

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE SISTEMAS**

**UNIDAD DE TITULACIÓN**

## **DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE PASAJES DE BUSES INTERPROVINCIALES**

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

**VICTOR FERNANDO CARRASCO MENA**

victor.carrasco@epn.edu.ec

**Director: Pamela Catherine Flores Naranjo, Ph.D.**

pamela.flores@epn.edu.ec

**Quito, febrero 2023**

## DECLARACIÓN

Yo, Victor Fernando Carrasco Mena, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

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



---

VICTOR FERNANDO CARRASCO MENA

## CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Victor Fernando Carrasco Mena bajo nuestra supervisión.

---

PAMELA FLORES, PhD.  
DIRECTOR DE PROYECTO

## **DEDICATORIA**

Dedico el presente documento a mis padres CARRASCO VINUEZA VICTOR HUGO y MENA MOLINA LILIAN SOFÍA, por ser siempre el apoyo que necesitaba, por darme la oportunidad de estudiar en una excelente universidad y por nunca dejar que falte el pan en la mesa.

A Vanessa por estar conmigo en los buenos y malos momentos y nunca dudar de mi incluso en los momentos más difíciles de mi vida siempre ha estado ahí cuando lo necesitaba.

A mi hermano Danilo, por el cual siempre estoy dispuesto a dar de mí mismo para poder dar un buen ejemplo.

A mis compañeros de universidad los cuales admiro y respeto mucho.

A mis maestros de la Facultad de Ingeniería en Sistemas por enseñarme las más valiosas lecciones no solo sobre la carrera si no también sobre ética, valores y principios.

Finalmente la dedico a todos mis amigos que me apoyaron y supieron darme su hombro en las épocas difíciles.

## **AGRADECIMIENTO**

Agradezco a los profesores de la Facultad de Ingeniería en Sistemas los cuales con sus enseñanzas y anécdotas me motivaron a ser el profesional de élite en el que me he convertido, también agradezco a mis compañeros por no solo ser parte del día a día en las aulas si no por ser un apoyo en todos los cursos que compartimos.

# Índice de Contenido

DECLARACIÓN .....	ii
CERTIFICACIÓN .....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
Índice de Contenido .....	1
Índice de Figuras.....	2
Capítulo 1 - Introducción .....	6
1.1 OBJETIVO GENERAL .....	6
1.2 OBJETIVOS ESPECÍFICOS .....	6
1.3 ESTADO DEL ARTE .....	6
1.3.1 Necesidades del transporte Interprovincial.....	6
1.3.2 Uso de aplicaciones móviles en el Ecuador .....	6
1.3.3 Uso de sistemas de información en compañías de transporte interprovincial .....	8
1.3.4 Realidad Actual .....	8
Capítulo 2 - Marco teórico .....	9
2.1 EXTREME PROGRAMMING (XP) .....	11
2.1.1 Prácticas Primarias.....	13
2.1.2 Practicas Corolarias.....	16
2.1.3 Equipo de XP .....	18
2.1.4 Ciclos de XP para el desarrollo de aplicaciones móviles .....	19
2.2 TEST DRIVEN DEVELOPMENT (TDD).....	22
Capítulo 3 – Desarrollo de la aplicación .....	23
3.1 FASE DE EXPLORACIÓN .....	23
3.1.1 Stakeholders .....	23
3.1.2 Alcance del proyecto.....	24
3.2 FASE DE INICIALIZACIÓN .....	27
3.2.1 Configuración del proyecto.....	27
3.2.3 Plan de Lanzamiento.....	29
3.2.4 Plan de Iteración .....	30
3.3 FASES DE PRODUCCIÓN Y PRUEBAS .....	31
3.3.2 Primera fase .....	34
3.3.2 Segunda fase.....	46
3.3.3 Tercera fase.....	52
3.3.4 Cuarta fase .....	57
3.3.5 Quinta fase.....	64
Capítulo 4 – Resultados .....	68
Capítulo 5 – Conclusiones y Recomendaciones .....	70
5.1 CONCLUSIONES.....	70
5.2 RECOMENDACIONES .....	71
Referencias .....	71

## Índice de Figuras

Fig. 1. Situación digital del Ecuador 2022 [2] .....	7
Fig. 2. Arquitectura general del sistema .....	26
Fig. 3. Arquitectura de software del sistema .....	29
Fig. 4. Modelo Entidad Relación de la base de datos .....	29
Fig. 5. Ciclo de lanzamiento de aplicaciones móviles propuesto por Microsoft para el desarrollo con Xammarin.Forms [5] .....	30
Fig. 6. Pantalla de inicio de sesión pasajero, Android .....	38
Fig. 7. Pantalla de inicio de sesión pasajero, iOS .....	38
Fig. 8. Pantalla de inicio de sesión conductor, Android .....	39
Fig. 9. Pantalla de inicio de sesión conductor, iOS .....	39
Fig. 10. Pantalla registro de pasajero, Android .....	40
Fig. 11. Pantalla registro de pasajero, iOS .....	40
Fig. 12. Pantalla registro de pasajero con datos, Android .....	40
Fig. 13. Pantalla registro de pasajero con datos, iOS .....	40
Fig. 14. Pantalla de confirmación de registro de pasajero, Android .....	41
Fig. 15. Pantalla de confirmación de registro de pasajero, iOS .....	41
Fig. 16. Pantalla registro de conductor Android .....	41
Fig. 17. Pantalla registro de conductor, iOS .....	41
Fig. 18. Pantalla registro de conductor con datos, Android .....	42
Fig. 19. Pantalla registro de conductor con datos, iOS .....	42
Fig. 20. Mensaje de confirmación de registro de conductor, Android .....	42
Fig. 21. Mensaje de confirmación de registro de conductor, iOS .....	42
Fig. 22. Pantalla olvido de contraseña de conductor, Android .....	43
Fig. 23. Pantalla olvido de contraseña de conductor, iOS .....	43
Fig. 24. Pantalla olvido de contraseña de conductor con datos, Android .....	43
Fig. 25. Pantalla olvido de contraseña de conductor con datos, iOS .....	43
Fig. 26. Mensaje de error al recuperar la contraseña de conductor, Android .....	44
Fig. 27. Mensaje de error al recuperar la contraseña de conductor, iOS .....	44
Fig. 28. Pantalla olvido de contraseña de pasajero, Android .....	44
Fig. 29. Pantalla olvido de contraseña de pasajero, iOS .....	44
Fig. 30. Pantalla olvido de contraseña de pasajero con datos, iOS .....	45
Fig. 31. Pantalla olvido de contraseña de pasajero con datos, iOS .....	45
Fig. 32. Mensaje de error al recuperar la contraseña de pasajero, Android .....	45
Fig. 33. Mensaje de error al recuperar la contraseña de pasajero, iOS .....	45

Fig. 34. Pantalla de gestión de viajes -pasajero, Android.....	49
Fig. 35. Pantalla de gestión de viajes -pasajero, iOS.....	49
Fig. 36. Pantalla de búsqueda de rutas-pasajero, Android.....	49
Fig. 37. Pantalla de búsqueda de rutas-pasajero, iOS.....	49
Fig. 38. Pantalla de gestión de unidades-conductor, Android.....	51
Fig. 39. Pantalla de gestión de unidades-conductor, iOS.....	51
Fig. 40. Pantalla de cambio de unidad-conductor, Android.....	51
Fig. 41. Pantalla de cambio de unidad-conductor, iOS.....	51
Fig. 42. Pantalla de Visualización de datos de la reserva- pasajero, Android.....	55
Fig. 43. Pantalla de Visualización de datos de la reserva- pasajero, iOS.....	55
Fig. 44. Pantalla de Selección de parada inicial- pasajero, Android.....	55
Fig. 45. Pantalla de Selección de parada inicial- pasajero, iOS.....	55
Fig. 46. Pantalla de visualización de rutas por unidad-conductor, Android.....	56
Fig. 47. Pantalla de visualización de rutas por unidad-conductor, iOS.....	56
Fig. 48. Pantalla de visualización de reservas-conductor, Android.....	56
Fig. 49. Pantalla de visualización de reservas-conductor, iOS.....	56
Fig. 50. Pantalla de visualización de tramos- conductor, Android.....	57
Fig. 51. Pantalla de visualización de tramos- conductor, iOS.....	57
Fig. 53. Pantalla de visualización de asientos- pasajero, Android.....	61
Fig. 54. Pantalla de visualización de asientos- pasajero, iOS.....	61
Fig. 55. Selección de asientos- pasajero, Android.....	62
Fig. 56. Selección de asientos- pasajero, iOS.....	62
Fig. 57. Des enmarcado de asientos-pasajero, Android.....	62
Fig. 58. Des enmarcado de asientos-pasajero, iOS.....	62
Fig. 59. Pantalla de confirmación de asientos para la reserva, Android.....	63
Fig. 60. Pantalla de confirmación de asientos para la reserva, iOS.....	63
Fig. 61. Pantalla de visualización de asientos- conductor, Android.....	64
Fig. 62. Pantalla de visualización de asientos- conductor, iOS.....	64
Fig. 63. Pantalla de pago de reserva- pasajero, Android.....	66
Fig. 64. Pantalla de pago de reserva- pasajero, iOS.....	66
Fig. 65. Pantalla de pago de reserva con datos, Android.....	67
Fig. 66. Pantalla de pago de reserva con datos, iOS.....	67
Fig. 67. Mensaje de confirmación de reserva pasajero, Android.....	67
Fig. 68. Mensaje de confirmación de reserva pasajero, iOS.....	67



## Índice de Tablas

Tabla 1. Cuadro comparativo de marcos de desarrollo ágil. ....	10
Tabla 2. Importancia de cada practica correspondiente a cada rol. ....	22
Tabla 3: Requerimientos Funcionales para la aplicación del pasajero.....	26
Tabla 4: Requerimientos Funcionales para la aplicación del conductor.....	26
Tabla 5. Ranking de impacto para evaluación de riesgos.....	32
Tabla 6. Ranking de probabilidades para evaluación de riesgos .....	33
Tabla 7. Matriz de evaluación de riesgos desarrollo de la aplicación.....	33
Tabla 8. Historia de usuario ingreso al sistema.....	34
Tabla 9. Historia de usuario Registro en el sistema .....	34
Tabla 10. Historia de usuario Ingreso al sistema (Conductor).....	35
Tabla 11. Historia de usuario Registro en el sistema (Conductor). ....	35
Tabla 12. Caso de prueba 1 para el inicio de sesión .....	35
Tabla 13. Caso de prueba 2 para el inicio de sesión .....	35
Tabla 14. Caso de prueba 3 para el inicio de sesión. ....	35
Tabla 15. Caso de prueba 4 para el inicio de sesión .....	36
Tabla 16. Caso de prueba 1 para la pantalla de registro de usuario.....	36
Tabla 17. Caso de prueba 2 para la pantalla de registro de usuario.....	36
Tabla 18. Caso de prueba 3 para la pantalla de registro de usuario.....	36
Tabla 19. Caso de prueba 4 para la pantalla de registro de usuario.....	36
Tabla 20. Caso de prueba 5 para la pantalla de registro de usuario.....	37
Tabla 21. Caso de prueba 1 para la pantalla de recuperación de contraseña. ....	37
Tabla 22. Caso de prueba 2 para la pantalla de recuperación de contraseña. ....	37
Tabla 23. Caso de prueba 3 para la pantalla de recuperación de contraseña. ....	37
Tabla 24. Historia de usuario gestión de viajes.....	46
Tabla 25. Historia de usuario búsqueda de rutas.....	46
Tabla 26. Historia de conductor visualización de unidades.....	47
Tabla 27. Caso de prueba 1 para la Pantalla de inicio (usuario).....	47
Tabla 28. Caso de prueba 2 para la Pantalla de inicio (usuario).....	47
Tabla 29. Caso de prueba 1 para la Pantalla búsqueda de rutas .....	47
Tabla 30. Caso de prueba 2 para la Pantalla búsqueda de rutas. ....	47
Tabla 31. Caso de prueba 3 para la Pantalla búsqueda de rutas. ....	47
Tabla 32. Caso de prueba 1 para la Pantalla de gestión de unidades .....	48
Tabla 33. Caso de prueba 2 para la Pantalla de gestión de unidades. ....	48

Tabla 34. Caso de prueba 1 para la Pantalla cambio de unidad.....	48
Tabla 35. Historia de usuario visualización de datos de la reserva.....	52
Tabla 36. Historia de usuario selección de parada inicial.....	52
Tabla 37. Historia de usuario visualización de rutas por unidad .....	53
Tabla 38. Historia de usuario Visualización de reservas.....	53
Tabla 39. Historia de usuario puntos de abordaje. ....	53
Tabla 40. Caso de prueba 1 para la Pantalla de visualización de datos de la reserva .....	53
Tabla 41. Caso de prueba 2 para la Pantalla de visualización de datos de la reserva. .....	54
Tabla 42. Caso de prueba 1 para la Pantalla de selección de parada inicial. ....	54
Tabla 43. Caso de prueba 2 para la Pantalla de selección de parada inicial. ....	54
Tabla 44. Caso de prueba 1 para la Pantalla de visualización de rutas por unidad. ....	54
Tabla 45. Caso de prueba 2 para la Pantalla de visualización de rutas por unidad. ....	54
Tabla 46. Caso de prueba 1 para la Pantalla de visualización de reservas por unidad. ....	54
Tabla 47. Historia de usuario selección de parada inicial.....	58
Tabla 48. Historia de usuario Visualización de asientos .....	58
Tabla 49. Caso de prueba 1 para la Pantalla de visualización de asientos (pasajero). .....	58
Tabla 50. Caso de prueba 2 para la Pantalla de visualización de asientos (pasajero). .....	58
Tabla 51. Caso de prueba 3 para la Pantalla de visualización de asientos (pasajero). .....	59
Tabla 52. Caso de prueba 1 para la Pantalla de visualización de asientos (conductor). ....	59
Tabla 53. Historia de usuario pago de pasajes .....	65
Tabla 54. Caso de prueba 1 para la Pantalla de pago de reserva. ....	65
Tabla 55. Caso de prueba 2 para la Pantalla de pago de reserva. ....	65
Tabla 56. Caso de prueba 3 para la Pantalla de pago de reserva. ....	65
Tabla 57. Caso de prueba 4 para la Pantalla de pago de reserva. ....	65
Tabla 58. Caso de prueba 5 para la Pantalla de pago de reserva. ....	66

# Capítulo 1 - Introducción

## 1.1 Objetivo general

Desarrollar una aplicación móvil para gestionar los procesos de compra y asignación de puestos en buses interprovinciales.

## 1.2 Objetivos específicos

- Recolectar los requerimientos funcionales y no funcionales de la empresa interesada en el desarrollo del sistema.
- Diseñar una aplicación móvil con el patrón MVVM (Model-View-View-Model)
- Implementar una aplicación móvil multiplataforma que sirva de apoyo a las empresas de transporte interprovincial en el proceso de compra de boletos, utilizando metodologías de desarrollo ágiles.
- Probar cíclicamente el funcionamiento de la lógica de programación mediante TDD.

## 1.3 Estado del arte

### 1.3.1 Necesidades del transporte Interprovincial

En el Ecuador el transporte interprovincial carece de varias cosas entre ellas un sistema centralizado de cobro de pasajes de forma digital y de un sistema informático que permita gestionar la asignación de asientos de acuerdo con su preferencia y disponibilidad. Adicionalmente debido a la emergencia sanitaria que vive el Ecuador y el mundo las empresas de transporte se ven obligadas no solo a controlar la capacidad de sus unidades sino también a tener un medio de pago que permita cumplir con el distanciamiento social, esto no sería posible con medio de pago tradicionales en efectivo. En base al contexto previamente establecido la mejor solución es evitar aglomeraciones en ventanillas y esto se logra con medios de pago digitales como el pago con tarjetas vía online y botones de pago que gestionen dichas transacciones [1].

### 1.3.2 Uso de aplicaciones móviles en el Ecuador

Las aplicaciones móviles se han vuelto cada vez más populares debido a que están al alcance de su bolsillo, no es de extrañar que para febrero de 2022 el 88.4% de los ecuatorianos tengan un smartphone con acceso a redes móviles como se puede apreciar en la figura 1.



Fig. 1. Situación digital del Ecuador 2022 [2]

### **1.3.3 Uso de sistemas de información en compañías de transporte interprovincial**

En el Ecuador, pocas empresas de transporte interprovincial tienen un sistema que permita la asignación de pasajes y el pago online, el mantenimiento y desarrollo de este tipo de herramientas significa gastos adicionales a los previstos en una compañía de transporte, sin mencionar que al no ser su negocio principal el desarrollo de software muchas de las veces van a tener que tercerizar este servicio. Para solucionar este problema es necesario una plataforma centralizada que cobre solo por el uso de la plataforma.

### **1.3.4 Realidad Actual**

Debido a la emergencia sanitaria provocado por el COVID-19 virus que afectó a la humanidad en el 2020 las empresas de transporte se han visto obligadas a tener un mayor control de sus procesos de compra y asignación de pasajes. También es importante recalcar que la movilización a un terminal de buses implica en gastos y en acercamiento a aglomeraciones innecesarias, ahorrar ese proceso al público en general le permitiría evitar ir a terminales y esperar largas filas para comprar pasajes, además si el bus que va a su destino pasa cerca de su casa podría ayudar significativamente a que se movilice innecesariamente.

## Capítulo 2 - Marco teórico

En este capítulo se especifica toda la teoría detrás del desarrollo de la aplicación, el marco de trabajo a utilizar, la planificación y las tecnologías que permitieron desarrollar el sistema como tal y avanzar en el proyecto. 2.1. Metodologías ágiles para desarrollo de aplicaciones móviles.

La filosofía “Ágil” nació en 2001, en Utah (EE. UU.), donde 17 programadores se reunieron para hablar sobre el futuro del desarrollo de software. Juntos llegaron a la conclusión de que los desarrolladores se centran mucho en la planificación y documentación de las etapas de desarrollo de software y se olvidaban de lo más importante, la satisfacción del cliente. De este modo nacieron los 12 principios y 4 valores de la filosofía ‘agile’ recopilados en un documento que se llamó el Manifiesto Ágil. El documento proponía los conceptos básicos para la gestión ágil de proyectos de desarrollo software.

En la actualidad existen diferentes metodologías ágiles para desarrollo de aplicaciones móviles como son: Scrum, Extreme Programming (XP), Scrumban, Hybrid Method Engineering (HME), Mobile-D, MASAN, de las cuales se detalla las ventajas y desventajas en la siguiente tabla:

METODOLOGIA AGIL	VENTAJAS	DESVENTAJAS
SCRUM	<p>Scrum proporciona un enfoque ágil para la gestión de proyectos aumentando la probabilidad de su éxito.</p> <ul style="list-style-type: none"> <li>• Resultados significativos en desenvolvimiento.</li> <li>• Mayor intercambio de información</li> <li>• Centrarse en cumplir los objetivos y metas del proyecto</li> <li>• Identificación clara de las debilidades del equipo.</li> <li>• Identificación diaria de impedimentos y riesgos.</li> <li>• Claridad de los objetivos del proyecto.</li> <li>• Mejor comunicación y reuniones más rápidas y objetivas.</li> <li>• Mejor distribución del trabajo</li> <li>• Mejor proactividad, motivación, integración y compromiso del equipo.</li> <li>• Monitoreo y control descentralizado por</li> </ul>	<ul style="list-style-type: none"> <li>• Scrum puede decirle al equipo que el código debe probarse bien, pero no especifica cómo se realiza esa prueba.</li> <li>• Requiere una exhaustiva definición de las tareas y sus plazos. Cuando estos dos aspectos no se definen adecuadamente, Scrum se desvanece. Recuerda que la división del trabajo en cada etapa (y de éstas en tareas específicas) son la esencia de esta metodología.</li> <li>• Exige que quienes la utilicen cuenten con una alta cualificación o formación. No es una modalidad de gestión propia de grupos junior o que apenas estén en proceso de formación. Gran parte del éxito de Scrum radica en la experiencia que aportan los profesionales de los equipos, quienes por lo general acumulan años de experiencia.</li> </ul>

	<p>parte del equipo.</p> <ul style="list-style-type: none"> <li>• Planificación del equipo.</li> </ul>	
eXtreme Programming (XP)	<ul style="list-style-type: none"> <li>• Resultados significativos en desenvolvimiento en algunos proyectos.</li> <li>• XP especificaría el desarrollo impulsado por pruebas (TDD), una receta de cómo el equipo desarrolla el código mediante pruebas; TDD</li> </ul>	<ul style="list-style-type: none"> <li>• XP como una serie de piezas de rompecabezas que encajan como un reloj suizo, a pesar del hecho de que XP es supuestamente adaptable a una amplia variedad de proyectos. Algunas prácticas, aunque no agregan valor, son imposibles de eliminar, ya que son necesarias para mantener las otras en su lugar.</li> </ul>
Scrumban	<ul style="list-style-type: none"> <li>• No hay necesidad de cambiar una organización completa a Scrumban: un solo grupo dentro de una empresa puede usar Scrumban indefinidamente.</li> <li>• Adoptar Scrumban es simple: un equipo puede decidir que se mudarán a Scrumban por la mañana y estarán en funcionamiento con notas adhesivas y carteles esa tarde.</li> <li>• Scrumban es evolutivo; en ese sentido, puede estar más cerca de LPD que Scrum.</li> </ul>	<ul style="list-style-type: none"> <li>• No suele ser una buena combinación para grandes proyectos de desarrollo de productos. Puede ser una buena opción para actividades sostenibles y proyectos de desarrollo más pequeños.</li> </ul>
Hybrid Method Engineering (HME)	<ul style="list-style-type: none"> <li>• Puede ser extremadamente poderoso para el desarrollo de aplicaciones, especialmente porque Scrum tiene una fuerte alineación con Lean.</li> </ul>	<ul style="list-style-type: none"> <li>• Se centra en el desarrollo de aplicaciones y se encuentra en un alto nivel de abstracción, y no proporciona detalles necesarios de sus fases para su aplicación al desarrollo personalizado. Teléfonos móviles (adaptación del código base a bordo).</li> </ul>
Mobile-D	<ul style="list-style-type: none"> <li>• El modelo de negocio es dictado por el mismo cliente (El fabricante de celulares).</li> </ul>	<ul style="list-style-type: none"> <li>• Este enfoque es superficial y no está completamente definido para ser utilizado literalmente en la práctica.</li> <li>• Otros autores han sugerido mejoras adicionales y el modelo podría mejorarse aún más utilizando técnicas ágiles híbridas.</li> </ul>
MASAN	<ul style="list-style-type: none"> <li>• Se recomienda usar el MASAM, metodología para pequeñas empresas que se centran en el desarrollo de aplicaciones de software móvil.</li> </ul>	<ul style="list-style-type: none"> <li>• Los autores de esta metodología no han presentado un estudio de caso de una implementación real de esta metodología en un entorno del mundo real, por lo que aún no es clara para su uso.</li> </ul>

*Tabla 1. Cuadro comparativo de marcos de desarrollo ágil.*

## 2.1 Extreme Programming (XP)

Este marco de trabajo define cuatro variables para cualquier proyecto: costo, tiempo, calidad y alcance. Además, se especifica que, de estas cuatro variables, sólo tres de ellas podrán ser fijadas arbitrariamente por actores externos al grupo de desarrolladores (clientes).

Se distingue de otras metodologías por:

- Sus cortos ciclos de desarrollo,
- Su tendencia a la planificación incremental.
- Su capacidad de flexibilizar el horario de implementación de funcionalidades.
- Su confianza en pruebas automatizadas escritas por programadores, clientes y probadores.
- Su confianza en la comunicación oral, pruebas, y código fuente.
- Su confianza en diseño de procesos evolucionario que dure tanto como dure el sistema.
- Su confianza en la colaboración cercana de personas con talento ordinario activamente comprometidas.
- Su confianza en prácticas que trabaje con ambas instinto a corto plazo de los miembros de equipo e intereses a largo plazo del proyecto.

XP puede ser descrito como:

- Ligerero, solo se realiza lo necesario para crear un valor para el cliente.
- Una metodología basada en abordar restricciones de desarrollo de software.
- Puede trabajar con equipos de todos los tamaños.
- Se adapta a requisitos vagos o que cambian rápidamente.

