

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**Desarrollo de una Aplicación Móvil y Web para visualizar la
Disponibilidad de Parqueaderos en Tiempo Real para el
Estacionamiento del Centro de Educación Continua de la Escuela
Politécnica Nacional.**

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

KEVIN FABRICIO CARATE CABRERA

kevin.carate@epn.edu.ec

RONNY SANTIAGO CABRERA CABRERA

ronny.cabrera@epn.edu.ec

DIRECTOR: Dr. TANIA ELIZABETH CALLE JIMÉNEZ


tania.calle@epn.edu.ec

Quito, Junio 2020

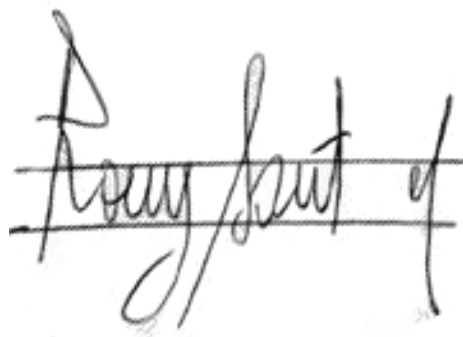
DECLARACIÓN

Nosotros, Kevin Fabricio Cárate Cabrera y Ronny Santiago Cabrera Cabrera, declaro 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 he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis 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.



Kevin Fabricio Cárate Cabrera



Ronny Santiago Cabrera Cabrera

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Kevin Fabricio Cárate Cabrera y Ronny Santiago Cabrera Cabrera, bajo mi supervisión.



Dr. TANIA ELIZABETH CALLE JIMÉNEZ

AGRADECIMIENTOS

A Dios, por darme salud y vida para poder llegar a este momento.

A mis padres, por el apoyo y esfuerzo constante, siempre pendiente de que nunca me falte lo necesario para seguir estudiando cada día, por los consejos dados en momentos difíciles y sobre todo por la paciencia que me han tenido durante toda mi vida académica, no me alcanzaré la vida para terminar de agradecer todo lo que han hecho por mí.

A mis hermanas, Andrea mi mayor inspiración para no desmayar en los momentos difíciles, Valery siempre formando parte de mis locuras para sacar una sonrisa a la familia, sin duda junto a mis padres son el motor para cada día ser mejor.

A mis tíos, primos y demás familia, por cada consejo y apoyo entregado cuando necesite de alguno de ustedes.

A la Escuela Politécnica Nacional por abrirme las puertas al conocimiento y permitirme ser un profesional.

A mis amigos y compañeros, por todas las experiencias vividas que me llevo en mi estadía por la universidad, hicieron que cada momento valiera la pena.

A Ronny, gracias por creer y confiar en mí, sin duda nos llevamos buenos recuerdos durante todo este último esfuerzo, de todo corazón te deseo éxitos en tu carrera profesional, espero volver a crear nuevas cosas junto a ti, gracias.

Por último y no menos importante a PhD. Tania Calle por confiar en nosotros, darnos la oportunidad y guiarnos en el desarrollo de este proyecto.

Kevin Cárate

DEDICATORIA

A la memoria de mi abuelito Segundo, quién, desde que yo era pequeño me enseñó el valor de la perseverancia. A la memoria de mi abuelita Anita, con quién compartí gran parte de mi vida y de quién tuve tantas enseñanzas; Espero que la humildad y el profundo amor que los caracterizaba me acompañen siempre.

“Ningún nieto tuvo mejores abuelos”

Kevin Cárate

AGRADECIMIENTOS

A la vida por permitirme tener a los mejores padres que un hijo pueda querer.

A mis padres que ante cualquier circunstancia adversa siempre han estado allí para mí, quienes han sido pilares fundamentales en mi vida, por inculcarme valores, por enseñarme a ser generoso, por ser pacientes durante toda mi vida y darme fuerzas para alcanzar mis metas.

A mis hermanos Alexis y Roberto por enseñarme que hay varios caminos para salir adelante y cumplir sus metas, por cuidarme y ayudarme, por estar ahí cuando necesite de un consejo o tomar un respiro del duro esfuerzo que la universidad requiere.

A mis amigos y compañeros, por todas las vivencias que pasamos a lo largo de nuestra vida universitaria y especialmente a aquellos con quien estoy creciendo laboralmente.

A nuestra tutora la PhD. Tania Calle por las palabras de aliento, por tenernos paciencia y convicción a lo largo de este proyecto de titulación y ayudarnos para poder culminar este gran paso en nuestras vidas.

Por último y no menos importante a mi amigo Kevin, por acompañarme y compartir sus conocimientos durante este proyecto, por confiar en mí para concluir esta etapa de nuestras vidas, confío que vas a crecer laboralmente y espero al igual que tu desarrollar nuevos proyectos junto a ti, gracias totales.

Ronny Cabrera

DEDICATORIA

Este esfuerzo tan grande es dedicado a mis padres, Rosa y Emilio, mis hermanos Alexis y Roberto. A la memoria de mi abuelito Teófilo quien fue un referente de respeto y trabajo. A la memoria de mi abuelita Maruja quien fue un referente de esfuerzo y humildad.

Ronny Cabrera

CONTENIDO

RESUMEN.....	xv
ABSTRACT	xvi
1. INTRODUCCIÓN.....	1
1.1 OBJETIVOS	2
1.1.1 Objetivo General	2
1.1.2 Objetivos Específicos	2
1.2 ALCANCE	2
1.3 MARCO DE REFERENCIA	3
1.3.1 Estado Actual de los Parqueaderos de la EPN.....	3
1.3.2 Aplicaciones Móviles	4
1.3.3 Servicios Web	5
1.3.4 Aplicaciones Web.....	6
1.3.5 Sensores.....	7
2. METODOLOGÍA.....	11
2.1 Comparación Scrum y XP.....	11
2.2 Aplicación de la Metodología	12
2.2.1 Equipo Scrum y roles	12
2.2.2 Eventos	13
2.2.3 Artefactos:.....	13
2.3 Arquitectura	14
2.4 Lenguajes y Herramientas de Programación	16
3. DESARROLLO DE LA APLICACIÓN	19
3.1 Roles	19
3.2 Requerimientos de la Aplicación	19
3.2.1 Contenido del Product Backlog	20

3.2.2	Product Backlog	21
3.3.	Release Plan	23
3.4	Sprint 0	24
3.4.1	Planificación	24
3.4.2	Sprint Review	24
3.4.3	Sprint Retrospective	25
3.5	Sprint 1	25
3.5.1	Planificación	25
3.5.2	Sprint Review	34
3.5.3	Sprint Retrospective	35
3.6	Sprint 2	36
3.6.1	Planificación	36
3.6.2	Sprint Review	42
3.6.3	Sprint Retrospective	43
3.7	Sprint 3	43
3.7.1	Planificación	43
3.7.2	Sprint Review	49
3.7.3	Sprint Retrospective	51
3.8	Sprint 4	51
3.8.1	Planificación	51
3.8.2	Sprint Review	58
3.8.3	Sprint Retrospective	59
3.9	Sprint 5	59
3.9.1	Planificación	59
3.9.2	Sprint Review	68
3.9.3	Sprint Retrospective	70
4.	PRUEBAS DE USABILIDAD Y RENDIMIENTO	71
4.1	Pruebas de Usabilidad	71
4.1.1	Pruebas de Remotas para Aplicación Móvil	71
4.1.2	Estructura de la Encuesta para Aplicación Móvil	72

4.1.3	Análisis de Encuestas para Aplicación Móvil	72
4.1.4	Pruebas en Tiempo Real.....	76
4.2	Pruebas de Rendimiento	78
4.2.1	Evaluación de rendimiento del Backend.....	79
4.2.2	Evaluación de rendimiento de Aplicación Web	85
4.2.3	Evaluación de rendimiento de Aplicación Móvil	90
4.3	Discusión de Resultados	98
5.	CONCLUSIONES Y RECOMENDACIONES.....	101
5.1	Conclusiones	101
5.2	Recomendaciones	102
6.	REFERENCIAS BIBLIOGRÁFICAS.....	104
7.	ANEXOS	112
7.1	Anexo 1 – Pila del Producto	112
7.2	Anexo 2 – Mockups.....	112
7.3	Anexo 3 – Base de Datos.....	112
7.4	Anexo 4 – Gráficos Burndown	112
7.5	Anexo 5 – Encuestas.....	112
7.6	Anexo 6 – Pruebas.....	112
7.7	Anexo 7 – Manuales.....	112
7.8	Anexo 8 – Scripts	112

ÍNDICE DE FIGURAS

Figura 1. Sensor Ultrasonido HC-SR04.	9
Figura 2. Raspberry Pi 2 Modelo B..	10
Figura 3. Patrón de diseño MVC.....	15
Figura 4. Arquitectura General MVC.....	16
Figura 5. Modelo de la Base de Datos.	29
Figura 6. Ejecución Backend.	30
Figura 7. Visualización de Módulos.....	31
Figura 8. Visualización de Mapa y Marcadores.....	32
Figura 9. Visualización Datos Parqueadero CEC-EPN.	33
Figura 10. Visualización Datos Parqueadero FIEE-EPN.	33
Figura 11. Gráfico Burndown Sprint 1.	35
Figura 12. Visualización de Lista de Parqueaderos.	39
Figura 13. Búsqueda de un parqueadero.....	39
Figura 14. Visualización de Información del Parqueaderos.....	40
Figura 15. Visualización de Subsuelo 1 del Parqueadero del CEC.	40
Figura 16. Visualización de Subsuelo 2 del Parqueadero del CEC.	41
Figura 17. Visualización de una Ruta.....	41
Figura 18. Gráfico Burndown Sprint 2.	42
Figura 19. Interfaz Registro de Usuario Administrador.....	46
Figura 20. Interfaz de Acceso.	47
Figura 21. Interfaz de Registro de Usuarios para determinados parqueaderos.....	47
Figura 22. Interfaz Recuperación de Contraseña.....	48
Figura 23. Interfaz Visualización de Datos de Usuario.	48
Figura 24. Interfaz Actualización de Datos de Usuario.....	49
Figura 25. Gráfico Burndown Sprint 3.	50
Figura 26. Interfaz Registro de Parqueaderos.....	54
Figura 27. Interfaz Registro de Horarios.	54
Figura 28. Interfaz Visualización de Parqueaderos.	55
Figura 29. Interfaz Visualización Datos Parqueaderos.....	55
Figura 30. Interfaz Actualización de Datos.....	56
Figura 31. Interfaz Actualización de Horarios.....	56
Figura 32. Interfaz Registro de Plazas.	57

Figura 33. Interfaz de Asignación de Tipo de Plaza.	57
Figura 34. Gráfico Burndown Sprint 4.	59
Figura 35. Información Acerca de la aplicación móvil.....	63
Figura 36. Información de Ayuda para la aplicación móvil.	64
Figura 37. Interfaz de Información de una determinada Plaza.	64
Figura 38. Interfaz de Actualización de Datos de una Plaza.	65
Figura 39. Interfaz Perfil de Usuario.....	65
Figura 40. Interfaz Actualización de Datos de Usuario.....	66
Figura 41. Interfaz Actualización de Contraseña de Usuario.....	66
Figura 42. Interfaz Información de Aplicativo Web.....	67
Figura 43. Ejecución Script Sensor Ultrasonido HC-SR04.....	67
Figura 44. Conexión del Subsistema para Obtener Disponibilidad.....	68
Figura 45. Gráfico Burndown Sprint 5.....	69
Figura 46. Pregunta 1 Encuesta de Usabilidad.....	73
Figura 47. Pregunta 2 Encuesta de Usabilidad.....	73
Figura 48 Pregunta 3 Encuesta de Usabilidad.....	74
Figura 49 Pregunta 4 Encuesta de Usabilidad.....	74
Figura 50 Pregunta 5 Encuesta de Usabilidad.....	75
Figura 51 Pregunta 6 Encuesta de Usabilidad.....	75
Figura 52 Pregunta 7 Encuesta de Usabilidad.....	76
Figura 53. Instalación del Subsistema para Obtener Disponibilidad.....	77
Figura 54. Funcionamiento de la Aplicación Park EPN en Tiempo Real.....	77
Figura 55. Definición de Parámetros de Consulta.....	80
Figura 56. Configuración Escenario Uno.....	80
Figura 57. Configuración Escenario Dos.....	82
Figura 58. Configuración Escenario Tres.....	83
Figura 59. Configuración Escenario Cuatro.....	84
Figura 60. Ejecución y Grabación de Rendimiento Chrome.....	86
Figura 61. Ejecución y Grabación de Rendimiento Firefox.....	86
Figura 62. Ejecución y Grabación de Rendimiento Opera.....	87
Figura 63 Análisis Rendimiento Herramienta GTmetrix.....	88
Figura 64 Análisis Dispositivo Móvil Herramienta PageSpeed.....	89
Figura 65 Análisis Dispositivo de Escritorio Herramienta PageSpeed.....	90

Figura 66. Monitoreo de Uso de CPU para Celular Gama Alta.	91
Figura 67. Pico más alto de Uso de CPU para Celular Gama Alta.	91
Figura 68. Monitoreo de Uso de Memoria para Celular Gama Alta.	92
Figura 69. Pico más alto de Uso de Memoria para Celular Gama Alta.	93
Figura 70. Monitoreo de Uso de Red para Celular Gama Alta.	93
Figura 71. Pico más alto de Uso de Red para Celular Gama Alta.	94
Figura 72. Monitoreo de Uso de CPU para Celular Gama Baja.	95
Figura 73. Pico más alto de Uso de CPU para Celular Gama Baja.	95
Figura 74. Monitoreo de Uso de Memoria para Celular Gama Baja.	96
Figura 75. Pico más alto de Uso de Memoria para Celular Gama Baja.	96
Figura 76. Monitoreo de Uso de Red para Celular Gama Baja.	97
Figura 77. Pico más alto de Uso de Red para Celular Gama Baja.	98

ÍNDICE DE TABLAS

Tabla 1 Distribución Parquederos de la EPN.	4
Tabla 2 Usuarios Globales de Smartphones por Región 2016-2021.....	5
Tabla 3 Top 5 de Marcas de Smartphones por Dispositivos Activos Mensuales Junio 2018...5	
Tabla 4 Diferencias, ventajas y desventajas entre aplicaciones de escritorio y las aplicaciones web.	7
Tabla 5 Especificaciones Técnicas Sensor Ultrasonido HC-SR04.	9
Tabla 6 Especificaciones Técnicas Raspberry Pi Modelo B.....	10
Tabla 7 Comparativa Metodologías Ágiles.....	12
Tabla 8 Roles de Equipo Scrum.	12
Tabla 9 Lenguajes y herramientas de programación.....	18
Tabla 10 Roles Scrum del Proyecto.....	19
Tabla 11 Puntos de Historia.....	21
Tabla 12 Product Backlog.....	22
Tabla 13 Descripción Historias de Usuario.	23
Tabla 14 Release Plan.....	24
Tabla 15 Historia de Usuario 1.1.....	26
Tabla 16 Historia de Usuario 1.2.....	26
Tabla 17 Historia de Usuario 1.3.....	27
Tabla 18 Sprint Backlog Sprint 1.....	28
Tabla 19 Sprint Review 1.....	34
Tabla 20 Historia de Usuario 2.1.....	36
Tabla 21 Historia de Usuario 2.2.....	36
Tabla 22 Historia de Usuario 3.1.....	37
Tabla 23 Sprint Backlog Sprint 2.....	38
Tabla 24 Sprint Review 2.....	42
Tabla 25 Historia de Usuario 5.1.....	44
Tabla 26 Historia de Usuario 5.2.....	44
Tabla 27 Historia de Usuario 5.3.....	44
Tabla 28 Historia de Usuario 7.1.....	45
Tabla 29 Sprint Backlog Sprint 3.....	46
Tabla 30 Sprint Review 3.....	50
Tabla 31 Historia de Usuario 6.1.....	51

Tabla 32 Historia de Usuario 8.1.....	52
Tabla 33 Historia de Usuario 8.3.....	52
Tabla 34 Sprint Backlog Sprint 4.....	53
Tabla 35 Sprint Review 4.....	58
Tabla 36 Historia de Usuario 1.4.....	60
Tabla 37 Historia de Usuario 4.1.....	60
Tabla 38 Historia de Usuario 7.2.....	61
Tabla 39 Historia de Usuario 8.2.....	61
Tabla 40 Historia de Usuario 9.1.....	61
Tabla 41 Sprint Backlog Sprint 5.....	63
Tabla 42 Sprint Review 5.....	69
Tabla 43 Estructura de la Encuesta para Aplicación Móvil.....	72
Tabla 44 Cuadro comparativo de herramientas para pruebas de carga y estrés.....	79
Tabla 45 Resultados de la ejecución del Escenario 1.....	81
Tabla 46 Resultados de la ejecución del Escenario 2.....	82
Tabla 47 Resultados de la ejecución del Escenario 3.....	84
Tabla 48 Resultados de la ejecución del Escenario 4.....	85
Tabla 49 Resultados Rendimiento Aplicación Web.....	87

RESUMEN

De acuerdo con estudios realizados por el INEC y ANT un problema a nivel nacional y específicamente de la ciudad de Quito es el crecimiento del parque automotor dificultando la posibilidad de encontrar un lugar para estacionarse. Por tal motivo, el presente proyecto se enfoca en realizar un prototipo que brinde una solución tecnológica en tiempo real para obtener información sobre la disponibilidad de plazas de un determinado parqueadero, utilizando una aplicación móvil y web. Como caso de estudio se utilizó el estacionamiento del Centro de Educación Continua (CEC).

La aplicación móvil ofrece a los usuarios la opción de visualizar los parqueaderos en un mapa interactivo y acceder a información relacionada a un parqueadero. Por otro lado, la aplicación web sirve a los usuarios que administran el parqueadero como una herramienta de gestión, permitiéndole al usuario administrar los parqueaderos para ser visualizados en el aplicativo móvil.

El proyecto fue desarrollado utilizando la metodología ágil SCRUM, la misma que permitió gestionar el desarrollo tanto de la aplicación móvil como web, dividiéndolo en cinco sprints de un mes cada uno y verificando sus resultados por medio del gráfico Burndown por cada sprint.

Se realizaron pruebas de rendimiento al Backend de la aplicación móvil y web para analizar el consumo de recursos necesarios para su funcionamiento. Finalmente, se realizaron pruebas de usabilidad por medio de encuestas, las mismas que muestran resultados satisfactorios conforme a su diseño, uso y funcionalidad.

Palabras clave: Parqueadero, Tiempo Real, SCRUM, Aplicación Móvil, Aplicación Web.

ABSTRACT

According to studies conducted out by INEC and ANT a problem at a national level and specifically in the city of Quito is the growth of the vehicle fleet making it difficult to find a place to park, therefore, this project focuses on making a prototype that provides a technological solution in real time to get information about the availability of places in a given parking, using a mobile application and web. As a case study, the parking of the Centro de Educación Continua (CEC) was used.

The mobile application offers users the option to visualize the parking lots on an interactive map and access related information related to a parking lot. Furthermore, the web application serves users who manage the parking lot as a management tool, allowing the user to manage the parking lots to be visualized in the mobile application.

The project was developed using the agile methodology SCRUM, which allowed managing the development of both the mobile and web application, dividing it into five sprints of one month each and verifying their results through the Burndown graph for each sprint.

Performance tests were carried out on the Backend mobile applications and web to analyze the consumption of resources necessary for its operation and finally usability tests were carried out by means of online surveys, which show satisfactory results according to their design, use and functionality

Key words: Parking, Real Time, SCRUM, Mobile Application, Web Application

1. INTRODUCCIÓN

En la provincia de Pichincha el parque automotor, según cifras del INEC y ANT, en el año 2016 tuvo 733.269 vehículos, para el 2017 este número descendió a 511.782 vehículos, pero existió un crecimiento del 8.8% a nivel nacional, esto implica que existe un aumento de la cantidad de vehículos que circulan en todo el país [1], [2].

Por otro lado, la mayor cantidad de la población de la ciudad tiende a movilizarse a la región centro-norte de la capital, debido a que, en esta zona denominada Hipercentro, existe gran cantidad de equipamientos urbanos, públicos y privados [3]. Debido al crecimiento del parque automotor, la población y la gran concentración de movilidad hacia el Hipercentro de la ciudad, existe la necesidad y una gran demanda de sitios de estacionamiento públicos y privados, principalmente en sectores de gran concentración de actividades.

Para dar solución a esta gran demanda de sitios de estacionamientos la Empresa Pública Metropolitana de Movilidad y Obras Públicas (EPMOP), cuenta con estacionamientos públicos no guiados como la Zona Azul [4], estacionamientos municipales guiados y estacionamientos en parques grandes de la capital [5]. También existen gran cantidad de parqueaderos privados, regados por todo el Hipercentro de la ciudad, los mismos que brindan servicio a todo público en general, así también como a grupos determinados de personas.

En muchas ocasiones varios de estos estacionamientos ya han sido ocupados y no se tiene la certeza de en qué momento serán desocupados. Por tal motivo, la tecnología en tiempo real puede ofrecernos una solución para realizar una correcta administración de los distintos puntos existentes dentro de los estacionamientos, en aquellos que son guiados.

El presente Proyecto de Titulación se encuentra enfocado en el estacionamiento del Centro de Educación Continua (CEC) que forma parte de los 17 parqueaderos de la Escuela Politécnica Nacional (EPN), estos son utilizados por cualquier persona que pertenezca a la EPN y al CEC, y cuenta con un sistema gestionado por la Dirección de Gestión de la Información y Procesos (DGIP) [6].

Sin embargo, el sistema actual del parqueadero del CEC no brinda información a los usuarios de la disponibilidad de parqueaderos. En este sentido se plantea desarrollar un sistema automatizado que permita informar a los usuarios de la disponibilidad en tiempo real de cada una de las plazas del parqueadero en una aplicación móvil, sirviendo como prototipo para cualquier parqueadero sea este municipal, público o privado.

1.1 OBJETIVOS

1.1.1 Objetivo General

Desarrollar una aplicación móvil y web para visualizar la disponibilidad de parqueaderos en tiempo real para el estacionamiento del Centro de Educación Continua de la Escuela Politécnica Nacional.

1.1.2 Objetivos Específicos

- Levantar los requisitos para la aplicación móvil y web, investigando el funcionamiento del estacionamiento CEC.
- Diseñar una aplicación móvil y web para visualizar y gestionar en tiempo real la disponibilidad de parqueaderos en el estacionamiento del CEC.
- Programar los módulos correspondientes a la aplicación móvil y web para visualizar y gestionar en tiempo real la disponibilidad de parqueaderos en el estacionamiento del CEC.
- Probar la usabilidad y rendimiento de la aplicación móvil y web respectivamente.

1.2 ALCANCE

El Proyecto de Titulación se encuentra dirigido a las personas que tengan la necesidad de conocer en tiempo real la disponibilidad y hacer uso de los parqueaderos de la EPN y específicamente se encuentra enfocado al parqueadero del CEC. El parqueadero del CEC posee dos etapas, dos accesos directos por la calle Toledo y un acceso secundario por la calle Alfredo Mena Caamaño y Andalucía mediante el uso de brazos de barrera, y un área delimitada en donde se puede encontrar un total de 539 plazas de parqueo.

Este Proyecto de Titulación se va a distribuir en dos aplicaciones con funcionalidades descritas en forma general:

- **Aplicación Web:** Se desarrollará un sistema web para que un usuario gestione la información de los parqueaderos. Se van a desarrollar los servicios web; para proporcionar la información de la disponibilidad de los parqueaderos obtenida de la base de datos a la aplicación móvil; y para gestionar la información de los parqueaderos y sus plazas en la base de datos
- **Aplicación Móvil:** Se desarrollarán módulos para: la visualización de un mapa con marcadores de los parqueaderos e información de su disponibilidad y que permitirá

marcar rutas de la ubicación actual a un parqueadero; visualizar información de los parqueaderos; y visualizar información de ayuda y acerca de la aplicación.

Además, una característica importante del proyecto es conocer la disponibilidad del parqueadero del CEC en tiempo real. Por tal motivo, se va a utilizar un sensor ultrasónico que conectado a un Raspberry Pi se va a poder conocer en tiempo real la disponibilidad de las plazas del parqueadero del CEC.

1.3 MARCO DE REFERENCIA

En el presente tema se muestra el marco teórico fundamental que justifica la base para el desarrollo de este Proyecto de Titulación. En primer lugar, se describe la distribución y estado actual de los parqueaderos de la EPN. Luego, se detalla el estado del mercado de Smartphones y los sistemas operativos más utilizados actualmente por los usuarios que explica la razón que se ha escogido el sistema operativo Android para el desarrollo de la aplicación móvil. Después, se definen los servicios web y los protocolos creados que muestra el por qué se ha elegido al protocolo REST para desarrollar los servicios web. Luego, se describen las diferencias entre aplicación web y de escritorio que aclara la razón de haber elegido desarrollar una aplicación web. Luego, se analizan los tipos de sensores para elegir al más óptimo para conocer la disponibilidad de una plaza y se detalla el dispositivo que acompañaría al sensor para conocer la disponibilidad en tiempo real. Posteriormente, se analiza la metodología de desarrollo de software Scrum y se la compara con la metodología XP y se explica el por qué se utilizó la metodología Scrum. Después, se detalla el funcionamiento de la arquitectura Modelo-Vista-Controlador (MVC) y se muestra la Arquitectura General. Por último, se definen las herramientas, tecnologías y lenguajes de programación, y se describen su utilización en el Proyecto de Titulación.

1.3.1 Estado Actual de los Parqueaderos de la EPN

Para conocer el estado actual de los parqueaderos de la EPN se utilizó información de una tesis realizada en 2014 junto con un estudio de campo realizado en 2019 para visualizar los cambios realizados en la señalética y distribución de los parqueaderos. En la actualidad la EPN cuenta con un total de 17 parqueaderos distribuidos por todo el campus José Rubén Orellana Ricaurte, posee 8 entradas en total para el acceso a los parqueaderos. En la Tabla 1 se detallan cada uno de los parqueaderos existentes en la EPN y sus características [6].

N°	Zona	# Plazas	# Plazas - personas con discapacidad	# Plazas – Bicicletas
1	Administración 1	75	1	0
2	Administración 2	23	3	1
3	Ciencias Administrativas	32	1	0
4	Laboratorio de la Vivienda	16	1	0
5	Civil	37	2	0
6	Visitas	8	0	0
7	Nucleares	33	2	0
8	Ciencias	16	1	0
9	Ciencias Básicas (ICB)	38	2	1
10	Geología y Petróleos	18	1	1
11	Tecnólogos 1	19	2	0
12	Facultad de Eléctrica	65	3	1
13	Acelerador de electrones	7	0	0
14	Tecnólogos 2	56	2	0
15	Metalurgia extractiva	6	0	0
16	CEC – subsuelo 1	264	5	4
17	CEC – subsuelo 2	275	5	0
	Total plazas por Tipo	988	31	8
	Total de plazas	1027		

Tabla 1. Distribución Parqueaderos de la EPN. Fuente: [6].

1.3.2 Aplicaciones Móviles

Un factor para tomar en cuenta en el desarrollo de la aplicación móvil de este Proyecto de Titulación es el aumento de usuarios de smartphones en el mundo. Newzoo un proveedor de análisis de datos manifiesta que en el año 2021 habrían aproximadamente 3.8 millones de usuarios de smartphones, como se observa en la Tabla 2 [7].

Año Países	2016	2017	2018	2019	2020	2021
Norte América	258M	268M	278M	288M	297M	305M
Latino América	203M	230M	259M	288M	317M	347M
Europa Oeste	291M	298M	306M	313M	319M	325M
Europa Este	192M	202M	212M	220M	229M	237M
Oriente Medio y África	279M	310M	342M	376M	413M	451M
Asia-Pacífico	354M	397M	441M	486M	532M	580M
India	227M	300M	375M	456M	531M	601M
China	687M	735M	783M	830M	875M	918M
TOTAL (Millones)	2491M	2741M	2995M	3257M	3513M	3763M

Tabla 2. Usuarios Globales de Smartphones por Región 2016-2021. Fuente: [7].

Además, es importante diferenciar la cantidad de usuarios de los dos principales fabricantes de smartphones siendo Samsung (con el sistema operativo Android) el líder con 893 millones de smartphones y Apple (con el sistema operativo iOS) con 800 millones de smartphones, como se observa en la Tabla 3 [7].

Rango	Marca	Dispositivos Activos (Millones)	Porcentaje de Mercado
1	Samsung	893.1M	26.8%
2	Apple	800.7M	24.0%
3	Oppo	331.3M	9.9%
4	Xiaomi	287.5M	8.6%
5	Huawei	280.5M	8.4%

Tabla 3. Top 5 de Marcas de Smartphones por Dispositivos Activos Mensuales Junio 2018.

Fuente: [7].

1.3.3 Servicios Web

Un servicio web es una tecnología de comunicación que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Los servicios web son componentes de aplicaciones distribuidas que están disponibles de forma externa. Se pueden

utilizar para integrar aplicaciones escritas en diferentes lenguajes y que se ejecutan en plataformas diferentes [8].

La primera introducción a los servicios web comenzó con el Protocolo Simple de Acceso a Objetos (SOAP), que establece cómo deben realizarse las comunicaciones entre máquinas. Sin embargo, como una alternativa más simple se creó la Transferencia de Estado Representacional (REST) y ya varios grandes proveedores están migrando a esta tecnología, incluyendo a Yahoo, Google y Facebook [8].

Servicio Web REST

El protocolo REST (Transferencia de Estado Representacional) define un conjunto de principios de arquitectura para diseñar servicios web enfocándose en los recursos del sistema. Para acceder a estos servicios se utiliza directamente HTTP [9].

Los 4 principios de REST

Los servicios web REST siguen cuatro diseños fundamentales: [9]

- Utiliza métodos HTTP de manera explícita, utilizando las siguientes operaciones:
 - POST para crear un recurso.
 - GET para obtener un recurso.
 - PUT para actualizar un recurso.
 - DELETE para eliminar un recurso.
- Los recursos se exponen en forma de directorios a través de su URI.
- Transfiere información entre cliente y servidor en formato y lenguaje XML, HTML, JSON.
- El cliente y el servidor no necesitan recordar ningún estado de las comunicaciones.

1.3.4 Aplicaciones Web

Un factor esencial en el desarrollo de la herramienta para gestionar la información de los parqueaderos es realizar una comparación entre aplicaciones web y de escritorio.

Una aplicación web es una herramienta en la que los usuarios pueden utilizarla accediendo a un servidor web por medio de internet mediante un navegador [10]. A continuación, se va a diferenciar las características de aplicaciones web y de escritorio, como se observa en la Tabla 4 [11].

Características	Aplicación Web	Aplicación Escritorio
Portabilidad	Óptima, ya que puede acceder desde cualquier dispositivo con acceso a Internet.	Limitada, ya que se utiliza únicamente en el ambiente en que se instaló.
Actualizaciones	Se realizan de forma automática y simultáneamente en todos los equipos desde una única ubicación.	Implica más tiempo y trabajo, ya que se deben recorrer todas las estaciones de trabajo.
Incompatibilidad de versiones	No se presenta, ya que las actualizaciones se realizan simultáneamente.	Puede presentarse si no se actualizan los equipos al mismo tiempo.
Requerimiento de software	No requiere.	Se requiere la adquisición e instalación de un software.
Instalación	No requiere.	Si requiere.
Interfaz	Amigables con el usuario.	Amigables con el usuario.
RespalDOS	Están centralizados.	Se encuentra en varias estaciones de trabajo.
Sistemas Operativos	No importa el Sistema Operativo instalado en el dispositivo, sólo se requiere conexión a Internet.	Sólo se puede utilizar en dispositivos que cuenten con determinado Sistema Operativo.
Conexión a Internet	Es indispensable para su uso.	Si requiere.
Tiempo de desarrollo	Menor tiempo de desarrollo, hoy en día se cuenta con herramientas (frameworks) que pueden acelerarlo.	Mayor tiempo de desarrollo.
Tiempo de respuesta	Más lento aunque se puede mejorar con tecnologías como AJAX.	Más rápido.
Seguridad	Es muy segura.	Es muy segura.

Tabla 4 Diferencias, ventajas y desventajas entre aplicaciones de escritorio y las aplicaciones web. Fuente: [11].

