



UNIVERSIDAD  
CATÓLICA  
DE CUENCA

**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE INFORMÁTICA,  
CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN  
TECNOLÓGICA**

**CARRERA DE INGENIERIA EN SISTEMAS DE  
INFORMACIÓN**

**PROPUESTA Y DESARROLLO DE UN PROTOTIPO DE  
APLICACIÓN MÓVIL HÍBRIDA PARA EL REGISTRO,  
CONSULTA Y PAGO DE MULTAS DE SEMERTAZ**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

**AUTOR: DIEGO GABRIEL FAJARDO UZHCA**

**MARLON BRIANT BERMEJO LEMA**

**DIRECTOR: ING. JUAN PABLO CUENCA TAPIA, MSC**

**AZOGUES - ECUADOR**

**2023**

**DIOS, PATRIA, CULTURA Y DESARROLLO**



**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE INFORMÁTICA,  
CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN  
TECNOLÓGICA**

**CARRERA DE INGENIERIA EN SISTEMAS DE  
INFORMACIÓN**

**PROPUESTA Y DESARROLLO DE UN PROTOTIPO DE  
APLICACIÓN MÓVIL HÍBRIDA PARA EL REGISTRO,  
CONSULTA Y PAGO DE MULTAS DE SEMERTAZ**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

**AUTOR: DIEGO GABRIEL FAJARDO UZHCA**

**MARLON BRIANT BERMEJO LEMA**

**DIRECTOR: ING. JUAN PABLO CUENCA TAPIA, MSC**

**AZOGUES - ECUADOR**

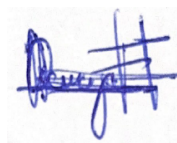
**2023**

**DIOS, PATRIA, CULTURA Y DESARROLLO**

**Declaratoria de Autoría y Responsabilidad**

**Diego Gabriel Fajardo Uzhca** portador de la cédula de ciudadanía N° **0302986153**. Declaro ser el autor de la obra: **“Propuesta y desarrollo de un prototipo de aplicación móvil híbrida para el registro, consulta y pago de multas de SEMERTAZ”**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Azogues, **02 de mayo del 2023**



F: .....

**Diego Gabriel Fajardo Uzhca**

**C.I. 0302986153**

**Declaratoria de Autoría y Responsabilidad**

**Marlon Briant Bermejo Lema** portador de la cédula de ciudadanía N° **0350259750**. Declaro ser el autor de la obra: “**propuesta y desarrollo de un prototipo de aplicación móvil híbrida para el registro, consulta y pago de multas de SEMERTAZ**”, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Azogues, **02 de mayo del 2023**



F: .....

**Marlon Briant Bermejo Lema**

**C.I. 0350259750**

## CERTIFICACIÓN DEL TUTOR

### A quien corresponda:

Yo, Juan Pablo Cuenca Tapia Docente de la Unidad Académica de Informática, Ciencias de la Computación e Innovación Tecnológica a petición de parte interesada:

Certifico que los estudiantes Diego Gabriel Fajardo Uzhca con C.I. 0302986153 y Marlon Briant Bermejo Lema con C.I. 0350259750 han desarrollado el trabajo de titulación **"PROPUESTA Y DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL HÍBRIDA PARA EL REGISTRO, CONSULTA Y PAGO DE MULTAS DE SEMERTAZ"**, previo a la obtención del título de Ingeniero de Sistemas, de la Unidad Académica de Informática, Ciencias de la Computación e Innovación Tecnológica, el mismo que ha estado bajo mi supervisión y cumple con todas las observaciones realizadas por el tribunal evaluador; por lo que las ideas u opiniones vertidas en el mismo son de exclusiva responsabilidad de los autores.

Es todo cuanto puedo certificar en honor a la verdad, autorizando al peticionario, hacer uso del presente documento, el uso que creyere conveniente.



Ing. Juan Pablo Cuenca Tapia, Msc.

**DIRECTOR**

## **DEDICATORIA**

Este logro está dedicado a mi madre Mariana, quien me enseñó el verdadero significado del amor, y mi hermano Patricio, quien me enseñó la importancia de la lealtad y el compromiso. Sé que ambos estarían muy orgullosos de mí y de todo lo que he logrado. Los llevo siempre en mi corazón y pensamientos.

**-Diego Fajardo**

## **AGRADECIMIENTOS**

Quiero expresar mi agradecimiento a mis padres Mariana y Héctor por su inquebrantable confianza en mí y por brindarme las herramientas necesarias para alcanzar mis metas académicas y profesionales. Sus consejos, enseñanzas y palabras de aliento me han guiado a lo largo de este proceso y han sido una fuente inagotable de inspiración para mí.

A mis hermanos, Geovanny, Nube y Diana por ser mis cómplices, amigos y compañeros de vida. Siempre han estado ahí para escucharme, ayudarme y brindarme su apoyo incondicional. Agradezco su paciencia y comprensión.

Sin su amor y respaldo, este logro no hubiera sido posible. Agradezco infinitamente todo lo que han hecho por mí y espero poder retribuirles de alguna manera todo lo que me han brindado a lo largo de mi vida.

**-Diego Fajardo**

## **DEDICATORIA**

Dedico esta tesis con mucho amor y A mi abuela que siempre ha sido una gran inspiración en mi vida, espero poder seguir honrando tu legado A mi abuelo por tus palabras sabias, tu paciencia y seguir aprendiendo de tus enseñanzas. A mis padres, quienes han sido mi mayor apoyo y mis guías a lo largo de mi vida. A mi tía, quien ha sido un gran modelo a seguir. Les dedico este logro con todo mi corazón y espero poder seguir contando con su amor y apoyo en mis futuros proyectos.

**-Marlon Bermejo**

## **AGRADECIMIENTOS**

Quiero expresar mi más profundo agradecimiento a Dios por guiarme y darme la fuerza necesaria para completar este importante logro en mi vida. Su amor y su guía me han sostenido en los momentos de dificultad y me han impulsado hacia adelante en los momentos de éxito.

También quisiera agradecer a mis padres Gladys y Nelson por su amor incondicional, su apoyo constante y sus sacrificios incalculables a lo largo de mi educación. Sin su orientación y motivación, no estaría aquí hoy celebrando este logro.

Agradezco a mis profesores por su dedicación, su paciencia y su experiencia en guiarme a través de los desafíos académicos y en ayudarme a desarrollar mis habilidades y conocimientos.

Por último, pero no menos importante, quisiera agradecer a mis amigos por su amor, su aliento y su apoyo incondicional. Su presencia en mi vida ha sido un regalo invaluable y no puedo expresar lo agradecido que estoy por su amistad.

Gracias a todos por creer en mí y por ser parte de mi viaje hacia la realización de mi sueño.

**-Marlon Bermejo**

## **Propuesta y desarrollo de un prototipo de aplicación móvil híbrida para el registro, consulta y pago de multas de SEMERTAZ**

Diego Gabriel Fajardo Uzhca, Marlon Briant Bermejo Lema, Ing. Juan Pablo Cuenca

Universidad Católica de Cuenca, [dgfajardou53@est.ucacue.edu.ec](mailto:dgfajardou53@est.ucacue.edu.ec);  
[mbermejol50@est.ucacue.edu.ec](mailto:mbermejol50@est.ucacue.edu.ec); [jcuenca@ucacue.edu.ec](mailto:jcuenca@ucacue.edu.ec)

### **RESUMEN.-**

El Servicio Municipal de Estacionamiento Rotativo Tarifado de Azogues (SEMERTAZ) es el encargado de controlar los espacios públicos de estacionamiento vehicular de la zona céntrica de la ciudad, con el objetivo de mejorar la movilidad vehicular. Sin embargo, SEMERTAZ no dispone de una aplicación móvil que permita el registro, consulta y pago de infracciones de manera automatizada.

Por lo expuesto anteriormente, se optó por desarrollar un prototipo de aplicación móvil híbrida en IONIC mediante la metodología ágil de desarrollo SCRUM, con el objetivo de mejorar los procesos que maneja SEMERTAZ. La aplicación se enfoca en la optimización del tiempo de registro, consulta y pago de infracciones usando dispositivos móviles en comparación a la gestión actual que ofrece la Empresa.

Para el desarrollo de esta investigación se describen 4 capítulos. En el Capítulo 1 se desarrolla el Marco Referencial, que consta de antecedentes, planteamiento del problema, justificación, objetos y alcance. En el Capítulo 2 se realiza la fundamentación teórica sobre el tipo de sistemas operativos móviles y aplicaciones móviles, herramientas útiles para el desarrollo de aplicaciones híbridas, el uso de bases de datos no relacionales y un análisis sobre metodologías ágiles de desarrollo. En el Capítulo 3 se aplica la metodología ágil seleccionada para el desarrollo de la aplicación híbrida móvil. Finalmente, en el Capítulo 4 se muestran los resultados del desarrollo de la aplicación, además se exponen las conclusiones y recomendaciones.

*Palabras clave.*- aplicaciones híbridas, SCRUM, dispositivos móviles, parqueo tarifado vehicular

## **Propuesta y desarrollo de un prototipo de aplicación móvil híbrida para el registro, consulta y pago de multas de SEMERTAZ**

Diego Gabriel Fajardo Uzhca, Marlon Briant Bermejo Lema, Ing. Juan Pablo Cuenca

Universidad Católica de Cuenca, [dgfajardou53@est.ucacue.edu.ec](mailto:dgfajardou53@est.ucacue.edu.ec);  
[mbbermejol50@est.ucacue.edu.ec](mailto:mbbermejol50@est.ucacue.edu.ec); [jcuenca@ucacue.edu.ec](mailto:jcuenca@ucacue.edu.ec)

### **ABSTRACT.-**

The Azogues Rotary Tarrified Municipal Parking Service (SEMERTAZ by its Spanish acronym) controls public parking spaces in the city's downtown area, aiming to improve vehicular mobility. However, SEMERTAZ has no mobile application permit automated registration, consultation, and payment of infractions.

Therefore, it was decided to develop a prototype of a hybrid mobile application in IONIC using the agile development methodology SCRUM to improve the processes managed by SEMERTAZ. The application focuses on optimizing the time of registration, consultation, and fines payment using mobile devices compared to the current management offered by the Company.

For the development of this research, four chapters are described. Chapter 1 concerns the reference frame: background, problem statement, justification, objects, and scope. Chapter 2 contains the theoretical foundation on the type of mobile operating systems and applications, valuable tools for developing hybrid applications, using non-relational databases, and analysis of agile development methodologies. Chapter 3 applies the selected agile methodology selected to design the hybrid mobile application. Finally, Chapter 4 shows the application development results, conclusions, and recommendations.

*Keywords.-* hybrid applications, SCRUM, mobile devices, paid parking

# TABLA DE CONTENIDO

<b>DEDICATORIA</b> .....	<b>IV</b>
<b>AGRADECIMIENTOS</b> .....	<b>V</b>
<b>DEDICATORIA</b> .....	<b>VI</b>
<b>AGRADECIMIENTOS</b> .....	<b>VII</b>
<b>RESUMEN.-</b> .....	<b>1</b>
<b>ABSTRACT.-</b> .....	<b>3</b>
<b>TABLA DE CONTENIDO</b> .....	<b>4</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>7</b>
<b>ÍNDICE DE ILUSTRACIONES</b> .....	<b>8</b>
<b>ÍNDICE DE ANEXOS</b> .....	<b>9</b>
<b>CAPÍTULO 1</b> .....	<b>10</b>
1. Marco referencial.....	10
1.1. Antecedentes.....	10
1.2. Planteamiento del problema .....	11
1.3. Justificación.....	12
1.4. Objetivos.....	13
Objetivo General .....	13
Objetivos Específicos .....	13
1.5. Alcance .....	14
1.6. Estado del Arte .....	14
<b>CAPÍTULO 2</b> .....	<b>17</b>
2. Marco Teórico .....	17
2.1. Sistemas Operativos Móviles .....	17
2.1.1. iOS.....	17
2.1.2. Android.....	17
2.2. Aplicaciones móviles.....	18
2.2.1. Aplicaciones básicas.....	18
2.2.2. Aplicaciones Nativas .....	18
2.2.3. Aplicaciones Híbridas .....	18
2.2.4. Aplicaciones Web.....	19
2.3. Frontend.....	19
2.3.1. Frameworks .....	20
2.3.1.1. Ionic .....	20
2.3.1.2. React Native.....	20
2.3.1.3. Flutter.....	21
2.4. Backend .....	22
2.4.1. Node.Js .....	22
2.4.2. Deno .....	23

2.4.3. Base de datos NoSQL.....	24
2.4.3.1. MongoDB.....	26
2.4.3.2. MongoDB Atlas .....	27
2.5. Protocolo de Transferencia de Hiper Texto.....	27
2.6. API REST .....	27
2.7. Arquitectura cliente - servidor.....	28
2.8. Metodologías ágiles de desarrollo de software .....	29
2.8.1. Scrum.....	30
2.8.1.1. Equipos Scrum .....	31
2.8.1.2. Artefactos de Scrum.....	31
2.8.1.3. Procesos Scrum .....	31
2.8.1.4. Eventos de Scrum.....	32
2.9. Editores de código .....	33
2.9.1. Visual Studio Code.....	33
2.9.2. Atom Editor .....	33
2.9.3. Sublime Text .....	33
2.10. Plataformas de pago en línea .....	35
2.10.1. PayPal.....	35
2.11. Servicio Municipal de Estacionamiento Rotativo Tarifado de Azogues.....	36
2.11.1. Objetivos de SEMERTAZ.....	37
2.11.2. Administración .....	38
2.11.3. Días y horario de funcionamiento .....	39
2.11.4. Tarifas.....	39
2.11.5. Infracciones y sanciones.....	39
<b>CAPÍTULO 3.....</b>	<b>41</b>
3. Desarrollo de la metodología.....	41
3.1. Análisis de requerimientos .....	41
3.2. Implementación de la metodología .....	41
3.3. Definición de los Sprints .....	43
3.4. Planificación de los Sprints .....	45
<b>CAPÍTULO 4.....</b>	<b>49</b>
4. Resultados.....	49
4.1. Resultados del Sprint 1 .....	49
4.2. Resultados del Sprint 2.....	59
4.3. Resultado del Sprint 3 .....	77
4.4. Entregables Finales.....	98
4.5. Conclusiones y recomendaciones.....	99
4.5.1. Conclusiones .....	99
4.5.2. Recomendaciones.....	99
Bibliografía.....	101
Anexos .....	104
Anexo A. Muestra de la población.....	104
Anexo B. Encuesta a controladores de tráfico de SEMETAZ .....	105

Anexo C. Resultados de la encuesta realizada a controladores de tráfico de SEMERTAZ..... 107

## ÍNDICE DE TABLAS

Tabla 1 - Aplicaciones móviles.....	19
Tabla 2 - Frameworks de desarrollo móvil.....	21
Tabla 3 - Lenguajes del servidor.....	23
Tabla 4 - Bases de datos NoSQL comparativa.....	25
Tabla 5 - Comparativa Metodologías Ágiles.....	30
Tabla 6 - Editores de código.....	34
Tabla 7 - Rutas SEMERTAZ.....	37
Tabla 8 - Valor de las sanciones.....	40
Tabla 9 - Requerimientos Funcionales.....	41
Tabla 10 - Ejemplo 1 de Historia de Usuario.....	42
Tabla 11 - Ejemplo 2 de Historia de Usuario.....	42
Tabla 12- Product Backlog.....	43
Tabla 13 - Sprint 1.....	44
Tabla 14 - Sprint 2.....	44
Tabla 15 - Sprint 3.....	45
Tabla 16 - Planificación del primer Sprint.....	46
Tabla 17 - Planificación del segundo Sprint.....	46
Tabla 18 - Planificación del tercer Sprint.....	46
Tabla 19 - Taskboard inicial.....	47
Tabla 20 - Taskboard del primer Sprint.....	49
Tabla 21 - Taskboard del primer Sprint.....	52
Tabla 22 - Taskboard del primer Sprint.....	58
Tabla 23 - Taskboard del segundo Sprint.....	60
Tabla 24 - Taskboard del segundo Sprint.....	70
Tabla 25 - Taskboard del segundo Sprint.....	76
Tabla 26 - Taskboard del tercer Sprint.....	78
Tabla 27 - Taskboard del tercer Sprint.....	83
Tabla 28 - Taskboard del tercer Sprint.....	97

# ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Bases de datos más utilizadas en 2022.....	26
Ilustración 2 - Arquitectura cliente - servidor. ....	28
Ilustración 3 - Procesos Scrum. ....	32
Ilustración 4 - Editores de código más utilizados en 2022.....	34
Ilustración 5 - Burndown chart inicial.....	48
Ilustración 6 - Burndown chart del primer Sprint. ....	50
Ilustración 7 - Base de datos.....	51
Ilustración 8 - Burndown chart del primer Sprint. ....	53
Ilustración 9 - Interfaz de registro de personal en IOS.....	54
Ilustración 10 - Interfaz de registro de personal en Android.....	55
Ilustración 11 - Interfaz de personal registrados en IOS. ....	56
Ilustración 12 - Interfaz de personal registrados en Android. ....	57
Ilustración 13 - Burndown chart del primer Sprint. ....	59
Ilustración 14 - Burndown chart del segundo Sprint.....	61
Ilustración 15 - Interfaz de registro de infracciones en IOS.....	62
Ilustración 16 - Interfaz de registro de infracciones en Android.....	63
Ilustración 17 - Interfaz de infracciones registradas en IOS. ....	64
Ilustración 18 - Interfaz de infracciones registradas en Android. ....	65
Ilustración 19 - Interfaz para editar infracciones en IOS. ....	66
Ilustración 20 - Interfaz para editar infracciones en Android.....	67
Ilustración 21 - Interfaz de infracciones pagadas en IOS.....	68
Ilustración 22 - Interfaz de infracciones pagadas en Android.....	69
Ilustración 23 - Burndown chart del Sprint 2. ....	71
Ilustración 24 - Interfaz de login en IOS.....	72
Ilustración 25 - Interfaz de login en Android. ....	73
Ilustración 26 - Interfaz de bienvenida en IOS.....	74
Ilustración 27 - Interfaz de bienvenida en Android.....	75
Ilustración 28 - Burndown chart del segundo Sprint.....	77
Ilustración 29 -Burndown chart del tercer Sprint.....	78
Ilustración 30 - Interfaz de búsqueda de infracciones en IOS.....	79
Ilustración 31 - Interfaz de búsqueda de infracciones en Android.....	80
Ilustración 32 - Interfaz de búsqueda realizada en IOS.....	81
Ilustración 33 - Interfaz de búsqueda realizada en Android.....	82
Ilustración 34 - Burndown chart del Sprint 3. ....	84
Ilustración 35 - Botón de pago en IOS. ....	85
Ilustración 36 - Botón de pago en Android. ....	86
Ilustración 37 - Interfaz detalle de pago en IOS.....	87
Ilustración 38 - Interfaz detalle de pago en Android.....	88
Ilustración 39 - Interfaz inicio de sesión PayPal en IOS. ....	89
Ilustración 40 - Interfaz inicio de sesión PayPal en Android. ....	90
Ilustración 41 - Formulario pago por tarjeta de crédito o débito en IOS. ....	91
Ilustración 42 - Formulario pago por tarjeta de crédito o débito en Android.....	92
Ilustración 43 - Interfaz para realizar el pago y facturación en IOS. ....	93
Ilustración 44 - Interfaz para realizar el pago y facturación en Android.....	94
Ilustración 45 - Interfaz del pago realizado en IOS.....	95
Ilustración 46 - Interfaz del pago realizado en Android.....	96
Ilustración 47 - Burndown chart del tercer Sprint.....	98

## ÍNDICE DE ANEXOS

ANEXOS.....	104
Anexo A. Muestra de la población.....	104
Anexo B. Encuesta a controladores de tráfico de SEMETAZ.....	105
Anexo C. Resultados de la encuesta realizada a controladores de tráfico de SEMERTA.....	107

# CAPÍTULO 1

## 1. Marco referencial

### 1.1. Antecedentes

Desde hace varios años en las ciudades del Ecuador se ha venido implementando un servicio rotativo tarifado para controlar los espacios públicos destinados al estacionamiento vehicular, siendo estos manejados netamente por los Municipios respectivos que a través de ordenanzas manejan dicho servicio de acuerdo a sanciones previamente establecidas y pago de las mismas.

El Gad Municipal de Azogues en el 2011, apegándose a las disposiciones del Código Orgánico de Organización Territorial (COOTAD) crea el Servicio Municipal de Estacionamiento Rotativo Tarifado de Azogues (SEMERTAZ) encargado de controlar el espacio público destinado al estacionamiento vehicular en el centro de la ciudad, con el objetivo de mejorar la movilidad y disminuir la congestión vehicular producto de la falta de espacios de estacionamiento. [1]

El Servicio Municipal de Estacionamiento Rotativo Tarifado de Azogues (SEMERTAZ) a través de la ordenanza “Sustitutiva que norma el servicio municipal de estacionamiento rotativo tarifado en las vías públicas” [2] elaboraron parámetros como horarios, tarifas, rutas, sanciones y control con el fin de regular y gestionar adecuadamente su funcionamiento garantizando el cumplimiento de sus servicios.

Actualmente SEMERTAZ cuenta con 35 controladores de tráfico que se encargan de recorrer la zona central de la ciudad para controlar el uso del espacio destinado al estacionamiento vehicular, la misma que se encuentra dividida en 18 rutas. Así mismo existen 21 tipos de sanciones con un valor económico diferente para penalizar a los ciudadanos que hagan mal uso del espacio vehicular.

## **1.2. Planteamiento del problema**

Según la organización Internacional de fabricantes de Motor [3], existen 1.400 millones de automóviles activos que circulan a nivel mundial, es por ello que existe congestión vehicular en varios países incluyendo Ecuador en donde, según el INEC en 2019 [4], el parque automotor creció con 1,4 millones de vehículos, lo que ubico la cifra encima de los 2,4 millones de unidades a 2018. [4] En la ciudad de Azogues con una población total de 37244 habitantes de acuerdo al último Censo Nacional realizado por el INEC [5], es una de las ciudades con más habitantes en la provincia del Cañar, por lo que en estos últimos años ha existido un incremento de tránsito vehicular ya que el parque automotor tiene 23500 vehículos automotrices matriculados con una tasa de crecimiento anual del 6% aproximadamente. [6]

Lo descrito anteriormente afecta directamente a los ciudadanos de Azogues por la aglomeración vehicular que se produce dentro de la ciudad, por tal motivo la Municipalidad de Azogues ha optado por crear el Servicio Municipal de Estacionamiento Rotativo Tarifario de Azogues (SEMERTAZ) que se encarga de controlar los espacios en las vías públicas destinadas al estacionamiento vehicular en la zona céntrica de la ciudad que está dividida en 18 zonas. El control se realiza mediante la generación de infracciones que se aplican a conductores que hacen mal uso del espacio de estacionamiento.

Las infracciones se realizan mediante una cámara fotográfica para capturar la placa vehicular, modelo, marca y color del automóvil; una radio de comunicación con el fin de pasar la información a la oficina de SEMERTAZ que se registran en tablas en Microsoft Excel con los datos previamente enviados por los controladores de tránsito, así mismo al final de la jornada laboral deben presentar la tarjeta que justifica el motivo de la infracción en la oficina mediante un formulario y guardar las fotografías como evidencia de la multa cometida y poder realizar el cobro de la misma.

