiada inconsistencia en esta aplicación. | 1 | 2 | 3 | 4 | 5 |
| 7. Imagino que la mayoría de personas aprendería rápidamente a usar esta aplicación. | 1 | 2 | 3 | 4 | 5 |
| 8. Encuentro que la aplicación es muy compleja de usar. | 1 | 2 | 3 | 4 | 5 |
| 9. Me sentí muy seguro/a al usar la aplicación. | 1 | 2 | 3 | 4 | 5 |
| 10. Necesité aprender muchas cosas antes de ser capaz de usar esta aplicación. | 1 | 2 | 3 | 4 | 5 |

Tabla 3.1: Ítems del cuestionario SUS

Un puntaje de SUS por encima de 68 se puede considerar superior al promedio y cualquier valor por debajo de 68 está por debajo del promedio [29].

3.1.3 Resultados

Por motivos de confinamiento el cuestionario se lo hizo en línea por medio de la plataforma Google Meet. Se presentó el aplicativo móvil a un grupo de estudiantes y profesionales, quienes a su vez pudieron interactuar con el aplicativo y evaluaron su usabilidad con el cuestionario. Las respuestas se muestran en la Tabla 3.2.

| Profesión | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | Resultado |
|----------------------------|--------|------|------|--------|--------|--------|--------|------|--------|--------|-----------|
| Ingeniero de sistemas 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 5 | 90 |
| Desarrollador 1 | 4 | 1 | 5 | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 90 |
| Estudiante de ingeniería 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 97.5 |
| Estudiante de ingeniería 2 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 2 | 5 | 1 | 100 |
| Ingeniero comercial | 5 | 4 | 4 | 4 | 3 | 2 | 3 | 2 | 4 | 5 | 55 |
| Ingeniero de sistemas 2 | 5 | 1 | 5 | 2 | 4 | 1 | 5 | 1 | 5 | 1 | 95 |
| Ingeniero de sistemas 3 | 5 | 1 | 5 | 1 | 5 | 1 | 4 | 1 | 5 | 2 | 95 |
| Estudiante de ingeniería 3 | 5 | 3 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 95 |
| Desarrollador 2 | 5 | 2 | 4 | 2 | 5 | 1 | 4 | 2 | 4 | 2 | 82.5 |
| Estudiante de artes | 5 | 1 | 5 | 1 | 5 | 4 | 5 | 1 | 5 | 1 | 92.5 |
| Diseñadora gráfica | 5 | 3 | 4 | 5 | 5 | 1 | 5 | 1 | 5 | 3 | 77.5 |
| Trabajadora social | 4 | 2 | 5 | 3 | 5 | 1 | 5 | 1 | 4 | 3 | 82.5 |
| Promedio | 4.8333 | 1.75 | 4.75 | 1.9167 | 4.6667 | 1.3333 | 4.5833 | 1.25 | 4.6667 | 2.1667 | 87.7083 |

Tabla 3.2: Resultados del cuestionario SUS

En la columna de resultado se observa la puntuación SUS obtenida por cada encuestado, como resultado promedio se obtuvo una puntuación de usabilidad de 87.7083/100. En palabras del PhD. Jeff Sauro [29], una puntuación obtenida superior a 80,3 se encuentra en el 10 % superior de las puntuaciones. También, es la zona en la cual es más probable que los usuarios recomienden el producto a un amigo.

Como se observa en la Tabla 3.2, en promedio las preguntas positivas (pares) se mantuvieron con una puntuación mayor a 4 y las preguntas negativas (impares) se mantuvieron con una puntuación menor a 2, a excepción de la última pregunta que indica que los usuarios necesitan tener un conocimiento previo antes de usar la aplicación. Sin embargo, en las demás preguntas y sobre todo en las preguntas relacionadas con la facilidad de uso del

aplicativo se obtuvieron excelentes resultados (un puntaje que difiere en 0.25 del puntaje perfecto).

3.2 PRUEBAS DE FUNCIONALIDAD

Para evaluar la funcionalidad del aplicativo móvil se realizaron las pruebas en base a las historias de usuario realizadas en el Sprint 3.

Se eligió el diseño de los casos de prueba para probar la funcionalidad del sistema dado que permiten validar y verificar las historias de usuario y por otro lado presentar defectos que existen en el sistema. [27]

3.2.1 Resultados

A continuación, se muestran los casos de prueba con sus respectivos resultados:

| Caso de Prueba 1 | |
|-------------------------|---|
| Funcionalidad | Registro de un usuario en el sistema |
| Entradas | <ul style="list-style-type: none"><input type="checkbox"/> Nombres (Campo Alfabético, Obligatorio)<input type="checkbox"/> Apellidos (Campo Alfabético, Obligatorio)<input type="checkbox"/> Correo (Campo Alfanumérico, Obligatorio)<input type="checkbox"/> Identificación (Campo Numérico, Obligatorio)<input type="checkbox"/> Contraseña (Obligatorio) |
| Antecedente | Ninguno |
| Acciones | <ol style="list-style-type: none">1. Ir a la pantalla de inicio de sesión.2. Dar clic en el botón de REGÍSTRATE3. Ingresar todos los campos obligatorios del formulario.4. Dar clic en el botón de REGISTRARSE |

| | |
|---------------------------|---------------------------------------|
| Roles | Usuario final del aplicativo |
| Resultado esperado | Mensaje de éxito y usuario registrado |
| Resultado obtenido | Mensaje de éxito y usuario registrado |

Tabla 3.3: Caso de prueba para registro de un usuario

| Caso de Prueba 2 | |
|---------------------------|--|
| Funcionalidad | Iniciar sesión |
| Entradas | <input type="checkbox"/> Correo (Obligatorio) <input type="checkbox"/> Contraseña (Obligatorio) |
| Antecedente | Estar registrado con éxito en el aplicativo |
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de inicio de sesión. 2. Ingresar todos los campos obligatorios del formulario. 3. Dar clic en el botón de INGRESAR |
| Roles | Usuario final del aplicativo |
| Resultado esperado | Mensaje de éxito y navegación a la pestaña de inicio |
| Resultado obtenido | Mensaje de éxito y navegación a la pestaña de inicio |

Tabla 3.4: Caso de prueba de inicio de sesión

| Caso de Prueba 3 | |
|-------------------------|--|
| Funcionalidad | Cerrar sesión |
| Entradas | <input type="checkbox"/> Correo (Obligatorio) <input type="checkbox"/> Contraseña (Obligatorio) |
| Antecedente | Haber ingresado con éxito en el aplicativo |

| | |
|---------------------------|--|
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de perfil. 2. Dar clic en el botón de CERRAR SESIÓN |
| Roles | Usuario final del aplicativo |
| Resultado esperado | Mensaje de éxito y navegación a la pestaña de inicio |
| Resultado obtenido | Mensaje de éxito y navegación a la pestaña de inicio |

Tabla 3.5: Caso de prueba de cerrar sesión

| Caso de Prueba 4 | |
|---------------------------|--|
| Funcionalidad | Visualizar paradas en el mapa de inicio |
| Entradas | Ninguna |
| Antecedente | Haber ingresado con éxito en el aplicativo |
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de inicio. |
| Roles | Usuario final del aplicativo |
| Resultado esperado | Las paradas se presentan con un icono y un nombre en el mapa de inicio |
| Resultado obtenido | Se muestran paradas con un icono y un nombre en el mapa de inicio |

Tabla 3.6: Caso de prueba de visualizar paradas en un mapa

| Caso de Prueba 5 | |
|-------------------------|--|
| Funcionalidad | Buscar paradas |
| Entradas | Nombre de la parada a buscar |
| Antecedente | Haber ingresado con éxito en el aplicativo |

| | |
|---------------------------|---|
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de inicio. 2. Dar clic en la barra de búsqueda que dice “busca una parada”. 3. Ingresar el nombre de la parada a buscar, ejemplo: “el trebol”. 4. Dar clic en el nombre de la parada de la lista que aparecerá debajo de la barra de búsqueda. |
| Roles | Usuario final del aplicativo |
| Resultado esperado | Se muestra la parada seleccionada en el mapa |
| Resultado obtenido | Carga un momento el mapa y luego la parada seleccionada se muestra en el mapa. |

Tabla 3.7: Caso de prueba de buscar paradas

| Caso de Prueba 6 | |
|---------------------------|--|
| Funcionalidad | Listar rutas de una parada |
| Entradas | Ninguna |
| Antecedente | Haber ingresado con éxito en el aplicativo |
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de inicio. 2. Dar clic en una parada. |
| Roles | Usuario final del aplicativo |
| Resultado esperado | Se muestra una lista de rutas que pasan por esa parada |
| Resultado obtenido | Se muestra una ventana con una lista de rutas de la parada |

Tabla 3.8: Caso de prueba de listar rutas de una parada

| Caso de Prueba 7 | |
|-------------------------|--|
| Funcionalidad | Agregar una parada a favoritos |
| Entradas | Ninguna |
| Antecedente | Haber ingresado con éxito en el aplicativo |

| | |
|---------------------------|---|
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de inicio. 2. Dar clic en una parada. 3. Dar clic en el botón de “Añadir a favoritos” 4. Cerrar la ventana 5. Ir a la pestaña Favoritos |
| Roles | Usuario final del aplicativo |
| Resultado esperado | Se muestra unas paradas, las cuales son las que el usuario debe haberlas guardado como favoritas. |
| Resultado obtenido | Se muestra la parada seleccionada en la lista de favoritos |

Tabla 3.9: Caso de prueba de agregar una parada a favoritos

| Caso de Prueba 8 | |
|---------------------------|--|
| Funcionalidad | Visualizar una parada favorita en el mapa |
| Entradas | Ninguna |
| Antecedentes | <ol style="list-style-type: none"> 1. Haber ingresado con éxito en el aplicativo. 2. Haber guardado al menos una parada como favorita. |
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de favoritos. 2. Buscar la parada favorita. 3. Arrastrar la parada favorita de izquierda a derecha. 4. Hacer clic en el botón “Ver mapa”. |
| Roles | Usuario final del aplicativo |
| Resultado esperado | En la pantalla de inicio se debe visualizar la parada favorita en el mapa. |
| Resultado obtenido | Se muestra la parada favorita en el mapa. |

Tabla 3.10: Caso de prueba de visualizar una parada favorita en el mapa

| Caso de Prueba 9 | |
|---------------------------|--|
| Funcionalidad | Visualizar los recorridos de una ruta. |
| Entradas | Ninguna |
| Antecedentes | <ol style="list-style-type: none"> 1. Haber ingresado con éxito en el aplicativo. |
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de inicio. 2. Buscar y seleccionar una parada. 3. Seleccionar una ruta de que pasa por la parada. |
| Roles | Usuario final del aplicativo |
| Resultado esperado | Se debe visualizar una lista de los recorridos que pertenecen a esa ruta. Además, el horario de los recorridos debe estar dentro del rango con la fecha y hora en la que se realiza la prueba. |
| Resultado obtenido | Se muestra la lista de los recorridos de la ruta conforme al horario actual. |

Tabla 3.11: Caso de prueba de visualizar los recorridos de una ruta

| Caso de Prueba 10 | |
|--------------------------|--|
| Funcionalidad | Realizar el seguimiento de una ruta. |
| Entradas | Ninguna |
| Antecedentes | <ol style="list-style-type: none"> 1. Haber ingresado con éxito en el aplicativo. |

| | |
|---------------------------|--|
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de inicio. 2. Buscar y seleccionar una parada. Ejemplo: Seminario Mayor 3. Seleccionar una ruta de que pasa por la parada. Ejemplo: La Ofelia - Playón de la Marín 4. Seleccionar un recorrido de la ruta. Ejemplo: R17 |
| Roles | Usuario final del aplicativo |
| Resultado esperado | Se debe visualizar un mapa en donde se muestra dibujado la trayectoria del recorrido y se debe mostrar también en el mapa la ubicación en tiempo real de las unidades de transporte asignados a ese recorrido. |
| Resultado obtenido | Se muestra dibujado en el mapa el recorrido de la ruta y se muestra en tiempo real la posición de una unidad de transporte asignada al recorrido en cuestión. Nota: Solo existe una unidad de transporte asignada para este recorrido. |

Tabla 3.12: Caso de prueba de realizar el seguimiento de una ruta

| Caso de Prueba 11 | |
|--------------------------|--|
| Funcionalidad | Ver la información de una unidad de transporte. |
| Entradas | Ninguna |
| Antecedentes | <ol style="list-style-type: none"> 1. Haber ingresado con éxito en el aplicativo. |

| | |
|----------------------------------|--|
| <p>Acciones</p> | <ol style="list-style-type: none"> 1. Ir a la pantalla de inicio. 2. Buscar y seleccionar una parada. Ejemplo: Seminario Mayor 3. Seleccionar una ruta de que pasa por la parada. Ejemplo: La Ofelia - Playón de la Marín 4. Seleccionar un recorrido de la ruta. Ejemplo: R17 5. Seleccionar una unidad de transporte dentro del mapa. |
| <p>Roles</p> | <p>Usuario final del aplicativo</p> |
| <p>Resultado esperado</p> | <p>Se debe visualizar lo siguiente:</p> <ol style="list-style-type: none"> 1. Las fotografías del interior de la unidad de transporte. 2. La velocidad con la que circula el vehículo en cuestión. 3. La información del conductor: <ul style="list-style-type: none"> <input type="checkbox"/> Nombres <input type="checkbox"/> Apellidos <input type="checkbox"/> Identificación <input type="checkbox"/> Correo electrónico |
| <p>Resultado obtenido</p> | <p>Se muestran fotografías del interior de un autobús, también se visualiza la velocidad actual de 38 Km/h y la siguiente información del conductor:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Nombres: Carolina Gabriela <input type="checkbox"/> Apellidos: Bohorquez Espín <input type="checkbox"/> Identificación: 1720655149 <input type="checkbox"/> Correo electrónico: carolina.bohorquez@gmail.com |

