



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

**APLICACIÓN WEB RESPONSIVE PARA LA GESTIÓN DE RECORRIDOS
Y COMUNICACIÓN ENTRE TRANSPORTISTAS PARTICULARES Y
ESTUDIANTES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS
HUACHI**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Tecnologías de la Información.

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Santiago Sebastian Nuñez Tite

TUTOR: Ing. Santiago David Jara Moya, Mg.

Ambato – Ecuador

febrero – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: APLICACIÓN WEB RESPONSIVE PARA LA GESTIÓN DE RECORRIDOS Y COMUNICACIÓN ENTRE TRANSPORTISTAS PARTICULARES Y ESTUDIANTES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS HUACHI, desarrollado bajo la modalidad Proyecto de Investigación por el señor Santiago Sebastian Nuñez Tite, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024.

Ing. Santiago David Jara Moya, Mg.
TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: APLICACIÓN WEB RESPONSIVE PARA LA GESTIÓN DE RECORRIDOS Y COMUNICACIÓN ENTRE TRANSPORTISTAS PARTICULARES Y ESTUDIANTES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS HUACHI es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024.



Santiago Sebastian Nuñez Tite

C.C. 1804428595

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.



Santiago Sebastian Nuñez Tite

C.C. 1804428595

AUTOR

APROBACIÓN DE TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Santiago Sebastian Nuñez Tite, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado APLICACIÓN WEB RESPONSIVE PARA LA GESTIÓN DE RECORRIDOS Y COMUNICACIÓN ENTRE TRANSPORTISTAS PARTICULARES Y ESTUDIANTES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS HUACHI, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Pablo Israel Morales Paredes, Mg
PROFESOR CALIFICADOR

Ing. Leonardo David Torres Valverde, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

El presente proyecto está dedicado a mi padre, porque, aunque la vida haya decidido que ya no esté a mi lado no significa que esté lejos de mi memoria. Su recuerdo siempre estará en mi corazón e inmortalizado en esta pequeña huella.

A mi madre, gracias por enseñarme a ser fuerte, a pesar de las adversidades siempre sabes cómo seguir adelante con tu propio esfuerzo y aunque no lo sepas, eres mi mayor admiración.

A todos mis amigos que me brindaron apoyo o un consejo en mis momentos más oscuros. Los llevo siempre conmigo. Muchísimas gracias...

-Santiago Sebastian Nuñez Tite

AGRADECIMIENTO

Agradezco a mi familia el impulso para seguir adelante y no desistir en el transcurso de mi carrera universitaria. Todo el apoyo que he recibido de ellos ha sido fundamental para lograr este objetivo.

A la vida que puso en mi camino personas y amigos excepcionales y que en mayor o menor medida aportaron para que yo pueda alcanzar esta meta. No me alcanza la vida para decirles gracias por todo y perdón si alguna vez los he lastimado o hecho daño.

A mi tutor Ing. Santiago Jara por brindarme el apoyo para el desarrollo de mi proyecto. Toda mi gratitud hacia su persona y sus futuros proyectos.

A todas las personas que forman o formaron parte de mi vida universitaria. Gracias.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR.....	iv
APROBACIÓN DE TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO.....	vii
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE TABLAS.....	xv
RESUMEN EJECUTIVO.....	xvii
ABSTRACT.....	xviii
CAPÍTULO I.- MARCO TEÓRICO.....	1
1.1. Tema de Investigación.....	1
1.1.1. Planteamiento del problema.....	1
1.2. Antecedentes investigativos.....	2
1.3. Fundamentación Teórica.....	4
1.4. Objetivos.....	14
1.4.1. Objetivo General.....	14
1.4.2. Objetivos Específicos.....	14
CAPÍTULO II.- METODOLOGÍA.....	15
2.1. Materiales.....	15
2.1.1. Preguntas realizadas para la encuesta.....	15
2.1.2. Formato de entrevista aplicada a los transportistas.....	18
2.2. Métodos.....	19
2.2.1. Modalidad de la Investigación.....	19
a. Investigación bibliográfica.....	19
b. Investigación de campo.....	19
c. Investigación aplicada.....	19
2.2.2. Población y muestra.....	19
2.2.3. Recolección de información.....	19
a. Análisis e interpretación de encuesta aplicada a estudiantes.....	22
b. Resultados de las entrevistas aplicadas a transportistas.....	32

2.2.4. Procesamiento y análisis de datos.....	34
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN	35
3.1. Análisis y discusión de resultados	35
3.1.1. Descripción del análisis de impacto de inconvenientes a los involucrados	35
3.1.2. Requisitos funcionales y no funcionales para el desarrollo de la aplicación web responsive.....	37
a. Requisitos funcionales para transportistas	37
b. Requisitos funcionales para estudiantes.....	38
c. Requerimientos no funcionales para el desarrollo de la aplicación.	38
3.1.3. Frameworks de desarrollo front-end.....	40
a. Angular	40
b. Vue.js.....	40
c. React JS	41
d. Selección de metodología front-end.....	42
3.1.4. Bootstrap y Material Design para desarrollo web responsive	42
a. Selección de tecnología web responsive	43
3.1.5. Tecnologías de back-end para la aplicación web	43
a. .NET CORE.....	43
b. Node JS	44
c. Comparación Tecnologías Backend	44
d. Selección de tecnología backend	44
3.1.6. Metodologías de desarrollo web	45
a. Determinación de metodología para la aplicación web.....	45
b. XP.....	45
c. OOHDM.....	46
d. ICONIX.....	48
3.1.7. Comparación entre metodologías	51
3.1.8. Selección de metodología	52
3.2. Desarrollo de la propuesta	52
3.2.1. Primera fase: recopilación de requisitos	52
a. Análisis de requerimientos de la aplicación	52
b. Actores	53
c. Relación de generalización entre actores	54
d. Diagrama de casos de uso	54

e. Documentación de casos de uso.....	55
3.2.2. Segunda etapa – Diseño conceptual.....	63
a. Diagrama de clase Usuario.....	63
b. Diagrama de clase Transportista	64
c. Diagrama de clase Estudiante.....	65
3.2.3. Tercera etapa – Diseño navegacional.....	66
a. Diagrama navegacional de un usuario.....	66
b. Diagrama navegacional de un estudiante	66
c. Diagrama navegacional del transportista.	67
d. Contexto navegacional de un usuario	68
e. Contexto navegacional de un estudiante	68
f. Contexto navegacional de un transportista	69
g. Diagrama de navegación final.....	70
3.2.4. Cuarta etapa – Diseño de interfaces abstractas.....	71
3.2.5. Quinta etapa – Implementación	79
a. Arquitectura de la aplicación.....	79
b. Modelado de la base de datos	79
c. Conexión a la base de datos	80
d. Uso de JWT para la API-REST en sesión de usuario	81
e. Documentación de la API-REST implementada	87
f. Desarrollo del frontend	93
g. Diseño Web Responsive Design con Bootstrap	105
3.2.6. Pruebas de aceptación.....	113
3.2.7. Implantación de aplicación web	114
CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES	119
4.1. Conclusiones	119
4.2. Recomendaciones	120
Bibliografía	121
ANEXOS	126

ÍNDICE DE FIGURAS

Figura 1: Calculo de Alfa de Cronbach mediante la herramienta SPSS	21
Figura 2: Resultados pregunta 1	22
Figura 3: Resultados Pregunta 2	23
Figura 4: Resultados pregunta 3	24
Figura 5:Resultados pregunta 4.....	25
Figura 6:Resultados pregunta 5.....	26
Figura 7:Resultados pregunta 6.....	27
Figura 8: Resultados pregunta 7.....	28
Figura 9: Resultados pregunta 8.....	29
Figura 10: Resultados pregunta 9.....	30
Figura 11:Resultados pregunta 10.....	31
Figura 12: Ciclo de vida metodología XP.....	46
Figura 13: Etapas de desarrollo metodología OOHDM.....	47
Figura 14:Resumen de procesos ICONIX.....	50
Figura 15: Relación de generalización entre actores.....	54
Figura 16: Casos de uso actores	55
Figura 17: Diseño conceptual.....	63
Figura 18: Diagrama de clase Usuario	64
Figura 19: Diagrama de clase transportista.....	65
Figura 20: Diagrama de clase estudiante	65
Figura 21:Diagrama navegacional: Usuario.....	66
Figura 22:Diagrama navegacional estudiante	67
Figura 23: Diagrama navegacional transportista.....	67
Figura 24:Contexto navegacional usuario.....	68
Figura 25: Diseño navegacional estudiante.....	69
Figura 26:Contexto navegacional Transportista	70
Figura 27: Diagrama navegación final.....	71
Figura 28: ADV Login	72
Figura 29: ADV: Formulario Registro	72
Figura 30:ADV: Cambiar contraseña.....	73
Figura 31: ADV: dashboard	73

Figura 32: ADV: menú lateral	74
Figura 33: ADV header	74
Figura 34: ADV: footer	75
Figura 35: ADV:CardTransportista	75
Figura 36: ADV: transportistas.....	76
Figura 37: ADV: CardRecorrido	76
Figura 38: ADV: Reservas	77
Figura 39: ADV:Busetas	77
Figura 40: ADV: Recorridos	78
Figura 41: ADV:Reservas	78
Figura 42: Arquitectura de la aplicación	79
Figura 43: Base de datos de la aplicación de la aplicación de recorridos y comunicación	80
Figura 44: Conexión Base de datos.....	80
Figura 45: Cadena conexión app.settings	81
Figura 46:Librería JWT	81
Figura 47: Añadir Jwt Bearer	81
Figura 48:Método registrar usuario.....	82
Figura 49: Método autenticar usuario	83
Figura 50: Método creación Token	84
Figura 51: Método cambiar contraseña.....	84
Figura 52:Método encriptar contraseña	85
Figura 53: Método verificar contraseña	85
Figura 54: Método enviar email.....	86
Figura 55:Método enviar email interfaz.....	87
Figura 56: Cadena conexión servidor SMTP.....	87
Figura 57: Autorización de token en los métodos del backend.....	89
Figura 58: UseAuthentication	90
Figura 59: Configurar CORS	90
Figura 60: modelo Usuario.....	91
Figura 61: Modelos backend.....	91
Figura 62:Controladores backend	92
Figura 63:Servicios backend.....	92

Figura 64: Helpers backend	92
Figura 65: Configurar servicios backend	93
Figura 66: Variable conexion al backend	93
Figura 67: Creación de Interceptor	94
Figura 68: Guards para control de rutas	95
Figura 69: providers en app.module.....	95
Figura 70: Obtener el rol de localStorage	96
Figura 71: Variables Behaviour Subject.....	96
Figura 72: Obtener id del token	96
Figura 73: modulo principal Angular	97
Figura 74: Rutas modulo auth	97
Figura 75: Rutas paginas principales	97
Figura 76: Ejemplo servicios Angular.....	98
Figura 77: Modelo Usuario	99
Figura 78: Método validar Nueva Contraseña	99
Figura 79: Arreglos rutas menú lateral.....	100
Figura 80: Contenedor para rol transportista	101
Figura 81: Formulario para registro de usuario	101
Figura 82: Método registrar con consumo de servicio.....	102
Figura 83: Método cargar mapa Mapbox	102
Figura 84: contenedor para el mapa	103
Figura 85: Crear marcador en el mapa	103
Figura 86: Establecer puntos en el mapa.....	104
Figura 87. Dibujar puntos en el mapa	104
Figura 88: Referencias scripts Bootstrap	105
Figura 89:Codigo HTML para el login	106
Figura 90:Codigo HTML para registro	106
Figura 91: Cuerpo modal Bootstrap.....	107
Figura 92: Header página principal.....	107
Figura 93: Breadcrumb Bootstrap.....	108
Figura 94: Tarjeta Bootstrap.....	108
Figura 95: Ejemplo de código CSS para tarjetas	109
Figura 96: Grupo de botones.....	109

Figura 97: Componentes del proyecto	110
Figura 98: Módulo shared	110
Figura 99: Modulo pages	111
Figura 100: Modulo Auth.....	111
Figura 101: Servicios creados	112
Figura 102: Modelos creados	112
Figura 103. Resultados cuestionario Nielsen.....	113
Figura 104. Creación de sitio web en servidor contratado.....	114
Figura 105. Publicación de web api	114
Figura 106. Verificación de publicación	115
Figura 107. Base de datos en sitio web	115
Figura 108. Creación de sitio web en Hostinger	116
Figura 109. Creación de dominio en Hostinger	116
Figura 110. Crear carpeta dist para producción	116
Figura 111. Configuración sitio web en HPanel	117
Figura 112. Pantalla de login en sitio web Hostinger	117
Figura 113. Pantalla home en sitio web Hostinger.....	118

ÍNDICE DE TABLAS

Tabla 1: Formato entrevista.....	18
Tabla 2. Población y muestra	19
Tabla 3. Varianza de los items encuesta aplicada.....	20
Tabla 4. Resultados pregunta 1	22
Tabla 5. Resultados pregunta 2	23
Tabla 6. Resultados pregunta 3	24
Tabla 7. Resultados pregunta 4	25
Tabla 8. Resultados pregunta 5	26
Tabla 9. Resultados pregunta 6	27
Tabla 10. Resultados pregunta 7	28
Tabla 11. Resultados pregunta 8	29
Tabla 12. Resultados pregunta 9	30
Tabla 13. Resultados pregunta 10	31
Tabla 14. Respuestas entrevista aplicada	33
Tabla 15. Análisis de impacto a los involucrados	37
Tabla 16. Requisitos funcionales transportistas	38
Tabla 17. Requisitos funcionales estudiantes.....	38
Tabla 18. Requisitos no funcionales.....	39
Tabla 19. Ventajas y desventajas de Angular	40
Tabla 20. Ventajas y desventajas de Vue JS	41
Tabla 21. Ventajas y desventajas de React JS	41
Tabla 22. Comparación Bootstrap y material Design	43
Tabla 23. Comparación tecnologías backend.....	44
Tabla 24. Resumen metodología OOHDM.....	48
Tabla 25. Comparación metodologías de desarrollo	52
Tabla 26. Actores.....	54
Tabla 27. Caso de uso 1.....	56
Tabla 28. Caso de uso 2.....	56
Tabla 29. Caso de uso 3.....	57
Tabla 30. Caso de uso 4.....	57
Tabla 31. Caso de uso 5.....	58

Tabla 32. Caso de uso 6.....	58
Tabla 33. Caso de uso 7.....	59
Tabla 34. Caso de uso 8.....	60
Tabla 35. Caso de uso 9.....	60
Tabla 36. Caso de uso 10.....	61
Tabla 37. Caso de uso 11.....	61
Tabla 38. Caso de uso 12.....	62
Tabla 39. Caso de uso 13.....	62
Tabla 40. Documentación de API.....	89

RESUMEN EJECUTIVO

A lo largo de los años, el uso de aplicaciones web para la difusión e intercambio de información se ha vuelto imprescindible debido a su utilidad, tanto es así que cuando se habla de ellas está implícito su adaptabilidad y funcionalidad, ya que estas aplicaciones deben poder adaptarse a los dispositivos que las personas dispongan, mostrando la información que sea significativa según sea el caso.

En el presente proyecto se detalla los elementos clave para el desarrollo de una aplicación web responsiva para el intercambio de información que permita ayudar a todas aquellas personas que prestan servicios de transporte estudiantil, los cuales no cuentan con un posicionamiento en plataformas digitales, así como estudiantes que lo utilicen. Para ello se basó en los requerimientos funcionales y no funcionales determinados a través de la recolección de información necesaria con el fin de ofrecer una aplicación adaptable que muestre información relevante y que futuros desarrolladores dispongan de una base al momento de desarrollar aplicaciones web de transporte.

La aplicación web desarrollada aplicando la metodología OOHDM (Método de diseño hipertexto objeto orientado) permitió un correcto seguimiento de las actividades del proyecto, así como las optimizaciones que se presentaron. Esto junto con el framework Angular para el frontend e integrando lenguajes como Bootstrap y Typescript junto con la tecnología ASP.NET Core para la parte del backend, proporcionó una buena funcionalidad en base a los requerimientos planteados.

De esta manera se cumplió el objetivo de la aplicación, donde los transportistas pueden publicar sus recorridos e información y optimizando el proceso de comunicación con los estudiantes.

Palabras clave: Aplicación web, metodología OOHDM, Angular, Bootstrap, transporte estudiantil.

ABSTRACT

Over the years, the use of web applications for the difusión and exchange of information has become essential due to their usefulness, so much so that when we talk about them, their adaptability and functionality is implicit, because these applications must be able to adapt to the different devices that people have, showing all the significant information in each case.

This project shows and details the fundamentals elements for the development of a responsive web application to allows all those who provide student transportation services, which do not have any positioning in digital platforms, as well as students who use that service. This was based on the functional and non-functional requirements determined through the collection of all necessary information in order to provide an adaptable application that shows relevant information and future developers have a grounding when they need to develop transport web applications.

The web application developed by applying the OOHDM (Objected Oriented Hypermedia Design Method) methodology allowed a correct follow-up of the project activities, as well as the optimizations that were presented. All this along with Angular framework for the frontend and integrating programming languages such as Bootstrap and Typescript and ASP.net core technology for the backend, provided a good functionality based on the requirements.

In this way, the principal objective of the application, where carriers can publish their routes and information and optimise the communication with the students, was achieved

Keywords: Web application, OOHDM methodology, Angular framework, Bootstrap, student transport

CAPÍTULO I.- MARCO TEÓRICO

1.1. Tema de Investigación

“APLICACIÓN WEB RESPONSIVE PARA LA GESTIÓN DE RECORRIDOS Y COMUNICACIÓN ENTRE TRANSPORTISTAS PARTICULARES Y ESTUDIANTES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS HUACHI”

1.1.1. Planteamiento del problema

El incremento de la población con acceso a la educación superior a nivel mundial en los últimos años ha generado un gran crecimiento en lo que a transporte se refiere, generando una gran demanda de movilidad por parte de este grupo de personas. Si bien es cierto que la movilidad para estos usuarios en las grandes urbes ha tenido un gran avance respecto a calidad y cantidad, también es cierto que su evolución no se detiene y sus mejoras han sido constantes y hoy en día su optimización se ha vuelto fundamental especialmente en América Latina donde las Tecnologías de la Información juegan un papel importante al momento de mejorar estos servicios, utilizando nuevas herramientas que se adapten a los usuarios y sus necesidades[1].

A nivel de Latinoamérica la gestión de transporte particular por parte de estudiantes en las universidades enfrenta varios retos que se deben analizar: por un lado, el uso de automóviles privados contratados de forma esporádica y mediante terceros hacen que estos servicios se contraten de manera ilegal, supliendo al transporte público. Por otro lado, estos transportes contratados se deben adecuar a la ruta y horario de cada estudiante en particular, en la mayoría de los casos por personas que residen muy lejos de su plantel de estudio. Estas condiciones identificadas en gran parte de ciudades latinoamericanas han provocado que la mayoría de los estudiantes tengan que comunicarse con estos servicios de transporte de parte de terceros o que se gasten excesivas cantidades de dinero en arriendo de mobiliarias, reflejando de manera negativa el desempeño estudiantil ya que estos deben movilizarse constantemente trayendo consigo frustración, ausentismo en clases, deserción, desmotivación,

pérdidas económicas. Todo esto junto con una mala coordinación de comunicación entre estudiantes y transportistas[2].

El crecimiento en el uso de aplicaciones web ha creado la necesidad de desarrollar diferentes programas que emplean o hacen uso de la tecnología Responsive Web Design que permita el diseño de páginas web con una adaptabilidad para los dispositivos móviles y ordenadores de escritorio actuales, dando una interacción con los usuarios más fácil y sencilla. En Ecuador a nivel nacional se hace referencia a la técnica de Responsive Web Design[3], haciendo énfasis en lo útil que son las páginas web adaptativas al momento de gestionar servicios, brindando una opción al momento de desarrollar una página web adaptativa orientada a gestión y comunicación de servicios de transporte a usuarios usando nuevas tecnologías[4].

Actualmente la comunidad universitaria de la Universidad Técnica de Ambato no cuenta con una herramienta de comunicación dedicada este servicio de transporte. La manera en la que los estudiantes acceden a esta es mediante información de terceros, mensajes privados o publicidad en redes sociales. Cabe recalcar que, en estas formas de comunicación con los transportistas no se adjuntan rutas u horarios o alguna información del transportista o del vehículo, así como sus especificaciones, por lo que resulta muy complejo e ineficiente el acceso nuevamente a la misma. Por otra parte, resulta complicado que los estudiantes logren comunicarse para contratar un servicio que se adapte a su disponibilidad como horarios, capacidad o ruta. Esto ocasiona pérdida de tiempo y frustración tanto para el usuario como para el transportista[5].

1.2. Antecedentes investigativos

Después de haber realizado el análisis de fuentes de investigación dentro de repositorios y páginas web educativas se han encontrado los siguientes trabajos:

En su artículo de investigación Olivares Adriana y Orquera María [6] se tiene que: La escasa diversidad funcional y accesibilidad de las zonas residenciales localizadas en los límites urbanos, influyen en el uso de medios motorizados y de largos tiempos de desplazamiento para la realización de la mayor parte de las actividades cotidianas de los estudiantes.

En su tesis Juan Caballero, Nathalia Vergara, Jonathan Coley, Paula Pérez, Iván Asprilla y Francisco Maza [7] se determina que: Los estudiantes anteponen la velocidad y un corto tiempo de recorrido al uso de vehículos particulares como transporte principal para su movilización y en relación con satisfacción y percepción ellos eligen estos transportes debido a los tiempos de espera o transbordo por un medio de transporte, asimismo con la empatía que tienen con el servicio y los avances tecnológicos usados en ellos. Por lo que se concluye que los estudiantes prefieren herramientas útiles para su traslado siempre que estos reduzcan su tiempo y se implemente tecnología acorde a ello.

Diego Samaniego en su proyecto de investigación [8] implementa una aplicación web con diseño responsivo para reserva de servicios de transporte creando una aplicación multiplataforma adaptable a cualquier dispositivo compatible y recomienda el uso de navegadores web actualizados, debido a que las aplicaciones responsivas cuentan con librerías actuales que poseen incompatibilidades con versiones anteriores y también recomienda interactuar con las personas que van a interactuar con la herramienta ya que cada usuario tiene una perspectiva diferente de la misma.

Según Walter Orrala y Roberto Castro en su proyecto [9] determinan que: Gracias a la implementación de un sistema web responsive adecuado se consigue un mejor ambiente de desarrollo para construir los diferentes módulos de la aplicación web que planteó su proyecto y la aplicación web responsive implementada es amigable y permite la facilidad de uso al momento de planificar y dar seguimiento a las visitas áulicas de su sistema.

En su tesis Yohand Malagón, Carlos Niño, Diego Silva y Santiago Vásquez [10] menciona que: la construcción de una aplicación web se aborda a partir de las necesidades de las empresas de servicio de transporte terrestre fundamentándose en los requerimientos funcionales, así como los no funcionales, los requisitos de interfaces del cliente y el hardware. Por lo tanto, se concluye que el principal objetivo de la aplicación es gestionar la información de manera más eficiente ya que la utilización de plataformas y aplicaciones en el ámbito de transporte es cada vez más común y da la posibilidad de obtener más clientes a futuro.

1.3.Fundamentación Teórica

Transporte urbano

El transporte público urbano en general y de forma particular en una ciudad es uno de los ejes eficientes que la mantienen en constante movimiento, ya que cumple la necesidad de las personas de trasladarse de su origen a su destino. Este servicio se basa en actividades que se llevan a cabo por intermedio de una empresa para poder responder e integrar las necesidades de los clientes. El servicio de transporte urbano es un conjunto de elementos intangibles, acciones personales y actitudes que se diseñan para satisfacer las necesidades de movilidad de los usuarios que lo requieran[11].

Transporte particular

Transporte particular se refiere al servicio de transporte que no está abierto o disponible para el público en general. Técnicamente, este tipo de transporte se diferencia del transporte público en tres aspectos: Primero, este tipo de transporte no está sujeto a rutas y el usuario selecciona el camino que considere más conveniente para llegar a su destino. Segundo, no depende de horarios, a diferencia del transporte público donde el horario del viaje está dispuesto a la disponibilidad de los servicios. Tercero, la velocidad es selección del usuario (dentro de las limitaciones del vehículo)[12].

Servicios y rutas de transportistas

Servicios de transporte

El servicio de transporte cumple con la función de transportar mercancía de un lugar determinado a otro. Este proceso es parte de una cadena logística, la cual se dedica a enviar uno o varios productos en el momento y lugar de destino indicado.

Con el transporte siendo una clave para negocios de suministro de mercancía, es imprescindible como empresa contar con una táctica de transporte confiable[13].

Para obtener los mejores resultados de transportación, es necesario que el servicio esté optimizado para el movimiento y manejo de mercancías. Ya sea para llevar la materia prima a proveedores, o el producto a un centro de distribución, o al cliente final, el sistema de transporte siempre es requerido[13].

Rutas de transporte

Una ruta de transporte es el recorrido que realiza el transportista durante la distribución y/o entrega de mercancía. En este sentido, la ruta de transporte debería funcionar de tal forma que la empresa pueda minimizar los costes de transporte y el uso de la flota, tanto como sea posible. También, debería permitir reducir óptimamente la distancia y el tiempo del recorrido de la ruta punto a punto[14].

Tipos de rutas de transporte

Los tipos de rutas de transporte pueden organizarse en dos categorías de acuerdo con el destino:

Ruta de transporte de distribución capilar: en este tipo de ruta el transporte de mercancías llega hasta donde está el cliente final. Esta logística de entrega de última milla se lleva a cabo dentro de los cascos urbanos[14].

Ruta de transporte de larga distancia: son aquellos tipos de rutas de transporte donde grandes cantidades del producto son trasladadas por camiones u otros medios, a nivel nacional e internacional[14].

Por otro lado, de acuerdo con la estrategia de planificación los tipos de rutas de transporte pueden ser:

Rutas de transporte fijas: son rutas de circuito cerrado que no están sujetas a cambio, donde los puntos de recolección y los puntos de entregas pueden ser continuos o discontinuos por un tiempo determinado[14].

Rutas de transporte dinámicas: estas rutas atienden a las distintas necesidades cambiantes del cliente, existiendo la posibilidad de reprogramarse. Es decir, se aprovecha de mejor forma el tiempo, las distancias recorridas y la unidad de transporte de productos[14].

Transporte universitario

El transporte representa una de las mayores necesidades de la sociedad y de la economía, además de ser uno de los motores que ayuda a impulsar su desarrollo ya que comunica y moviliza a las personas, cargas y en general a todos los recursos. De tal forma el servicio de transporte radica en conectar a los trabajadores con sus puestos de trabajo y a los consumidores con los centros de comercio, dinamizando así la economía, consecuentemente ayudando a los estudiantes a moverse a los centros universitarios[5].

Gestión de recorridos y comunicación de transportistas

Gestión de recorridos

La planificación de rutas es el proceso de elegir rutas más estratégicas en el transporte de carga y movilidad de equipos externos. El objetivo principal es determinar la mejor ruta desde el embarque hasta el último cliente del cronograma, incluyendo también todos los puntos de parada y regreso[15].

Esta estrategia se puede implementar en todos los nichos comerciales, independientemente del tamaño de la empresa o de la flota. La idea es democrática y tiene como objetivo aumentar la eficiencia de la entrega, evitando costos, riesgos y obstáculos innecesarios en el camino[15].

Aunque es posible realizar esa planificación manualmente, lo más adecuado es confiar en soluciones tecnológicas de ruteo. Después de todo, las empresas que apliquen esta estrategia manualmente tendrán dificultades para mantenerse al día con los cambios de tráfico en tiempo real, además de perder el potencial de escalabilidad[15].