**Se realizó una encuesta a 32 de los 35 controladores de tráfico de acuerdo a la muestra obtenida con una confiabilidad del 95% y margen de error del 5% [Anexos**

Anexo A. Muestra de la población En el transcurso del día registran entre 5 a 10 infracciones (62.5%), siendo las más comunes: vehículo sin tarjeta (56.3%), mal uso de tarjeta de parqueo (53.1%) y por tiempo excedido (50%). Además, del total de infracciones registradas en un día, 5 (59.4%) presentan errores al momento de procesar la placa vehicular mediante la radio a la oficina. [Anexo C. Resultados de la encuesta realizada a controladores de tráfico de SEMERTAZ

Por otra parte, como usuarios de SEMERTAZ no se cuenta con un servicio para consultar y pagar las infracciones cometidas, siendo un inconveniente el trasladarse a la oficina central y el horario de atención.

Debido a lo antes mencionado, es necesario desarrollar una aplicación móvil híbridas bajo metodologías ágiles para el registro de multas de SEMERTAZ, permitiendo al controlador de tráfico generar y subir las multas directamente a una base de datos, dejando de lado la toma de fotografías y la comunicación de radio con la oficina optimizando su tiempo y evitar inconvenientes en la información que se registra, de la misma forma, las multas generadas estarán disponibles inmediatamente para su consulta y pago por parte de usuarios externos.

### **1.3. Justificación**

El Servicio Municipal de Estacionamiento Rotativo Tarifado de Azogues (SEMERTAZ) es el encargado de administrar y controlar el espacio público destinado al estacionamiento vehicular.

El problema surge que, al no contar con un sistema, existen inconvenientes al momento de registrar infracciones y consultarlas, debido a lo antes mencionado este documento se enfocará en automatizar los procesos de registro y consulta de infracciones.

Mediante el desarrollo de una aplicación híbrida para el registro, consulta y pago de infracciones de SEMERTAZ, los controladores de tráfico podrán registrar infracciones digitalmente y de forma instantánea, la aplicación permitirá seleccionar los datos como el tipo de sanción y su valor a cancelar de manera dinámica para evitar errores al registrar manualmente la infracción y tabular en Excel.

De igual manera, la información de la infracción registrada a través de la aplicación estará disponible inmediatamente gracias a una base de datos en la nube que permitirá al coordinador general o secretario acceder a la base de datos para realizar el cobro, editar en caso de que algún campo contenga errores o simplemente borrar la infracción.

El pago de infracciones registradas en SEMERTAZ se realizan en efectivo en la oficina central únicamente los días laborables en el horario de atención de 8:00 AM a 17:00 PM. Por lo que se pretende que la aplicación propuesta permita realizar pagos online desde cualquier lugar que estara disponibles las 24 horas del día siendo una opción favorable para los clientes.

## **1.4. Objetivos**

### **Objetivo General**

Desarrollar un prototipo de aplicación móvil híbridas a través de metodologías ágiles de desarrollo de software para automatizar el registro, consulta y pago de multas a infracciones de estacionamiento del Servicio Municipal de Estacionamiento Rotativo Tarifado de Azogues (SEMERTAZ).

### **Objetivos Específicos**

- Indagar sobre los conceptos teóricos de tecnologías de desarrollo híbrido y metodologías ágiles que brinden las bases necesarias para la fundamentación del proyecto.

- Recolectar información sobre el funcionamiento del Servicio Municipal de Estacionamiento Rotativo Tarifado de Azogues (SEMERTAZ).
- Construir el prototipo de la aplicación móvil híbridas ejecutando las fases de la metodología ágil que permita un óptimo desarrollo de software.

### **1.5. Alcance**

El presente trabajo incluye el desarrollo de una aplicación móvil híbrida para SEMERTAZ, la misma que estará disponible en diferentes sistemas operativos móviles que existen en el mercado. Previamente al desarrollo de la aplicación se realizará una comparativa entre frameworks y metodologías de desarrollo para seleccionar las que se adapten a las necesidades del proyecto, se creará el modelo de la base de datos no relacional en la nube, mientras que el servidor estará alojado localmente.

A continuación, se llevará a cabo la revisión de ordenanzas Municipales del uso del espacio público para recolectar información sobre el funcionamiento de SEMERTAZ, debido a políticas internas que limitan el acceso a la información respecto a infracciones y cobro de la mismas.

Además, se aplicará una encuesta a 32 de los 35 controladores de tráfico de acuerdo a la muestra calculada para conocer problemas y el tiempo empleado al momento de registrar una infracción. El periodo para realizar la encuesta y recolección de información es de 2 semanas y se dará por finalizado cuando los datos obtenidos sean tabulados. El objetivo final del proyecto es automatizar el registro, consulta y pago de multas mediante una aplicación híbrida. Para lograr este objetivo se utilizará de forma responsable la información recolectada.

El desarrollo de la aplicación se realizará en dos módulos; el primer módulo será exclusiva para el registro de multas por parte del personal de SEMERTAZ, el segundo módulo será exclusivo para la consulta y pago online de infracciones por parte de los usuarios externos.

## 1.6. Estado del Arte

En el año 2014 en la ciudad de Loja los estudiantes Luz Carmen Mato Jaramillo y María Jhoana Villa Buri dieron a conocer su trabajo de titulación denominado “tecnificación del sistema municipal de estacionamiento rotativo tarifado (SIMEERT) de la ciudad de Lola, moduló de servicios en aplicaciones móviles” que tiene como objetivo automatizar los procesos del Sistema de Estacionamiento Rotativo Tarifado de Loja (SIMERT) mediante el desarrollo de una aplicación destinada a la administración de vehículos, clientes, proveedores y generar reportes, además cuenta con funcionalidades para realizar el pago de multas y verificar espacios disponibles de estacionamiento. Para el desarrollo de esta solución se aplicó la metodología RUP, la guía de ingeniería SWEBOK y el framework CodeIgniter. [7]

En el año 2014 en la ciudad de Ibarra los estudiantes Irving Marlon Reascos Paredes y Luis Rodolfo Guamán Guamán en su trabajo de titulación denominado “Sistema de control del parqueo tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra” que tiene como objetivo mejorar y controlar los servicios de SISMERT, a través de un sistema informático que permite el registro de parqueo por parte del inspector de tráfico ya que en muchas ocasiones no llevan información real de cada registro. Para el desarrollo de este sistema se utilizó el framework Ext Js y las metodologías de desarrollo SCRUM y XP. [8]

En el año 2016 en la ciudad de Ibarra mediante la ordenanza que regula el SISMERT se optó por implementar una aplicación llamada SIP (Sistema Integral de Parqueo) para realizar un control más eficiente por parte de los controladores de tránsito, saber el histórico de cada vehículo. En cuanto al usuario a través de la aplicación móvil tiene la opción de realizar pagos directos de infracciones, conocer espacios disponibles de parqueo y recargar saldo. Al usar la aplicación se obtuvo un incremento del 16% del cobro de infracciones, se evidencio una mejora

del 10% en el control de los espacios disponibles de parqueo en comparación con años anteriores y el servicio al cliente mejoro notablemente. [9]

En el 2018 la Empresa Pública Municipal de Movilidad, Tránsito y Transporte de Cuenca (EMOV EP) implemento la aplicación appSERT para optimizar los procesos del Sistema de Estacionamiento Rotativo Tarifado (SERT). Los beneficios obtenidos al usar la aplicación por parte de los controladores de tráfico son: llevar a cabo un control, verificación y registro de infracciones digitalmente, y para los ciudadanos los ciudadanos tienen la posibilidad de comprobar los espacios de estacionamiento disponibles y cancelar infracciones adeudadas mediante métodos de pago en línea. [10]

En el estado del arte se ha demostrado investigaciones enfocadas en automatizar los procesos de los Servicios de Estacionamiento Rotativo Tarifado de distintas ciudades del Ecuador haciendo uso de tecnologías modernas y metodologías de desarrollo en la creación de una aplicación móvil con el fin de resolver problemas que se presentaban cotidianamente.

## **CAPÍTULO 2**

### **2. Marco Teórico**

#### **2.1. Sistemas Operativos Móviles**

Un sistema operativo es aquel que permite la interacción del software, hardware y usuario, se encarga de controlar y administrar los recursos disponibles en un dispositivo, principalmente fueron creados para computadoras, pero con el avance tecnológico y la aparición de nuevos dispositivos se crearon sistemas operativos exclusivos para dispositivos móviles, que de igual forma permiten la interacción y administración antes ya mencionada entre el software, hardware y usuario.[11]

##### **2.1.1. iOS**

Es un sistema operativo móvil de código cerrado creado por Apple en el 2007, desarrollado exclusivamente para el iPhone y posteriormente fue usado en el iPad, no permite la instalación de aplicaciones desarrolladas por terceros, uno de sus puntos fuertes es su estabilidad ya que además de ser de código cerrado, las aplicaciones creadas para este sistema operativo son revisadas minuciosamente antes de su publicación contando con una gran seguridad. [11]

##### **2.1.2. Android**

Es un sistema operativo móvil basado en Linux desarrollado por Google en el 2008, es de código abierto, permitiendo realizar modificaciones a los diferentes fabricantes de dispositivos móviles, aunque estos deben contar con una licencia para poder usarlo. Ofrece flexibilidad en cuanto a su personalización y además permite la instalación de aplicaciones desarrolladas por terceros de manera fácil, aunque en algunos casos su seguridad se ve vulnerada.[11]

## **2.2. Aplicaciones móviles**

Una aplicación móvil es un software diseñado específicamente para dispositivos móviles inteligentes. En la actualidad los dispositivos móviles han tenido un mayor crecimiento ya que son indispensables en el diario vivir de personas de diferentes edades, por lo que el desarrollo de aplicaciones móviles ha tenido un gran crecimiento y con ello se han creado diferentes tipos de aplicaciones móviles que se usan en diversos contextos.

### **2.2.1. Aplicaciones básicas**

Son aplicaciones que existen desde hace bastante tiempo, son simples y poco complejas, se basan en el envío y recepción de información puntual como mensajes de texto SMS o MMS, aunque son fáciles de usar y en su momento tuvieron una gran cantidad de usuarios y aceptación, se están convirtiendo en obsoletas debido a sus limitaciones y la aparición de nuevas aplicaciones. [12]

### **2.2.2. Aplicaciones Nativas**

Son aplicaciones móviles desarrolladas exclusivamente para una sola plataforma mediante un SDK específico, podemos encontrarlas en las tiendas oficiales de aplicaciones. Su punto fuerte es que no necesitan una conexión a internet para funcionar, utilizan tanto el hardware como el software para explotar todas las funcionalidades y capacidades de un dispositivo móvil brindando un gran rendimiento, seguridad y mejorando la experiencia del usuario. Sin embargo, el código no es reutilizable si queremos desarrollar la misma aplicación para otro sistema operativo móvil, viéndose afectado el tiempo y costo de desarrollo. [13]

### **2.2.3. Aplicaciones Híbridas**

Son aplicaciones móviles basadas en tecnologías web y que a su vez se combinan con el desarrollo nativo, se crean a través de frameworks dedicados al desarrollo híbrido por lo que se puede utilizar el mismo código fuente y reutilizarlo para crear una aplicación que funcione en

diferentes plataformas siendo una ventaja en el tiempo y costo de desarrollo. Usa un componente especial nativo y un navegador web embebido. Sin embargo, presenta algunos inconvenientes ya que, a diferencia de las nativas, pueden llegar a necesitar una conexión a internet para funcionar o simplemente no pueden aprovechar todo el potencial del hardware ocasionando que el rendimiento sea menor. [13]

#### 2.2.4. Aplicaciones Web

Son aplicaciones desarrolladas con tecnología web destinadas a correr mediante un navegador web sin la necesidad de instalarlas y adecuarlas a un dispositivo móvil. Sin embargo, este tipo de aplicaciones presentan varios inconvenientes, ya que al necesitar de una conexión estable a internet y un navegador web tiene limitaciones en cuanto al acceso de las capacidades que ofrece el hardware, afectando directamente al rendimiento y la experiencia del usuario. [14]

CRITERIO	BASICAS	NATIVAS	HÍBRIDAS	WEB
<b>Costo</b>	Bajo	Alto	Medio	Medio
<b>Rendimiento</b>	Bajo	Alto	Medio	Medio
<b>Tiempo de desarrollo</b>	Bajo	Alto	Medio	Medio
<b>Funcionalidad</b>	Baja	Alta	Media	Baja
<b>Código reutilizable</b>	Si	No	Si	Si
<b>Multiplataforma</b>	Si	No	Si	Si

Tabla 1 - Aplicaciones móviles.

Fuente: Autores.

De acuerdo a la Tabla 1, hemos decidido desarrollar una aplicación híbrida ya que se adapta perfectamente al tiempo y costo de desarrollo al permitirnos reutilizar código y ser multiplataforma.

### 2.3. Frontend

El frontend o interfaz de usuario es la parte dedicada al desarrollo que se presenta al usuario final, es decir se dedica a la estructura, colores, animaciones, efectos y fondos de una aplicación para que sea agradable a la interfaz del usuario.

Para el desarrollo del frontend existen muchas tecnologías como CSS, HTML y JavaScript, además, se puede utilizar frameworks o librerías dedicadas que ayudan al momento de desarrollar una interfaz de usuario.

### **2.3.1. Frameworks**

#### **2.3.1.1. Ionic**

Es un framework para el desarrollo de aplicaciones híbridas de código abierto muy utilizado ya que cuenta con un alto rendimiento y calidad, utiliza tecnologías web como: HTML, CSS y JavaScript que se integran fácilmente a frameworks como React, Angular y Vue además cuenta con varios plugin permitiéndonos crear interfaces de usuario flexibles desde una sola base de código compartida y actualizable a lo largo del tiempo. [15]

Como hemos mencionado anteriormente, Ionic utiliza tecnologías web por lo que no es necesario aprender un nuevo lenguaje de programación dándonos una cierta ventaja al utilizarlo. [16]

En cuanto a su estabilidad e integración, podemos decir que es uno de sus puntos fuertes ya que cuenta con una gran variedad de componentes y gracias a la implementación de Córdova e Ionic Native que hacen posible trabajar con componentes híbridos, además como ya hemos mencionado su integración con frameworks frontend como: React, Angular y Vue ayudan al desarrollo y diseño de interfaces de alto nivel de forma rápida. [16]

#### **2.3.1.2. React Native**

Es un framework desarrollado por Facebook para la creación de aplicaciones nativas tanto para Android como para IOS utilizando React, cuenta como una de las mejores bibliotecas de JavaScript basado en React, se enfoca en crear interfaces de usuarios iterativas y componentes visuales, siendo este un factor diferenciador ya que, en lugar de ejecutarse en navegador, se ejecuta directamente en la plataforma móvil nativa que utilicemos. [17]

Su aprendizaje React Native es fácil de entender y aprender ya que usa JavaScript y componentes HTML, aunque se necesita cierto conocimiento sobre desarrollo nativo al momento de incorporar código nativo en la base de código de React native. [18]

En cuanto a su estabilidad y su integración, React Native nos permite crear componentes nativos independientes de la plataforma que utilicemos para utilizar la misma base de código para múltiples plataformas mejorando el rendimiento de las aplicaciones a través de un control nativo, aunque, por otra parte, estas se limitan a la lógica del sistema que estemos utilizando. [18], [19]

### 2.3.1.3. Flutter

Es un framework de código abierto desarrollado por Google, para desarrollar aplicaciones tanto nativas como multiplataforma con una gran variedad de bibliotecas desde una sola base de código utilizado el lenguaje de programación Dart, además cuenta componentes flexibles y widgets para crear interfaces de usuario amigables y Hot Reload que nos permite visualizar los cambios mientras desarrollamos la aplicación. [20]

Flutter nos permite compilar una aplicación en nativo, ya sea Android o IOS, además su desarrollo es muy rápido, aunque una de sus desventajas es su lenguaje de programación, como hemos mencionado es Dart, y al no ser un lenguaje usado comúnmente en el desarrollo de aplicaciones puede presentar inconvenientes al momento de aprenderlo. [21]

CRITERIO	IONIC	REACT NATIVE	FLUTTER
Estabilidad e integración	5	4	4
Aprendizaje	5	4	3
Documentación	5	5	5
Código reutilizable	5	5	4
Lenguaje de programación	TypeScript, HTML, CSS	JavaScript, HTML.	Dart
<b>Total</b>	20	19	16

Tabla 2 - Frameworks de desarrollo móvil.

Fuente: Autores.

De acuerdo a los resultados de la Tabla 2 se decidió utilizar Ionic como framework para el desarrollo de la aplicación móvil, ya que se adapta a nuestras necesidades en cuanto al alcance de nuestro proyecto.

## **2.4. Backend**

El backend o programación del lado del servidor es la parte interna de una aplicación que no es visible para el usuario final ya que no cuenta con una interfaz gráfica. Esta parte se encarga de la seguridad y funcionalidad de una aplicación, además optimiza los recursos que se encuentran fuera del dispositivo del cliente.

Para el desarrollo del Backend podemos encontrar varios lenguajes de programación como JavaScript, PHP, .NET, ASP, Ruby, entre otros, que pueden ser usados mediante un entorno de ejecución como Node.js o Deno.

### **2.4.1. Node.js**

Es un entorno de ejecución de código abierto construido sobre el motor V8 disponible para Windows, Mac Os y Linux, orientado a eventos asíncronos y diseñado para desarrollar aplicaciones web o real time escalables utilizando JavaScript del lado del servidor y navegador, permitiendo el soporte de concurrencias altas, por lo que puede ejecutar varios procesos simultáneamente, siendo muy fácil de integrar con peticiones HTTP. [22]

Una de sus principales características es que podemos utilizar sus eventos asíncronos y no se utilizan hilos del sistema operativo ya que son ineficientes y difíciles de utilizar, por lo que si no hay trabajo por hacer Node.js permanece dormido hasta que exista una conexión al servidor para que se active mediante una llamada al servidor o callback, por ello es muy beneficioso utilizar Node.js. Además, acepta datos en formato JSON al igual que algunas bases de datos. [22]

Node.js tiene un instalador de paquetes llamado NPM que permite a los desarrolladores instalar, actualizar y administrar paquetes de software necesarios para el desarrollo de aplicaciones. Además, NPM ofrece un método sencillo para gestionar las versiones de los paquetes de software y sus dependencias.

### 2.4.2. Deno

Es un runtime para JavaScript y TypeScript, construido sobre el motor V8, Tokio y Rúst. Nos proporciona herramientas de desarrollo integradas como: Deno fmt, Deno lit y Deno test, además cuenta con valores predeterminados muy seguros, por lo que el acceso a red, archivos o entornos se encuentran deshabilitados a menos que los habilitemos. [23]

Deno se distribuye como un solo ejecutable que cuenta con varios módulos estándar previamente revisados y garantizados para trabajar, además cuenta con un servidor HTTP eficiente y sus eventos asíncronos devuelven promesas siendo sus APIS diferentes a las de Node.js [23]

CRITERIO	NODEJS	DENO
Escalable	SI	NO
Documentación	SI	SI
Asíncrono	SI	SI
Módulos	SI	NO
Código reutilizable	SI	NO
Formato JSON	SI	NO
<b>Total</b>	<b>6</b>	<b>2</b>

Tabla 3 - Lenguajes del servidor.

Fuente: Autores.

De acuerdo a la Tabla 3 se ha decidido utilizar Node.js para el desarrollo del backend.

La información que normalmente se maneja en una aplicación híbrida debe ser almacenada, se ha decidido utilizar una base de datos NoSQL por su escalabilidad, seguridad y automatización al momento de utilizarla.

### 2.4.3. Base de datos NoSQL

Las bases de datos NoSQL o también conocidas como no relacionales no necesitan identificadores para las diferentes relaciones que existen entre las diferentes tablas que se pueden encontrar en una base de datos, además no es necesario contar con un lenguaje estructurado para manejar los datos ya que la información puede ser organizada y almacenada en un documento documentos JSON o XML, según sea el caso. [24]

Las bases de datos NoSQL, se han convertido en una gran alternativa y tendencia al momento de desarrollar una aplicación híbrida debido a que brindan soluciones a problemas que pueden presentarse al momento de utilizar una base de datos SQL, como: soportar la conexión de millones de usuarios sin limitar su rendimiento, escalabilidad y manejo de grandes cantidades de datos. [25]

Dentro de las bases de datos NoSQL podemos encontrar algunos tipos, como:

- **Documentos:** La información se almacena en forma de un documento encapsulado en formato XML, JSON o BSON, la información almacenada en forma de documentos puede ser manejados de forma sencilla al momento de implementarlas o realizar consultas. [26]
- **Clave-Valor:** Son las más simples en cuanto a su uso e implementación, ya que la información se almacena en un conjunto de clave/valor, en donde la clave es un identificador el cual nos permite retornar un objeto de información denominado valor. [26]

- **Columna:** En este tipo la información se almacena en forma de columnas dejando de lado las filas, pueden ser híbridas en cuanto a base de datos relacionales y orientadas a columna. [26]
- **Grafos:** En este tipo la información se representa bajo el concepto de un grafo permitiendo almacenar la información como un nodo de un grafo y relaciones con otros nodos, se aplica la teoría de grafos para poder manejar la información. [26]

CRITERIO	MONGODB	FIREBASE
<b>Rendimiento</b>	Alto	Medio
<b>Seguridad</b>	Alta	Media
<b>Aplicaciones</b>	Para aplicaciones a gran escala	Para aplicaciones a mediana escala
<b>Documentación</b>	SI	SI
<b>Formato JSON</b>	SI	SI

Tabla 4 - Bases de datos NoSQL comparativa.

Fuente: Autores.

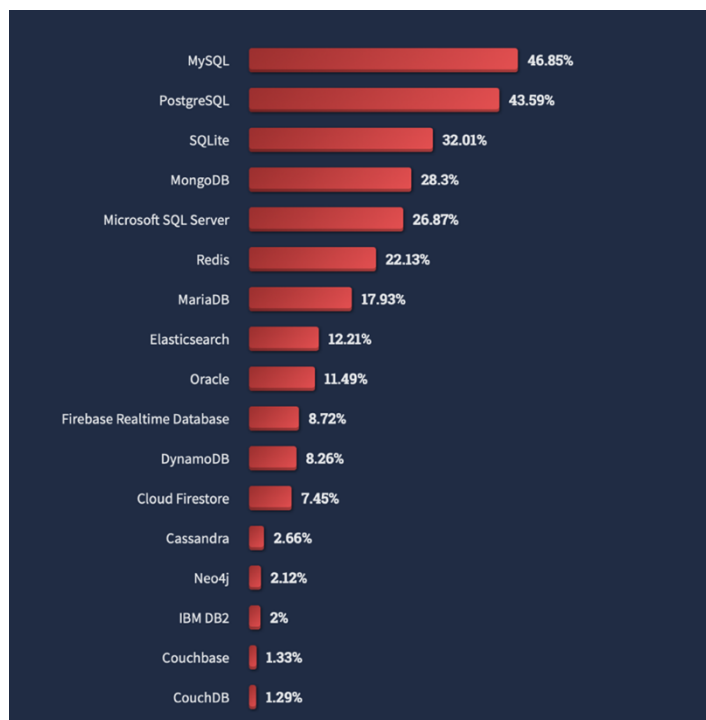


Ilustración 1 - Bases de datos más utilizadas en 2022.