Tabla 3.13: Caso de prueba ver la información de una unidad de transporte

| Caso de Prueba 12 | |
|---------------------------|--|
| Funcionalidad | Iniciar un recorrido. |
| Entradas | <input type="checkbox"/> Vehículo: TCQ-8036 - Tesla <input type="checkbox"/> Ruta: C2 - La Ofelia - Playón de la Marín <input type="checkbox"/> Recorrido y Horario: R17 Horario Infinito 01:00:00 24:59:00 |
| Antecedentes | <ol style="list-style-type: none"> 1. Haber ingresado con éxito en el aplicativo y poseer el cargo de transportista, ayudante transportista o administrador. |
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pestaña de perfil. 2. Clic en el botón de "Iniciar Recorrido". 3. Seleccionar las entradas en el formulario. 4. Clic en el botón "Iniciar Recorrido". 5. Hacer seguimiento de la parada Seminario Mayor, ruta La Ofelia - Playón de la Marín y recorrido R17 (como se describe en el caso de prueba 11). 6. Moverse con el dispositivo móvil por lo menos 10 metros del punto de origen. |
| Roles | Usuario transportista, ayudante transportista o administrador. |
| Resultado esperado | Mensaje de éxito y mostrar la ubicación actual del dispositivo móvil en el mapa con un icono de un autobús. |
| Resultado obtenido | Se muestra un mensaje de éxito y se mostró un icono de un bus en el mapa en la ubicación actual del dispositivo móvil. |

Tabla 3.14: Caso de prueba iniciar un recorrido

| Caso de Prueba 13 | |
|---------------------------|---|
| Funcionalidad | Quitar una parada de favoritos. |
| Entradas | Ninguna |
| Antecedentes | <ol style="list-style-type: none"> 1. Haber ingresado con éxito en el aplicativo. 2. Haber guardado al menos una parada como favorita. |
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de favoritos. 2. Arrastrar una parada favorita de derecha a izquierda. 3. Hacer clic en el botón "Quitar de Favoritos". |
| Roles | Usuario final del aplicativo. |
| Resultado esperado | Mensaje de éxito y lista de paradas favoritas sin la parada elegida para quitar. |
| Resultado obtenido | Mensaje de éxito y lista de paradas favoritas sin la parada elegida para quitar. |

Tabla 3.15: Caso de prueba quitar una parada de favoritos

| Caso de Prueba 13 | |
|--------------------------|---|
| Funcionalidad | Actualizar datos del perfil. |
| Entradas | <input type="checkbox"/> Nombres: José Rubén <input type="checkbox"/> Apellidos: Orellana Ricaurte |
| Antecedentes | <ol style="list-style-type: none"> 1. Haber ingresado con éxito en el aplicativo. |

| | |
|---------------------------|--|
| Acciones | <ol style="list-style-type: none"> 1. Ir a la pantalla de perfil. 2. Ingresar los datos como entradas en el formulario de Actualizar Datos. 3. Clic en el botón "Actualizar Datos". |
| Roles | Usuario final del aplicativo. |
| Resultado esperado | Mensaje de éxito y los nombres y apellidos que se ingresaron en el formulario. |
| Resultado obtenido | Mensaje de éxito y los nombres y apellidos que se ingresaron en el formulario. |

Tabla 3.16: Caso de prueba actualizar datos del perfil

4 CONCLUSIONES

4.1 CONCLUSIONES

- ❑ Basándose en la encuesta realizada sobre el uso del transporte público, se pudo constatar el interés y los requerimientos de las personas en usar un aplicativo que les permita conocer y visualizar las unidades de transporte que circulan en las rutas del Distrito Metropolitano de Quito, así como observar su interior; para de esta manera, ayudar a mitigar el principal problema manifestado en la encuesta, que es el no respetar el límite de pasajeros en las unidades de transporte.
- ❑ Durante la presentación de la aplicación a un determinado grupo de personas, el grupo mostró su interés en el aplicativo. Además, se puede afirmar que el sistema tendría una buena acogida por parte de los interesados en caso de ser desplegado al público. Debido a que a través de la aplicación se consigue información sobre cómo se gestionan las paradas, rutas y el seguimiento en tiempo real de los vehículos, lo cual permite a los usuarios decidir si tomar un determinado autobús.
- ❑ El uso de las metodologías ágiles, específicamente el marco de trabajo Scrum, permitió entregar incrementos con valor en la aplicación por cada Sprint. Esta metodología nos ayudó a recibir la respectiva retroalimentación de manera oportuna y poder realizar los cambios solicitados por parte del product owner en los incrementos posteriores de una forma eficiente.
- ❑ La aplicación desarrollada fue presentada como una PWA (Aplicación Web Progresiva), permitiendo a los interesados interactuar con ésta por medio de una URL en un navegador web, pudiendo instalarse fácilmente en un dispositivo móvil. Debido a que se lo realizó como PWA, la aplicación es fácilmente desplegable en servidores web y plataformas de software como servicio (SaaS).

- ❑ El empleo de paquetes y API's tanto de terceros, como los utilizados de la empresa de "Manticore Labs", permitió un desarrollo rápido y eficiente, debido a que no se gastó demasiado tiempo en crear ciertas partes del código que ya estuvieron creadas o eran repetitivas y gracias a esto, se dedicaron los esfuerzos en desarrollar las funcionalidades principales de la PWA.
- ❑ La implementación del protocolo MQTT a través de un micro-servicio permitió ahorrar recursos de red en el envío de las fotografías, por el hecho de que aquel protocolo anteriormente mencionado es ideal para poder trabajar sobre redes con un limitado ancho de banda en aplicaciones del internet de las cosas (IOT).
- ❑ Mediante el cuestionario SUS (Sistema de Escalas de Usabilidad), en el cual se obtuvo una puntuación promedio de 87.7 en la población encuestada, se evidenció que los usuarios no presentan mayores problemas de usabilidad en el manejo del aplicativo y que por el contrario, las funcionalidades propuestas en la aplicación serían una mejora en el uso del transporte público para los usuarios.
- ❑ Las pruebas de funcionalidad comprobaron que el aplicativo funciona correctamente, en estas pruebas se lograron cumplir con los criterios de aceptación de las historias de usuario y se obtuvieron los resultados esperados de los casos de prueba planteados.

4.2 RECOMENDACIONES

- ❑ En futuras implementaciones con los permisos pertinentes, poder usar las imágenes del interior de las unidades de transporte para que mediante el uso de inteligencia artificial y etiquetado de las imágenes, determinar automáticamente si una unidad de transporte se encuentra en su límite de pasajeros.
- ❑ Es necesario que los usuarios que estén operando la unidad de transporte posean una conexión a internet estable en su dispositivo móvil con el fin de que la geolocalización e imágenes de la unidad de transporte puedan emitirse correctamente.
- ❑ En caso de desplegarse el aplicativo al público, garantizar el acceso al aplicativo y su mantenimiento posterior. Así como utilizar servicios de autenticación del estándar OAuth que permitan a los usuarios acceder al aplicativo por medio de cuentas de Facebook y Google. De esta manera pueden tener una alternativa más sencilla y rápida de autenticación y registro.

- ❑ En la PWA, en la pantalla del seguimiento de un recorrido, adicionalmente mostrar en el mapa las paradas del recorrido en conjunto con los giros y trayectorias de las unidades de transporte, con el fin de que el usuario final tenga una mayor orientación de la posición del vehículo.
- ❑ En la pantalla de seguimiento del recorrido emitir una notificación cuando la unidad de transporte este próxima a la ubicación del usuario.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] M. B. Lema Vera y A. E. Becerra Punguil, «Imaginarios, discursos y lenguaje coloquial en el transporte público de Quito: Estudio de caso Cooperativa Disutran SA», B.S. thesis, 2012.
- [2] F. Arauz Ríos, «Informe de resultados de la encuesta realizada a usuarios, sobre el mejoramiento de la calidad del servicio de transporte público en el Distrito Metropolitano de Quito», Agencia Metropolitana de Tránsito, 2017, págs. 9-15. dirección: <https://bit.ly/encuAMT17>.
- [3] R. Cevallos, «PRIORIZACIÓN DE INDICADORES PARA EL ESTUDIO DE IMPACTO DEL PROYECTO QUITO CABLES EN EL DISTRITO METROPOLITANO DE QUITO. LÍNEA OFELIA - JAIME ROLDÓS.», B.S. thesis, Pontificia Universidad Católica del Ecuador, 2017.
- [4] Ionic, *Progressive Web Apps - Ionic Documentation*, 2020. dirección: <https://ionicframework.com/docs/core-concepts/what-are-progressive-web-apps>.
- [5] A. Osmani, *Getting started with Progressive Web Apps*, 2015. dirección: <https://addyosmani.com/blog/getting-started-with-progressive-web-apps/>.
- [6] M. Jailia, A. Kumar, M. Agarwal e I. Sinha, «Behavior of MVC (Model View Controller) based Web Application developed in PHP and .NET framework», en *2016 International Conference on ICT in Business Industry Government (ICTBIG)*, 2016.
- [7] M. Cohn, *Scrum Methodology and Project Management*, 2020. dirección: <http://www.mountaingoatsoftware.com/agile/scrum>.
- [8] Atlassian, *Scrum: qué es, cómo funciona y por qué es excelente*, 2020. dirección: <https://www.atlassian.com/es/agile/scrum>.
- [9] *¿Qué es el Internet de las cosas?*, 2020. dirección: <https://www.redhat.com/es/topics/internet-of-things/what-is-iot>.
- [10] *Angular Docs*, 2020. dirección: <https://angular.io/docs>.

- [11] Node.js, *Node.js*, 2020. dirección: <https://nodejs.org/es/>.
- [12] *Documentation | NestJS - A progressive Node.js framework*, 2020. dirección: <https://docs.nestjs.com/>.
- [13] Ionic, *Ionic Framework - Ionic Documentation*, 2020. dirección: <https://ionicframework.com/docs>.
- [14] *GitLab*, 2016. dirección: <https://about.gitlab.com/>.
- [15] *Typeorm.io*, 2020. dirección: <https://typeorm.io/#/>.
- [16] *What is MySQL?*, 2020. dirección: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>.
- [17] *MongoDB*, 2020. dirección: <https://www.mongodb.com/es>.
- [18] *¿Qué es DOCKER?*, 2020. dirección: <https://www.redhat.com/es/topics/containers/what-is-docker>.
- [19] *Getting started with WebStorm*, 2020. dirección: <https://www.jetbrains.com/help/webstorm/getting-started-with-webstorm.html>.
- [20] F. Redondo, *Postman: gestiona y construye tus APIs rápidamente*, <https://www.paradigmadigital.com/dev/postman-gestiona-construye-tus-apis-rapidamente/>, [Online; accessed 24-April-2020], 2019.
- [21] S. Rider, *About DBeeer*, <https://github.com/dbeaver/dbeaver/wiki>, [Online; accessed 24-April-2020], 2020.
- [22] L. Morela, *¿Qué es Adobe XD y para qué sirve?*, <https://blog.hubspot.es/marketing/para-que-sirve-adobe-xd>, [Online; accessed 24-April-2020], 2019.
- [23] T. Norris, *What is npm?*, <https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/>, [Online; accessed 24-April-2020], 2011.
- [24] K. S. David Berlind Wendell Santos, *Open Street maps API*, <https://www.programmableweb.com/api/openstreetmap>, [Online; accessed 24-April-2020].
- [25] V. S. García, *Introducción a Raspberry Pi*, [Online; accessed 24-April-2020], 2015. dirección: <https://hacklabalmeria.net/recursos/2015-10-13/intropi.pdf>.

- [26] A. M. de Transito, *Informe de los resultados de la encuesta realizada a usuarios, sobre el mejoramiento de la calidad del servicio de transporte público en el Distrito Metropolitano de Quito*, [Online; accessed 30-April-2020], 2017. dirección: http://www7.quito.gob.ec/mdmq_ordenanzas/Sesiones%20del%20Concejo/2017/Sesi%C3%B3n%20Extraordinaria%202017-06-27/Continuaci%C3%B3n%20Primer%20debate/Informe%20Resultados%20Encuesta%20-%20AMT.pdf.
- [27] I. Sommerville, *Ingeniería de software*, Novena edición. Naucalpan de Juárez, Estado de México: Pearson Educación, 2011.
- [28] A. S. for Public Affairs, *System Usability Scale (SUS) | Usability.gov*, 2020. dirección: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>.
- [29] J. Sauro, *Measuring Usability with the System Usability Scale (SUS)*, 2011. dirección: <https://measuringu.com/sus/>.