Comunicación entre transportistas

Para Magallanes (2005) la comunicación en su sentido estricto "es un proceso de doble vía, es decir la relación entre un emisor, receptor, y viceversa". Además, ambos interlocutores utilizan el mismo código. Esta definición da a entender que la comunicación se produce a través de la generación, transmisión y recepción de mensajes[16]

Metodología de desarrollo de software

Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo ha evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. La importancia de este dentro de un proyecto es fundamental, debido a que la metodología aplicada ayuda a definir el plazo del proyecto, presupuesto, actividades y tiempo de revisión de este. Las metodologías de desarrollo de software se las pueden diferenciar por familias, entre ellas están: ágiles, tradicionales e híbridas. Las metodologías ágiles se basan en procesos iterativos y de corta duración, los métodos tradicionales se basan en un control de proceso, en donde se establece las actividades involucradas, el alcance del proyecto, los entregables a generar y las herramientas y seguimiento que se utilizarán, y por último las metodologías híbridas combinan las mejores prácticas de las metodologías tradicionales y ágiles, donde se reúne las ventajas de ambas[17].

Singh, Bagga & Kaur afirman que existen diversos modelos para desarrollar un software que implementa un modelo predictivo, y entre ellos proponen las metodologías tradicionales y ágiles, de los cuales destacan los modelos: Cascada, Espiral, Scrum y XP. Con ellos induce que se debe aplicar una de estas estrategias para que en el futuro el software a desarrollar sea adaptable y presente la menor cantidad de errores[17].

Las metodologías ágiles de desarrollo de software buscan proporcionar en poco tiempo piezas pequeñas de sistemas de software en funcionamiento para mejorar la satisfacción del cliente. Por otro lado, las metodologías tradicionales se caracterizan por definir total y con rigidez los requisitos al inicio de los proyectos de ingeniería de software, y por último las metodologías híbridas explotan las fortalezas de ambas metodologías para conseguir atender un calendario de proyecto predefinido donde la gestión de las tareas sigue una filosofía más distribuida[17].

Web 2.0

La Web 2.0 marca el paso de un sistema rígido de aplicaciones de escritorio focalizado en la información distribuida por servicios centralizados en un sistema flexible de aplicaciones web distribuidas por servicios descentralizados (los sitios pequeños también tienen voz) centrados en el usuario y su colaboración con otros usuarios en la creación de contenidos y en la interacción social. La innovación ya no está sólo al servicio de los grandes desarrolladores, sino incluso de los mismos usuarios: los productos se actualizan continuamente, en aquello que se ha denominado servicios de beta perpetuos[18].

Aplicaciones web adaptativas

El diseño web adaptativo es una buena estrategia para construir una única solución la cual pueda adaptarse, permitiendo al usuario acceder sin dificultad, independientemente de cuál sea el dispositivo que está utilizando para navegar por

ese sitio. Actualmente existe una clara tendencia en el crecimiento del uso de los dispositivos móviles para conectarse a internet[19].

Cada vez menos personas esperan llegar a su casa y sentarse frente a una computadora para consultar los emails, leer alguna noticia o consultar alguna página particular. La disponibilidad y practicidad de los dispositivos móviles hacen que sean la primera fuente de referencia para estos casos. Sin embargo, al querer ingresar a muchos sitios web desde los dispositivos móviles surgen diversos problemas debido a que la mayoría de los sitios no fueron diseñados y estructurados pensando en dispositivos móviles, sino que fueron pensados para ser visualizados desde una computadora. Por ello se recomienda la utilización del enfoque mobile first (pensar primero en el usuario móvil), “es pensar en el móvil como punto de partida, esto obliga a concentrarse en lo esencial de un producto y a hacer foco solo en lo que tiene sentido para este dispositivo”. La utilización del diseño adaptativo permite diseñar la web para pantallas reducidas y adaptarlas luego a pantallas de mayores tamaños[19].

Es importante comprender que el diseño adaptativo no es una versión diferente del sitio web, sino que es el mismo sitio que se va adaptando y va acomodando sus elementos para una mejor visualización en el dispositivo de acceso. Con lo cual no cuenta con todas las ventajas que tendría la construcción de un sitio móvil. El diseño adaptativo se basa en la utilización de hojas de estilo de nivel 3 con cascading style sheets (CSS) y Media Queries lo que permite aplicar distintos estilos según sea el tamaño de la pantalla en la cual se está visualizando el sitio web. De esta forma, una misma página se podrá ver “correctamente” indistintamente del tamaño en la que se esté visualizando. “El diseño web adaptativo utiliza hojas de estilo en cascada complejas, para modificar de forma fluida la apariencia de un sitio web dependiendo del ancho de la ventana de visualización del dispositivo en uso”[19].

Protocolo HTTP

El protocolo HTTP (hypertext transfer protocol) es el protocolo base de la WWW. Se trata de un protocolo simple, orientado a conexión y sin estado. La razón de que esté orientado a conexión es que emplea para su funcionamiento un protocolo de comunicaciones (TCP, transport control protocol) de modo conectado, un

protocolo que establece un canal de comunicaciones de extremo a extremo (entre el cliente y el servidor) por el que pasa el flujo de bytes que constituyen los datos que hay que transferir, en contraposición a los protocolos de datagrama o no orientados a conexión que dividen los datos en pequeños paquetes (datagramas) y los envían, pudiendo llegar por vías diferentes del servidor al cliente[20].

JWT

Basados en las metodologías existentes como Json Web Token es un conjunto de medios de seguridad que permiten gestionar la seguridad de los servicios web de tipo REST. Pese a ser un elemento fundamental a la hora de desarrollar un servicio web, no todas las empresas e instituciones lo aplican debido a desconocimiento o adaptaciones a nuevos estándares de seguridad. El objetivo de la implementación de este estándar es con la finalidad de tratar de mitigar o reducir los riesgos de vulnerabilidad de los servicios web de tipo REST. Los JWT están formados por tres partes[21]:

- Encabezado (header): identifica el algoritmo utilizado para generar, normalmente HS256[21]
- Contenido (payload): Contiene la información sobre la identidad del usuario y sus privilegios[21].
- Firma (signature): se calcula codificando el encabezado y el contenido en base64url, concadenándose ambas partes con un punto[21].

Servicio web

Los servicios web son aplicaciones auto-contenidas, auto-descriptivas y modulares, que pueden ser publicadas, localizadas e invocadas a través del web y que cuentan con un mecanismo estándar para establecer la comunicación con otros tipos de software a través de la red[22].

MySQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones

comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo[23].

Lenguaje UML

El UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación. Un modelo es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo; el modelo omite detalles que no resultan esenciales para la comprensión del original y por lo tanto facilita dicha comprensión[24].

Arquitectura cliente- servidor

La red cliente-servidor funciona como una computadora distribuida. Los usuarios descargan los programas del servidor a sus propias computadoras y ejecutan los programas ahí. Los datos para sus programas también pueden descargarse desde el servidor, ser introducidos por los usuarios o ambos. Si un cálculo requiere una computadora más grande que la que el usuario tiene en su escritorio, entonces el usuario ejecutará ese programa en el servidor, pero en general, la idea detrás de una red cliente servidor es realizar tanto trabajo como sea posible en la computadora de cada usuario[25].

Tipos de diseño web adaptativos

Diseño web líquido o fluido: En este tipo de diseño se usan porcentajes o ems (unidad de medida tipográfica) en vez de píxeles, para que los elementos se adapten según el ancho de la pantalla. Si bien el resultado puede parecer atractivo en pantallas medianas, como computadores de escritorio y tablets, se producen muchos problemas en las pantallas grandes y pequeñas. Por ejemplo, en televisores las imágenes se estiran mucho y en teléfonos los textos son difíciles de leer[26].

Diseño web adaptativo, adaptable o adaptive: Usa plantillas estáticas basadas en puntos de quiebre. Cuando la pantalla alcanza cierto límite de tamaño, se cambia a otro diseño[26].

Diseño web responsivo o Responsive web design: En el responsive web design el diseño y el contenido se adaptan a cada pantalla, entregando una experiencia de usuario muy similar en resoluciones bajas, altas o en formatos de distintas pulgadas. Para lograr esto, los contenidos se ordenan en bloques que se reorganizan según las características de la pantalla y el navegador que se utiliza. Las partes y la jerarquía de los elementos se definen según una serie de parámetros, entre ellos:

- Ancho y alto de la ventana del navegador.
- Orientación del dispositivo.
- Proporción entre el alto y ancho de la pantalla.
- Resolución del dispositivo, es decir, la precisión del detalle en las imágenes de mapa de bits.

Experiencia e interfaz de usuario

Mientras que interfaz de usuario se refiere a la interfaz visual de una herramienta de software, experiencia de usuario se refiere específicamente a la experiencia del usuario mientras la utiliza. El diseño de una interfaz de usuario, más allá de resultar estéticamente agradable, debe tomar en cuenta parámetros a los que en ocasiones no se les da debida atención, lo cual puede perjudicar por completo la experiencia del usuario con la herramienta, ya que antes del agrado visual producido por la experiencia, los usuarios le dan más importancia a su nivel de frustración durante esta[27].

Al diseñar una interfaz de usuario se debe tener como prioridad el obtener una herramienta de uso fácil y conveniencia para el usuario. Jakob Nielsen (2012) define el termino usabilidad como una característica propia de una herramienta de software que debe cumplir con cinco condiciones relevantes:

- **Debe ser fácil de usar:** aprender de forma rápida e intuitiva.
- **Deber ser eficiente:** mejora de la productividad.

- **Fácil de recordar:** recordar el funcionamiento después de cierto tiempo.
- **Baja tasa de error:** ser menos propenso a cometer errores.
- **Agradable de utilizar:** generar satisfacción al usuario[27].

Ventajas de las aplicaciones web responsive

- Elimina los riesgos de crear contenido duplicado.
- Respetar las pautas de Google.
- Mejora las clasificaciones de búsqueda en dispositivos móviles.
- Baja tasa de rebote.
- Mejor experiencia de usuario[28].

1.4.Objetivos

1.4.1. Objetivo General

- Implantar una aplicación web responsive para la gestión de recorridos y comunicación entre transportistas particulares y estudiantes de la Universidad Técnica de Ambato campus Huachi.

1.4.2. Objetivos Específicos

- Analizar los inconvenientes que presentan los estudiantes al momento de comunicarse y gestionar recorridos y horarios que ofrecen los transportistas particulares en los alrededores de la Universidad Técnica de Ambato.
- Establecer los requisitos funcionales que exige una aplicación web responsive enfocada a la gestión de servicio de recorridos de transporte.
- Desarrollar la aplicación web responsive que permita gestionar recorridos y comunicación entre transportistas particulares en los alrededores de la institución educativa.

CAPÍTULO II.- METODOLOGÍA

2.1. Materiales

Este proyecto de investigación tuvo encuestas como instrumento para la recolección de información, estas encuestas se realizaron a los estudiantes que utilizan el servicio de transporte particular establecido alrededor de la universidad para saber cómo se comunican y como gestionan sus recorridos con las diferentes personas encargadas de brindarlo mediante la formulación de 10 preguntas. También se realizaron entrevistas a los choferes con el fin de conocer los métodos de comunicación con el usuario, que herramientas utilizan para publicitar su servicio y su opinión acerca de la implementación de una herramienta web como un complemento para facilitarlos.

2.1.1. Preguntas realizadas para la encuesta

Objetivo: Determinar el beneficio y la opinión sobre la implementación de una aplicación web para la comunicación y gestión de servicios de transporte particular para estudiantes de la Universidad Técnica de Ambato.

1. ¿Considera usted que se debe mejorar el proceso de gestión de recorridos y comunicación entre transportistas particulares y estudiantes de la Universidad Técnica de Ambato?

- a) Totalmente de acuerdo
- b) Muy de acuerdo
- c) De acuerdo
- d) Un poco en desacuerdo
- e) Totalmente en desacuerdo

2. ¿Con que regularidad hace uso de servicios de transporte particular desde su domicilio hacia la universidad?

- a) Todos los días
- b) 3-4 días a la semana

c) 1-2 días a la semana

d) Nunca

3. ¿Como considera la forma de comunicarse y gestionar servicios de transporte particular actualmente?

a) Muy buena

b) Buena

c) Regular

d) Mala

4. ¿Considera usted que la falta de opciones al poder contratar un servicio de transporte hacia la universidad puede afectar su desempeño estudiantil?

a) Si

b) No

c) No estoy seguro

5. ¿Qué dispositivo electrónico utiliza con más frecuencia? Seleccionar una o varias:

a) Teléfono celular

b) Laptop

c) Tablet

d) Otros

6. ¿Alguna vez ha tenido problemas al momento de querer contratar un servicio de transporte particular alrededor de la universidad?

a) Totalmente de acuerdo

b) Muy de acuerdo

c) De acuerdo

d) Un poco en desacuerdo

e) Totalmente en desacuerdo

7. ¿Conoce algún software destinado a la gestión de recorridos de transporte particular?

a) Si

b) No

8. ¿Considera usted que es necesario implementar una aplicación para la gestión de recorridos y comunicación entre estudiantes de la UTA y transportistas particulares?

a) Totalmente de acuerdo

b) Muy de acuerdo

c) De acuerdo

d) Un poco en desacuerdo

e) Totalmente en desacuerdo

9. ¿Considera usted que una aplicación web para la gestión de recorridos y comunicación entre transportistas particulares y estudiantes ayudará a organizar y reducir su tiempo de movilización hacia la Universidad Técnica de Ambato?

a) Totalmente de acuerdo

b) Muy de acuerdo

c) De acuerdo

d) Un poco en desacuerdo

e) Totalmente en desacuerdo

10. ¿Cree usted que el uso de una aplicación ayudará a que más estudiantes puedan acceder a estos servicios de transporte?

a) Totalmente de acuerdo

b) Muy de acuerdo

c) De acuerdo

d) Un poco en desacuerdo

e) Totalmente en desacuerdo

2.1.2. Formato de entrevista aplicada a los transportistas

El objetivo de la entrevista fue determinar la forma en que los transportistas se comunican con sus usuarios para ofrecer sus servicios de transporte y sus recorridos

Preguntas	Respuesta	Observación
¿Qué herramientas tecnológicas utilizan para promocionar recorridos?		
¿Cuáles son las formas de dar a conocer los distintos recorridos y horarios que puede ofrecer su servicio de transporte?		
¿Considera que una herramienta destinada a gestionar su servicio ayude a llegar a más estudiantes y poder generar más ingresos?		
¿Alguna vez ha tenido un problema con los recorridos de estudiantes? ¿Qué problemas?		
¿Sus recorridos varían o siempre siguen una ruta preestablecida?		
¿Qué opciones disponen los estudiantes si quieren comunicarse para contratar su servicio?		
¿Su servicio permite reservar recorridos?		

Tabla 1: Formato entrevista

2.2. Métodos

La investigación se contextualizó en la modalidad de investigación de campo y bibliográfica-documental.

2.2.1. Modalidad de la Investigación

a. Investigación bibliográfica

Se realizó una investigación bibliográfica – documental donde se recopiló información referente al diseño responsivo para aplicaciones web, así como de herramientas útiles para el desarrollo de esta.

b. Investigación de campo

Se realizó una investigación de campo, esta modalidad fue necesaria ya que debido al contexto de la investigación fue necesario recopilar información de los estudiantes involucrados y su forma de interactuar con los transportistas.

c. Investigación aplicada

Se empleó los conocimientos adquiridos durante el transcurso de la carrera para poder alcanzar los objetivos propuestos en este proyecto.

2.2.2. Población y muestra

Esta investigación trabajó con un pequeño grupo de la población estudiantil que reside en sectores alejados de la institución o que conocen de estos servicios de transporte, así como con los transportistas que se encargan del transporte a los mismos. Esta fue la población por estudiar. Como estos grupos son inferiores a cien por cada recorrido no se realizó la muestra.

Población	Numero	Porcentaje
Estudiantes	40	86.95%
Transportistas - choferes	6	13.04%
Total	46	100%

Tabla 2. Población y muestra

2.2.3. Recolección de información

La recolección de información para este proyecto se realizó mediante formularios de Google, debido a que el alcance, recolección e interpretación de datos son facilitados por la plataforma, también se realizó entrevistas a los choferes para recolectar

información de cómo se realiza la comunicación y forma de gestión de recorridos en su servicio. Esta información se validó mediante el cálculo de Alfa de Cronbach.

Validación del instrumento

Para validar el instrumento de recolección de datos se utilizó Alfa de Cronbach, este es un coeficiente de tipo estadístico que permite validar instrumentos con denominación cuantitativa, es decir datos medibles.

Para calcular el coeficiente se utiliza la formula:

$$\alpha = \frac{K}{K - 1} \left[1 - \frac{\sum \sigma_i^2}{\sigma^2} \right]$$

Donde:

K = número de ítems.

σ_i^2 = varianza de cada ítem.

σ^2 = varianza total.

Aplicación

Se convirtió las respuestas de la encuesta a una escala de Likert en donde 5 es la respuesta de mayor importancia y 1 la de menor importancia. Se tomaron en cuenta solo las preguntas efectuadas con esa escala.

Pregunta	Varianza del ítem	Varianza total
1	0.3348	.814
4	0.6905	
8	0.9296	
9	0.7755	
10	0.4947	
Total	3.22	

Tabla 3. Varianza de los ítems encuesta aplicada

➔ **Fiabilidad**

Escala: Alfa

Resumen de procesamiento de casos

		N	%
Casos	Válido	50	89,3
	Excluido ^a	6	10,7
	Total	56	100,0

a. La eliminación por lista se basa en todas las variables del procedimiento.

Estadísticas de fiabilidad

Alfa de Cronbach	N de elementos
,814	5

Estadísticas de elemento

Figura 1: Calculo de Alfa de Cronbach mediante la herramienta SPSS

Debido a que el alfa de Cronbach es de 0.814 se considera que esta dentro de los rangos aceptables del instrumento.

a. Análisis e interpretación de encuesta aplicada a estudiantes

La encuesta fue aplicada a un total de 49 estudiantes de la Universidad Técnica de Ambato, obteniendo la siguiente información:

Resultados de la pregunta 1

Pregunta 1	Respuestas	Frecuencia	Porcentaje
¿Considera usted que se debe mejorar el proceso de gestión de recorridos y comunicación entre transportistas particulares y estudiantes de la Universidad Técnica de Ambato?	Totalmente de acuerdo	36	73.5%
	Muy de acuerdo	12	24.5%
	De acuerdo	0	0%
	Un poco en desacuerdo	1	2%
	Totalmente en desacuerdo	0	0%
Total		49	100%

Tabla 4. Resultados pregunta 1

1. ¿Considera usted que se debe mejorar el proceso de gestión de recorridos y comunicación entre transportistas particulares y estudiantes de la Universidad Técnica de Ambato?

49 respuestas

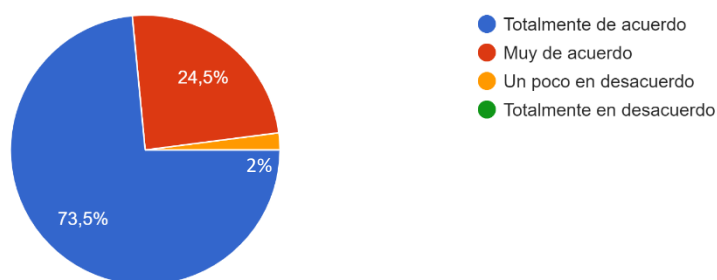


Figura 2: Resultados pregunta 1

Análisis e interpretación de resultados

Según la encuesta aplicada el 73.5% de estudiantes están totalmente de acuerdo en que se debe mejorar el proceso de gestión de recorridos y una mejor comunicación entre ellos junto con el 24% restante de los estudiantes encuestados. Esto sugiere que a pesar de que existen formas de poder gestionar este servicio los estudiantes están abiertos a optar por otras herramientas ya que el manejo actual no es el más adecuado o el más óptimo.

Resultados de la pregunta 2

Pregunta 2	Respuestas	Frecuencia	Porcentaje
¿Con que regularidad hace uso de servicios de transporte particular desde su domicilio hacia la universidad?	Todos los días	16	32.7%
	3-4 días a la semana	11	22.4%
	1-2 días a la semana	12	24.5%
	Nunca	10	20.4%
Total		49	100%

Tabla 5. Resultados pregunta 2

2. ¿Con que regularidad hace uso de servicios de transporte particular desde su domicilio hacia la universidad?

49 respuestas

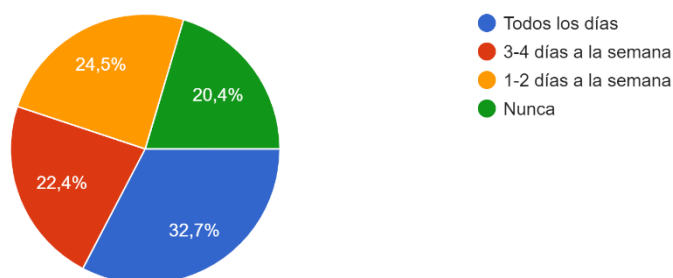


Figura 3: Resultados Pregunta 2

Análisis e interpretación de resultados

Según la encuesta aplicada el 32.7% de estudiantes hace uso de este servicio todos los días, junto con el 22.4% que los hacen de 3 a 4 veces por semana y el 24.5% de 1 a 2 días por semana. Esto quiere decir que los estudiantes usan frecuentemente los servicios de transporte particular alrededor de la universidad. Sin embargo, el 20.4% nunca usa este servicio ya sea por desinformación o falta de herramientas para poder acceder a este.

Pregunta 3:

Pregunta 3	Respuestas	Frecuencia	Porcentaje
¿Como considera la forma de comunicarse y gestionar servicios de transporte particular actualmente?	Muy buena	8	16.3%
	Buena	17	34.7%
	Regular	18	36.7%
	Mala	6	12.2%
Total		49	100%

Tabla 6. Resultados pregunta 3

3. ¿Cómo considera la forma de comunicarse y gestionar servicios de transporte particular actualmente?

49 respuestas

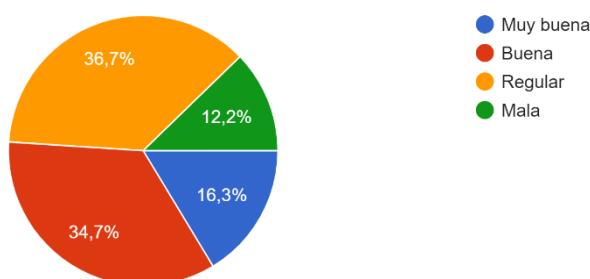


Figura 4: Resultados pregunta 3

Análisis e interpretación de resultados

Según la encuesta aplicada el 36.7% de estudiantes consideran que la forma de gestionar recorridos actualmente es regular junto al 34.4% considerándola buena, pero considerando que podría ser más eficiente. El 12.2% considera que es mala. Por lo que se determina que una gran parte de estudiantes consideran que, aunque existen formas de comunicación con el transportista y su servicio, esta interacción se puede hacerla de manera más eficiente.

Pregunta 4:

Pregunta 4	Respuestas	Frecuencia	Porcentaje
¿Considera usted que la falta de opciones al poder contratar un servicio de transporte hacia la universidad puede afectar su desempeño estudiantil?	Si	31	63.3%
	No	7	14.3%
	No estoy seguro	11	22.4%
Total		49	100%

Tabla 7. Resultados pregunta 4

4. ¿Considera usted que la falta de opciones al poder contratar un servicio de transporte hacia la universidad puede afectar su desempeño estudiantil?

49 respuestas

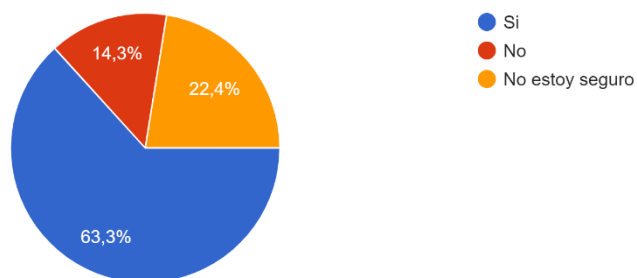


Figura 5:Resultados pregunta 4

Análisis e interpretación de resultados

Según la encuesta aplicada el 63.3% de estudiantes de la universidad consideran que la falta de opciones al poder contratar un servicio de transporte puede afectar su desempeño en calidad de estudiante. El 22.4% dice no estar seguro si esto puede afectar su desempeño y un pequeño porcentaje de 14.3% considera que esto no afecta directamente su desempeño. Por lo tanto, se determina que la falta de opciones al poder contratar un servicio de transporte puede afectar su desempeño en el ámbito estudiantil universitario.

Pregunta 5

Pregunta 5	Respuestas	Porcentaje
¿Qué dispositivo electrónico utiliza con más frecuencia? Seleccionar uno o varios	Teléfono celular	100%
	Laptop	38.8%
	Tablet	2%
	Otros	2%

Tabla 8. Resultados pregunta 5

5. ¿Qué dispositivo electrónico utiliza con más frecuencia? Seleccionar uno o varios
49 respuestas

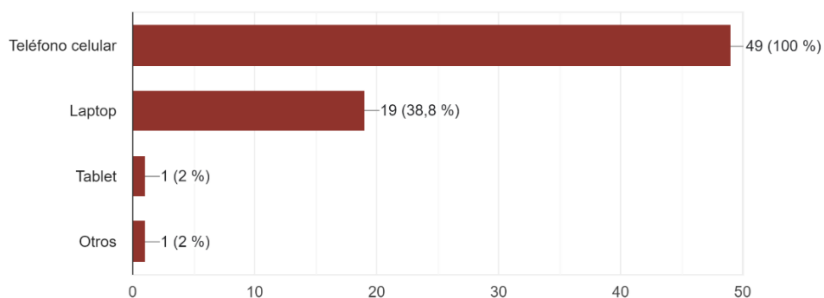


Figura 6: Resultados pregunta 5

Análisis e interpretación de resultados

Según los resultados de la pregunta 5, el 100% de los encuestados respondió que su dispositivo de uso más frecuente es su teléfono celular, el 38.8% también respondió que su dispositivo de uso más frecuente es su laptop junto con un 2% que menciona que también usan tabletas como dispositivo. Por lo tanto, se determina que, aunque la mayoría de los estudiantes utilicen celulares en su día a día también hacen uso de otros dispositivos haciendo que una aplicación web sea una buena opción ya que no se limita a un solo dispositivo y puede funcionar en otros como tabletas o computadoras portátiles.

Pregunta 6:

Pregunta 6	Respuestas	Frecuencia	Porcentaje
¿Alguna vez ha tenido problemas al momento de querer contratar un servicio de transporte particular alrededor de la universidad?	Totalmente de acuerdo	7	14.3%
	Muy de acuerdo	12	24.5%
	De acuerdo	19	38.8%
	Un poco en desacuerdo	7	14.3%
	Totalmente en desacuerdo	4	8.2%
Total		49	100%

Tabla 9. Resultados pregunta 6

6. ¿Alguna vez ha tenido problemas al momento de querer contratar un servicio de transporte particular alrededor de la universidad?

49 respuestas

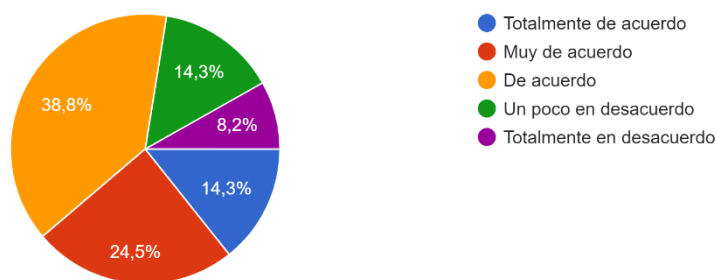


Figura 7: Resultados pregunta 6

Análisis e interpretación de resultados

Según la encuesta aplicada el 38.8% de estudiantes encuestados están de acuerdo con que cuando han hecho uso o han querido contratar servicios de transporte alrededor de la universidad se han encontrado con diversos problemas. El 24.5% de encuestados están muy de acuerdo con que han surgido problemas al contratarlo y el 14.3% están totalmente de acuerdo con que definitivamente han tenido algún problema. Esto determina que existen problemas con el modelo actual de gestión de servicio de transporte particular en la universidad y que es frecuente que existan inconvenientes.