Fuente: <https://survey.stackoverflow.co/2022/#databases>.

En la Tabla 4 - Bases de datos NoSQL comparativa.

Fuente: Autores.

se representa los resultados de la encuesta realizada por Stackoverflow en el 2022, en donde se evidencia que MongoDB se posiciona en el Top 3 de las bases de datos preferidas por desarrolladores.

De acuerdo a la Tabla 4 y la Tabla 4 - Bases de datos NoSQL comparativa.

Fuente: Autores.

se decidió utilizar MongoDB como base de datos NoSQL para el almacenamiento de la información que se manejara en la aplicación móvil híbridas.

### 2.4.3.1. MongoDB

Es una base de datos no relacional o NoSQL multiplataforma y de código abierto desarrollada en C++. MongoDB almacena los datos en forma de documentos similares a objetos JSON y

además cuenta con una alta disponibilidad ya que cuenta con un conjunto de replicas, que en otras palabras es un grupo de servidores con los mismos datos y su escalabilidad es horizontal. [27]

#### **2.4.3.2. MongoDB Atlas**

Es una base de datos en la nube creada y administrada por MongoDB, la cual permite la distribución de datos en varios nodos con el objetivo de ampliar la cobertura al momento de acceder a dichos datos. Atlas nos ofrece la posibilidad de simplificar la implementación, gestión, rendimiento y versatilidad en cuanto al manejo de datos garantizando la escalabilidad, disponibilidad y los controles de seguridad estandarizados para la privacidad de datos. [28]

### **2.5. Protocolo de Transferencia de Hiper Texto**

El protocolo de transferencia de hiper texto (HTTP) es un protocolo de comunicación para la transferencia de datos entre el servidor y el cliente. Funciona como un protocolo de solicitud – respuesta que permite al cliente realizar una solicitud al servidor y este responde con la información solicitada a través de métodos de peticiones HTTP que especifican la acción que debe realizarse en un recurso identificado por una URL (Uniform Resource Locator).

Los métodos de petición HTTP que comúnmente se utilizan son:

- **GET:** Este método petición nos permite recuperar un recurso al servidor.
- **POST:** Este método petición nos permite enviar un recurso al servidor.
- **PUT:** Este método petición nos permite actualizar un recurso del servidor
- **DELETE:** Este método petición nos permite eliminar un recurso del servidor.

### **2.6. API REST**

API REST (Interfaz de Programación de Aplicaciones de Transferencia de Estado Representacional) es una interfaz que permite interactuar a las aplicaciones entre si mediante

solicitudes HTTP, considerada altamente escalable y segura ya que cada solicitud que realiza el cliente es independiente por lo que no afecta a otras solicitudes.

Una API-REST utiliza los métodos de peticiones HTTP (get, post, put, delete) para acceder a los recursos del servidor mediante una URL, dichos recursos son representados en formato de intercambio de texto como XML o JSON proporcionando una gran flexibilidad en la forma en que los datos se pueden consumir y procesar.

Una API REST es muy escalable, ya que utiliza un protocolo de comunicación estándar (HTTP) y no requiere un estado de sesión. Esto permite que la API sea consumida por un gran número de clientes simultáneamente sin afectar su rendimiento desde diferentes plataformas, como aplicaciones web, móviles y de escritorio.

## 2.7.Arquitectura cliente - servidor

La arquitectura cliente-servidor se basa en un modelo donde existe el servidor, que es responsable de procesar las solicitudes enviadas por los clientes hacia una base de datos. Los clientes envían solicitudes HTTP al servidor, que las procesa y envía una respuesta de vuelta al cliente utilizando funciones predefinidas en el servidor. Este proceso es muy rápido y se realiza en fracciones de segundo, incluso si hay múltiples clientes enviando solicitudes simultáneamente mientras la aplicación está en funcionamiento.

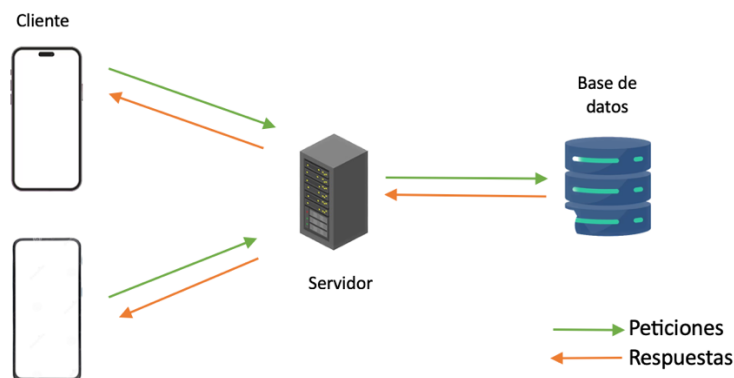


Ilustración 2 - Arquitectura cliente - servidor.

Fuente: Autor.

La arquitectura cliente-servidor permite escalar verticalmente o horizontalmente, lo que significa que se puede aumentar la capacidad del sistema a medida que crece la demanda. Debido a que los componentes de la arquitectura cliente-servidor se pueden desarrollar y mantener por separado, se pueden realizar actualizaciones y correcciones de manera más rápida y sencilla sin tener que detener todo el sistema.

## **2.8. Metodologías ágiles de desarrollo de software**

Las metodologías ágiles nacen por las necesidades que las metodologías tradicionales no podían cubrir en cuanto a la creación y desarrollo de software. Son flexibles y pueden ser modificadas cuando el proyecto de desarrollo lo requiera, además a través de la entrega constante de prototipos permite adaptarse a las necesidades y requisitos del cliente para entregar un producto final acorde a las necesidades del cliente.

Estas metodologías fomentan el trabajo en equipo bajo una comunicación constante, cada etapa del proyecto puede ser realizado por diferentes equipos multidisciplinarios según las necesidades del cliente y la evolución del proyecto permitiendo identificar errores y corregirlos inmediatamente ahorrándonos tiempo y costes de desarrollo.

CRITERIO	SCRUM	RUP
<b>Alcance</b>	Se evalúa al finalizar un Sprint	Se define antes o al comienzo del proyecto.
<b>Ciclo</b>	Sprint	Cuatro fases
<b>Entregables</b>	Software operativo	Plan de desarrollo, documentos de visión, alcance, lista de riesgos, Casos de uso
<b>Enfoque</b>	Iterativo	Iterativo
<b>Producto</b>	Mejoras rápidas	Proyecto de largo tiempo y alcance
<b>Roles</b>	SI	SI
<b>Conclusión</b>	Enfocado en el cliente adaptándose a las necesidades de acuerdo a revisiones constantes	Si las necesidades del cliente cambian se debe reformular todo desde el inicio

Tabla 5 - Comparativa Metodologías Ágiles.

Fuente: Autores.

De acuerdo a lo expuesto en la Tabla 5, se decidió utilizar la metodología SCRUM, ya que nos permite realizar cambios al finalizar cada Sprint, asimismo sus entregables constan de software operativo que permitirá realizar pruebas para comprobar su funcionamiento.

### 2.8.1. Scrum

Scrum es una metodología ágil de desarrollo de software que se enfoca en el trabajo en equipo, y aprendizaje constante, su estructura es flexible, cambiante y dinámica por lo que nos permite realizar cambios durante el proceso de desarrollo a través de Sprints. El objetivo de Scrum es planear y controlar proyectos con gran incertidumbre.

Scrum se centra en el cliente para ajustar las necesidades y cumplir con sus exigencias mediante la entrega de prototipos que sirven para corregir errores y alinear el producto con los cambios sugeridos y así obtener un producto final de calidad. La forma de trabajo de la metodología Scrum es la siguiente:

### 2.8.1.1. Equipos Scrum

Conformado por:

- **Product Owner:** Es la persona encargada de relacionarse con el cliente, comprende y entiende la visión del negocio y producto, además es el encargado de escribir y ordenar la opinión del cliente para crear el Product Backlog. [29]
- **Scrum Master:** Se encargada de revisar y comprobar que la metodología funciona correctamente, eliminando obstáculos que se presenten en el proceso, además se relaciona con el usuario. [29]
- **Equipo de desarrollo:** Grupo multifuncional conformado normalmente por 4 a 8 personas, cuentan con permisos en la toma de decisiones que ayuden a cumplir su objetivo, se encargan de crear el producto de acuerdo al Backlog. [29]

### 2.8.1.2. Artefactos de Scrum

- **Product Backlog:** Es una lista en la que consta las necesidades del cliente, así como los requerimientos del producto, los cuales pueden aparecer en las iteraciones sucesivas. [29]
- **Sprint Backlog:** Está conformada por tareas planificadas dentro de una lista ejecutadas en un Sprint y asignadas a personas con un tiempo estimado para finalizarlas. [29]
- **Incremento:** Es una parte desarrollada o añadida totalmente operativa en un Sprint. El cliente puede realizar cambios necesarios del proyecto basándose en los resultados que se obtienen. [29]

### 2.8.1.3. Procesos Scrum

Los proyectos desarrollados bajo esta metodología se realizan a través de fases iterativas que normalmente duran entre dos a cuatro semanas, en las cuales se realiza un feedback del producto y así obtener resultados en base a las iteraciones a fin de agregar valor al producto final. [30]

En la Ilustración 3 se puede observar cómo son utilizados los procesos Scrum.

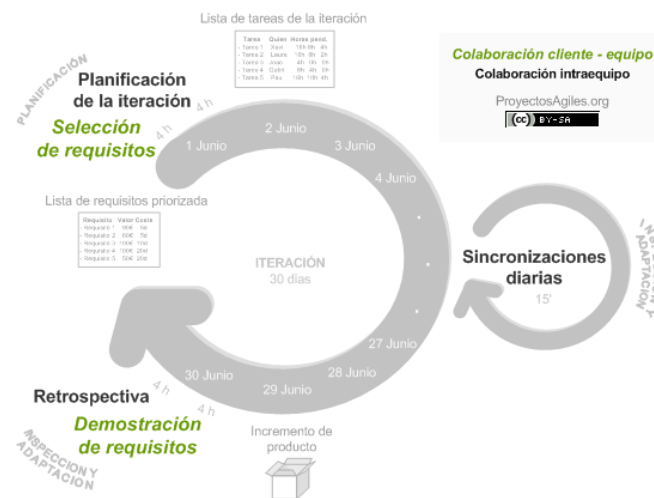


Ilustración 3 - Procesos Scrum.

Fuente: <https://proyectosagiles.org/que-es-scrum/>

#### 2.8.1.4. Eventos de Scrum

- **Sprint:** Un Sprint es un intervalo de tiempo fijo normalmente de un mes para crear un entregable funcional del producto final. Se desarrollan de acuerdo a las especificaciones y requisitos del cliente que constan en el Product Backlog. Es importante mencionar que la finalización de un Sprint es el inicio de otro. [31]
- **Planificación de Sprint:** Se establece el trabajo que se realizara por parte de todo el equipo involucrado en el desarrollo del producto. Generalmente se realiza una reunión en donde se discute lo que se entregara, como se realizara y se designaran las tareas a cumplirse a los miembros del equipo. [31]
- **Scrum Diario:** Son reuniones de 15 minutos de duración, en donde se inspecciona el avance del Sprint y el cumplimiento del objetivo. Las reuniones se efectúan para adaptar, analizar, recomendar y ajustar cambios con el objetivo terminar el Sprint previamente planificado. [31]
- **Revisión del Sprint o reunión demo:** Consiste en una reunión con el equipo y el cliente en donde se inspeccionan los resultados obtenidos, además se determinan

futuros cambios o adaptaciones para cumplir con el objetivo del producto y actualizar el Product Backlog y adaptarlo de ser necesario. [31]

## **2.9. Editores de código**

### **2.9.1. Visual Studio Code**

Es un editor de código fuente gratuito desarrollado por Microsoft, está disponible para Mac, Windows y Linux, tiene soporte integrado para JavaScript, TypeScript, entre otros. Es uno de los editores de código más ligeros y potentes que existen, además posee un ecosistema de varias extensiones creadas por Microsoft o terceros las cuales funcionan para otros lenguajes de programación (C++, java, C#, PHP, Python...), y de igual forma soporte para bibliotecas de diferentes frameworks como: Ionic, Angular, React o Vue.js y por último se integra fácilmente con Git. [32]

### **2.9.2. Atom Editor**

Creado por GitHub, el cual lo define como un editor de código hackeable, se encuentra disponible para Windows, IOS y Linux, cuenta un control integrado de versiones Git y varios plugin desarrollados en Node.js, además es compatible con varios lenguajes de programación mediante la instalación de paquetes a través de “instalador de paquetes incluido convirtiéndose en un IDE potente al momento de desarrollar software. [33]

### **2.9.3. Sublime Text**

Es un editor de código multiplataforma disponible en Windows, IOS y Linux nos permite programar de manera ligera y fluida, soporta varios lenguajes de programación y puede ser adaptada según las necesidades de los usuarios a través de complementos desarrollados por la comunidad convirtiéndolo en uno de los editores de texto más fáciles de utilizar. Cuenta con una versión de prueba de tiempo ilimitado y también una versión pagada de \$70.00. [34]

CRITERIO	VISUAL STUDIO CODE	ATOM EDITOR	SUBLIME TEXT
Open source	SI	SI	NO
Multiplataforma	SI	SI	SI
Documentación	SI	SI	SI
Compatibilidad con lenguajes de programación	SI	SI	SI
Terminal integrado	SI	NO	SI
Extensiones	SI	SI	SI
<b>Total</b>	<b>6</b>	<b>5</b>	<b>5</b>

Tabla 6 - Editores de código.

Fuente: Autores.

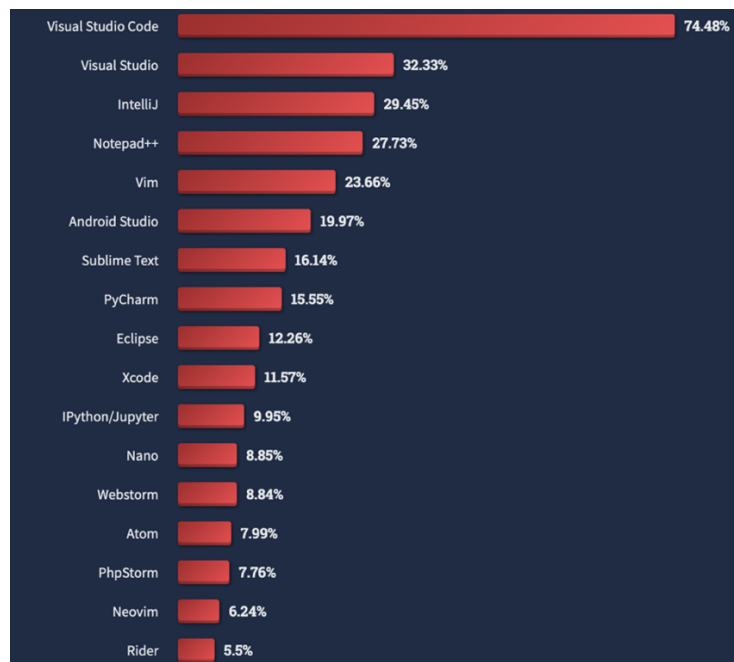


Ilustración 4 - Editores de código más utilizados en 2022.

Fuente: <https://survey.stackoverflow.co/2022/#section-most-popular-technologies-integrated-development-environment>

En la Tabla 6 - Editores de código.

Fuente: Autores.

, se muestran los resultados de la investigación hecha por StackOverFlow en 2022. [35] En donde se observa que Visual Studio Code ocupa el primer lugar de editores de código preferido por desarrolladores y aficionados con un total de el 74.48% mientras que Visual Studio ocupa

el segundo lugar con un 333% e IntelliJ ocupando el tercer lugar con un total de 29.45% respectivamente.

Basándonos en la Tabla 6 y en la Ilustración 3, se decidió utilizar Visual Studio Code para desarrollar el proyecto ya que nos proporciona varias librerías y extensiones que son útiles al momento de codificar.

## **2.10. Plataformas de pago en línea**

Una plataforma de pagos en línea es un servicio que permite a los usuarios realizar transacciones financieras de forma segura y rápida. Ofrecen un método sencillo para enviar y recibir pagos, realizar compras en línea y mover dinero a través de cuentas bancarias.

En comparación con los métodos de pago convencionales como cheques o efectivo, estas plataformas proporcionan una forma más rápida, segura y sencilla de hacer transacciones financieras, lo que ha transformado la industria financiera y ha facilitado el comercio en línea.

### **2.10.1. PayPal**

PayPal es una plataforma de pago en línea creada por la empresa PayPal Holdings, Inc. en 1998. Permite a los usuarios enviar y recibir pagos de forma rápida y segura a través de Internet convirtiéndose en uno de los servicios de pagos más conocidos en todo el mundo.

Para realizar transacciones, los usuarios de PayPal pueden añadir tarjetas de crédito o débito o conectar sus cuentas bancarias. La plataforma también permite a los usuarios realizar compras en línea en sitios web que tengan implementado PayPal como método de pago. Además, ofrece medidas de seguridad altas ya que cuenta con verificación y protección de datos mediante encriptación de los mismos.

De la misma manera, PayPal proporciona a los desarrolladores una API de pago que les facilita la incorporación de la plataforma en sus sitios web o aplicaciones móviles. Esto significa que

los desarrolladores pueden añadir rápida y fácilmente la funcionalidad de pago a sus proyectos sin tener que empezar de cero y desarrollar una solución de pago.

## 2.11. Servicio Municipal de Estacionamiento Rotativo Tarifado de Azogues

El Servicio Municipal de Estacionamiento Rotativo Tarifado de Azogues (SEMERTAZ) se encarga de controlar los espacios públicos de estacionamiento vehicular dentro de la ciudad de Azogues a fin de mejorar la movilidad, reducir el congestionamiento vehicular y contaminación dentro de la ciudad mediante un control constante y generación de multas a conductores que hagan mal uso de los espacios públicos que actualmente consta de 18 rutas divididas en la zona central de la ciudad.

RUTAS	DESCRIPCIÓN
RUTA UNO	Calle Oriente Calle Azuay Batalla de Ayacucho Juan Bautista Cordera Gral. Vintimilla
RUTA DOS	Calle Oriente Calle Guayas C. Emilio Abad C. Emilio Abad C. Bartolomé Serrano
RUTA TRES	C. Simón Bolívar C. Benigno Malo Gral. Vintimilla
RUTA CUATRO	Juan Bautista Cordero C. Matovelle C. Azuay C. benigno Malo C. Rafael M. García
RUTA CINCO	C. Antonio José de Sucre C. Guayas C. Emilio Abad C. Simón Bolívar C. 4 de noviembre
RUTA SEIS	C. José María Matovelle C. Bartolomeo Serrano C. Luis Cordero
RUTA SIETE	C. Fray Vicente Solano C. Antonio J. de Sucre C. Luis Cordero

RUTA OCHO	C. Fray Vicente Solano Av. 24 de Mayo C. Luis Cordero
RUTA NUEVE	C. José María Matovelle C. Simón Bolívar C. 10 de Agosto C. Cacique de Tenemaza
RUTA DIEZ	C. Luis Cordero C. Antonio José de Sucre C. 3 de Noviembre Av. 24 de Mayo
RUTA ONCE	C. Luis Cordero Av. 24 de Mayo C. Cacique Tenemaza C. 10 de Agosto
RUTA DOCE	C. 3 Noviembre C. Benigno Rivera C. Luis Cordero C. 10 de Agosto
RUTA TRECE	C. General Alberto Enríquez C. Benigno Rivera
RUTA CATORCE	C. Vicente Maldonado C. Cacique Tenemaza C. Emilio Abad Av. Aurelio Jaramillo
RUTA QUINCE	C. tres de noviembre C. José J. Olmedo Av. De la Virgen C. Luis Cordero
RUTA DIECISÉIS	C. Simón Bolívar Av. 24 de mayo Av. Aurelio Jaramillo C. Samuel Abad
RUTA DIECISIETE	C. General Enríquez C. tres de Noviembre C. Antonio J. de Sucre C. Antonio José de Sucre
RUTA DIECIOCHO	C. Tres de Noviembre C. 10 de Agosto C. General Enríquez

Tabla 7 - Rutas SEMERTAZ.

Fuente: Municipalidad de Azogues.

### 2.11.1. Objetivos de SEMERTAZ

- a) Disminuir el tráfico vehicular en el sistema vial ventral por el desplazamiento de vehículos cuyos conductores buscan plazas de estacionamiento. [36]

- b) Racionalizar y democratizar el uso de las vías públicas para el estacionamiento vehicular. [36]
- c) Elevar la rotación de las plazas de estacionamiento de las vías afectadas por SEMERTAZ a fin de disminuir los largos tiempos de estacionamiento e incrementar la oferta. [36]
- d) Controlar y eliminar la ocupación con vehículos de los espacios prohibidos para el estacionamiento y mejorar los niveles de seguridad y eficiencia del tráfico vehicular dentro del cantón. [36]
- e) Mejorar el nivel de accesibilidad al centro de la ciudad recuperando sus valores y el interés de la población por trasladarse hacia este.
- f) Facilitar el acceso vehicular y crear condiciones favorables para el estacionamiento temporal de los vehículos de ciudadanos que residen en la ciudad, pero que no cuentan con un garaje propio. [36]
- g) Mejorar las condiciones de accesibilidad al territorio en reverencia a las personas con discapacidad. [36]
- h) Mejorar las condiciones de movilidad y habitabilidad del cantón Azogues. [36]

### **2.11.2. Administración**

SEMERTAZ cuenta con un coordinador general, dos tabuladores de datos, uno de ellos también realiza las tareas de secretaría y 35 controladores de tráfico. A continuación, se detallan las actividades que realiza cada uno de ellos.

- **Coordinador general:** Se encarga de conocer, desarrollar, coordinar y ejecutar las acciones relacionadas al funcionamiento de SEMERTAZ. [36]
- **Tabuladores:** Se encargan de tabular los datos enviados por los controladores de tráfico en una hoja de Excel que consta de multas. [36]
- **Secretaria:** Se encarga de la atención al cliente. [36]

- **Controladores de tráfico:** Se encargan de recorrer y controlar los espacios públicos vehiculares en las 18 zonas delimitadas mediante la generación de multas o sanciones a conductores que hacen mal uso de dicho espacio.

### **2.11.3. Días y horario de funcionamiento**

SEMERTAZ atiende en los horarios de lunes a viernes de 07h30 a 19h00 y los sábados de 08h00 a 16h00, además el servicio no estará en funcionamiento los días domingos, feriados decretados por la ley y situaciones excepcionales. [36]

En cuanto al funcionamiento, el tiempo mínimo del uso del espacio de estacionamiento en una zona Azul es de media hora, mientras que en una zona Verde es de dos horas, además se establece que los conductores tendrán cinco minutos de tolerancia para colocar la tarjeta y marcar el tiempo que se va a utilizar el servicio. De igual manera existe una de tolerancia de diez minutos después de haberse cumplido el tiempo marcado en la tarjeta.

### **2.11.4. Tarifas**

SEMERTAZ se rige por las siguientes tarifas.