6 ANEXOS

6.1 ANEXO 1: REPRESENTACIÓN GRÁFICA DE RESULTADOS DE LA ENCUESTA SOBRE EL USO DEL TRANSPORTE PÚBLICO

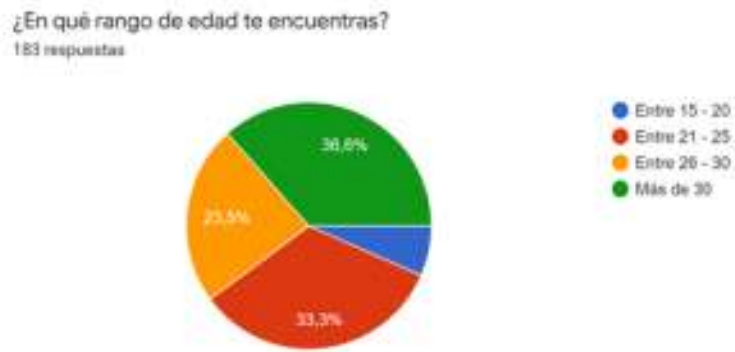


Figura 6.1: Pregunta 1 Encuesta sobre el uso del transporte público



Figura 6.2: Pregunta 2 Encuesta sobre el uso del transporte público

¿Cuál es el tipo de transporte público que más utilizas?
183 respuestas

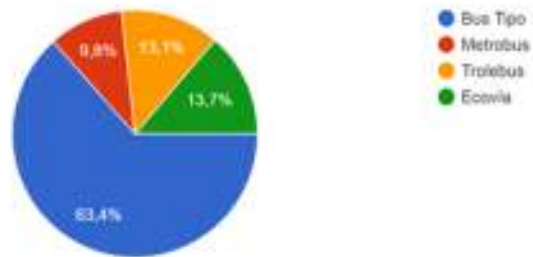


Figura 6.3: Pregunta 3 Encuesta sobre el uso del transporte público

¿En qué rango de horarios haces uso del servicio del transporte público?
183 respuestas

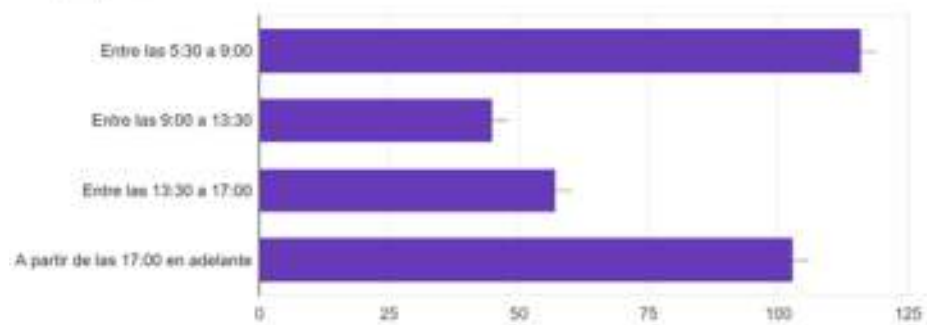


Figura 6.4: Pregunta 4 Encuesta sobre el uso del transporte público

¿Cuáles crees que son los principales problemas dentro del servicio de transporte público?
183 respuestas

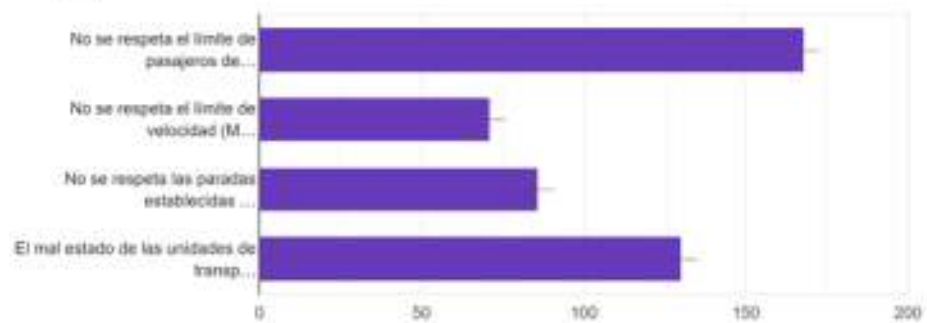


Figura 6.5: Pregunta 5 Encuesta sobre el uso del transporte público

¿Cuánto tiempo esperas para tomar una unidad de transporte?
183 respuestas

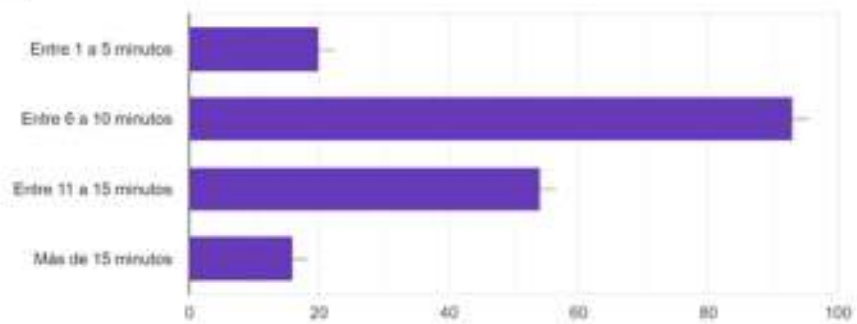


Figura 6.6: Pregunta 6 Encuesta sobre el uso del transporte público

En las paradas encuentras rótulos que contienen información sobre las rutas y paradas. ¿Estos son claros y entendibles?
183 respuestas

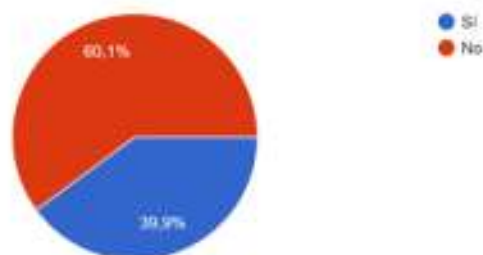


Figura 6.7: Pregunta 7 Encuesta sobre el uso del transporte público

¿Harias uso de la unidad de transporte si supieras que la misma está al límite de su capacidad de pasajeros?
183 respuestas

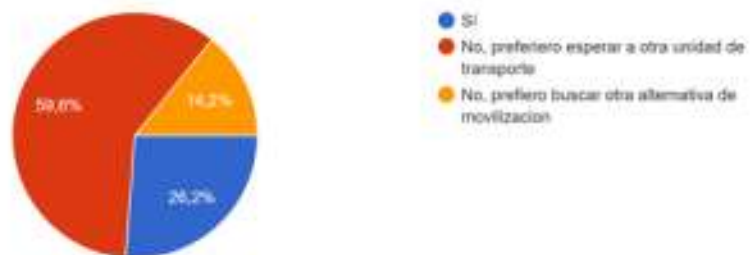


Figura 6.8: Pregunta 8 Encuesta sobre el uso del transporte público

¿Consideras útil visualizar el interior del autobús que estás esperando para determinar si está lleno?

181 respuestas

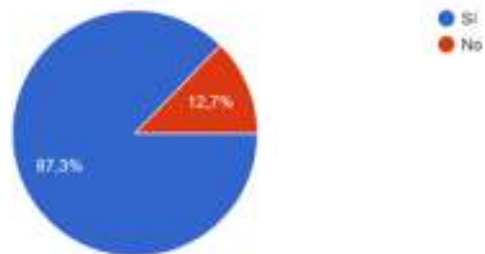


Figura 6.9: Pregunta 9 Encuesta sobre el uso del transporte público

¿Consideras útil conocer cuáles unidades de transporte te permitirán llegar a tu parada de destino?

183 respuestas



Figura 6.10: Pregunta 10 Encuesta sobre el uso del transporte público

¿Consideras útil conocer cuándo llega o dónde está la unidad de transporte que estás esperando?

183 respuestas



Figura 6.11: Pregunta 11 Encuesta sobre el uso del transporte público

A partir de la información anterior que has considerado útil, ¿Te gustaría que estuviera a tu disposición a través de una solución tecnológica como lo es una aplicación móvil?

183 respuestas



Figura 6.12: Pregunta 12 Encuesta sobre el uso del transporte público

¿Consideras conveniente incluir la información del conductor que opera la unidad de transporte en la aplicación? (Al igual como lo hacen las aplicaciones de Uber o Cabify)