Pregunta 7:

Pregunta 7	Respuestas	Frecuencia	Porcentaje
¿Conoce algún software destinado a la gestión de recorridos de transporte particular?	Si	9	18.4%
	No	40	81.6%
Total		49	100%

Tabla 10. Resultados pregunta 7

7. ¿Conoce algún software destinado a la gestión de recorridos de transporte particular?

49 respuestas

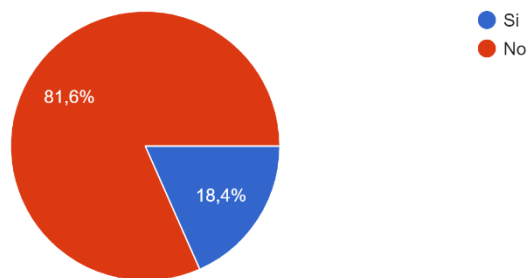


Figura 8: Resultados pregunta 7

Análisis e interpretación de resultados

Según los resultados de la encuesta el 81.6% de estudiantes no conocen algún software destinado al servicio de transporte en la universidad. El 18.4% de estudiantes si conocen de alguna herramienta. Esto quiere decir que un gran porcentaje de estudiantes desconoce que existe alguna herramienta de software que permita acceder a los estudiantes de la universidad a los servicios que brindan los transportistas.

Pregunta 8

Pregunta 8	Respuestas	Frecuencia	Porcentaje
¿Considera usted que es necesario implementar una aplicación para la gestión de recorridos y comunicación entre estudiantes de la UTA y transportistas particulares?	Totalmente de acuerdo	28	57.1%
	Muy de acuerdo	8	16.3%
	De acuerdo	12	24.5%
	Un poco en desacuerdo	0	0%
	Totalmente en desacuerdo	1	2%
Total		49	100%

Tabla 11. Resultados pregunta 8

8. ¿Considera usted que es necesario implementar una aplicación para la gestión de recorridos y comunicación entre estudiantes de la UTA y transportistas particulares?

49 respuestas

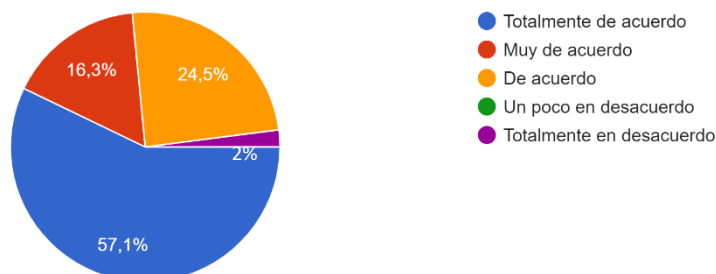


Figura 9: Resultados pregunta 8

Análisis e interpretación de resultados

Según la encuesta aplicada el 57.1% de estudiantes consideran que están totalmente de acuerdo en que se implemente una aplicación que permita comunicarse de mejor manera con los transportistas y sus servicios. El 16.3% respondieron que están muy de acuerdo con que se implemente una aplicación y el 24.5% también están de acuerdo. Por lo tanto, es evidente que la mayoría de los estudiantes considere que la manera de gestionar este servicio de transporte se lleve a cabo mediante una aplicación web interactiva que contenga diversos elementos para poder acceder a la misma.

Pregunta 9

Pregunta 9	Respuestas	Frecuencia	Porcentaje
¿Considera usted que una aplicación web para la gestión de recorridos y comunicación entre transportistas particulares y estudiantes ayudará a organizar y reducir su tiempo de movilización hacia la Universidad Técnica de Ambato?	Totalmente de acuerdo	27	55.1%
	Muy de acuerdo	10	20.4%
	De acuerdo	11	22.4%
	Un poco en desacuerdo	1	2%
	Totalmente en desacuerdo	0	0%
Total		49	100%

Tabla 12. Resultados pregunta 9

9. ¿Considera usted que una aplicación web para la gestión de recorridos y comunicación entre transportistas particulares y estudiantes ayudará ...vilización hacia la Universidad Técnica de Ambato?

49 respuestas

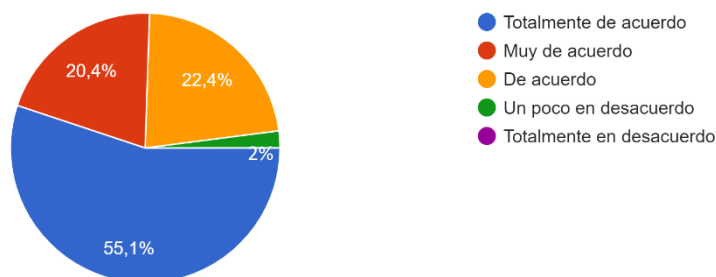


Figura 10: Resultados pregunta 9

Análisis e interpretación de resultados

Según la encuesta aplicada a los estudiantes, el 55.1% de estos consideran que al implementar una aplicación va a ayudar a que su tiempo de movilización hacia la universidad se reduzca ya que evitarán perder tiempo al querer comunicarse con los transportistas. Esto junto con el 20.4% que están muy de acuerdo y el 22.4% que están de acuerdo. Existe un pequeño porcentaje del 2% que está un poco en desacuerdo. Esto evidencia que la implementación de la aplicación puede permitir al estudiante a que organice mejor su tiempo y tenga opciones al querer movilizarse hacia la universidad

Pregunta 10

Pregunta 1	Respuestas	Frecuencia	Porcentaje
¿Cree usted que el uso de una aplicación ayudará a que más estudiantes puedan acceder a estos servicios de transporte?	Totalmente de acuerdo	31	66.3%
	Muy de acuerdo	12	24.5%
	De acuerdo	6	12.2%
	Un poco en desacuerdo	0	0%
	Totalmente en desacuerdo	0	0%
Total		49	100%

Tabla 13. Resultados pregunta 10

10. ¿Cree usted que el uso de una aplicación ayudará a que más estudiantes puedan acceder a estos servicios de transporte?

49 respuestas

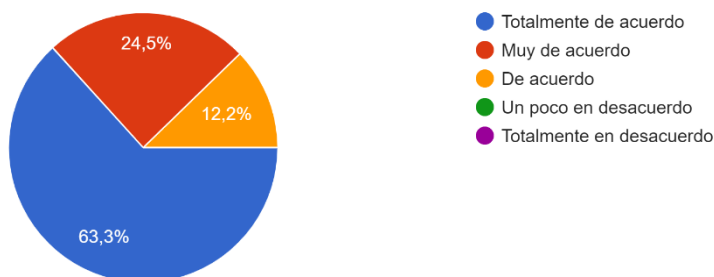


Figura 11: Resultados pregunta 10

Análisis e interpretación de resultados

Según los resultados obtenidos en la encuesta el 63.3% respondió que están totalmente de acuerdo en que la implementación y uso de una aplicación web atraerá a más estudiantes hacia el uso de estos servicios junto con el 24.5% que están muy de acuerdo en que se atraerá más estudiantes que lo necesiten y el 12.2% que opinan que están de acuerdo que con esta aplicación se llegara a más personas. Po lo que se determina que la implantación de la aplicación atraerá a más estudiantes que lo necesiten.

b. Resultados de las entrevistas aplicadas a transportistas

A continuación, se detalla un resumen general de las respuestas en las entrevistas dirigidas y aplicadas a los transportistas particulares establecidos alrededor de la universidad:

Preguntas	Respuestas	Observaciones
¿Qué herramientas tecnológicas utilizan para promocionar recorridos?	Usan principalmente redes sociales como WhatsApp y Facebook. Estas herramientas tienen grupos de estudiantes donde promocionan sus servicios. Los estudiantes comparten información de contacto entre sí.	Actualmente no se cuenta con una herramienta específica para comunicación de estudiantes y transportistas.
¿Cuáles son las formas de dar a conocer los distintos recorridos y horarios que puede ofrecer su servicio de transporte?	Mediante publicaciones y estados en redes sociales. La universidad apoya a estos servicios, pero no tienen nada oficial. Una opción sería poder publicar los recorridos en la página oficial de la universidad.	La manera en la que se publicita información de rutas no es la más óptima.
¿Considera que una herramienta destinada a gestionar su servicio ayude a llegar a más estudiantes y poder generar más ingresos?	Una herramienta que permita llegar a más estudiantes es de mucha utilidad para poder brindar un mejor servicio a todo el que lo necesite.	Las personas que brindan el servicio están abiertos a nuevas herramientas para el conocimiento de este a los estudiantes.

¿Alguna vez ha tenido un problema con los recorridos de estudiantes? ¿Qué problemas?	Los transportistas tienen contratos con la agencia de tránsito de Ambato. al comienzo se tenían problemas con los agentes, pero ya se solucionó y con la ayuda del rector de la universidad se puede brindar este servicio al estudiante que lo requiera.	Existen dificultades al momento de brindar el servicio.
¿Sus recorridos varían o siempre siguen una ruta preestablecida?	Los recorridos son preestablecidos. Varían de acuerdo con el horario o al conductor. Existen recorridos en la mañana, tarde y noche hacia distintos destinos.	Varios recorridos, pero poca información a los estudiantes sobre estos. Los recorridos se pueden publicitar de mejor manera.
¿Qué opciones disponen los estudiantes si quieren comunicarse para contratar su servicio?	Se promociona el contacto y la ruta por medio de WhatsApp. También se usa Facebook. Los estudiantes comparten la información de contacto.	Los medios de comunicación entre estudiante-transportista son tradicionales.
¿Su servicio permite reservar recorridos?	Si, el estudiante se contacta y se puede reservar siempre y cuando exista disponibilidad.	Se pueden reservar recorridos por medio del contacto directo al transportista.
<p>Conclusión:</p> <p>La forma de comunicación entre estudiante-transportista es básica, lo cual no permite que se promocioe de la mejor manera un recorrido en específico o que nuevos estudiantes puedan acceder a información de este servicio. Esto genera que no se tenga una información centralizada y que los estudiantes desconozcan como se puede acceder a los distintos recorridos que hay o su disponibilidad.</p>		

Tabla 14. Respuestas entrevista aplicada

2.2.4. Procesamiento y análisis de datos

En base a la información recolectada mediante las encuestas y entrevistas, una vez analizada objetivamente la información se determina que:

- Los estudiantes en su mayoría utilizan redes sociales como fuente de información ya que no existe una herramienta que brinde información sobre los servicios de transporte particular alrededor de la universidad.
- Los estudiantes de nuevo ingreso no hacen uso de este servicio hasta avanzado su transcurso en la carrera por desconocimiento o porque no conocen las rutas, horarios o números telefónicos que ofertan los transportistas particulares, pudiendo llegar a afectar el desempeño estudiantil ya que la página oficial de la universidad no promociona este tipo de servicio.
- Aunque las personas que brindan transporte están abiertas a que existan mejores formas de comunicación y gestión del servicio, todavía no existen muchos métodos, así como herramientas para que esta información llegue al estudiante que lo requiera.
- Aunque el medio más seleccionado para la comunicación entre involucrados fueron redes sociales, los transportistas están abiertos a que una aplicación web destinada a ofertar su servicio sea un complemento que ayude a llegar a más estudiantes y así establecer una mejor comunicación.
- Es necesario llevar un control adecuado de los recorridos que ofrecen los transportistas y en que horarios están disponibles. Es decir que esta información sea de rápido acceso al estudiante y eficaz al momento de acceder a ella.
- Es importante que la herramienta creada para el propósito de este proyecto siga pautas de diseño ya que será diseñada para poder funcionar en cualquier dispositivo que el estudiante posea.

CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de resultados

Luego de obtener la información mediante la aplicación de instrumentos de recolección se pudo determinar las formas de comunicarse entre estudiantes y transportistas particulares de la universidad, cuándo se solicita el servicio y la forma de hacerlo. Se evidencia las opciones que se tiene al momento de gestionar este servicio, se busca minimizar el tiempo y centralizar las opciones al momento de registro de información de los usuarios, así como de choferes y sus recorridos y brindar una información eficaz. Por lo tanto, existe la necesidad de contar con una aplicación web responsiva la cual ayude al registro de esta información concerniente a la gestión y comunicación del servicio de transporte mejorando el intercambio de información y la comunicación entre implicados.

3.1.1. Descripción del análisis de impacto de inconvenientes a los involucrados

El alcance a estudiantes y usuarios del transporte particular se basa en como este llega a las personas que consumen este servicio. Cada problema o inconveniente que se presenta en la comunicación entre el estudiante o el transportista tiene un impacto en estos, ya sea individualmente o a cada uno en específico que se detalla a continuación:

Estudiantes: usuarios del servicio de transporte.

Transportistas: propietarios de vehículos particulares que ofertan el servicio.

Problema	Impacto en el transportista	Impacto en los estudiantes	Nivel
Comunicación elemental	La comunicación a través de redes sociales no difunde aspectos detallados sobre el servicio. Los transportistas divulgan su información mediante mensajes directos sin información detallada.	La comunicación elemental hace que nuevos estudiantes desconozcan información sobre el servicio y su accesibilidad, así como el precio, registro y disponibilidad	Alto
Escasa información de transportistas	Los transportistas no difunden su información como: nombre, teléfono, horario, capacidad de su vehículo, ubicación.	Los estudiantes no pueden optar por tomar decisiones al momento de contratar el servicio de acuerdo con la información disponible del transportista	Medio
Publicación y acceso a información de rutas disponibles	El transportista puede tener varios recorridos el mismo día. No se puede ofertar sus horarios detalladamente por medio de mensajes directos	El estudiante no dispone de una fuente de información eficiente. Si el transportista varía su recorrido este debe volver a comunicarse nuevamente.	Alto
Acceso y gestión de recorridos	El transportista debe crear y mostrar sus recorridos. Esto se puede centralizar en la aplicación	Los estudiantes no gestionan sus recorridos y solo pueden acceder a los que se ofertan por mensajes o grupos	Alto
Disponibilidad de horarios	La disponibilidad de horarios del transportista está integrada a los recorridos. La disponibilidad del transportista se puede publicar en la aplicación.	Los estudiantes pueden comunicarse de mejor manera con el transportista si hay una manera directa de hacerlo.	Medio
Visualización de usuarios	El transportista no almacena gran información de los usuarios que usan su servicio. Se puede gestionar de mejor manera la información de sus usuarios	Los estudiantes no pueden almacenar su información para que otros transportistas accedan a ella. Así pudiendo ofrecer su servicio mediante la información de residencia del estudiante	Medio

Referencias acerca del servicio	El transportista no dispone de un sitio donde poder acceder a reseñas.	El estudiante actualmente no puede comentar sobre los recorridos para que otros estudiantes tengan referencias acerca de ello.	Bajo
---------------------------------	--	--	------

Tabla 15. Análisis de impacto a los involucrados

Según la tabla de impactos se puede determinar que la aplicación web va a ser destinada a manejar dos roles: el rol de transportista que se define como el propietario del vehículo particular y los clientes que son los estudiantes. En base a estas observaciones se pueden establecer los requisitos funcionales y los requisitos no funcionales responsivos que se describen a continuación:

3.1.2. Requisitos funcionales y no funcionales para el desarrollo de la aplicación web responsive

a. Requisitos funcionales para transportistas

De acuerdo con los requerimientos recolectados a través de la recolección de datos mediante encuestas, entrevistas aplicadas y tabla de impactos se pudieron determinar los requisitos funcionales que se pueden implementar en la aplicación web las cuales se detallan en la siguiente tabla:

Requisito	Descripción del requisito
Registro de información del transportista	Los transportistas que utilicen la aplicación deben registrar su información básica como: nombre, teléfono, recorridos que ofertan, ubicación.
Publicar recorridos	El transportista puede publicar el recorrido además del horario de salida, precio y disponibilidad.
Hacer reservas	La aplicación ofrece al estudiante la capacidad de reservar un recorrido que el transportista oferte.
Gestionar recorridos	El transportista puede agregar todos los recorridos que ofertan a diferentes rutas

Acceptar recorridos	La aplicación permite al transportista aceptar los recorridos reservados de los estudiantes.
Visualizar reseñas	El transportista puede observar comentarios acerca de las rutas.

Tabla 16. Requisitos funcionales transportistas

b. Requisitos funcionales para estudiantes

De acuerdo con la tabla de impactos, así como la información recolectada mediante investigación y en las encuestas aplicadas se determinaron requisitos funcionales para implementar en la aplicación que se detallan a continuación:

Requisito	Descripción del requisito
Visualizar información del transportista y sus recorridos	permitir al estudiante visualizar información relevante referente al transportista, además de constatar sus recorridos disponibles.
Comunicación con el transportista	permitir al estudiante poder comunicarse directamente con el transportista mediante notificaciones
Búsqueda personalizada	Los estudiantes pueden revisar los recorridos que han publicado los transportistas en la aplicación.
Registrar comentarios	El estudiante puede dejar comentarios acerca de las rutas de transporte utilizado.
Ubicación de recorridos	La aplicación debe permitir la visualización de los recorridos a través de un mapa.

Tabla 17. Requisitos funcionales estudiantes

c. Requerimientos no funcionales para el desarrollo de la aplicación.

Los requisitos no funcionales van a representar las características generales y restricciones que tendrá la aplicación. Estos requisitos servirán para evaluar la operación de los servicios que va a manejar la aplicación web responsiva. Con el diseño web responsive se redimensionará y colocará los elementos de la web para que se adapten al ancho del dispositivo para permitir una correcta visualización y una mejor experiencia de usuario, así también la integridad de los datos.

Requisito	Descripción del requisito
Diseño adaptable	<ul style="list-style-type: none"> • La aplicación web contará con un front-end responsive que se adapte al tamaño de pantalla del dispositivo utilizado. El contenido destacable debe estar disponible rápidamente • Todos los elementos gráficos deben estar optimizados (Imágenes, mapas, texto) de acuerdo con el tamaño de pantalla • La interfaz gráfica de la aplicación debe disponer de ventanas con botones, listas y campos de texto adaptables que serán visualizados desde navegadores de Internet más utilizados (Chrome, Edge o Firefox)
Usabilidad	<ul style="list-style-type: none"> • La interfaz por tipo de usuario debe ser cómoda y sencilla de utilizar, • La interfaz debe dividirse por secciones según la información requerida y el tipo de pantalla • La optimización del texto y los elementos gráficos deben estar diseñados de acuerdo con la página para evitar errores • Uso de tonalidades y colores para remarcar secciones importantes y diseño agradable. • Evitar efectos para evitar tiempos de carga. • La aplicación debe utilizar iconos representativos. • Tipografía de acuerdo con la aplicación. • Creación de mensaje de errores para la aplicación.
Eficiencia	<ul style="list-style-type: none"> • Consulta de información publicada en la aplicación • Ver recorridos
Seguridad	<ul style="list-style-type: none"> • Conexiones seguras para las bases de datos. • Sistema de autenticación por usuarios para restringir la información mediante perfiles utilizando Json Web Token • Garantizar la correcta entrada de datos
Integración	<ul style="list-style-type: none"> • La aplicación debe poder usarse en varios dispositivos de acuerdo con el tamaño de la pantalla o al navegador utilizado para acceder a esta.

Tabla 18. Requisitos no funcionales

Una vez identificados los requisitos funcionales se puede establecer las herramientas para el desarrollo de la aplicación: Frameworks front-end y tecnologías back-end, establecimiento de servidores de alojamiento de la aplicación, así como la metodología para el desarrollo de la propuesta.

3.1.3. Frameworks de desarrollo front-end

a. Angular

Angular es un framework desarrollado por Google y lanzado en 2010. Es un marco de JavaScript basado en TypeScript y posee un marco estructural para aplicaciones web dinámicas. Angular posee un gran número de librerías que ayudan a la facilitación de codificar procesos, su lenguaje TypeScript permite crear páginas web dinámicas e interactivas haciéndolas mucho más atractivas del lado del cliente[29].

Con la implementación de TypeScript, HTML, CSS, se van a realizar toda la parte visual del lado del cliente y servicios Ajax para hacer peticiones a los procesos realizados en el servidor estructurado con Node.js y las librerías para su correcto y óptimo funcionamiento[29].

Ventajas	Desventajas
<ul style="list-style-type: none"> • Evita el código repetitivo y proporciona consistencia al código. • Emplea buenas prácticas de programación. • Se puede integrar y reutilizar sus componentes reduciendo el tiempo de creación de una aplicación. • Soporta variedad de entornos de desarrollo. 	<ul style="list-style-type: none"> • Barrera de idioma. • Recursos limitados. • Se requiere bastante conocimiento para poder manejarla correctamente. • Falta de desarrolladores experimentados. • Aplicaciones dinámicas no tienen buen rendimiento.

Tabla 19. Ventajas y desventajas de Angular

b. Vue.js

Vue es un framework open source de JavaScript que permite elaborar interfaces de usuario de forma sencilla. Su curva de aprendizaje es sencilla, sin embargo, se debe dominar el lenguaje JavaScript, así como conocer el funcionamiento de callbacks,

promesas, objetos, entre otros. Vue simplifica el desarrollo web de aplicaciones y sitios web. Al ser un framework progresivo este tiene la capacidad de incluir las partes necesarias encima de la biblioteca central y luego incluir otras bibliotecas como si fueran módulos separados además de incluir un ecosistema variado que cubre todas las necesidades del desarrollo[30].

Ventajas	Desventajas
<ul style="list-style-type: none"> • Fácil de aprender • Sistema de herramientas sólido. • Capacidad de integración y flexibilidad • Soporte comunitario • Documentación y soporte. 	<ul style="list-style-type: none"> • Barrera de idioma (chino e inglés) • No es adecuado para proyectos a gran escala • Flexibilidad excesiva • Recursos limitados

Tabla 20. Ventajas y desventajas de Vue JS

c. React JS

React es una biblioteca JavaScript de código abierto diseñada para crear interfaces de usuario para facilitar el desarrollo de aplicaciones en una sola página. Este framework ayuda a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser muy sencillo y fácil de combinar. React maneja la interfaz de usuario en una aplicación usando el contexto MVC (modelo-vista-controlador). También puede ser utilizado con sus extensiones React-based que se encargan de las partes no-UI (no forman parte de la interfaz) de una aplicación web[31].

Ventajas	Desventajas
<ul style="list-style-type: none"> • Framework ligero: los datos del usuario pueden representarse en el servidor al mismo tiempo • Sencilla migración entre versiones • DOM virtual • No se debe aprender typescript como en Angular 	<ul style="list-style-type: none"> • Ausencia de documentación • No tiene un estándar de desarrollo • Mucho tiempo de aprendizaje, se requiere profundo conocimiento de la herramienta

Tabla 21. Ventajas y desventajas de React JS

d. Selección de metodología front-end

Una vez terminada la comparación entre las tecnologías para el manejo de frontend se elige Angular debido a las características que ofrece en el desarrollo y diseño como:

- Permite un desarrollo modular, es decir, cada componente es independiente de otro permitiendo la facilidad de desarrollo en el código además del uso de servicios mediante inyección de dependencias.
- Compatibilidad con CSS, esto permite que cada vista de la interfaz tenga su propio diseño ya que los archivos de diseño están encapsulados para cada componente.

3.1.4. Bootstrap y Material Design para desarrollo web responsive

Bootstrap nació en 2012 a partir de estilos gráficos de Twitter complementados con funcionalidades de jQuery para maquetación. Los elementos más destacables con los que trabaja son los grid, sus estilos de texto y sliders. Debido a que este depende en gran medida de JavaScript cuenta con amplio soporte y documentación para la facilidad de uso[32].

Por su parte Material Design fue desarrollado en 2014 por Google. Es un lenguaje de diseño mobile-first. Esta plataforma a diferencia de Bootstrap no depende de marcos y tampoco de bibliotecas JavaScript, este cuenta con su propio sistema de grid lo cual permite la realización de páginas complejas responsivas con estilo y funcionalidad[32].

Aunque ambos son utilizados para desarrollo web, material está orientado hacia la apariencia de una aplicación web mientras que Bootstrap se centra en crear de manera sencilla sitios web responsive y aplicaciones web que sean funcionales en contexto de experiencia de usuario. Algunas funcionalidades se comparan en la siguiente tabla[32]:

Funcionalidades	Bootstrap	Material Design
Color	Limitado	Extenso
Tipografía (Principal)	Arial	Roboto
Elementos HTML	Gran variedad	Limitados
Web Responsive Design	Si	Si
Preprocesadores CSS	LESS y SASS	SASS
Componentes	Dropdowns Grupos de botones	Etiquetas Botones

	Grupos de inputs Barras de navegación Mígas de pan Paginador Etiquetas Alertas y notificaciones Tarjetas	Paginadores Tarjetas Footer Formularios Barra de navegación Barras de progreso
Funcionalidades jQuery	Transiciones Modal Dropdown Menú con scroll Pestañas Tooltips Popovers Alertas Botones Acordeones Menú fijo	Acordeones Notificaciones Dropdown Modales Menú fijo Menú lateral móvil Pestañas Transiciones Animaciones

Tabla 22. Comparación Bootstrap y material Design

a. Selección de tecnología web responsive

Después de la comparación entre las tecnologías responsive se opta por utilizar Bootstrap ya que esta es muy utilizada en sitios web hechos con Javascript habiendo mucha documentación para proyectos diseñados con Bootstrap e implementados en proyectos de diseño con Angular.

3.1.5. Tecnologías de back-end para la aplicación web

Las tecnologías back-end son utilizadas para gestionar las diferentes peticiones de información que llegan y después que esta información es tratada es devuelta al dispositivo para ser visualizada mediante el front-end. Se realizó una comparación entre ASP.net y node JS con el fin de determinar cuál es la que se adapta de mejor manera al proyecto.

a. .NET CORE

Es un marco de aplicación web open source basado en C# que ofrece una amplia gama de formularios web, plantillas y más para la creación de aplicaciones web funcionales y dinámicas. Algunas características importantes son[33]:

- Personalización en el almacenamiento de datos

- Asistencia Microsoft
- Sencillez de mantenimiento
- Gran seguridad y protección[33].

b. Node JS

Node JS es un entorno de ejecución de JavaScript que permite a los desarrolladores ejecutar código JavaScript en el lado del servidor y permite el desarrollo de aplicaciones web en tiempo real. Es de arquitectura asíncrona y permite hacer llamadas a la API mediante eventos ya que usa el motor Chrome V8. Algunas características importantes son[34]:

- Rendimiento y escalabilidad:
- Facilidad de aprendizaje
- gran ecosistema de paquetes y comunidad
- Programación asíncrona y manejo de errores[34]

c. Comparación Tecnologías Backend

Características	NODE JS	.NET CORE
Lenguaje	Javascript	C#
Comunidad	Grande	Grande
Procesamiento de datos	Modelo de E/S asíncrono de un solo hilo	Modelo de E/S síncrono multiproceso
Plataformas	Linux, Microsoft, MacOS	Linux, Microsoft, MacOS
Herramientas disponibles	Media	Alta

Tabla 23. Comparación tecnologías backend

d. Selección de tecnología backend

Después de la comparación entre las tecnologías para el desarrollo del backend se opta por utilizar .Net Core para el desarrollo del backend de la aplicación, ya que esta permite el desarrollo de web apis que envían información al frontend mediante peticiones http haciendo uso de librerías para la seguridad, para la conexión a la base de datos y para lo que se requiera permitiendo al desarrollador acceder a una gran variedad de documentación y una comunidad activa.

3.1.6. Metodologías de desarrollo web

a. Determinación de metodología para la aplicación web

Para el desarrollo de esta propuesta es fundamental elegir una metodología de desarrollo, esta metodología se encargará de asegurar el desarrollo de un software de calidad, estas metodologías entre sus servicios abarcan buenas prácticas y procesos orientados a mantener la flexibilidad ante los cambios.