- a) 25 ctvs. (0,25 USD) por la ocupación de un espacio de estacionamiento permitido por media hora. [36]
- b) 75 dólares (75 USD) mensuales por la ocupación de un espacio de estacionamiento para vehículos de instituciones públicas, privadas comunitarias y personas particulares. [36]

### **2.11.5. Infracciones y sanciones**

Los conductores sancionados serán notificados mediante un adhesivo colocado en el vehículo por parte del controlador. Dicho adhesivo consta de: lugar y fecha, ubicación, marca, color, placa o código de identificación vehicular, el valor de la multa que debe pagar y la firma del controlador. El pago de las multas se realiza en la tesorería Municipal en un plazo de 7 días o se cobrara intereses de mora. [36]

Las infracciones que constan dentro de SEMERTAZ son:

<b>INFRACCIÓN</b>	<b>VALOR</b>
Vehículo sin tarjeta	9 USD
Tarjeta no legible	9 USD
Tarjeta no vigente	9 USD
Mal uso de tarjeta	9 USD
Tarjeta alterada	9 USD
Tiempo excedido	9 USD
Ventas en la vía pública	9 USD
Tiempo excedido con carnet de discapacidad	9 USD
Estacionado fuera del lugar para motos	9 USD
Vehículo en pesado	18 USD
Vehículo en paso cebra	15 USD
Vehículo en línea amarilla	15 USD
Vehículo en boca calle	15 USD
Vehículo en espacio de discapacitados	15 USD
Vehículo reservado para particulares	15 USD
Vehículo en parada de bus	15 USD
Vehículo frente a hidrante	15 USD
Vehículo en entrada a garaje	15 USD
Vehículo sobre vereda	15 USD
Vehículo doble fila	15 USD
Vehículo en espacio de motocicletas	15 USD

Tabla 8 - Valor de las sanciones.

Fuente: Ordenanza Municipal 2021.

## CAPÍTULO 3

### 3. Desarrollo de la metodología

#### 3.1. Análisis de requerimientos

Es importante mencionar que al ser un prototipo de aplicación móvil híbridas a través de metodologías ágiles el acceso a la información fue restringida por políticas internas de SEMERTAZ, por lo que se realizó una revisión de la ordenanza municipal “Sustitutiva que norma el servicio municipal de estacionamiento rotativo tarifado en las vías públicas” [36] publicada en el 2021 por la municipalidad de Azogues en su página web oficial y la encuesta realizada para identificar los roles e involucrados en los procesos y procedimientos que se realizan en SEMERTAZ a fin de comprender los requerimientos funcionales de la aplicación a desarrollarse.

Requerimientos Funcionales
Almacenar la información asociada a SEMERTAZ
Mantenimiento de infracciones de tránsito de SEMERTAZ
Mantenimiento del personal que labora en SEMERTAZ
Crear el login para el acceso del personal de SEMERTAZ
Crear Interfaz de consulta de Infracciones registradas en SEMERTAZ
Implementar un Botón de pago para infracciones registradas de SEMERTAZ

Tabla 9 - Requerimientos Funcionales

Fuente: Autores.

#### 3.2. Implementación de la metodología

Con la información obtenida en el análisis de requerimientos funcionales se elaboró el Product Backlog e historias de usuario que explican la funcionalidad del prototipo de aplicación; es importante mencionar que cada historia de usuario consta de:

- **Código:** Es el identificador de cada historia.
- **Nombre:** Detalla brevemente la historia de usuario

- **Importancia:** Es el orden de prioridad e importancia que va a tener la historia de usuario.
- **Descripción.** Es una explicación concisa que quiere el usuario de acuerdo a sus necesidades.
- **Estimación:** Es el tiempo en el que la historia de usuario debe estar lista.

En la Tabla 10 y Tabla 11 se puede observar el modelo de cómo se realizó cada historia de usuario.

Historia de usuario	
<b>Código</b>	05
<b>Nombre</b>	Crear la interfaz de búsqueda de infracciones
<b>Descripción</b>	Como usuario externo quiero consultar infracciones de tránsito de SEMETAZ
<b>Importancia</b>	5
<b>Estimación</b>	2

Tabla 10 - Ejemplo 1 de Historia de Usuario.

Fuente: Autores.

Historia de usuario	
<b>Código</b>	06
<b>Nombre</b>	Implementar el botón de pagos
<b>Descripción</b>	Como usuario quiero realizar pagos online de infracciones registradas en SEMERTAZ
<b>Importancia</b>	5
<b>Estimación</b>	2

Tabla 11 - Ejemplo 2 de Historia de Usuario.

Fuente: Autores.

En la Tabla 12 se observa el Product Backlog, en donde se enlista las historias del usuario planteadas mediante la investigación realizada.

<b>Id</b>	<b>Nombre de la Historia de Usuario</b>	<b>Importancia</b>	<b>Estimación</b>	<b>Comentario</b>
1	Creación de la base de datos	5	2	Crear la base de datos para almacenar la información asociada a SEMERTAZ
2	Mantenimiento del personal	5	2	Registrar el personal que trabaja en SEMERTAZ
3	Mantenimiento de infracciones	5	2	Registrar, consultar, editar y eliminar las infracciones registradas
4	Crear el login	5	2	Crear la interfaz de inicio de sesión para el personal
5	Crear la interfaz de búsqueda de infracciones	4	1	Crear la interfaz de búsqueda de infracciones adeudadas
6	Implementar el botón de pagos	4	1	Implementar un botón de pagos para infracciones adeudadas

Tabla 12- Product Backlog.

Fuente: Autores.

### 3.3. Definición de los Sprints

Terminado el Product Backlog, se agruparon las historias de usuario de acuerdo a su nivel de estimación e importancia y de tal modo que se puedan cumplir en el espacio de tiempo determinado.

En la estimación de horas de trabajo de los Sprints se asignó un 90% para el trabajo optimo y un 10% se asignó para retrasos o distracciones que se presenten en el cumplimiento de cada Sprint.

La estimación de tiempo se clasifico de entre dos y cuatro semanas, mientras que la importancia se clasifico de la siguiente manera:

5 = Muy Alta.

4 = Alta.

3 = Media.

2 = Baja.

1 = Muy Baja.

En la Tabla 13 se observan las historias de usuario de acuerdo al primer Sprint.

<b>Id</b>	<b>Nombre</b>	<b>Importancia</b>	<b>Estimación</b>	<b>Comentario</b>
<b>1</b>	Creación de la base de datos	5	2	Crear la base de datos para almacenar la información asociada a SEMERTAZ
<b>2</b>	Mantenimiento del personal	5	2	Registrar el personal que trabaja en SEMERTAZ

Tabla 13 - Sprint 1.

Fuente: Autores.

En la Tabla 14 se observa el segundo Sprint con las historias de usuario planificadas.

<b>Id</b>	<b>Nombre</b>	<b>Importancia</b>	<b>Estimación</b>	<b>Comentario</b>
<b>3</b>	Mantenimiento de infracciones	5	2	Registrar, Consultar, editar y eliminar las infracciones provocadas por clientes externos
<b>4</b>	Crear el login	5	2	Crear la interfaz de inicio de sesión para el personal

Tabla 14 - Sprint 2.

Fuente: Autores.

En la Tabla 15 se observa el tercer Sprint con las historias de usuario planificadas.

<b>Id</b>	<b>Nombre</b>	<b>Importancia</b>	<b>Estimación</b>	<b>Comentario</b>
5	Crear la interfaz de búsqueda de infracciones	4	1	Crear la interfaz de búsqueda de infracciones.
6	Implementar el botón de pagos	4	1	Implementar un botón de pagos para infracciones adeudadas

Tabla 15 - Sprint 3.

Fuente: Autores.

### 3.4. Planificación de los Sprints

En cada Sprint se organizan revisiones con el objetivo de validar los avances para generar retroalimentación o simplemente seguir con las acciones para realizar mejoras que aportan a entregas futuras del proyecto.

Para evidenciar los avances de cada Sprint se elabora un Taskboard, en el cual se observan los avances de cada Sprint y nos muestra si una tarea está en desarrollo o ya ha Terminado, además se puede implementar un Burndown chart para comprobar la velocidad con la que se está desarrollando el proyecto y si se está cumpliendo con la estimación de tiempo acordado.

A continuación, en la Tabla 16 se observan las tareas de desarrollo planificadas para cumplir con el primer Sprint.

<b>Sprint 1.</b>	
<b>Autores:</b> Diego Fajardo y Marlon Bermejo	
<b>Inicio</b>	06 de diciembre del 2022
<b>Fin</b>	06 de enero del 2023
<b>Tiempo de desarrollo previsto</b>	4 semanas
<b>Tareas a desarrollarse</b>	Creación de la base de datos
	Mantenimiento del personal

Tabla 16 - Planificación del primer Sprint.

Fuente: Autores.

La Tabla 17 muestra las tareas de desarrollo planificadas para cumplir con el segundo Sprint.

<b>Sprint 2</b>	
<b>Autores:</b> Diego Fajardo y Marlon Bermejo	
<b>Inicio</b>	9 de Enero del 2023
<b>Fin</b>	9 de Febrero del 2023
<b>Tiempo de desarrollo previsto</b>	4 semanas
<b>Tareas a desarrollarse</b>	Mantenimiento de infracciones
	Crear el login

Tabla 17 - Planificación del segundo Sprint.

Fuente: Autores.

La Tabla 18 muestra tareas de desarrollo planificadas para cumplir con el tercer Sprint.

<b>Sprint 3</b>	
<b>Autores:</b> Diego Fajardo y Marlon Bermejo	
<b>Inicio</b>	12 de febrero del 2023
<b>Fin</b>	06 de Marzo del 2023
<b>Tiempo de desarrollo previsto</b>	3 semanas
<b>Tareas a desarrollarse</b>	Crear de la interfaz de búsqueda de infracciones
	Implementar del botón de pagos

Tabla 18 - Planificación del tercer Sprint.

Fuente: Autores.

## Taskboard inicial

La Tabla 19 muestra el Taskboard establecido con las tareas de desarrollo, así como su estado (Pendiente, En desarrollo y Terminado).

N. Sprints	Inicio: 06/ Diciembre /2022	Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Fin: 06/ Marzo /2023	Pendiente	En desarrollo	Terminado
Sprint 1	Creación de la base de datos	√		
	Mantenimiento del personal	√		
Sprint 2	Mantenimiento de infracciones	√		
	Crear el login	√		
Sprint 3	Crear la interfaz de búsqueda de infracciones	√		
	Implementar el botón de pagos	√		

Tabla 19 - Taskboard inicial.

Fuente: Autores.

## Burndown chart inicial

La Ilustración 5 muestra el Burndown chart de las tareas planificadas a desarrollarse.

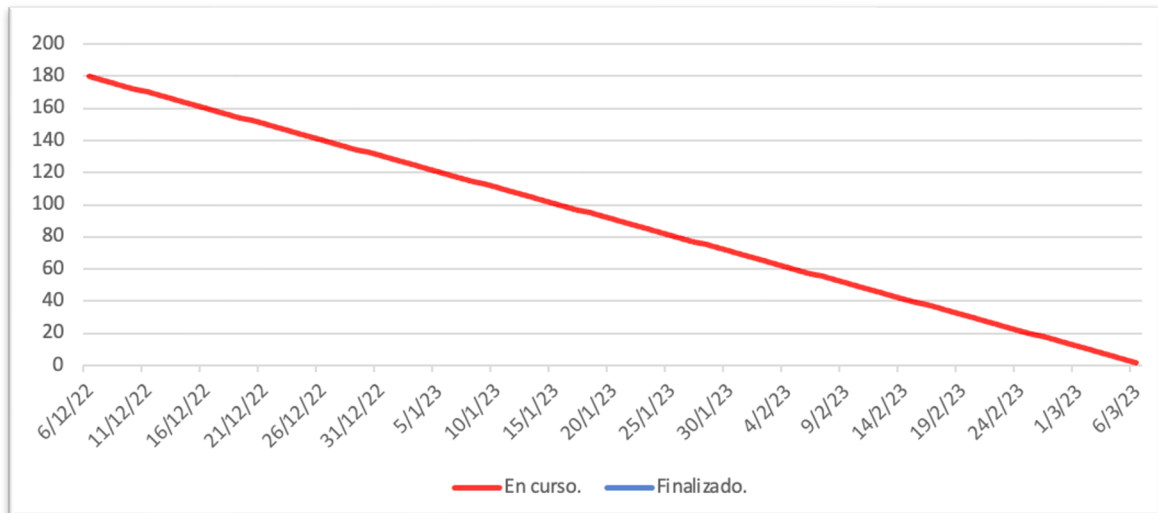


Ilustración 5 - Burndown chart inicial.

## CAPÍTULO 4

### 4. Resultados

#### 4.1. Resultados del Sprint 1

- **Creación de la base de datos.**
- **Mantenimiento del personal.**

La Tabla 20 muestra el Taskboard del primer Sprint, en donde la tarea de desarrollo “Creación de la base de datos” se encuentra en desarrollo.

N. Sprints	Inicio: 06/ Diciembre /2022 Fin: 06 / Enero /2023	Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Tareas a desarrollar	Pendiente	En desarrollo	Terminado
Sprint 1	Creación de la base de datos		√	
	Mantenimiento del personal	√		
Sprint 2	Mantenimiento de infracciones	√		
	Crear el login	√		
Sprint 3	Crear la interfaz de búsqueda de infracciones	√		
	Implementar el botón de pagos	√		

Tabla 20 - Taskboard del primer Sprint.

Fuente: Autores.

La Ilustración 6 muestra el Burndown chart correspondiente al primer Sprint con tareas pendientes y en desarrollo según el tiempo de desarrollo planificado, siendo el eje “X” el tiempo

de desarrollo y el eje “Y” las tareas que se realizan, tomando como línea de referencia al color rojo que representa la conexión entre el eje “Y” y el eje “X” de acuerdo al tiempo planificado y la línea color azul se actualizara con cada tarea que se termine.

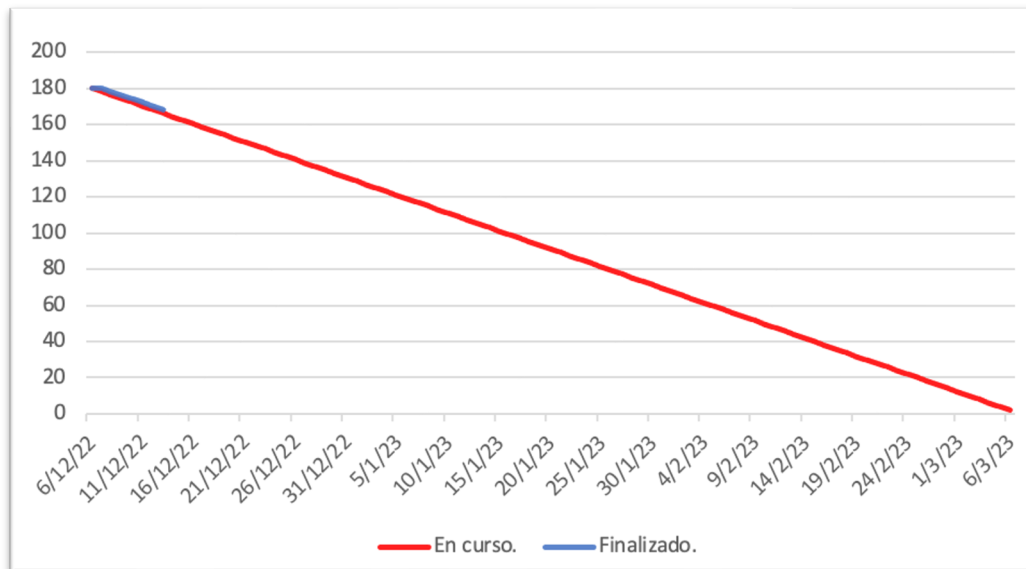


Ilustración 6 - Burndown chart del primer Sprint.

Fuente: Autores.

### Creación de la base de datos.

La Ilustración 7 muestra la base de datos no relacional creada en MongoDB Atlas.

#### Empleados.

```
_id: ObjectId('6406676617ea3537713f7bf1')
Cedula: "0987654321"
Nombre: "Marlon Bryant"
Apellido: "Bermejo Lema"
Cargo: " Coordinador General "
Direccion: "Bilian"
Telefono: "0987783223"
Correo: "marlonbl@gmail.com"
Contraseña: null
```

#### Infracciones.

```
_id: ObjectId('6406683717ea3537713f7bf3')
PlacaVehicular: "ABX-1234"
Sancion: "Tarjeta no legible"
Valor: 9
Pagado: false
Marca: "Chevrolet"
ModeloVehiculo: "Aveo"
TipoVehiculo: "Auto"
ColorVehiculo: " Azul "
Ruta: " Ruta 01 "
createdAt: 2023-03-06T22:24:55.538+00:00
updatedAt: 2023-03-06T22:24:55.538+00:00
```

#### Ilustración 7 - Base de datos.

Fuente: Autores.

La Tabla 21 muestra el Taskboard del primer Sprint con la tarea “Creación de la base de datos” terminada, mientras que la tarea “Mantenimiento de personal” se encuentra en desarrollo.

N. Sprints	Inicio: 06/ Diciembre /2022	Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Fin: 06 / Enero /2023			
	Tareas a desarrollar	Pendiente	En desarrollo	Terminado
Sprint 1	Creación de la base de datos			√
	Mantenimiento de personal		√	
Sprint 2	Mantenimiento de infracciones	√		
	Crear el login	√		
Sprint 3	Crear la interfaz de búsqueda de infracciones	√		
	Implementar el botón de pagos	√		

Tabla 21 - Taskboard del primer Sprint.

Fuente: Autores.

La Ilustración 8 muestra el Burndown chart del primer Sprint con tareas terminadas y en desarrollo según el tiempo de desarrollo planificado.



Ilustración 8 - Burndown chart del primer Sprint.

Fuente: Autores.

### Mantenimiento del personal.

La Ilustración 9 e Ilustración 10 muestra la interfaz del formulario que permite registrar un nuevo empleado.

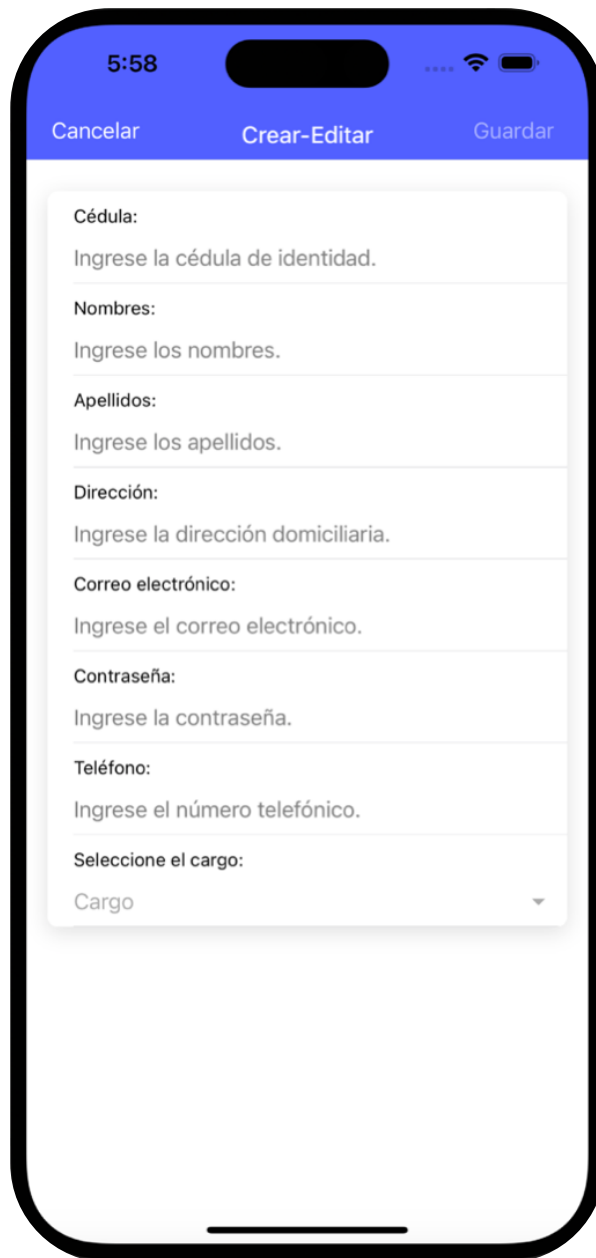


Ilustración 9 - Interfaz de registro de personal en IOS.

Fuente: Autores.



Ilustración 10 - Interfaz de registro de personal en Android.

Fuente: Autores.

En la Ilustración 11 e Ilustración 12 se observa La interfaz del personal registrados con botones de editar y eliminar.

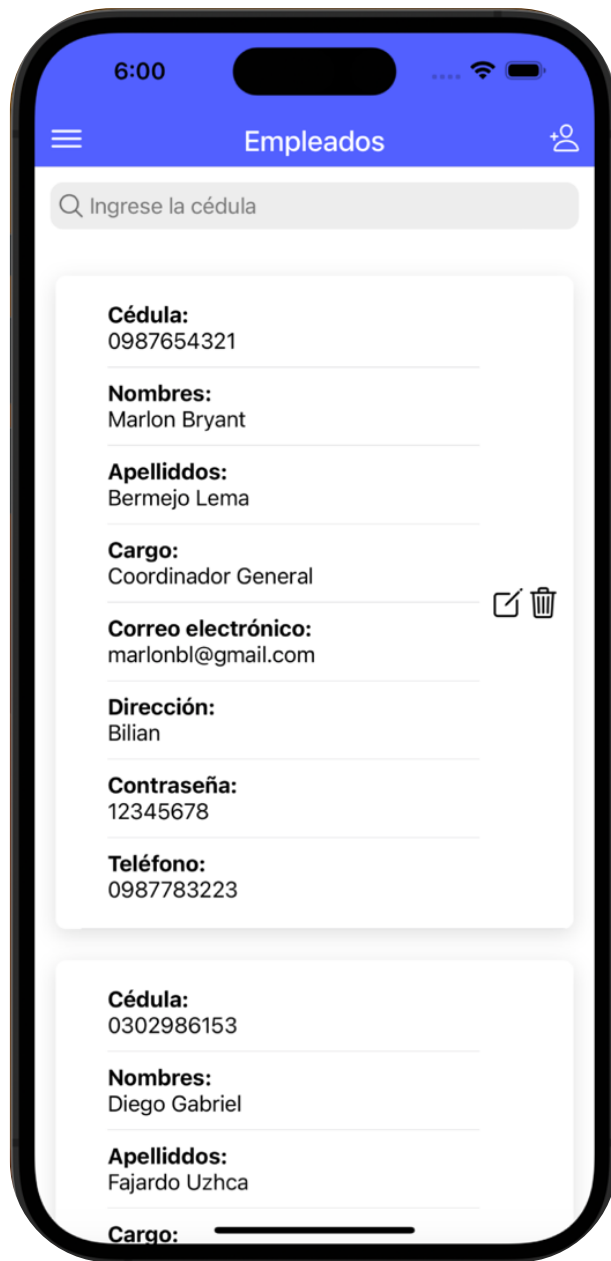


Ilustración 11 - Interfaz de personal registrados en iOS.