1.3.5 Sensores

Los sensores son dispositivos de gran importancia en la actualidad debido a que se puede medir, limitar o automatizar algún proceso, nos permiten captar la información del medio físico que nos rodea [12].

Los sensores permiten medir las magnitudes físicas y transformarlas en señales eléctricas capaces de ser entendidas por un microcontrolador, el tiempo de respuesta en los sensores

es instantáneo, una vez que el objeto que se necesita detectar cae dentro de su rango de operación en el que trabaja o en el que se ajustó el sensor [12].

Tipos de Sensores

- Sensor de Temperatura: Este sensor permite monitorear los cambios de la temperatura mediante la expansión o contracción de sólidos [13].
- Sensor de Movimiento: Este sensor registran el movimiento en un área determinada mediante emisores y receptores de señales, se pueden utilizar como detectores de presencia [14].
- Sensor de Distancia: Este sensor permiten realizar la medida de distancia lineal, se pueden adaptar para medir la distancia o ser utilizados como sensores de presencia [15].
- Sensor de Luz: Este sensor permiten sentir la presencia de luz y el cambio en la intensidad de la luz [16].
- Sensor de Proximidad: Este sensor registran la presencia de un objeto sin la necesidad de contacto y se pueden configurar la medición de distancia [17].
- Sensor de Presión: Este sensor transformas la magnitud física de presión en una señal normaliza [18].
- Sensor de Posición: Este sensor pueden ser de posición lineal y posición angular, permiten determinar la ubicación de un objeto [19].
- Sensor de Color: Este sensor convierten de luz a frecuencia, permitiendo la detección de colores a partir de la radiación reflejada [20].
- Sensor de Humedad: Este sensor miden el nivel de líquido de un área establecida, permitiendo controlar la temperatura y la humedad del aire [21].
- Sensor Magnético: Este sensor permite realizar mediciones sin contacto de alta precisión y en tiempo real [22].
- Sensor Mecánico: Este sensor mide la deformación o el esfuerzo, permitiendo conocer si un objeto está sujeto a niveles de carga segura [23].

Una vez analizado estos tipos de sensores, sus características y funcionalidades, se seleccionó el sensor más óptimo para conocer la disponibilidad de una plaza. Los sensores ultrasónicos permiten conocer la presencia de un objeto dentro de un espacio, estos sensores emiten pulsos haciendo que las ondas lanzadas reboten en la superficie, si un objeto interfiere en el retorno de la onda se detecta un cambio en el tiempo de retorno [14].

Sensor Ultrasonido HC-SR04

Este es un sensor de distancia que utiliza ultrasonido para medir la distancia de un objeto en un rango de 2 a 450cm, siendo esta la razón de su elección y teniendo en cuenta las dimensiones del Parqueadero del CEC y más específicamente las dimensiones de cada plaza. Además, se eligió este sensor por su bajo costo, pequeño tamaño, bajo consumo energético y buena precisión [24].



Figura 1. Sensor Ultrasonido HC-SR04. Fuente: [24].

Especificaciones Técnicas	
Voltaje de Operación	5V DC
Corriente de Reposo	< 2mA
Corriente de Trabajo	15mA
Rango de Medición	2cm a 450cm
Precisión	+/- 3mm
Ángulo de Apertura	15°
Frecuencia de Ultrasonido	40KHz
Duración mínima del pulso de disparo TRIG (nivel TTL)	10µS
Duración del pulso ECO de salida (nivel TTL)	100-25000 µS
Dimensiones	45mm x 20mm x 15mm
Tiempo mínimo de espera entre una medida y el inicio de otra	20ms (recomendable 50ms)

Tabla 5. Especificaciones Técnicas Sensor Ultrasonido HC-SR04. Fuente: [24].

Teniendo en cuenta las especificaciones mencionadas del sensor ultrasonido, y también que la disponibilidad de cada plaza debe ser enviada a un servidor, para que pueda ser visualizada en la aplicación móvil y web, se debe utilizar un método para llevar a cabo estas tareas. Por

tal motivo, se decide utilizar un Raspberry Pi para conectar el sensor y enviar el estado de la plaza.

Raspberry Pi

Es una placa de computadora de bajo costo desarrollada en Reino Unido por la Fundación Raspberry Pi, el objetivo de este dispositivo es estimular la enseñanza de informática en las escuelas [25]. Este dispositivo funciona como una minicomputadora a la que se le puede conectar distintos sensores, dentro de estos el sensor ultrasonido HC-SR04.



Figura 2. Raspberry Pi 2 Modelo B. Fuente: [26].

Especificaciones Técnicas	
Procesador	Broadcom BCM2836, 900MHz quad-core ARM Cortex-A7
Memoria RAM	SSDRAM LPDDR2 1GB
Gráfica	VideoCore IV 3D graphics core
Salida de Video	HD 1080p Compuesto (PAL/NTSC)
Salida de Audio	Estéreo
Conector Ethernet	RJ45 10/100 BaseT
Conector de Video/Audio	HDMI 1.3 y 1.4
Conector de Video	Compuesto/audio de 3.5mm 4 polos
Conector USB	4 conectores USB 2.0
Conector de Cámara	MPI CSI-2 de 15vías
Conector de Memoria	Tarjeta MicroSD
Conector GPIO	40 conectores macho para buses serie
Fuente de Alimentación	+5V a 2A a través de MicroUSB

Tabla 6. Especificaciones Técnicas Raspberry Pi Modelo B. Fuente: [26].

2. METODOLOGÍA

En el presente capítulo se detalla en primer lugar una comparación de las metodologías ágiles existentes, a continuación, se describen los componentes de Scrum; Luego, se detalla la arquitectura del presente proyecto. Finalmente, se describen las herramientas y lenguajes de programación utilizados para el desarrollo del presente proyecto.

2.1 Comparación Scrum y XP

En el presente Proyecto de Titulación se utilizó la metodología Scrum por las características, alcance y requisitos de este en comparación a la metodología XP.

A continuación, se muestra un cuadro comparativo entre las metodologías Ágiles Scrum y XP, como se observa en la Tabla 7 [27].

Criterio	Metodología	
	Scrum	XP
Tipo de Iteraciones	Iteraciones de plazo fijo	Iteraciones de plazo variable
Facilitador	Scrum Master	Coach, Big Boss
Administrador Requerimientos	Product Owner	Cliente
Equipo del Proyecto	Equipo de Desarrollo	Programador, Tester
Equipos	Multifuncional	Especializados
Tamaño del equipo	9	máx. 12
Limitación Work In Progress	Limitación por iteración	Limitación por iteración
Incorporación de Tareas	No es posible hasta finalizar el sprint	No es posible hasta terminar la iteración
Seguimiento de Tareas	Gráfico Burn-down	Velocity
Proceso de Gestión del proyecto	Planeación de Sprint reuniones diarias	Juego de planificación
Trabajo en Equipo	Cada miembro trabaja de manera individual	Los miembros del equipo trabajan en parejas

Requerimientos	No modificables en el tiempo durante el Sprint	Modificables en el tiempo
-----------------------	--	---------------------------

Tabla 7. Comparativa Metodologías Ágiles. Fuente: [27].

Como se observa en la Tabla 7 Scrum a diferencia de las metodologías tradicionales y de otras metodologías ágiles como XP nos permitirá adaptarnos a los cambios que se presenten en el desarrollo de la aplicación, tener una mejor comunicación con el cliente y realizar una programación individual, ideal para nuestro equipo de desarrollo limitado, para generar entregas funcionales de la aplicación, por tal motivo Scrum se acopla perfectamente como metodología para el desarrollo de la aplicación web y móvil.

Scrum es un marco de trabajo para el desarrollo y el mantenimiento de productos complejos, mediante el cual las personas pueden abordar problemas adaptativos complejos, para entregar productos del máximo valor posible productiva y creativamente [28].

El marco de trabajo Scrum consiste en los Equipos Scrum y sus roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso [28].

A continuación, se describen los componentes de Scrum:

2.2 Aplicación de la Metodología

2.2.1 Equipo Scrum y roles

El equipo de Scrum es autoorganizado y multifuncional y se encuentra conformado por los siguientes roles, como se observa en la Tabla 8 [28].

Rol	Descripción
Product Owner – Dueño del Producto	Responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo. Es la única persona responsable de gestionar el Product Backlog.
Development Team – Equipo de Desarrollo	Responsables de entregar un incremento al final de cada Sprint. Son autoorganizados y multifuncionales.
Scrum Máster	Responsable de asegurar que el equipo Scrum se adapte a la teoría, prácticas y reglas de SCRUM.

Tabla 8. Roles de Equipo Scrum. Fuente: [28].

2.2.2 Eventos

Los eventos en Scrum son bloques de tiempo con una determinada duración, la duración de cada Sprint es fija y esta no puede acortarse ni alargarse [28].

Un Sprint es un bloque de tiempo durante el cual se crea un incremento de producto, el mismo que puede ser puesto en producción al final del sprint, tiene una duración de un mes o menos. Al término de cada Sprint inmediatamente comienza uno nuevo. Cada Sprint tiene una definición de lo que se construirá, un diseño, y un plan flexible que guiará su construcción, el trabajo del equipo y el producto restante [28].

Los Sprints contienen y consisten en:

- Planificación del Sprint (Sprint Planning): Durante este evento se planifica el trabajo a realizar durante el Sprint [28].
- Scrums Diarios (Daily Scrums): Es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas [28].
- Revisión del Sprint (Sprint Review): Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de Producto (Product Backlog) si fuese necesario [28].
- Retrospectiva del Sprint (Sprint Retrospective): La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y de crear un plan de mejoras que sean abordadas durante el siguiente Sprint [28].

2.2.3 Artefactos:

Los artefactos representan trabajo o valor para proporcionar transparencia y oportunidades para la inspección y adaptación. Scrum define los siguientes artefactos:

- Lista de Producto (Product Backlog) es una lista ordenada y la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Product Owner es el responsable de la lista, incluyendo su contenido, disponibilidad y ordenación.
El Product Backlog evoluciona a medida que el producto y el entorno también lo hacen, esto debido a que dicha lista no está completa al inicio del desarrollo del producto. La

lista es dinámica, cambia constantemente para adaptarse a lo que el producto necesita para ser adecuado, competitivo y útil [28].

- Lista de Pendientes (Sprint Backlog) es el conjunto de elementos seleccionados del Product backlog para el siguiente Sprint, así como también un plan con la funcionalidad que formará parte del nuevo sprint y de cómo lograr dicha funcionalidad para generar un incremento Terminado.

Incremento es la suma de todos los elementos del Product backlog completados durante un sprint y el valor de los incrementos de todos los sprint anteriores [28].

- Historia de Usuario: son una breve descripción de la funcionalidad del software tal y como lo percibe el usuario. Se describe lo que el cliente o usuario quiere que se implemente y se escriben con una o dos frases utilizando el lenguaje común del usuario.
- Historia de Usuario Épica: se denomina así a una superhistoria de usuario que se distingue por su gran tamaño y su alta granularidad. Esta se puede dividir en historias de usuario más pequeñas debido a que cuyo esfuerzo es demasiado grande para ser completada en un solo sprint.
- Incremento: es el resultado de sumar todos los elementos del Product Backlog completados durante un sprint y el valor de los incrementos anteriores.
- Gráfico de trabajo pendiente (Burndown chart): El Sprint Burndown Chart hace visible el trabajo del Equipo. Es una representación gráfica de la velocidad a la que se completa el trabajo y cuánto trabajo queda por hacer.

2.3 Arquitectura

Para entender la importancia de la arquitectura de software, vamos a presentar la definición propuesta por en el documento de la IEEE Std 1471-2000:

- “Arquitectura es definida por la práctica habitual como la organización fundamental de un sistema, expresada en términos de componentes, las relaciones entre ellos y el entorno, y los principios que gobiernan su diseño y evolución” [29].

Existen varios tipos de arquitectura de software, en este espacio nos vamos a centrar en la Arquitectura MVC.

Arquitectura MVC

El patrón de diseño MVC proviene de las siglas (Modelo Vista Controlador). Una correcta implementación y uso de este patrón logran separar la lógica de negocios (datos o modelo), y la interfaz de usuario (vista), mientras que el controlador es el encargado de relacionar estos dos. De tal forma que, cualquier cambio en la lógica de negocios se refleje automáticamente en cada una de las interfaces, reduciendo el esfuerzo de programación necesaria. En la Figura 3 se puede observar la separación de las tres capas y los componentes que la hacen funcional. [30], [31]

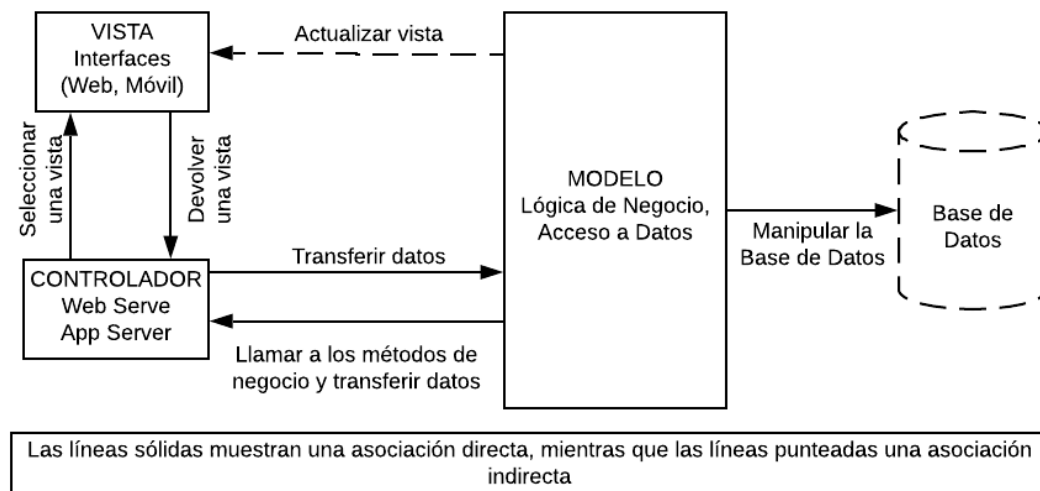


Figura 3. Patrón de diseño MVC.

Este patrón de diseño presenta las siguientes ventajas:

- Una Interfaz de Programación de Aplicaciones, ya que cualquiera que utilice el API podrá modificar los componentes de la arquitectura (Modelo, Vista y Controlador) sin dificultad.
- Permite implementar los componentes por separado, debido a una clara separación entre estos.

Modelo

Es el componente que representa la información (datos) y reglas de negocio usadas para manipular todas las transformaciones de los datos. Gestiona los mecanismos de persistencia de la información en la base de datos [30], [31], como se observa en la Figura 4.

Vista

Es el componente que maneja la presentación visual de la información (datos) contenida en el Modelo en una forma que permita interacción a través de una interfaz de usuario [30], [31], como se observa en la Figura 4.

Controlador

Es el componente que procesa y responde a acciones del usuario, actuando sobre la información (datos) representada por el Modelo y, en última instancia, selecciona una vista para mostrarla. Centra toda la interacción entre la Vista y el Modelo [30], [31], como se observa en la Figura 4.

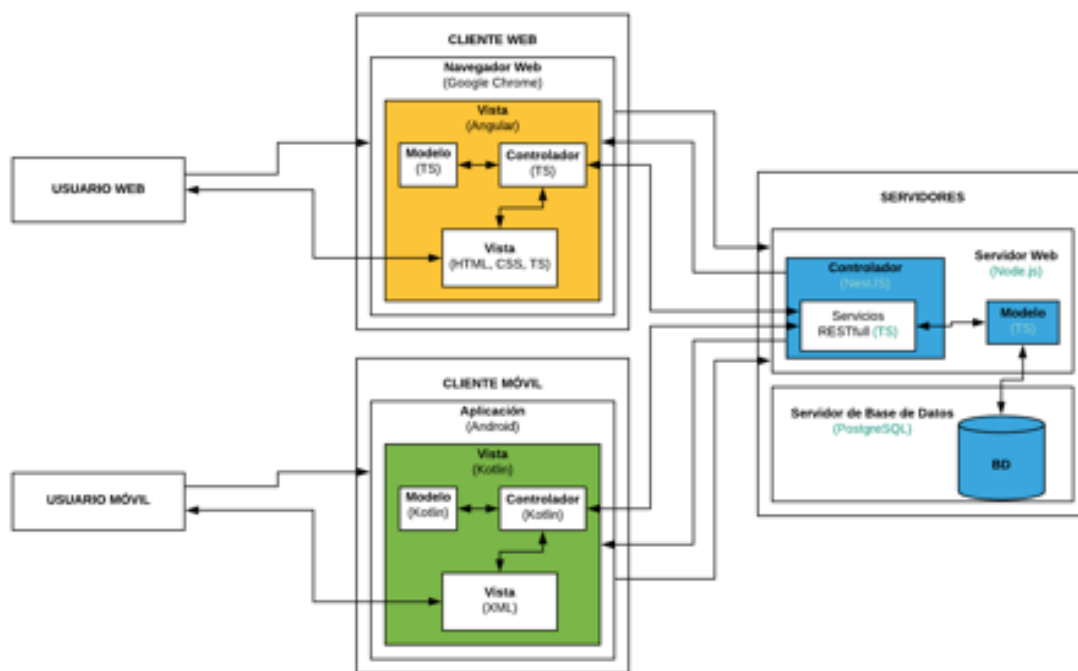


Figura 4. Arquitectura General MVC.

Consideramos que a lo largo de la explicación del patrón de diseño MVC, queda justificada su elección en la implementación de este Proyecto de Titulación. Además, utilizando este diseño se logra una estructura bien definida que ayuda a que un proyecto sea organizado y bien desarrollado. Existen varios frameworks basados en este diseño que facilitan su implementación, los mismos que se han utilizado como se describen en el apartado 2.3

2.4 Lenguajes y Herramientas de Programación

En el desarrollo del presente Proyecto de Titulación, se utilizaron varios lenguajes de programación y herramientas especializadas en prototipado, repositorios, base de datos, entornos de desarrollo y frameworks, como se observa en la Tabla 9.

Nombre	Descripción	Utilización
Balsamiq Mockups 3	Es una herramienta que permite crear prototipos para diseño de interfaces web, móviles y de escritorio [32].	Desarrollo del prototipo para la interfaz móvil y web.
Github	Es una plataforma que ayuda a las personas a resolver problemas creando software juntos [33].	Creación de los repositorios para la aplicación móvil y web.
PowerDesigner	Es una herramienta especializada en la modelización de datos, permite crear diferentes técnicas de modelización (modelo conceptual, tradicional, físico y lógico) [34].	Desarrollo del modelo conceptual, lógico y físico de la base de datos.
PostgreSQL	Es un poderoso sistema de base de datos relacional de objetos de código abierto. Usa el lenguaje SQL con características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas, debido a su arquitectura comprobada, confiabilidad, integridad de datos, conjunto de características robustas y extensibilidad [35].	Creación de la base de datos para almacenar la información de las aplicaciones móvil y web.
Android Studio	Es un entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android [36].	Desarrollo de la aplicación móvil.
Kotlin	Es un lenguaje de programación para desarrollar aplicaciones Android, ofrece todas las ventajas de un lenguaje moderno sin introducir nuevas restricciones [37].	Lenguaje de programación para el desarrollo de la aplicación móvil.
WebStorm	Es un IDE de la plataforma IntelliJ para la asistencia de codificación inteligente para JavaScript y lenguajes compilados a JavaScript, Node.js, HTML y CSS. Además, una asistencia de codificación avanzada para Angular, React, Vue.js y Meteor [38].	Desarrollo de los servicios web y de la aplicación web.

TypeScript	Es un lenguaje de programación con herramientas de programación orientada a objetos. Es un superset de Javascript que se compila en JavaScript simple [39].	Lenguaje de programación para el desarrollo de los servicios y la aplicación web.
Node.js	Es un entorno de ejecución para Javascript orientado a eventos asíncronos y diseñado para desarrollar aplicaciones en red escalable. Es ideal para ser un framework web, ya que esta diseñado con operaciones de streaming y baja latencia en mente [40].	Entorno de ejecución para el desarrollo de los servicios y la aplicación web.
NestJS	Es un framework para desarrollar aplicaciones para funcionar como servidor escalable y eficiente. Utiliza JavaScript progresivo y es compatible con TypeScript. Hace uso de robustos marcos de Servidor HTTP como Express [41].	Desarrollo de los servicios para la aplicación móvil y web.
Angular	Es una plataforma y un framework para crear aplicaciones clientes en HTML y TypeScript [42].	Desarrollo de la aplicación web.

Tabla 9. Lenguajes y herramientas de programación.

3. DESARROLLO DE LA APLICACIÓN

El presente proyecto se enfoca en brindar información, ubicación geográfica y disponibilidad de cualquier parqueadero; teniendo como objetivo principal mostrar las plazas disponibles de dicho parqueadero y siendo de utilidad para cualquier persona que tenga a disposición un dispositivo móvil, para esto nos hemos centrado en el parqueadero CEC como campo de estudio.

Para el desarrollo de este proyecto se hará uso de la metodología ágil iterativa Scrum, descrita en el apartado Marco Teórico, la misma que permitirá realizar de forma incremental las soluciones para cada una de las funcionalidades de las aplicaciones móvil y web. Scrum con respecto a otras metodologías ágiles como XP, resulta ser la más adecuada debido a que los requerimientos tanto de las aplicaciones móvil y web pueden sufrir cambios, estos cambios Scrum los maneja como nuevos requerimientos mientras que otras metodologías lo realizan durante la iteración, evitando así que existan problemas en el desarrollo.

3.1 Roles

A continuación, en la Tabla 10 se describen las personas a cargo del presente proyecto y el rol que ocupa cada uno.

Persona	Contacto	Rol
Calle Tania	tania.calle@epn.edu.ec	Scrum Master
Mónica Beltrán	monica.beltran@epmmop.gob.ec	Product Owner
Kevin Cárate	carate.kevin@epn.edu.ec	Team Development
Cabrera Ronny	ronny.cabrera@epn.edu.ec	Team Development

Tabla 10. Roles Scrum del Proyecto.

3.2 Requerimientos de la Aplicación

En un principio se tomó como referencia a Mónica Beltrán representante de la EPMMOP para la toma de requerimientos, pero estos se completaron y modificaron con la información obtenida en base al caso de estudio del parqueadero del CEC. Los requerimientos obtenidos se pueden observar en el Anexo 1 - Historias de Usuario.

3.2.1 Contenido del Product Backlog

A continuación, se define el contenido del Product backlog.

Identificador (ID) de la Historia: Código que identifica a la historia de usuario de forma unívoca y que no puede ser reutilizado. En el Product backlog se ha adoptado el código HUE correspondiente a Historia de Usuario Épica precedido de un número ordinal.

Código de Historia de Usuario: Tiene el mismo significado que el Identificador de la historia, pero adaptado a una historia de usuario normal. En la Tabla 13 Descripción de historias de usuario, se ha adoptado el código HU correspondiente a Historia de Usuario precedido del número de la historia épica a la que pertenece y un número ordinal.

Enunciado de la Historia de Usuario: Es el nombre de la historia de usuario, el cual está utilizando el siguiente formato:

Como [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón], esto último es opcional.

Estado: Una historia de usuario, durante su ciclo de vida puede pasar por los siguientes estados:

- Vacío: Si la historia de usuario no ha sido asignada a una iteración.
- Planificada: Si la historia de usuario ha sido previamente asignada pero aún no ha comenzado su ejecución o desarrollo.
- En Proceso: Si la historia de usuario está en proceso de desarrollo por el equipo de trabajo.
- Realizado: Si la historia de usuario ha sido desarrollada, integrada y se han realizado pruebas sobre su funcionalidad.
- Descartada: Si la historia de usuario ha sido cancelada o incluida en otro grupo de historia.

Dimensión / Esfuerzo: Es la medida del esfuerzo que implica desarrollar una determinada historia de usuario, para medir el esfuerzo se ha empleado el método “puntos de historia” que se basa en escoger una historia de usuario sencilla y emplearla como referencia para calcular el esfuerzo de las demás historias de usuario [43].

La historia de usuario escogida es la HUE04 de la Tabla 12 Product Backlog y se le ha asignado un valor de 3 puntos, siendo el valor más bajo y 8 puntos el valor máximo; a continuación, se detallan los posibles valores que se asignarán a una determinada historia de usuario como se observa en la Tabla 11.

Puntos	Descripción
3	Si la historia de usuario es sencilla de desarrollar.
5	Si la historia de usuario tomará el doble de esfuerzo en ser desarrollada.
8	Si la historia de usuario tomará demasiado esfuerzo en ser desarrollada.

Tabla 11. Puntos de Historia.

Prioridad: Es el nivel de importancia que toma desarrollar una historia de usuario y va de la mano con el esfuerzo que se debe realizar, en el presente trabajo se ha optado por tener tres niveles de prioridad: **Alta, Media, Baja**, siendo Alta la de mayor prioridad y Baja la de menor prioridad.

3.2.2 Product Backlog

A continuación, en la Tabla 12 se presenta el Product Backlog con la lista de requerimientos, denominadas como historias de usuario épicas, para las aplicaciones móvil y web, dimensión y prioridad de cada una.

PRODUCT BACKLOG VERSIÓN 1.0				
Identificador (ID) de la Historia	Enunciado de la Historia de Usuario Épica	Estado	Dimensión / Esfuerzo	Prioridad
HUE01	Como usuario, necesito visualizar la disponibilidad de plazas de un parqueadero, con la finalidad de conocer la cantidad de plazas libres y su ubicación.	Planificada	8	Alta
HUE02	Como usuario, necesito visualizar los parqueaderos en un mapa.	Planificada	8	Alta
HUE03	Como usuario, necesito trazar una ruta desde mi ubicación en tiempo real hasta el	Planificada	8	Alta

	estacionamiento, con la finalidad de guiarme en un mapa.			
HUE04	Como Usuario, necesito tener un módulo de ayuda e información de la aplicación móvil.	Planificada	3	Baja
HUE05	Como Administrador, necesito crear una cuenta de usuario, con la finalidad de gestionar la información de los parqueaderos.	Planificada	8	Alta
HUE06	Como Administrador, necesito observar la disponibilidad de las plazas en el aplicativo web, con la finalidad de tener un control de las mismas.	Planificada	8	Alta
HUE07	Como Administrador, necesito gestionar las cuentas de usuarios, con la finalidad de realizar modificaciones.	Planificada	5	Media
HUE08	Como Administrador, necesito gestionar la información de los parqueaderos.	Planificada	8	Alta
HUE09	Como Administrador, necesito tener un módulo de ayuda e información de la aplicación web.	Planificada	3	Baja

Tabla 12. Product Backlog.

Las historias de usuario épicas pueden llegar a ser complejas por lo tanto se las puede descomponer en historias de usuario mucho más específicas.

A continuación, en la Tabla 13 se detallan las historias de usuario específicas para cada una de las historias de usuarios épicas.

Identificador Historia de Usuario Épica	Código Historia de Usuario	Enunciado de la Historia de Usuario Épica	Dimensión / Esfuerzo	Prioridad
HUE01	HU1.1	Diseño de Base de Datos	8	Alta

	HU1.2	Diseño y construcción de servicios	8	Alta
	HU1.3	Diseño de interfaz principal - App Móvil	8	Alta
	HU1.4	Configuración y Programación del Sensor	5	Media
HUE02	HU2.1	Diseño de interfaz para la información de parqueaderos - App Móvil	8	Alta
	HU2.2	Diseño de interfaz para las plazas de parqueadero CEC - App Móvil	8	Alta
HUE03	HU3.1	Implementación de ruta - App Móvil	8	Alta
HUE04	HU4.1	Diseño interfaz de información y ayuda - App Móvil	3	Baja
HUE05	HU5.1	Cuentas de usuario - App Web	8	Alta
	HU5.2	Interfaz de acceso - App Web	5	Media
	HU5.3	Interfaz de Recuperación de contraseña - App Web	3	Baja
HUE06	HU6.1	Interfaz de registro de plazas y sensores - App Web	8	Alta
HUE07	HU7.1	Interfaz Gestión de cuentas de usuario - App Web	5	Media
	HU7.2	Interfaz Cambio de contraseña - App Web	3	Baja
HUE08	HU8.1	Interfaz de creación de parqueaderos - App Web	8	Alta
	HU8.2	Interfaz de actualización de plazas y sensores - App Web	8	Alta
	HU8.3	Interfaz Gestión de información de parqueaderos - App Web	8	Alta
HUE09	HU9.1	Interfaz de información y ayuda - App Web	3	Baja

Tabla 13. Descripción Historias de Usuario.

3.3. Release Plan

En el Release Plan se detallan las historias de usuario que se realizarán en cada uno de los sprint como se observa en la Tabla 14.

Release Plan				
SPRINT 1	SPRINT 2	SPRINT 3	SPRINT 4	SPRINT 5
HU1.1	HU2.1	HU5.1	HU6.1	HU1.4
HU1.2	HU2.2	HU5.2	HU8.1	HU4.1
HU1.3	HU3.1	HU5.3	HU8.3	HU7.2
		HU7.1		HU8.2
				HU9.1
24	24	21	24	22

Tabla 14. Release Plan.

3.4 Sprint 0

3.4.1 Planificación

En el Sprint 0 se realizó las siguientes tareas:

- Estudio de campo del estado actual en el parqueadero del CEC.
- Creación de mockups de la aplicación móvil y web, se pueden observar a detalle en el Anexo 2 - Mockups.
- Definición de las herramientas de desarrollo a ser utilizadas para la aplicación móvil y web.
- Instalación de las herramientas definidas en la Tabla 9 Lenguajes y Herramientas de Programación.
- Definición de la Arquitectura de la aplicación definida en la Figura 4 Arquitectura General MVC.
- Definición de los requisitos de la aplicación.
- Creación de las historias de usuario Épicas y Específicas.

3.4.2 Sprint Review

La preparación del entorno de desarrollo se ha podido cumplir satisfactoriamente y este ha sido establecido de manera local.

Durante el desarrollo del presente Sprint no se han presentado inconvenientes al momento de la instalación de las herramientas, ni al momento de definir la arquitectura de la aplicación y los requisitos previos para el desarrollo de las aplicaciones móvil y web.

En el Sprint 0 se han definido las historias de usuario Épicas y posteriormente han sido desglosadas en historias de usuario específicas, también se ha definido el esfuerzo de cada una y se las ha agrupado para ser realizadas en los diferentes Sprints del presente trabajo.

3.4.3 Sprint Retrospective

El desarrollo del Sprint 0 se llevó a cabo sin inconvenientes debido a que solamente se enfoca en la definición y preparación del entorno de desarrollo, la definición de requisitos, arquitectura y el estudio de campo del parqueadero del CEC.

3.5 Sprint 1

3.5.1 Planificación

El objetivo del Sprint 1 es realizar el diseño y la implementación de la base de datos en PostgreSQL, construcción de los servicios para el tratamiento de los datos, es decir el desarrollo del Backend que servirá tanto para la aplicación móvil como web y finalmente el diseño y desarrollo de la interfaz principal del aplicativo móvil.

Historias de Usuario del Sprint 1

HISTORIA DE USUARIO	HU1.1
Usuario: Usuario del sistema	
Título: Diseño de Base de Datos.	
Descripción: Diseñar y construir la base de datos para almacenar los datos pertenecientes a los parqueaderos con sus respectivos estados.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación: <ol style="list-style-type: none">1. La base de datos debe almacenar información relevante sobre un determinado parqueadero, como: Nombre, ubicación, número de pisos, dirección, número de plazas, tarifas.	

<ol style="list-style-type: none"> 2. La base de datos debe almacenar información relevante sobre las plazas de un determinado parqueadero, como: estado, tipo, código, ubicación. 3. La base de datos debe almacenar información relevante sobre los sensores de cada plaza como: código, estado, dirección. 4. La base de datos debe almacenar información relevante sobre los usuarios como: rol, información personal.

Tabla 15. Historia de Usuario 1.1.

HISTORIA DE USUARIO	HU1.2
Usuario: Usuario del sistema	
Título: Diseño y construcción de servicios	
Descripción: Diseñar y construir los servicios necesarios para realizar las operaciones CRUD sobre la base de datos.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación:	
<ol style="list-style-type: none"> 1. Permitir realizar peticiones Get, Post, Put y Delete con autorización para cada una de las rutas pertenecientes a los servicios. 	

Tabla 16. Historia de Usuario 1.2.