Las metodologías ágiles de desarrollo están especialmente indicadas en proyectos con requisitos poco definidos o cambiantes. Estas metodologías se aplican bien en equipos pequeños que resuelven problemas concretos, lo que no está reñido con su aplicación en el desarrollo de grandes sistemas, ya que una correcta modularización de los mismos es fundamental para su exitosa implantación. Dividir el trabajo en módulos abordables minimiza los fallos y el coste. Las metodologías ágiles presentan diversas ventajas, entre las que se destaca[35]:

- Capacidad de respuesta a cambios de requisitos a lo largo de desarrollo.
- Entrega continua y en plazos breves de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo.
- Importancia de la simplicidad, eliminando el trabajo innecesario.
- Atención continua a la excelencia técnica y al buen diseño.
- Mejora continua de los procesos y el equipo de desarrollo[35].

Sabiendo que las metodologías ágiles son esenciales en proyectos de desarrollo cabe hacerse la pregunta: ¿qué metodología de desarrollo de software se debe utilizar? Estas metodologías se deben adoptar en función del tamaño del proyecto, así como el tiempo de desarrollo y el costo, para agilizar la entrega de un proyecto a los involucrados[35].

b. XP

La metodología XP es usada por pequeños equipos que necesitan completar un proyecto en un periodo corto de tiempo, esta metodología se caracteriza por pequeñas entregas funcionales representadas por historias de usuario que pueden estar basadas en Test Driven Development o TDD, lo que garantiza que estos pequeños entregables sean parte de un todo funcional. El éxito depende de cuan bien este diseñado el proceso completo, la Planificación, Diseño, Codificación, Pruebas y Lanzamientos garantizan el éxito o fracaso a la hora de cumplir con un objetivo. Todas estas fases tienen un

esquema cíclico, es decir, se puede volver al punto de partida y reevaluar nuevos parámetros dentro de su implementación en el caso de ser necesario, para así poder tener una constante retroalimentación. Una de las desventajas de XP es que tiene un costo elevado a la hora de adaptar el proyecto en caso de una modificación del equipo de trabajo o requerimientos[36].

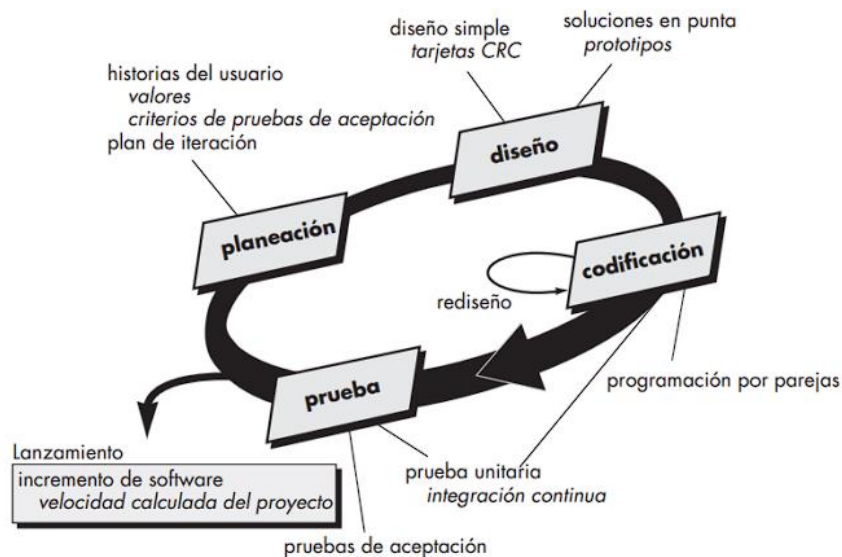


Figura 12: Ciclo de vida metodología XP

c. OOHDM

La metodología OOHDM cuyas siglas en español son Método de Diseño e Hipermmedia Orientado a Objetos, tiene similitud con la metodología HDM con la diferencia de que tiene un proceso que indica las actividades a ejecutar y el producto o entregable que debe hacerse al finalizar una fase. Este método toma como punto de partida el modelo de clases obtenido durante la primera fase del desarrollo de software denominado modelo conceptual, además permite modelar aplicaciones de grandes tamaños o con grandes volúmenes de información y pueden ser usados en diversos tipos de aplicaciones navegables, sitios web, sistemas de información o presentaciones multimedia[37].

OOHDM es una de las metodologías que más se utilizan hoy en día debido a que permiten reducir los tiempos de desarrollo, reutilizar diseño, simplificar la evolución y el mantenimiento de la aplicación. Las fases de esta metodología se pueden ver en la siguiente figura[37]:

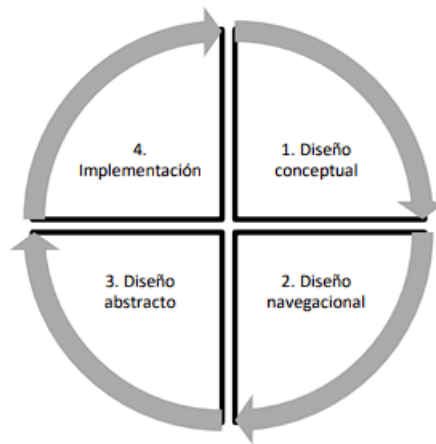


Figura 13: Etapas de desarrollo metodología OOHDM

OOHDM permite poseer una captura en forma visual de los requisitos de la aplicación para el desarrollo del software. Esta es una metodología que se basa en la recolección de requisitos funcionales[37].

Fases de la metodología OOHDM

Fase	Productos	Mecanismos	Preocupaciones de diseño
Recopilación de requisitos	Casos de uso, actores, escenarios[38]	Identificación de roles y tareas, especificación de escenarios, diagramas de interacción de usuarios[38]	Identificación del dominio de la aplicación[38]
Diseño conceptual	Clases, subsistemas, relaciones, atributos[38]	Clasificación, generalización y especialización [38]	Modelado de la semántica del dominio de la aplicación[38]
Diseño navegacional	Nodos, enlaces, estructuras de acceso, contextos de navegación[38]	Mapeo entre objetos conceptuales y de navegación[38]	Perfil y tarea del usuario. Énfasis en aspectos cognitivos[38]
Diseño de interfaz abstracta	Objetos abstractos de interfaz, respuesta a eventos externos[38]	Mapeo entre navegación y objetos perceptibles[38]	Modelado de objetos perceptibles. Describir

			la interfaz para objetos de navegación[38].
Implementación	Aplicación de ejecución[38]	Proporcionados por el entorno de destino[38]	Rendimiento, integridad[38]

Tabla 24. Resumen metodología OOHDM

d. ICONIX

Es una metodología de desarrollo de software que se halla entre las metodologías RUP (Rational Unified Processes) y XP, donde unifica un conjunto de métodos orientado a objetos con el objetivo de tener el control estricto sobre todo el ciclo de vida de un proyecto. Cuenta con una secuencia de pasos que se debe seguir para determinar claramente las actividades a desarrollar en cada etapa de un proyecto de desarrollo[39].

Características:

Iterativo e incremental: durante el desarrollo del modelo del dominio y la definición de los casos de uso se producen varias iteraciones. El ciclo de vida incremental consiste en desarrollar por partes el producto de manera que se pueda integrar funcionalmente, y en cada ciclo de iteración se revisa y mejora el producto[39].

Trazabilidad: cada paso realizado está definido por un requisito, se define la trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos de software producidos[39].

Dinámica del UML: ofrece un uso dinámico del UML porque se utiliza algunos diagramas UML[39].

Fases de ICONIX

Revision de requisitos / Análisis de requisitos: Identificar en el mundo real, los objetos y todas las relaciones de agregación y generalización entre ellos. Se deben analizar todos los requisitos que formaran parte de los sistemas y con estos construir el diagrama de clases, que representa las agrupaciones funcionales que estructuran el sistema en desarrollo[39].

Para esta fase se utilizan 3 herramientas:

- Modelo de dominio: identificar objetos y cosas del mundo real que intervienen con el sistema.
- Modelo de casos de uso: describe las acciones o el comportamiento que un usuario realiza dentro del sistema: actores, casos de uso y el sistema.
- Prototipo de interfaz de usuario: implica la creación de un modelo del sistema, en el que analistas y clientes deben estar de acuerdo[39].

Revisión del diseño preliminar / Análisis y diseño preliminar

En esta fase se obtendrán una ficha de caso de uso, está formada por un nombre descripción, una precondición que debe cumplirse antes de iniciarse, una postcondición que debe cumplir al terminar si termina correctamente.

Diagrama de Robustez: Un diagrama de robustez es un híbrido entre un Diagrama de Clases y un Diagrama de Actividades. Es una herramienta que permite capturar el Que hacer y a partir de eso él como hacerlo[39].

El diagrama de Robustez se divide en:

- Objetos fronterizos: usado por los actores para comunicarse con el sistema.
- Objetos entidad: son objetos del modelo del dominio.
- Objetos de Control: es la unión entre la interfaz y los objetos de entidad[39].

Diagrama de Clases: describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos[39].

Revisión crítica del diseño/Diseño

En esta fase se reconocen todos los elementos que forman parte del sistema.

- Diagramas de Secuencia: muestra los métodos que llevarán las clases del sistema. Muestra todos los cursos alternos que pueden tomar todos los casos de uso.
- Se debe terminar el modelo estático, añadiendo los detalles del diseño en el diagrama de clases[39].

Implementación y pruebas

En esta fase a partir del buen diseño logrado se creará el software; que posteriormente se entregará. Pero además se debe tener en cuenta factores como:

- Reusabilidad: es la posibilidad de hacer uso de los componentes en diferentes aplicaciones.
- Extensibilidad: consiste en modificar con facilidad el software.
- Confiabilidad: realización de sistemas descartando las posibilidades de error.
- Realizar pruebas: Test de unidades, de casos, datos y resultados[39].

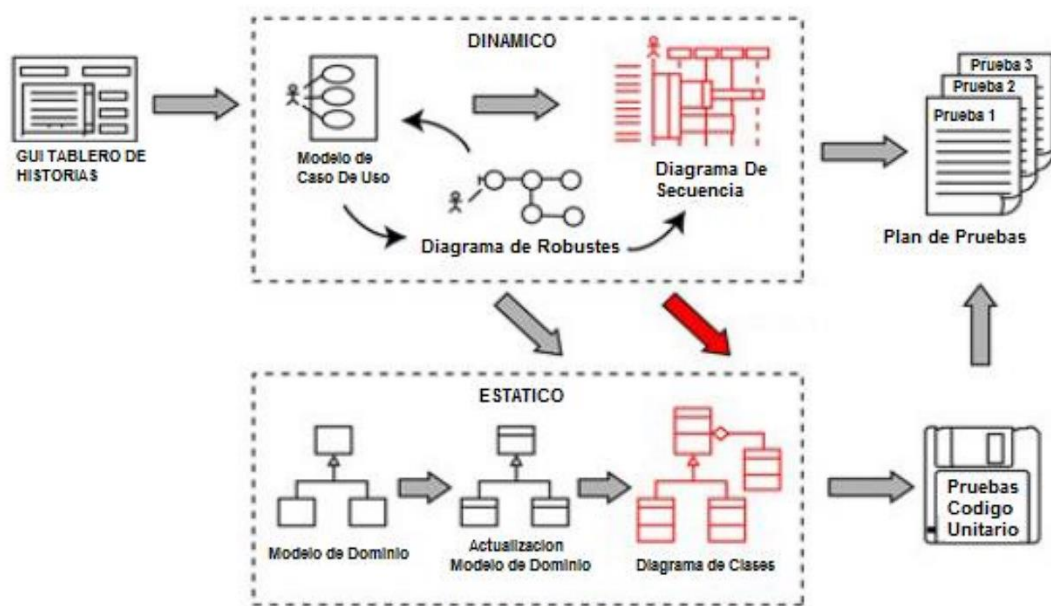


Figura 14: Resumen de procesos ICONIX

3.1.7. Comparación entre metodologías

A continuación, se detalla una tabla comparativa entre las metodologías para aplicaciones web seleccionadas para tal análisis: XP, OOHDM y ICONIX, para determinar la correcta metodología acorde al proyecto:

	XP	OOHDM	ICONIX
Tamaño del proyecto	Pequeño/mediano	Pequeño/mediano	Pequeño/mediano
Tamaño del equipo	Pequeño	Pequeño	Pequeño
Enfoque	Marco limpio, diseño sencillo, creando roles y tareas por separado para la iteración constante con la empresa, en cada fase se realiza análisis, diseño y pruebas	Diseño de vistas de datos abstractos para describir la interfaz de usuario de la aplicación Modelos orientados a objetos	Semi-ágil Métodos orientados a objetos presentando actividades en cada fase ofreciendo soporte UML dinámico
Entorno de desarrollo	Sencillo	Medio	Ligero
Periodicidad de eventos	Fases	Etapas: Diseño conceptual, diseño navegacional, diseño abstracto e implementación	Fases: Análisis de requisitos, revisión del diseño preliminar, revisión crítica de diseño, implementación
Riesgos	Bajo	Bajo	Bajo
Roles	Programador, cliente, encargado de pruebas, encargado de seguimientos, coach, mánager	No prescribe roles	Jefe de proyecto, diseñadores, programadores, encargados de pruebas
Duración del proyecto	Corto	Medio	Medio/corto
Requisitos de dirección	Gestionado por tarjetas de historia	Gestionado por etapas	Gestionado por etapas
Iteraciones	1 a 3 semanas	Flujo continuo	incremental

Orientación	Desarrollo de software, técnicas de programación	Desarrollo de aplicaciones web y software, sistemas de información	Desarrollo de software, sitios web de pequeña y mediana dificultad
Gestión de cambios	Se pasa a la siguiente historia del usuario	Se pasa a la siguiente etapa de desarrollo	Se pasa a la siguiente fase de desarrollo
Aumento de productividad	Alto	Alto	Alto

Tabla 25. Comparación metodologías de desarrollo

3.1.8. Selección de metodología

Según la tabla se eligió la metodología OOHDm debido a que presenta la mejor opción para el tipo de proyecto, definiéndolo como una metodología tradicional orientada a aplicaciones web, centrada en fases cada una con sus respectivos entregables, no prescribiendo roles, entregando así un proyecto con una metodología de calidad y acorde al desarrollo propuesto.

3.2. Desarrollo de la propuesta

De acuerdo con la información recolectada, la problemática y el análisis de herramientas de desarrollo y sus componentes servirán de apoyo para la aplicación de la solución.

3.2.1. Primera fase: recopilación de requisitos

Esta primera fase se divide en las siguientes etapas:

a. Análisis de requerimientos de la aplicación

Para el análisis de requerimientos se requirió hacer una recolección de información a los actores involucrados y a partir de ello se encontró características como la disponibilidad del servicio, la promoción y la comunicación entre implicados.

Requisitos funcionales

La implementación de la aplicación web responsive para la gestión de recorridos y la comunicación para transportistas particulares requiere que esté disponible siempre que el estudiante o el que gestiona el servicio lo requiera. A partir de los requisitos funcionales planteados con anterioridad se resumen los siguientes:

- Registrar y autenticar usuarios
- Visualizar información de usuarios
- Publicar recorridos
- Hacer reservaciones
- Ingresar disponibilidad del vehículo
- Gestionar recorridos
- Aceptar recorridos
- Visualizar comentarios
- Comunicación entre usuarios
- Búsqueda de recorridos
- Crear recorridos mediante mapas

Requisitos no funcionales

Dentro de los requerimientos no funcionales del desarrollo de la aplicación se considera:

- Diseño adaptable mediante el uso de responsive Web Design
- Interfaz amigable con el usuario
- Eficiencia de la aplicación web
- Seguridad e integridad de la aplicación
- La aplicación se debe acceder mediante un servidor en la web

b. Actores

La aplicación web será utilizada por dos actores principales: los transportistas que gestionan su servicio y los estudiantes que lo utilizan. Se detallan las actividades que realizan cada uno:

Actor	Actividad
Transportista	<ul style="list-style-type: none"> • Gestionar recorridos, ver comentarios, enviar mensajes, información de reservas de estudiantes registrados. • Ingresar nuevos recorridos • Ingresar busetas
Estudiante	<ul style="list-style-type: none"> • Visualizar opciones de recorridos registrados

	<ul style="list-style-type: none"> • Enviar mensajes • Visualizar horarios • Ingresar comentarios
--	--

Tabla 26. Actores

c. Relación de generalización entre actores

Los actores están clasificados de acuerdo con su rol en la aplicación

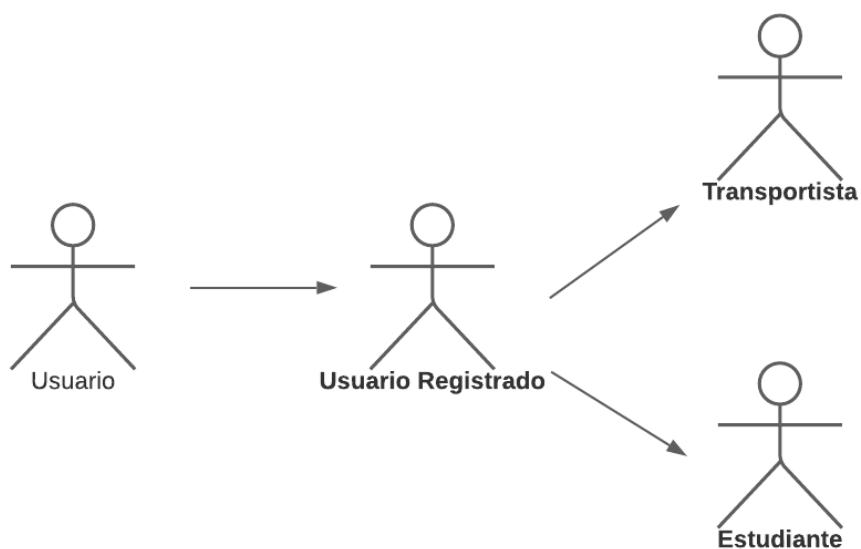


Figura 15: Relación de generalización entre actores

d. Diagrama de casos de uso

El diagrama muestra a los actores y sus funciones dentro de la aplicación web, cada uno con tareas específicas en la aplicación.

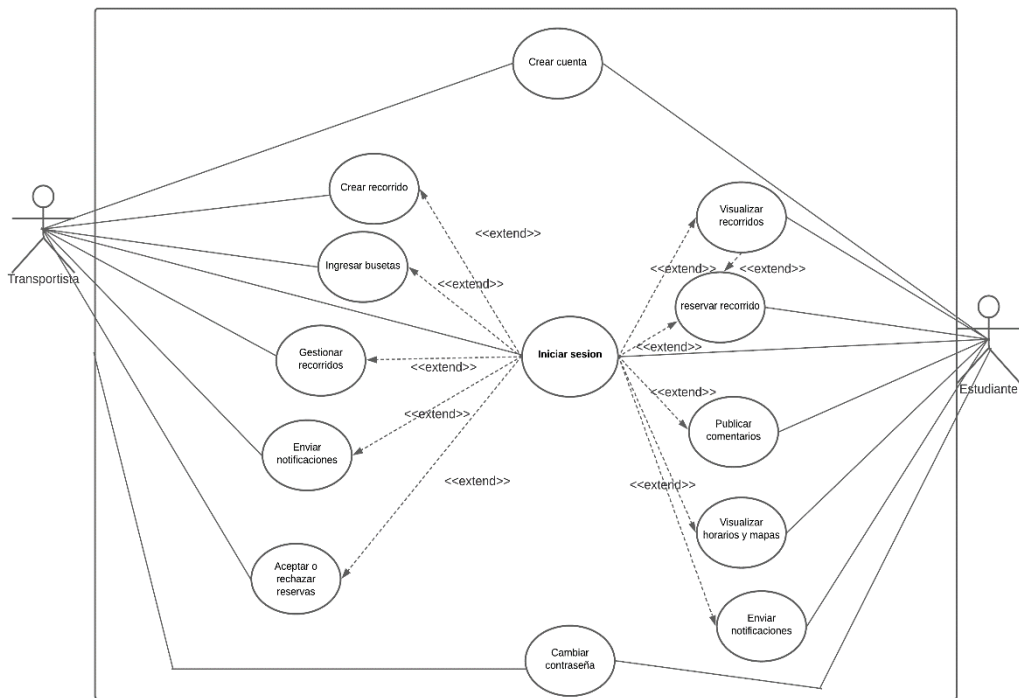


Figura 16: Casos de uso actores

e. Documentación de casos de uso

Caso 1: iniciar sesión

Caso de uso	Iniciar sesión				CU1
Actor(es)	Transportista, Estudiante				
Precondición	Los actores deben estar registrados				
Postcondición	Los actores tienen la sesión activa para realizar las actividades de la aplicación				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir el ingreso a la aplicación mediante el registro o login				
Resumen	Los actores proporcionan sus credenciales en el formulario, se validan los datos y se accede a la aplicación				
Curso/Flujo Normal (Básico)					
Paso	Actor	Sistema			
1	Ingresar email y contraseña				
2	Presionar el botón inicio	Valida los datos ingresados y redirecciona			
Curso/Flujo Alterno (Excepcional)					
Paso	Actor	Sistema			

2.1	Datos incorrectos	Muestra un mensaje de error si existen datos erróneos.
-----	-------------------	--

Tabla 27. Caso de uso 1

Caso 2: Crear cuenta

Caso de uso	Crear cuenta				CU2
Actor(es)	Transportista, Estudiante				
Precondición	Los actores no están registrados				
Postcondición	Los actores se registran en la base de datos y pueden iniciar sesión				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Crear una cuenta según el rol que se elija				
Resumen	Los actores llenan el formulario de registro y se validan para la creación de la cuenta				
Curso/Flujo Normal (Básico)					
Paso	Actor	Sistema			
1	Ingresar nombre, apellido, teléfono, dirección, email y contraseña	Valida los datos que se ingresan			
2	Presionar el botón registrar	Registra los datos ingresados y redirecciona			
Curso/Flujo Alterno (Excepcional)					
Paso	Actor	Sistema			
2.1	Datos incorrectos	Muestra un mensaje de error si existen datos erróneos o correo duplicado.			

Tabla 28. Caso de uso 2

Caso de uso 3: Actualizar contraseña

Caso de uso	Cambiar contraseña				CU3
Actor(es)	Transportista, Estudiante				
Precondición	Los actores deben tener una cuenta en la base de datos				
Postcondición	La contraseña registrada en la base de datos va a actualizarse				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir el cambio de contraseña en caso de olvido				

Resumen	Los actores proporcionan su correo electrónico y se envía un correo a su cuenta	
Curso/Flujo Normal (Básico)		
Paso	Actor	Sistema
1	Ingresar email	
2	Presionar el botón enviar	Valida el correo ingresado y envía la solicitud
Curso/Flujo Alterno (Excepcional)		
Paso	Actor	Sistema
2.1	Datos incorrectos	Muestra un mensaje de error si el correo no es válido.

Tabla 29. Caso de uso 3

Caso de uso 4

Caso de uso	Visualizar recorridos				CU4
Actor(es)	Estudiante				
Precondición	Iniciar sesión				
Postcondición	Los actores pueden ver los recorridos e información ingresada por los transportistas				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir a los actores enviar una notificación de reserva				
Resumen	El actor visualiza la información que necesita y selecciona una opción				
Curso/Flujo Normal (Básico)					
Paso	Actor	Sistema			
1	Accede a la opción recorridos	Muestra la lista de recorridos creados			
Curso/Flujo Alterno (Excepcional)					
Paso	Actor	Sistema			
2.1		No existen recorridos creados.			

Tabla 30. Caso de uso 4

Caso de uso 5

Caso de uso	Reservar recorridos				CU5
Actor(es)	Estudiante				
Precondición	Iniciar sesión				
Postcondición	El actor debe escoger una ruta				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir al actor enviar una petición de reserva				
Resumen	El actor visualiza los recorridos y escoge uno enviando una notificación de reserva				
Curso/Flujo Normal (Básico)					
Paso	Actor	Sistema			
1	Seleccionar la ruta	Guarda la información del estudiante que reserva			
2	Presionar el botón de acción	Crea la reserva al transportista que la creó			
Curso/Flujo Alterno (Excepcional)					
Paso	Actor	Sistema			
2.1	Error de solicitud	La reserva no se crea.			

Tabla 31. Caso de uso 5

Caso de uso 6

Caso de uso	Publicar comentarios				CU6
Actor(es)	Estudiante				
Precondición	Iniciar sesión				
Postcondición	Ingresar un comentario sobre las rutas				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir a los actores crear comentarios				
Resumen	Los actores crean comentarios basándose en las rutas que se muestran				
Curso/Flujo Normal (Básico)					
Paso	Actor	Sistema			
1	Crear comentario	Guarda el comentario y lo muestra			
Curso/Flujo Alterno (Excepcional)					
Paso	Actor	Sistema			

Tabla 32. Caso de uso 6

Caso de uso 7

Caso de uso	Enviar notificaciones				CU7
Actor(es)	Estudiante				
Precondición	Iniciar sesión				
Postcondición	Los actores envían una notificación				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir la comunicación con el transportista				
Resumen	El actor selecciona un transportista de la lista y se envía un mensaje directo predeterminado al número registrado en la base de datos				
Curso/Flujo Normal (Básico)					
Paso	Actor	Sistema			
1	Dar clic en la opción enviar mensaje	Carga el numero registrado del transportista y envía un mensaje predeterminado			
2	Presionar el botón enviar	Envía un mensaje utilizando WhatsApp			
Curso/Flujo Alterno (Excepcional)					
Paso	Actor	Sistema			
2.1		El sistema no envía el mensaje			

Tabla 33. Caso de uso 7

Caso de uso 8

Caso de uso	Visualizar horarios y mapas				CU8
Actor(es)	Estudiante				
Precondición	Iniciar sesión				
Postcondición	Al actor se le mostrará la información sobre rutas				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir acceder a la información sobre recorridos publicados, horario, precio, hora de salida				
Resumen	Los actores pueden visualizar información sobre las rutas				
Curso/Flujo Normal (Básico)					
Paso	Actor	Sistema			
1					
2	Ingresa a la opción reservas	Carga las rutas creadas			
Curso/Flujo Alterno (Excepcional)					

Paso	Actor	Sistema
2.1		No carga las rutas.

Tabla 34. Caso de uso 8

Caso de uso 9

Caso de uso	Crear recorridos					CU9
Actor(es)	Transportista					
Precondición	Iniciar sesión					
Postcondición	El actor crea una ruta para ser mostrada					
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0	
Propósito	Permitir acceder a la información sobre el recorrido					
Resumen	Los actores crean una ruta y el sistema la muestra					
Curso/Flujo Normal (Básico)						
Paso	Actor		Sistema			
1	Llena los campos de información del formulario y los puntos en el mapa		Valida los datos			
2	Presiona la opción crear recorrido		Guarda los recorridos en la base de datos			
Curso/Flujo Alterno (Excepcional)						
Paso	Actor		Sistema			
2.1			Los puntos no se crean.			

Tabla 35. Caso de uso 9

Caso de uso 10

Caso de uso	Ingresar busetas					CU10
Actor(es)	Transportista					
Precondición	Iniciar sesión					
Postcondición	El actor ingresa una buseta					
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0	
Propósito	Permitir al actor ingresar una buseta para crear un recorrido					
Resumen	Los actores crean una buseta					

Curso/Flujo Normal (Básico)		
Paso	Actor	Sistema
1	Llena los campos de información de la buseta	Valida los datos
2	Presiona la opción crear buseta	Guarda la buseta en la base
Curso/Flujo Alterno (Excepcional)		
Paso	Actor	Sistema
2.1		La buseta no se crea.