Fuente: Autores.

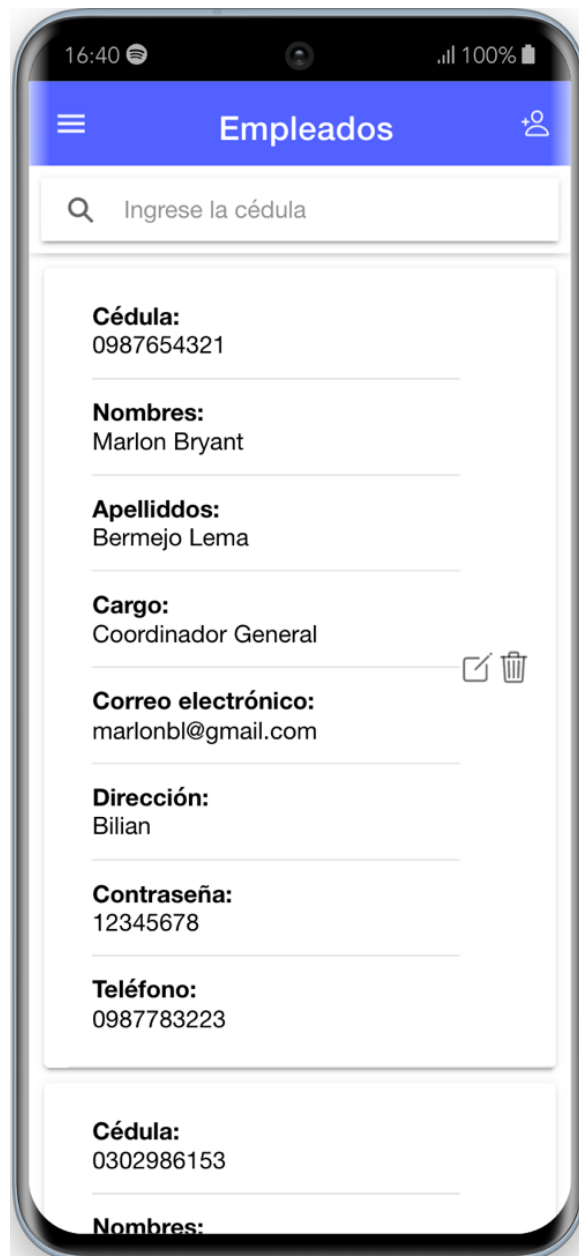


Ilustración 12 - Interfaz de personal registrados en Android.

Fuente: Autores.

La Tabla 22 muestra el Taskboard del primer Sprint, con las tareas planificadas terminadas.

N. Sprints	Inicio: 06/ Diciembre /2022	Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Fin: 06/ Enero /2023			
	Tareas a desarrollar	Pendiente	En desarrollo	Terminado
Sprint 1	Creación de la base de datos			√
	Mantenimiento del personal			√
Sprint 2	Mantenimiento de infracciones	√		
	Crear el login	√		
Sprint 3	Crear la interfaz de búsqueda de infracciones	√		
	Implementar botón de pagos	√		

Tabla 22 - Taskboard del primer Sprint.

Fuente: Autores.

La Ilustración 13 muestra el Burndown chart del primer Sprint, el cual se encuentra terminado de acuerdo al tiempo de desarrollo planificado.



Ilustración 13 - Burndown chart del primer Sprint.

Fuente: Autores.

## 4.2. Resultados del Sprint 2

- **Mantenimiento de infracciones.**
- **Crear el login.**

La Tabla 23 muestra el Taskboard del segundo Sprint, con la tarea “Mantenimiento de infracciones” en desarrollo.

N. Sprints	Inicio: 09/ Enero /2023	Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Fin: 09/ Febrero /2023	Pendiente	En desarrollo	Terminado
Sprint 1	Creación de la base de datos			√
	Mantenimiento del personal			√
Sprint 2	Mantenimiento de infracciones		√	
	Crear el login	√		
Sprint 3	Crear la interfaz de búsqueda de infracciones	√		
	Implementar el botón de pagos	√		

Tabla 23 - Taskboard del segundo Sprint.

Fuente: Autores.

La Ilustración 14 muestra el Burndown chart del segundo Sprint, con tareas pendientes y En desarrollo según el tiempo de desarrollo planificado.

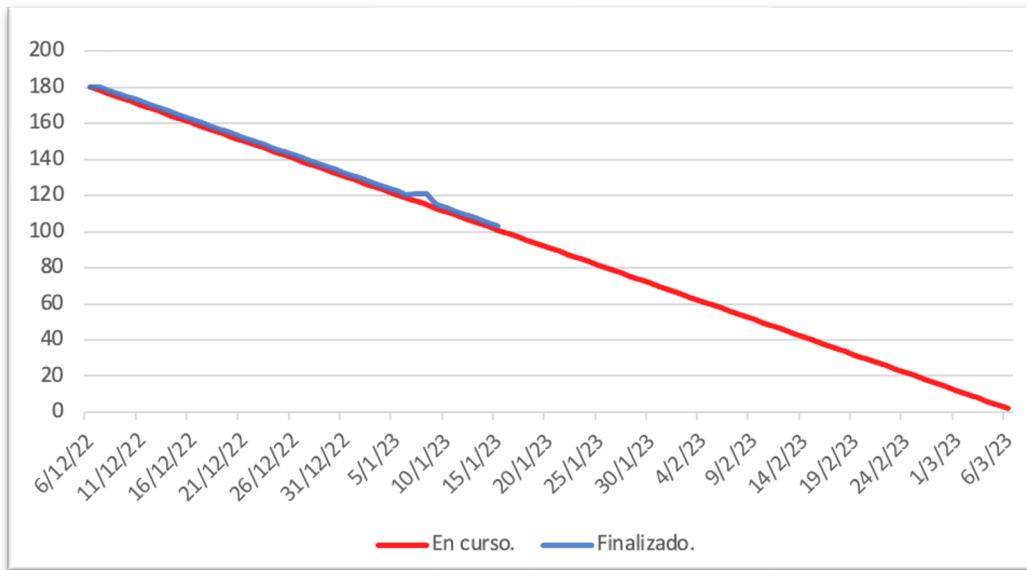


Ilustración 14 - Burndown chart del segundo Sprint.

Fuente: Autores.

### Mantenimiento de infracciones.

La Ilustración 15 e Ilustración 16 muestran la interfaz con el formulario y los campos necesarios para registrar una infracción.

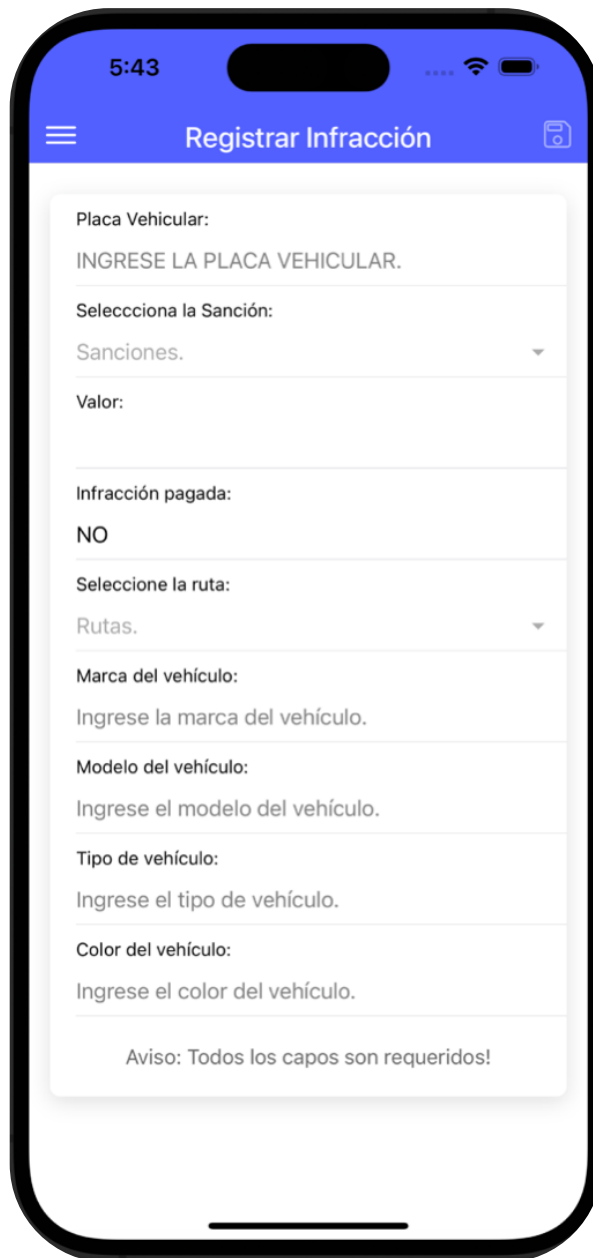


Ilustración 15 - Interfaz de registro de infracciones en IOS.

Fuente: Autores.

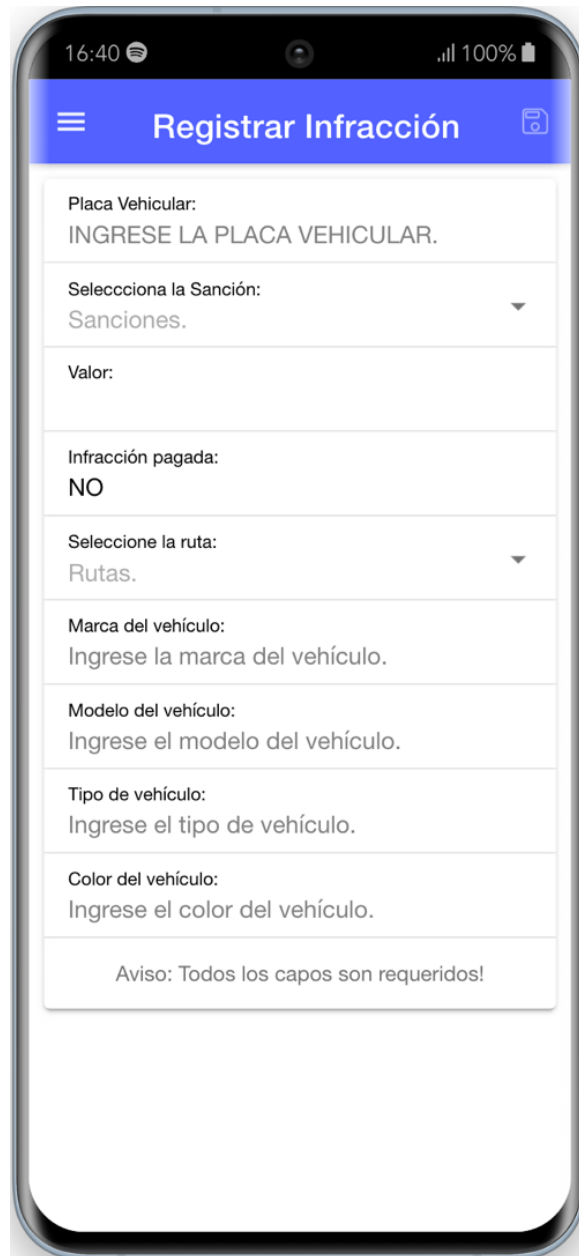


Ilustración 16 - Interfaz de registro de infracciones en Android.

Fuente: Autores.

En la Ilustración 17 e Ilustración 18 se observa la interfaz con infracciones registradas, botones para editar y borrar.

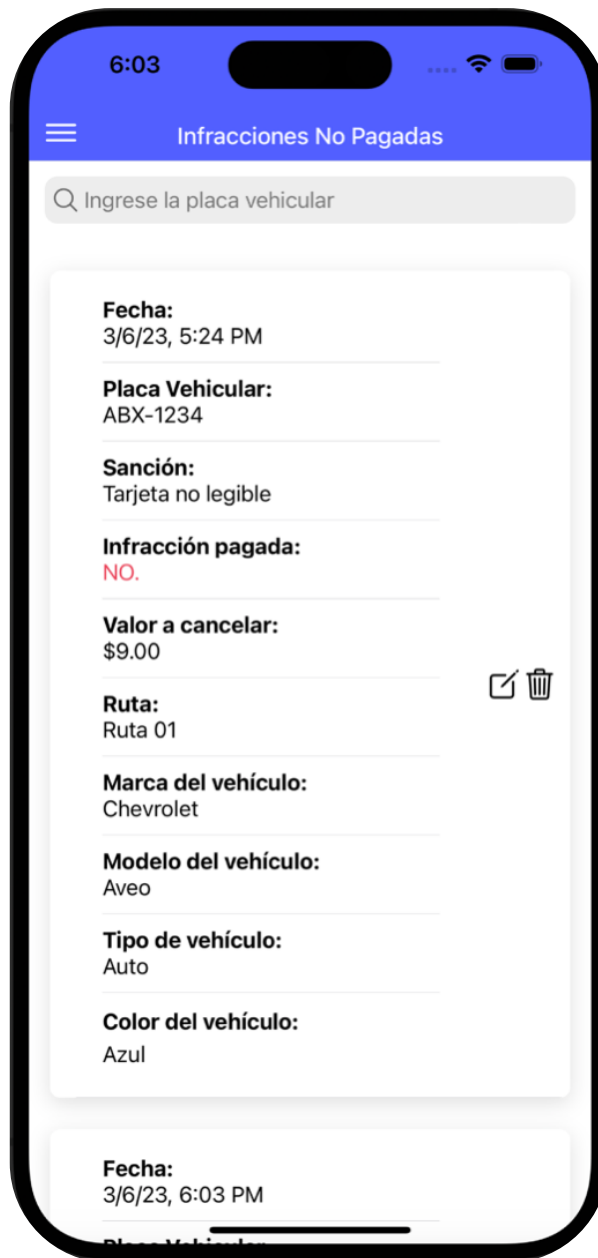


Ilustración 17 - Interfaz de infracciones registradas en IOS.

Fuente: Autores.

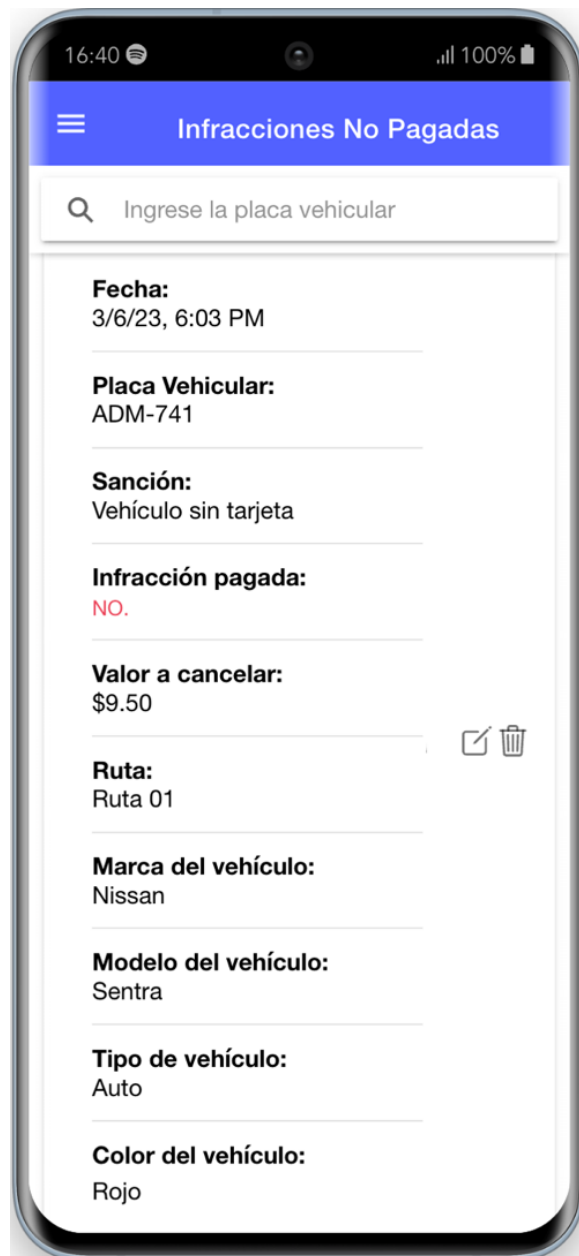


Ilustración 18 - Interfaz de infracciones registradas en Android.

Fuente: Autores.

En la Ilustración 19 e Ilustración 20 se observa la interfaz para editar los datos de una infracción previamente registrada.

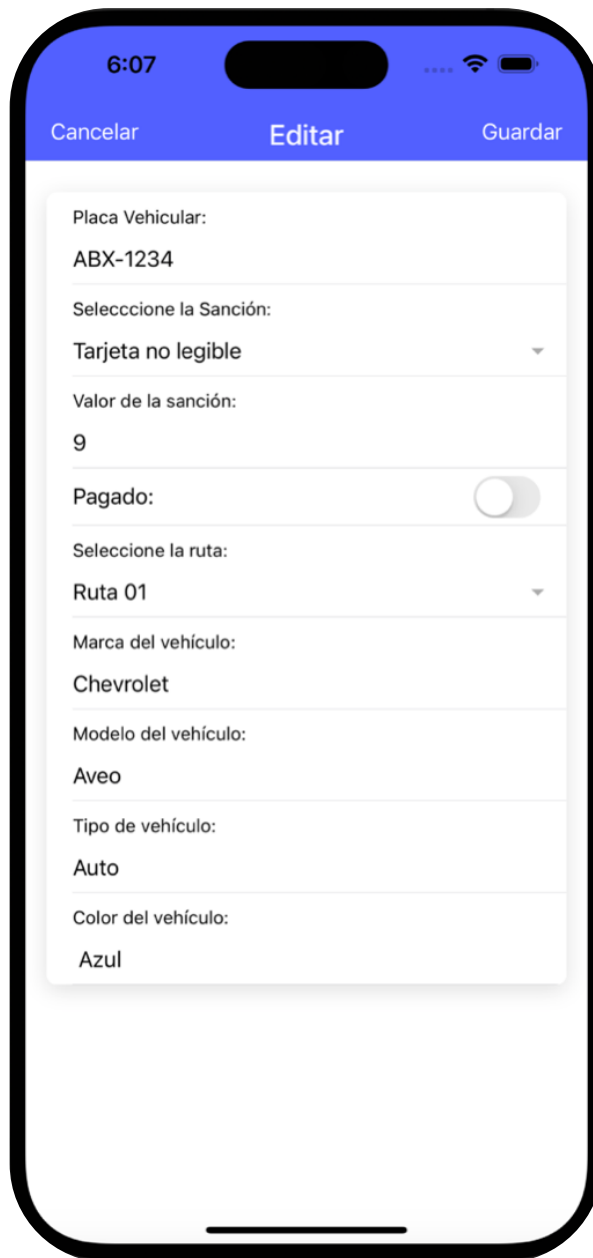


Ilustración 19 - Interfaz para editar infracciones en IOS.

Fuente: Autores.

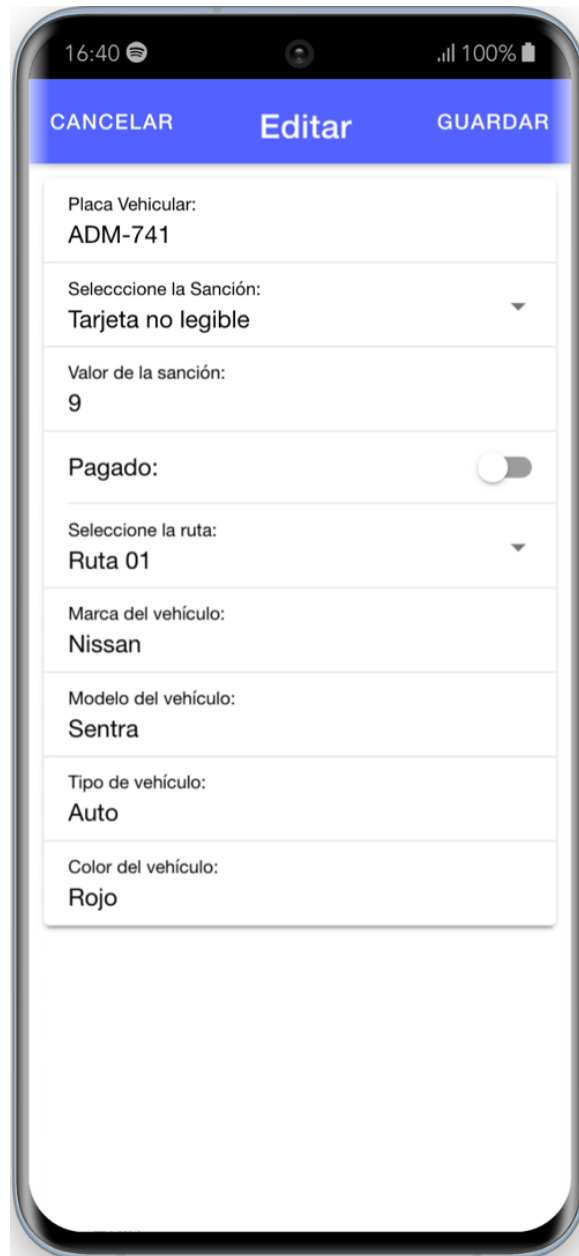


Ilustración 20 - Interfaz para editar infracciones en Android.

Fuente: Autores.

En la Ilustración 21 e Ilustración 22 se observa la interfaz con una infracción pagada.

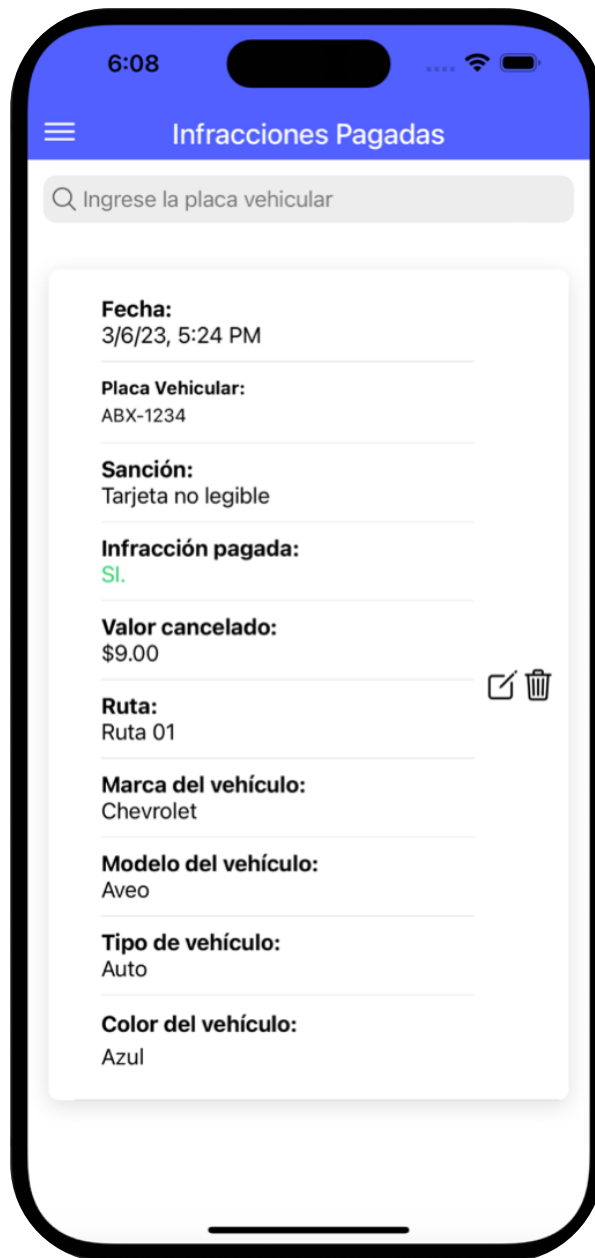


Ilustración 21 - Interfaz de infracciones pagadas en IOS.