HISTORIA DE USUARIO	HU1.3
Usuario: Usuario del sistema	
Título: Diseño de Interfaz Principal - App Móvil	
Descripción: Diseño de interfaz Principal.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación:	
<ol style="list-style-type: none"> 1. Visualizar la navegación de los módulos: Mapa, Parqueaderos, Acerca, Ayuda 2. Visualizar mapa de google Maps. 3. Visualizar los marcadores en donde se encuentran los parqueaderos. 	

4. Visualizar el nombre del parqueadero, cantidad de parqueaderos disponibles.

Tabla 17. Historia de Usuario 1.3.

Sprint Backlog Sprint 1

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Actividades
HU1.1	Diseño de Base de Datos	La base de datos debe almacenar información relevante sobre un determinado parqueadero, como: Nombre, ubicación, número de pisos, dirección, número de plazas, tarifas.	<ul style="list-style-type: none"> • Diseño del Diagrama Entidad Relación de la BD en la herramienta PowerDesigner. • Implementación de la BD en PostgreSQL. • Creación entidad Parqueadero
		La base de datos debe almacenar información relevante sobre las plazas de un determinado parqueadero, como: estado, tipo, código, ubicación.	<ul style="list-style-type: none"> • Creación entidad Plaza
		La base de datos debe almacenar información relevante sobre los sensores de cada plaza como: código, estado, dirección.	<ul style="list-style-type: none"> • Creación entidad Sensores
		La base de datos debe almacenar información relevante sobre los usuarios como: rol, información personal.	<ul style="list-style-type: none"> • Creación entidad Usuario • Creación entidades restantes.
HU1.2	Diseño y construcción de servicios	Permitir realizar peticiones Get, Post, Put y Delete con autorización para cada una de las rutas pertenecientes a los servicios.	<ul style="list-style-type: none"> • Creación de un nuevo proyecto en la herramienta WebStorm. • Creación de servicios mediante NestJS • Creación de Controladores para cada servicio creado.

			<ul style="list-style-type: none"> • Conexión entre BD y Servicios.
HU1.3	Diseño de Interfaz Principal - App Móvil	Visualizar la navegación de los módulos: Mapa, Parqueaderos, Acerca, Ayuda	<ul style="list-style-type: none"> • Creación de navegabilidad de aplicación. • Creación de Módulos.
		Visualizar mapa de google Maps.	<ul style="list-style-type: none"> • Creación de fragmento para mostrar mapa de Google Maps. • Creación de certificado SHA-1 de la aplicación en Android Studio. • Creación de proyecto en Google Cloud Platform para obtener la clave de aplicación.
		Visualizar los marcadores en donde se encuentran los parqueaderos.	<ul style="list-style-type: none"> • Creación de los controladores para obtener coordenadas geográficas. • Consumo de servicios del Backend para obtener coordenadas geográficas de los parqueaderos. • Creación de los marcadores en Mapa con los datos obtenidos.
		Visualizar el nombre del parqueadero, cantidad de parqueaderos disponibles.	<ul style="list-style-type: none"> • Creación de los controladores para obtener nombre y disponibilidad de parqueaderos. • Consumo de servicios del Backend para obtener nombre y disponibilidad de parqueaderos. • Creación de una ventana personalizada para mostrar los datos.

Tabla 18. Sprint Backlog Sprint 1.

Prototipo

A continuación, se presenta el resultado de las actividades realizadas en el Sprint 1 y el prototipo de las siguientes interfaces:

La Figura 5 presenta el modelo de la base de datos. Las tablas que se han definido son las siguientes:

- Usuario. - son los registros de los usuarios que pueden ingresar a la aplicación web.
- Rol. - son los registros de los roles que puede tener un usuario.
- Permiso. - son los registros que definen las funcionalidades que puede realizar un usuario en la aplicación web dependiendo de su Rol.
- Parqueadero. - son los registros de los parqueaderos que tiene la aplicación.
- Plaza. - son los registros de las plazas que puede tener un parqueadero y los sensores enlazados.
- Horario. - son los registros de los horarios que puede tener un parqueadero.

Los modelos físico y lógico se pueden observar en el Anexo 3 - Base de Datos.

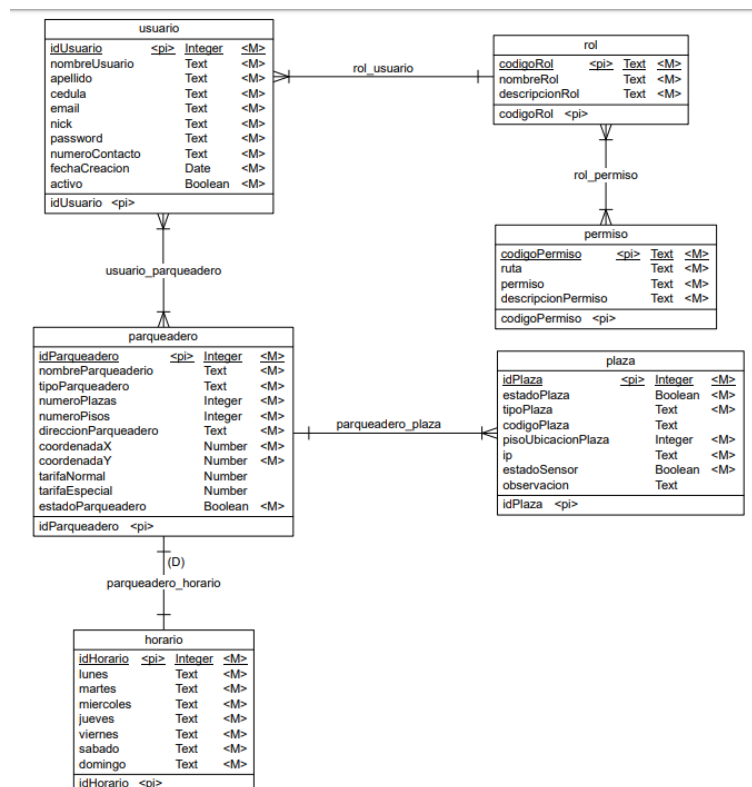


Figura 5. Modelo de la Base de Datos.

La Figura 6 presenta el funcionamiento del Backend que contiene los servicios que serán utilizados en los aplicativos móvil y web. Se puede ver la ejecución de varios servicios como:

- ParqueaderoController. - contiene funcionalidades para obtener información del parqueadero, ingresar nuevos parqueaderos, etc.
- PlazaController. - contiene funcionalidades para obtener la disponibilidad de plazas, asignar plazas a un parqueadero, etc.

```
[Nest] 3352 - 2020-01-16 16:01:34 [RoutesResolver] ParqueaderoController (/parqueadero): +63ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/, POST) route +40ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/, GET) route +40ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/id/:id, GET) route +19ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/:name, GET) route +21ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/:id, PUT) route +4ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/:id, DELETE) route +5ms
[Nest] 3352 - 2020-01-16 16:01:34 [RoutesResolver] ParqueaderoUsuarioController (/parqueaderoUsuario): +7ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/, POST) route +4ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/, GET) route +3ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/user/:id, GET) route +14ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/:codigo, GET) route +14ms
[Nest] 3352 - 2020-01-16 16:01:34 [RoutesResolver] PermisoController (/permiso): +8ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/, POST) route +4ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/, GET) route +10ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/:id, GET) route +4ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/:id, PUT) route +3ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/:id, DELETE) route +5ms
[Nest] 3352 - 2020-01-16 16:01:34 [RoutesResolver] PlazaController (/plaza): +8ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/, POST) route +5ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/, GET) route +4ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/:id, GET) route +13ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/codigo/:codigo, GET) route +9ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/parqueadero/:id, GET) route +9ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/estado/:estado, GET) route +4ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/parqueadero/:id/:estado, GET) route +13ms
[Nest] 3352 - 2020-01-16 16:01:34 [RouterExplorer] Mapped (/:id, PUT) route +6ms
```

Figura 6. Ejecución Backend.

La Figura 7 presenta la navegación y los módulos de la aplicación: Mapa, Parqueaderos, Acerca de Park EPN, Ayuda.

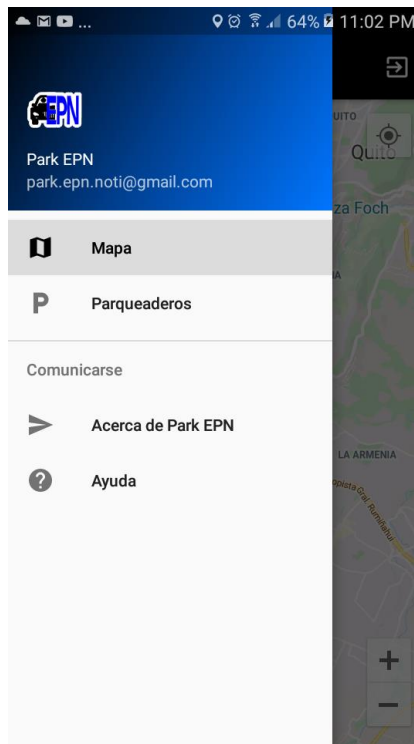


Figura 7. Visualización de Módulos.

La Figura 8 presenta el mapa de Google Maps y los marcadores de los parqueaderos en su ubicación geográfica. Existen marcadores con tres colores:

- Verde. - significa que un parqueadero se encuentra ocupado menos el 30% de su capacidad,
- Amarillo. - significa que un parqueadero se encuentra ocupado entre el 30% y 60% de su capacidad.
- Rojo. - significa que un parqueadero se encuentra ocupado más del 60% de su capacidad.



Figura 8. Visualización de Mapa y Marcadores.

La Figura 9 y Figura 10 presentan una ventana sobre los marcadores con la información de los parqueaderos: Nombre del Parqueadero y Plazas Ocupadas.



Figura 9. Visualización Datos Parqueadero CEC-EPN.



Figura 10. Visualización Datos Parqueadero FIEE-EPN.

3.5.2 Sprint Review

Durante el desarrollo del Sprint 1 se logró cumplir con el objetivo planteado, el diseño de la Base de Datos se realizó en base a los requerimientos planteados, en la implementación del Backend no se tuvo mayor complicación debido a que NestJS presenta facilidades para la creación de servicios, finalmente se logró crear la interfaz principal del aplicativo móvil en el cual se tuvo un poco de complicación al momento de mapear los marcadores correspondientes a los parqueaderos.

En la Tabla 19 se muestra si las actividades desarrolladas durante el sprint 1 fueron cumplidas.

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Cumplido
HU1.1	Diseño de Base de Datos	La base de datos debe almacenar información relevante sobre un determinado parqueadero, como: Nombre, ubicación, número de pisos, dirección, número de plazas, tarifas.	SI
		La base de datos debe almacenar información relevante sobre las plazas de un determinado parqueadero, como: estado, tipo, código, ubicación.	SI
		La base de datos debe almacenar información relevante sobre los sensores de cada plaza como: código, estado, dirección.	SI
		La base de datos debe almacenar información relevante sobre los usuarios como: rol, información personal.	SI
HU1.2	Diseño y construcción de servicios	Permitir realizar peticiones Get, Post, Put y Delete con autorización para cada una de las rutas pertenecientes a los servicios.	SI
HU1.3	Diseño de Interfaz Principal - App Móvil	Visualizar la navegación de los módulos: Mapa, Parqueaderos, Acerca, Ayuda	SI
		Visualizar mapa de google Maps.	SI
		Visualizar los marcadores en donde se encuentran los parqueaderos.	SI
		Visualizar el nombre del parqueadero, cantidad de parqueaderos disponibles.	SI

Tabla 19. Sprint Review 1.

Gráfico Burndown Sprint 1

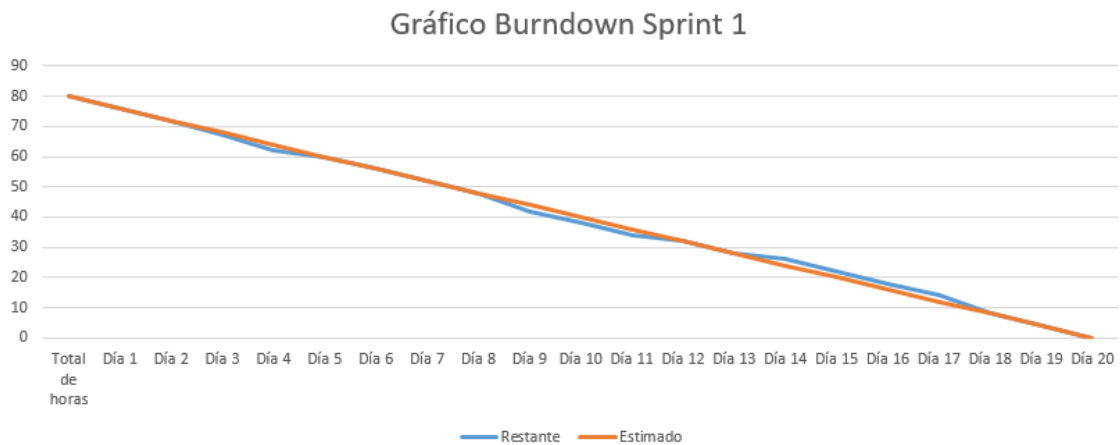


Figura 11. Gráfico Burndown Sprint 1.

Como se observa en la Figura 11 Gráfico Burndown Sprint 1, durante la primera semana se empezó con un ritmo constante, en el cuarto día se trabajó más horas de lo normal, posteriormente se retoma el ritmo constante hasta los días 9, 10 y 11 en los cuáles nuevamente existen más horas trabajadas que las permitidas, ya en los días 14, 15, 16 y 17 existen atrasos en el desarrolló y finalmente en el día 18 se retoma el ritmo constante. El desarrollo del Gráfico Burndown del Sprint 1 se puede observar con más detalle en el Anexo 4 - Gráficos Burdown.

3.5.3 Sprint Retrospective

El desarrollo del Sprint 1 se realizó sin mayores complicaciones a parte de los mencionados en el Sprint Review del presente Sprint, como se observa en el gráfico Burndown se tuvo inconvenientes entre la semana 3 y 4, para esto se debe realizar una mejor planificación de las tareas para evitar que algunas tareas se desarrollen en menor o mayor tiempo del establecido en un inicio, finalmente a mitad de la semana 4 las actividades retomaron sus tiempos establecidos.

3.6 Sprint 2

3.6.1 Planificación

El objetivo del Sprint 2 es realizar el diseño y desarrollo de la interfaz para visualizar la información de los parqueaderos y la distribución de las plazas dentro del parqueadero del CEC en aplicativo móvil.

Historias de Usuario del Sprint 2

HISTORIA DE USUARIO	HU2.1
Usuario: Usuario del sistema	
Título: Diseño de interfaz para la información de parqueaderos - App Móvil	
Descripción: Diseño de la interfaz que contiene la información de los diferentes parqueaderos registrados.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación: <ol style="list-style-type: none">1. Visualizar la lista de parqueaderos.2. Buscar un parqueadero dentro de la lista.3. Mostrar la información de un parqueadero al seleccionar uno.	

Tabla 20. Historia de Usuario 2.1.

HISTORIA DE USUARIO	HU2.2
Usuario: Usuario del sistema	
Título: Diseño de interfaz para las plazas de parqueadero CEC - App Móvil	
Descripción: Diseño de la interfaz que contiene la distribución del parqueadero del CEC.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación: <ol style="list-style-type: none">1. Visualizar la distribución de los pisos del parqueadero del CEC.2. Visualizar la distribución de las plazas del parqueadero del CEC.	

Tabla 21. Historia de Usuario 2.2.

HISTORIA DE USUARIO	HU3.1
Usuario: Usuario del sistema	
Título: Implementación de ruta - App Móvil	
Descripción: Implementación de la ruta, desde la ubicación del usuario hacia el parqueadero seleccionado.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación:	
1. Visualizar la ruta desde la posición actual hasta un determinado parqueadero.	

Tabla 22. Historia de Usuario 3.1.

Sprint Backlog Sprint 2

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Actividades
HU2.1	Diseño de interfaz para la información de parqueaderos - App Móvil	Visualizar la lista de parqueaderos.	<ul style="list-style-type: none"> • Creación de fragmento para mostrar la lista de los parqueaderos. • Creación de los controladores para obtener nombre de parqueaderos. • Consumo de servicios del Backend para obtener nombre de parqueaderos. • Creación de una ventana personalizada para mostrar los datos.
		Buscar un parqueadero dentro de la lista.	<ul style="list-style-type: none"> • Creación de un filtro para buscar los parqueaderos por el nombre.
		Mostrar la información de un parqueadero al seleccionar uno.	<ul style="list-style-type: none"> • Creación de los controladores para información de parqueaderos. • Consumo de servicios del Backend para información de parqueaderos

			<ul style="list-style-type: none"> • Creación de una ventana personalizada para mostrar los datos.
HU2.2	Diseño de interfaz para las plazas de parqueadero CEC - App Móvil	Visualizar la distribución de los pisos del parqueadero del CEC.	<ul style="list-style-type: none"> • Creación de fragmento para mostrar los pisos del parqueadero del CEC.
		Visualizar la distribución de las plazas del parqueadero del CEC.	<ul style="list-style-type: none"> • Creación de los controladores para obtener la disponibilidad de las plazas. • Consumo de servicios del Backend para obtener la disponibilidad de las plazas. • Creación de ventana para marcar la disponibilidad de plazas.
HU3.1	Implementación de ruta - App Móvil	Visualizar la ruta desde la posición actual hasta un determinado parqueadero	<ul style="list-style-type: none"> • Habilitar la API de Dirección en Google Cloud Platform para obtener la clave de aplicación. • Creación de la ruta entre la ubicación actual y un parqueadero.

Tabla 23. Sprint Backlog Sprint 2.

Prototipo

A continuación, se presenta el resultado de las actividades realizadas en el Sprint 2 y el prototipo de las siguientes interfaces:

La Figura 12 presenta la lista de los parqueaderos verticalmente con el icono y el nombre del parqueadero.

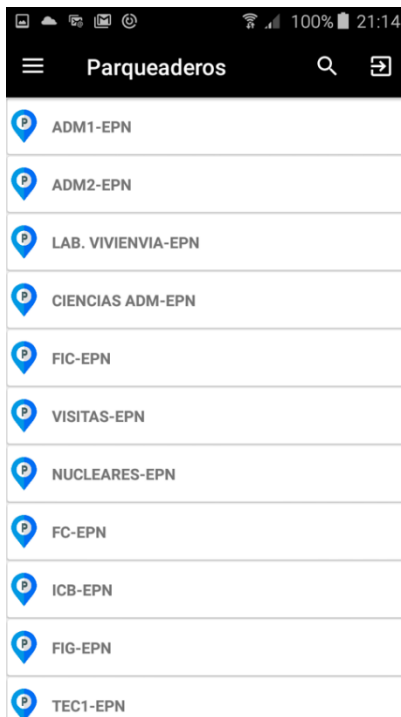


Figura 12. Visualización de Lista de Parqueaderos.

La Figura 13 presenta la búsqueda de un parqueadero dentro de la lista.

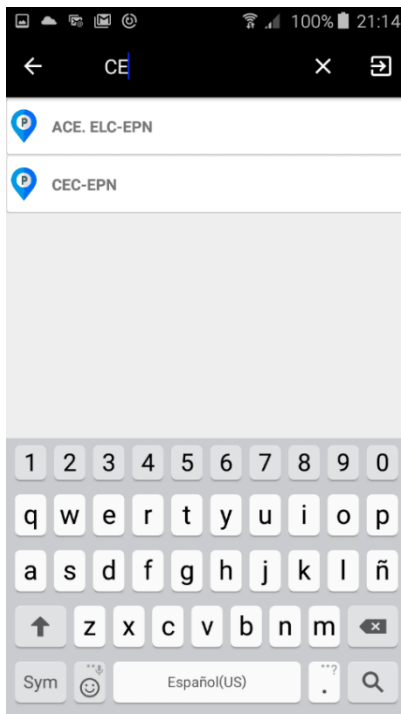


Figura 13. Búsqueda de un parqueadero.

La Figura 14 presenta la información de un parqueadero: Nombre de Parqueadero, Estado, Tipo de Parqueadero, Número de Plazas, Número de Pisos, Dirección, Tarifa Normal, Tarifa Especial y Horarios.



Figura 14. Visualización de Información del Parquederos.

La Figura 15 presenta la distribución y marca la disponibilidad de las plazas del subsuelo 1 del parqueadero del CEC.

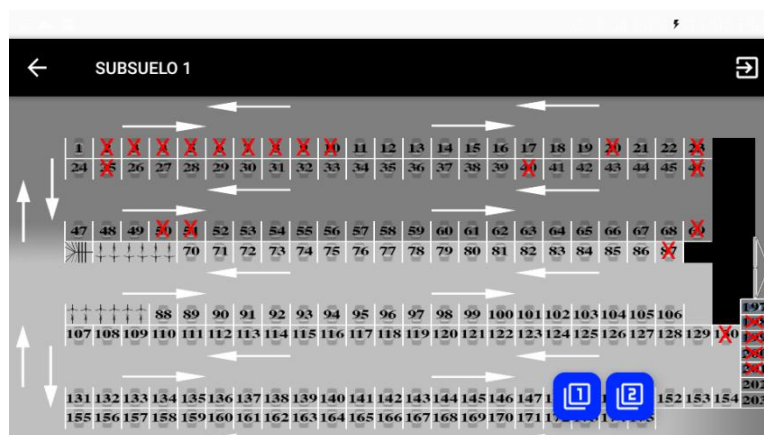


Figura 15. Visualización de Subsuelo 1 del Parquedero del CEC.

La Figura 16 presenta la distribución y marca la disponibilidad de las plazas del subsuelo 2 del parqueadero del CEC.

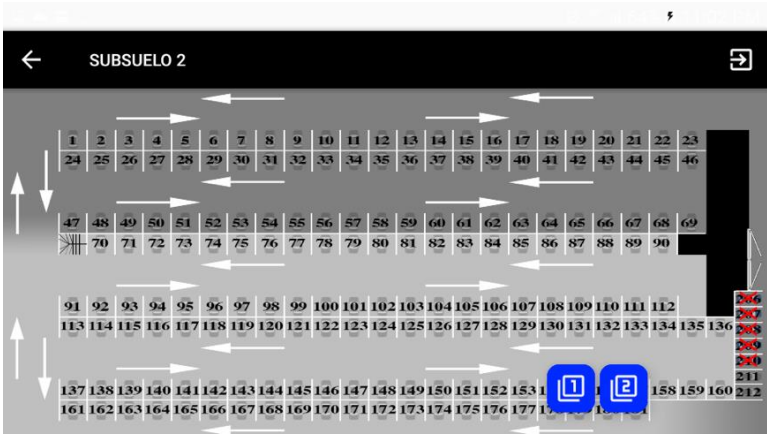


Figura 16. Visualización de Subsuelo 2 del Parqueadero del CEC.

La Figura 17 presenta una ruta definida con color azul en Google Maps entre la posición actual de un usuario y un parqueadero.

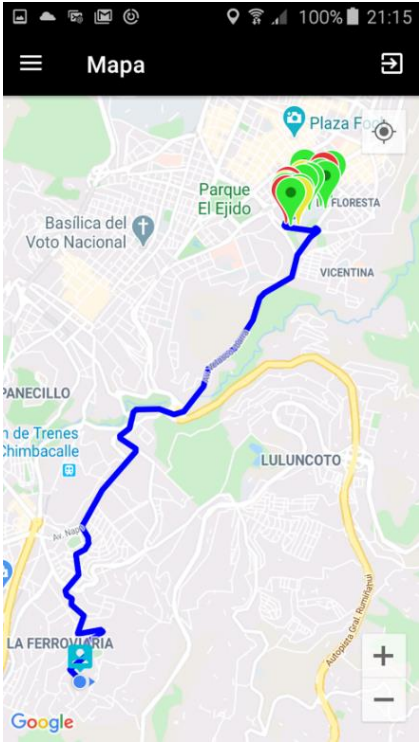


Figura 17. Visualización de una Ruta.

3.6.2 Sprint Review

Durante el desarrollo del Sprint 2 se logró cumplir con el objetivo planteado, la creación de la lista de los parqueadero y visualización de la información, la creación de los subsuelos del parqueadero del CEC en donde se complicó marcar la disponibilidad de las plazas de manera responsiva, finalmente se consiguió dibujar la ruta en Google Maps entre la posición actual del usuario y un parqueadero.

En la Tabla 24 se muestra si las actividades desarrolladas durante el sprint 2 fueron cumplidas.

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Cumplido
HU2.1	Diseño de interfaz para la información de parqueaderos - App Móvil	Visualizar la lista de parqueaderos.	SI
		Buscar un parqueadero dentro de la lista.	SI
		Mostrar la información de un parqueadero al seleccionar uno.	SI
HU2.2	Diseño de interfaz para las plazas de parqueadero CEC - App Móvil	Visualizar la distribución de los pisos del parqueadero del CEC.	SI
		Visualizar la distribución de las plazas del parqueadero del CEC.	SI
HU3.1	Implementación de ruta - App Móvil	Visualizar la ruta desde la posición actual hasta un determinado parqueadero.	SI

Tabla 24. Sprint Review 2.

Gráfico Burndown Sprint 2

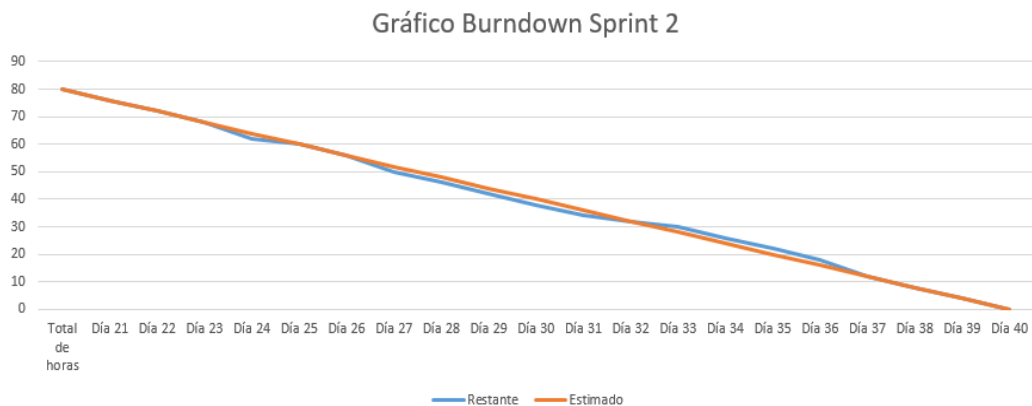


Figura 18. Gráfico Burndown Sprint 2.

Como se observa en la Figura 18 Gráfico Burndown Sprint 2, la primera semana se trabajó a un ritmo constante, en la segunda y tercera semana existen pequeños desfases ya que existieron adelantos y posteriormente atrasos, en el desarrollo, respectivamente. Finalmente, la cuarta semana se desarrolla de manera constante hasta terminar el Sprint. El desarrollo del Gráfico Burndown del Sprint 2 se puede observar con más detalle en el Anexo 4 - Gráficos Burndown.

3.6.3 Sprint Retrospective

El desarrollo del Sprint 2 se realizó sin mayores complicaciones a parte de los mencionados en el Sprint Review del presente Sprint, como se observa en el gráfico Burndown existieron desfases entre la semana 3 y 4, debido a la dificultad en la Historia de Usuario 2.2 que aumento su realización, pero finalmente a mitad de la semana 4 las actividades retomaron sus tiempos establecidos.

3.7 Sprint 3

3.7.1 Planificación

El objetivo del Sprint 3 es realizar el diseño y desarrollo de las interfaces de: acceso al aplicativo web, creación de cuentas de usuario, gestión de cuentas de usuario, recuperación y cambio de contraseña del aplicativo web.

Historias de Usuario del Sprint 3

HISTORIA DE USUARIO	HU5.1
Usuario: Administrador del sistema	
Título: Cuentas de usuario - App Web	
Descripción: Creación de cuentas de usuario para la administración de la información de los parqueaderos.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación:	
<ol style="list-style-type: none"> 1. La aplicación web permitirá crear cuentas de usuario con el perfil de "Administrador". 2. Se podrá crear una o más cuentas para un determinado parqueadero. 	

Tabla 25. Historia de Usuario 5.1.

HISTORIA DE USUARIO	HU5.2
Usuario: Administrador del sistema	
Título: Interfaz de acceso - App Web	
Descripción: Creación de la interfaz de inicio de sesión en la aplicación web.	
Prioridad: Media	Esfuerzo: 5
Criterios de aceptación:	
<ol style="list-style-type: none"> 1. Validación de usuario y contraseña. 2. Ingreso a la interfaz principal. 	

Tabla 26. Historia de Usuario 5.2.

HISTORIA DE USUARIO	HU5.3
Usuario: Administrador del sistema	
Título: Interfaz de Recuperación de contraseña - App Web	
Descripción: Creación de la interfaz de recuperación de contraseña.	
Prioridad: Baja	Esfuerzo: 3
Criterios de aceptación:	
<ol style="list-style-type: none"> 1. Envío de mensaje de contraseña temporal al correo del usuario. 2. Validación de correo electrónico o Nick. 	

Tabla 27. Historia de Usuario 5.3.

HISTORIA DE USUARIO	HU7.1
Usuario: Administrador del sistema	
Título: Interfaz Gestión de cuentas de usuario - App Web	
Descripción: Diseño de interfaz para gestionar una determinada cuenta de usuario, como actualizar datos o eliminación de cuentas	
Prioridad: Media	Esfuerzo: 5
Criterios de aceptación:	

1. La aplicación web permitirá modificar una determinada cuenta de usuario.
2. La aplicación web permitirá activar y desactivar una determinada cuenta de usuario.

Tabla 28. Historia de Usuario 7.1.

Sprint Backlog Sprint 3

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Actividades
HU5.1	Cuentas de usuario - App Web	La aplicación web permitirá crear cuentas de usuario con el perfil de "Administrador"	<ul style="list-style-type: none"> • Creación de la Interfaz para registrar usuarios. • Consumo de servicios del Backend. • Validaciones.
		Se podrá crear una o más cuentas para un determinado parqueadero	<ul style="list-style-type: none"> • Creación de interfaz para crear nuevos usuarios asociados a un parqueadero. • Creación y consumo de servicios del Backend.
HU5.2	Interfaz de acceso - App Web	Validación de usuario y contraseña	<ul style="list-style-type: none"> • Creación de Interfaz de acceso - login. • Validación de acceso mediante correo y contraseña (hash)
		Ingreso a la interfaz principal	<ul style="list-style-type: none"> • Enlace con la ruta de la interfaz principal.
HU5.3	Interfaz de Recuperación de contraseña - App Web	Envío de mensaje de contraseña temporal al correo del usuario	<ul style="list-style-type: none"> • Creación de interfaz para recuperación de contraseña. • Creación y configuración de correo electrónico de Gmail. • Creación y consumo de servicios del Backend
		Validación de correo electrónico o Nick.	<ul style="list-style-type: none"> • Validación de usuario registrado con el correo o Nick ingresado.

HU7.1	Gestión de cuentas de usuario - App Web	La aplicación web permitirá modificar una determinada cuenta de usuario	<ul style="list-style-type: none"> • Creación de la interfaz para visualizar y modificar los usuarios registrados.
		La aplicación web permitirá activar y desactivar una determinada cuenta de usuario	<ul style="list-style-type: none"> • Validación de usuario desactivado al momento de intentar acceder al aplicativo web.
		Notificar del cambio de contraseña al usuario vía correo electrónico	<ul style="list-style-type: none"> • Consumo de servicio Backend

Tabla 29. Sprint Backlog Sprint 3.

Prototipo

A continuación, se presenta el resultado de las actividades realizadas en el Sprint 3 y el prototipo de las siguientes interfaces:

La Figura 19 presenta la interfaz para el registro de un usuario nuevo, el mismo que tendrá el perfil de Administrador del sistema.

The image shows a web browser window with the URL '18.224.229.129:3000/#/registro'. The page content is a registration form on a dark blue background. The form has the following fields:

- Nombre:** Input field with a person icon.
- Apellido:** Input field with a person icon.
- Cédula:** Input field with a document icon.
- Contacto:** Input field with a phone icon.
- Correo:** Input field with an envelope icon.
- Nick:** Input field with a person icon.
- Contraseña:** Input field with a lock icon.
- Confirmar Contraseña:** Input field with a lock icon.

A blue button labeled 'Registrarse' is positioned below the password fields. At the top of the form, there is a link 'o bien, inicia sesión'.