XP es una disciplina de desarrollo de software que maneja el riesgo en todos los niveles del proceso de desarrollo de software. XP abarca cinco **valores** para guiar el desarrollo: comunicación, simplicidad, retroalimentación, coraje y respeto [3].

- Comunicación: la comunicación es muy importante en un equipo de desarrollo ya que permite que los problemas sean resueltos de mejor manera ahorrando tiempo ya que lo más probable es que cualquier persona del equipo de desarrollo ya tenga la solución que otra persona del equipo está buscando.
- Simplicidad: la simplicidad es uno de los valores más demandantes intelectualmente hablando de XP, generalmente la simplicidad es relativa al tiempo en que se desarrolla un proyecto y puede que los requerimientos de ahora no sean tan simples o complejos como los requerimientos de mañana por lo que es difícil preparar una solución simple a un problema.



- Retroalimentación: La retroalimentación en XP debe ser constante, tan pronto como se pueda recolectar la retroalimentación más rápido se podrá abordar el contexto del problema. En comparación con otros marcos de trabajo la retroalimentación en XP debe ser recolectada en horas o en días y no en semanas.
- Valentía: “El coraje de decir verdades, agradables o desagradables, fomenta la comunicación y la confianza. El coraje de descartar soluciones defectuosas y buscar otras nuevas fomenta la simplicidad. El coraje de buscar respuestas reales y concretas genera retroalimentación” [3].
- Respeto: El respeto es fundamental dentro de un equipo de trabajo si no hay respeto ninguno de los anteriores valores tiene sentido, ningún miembro del equipo es menos importante que el otro.

XP también se basa en principios fundamentales los cuales hacen que los valores cobren sentido, estos principios son:

- Humanidad: Recordar que el software está diseñado por humanos para humanos, por tanto, debe de adaptarse a la realidad de los clientes e interesados en el desarrollo del producto.
- Economía: Todo desarrollo de software que está orientado a obtener un beneficio económico debe de tomar en cuenta que la inversión que se hace ahora puede ser mayor mañana, por eso es recomendable utilizar un modelo de negocio pague por usar. Este modelo garantiza que los cambios futuros se realicen de manera sostenible.
- Beneficio mutuo: Cada actividad debe beneficiar a todos los involucrados.
- Autosimilitud: Se debe tomar en cuenta que un proceso que funciona se lo puede replicar a diferentes escalas, sin embargo, no siempre se podrá replicar, por lo tanto, es importante saber reutilizar procesos o porciones de código en donde sea posible. Una forma recomendable de aplicar XP al flujo de trabajo de un desarrollador es realizar primero las pruebas y luego el código. Esta práctica reduce el estrés y ahorra tiempo.
- Mejora: El enfoque de XP es que el software esté en constante mejora, cada día se debe mejorar de alguna manera, sin embargo, no se debe de parar un proceso solo porque no es lo suficientemente bueno, la mejora debe de ser continua siguiendo el flujo diario de trabajo.
- Diversidad: Los equipos de trabajo deben de ser diversos para juntar habilidades, actitudes y perspectivas para ver los problemas y las dificultades. Esto mejora la capacidad de resolver problemas.

- Reflexión: se refiere a que el equipo de trabajo siempre debe reflexionar acerca de su trabajo, no necesariamente se necesita hacer un ejercicio mental grande para poder reflexionar acerca del trabajo del equipo, pero si es necesario establecer métricas que ayuden a reflexionar acerca de cómo se está realizando el trabajo [3].
- Flujo: las prácticas de XP están orientadas a un flujo continuo de actividades. Por tanto, se deben de resolver los problemas que interrumpen el flujo y en cuanto sea posible se debe de regresar al flujo normal de trabajo.
- Oportunidad: lo que hace XP diferente de otros marcos de trabajo es que promueve a que el equipo vea cada dificultad como una oportunidad, para lo cual es importante que el equipo de trabajo que aplique XP aprenda a ver los problemas como oportunidades.
- Redundancia: en XP se debe considerar varias formas de resolver un problema, de esta manera si no se puede solucionar el problema de una forma siempre habrá otra. Si bien esta práctica puede ser tediosa ayuda a minimizar los errores y defectos que pueda tener un software.
- Fracaso: el fracaso cuando viene acompañado de aprendizaje siempre es bueno, si por alguna razón una historia de usuario es difícil se debe de realizar todas las maneras posibles de resolverlo y si todas fracasan al menos queda el aprendizaje y el aprendizaje es muy valioso en una organización.
- Calidad: se debe de buscar un balance entre entregar prototipos rápidamente y con una calidad mediocre a entregar prototipos con la mejor calidad posible, pero en más tiempo.
- Pequeños pasos: se deben implementar cambios de manera paulatina en vez de abordar grandes cambios en cada iteración.
- Responsabilidad aceptada: "La responsabilidad no puede ser asignada; solo puede ser aceptada". El líder del equipo no solo debe asignar responsabilidades si no que debe compartirlas [3].

Adicional a los valores se tiene las prácticas, por lo general las prácticas dependen de la situación, por otro lado, los valores no deben cambiar y los principios se deben aplicar cuando se cambia de dominio.

### **2.1.1 Prácticas Primarias**

En esta subsección se mostrará las practicas primarias las mismas que fueron tomadas de:

- Sentarse juntos

Trabajar en un espacio de trabajo abierto y que tenga módulos individuales pero cercanos que den la sensación de privacidad sin separarse del todo del equipo de trabajo, mantener el espacio entre módulos lo suficientemente cercano y abierto para que cualquier pueda hablar y ser escuchado por cualquier persona del equipo.

- Todo el equipo

Incluir en el equipo personas con todas las habilidades necesarias para que el proyecto tenga éxito. Las personas deben sentirse incluidas, escuchadas y que son parte integral del equipo.

- Espacio de trabajo informativo

El espacio de trabajo debe de mostrar a primera vista de que se trata el proyecto, de tal modo si alguna persona externa al proyecto entra en él, pueda saber grosso modo en que se está trabajando.

- Trabajo energizado

Se debe trabajar solamente la cantidad de horas en las que se es productivo. Trabajar en exceso solo lleva a tomar pésimas decisiones y a tener que refactorizar el código mal escrito en esas horas.

- Programación en parejas

Escribir todo el código que se va a llevar a producción con dos personas sentadas en la misma estación de trabajo. Esto permitirá que el código resultante tenga la menor cantidad de errores de diseño.

- Historias

Planificar el desarrollo del proyecto usando unidades de funcionalidad visibles al cliente, esas unidades deben provenir de parte del cliente y deben ser validadas por el equipo de desarrollo.

- Ciclo semanal

Se debe planificar una semana a la vez. Cada semana debe empezar con una reunión que aborde los siguientes temas:

- 1) El progreso que se ha tenido hasta la fecha.

- 2) La historia o historias de usuario escogidas por el cliente que se deben realizar esta semana
- 3) Sintetizar las historias en actividades.

Luego se deben realizar pruebas automatizadas, que corran cuando se completen las historias. Luego ocupar el resto de la semana codificando el código necesario para que las pruebas pasen con éxito.

- Ciclo Trimestral

Planificar el trabajo trimestralmente. Cada trimestre se debe reflejar que tan alineadas están las metas con el equipo, el proyecto y el progreso. Durante la planificación del trimestre se debe:

- Identificar cuellos de botella
- Iniciar reparaciones
- Planificar los temas o tema del trimestre.
- Escoger las historias que permitan cumplir esos temas.
- Enfocarse en lo general, donde el proyecto compagine con la organización.

- Holgura

Incluso en proyectos con cortos periodo de entrega, llenos de historias de usuario y con presupuesto reducido, se debe tener un tiempo de holgura para poder motivar al equipo. Si el equipo es muy conflictivo el cumplir con las promesas ayuda a que sea más unido.

- Construcción de diez minutos

Construir todo el sistema automáticamente y correr todas sus pruebas en 10 minutos. Normalmente este proceso toma varios minutos pero que funcione en no más de 10 minutos permite al equipo de desarrollo seguir trabajando en otros temas puntuales.

- Integración continua

Integrar y probar los cambios en no más de un par de horas. La programación es un problema de divide, conquista e integración. Mientras más tiempo tome este proceso, más impredecible y costoso se vuelve.

- Programación Test-First

Escribir pruebas que no pasen antes de escribir cualquier código ayuda en muchas maneras, por ejemplo:

- Permite enfocarse en el código. Si se requiere añadir más código, primero se debe escribir otra prueba
- Si por algún motivo es difícil de escribir una prueba, es señal de que tenemos un problema de diseño.
- Escribir código limpio que funcione y demostrando las intenciones con pruebas automatizadas, da a los compañeros del equipo un motivo para confiar.
- Queda más claro que hacer después ya sea escribir otra prueba o crear código que pase esa prueba.

- Diseño incremental

Cada día se debe trabajar en el diseño del sistema. El resultado es un sistema incremental que empieza siendo pequeño y crece tanto como se requiera sin costos exorbitantes.

### **2.1.2 Practicas Corolarias**

Estas prácticas solo se deben de implementar cuando se hayan masterizado las practicas primarias.

- Participación real del cliente

El objetivo de involucrar a los clientes es reducir la pérdida de esfuerzo poniendo a las personas con las necesidades en contacto directo con las personas que pueden cubrir esas necesidades.

- Implementación Incremental

Las implementaciones grandes requieren de mucho tiempo y recursos. Se debe encontrar una pequeña pieza de funcionalidad o un grupo limitado de datos manejables para ser implementados en producción.

- Continuidad del equipo

Mantener a los equipos efectivos unidos. Se debe tener en cuenta que los miembros de un equipo son muy importantes y que cuando se tiene un equipo con buena sinergia se debe conservarlo ya que se ha demostrado que rinde y lo seguirá haciendo.

- Encogiendo Equipos

Conforme la organización o empresa empieza a crecer es recomendable mantener varios equipos pequeños en lugar de uno solo grande. Esto permite que exista una mejor comunicación.

- Análisis de la causa raíz

Cada que se encuentre un error se debe solucionarlo y encontrar su causa raíz. A continuación, se detalla el proceso de cómo tratar con errores en XP:

1. Escribir una prueba automatizada que demuestre el defecto incluyendo el comportamiento deseado.
2. Escribir una prueba unitaria con el menor alcance posible que también reproduzca el error.
3. Corregir el sistema para que las pruebas funcionen. De no funcionar regresar al paso 2.
4. Una vez que el defecto se corrija descubrir el por qué se produjo y por qué no fue manejado. Empezar los cambios necesarios para prevenir este tipo de defectos en el futuro. Preguntar 5 veces por qué ocurre un problema.

- Código compartido

Cualquier persona en el equipo puede modificar el código del sistema a cualquier hora. Si el sistema presenta algún fallo que cualquiera del equipo lo pueda resolver y no aumenta el alcance del proyecto, se debe resolver lo antes posible.

- Codificar y Probar

Mantener solo el código y las pruebas como artefactos permanentes. Generar otros documentos para el código y las pruebas.

- Base de código único

Solo debe haber una fuente de código. Si se va a utilizar una rama secundaria combinarla lo antes posible y no permitir que dicha rama sobreviva más de unas pocas horas.

- Despliegue Diario

Todo nuevo software debe ser puesto en producción cada noche. Cualquier brecha entre lo que tiene el programador en su escritorio y lo que está en producción es un riesgo.

- Contrato de alcance negociado

Definir un contrato para el desarrollo de software que fije un tiempo, estimaciones y calidad. Negociar el alcance del proyecto en base a las estimaciones del equipo. Para reducir el riesgo se pueden escribir varios contratos cortos en lugar de uno largo.

- Pague por uso

Desarrollar sistema de pague por uso, se puede cargar al usuario un monto cada vez que utilice el sistema. "El dinero es la retroalimentación definitiva, no solo es concreto si no que se puede gastar" [3].

### **2.1.3 Equipo de XP**

- Testers

Ayudan a los clientes a escribir pruebas a nivel de sistema automatizadas y enseñan a los programadores sobre técnicas de testeo.

- Diseñadores de Iteración.

Escogen metáforas generales del sistema, escriben historias y evalúan el uso del sistema en producción para encontrar posibles nuevas historias. Trabajan con los clientes ayudándolos a escribir historias de usuario.

- Arquitectos

Buscan y ejecutan refactorizaciones a gran escala, escriben pruebas a nivel de sistema que pongan a prueba el sistema al máximo. Ellos direccionan la arquitectura del proyecto a lo largo de su evolución.

- Project Manager

Están encargados de enfatizar la comunicación entre el cliente y el equipo de desarrollo, así como con proveedores y el resto de la organización. Planificar en XP es una actividad no una fase.

- Product Manager

Escriben historias, seleccionan temas e historias en el ciclo trimestral, escogen historias en el ciclo semanal y responden a preguntas como historias que no se han especificado.

- Ejecutivos

Proveen al equipo de XP con valor, confianza y responsabilidad. Uno de sus trabajos es articular y mantener metas a largo plazo. Otra de sus funciones es monitorizar, empoderar y facilitar la mejora.

- Escritores Técnicos

El rol que cumplen las publicaciones técnicas es proveer retroalimentación temprana acerca de nuevas funcionalidades y crear relaciones cercanas con los usuarios. Se puede intentar tener la documentación lista la misma semana que las historias.

- Usuarios

Los usuarios en XP ayudan a escribir y obtener historias y tomar decisiones de dominio durante el desarrollo.

- Programadores

Los programadores en un equipo de XP estiman las historias y tareas, sintetizan las historias en tareas, escriben pruebas, escriben código para implementar funcionalidades, automatizan procesos de desarrollo tediosos y gradualmente mejoran el diseño del sistema.

Los programadores trabajan entre ellos, en parejas en el código de producción, así que necesitan tener buenas habilidades sociales y de relación [4].

#### **2.1.4 Ciclos de XP para el desarrollo de aplicaciones móviles**

La planificación de XP se debe de realizar semanalmente, teniendo en cuenta una reunión semanal donde se debe de abordar lo siguiente:

- El progreso hasta la fecha, incluyendo que tanto progreso se ha obtenido comparando con la semana anterior.
- Elección de una historia de usuario para implementar en la semana, de parte del cliente.



- Desglosar las historias en tareas y estimar el tiempo de estas.

La semana se debe empezar escribiendo pruebas automatizadas que corran cuando las historias se hayan completado. Luego el resto de la semana se debe enfocar en completar las historias para que pasen las pruebas previamente realizadas. La meta es tener un prototipo funcional al final de la semana [3].

- Planificar XP

Cada equipo XP tiene expertos del lado del cliente, quienes son responsables de tomar decisiones. Los programadores se encargan de proveer estimaciones y sugerencias, las cuales están ligadas con las prioridades de los clientes.

- Análisis

En lugar de sentarse a analizar en una fase del proyecto para definir requerimientos, los clientes se sientan junto al equipo a tiempo completo. Los clientes in situ son responsables de encontrar la manera de obtener requerimientos para el software.

- Diseño y codificación

Este Trabajo se lo realiza mediante Test Driven Development (TDD). Para completar este proceso los programadores trabajan en parejas con el objetivo de incrementar el poder mental para asegurarse de que por lo menos una persona tenga capacidad de analizar problemas complejos. Los programadores integran su código cada cierto tiempo con intervalos pequeños y se aseguran de que cada integración sea capaz de ser implementada en producción. También mantienen estándares de codificación y comparten su propio código. Cada programador tiene la capacidad de resolver problemas en el código sin importar quien lo escribió.

- Pruebas

“En XP todos los miembros del equipo contribuyen a la calidad del software” [4]. Se utiliza TDD para producir pruebas unitarias automatizadas y pruebas de integración.

Las pruebas que realiza el cliente ayudan a verificar que el desarrollo va de la mano con las necesidades del negocio. Los testers ayudan al equipo de desarrollo a entender en donde los esfuerzos focalizados están produciendo código de alta calidad. Cuando un tester encuentra un fallo o bug el equipo de desarrollo se encarga de encontrar la causa raíz, analizarla y considerar en cómo mejorar los procesos para evitar errores similares. Los testers también

se encargan de probar el rendimiento y la estabilidad del sistema. Los clientes utilizan esta información para decidir qué otras historias crear.

Cuando se encuentran fallos los programadores realizan pruebas automatizadas para demostrar que ya han sido resueltos.

- Despliegue

Los equipos de XP mantienen su código listo para ser desplegado al final de cada iteración. El despliegue a clientes reales se agenda de acuerdo con las necesidades del negocio. En las últimas semanas se genera documentación como manuales y diagramas que servirán para capacitar.

La siguiente tabla muestra los roles y las prácticas de cada miembro del equipo de XP:

<i>Practicas XP</i>	<i>Clientes in situ</i>	<i>Programadores</i>	<i>Testers</i>	<i>Entrenadores</i>
<b>Pensamiento</b>				
<i>Programación en parejas</i>		✓		
<i>Trabajo Energizado</i>	⊗	⊗	⊗	✓
<i>Espacio de trabajo informativo</i>	⊗	⊗	⊗	✓
<i>Análisis causa raíz</i>	⊗	⊗	⊗	✓
<i>Retrospectivas</i>	⊗	⊗	⊗	✓
<b>Colaboración</b>				
<i>Confianza</i>	⊗	⊗	⊗	✓
<i>Sentarse Juntos</i>	⊗	⊗	⊗	✓
<i>Involucramiento real del cliente</i>	⊗	⊗	⊗	✓
<i>Lenguaje omnipresente</i>		✓		
<i>Reuniones de pie</i>	⊗	⊗	⊗	✓
<i>Estándares de codificación</i>		⊗		✓
<i>Demostración de iteración</i>	✓	✓	⊗	⊗
<i>Reportar</i>	⊗	⊗	⊗	✓
<b>Lanzamiento</b>				
<i>“Realizado Realizado”</i>	✓	✓	✓	✓
<i>Sin bugs</i>	⊗	✓	✓	✓
<i>Control de versiones</i>	⊗	✓	⊗	
<i>Compilación de 10 minutos</i>		✓		
<i>Integración Continua</i>		✓		
<i>Propiedad colectiva del código</i>		✓		
<i>Documentación</i>	⊗	⊗		✓
<b>Planificación</b>				
<i>Visión</i>	✓			✓
<i>Plan de lanzamientos</i>	✓			✓
<i>El juego de la planificación</i>	⊗	⊗		✓
<i>Gestion de riesgo</i>	⊗	⊗	⊗	✓
<i>Plan de iteración</i>	⊗	✓		✓
<i>Holgura</i>		⊗		✓

Historias	●	●	●	✓
Estimación		✓		
<b>Desarrollo</b>				
Requerimientos incrementales	✓		✓	✓
Pruebas del cliente	✓	✓	✓	
Test-Driven Development		✓	●	
Refactorización		✓		
Diseño simple		✓		
Diseño y Arquitectura incremental		✓		
Soluciones acertadas		✓		
Optimización de rendimiento	●	✓		
Pruebas Exploratorias				✓

Tabla 2. Importancia de cada practica correspondiente a cada rol.

✓ = se debe estudiar la practica cuidadosamente

● = Usted estará involucrado con esta práctica. Estudiarlo será útil, pero no necesario [3]

## 2.2 Test Driven Development (TDD)

Es una práctica de programación bastante utilizada en XP, consiste básicamente en empezar un desarrollo desde la creación de pruebas unitarias, dichas pruebas deben ser analizadas en base al código que se va a desarrollar. Para ello se debe hacer un análisis previo de qué funcionalidades se van a incluir dentro del sprint, estas funcionalidades tienen que ser probadas al final del sprint y pasar todos los Test que se hayan escrito al inicio del sprint.

Una de las ventajas de esta práctica de programación es que se pueden identificar tempranamente funcionalidades más importantes que otras y de igual forma se puede identificar si es que un proyecto está bien estructurado directo mal estructurado implica unas pruebas mal estructuradas. Lo que nos permite tener un código más limpio y garantizar que funcione correctamente desde el inicio de la concepción de la idea. También esta práctica nos permite poder hacer cambios en el código y garantizar de qué no se afecte la funcionalidad inicial.

## Capítulo 3 – Desarrollo de la aplicación

En este capítulo se abordan todos los temas relacionados con el desarrollo del sistema. Se incluye la planificación general del proyecto, el desarrollo de este y las pruebas respectivas para garantizar su correcto funcionamiento.

### 3.1 Fase de exploración

El cliente ha solicitado una aplicación móvil que esté disponible en iOS y en Android la cual permita a las compañías de transporte interprovincial poder ofrecer sus viajes y gestionar la venta de asientos para cada unidad. El cliente es una compañía de desarrollo de software que está interesada en ofrecer la plataforma previamente mencionada a empresas de transporte.

Inicialmente se planteó el hecho de poder comprar pasajes para buses interprovinciales para de esta manera evitar traslados innecesarios a boleterías y terminales. En esta fase de exploración se identificó un mercado potencial en personas con un teléfono celular y un plan de datos quienes dispondrían de mayor facilidad para adquirir boletos en cooperativas de buses interprovinciales, de igual manera sería una ventaja para los dueños de flotas de buses ya que de esta manera podría gestionar mejor las ventas de sus boletos y el uso de sus unidades.

#### 3.1.1 Stakeholders

De acuerdo con datos obtenidos del INEC se sabe que en el Ecuador existe un gran número de personas que poseen un dispositivo móvil y que tienen un plan de datos, los cuales están constantemente movilizándose, por tanto, existe un gran porcentaje de personas a las cuales este proyecto busca abarcar para que dichos usuarios tengan una alternativa rápida, segura y económica de adquirir sus boletos sin necesidad de ir a un terminal a hacer largas filas y tener que soportar aglomeraciones.

Por lo tanto, debido a los datos obtenido las personas involucradas en este proyecto y a las cuales se podrían beneficiar del mismo son:

- Usuarios finales
- Cooperativas de transporte de buses interprovinciales
- Conductores de buses
- Empresa administradora del sistema
- Empresa que administra el sistema de cobro.

Inicialmente el proyecto se planteó por una empresa, pero debido a la situación económica que vivió el país en 2020 debido a la pandemia ya no pudo continuar con el proyecto, no obstante, puede ser de gran interés para cualquier empresa que se encuentre en el rubro del transporte terrestre.

### **3.1.2 Alcance del proyecto**

El proyecto partió con la propuesta inicial de la empresa Geckotech que buscaba facilitar al usuario la compra de pasajes de buses interprovinciales mediante el uso de herramientas digitales como son las aplicaciones móviles. Los requerimientos se obtuvieron a partir de varias reuniones que se tuvieron con la empresa mencionada anteriormente, en las primeras reuniones se propuso la idea de manera general lo cual era un modelo bastante primitivo de cómo sería el gestor de pasajes. Luego de varias reuniones se encontró que no mejor forma para acercarse a los usuarios sería a través de una aplicación móvil ya que se trataba de un servicio de transporte es necesario tener a la mano un dispositivo que se puede llevar a cualquier parte en donde se pueda tener control no sólo de los pasajes sino también de la ubicación del autobús y de las rutas más cercanas. Luego de varios análisis y de varias reuniones con diferentes cooperativas de transporte como trans esmeraldas nos dimos cuenta de qué un sistema centralizado era lo mejor debido a que permitiría a la empresa seguir enfocada en el core del su negocio sin tener que involucrarse mucho en temas de desarrollo. Después de llevar a cabo varios elementos a la idea original, se descubrió que algunos modelos planteados inicialmente no iban a ser necesarios en este proyecto debido a que tomarían demasiado tiempo. Por otra parte, luego de empezar el desarrollo y durante el mismo se encontraron varios inconvenientes los cuales comprometieron el alcance del proyecto, pero a su vez también permitieron ver oportunidades de mejora que no se contemplaron al inicio de este. Luego de todo un proceso de prueba y error se definieron las principales funcionalidades del sistema que servirían para tener un producto mínimo viable, para lo cual se definió la creación de dos aplicaciones móviles disponibles tanto para el sistema operativo iOS como para Android: una para usuarios y otra para conductores, que cumplan los siguientes requerimientos:

Aplicación móvil para usuarios, que deberá permitir:

- Verificar rutas disponibles
- Reservar asientos para una o varias personas
- Revisar el horario de viajes
- Ver un historial de viajes
- Entregar sus boletos virtuales

Aplicación móvil para conductores que deberá permitir:

- Ver las rutas asignadas
- Ver disponibilidad de asientos
- Registrar los boletos
- Llevar un control de rutas
- Organizar de mejor manera sus viajes con el visualizador de horarios

Con estos requerimientos, se buscó que el sistema permita a los usuarios escoger asientos, rutas, el horario cuando desean viajar, y que los conductores puedan ver en tiempo real los boletos que se hayan vendido y cuales asientos están disponibles.