Tabla 36. Caso de uso 10

Caso de uso 11

Caso de uso	Gestionar recorrido				CU11
Actor(es)	Transportista				
Precondición	Iniciar sesión				
Postcondición	El actor puede eliminar un recorrido o crear otro				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir crear múltiples recorridos				
Resumen	Los actores crean un recorrido o eliminan uno ya existente				
Curso/Flujo Normal (Básico)					
Paso	Actor	Sistema			
1	Crea o eliminar un recorrido	Valida los datos			
2	Presiona la opción correspondiente	Guarda los recorridos en la base de datos			
Curso/Flujo Alterno (Excepcional)					
Paso	Actor	Sistema			

Tabla 37. Caso de uso 11

Caso de uso 12

Caso de uso	Enviar notificaciones				CU12
Actor(es)	Transportista				
Precondición	Iniciar sesión				
Postcondición	El actor envía una notificación al usuario que creó una reserva				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir comunicarse con el estudiante sobre la reserva				
Resumen	Los actores envían una notificación al estudiante mediante mensaje				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Visualiza la información del estudiante		Muestra los datos		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		

Tabla 38. Caso de uso 12

Caso de uso 13

Caso de uso	Aceptar o rechazar reservas				CU13
Actor(es)	Transportista				
Precondición	Iniciar sesión				
Postcondición	El actor acepta o rechaza una ruta				
Autor	Santiago Nuñez	Fecha	10/06/2023	Versión	1.0
Propósito	Permitir crear reservas				
Resumen	Los actores aceptan o rechazan una reserva				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Visualiza la reserva		Muestra la información		
2	Presiona la opción correspondiente		Envía un mensaje o elimina la reserva		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
2.1			La reserva no se carga.		

Tabla 39. Caso de uso 13

3.2.2. Segunda etapa – Diseño conceptual

En esta etapa se determina el modelo conceptual que está formado por las entidades de la aplicación junto con sus clases, relaciones, operaciones y los atributos que pertenecen a ellas.

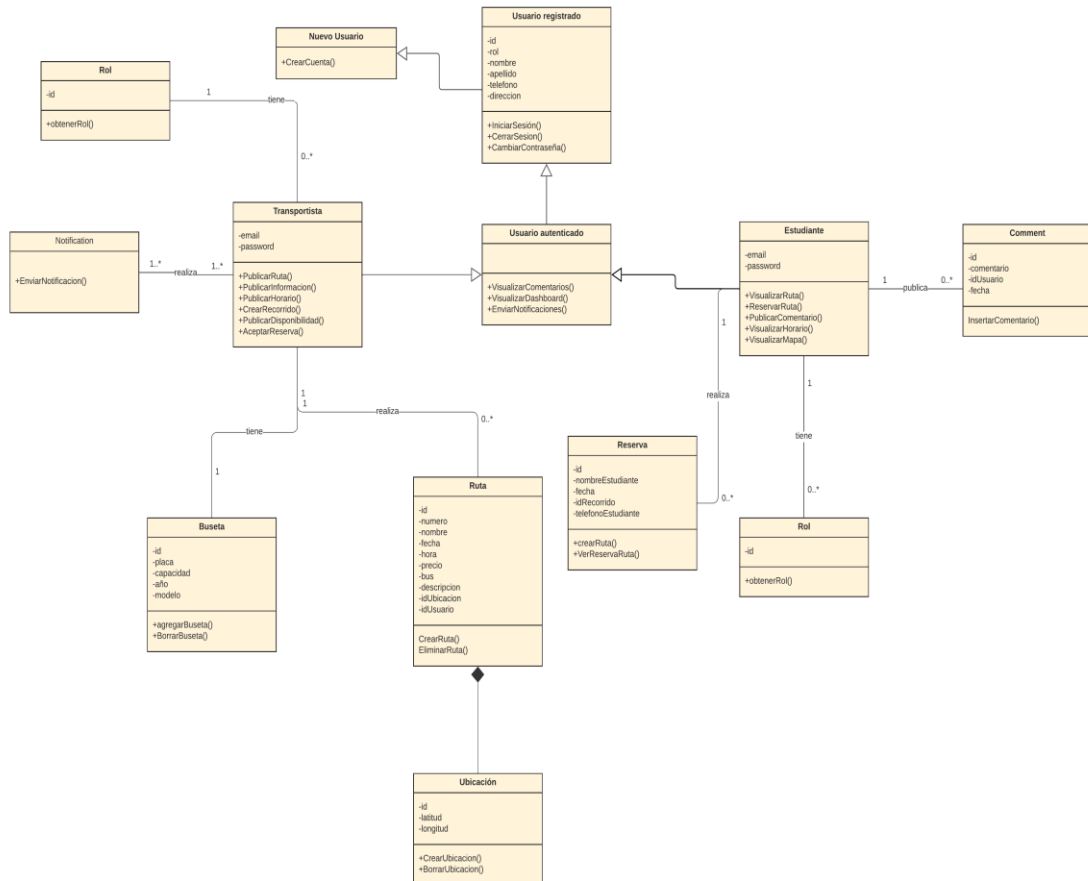


Figura 17: Diseño conceptual

Dentro de este esquema se pueden evidenciar tres clases principales que van a participar en la aplicación:

a. Diagrama de clase Usuario

Esta clase representa al usuario que comience a interactuar en la aplicación. Esta va a heredar sus atributos a la entidad usuario autenticado. La clase usuario autenticado es la que el usuario va a tomar una vez que se ingrese en la aplicación, esa clase es la que luego va a pasar sus atributos y métodos a las dos siguientes clases principales: estudiante y transportista:

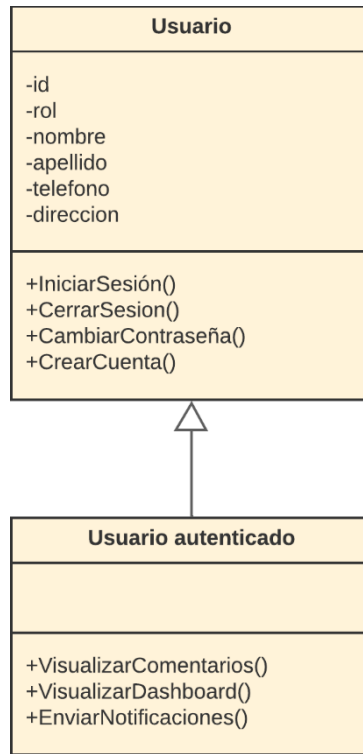


Figura 18: Diagrama de clase Usuario

b. Diagrama de clase Transportista

Este diagrama representa la relación entre el transportista y las entidades que van a interactuar con él, principalmente la entidad Recorrido y la entidad Busetas, donde el transportista realiza acciones como publicar ruta y agregar buseta además de tener el rol correspondiente al usar la aplicación

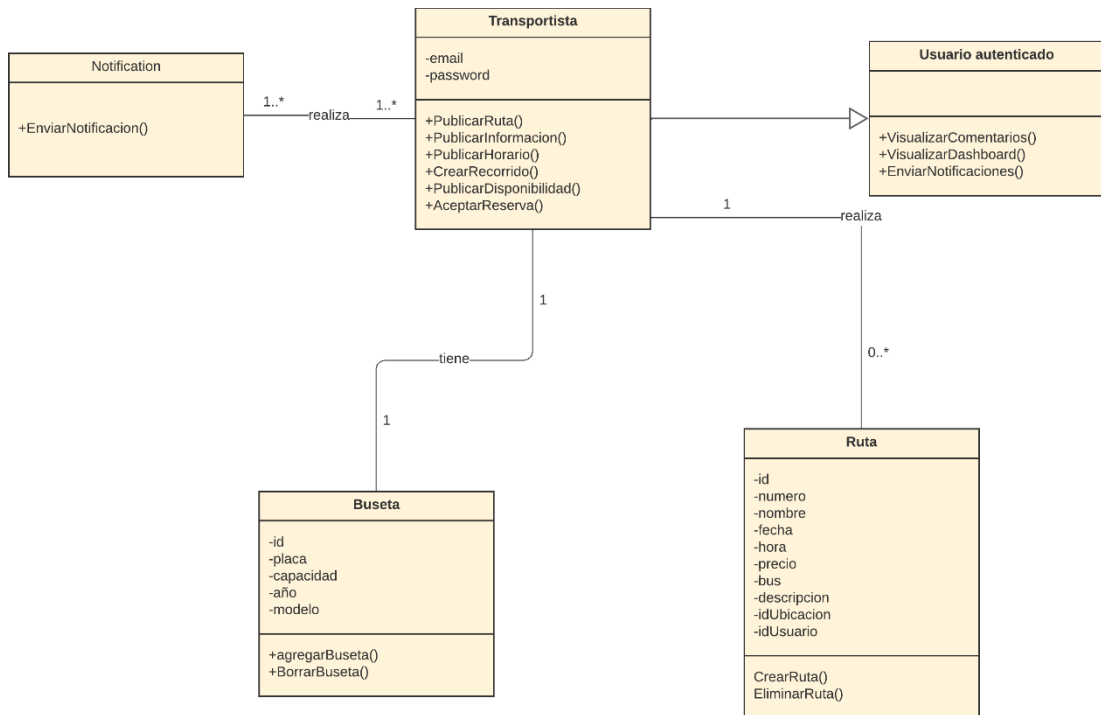


Figura 19: Diagrama de clase transportista

c. Diagrama de clase Estudiante

El siguiente diagrama muestra como la entidad Estudiante se relaciona con las entidades Comentario, Reserva y Rol. Estas clases permiten que la entidad estudiante tenga varias acciones a realizar como visualizar rutas y publicar comentarios, todo esto con su respectivo rol en la aplicación. la entidad Reserva permite que el estudiante tenga varias acciones como crear y ver reservas. En el contexto de la relación la entidad estudiante también permite realizar las acciones de un usuario autenticado

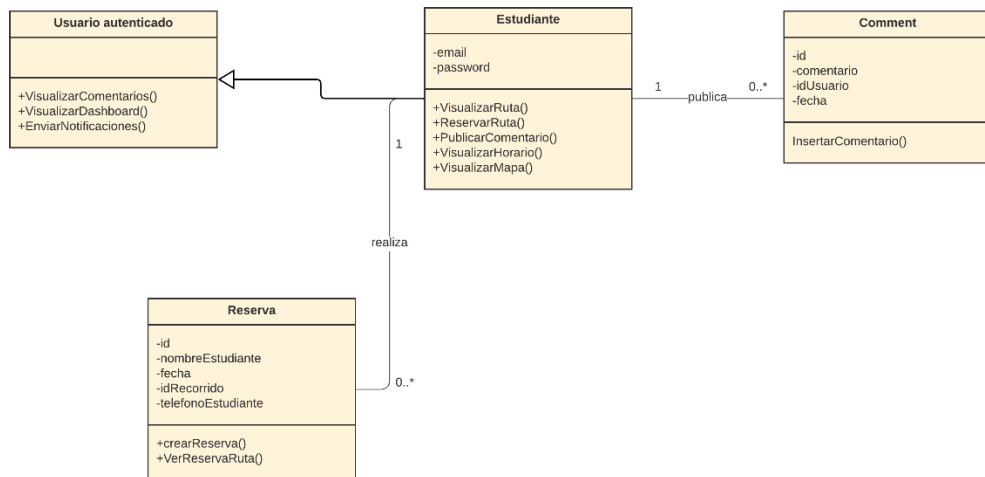


Figura 20: Diagrama de clase estudiante

3.2.3. Tercera etapa – Diseño navegacional

Esta tercera etapa supone un primer acercamiento de la aplicación con la interacción que se tendrá con los usuarios. Aquí se sintetiza las tres clases del modelo conceptual en contextos para la navegación en la aplicación.

- Usuario
- Estudiante
- Transportista

Una vez que se tienen las clases definidas se procede a diseñar los diagramas de clases navegacionales.

a. Diagrama navegacional de un usuario

En el diagrama de un usuario que accede al dominio de la aplicación se muestran los nodos a los que tiene permitido en la navegación dentro de la plataforma. Estos nodos están representados por las opciones que tiene el usuario al momento de acceder y definidos anteriormente como registrarse e iniciar sesión

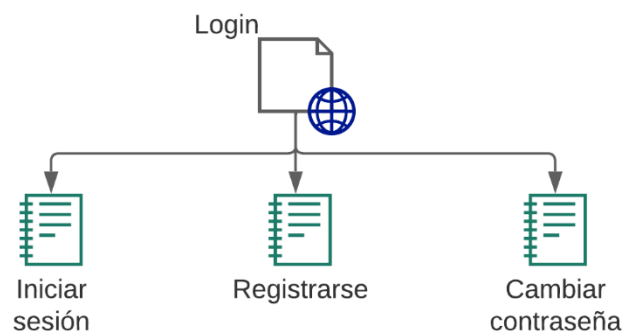


Figura 21: Diagrama navegacional: Usuario

b. Diagrama navegacional de un estudiante

El siguiente esquema muestra los nodos identificados para la navegación de un estudiante que accede a la aplicación y su flujo de navegación. Estos nodos están definidos como transportistas, home o página principal en donde se muestra el nombre del usuario y el nodo reservas

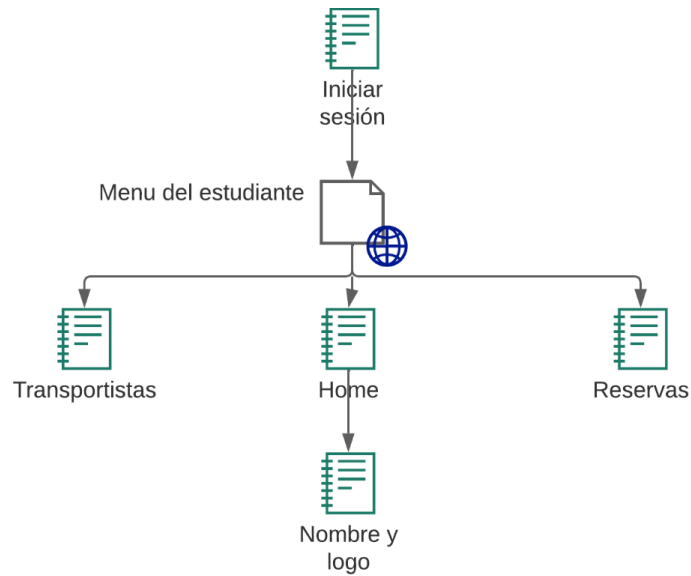


Figura 22: Diagrama navegacional estudiante

c. Diagrama navegacional del transportista.

En este esquema se identifican los nodos a los que se puede acceder a la aplicación como transportista y sus enlaces de navegación. Estos nodos están representados como buseta, home, recorridos y reservas.

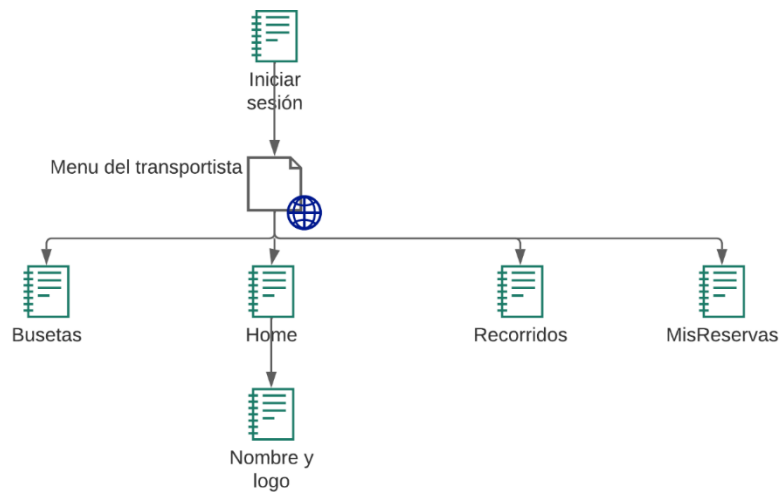


Figura 23: Diagrama navegacional transportista

Una vez establecido los diagramas de navegación se procede a crear los contextos navegacionales de la aplicación.

d. Contexto navegacional de un usuario

El contexto navegacional muestra el flujo de navegación de un usuario que visita la aplicación. en el diagrama se muestra las opciones que el usuario tendrá disponible en ese momento en un nodo en específico. Por lo tanto, el usuario empieza en el modo login y a su vez este enlaza a tres nodos diferentes:

Inicio de sesión: este nodo contiene el formulario de ingreso de datos para acceder a la aplicación, aquí se ingres usuario y contraseña

Registrarse: este nodo contiene el formulario de registro para la creación de un usuario nuevo, aquí se selecciona el rol y se ingresa los datos

Cambiar contraseña: este nodo enlaza al formulario de correo para cambiar la contraseña

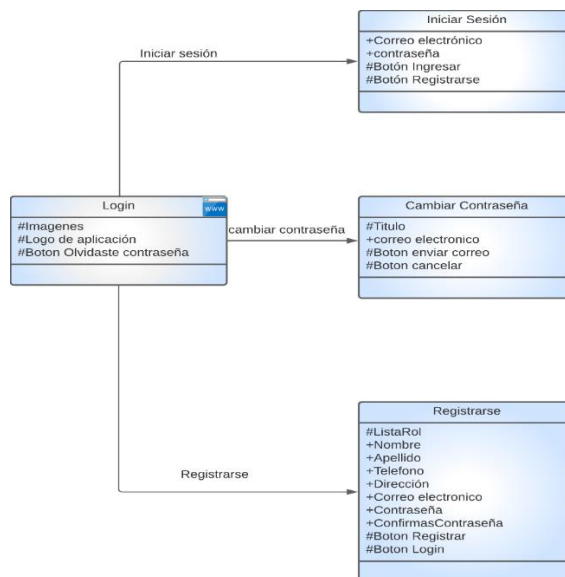


Figura 24: Contexto navegacional usuario

e. Contexto navegacional de un estudiante

El contexto de estudiante muestra el flujo de navegación de un usuario registrado como estudiante, al tener la sesión activa se crea la opción de acceder hacia nuevos nodos en el sistema. Estos nodos enlazan hacia nuevas opciones como:

Menú Estudiante: este nodo se enlaza con el nodo iniciar sesión, que a su vez enlaza con tres nodos diferentes

Reservas: este nodo se enlaza del menú estudiantes, esta función solo aparece si se accede primero al menú estudiantes.

Home: este nodo se muestra después de iniciar sesión

Nombre y logo: al momento de acceder al nodo home se muestra este nodo

Transportistas: este nodo dispone de información referente a los transportistas registrados. Este nodo solo los ve el estudiante

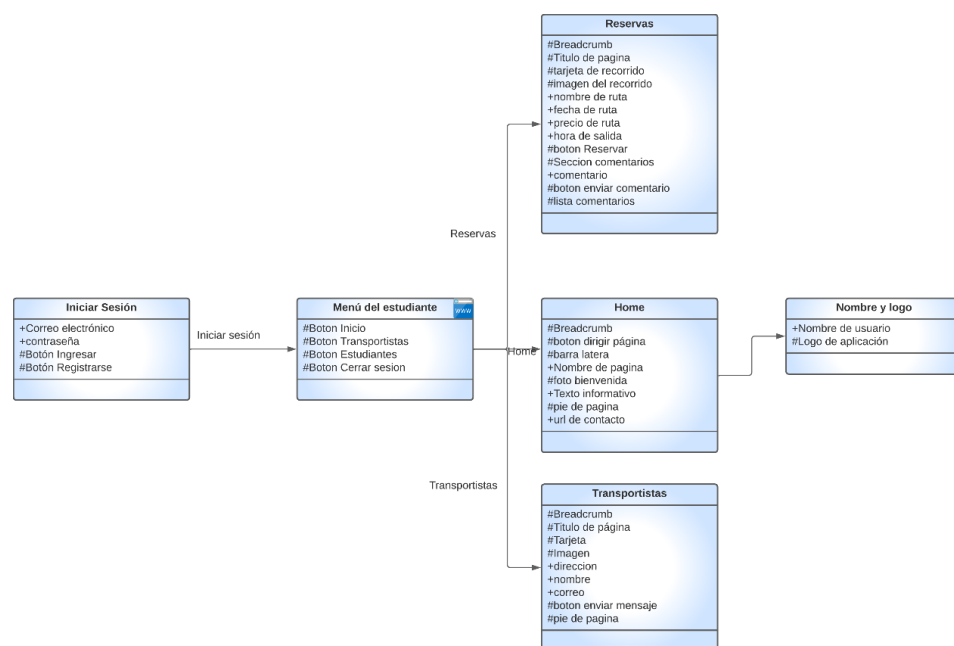


Figura 25: Diseño navegacional estudiante

f. Contexto navegacional de un transportista

El siguiente contexto muestra el flujo de navegación de un usuario registrado como transportista. Aquí el usuario accede a nodos propios que corresponden a la gestión de sus recorridos. El menú de transportista deriva del nodo iniciar sesión y este a su vez permite el acceso a nodos como:

Busetas: enlaza al nodo que contiene un formulario de registro de información

Home: enlaza al nodo donde se aloja la información que los usuarios observan al iniciar sesión.

Nombre y logo: deriva del nodo home. Por defecto todos los usuarios que acceden a los dos tipos de menús acceden a este nodo

Recorridos: este nodo enlaza al formulario de registro para crear un recorrido ζ , con el fin de que los estudiantes puedan acceder a esta información

Reservas: este nodo enlaza a la información que se obtiene después que un estudiante haya pedido una reserva la cual un transportista haya creado previamente en la aplicación.

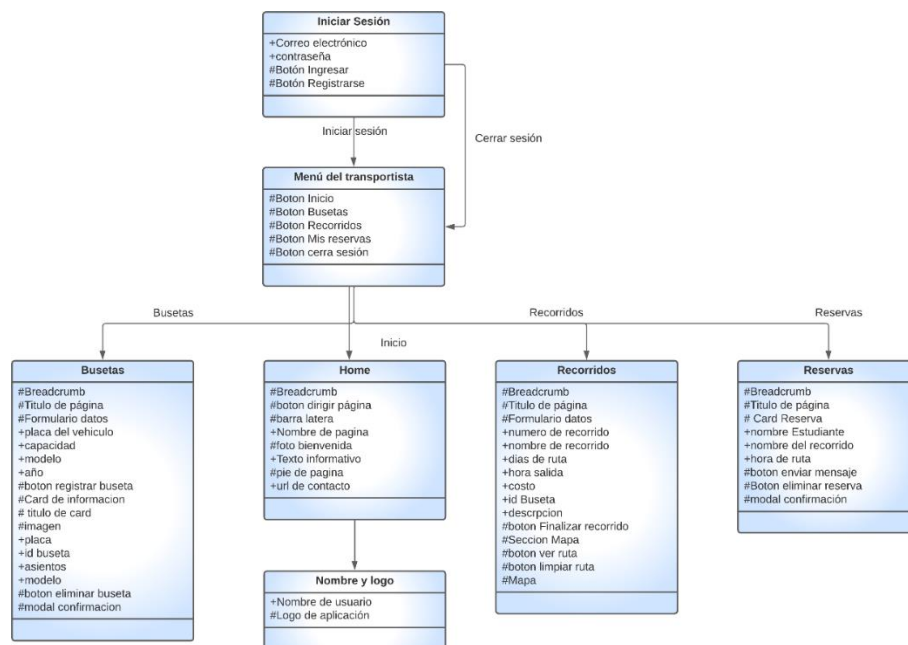


Figura 26: Contexto navegacional Transportista

g. Diagrama de navegación final

Una vez que se definió todo el contexto navegacional para todos los tipos de usuario en la aplicación se diseñó el diagrama de navegación final para poder tener una vista general de flujo total de navegación de la aplicación. Como se observa en la figura el diagrama parte desde el login y a partir de ahí se crea tres ramas dependiendo de que opción se elija. A diferencia de los diagramas anteriores aquí se obtiene una vista

general de los enlaces y nodos para poder establecer la delimitación de todo el espacio de navegación.

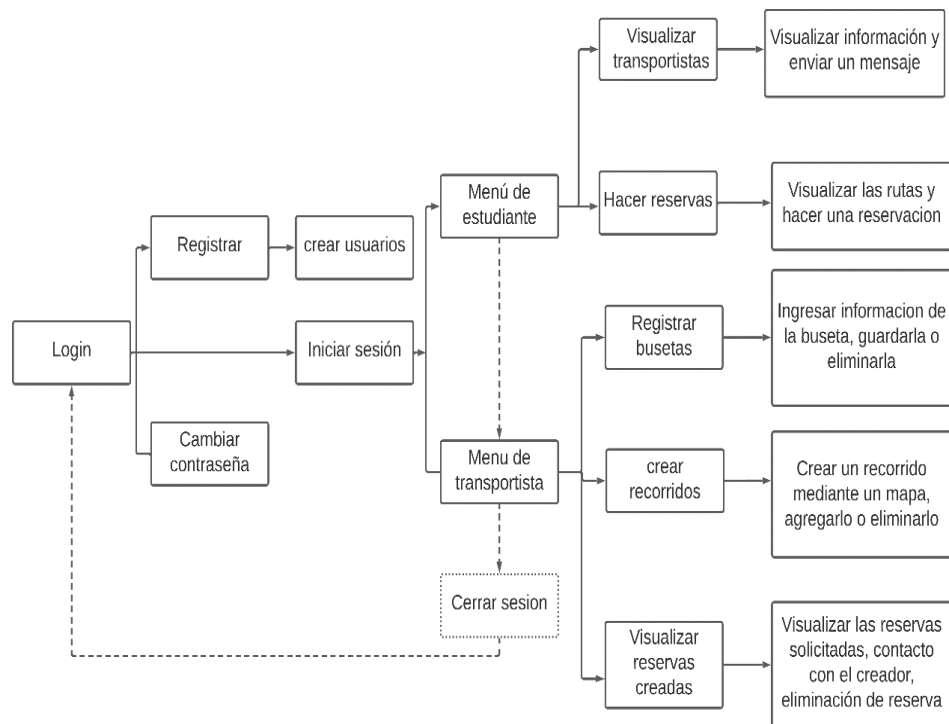


Figura 27: Diagrama navegación final

3.2.4. Cuarta etapa – Diseño de interfaces abstractas

El diseño de las ADV o vistas abstractas de datos por sus siglas en inglés permite elaborar bocetos o mockups acerca de cómo se verá la interfaz gráfica que percibirá el usuario. Como propone la metodología OOHDHM se aplicó este modelo porque se enfoca en la representación del estado y la interfaz, en la que cada vista se representa los atributos que van a tener dichas interfaces para poder establecer la estructura general de la aplicación.

La vista “login” representa a la página que ve el usuario al cargar la aplicación. Esta vista contiene una imagen relacionada a la aplicación junto con un pequeño formulario de ingreso de información.