182 respuestas

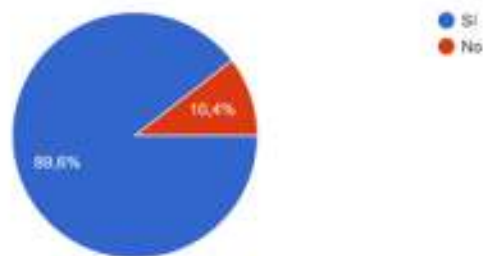


Figura 6.13: Pregunta 13 Encuesta sobre el uso del transporte público

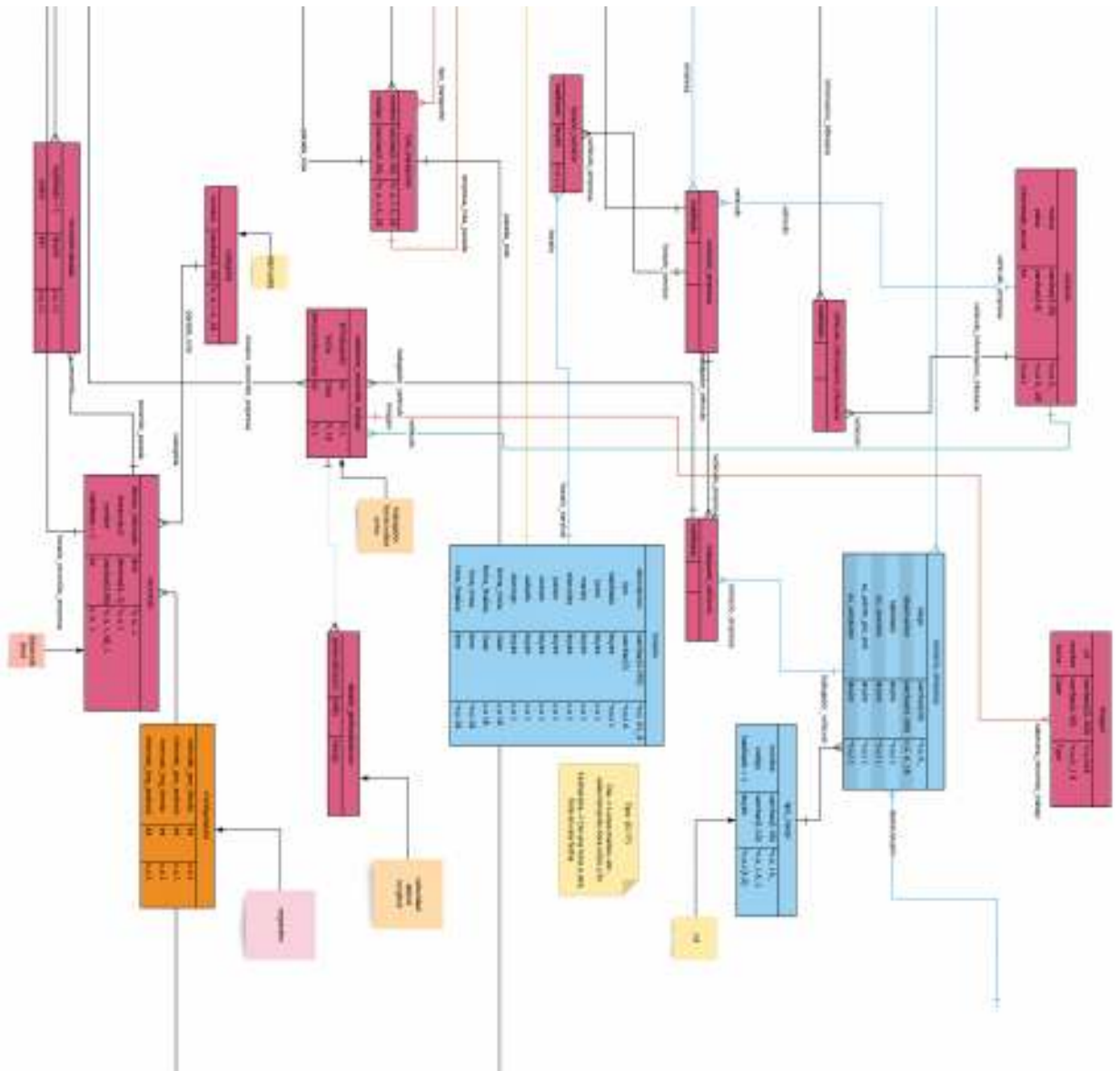


Figura 6.15: Diagrama entidad relación parte 2/2

6.3 ANEXO 3: ARQUITECTURA DE MICROSERVICIOS

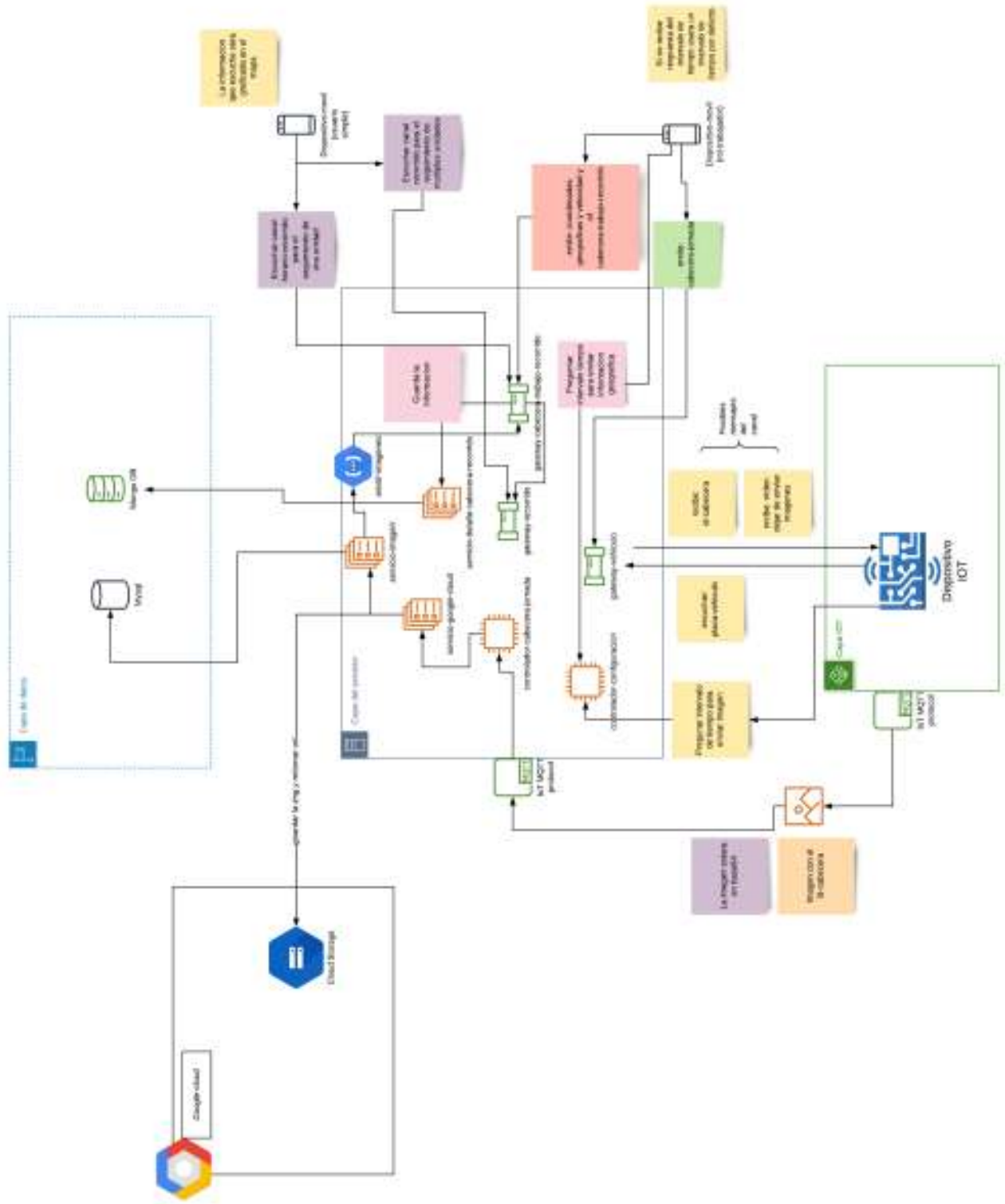


Figura 6.16: Interacción con microservicios y protocolos

6.4 ANEXO 4: HISTORIAS DE USUARIO

| Código | Descripción | Prioridad | Peso |
|---------|--|-----------|------|
| HN000 | Realizar encuesta | Alta | 5 |
| HN001 | Diseño del modelo lógico de la base de datos | Alta | 5 |
| HNWM002 | Diseño de interfaces gráficas de usuario (Mockups) | Alta | 5 |
| HN003 | Diseño de la arquitectura del sistema | Alta | 5 |
| HN004 | Levantar el ambiente de desarrollo para el servidor web (backend) | Alta | 8 |
| HN005 | Levantar el ambiente de desarrollo para el frontend | Alta | 8 |
| HN006 | Levantar el ambiente de desarrollo para el aplicativo móvil | Alta | 5 |
| HNB007 | Implementar la entidad vehículo en el Backend | Baja | 3 |
| HNB008 | Implementar la entidad contacto-empresa en el Backend | Media | 3 |
| HNB009 | Implementar la entidad trabajador-vehículo en el Backend | Media | 3 |
| HNB010 | Implementar la entidad vehículo-informacion-tributaria en el Backend | Baja | 3 |
| HNB011 | Implementar la entidad vehículo-empresa en el Backend | Media | 3 |
| HNB012 | Implementar la entidad informacion-tributaria en el Backend | Baja | 3 |
| HNB013 | Implementar la entidad empresa en el Backend | Alta | 3 |
| HNB014 | Implementar la entidad horario en el Backend | Media | 3 |
| HNB015 | Implementar la entidad horario-servicio en el Backend | Media | 3 |
| HNB016 | Implementar la entidad empresa-ruta-transporte en el Backend | Media | 3 |
| HNB017 | Implementar la entidad tipo-transporte en el Backend | Baja | 3 |
| HNB018 | Implementar la entidad parada en el Backend | Alta | 3 |
| HNB019 | Implementar la entidad localización en el Backend | Alta | 3 |
| HNB020 | Implementar la entidad lugar en el Backend | Baja | 3 |
| HNB021 | Implementar la entidad datos-usuario en el Backend | Media | 3 |
| HNB022 | Implementar la entidad parada-favorita en el Backend | Alta | 3 |
| HNB023 | Implementar la entidad tipo-transporte-parada en el Backend | Alta | 3 |
| HNB024 | Implementar la entidad ruta-transporte en el Backend | Alta | 3 |

| | | | |
|--------|--|-------|---|
| HNB025 | Implementar la entidad horario-recorrido-empresa en el Backend | Alta | 3 |
| HNB026 | Implementar la entidad parada-ruta en el Backend | Alta | 3 |
| HNB027 | Implementar la entidad categoría en el Backend | Baja | 3 |
| HNB028 | Implementar la entidad recorrido-parada en el Backend | Alta | 3 |
| HNB029 | Implementar la entidad recorrido en el Backend | Alta | 3 |
| HFW030 | Como usuario administrador, yo puedo gestionar empresas | Media | 5 |
| HFW031 | Como usuario administrador, yo puedo gestionar vehículos | Media | 5 |
| HFW032 | Como usuario administrador, yo puedo asignar vehículos a una determinada empresa | Media | 5 |
| HFW033 | Como usuario administrador, yo puedo asignar dueños a los vehículos | Baja | 5 |
| HFW034 | Como usuario administrador, yo puedo gestionar horarios para los Recorridos | Media | 8 |
| HFW035 | Como usuario administrador, yo puedo gestionar horarios para los Vehículos | Media | 5 |
| HFW036 | Como usuario administrador, yo puedo gestionar lugares | Alta | 3 |
| HFW037 | Como usuario administrador, yo puedo gestionar paradas para un lugar determinado | Alta | 5 |
| HFW038 | Como usuario administrador, yo puedo gestionar rutas de transporte | Alta | 5 |
| HFW039 | Como usuario administrador, yo puedo gestionar recorridos para una ruta de transporte | Alta | 8 |
| HFW040 | Como usuario administrador, yo puedo gestionar Paradas-Ruta de un recorrido específico | Alta | 8 |
| HFW041 | Como usuario administrador, yo puedo gestionar paradas de una ruta de transporte | Alta | 5 |
| HFW042 | Como usuario administrador, yo puedo gestionar categorías | Baja | 2 |
| HFW043 | Como usuario administrador, yo puedo gestionar tipos de transporte | Baja | 2 |
| HFM044 | Como usuario, yo puedo registrarme en el sistema | Baja | 5 |
| HFM045 | Como usuario registrado, yo puedo ingresar al sistema | Baja | 3 |
| HFM046 | Como usuario, yo puedo visualizar paradas en un mapa | Alta | 5 |

| | | | |
|----------|--|------|---|
| HFM047 | Como usuario registrado, yo puedo buscar paradas | Alta | 5 |
| HFM048 | Como usuario registrado, yo puedo ver las rutas de una parada determinada | Alta | 3 |
| HFM049 | Como usuario registrado, yo puedo agregar una parada a mis paradas favoritas | Alta | 3 |
| HFM050 | Como usuario registrado, yo puedo gestionar mis paradas favoritas | Alta | 3 |
| HNIOT051 | Permitir que el "Raspberry PI" tome fotografías a través de sus dispositivos y las envíe al servidor web central en un periodo de tiempo determinado | Alta | 8 |
| HFM052 | Como usuario registrado, yo puedo visualizar en un mapa las unidades de transporte de una ruta de transporte en específico | Alta | 8 |
| HFM053 | Como usuario registrado, yo puedo visualizar la información de una determinada unidad de transporte | Alta | 8 |

6.5 ANEXO 5: SPRINT BACKLOG SPRINT 0

| | |
|--------------------------------|---|
| Código | HN000 |
| Descripción | Realizar la encuesta |
| Prioridad | Alta |
| Peso | 5 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Definir el tipo de encuesta que se va a realizar <input type="checkbox"/> Formular preguntas para la encuesta. <input type="checkbox"/> Realizar la encuesta de manera digital. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> La encuesta debe estar sustentada en base algún estudio u alguna encuesta ya realizada previamente. <input type="checkbox"/> Las preguntas de la encuesta deben estar aprobadas por el product owner. |

Tabla 6.2: Historia de usuario: HN000

| | |
|--------------------------------|---|
| Código | HN001 |
| Descripción | Diseño del modelo lógico de la base de datos |
| Prioridad | Alta |
| Peso | 5 |
| Tareas | <input type="checkbox"/> Realizar el modelo entidad relación para el aplicativo. |
| Criterios de aceptación | <input type="checkbox"/> El diagrama entidad relación debe aplicar los conceptos de normalización de bases de datos |

Tabla 6.3: Historia de usuario: HN001

| | |
|--------------------------------|--|
| Código | HNWM002 |
| Descripción | Diseño de interfaces gráficas de usuario (Mockups) |
| Prioridad | Alta |
| Peso | 5 |
| Tareas | <input type="checkbox"/> Realizar Mockups para el aplicativo web (opcional). <input type="checkbox"/> Realizar Mockups para el aplicativo móvil |
| Criterios de aceptación | <input type="checkbox"/> Los Mockups deben ser claros y entendibles. |

Tabla 6.4: Historia de usuario: HNWM002

| | |
|--------------------|---------------------------------------|
| Código | HN003 |
| Descripción | Diseño de la arquitectura del sistema |
| Prioridad | Alta |
| Peso | 5 |

| | |
|--------------------------------|--|
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Realizar diagrama del sistema para la capa de datos. <input type="checkbox"/> Realizar el diagrama del sistema para la capa del negocio. <input type="checkbox"/> Realizar el diagrama del sistema para la capa del sistema para la capa de presentación. <input type="checkbox"/> Realizar el diagrama del sistema para la capa de presentación. Todos los diagramas deben tener interacción entre ellos. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> El modelo de la arquitectura de datos debe estar aprobado por el Product Owner. |

Tabla 6.5: Historia de usuario: HN003

| | |
|--------------------|--|
| Código | HN004 |
| Descripción | Levantar el ambiente de desarrollo para el servidor web (backend) |
| Prioridad | Alta |
| Peso | 8 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Crear el repositorio git del submódulo de transportes. <input type="checkbox"/> Clonar el repositorio con el proyecto base llamado "fear" y crear una rama llamada "transportes" a partir de la rama "empresa". <input type="checkbox"/> Agregar el submódulo transportes al proyecto base. <input type="checkbox"/> Agregar el submódulo empresa al proyecto. <input type="checkbox"/> Agregar el submódulo roles al proyecto. <input type="checkbox"/> Agregar el submódulo auth0 al proyecto. <input type="checkbox"/> Agregar el submódulo datos-de-prueba al proyecto. <input type="checkbox"/> Dentro de cada submódulo crear la rama "transportes". |

| | |
|--------------------------------|---|
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> No deben existir conflictos en los archivos al agregar los submódulos al repositorio principal del proyecto. <input type="checkbox"/> No deben existir conflictos de dependencias externas, si alguno de los submódulos está haciendo uso de alguna librería. |
|--------------------------------|---|