Para el fin académico de la presente tesis se planteó que para la comprobación de la reserva se emita un código numérico que llegaría al correo electrónico registrado por el usuario y que la gestión de todo el sistema solo se administre para una cooperativa de buses interprovinciales ya que el proyecto se extendía mucho más en caso de tener más cooperativas de transporte.

Para un mayor entendimiento se procesaron los requerimientos de ambas aplicaciones y los mismos se encuentran descritos en la Tabla 3, el código RF indica que se trata de un requerimiento funcional, mientras que el código RNF los marca como requerimientos no funcionales.

En cuanto a la arquitectura del sistema se pensó inicialmente en una arquitectura serverless pero luego de un análisis más profundo y de comprobar lo difícil que sería realizar pruebas unitarias de un servicio que está en constante cambio y que implica un llamado específico para cada sistema operativo se dictaminó que se complicaría cumplir con el marco de desarrollo y se decidió utilizar una arquitectura cliente servidor más tradicional la cual se puede observar en la Figura 2.

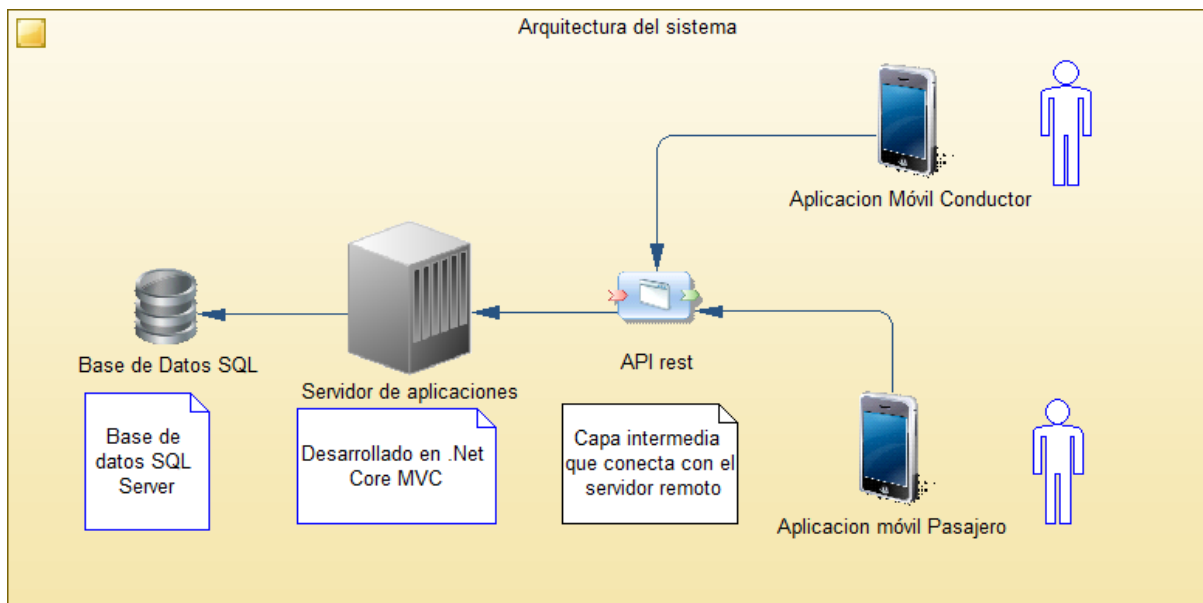
Código	Requerimiento
Requerimientos Funcionales	
RFP1	La aplicación debe permitir seleccionar la ciudad de origen y un sector para ser recogido
RFP2	La aplicación debe permitir seleccionar una ciudad de destino y un sector en la ciudad de destino
RFP3	La aplicación debe permitir que el usuario seleccione la fecha de partida
RFP4	La aplicación debe permitir seleccionar el número de pasajes
RFP5	La aplicación mostrará un esquema con los asientos disponibles.
RFP6	La aplicación debe permitir al pasajero seleccionar los asientos a ocupar.
RFP7	La aplicación debe permitir seleccionar el punto de abordaje y el punto de desembarque.
RFP8	La aplicación debe permitir ingresar la siguiente información de los pasajeros: Nombre, edad, número de cédula y correo electrónico.
Requerimientos no funcionales	

RNFP1	La aplicación debe ser desarrollada para los sistemas operativos Android e iOS
RNFP2	Los usuarios posibles son Conductor y Pasajero
RNFP3	El sistema debe estar disponible 24/7
RNFP4	El sistema debe ser en tiempo real
RNFP5	El sistema debe permitir aumentar nuevos módulos cuando así se requiera
RNFP6	La aplicación móvil debe ser intuitiva y con un diseño amigable

*Tabla 3: Requerimientos Funcionales para la aplicación del pasajero*

Código	Requerimiento
Requerimientos Funcionales	
RFC1	La aplicación debe permitir conocer los asientos vendidos y los disponibles
RFC2	La aplicación debe permitir seleccionar asientos y marcarlos como reservados.
RFC3	La aplicación debe permitir ingresar la siguiente información de los conductores: Nombre, número de cédula y correo electrónico.
RFC4	La aplicación debe permitir ver las reservas por pasajero
RFC5	La aplicación debe permitir ver las rutas asignadas a una unidad
RFC6	La aplicación debe permitir ver el punto de subida y el punto de llegada de cada pasajero.
Requerimientos Funcionales	
RNFC1	La aplicación debe ser desarrollada para los sistemas operativos Android e iOS
RNFC2	Los usuarios posibles son Conductor y Pasajero
RNFC3	El sistema debe estar disponible 24/7
RNFC4	El sistema debe ser en tiempo real
RNFC5	El sistema debe permitir aumentar nuevos módulos cuando así se requiera
RNFC6	La aplicación móvil debe ser intuitiva y con un diseño amigable

*Tabla 4: Requerimientos Funcionales para la aplicación del conductor*



*Fig. 2. Arquitectura general del sistema*

## 3.2 Fase de inicialización

Debido a que se ha tomado como modelo a XP la planificación es una actividad recurrente en cada ciclo del mencionado modelo, sin embargo, si se realizó una planificación general la cual involucra de manera general las historias de usuario. Se consideraron los siguientes planes para el proyecto:

- Plan de lanzamiento (Release Planning)
- Plan de iteración (Iteration Planning)

El plan de lanzamiento involucra a las historias de usuario y el alcance del proyecto como tal, en cambio el plan de iteración involucra más a los ciclos mensuales y trimestrales de todo el proyecto.

### 3.2.1 Configuración del proyecto

Inicialmente se planteó el uso de arquitectura serverless con la tecnología Firebase con la finalidad de facilitar y reducir la cantidad de código que tenía que ser escrita y que el desarrollo sea más fluido y fácil de manejar; sin embargo, no se pudo implementar esta tecnología pues presentó problemas de compatibilidad entre Firebase y Xamarin Forms.

Entre los principales problemas que se encontraron en Firebase fueron los siguientes:

- La integración con Xamarin.Forms no es tan completa y requiere que se escriba bastante código, en los proyectos dedicados de Android y iOS.
- Para probar la implementación de Firebase en iOS obligadamente se necesita un dispositivo físico que cuente con dicho sistema operativo debido a un problema de certificados.
- Debido a que muchas partes del código debían hacerse en los proyectos específicos de cada plataforma las pruebas unitarias se complicaban cada vez más.
- El implementar nuevas funcionalidades se volvió un trabajo de investigación tedioso y poco fructífero debido a la poca documentación que se tiene de la integración con Xamarin.

Por tal motivo se optó por desarrollar también el back-end con una tecnología compatible con Xamarin Forms, la cual es ASP.net core., esta permitió que fluyera de mejor manera el consumo de servicios y permitió organizar de mejor manera el esquema de base de datos.

Para notificar a los clientes de qué se realizó la reserva de los pasajes se utilizó un sistema de mensajes de correo electrónico de esta manera se envió un IDE el cual se va a utilizar



mediante la aplicación del conductor para validar la compra del boleto. Para esto se utilizó un servidor de correos que está alojado en un servidor remoto para poder hacer pruebas. De igual forma el servidor de aplicaciones el cual aloja todo el backend se estableció en un servidor remoto.

Para el almacenamiento de datos se utilizó una base de datos relacional muy utilizada en .Net la cual es Microsoft SQLServer. Por otra parte, para el desarrollo de todo el sistema se utilizó el entorno de desarrollo Visual Studio 2019 luego de la actualización que se realizó en el 2021, se empezó a utilizar la versión 2022 ya que incluye mejoras en el rendimiento y mejor compatibilidad con el framework Xammarin.Forms.

Se utilizó el lenguaje de programación C# el cual es un lenguaje multi propósito desarrollado por Microsoft, ampliamente utilizado en desarrollo de sistemas complejos como videojuegos y aplicaciones web o móviles empresariales.

Cómo patrón de diseño se utilizó el modelo MVVM el cual hace referencia a Modelo, Vista y Vista Modelo, la ventaja de usar este framework es que permite separar la lógica del negocio de la vista que se presentará de cara al usuario, lo que facilita la mantenibilidad del código y también permite realizar pruebas unitarias de una manera más sencilla.

Debido a la necesidad de poder geolocalizar las rutas de los buses, se planteó el uso de una tecnología open source como lo es Open Street Map, sin embargo, no tiene mucho soporte en Xammarin y las pocas librerías no tienen la documentación suficiente, por lo que se optó por utilizar Google maps tanto para iOS como para Android. asumiendo que por ser un proyecto académico no tendría mayor costo, pero ya en el ámbito productivo si se debe considerar un costo.

Para la gestión de usuarios y el registro en el sistema de rutas y demás funcionalidades se implementó toda la lógica en un servidor remoto el cual a su vez hace uso de ASP.net.

La gestión de tickets se la realiza mediante tokens que se deben validar desde el lado de la aplicación del conductor.

Para las pruebas unitarias se utilizó el framework Nunit el cual está diseñado específicamente para Xammarin, tiene ciertas facilidades.

Para las pruebas de interfaz se utilizó una librería propia de Xammarin, la cual es limitada para el caso de la versión de visual studio community que es la que se utilizó.

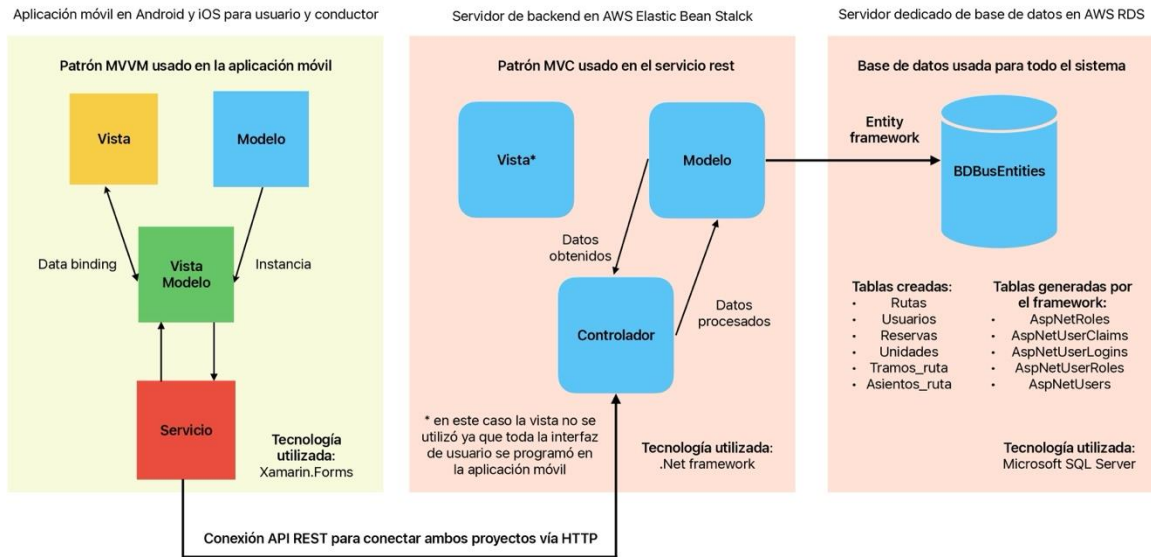


Fig. 3. Arquitectura de software del sistema

### 3.2.2. Diseño de la base de datos

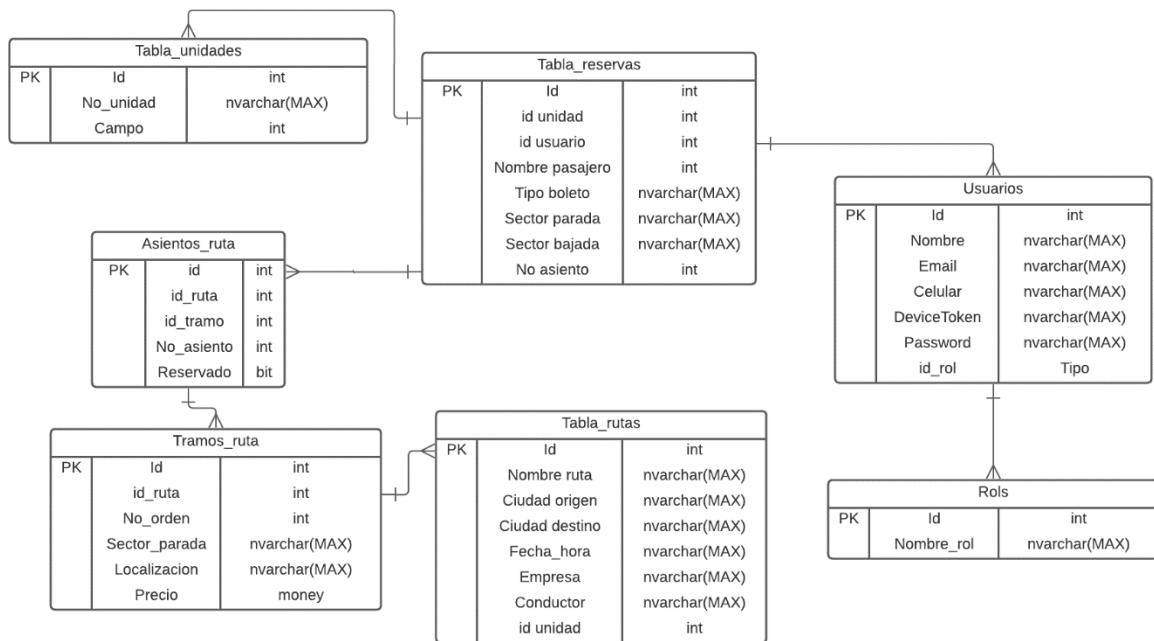


Fig. 4. Modelo Entidad Relación de la base de datos

### 3.2.3 Plan de Lanzamiento

El plan de lanzamiento consta de los siguientes pasos o componentes a liberar:

1. Se trabaja en el análisis de la historia de usuario.
2. Se implementa un pequeño esquema de base de datos.
3. Se desarrolla el esquema de base de datos mediante Entity Framework para facilitar el desarrollo.

4. Se implementa toda la lógica en el backend para acceder a los datos.
5. Se implementa el código necesario para consumir los datos del backend desde la aplicación móvil.
6. De ser necesario se implementa la pantalla de ingreso de datos por parte del cliente.
7. Finalmente se realizan las pruebas de unitarias o de interfaz de ser el caso.

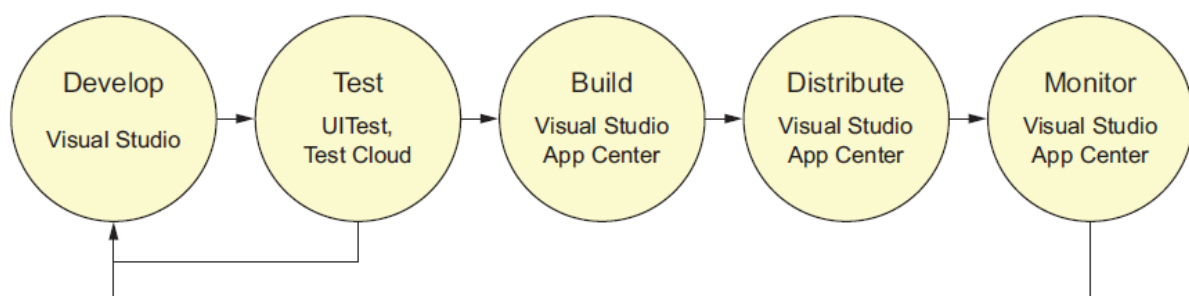
Debido a que se planteó un enfoque de desarrollo basado en TDD las pruebas de interfaz o las pruebas unitarias se deberían realizar al inicio del desarrollo, pero lamentablemente fue muy difícil adaptarse a este esquema de desarrollo por lo que en ciertas ocasiones se empezó desde el esquema de base de datos hasta finalizar con el tema de las pruebas unitarias o de interfaz.

En muchas partes del desarrollo los entregables fueron simplemente esquemas de bases de datos ya que hubo bastantes complicaciones con el desarrollo.

La metodología de XP no se pudo implementar en su totalidad debido a que algunas prácticas como Pair Programming implican en el equipo más de un desarrollador sin embargo el resto de las prácticas y valores de XP fueron de gran utilidad para que el desarrollo fluyera de una manera aceptable.

Para el plan de lanzamientos se considerará el ciclo descrito en la Figura 2. Debido a que la aplicación está pensada solo para fines académicos solo se llegará hasta la fase denominada Build pero solo se generarán los archivos de instalación localmente.

Para las pruebas de interfaz se utilizará la herramienta que incluye Xamarin por defecto y debido a limitaciones de licencia solo se realizaran pruebas que no involucren sensores específicos del dispositivo como GPS, giroscopio, etc.



*Fig. 5. Ciclo de lanzamiento de aplicaciones móviles propuesto por Microsoft para el desarrollo con Xamarin.Forms [5]*

### 3.2.4 Plan de Iteración

Para este plan se tuvo en cuenta el esquema de metodología XP el cual nos dice que en cada sprint o ciclo de desarrollo debe existir por lo menos una funcionalidad mínima del sistema,

de esta manera se garantiza de qué en cada ciclo de desarrollo exista un entregable por lo menos que sirva para mostrar al cliente el avance del proyecto, para nuestro caso particular los ciclos de desarrollo tuvieron algunas modificaciones y se extendió el proyecto más de lo planteado debido a que las tecnologías que se pensó utilizar al inicio presentaron varios problemas de compatibilidad como la dificultad de realizar pruebas unitarias a servicios cuya implementación implica considerar cambios constantes y diferencias entre plataformas.

### **3.3 Fases de producción y pruebas**

Para obtener las siguientes historias de usuario se tuvo que realizar varias reuniones con el cliente para poder llegar al estado en el que se encuentran actualmente, en el caso inicial estaba para más de 30 historias de usuario, pero el tiempo planteado para el desarrollo de este proyecto de titulación no alcanzaba para todas esas funcionalidades por lo que al final se dejaron las más importantes.

Luego de un análisis preliminar se separaron las diferentes funcionalidades en varios Sprint ya que XP maneja un esquema incremental al igual que en Scrum se dividió el desarrollo en varias iteraciones las cuales se detallan más adelante en esta sección.

Para cada fase de desarrollo se tomaron en cuenta el desarrollo del backend y del frontend donde el backend hace referencia al servidor de aplicaciones el cual se encarga de gestionar las peticiones que se realizan a la base de datos.

#### **3.3.1. Fase de evaluación de riesgos en el desarrollo de software**

Es aconsejable que en las fases iniciales del desarrollo de un proyecto se identifiquen los riesgos que pueden existir en el desarrollo del software para reducir la probabilidad de que sucedan; o el impacto que podría producir si ocurriesen, o, si es posible, los dos a la vez.

Se puede definir riesgo como: “combinación de la probabilidad de un suceso y de su consecuencia (International Organization for Standardization. International Electrotechnical Commission, 2009). Efecto de la incertidumbre sobre la consecución de los objetivos (Asociación Española de Normalización y Certificación, 2010)”. [6]

Para medir el riesgo se emplea la fórmula:

$$I * P = R$$

Donde:

I: Impacto presente en cada riesgo.

P: Probabilidad presente en cada riesgo.

R: Riesgo presente en un proyecto.

Las variables Impacto y Probabilidad pueden tomar diferentes valores; la ecuación queda plasmada de esta forma:

$$Im * Pm = R.$$

Donde:

Im: Impacto en el valor dado presente en cada riesgo.

Fm: Frecuencia en el valor dado presente en cada riesgo.

El índice m puede tomar valores como: insignificante, bajo, moderado, alto, muy alto [6].

Para evaluar los riesgos en el desarrollo de la aplicación en cada historia de usuario, se definió mediante una matriz que relaciona el impacto de la actividad y la probabilidad de ocurrencia durante el desarrollo como se puede ver en la tabla 7. Matriz de evaluación de riesgos desarrollo de la aplicación.

Clase	Impacto <b>RIESGO -</b> / <b>OPORTUNIDAD +</b>	Ranking
Insignificante	<b>Sin efecto</b> / <b>SIN EFECTO</b>	1
Bajo	<b>Es leve (se puede solventar en sitio)</b> / <b>BAJO IMPACTO</b>	2
Moderado	<b>Afecta, pero se puede solventar en un tiempo determinado</b> / <b>PUEDA DAR RESULTADOS</b>	3
Alto	<b>Afecta la ejecución</b> / <b>IMPACTO ALTO</b>	4
Muy alto	<b>No permite realizar la operación</b> / <b>IMPACTO INMINENTE</b>	5

Tabla 5. Ranking de impacto para evaluación de riesgos.

Clase	Probabilidad de Ocurrencia	Ranking
Improbable	Nunca ha sucedido, su probabilidad de ocurrencia es muy remota.	1
Remoto	Sería una coincidencia rara que suceda.	2
Ocasional	Sucede de manera esporádica	3
Moderado	Es posible que ocurra.	4

Frecuente	Suele ocurrir siempre.	5
-----------	------------------------	---

*Tabla 6. Ranking de probabilidades para evaluación de riesgos*

Según el ranking de impacto y probabilidad dadas en las tablas 5 y 6; se completa la matriz de riesgos para cada historia de usuario:

### MATRIZ DE EVALUACIÓN DE RIESGOS DESARROLLO DE APLICACIÓN

FECHA DE EVALUACIÓN			13/08/2020		
No.	DESCRIPCION	IMPACTO	PROBABILIDAD	RIESGO	Riesgo en desarrollo
1	HU1.- Ingreso al sistema	Alto	Frecuente	20	ALTO
2	HU2.- Registro en el sistema	Alto	Ocasional	15	ALTO
3	HU3.- Gestión de viajes	Bajo	Frecuente	10	MEDIO
4	HU4.- Búsqueda de rutas	Alto	Frecuente	20	ALTO
5	HU5.- Visualización de datos de la reserva	Alto	Frecuente	20	ALTO
6	HU6.- Selección de parada inicial	Moderado	Frecuente	15	ALTO
7	HU7.- Selección de asientos	Alto	Frecuente	20	ALTO
8	HU8.- Pago de pasajes	Alto	Frecuente	20	ALTO
9	HU9.- Ingreso al sistema (conductor)	Alto	Frecuente	20	ALTO
10	HU10.- Registro en el sistema (conductor)	Alto	Frecuente	20	ALTO
11	HU11.- Visualización de unidades (conductor)	Alto	Frecuente	20	ALTO
12	HU12.- Visualización de rutas por unidad (conductor)	Alto	Frecuente	20	ALTO
13	HU13.- Visualización de reservas (conductor)	Alto	Frecuente	20	ALTO
16	HU14.- Puntos de abordaje (conductor)	Alto	Frecuente	20	ALTO
17	HU15.- Visualización de asientos	Alto	Frecuente	20	ALTO

*Tabla 7. Matriz de evaluación de riesgos desarrollo de la aplicación*

### 3.3.2 Primera fase

Las historias de usuario detalladas a continuación fueron minuciosamente preparadas en base a los requerimientos funcionales y no funcionales de la aplicación. Estas se levantaron comprendiendo los requerimientos que buscaba inicialmente la empresa Geckotech en la reunión inicial y las cinco reuniones siguientes, posterior a esto las historias de usuario se fueron definiendo en base a la observación de las necesidades actuales que tiene el negocio del transporte interprovincial.