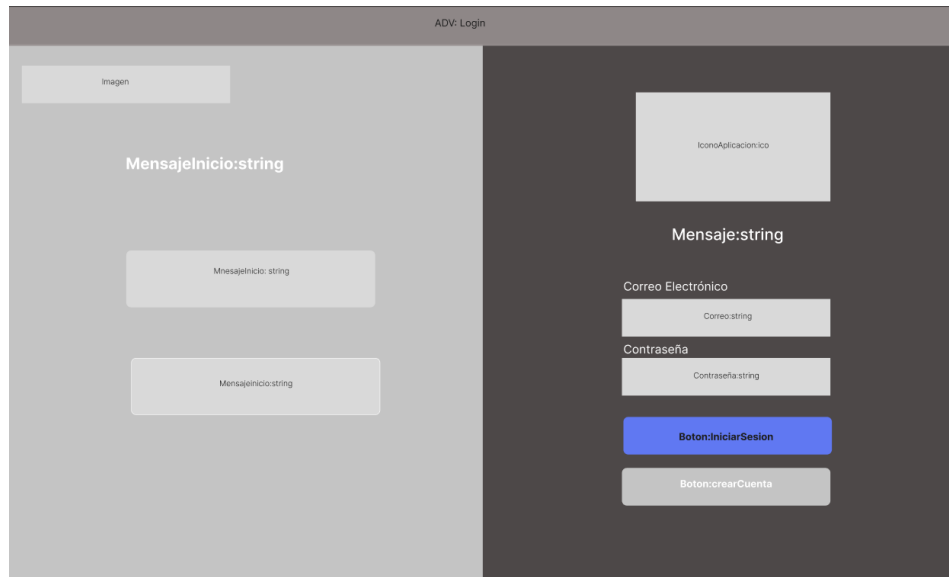


Figura 28: ADV Login

La vista “registro” representa la página donde se encuentra el formulario de ingreso de datos para el registro de usuarios

Figura 29: ADV: Formulario Registro

La vista “cambiar contraseña” representa al modal que contiene el campo de ingreso de correo electrónico para el envío del correo



Figura 30:ADV: Cambiar contraseña

La vista “MenuEstudiante” y menuTransportista representa a la página principal que visualiza el usuario que entra a la aplicación según el rol

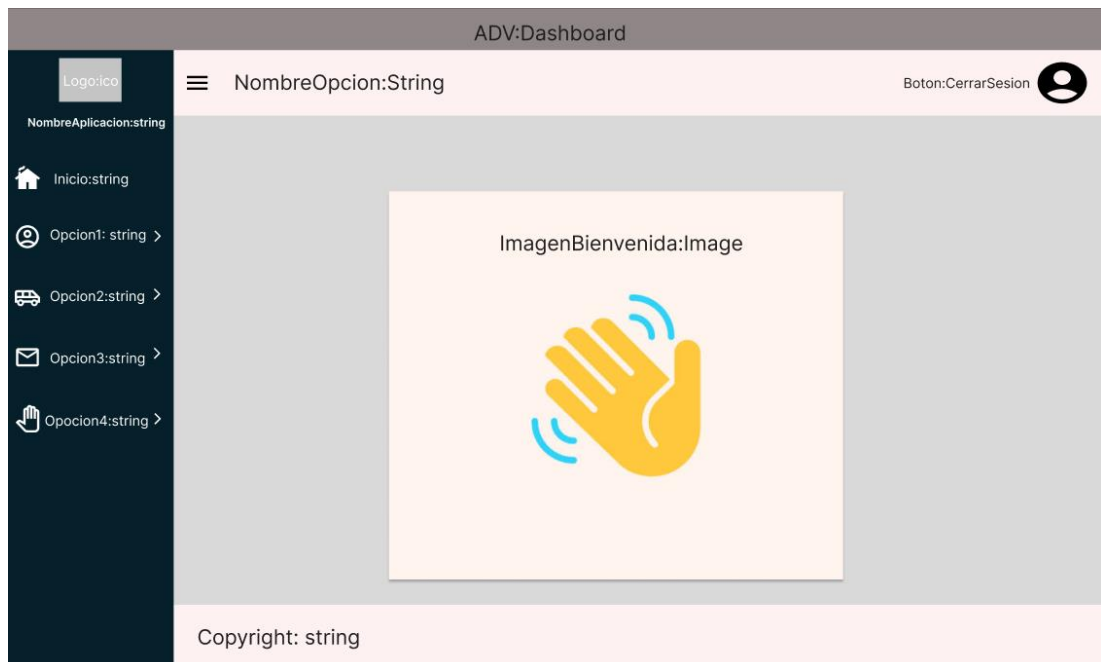


Figura 31: ADV: dashboard

La vista “menú lateral” representa al elemento que se posiciona en la parte izquierda de la página principal y contiene las referencias a las diferentes páginas a las que puede acceder el usuario por rol

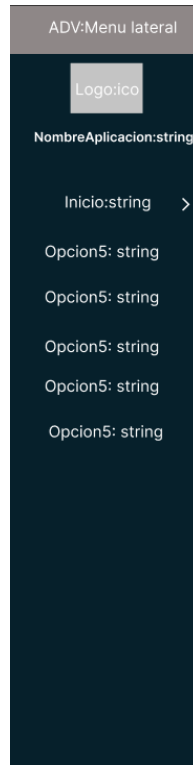


Figura 32: ADV: menú lateral

La vista “header” representa el encabezado de la aplicación donde se muestra el nombre de la página seleccionada, un icono de barra lateral y un botón cerrar sesión

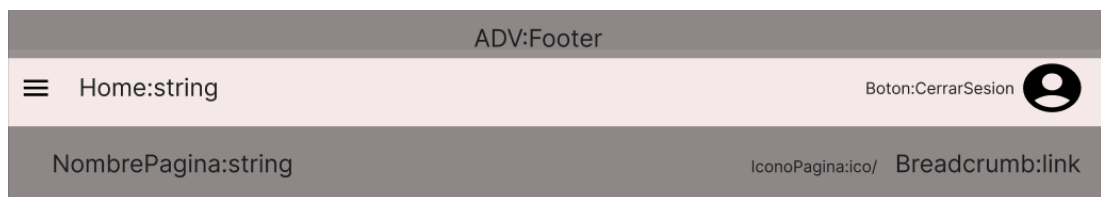


Figura 33: ADV header

La vista “footer” representa el pie de página que contiene la información sobre el estudiante y las referencias hacia sus redes sociales. Esta aparece en todas las páginas.



Figura 34: ADV: footer

La vista “CardTransportista” representa a la tarjeta que carga la información de los transportistas registrados cada vez que se accede a esa sección.

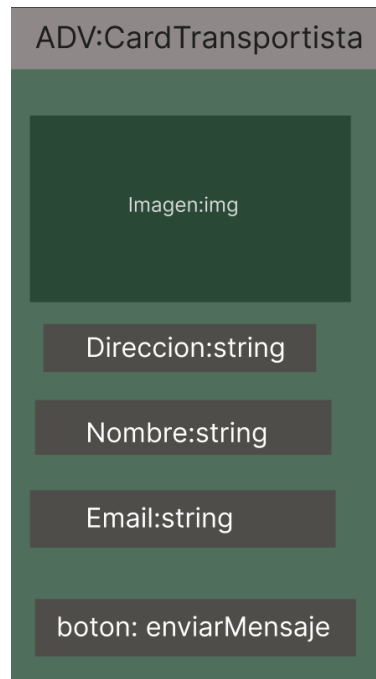


Figura 35: ADV:CardTransportista

La vista “Transportistas” muestra la vista CardTransportista y se posiciona tantas veces como transportistas haya registrados para el acceso de información

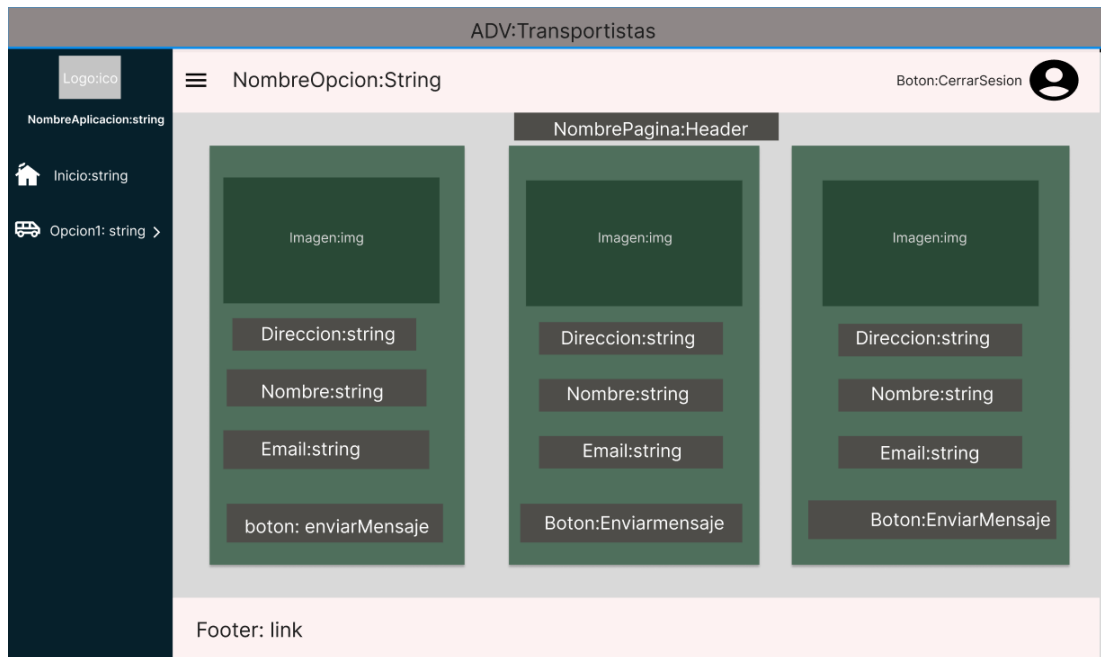


Figura 36: ADV: transportistas

La vista “CardRecorrido” muestra al elemento que se posiciona en la página Recorridos. Esta contiene la información sobre las rutas creadas y un botón para acceder a la página reservas.



Figura 37: ADV: CardRecorrido

La vista “Reservas” representa a la página donde se van a cargar las tarjetas de recorrido con la información. Esta a su vez contiene una sección de comentarios donde se muestra al elemento para añadir comentario y una lista.

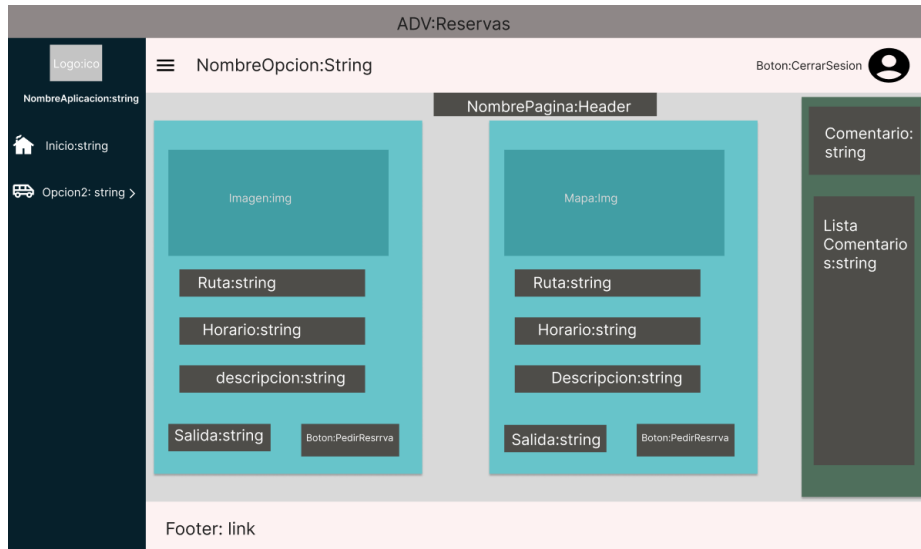


Figura 38: ADV: Reservas

La vista “Busetas” representa a la página donde se carga el formulario para el ingreso de información de busetas. Esta a su vez muestra una lista de tarjetas que contienen información sobre busetas creadas

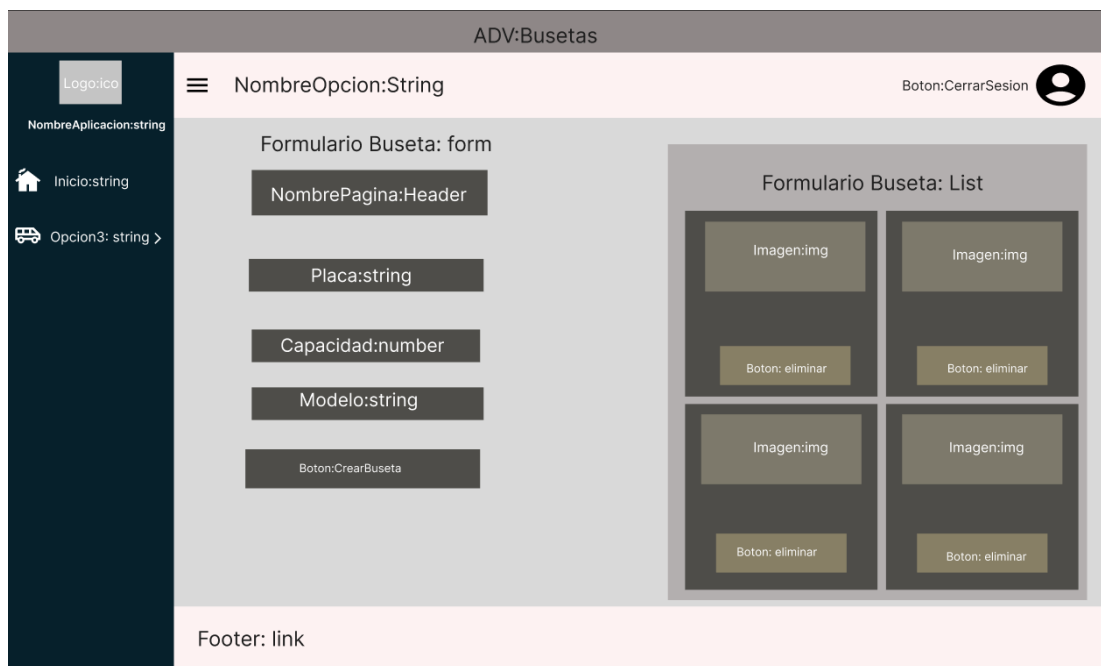


Figura 39: ADV:Busetas

La vista “Recorridos” representa a la página donde se muestra el formulario para crear un recorrido y que se puedan crear los puntos de la ruta utilizando un mapa. También contiene dos botones para procesar la información en el mapa

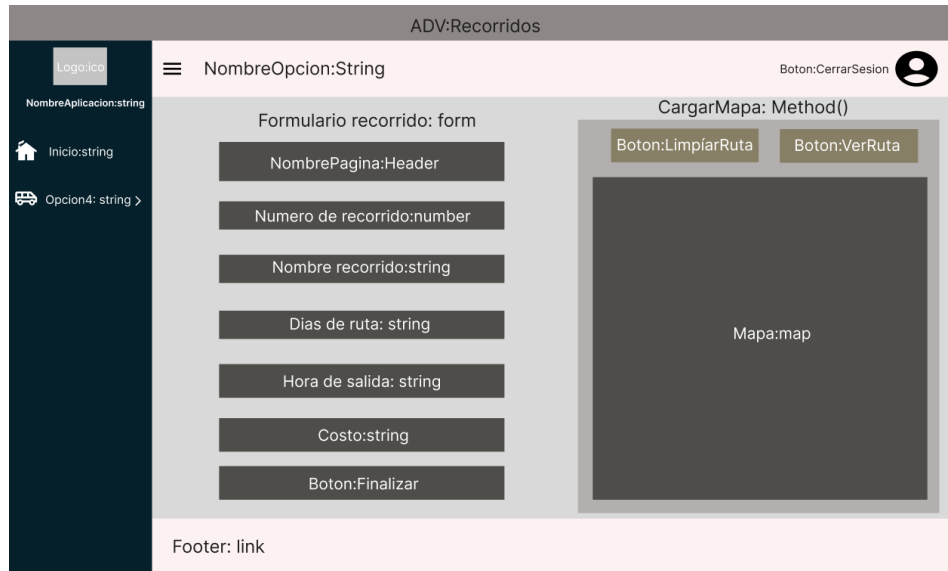


Figura 40: ADV: Recorridos

La vista “Reservas” representa a la página que contiene reservas hechas por estudiantes de la aplicación

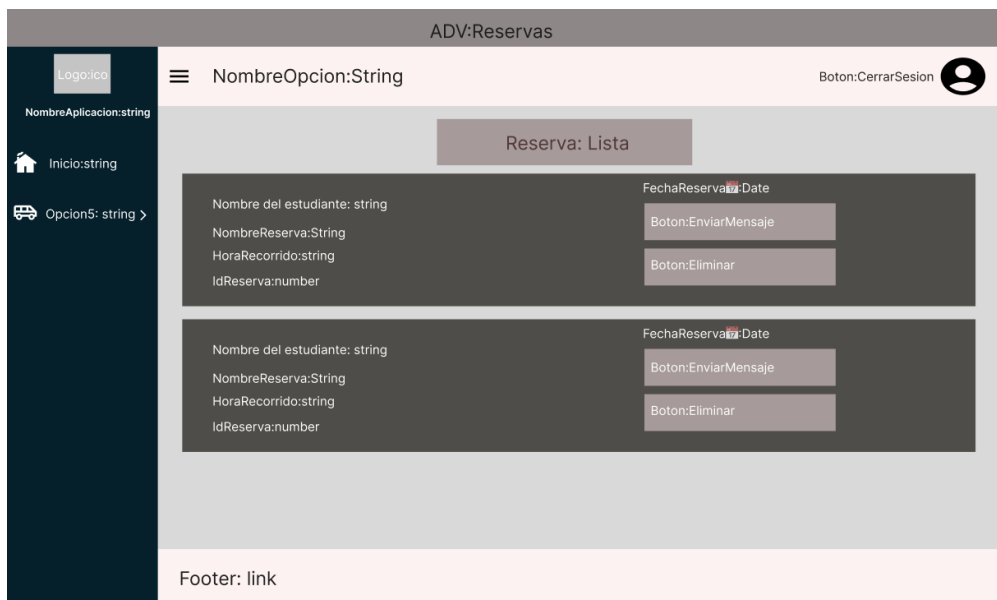


Figura 41: ADV:Reservas

3.2.5. Quinta etapa – Implementación

a. Arquitectura de la aplicación

Una vez que se culminó con éxito las anteriores etapas se procedió a llevar los objetos creados a un lenguaje de programación para obtener el código ejecutable de la aplicación. para la parte del back-end se decidió crear una web API utilizando .net Core y para la parte de front-end se desarrolló en el marco de trabajo Angular utilizando lenguaje Typescript.

Para establecer el funcionamiento del sistema la arquitectura en la que se basó este proyecto es la de cliente-servidor, donde la parte frontend hace peticiones http al backend devolviendo una API REST que recibe las peticiones y las devuelve en formato JSON.

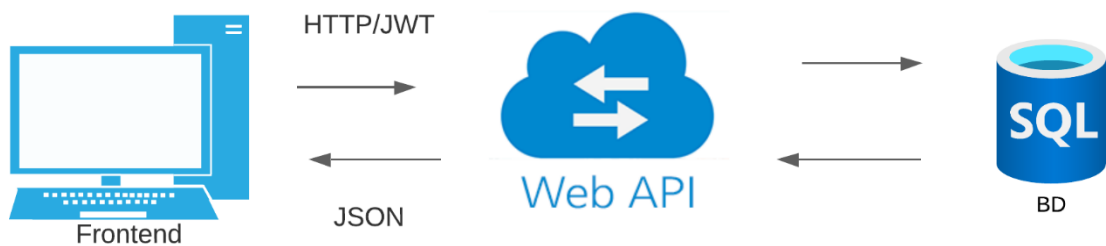


Figura 42: Arquitectura de la aplicación

b. Modelado de la base de datos

El modelado de la base de datos se elaboró a partir del modelo conceptual. Este modelo contiene las tablas que se crean en la base de datos y sus respectivas relaciones. La Base de datos se creó utilizando MySQL Workbench.

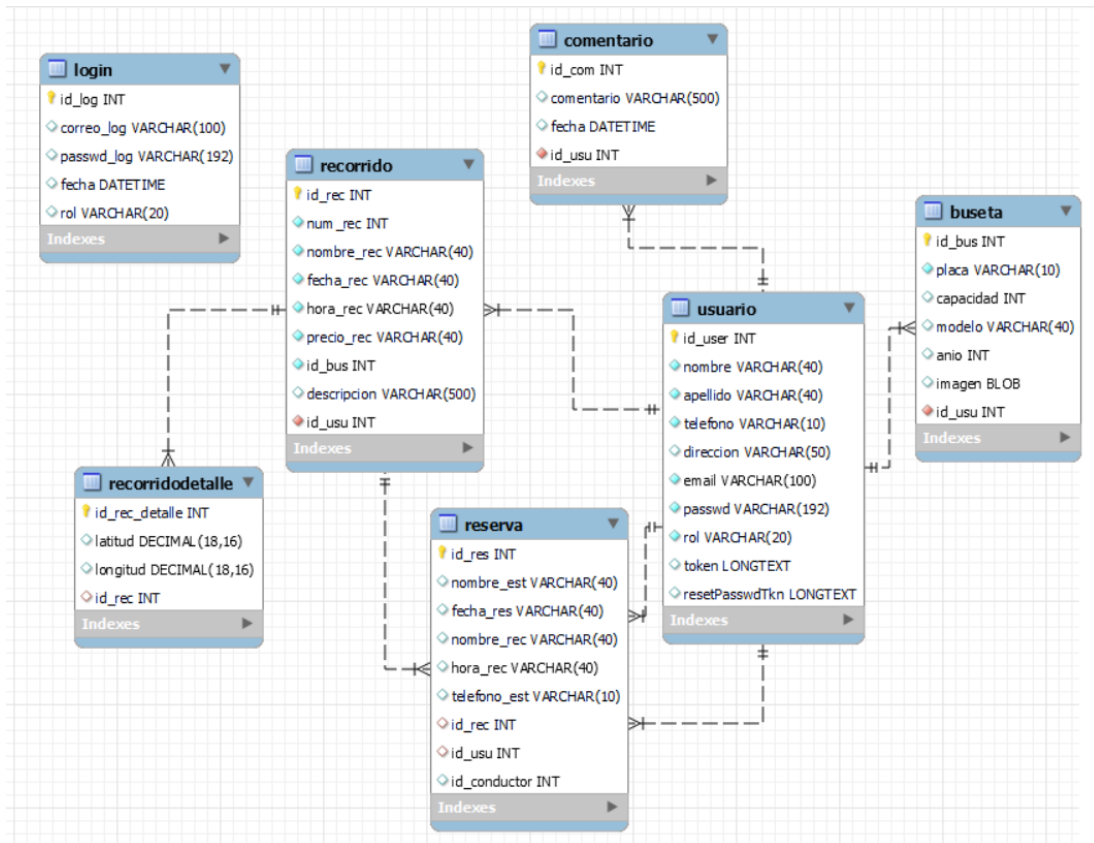


Figura 43: Base de datos de la aplicación de la aplicación de recorridos y comunicación

c. Conexión a la base de datos

Para conectarse a la base de datos en MySQL se generó una cadena de conexión en el archivo program.cs. haciendo referencia al archivo appsettings. Para poder hacer esta cadena de conexión se debe instalar la librería Pomelo si la versión de .net utilizada es la 6.0 en adelante.

```
//cadena de conexion para mysql (para corregir los errores se agregó builder)
string mySqlConnectionStr = builder.Configuration.GetConnectionString("DefaultConnection");
builder.Services.AddDbContext<AplicacionRecorridos2024Context>(options => options.UseMySQL(
    mySqlConnectionStr, ServerVersion.AutoDetect(mySqlConnectionStr)));
```

Figura 44: Conexión Base de datos

En el archivo appsettings se configuró la cadena de conexión hacia la base de datos

```
}
"AllowedHosts": "*",
//esta es la cadena de conexion de mysql, despues de crear esto ir a program.cs
"ConnectionStrings": {
  "DefaultConnection": "Server=localhost;Database=AplicacionRecorridos2024;Uid=root;Pwd=root"
}
}
```

Figura 45: Cadena conexión app.settings

d. Uso de JWT para la API-REST en sesión de usuario

Para establecer un nivel de seguridad en la aplicación desde la parte del servidor se hizo uso del estándar JWT (JSON Web Tokens), este recurso crea tokens de acceso para autenticar usuarios que realicen peticiones desde la parte del frontend. Para hacer uso de esta librería se debe instalar en el proyecto el paquete JwtBearer.

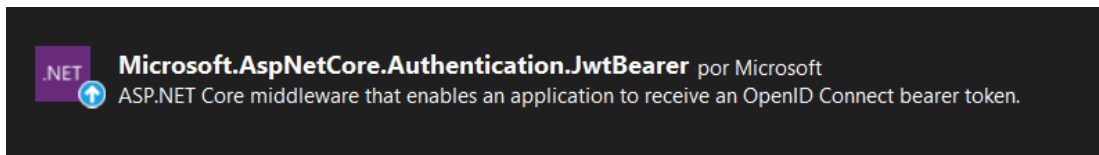


Figura 46: Librería JWT

Este token generado por el servidor debe estar firmado por una llave secreta que el programador establezca. En el archivo program.cs se establece que el proyecto va a usar este servicio al momento de autenticar usuarios.

```
AddJwtBearer(x =>
{
    x.RequireHttpsMetadata = false;
    x.SaveToken = true;
    x.TokenValidationParameters = new TokenValidationParameters
    {
        ValidateIssuerSigningKey = true,
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("Clave...")),
        ValidateAudience = false,
        ValidateIssuer = false,
        ClockSkew= TimeSpan.Zero
    };
});
```

Figura 47: Añadir Jwt Bearer

Para que el proceso de autenticación empiece se debe validar la información que el servidor envía o recibe, esto se hace mediante el uso del token para cada aspecto de la autenticación, ya sea en el registro, el inicio o fin de una sesión.

En el proceso de registro de usuarios se guarda la información del usuario y el sistema valida los datos ingresados, comprueba que el correo ingresado no exista en la base de datos, la fuerza de la contraseña y la encripta, por último, crea el token para el nuevo usuario y guarda los cambios en la base. Este método utiliza la petición Post

```
public async Task<IActionResult> RegisterUser([FromBody] Usuario userObj)
{
    if (userObj == null)
    {
        return BadRequest();
    }
    else
    {
        if (await CheckEmailExistAsync(userObj.Email))
        {
            return BadRequest(new { Message = "el email ya está registrado!" });
        }
        var pass = CheckPasswordStrength(userObj.Passwd);

        if (!string.IsNullOrEmpty(pass))
        {
            return BadRequest(new { Message = pass.ToString() });
        }
        userObj.Passwd = PasswordHasher.HashPassword(userObj.Passwd);

        userObj.Token = CreateJwt(userObj);
        await _context.Usuarios.AddAsync(userObj);
        await _context.SaveChangesAsync();
        return Ok(new { Message = "Usuario Registrado correctamente!" });
    }
}
```

Figura 48: Método registrar usuario

En el inicio de sesión se valida que el objeto usuario exista en la base de datos, se verifica si su correo existe y se verifica la contraseña. Si todo se cumple envía la respuesta y el token del usuario.

```

0 referencias
public async Task<IActionResult> Authenticate([FromBody] Usuario userObj)
{
    if (userObj == null)
    {
        return BadRequest(new { Message = "Usuario no encontrado!" });
    }

    var user = await _context.Usuarios.FirstOrDefaultAsync(x => x.Email == userObj.Email);
    if (user == null)
    {
        return NotFound(new { Message = "Usuario no encontrado!" });
    }

    if (!PasswordHasher.VerifyPassword(userObj.Passwd, user.Passwd))
    {
        return BadRequest(new { Message = "Contraseña incorrecta" });
    }

    //
    user.Token = CreateJwt(user);

    return Ok(new
    {
        Token = user.Token,
        Message = "Acceso exitoso!"
    });
}

```

Figura 49: Método autenticar usuario

Tanto al crear el inicio de sesión como al registrar un usuario se utiliza un método para crear un nuevo token. El token está conformado por tres partes: el encabezado, la llave y el contenido. En la llave está firmada la firma del servidor y en el contenido se envió el id del usuario para poder usarlo en el front-end, nombre, apellido y el rol. Estos datos sirvieron para identificar al usuario que está en el lado del cliente.