Fuente: Autores.

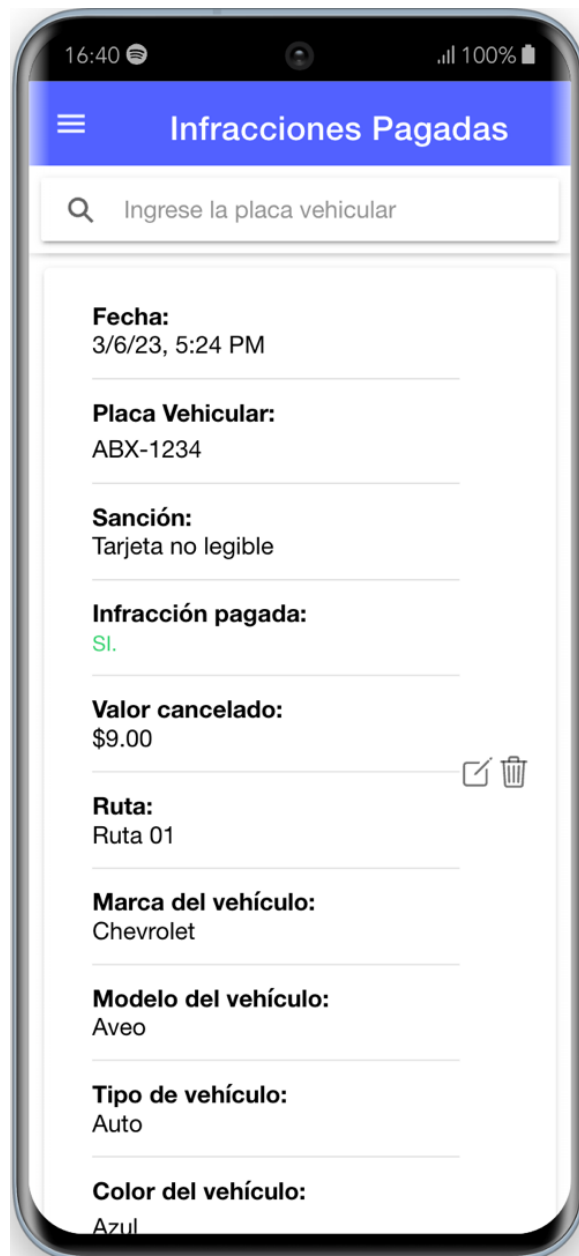


Ilustración 22 - Interfaz de infracciones pagadas en Android.

Fuente: Autores.

La Tabla 24 indica el Taskboard del segundo Sprint en donde la tarea “Mantenimiento de infracciones” está terminada, mientras que la “Crear el login” se encuentra en desarrollo.

N. Sprints	Inicio: 09/ Enero /2023	Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Fin: 09/ Febrero /2023	Pendiente	En desarrollo	Terminado
Sprint 1	Creación de la base de datos			√
	Mantenimiento del personal			√
Sprint 2	Mantenimiento de infracciones			√
	Crear el login		√	
Sprint 3	Crear la interfaz de búsqueda de infracciones	√		
	Implementar el botón de pagos	√		

Tabla 24 - Taskboard del segundo Sprint.

Fuente: Autores.

La Ilustración 23 muestra el progreso del primer Sprint, con tareas finalizadas y En desarrollo según el tiempo de desarrollo planificado.

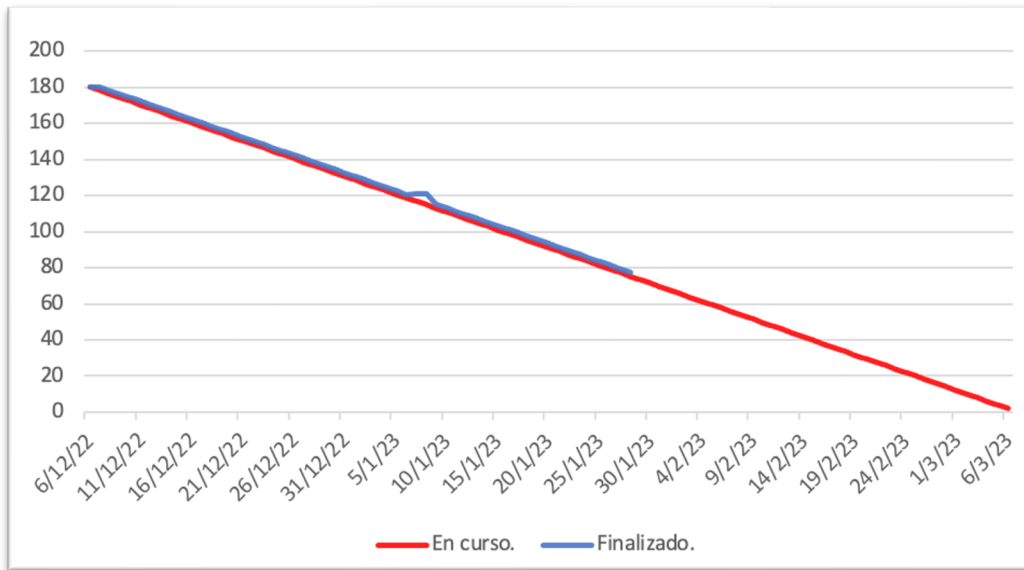


Ilustración 23 - Burndown chart del Sprint 2.

Fuente: Autores.

### Crear el login.

La Ilustración 24 e Ilustración 25 muestran la interfaz del formulario que permite iniciar sesión en la aplicación.

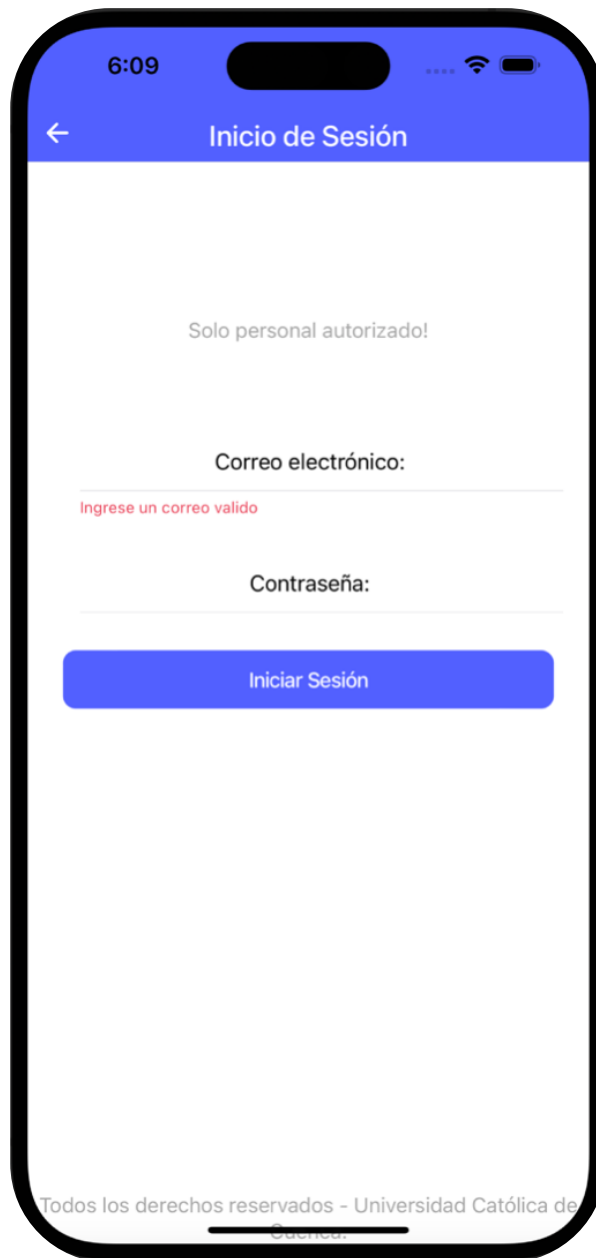


Ilustración 24 - Interfaz de login en IOS.

Fuente: Autores.



Ilustración 25 - Interfaz de login en Android.

Fuente: Autores.

La Ilustración 26 e Ilustración 27 muestra la interfaz de bienvenida al iniciar sesión.



Ilustración 26 - Interfaz de bienvenida en IOS.

Fuente: Autores.



Ilustración 27 - Interfaz de bienvenida en Android.

Fuente: Autores.

La Tabla 25 muestra el Taskboard del segundo Sprint, con las tareas planificadas terminadas de acuerdo al tiempo de desarrollo planificado.

N. Sprints	Inicio: 09/ Enero /2023	Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Fin: 09/ Febrero /2023			
	Tareas a desarrollar	Pendiente	En desarrollo	Terminado
Sprint 1	Creación de la base de datos			√
	Mantenimiento del personal			√
Sprint 2	Mantenimiento de infracciones			√
	Crear el login			√
Sprint 3	Crear la interfaz de búsqueda de infracciones	√		
	Implementar el botón de pagos	√		

Tabla 25 - Taskboard del segundo Sprint.

Fuente: Autores.

La Ilustración 28 muestra el Burndown chart del segundo Sprint, el cual se encuentra Terminado de acuerdo al tiempo de desarrollo planificado.

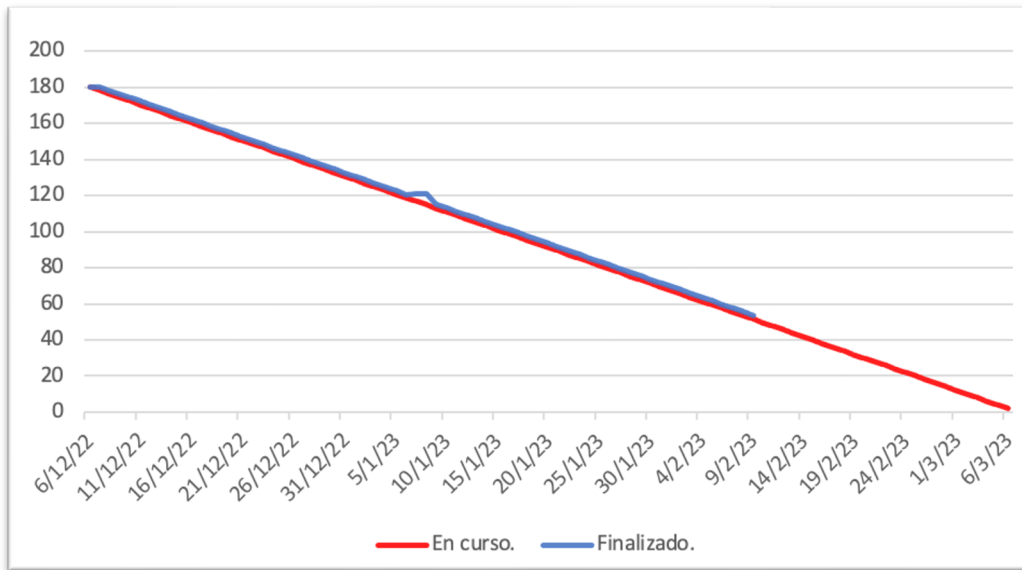


Ilustración 28 - Burndown chart del segundo Sprint.

Fuente: Autores.

### 4.3. Resultado del Sprint 3

- Crear interfaz de búsqueda de infracciones.
- Crear el login.

La Tabla 26 muestra el progreso del tercer Sprint, en donde la tarea de desarrollo “Crear la interfaz de búsqueda de infracciones” se encuentra En desarrollo.

Sprints	Inicio: 12/ Febrero /2023 Fin: 06/ Marzo /2023		Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Tareas a desarrollar	Pendiente	En desarrollo	Terminado	
Sprint 1	Creación de la base de datos.			√	
	Mantenimiento del personal.			√	
Sprint 2	Mantenimiento de infracciones.			√	
	Crear el login.			√	

<b>Sprint 3</b>	Crear la interfaz de búsqueda de infracciones.		√	
	Implementar el botón de pagos.	√		

Tabla 26 - Taskboard del tercer Sprint.

Fuente: Autores.

La Ilustración 29 muestra el avance del tercer Sprint tareas pendientes y en desarrollo según el tiempo de desarrollo planificado.

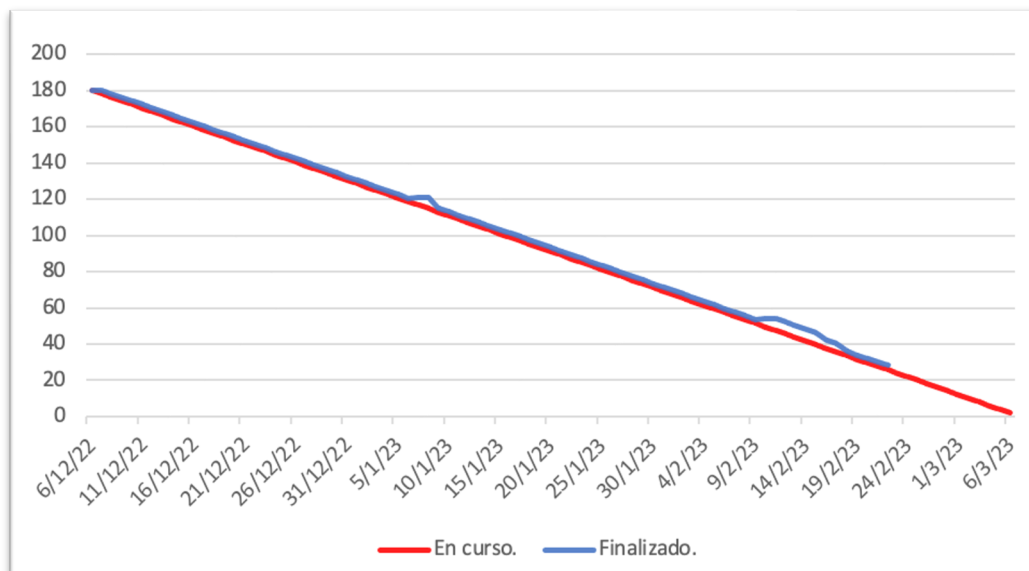


Ilustración 29 -Burndown chart del tercer Sprint.

Fuente: Autores.

### Crear la interfaz de búsqueda de infracciones.

La Ilustración 30 e Ilustración 31 muestran la interfaz de búsqueda de infracciones.

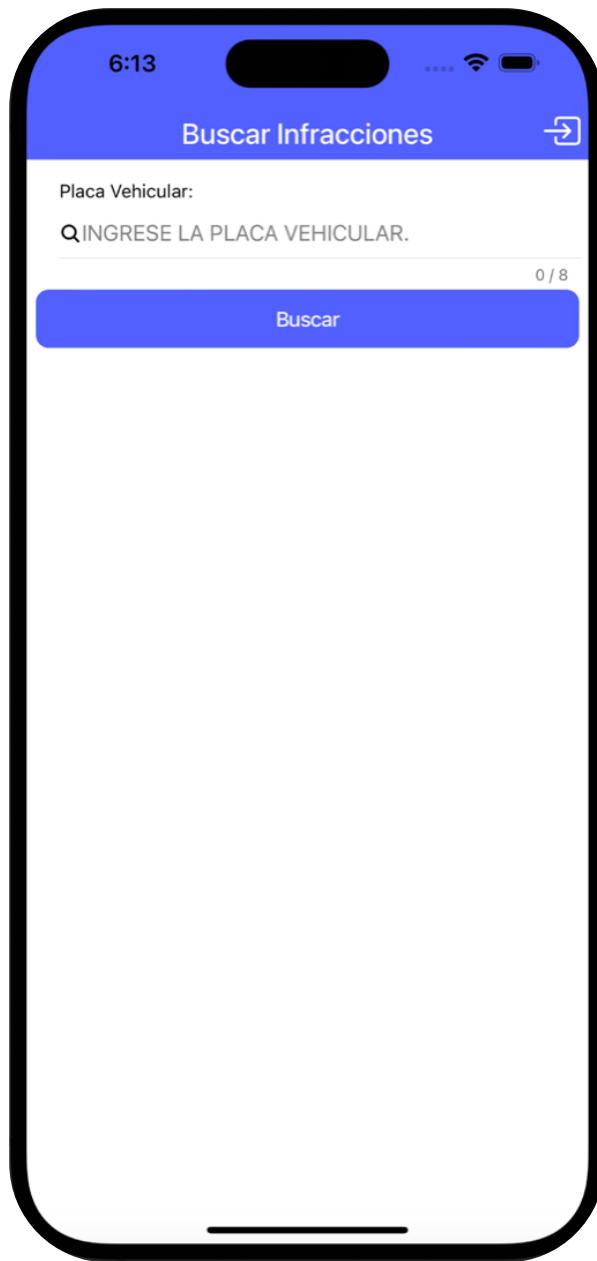


Ilustración 30 - Interfaz de búsqueda de infracciones en IOS.

Fuente: Autores.



Ilustración 31 - Interfaz de búsqueda de infracciones en Android.

Fuente: Autores.

En la Ilustración 32 e Ilustración 33 se observa la interfaz con una búsqueda realizada.

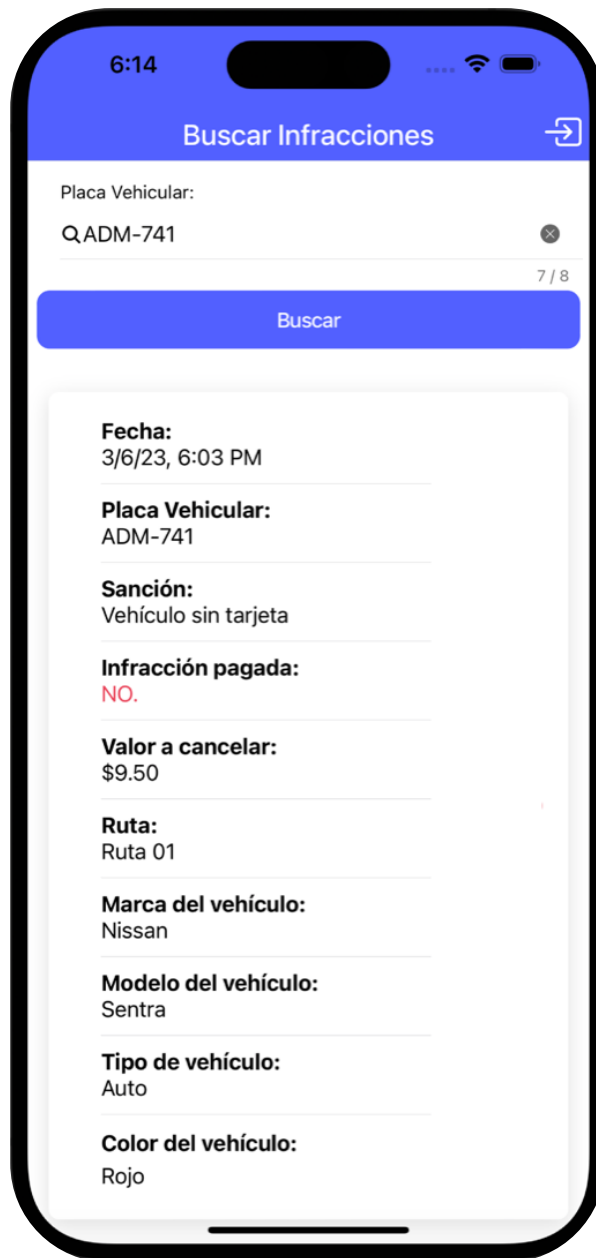


Ilustración 32 - Interfaz de búsqueda realizada en IOS.

Fuente: Autores.

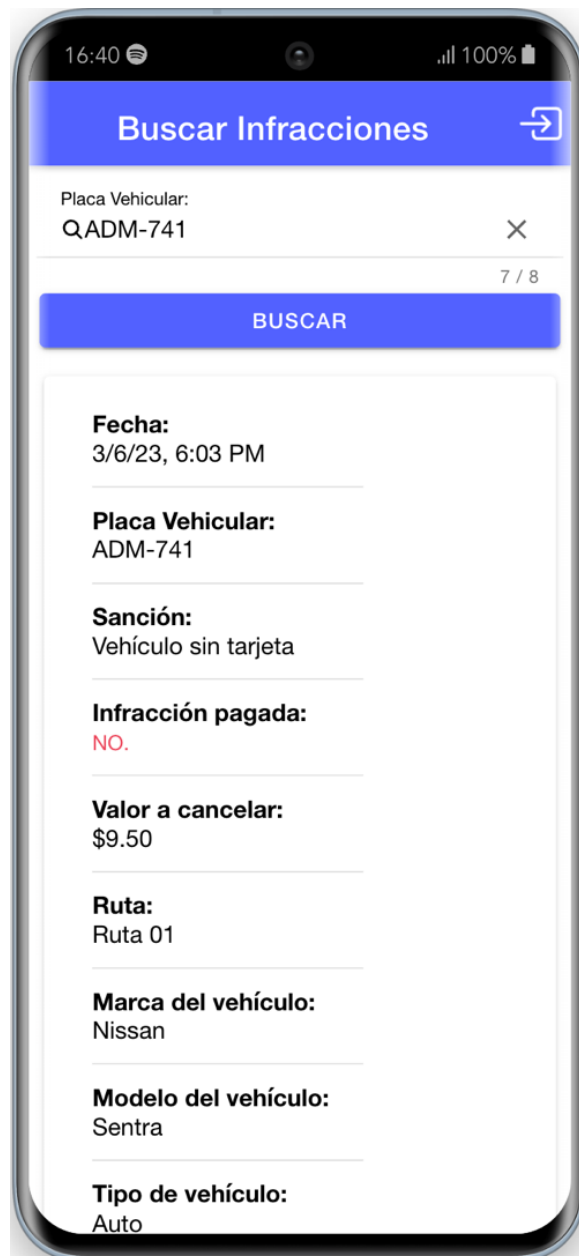


Ilustración 33 - Interfaz de búsqueda realizada en Android.

Fuente: Autores.

La Tabla 27 muestra el Taskboard del tercer Sprint, con la tarea “Crear la interfaz de búsqueda de infracciones” se encuentra finalizada mientras que la tarea “Implementar el botón de pagos” se encuentra en desarrollo.

N. Sprints	Inicio: 12/ Febrero /2023	Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Fin: 06/ Marzo /2023			
	Tareas a desarrollar	Pendiente	En desarrollo	Terminado
Sprint 1	Creación de la base de datos			√
	Mantenimiento del personal.			√
Sprint 2	Mantenimiento de infracciones			√
	Crear el login			√
Sprint 3	Crear la interfaz de búsqueda de infracciones			√
	Implementar el botón de pagos		√	

Tabla 27 - Taskboard del tercer Sprint.

Fuente: Autores.

La Ilustración 34 muestra el Burndown chart del tercer Sprint con tareas terminadas y en desarrollo según el tiempo de desarrollo planificado.