Para la aplicación de Usuario:

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 1	<b>Usuario:</b> Pasajero
<b>Nombre Historia:</b> Ingreso al sistema	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder iniciar sesión en la aplicación móvil con su número de teléfono y con una contraseña.	
<b>Observación:</b> Solo los usuarios que tenga un número de celular y una contraseña válidos podrán acceder al sistema	

*Tabla 8. Historia de usuario ingreso al sistema*

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 2	<b>Usuario:</b> Pasajero
<b>Nombre Historia:</b> Registro en el sistema	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder registrarse en el sistema con los siguientes datos: Nombre, edad, teléfono móvil, cédula, correo electrónico.	
<b>Observación:</b> Se deben validar la cédula, el teléfono y el correo mediante envío de un email de confirmación	

*Tabla 9. Historia de usuario Registro en el sistema*

Para la aplicación de Conductor:

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 9	<b>Usuario:</b> Conductor
<b>Nombre Historia:</b> Ingreso al sistema	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador Responsable:</b> Fernando Carrasco	

<b>Descripción:</b> El usuario debe poder iniciar sesión en la aplicación móvil con su número de teléfono y con una contraseña de mínimo 8 caracteres
<b>Observación:</b> Solo los usuarios que tenga un número de celular y una contraseña válidos podrán acceder al sistema

*Tabla 10. Historia de usuario Ingreso al sistema (Conductor)*

HISTORIA DE USUARIO	
<b>Número:</b> 10	<b>Usuario:</b> Conductor
<b>Nombre Historia:</b> Registro en el sistema	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder registrarse en el sistema con los siguientes datos: Nombre, edad, teléfono móvil, cédula, correo electrónico.	
<b>Observación:</b> Se deben validar la cédula, el teléfono y el correo mediante envío de un email de confirmación, también se debería poder escoger la compañía y la unidad asignada	

*Tabla 11. Historia de usuario Registro en el sistema (Conductor).*

Se realizaron las pruebas unitarias por pantalla. Para la pantalla de inicio de sesión se realizaron los casos de prueba descritos en las tablas 12, 13, 14 y 15:

Campo	Entrada	Salida Esperada
Correo Electrónico	fernando@correo.com	Ingreso Exitoso
Contraseña	abc12345	Ingreso Exitoso

*Tabla 12. Caso de prueba 1 para el inicio de sesión*

Campo	Entrada	Salida Esperada
Correo Electrónico		Entidad Inválida
Contraseña		Entidad Inválida

*Tabla 13. Caso de prueba 2 para el inicio de sesión*

Campo	Entrada	Salida Esperada
Correo Electrónico	.email@correo.com	Entidad Inválida
Correo Electrónico	email@correo..com	Entidad Inválida
Correo Electrónico	abc..123@correo.com	Entidad Inválida

*Tabla 14. Caso de prueba 3 para el inicio de sesión.*



Campo	Entrada	Salida Esperada
Contraseña	1	Entidad Inválida
Contraseña	abc12	Entidad Inválida
Contraseña	12345	Entidad Inválida
Contraseña	12345678	Entidad Inválida

*Tabla 15. Caso de prueba 4 para el inicio de sesión*

Para la pantalla de registro de usuario se plantearon los casos de prueba descritos en las tablas 16, 17, 18, 19, 20.

Campo	Entrada	Salida Esperada
Nombre y Apellido	Fernando	Entidad Creada
Número de Teléfono	0993677882	Entidad Creada
Fecha de Nacimiento	25/05/1982	Entidad Creada
Correo Electrónico	fernando@correo.com	Entidad Creada
Contraseña	abc12458	Entidad Creada
Confirmar Contraseña	abc12458	Entidad Creada

*Tabla 16. Caso de prueba 1 para la pantalla de registro de usuario.*

Campo	Entrada	Salida Esperada
Nombre y Apellido		Entidad inválida
Número de Teléfono		Entidad inválida
Fecha de Nacimiento		Entidad inválida
Correo Electrónico		Entidad inválida
Contraseña		Entidad inválida
Confirmar Contraseña		Entidad inválida

*Tabla 17. Caso de prueba 2 para la pantalla de registro de usuario.*

Campo	Entrada	Salida Esperada
Correo Electrónico	.email@correo.com	Entidad Inválida
Correo Electrónico	email@correo..com	Entidad Inválida
Correo Electrónico	abc..123@correo.com	Entidad Inválida

*Tabla 18. Caso de prueba 3 para la pantalla de registro de usuario.*

Campo	Entrada	Salida Esperada
Contraseña	abc12345	Entidad Creada
Confirmar Contraseña	abc22345	Entidad Inválida

*Tabla 19. Caso de prueba 4 para la pantalla de registro de usuario.*

Campo	Entrada	Salida Esperada
-------	---------	-----------------

Contraseña	1	Entidad Inválida
Contraseña	abc12	Entidad Inválida
Contraseña	12345	Entidad Inválida
Contraseña	12345678	Entidad Inválida
Confirmar Contraseña	1	Entidad Inválida
Confirmar Contraseña	abc12	Entidad Inválida
Confirmar Contraseña	12345	Entidad Inválida
Confirmar Contraseña	12345678	Entidad Inválida

*Tabla 20. Caso de prueba 5 para la pantalla de registro de usuario.*

Para la pantalla de recuperación de contraseña se realizaron los casos de prueba descritos en las tablas 21, 22 y 23.

Campo	Entrada	Salida Esperada
Correo Electrónico	fernando@correo.com	Envío Exitoso

*Tabla 21. Caso de prueba 1 para la pantalla de recuperación de contraseña.*

Campo	Entrada	Salida Esperada
Correo Electrónico		Entidad inválida

*Tabla 22. Caso de prueba 2 para la pantalla de recuperación de contraseña.*

Campo	Entrada	Salida Esperada
Correo Electrónico	.email@correo.com	Entidad Inválida
Correo Electrónico	email@correo..com	Entidad Inválida
Correo Electrónico	abc..123@correo.com	Entidad Inválida

*Tabla 23. Caso de prueba 3 para la pantalla de recuperación de contraseña.*

Para esta primera fase se tomó en cuenta el tipo de autenticación que requería el usuario se determinó que la mejor forma de autenticarse fuera por correo electrónico y contraseña de más de 8 caracteres para asegurar un nivel de seguridad aceptable.

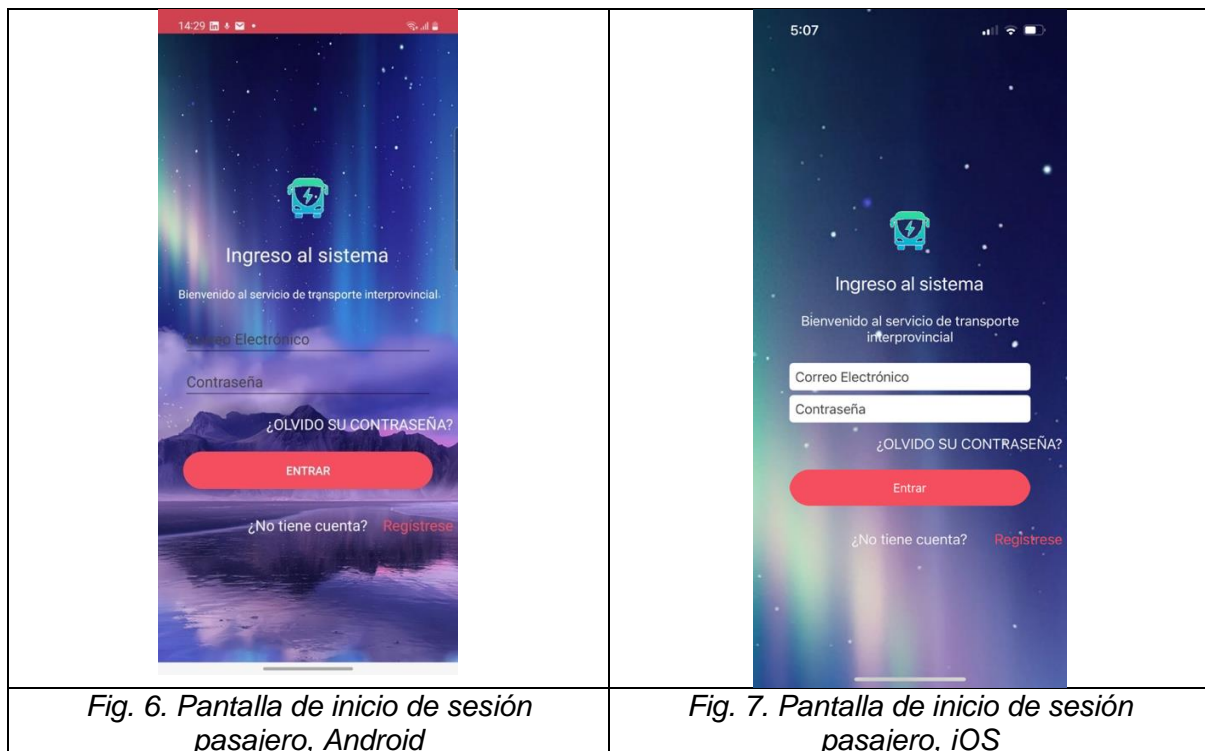
Cómo esquema de autenticación se tomó en cuenta el uso de doble factor debido a que es un sistema que se van a realizar pagos, pero luego se analizó de mejor manera y se determinó que la implementación de doble factor involucraría un aumento de tiempo en el proyecto el

cual sólo por motivos académicos no va a contar con un gestor de pagos por ende se retiró el uso del doble factor para la autenticación.

Para el tema de autenticación se utilizó una tabla en la cual se visualiza el nombre de usuario y adicionalmente la contraseña.

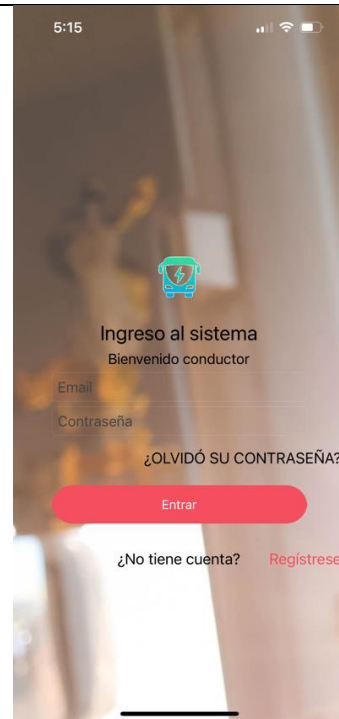
Para esta primera parte también se tomó en cuenta el esquema de notificaciones inicialmente se planteó implementar un esquema de envío de correo, sin embargo; esto implicaba implementar un esquema de segundo factor al final se decidió que no debería llegar ninguna notificación y que simplemente se redirigiera directamente a la pantalla principal de la aplicación.

Luego de varios análisis la pantalla de inicio de sesión quedó de la siguiente forma:





*Fig. 8. Pantalla de inicio de sesión conductor, Android*

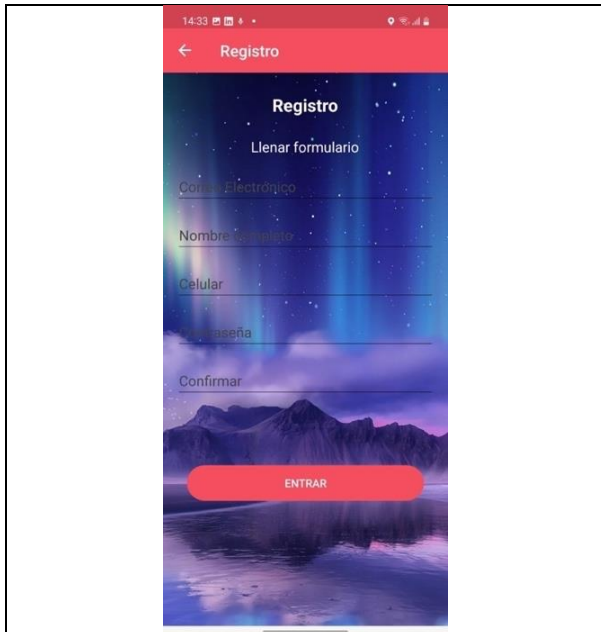


*Fig. 9. Pantalla de inicio de sesión conductor, iOS*

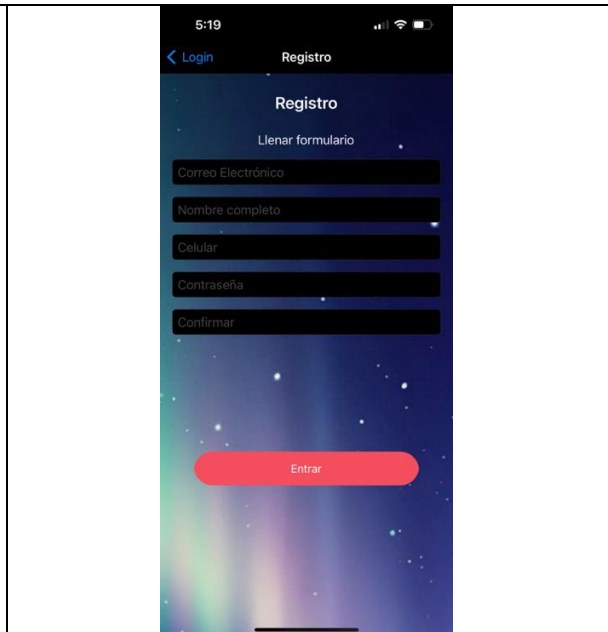
Para que el usuario pueda registrarse en la aplicación dentro de la pantalla de inicio de sesión se implementó un botón el cual le permite registrarse en la aplicación. Una vez que el usuario de clic en ese botón le permitirá ingresar varios valores mediante un formulario el cual consta de los siguientes campos:

- Nombre completo
- Correo electrónico
- Fecha de nacimiento
- Número celular

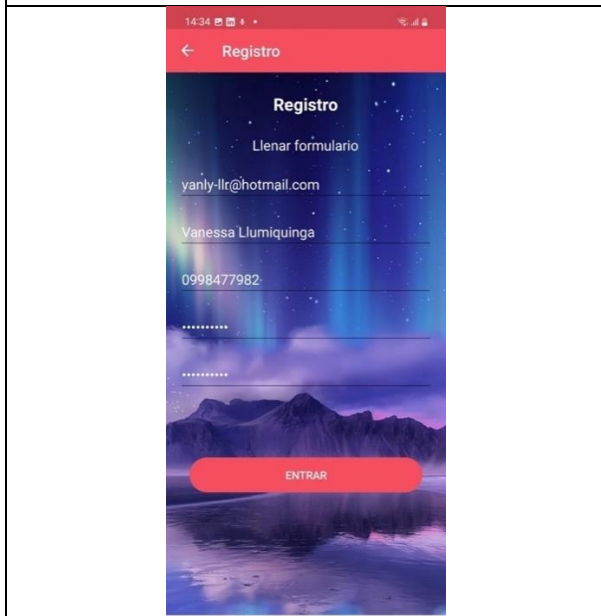
También se contempló el caso de qué los usuarios olviden su contraseña, para lo cual el usuario deberá ingresar su correo electrónico y le llegará a este la contraseña.



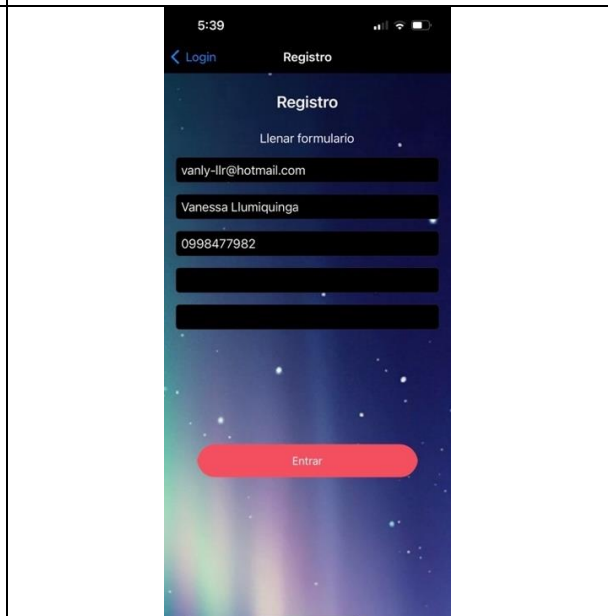
*Fig. 10. Pantalla registro de pasajero, Android*