Figura 19. Interfaz Registro de Usuario Administrador.

La Figura 20 presenta la Interfaz de acceso al sistema, se validan el correo y la contraseña para el ingreso.

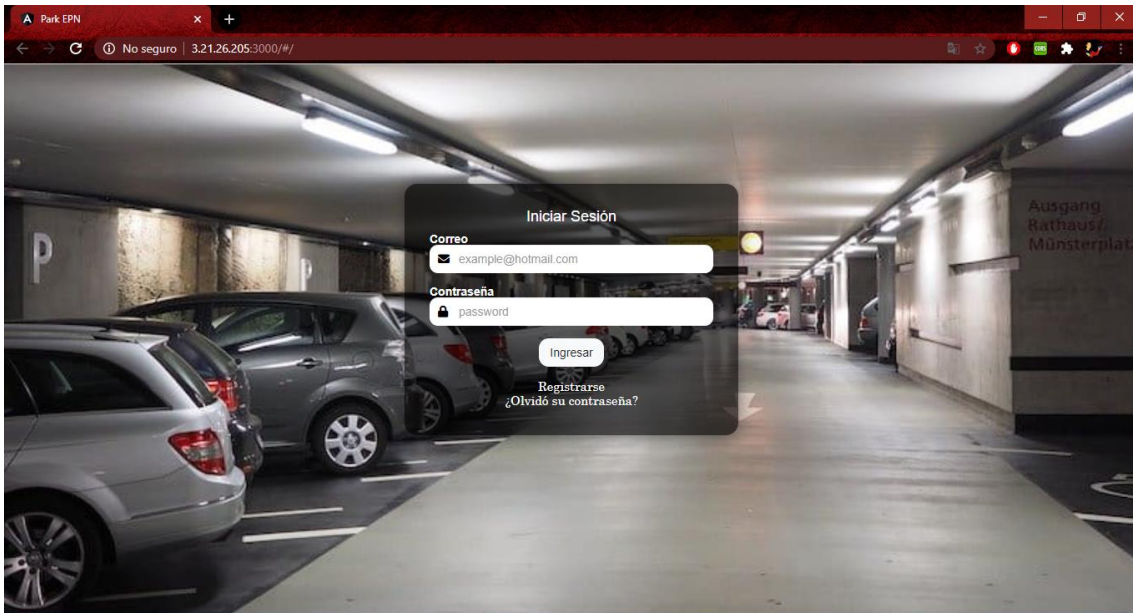


Figura 20. Interfaz de Acceso.

La Figura 21 presenta la Interfaz para el registro de usuarios los mismos que pueden ser Administradores u Operadores de los ciertos parqueaderos.

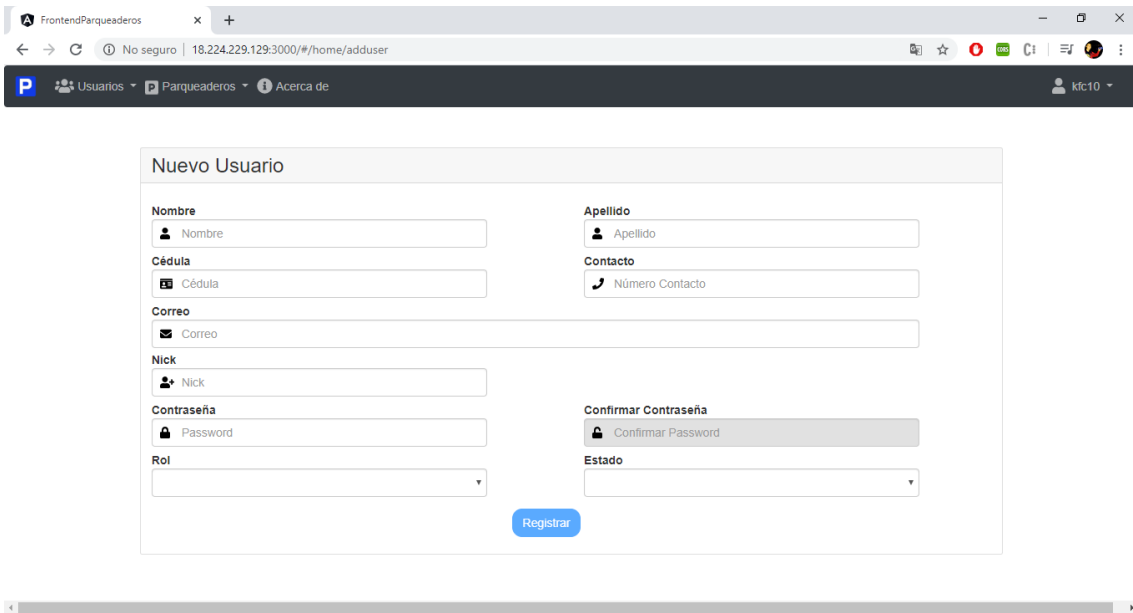


Figura 21. Interfaz de Registro de Usuarios para determinados parqueaderos.

La Figura 22 presenta la Interfaz para recuperar la contraseña de una determinada cuenta.

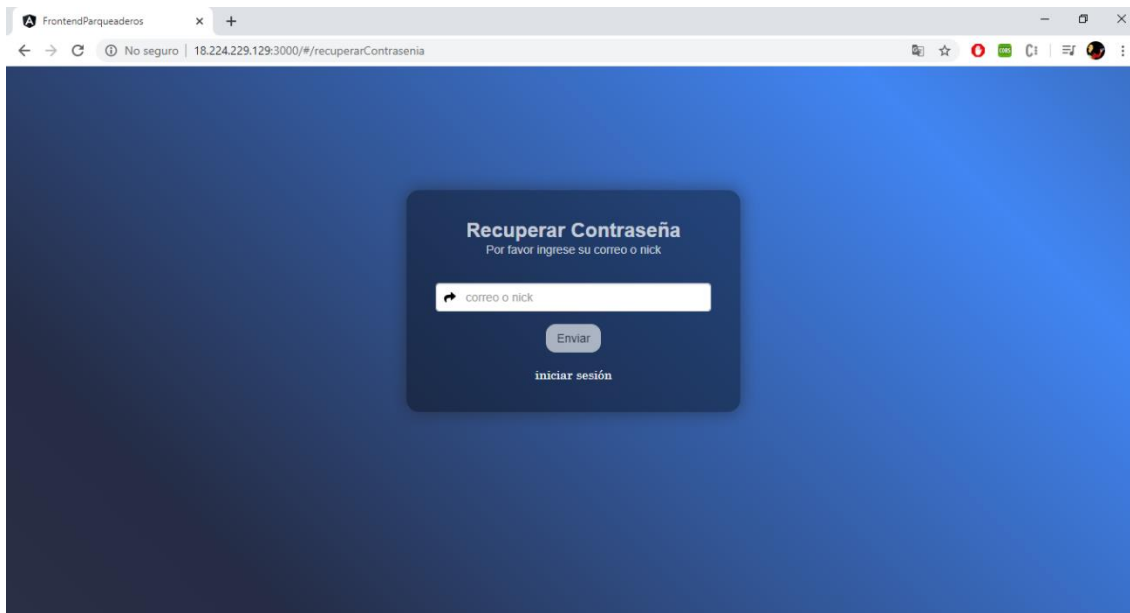


Figura 22. Interfaz Recuperación de Contraseña.

La Figura 23 presenta la Interfaz para visualizar la información de un determinado usuario del sistema.

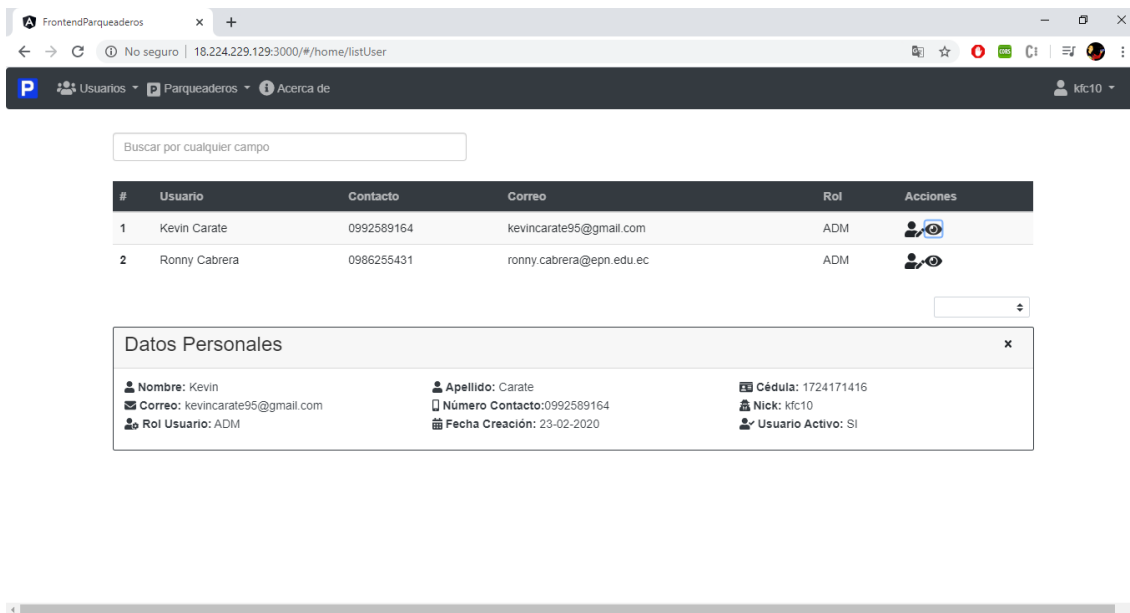


Figura 23. Interfaz Visualización de Datos de Usuario.

La Figura 24 presenta la Interfaz para actualizar la información de un determinado Usuario del sistema.

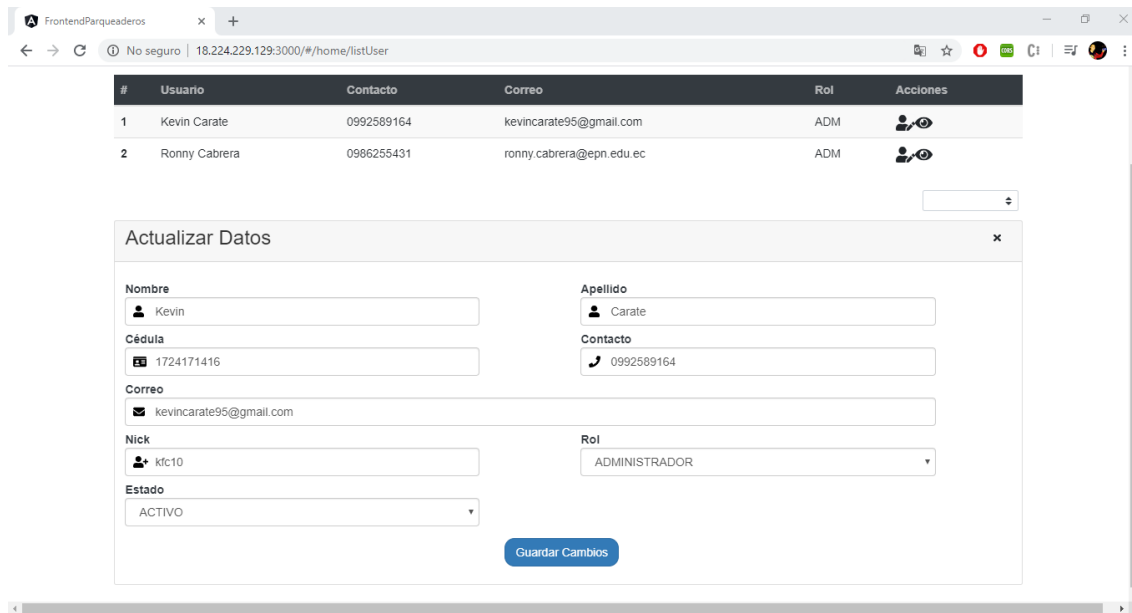


Figura 24. Interfaz Actualización de Datos de Usuario.

3.7.2 Sprint Review

Durante el desarrollo del Sprint 3 se logró cumplir con el objetivo planteado, las interfaces que presentaron mayor esfuerzo en su desarrollo fueron: Interfaz de registro de usuarios y la Interfaz de acceso o Login, para las interfaces de registro de usuarios asociados a un parqueadero y actualización de datos se tomó el modelo del componente registro de usuarios.

Las complicaciones que se presentaron, durante el desarrollo de las interfaces, fueron con respecto al diseño responsive de las interfaces y principalmente en el diseño de tablas. Finalmente se tuvo que configurar el correo electrónico para el envío de notificaciones vía email para la recuperación de contraseña.

En la Tabla 30 se muestra si las actividades desarrolladas durante el sprint 3 fueron cumplidas.

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Cumplido
HU5.1	Cuentas de usuario - App Web	La aplicación web permitirá crear cuentas de usuario con el perfil de "Administrador"	SI
		Se podrá crear una o más cuentas para un determinado parqueadero	SI

HU5.2	Interfaz de acceso - App Web	Validación de usuario y contraseña	SI
		Ingreso a la interfaz principal	SI
HU5.3	Interfaz de Recuperación de contraseña - App Web	Envío de mensaje de contraseña temporal al correo del usuario	SI
		Validación de correo electrónico o Nick.	SI
HU7.1	Gestión de cuentas de usuario - App Web	La aplicación web permitirá modificar una determinada cuenta de usuario	SI
		La aplicación web permitirá activar y desactivar una determinada cuenta de usuario	SI
		Notificar del cambio de contraseña al usuario vía correo electrónico	SI

Tabla 30. Sprint Review 3.

Gráfico Burndown Sprint 3

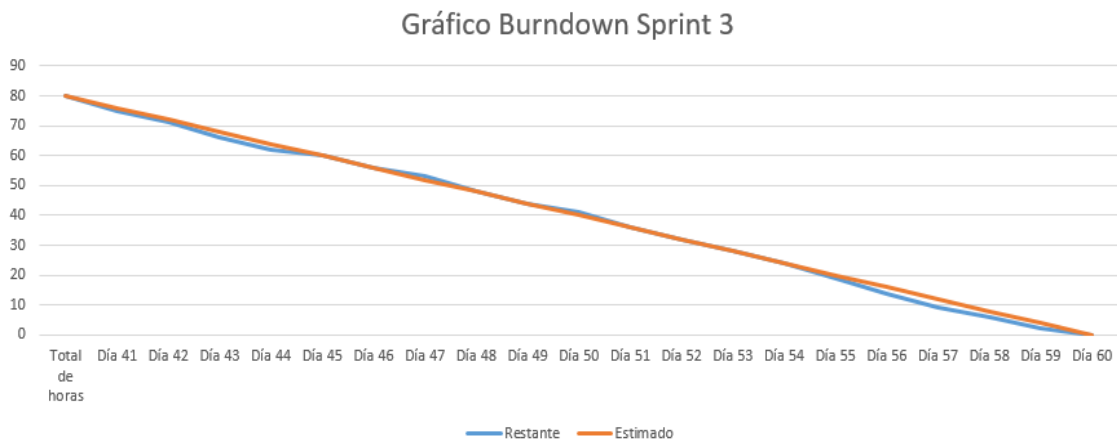


Figura 25. Gráfico Burndown Sprint 3.

Como se observa en la Figura 25 Gráfico Burndown Sprint 3, la mayoría del desarrollo del Sprint 3 se lo realizó de una manera constante, existen desfases entre los días 42, 43, 57, 58 y 59 en los mismos que se trabajó un poco más de tiempo, pero posteriormente se equilibraba el esfuerzo realizado en días posteriores. El desarrollo del Gráfico Burndown del Sprint 3 se puede observar con más detalle en el Anexo 4 - Gráficos Burndown.

3.7.3 Sprint Retrospective

El desarrollo del Sprint 3 se realizó sin mayores complicaciones a parte de los mencionados en el Sprint Review del presente Sprint, durante todo su desarrollo se trabajó a un ritmo constante, esto implica que se realizó una mejor planificación con respecto a los Sprint 1 y 2.

3.8 Sprint 4

3.8.1 Planificación

El Objetivo del Sprint 4 es realizar el diseño y desarrollo de las interfaces de: Registro de parqueaderos, registro de plazas correspondientes a un determinado parqueadero y gestión de la información de parqueaderos.

Historias de Usuario del Sprint 4

HISTORIA DE USUARIO	HU6.1
Usuario: Administrador del sistema.	
Título: Interfaz de registro de plazas y sensores - App Web	
Descripción: Diseño de interfaz para registrar todas las plazas de un determinado parqueadero junto con su sensor asignado.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación: <ol style="list-style-type: none">1. Ingreso del total de plazas por cada piso que tenga el parqueadero.2. Permitir escoger que tipo de plaza.3. El código del sensor es único para cada una de las plazas de un parqueadero.4. Una plaza debe estar asociada con un único sensor.5. Se debe ingresar como datos de un sensor su código y dirección ip, su estado inicial (opcional).	

Tabla 31. Historia de Usuario 6.1.

HISTORIA DE USUARIO	HU8.1
Usuario: Administrador del sistema.	

Título: Interfaz de creación de parqueaderos - App Web	
Descripción: Diseño de la interfaz para ingresar la información de un nuevo parqueadero.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación:	
<ol style="list-style-type: none"> 1. Registro de la información de un nuevo parqueadero. 2. Ingreso de las coordenadas (ubicación geográfica) de un parqueadero (latitud, longitud). 	

Tabla 32. Historia de Usuario 8.1.

HISTORIA DE USUARIO	HU8.3
Usuario: Administrador del sistema.	
Título: Interfaz Gestión de información de parqueaderos - App Web	
Descripción: Diseño de la interfaz que contendrá información de un determinado parqueadero.	
Prioridad: Alta	Esfuerzo: 8
Criterios de aceptación:	
<ol style="list-style-type: none"> 1. Visualizar la información de un determinado parqueadero. 2. Actualizar información de un determinado parqueadero. 	

Tabla 33. Historia de Usuario 8.3.

Sprint Backlog Sprint 4

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Actividades
HU6.1	Interfaz de registro de plazas y sensores - App Web	Ingreso del total de plazas por cada piso que tenga el parqueadero	<ul style="list-style-type: none"> • Creación de Interfaz para registro de plazas. • Consumo de Servicios. • Validaciones.
		Permitir escoger que tipo de plaza.	<ul style="list-style-type: none"> • Algoritmo para ver el estado actual.

		El código del sensor es único para cada una de las plazas de un parqueadero.	<ul style="list-style-type: none"> • Creación de interfaz con la información generada automáticamente. • Validaciones
		Una plaza debe estar asociada con un único sensor	<ul style="list-style-type: none"> • Consumo de servicios
		Se debe ingresar como datos de un sensor su código y dirección ip, su estado inicial (opcional).	<ul style="list-style-type: none"> • Consumo de servicios
HU8.1	Interfaz de Creación de Parqueaderos – App Web	Registro de la información de un nuevo parqueadero.	<ul style="list-style-type: none"> • Creación de Interfaz para registro de parqueadero. • Creación de Interfaz para registro de horarios. • Consumo de servicios. • Validaciones
		Ingreso de las coordenadas (ubicación geográfica) de un parqueadero (latitud, longitud).	<ul style="list-style-type: none"> • Validaciones
HU8.3	Interfaz Gestión de Información de Parqueaderos - App Web	Visualizar la información de un determinado parqueadero.	<ul style="list-style-type: none"> • Creación de Interfaz de visualización de datos de un parqueadero. • Consumo de servicios
		Actualizar información de un determinado parqueadero.	<ul style="list-style-type: none"> • Creación de Interfaz de actualización de datos de un parqueadero. • Consumo de Servicios. • Validaciones.

Tabla 34. Sprint Backlog Sprint 4.

Prototipo

A continuación, se presenta el resultado de las actividades realizadas en el Sprint 4 y el prototipo de las siguientes interfaces:

La Figura 26 presenta la Interfaz para el registro de un nuevo parqueadero en el Sistema.

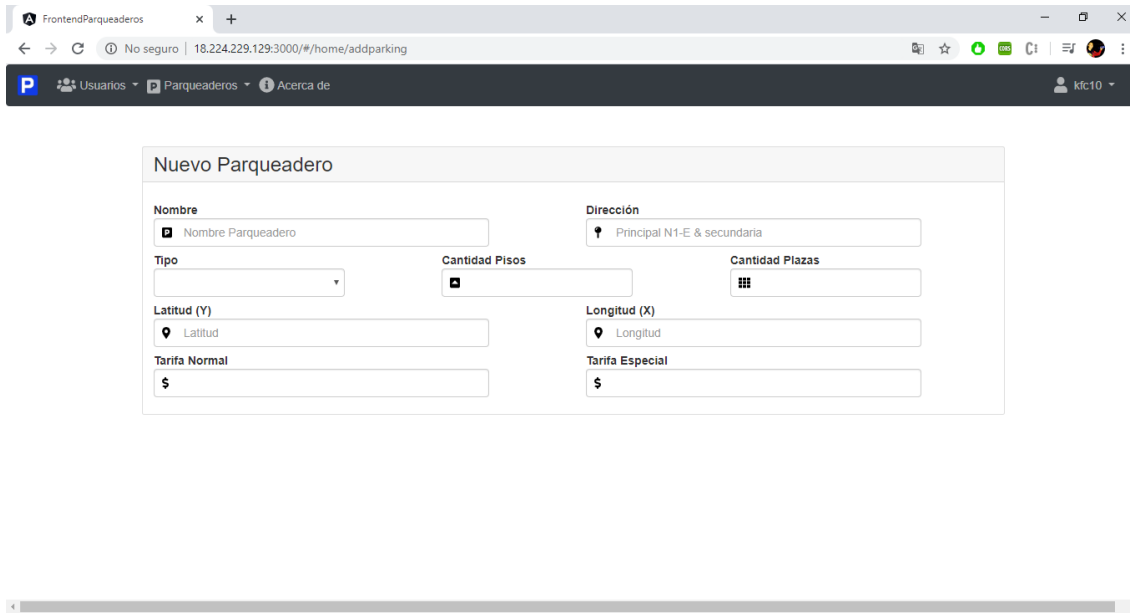


Figura 26. Interfaz Registro de Parquaderos.

La Figura 27 presenta la Interfaz para el registro de horarios de disponibilidad de un parquadero.

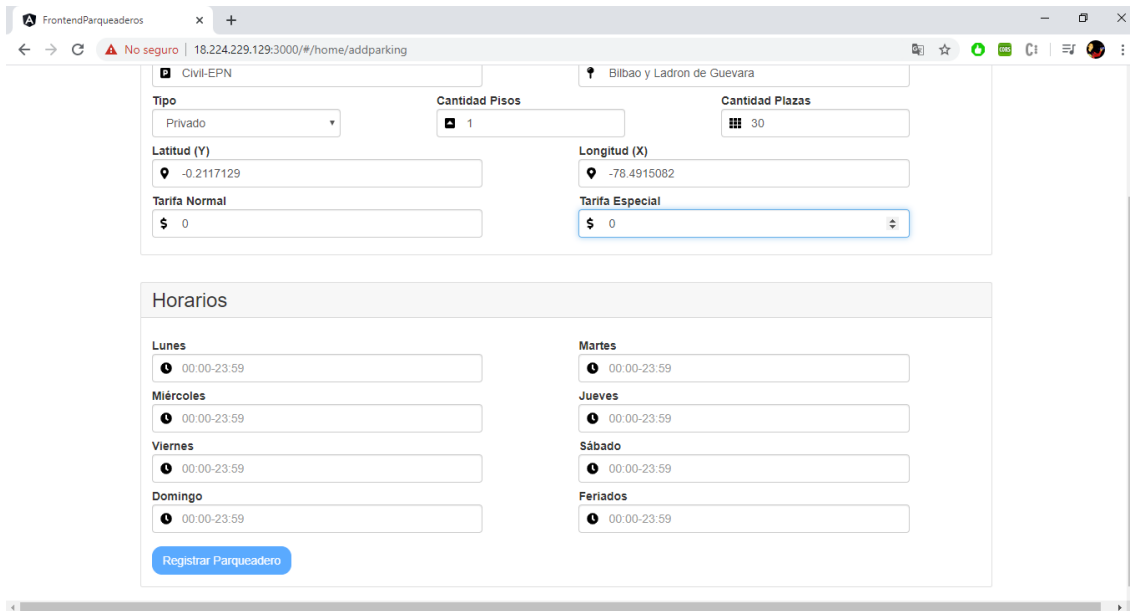


Figura 27. Interfaz Registro de Horarios.

La Figura 28 presenta la Interfaz de visualización de todos los parquaderos registrados en el sistema.

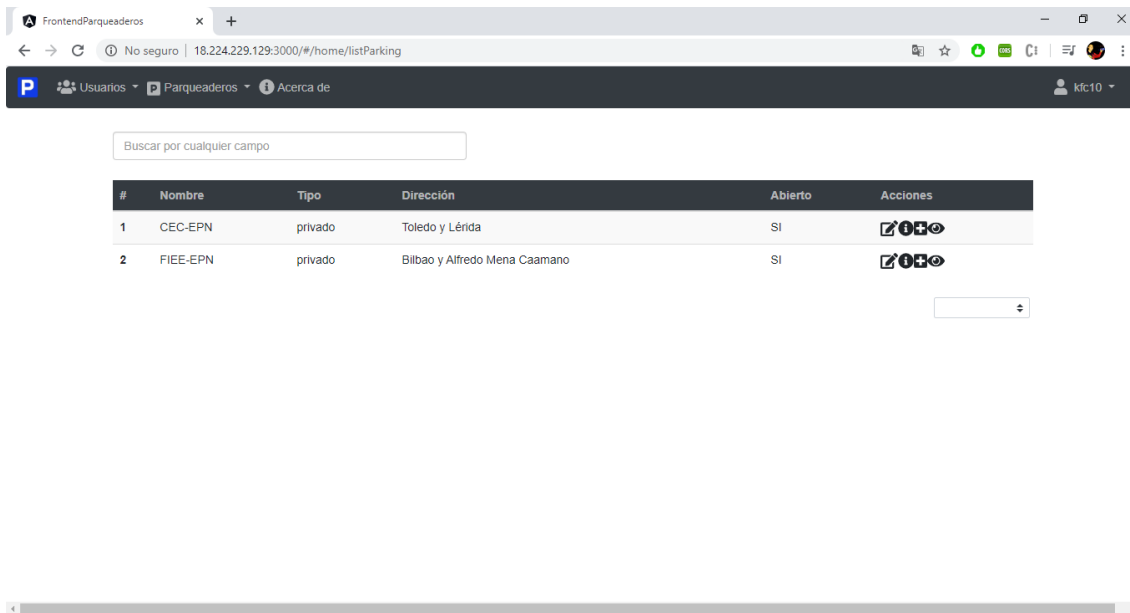


Figura 28. Interfaz Visualización de Parqueaderos.

La Figura 29 presenta la Interfaz de visualización de datos de un determinado parqueadero registrado.

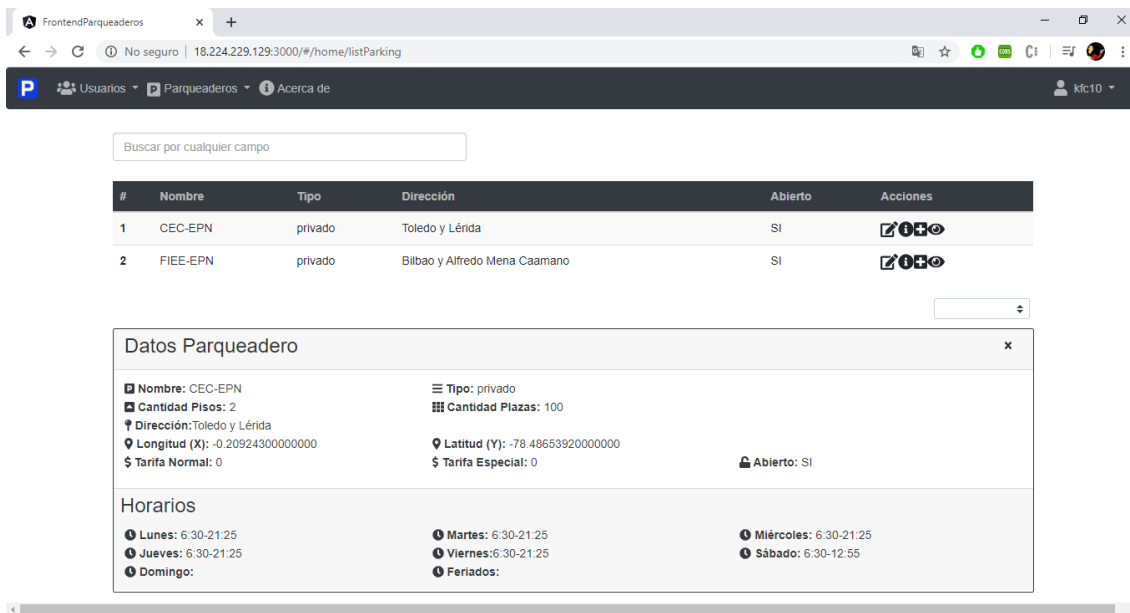


Figura 29. Interfaz Visualización Datos Parqueaderos.

La Figura 30 presenta la Interfaz para actualizar los datos de un determinado parqueadero registrado.

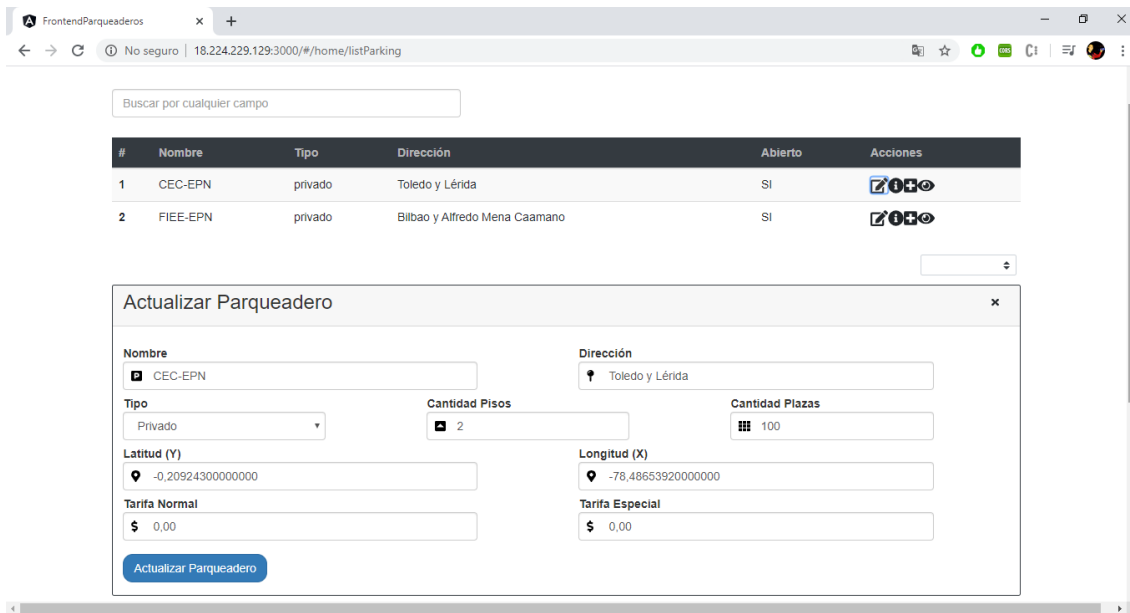


Figura 30. Interfaz Actualización de Datos.

La Figura 31 presenta la Interfaz para actualizar los horarios de un determinado parqueadero registrado.