Tabla 6.6: Historia de usuario: HN004

| | |
|--------------------|--|
| Código | HN005 |
| Descripción | Levantar el ambiente de desarrollo para el frontend |
| Prioridad | Alta |
| Peso | 8 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Crear el repositorio git del submódulo de transportes-front. <input type="checkbox"/> Clonar el repositorio con el proyecto base llamado "fear" y crear una rama llamada "transportes" a partir de la rama "empresa". <input type="checkbox"/> Agregar el submódulo transportes-front al proyecto base. <input type="checkbox"/> Agregar el submódulo empresa-front al proyecto. <input type="checkbox"/> Agregar el submódulo roles-front al proyecto. <input type="checkbox"/> Agregar el submódulo articulo-front al proyecto. <input type="checkbox"/> Agregar el submódulo pedido-front al proyecto. <input type="checkbox"/> Agregar el submódulo pedido-front al proyecto. <input type="checkbox"/> Agregar el submódulo vendedor-front al proyecto. <input type="checkbox"/> Agregar el submódulo inventario-front al proyecto. <input type="checkbox"/> Agregar el submódulo menu-lateral-front al proyecto. <input type="checkbox"/> Agregar el submódulo cargas-masivas al proyecto. <input type="checkbox"/> Agregar el submódulo internacionalizacion-front al proyecto. <input type="checkbox"/> Agregar el submódulo front-comun al proyecto. <input type="checkbox"/> Dentro de cada submódulo crear desde la rama "empresa" la rama "transportes". |

| | |
|--------------------------------|---|
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> No deben existir conflictos en los archivos al agregar los submódulos al repositorio principal del proyecto. <input type="checkbox"/> No deben existir conflictos de dependencias externas. |
|--------------------------------|---|

Tabla 6.7: Historia de usuario: HN005

| | |
|--------------------------------|---|
| Código | HN006 |
| Descripción | Levantar el ambiente de desarrollo para el aplicativo móvil |
| Prioridad | Alta |
| Peso | 5 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Crear el repositorio git del submódulo transportes-movil. <input type="checkbox"/> Clonar el repositorio con el proyecto base llamado "fear" y crear una rama llamada "transportes" a partir de la rama "menu-pedidos". <input type="checkbox"/> Agregar el submódulo auth0-movil al proyecto. <input type="checkbox"/> Para el submódulo crear desde la rama "menu-pedidos" la rama "transportes". |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> No deben existir conflictos en los archivos al agregar los submódulos al repositorio principal del proyecto. <input type="checkbox"/> No deben existir conflictos de dependencias externas. |

Tabla 6.8: Historia de usuario: HN006

6.6 ANEXO 6: SPRINT BACKLOG SPRINT 1

| | |
|--------------------|---|
| Código | HNB007 |
| Descripción | Implementar la entidad vehículo en el Backend |
| Prioridad | Baja |

| | |
|--------------------------------|--|
| Peso | 3 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.9: Historia de usuario: HNB007

| | |
|--------------------|---|
| Código | HNB008 |
| Descripción | Implementar la entidad contacto-empresa en el Backend |
| Prioridad | Media |
| Peso | 3 |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda mediante el uso de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.10: Historia de usuario: HNB008

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB009</p> |
| <p>Descripción</p> | <p>Implementar la entidad trabajador-vehículo en el Backend</p> |
| <p>Prioridad</p> | <p>Media</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda con el apoyo de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.11: Historia de usuario: HNB009

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB010</p> |
| <p>Descripción</p> | <p>Implementar la entidad vehículo-informacion-tributaria en el Backend</p> |
| <p>Prioridad</p> | <p>Baja</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.12: Historia de usuario: HNB010

| | |
|---------------------------|--|
| <p>Código</p> | <p>HNB011</p> |
| <p>Descripción</p> | <p>Implementar la entidad vehículo-empresa en el Backend</p> |
| <p>Prioridad</p> | <p>Media</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.13: Historia de usuario: HNB011

| | |
|---------------------------|--|
| <p>Código</p> | <p>HNB012</p> |
| <p>Descripción</p> | <p>Implementar la entidad información-tributaria en el Backend</p> |
| <p>Prioridad</p> | <p>Baja</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.14: Historia de usuario: HNB012

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB013</p> |
| <p>Descripción</p> | <p>Implementar la entidad empresa en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.15: Historia de usuario: HNB013

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB014</p> |
| <p>Descripción</p> | <p>Implementar la entidad horario en el Backend</p> |
| <p>Prioridad</p> | <p>Media</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.16: Historia de usuario: HNB014

| | |
|---------------------------|--|
| <p>Código</p> | <p>HNB015</p> |
| <p>Descripción</p> | <p>Implementar la entidad horario-servicio en el Backend</p> |
| <p>Prioridad</p> | <p>Media</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.17: Historia de usuario: HNB015

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB016</p> |
| <p>Descripción</p> | <p>Implementar la entidad empresa-ruta-transporte en el Backend</p> |
| <p>Prioridad</p> | <p>Media</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.18: Historia de usuario: HNB016

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB017</p> |
| <p>Descripción</p> | <p>Implementar la entidad tipo-transporte en el Backend</p> |
| <p>Prioridad</p> | <p>Baja</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.19: Historia de usuario: HNB017

| | |
|---------------------------|--|
| <p>Código</p> | <p>HNB018</p> |
| <p>Descripción</p> | <p>Implementar la entidad parada en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.20: Historia de usuario: HNB018

| | |
|---------------------------|--|
| <p>Código</p> | <p>HNB019</p> |
| <p>Descripción</p> | <p>Implementar la entidad localización en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.21: Historia de usuario: HNB019

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB020</p> |
| <p>Descripción</p> | <p>Implementar la entidad lugar en el Backend</p> |
| <p>Prioridad</p> | <p>Baja</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.22: Historia de usuario: HNB020

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB021</p> |
| <p>Descripción</p> | <p>Implementar la entidad datos-usuario en el Backend</p> |
| <p>Prioridad</p> | <p>Media</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.23: Historia de usuario: HNB021

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB022</p> |
| <p>Descripción</p> | <p>Implementar la entidad parada-favorita en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.24: Historia de usuario: HNB022

| | |
|---------------------------|--|
| <p>Código</p> | <p>HNB023</p> |
| <p>Descripción</p> | <p>Implementar la entidad tipo-transporte-parada en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.25: Historia de usuario: HNB023

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB024</p> |
| <p>Descripción</p> | <p>Implementar la entidad ruta-transporte en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.26: Historia de usuario: HNB024

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB025</p> |
| <p>Descripción</p> | <p>Implementar la entidad horario-recorrido-empresa en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.27: Historia de usuario: HNB025

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB026</p> |
| <p>Descripción</p> | <p>Implementar la entidad parada-ruta en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.28: Historia de usuario: HNB026

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB027</p> |
| <p>Descripción</p> | <p>Implementar la entidad categoría en el Backend</p> |
| <p>Prioridad</p> | <p>Baja</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.29: Historia de usuario: HNB027

| | |
|---------------------------|--|
| <p>Código</p> | <p>HNB028</p> |
| <p>Descripción</p> | <p>Implementar la entidad recorrido-parada en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.30: Historia de usuario: HNB028

| | |
|---------------------------|---|
| <p>Código</p> | <p>HNB029</p> |
| <p>Descripción</p> | <p>Implementar la entidad recorrido en el Backend</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar atributos. <input type="checkbox"/> Implementar relaciones entre las entidades respectivas según el diagrama entidad relación. <input type="checkbox"/> Implementar validaciones de los atributos. <input type="checkbox"/> Implementar controladores CRUD. <input type="checkbox"/> Implementar servicios CRUD. <input type="checkbox"/> Crear datos de prueba para esta entidad, <input type="checkbox"/> Implementar políticas de seguridad (solo definición) <input type="checkbox"/> Implementar gateways (en caso de ser necesario) <input type="checkbox"/> Implementar resolvers (en caso de ser necesario) |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Se debe poder insertar registros a través de la API REST con el método POST. <input type="checkbox"/> Se deben listar todos los registros con o sin criterios de búsqueda a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder editar registros a través de la API REST con el método PUT. <input type="checkbox"/> Se debe poder buscar un registro por su ID como parámetro de ruta a través de la API REST con el método GET. <input type="checkbox"/> Se debe poder eliminar un registro por su ID como parámetro de ruta a través de la API REST con el método DELETE. |

Tabla 6.31: Historia de usuario: HNB029

6.7 ANEXO 7: DOCUMENTACIÓN DETALLADA DEL SPRINT: 1

Implementación de las entidades del diagrama entidad relación

En este sprint se implementó todas las entidades en el backend descritas en el diagrama de base de datos. La implementación por lo general fue un proceso repetitivo para cada

entidad, por lo cual solamente se mostrará la implementación de la entidad 'recorrido' en los temas posteriores.

Implementación de la entidad (modelo de datos)

La implementación del modelo de datos para la entidad recorrido se la realizó a través de las bibliotecas proporcionadas por el framework 'NestJS' y por las bibliotecas de 'Manticore-labs'. En la siguiente figura se muestra la implementación de la entidad con el tipo de dato correspondiente para cada atributo y con sus 'check-constraints' respectivos.

```
@Entity( name: 'recorrido')
export class RecorridoEntity extends PrincipalEntity {
  @Column(
    options: {
      name: 'distancia',
      type: 'decimal',
      precision: 4,
      scale: 2,
      default: 0,
    },
  ),
  distancia: number = 0;

  @Column( options: {
    type: 'varchar',
    name: 'nombre',
    length: 80,
  })
  nombre: string = null;

  @Column( options: {
    type: 'time',
    name: 'tiempo_estimado',
  })
  tiempoEstimado: string = null;

  @Column( options: {
    name: 'habilitado',
    type: 'tinyint',
    default: 1,
  })
  habilitado: 0 | 1 = 1;
}
```

Figura 6.17: Entidad: Recorrido

En la siguiente figura se muestra la continuación de la implementación de la entidad con la definición de sus relaciones: 'uno a muchos' y 'muchos a uno'.

```
@ManyToOne(  
    typeFunctionOrTarget: type => RutaTransporteEntity,  
    inverseSide: rutaTransporte => rutaTransporte.recorridos,  
)  
rutaTransporte: RutaTransporteEntity;  
  
@ManyToOne(  
    typeFunctionOrTarget: type => CategoriaEntity,  
    inverseSide: categoria => categoria.recorridos,  
    options: {  
        nullable: false,  
    },  
)  
categoria: CategoriaEntity;  
  
@OneToMany(  
    typeFunctionOrTarget: type => RecorridoParadaEntity,  
    inverseSide: recorridosParada => recorridoParada.recorrido,  
)  
recorridosParadas: RecorridoParadaEntity[];  
  
@OneToMany(  
    typeFunctionOrTarget: type => HorarioRecorridoEmpresaEntity,  
    inverseSide: horarioRutaEmpresa => horarioRutaEmpresa.recorrido,  
)  
horarioRecorridoEmpresa: HorarioRecorridoEmpresaEntity[];  
}
```

Figura 6.18: Entidad: Recorrido

Implementación de los DTO's para actualizar y crear.

Se definieron dos tipos de DTO's (Data Transfer Objects):

❑ **Para crear:** Se definió el DTO para validar la información de un registro del tipo 'Recorrido' antes de guardarlo en la base de datos, con los siguientes criterios:

- Validar que el campo 'tiempoEstimado' no este vacío y que acepte cadenas de texto.
- Validar que el campo 'distancia' no este vacío y que acepte números positivos.
- Validar que el campo 'nombre' no este vacío y que acepte cadenas de texto con una longitud de caracteres mínima de 3 y una longitud máxima de 80.
- El campo 'habilitado' es opcional y aceptará solamente datos numéricos que sean solamente 1 o 0.

```

export class RecorridoCreateDto extends PrincipalDto {
    @NotEmpty()
    @IsString()
    tiempoEstimado: string;

    @NotEmpty()
    @IsNumber()
    @IsPositive()
    distancia: number;

    @NotEmpty()
    @IsString()
    @Length( min: 3, max: 80)
    nombre: string;

    @IsOptional()
    @IsNumber()
    @ValorEstaDentroDeArreglo( property: [0, 1])
    habilitado: 0 | 1;
}

```

Figura 6.19: DTO para crear un Recorrido

❑ **Para actualizar:** Se definió el siguiente DTO para validar la información de un registro del tipo 'Recorrido' antes de actualizarlo en la base de datos:

- Validar que el campo 'tiempoEstimado' no este vacío y que acepte cadenas de texto.
- El campo 'distancia' es opcional y que acepte números positivos.
- El campo 'nombre' es opcional y que acepte cadenas de texto con una longitud de caracteres mínima de 3 y una longitud máxima de 80.
- El campo 'habilitado' es opcional y aceptará solamente datos numéricos que sean solamente 1 o 0.

```

export class RecorridoUpdateDto extends PrincipalDto {
    @IsOptional()
    @IsString()
    tiempoEstimado: string;

    @IsOptional()
    @IsNumber()
    @IsPositive()
    distancia: number;

    @IsOptional()
    @IsString()
    @Length( min: 3, max: 80)
    nombre: string;

    @IsOptional()
    @IsNumber()
    @ValorEstaDentroDeArreglo( property: [0, 1])
    habilitado: 0 | 1;
}

```

Figura 6.20: DTO para actualizar un Recorrido

Implementación de servicios (interacción con la capa de datos).