*Fig. 11. Pantalla registro de pasajero, iOS*



*Fig. 12. Pantalla registro de pasajero con datos, Android*



*Fig. 13. Pantalla registro de pasajero con datos, iOS*

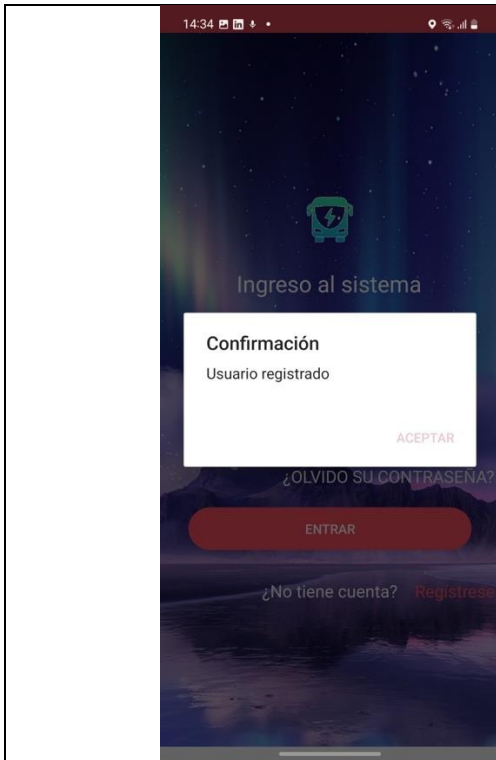


Fig. 14. Pantalla de confirmación de registro de pasajero, Android

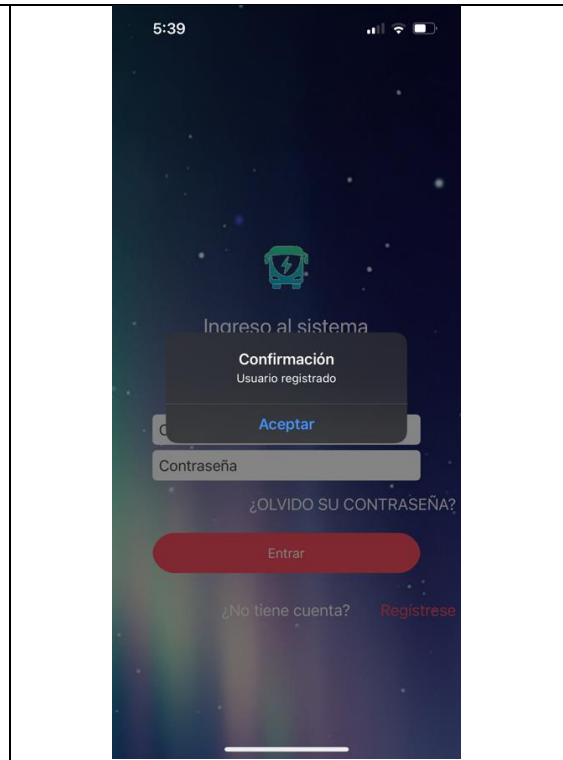


Fig. 15. Pantalla de confirmación de registro de pasajero, iOS

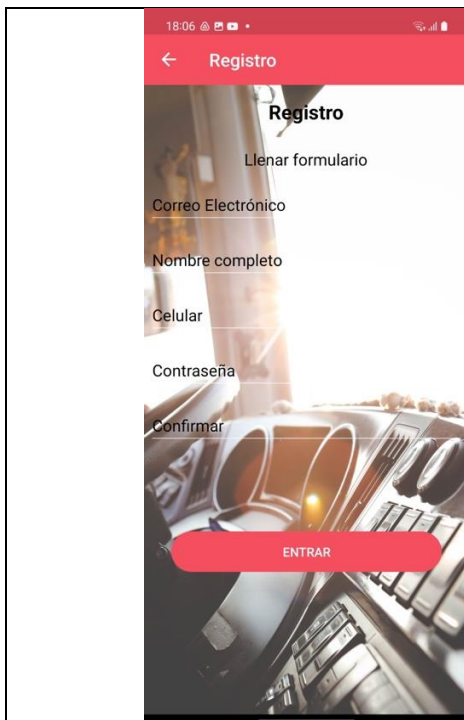


Fig. 16. Pantalla registro de conductor Android

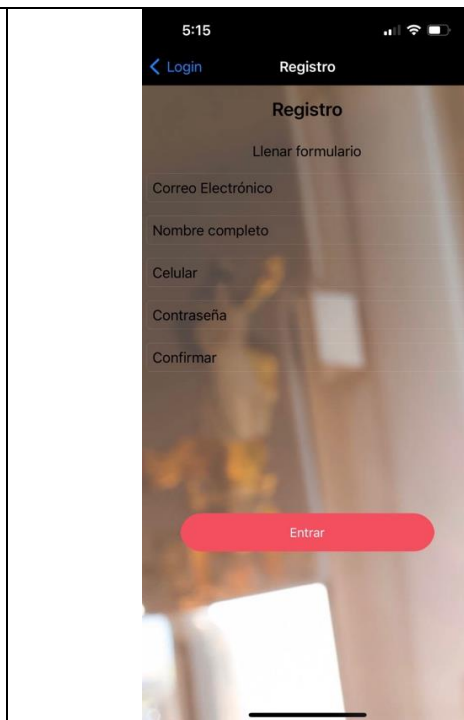


Fig. 17. Pantalla registro de conductor, iOS

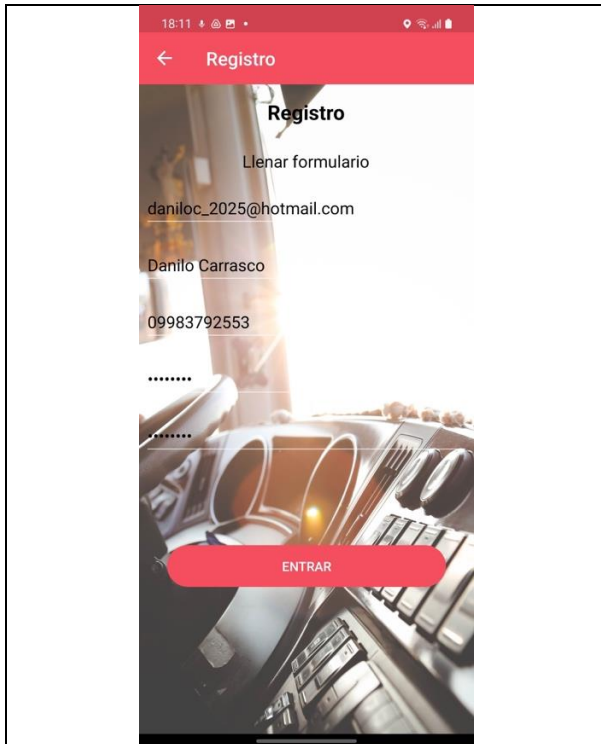


Fig. 18. Pantalla registro de conductor con datos, Android

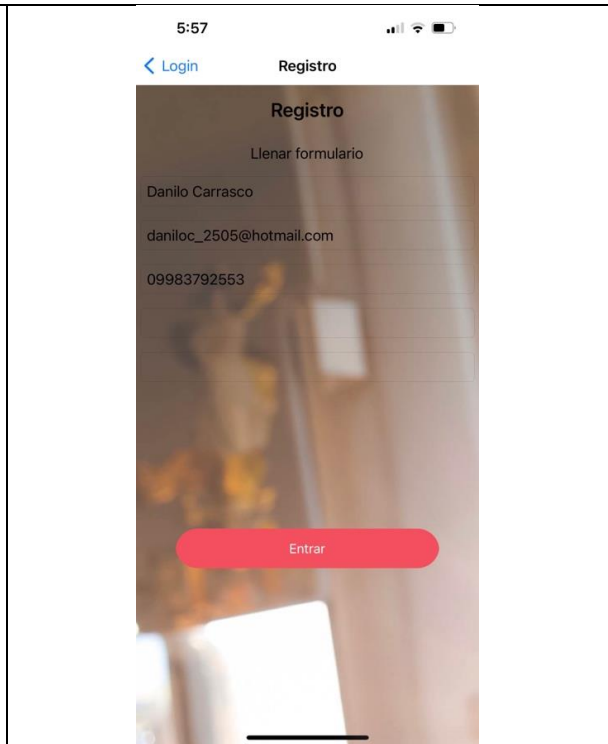


Fig. 19. Pantalla registro de conductor con datos, iOS

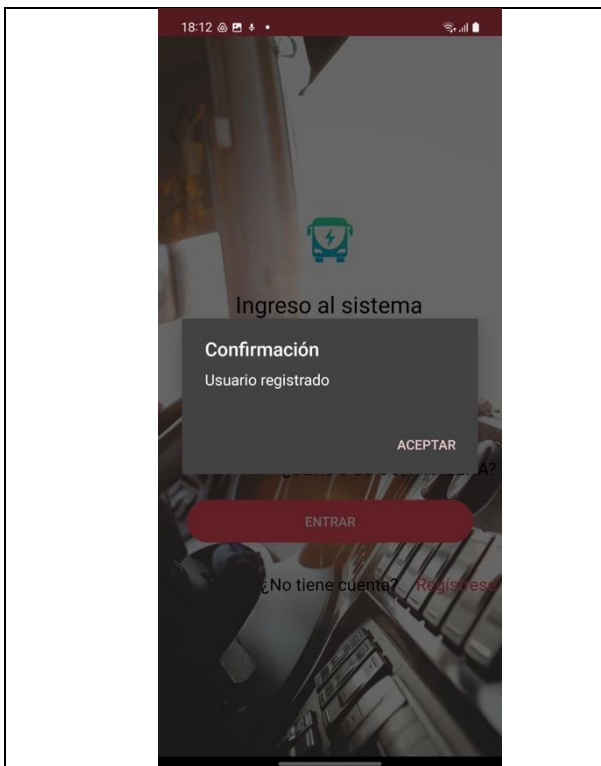


Fig. 20. Mensaje de confirmación de registro de conductor, Android

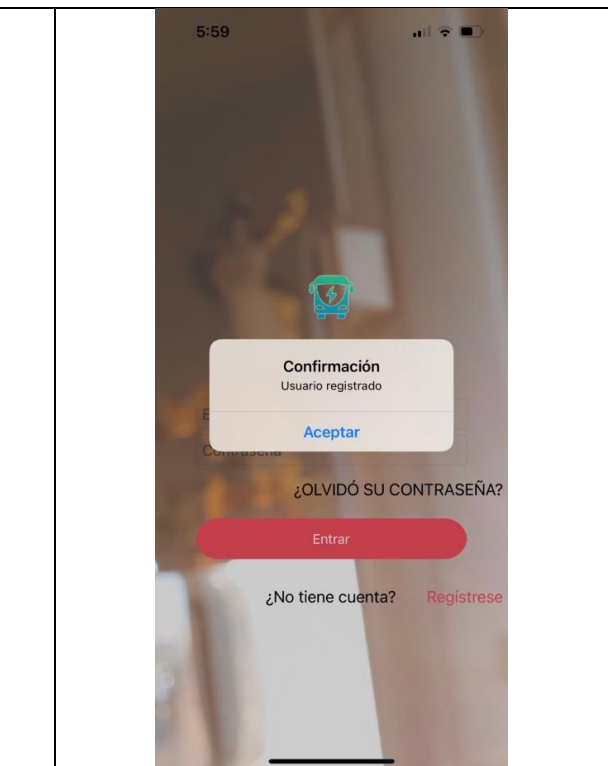


Fig. 21. Mensaje de confirmación de registro de conductor, iOS



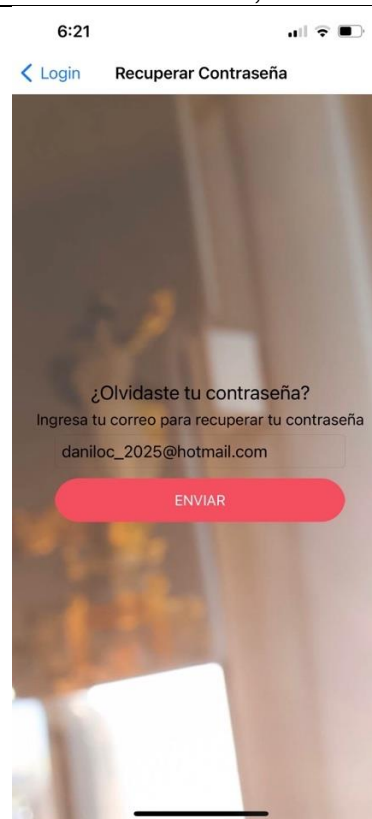
*Fig. 22. Pantalla olvido de contraseña de conductor, Android*



*Fig. 23. Pantalla olvido de contraseña de conductor, iOS*



*Fig. 24. Pantalla olvido de contraseña de conductor con datos, Android*



*Fig. 25. Pantalla olvido de contraseña de conductor con datos, iOS*



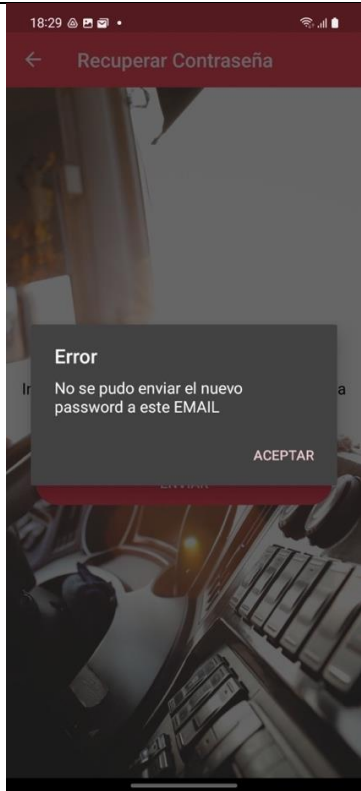


Fig. 26. Mensaje de error al recuperar la contraseña de conductor, Android

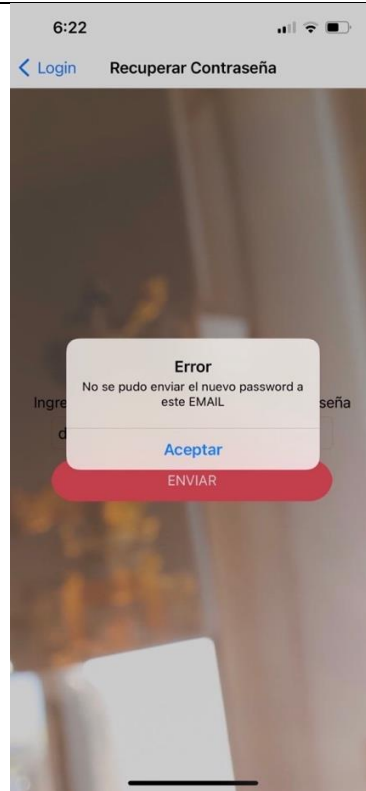


Fig. 27. Mensaje de error al recuperar la contraseña de conductor, iOS



Fig. 28. Pantalla olvido de contraseña de pasajero, Android

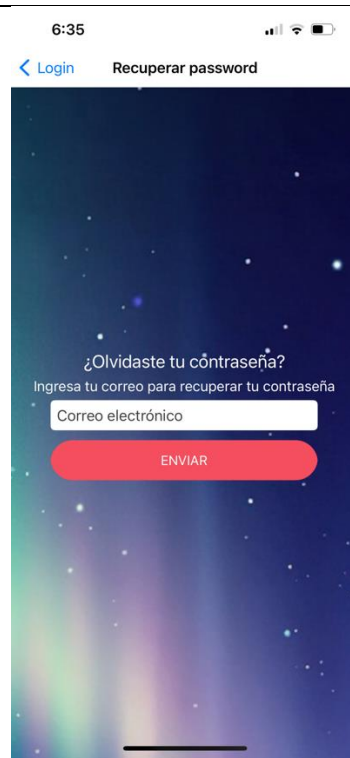
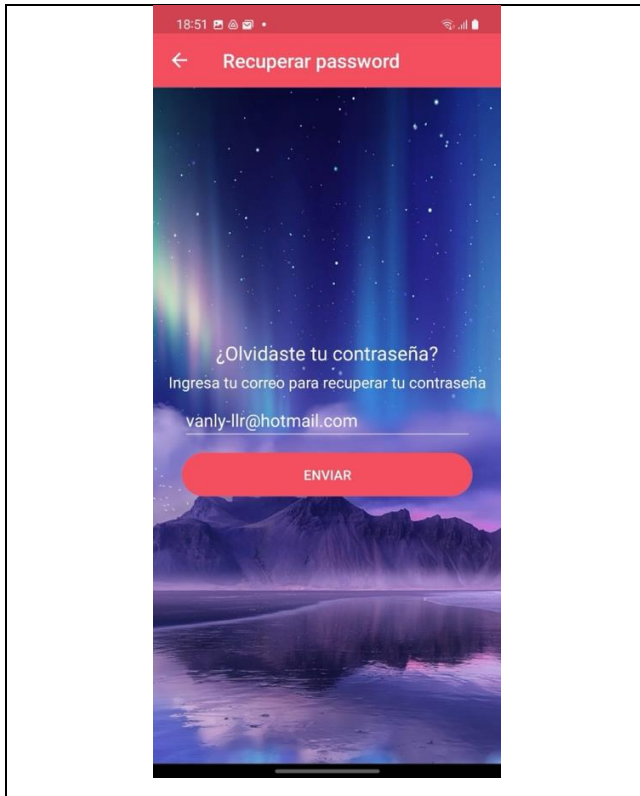
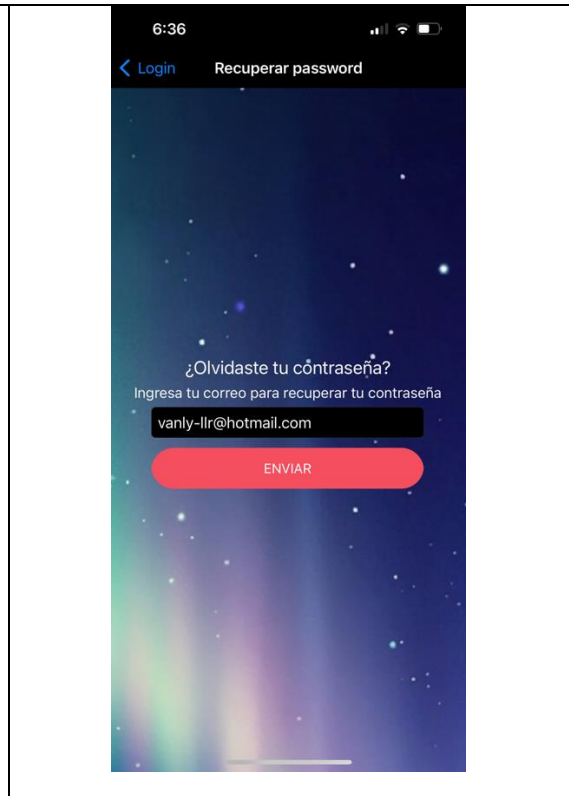


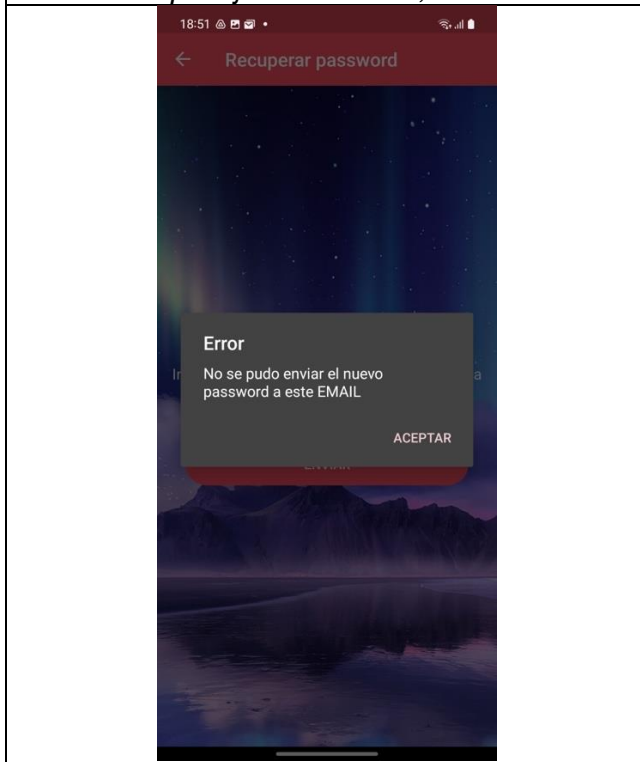
Fig. 29. Pantalla olvido de contraseña de pasajero, iOS



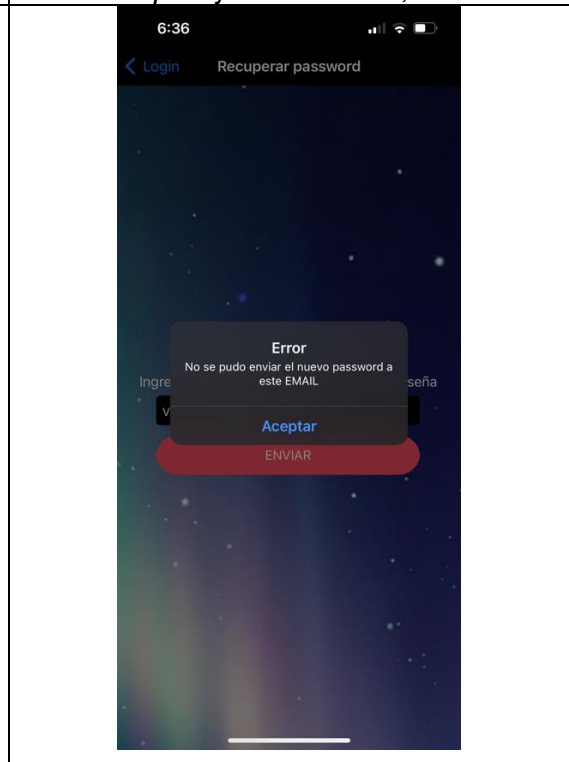
*Fig. 30. Pantalla olvido de contraseña de pasajero con datos, iOS*



*Fig. 31. Pantalla olvido de contraseña de pasajero con datos, iOS*



*Fig. 32. Mensaje de error al recuperar la contraseña de pasajero, Android*



*Fig. 33. Mensaje de error al recuperar la contraseña de pasajero, iOS*

Luego de pasar exitosamente todas las pruebas unitarias se dio por finalizada la primera fase. Para este punto la autenticación es común tanto para la aplicación del conductor como la aplicación del pasajero.

### 3.3.2 Segunda fase

Las historias detalladas a continuación fueron minuciosamente preparadas en base a los requerimientos funcionales y no funcionales de la aplicación. Se definieron las siguientes historias de usuario para la segunda fase:

Para la aplicación de Usuario:

HISTORIA DE USUARIO	
<b>Número:</b> 3	<b>Usuario:</b> Pasajero
<b>Nombre Historia:</b> Gestión de viajes	
<b>Prioridad en Negocio:</b> Baja	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos Estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder escoger entre buscar un viaje o ver el historial de sus viajes.	
<b>Observación:</b> Ninguna	

*Tabla 24. Historia de usuario gestión de viajes*

HISTORIA DE USUARIO	
<b>Número:</b> 4	<b>Usuario:</b> Pasajero
<b>Nombre Historia:</b> Búsqueda de rutas	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder buscar una ruta la cual debe estar parametrizada en la base de datos. Solo se muestran las rutas que tengan fecha y hora posterior al tiempo actual.	
<b>Observación:</b> Solo se validarán la cédula y el teléfono	

*Tabla 25. Historia de usuario búsqueda de rutas*

Para la aplicación de Conductor:

HISTORIA DE USUARIO	
<b>Número:</b> 11	<b>Usuario:</b> Conductor
<b>Nombre Historia:</b> Visualización de unidades	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto

<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder visualizar las unidades de transporte disponibles en una lista	
<b>Observación:</b> Debe de haber un buscador para acceder más rápido a la unidad deseada.	

*Tabla 26. Historia de conductor visualización de unidades*

Se realizaron las pruebas unitarias por pantalla. Para la pantalla de inicio se realizaron los casos de prueba descritos en las tablas 27, 28, 29,30, 31,32,33 y 34:

Elemento	Acción	Salida Esperada
Botón buscar viaje	Dar clic	Pantalla de búsqueda de rutas

*Tabla 27. Caso de prueba 1 para la Pantalla de inicio (usuario).*

Elemento	Acción	Salida Esperada
Botón mis viajes	Dar clic	Pantalla de historial de viajes

*Tabla 28. Caso de prueba 2 para la Pantalla de inicio (usuario).*

Elemento	Acción	Salida Esperada
Botón de atras	Dar clic	Pantalla de inicio

*Tabla 29. Caso de prueba 1 para la Pantalla búsqueda de rutas*

Elemento	Acción	Salida Esperada
Buscador	Ingresar: Esmeraldas	Rutas disponibles para destinos que contengan la palabra Esmeraldas
Buscador	Ingresar: ""	No permite ingresar esos caracteres
Buscador	Ingresar: "1234"	No permite ingresar esos caracteres
Buscador	Ingresar: 😊 😞 😏 😬	No permite ingresar esos caracteres
Buscador	Ingresar: ESMERALDAS	Rutas disponibles para destinos que contengan la palabra ESMERALDAS
Buscador	Ingresar: esmeraldas	Rutas disponibles para destinos que contengan la palabra esmeraldas
Buscador	Ingresar: # \$ & * ( ) ' " . : ? +	No permite ingresar esos caracteres

*Tabla 30. Caso de prueba 2 para la Pantalla búsqueda de rutas.*

Elemento	Acción	Salida Esperada
Tarjeta de ruta	Dar clic	Pantalla de información de la ruta

*Tabla 31. Caso de prueba 3 para la Pantalla búsqueda de rutas.*

Elemento	Acción	Salida Esperada
Buscador	Ingresar: "1001"	Unidades disponibles con el número 1001
Buscador	Ingresar: ""	No permite ingresar esos caracteres
Buscador	Ingresar: 😊 😞 😏 😬	No permite ingresar esos caracteres
Buscador	Ingresar: ESMERALDAS	No permite ingresar esos caracteres
Buscador	Ingresar: Esmeraldas	No permite ingresar esos caracteres
Buscador	Ingresar: #\$&*()".¿?+	No permite ingresar esos caracteres

*Tabla 32. Caso de prueba 1 para la Pantalla de gestión de unidades*

Elemento	Acción	Salida Esperada
Tarjeta de unidad	Dar clic	Pantalla de visualización de rutas por unidad

*Tabla 33. Caso de prueba 2 para la Pantalla de gestión de unidades.*

Elemento	Acción	Salida Esperada
Botón de cerrar sesión	Dar clic	Pantalla de inicio de sesión

*Tabla 34. Caso de prueba 1 para la Pantalla cambio de unidad*

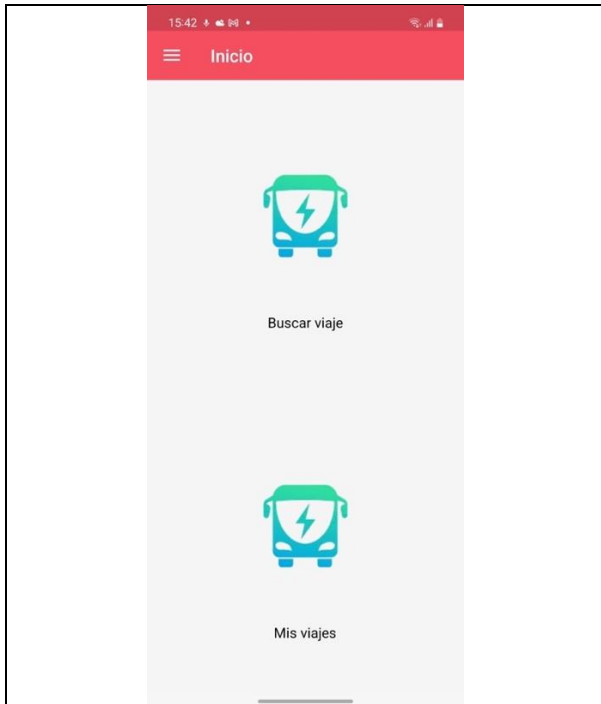
Para esta fase del proyecto se tomó en cuenta varios factores entre ellos para la aplicación del pasajero se tomarán en cuenta los siguientes:

- Lugar de destino del cliente
- Lugar de origen
- Rutas disponibles

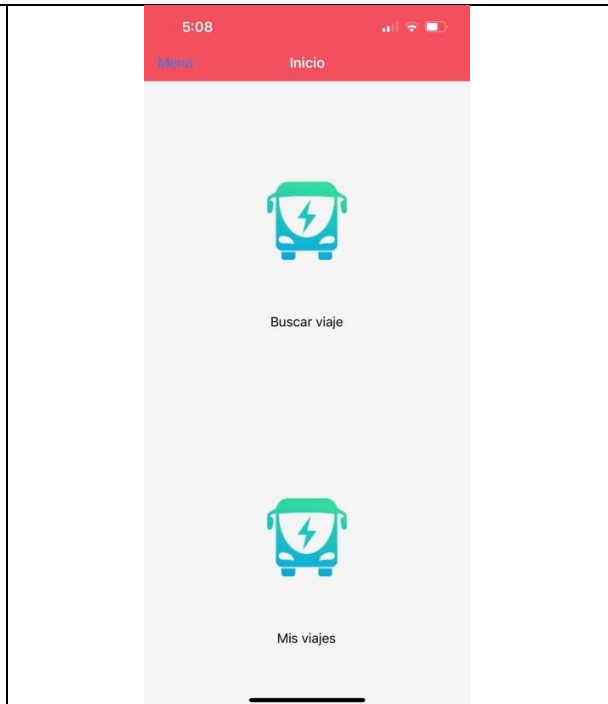
En la Figura 6 se puede apreciar la pantalla inicial de la aplicación móvil del pasajero una vez se hayan ingresado las credenciales. El botón "Buscar viaje" le permite al usuario seleccionar el viaje que desea realizar, por otro lado, el botón "Mis viajes" le permite visualizar un historial de todos los viajes realizados.

En la Figura 7 se puede apreciar la pantalla de búsqueda de rutas, donde se puede realizar una búsqueda rápida de la ruta que se desea realizar. Dentro de cada tarjeta se pueden apreciar los siguientes datos de cada ruta:

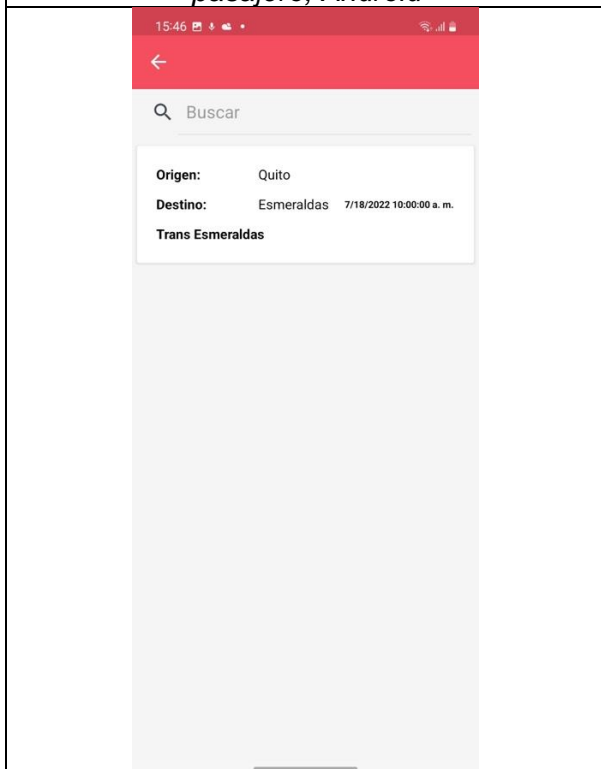
- Origen: se refiere al punto de partida de la ruta
- Destino: se refiere al punto de llegada de la ruta
- Nombre de la cooperativa: para este caso es "Trans-Esmeraldas"
- Fecha y hora de salida



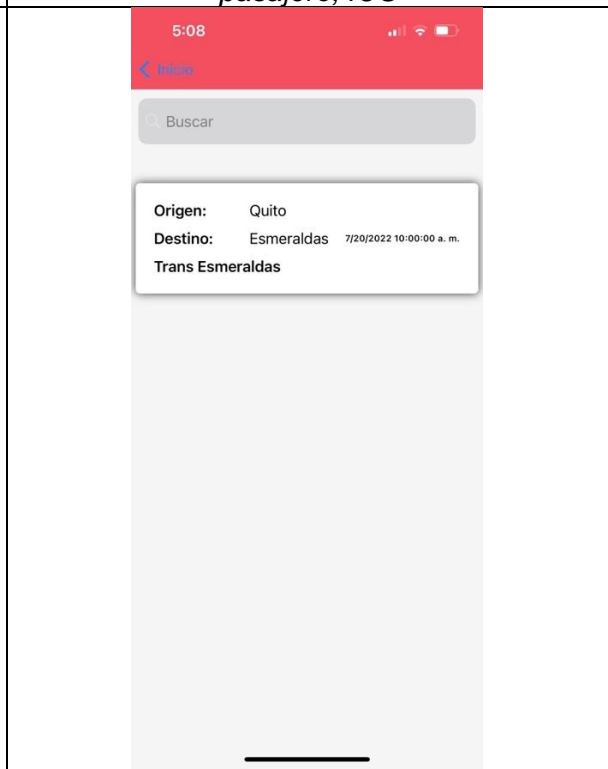
*Fig. 34. Pantalla de gestión de viajes - pasajero, Android*



*Fig. 35. Pantalla de gestión de viajes - pasajero, iOS*



*Fig. 36. Pantalla de búsqueda de rutas - pasajero, Android*



*Fig. 37. Pantalla de búsqueda de rutas - pasajero, iOS*

Para la aplicación del conductor se tomó en cuenta la gestión de unidades, debido a que el sistema está diseñado para una sola compañía de transporte la pantalla inicial muestra el

número de unidad. En el buscador de la pantalla se puede acceder fácilmente a la unidad deseada.