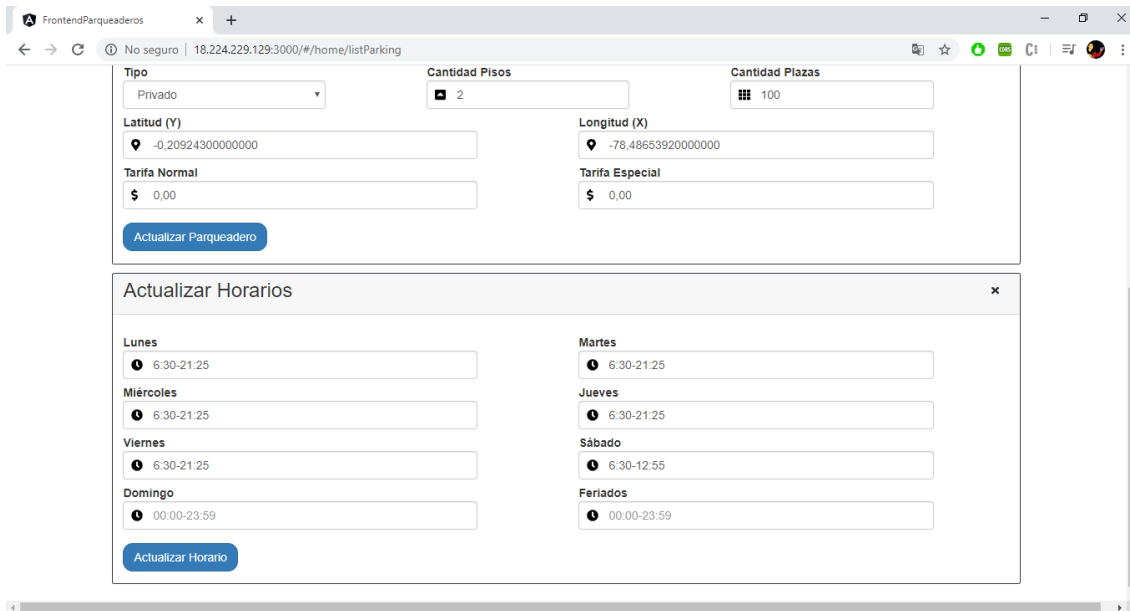


Figura 31. Interfaz Actualización de Horarios.

La Figura 32 presenta la interfaz de registro de plazas de un determinado parqueadero por cada piso que tenga dicho parqueadero.

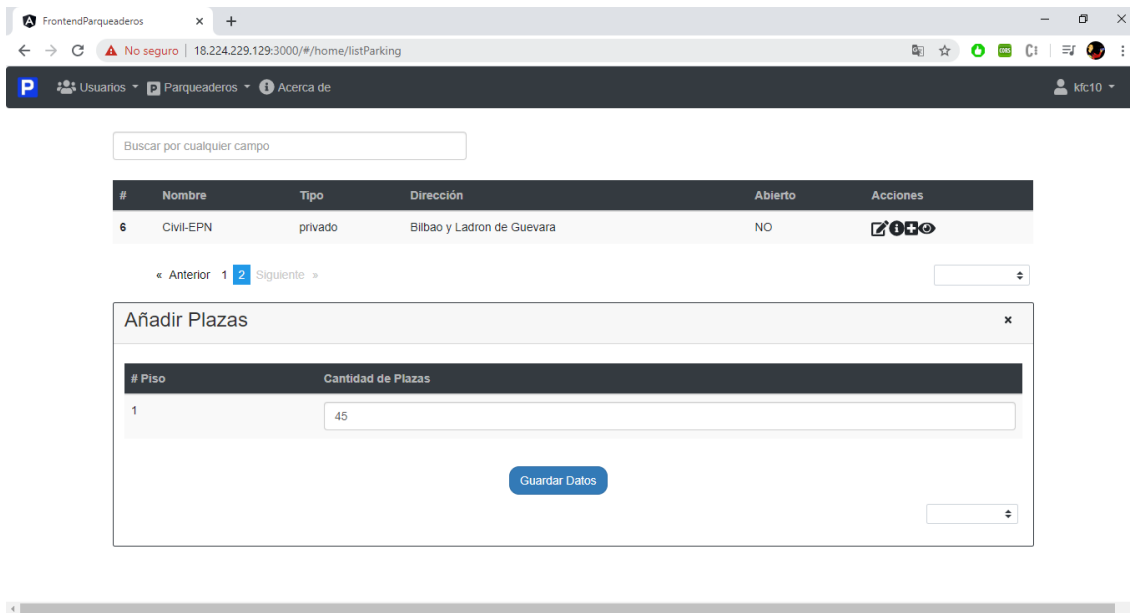


Figura 32. Interfaz Registro de Plazas.

La Figura 33 presenta la interfaz para asignar el tipo de plaza y la ip del sensor, los datos son generados automáticamente y sólo se pueden cambiar el tipo de plaza y la ip del sensor.

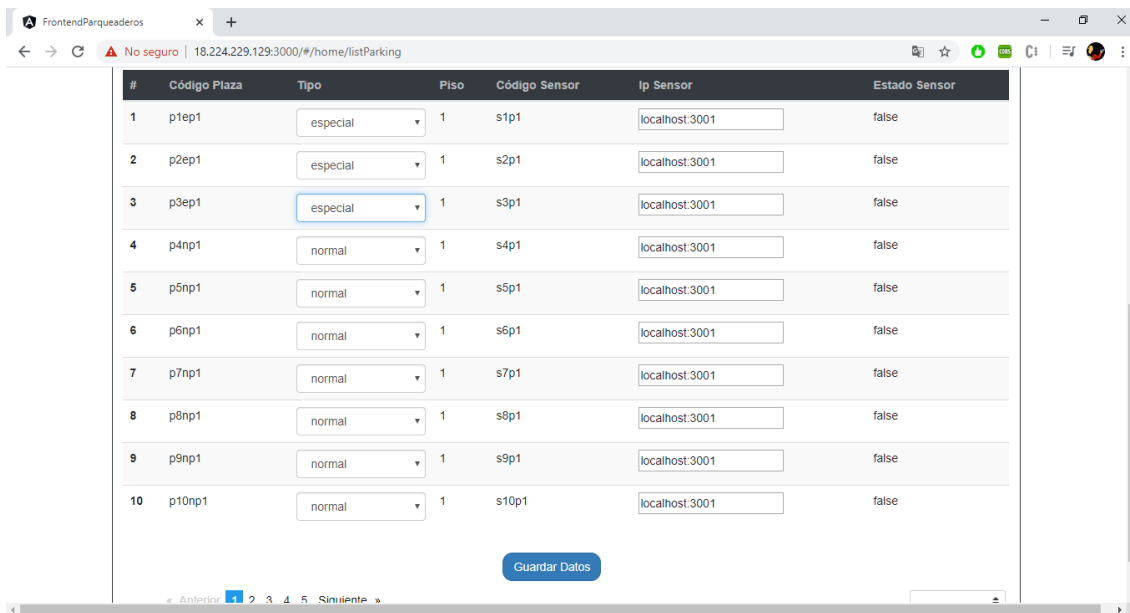


Figura 33. Interfaz de Asignación de Tipo de Plaza.

3.8.2 Sprint Review

Durante el desarrollo del Sprint 4 se logró cumplir con el objetivo planteado, las interfaces que presentaron mayor esfuerzo en su desarrollo fueron: Interfaz de registro de parqueaderos y registro de plazas, se debió controlar el flujo de datos y realizarlos como una sola transacción en bloque; las interfaces de visualización de datos y actualización no presentaron mayor complicación debido a que se basaron en interfaces anteriores.

Las complicaciones que se presentaron, durante el desarrollo de las interfaces, fueron con respecto a la validación de los datos en tablas, esto porque tenemos que recorrer la tabla e ir validando los datos de cada fila.

En la Tabla 35 se muestra si las actividades desarrolladas durante el sprint 4 fueron cumplidas.

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Cumplido
HU6.1	Interfaz de registro de plazas y sensores - App Web	Ingreso del total de plazas por cada piso que tenga el parqueadero	SI
		Permitir escoger que tipo de plaza.	SI
		El código del sensor es único para cada una de las plazas de un parqueadero.	SI
		Una plaza debe estar asociada con un único sensor	SI
		Se debe ingresar como datos de un sensor su código y dirección ip, su estado inicial (opcional).	SI
HU8.1	Interfaz de Creación de Parqueaderos – App Web	Registro de la información de un nuevo parqueadero.	SI
		Ingreso de las coordenadas (ubicación geográfica) de un parqueadero (latitud, longitud).	SI
HU8.3	Interfaz Gestión de Información de Parqueaderos - App Web	Visualizar la información de un determinado parqueadero.	SI
		Actualizar información de un determinado parqueadero.	SI

Tabla 35. Sprint Review 4.

Gráfico Burndown Sprint 4

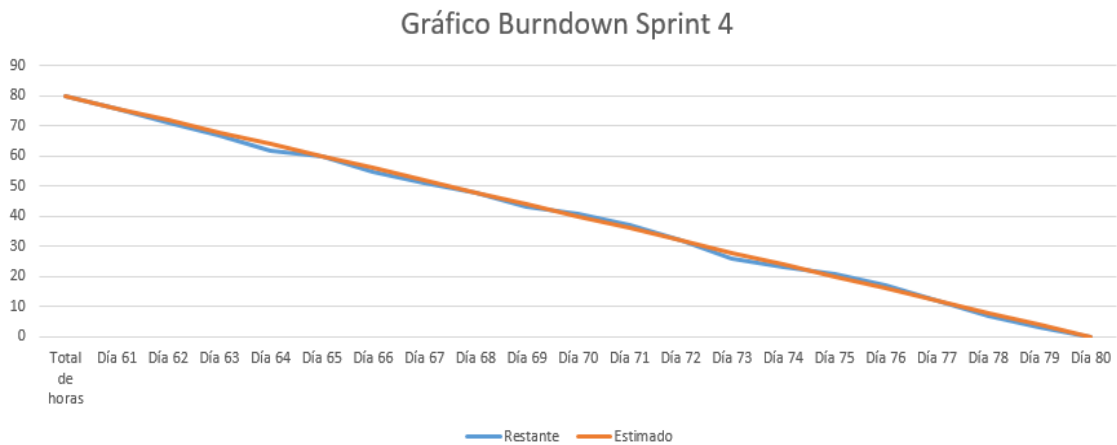


Figura 34. Gráfico Burndown Sprint 4.

Como se observa en la Figura 34 Gráfico Burndown Sprint 4, la mayoría del desarrollo del Sprint 4 se lo realizó de una manera constante, existen desfases en los días 63, 66, 73, 78 y 79 en los que se trabajó un poco más de tiempo, por otro lado, los días 70 y 71 reflejan que no se trabajó completamente, finalmente podemos observar que el esfuerzo realizado ha permanecido equilibrado durante todo el desarrollo del Sprint. El desarrollo del Gráfico Burndown del Sprint 4 se puede observar con más detalle en el Anexo 4 - Gráficos Burndown.

3.8.3 Sprint Retrospective

El desarrollo del Sprint 4 se realizó sin mayores complicaciones a parte de los mencionados en el Sprint Review del presente Sprint, durante todo su desarrollo se trabajó a un ritmo constante muy similar a lo ocurrido en el sprint 3 en el cual se realizó una mejor planificación con respecto a los dos primeros Sprints.

3.9 Sprint 5

3.9.1 Planificación

El Objetivo del Sprint 5 es realizar el diseño y desarrollo de las interfaces de: Cambio de contraseña de un determinado usuario, actualización de plazas y sensores de un determinado parqueadero, ayuda e información tanto del aplicativo móvil como web y finalmente programar y configurar el sensor que servirá para obtener el estado de una plaza en tiempo real.

Historias de Usuario del Sprint 5

HISTORIA DE USUARIO	HU1.4
Usuario: Usuario del sistema	
Título: Configuración y Programación del Sensor	
Descripción: Configuración y programación del sensor para visualizar en tiempo real es estado de una determinada plaza	
Prioridad: Media	Esfuerzo: 5
Criterios de aceptación: 1. Actualizar automáticamente el estado de la plaza a la que es asignado el sensor.	

Tabla 36. Historia de Usuario 1.4.

HISTORIA DE USUARIO	HU4.1
Usuario: Usuario del sistema	
Título: Diseño interfaz de información y ayuda - App Móvil.	
Descripción: Diseño de la interfaz que contendrá información y ayuda acerca del manejo de la aplicación móvil.	
Prioridad: Baja	Esfuerzo: 3
Criterios de aceptación: 1. Visualizar información sobre el uso y manejo de la aplicación móvil.	

Tabla 37. Historia de Usuario 4.1.

HISTORIA DE USUARIO	HU7.2
Usuario: Administrador del sistema	
Título: Interfaz Cambio de contraseña - App Web	
Descripción: Implementación del cambio de contraseña de una determinada cuenta de usuario.	
Prioridad: Baja	Esfuerzo: 3

<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. cambio de contraseña de una determinada cuenta de usuario. 2. notificar del cambio de contraseña al usuario vía correo electrónico.
--

Tabla 38. Historia de Usuario 7.2.

HISTORIA DE USUARIO	HU8.2
Usuario: Administrador del sistema.	
Título: Interfaz de actualización de plazas y sensores - App Web	
Descripción: Diseño de la interfaz para actualizar la información correspondiente a plazas con su respectivo sensor.	
Prioridad: Alta	Esfuerzo: 8
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Se debe poder cambiar el tipo de plaza. 2. Se debe poder cambiar el estado de una plaza. 3. Se debe poder cambiar la dirección ip del sensor. 	

Tabla 39. Historia de Usuario 8.2.

HISTORIA DE USUARIO	HU9.1
Usuario: Administrador del sistema.	
Título: Diseño interfaz de información y ayuda - App Web	
Descripción: Diseño de la interfaz que contendrá información y ayuda acerca del manejo de la aplicación web	
Prioridad: Baja	Esfuerzo: 3
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Visualizar información sobre el contenido y uso de la aplicación web. 	

Tabla 40. Historia de Usuario 9.1.

Sprint Backlog Sprint 5

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Actividades
HU1.4	Configuración y Programación del Sensor	Actualizar automáticamente el estado de la plaza a la que es asignado el sensor.	<ul style="list-style-type: none"> • Conexión de sensor con la Raspberry. • Desarrollo del programa para medir la distancia. • Conexión con Backend. • Creación demonio para ejecutar el script automáticamente al reinicio de la Raspberry.
HU4.1	Diseño interfaz de información y ayuda - App Móvil	Visualizar información sobre el uso y manejo de la aplicación móvil	<ul style="list-style-type: none"> • Consumo de página web del Frontend para información de aplicación y ayuda. • Creación de una ventana para mostrar información acerca de la aplicación. • Creación de una ventana para mostrar información de ayuda de la aplicación.
HU7.2	Interfaz Cambio de contraseña - App Web	cambio de contraseña de una determinada cuenta de usuario	<ul style="list-style-type: none"> • Creación de Interfaz de Perfil de Usuario. • Creación de Interfaz de cambio de contraseña. • Consumo de servicios. • Validaciones.
		notificar del cambio de contraseña al usuario vía correo electrónico	<ul style="list-style-type: none"> • Consumo de servicios. • Validaciones.
HU8.2	Interfaz de actualización de plazas y sensores - App Web	Se debe poder cambiar el tipo de plaza	<ul style="list-style-type: none"> • Creación de Interfaz para ver información de una determinada plaza. • Creación de Interfaz para actualización de datos. • Consumo de Servicios. • Validaciones.
		Se debe poder cambiar el estado de una plaza	

HU9.1	Diseño interfaz de información y ayuda - App Web	Visualizar información sobre el contenido y uso de la aplicación web	<ul style="list-style-type: none"> • Creación de Interfaz de Información.
-------	--	--	--

Tabla 41. Sprint Backlog Sprint 5.

Prototipo

A continuación, se presenta el resultado de las actividades realizadas en el Sprint 5 y el prototipo de las siguientes interfaces:

La Figura 35 presenta la página web con la información Acerca de la aplicación móvil.

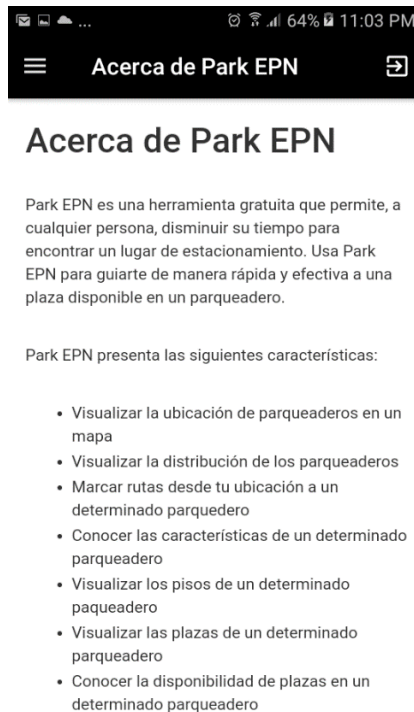


Figura 35. Información Acerca de la aplicación móvil.

La Figura 36 presenta la página web con la información de Ayuda de la aplicación móvil.

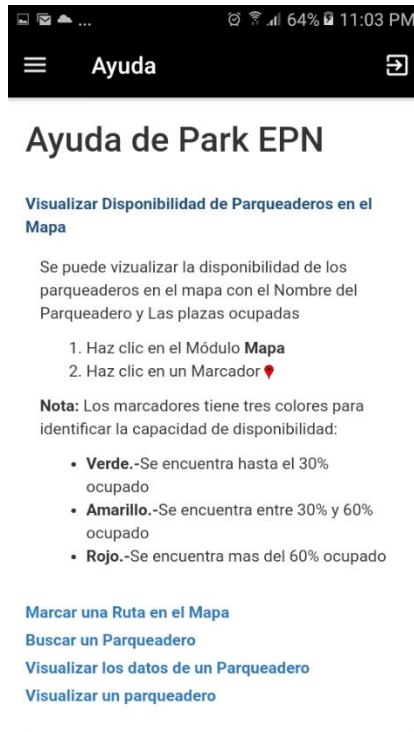


Figura 36. Información de Ayuda para la aplicación móvil.

La Figura 37 presenta la Interfaz para visualizar la información de una determinada plaza junto a la información de su sensor.

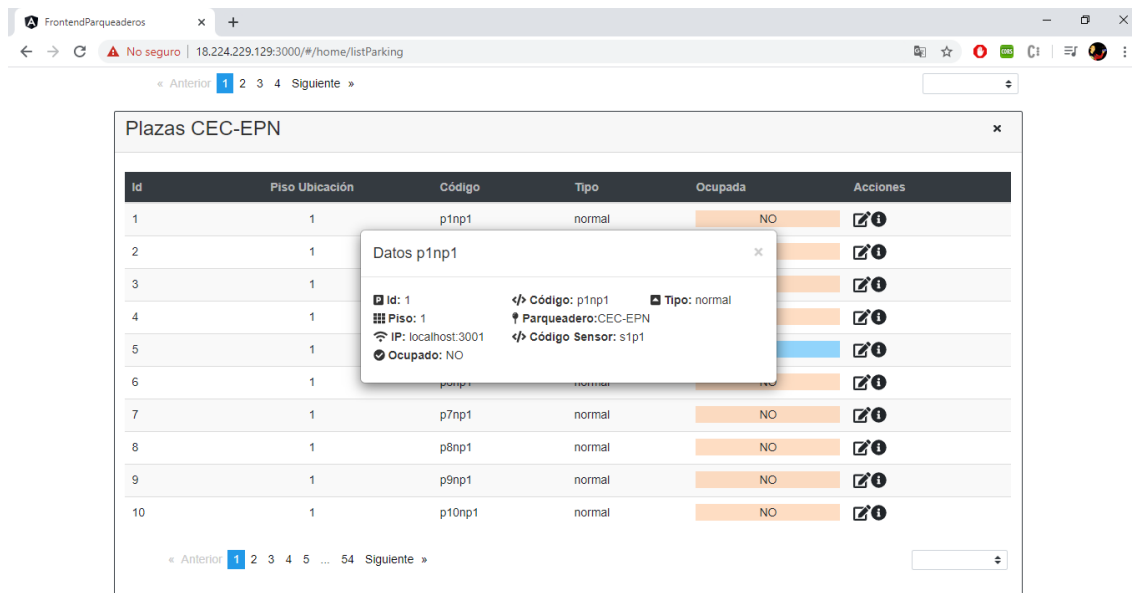


Figura 37. Interfaz de Información de una determinada Plaza.

La Figura 38 presenta la Interfaz para actualizar datos de una determinada plaza y los datos correspondientes al sensor de dicha plaza.

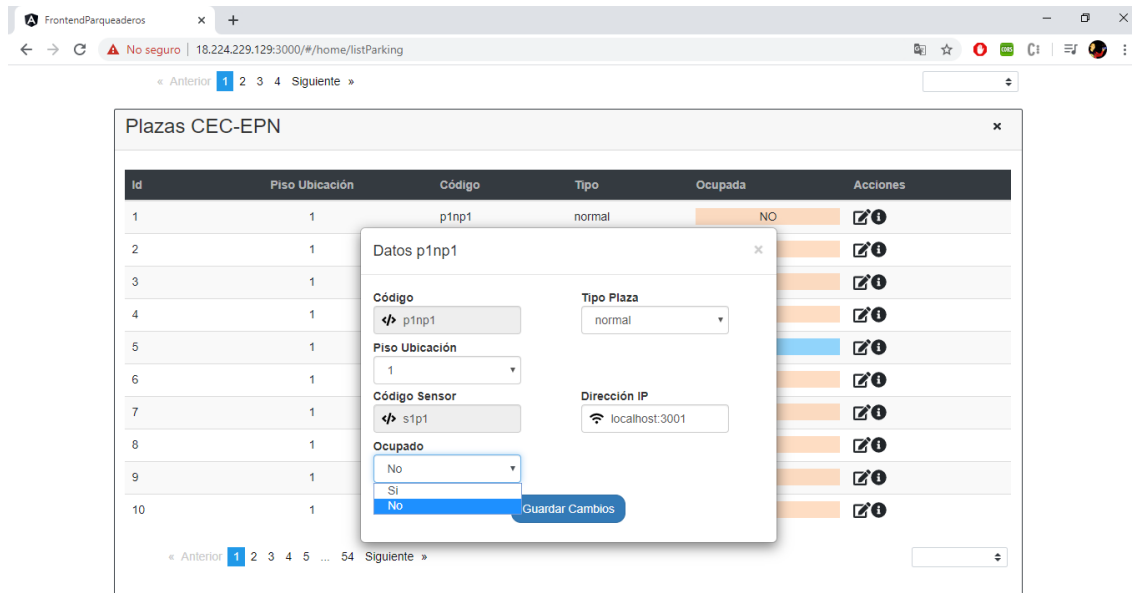


Figura 38. Interfaz de Actualización de Datos de una Plaza.

La Figura 39 presenta la Interfaz para ver el perfil de usuario logueado.

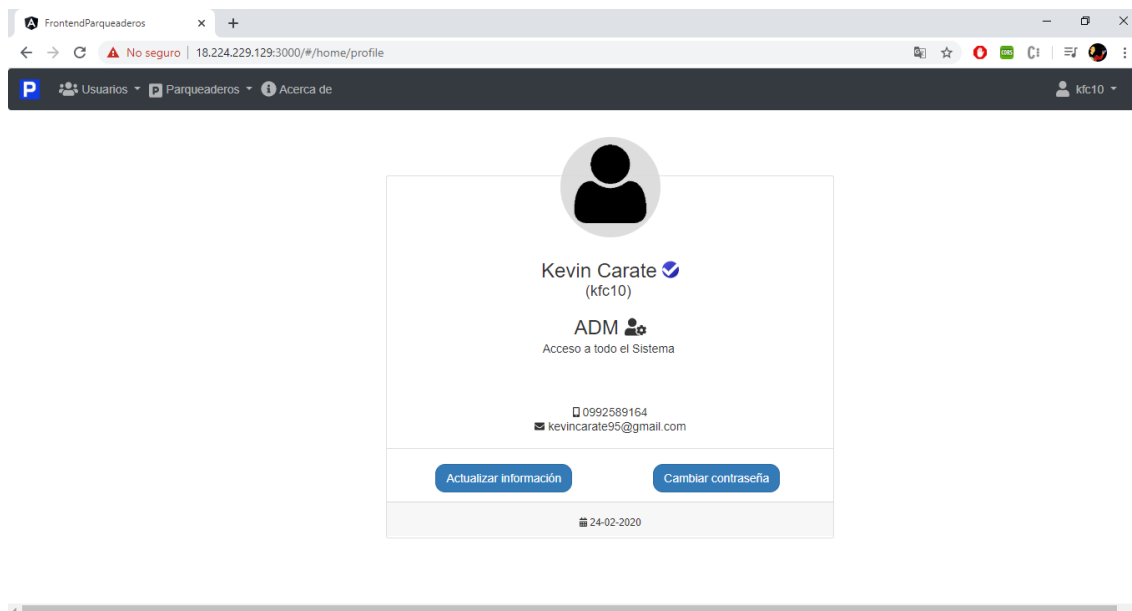


Figura 39. Interfaz Perfil de Usuario.

La Figura 40 presenta la Interfaz para actualizar los datos del usuario logueado.

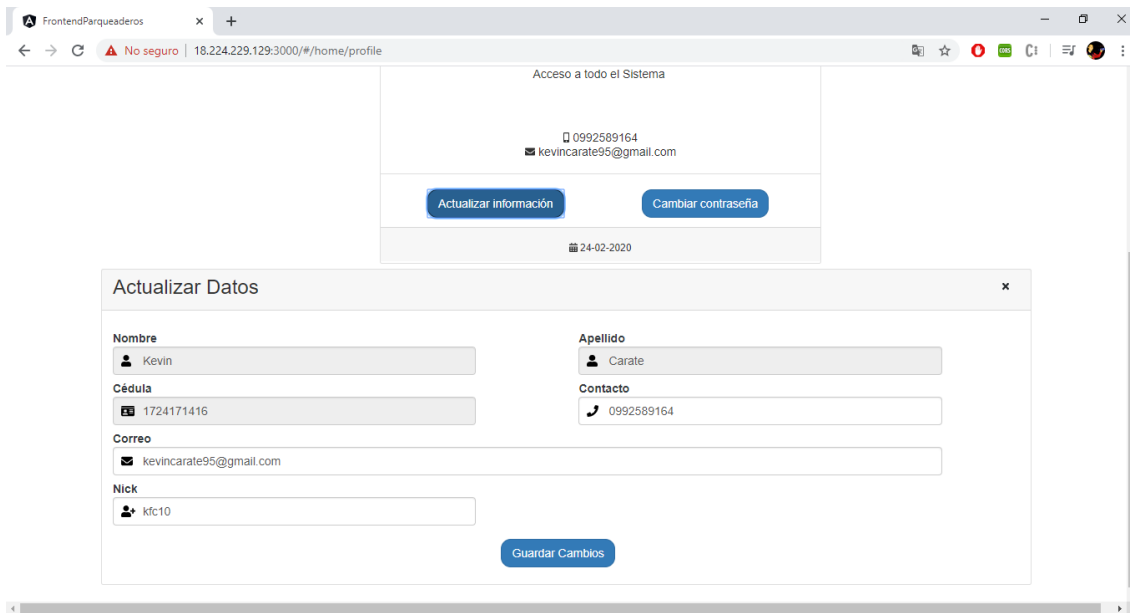


Figura 40. Interfaz Actualización de Datos de Usuario.

La Figura 41 presenta la Interfaz para cambiar la contraseña del usuario logueado.

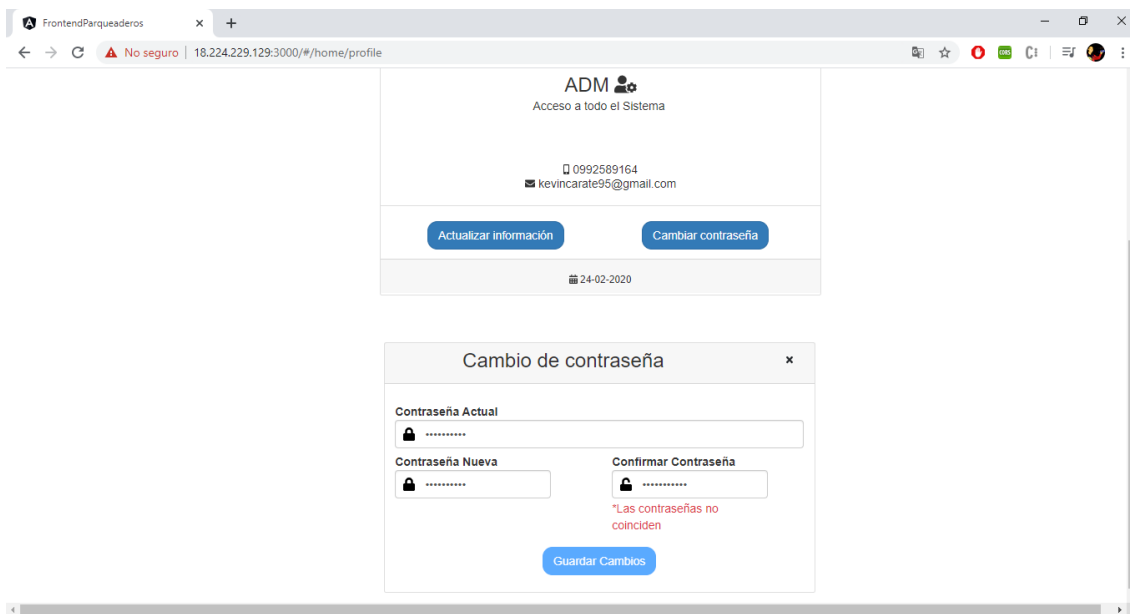


Figura 41. Interfaz Actualización de Contraseña de Usuario.

La figura 42 presenta la interfaz de Información de la aplicación web.



Figura 44. Conexión del Subsistema para Obtener Disponibilidad.

3.9.2 Sprint Review

Durante el desarrollo del Sprint 5 se logró cumplir con el objetivo planteado, el desarrollo de las interfaces en el presente sprint no tuvo mayor dificultad, debido a que en algunas se reutilizó componentes y servicios ya realizados anteriormente, así como métodos de validaciones de otras interfaces.

Una pequeña parte del tiempo empleado en el presente sprint sirvió para corregir ciertos bugs encontrados en las diferentes interfaces tanto del aplicativo móvil como web.

En la Tabla 42 se muestra si las actividades desarrolladas durante el sprint 5 fueron cumplidas.

Código Historia de Usuario	Enunciado de la Historia de Usuario	Criterios de Aceptación	Cumplido
----------------------------	-------------------------------------	-------------------------	----------

HU1.4	Configuración y Programación del Sensor	Actualizar automáticamente el estado de la plaza a la que es asignado el sensor.	SI
HU4.1	Diseño interfaz de información y ayuda - App Móvil	Visualizar información sobre el uso y manejo de la aplicación móvil	SI
HU7.2	Interfaz Cambio de contraseña - App Web	cambio de contraseña de una determinada cuenta de usuario	SI
		notificar del cambio de contraseña al usuario vía correo electrónico	SI
HU8.2	Interfaz de actualización de plazas y sensores - App Web	Se debe poder cambiar el tipo de plaza	SI
		Se debe poder cambiar el estado de una plaza	SI
HU9.1	Diseño interfaz de información y ayuda - App Web	Visualizar información sobre el contenido y uso de la aplicación web	SI

Tabla 42. Sprint Review 5.

Gráfico Burndown Sprint 5

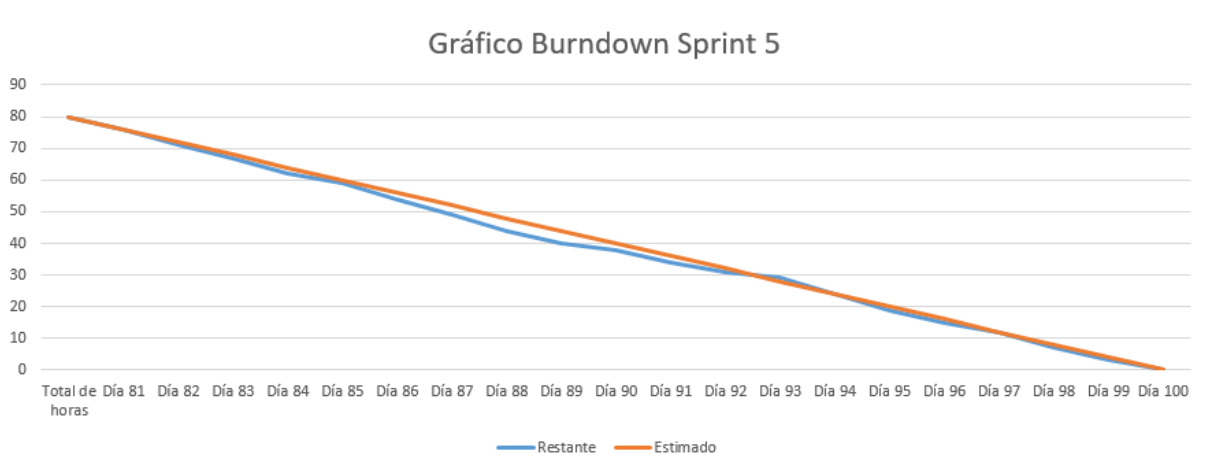


Figura 45. Gráfico Burndown Sprint 5.