Por último, se definió la duración de validez del token a un día, aunque si se requiere que las sesiones de usuario activas tengan más seguridad se debe establecer menor duración o incluso crear un método para refrescar el token.

```

private string CreateJwt(Usuario user)
{
    var jwtTokenHandler = new JwtSecurityTokenHandler();
    var key = Encoding.ASCII.GetBytes("AplicacionTesisFinal....");
    var identity = new ClaimsIdentity(new Claim[]
    {
        new Claim(ClaimTypes.NameIdentifier, user.IdUser.ToString()), //id del usuario
        new Claim(ClaimTypes.Role, user.Rol), //rol
        new Claim(ClaimTypes.Name, $"{user.Nombre} {user.Apellido}") //nombre y apellido
    });

    var credentials = new SigningCredentials(new SymmetricSecurityKey(key), SecurityAlgorithms.HmacSha256);

    var tokenDescriptor = new SecurityTokenDescriptor
    {
        Subject = identity,
        Expires = DateTime.Now.AddDays(1),
        SigningCredentials = credentials
    };

    var token = jwtTokenHandler.CreateToken(tokenDescriptor);
    return jwtTokenHandler.WriteToken(token);
}

```

Figura 50: Método creación Token

El método ResetPassword permite a los usuarios que tenga un correo registrado en la base de datos cambiar la contraseña registrada anteriormente, para ello verifica que el correo exista en la base y si el usuario existe crea un nuevo token y lo guarda en la base de datos, pero si no existe envía un mensaje de error. A su vez encripta la contraseña y modifica el estado del usuario.

```

public async Task<IActionResult> ResetPassword(ResetPasswordDto resetPasswordDto)
{
    var newToken = resetPasswordDto.EmailToken.Replace(" ", "+");
    var user = await _context.Usuarios.AsNoTracking().FirstOrDefaultAsync(a => a.Email == resetPasswordDto.Email);
    if (user is null)
    {
        return NotFound(new
        {
            StatusCode = 404,
            Message = "el usuario no existe"
        });
    }

    var tokenCode = user.ResetPasswdTkn;
    if (tokenCode != resetPasswordDto.EmailToken)
    {
        return BadRequest(new
        {
            StatusCode = 400,
            Message = "link de correo inválido"
        });
    }

    user.Password = PasswordHasher.HashPassword(resetPasswordDto.NewPassword);
    _context.Entry(user).State = EntityState.Modified;
    await _context.SaveChangesAsync();
    return Ok(new
    {
        StatusCode = 200,
        Message = "Contraseña cambiada correctamente"
    });
}

```

Figura 51: Método cambiar contraseña

Para encriptar la contraseña del usuario en el servidor se hizo uso del método HashPassword que encripta la contraseña y guarda los bytes en un arreglo creando saltos que el programador defina, así mismo para desencriptarla se convierte el array de bytes a un string de base 64 como se muestra en la figura:

```
2 referencias
public static string HashPassword(string password)
{
    byte[] salt;
    rng.GetBytes(salt = new byte[SaltSize]);
    var key = new Rfc2898DeriveBytes(password, salt, Iterations);
    var hash = key.GetBytes(HashSize);

    var hashBytes = new byte[SaltSize + HashSize];
    Array.Copy(salt, 0, hashBytes, 0, SaltSize);
    Array.Copy(hash, 0, hashBytes, SaltSize, HashSize);
    var base64Hash = Convert.ToBase64String(hashBytes);
    return base64Hash;
}
```

Figura 52: Método encriptar contraseña

```
1 referencia
public static bool VerifyPassword(string password, string base64Hash)
{
    var hashBytes = Convert.FromBase64String(base64Hash);
    var salt = new byte[SaltSize];
    Array.Copy(hashBytes, 0, salt, 0, SaltSize);

    var key = new Rfc2898DeriveBytes(password, salt, Iterations);
    byte[] hash = key.GetBytes(HashSize);

    for (var i = 0; i < HashSize; i++)
    {
        if (hashBytes[i + SaltSize] != hash[i])
        {
            return false;
        }
    }
    return true;
}
```

Figura 53: Método verificar contraseña

El método SendEmail permite a un usuario que quiera cambiar su contraseña enviar un email a su cuenta de correo utilizando un servidor de correo smtp. En este método se genera un nuevo token de cambio de contraseña y se agrega al cuerpo del email que recibirá el método sendEmail creado en la clase emailService.

```
public async Task<IActionResult> SendEmail(string email)
{
    var user = await _context.Usuarios.FirstOrDefaultAsync(a => a.Email == email);
    if (user is null)
    {
        return NotFound(new
        {
            StatusCode = 404,
            Message = "el correo no existe"
        });
    }
    var tokenBytes = RandomNumberGenerator.GetBytes(64);
    var emailToken = Convert.ToBase64String(tokenBytes);
    user.ResetPasswdTkn = emailToken;
    //user.ResetPasswordExpiry = DateTime.Now.AddMinutes(15);
    string from = _config["EmailSettings:From"];
    var emailModel = new Email(email, "Cambia tu contraseña!", EmailBody.EmailStringBody(email, emailToken));
    _emailService.SendEmail(emailModel);
    _context.Entry(user).State = EntityState.Modified;
    await _context.SaveChangesAsync();
    return Ok(new
    {
        StatusCode = 200,
        Message = "Correo enviado!"
    });
}
```

Figura 54: Método enviar email

En la clase emailService que implementó la interfaz IEmailService se creó el cuerpo del correo que recibirá el usuario al momento de hacer la petición. Con el método cliente.Connect se conectó al servidor de correo y se utiliza el puerto 465 siendo un puerto seguro para https.


```

2 referencias
public void SendEmail(Email emailModel)
{
    var emailMessage = new MimeMessage();
    var from = _config["EmailSettings:From"];
    emailMessage.From.Add
        (new MailboxAddress("servirUTA", from));
    emailMessage.To.Add(new MailboxAddress(emailModel.To, emailModel.To));
    emailMessage.Subject = emailModel.Subject;
    emailMessage.Body = new TextPart(MimeKit.Text.TextFormat.Html)
    {
        Text = string.Format(emailModel.Content)
    };

    using (var client = new SmtplibClient())
    {
        try
        {
            client.Connect(_config["EmailSettings:SmtplibServer"], 465, true);
            client.Authenticate(_config["EmailSettings:From"], _config["EmailSettings:Password"]);
            client.Send(emailMessage);
        }
        catch (Exception ex)
        {
            throw;
        }
        finally { client.Disconnect(true); client.Dispose(); }
    }
}

```

Figura 55: Método enviar email interfaz

En la cadena de conexión hacia el servidor de correo utilizado se configuró el email desde donde se enviará la petición. Nombre del servidor y puerto. El nombre de usuario y la contraseña como se observa en la figura:

```

//cadena de conexión para smtp
"EmailSettings": {
    "From": "nunezsantiagole@gmail.com",
    "SmtplibServer": "smtp.gmail.com",
    "Port": 465,
    "Username": "nunezsantiagole@gmail.com",
    "Password": ██████████
}

```

Figura 56: Cadena conexión servidor SMTP

e. Documentación de la API-REST implementada

La API-REST creada para el proyecto provee un conjunto de datos que se acceden mediante solicitudes http por medio de rutas o controladores los cuales establecen un

canal de comunicación entre el extremo de la API y el cliente en el lado del front-end a través de una URL para resolver la petición del recurso solicitado.

La API creada cuenta con diferentes puntos finales para acceder a sus recursos como se detalla la tabla a continuación:

Método	Ruta	Función	Descripción
BusetaController			
POST	/api/Buseta/regar	GuardarBuseta	Guarda la buseta de un transportista
GET	/api/Buseta/{id}	listarBusetaUsuario	Lista las busetas que ha registrado un usuario por medio de su id
DELETE	/api/Buseta/id	eliminarBuseta	Elimina una buseta recibiendo el id de la buseta en cuestión
ComentarioController			
GET	/api/Comentario	ObtenerListaComentarios	Obtiene todos los comentarios registrados
POST	/api/Comentario/ registrarcomentario	GuardarComentario	Guarda los comentarios de los usuarios
RecorridoController			
GET	/api/Recorrido	ObtenerListaRecorridos	Obtiene todos los recorridos registrados
GET	/api/Recorrido/{id}	ObtenerListaRecorridosI d	Obtiene los recorridos de un usuario en específico
POST	/api/Recorrido/ guardarrecorrido	GuardarRecorrido	Guarda los recorridos creados por los usuarios
RecorridoDetalleController			
POST	/api/Recorridodetalle/ guardardetalle	GuardarDetalle	Guarda las ubicaciones o coordenadas de los recorridos
ReservaController			
POST	/api/Reserva/reserva	GuardarReserva	Guarda las reservas de los usuarios
GET	/api/Reserva/{id}	listarReserva	Lista las reservas hechas por un usuario por medio del id

DELETE	/api/Reserva/{id}	eliminarReserva	Eliminar la reserva de un usuario en específico
TransportistaController			
GET	/api/Transportista	ObtenerTransportistas	Obtiene todos los transportistas registrados
UsuarioController			
POST	/api/Usuario/authenticate	Authenticate	Permite el acceso de usuarios a la aplicación
POST	/api/Usuario/register	RegisterUser	Registra un usuario nuevo en la base de datos
GET	/api/Usuario	GetAllUsers	Lista todos los usuarios de la aplicación
GET	/api/Usuario/{id}	ListarUsuario	Lista un usuario en específico por su id
POST	/api/Usuario/send-reset-email/{email}	SendEmail	envía el email de cambio de contraseña al usuario
POST	/api/Usuario/reset-apssword	ResetPassword	envía la contraseña nueva a la base de datos

Tabla 40. Documentación de API

Para que la API-REST implemente la seguridad que se define en el token se debió establecer en los métodos que hacen las peticiones la opción Authorize para que validen la información y que solo se pueda acceder a ella si se envía el token en el header de las peticiones en el lado del cliente

```

[Authorize]
[HttpGet]
0 referencias
public async Task<Acti
{
    ...
    return Ok(await _c
}

```

Figura 57: Autorización de token en los métodos del backend

En el archivo program.cs se debió definir que el servidor use la autenticación del token cada que se haga una petición

```
app.UseCors("MyPolicy");  
app.UseAuthentication();  
app.UseAuthorization();
```

Figura 58: UseAuthentication

Se definió CORS (Cross-Origin Resource Sharing) para que el servidor envíe datos de una única fuente

```
builder.Services.AddCors(option =>  
{  
    option.AddPolicy("MyPolicy", builder => {  
        builder.AllowAnyOrigin()  
        .AllowAnyMethod()  
        .AllowAnyHeader();  
    });  
});
```

Figura 59: Configurar CORS

El backend de este proyecto se basó en el patrón de diseño MVC (modelo vista-controlador) e integra una capa extra llamada Service donde se encuentra toda la lógica al momento de traer los datos desde la base. Los modelos se definieron de acuerdo con las tablas existentes y los controladores se crearon de acuerdo con los modelos. Para que la capa vista sea delegada al frontend en Angular.

El modelo Usuario se definió como se muestra en la figura, donde se declara el nombre del modelo, junto con sus atributos y los métodos get y set para obtener y colocar información

```
public partial class Usuario
{
    4 referencias
    public int IdUser { get; set; }
    2 referencias
    public string Nombre { get; set; }
    2 referencias
    public string Apellido { get; set; }
    1 referencia
    public string Telefono { get; set; }
    1 referencia
    public string Direccion { get; set; }
    7 referencias
    public string Email { get; set; }
    7 referencias
    public string Passwd { get; set; }
    3 referencias
    public string Rol { get; set; }
    4 referencias
    public string Token { get; set; }
    3 referencias
    public string ResetPasswdTkn { get; set; }
}
```

Figura 60: modelo Usuario

Finalmente, todos los modelos utilizados en la aplicación se estructuraron como se indica en la figura:

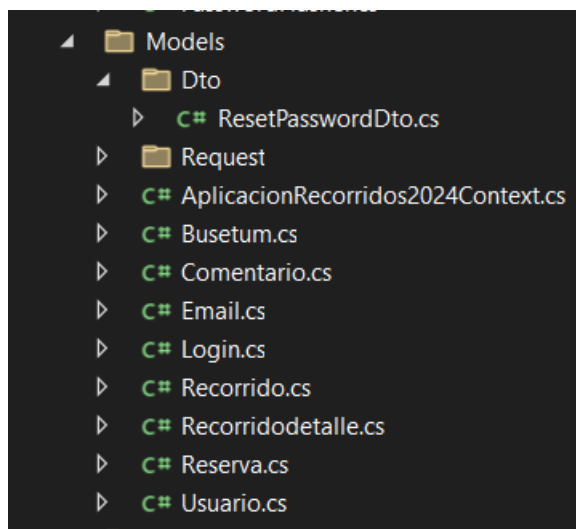


Figura 61: Modelos backend

Se establecieron los controladores que utilizarán estos modelos para crear los métodos http.

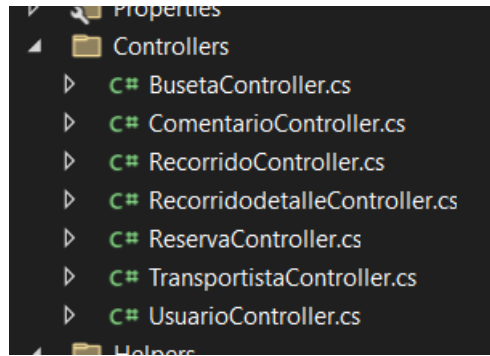


Figura 62: Controladores backend

Después se establecieron los servicios que ocuparán los controladores, estos tienen toda la lógica al momento de hacer una consulta a la base de datos

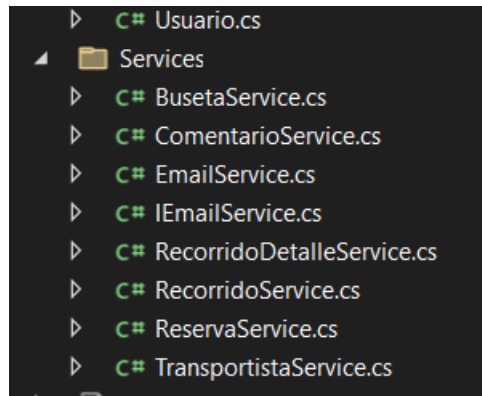


Figura 63: Servicios backend

También se definieron clases que sirvieron de ayuda al momento de convertir la información obtenida del cliente. Estas clases permitieron la conversión de datos obtenidos.

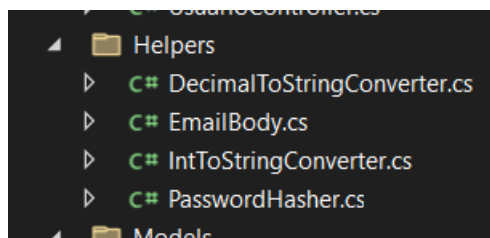


Figura 64: Helpers backend

Para que los controladores puedan acceder a acceder a los servicios creados en el backend se debió utilizar los métodos Addscoped e inyectarlos en estos para que se ocupen en los métodos de los controladores.

```
builder.Services.AddScoped<IEmailService, EmailService>();
builder.Services.AddScoped<TransportistaService, TransportistaService>();
builder.Services.AddScoped<BusetaService, BusetaService>();
builder.Services.AddScoped<RecorridoService, RecorridoService>();
builder.Services.AddScoped<RecorridoDetalleService, RecorridoDetalleService>();
builder.Services.AddScoped<ComentarioService, ComentarioService>();
builder.Services.AddScoped<ReservaService, ReservaService>();
```

Figura 65: Configurar servicios backend

f. Desarrollo del frontend

El cliente fue desarrollado en angular y este está vinculado a la API-REST mediante peticiones que se crean en los servicios del proyecto en Angular, para ello se debió establecer la ruta de acceso, esto se realizó creando una variable tipo string en los servicios creados. Para tener una sola variable global en el proyecto se debe crearla en el archivo enviroments del proyecto, pero desde la versión 16 de Angular este archivo ya no se utiliza.

```
export class AuthService {
    private url:string="https://localhost:7005/api/Usuario/"
    private userDownload: any; //esta variable guarda al token
```

Figura 66: Variable conexion al backend

Para las peticiones HTTP que requieren el uso de autorización se debió agregar el token al encabezado de estas. Este nuevo encabezado se creó y se agregó al ya existente mediante el uso de interceptores. Un interceptor capta la solicitud del cliente y clona o duplica la petición, envía esta petición modificada y le agrega a esta un encabezado donde se aloja el token de autenticación, con ello el servidor recibe la petición y la valida y devuelve una respuesta positiva o negativa. Un interceptor en Angular implementa la interfaz HttpInterceptor para poder usar los métodos que se requieren.

El interceptor adquiere el token del local Storage y lo agrega al header como se observa en la figura

```
@Injectable()
export class TokenInterceptor implements HttpInterceptor {

  constructor(private auth: AuthService, private toast: NgToastService, private router: Router) {
  }

  intercept(request: HttpRequest<unknown>, next: HttpHandler): Observable<HttpEvent<unknown>> {
    const myToken=this.auth.getToken();
    if(myToken){
      request=request.clone({
        setHeaders:{Authorization: `Bearer ${myToken}` }
      })
    }
    return next.handle(request)
  }
}
```

Figura 67: Creación de Interceptor

Una interfaz Guard en Angular contiene un método que se ejecuta bajo las condiciones que el proyecto requiera para acceder a las rutas creadas para el acceso a las páginas. Estos métodos solo devuelven verdadero o falso según la condición creada. Para proteger las rutas del proyecto se estableció que solo se pueden acceder a las rutas si el usuario tiene iniciado una sesión. Implementado la interfaz CanActivate se devuelve a la página del login al usuario que no tenga una sesión activa.


```

@Injectables({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate{
  constructor( private auth: AuthService, private router: Router, private toast: NgToastService){
  }
  canActivate(): boolean{
    if(this.auth.isLoggedIn()){
      return true;
    }else{
      this.toast.error({detail:"Ups", summary:"Inicie sesión primero"})
      this.router.navigate(['login']);
      return false;
    }
  }
}

```

Figura 68: Guards para control de rutas

Después de crear el interceptor y la interfaz Guard se debió modificar el archivo app.module que es el módulo principal del proyecto. Allí en la sección providers se declaró que se utiliza la clase TokenInterceptor

```

providers: [{
  provide: HTTP_INTERCEPTORS,
  useClass:TokenInterceptor,
  multi:true
}],
bootstrap: [AppComponent]
})

```

Figura 69: providers en app.module

Para decodificar el token que se envía desde el backend en el frontend se debió instalar un paquete para el proyecto y después crear los métodos que obtiene del localStorage

```

public getRoleFromStore(){
    return this.role$.asObservable();
}

public setRoleForStore(role:string){
    this.role$.next(role);
}

```

Figura 70: Obtener el rol de localStorage

Estos métodos se suscriben a observables y van a devolver un valor cada vez que dichos métodos lo requieran, así que se debieron definir variables de tipo BehaviorSubject

```

private Nombre$ = new BehaviorSubject<string>(""); //est
private role$ = new BehaviorSubject<string>("");
private nameid$ = new BehaviorSubject<string>("");
//public id!: string;

```

Figura 71: Variables Behaviour Subject

Estos métodos se debieron crear en un servicio e inyectarlos en los componentes que requieran la información como el id del usuario.

```

public idReservaEliminada!:number;
constructor(private reservaService: ReservaService, priv
    this.userHome.getIdFromStore().subscribe(val=>{
        const idFromToken = this.auth.getIdFromToken();
        this.idUser=val||idFromToken;
    })
}
ngOnInit(): void {

```

Figura 72: Obtener id del token

Para este proyecto se definió no sobrecargar el módulo principal declarando los componentes que se utilice en el proyecto y se creó un módulo para cada parte del proyecto. Para sección auth que se encarga del inicio de sesión, registro y carga de contraseña se creó un módulo y su correspondiente archivo de rutas

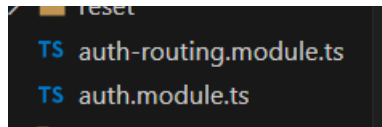


Figura 73: modulo principal Angular

En ese módulo se declaró únicamente los componentes que se ocupan para ese modulo y sus respectivas rutas. Como para el acceso de esas rutas no es necesario estar autenticado no se definió la protección de rutas.

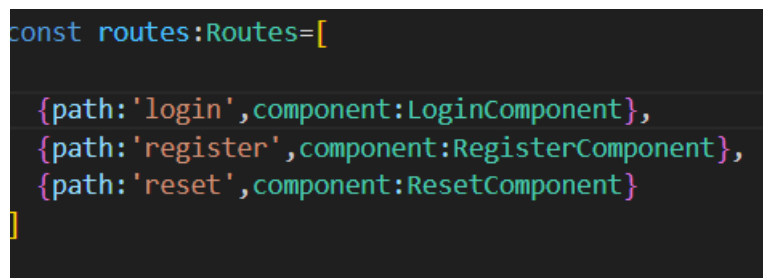


Figura 74: Rutas modulo auth

La sección de Pages contiene todas las páginas que si necesitan autenticación para poder acceder a ellas por lo tanto también se creó un módulo y un archivo de rutas para esa sección. En el archivo de rutas se utilizó la clase guard creada. La ruta home es la principal y de estas heredan la protección las rutas hijas. También se agregó un título que aparece en las pestañas del navegador con su respectivo nombre.



Figura 75: Rutas paginas principales

Servicios

Los servicios en Angular sirven para hacer las peticiones al servicio del backend, un ejemplo de los servicios que se ocupó en el proyecto lo muestra la figura

```
private url:string="https://localhost:7005/api/Usuario/"

private userPayload:any;

constructor(private http: HttpClient, private router: Router) {
  this.userPayload=this.decodedToken();
}

obtenerUsuarioId(id: number){
  return this.http.get<any>(`${this.url}${id}`);
}

register(userObj:any){
  return this.http.post<any>(`${this.url}register`,userObj);
}

login(loginObj:any){
  return this.http.post<any>(`${this.url}authenticate`,loginObj);
}
```

Figura 76: Ejemplo servicios Angular

Los modelos en Angular son clases que deben tener los mismos campos que se obtienen de la base de datos. Si un servicio recibe una petición de un servidor, este devuelve una respuesta del tipo modelo que se haya creado y usado en el servicio

```

export class Usuario{
  iduser!: number;
  nombre!:string;
  apellido!: string;
  telefono!: string;
  direccion!:string;
  email!:string;
}

```

Figura 77: Modelo Usuario

Los archivos helpers son clases que ayudaron a crear métodos que se ocupan en todos los componentes que se requieran como se observa en la figura:

```

export function confirmPasswordValidator(controlName: string, matchControlName:string){
  return(formGroup:FormGroup)=>{
    const passwordControl=formGroup.controls[controlName];
    const confirmPasswordControl=formGroup.controls[matchControlName];
    if(confirmPasswordControl.errors && confirmPasswordControl.errors['confirmPasswordValidator']){
      return;
    }

    if(passwordControl.value !== confirmPasswordControl.value){
      confirmPasswordControl.setErrors({confirmPasswordValidator:true});
    }else{
      confirmPasswordControl.setErrors(null);
    }
  }
}

```

Figura 78: Método validar Nueva Contraseña

Menú lateral

Para este proyecto se definió dos usuarios principales: el rol de estudiante y el rol de transportista. Para esto se debió crear un servicio y crear un arreglo que contenga las rutas que va a ser utilizadas por cada rol en la barra lateral como se observa en la figura:

```

@Injectable({
  providedIn: 'root'
})
export class SidebarService {
  menuEstudiantes:any[]={
    titulo:'Inicio',
    icono:'nav-icon fas fa-home',
    submenu:[
      {titulo:'Transportistas', url:'transportistas', icono:'fa fa-users'},
      {titulo:'Reservas', url:'reservas', icono:'fa fa-tag'},
    ]
  ]
}

menuTransportistas:any[]={
  titulo:'Inicio',
  icono:'nav-icon fas fa-home',
  submenu:[
    {titulo:'Busetas', url:'busetas', icono:'fa fa-bus'},
    {titulo:'Recorridos', url:'recorridos', icono:'fa fa-map'},
    {titulo:'Mis Reservas', url:'reservasestudiantes', icono:'fa fa-tags'},
  ]
}
}

```

Figura 79: Arreglos rutas menú lateral

En la sección html de la barra lateral se creó un ng-container que alojará las redirecciones a las rutas según el rol del usuario accediendo al arreglo que se creó el servicio

```

    <ng-container *ngIf="role==='transportista'">
  <nav class="mt-2">
    <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" data-accordion="false">
      <li class="nav-item has-treeview" *ngFor="let item of menuItemsTransportistas">
        <a href="" class="nav-link ">
          <i class="{{item.icono}}"></i>
          <p class="ml-2">
            {{item.titulo}}
            <span class="badge badge-primary ml-2">{{item.submenu.length}}</span>
            <i class="right fas fa-angle-left"></i>
          </p>
        </a>
        <ul class="nav nav-treeview">
          <li class="nav-item" *ngFor="let subMenuItem of item.submenu">
            <a [routerLink]="subMenuItem.url" class="nav-link" style="cursor: pointer;">
              <i class="{{subMenuItem.icono}} ml-1"></i>
              <p class="ml-2">{{subMenuItem.titulo}}</p>
            </a>
          </li>
        </ul>
      </li>
      <li class="nav-item" style="cursor: pointer;">
        <a (click)="signOut()" class="nav-link ">
          <i class="fas fa-sign-out-alt ml-2"></i>
          <p class="ml-2">
            Cerrar sesión
          </p>
        </a>
      </li>
    </ul>
  </nav>

```

Figura 80: Contenedor para rol transportista

Formularios reactivos

Se ocupó formularios reactivos para poder hacer validaciones en la información que se ingresa a los formularios en la parte del frontend. Un ejemplo de formulario reactivo se muestra en la figura:

```

this.registerForm=this.fb.group({
  nombre:['',Validators.required],
  apellido:['',Validators.required],
  telefono:['',Validators.required],
  direccion:['',Validators.required],
  email:['',[Validators.required,Validators.email]],
  passwd:['',Validators.required],
  rol:['',Validators.required]
})

```

Figura 81: Formulario para registro de usuario

El método onRegister se creó para que guarde la información del usuario validando los campos del formulario de registro. Cuando se valida se instancia al servicio inyectado en el archivo .ts del componente y se suscribe la información. Si existe un error en el servidor se muestra mediante notificaciones tipo toast

```
onRegister(){
  if(this.registerForm.valid){
    this.auth.register(this.registerForm.value).subscribe({
      next:(res)=>{
        this.toast.success({detail:"Registrado!",summary: res.message, duration: 5000});
        this.registerForm.reset();
        this.router.navigate(['login']);
      },
      error:(err)=>{
        this.toast.error({detail:"Ups!",summary: err.error.message, duration: 5000});
      }
    })
  }else{
    this.toast.error({detail:"Ups!",summary: "Ingrese todos los campos", duration: 5000});
  }
}
```

Figura 82: Método registrar con consumo de servicio

Para poder crear las rutas que se muestran en la opción Recorridos se creó un componente y se creó un mapa utilizando Mapbox JS. Para poder cargar el mapa se debió crear un método con el utilizando el token y colocando como punto de partida las coordenadas de la universidad

```
//metodo para cargar mapa
cargarMapa(){
  (mapboxgl as typeof mapboxgl).accessToken = 'token';
  this.map = new mapboxgl.Map({
    container: 'map',
    style: 'mapbox://styles/mapbox/satellite-streets-v12',
    center: [-78.625736, -1.2690819],
    zoom: 16.6,});
  //metodo para crear marcador
  const popup = new mapboxgl.Popup().setHTML(`<h6>Cree su ruta</h6><span>Punto Seleccionado</span>`);
  const marker=new mapboxgl.Marker({draggable:true}).setLngLat([-78.625736, -1.2690819]).addTo(this.map).setPopup(popup);
  this.crearMarcador();
}
```

Figura 83: Método cargar mapa Mapbox

Luego se creó un contenedor en el código html para centrar el mapa como se muestra en la figura:

```
<div class="col-lg-7 d-flex flex-column">
  <div class="btn btn-group">
    <button (click)="mostrarRuta()" class="btn btn-outline-success">Ver ruta</button><!--w-100 es para
    <button (click)="limpiarRuta()" class="btn btn-outline-danger" >Limpiar ruta</button>
  </div>
  <div class="p-4">
    <div id="map-container">
      <div id="map">
        
      </div>
    </div>
  </div>
</div>
```

Figura 84: contenedor para el mapa

Para colocar marcadores por cada punto que marca el usuario se creó un método llamado crear marcador en el evento onclick del ratón

```
crearMarcador(){
  const popup = new mapboxgl.Popup().setHTML(`<h6>Punto seleccionado</h6><span></span>`);
  this.map.on('click', (e) => {
    var lngLat = e.lngLat;
    var lat = lngLat.lat;
    var lon = lngLat.lng;
    var marker = new mapboxgl.Marker({ draggable: true }).setLngLat([lon, lat]).addTo(this.map).setPopup(popup);
    //this.currentMarkers.push(marker);
    let c: Coordenada = new Coordenada();
    c.longitud = lon.toString();
    c.latitud = lat.toString();
    //console.log(c.latitud, c.longitud)
    this.ruta.push(c);
    console.log(this.ruta)
  })
}
```

Figura 85: Crear marcador en el mapa

Por último, para poder dibujar el recorrido de la trayectoria creada se definió un método donde se añade una capa de líneas al mapa cada vez que se lo llama tomando como puntos las coordenadas que se seleccionan en el mapa

```

this.map.addSource('route',{
  'type': 'geojson',
  'data':{
    'type':'FeatureCollection',
    'features':[
      {
        'type': 'Feature',
        'properties':{},
        'geometry':{
          'type': 'LineString',
          'coordinates': ubicaciones
        }
      }
    ]
  }
});

```

Figura 86: Establecer puntos en el mapa

```

this.map.addLayer({
  'id':'route',
  'type':'line',
  'source':'route',
  'layout': {
    'line-join': 'round',
    'line-cap': 'round',
  },
  'paint':{
    'line-color':'yellow',
    'line-width':6
  }
});

```

Figura 87. Dibujar puntos en el mapa

g. Diseño Web Responsive Design con Bootstrap

Para el desarrollo de las páginas en el proyecto se utilizó el enfoque responsivo que se centra en el entorno del usuario dentro del sitio web, este entorno va a depender del dispositivo usado, usando siempre el mismo código HTML en todos los dispositivos y CSS para la renderización.