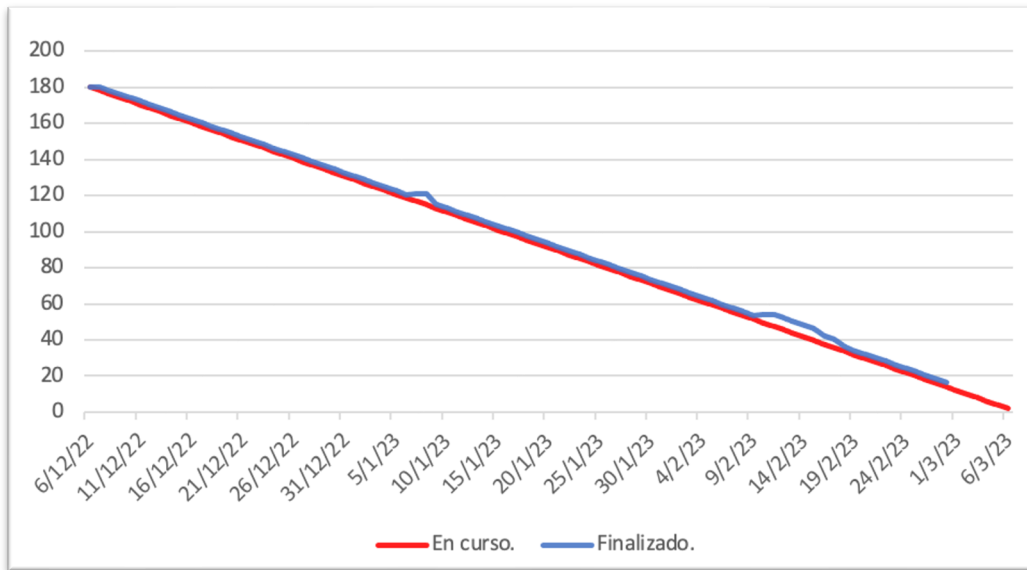


Ilustración 34 - Burndown chart del Sprint 3.

Fuente: Autores.

### Implementar el botón de pago.

En la Ilustración 35 e Ilustración 36 se observa la interfaz con el botón de pagos implementado.

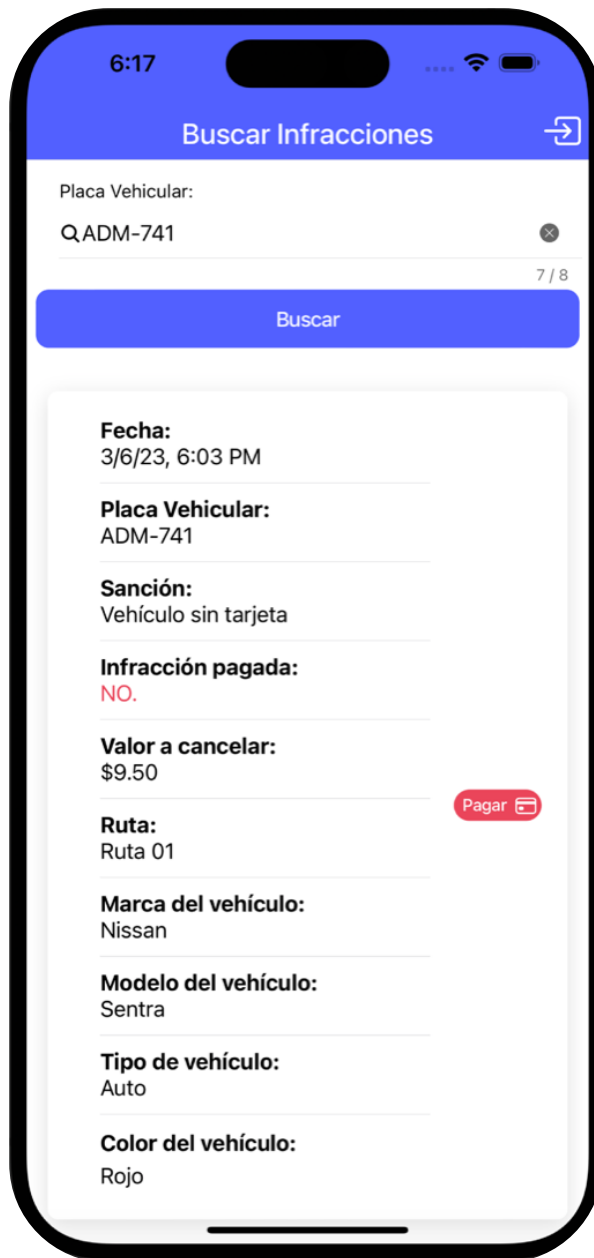


Ilustración 35 - Botón de pago en IOS.

Fuente: Autores.

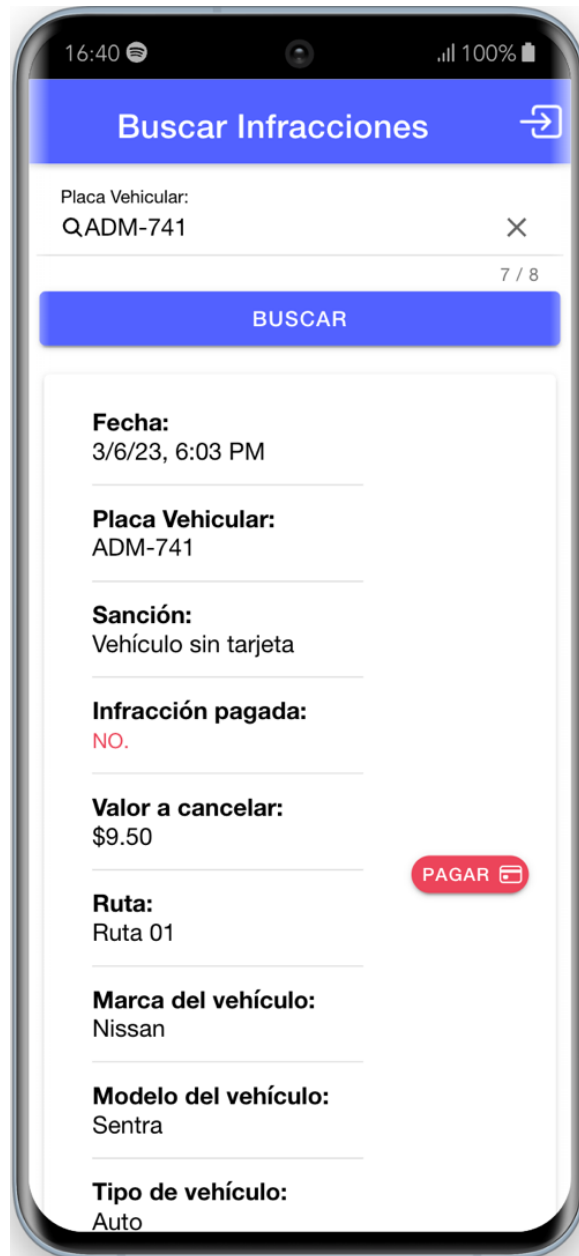


Ilustración 36 - Botón de pago en Android.

Fuente: Autores.

En la Ilustración 37 e Ilustración 38 se observa la interfaz de los detalles de la infracción a pagar.

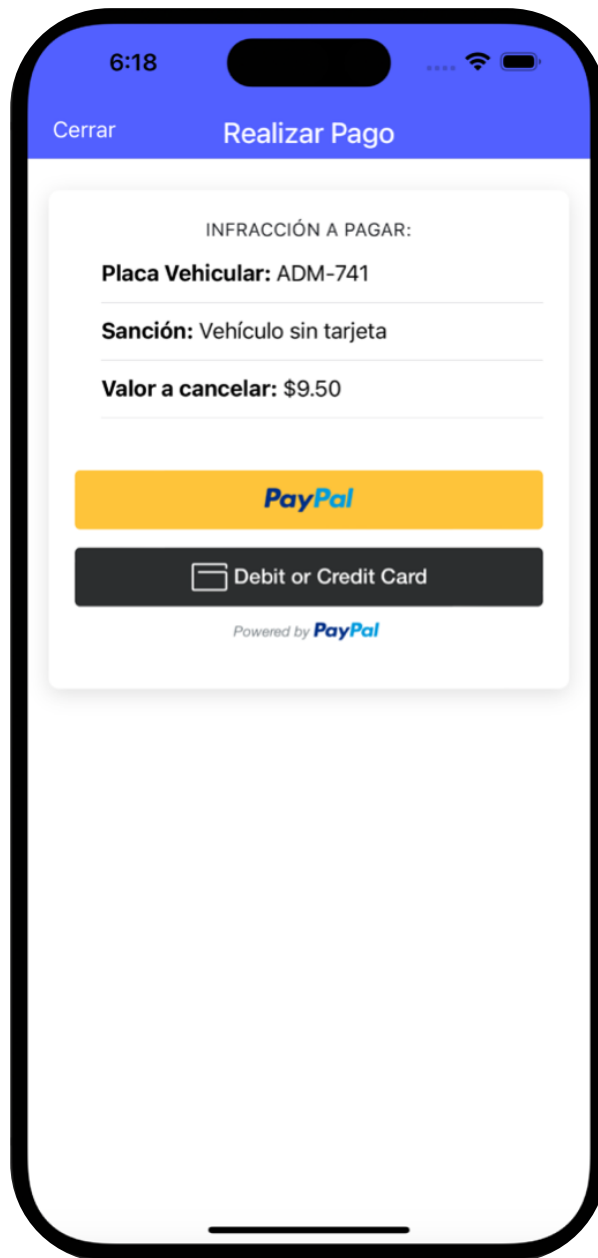


Ilustración 37 - Interfaz detalle de pago en IOS.

Fuente: Autores.

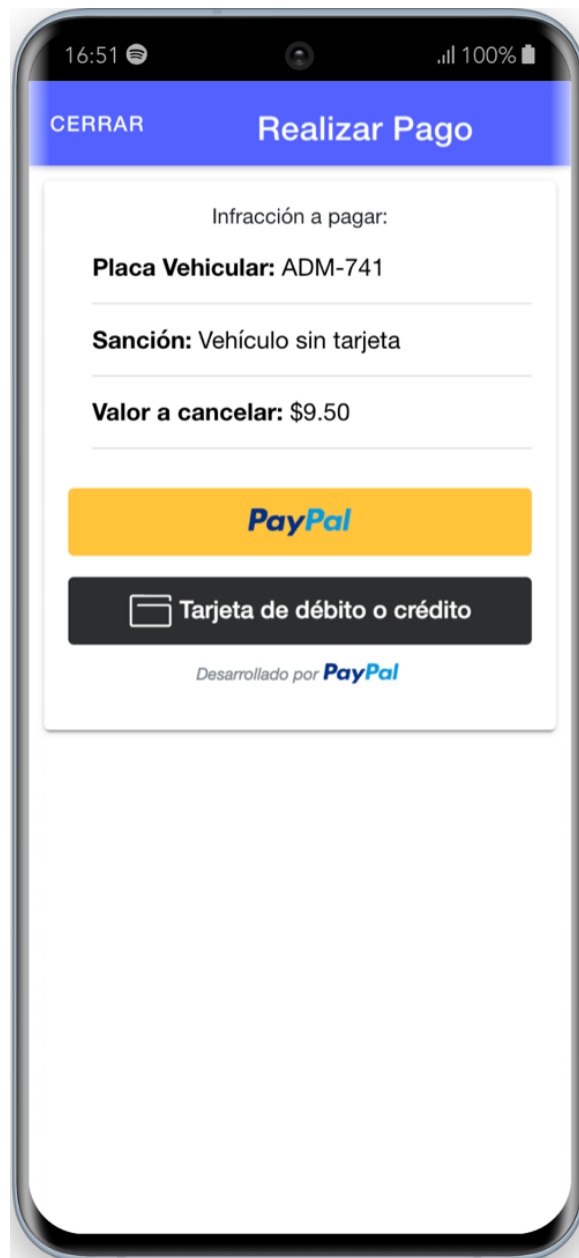


Ilustración 38 - Interfaz detalle de pago en Android.

Fuente: Autores.

En la Ilustración 39 e Ilustración 40 se observa la interfaz de inicio de sesión de PayPal para cancelar la infracción.

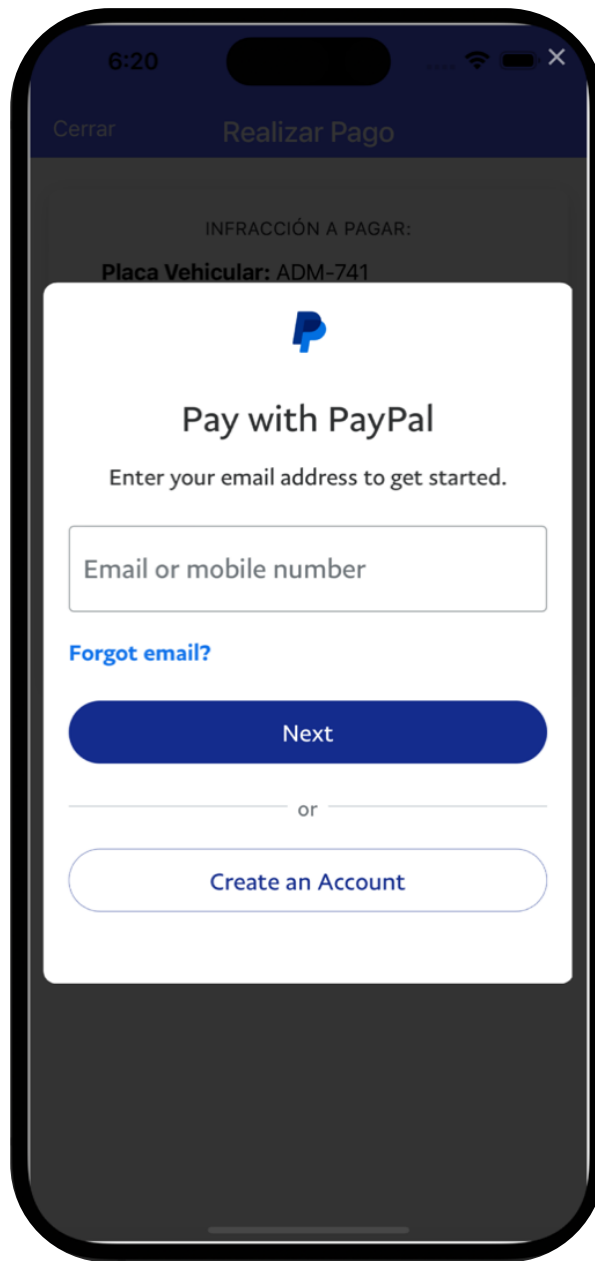


Ilustración 39 - Interfaz inicio de sesión PayPal en iOS.

Fuente: Autores.

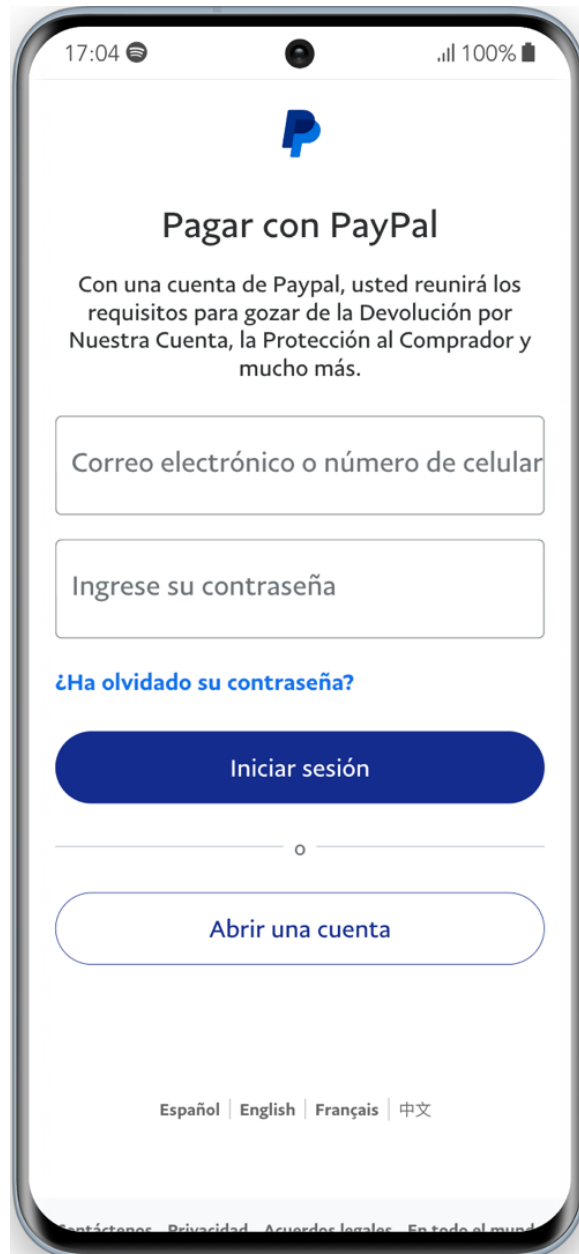


Ilustración 40 - Interfaz inicio de sesión PayPal en Android.

Fuente: Autores.

En la Ilustración 41 e Ilustración 42 se observa la interfaz para realizar el pago sin iniciar sesión en PayPal mediante una tarjeta de crédito o débito.

6:26

Cerrar Realizar Pago

Debit or Credit Card

N.º de la tarjeta

Fecha de vencimiento CSC

Dirección de la tarjeta

Nombre Apellidos

Calle y número de casa

Dirección (continuación)

Código postal

Población

Móvil +593

Correo electrónico

Enviar a la dirección de mi tarjeta

Confirmando que soy mayor de edad y acepto el [Aviso de privacidad](#) de PayPal.

Pagar ahora

Ilustración 41 - Formulario pago por tarjeta de crédito o débito en IOS.

Fuente: Autores.



Ilustración 42 - Formulario pago por tarjeta de crédito o débito en Android.

Fuente: Autores.

En la Ilustración 43 e Ilustración 44 se observa la interfaz de la pasarela de pagos de PayPal con los datos de la factura.

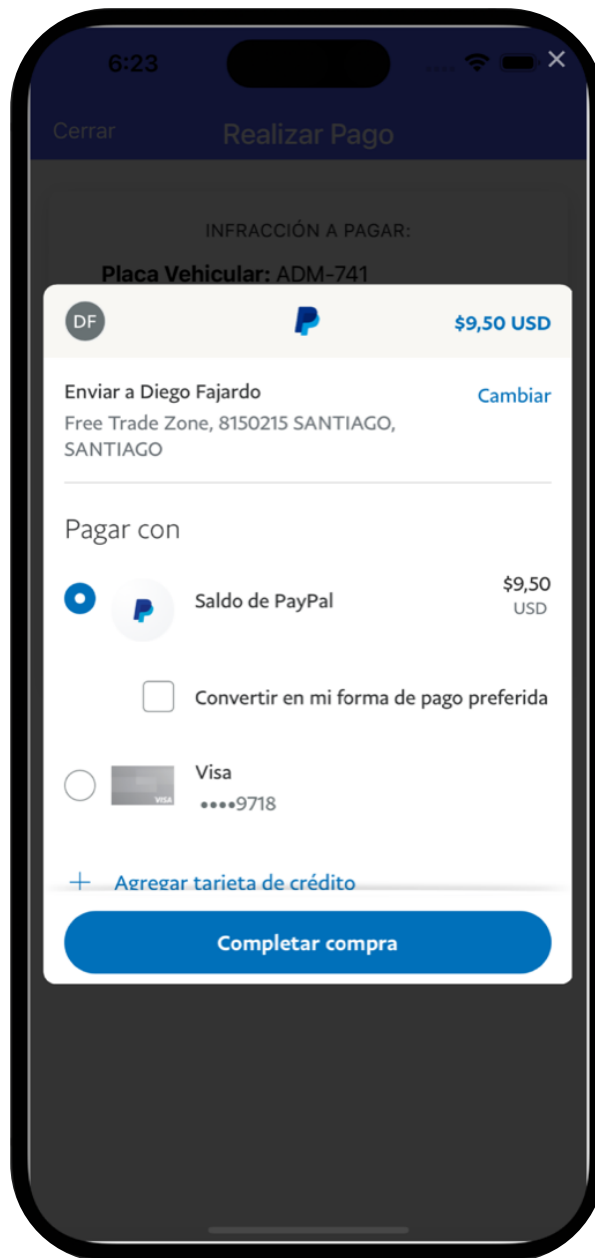


Ilustración 43 - Interfaz para realizar el pago y facturación en IOS.

Fuente: Autores.

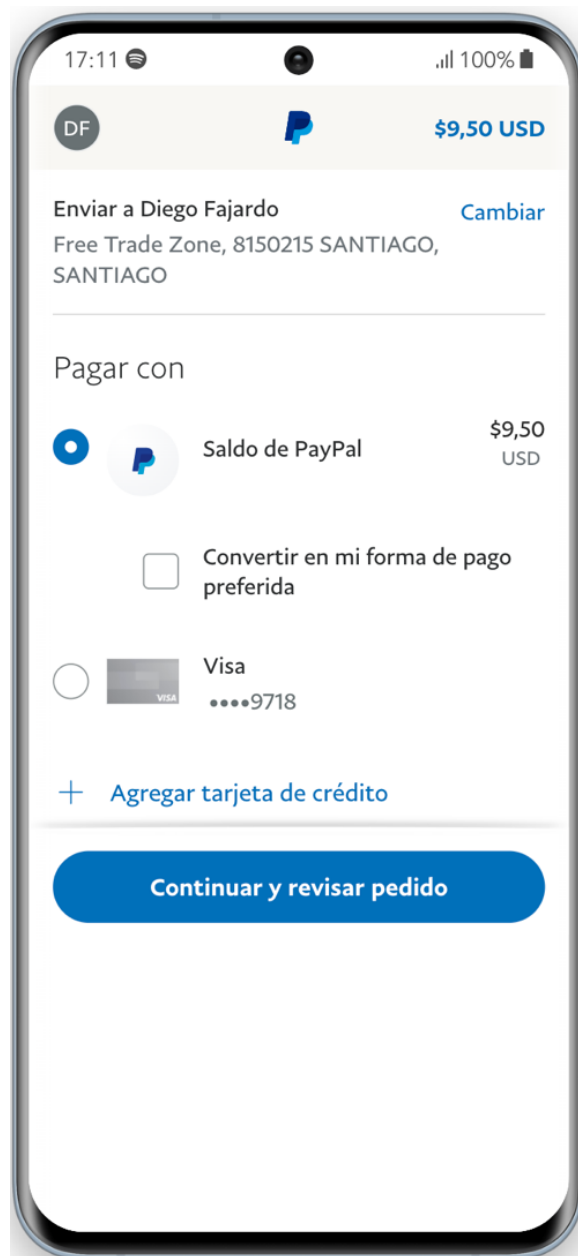


Ilustración 44 - Interfaz para realizar el pago y facturación en Android.

Fuente: Autores.

En la Ilustración 45 e Ilustración 46 se observa la interfaz de la infracción con el valor cancelado.