La implementación de los servicios se la hizo a través de la definición de una clase específica para la entidad y que extiende de una clase genérica principal, esta clase genérica tiene todos los métodos CRUD implementados.

```

@Injectables()
export class RecorridoService extends PrincipalService<
    RecorridoEntity,
    RecorridoCreateDto,
    RecorridoUpdateDto> {
    constructor(@InjectRepository(RecorridoEntity)
        private readonly _recorridoRepository: Repository<RecorridoEntity>) {
        super(_recorridoRepository, RecorridoEntity, nombreRepositorio: 'default');
    }
}

```

Figura 6.21: Servicio para la entidad 'recorrido'

Implementación de controladores.

Al igual que los servicios, se implemento una clase para la entidad que se extiende de una clase genérica principal, esta clase principal tiene ya los métodos CRUD implementados.

```
@Controller( prefix: 'recorrido')
export class RecorridoController extends PrincipalController<
  RecorridoCreateDto,
  RecorridoUpdateDto> {
  constructor(private readonly _recorridoService: RecorridoService) {
    super( politicasRecorrido, // politicas de seguridad
    _recorridoService, // servicio
    _dto: { // Dto
      CreateDto: RecorridoCreateDto,
      UpdateDto: RecorridoUpdateDto,
    },
    _skip: 0, // skip
    _take: 30, // take
    mensajesRecorrido,
    contexto: undefined, // contexto
  );
  this.contexto = this;
}
}
```

Figura 6.22: Controlador para la entidad 'recorrido'

Creación de datos de prueba.

Para poder realizar pruebas en el sistema se crearon datos de prueba, estos datos se los guardo en formato '.json' y se los carga al sistema cuando se levanta el servidor web.

```
const respuestaRecorrido = await crearDatos(
  this._recorridoService,
  rutaDatos: '../..../submodulo-datos-de-prueba/datos-prueba-3/datos-recorrido.json',
);
```

Figura 6.23: Creación de los datos de prueba para 'recorridos'

```
{
  "id": 1,
  "tiempoEstimado": "20:30:46",
  "distancia": "1.02",
  "nombre": "R1",
  "rutaTransporte": 4,
  "categoria": 2
},
{
  "id": 2,
  "tiempoEstimado": "12:41:1",
  "distancia": "20.65",
  "nombre": "R2",
  "rutaTransporte": 8,
  "categoria": 3
},
{
  "id": 3,
  "tiempoEstimado": "17:1:26",
  "distancia": "24.88",
  "nombre": "R3",
  "rutaTransporte": 2,
  "categoria": 1
},
}
```

Figura 6.24: Datos de prueba para la entidad 'recorrido'

Pruebas de la API REST.

Para poder probar la API REST se hizo uso de la herramienta 'POSTMAN', se probó la API REST para los siguientes casos:

- ❑ **Listar registros:** Por defecto se listan los diez últimos registros y se retornan los resultados en un arreglo en donde el primer elemento son los diez últimos registros y el segundo elemento es el total de elementos que existen en la base de datos. Para ejecutar esta consulta se emplea el método de petición HTTP GET.

```
GET http://localhost:8080/recorrido/

Pretty Raw Preview Visualize JSON

140
141     "nombre": "R16",
142     "tiempoEstimado": "00:42:45",
143     "habilitado": 1,
144     "id": 16,
145     "createdAt": "2020-05-16T05:28:33.000Z",
146     "updatedAt": "2020-05-16T05:28:33.000Z"
147   },
148   {
149     "distancia": "23.44",
150     "nombre": "R17",
151     "tiempoEstimado": "12:37:52",
152     "habilitado": 1,
153     "id": 17,
154     "createdAt": "2020-05-16T05:28:33.000Z",
155     "updatedAt": "2020-05-16T05:28:33.000Z"
156   },
157   {
158     "distancia": "31.46",
159     "nombre": "R18",
160     "tiempoEstimado": "12:43:26",
161     "habilitado": 1,
162     "id": 18,
163     "createdAt": "2020-05-16T05:28:33.000Z",
164     "updatedAt": "2020-05-16T05:28:33.000Z"
165   },
166   {
167     "distancia": "16.06",
168     "nombre": "R19",
169     "tiempoEstimado": "12:54:48",
170     "habilitado": 1,
171     "id": 19,
172     "createdAt": "2020-05-16T05:28:33.000Z",
173     "updatedAt": "2020-05-16T05:28:33.000Z"
174   },
175   {
176     "distancia": "26.42",
177     "nombre": "R20",
178     "tiempoEstimado": "18:30:39",
179     "habilitado": 1,
180     "id": 20,
181     "createdAt": "2020-05-16T05:28:33.000Z",
182     "updatedAt": "2020-05-16T05:28:33.000Z"
183   }
184 ],
185 20
```

Figura 6.25: Listar los 10 últimos Recorridos

- Listar registros con criterios de búsqueda:** En este caso se van a listar los 'recorridos' que estén no estén habilitados.

```
GET http://localhost:8080/recorrido?criterioBusqueda={"where":{"habilitado":0}}

Pretty Raw Preview Visualize JSON

1 []
2 []
3 0
4 []
```

Figura 6.26: Listar Recorridos deshabilitados

❑ **Crear un registro:** Se crea un nuevo registro con el método de petición HTTP POST, a manera de prueba se hizo con los siguientes valores:

- distancia: 5.2
- nombre: R89
- tiempoEstimado: 02:30:05
- categoria: 1

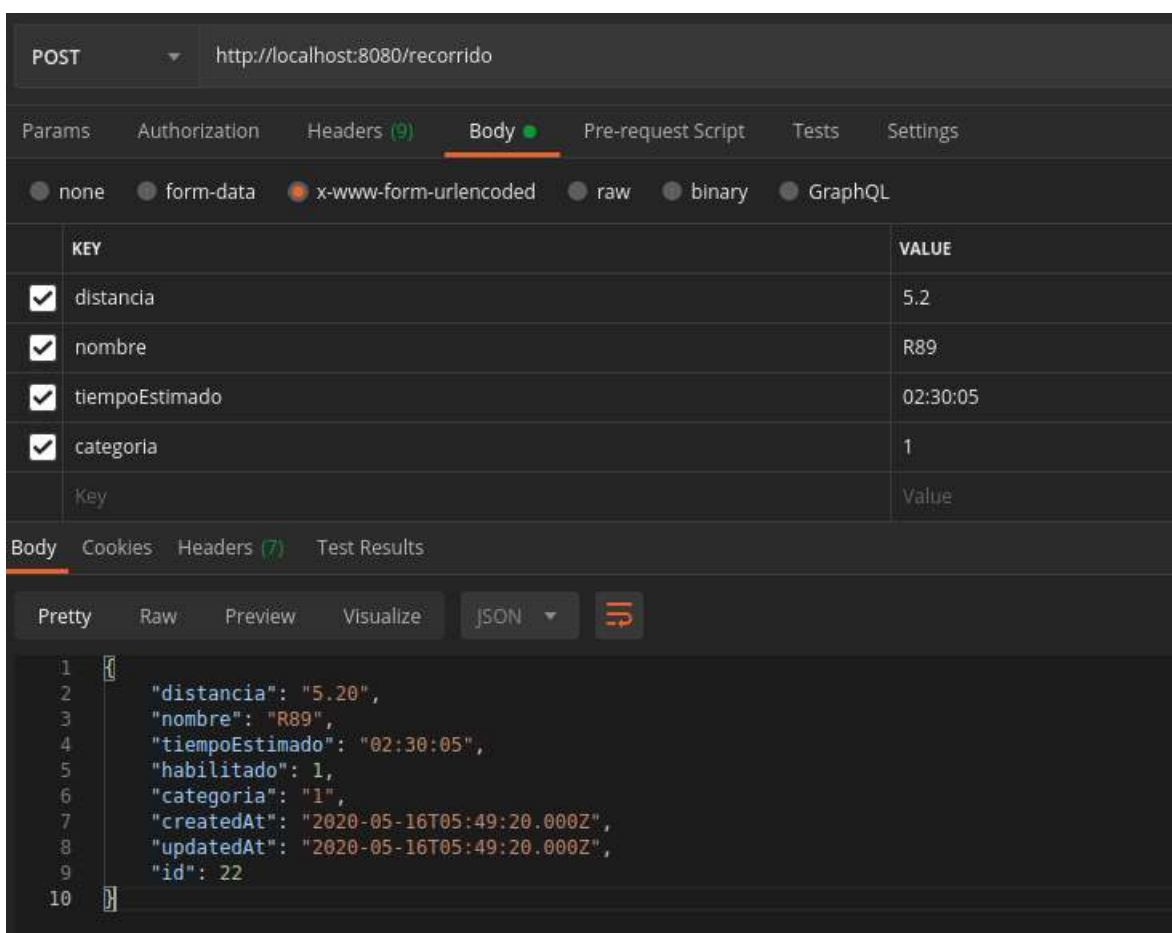


Figura 6.27: Crear un recorrido

❑ **Actualizar un registro:** Se actualiza un registro con el método de petición HTTP PUT, a manera de prueba se hizo sobre el recorrido con id 22 y se actualizó el siguiente valor:

- habilitado: 0


PUT http://localhost:8080/recorrido/22

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

| KEY | VALUE |
|--|----------|
| <input checked="" type="checkbox"/> habilitado | 0 |
| <input type="checkbox"/> nombre | R89 |
| <input type="checkbox"/> tiempoEstimado | 02:30:05 |
| <input type="checkbox"/> categoria | 1 |
| <input type="checkbox"/> distancia | 5.20 |
| Key | Value |

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON 

```
1 {
2   "distancia": "5.20",
3   "nombre": "R89",
4   "tiempoEstimado": "02:30:05",
5   "habilitado": "0",
6   "id": 22,
7   "createdAt": "2020-05-16T05:52:22.000Z",
8   "updatedAt": "2020-05-16T05:52:38.939Z"
9 }
```

Figura 6.28: Actualizar un recorrido

6.8 ANEXO 8: SPRINT BACKLOG SPRINT 2

| | |
|--------------------|---|
| Código | HFW030 |
| Descripción | Como usuario administrador, yo puedo gestionar empresas |
| Prioridad | Media |
| Peso | 5 |
| Tareas | <ul style="list-style-type: none"><input type="checkbox"/> Implementar la navegación entre el menú principal y la ruta gestión de empresas.<input type="checkbox"/> Implementar el formulario para editar una empresa<input type="checkbox"/> Implementar el formulario para crear una sub-empresa.<input type="checkbox"/> Implementar el modal para el formulario.<input type="checkbox"/> Listar mi empresa con un menú con los siguientes elementos:<input type="checkbox"/> Gestión de Vehículos<input type="checkbox"/> Gestión de Rutas de Transporte<input type="checkbox"/> Gestión de Trabajadores<input type="checkbox"/> Cargos de los Trabajadores |

| | |
|---------------------------------------|--|
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar Gestión de Vehículos y navegar a la ruta de gestión de los vehículos de la empresa <input type="checkbox"/> Seleccionar Gestión de Rutas de Transporte y navegar a la ruta de asignación de rutas de transporte a la empresa <input type="checkbox"/> Seleccionar Gestión de Trabajadores y navegar a la ruta de gestión de los trabajadores de la empresa <input type="checkbox"/> Seleccionar Cargos de los Trabajadores y navegar a la ruta de gestión de los trabajadores de la empresa <input type="checkbox"/> Editar mi empresa <input type="checkbox"/> Cambiar el estado (activo o inactivo) de una empresa <input type="checkbox"/> Buscar empresas o sub-empresas por su razón social o RUC. <input type="checkbox"/> Filtrar las empresas por el estado: Activo e Inactivo <input type="checkbox"/> Las sub-empresas se crearán con los siguientes campos: <ul style="list-style-type: none"> • Nombre Comercial (Campo Alfabético, Obligatorio) • Razón Social (Campo Alfabético, Obligatorio) • RUC (Campo Numérico, Obligatorio) • Dirección matriz (Campo Alfanumérico, Obligatorio) • Teléfono (Campo Numérico, Obligatorio) • Correo (Campo Alfanumérico, Obligatorio) • Código (Campo Numérico, Obligatorio) • Tipo de Contribuyente (Campo de selección: Persona natural/Sociedades/Especial, Obligatorio) • Obligado a contabilidad (Campo de selección: Sí o No, Obligatorio) |
|---------------------------------------|--|

Tabla 6.32: Historia de usuario: HFW030

| | |
|--------------------------------|---|
| Código | HFW031 |
| Descripción | Como usuario administrador, yo puedo gestionar vehículos |
| Prioridad | Media |
| Peso | 5 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la navegación entre el menú principal y la ruta gestión de vehículos. <input type="checkbox"/> Implementar el formulario para editar y crear un vehículo. <input type="checkbox"/> Implementar el modal para el formulario de editar y crear. <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros de vehículos por página. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar un vehículo y navegar a su respectiva ruta para gestionar sus dueños <input type="checkbox"/> Crear un Vehículo <input type="checkbox"/> Editar un Vehículo <input type="checkbox"/> Buscar vehículos por placa <input type="checkbox"/> Los vehículos se crearán o editarán con los siguientes campos <ul style="list-style-type: none"> • Marca (Campo Alfabético, Obligatorio) • Placa (Campo Alfabético, Obligatorio) • Kilometraje Actual (Campo Numérico, Obligatorio) |