Tomando en cuenta todas estas variables, la interfaz gráfica de búsqueda de rutas se puede apreciar en la Figura 35 y 36.

Por otra parte, luego de seleccionar una unidad se despliega una vista con pestañas donde cada pestaña nos muestra una pantalla diferente, las pestañas se dividen de la siguiente manera:

- Reservas: En esta pantalla se lista todas las reservas realizadas en la unidad seleccionada.
- Tramos: En esta pantalla se listan los tramos que realizó y realizará la unidad.
- Asientos: En esta pantalla se muestra una gráfica al usuario la cual representa el número de asientos de la unidad, los asientos ocupados y asientos disponibles. Para identificar a los asientos ocupados se utiliza el color rojo, por defecto cuando se tienen asientos libres se los identifica con color blanco.

Para esta funcionalidad se utilizó un tipo de vista especial llamada tabbed page la cual es una página que alberga otras páginas del tipo content page, la ventaja de utilizar este tipo de vista es que el comportamiento en cada plataforma es diferente y se ajusta a los estándares de diseño tanto de Android como de iOS. Otra de las ventajas es que el usuario puede ver en una sola pantalla muchas funcionalidades del sistema sin tener que esperar mucho tiempo de carga y sin tener que ir y volver entre pantallas.

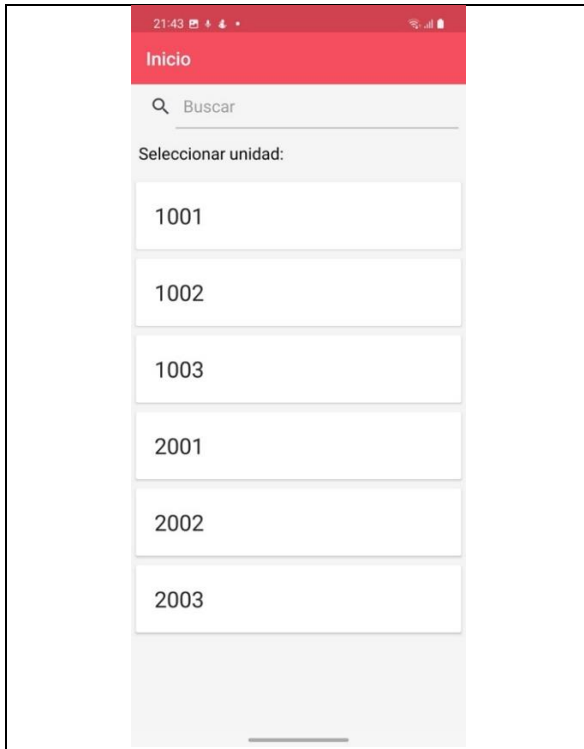


Fig. 38. Pantalla de gestión de unidades-conductor, Android

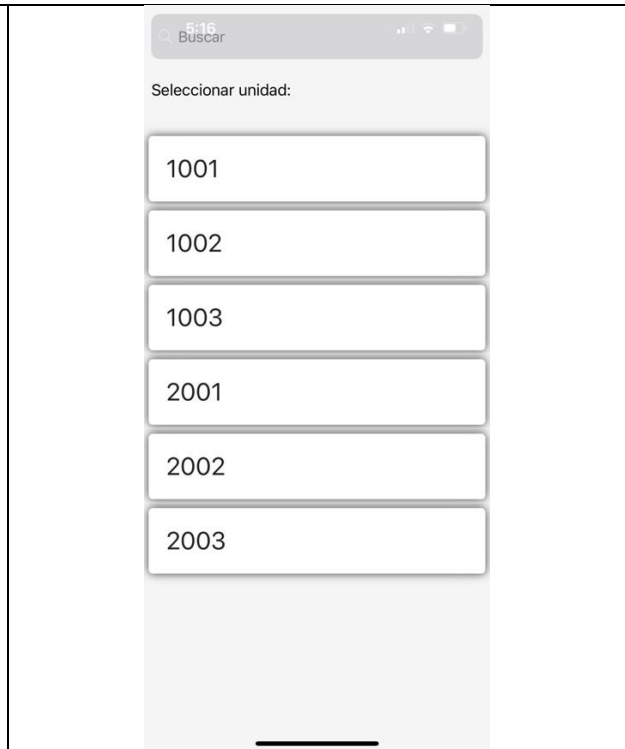


Fig. 39. Pantalla de gestión de unidades-conductor, iOS

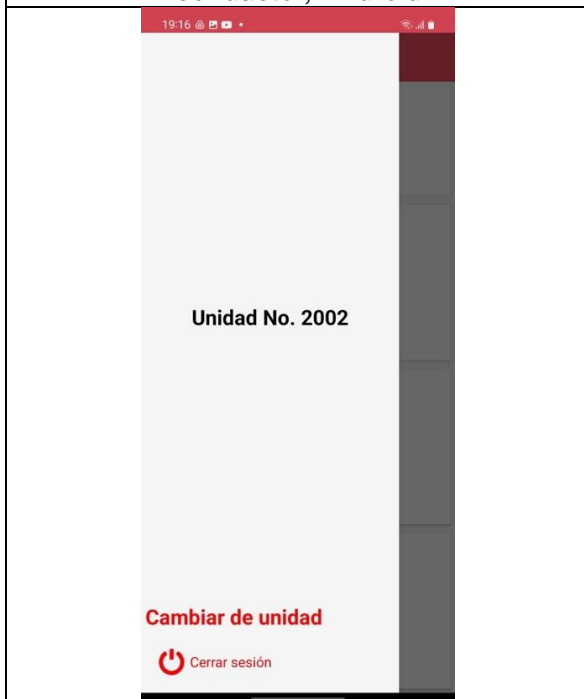


Fig. 40. Pantalla de cambio de unidad-conductor, Android

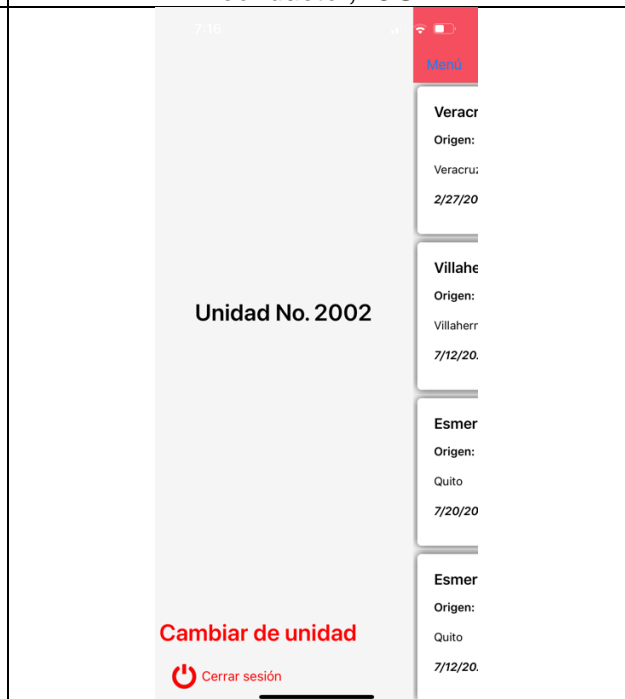


Fig. 41. Pantalla de cambio de unidad-conductor, iOS

Luego de pasar exitosamente todas las pruebas unitarias se dio por finalizada la segunda fase. En esta fase la pantalla de inicio difiere en la aplicación del conductor y la aplicación del pasajero.



### 3.3.3 Tercera fase

Para la presenta fase se tomó en cuenta la geolocalización de las paradas donde se van a recoger a los pasajeros, también se tomó en cuenta datos de vital importancia como son las paradas de abordaje y las paradas de desembarque, finalmente se tomó en cuenta la visualización de las reservas del lado del conductor y los datos que corresponden al pasajero. Las historias de usuario que competen a estas funcionalidades se las puede encontrar en las tablas 35 y 36 para la aplicación del pasajero y en las tablas 37,38 y 39 para el conductor.

Para la aplicación de Usuario:

HISTORIA DE USUARIO	
<b>Número:</b> 5	<b>Usuario:</b> Pasajero
<b>Nombre Historia:</b> Visualización de datos de la reserva	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder visualizar la fecha y hora del viaje, la ruta (ciudad, sector de origen y destino) y el horario seleccionado, además visualizar el costo del pasaje en cada tramo que integra la ruta completa.	
<b>Observación:</b> Los tramos de la rutas, el costo de cada tramo, la fecha y demás datos deben ser parametrizables desde base de datos.	

*Tabla 35. Historia de usuario visualización de datos de la reserva*

HISTORIA DE USUARIO	
<b>Número:</b> 6	<b>Usuario:</b> Pasajero
<b>Nombre Historia:</b> Selección de parada inicial	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder seleccionar la parada inicial de su viaje.	
<b>Observación:</b> Se debe mostrar la ubicación exacta de la parada en un mapa.	

*Tabla 36. Historia de usuario selección de parada inicial*

Para la aplicación de Conductor:

HISTORIA DE USUARIO	
<b>Número:</b> 12	<b>Usuario:</b> Conductor
<b>Nombre Historia:</b> Visualización de rutas por unidad	

<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder visualizar las rutas asignadas a una unidad de transporte donde se muestra el origen, destino y horario.	
<b>Observación:</b> Las rutas deben mostrarse en forma de lista de tarjetas	

*Tabla 37. Historia de usuario visualización de rutas por unidad*

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 13	<b>Usuario:</b> Conductor
<b>Nombre Historia:</b> Visualización de reservas	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder visualizar las reservas realizadas por los pasajeros donde conste el nombre del pasajero, el número de asiento, el tipo de bolero, la parada donde aborda la unidad y la parada donde desembarca.	
<b>Observación:</b> Se debe visualizar la información en un formato tipo lista.	

*Tabla 38. Historia de usuario Visualización de reservas*

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 14	<b>Usuario:</b> Conductor
<b>Nombre Historia:</b> Puntos de abordaje	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 4
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder visualizar la ubicación de las paradas en un mapa. El mapa debe mostrar la ubicación exacta de cada parada.	
<b>Observación:</b> Ninguna	

*Tabla 39. Historia de usuario puntos de abordaje.*

Se realizaron las pruebas unitarias por pantalla. Para las pantallas de selección de rutas se realizaron los casos de prueba descritos en las tablas 40, 41, 42, 43, 44, 45 y 46:

Elemento	Acción	Salida Esperada
Botón de atras	Dar clic	Pantalla búsqueda de rutas

*Tabla 40. Caso de prueba 1 para la Pantalla de visualización de datos de la reserva*

Elemento	Acción	Salida Esperada
Botón de reserva	Dar clic	Pantalla de selección de parada inicial

*Tabla 41. Caso de prueba 2 para la Pantalla de visualización de datos de la reserva.*

Elemento	Acción	Salida Esperada
Botón cancelar "X"	Dar clic	Pantalla de visualización de datos de la reserva

*Tabla 42. Caso de prueba 1 para la Pantalla de selección de parada inicial.*

Elemento	Acción	Salida Esperada
Botón seleccionar	Dar clic	Pantalla de selección de asientos

*Tabla 43. Caso de prueba 2 para la Pantalla de selección de parada inicial.*

Elemento	Acción	Salida Esperada
Botón Menú (icono hamburguesa en Android)	Dar clic	Pantalla cambio de unidad

*Tabla 44. Caso de prueba 1 para la Pantalla de visualización de rutas por unidad.*

Elemento	Acción	Salida Esperada
Tarjeta de información de ruta	Dar clic	Pantalla de reservas

*Tabla 45. Caso de prueba 2 para la Pantalla de visualización de rutas por unidad.*

Elemento	Acción	Salida Esperada
Botón unidad (flecha atras en Android)	Dar clic	Pantalla de rutas por unidad

*Tabla 46. Caso de prueba 1 para la Pantalla de visualización de reservas por unidad.*

Para la tercera fase se tomaron en cuenta varios factores como son los siguientes:

- Lugar de origen
- Hora de salida
- Fecha de salida
- Lugar de destino
- Compañía de transporte
- Conductor asignado
- Número de la unidad
- Tramos de la ruta seleccionada
- Costo de cada tramo

Tomando en cuenta todas estas variables, la interfaz gráfica de visualización de datos para la reserva queda de la siguiente manera:

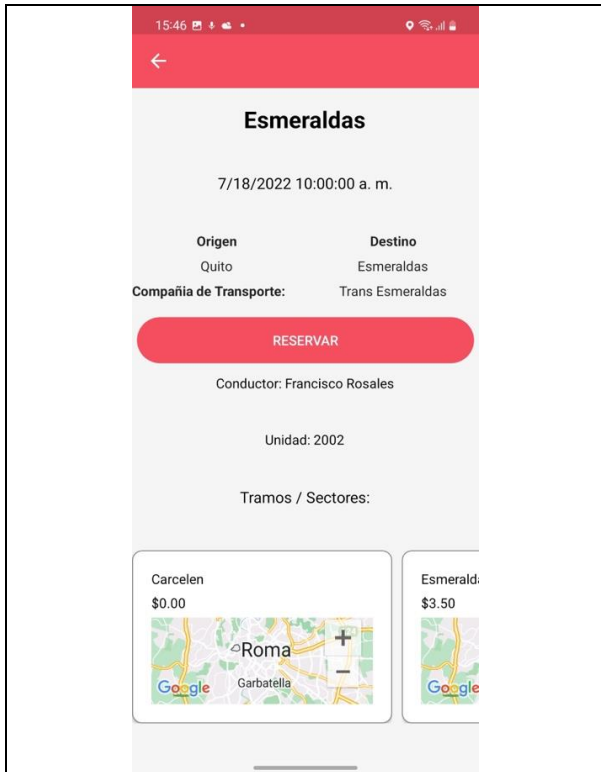


Fig. 42. Pantalla de Visualización de datos de la reserva- pasajero, Android

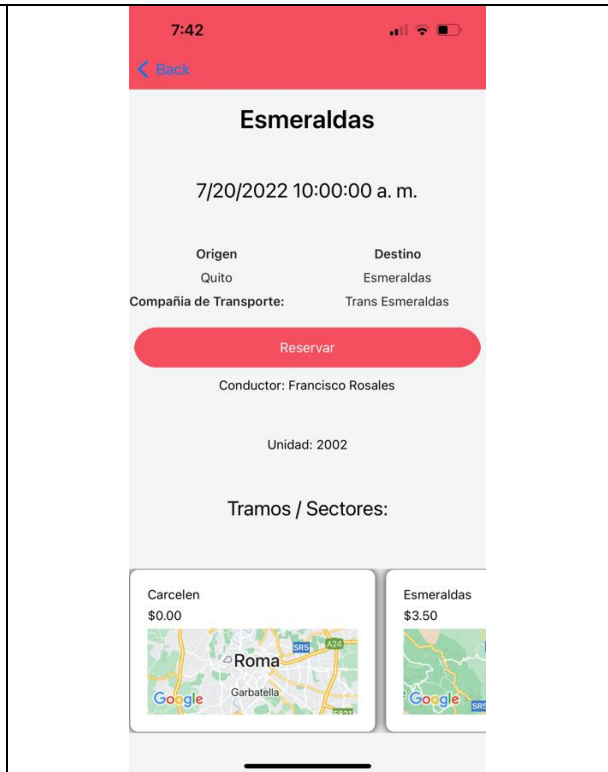


Fig. 43. Pantalla de Visualización de datos de la reserva- pasajero, iOS

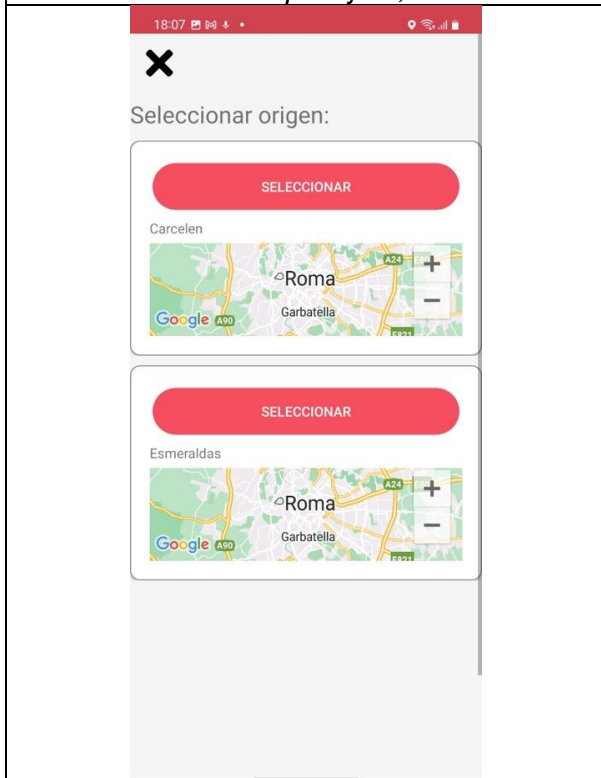


Fig. 44. Pantalla de Selección de parada inicial- pasajero, Android

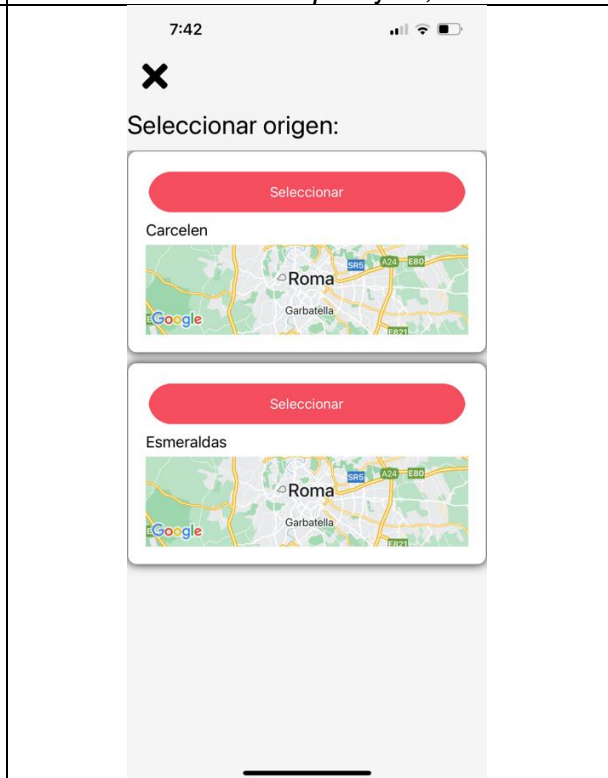


Fig. 45. Pantalla de Selección de parada inicial- pasajero, iOS

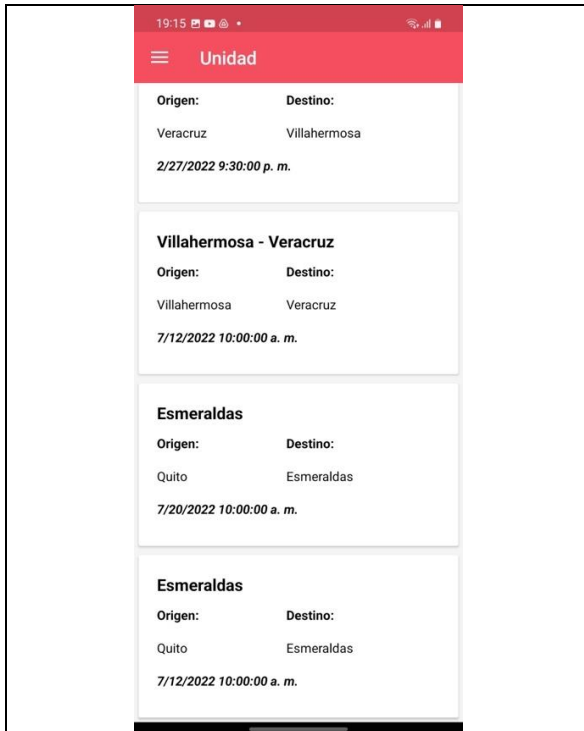


Fig. 46. Pantalla de visualización de rutas por unidad-conductor, Android



Fig. 47. Pantalla de visualización de rutas por unidad-conductor, iOS

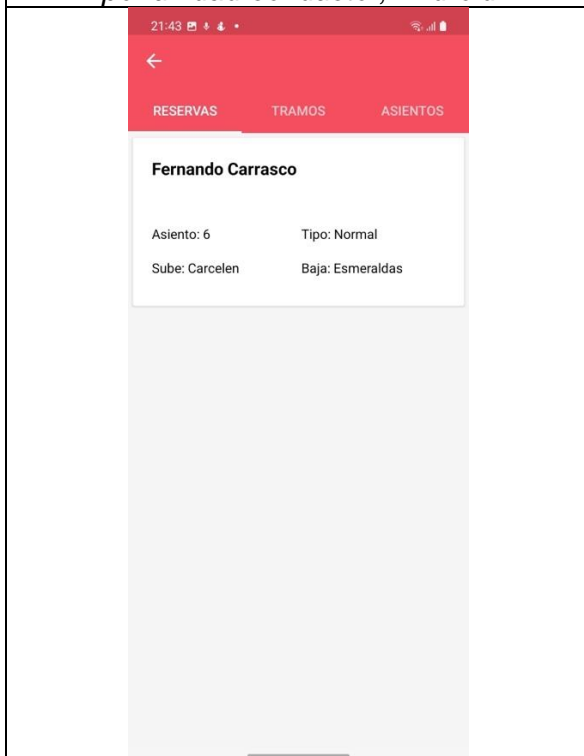


Fig. 48. Pantalla de visualización de reservas-conductor, Android

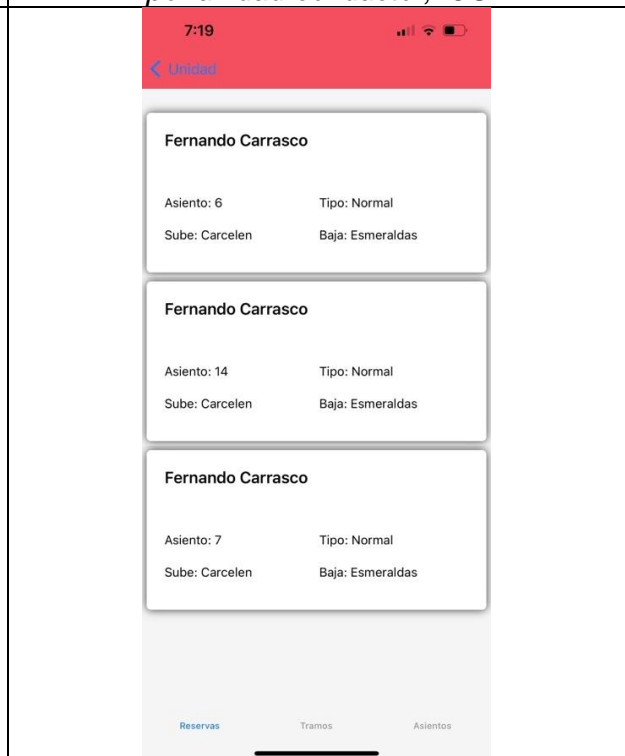


Fig. 49. Pantalla de visualización de reservas-conductor, iOS

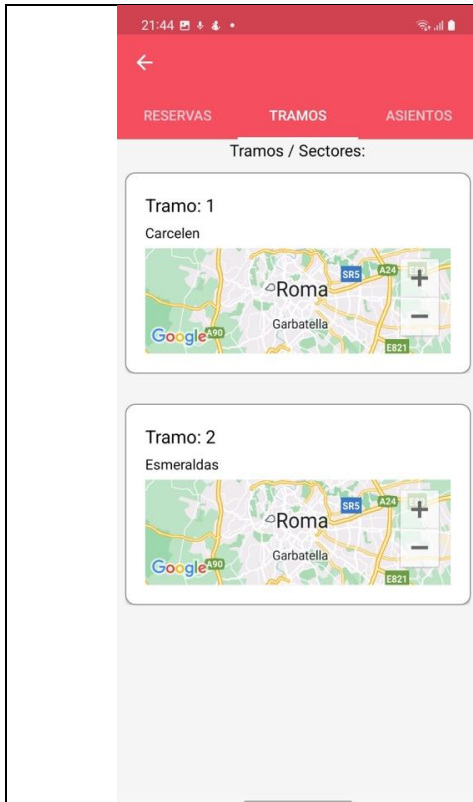


Fig. 50. Pantalla de visualización de tramos- conductor, Android

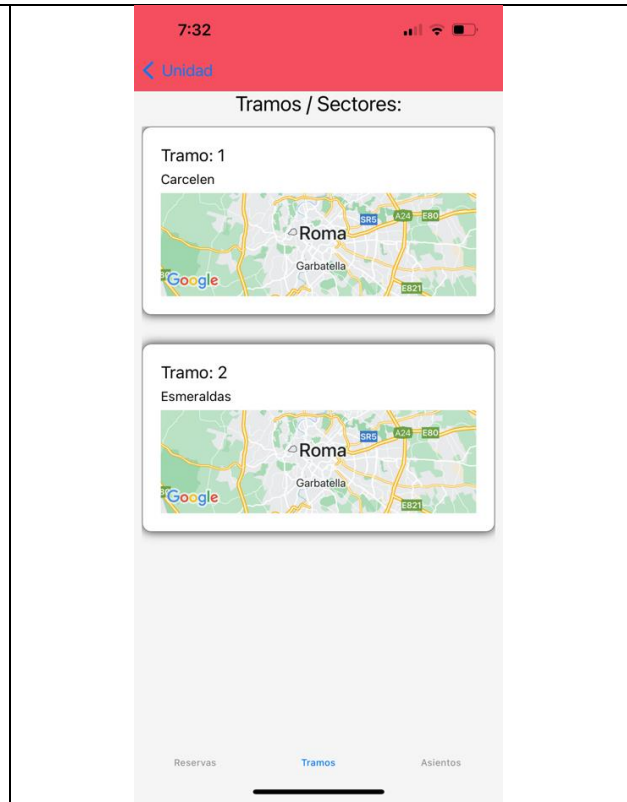


Fig. 51. Pantalla de visualización de tramos- conductor, iOS

Luego de pasar exitosamente todas las pruebas unitarias se dio por finalizada la tercera fase. En esta fase se trabajo mas del lado del conductor ya que se tenían que visualizar las reservas de los clientes y las paradas de cada ruta del conductor.

### 3.3.4 Cuarta fase

Para esta fase se tomaron en cuenta las historias de usuario detalladas en las Tablas 47 y 48 las cuales tratan acerca de la selección y visualización de asientos reservados.

Para la aplicación de Usuario:

HISTORIA DE USUARIO	
<b>Número:</b> 7	<b>Usuario:</b> Pasajero
<b>Nombre Historia:</b> Selección de asientos	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración asignada:</b> 4
<b>Programador Responsable:</b> Fernando Carrasco	

<b>Descripción:</b> El usuario debe poder seleccionar el asiento o asientos que estén disponibles mediante una interfaz gráfica que represente los asientos de una unidad de transporte
<b>Observación:</b> Puede ser que existan varios modelos de unidades por lo que esta interfaz debe poder adaptarse a esas diferencias.

*Tabla 47. Historia de usuario selección de parada inicial*

Para la aplicación de Conductor:

HISTORIA DE USUARIO	
<b>Número:</b> 15	<b>Usuario:</b> Conductor
<b>Nombre Historia:</b> Visualización de asientos	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración asignada:</b> 2
<b>Programador Responsable:</b> Fernando Carrasco	
<b>Descripción:</b> El usuario debe poder visualizar los asientos vendidos y los disponibles en un gráfico que contenga los números de asientos respectivos.	
<b>Observación:</b> El gráfico debe variar según el número de asientos de la unidad.	

*Tabla 48. Historia de usuario Visualización de asientos*

Se realizaron las pruebas unitarias por pantalla. Para las pantallas de selección y visualización de asientos se realizaron los casos de prueba descritos en las tablas 49, 50, 51 y 52:

Elemento	Acción	Salida Esperada
Botón unidad (flecha atras en Android)	Dar clic	Pantalla de rutas por unidad

*Tabla 49. Caso de prueba 1 para la Pantalla de visualización de asientos (pasajero).*

Elemento	Acción	Salida Esperada
Asiento numerado de color blanco	Dar clic	Cambio de color en el asiento a rojo
Asiento numerado de color rojo	Dar clic	Cambio de color en el asiento a blanco en caso de no haber sido reservado con anterioridad por otro usuario

*Tabla 50. Caso de prueba 2 para la Pantalla de visualización de asientos (pasajero).*

Elemento	Acción	Salida Esperada
Botón Confirmar Reserva	Dar clic	En caso de que el número de asientos seleccionados sea diferente de cero se debe mostrar la pantalla de pago

*Tabla 51. Caso de prueba 3 para la Pantalla de visualización de asientos (pasajero).*

Elemento	Acción	Salida Esperada
Botón unidad (flecha atras en Android)	Dar clic	Pantalla de rutas por unidad.

*Tabla 52. Caso de prueba 1 para la Pantalla de visualización de asientos (conductor).*

Para la cuarta fase se tomaron en cuenta varios factores como son los siguientes:

- Lugar de origen
- Hora de salida
- Fecha de salida
- Lugar de destino
- Compañía de transporte
- Numero de asiento

Tomando en cuenta todas estas variables y que el comportamiento natural de los usuarios es seleccionar unos asientos y luego decidirse por otros se implementó un algoritmo que permita controlar el evento que se genera al dar clic en cada imagen de los asientos, dicho algoritmo quedó de la siguiente manera:



```

private void SelectAsiento()
{
    if ((bool)this.Reservado)
    {
        this.Reservado = false;
        this.ColorReserva = Color.White;
        MainViewModel.GetInstance().NoBoletos =
MainViewModel.GetInstance().NoBoletos - 1;
        if (MainViewModel.GetInstance().AsientosReservaList != null)
        {
            MainViewModel.GetInstance().AsientosReservaList.Remove(this);
        }

        if (MainViewModel.GetInstance().Reservar.SeleccionAsientos !=
null)
        {
MainViewModel.GetInstance().Reservar.SeleccionAsientos.Remove(this);
        }
    }
    else
    {
        this.Reservado = true;
        this.ColorReserva = Color.Red;
        MainViewModel.GetInstance().NoBoletos =
MainViewModel.GetInstance().NoBoletos + 1;

        MainViewModel.GetInstance().AsientosReservaList.Add(this);
        if (MainViewModel.GetInstance().AsientosReservaList == null)
        {
            MainViewModel.GetInstance().AsientosReservaList = new
List<Asientos_ruta>();
            MainViewModel.GetInstance().AsientosReservaList.Add(this);
        }
        MainViewModel.GetInstance().Reservar.SeleccionAsientos.Add(this);
    }
    else
    {
        MainViewModel.GetInstance().AsientosReservaList.Add(this);
    }
    MainViewModel.GetInstance().Reservar.SeleccionAsientos.Add(this);
}
}
}

```

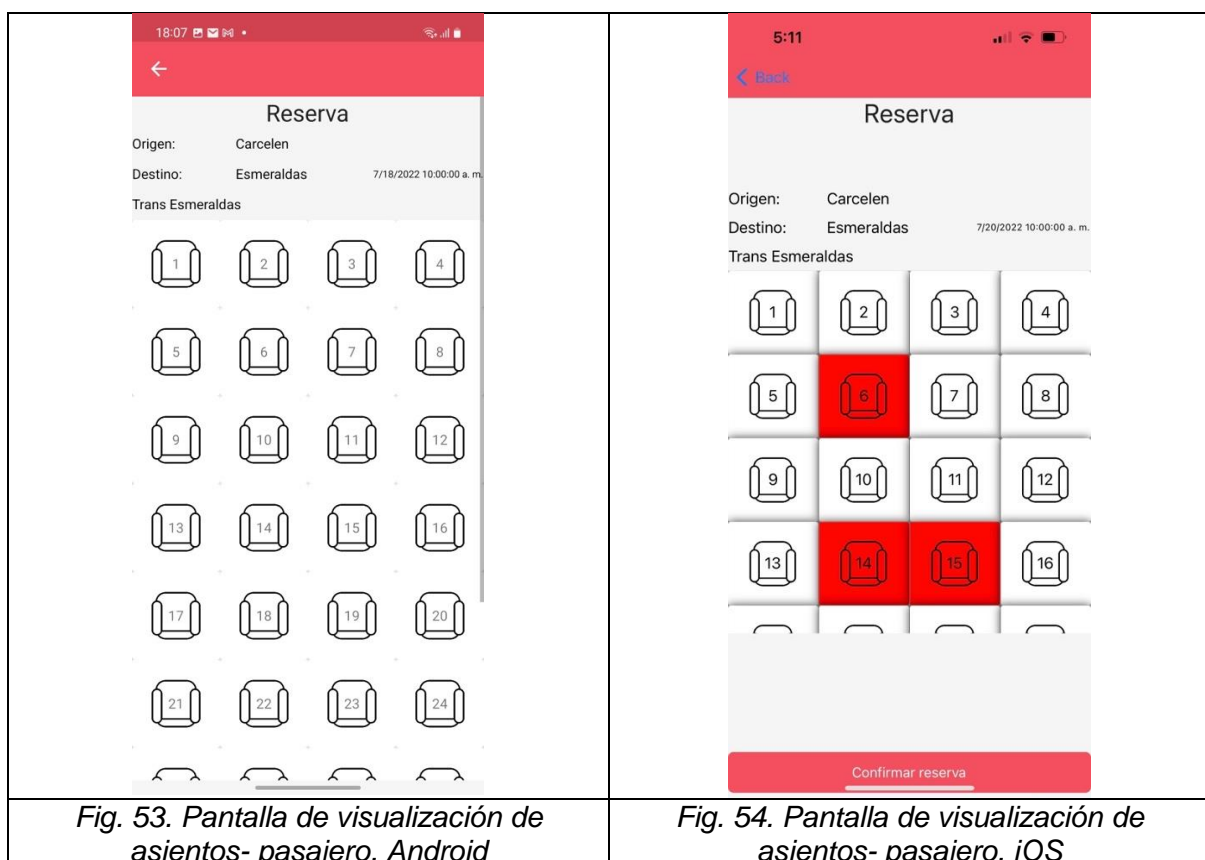
Figura 52. Algoritmo de selección de asientos.

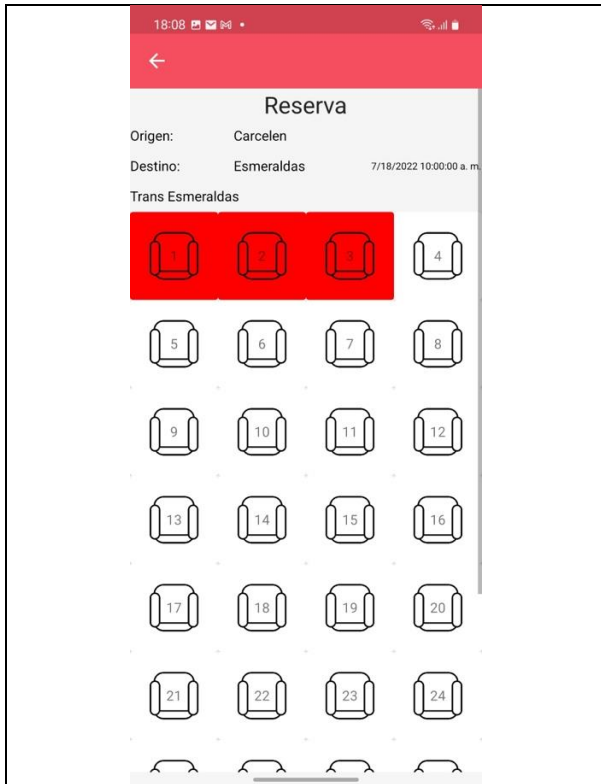
Cómo se puede observar en el algoritmo se hizo uso de listas para agregar o quitar la reserva de un asiento, de igual forma se utilizó un booleano para validar si un asiento ha sido reservado o no, el booleano por defecto permanece en false hasta que la imagen sea presionada, al ser presionada por primera vez la imagen del asiento esta cambia su color a rojo y su valor booleano pasa a true, adicionalmente se agrega el contador de número de asiento en 1 y se agrega el asiento reservado a la lista de asientos. Luego al volver a presionar

cambia de color a blanco y su valor booleano a false, adicionalmente se disminuye el valor del número de asientos reservados en 1 y se remueve de la lista de asientos el asiento seleccionado.

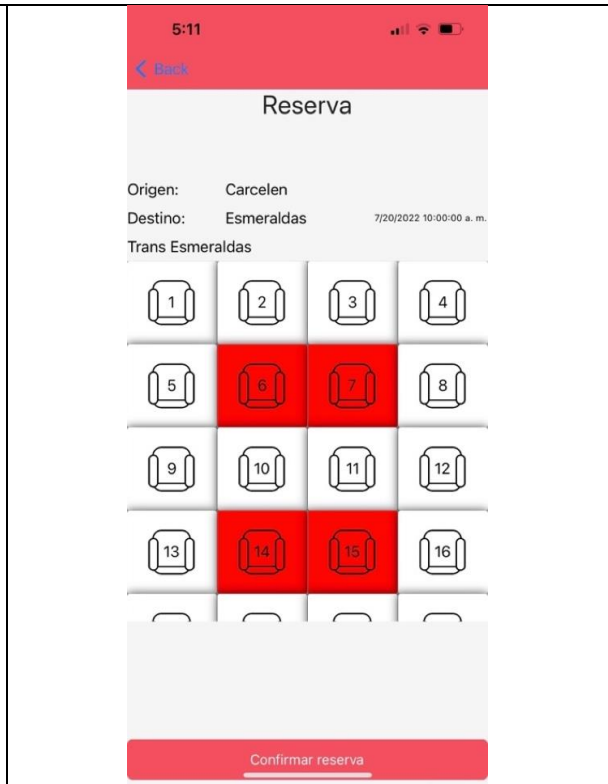
En caso de que algún usuario desee escoger los mismos asientos que un usuario ya los haya reservado no va a poder ya que cuando un asiento viene reservado desde base de datos no puede cambiarlo con un simple clic, para controlar esto se agregó un campo en la base de datos el cual nos dice si el asiento está habilitado o no. En las Figuras 53, 54, 55, 56, 57 y 58 podemos observar cómo se comporta la selección de asientos en ambas plataformas.

Luego de que el usuario termina de seleccionar los asientos y al dar clic en el botón “Confirmar reserva” la lista de asientos generada se pasa a base de datos donde se actualiza el valor de los asientos reservados. Adicionalmente se guarda el valor de cuantos asientos han sido reservados de esta manera en la siguiente pantalla se puede calcular el monto a pagar dependiendo de la cantidad de asientos reservados. El guardado de los asientos para ambas plataformas se puede ver en las Figuras 59 y 60.

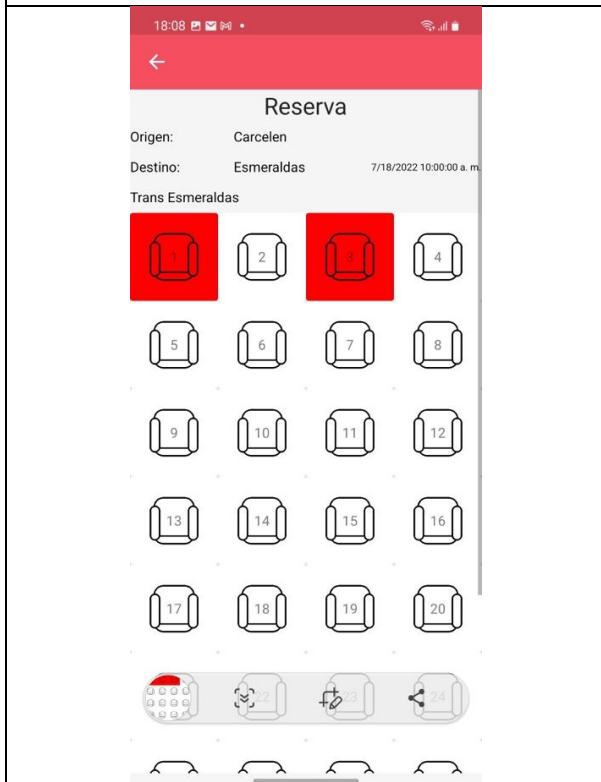




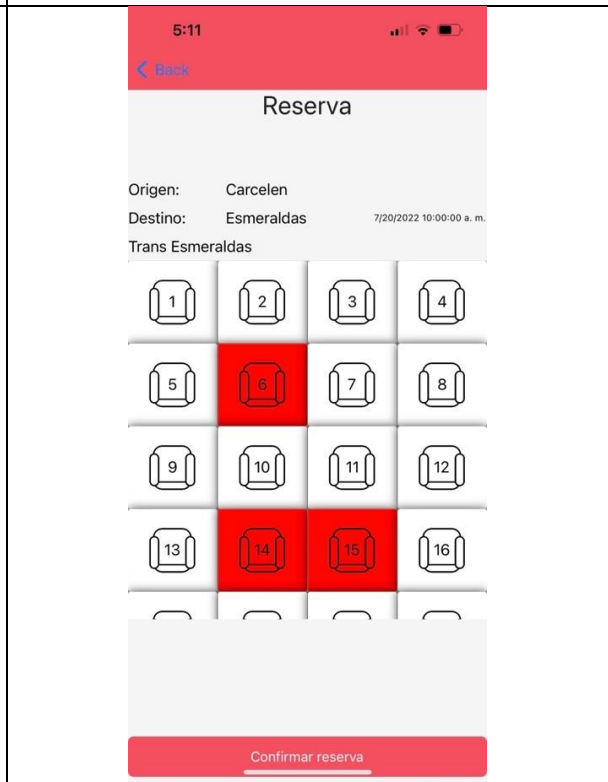
*Fig. 55. Selección de asientos- pasajero, Android*



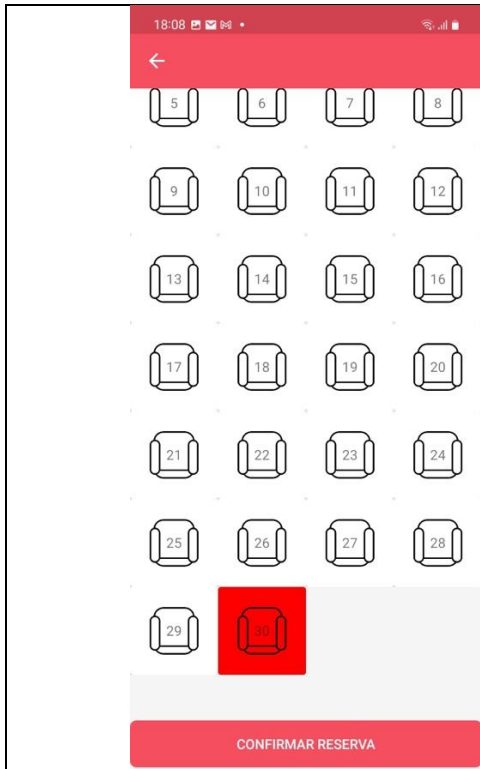
*Fig. 56. Selección de asientos- pasajero, iOS*



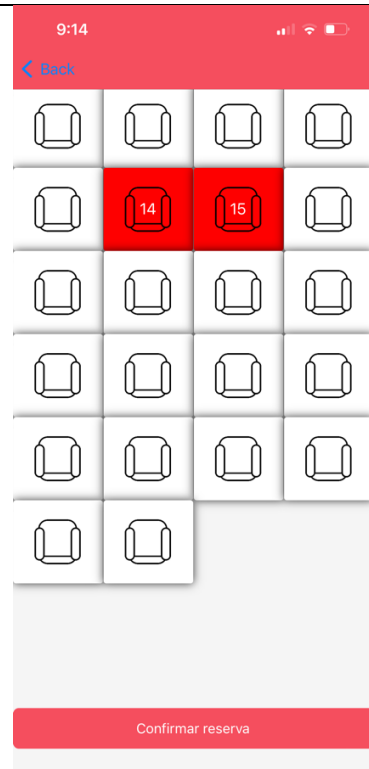
*Fig. 57. Des enmarcado de asientos- pasajero, Android*



*Fig. 58. Des enmarcado de asientos- pasajero, iOS*



*Fig. 59. Pantalla de confirmación de asientos para la reserva, Android*



*Fig. 60. Pantalla de confirmación de asientos para la reserva, iOS*

Para que el conductor tenga una mejor visualización de que asientos ya están ocupados se implementó un gráfico donde se muestran los números de los asientos con el siguiente código de colores:

- Rojo: asiento ocupado
- Blanco: asiento disponible

Los datos se obtienen de la misma tabla donde se guardan los valores de la Historia de Usuario 6 mencionada en la tabla 36. Adicionalmente se muestran los siguientes datos:

- Origen: lugar de origen de la ruta
- Destino: lugar de destino de la ruta
- Compañía de transporte a la que pertenece la unidad
- Fecha y hora de salida de la ruta.

La implementación de esta pantalla se puede ver en las Figuras 60 y 61.