Como se observa en la Figura 45 Gráfico Burndown Sprint 5, la mayoría del desarrollo del Sprint 5 se lo realizó de una manera acelerada ya que presenta desfases en el esfuerzo y tiempo empleado (línea azul) con respecto a lo estimado (línea naranja). El desarrollo del

Gráfico Burndown del Sprint 5 se puede observar con más detalle en el Anexo 4 - Gráficos Burndown.

3.9.3 Sprint Retrospective

El desarrollo del Sprint 5 se realizó sin mayores complicaciones, como se menciona en el apartado 3.9.2 Sprint Review durante su desarrollo se logró corregir bugs encontrados en interfaces desarrolladas anteriormente.

Finalmente, con respecto a su planificación, su gráfico Burndown refleja que se trabajó de una manera acelerada con respecto a anteriores Sprint.

4. PRUEBAS DE USABILIDAD Y RENDIMIENTO

En el presente capítulo se muestra las pruebas realizadas a las aplicaciones móvil y web, y los servicios para las aplicaciones. En primer lugar, se describe la Norma ISO/IEC 25010 que se va a tomar como referencia para realizar las Pruebas de Usabilidad de la aplicación móvil, el formato de las encuestas a realizarse y su respectivo análisis. Después, se detallan las Pruebas en Tiempo Real para visualizar el funcionamiento de la aplicación móvil. Luego, se definen las características según la Norma ISO/IEC 25010 para realizar las Pruebas de Rendimiento a los servicios para las aplicaciones móviles, a la aplicación web en diferentes navegadores y, por último, a la aplicación móvil en dispositivos de gama alta y baja. Finalmente, se realizó una discusión de los resultados obtenidos en las pruebas realizadas.

4.1 Pruebas de Usabilidad

Para la evaluación de usabilidad de la aplicación móvil se tomará como referencia la Norma ISO/IEC 25010, este modelo de calidad contiene ocho características para evaluar las propiedades de un producto software [44]. En este Proyecto de Titulación nos enfocaremos en la característica de usabilidad, que comprende la capacidad de un producto software para ser entendido, aprendido, usado y resultar ser atractivo para el usuario [45].

En este Proyecto de Titulación se realizará pruebas de usabilidad para la aplicación móvil, las mismas que serán evaluadas por medio de pruebas remotas, que nos permitirá obtener una idea de los aspectos relacionados al uso real de las aplicaciones.

4.1.1 Pruebas de Remotas para Aplicación Móvil

Para las pruebas de usabilidad de la aplicación móvil se ha definido realizar pruebas remotas a veinte personas, ya que, según Nielsen, para identificar los problemas de usabilidad más importantes de un diseño, generalmente es suficiente probar con cinco usuarios [46].

Las pruebas remotas consisten en que cada persona va a descargar e instalar la aplicación móvil, después utilizar la aplicación móvil navegando por todos los módulos y finalmente deberán llenar una encuesta definida en el punto 4.1.2, enfocada en las métricas de usabilidad de la ISO/IEC 25000, las encuestas se pueden observar en el Anexo 5 - Encuestas.

4.1.2 Estructura de la Encuesta para Aplicación Móvil

Para calificar las encuestas se utilizarán los siguientes parámetros: 1: Malo, 2: Regular, 3: Bueno, 4 Muy Bueno.

Se utilizó Google Forms para crear la encuesta y contar con un registro digitalizado de los resultados. A continuación, en la Tabla 43 se detalla la estructura de la encuesta:

Nº	Pregunta	Puntuación			
		1	2	3	4
1	¿Qué tan útil sería esta aplicación para conocer la disponibilidad de plazas del parqueadero del CEC?				
2	¿Qué tan buena fue la navegación en la aplicación?				
3	¿Qué tan eficiente es la información de un parqueadero?				
4	¿Qué tan eficiente es tener una ruta para dirigirse a un parqueadero?				
5	¿Qué tan adecuado es el diseño de las pantallas para este tipo de aplicación?				
6	¿Qué tan útil es la búsqueda de un parqueadero?				
7	¿Qué tan clara fue la información de ayuda de la aplicación?				

Tabla 43. Estructura de la Encuesta para Aplicación Móvil.

4.1.3 Análisis de Encuestas para Aplicación Móvil

La tabulación de resultados de las pruebas de usabilidad y desarrollo de gráficos se puede observar en el Anexo 6 – Tabulación de Resultados Encuestas de Aplicación Móvil ParkePN.

En la Figura 46 se observa los resultados a la pregunta 1 de la encuesta en donde el 80% de personas consideran que la aplicación es Muy Buena para conocer la disponibilidad de plazas del parqueadero del CEC y el 20% restante consideran que es Buena.

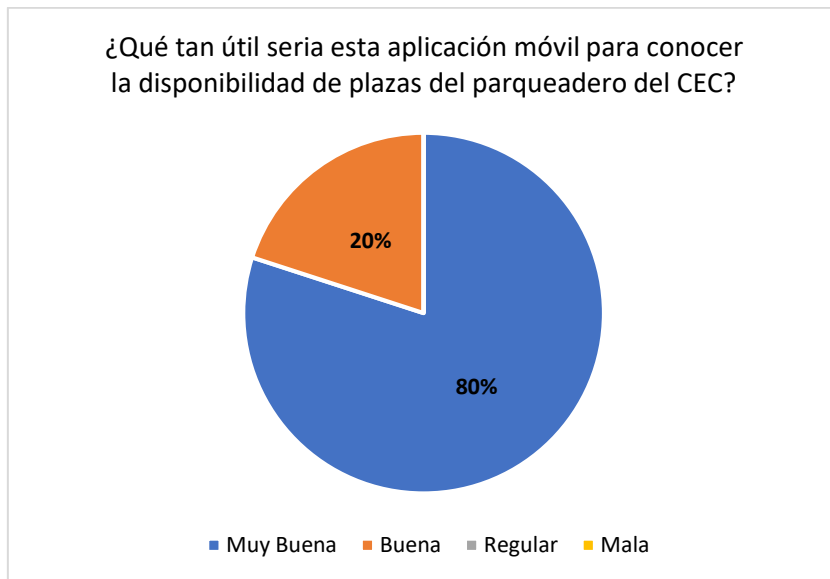


Figura 46. Pregunta 1 Encuesta de Usabilidad

En la Figura 47 se observa los resultados a la pregunta 2 de la encuesta en donde el 60% de personas consideran que la aplicación tiene Muy Buena navegación entre las pantallas y el 40% restante considera que es Buena.

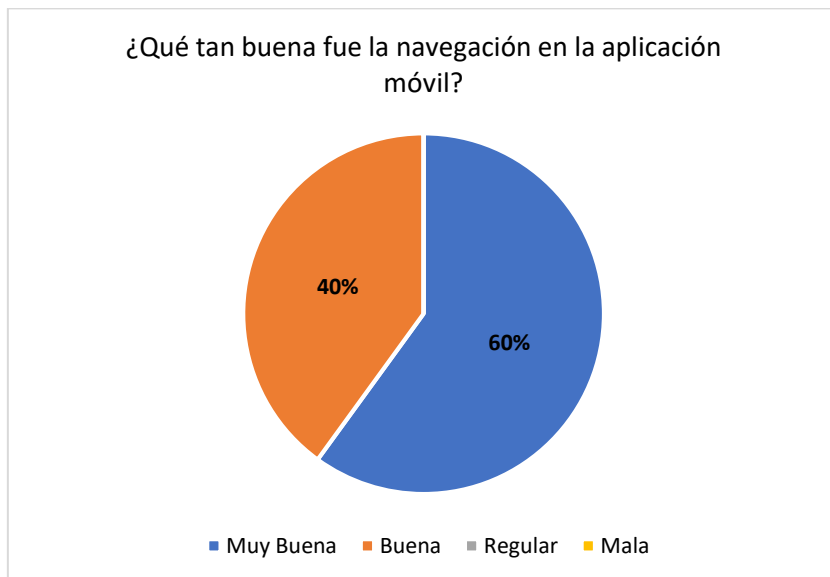


Figura 47. Pregunta 2 Encuesta de Usabilidad

En la Figura 48 se observa los resultados a la pregunta 3 de la encuesta en donde el 65% de personas consideran que la información acerca de los parqueaderos es Muy Buena y el 35% restante considera que es Buena.

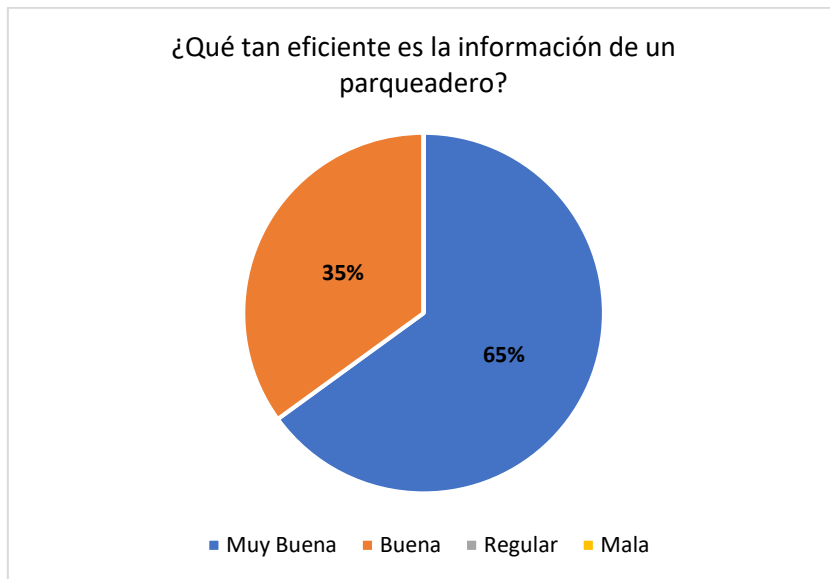


Figura 48 Pregunta 3 Encuesta de Usabilidad

En la Figura 49 se observa los resultados a la pregunta 4 de la encuesta en donde el 70% de personas consideran que es Muy Buena la posibilidad de tener una ruta para dirigirse a un parqueadero y el 30% restante considera que es Buena.

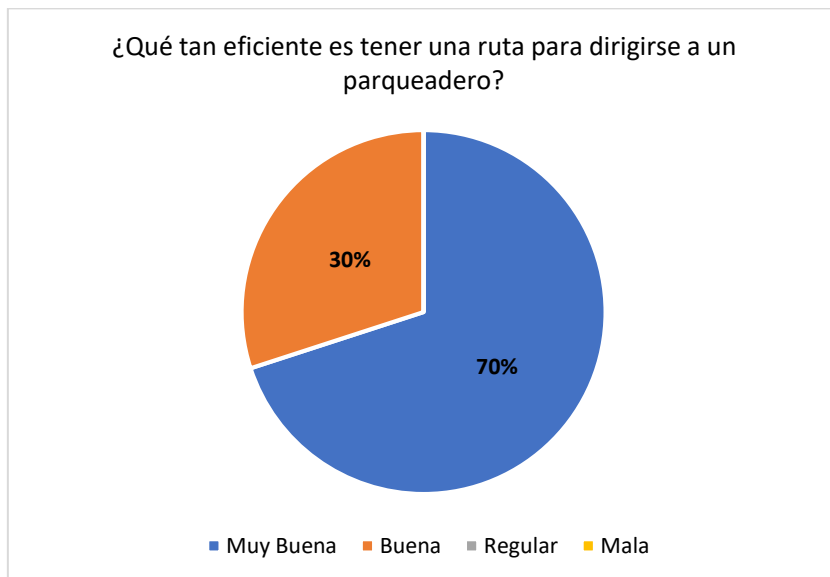


Figura 49 Pregunta 4 Encuesta de Usabilidad

En la Figura 50 se observa los resultados a la pregunta 5 de la encuesta en donde el 65% de personas consideran que el diseño de las pantallas es Muy Buena para una aplicación móvil y el 35% restante considera que es Buena.

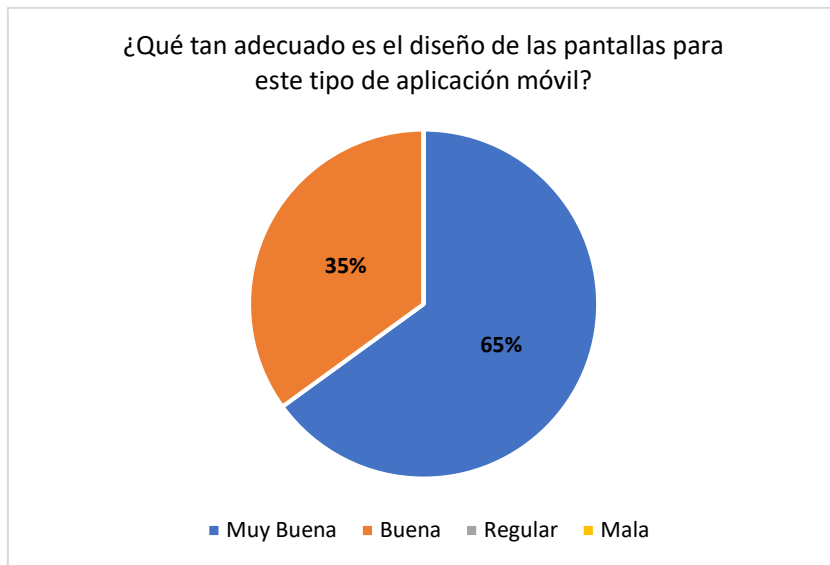


Figura 50 Pregunta 5 Encuesta de Usabilidad

En la Figura 51 se observa los resultados a la pregunta 6 de la encuesta en donde el 80% de personas consideran que es Muy Buena la funcionalidad de buscar un parqueadero en la aplicación móvil y el 20% restante considera que es Buena.

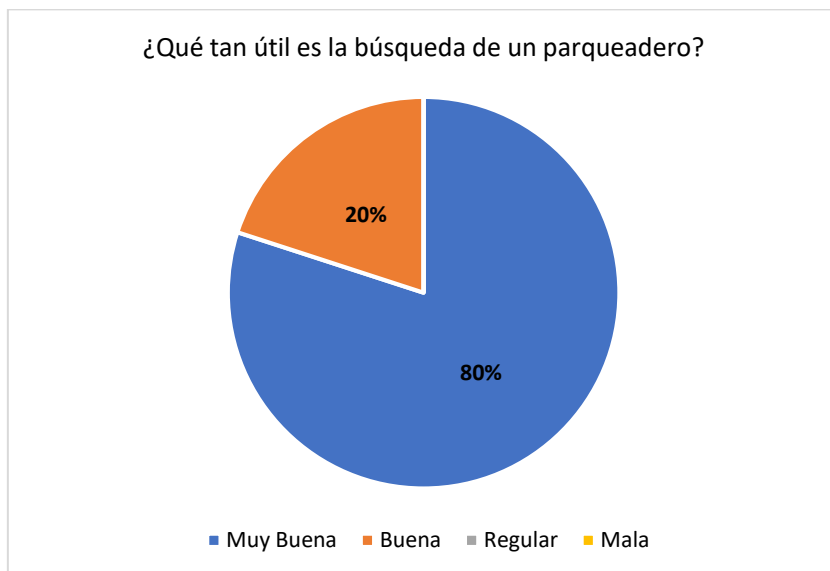


Figura 51 Pregunta 6 Encuesta de Usabilidad

En la Figura 52 se observa los resultados a la pregunta 7 de la encuesta en donde el 60% de personas consideran que es Muy Buena la información de ayuda de la aplicación móvil y el 40% restante considera que es Buena.

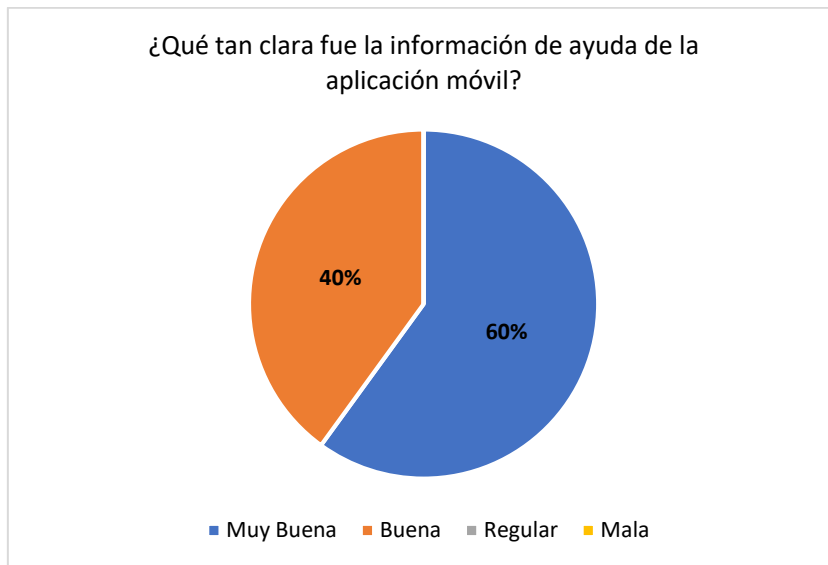


Figura 52 Pregunta 7 Encuesta de Usabilidad

4.1.4 Pruebas en Tiempo Real

Para estas pruebas se definió realizar la implementación del Subsistema para Obtener Disponibilidad que comprende la instalación en una plaza del parqueadero del CEC, pero, por motivos relacionados a la emergencia sanitaria por la que atraviesa el mundo y el país no se pudo llevar a cabo dicha implementación en el establecimiento del CEC. Sin embargo, se logró realizar estas pruebas en otras instalaciones, con el fin de cumplir con el objetivo de mostrar la disponibilidad de una plaza en tiempo real.

En la Figura 53 se puede observar el Subsistema para Obtener Disponibilidad enmarcado en un recuadro rojo, el mismo que se encuentra configurado para simular la plaza número 1 del parqueadero del CEC. La instalación del prototipo del dispositivo se lo realizó en el techo de una plaza a una altura aproximadamente de 3 metros de manera que el sensor pueda detectar un vehículo.

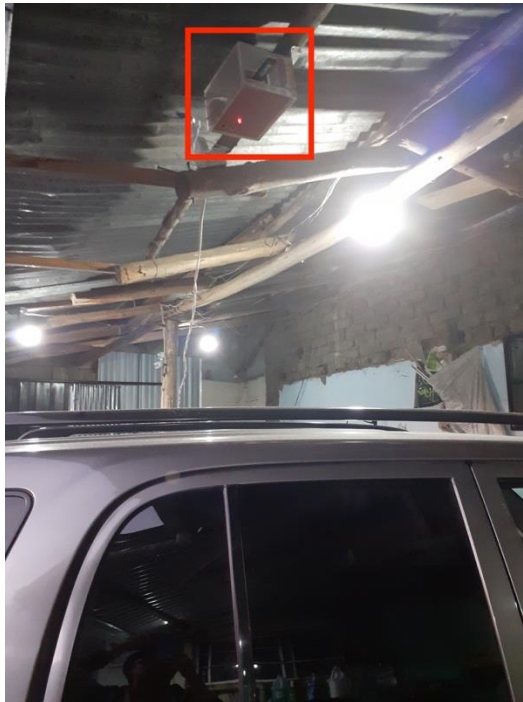


Figura 53. Instalación del Subsistema para Obtener Disponibilidad

En la Figura 54, se puede observar la aplicación móvil Park EPN en la Pantalla “Subsuelo 1” donde se visualiza la plaza número 1, encerrado en un recuadro azul, del parqueadero del CEC en un estado ocupado y detrás el vehículo parqueado.

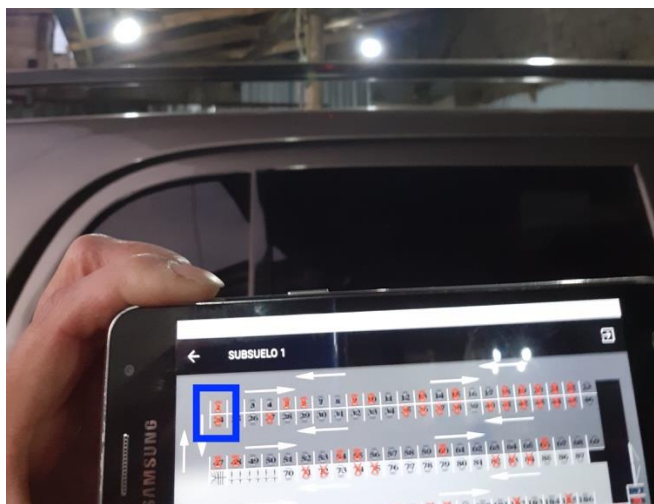


Figura 54. Funcionamiento de la Aplicación Park EPN en Tiempo Real

Se puede observar el Anexo 6 – Pruebas en Tiempo Real, para visualizar el proceso completo de las pruebas en tiempo real del sensor y la aplicación móvil Park EPN.

4.2 Pruebas de Rendimiento

Según el ISTQB (International Software Testing Qualifications Board) el rendimiento, de una aplicación, es un factor importante para proporcionar una buena experiencia a los usuarios que hacen uso de dicha aplicación [47].

En este apartado también tomaremos como referencia la norma ISO/IEC 25010, enfocándonos en la característica “Eficiencia de Desempeño” la misma que representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones y que comprende las siguientes subcaracterísticas:

- Comportamiento temporal [44].
- Utilización de recursos [44].
- Capacidad [44].

El ISTQB define a las pruebas de rendimiento como cualquier tipo de prueba centrada en la capacidad de respuesta de un sistema bajo diferentes volúmenes de carga [47].

Existen varios tipos de pruebas de rendimiento, y tomando en cuenta las subcaracterísticas de Rendimiento antes mencionadas, en el presente proyecto se realizarán pruebas de carga y estrés al Backend de nuestra aplicación y para medir el rendimiento de la aplicación móvil y web observaremos el consumo de recursos como CPU, Memoria y Red.

En la Tabla 44 se realizó una comparativa de las herramientas más utilizadas para realizar pruebas de carga y estrés.

Herramienta	Ventajas	Desventajas
Apache JMeter	<ul style="list-style-type: none">• Versión gratuita.• Interfaz Amigable.• Soporta Protocolos Básicos.• Documentación extensa.• Visualización de Resultados.• Independencia de Plataforma [48].	<ul style="list-style-type: none">• No se comporta como en un navegador.• No interpreta código JavaScript.

LoadUI	<ul style="list-style-type: none"> • Versión gratuita y de pago. • Interfaz Amigable. • Visualización de Resultados. • Pruebas de carga paralelas. • Automatización de pruebas [49]. 	<ul style="list-style-type: none"> • Funcionalidades avanzadas en versión pagada. • Necesita de SoapUI
Neoload	<ul style="list-style-type: none"> • Versión gratuita y de pago. • Compatible con últimas tecnologías. • Automatización de pruebas. • Interfaz Amigable [50]. 	<ul style="list-style-type: none"> • No disponible para Mac OS. • Funcionalidades avanzadas en versión pagada.
HP LoadRunner	<ul style="list-style-type: none"> • Versión gratuita y de pago. • Soporta varios protocolos. • Automatización de pruebas. • Interfaz Amigable [51]. 	<ul style="list-style-type: none"> • Solo funciona en Windows. • Funcionalidades avanzadas en versión pagada.

Tabla 44. Cuadro comparativo de herramientas para pruebas de carga y estrés.

Una vez analizadas las herramientas de la Tabla 44, para el presente proyecto se ha decidido utilizar JMeter para hacer pruebas de rendimiento al Backend de nuestra aplicación debido a que es totalmente gratuita y no presenta limitaciones en sus funcionalidades, también utilizaremos la herramienta Android Profiler para medir los recursos de la aplicación móvil, no hemos realizado ninguna tabla comparativa de Android Profiler debido a que esta herramienta es propia del IDE Android Studio en el cual se realizó todo el desarrollo de la aplicación móvil; Finalmente para medir los recursos de la aplicación web utilizaremos las herramientas de desarrollador que son propias de cada navegador.

4.2.1 Evaluación de rendimiento del Backend

Para más información se puede visualizar el Anexo 6 - Pruebas de Rendimiento Park EPN, en el cual se detalla la configuración, características, procedimiento, ejecución y resultados de las pruebas descritas en esta sección.

Para las pruebas de rendimiento del Backend se ha definido cuatro escenarios de pruebas que comprenden 68, 135, 270 y 370 usuarios concurrentes respectivamente.

En la Figura 55 se observa la definición de parámetros para realizar peticiones concurrentes al servicio con más consumo dentro de las aplicaciones tanto web como móvil. Se ha definido la dirección IP, el protocolo HTTP con el método GET, el puerto y la ruta a la cual hacer la petición.

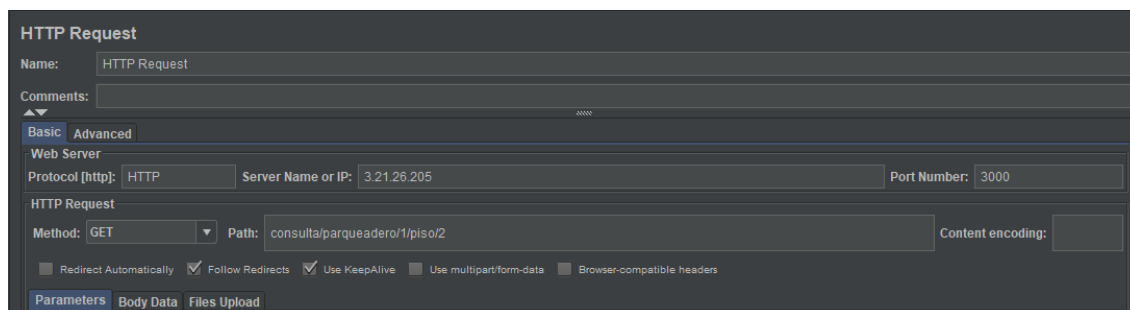


Figura 55. Definición de Parámetros de Consulta.

Escenario Uno

En la Figura 56 se observa la configuración del Escenario Uno, para la prueba se especificaron la cantidad de 68 usuarios concurrentes que realizarán la petición en 1 segundo y esta prueba se la realizará 50 veces.

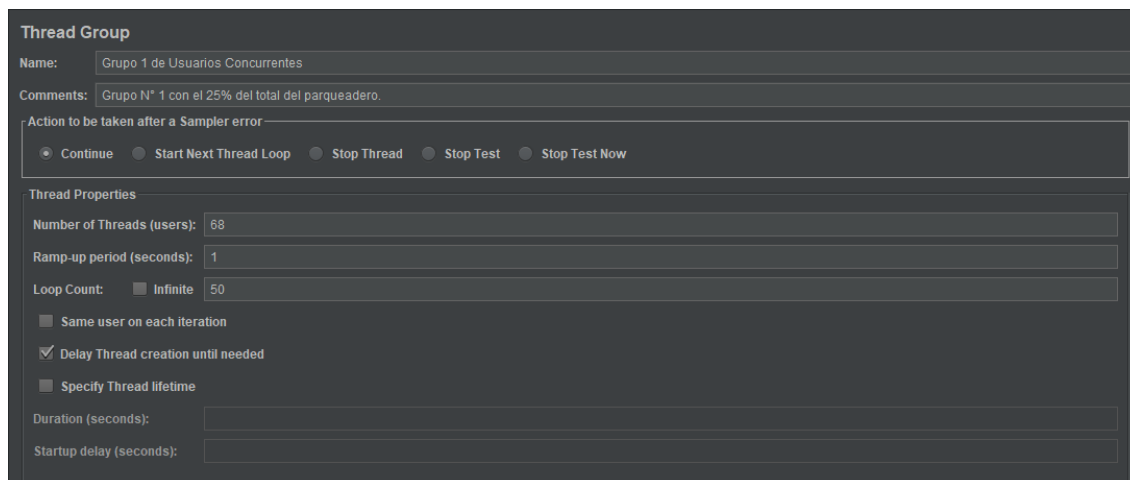


Figura 56. Configuración Escenario Uno.

Resultados Escenario Uno

A continuación, se detallan los resultados de la ejecución del Escenario Uno.

Etiqueta	Petición HTTP Resultado
# Muestras	3400
Tiempo Promedio (ms)	2095
Tiempo Mínimo (ms)	282
Tiempo Máximo (ms)	32235
Desviación Estándar	2585.65
Porcentaje de Error (%)	0.24
Rendimiento	25.6/seg
Cantidad de datos Recibidos (KB/seg)	411.09
Cantidad de datos Enviados (KB/seg)	3.74
Media en Bytes	16462.6
Cantidad de Peticiones Exitosas	3392
Cantidad de Peticiones Erróneas	8

Tabla 45. Resultados de la ejecución del Escenario 1.

Como se observa en la Tabla 45 se realizaron un total de 3400 peticiones al Backend de los cuales 8 de estas peticiones resultaron erróneas, también se observa que el tiempo mínimo de respuesta es de 282 milisegundos, es decir 0.28 segundos, estos tiempos de respuesta principalmente se observaron en un porcentaje alto de la prueba, sin embargo, existieron peticiones que tardaron un tiempo máximo de 32.23 segundos, debido a la saturación en el servidor y por tal motivo esto conlleva a que el tiempo de respuesta promedio sea de 2.09 segundos. La desviación estándar de 2585.65 está por encima de la media debido al porcentaje pequeño de peticiones con tiempo de respuesta grandes. Finalmente, el rendimiento en este escenario implica que se pueden recibir 25.6 peticiones por segundo conforme a las características del servidor.

Escenario Dos

En la Figura 57 se observa la configuración del Escenario Dos, para la prueba se especificaron la cantidad de 135 usuarios concurrentes que realizarán la petición en 1 segundo y esta prueba se la realizará 25 veces.

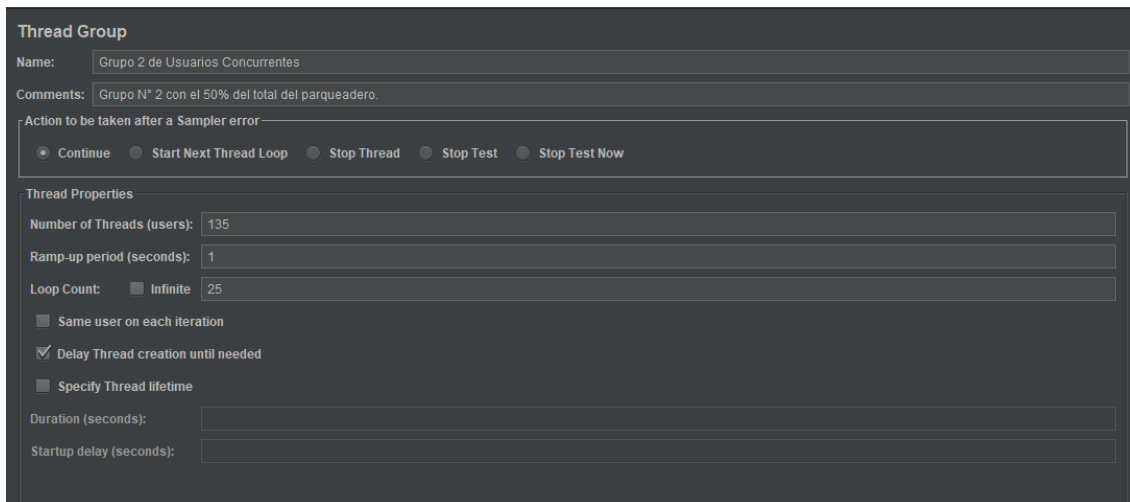


Figura 57. Configuración Escenario Dos.

Resultados Escenario 2

Etiqueta	Petición HTTP Resultado
# Muestras	3375
Tiempo Promedio (ms)	4437
Tiempo Mínimo (ms)	282
Tiempo Máximo (ms)	283715
Desviación Estándar	9769.79
Porcentaje de Error (%)	1.54
Rendimiento	10.9/seg
Cantidad de datos Recibidos (KB/seg)	172.94
Cantidad de datos Enviados (KB/seg)	1.57
Media en Bytes	16280.2
Cantidad de Peticiones Exitosas	3323
Cantidad de Peticiones Erróneas	52

Tabla 46. Resultados de la ejecución del Escenario Dos.