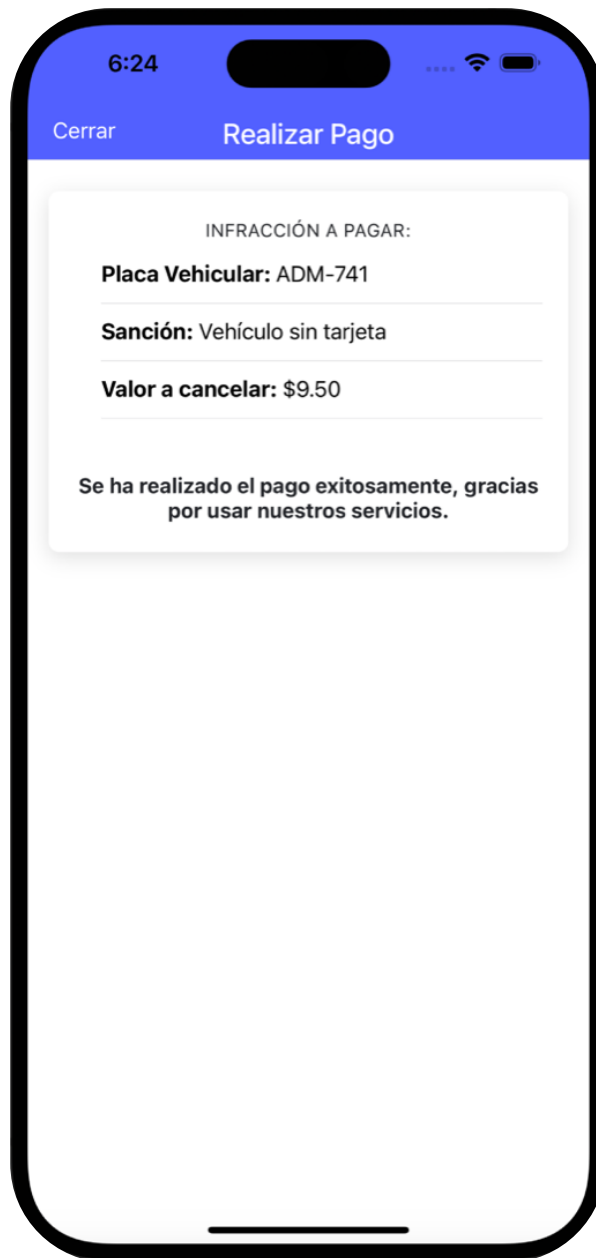


Ilustración 45 - Interfaz del pago realizado en IOS.

Fuente: Autores.

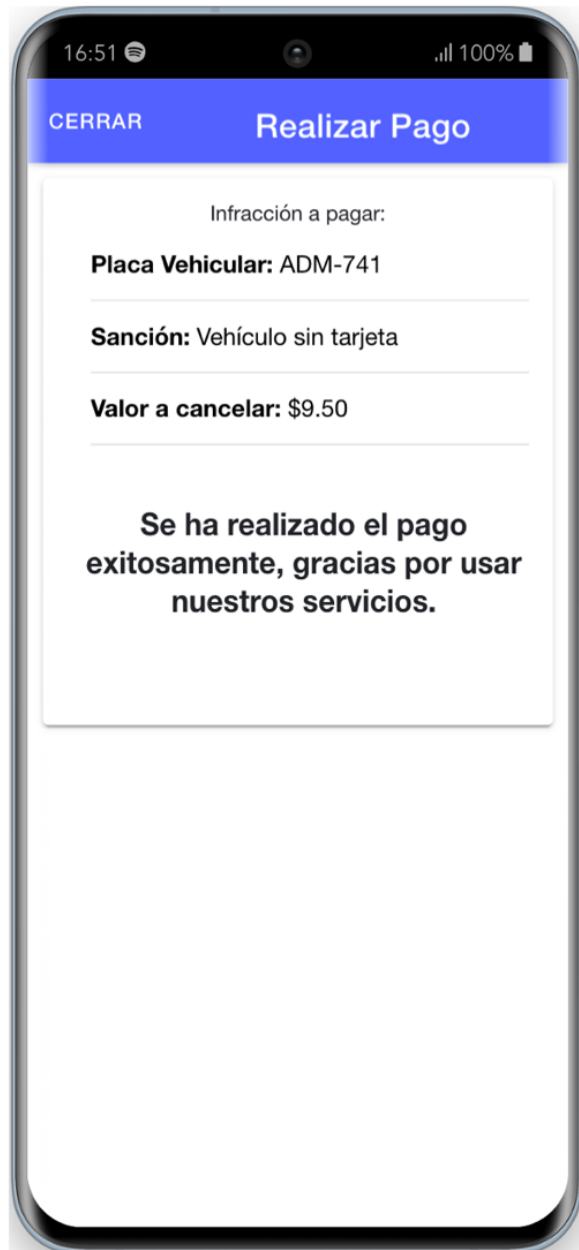


Ilustración 46 - Interfaz del pago realizado en Android.

Fuente: Autores.

La Tabla 28 muestra Taskboard del tercer Sprint, con las tareas planificadas ya terminadas.

N. Sprints	Inicio: 12/ Febrero /2023	Equipo de trabajo: Diego Fajardo y Marlon Bermejo		
	Fin: 06/ Marzo /2023	Pendiente	En desarrollo	Terminado
Sprint 1	Creación de la base de datos			√
	Mantenimiento del personal			√
Sprint 2	Mantenimiento de infracciones			√
	Crear el login			√
Sprint 3	Crear la interfaz de búsqueda de infracciones.			√
	Implementar el botón de pagos			√

Tabla 28 - Taskboard del tercer Sprint.

Fuente: Autores.

La Ilustración 47 muestra el Burndown chart del del tercer Sprint, el cual se encuentra Terminado de acuerdo al tiempo de desarrollo planificado.

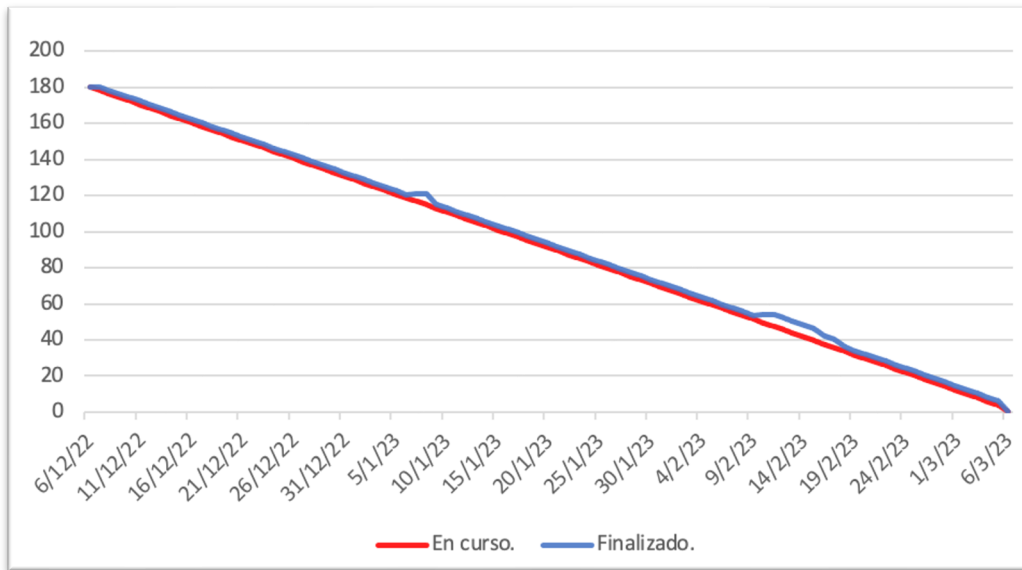


Ilustración 47 - Burndown chart del tercer Sprint.

Fuente: Autor.

#### 4.4. Entregables Finales

De acuerdo a la finalización del último Sprint conforme al tiempo establecido, se da por concluido el desarrollo del prototipo de aplicación híbrida para el registro, consulta y pago de multas de SEMERTAZ.

Al ser un prototipo de aplicación híbrida, el servidor y la aplicación se encuentran configuradas localmente, para implementar el prototipo es necesario contar con un servicio en la nube en donde se aloje el backend, y tener los permisos necesarios para publicar la aplicación en las tiendas oficiales de IOS y Android.

De igual forma, los pagos de infracciones que se realizan mediante el botón de PayPal, es una simulación funcional, por lo que, al momento de realizar la implementación se deberá regular mediante los procedimientos y métodos propios del Servicio de Rentas Internas y Ley Orgánica de Administración Financiera y Control, así como las leyes de comercio electrónico del Ecuador en el cual se incluyen los pagos electrónicos.

## **4.5. Conclusiones y recomendaciones**

### **4.5.1. Conclusiones**

El estudio en diferentes fuentes bibliográficas sobre tecnologías de desarrollo híbrido y metodologías ágiles tales como revistas de alto impacto, artículos e investigaciones nos permitió tener con una base sólida de conocimientos para elegir las herramientas más óptimas en el desarrollo de aplicaciones multi-plataforma y el uso de la metodología ágil para tener un marco de trabajo flexible que se adapta a las necesidades y cambios del proyecto. De modo que, los tiempos y costes de desarrollo son menores en comparación al desarrollar aplicaciones nativas.

La información recolectada del Servicio Municipal de Estacionamiento Rotativo de Azogues permitió comprender cómo funciona el servicio en el entorno, los procesos de gestión de multas y pagos, las regulaciones y políticas aplicables, y cualquier otro aspecto relevante en el levantamiento de los requisitos funcionales de la aplicación híbrida propuesta.

El prototipo de aplicación móvil híbrida se desarrolló aplicando las fases de la metodología ágil SCRUM, estas se acoplaron a las necesidades identificadas del Servicio de Estacionamiento Rotativo Tarifado de Azogues. Es necesario considerar que se podrán automatizar todos los procesos que se realizan cotidianamente y que al ser implementada será una solución útil y eficiente.

### **4.5.2. Recomendaciones**

- Implementar la aplicación para mejorar y optimizar las actividades del Servicio de Estacionamiento Rotativo Tarifado de Azogues.
- Poner a prueba un semestre el prototipo de aplicación móvil híbrida y observar los beneficios que otorga.

- Revisar las leyes de comercio electrónico y políticas tributarias al momento de implementar pagos en línea.

## Bibliografía

- [1] G. Campoverde Crespo, "Análisis de la política de estacionamiento público aplicado en la ciudad de azogues enfocado a la alternabilidad por tiempo," Universidad del Azuay, 2016.
- [2] Municipio de Azogues, "Ordenanzas Municipales. Cantón Azogues: Sustitutiva que norma el servicio municipal de estacionamiento rotativo tarifado en las vías públicas del área central de la ciudad," 2017. <https://vlex.ec/vid/canton-azogues-sustitutiva-norma-587449418>.
- [3] J. Montoro, "¿Cuántos coches hay en el mundo en 2021? | carwow," Aug. 05, 2021.
- [4] el Comercio, "Parque automotor de Ecuador creció en 1,4 millones de vehículos en una década - El Comercio," 2019. <https://www.elcomercio.com/actualidad/negocios/parque-automotor-ecuador-crecimiento-decada.html>.
- [5] INEC, "Población y Demografía," 2010. <https://www.ecuadorencifras.gob.ec/censo-de-poblacion-y-vivienda/>.
- [6] I. Milton and P. Urgilés, "Estimación de un inventario de emisiones de fuentes móviles terrestres para la ciudad de Azogues aplicando el Modelo Internacional de Emisiones Vehiculares," 2019[Online]. Available: <http://dspace.uazuay.edu.ec/handle/datos/8894>
- [7] P. Ordoñez, L. Mato, and M. Villa, "Tecnificación del sistema municipal de estacionamiento rotativo tarifado (SIMERT) de la ciudad de Loja, módulo de servicios en aplicaciones móviles." 2014. [Online]. Available: <https://dspace.unl.edu.ec//handle/123456789/13981>
- [8] L. Guamán and M. Reascos, "Sistema de control del parqueo tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra." Jun. 2014, 2023. [Online]. Available: <http://repositorio.utn.edu.ec/handle/123456789/2663>
- [9] Gobierno Autónomo Descentralizado de la ciudad de Ibarra, "RENDICIÓN DE CUENTAS SISMERT POR CAMBIO DE DIRECCIÓN," 2016. <http://www.ibarra.gob.ec/web/docs/2016rc/06ProcesosDeRendicionDeCuentas/FASE1/SISMERT.pdf>
- [10] CONCEJO CANTONAL DE CUENCA, "ORDENANZA QUE REGULA Y CONTROLA LA OCUPACIÓN DE LAS VIAS PÚBLICAS POR LOS VEHÍCULOS MOTORIZADOS DENTRO DEL CANTÓN CUENCA Y FUNCIONAMIENTO DEL SISTEMA DE ESTACIONAMIENTO ROTATIVO TARIFADO Y PARQUEO INDEBIDO-SERT-. EL I. CONCEJO CANTONAL DE CUENCA CONSIDERANDO," 2012.
- [11] F. S. Illanes Rumiguano, "Desarrollo de una aplicación Web-móvil para rastreo, seguimiento y reporte de contingencias del transporte escolar. Caso de estudio: Unidad Educativa San José La Salle," PUCE-Quito, 2022.
- [12] R. R. Vique, "Métodos para el desarrollo de aplicaciones móviles," 2019.
- [13] E. D. Torres Castillo, "Desarrollo de una aplicación móvil para la enseñanza de guitarra y Ukelele con realidad aumentada. Caso de estudio: Import Music," PUCE-Quito, 2022.
- [14] P. Thomas *et al.*, "Tendencias en el desarrollo de Aplicaciones para Dispositivos Móviles," in *Workshop de Investigadores en Ciencias de la Computación*, 2018.
- [15] Ionic Framework, "Ionic Framework - The Cross-Platform App Development Leader." <https://ionicframework.com/>.

- [16] F. X. Marchán Salgado, "Análisis, diseño y desarrollo de un sistema informático con interfaz Web y móvil para la gestión de despacho de combustible en activos fijos y control de inventario del consumo de bodegas. Caso de estudio: servicios de hidrocarburos SERTECPET SA-Ecuador," PUCE-Quito, 2022.
- [17] P. G. Cuascota Cabascango, "Desarrollo de una aplicación móvil para fortalecer la inclusión de mujeres en las áreas de ciencia, tecnología, ingeniería y matemáticas (stem) en la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte, integrando las tecnologías Api Rest y el Framework Ionic.," 2022.
- [18] X. P. Ye, "Diseño e implantación de un sistema de autenticación Cross-platform para React y React Native," 2022.
- [19] E. V. Macias Vera, "Estudio comparativo de los frameworks del desarrollo móvil nativo" Flutter" y" React Native",," BABAHOYO: UTB, 2021, 2021.
- [20] L. F. Litano Ramos, "Desarrollo de una aplicación móvil utilizando Flutter y Firebase para realizar el seguimiento de los tratamientos farmacológicos de un paciente," 2021.
- [21] L. M. M. Fernanda, R. O. Robles, and A. O. Arciniega, "Flutter y Laravel una Alternativa para el Desarrollo de Sitios Web.," 2020.
- [22] Node, "Acerca | Node.js." <https://nodejs.org/es/about/>.
- [23] Deno, "Introduction | Manual | Deno." <https://deno.land/manual@v1.26.1/introduction>.
- [24] L. A. Saltos Pérez, "Estudio comparativo entre bases de datos relacional y no relacional," 2022. [Online]. Available: <http://dspace.utb.edu.ec/handle/49000/11669>
- [25] B. Campoverde Vega, V. Andree, M. O. Moreno, and R. Liliana, "Análisis comparativo de rendimiento en gestores de bases de datos relacionales y no relacionales," *Repositorio Institucional - USS*, 2022. [Online]. Available: <http://repositorio.uss.edu.pe/handle/20.500.12802/9211>
- [26] L. Marrero *et al.*, "Un estudio comparativo de bases de datos relacionales y bases de datos NoSQL," *PWA para unificar el desarrollo Desktop, Web y Mobile*, pp. 1–9, 2019. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/91403>
- [27] MongoDB, "Introduction to MongoDB — MongoDB Manual," 2018. <https://www.mongodb.com/docs/manual/introduction/>
- [28] MongoDB, "What is MongoDB Atlas? — MongoDB Atlas," 2019. <https://www.mongodb.com/docs/atlas/>
- [29] M. Trigás Gallego, "Metodología Scrum," Jun. 2012. [Online]. Available: <https://openaccess.uoc.edu/handle/10609/17885>
- [30] J. A. Peña Peña and D. X. González Miranda, "Sistema de información web para la gestión de datos referente a la recolección de desechos domiciliarios aplicando geolocalización en el GAD Municipal del Cantón Azogues," 2021. [Online]. Available: <https://dspace.ucacue.edu.ec/handle/ucacue/10422>
- [31] K. Schwaber, J. Sutherland, and L. G. Definitiva, "La Guía Scrum," 2020.
- [32] Visual Studio Code, "Visual Studio Code - Code Editing. Redefined." <https://code.visualstudio.com/>.

- [33] R. Martin Morales and others, "Extensión para el editor Atom. Web Bookmarks," 2019.
- [34] Community, "sublimetext3 eBook." <https://riptutorial.com/ebook/sublimetext3>.
- [35] Stack Overflow, "Stack Overflow Developer Survey 2022," 2022.  
<https://survey.stackoverflow.co/2022/#section-most-popular-technologies-integrated-development-environment>.
- [36] Municipio de Azogues, "ORDENANZA MUNICIPAL: Cantón Azogues: Sustitutiva que norma el servicio municipal de estacionamiento rotativo tarifado en las vías públicas," Azogues , 2021.

## Anexos

### Anexo A. Muestra de la población

$$n = \frac{N Z^2 pq}{d^2(N - 1) + Z^2 pq}$$

En donde:

- N = Número total de la población.
- Z = Nivel de confianza.
- P = Proporción aproximada del fenómeno en estudio en la población de referencia
- Q = Proporción de la población de referencia que no presenta el fenómeno en estudio
- D = Margen de error.

El resultado de la muestra se presenta a continuación:

$$n = \frac{(35)(1.96)^2(0.50)(0.50)}{0.05^2(35 - 1) + 1.96^2(0.50)(0.50)}$$

$$n = \frac{33,61}{1,05}$$

$$n = 32,00$$

## **Anexo B. Encuesta a controladores de tráfico de SEMETAZ**

### **1. ¿Cuál es el tiempo aproximado para registrar una multa en el sistema?**

- 10 minutos.
- 20 minutos.
- 30 minutos.
- 40 minutos.
- 50 minutos.
- 1 hora.
- Más de una hora.

### **2. ¿Cuántas multas registra diariamente?**

- De 1 a 5 multas.
- De 5 a 10 multas.
- De 10 a 15 multas.
- De 15 a 20 multas.

### **3. Del total del infracciones registradas en un día, Cuantas presentan errores?**

- 1.
- 5.
- 10.
- 15.
- 20.
- Más de 20.

### **4. ¿Cuáles son las sanciones que más se registran?**

- Vehículo sin tarjeta.
- Tarjeta no legible.
- Tarjeta no vigente.
- Mal uso de tarjeta.
- Tarjeta alterada.
- Tiempo excedido.
- Ventas en la vía pública.
- Tiempo excedido con carnet de discapacidad.

- Estacionado fuera del lugar para motos.
- Vehículo pesado.
- Vehículo en paso cebra.
- Vehículo en línea amarilla.
- Vehículo en boca calle.
- Vehículo en espacio de discapacitados.
- Vehículo reservado para particulares.
- Vehículo en parada de bus.
- Vehículo frente a hidrante.
- Vehículo en entrada a garaje.
- Vehículo sobre vereda.
- Vehículo doble fila.
- Vehículo en espacio de motocicletas.

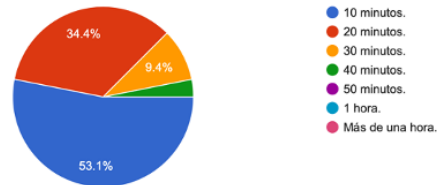
**5. ¿Cuál es el error que comúnmente se presenta al momento de registrar una infracción?**

- Comunicación con la oficina.
- Al enviar los datos de la placa vehicular.
- Al registrar los datos en la tarjeta física.
- Ningún problema.
- Otros.

# Anexo C. Resultados de la encuesta realizada a controladores de tráfico de SEMERTAZ

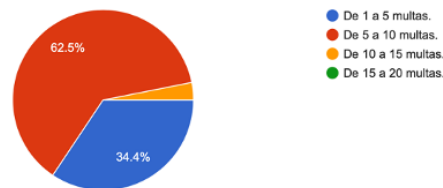
1. ¿Cuál es el tiempo aproximado para registrar una multa en el sistema?

32 respuestas



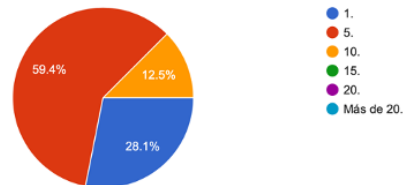
2. ¿Cuántas multas registra diariamente?

32 respuestas



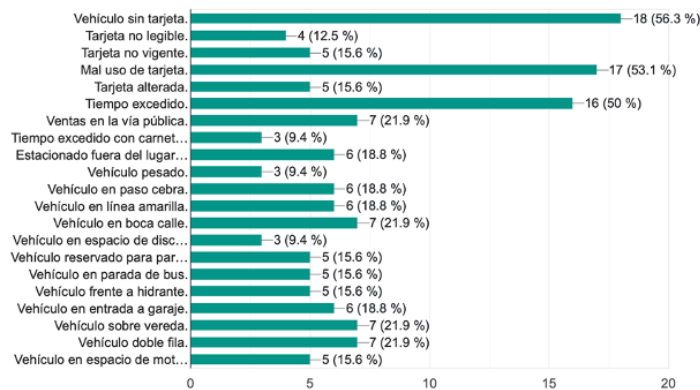
3. Del total de infracciones registradas en un día, Cuantas presentan errores?

32 respuestas



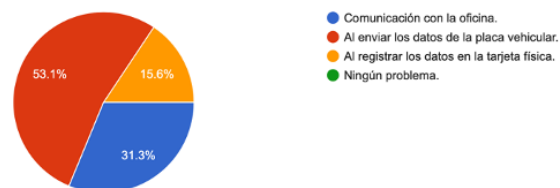
4. ¿Cuáles son las sanciones que más se registran?

32 respuestas



5. ¿Cuál es el error que comúnmente se presenta al momento de registrar una infracción?

32 respuestas



**Diego Gabriel Fajardo Uzhca** portador de la cédula de ciudadanía N° **0302986153**. En calidad de autor y titular de los derechos patrimoniales del trabajo de titulación **“Propuesta y desarrollo de un prototipo de aplicación móvil híbrida para el registro, consulta y pago de multas de SEMERTAZ”** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de éste trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, **02 de mayo de 2023**



F: .....

**Diego Gabriel Fajardo Uzhca**

**C.I. 0302986153**

**Marlon Briant Bermejo Lema** portador de la cédula de ciudadanía N° **0350259750**. En calidad de autor y titular de los derechos patrimoniales del trabajo de titulación **“Propuesta y desarrollo de un prototipo de aplicación móvil híbrida para el registro, consulta y pago de multas de SEMERTAZ”** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de éste trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, **02 de mayo de 2023**



F: .....

**Marlon Briant Bermejo Lema**

**C.I. 0350259750**