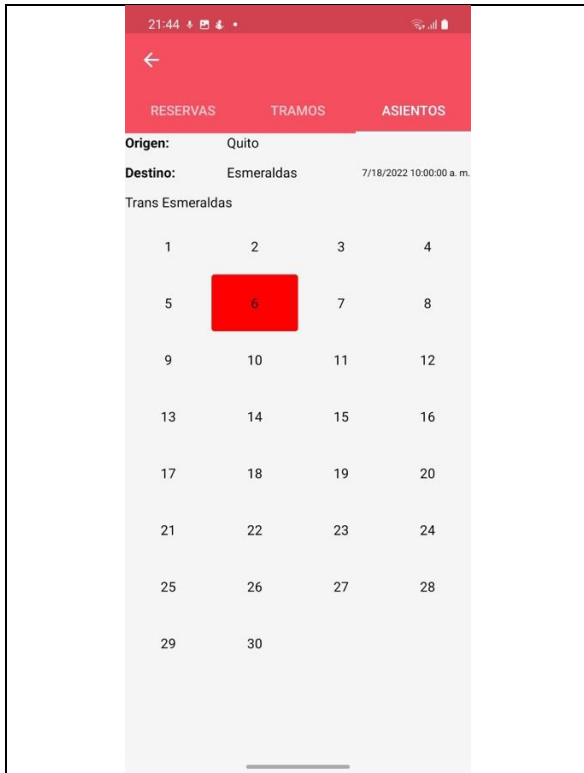


Fig. 61. Pantalla de visualización de asientos- conductor, Android

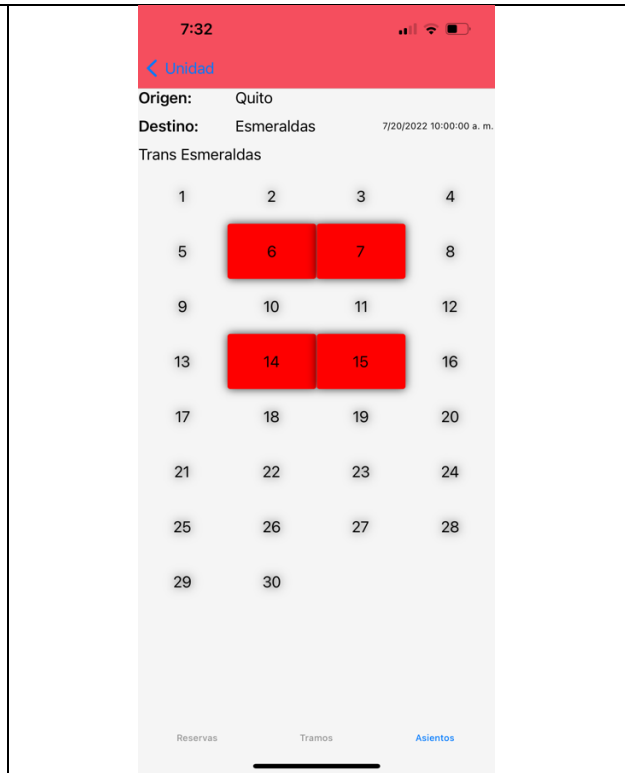


Fig. 62. Pantalla de visualización de asientos- conductor, iOS

Luego de pasar exitosamente todas las pruebas unitarias se dio por finalizada la cuarta fase. En esta fase se realizó un esfuerzo mayor ya que se implementó un algoritmo para poder escoger los asientos sin afectar a los asientos ya reservados y con la posibilidad de quitar la selección de un asiento con un solo clic.

### 3.3.5 Quinta fase

Para esta fase se tomó en cuenta el pago del pasaje por parte del pasajero, debido a que el presente proyecto solo tiene motivo académico la forma en que se realiza el pago es simbólica y no incluye un botón de pagos que esté conectado con el sistema financiero nacional o internacional. La historia de usuario a tratar en esta fase está descrita en la tabla 53.

Para la aplicación de Usuario:

HISTORIA DE USUARIO	
<b>Número:</b> 8	<b>Usuario:</b> Pasajero
<b>Nombre Historia:</b> Pago de pasajes	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos Estimados:</b> 3	<b>Iteración asignada:</b> 5
<b>Programador Responsable:</b> Fernando Carrasco	

<b>Descripción:</b> El usuario debe poder seleccionar la forma de pago, el pago debe ser calculado de acuerdo con el número de pasajes.
<b>Observación:</b> Por motivos académicos el botón de pago no está conectado con una entidad bancaria

*Tabla 53. Historia de usuario pago de pasajes*

Para la aplicación de Conductor:

No hay historias para este usuario en la quinta fase.

Se realizaron las pruebas unitarias por pantalla. Para las pantallas de pago de pasajes se realizaron los casos de prueba descritos en las tablas 54, 55, 56, 57 y 58:

Elemento	Acción	Salida Esperada
Botón atras(icono de flecha en Android)	Dar clic	Pantalla de rutas por unidad

*Tabla 54. Caso de prueba 1 para la Pantalla de pago de reserva.*

Campo	Entrada	Salida Esperada
Número de tarjeta	6579 6474 3498 3426	Número de tarjeta válido
Número de tarjeta	Abcdefg	No permite ingresar esos caracteres
Número de tarjeta	12345678	Número de caracteres insuficiente
Número de tarjeta	6579647434983420	Número de tarjeta válido
Número de tarjeta	#\$&*()".¿?+	No permite ingresar esos caracteres
Número de tarjeta		Este campo es obligatorio

*Tabla 55. Caso de prueba 2 para la Pantalla de pago de reserva.*

Campo	Entrada	Salida Esperada
Fecha de expiración	0225	Fecha de expiración válida
Fecha de expiración	Abcdefg	No permite ingresar esos caracteres
Fecha de expiración	12345678	Número de caracteres no permitido
Fecha de expiración	225	Número de caracteres insuficiente
Fecha de expiración	#\$&*()".¿?+	No permite ingresar esos caracteres
Fecha de expiración		Este campo es obligatorio

*Tabla 56. Caso de prueba 3 para la Pantalla de pago de reserva.*

Campo	Entrada	Salida Esperada
Código de verificación de tarjeta(CCV)	984	CCV Válido
Código de verificación de tarjeta(CCV)	Abc	No permite ingresar esos caracteres
Código de verificación de tarjeta(CCV)	12345678	Número de caracteres no permitido
Código de verificación de tarjeta(CCV)	22	Número de caracteres insuficiente
Código de verificación de tarjeta(CCV)	#\$&*()".¿?+	No permite ingresar esos caracteres
Código de verificación de tarjeta(CCV)		Este campo es obligatorio

*Tabla 57. Caso de prueba 4 para la Pantalla de pago de reserva.*

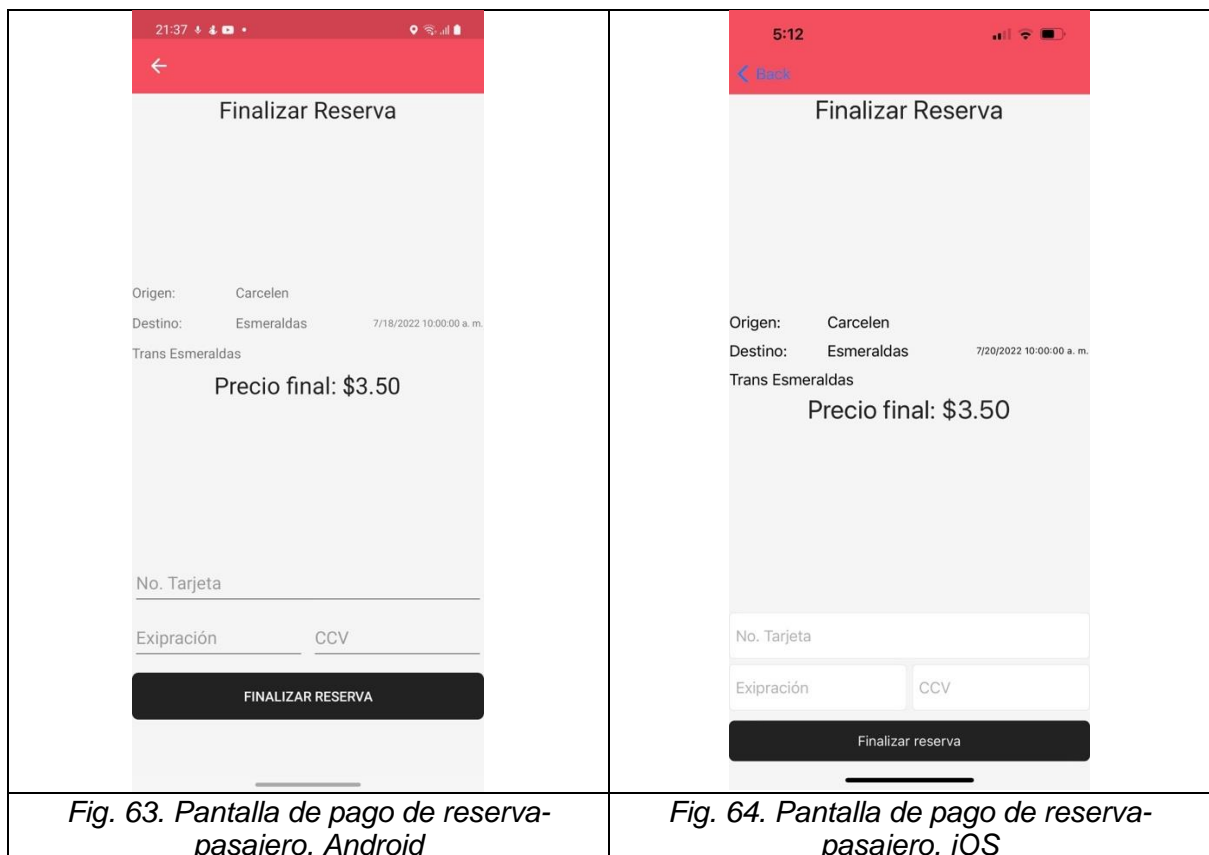
Elemento	Acción	Salida Esperada
Botón Finalizar Reserva	Dar clic	Alerta de confirmación de compra en caso de que todos los campos estén correctos
Botón Finalizar Reserva	Dar clic	Alerta de error en la compra en caso de algún campo ingresado erróneamente

*Tabla 58. Caso de prueba 5 para la Pantalla de pago de reserva.*

Para la quinta fase se tomaron en cuenta varios factores como son los siguientes:

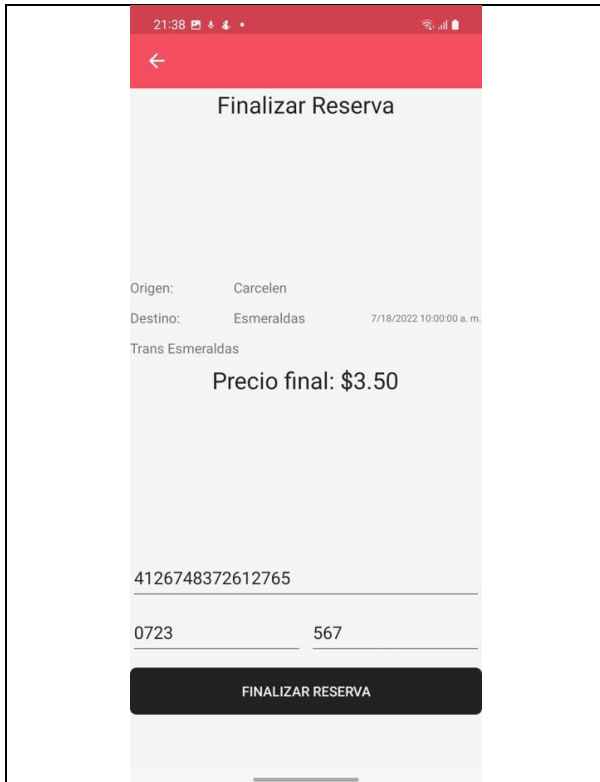
- Lugar de origen
- Hora de salida
- Fecha de salida
- Lugar de destino
- Compañía de transporte
- Ingreso de datos de tarjeta débito

Tomando en cuenta todas estas variables, la interfaz gráfica de búsqueda de rutas queda de la siguiente manera:

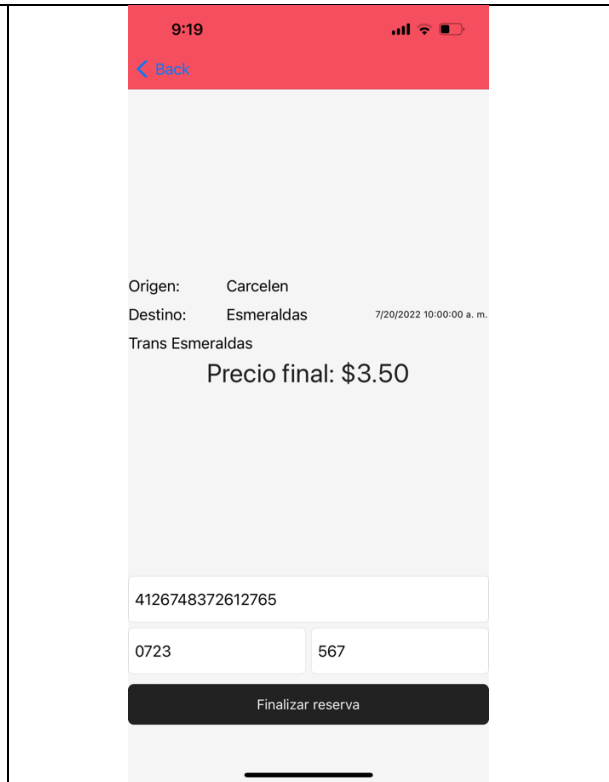


*Fig. 63. Pantalla de pago de reserva-pasajero, Android*

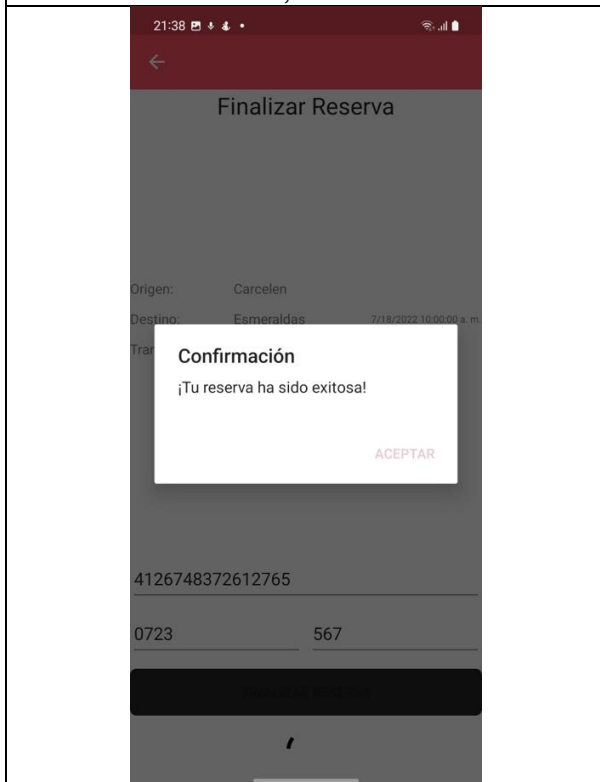
*Fig. 64. Pantalla de pago de reserva-pasajero, iOS*



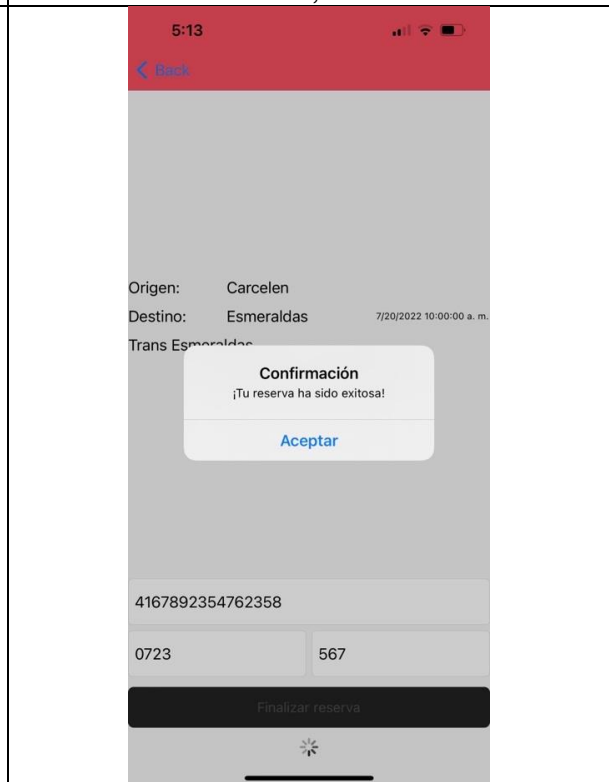
*Fig. 65. Pantalla de pago de reserva con datos, Android*



*Fig. 66. Pantalla de pago de reserva con datos, iOS*



*Fig. 67. Mensaje de confirmación de reserva pasajero, Android*



*Fig. 68. Mensaje de confirmación de reserva pasajero, iOS*

Luego de pasar exitosamente todas las pruebas unitarias se dio por finalizada la quinta fase.



## Capítulo 4 – Resultados

En esta sección se describen los resultados obtenidos luego de todo el proceso de desarrollo.

Como resultado se obtuvieron al final cuatro aplicaciones móviles las cuales corresponden a:

- Aplicación móvil del conductor para Android.
- Aplicación móvil del pasajero para Android.
- Aplicación móvil del conductor para iOS.
- Aplicación móvil del pasajero para iOS.

Adicionalmente se configuró un servidor remoto para alojar la base de datos y toda la lógica detrás del backend.

Debido a la complejidad del proyecto que se planteó inicialmente se tuvieron que modificar algunas historias sobre la marcha para no extender el desarrollo del sistema más de la cuenta, pero se mantuvieron las funcionalidades principales como son la selección de asientos desde una interfaz gráfica, la visualización de las paradas en un mapa, la compra de pasajes y la gestión de unidades de transporte.

Para todo el desarrollo del proyecto se utilizó la metodología XP la cual permitió un flujo de trabajo ágil. También se planteó el uso del enfoque TDD que se encuentra dentro de las prácticas de XP la cual se utilizó a nivel de pruebas de la aplicación como tal y no del lado del servidor. Adicionalmente se realizaron entregables lo más pronto posible en cada sprint priorizando la funcionalidad del proyecto antes que la estética, por tal motivo el sistema mantiene una sólida estabilidad en todos sus procesos.

En cuanto a la configuración inicial del proyecto se plantearon varios temas que son los siguientes:

- Utilizar una arquitectura serverless con Firebase como tecnología de apoyo.
- Utilizar una base de datos no relacional basada en Firebase firestore.
- Utilizar open Street map para el manejo de mapas.
- Utilizar una librería syncfusion para el diseño de interfaces.

Luego de varias pruebas y análisis se decidió cambiar esos planteamientos iniciales con lo que quedaron de la siguiente manera:

- Se utilizó una arquitectura cliente servidor donde la tecnología que se implementó fue ASP.NET core la cual es totalmente compatible con Xamarin.
- Se utilizó el sistema de base de datos Microsoft SQL Server.

- Se utilizó Google maps para ambas plataformas debido a que existe mayor documentación y librerías probadas que facilitan su desarrollo.
- Se diseñó toda la interfaz desde cero para tener un mejor control del sistema.

El motivo por el cual se decidió migrar a la arquitectura de cliente servidor fue la falta de documentación que se puede encontrar de cómo implementar firebase con Xamarin, adicionalmente implicaba el uso de código específico de cada plataforma lo que aumentaba la complejidad y limitaba el uso de pruebas unitarias.

Por otra parte, el uso de la librería syncfusion agregaba complejidad al proyecto debido a que la librería incrustaba una cantidad de código importante al proyecto lo cual en un principio era beneficioso, pero lastimosamente a medida que fue creciendo la cantidad de código se volvió más complejo encontrar donde modificar ciertas partes de la interfaz y cómo corregir errores que se presentaban. Una vez superadas todas las limitaciones el desarrollo de la aplicación fluyó de manera normal.

Para la creación de la base de datos se utilizó Entity Framework el cual es un ORM (Object Relation Mapping) un modelo de programación que permite mapear las estructuras de una base de datos relacional sobre una estructura lógica de entidades con el objetivo de crear y modificar bases de datos a partir de código escrito en C#.

En cada ciclo de desarrollo el uso de TDD permitió identificar y prever los posibles errores para avanzar de un ciclo a otro exitosamente gracias a su enfoque de pensar en las pruebas primero. Por ejemplo, en la primera iteración ayudó a validar que cada campo se llene solo con los valores necesarios evitando inconsistencia de datos. Para la segunda iteración se validó que se siga el flujo correcto de escoger el origen del viaje y que el buscador funcione solo con los caracteres permitidos. En la tercera iteración se evaluó que la interfaz gráfica facilita que el usuario pueda escoger el viaje sin problemas. Seguidamente, en la cuarta iteración las pruebas ejecutadas permitieron comprobar que el usuario puede seleccionar los asientos sin intervenir en selecciones previas de otros usuarios y que puede escoger sin problemas los asientos deseados pasando correctamente a la pantalla de pago. Por último, para la quinta iteración las pruebas ayudaron a validar correctamente el ingreso de los datos de la tarjeta de débito o crédito para proceder con el pago y que así se confirme la reserva.

## Capítulo 5 – Conclusiones y Recomendaciones

### 5.1 Conclusiones

La decisión de diseñar la interfaz desde cero facilitó la velocidad con la que se desarrollaron las pantallas y permitió un mayor control del sistema.

Debido a la complejidad de la lógica de ciertas interfaces como la de selección de asientos, el utilizar el patrón MVVM permitió separar la lógica de la interfaz de la lógica del llamado a los servicios, con lo que fue mas facil de probar y a futuro permitiría tener un código mantenible y seguro.

Se evidenció que el uso de TDD tiene varias ventajas por ejemplo asegura que cuando se implemente una nueva funcionalidad al proyecto, no se afecte a procesos anteriores y dependiendo de la calidad y cantidad de los casos de prueba permite entregar un código funcional y estable en la mayoría de los casos; la desventaja es que no todos los programadores están acostumbrados a ese enfoque lo que puede ocasionar retrasos en las actividades del sprint al inicio del proyecto.

Al momento de implementar TDD como buena practica de programación se presentaron varios retos como la incompatibilidad de probar servicios externos dentro de la lógica de programación, la complejidad de probar pantallas que requieren del uso de sensores del dispositivo como el GPS y la demora que representa el sentarse a pensar todos los escenarios posibles para luego volver a plantearlos hasta llegar a un punto intermedio que asegure la calidad del sistema.

El uso de Xamarin.Forms en el presente proyecto permitió desarrollar aplicaciones móviles multiplataforma con una sola base de código y con un rendimiento de aplicación nativa, adicionalmente el trabajar solo con tecnologías de Microsoft demuestra que se puede crear software de alta calidad con soluciones propietarias sin gastar mucho dinero, ya que muchas de estas tecnologías son open source.

Se demostró que para poder utilizar una arquitectura serverless en aplicaciones móviles desarrolladas en Xamarin.Forms es conveniente primero validar que se tenga la documentación necesaria y verificar si se debe realizar un análisis sobre la conveniencia en caso de empresas donde la información es más sensible de lo normal, en dichos caso es

preferible manejar una arquitectura de tipo cliente servidor, la cual permite al negocio tener mayor propiedad de los datos que genera la aplicación.

La toma de requerimientos por medio de reuniones con los stakeholders fue de gran ayuda para obtener las funcionalidades iniciales, sin embargo; a lo largo del desarrollo hizo falta el seguimiento de las historias de usuarios por parte de los stakeholders, lo cual retrasó el proyecto y se tuvo que redefinir historias para no ver afectado el tiempo de entrega del mismo.

Para temas de autenticación se evidenció que el uso de correo electrónico permite disminuir costos ya que el servidor de correos se lo puede desarrollar fácilmente de forma interna, pero si utilizáramos el número celular se tendría que depender de servidores externos los cuales suelen tener un costo mayor que un proveedor externo de correo.

La arquitectura serverless si bien tiene ciertas ventajas en algunas tecnologías de desarrollo móvil como flutter, lastimosamente en tecnologías como Xamarin.Forms no presenta un soporte tan amplio.

## 5.2 Recomendaciones

Considero que para un desarrollo de la aplicación más comercial y de uso masivo es necesario utilizar librerías como Syncfusion que permiten facilitar el desarrollo de las interfaces sin perder la calidad y facilidad de uso de estas.

Se recomienda utilizar siempre el patrón de diseño MVVM para el desarrollo de aplicaciones móviles en Xamarin.Forms ya que permite separar la lógica del negocio del diseño de la interfaz gráfica lo que facilita el uso de pruebas unitarias, así como las pruebas de interfaz.

## Referencias

- [1] INEC, «Anuario de Estadísticas de Transporte 2019,» INEC, 5 Enero 2019. [En línea].
- [2] Y. M. S. Xie, «Yi Min Shum Xie,» 26 Abril 2022. [En línea]. Available: <https://yiminshum.com>.
- [3] K. Beck y C. Andres, Extreme Programming Explained: embrace change, Massachusetts: Addison Wesley Professional, 2000.
- [4] K. Beck y M. Fowler, Planning Extreme Programming, New Jersey: Addison Wesley Professional, 2001.

- [5] J. Bennett, Xamarin in Action: Creating native cross-platform mobile apps, Simon and Schuster, 2018.
- [6] I. C. R. H. I. M. M. C. Ing. Manuel Cortés Iglesias, «SCIELO,» diciembre 2016. [En línea]. Available: [http://scielo.sld.cu/scielo.php?pid=S2218-36202016000400010&script=sci\\_arttext&tlng=en](http://scielo.sld.cu/scielo.php?pid=S2218-36202016000400010&script=sci_arttext&tlng=en). [Último acceso: 21 febrero 2023].