Como se observa en la Tabla 46 de un total de 3375 peticiones, 52 de estas resultaron erróneas, el tiempo mínimo de respuesta del servidor fue de 0.28 segundos, tuvo un tiempo máximo de respuesta de 283715 milisegundos, una sola petición alcanzo este valor y fue errónea, esto debido a que existió saturación en el servidor y por tal motivo conlleva a tener 4.43 segundos como tiempo de respuesta promedio. La desviación estándar varia

aproximadamente el doble de la media debido a la petición que se demoró alrededor de 4.72 minutos. Finalmente, en esta prueba se puede observar que el rendimiento en este escenario implica que se pueden receptor 10.9 peticiones por segundo debido a las características del servidor.

Escenario Tres

En la Figura 58 se observa la configuración del Escenario Tres, para la prueba se especificaron la cantidad de 270 usuarios concurrentes que realizarán la petición en 1 segundo y esta prueba se la realizará 15 veces.

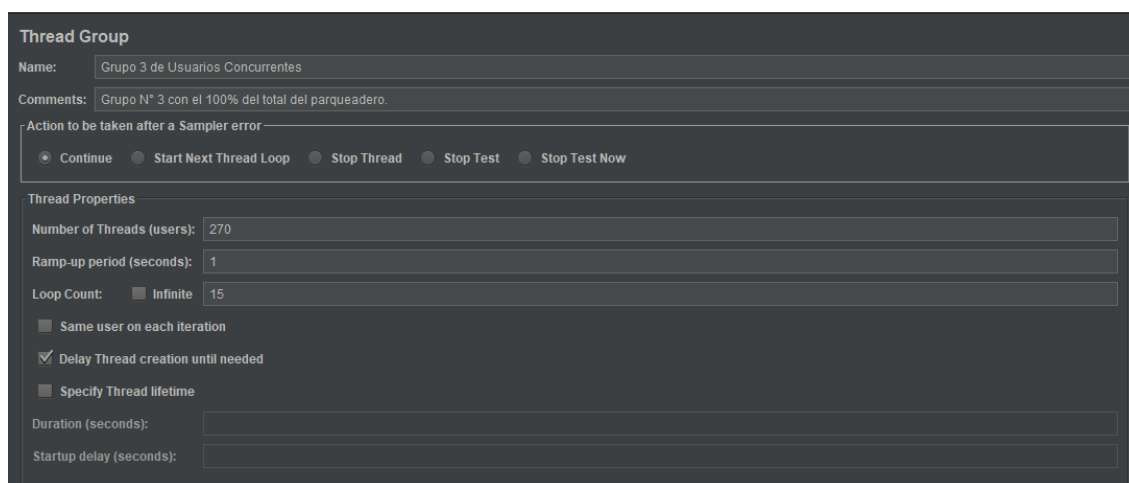


Figura 58. Configuración Escenario Tres.

Resultados Escenario Tres

Etiqueta	Petición HTTP Resultado
# Muestras	4050
Tiempo Promedio (ms)	7964
Tiempo Mínimo (ms)	282
Tiempo Máximo (ms)	293004
Desviación Estándar	22388.52
Porcentaje de Error (%)	3.95
Rendimiento	13.2/seg
Cantidad de datos Recibidos (KB/seg)	205.62
Cantidad de datos Enviados (KB/seg)	1.86

Media en Bytes	15939.9
Cantidad de Peticiones Exitosas	3890
Cantidad de Peticiones Erróneas	160

Tabla 47. Resultados de la ejecución del Escenario Tres.

Como se observa en la Tabla 47 de un total de 4050 peticiones, 160 resultaron erróneas, el tiempo mínimo de una petición fue de 0.28 segundos, el tiempo máximo de una petición fue de 293 segundos, por tal motivo se encuentra una media de 7964 milisegundos lo que implica un promedio de 7.96 segundos al realizar una petición. La desviación estándar casi triplica la media debido a los 4.88 minutos de demora a una petición. Finalmente, en esta prueba se puede observar que el rendimiento en este escenario implica que se pueden receptor 13.2 peticiones por segundo debido a las características del servidor.

Escenario Cuatro

En la Figura 59 se observa la configuración del Escenario Cuatro, para la prueba se especificaron la cantidad de 370 usuarios concurrentes que realizarán la petición en 1 segundo y esta prueba se la realizará 5 veces.

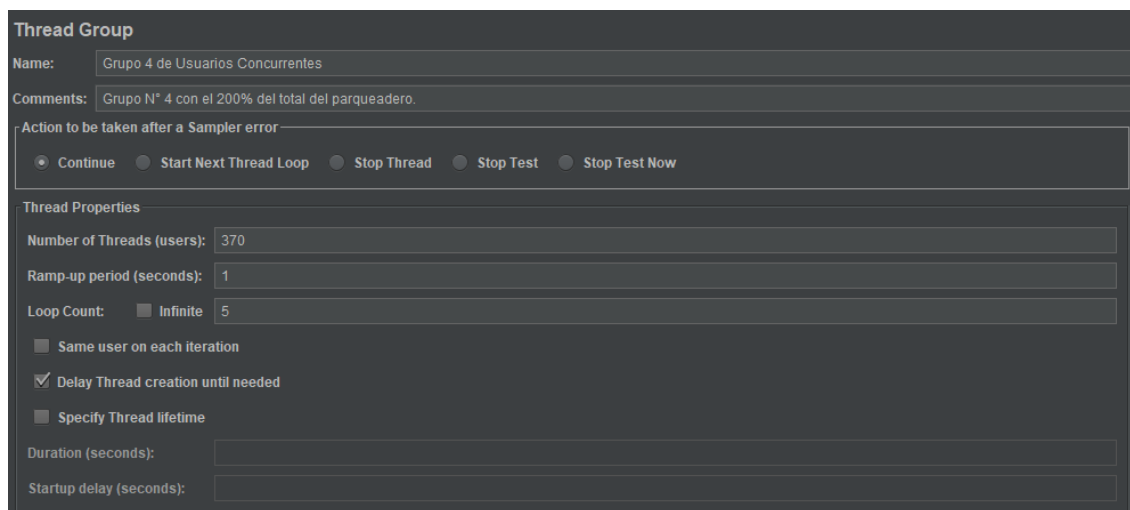


Figura 59. Configuración Escenario Cuatro.

Resultados Escenario Cuatro

Etiqueta	Petición HTTP Resultado
# Muestras	1850

Tiempo Promedio (ms)	10577
Tiempo Mínimo (ms)	287
Tiempo Máximo (ms)	252060
Desviación Estándar	25561.01
Porcentaje de Error (%)	5.03
Rendimiento	6.8/seg
Cantidad de datos Recibidos (KB/seg)	104.30
Cantidad de datos Enviados (KB/seg)	0.94
Media en Bytes	15781.7
Cantidad de Peticiones Exitosas	1757
Cantidad de Peticiones Erróneas	93

Tabla 48. Resultados de la ejecución del Escenario Cuatro.

Como se observa en la Tabla 48 de un total de 1850 peticiones, 93 resultaron erróneas, el tiempo mínimo de respuesta fue de 0.28 segundos, el tiempo máximo fue de 252.06 segundos o lo que representa 4.20 minutos, en base a estos resultados se obtuvo un tiempo de respuesta promedio de 10.57 segundos. La desviación estándar duplica la media debido a los 4.20 minutos de repuesta a una petición. Finalmente, en esta prueba se puede observar que el rendimiento en este escenario implica que se pueden receptor 6.8 peticiones por segundo debido a las características del servidor.

4.2.2 Evaluación de rendimiento de Aplicación Web

Para realizar las pruebas de rendimiento de la aplicación web se ha definido usar tres navegadores, los cuales son: Google Chrome, Mozilla Firefox y Opera, estos navegadores accederán a la aplicación web en modo incognito para evitar que otras páginas y extensiones afecten al rendimiento.

Finalmente, se utilizarán las herramientas online PageSpeed Insights de Google y GTmetrix para evaluar la velocidad de carga de nuestra aplicación web.

Ejecución Navegador Chrome

En la Figura 60 observamos la ejecución y grabación de rendimiento en el navegador Chrome.

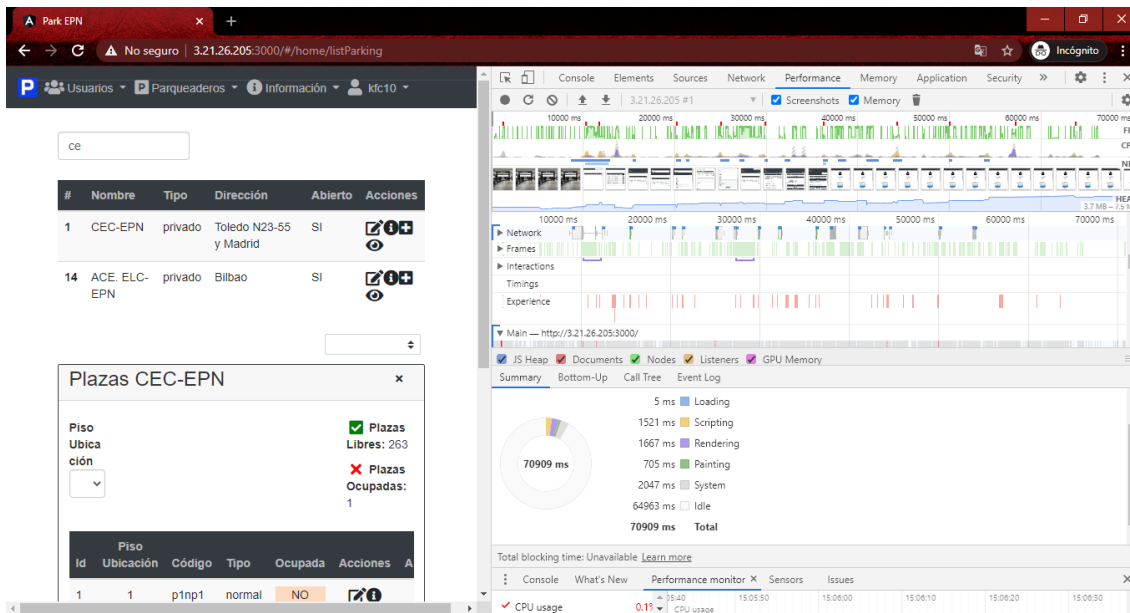


Figura 60. Ejecución y Grabación de Rendimiento Chrome.

Ejecución Navegador Firefox

En la Figura 61 observamos la ejecución y grabación de rendimiento en el navegador Firefox.

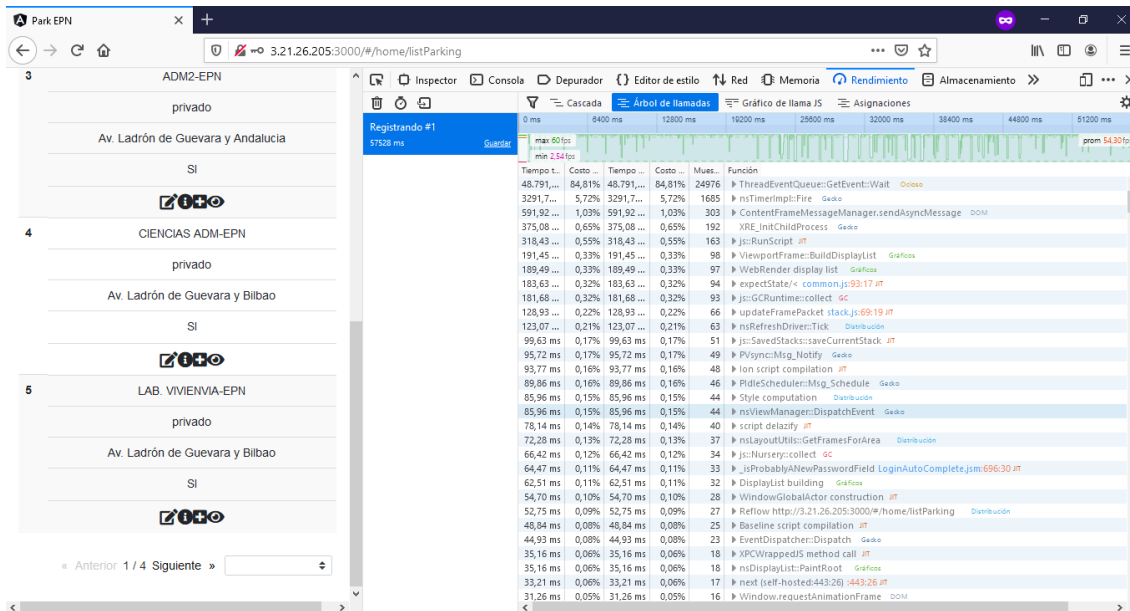


Figura 61. Ejecución y Grabación de Rendimiento Firefox.

Ejecución Navegador Opera

En la Figura 62 observamos la ejecución y grabación de rendimiento en el navegador Opera.

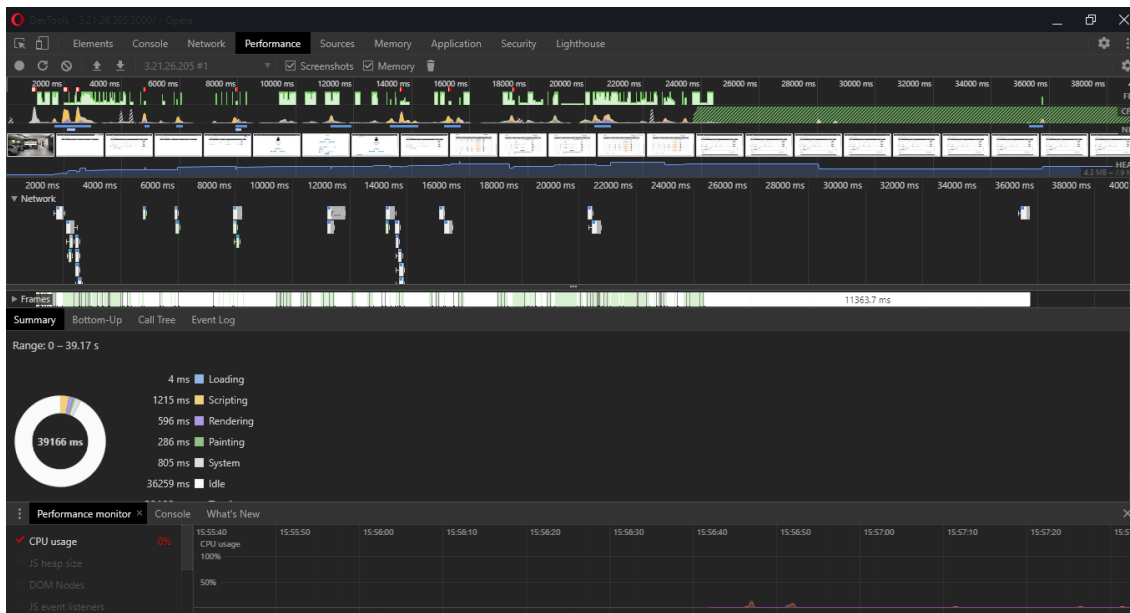


Figura 62. Ejecución y Grabación de Rendimiento Opera.

Resultados

En la Tabla 49 podemos observar el resumen de los resultados obtenidos al ejecutar la aplicación web en los diferentes navegadores definidos anteriormente.

	CPU	Memoria Obtenida	Uso Red	
			Cantidad datos Recibidos (máx)	Tiempo de Respuesta (máx)
Google Chrome	2.6 %	53.15 MB	249 KB	1.41 s
Mozilla Firefox	-	18.5 MB	249 KB	0.28 s
Opera	1.2 %	32.77 MB	249 KB	0.51 s

Tabla 49. Resultados Rendimiento Aplicación Web

Como observamos en la Tabla 49 el consumo de CPU promedio es de 2.6% en el caso del navegador Chrome, 1.2% en el navegador Opera y en Firefox no se pudo obtener el consumo de CPU debido a que en las herramientas de desarrollador no se encontró dicha medición, cabe recalcar que la máxima cantidad de consumo de CPU encontrada en todos los navegadores fue cuando se consulta el estado de las plazas de un determinado parqueadero.

Con respecto a la memoria consumida en el navegador Chrome se registró un resultado de 53.15 MB, 32.77 MB en Opera y en Firefox se obtuvo el valor de 18.5 MB de uso de memoria como máximo.

Con respecto al uso de red se pudo observar que el tiempo de respuesta de la petición más consumida fue de 1.41 segundos en Chrome, en el caso de Firefox se obtuvo un tiempo de respuesta de 0.51 segundos y en Opera el tiempo de respuesta fue de 0.28 segundos, podemos observar que estos valores se encuentran dentro del rango registrados en la evaluación de rendimiento del escenario 1.

Finalmente, la cantidad máxima de datos recibidos fue de 249 KB en todos los navegadores, esto debido a que todos reciben la misma información.

Herramienta GTmetrix

La herramienta Gtmetrix analiza las siguientes características en la página web: el rendimiento de una página web basándose en indicadores clave de velocidad de carga de la página y en relación con el promedio de todos los sitios analizados en Gtmetrix, la carga de la página desde 28 servidores ubicados en diferentes regiones del mundo y utiliza un dispositivo Android real para obtener datos de rendimiento para usuarios móviles [52].

La utilización de la herramienta GTmetrix permitió conocer deficiencias en la aplicación web como se puede observar en el Anexo 6 – Pruebas de Rendimiento. Con el conocimiento de las deficiencias se procedió a realizar los cambios necesarios, obteniendo una calificación de A(100%) en el Test de PageSpeed y una calificación de A(94%) en el Test de Yslow Score como se puede observar en la Figura 63.

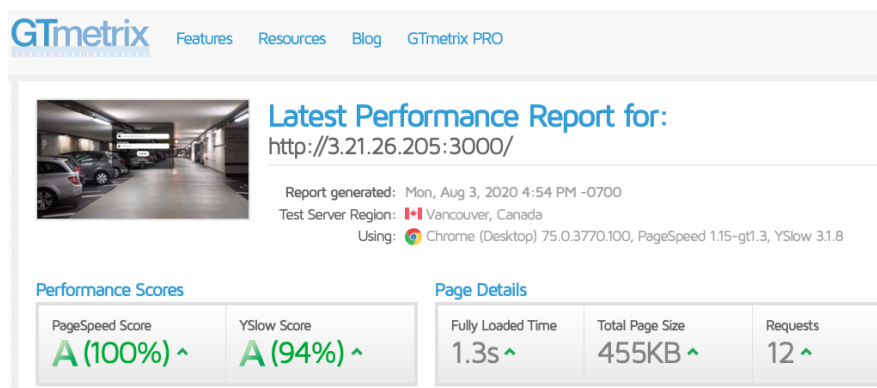


Figura 63 Análisis Rendimiento Herramienta GTmetrix

API PageSpeed

La API PageSpeed utiliza la herramienta Lighthouse que analiza el rendimiento de las páginas en dispositivos móviles como en ordenadores y ofrece sugerencias para mejorarlas [53].

Los resultados de la herramienta Lighthouse pueden fluctuar debido a: [54]

- Pruebas A/B o cambios en los anuncios que se publican.
- Cambios en el enrutamiento del tráfico de Internet.
- Pruebas en diferentes dispositivos, como una computadora de escritorio de alto rendimiento y una portátil de bajo rendimiento.
- Extensiones de navegador que inyectan JavaScript.
- Software de Antivirus.

La utilización de la herramienta Lighthouse permitió conocer deficiencias en la aplicación web como se puede observar en el Anexo 6 – Pruebas de Rendimiento. Con el conocimiento de las deficiencias se procedió a realizar los cambios necesarios, obteniendo una calificación de 80 en un dispositivo móvil como se puede observar en la Figura 64 y una calificación de 99 en un dispositivo de escritorio como se puede observar en la Figura 65.

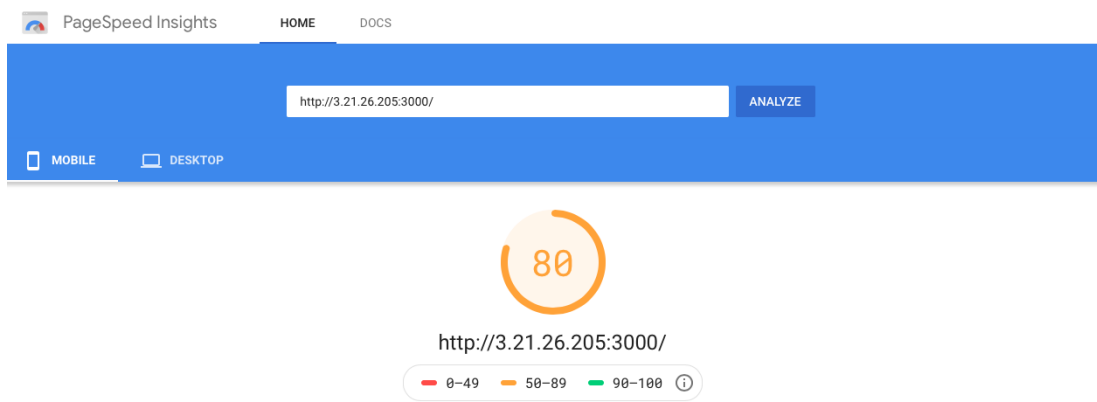


Figura 64 Análisis Dispositivo Móvil Herramienta PageSpeed

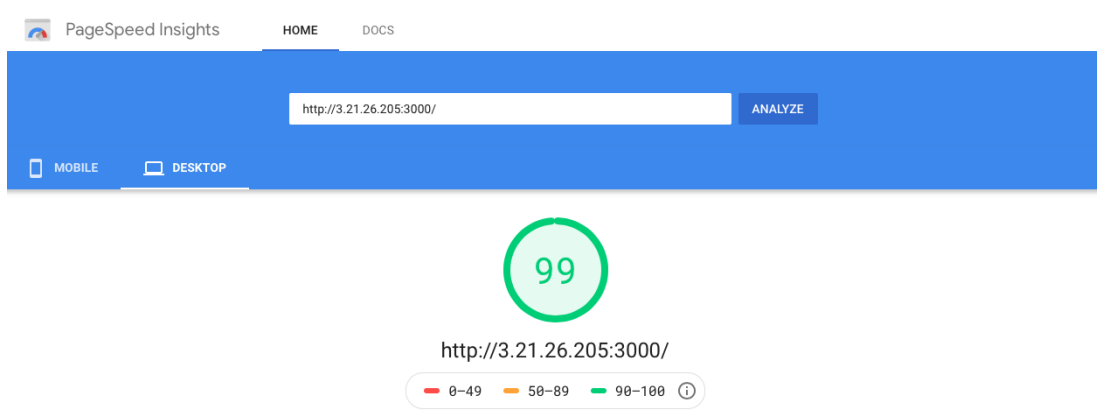


Figura 65 Análisis Dispositivo de Escritorio Herramienta PageSpeed

4.2.3 Evaluación de rendimiento de Aplicación Móvil

Para realizar las pruebas de rendimiento de la aplicación móvil se ha definido usar dos celulares uno de gama alta y otro de gama baja, con el propósito de conocer los recursos necesarios para el correcto funcionamiento de la aplicación móvil

Características del Celular Gama Alta

- Samsung Galaxy A10
- Modelo SM-105M
- Versión de Android 9
- Procesador Exynos 7884 1.6GHz
- Memoria RAM 2GB
- Almacenamiento 32GB

Uso de CPU

Para tener una correcta medición del uso de CPU de la aplicación, se navegó por todas las pantallas y se realizó todas las acciones que permite la aplicación, lo que permitió monitorear el uso de CPU como se observa en la Figura 66.

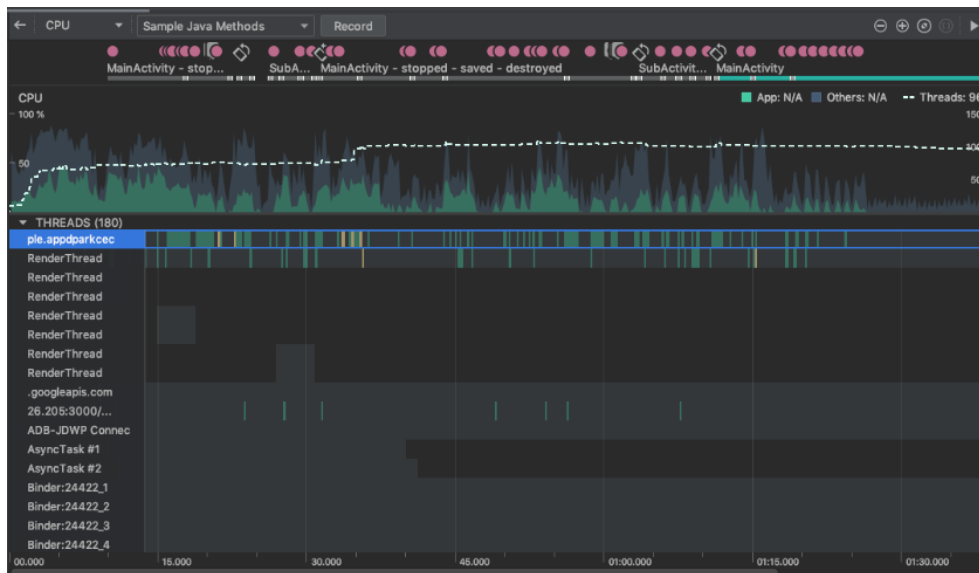


Figura 66. Monitoreo de Uso de CPU para Celular Gama Alta.

En la Figura 66, se pueden observar varios picos altos de uso de CPU, sin embargo, el pico más alto se registró en la pantalla principal donde se encuentra el mapa de Google Maps con los marcadores de todos los parqueaderos. La aplicación hace uso del 47% de CPU en su pico más alto como se observa en la Figura 67.

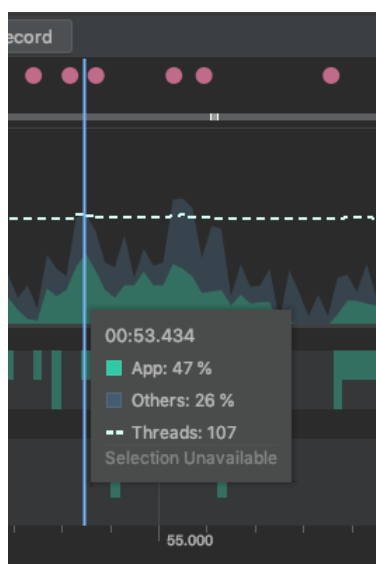


Figura 67. Pico más alto de Uso de CPU para Celular Gama Alta.

Uso de Memoria

El uso de memoria de la aplicación aumenta conforme se navega entre las pantallas, por tal motivo, se navegó por la aplicación más de una ocasión permitiendo monitorear el uso de memoria como se observa en la Figura 68.

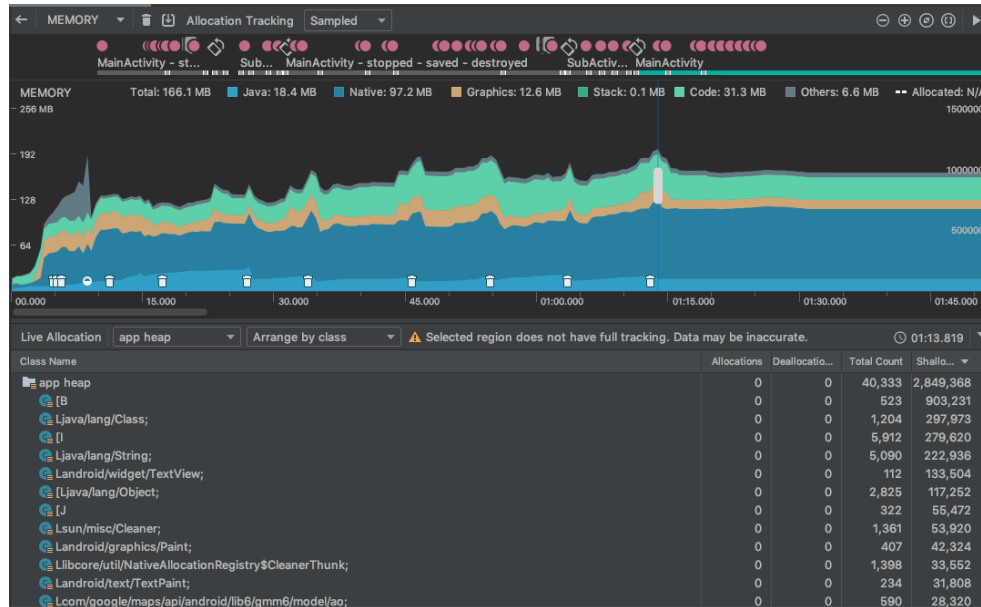


Figura 68. Monitoreo de Uso de Memoria para Celular Gama Alta.

En la Figura 69, se puede observar un constante aumento de la memoria mientras se utiliza la aplicación, pero, el pico más alto se registró en la pantalla principal donde se encuentra el mapa de Google Maps con los marcadores de todos los parqueaderos. El mayor consumo ocurre por la memoria nativa con 110.3 MB como se observa en la Figura 66, la memoria nativa hace referencia al manejo de imágenes y otros gráficos [55].

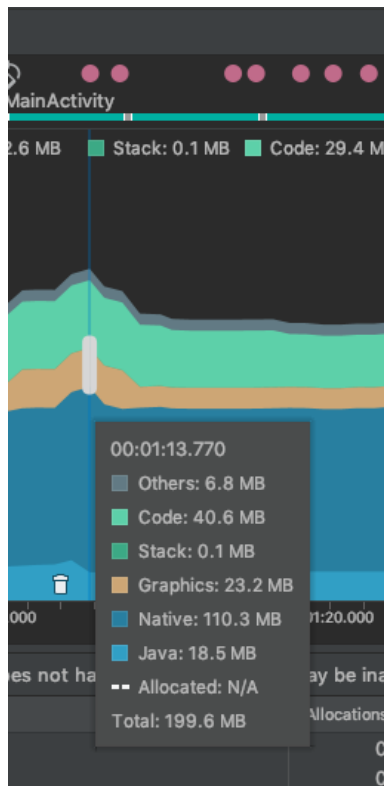


Figura 69. Pico más alto de Uso de Memoria para Celular Gama Alta.

Uso de Red

En la aplicación móvil se envían y reciben datos en todas las pantallas, por lo tanto, para conocer la cantidad de información que se transfiere y el tiempo que se demora se realizó el monitoreo del uso de red como se observa en la Figura 70.

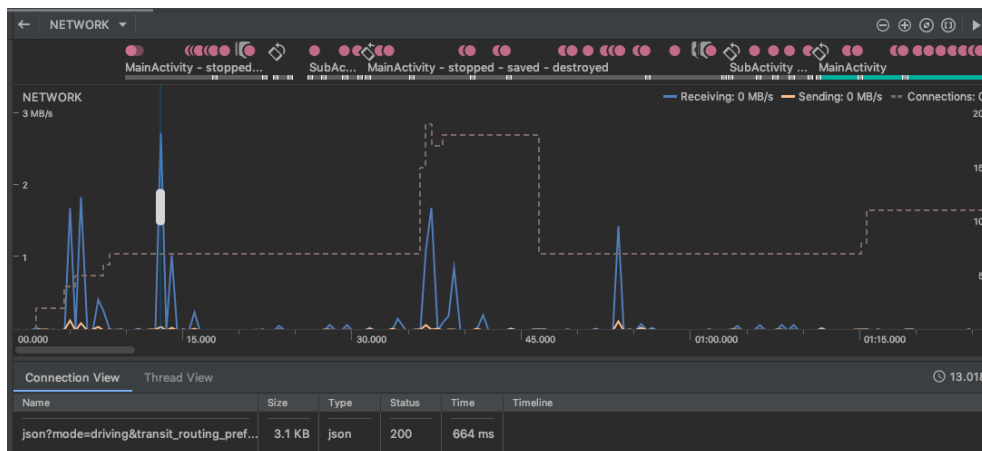


Figura 70. Monitoreo de Uso de Red para Celular Gama Alta.

En la Figura 70, se puede observar varios picos que hacen referencia a la información recibida por la Api de Google Maps y los Servicios del Backend desarrollados en el presente proyecto. El pico más alto se registró en la pantalla principal donde se encuentra el mapa de Google Maps con los marcadores de todos los parqueaderos, en esta pantalla se recibieron 2.7 MB/s y la petición con más tamaño es de 245 KB en un tiempo de 479 ms como se observa en la Figura 71.