Tabla 6.33: Historia de usuario: HFW031

| | |
|--------------------|--|
| Código | HFW032 |
| Descripción | Como usuario administrador, yo puedo asignar vehículos a una determinada empresa |
| Prioridad | Media |
| Peso | 5 |

| | |
|---------------------------------------|---|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar el componente de arrastrar y soltar para asignar y desasignar un vehículo de una empresa. <input type="checkbox"/> Implementar el modal para el componente de asignar y desasignar. <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros de vehículos de la empresa por página. |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar Gestión de Horarios de Servicio de un vehículo de una empresa y navegar a su respectiva ruta para gestionar sus horarios de servicio <input type="checkbox"/> Seleccionar Asignación de Trabajadores de un vehículo de una empresa y navegar a su respectiva ruta para gestionar sus trabajadores <input type="checkbox"/> Asignar vehículos a una empresa <input type="checkbox"/> Desasignar vehículos a una empresa <input type="checkbox"/> Buscar los vehículos asignados por placa <input type="checkbox"/> Cambiar el estado (activo o inactivo) de un lugar |

Tabla 6.34: Historia de usuario: HFW032

| | |
|---------------------------|--|
| <p>Código</p> | <p>HFW033</p> |
| <p>Descripción</p> | <p>Como usuario administrador, yo puedo asignar dueños a los vehículos</p> |
| <p>Prioridad</p> | <p>Baja</p> |
| <p>Peso</p> | <p>5</p> |

| | |
|--------------------------------|--|
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar el formulario para asignar un dueño al vehículo. <input type="checkbox"/> Implementar el modal para el formulario de asignación. <input type="checkbox"/> Implementar una tabla con paginación que liste hasta 10 registros de dueños de un vehículo por página. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Asignar dueños a un vehículo del sistema <input type="checkbox"/> Cambiar el estado (activo o inactivo) de un lugar <input type="checkbox"/> Buscar los dueños asignados a un vehículo por su razón social <input type="checkbox"/> Los lugares se crearán o editarán con los siguientes campos <ul style="list-style-type: none"> • Dueño (Campo auto-completador, Obligatorio) |

Tabla 6.35: Historia de usuario: HFW036

| | |
|--------------------|---|
| Código | HFW034 |
| Descripción | Como usuario administrador, yo puedo gestionar horarios para los Recorridos |
| Prioridad | Media |
| Peso | 8 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar el formulario para editar y crear un horario de un recorrido. <input type="checkbox"/> Implementar el modal para el formulario de editar y crear. <input type="checkbox"/> Implementar una tabla con paginación que liste hasta 10 registros de horarios de los recorridos por página. |

| | |
|---------------------------------------|--|
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar una ruta de transporte y listar los recorridos con sus horarios creados para esa ruta <input type="checkbox"/> Crear un Horario de un Recorrido de tipo fecha-hora <input type="checkbox"/> Crear un Horario de un Recorrido de Tipo día <input type="checkbox"/> Editar un Horario de un Recorrido <input type="checkbox"/> Seleccionar los recorridos disponibles de la ruta de transporte seleccionada <input type="checkbox"/> Cambiar el estado (activo o inactivo) de un horario de un recorrido <input type="checkbox"/> Buscar un horario de un recorrido por el nombre del recorrido <input type="checkbox"/> Los horarios para los recorridos se crearán o editarán con los siguientes campos <ul style="list-style-type: none"> • Recorrido (Campo auto-completador, Obligatorio) • Descripción (Campo Alfabético, Obligatorio) • Tipo (Campo de selección: Día o Fecha Hora, Obligatorio) • Fecha de Inicio (Campo de fecha, Opcional) • Fecha de Fin (Campo de fecha, Opcional) • Horario Inicio (Campo de tiempo, Obligatorio) • Horario Fin (Campo de tiempo, Obligatorio) • Lunes (Checkbox, Opcional) • Martes (Checkbox, Opcional) • Miércoles (Checkbox, Opcional) • Jueves (Checkbox, Opcional) • Viernes (Checkbox, Opcional) • Sábado (Checkbox, Opcional) • Domingo (Checkbox, Opcional) |
|---------------------------------------|--|

Tabla 6.36: Historia de usuario: HFW034

| | |
|--------------------------------|--|
| Código | HFW035 |
| Descripción | Como usuario administrador, yo puedo gestionar horarios para los Vehículos |
| Prioridad | Media |
| Peso | 5 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar el formulario para editar y crear un horario de un vehículo. <input type="checkbox"/> Implementar el modal para el formulario de editar y crear. <input type="checkbox"/> Implementar una tabla con paginación que liste hasta 10 registros de horarios de servicio de los vehículos por página. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar un vehículo de una empresa y listar los horarios creados para ese vehículo <input type="checkbox"/> Crear un Horario para un Vehículo de tipo fecha-hora <input type="checkbox"/> Crear un Horario para un Vehículo de Tipo día <input type="checkbox"/> Editar un Horario para un Vehículo <input type="checkbox"/> Cambiar el estado (activo o inactivo) de un horario para un vehículo <input type="checkbox"/> Buscar un horario de un vehículo por la descripción del horario |

| | |
|---------------------------------------|--|
| <p>Criterios de aceptación</p> | <p><input type="checkbox"/> Los horarios para un vehículo se crearán o editarán con los siguientes campos</p> <ul style="list-style-type: none"> • Descripción (Campo Alfabético, Obligatorio) • Tipo (Campo de selección: Día o Fecha Hora, Obligatorio) • Fecha de Inicio (Campo de fecha, Opcional) • Fecha de Fin (Campo de fecha, Opcional) • Horario Inicio (Campo de tiempo, Obligatorio) • Horario Fin (Campo de tiempo, Obligatorio) • Lunes (Checkbox, Opcional) • Martes (Checkbox, Opcional) • Miércoles (Checkbox, Opcional) • Jueves (Checkbox, Opcional) • Viernes (Checkbox, Opcional) • Sábado (Checkbox, Opcional) • Domingo (Checkbox, Opcional) |
|---------------------------------------|--|

Tabla 6.37: Historia de usuario: HFW035

| | |
|---------------------------|--|
| <p>Código</p> | <p>HFW036</p> |
| <p>Descripción</p> | <p>Como usuario administrador, yo puedo gestionar lugares</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>3</p> |
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la navegación entre el menú principal y la ruta gestión de lugares. <input type="checkbox"/> Implementar el formulario para editar y crear un lugar. <input type="checkbox"/> Implementar el modal para el formulario de editar y crear. <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros por página. |

| | |
|--------------------------------|--|
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar un lugar y navegar a su respectiva ruta para gestionar sus paradas <input type="checkbox"/> Seleccionar un lugar y navegar a su respectiva ruta para gestionar sus rutas de transporte <input type="checkbox"/> Crear un Lugar <input type="checkbox"/> Editar un Lugar <input type="checkbox"/> Navegar a la gestión de paradas del lugar <input type="checkbox"/> Navegar a la gestión de rutas de transporte del lugar <input type="checkbox"/> Cambiar el estado (activo o inactivo) de un lugar <input type="checkbox"/> Buscar lugares por nombre <input type="checkbox"/> Filtrar los lugares por el estado: Activo e Inactivo <input type="checkbox"/> Los lugares se crearán o editarán con los siguientes campos <ul style="list-style-type: none"> • Nombre (Campo Alfabético, Obligatorio) • Nivel (Campo Numérico, Obligatorio) • Zona Horaria (Campo Alfanumérico, Obligatorio) • Es ultimo nodo (Campo de selección: Si o No, Obligatorio) • Lugar Padre (Campo auto-completador, Opcional) |
|--------------------------------|--|

Tabla 6.38: Historia de usuario: HFW036

| | |
|--------------------|---|
| Código | HFW037 |
| Descripción | Como usuario Administrador deseo crear paradas de un lugar. |
| Prioridad | Alta |
| Peso | 5 |

| | |
|---------------------------------------|---|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la navegación entre la ruta de gestión lugares y la ruta gestión de paradas. <input type="checkbox"/> Implementar el formulario para editar y crear una parada. <input type="checkbox"/> Implementar el modal para el formulario de editar y crear. <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros por página. |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Listar en una tabla las paradas con el campo nombre. <input type="checkbox"/> Crear una Parada <input type="checkbox"/> Editar una Parada <input type="checkbox"/> Ver los tipos de transporte que pasan por esta parada <input type="checkbox"/> Asignar los tipos de transporte que pasan por esta parada <input type="checkbox"/> Cambiar el estado (activo o inactivo) de una parada <input type="checkbox"/> Buscar paradas por nombre <input type="checkbox"/> Filtrar las paradas por el estado: Activo e Inactivo <input type="checkbox"/> Al crear o editar una parada, permitir al usuario que seleccione la ubicación geográfica de la parada a través de un mapa. <input type="checkbox"/> Las paradas se crearán o editarán con los siguientes campos <ul style="list-style-type: none"> • Nombre (Campo Alfabético, Obligatorio) • Coordenadas Geográficas (Campo Numérico, pero se seleccionará la parada desde el mapa, Obligatorio) |

Tabla 6.39: Historia de usuario: HFW037

| | |
|---------------------------|--|
| <p>Código</p> | <p>HFW038</p> |
| <p>Descripción</p> | <p>Como usuario Administrador deseo crear rutas de transporte.</p> |
| <p>Prioridad</p> | <p>Alta</p> |
| <p>Peso</p> | <p>5</p> |

| | |
|---------------------------------------|---|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la navegación entre la ruta de gestión lugares y la ruta gestión de rutas de transporte. <input type="checkbox"/> Implementar el formulario para editar y crear una ruta de transporte. <input type="checkbox"/> Implementar el modal para el formulario de editar y crear. <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros por página. |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar una ruta de transporte y navegar a su respectiva ruta para gestionar sus paradas. <input type="checkbox"/> Al seleccionar la ruta de transporte, navegar a su respectiva ruta para gestionar sus recorridos. <input type="checkbox"/> Crear una ruta de transporte. <input type="checkbox"/> Editar una ruta de transporte. <input type="checkbox"/> Navegar a la gestión de paradas de una ruta de transporte. <input type="checkbox"/> Navegar a la gestión de recorridos de ruta de transporte de transporte. <input type="checkbox"/> Cambiar el estado (activo o inactivo) de una ruta de transporte. <input type="checkbox"/> Buscar ruta de transporte por nombre. <input type="checkbox"/> Filtrar las rutas de transporte por el estado: Activo e Inactivo. <input type="checkbox"/> Filtrar las rutas de transporte por código (circuito). <input type="checkbox"/> Asignar paradas a la ruta de transporte. <input type="checkbox"/> Las rutas de transporte se crearán o editarán con los siguientes campos <ul style="list-style-type: none"> • Nombre (Campo Alfabético, Obligatorio) • Código (Campo Numérico, Obligatorio) • Tipo de transporte (Campo auto-completador, Obligatorio) |

Tabla 6.40: Historia de usuario: HFW038

| | |
|--------------------------------|--|
| Código | HFW039 |
| Descripción | Como usuario administrador, yo puedo gestionar recorridos para una ruta de transporte |
| Prioridad | Alta |
| Peso | 8 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la navegación entre la ruta de gestión rutas de transporte y la ruta gestión de recorridos. <input type="checkbox"/> Implementar el formulario para editar y crear un recorrido. <input type="checkbox"/> Implementar Modal para el formulario de editar y crear. <input type="checkbox"/> Implementar el modal para dibujar la trayectoria del recorrido en un mapa. <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros por página. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar un recorrido y navegar a su respectiva ruta para gestionar sus paradas-ruta. <input type="checkbox"/> Crear un recorrido. <input type="checkbox"/> Editar un recorrido. <input type="checkbox"/> Navegar a la gestión de paradas-ruta del recorrido. <input type="checkbox"/> Cambiar el estado (activo o inactivo) de un recorrido. <input type="checkbox"/> Buscar recorridos por nombre. |

| | |
|--------------------------------|--|
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Ver en un mapa la trayectoria del recorrido. <input type="checkbox"/> Dibujar en un mapa la trayectoria del recorrido. <input type="checkbox"/> Editar en un mapa la trayectoria del recorrido. <input type="checkbox"/> Los recorridos se crearán o editarán con los siguientes campos <ul style="list-style-type: none"> • Nombre (Campo Alfabético, Obligatorio) • Categoría (Campo auto-completador, Obligatorio) • Tiempo Estimado (Campo Numérico) • Distancia (Campo Numérico) |
|--------------------------------|--|