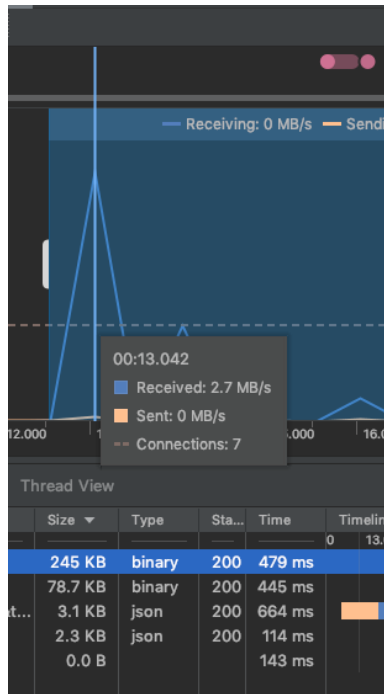


Figura 71. Pico más alto de Uso de Red para Celular Gama Alta.

Características del Celular Gama Baja

- Samsung Galaxy J2 Prime
- Modelo SM-G532M
- Versión de Android 6.0.1
- Procesador Mediatek MT6737T 1.4GHz
- Memoria RAM 1.5GB
- Almacenamiento 8GB

Uso de CPU

Para tener una correcta medición del uso de CPU de la aplicación, se realizó el mismo proceso que en el celular de gama alta, lo que permitió monitorear el uso de CPU como se observa en la Figura 72.

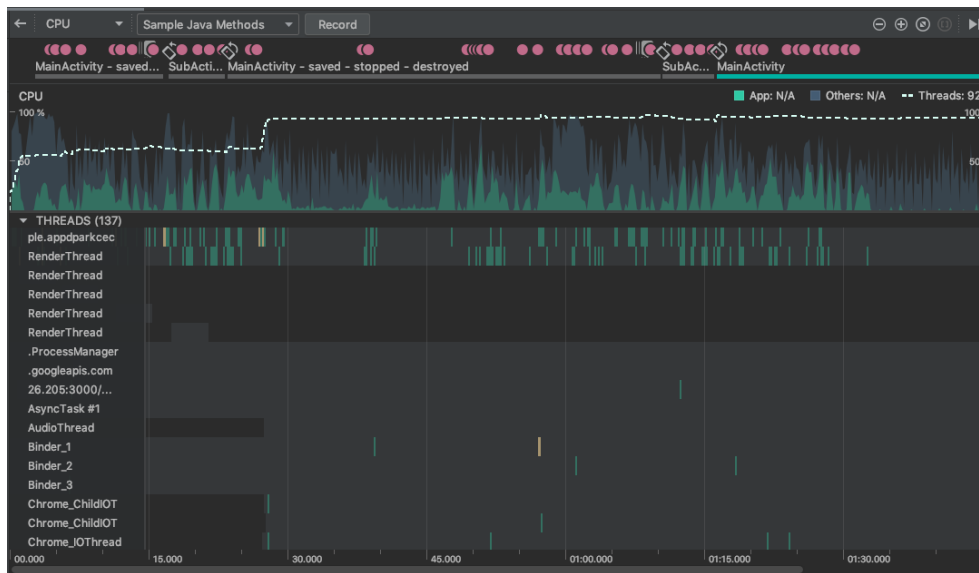


Figura 72. Monitoreo de Uso de CPU para Celular Gama Baja.

En la Figura 72, se pueden observar varios picos altos de uso de CPU, sin embargo, el pico más alto se registró en la pantalla principal donde se encuentra el mapa de Google Maps con los marcadores de todos los parqueaderos. La aplicación hace uso del 68% de CPU en su pico más alto como se observa en la Figura 73.

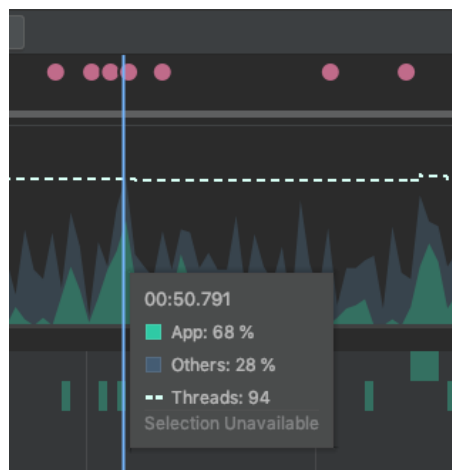


Figura 73. Pico más alto de Uso de CPU para Celular Gama Baja.

Uso de Memoria

Para obtener el uso de memoria de la aplicación se siguió el mismo proceso que el celular de gama alta, por lo que, se navegó por la aplicación más de una ocasión permitiendo monitorear el uso de memoria como se observa en la Figura 74.

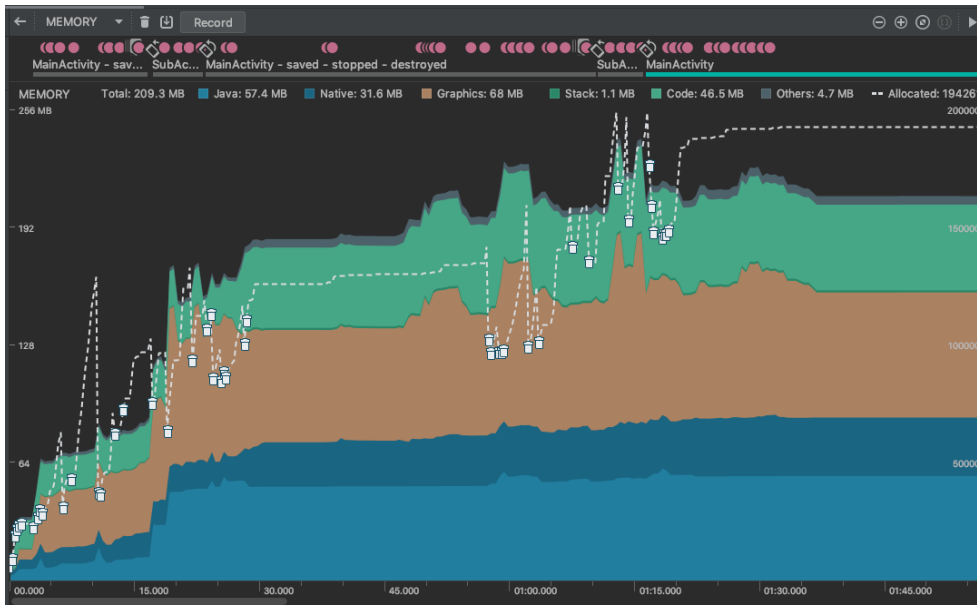


Figura 74. Monitoreo de Uso de Memoria para Celular Gama Baja.

En la Figura 74, se puede observar un constante aumento de la memoria mientras se utiliza la aplicación, pero, el pico más alto se registró en la para visualizar las plazas del parqueadero del CEC. El mayor consumo ocurre por la memoria gráfica con 102.9 MB como se observa en la Figura 75, la memoria gráfica tiene el trabajo de mostrar píxeles en pantalla [55].

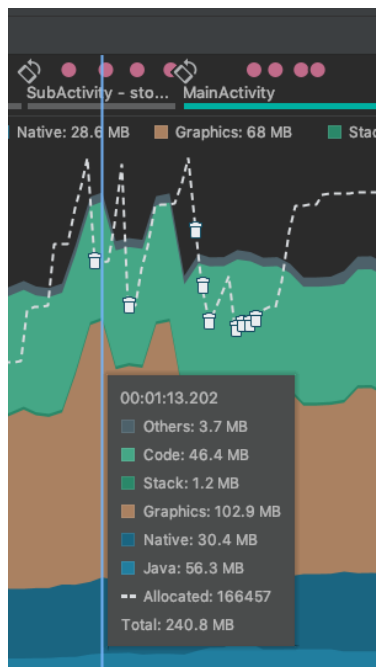


Figura 75. Pico más alto de Uso de Memoria para Celular Gama Baja.

Uso de Red

Para conocer el uso de red se realizó el mismo proceso que el celular de gama alta, por lo tanto, para conocer la cantidad de información que se transfiere y el tiempo que se demora se realizó el monitoreo del uso de red como se observa en la Figura 76.

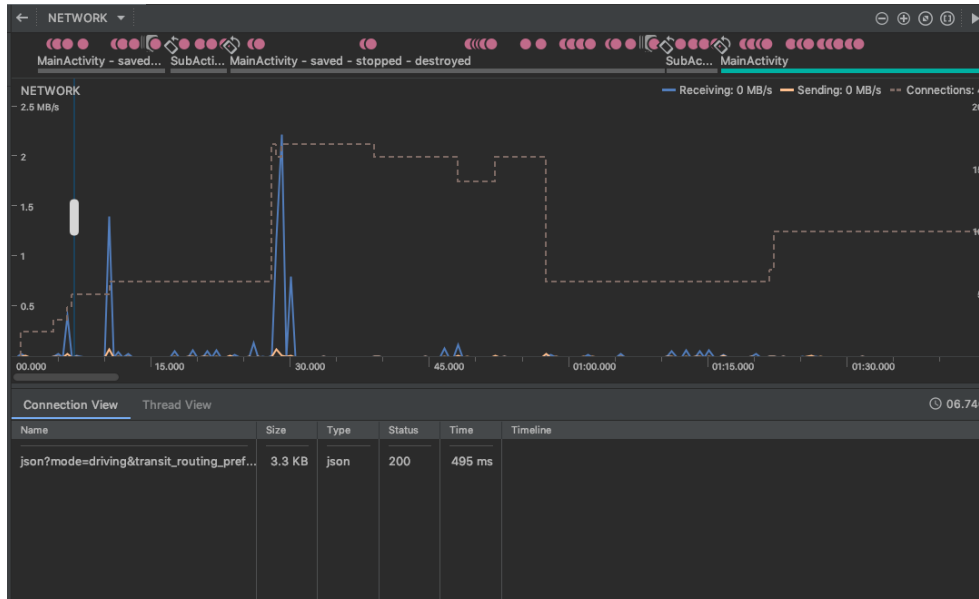


Figura 76. Monitoreo de Uso de Red para Celular Gama Baja.

En la Figura 76, se puede observar varios picos que hacen referencia a la información recibida por la Api de Google Maps y los Servicios del Backend desarrollados en el presente proyecto. El pico más alto se registró en la pantalla principal donde se encuentra el mapa de Google Maps con los marcadores de todos los parqueaderos, en esta pantalla se recibieron 2.2 MB/s como se observa en la Figura 77.

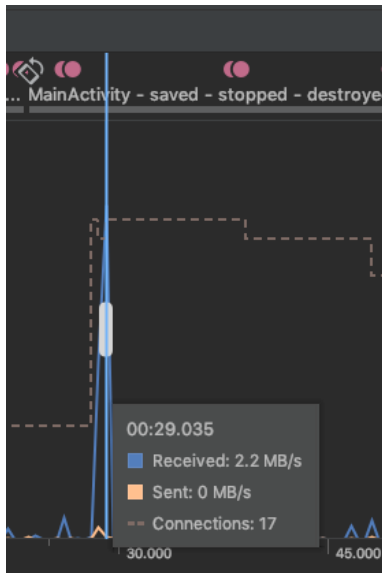


Figura 77. Pico más alto de Uso de Red para Celular Gama Baja.

4.3 Discusión de Resultados

Pruebas de Usabilidad

Los resultados obtenidos en la encuesta mostraron que la aplicación móvil denominada Park EPN tuvo una gran aceptación conforme a su diseño, funcionalidad y uso. Además, los usuarios supieron expresar que las pantallas y la información presentada son de gran ayuda para conocer la disponibilidad de un parqueadero.

También, los usuarios comentaron que la posibilidad de conocer la disponibilidad de plazas del parqueadero del CEC es una gran idea, que se debería expandir a los demás parqueaderos.

Rendimiento de Backend

En base a los resultados obtenidos por cada escenario propuesto podemos observar que, en todos los escenarios el tiempo mínimo de respuesta es 0.28 segundos, el mismo que resulta ser bueno cuando se realiza un conjunto pequeño de peticiones concurrentes como es el caso de los escenarios con 68 y 135 usuarios.

Observamos que en los escenarios con 135, 270 y 370 usuarios se generaron cuellos de botella cuando las peticiones concurrentes tienden a repetirse en lapsos de tiempo seguidos, debido a que el servidor tiende a saturarse al recibir tanta carga de manera seguida y por la

cantidad de datos que debe responder, esto porque las características del servidor en este caso no son las adecuadas ya que cuenta con solo 1GB de memoria RAM, más detalles del servidor se pueden observar en el Anexo 6 – Pruebas de Rendimiento.

Sin embargo, podemos decir que a pesar de las limitaciones del servidor el rendimiento en los escenarios con 68, 135 y 270 usuarios fueron los esperados, consideramos que los escenarios con 68 y 135 usuarios serán los que más ocurran en tiempos seguidos mientras que los escenarios con 270 y 370 usuarios sucederán en muy pocas ocasiones y no a los niveles que se realizaron en las pruebas, en los cuales se llegó a tener un máximo de 4050 peticiones en tiempos muy cortos.

Rendimiento de Aplicación Web

Con respecto a las pruebas de rendimiento de la aplicación web denominada Park EPN Management observamos que el consumo de CPU es elevado cuando hacemos peticiones que nos traen una gran cantidad de datos como es en el caso para ver las plazas de un determinado parqueadero y esto debido a que constantemente se está consultando el estado de dichas plazas, sin embargo, el consumo de CPU es adecuado cuando se están realizando otras acciones.

Con respecto al uso de memoria observamos que no presenta una carga elevada de memoria, sin embargo, destacamos al navegador Firefox ya que en este se obtuvo la menor cantidad de memoria con respecto a los otros navegadores analizados.

Con respecto al uso de RED observamos que los tiempos de respuesta están conforme a las pruebas obtenidas en el rendimiento del servidor y nuevamente podemos destacar al navegador Firefox debido a que tuvo el menor tiempo de respuesta al realizar la misma petición al servidor con respecto de los otros dos navegadores.

Finalmente, el uso de las herramientas online PageSpeed y GTmetrix ayudaron a mejorar la velocidad de carga de nuestra aplicación web, esto porque se pudo cumplir con algunas de las sugerencias planteadas en los análisis realizados por dichas herramientas.

Rendimiento de Aplicación Móvil

En base a los resultados obtenidos de las pruebas de rendimiento de la aplicación móvil observamos que el consumo de CPU es elevado en las pantallas donde se encuentra el mapa

de Google Maps y las plazas del parqueadero del CEC, ya que en estas pantallas se maneja una gran cantidad de información por procesar.

Con respecto al consumo de Memoria observamos un intervalo de uso total entre 190 MB y 250 MB de memoria y los tipos de memoria que más se usan son la memoria nativa y la memoria gráfica, estos consumos altos ocurre en las pantallas donde se muestra el mapa de Google Maps y las plazas del parqueadero del CEC.

Por último, en el consumo de Red observamos que el tiempo de respuesta en las pantallas que usan los servicios del Backend son congruentes a las pruebas obtenidas en el rendimiento del servidor.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Las aplicaciones desarrolladas en este Proyecto de Titulación son de gran utilidad para complementar el actual sistema de ingreso al parqueadero del CEC, permiten gestionar el parqueadero del CEC con el uso de una aplicación web y visualizar la disponibilidad de plazas en tiempo real por medio de una aplicación móvil lo cual no se encuentra implementado actualmente en la EPN.
- Las herramientas de desarrollo utilizadas para llevar a cabo este proyecto apoyaron de manera efectiva a la creación de estas aplicaciones. Debido a que permitieron facilitar el trabajo para desarrollar las aplicaciones móvil y web, así como también implementar una respuesta en tiempo real con la utilización del servidor y los dispositivos como el sensor ultrasónico y la Raspberry Pi.
- El análisis de los distintos tipos de sensores permitió seleccionar al sensor ultrasónico. El cuál fue adecuado para conocer la disponibilidad de una plaza en un parqueadero y con la integración de un Raspberry Pi se logró enviar la información en tiempo real para poder visualizarla en la aplicación móvil y web.
- El uso de la metodología Scrum fue adecuada para el desarrollo de la aplicación móvil y web. Ya que, esta metodología permitió adaptarse a las necesidades del proyecto. Además, las Reuniones Diarias y el uso de gráficos Burndown permitió observar a detalle la evolución de cada uno de los Sprints realizados. Sin embargo, en algunas tareas nos encontramos con retrasos, los mismos que fueron solucionados al aumentar el esfuerzo en próximos días, logrando cumplir con las fechas establecidas en el proyecto.
- Para la toma de requerimientos se realizó reuniones con un representante, el mismo que nos proporcionó la mayoría de los requerimientos, además se realizó un estudio de campo en el parqueadero del CEC para obtener información actual y fundamental que permitió describir los requisitos definitivos.
- En el país actualmente existen aplicaciones como AppSert, SSpot y SEQ que permiten conocer los estacionamientos cercanos a un determinado punto, estas características también la podemos encontrar en Google Maps. Sin embargo, la aplicación Park EPN permite ver la disponibilidad de plazas no solamente en cifras sino también permite observar cuales se encuentran libres en una imagen descriptiva del Parqueadero del

CEC, y su distribución en cada uno de los subsuelos. Además, proporciona información fundamental y necesaria relacionada al parqueadero como: las entradas y salidas en cada uno de los subsuelos.

- Los resultados de las pruebas de usabilidad realizadas a la aplicación móvil mostraron que la funcionalidad principal para conocer la disponibilidad de parqueaderos tuvo una aceptación muy buena del 80% y buena del 20%. Cumpliendo con el objetivo principal del presente proyecto.
- Los resultados de las pruebas en tiempo real mostraron que la ubicación del dispositivo fue el adecuado, debido a que el sensor logró captar la señal de la disponibilidad de una plaza cuando ésta se encontraba libre u ocupada por un vehículo, así mismos se logró visualizar dicha disponibilidad en tiempo real en la aplicación móvil Park EPN.

5.2 Recomendaciones

- En el presente Proyecto de Titulación se utilizaron dispositivos basados en su funcionamiento y principalmente costo, debido a limitaciones en el presupuesto, pero consideramos que no son los adecuados para ambientes productivos, por tal motivo, recomendamos investigar dispositivos especializados en conocer la disponibilidad de parqueaderos que puedan suplantar a los sensores utilizados.
- El presente proyecto podría ser de gran ayuda para parqueaderos públicos como privados. Por tal motivo, se recomienda plantear la opción de implementarlo en otros establecimientos, que deseen exhibir su servicio de parqueo y cuenten con un presupuesto limitado.
- Se recomienda continuar con el desarrollo de las aplicaciones presentadas en el presente Proyecto de Titulación, ya que se pueden añadir más funcionalidades, como, por ejemplo: reserva de plazas, visualización virtual del parqueadero, que por motivos del alcance en el presente proyecto no se las llegó a implementar.
- En el presente Proyecto de Titulación se utilizó un Raspberry Pi versión 2B, al mismo que se tuvo que integrar un adaptador wifi USB, por lo que se recomienda utilizar un Raspberry Pi versión 3 o superior, ya que estos cuentan con un módulo wifi integrado.
- Se recomienda alojar al Backend de las aplicaciones web y móvil en un servidor con mejores características en capacidad de RAM y disco, de esta manera se puede aumentar la capacidad de usuario concurrentes en las aplicaciones y también evitar

cuellos de botella cuando la carga de usuarios supere los límites establecidos en las pruebas realizadas en el presente proyecto.

- Se recomienda la utilización de herramientas online como: GTmetrix y PageSpeed, que permiten conocer deficiencias en páginas web con relación a su rendimiento. Además, facilitan recomendaciones para solventar estas deficiencias en ambientes reales móviles y de escritorio.

6. REFERENCIAS BIBLIOGRÁFICAS

- [1] INEC, «Anuario de Estadísticas de Transporte 2017,» 2018. [En línea]. Disponible en: http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Estadistica%20de%20Transporte/2017/2017_TRANSPORTE_PRESENTACION.pdf. [Último acceso: 18 Marzo 2019].
- [2] INEC, «Anuario de Transporte 2016,» 2016. [En línea]. Disponible en: http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Estadistica%20de%20Transporte/2016/2016_AnuarioTransportes_%20Principales%20Resultados.pdf. [Último acceso: 18 Marzo 2019].
- [3] K. Guamaní, «Estimación de los Costos Económicos de la Gestión Vehicular en Quito en el año 2016,» Escuela politécnica Nacional, 2017. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/18787>. [Último acceso: 18 Marzo 2019].
- [4] Secretaría de Movilidad, «Diagnostico de la Movilidad en el Distrito Metropolitano de Quito Para el Plan Metropolitano de Desarrollo Territorial (PMOT),» 2014, [En Línea]. Disponible en: <http://gobiernoabierto.quito.gob.ec/wp-content/uploads/documentos/pdf/diagnosticomovilidad.pdf?fbclid=IwAR14UiAnnndwEx51SpmqLaUvDYraJTah9b2vctMg2nnNYvHRFNG2kl-KnBU>. [Último acceso: 21 04 2019].
- [5] Municipio de Quito «Ordenanza Metropolitana N° 170,» 2017. [En Línea]. Disponible en: http://www7.quito.gob.ec/mdmq_ordenanzas/Ordenanzas/ORDENANZAS%20MUNICIPALES%202017/ORDM%20-%20170%20%20%20%20%20%20%20%20%20Sistema%20de%20Estacionamientos%20y%20Terminales%20Terrestres%20del%20DMQ..pdf. [Último acceso: 01 Marzo 2020].

- [6] D. Veintimilla y Y. Siguenca, «Diseño de un Sistema Inteligente de Parqueo Vehicular Mediante Videograbación e Implementación de un Prototipo de Prueba para la FIEE,» Universidad Escuela Politécnica Nacional, 2014, [En línea]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/7853/4/CD-5666.pdf>. [Último acceso: 01 Marzo 2020].
- [7] Xatako, «A finales de 2018 habrá 3.000 millones de usuarios de smartphones en el mundo, según Newzoo,» 2020, [En línea]. Disponible en: <https://www.xatakamovil.com/mercado/a-finales-2018-habra-3-000-millones-usuarios-smartphones-mundo-newzoo>. [Último acceso: 01 Marzo 2020].
- [8] G. Chanchí, W. Y. Campo, J. P. Amaya y J. L. Arciniegas, «Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON,» 2011, [En línea]. Disponible en: <https://www.seer.ufrgs.br/cadernosdeinformatica/article/view/v6n1p233-240/11807>. [Último acceso: 19 Noviembre 2019].
- [9] E. R. Zulian, «Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST,» Universidad de Belgrado, 2010, [En línea]. Disponible en: http://190.221.29.250/bitstream/handle/123456789/640/354_Tesina_Zulian.pdf?sequence=1&isAllowed=y. [Último acceso: 19 Noviembre 2019].
- [10] Wiboo, «¿Qué son las Aplicaciones Web? Ventajas y Tipos de Desarrollo Web,» 2010, [En línea]. Disponible en: <https://wiboomedia.com/que-son-las-aplicaciones-web-ventajas-y-tipos-de-desarrollo-web/>. [Último acceso: 01 Marzo 2020].
- [11] M. Gendra, «Aplicaciones Web Vs Aplicaciones de Escritorio,» 2010, [En línea]. Disponible en: <http://www.marianogendra.com.ar/Articulos/aplicaciones-web-vs-escritorio>. [Último acceso: 01 Marzo 2020].

- [12] Mecatrónica Latam, «Sensores,» 2010, [En línea]. Disponible en:
<https://www.mecatronicalatam.com/es/tutoriales/sensores/>. [Último acceso: 01 Marzo 2020].
- [13] Mecatrónica Latam, «Sensor de Temperatura,» 2010, [En línea]. Disponible en:
<https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-temperatura>.
[Último acceso: 01 Marzo 2020].
- [14] Mecatrónica Latam, «Sensor de Movimiento,» 2010, [En línea]. Disponible en:
<https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-movimiento>.
[Último acceso: 01 Marzo 2020].
- [15] Mecatrónica Latam, «Sensor de Distancia,» 2010, [En línea]. Disponible en:
<https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-distancia>.
[Último acceso: 01 Marzo 2020].
- [16] Mecatrónica Latam, «Sensor de Luz,» 2010, [En línea]. Disponible en:
<https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-luz>. [Último
acceso: 01 Marzo 2020].
- [17] Mecatrónica Latam, «Sensor de Proximidad,» 2010, [En línea]. Disponible en:
<https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-proximidad>.
[Último acceso: 01 Marzo 2020].
- [18] Mecatrónica Latam, «Sensor de Presión,» 2010, [En línea]. Disponible en:
<https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-presion>.
[Último acceso: 01 Marzo 2020].
- [19] Mecatrónica Latam, «Sensor de Posición,» 2010, [En línea]. Disponible en:
<https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-posicion>.
[Último acceso: 01 Marzo 2020].

- [20] Mecatrónica Latam, «Sensor de Color,» 2010, [En línea]. Disponible en: <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-color>. [Último acceso: 01 Marzo 2020].
- [21] Mecatrónica Latam, «Sensor de Humedad,» 2010, [En línea]. Disponible en: <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-humedad>. [Último acceso: 01 Marzo 2020].
- [22] Mecatrónica Latam, «Sensor Magnético,» 2010, [En línea]. Disponible en: <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-magnetico>. [Último acceso: 01 Marzo 2020].
- [23] Mecatrónica Latam, «Sensor Mecánico,» 2010, [En línea]. Disponible en: <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensores-mecanicos>. [Último acceso: 01 Marzo 2020].
- [24] Nailamp Mechatronics, «Sensor Ultrasonido HC-SR04,» 2011, [En línea]. Disponible en: <https://nailampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>. [Último acceso: 01 Marzo 2020].
- [25] Raspberry Shop, «Raspberry Pi,» 2019, [En línea]. Disponible en: <https://www.raspberrypi.es>. [Último acceso: 01 Marzo 2020].
- [26] Bricogeek, «Raspberry Pi 2 Model B (1GB),» 2019, [En línea]. Disponible en: <https://tienda.bricogeek.com/descatalogado/718-raspberry-pi-2-model-b-1gb.html>. [Último acceso: 01 Marzo 2020].
- [27] C. Avalos, «Comparativa Metodologías Agiles,» 2015, [En línea]. Disponible en: https://www.academia.edu/8058844/2.6_Comparativa_Metodolog%C3%ADas_Agiles. [Último acceso: 19 Noviembre 2019].

- [28] K. Schwaber y J. Sutherland, «La Guía de Scrum,» 2016, [En línea]. Disponible en: <https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf#zoom=100>. [Último acceso: 19 Noviembre 2019].
- [29] Software Engineering Standards Committee, «IEEE recommended practice for architectural description of software-intensive systems,» Technical Report IEEE Std 1471-2000, IEEE Computer Society, 2000. [Último acceso: 19 Noviembre 2019].
- [30] Y. F. Romero, Y. D. González, «Patrón Modelo-Vista-Controlador,» 2012, [En línea]. Disponible en: <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10>. [Último acceso: 21 Enero 2020].
- [31] E. R. Zulian, «Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST,» Universidad de Belgrado, 2010, [En línea]. Disponible en: http://190.221.29.250/bitstream/handle/123456789/640/354_Tesina_Zulian.pdf?sequence=1&isAllowed=y. [Último acceso: 19 Noviembre 2019].
- [32] Balsamiq Studios, «Balsamiq Documentation» 2019, [En línea]. Disponible en: <https://balsamiq.com/wireframes/desktop/docs/overview/>. [Último acceso: 21 Enero 2020].
- [33] GitHub, «How developers work» 2019, [En línea]. Disponible en: <https://github.com/features>. [Último acceso: 21 Enero 2020].
- [34] Novalys, «What is SAP PowerDesigner» 2019, [En línea]. Disponible en: <https://www.powerdesigner.biz>. [Último acceso: 21 Enero 2020].
- [35] The PostgreSQL Global Development Group, «PostgreSQL About,» 2019, [En línea]. Disponible en: <https://www.postgresql.org/about/>. [Último acceso: 19 Noviembre 2019].

- [36] Android Studio, «Kotlin,» 2019, [En línea]. Disponible en:
<https://developer.android.com/kotlin/learn>. [Último acceso: 19 Noviembre 2019].
- [37] Jet Brains y Kotlin Foundation, «Kotlin,» 2019, [En línea]. Disponible en:
<https://kotlinlang.org/docs/reference/android-overview.html>. [Último acceso: 19 Noviembre 2019].
- [38] IntelliJ IDEA «WebStorm,» 2019, [En línea]. Disponible en:
<https://www.jetbrains.com/webstorm/features/>. [Último acceso: 19 Noviembre 2019].
- [39] Microsoft, «TypeScript,» 2019, [En línea]. Disponible en:
<https://www.typescriptlang.org>. [Último acceso: 19 Noviembre 2019].
- [40] Linux Foundation y Node.js Foundation, «Acerca de Node.js,» 2019, [En línea].
Disponible en: <https://nodejs.org/es/about/>. [Último acceso: 19 Noviembre 2019].
- [41] J. Staron, «Nest Js,» 2019, [En línea]. Disponible en: <https://docs.nestjs.com/>.
[Último acceso: 19 Noviembre 2019].
- [42] Angular, «Architecture overview,» 2019, [En línea]. Disponible en:
<https://angular.io/guide/architecture>. [Último acceso: 19 Noviembre 2019].
- [43] Dan Radigan, «Puntos de historia y estimación,» 2019, [En línea]. Disponible en:
<https://www.atlassian.com/es/agile/project-management/estimation>. [Último acceso:
01 Marzo 2020].
- [44] ISO25000, «ISO/IEC 25000,» 2019, [En línea]. Disponible en:
<https://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&limitstart=0>.
[Último acceso: 01 Agosto 2020].
- [45] ISO25000, «ISO/IEC 25000,» 2019, [En línea]. Disponible en:
<https://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&start=3>.
[Último acceso: 01 Marzo 2020].

- [46] Nielsen Norman Group, «Usability 101: Introduction to Usability,» 2019, [En línea]. Disponible en: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>. [Último acceso: 01 Marzo 2020].
- [47] ISTQB, «Foundation Level Specialist Syllabus Performance Testing,» 2018, [En línea]. Disponible en: <https://www.istqb.org/downloads/send/59-performance-testing/239-istqb-ctfl-pt-syllabus-2018-ga.html>. [Último acceso: 01 Marzo 2020].
- [48] Apache Software Foundation, «Apache JMeter,» 2019, [En línea]. Disponible en: <https://jmeter.apache.org/>. [Último acceso: 01 Marzo 2020].
- [49] SmartBear Software, «SoapUI,» 2019, [En línea]. Disponible en: <https://www.soapui.org/professional/loadui-pro/features/>. [Último acceso: 01 Marzo 2020].
- [50] Neotys, «NeoLoad Features,» 2019, [En línea]. Disponible en: <https://www.neotys.com/neoload/features>. [Último acceso: 01 Marzo 2020].
- [51] Micro Focus, «LoadRunner,» 2019, [En línea]. Disponible en: <https://www.microfocus.com/en-us/products/loadrunner-professional/overview>. [Último acceso: 01 Marzo 2020].
- [52] GTmetrix, «Make your website pages fast on all devices,» 2019, [En línea]. Disponible en: <https://gtmetrix.com>. [Último acceso: 01 Agosto 2020].
- [53] PageSpeed, «About PageSpeed Insights,» 2019, [En línea]. Disponible en: <https://developers.google.com/speed/docs/insights/v5/about>. [Último acceso: 01 Agosto 2020].
- [54] Lighthouse, «Lighthouse performance scoring,» 2019, [En línea]. Disponible en: <https://web.dev/performance-scoring/>. [Último acceso: 01 Agosto 2020].

- [55] Android Studio, «Cómo ver asignaciones de memoria y pila de Java con Memory Profiler,» 2019, [En línea]. Disponible en:
<https://developer.android.com/studio/profile/memory-profiler?hl=es-419>. [Último acceso: 01 Marzo 2020].

7. ANEXOS

7.1 Anexo 1 – Pila del Producto

7.2 Anexo 2 – Mockups

7.3 Anexo 3 – Base de Datos

7.4 Anexo 4 – Gráficos Burndown

7.5 Anexo 5 – Encuestas

7.6 Anexo 6 – Pruebas

7.7 Anexo 7 – Manuales

7.8 Anexo 8 – Scripts