Tabla 6.41: Historia de usuario: HFW039

| | |
|--------------------|--|
| Código | HFW040 |
| Descripción | Como usuario administrador, yo puedo gestionar Paradas-Ruta de un recorrido específico |
| Prioridad | Alta |
| Peso | 8 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la navegación entre la ruta de gestión de recorridos y la ruta gestión de paradas-ruta. <input type="checkbox"/> Implementar Modal para asignar una parada-ruta (parada asignada a una ruta de transporte). <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros por página. |

| | |
|---------------------------------------|--|
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar un recorrido y navegar a su respectiva ruta para gestionar sus paradas-ruta. <input type="checkbox"/> Seleccionar una parada-ruta y des-asignarla del recorrido. <input type="checkbox"/> Cambiar el estado (activo o inactivo) de una parada-ruta. <input type="checkbox"/> Buscar paradas-ruta por nombre. <input type="checkbox"/> Gestionar el orden de las paradas-ruta del recorrido. <input type="checkbox"/> Asignar y Des-asignar paradas-ruta al recorrido de forma masiva. <input type="checkbox"/> La asignación de paradas-ruta se realizará de la siguiente manera. <ul style="list-style-type: none"> • Se presentarán 2 listas: <ul style="list-style-type: none"> ◦ En la primera lista estarán las paradas-ruta no asignadas ◦ En la segunda lista estarán las paradas-ruta asignadas. • Las paradas-ruta asignadas que se muevan a la primera lista serán des-asignadas al recorrido. • Las paradas-ruta des-asignadas que se muevan a la segunda lista serán asignadas al recorrido. |
|---------------------------------------|--|

Tabla 6.42: Historia de usuario: HFW040

| | |
|--------------------|--|
| Código | HFW041 |
| Descripción | Como usuario administrador, yo puedo gestionar paradas de una ruta de transporte |
| Prioridad | Alta |
| Peso | 5 |

| | |
|---------------------------------------|---|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la navegación entre la ruta de gestión rutas de transporte y la ruta de gestión de paradas de una ruta de transporte. <input type="checkbox"/> Implementar Modal para asignar una parada a una determinada ruta de transporte. <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros por página. |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Seleccionar un recorrido y navegar a su respectiva ruta para gestionar sus paradas-ruta. <input type="checkbox"/> Seleccionar una parada-ruta y des-asignarla de la misma ruta de transporte. <input type="checkbox"/> Cambiar el estado (activo o inactivo) de una parada de la ruta de transporte. <input type="checkbox"/> Buscar paradas de la ruta de transporte por su nombre <input type="checkbox"/> Asignar paradas a la ruta de transporte de manera múltiple. <input type="checkbox"/> Ver en un mapa las paradas asignadas a la ruta de transporte. |

Tabla 6.43: Historia de usuario: HFW041

| | |
|---------------------------|--|
| <p>Código</p> | <p>HFW042</p> |
| <p>Descripción</p> | <p>Como usuario administrador, yo puedo gestionar categorías</p> |
| <p>Prioridad</p> | <p>Baja</p> |
| <p>Peso</p> | <p>2</p> |

| | |
|---------------------------------------|--|
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la navegación entre la ruta del menú principal y la ruta de gestión de categorías. <input type="checkbox"/> Implementar el formulario para editar y crear una categoría. <input type="checkbox"/> Implementar el modal para el formulario de editar y crear. <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros por página. |
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Crear una Categoría. <input type="checkbox"/> Editar una Categoría <input type="checkbox"/> Cambiar el estado (activo o inactivo) de una categoría. <input type="checkbox"/> Buscar categorías por nombre. <input type="checkbox"/> Las categorías se crearán o editarán con el siguiente campo: <ul style="list-style-type: none"> • Nombre (Campo Alfabético, Obligatorio). |

Tabla 6.44: Historia de usuario: HFW042

| | |
|---------------------------|---|
| <p>Código</p> | <p>HFW043</p> |
| <p>Descripción</p> | <p>Como usuario administrador, yo puedo gestionar tipos de transporte</p> |
| <p>Prioridad</p> | <p>Baja</p> |
| <p>Peso</p> | <p>2</p> |
| <p>Tareas</p> | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la navegación entre la ruta del menú principal y la ruta de gestión de tipos de transporte. <input type="checkbox"/> Implementar el formulario para editar y crear un tipo de transporte. <input type="checkbox"/> Implementar el modal para el formulario de editar y crear. <input type="checkbox"/> Implementar una tabla con paginación de que liste 10 registros por página. |

| | |
|--------------------------------|---|
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Crear un tipo de transporte. <input type="checkbox"/> Editar un tipo de transporte. <input type="checkbox"/> Cambiar el estado (activo o inactivo) de un tipo de transporte. <input type="checkbox"/> Buscar tipos de transporte por nombre. <input type="checkbox"/> Los tipos de transporte se crearán o editarán con el siguiente campo: <ul style="list-style-type: none"> • Nombre (Campo Alfabético, Obligatorio). |
|--------------------------------|---|

Tabla 6.45: Historia de usuario: HFW043

6.9 ANEXO 9: SPRINT BACKLOG SPRINT 3

| | |
|--------------------------------|---|
| Código | HFM044 |
| Descripción | Como usuario, yo puedo registrarme en el sistema |
| Prioridad | Media |
| Peso | 5 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar el formulario para registro de usuarios. <input type="checkbox"/> Implementar la pantalla o actividad para registrar usuarios. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Los usuarios se registrarán con los siguientes campos: <ul style="list-style-type: none"> • Nombres (Campo Alfabético, Obligatorio) • Apellidos (Campo Alfabético, Obligatorio) • Correo (Campo Alfanumérico, Obligatorio) • Identificación (Campo Numérico, Obligatorio) • Contraseña (Obligatorio) |

Tabla 6.46: Historia de usuario: HFM044

| | |
|--------------------------------|--|
| Código | HFM045 |
| Descripción | Como usuario registrado. yo puedo iniciar y cerrar sesión |
| Prioridad | Baja |
| Peso | 3 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la funcionalidad para ingresar al sistema. <input type="checkbox"/> Implementar la pantalla o actividad para ingresar al sistema. <input type="checkbox"/> Implementar la funcionalidad para cerrar sesión del sistema. <input type="checkbox"/> Implementar la navegación entre la pantalla de registro e ingreso al sistema. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> El formulario para ingresar al sistema contará con los siguientes campos: <ul style="list-style-type: none"> • Correo (Campo Alfabético, Obligatorio) • Contraseña (Campo Alfabético, Obligatorio) <input type="checkbox"/> Botón de cerrar sesión con una alerta de confirmación |

Tabla 6.47: Historia de usuario: HFM045

| | |
|--------------------|---|
| Código | HFM046 |
| Descripción | Como usuario, yo puedo visualizar paradas en un mapa |
| Prioridad | Alta |
| Peso | 5 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar el mapa de las paradas en un componente. <input type="checkbox"/> Implementar la pantalla o actividad para ver el mapa. |

| | |
|--------------------------------|---|
| Criterios de aceptación | <input type="checkbox"/> Las paradas se presentarán en el mapa con un icono característico y con el respectivo nombre de la parada. |
|--------------------------------|---|

Tabla 6.48: Historia de usuario: HFM046

| | |
|--------------------------------|---|
| Código | HFM047 |
| Descripción | Como usuario registrado, yo puedo buscar paradas |
| Prioridad | Baja |
| Peso | 5 |
| Tareas | <input type="checkbox"/> Implementar el componente autocompletador y conectarlo con el servicio REST respectivo para buscar paradas por nombre. |
| Criterios de aceptación | <input type="checkbox"/> Al buscar una parada, la misma debe ser ubicada en el mapa. |

Tabla 6.49: Historia de usuario: HFM047

| | |
|--------------------|--|
| Código | HFM048 |
| Descripción | Como usuario registrado, yo puedo ver las rutas de una parada determinada |
| Prioridad | Alta |
| Peso | 3 |
| Tareas | <input type="checkbox"/> Implementar el modal con una lista para visualizar las rutas que pasen por la parada. |

| | |
|--------------------------------|--|
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Al seleccionar una parada en el mapa, mostrar el modal con sus respectivas rutas en una lista. <input type="checkbox"/> Las rutas se mostrarán por su nombre en la lista mencionada anteriormente. |
|--------------------------------|--|

Tabla 6.50: Historia de usuario: HFM048

| | |
|--------------------------------|---|
| Código | HFM049 |
| Descripción | Como usuario registrado, yo puedo agregar una parada a mis paradas favoritas |
| Prioridad | Alta |
| Peso | 3 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar botón para agregar y quitar la parada de favoritos en el modal descrito en la historia “HFM048”. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Al seleccionar una parada en el mapa, mostrar además de las rutas. |

Tabla 6.51: Historia de usuario: HFM049

| | |
|--------------------|---|
| Código | HFM050 |
| Descripción | Como usuario registrado, yo puedo gestionar mis paradas favoritas |
| Prioridad | Alta |
| Peso | 3 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la pantalla o actividad para visualizar las paradas favoritas del usuario. <input type="checkbox"/> Desarrollar una lista paradas favoritas en un componente. |

| | |
|--------------------------------|---|
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Cada parada en la lista tendrá las siguientes acciones: <ul style="list-style-type: none"> • Quitar la parada de favoritos • Ver la parada en el mapa, es decir que el usuario sera llevado a la pantalla del mapa de paradas y la parada en cuestión será mostrada en el mapa. |
|--------------------------------|---|

Tabla 6.52: Historia de usuario: HFM051

| | |
|--------------------------------|--|
| Código | HNIOT051 |
| Descripción | Permitir que el "Raspberry PI" tome fotografías y las envíe al servidor web central en un periodo de tiempo determinado |
| Prioridad | Alta |
| Peso | 8 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar el programa encargado de manejar toda la lógica del dispositivo IOT. <input type="checkbox"/> Dentro del programa implementar un cliente HTTP para consultar el intervalo de tiempo en que se emitirá la información recopilada por el dispositivo. <input type="checkbox"/> Implementar el cliente para websockets con el fin de estar atento a qué recorrido emitir la información. <input type="checkbox"/> Implementar el cliente MQTT con el fin de enviar las fotografías al servidor. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Usar un intervalo de tiempo por defecto si la consulta realizada por el cliente HTTP falla. <input type="checkbox"/> Las imágenes o fotografías serán enviadas al servidor en el formato "Base64". |

Tabla 6.53: Historia de usuario: HNIOT051

| | |
|--------------------------------|---|
| Código | HFM052 |
| Descripción | Como usuario registrado, yo puedo visualizar en un mapa las unidades de transporte de una ruta de transporte en específico |
| Prioridad | Alta |
| Peso | 8 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la pantalla o actividad para visualizar en un mapa las unidades de transporte. <input type="checkbox"/> Implementar el modal para seleccionar el recorrido que voy a escuchar. <input type="checkbox"/> Implementar el mapa de recorridos en un componente. <input type="checkbox"/> Implementar el cliente de websockets para escuchar las unidades de transporte en base al recorrido seleccionado. |
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Al seleccionar el recorrido en el modal, se debe dibujar en el mapa de recorridos el camino que el recorrido debe cumplir. <input type="checkbox"/> Representar en el mismo mapa las unidades de transporte que se estén escuchando. Cada unidad de transporte será representada con un icono determinado y con su número de placa. |

Tabla 6.54: Historia de usuario: HFM052

| | |
|--------------------|---|
| Código | HFM053 |
| Descripción | Como usuario registrado, yo puedo visualizar la información de una determinada unidad de transporte |
| Prioridad | Alta |
| Peso | 8 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar la pantalla o actividad para visualizar la información de la unidad de transporte. |

| | |
|--------------------------------|--|
| Criterios de aceptación | <ul style="list-style-type: none"> <input type="checkbox"/> Las fotografías del interior de la unidad de transporte serán mostradas a través de un componente tipo “carrusel”. <input type="checkbox"/> Mostrar la velocidad de la unidad de transporte. <input type="checkbox"/> Mostrar el nombre conductor de la unidad de transporte. |
|--------------------------------|--|

Tabla 6.55: Historia de usuario: HFM053

| | |
|--------------------|---|
| Código | HFM054 |
| Descripción | Como usuario trabajador registrado, yo puedo iniciar un recorrido |
| Prioridad | Alta |
| Peso | 8 |
| Tareas | <ul style="list-style-type: none"> <input type="checkbox"/> Implementar el formulario para crear una cabecera de recorrido trabajo. <input type="checkbox"/> Implementar el servicio de emisión de los datos de geolocalización y velocidad de la unidad de transporte. <input type="checkbox"/> Implementar el servicio de websockets para emitir los datos de la unidad de transporte y del conductor. |

| | |
|---------------------------------------|--|
| <p>Criterios de aceptación</p> | <ul style="list-style-type: none"> ❑ El formulario constará de los siguientes campos: <ul style="list-style-type: none"> ● Unidad de transporte del conductor (Campo de selección, Obligatorio) ● Ruta de la unidad de transporte (Campo de selección, Obligatorio) ● Recorrido y Horario de la unidad de transporte (Campo de selección, Obligatorio) ❑ Emisión de la geolocalización y la velocidad del vehículo en segundo plano. ❑ Los datos a emitir por websockets son: <ul style="list-style-type: none"> ● Datos del conductor: nombres, apellidos, cédula y correo electrónico. ● Latitud, longitud, velocidad y placa del vehículo. ● Identificador de cabecera de recorrido de trabajo. ● Identificador de horario recorrido. |
|---------------------------------------|--|

Tabla 6.56: Historia de usuario: HFM054