Una aplicación diseñada con web responsive adapta el diseño para la visualización utilizando cuadrículas adaptables, imágenes y queries.

Cuadrículas: los tamaños de los elementos de la página se basan en porcentajes en lugar de puntos

Imágenes adaptables: se dimensionan de acuerdo con el contenedor que la contenga

Estilo CSS: reglas basadas en las características del dispositivo donde se muestra la aplicación calculando la anchura de la pantalla.

El sistema de cuadrícula de Bootstrap utiliza una serie de filas y columnas para alinear el contenido implementando flexbox. Su cuadrícula implementa 12 cuadrículas en total, basando el diseño en la cuadrícula que implementa Bootstrap para el diseño de las páginas se procedió a instanciar el framework en el proyecto.

En el archivo index se debió crear las URLs de los scripts para que implementar Bootstrap en el proyecto

```
<!-- jQuery -->
<script src="./assets/plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4.5 -->
<script src="./assets/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- AdminLTE App -->
<script src="./assets/dist/js/adminlte.min.js"></script>
<!-- mapbox -->
<script src='https://api.mapbox.com/mapbox-gl-js/v3.0.1/mapbox-gl.js'></script>
```

Figura 88: Referencias scripts Bootstrap

Para el login el diseño se basó en el sistema de fila con 12 columnas en las cuales 7 columnas ocuparon el carousel de imágenes principales y los 5 restantes para el formulario de inicio, esto para pantallas grandes. Para pantallas móviles pequeñas el display del carousel se bloquea.

```

<body class="bg-dark">
  <section>
    <div class="row g-0">
      <!-- aqui comienza el carusel -->
      <div class="col-lg-7 d-none d-lg-block">
        <div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
          <ol class="carousel-indicators">
            <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
            <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
          </ol>
          <div class="carousel-inner carousel-fade">
            <div class="carousel-item min-vh-100 active">
              
              <div class="carousel-caption d-none d-md-block">
                <h5>Prueba la aplicación</h5>
                <p>Accede desde cualquier dispositivo en cualquier lugar solo ingresando serviruta.com en tu navegador web</p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>

```

Figura 89: Código HTML para el login

Para la pantalla de registro se creó un contenedor de tipo panel dentro de otro contenedor que ocupa 5 columnas en pantallas grandes y 5 en pantallas pequeñas

```

<div class="outer">
  <div class="container" >
    <div class="row py-3">
      <div class="offset-md-2 col-lg-5 col-md-7 offset-lg-4 offset-md-3">
        <div class="panel border bg-white">
          <div class="panel-heading" >
            <h3 class="pt-3 font-weight-bold">Regístrate en la aplicación</h3>
          </div>
          <div class="panel-body p-3">
            <form [formGroup]="registerForm" >
              <div class="form-group pt-3">
                <div class="input-field">
                  <span class="fa fa-users p-2"></span>
                  <select class="select" FormControlName="rol" placeholder="seleccione un rol">
                    <option value="" disabled selected hidden>Quiero registrarme como</option>
                    <option value="estudiante" type="text">Estudiante</option>
                    <option value="transportista" type="text">Transportista</option>
                  </select>
                </div>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>

```

Figura 90: Código HTML para registro

Para el envío de correo de restablecimiento de contraseña se implementó otro componente de Bootstrap: modales. Los modales sirven para mostrar un cuadro de información extra al usuario al dar click, la clase tipo modal se divide en el encabezado, cuerpo y pie de página como se muestra en la figura:

```

<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModallabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <p class="modal-title" id="exampleModallabel"><span style="font-size: 25px; font-weight: 500">Olvidaste tu contraseña?</span><br>
        <small>Porfavor ingresa tu correo para enviar el link de reinicio de contraseña</small></p>
        <button type="button" class="btn-close" data-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <div class="form-group pt-3">
          <div class="input-field">
            <span class="fa fa-at p-2"></span>
            <input [(ngModel)]="resetPasswordEmail" (ngModelChange)="checkValidEmail($event)" type="text" placeholder="Correo electrónico">
          </div>
          <small *ngIf="!resetPasswordEmail" class="text-danger">*El correo es requerido</small>
          <small *ngIf="!isValidEmail && resetPasswordEmail" class="text-danger">*El correo no es válido</small>
        </div>
      </div>
      <div class="modal-footer">
        <button id="closeBtn" type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>
        <button type="button" class="btn btn-warning" (click)="confirmToSend()" class="btn btn-primary">Enviar</button>
      </div>
    </div>
  </div>
</div>

```

Figura 91: Cuerpo modal Bootstrap

Para la página principal se mostró una etiqueta header que contiene un div con una clase flex para alinear los elementos al centro.

```

<header class="bg-primary text-white text-center">
  <div class="container d-flex align-items-center flex-column">
    
    <h1 class="masthead-heading text-uppercase mb-0">ServirUTA</h1>
    <div class="divider-custom divider-light">
      <div class="divider-custom-line"></div>
      <div class="divider-custom-icon"><i class="fas fa-star"></i></div>
      <div class="divider-custom-line"></div>
    </div>
    <p class="masthead-subheading font-weight-light mb-0">Recorridos - Reservas - Comunicación</p>
  </div>
</header>

```

Figura 92: Header página principal

Otro componente utilizado con Bootstrap son las migas de pan o breadcrumbs, estas sirven para mostrar al usuario la página en la que se encuentra o dirigirlo a otra página de la aplicación como se muestra en la figura:

```

<section id="bread" class="content-header">
  <div class="container-fluid">
    <div class="row mb-2">
      <div class="col-sm-6">
        <h1 class="text-light">{{titulo}}</h1>
      </div>
      <div class="col-sm-6">
        <ol id="bread" class="breadcrumb float-sm-right">
          <li class="breadcrumb-item"><a routerLink="/home"><i class="fas fa-home"></i></a></li>
          <li class="breadcrumb-item active">{{titulo}}</li>
        </ol>
      </div>
    </div>
  </div>

```

Figura 93: Breadcrumb Bootstrap

Las tarjetas o cards en Bootstrap son un componente importante al momento de mostrar información. Las tarjetas para mostrar la información de recorridos se muestran en un contenedor de tipo card que utiliza la clase flex para alinearlos en la pantalla de acuerdo con el tamaño de esta. Con la clase justify-content se justifica el contenido que muestra la tarjeta y la clase shadow muestra una difuminación trasera. Por último, con la etiqueta style se define el tamaño como se muestra en la figura:

```

<div class="d-flex justify-content flex-wrap margin-left-3">
  <div *ngFor="let rec of recorridos" class="card mx-4 shadow-lg p-3 mb-3" style="width:30rem;">
    
    <div class="card-body">
      <h5 class="card-title">{{rec.numRec}} {{rec.nombreRec}}</h5>
      <h4 class="card-text">Dias de la ruta: {{rec.fechaRec}}</h4>
      <h4 class="card-text">Precio: {{rec.precioRec}} $</h4>
      <h4 class="card-text">{{rec.descripcion}}</h4>
    </div>
    <div class="d-flex justify-content-around mb-5">
      <h3>Salida: {{rec.horaRec}}</h3>
      <button data-toggle="modal" (click)="capturarIdRec(rec.idRec)" data-target="#exampleModal" class="btn btn-warn
    </div>
  </div>
</div>

```

Figura 94: Tarjeta Bootstrap

Todas las tarjetas que se muestran en el proyecto usaron las mismas clases, pero para diferenciarlas se tuvo que modificar los archivos css para darles color, forma a los botones y tamaño

```

.card-img-top{
  border-radius: 50px;
  padding:20px
}

.card{
  border-radius: 30px;
  box-shadow: 0px 0px 0px 0.3px, 0px 19px 38px, 0px 0px 0px 0.22px, 0px 15px 12px;
  overflow: hidden;
  cursor: default;
  transition: all 400ms ease;
}

.card:hover{
  box-shadow: 5px 5px 20px rgba(0,0,0,0.4);
  transform: translateY(-3%);
  background-color: #161616;
}

.card-body{
  padding: 25px;
  margin-top: -15px;
}

```

Figura 95: Ejemplo de código CSS para tarjetas

Los botones de la aplicación también se configuraron con la clase btn-group

```

<div class="btn btn-group">
<button (click)="mostrarRuta()" class="btn btn-outline-success">Ver ruta</button><!--
<button (click)="limpiarRuta()" class="btn btn-outline-danger" >Limpiar ruta</button>

```

Figura 96: Grupo de botones

La estructura de la aplicación se dividió en módulos, esto con la finalidad de no sobrecargar el módulo principal y tener un mejor manejo de código.

Las páginas en Angular se representan por componentes cada componente esta agrupado como se muestra en la figura

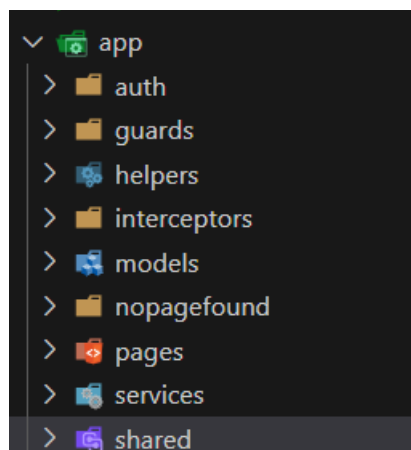


Figura 97: Componentes del proyecto

La carpeta shared representa a los componentes que son compartidos y utilizados en todas las páginas de la aplicación

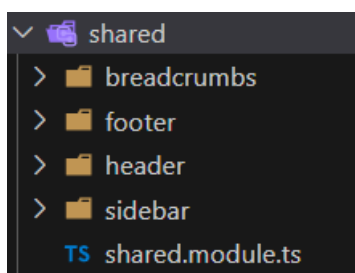


Figura 98: Módulo shared

El módulo pages representa todas las páginas que son accedidas por el usuario que inició sesión.

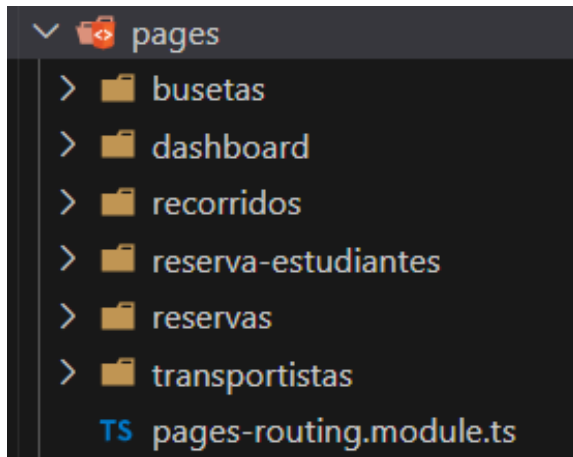


Figura 99: Modulo pages

El módulo auth representa los componentes que muestra la aplicación sin iniciar sesión

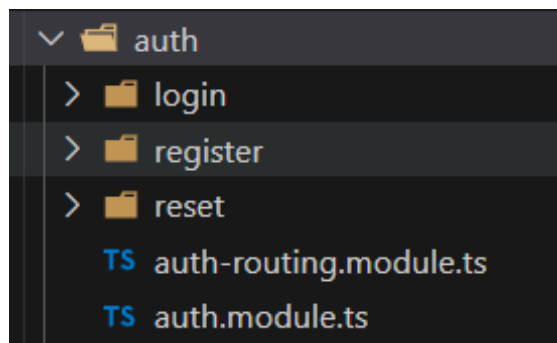


Figura 100: Modulo Auth

La carpeta services representa a los servicios utilizados en la aplicación para hacer las peticiones al servidor

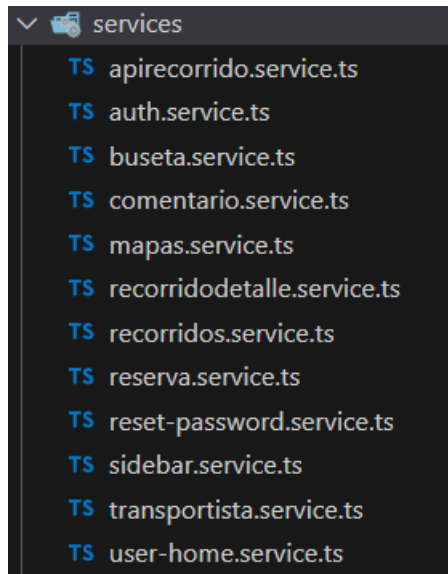


Figura 101: Servicios creados

La carpeta models representa todas las clases que se utilizaron para mapear los datos de las peticiones http extraídas de la base de datos

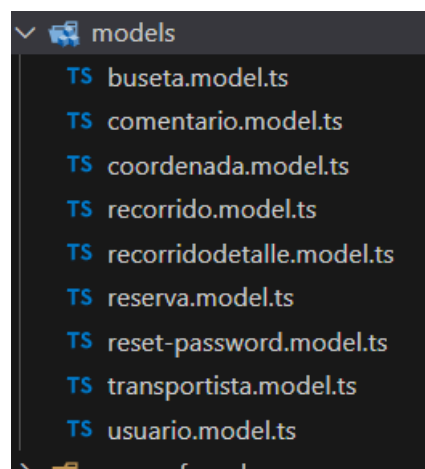


Figura 102: Modelos creados

3.2.6. Pruebas de aceptación

Dado que los estudiantes de la Universidad Técnica de Ambato que ocupan el servicio de busetas de la universidad son los principales usuarios de la aplicación, así como los choferes que proveen el servicio, como parte de las pruebas se realizó un cuestionario detallado en el Anexo 2 basado en los diez principios heurísticos de Nielsen para evaluar la adaptabilidad y usabilidad de la aplicación. el cuestionario se aplicó a 10 estudiantes y a tres choferes que disponen de busetas. Se obtuvo los siguientes resultados:

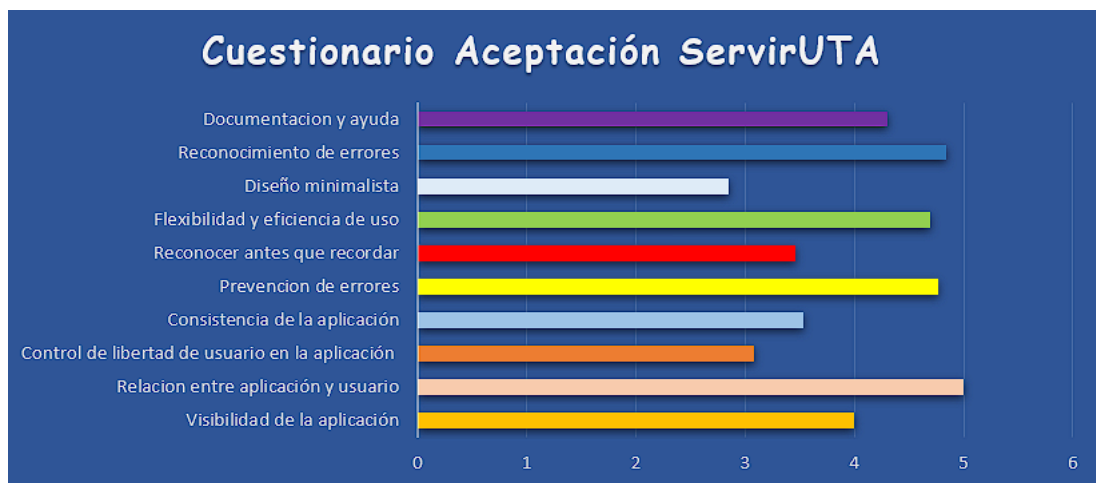


Figura 103. Resultados cuestionario Nielsen

El resultado del cuestionario dio un puntaje promedio de 4.053/5 que equivale a un porcentaje de 81% lo que demuestra que la aplicación cumple con los estándares de usabilidad y adaptabilidad web que propone los principios de Nielsen.

3.2.7. Implantación de aplicación web

Para subir la parte de backend se utilizó un servidor llamado SmarterASP.NET donde se creó un sitio web para alojar la publicación a un precio económico para nuevos usuarios

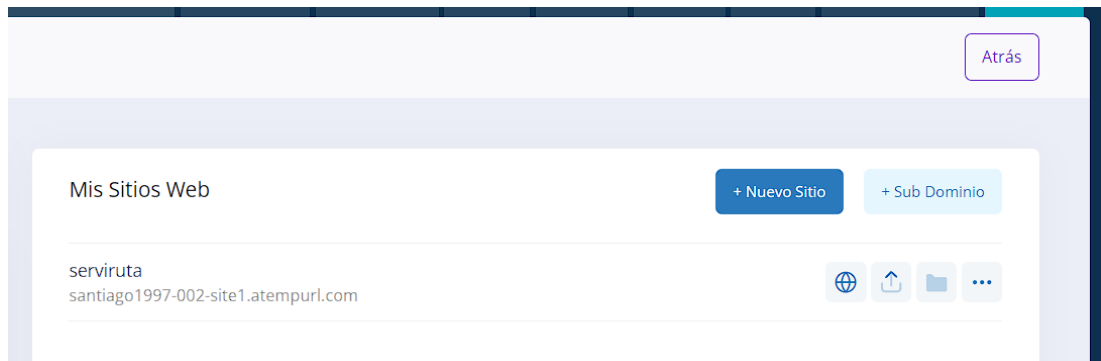


Figura 104. Creación de sitio web en servidor contratado

Para publicar el proyecto se descargó la información de despliegue del sitio y se importó el perfil en el web api del proyecto

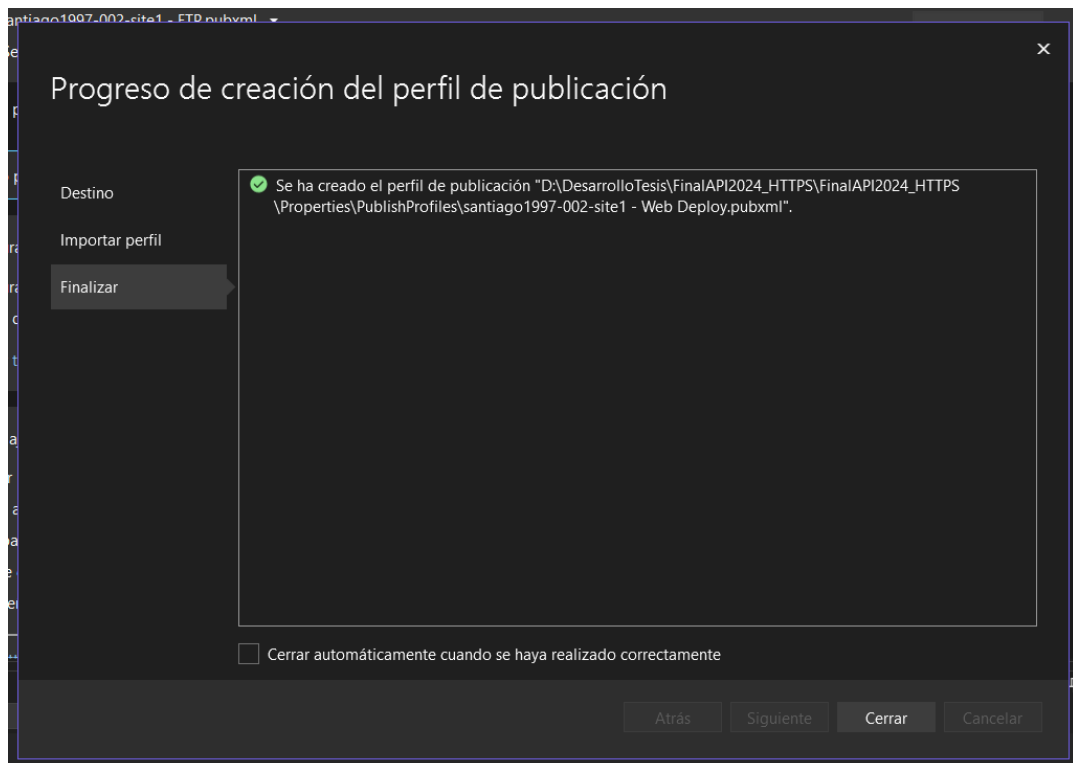


Figura 105. Publicación de web api

Después que la web api se publicó correctamente se procedió a crear la base de datos con el script:

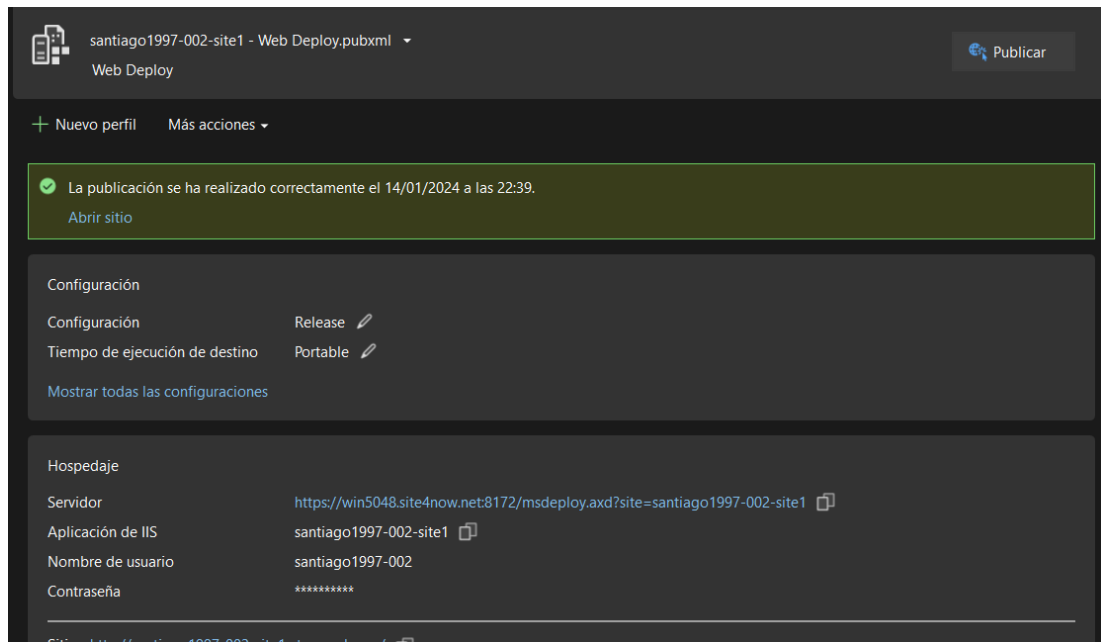


Figura 106. Verificación de publicación

Se configuró la cadena de conexión de la base en el sitio web en el web api.

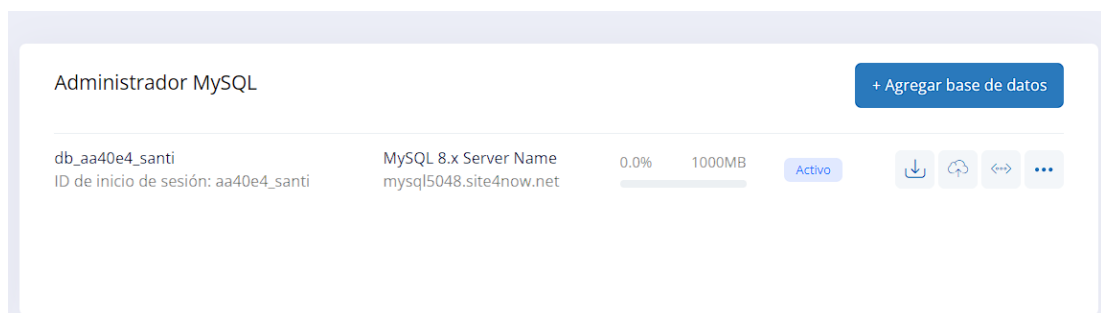


Figura 107. Base de datos en sitio web

Implantación de frontend en Hostinger

Hostinger es un servicio de alojamiento para aplicaciones web que ofrece precios económicos tanto para contratar el sitio como para rentar un dominio. Cuenta con una sola interfaz donde acceder a todas las herramientas de gestión de hosting.

Para subir la aplicación se tuvo que contratar un sitio web en el sitio:

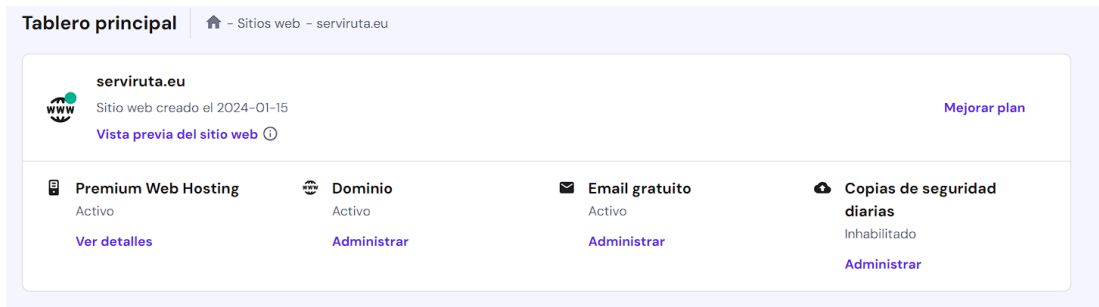


Figura 108. Creación de sitio web en Hostinger

Para poder acceder al sitio también se tuvo que contratar un dominio para la aplicación. Se optó por contratar el dominio serviruta.eu

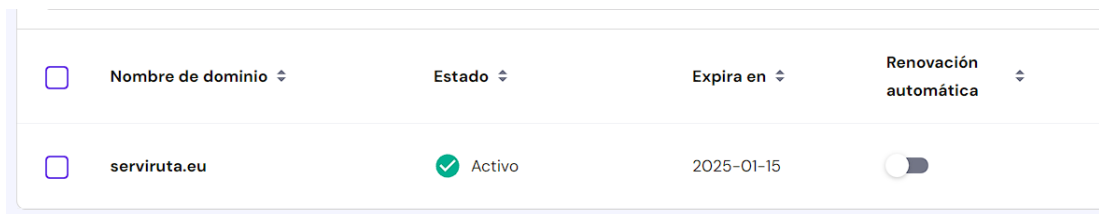


Figura 109. Creación de dominio en Hostinger

La aplicación de Angular se exportó con el comando `ng build --prod` lo que generó la carpeta que se debe cargar en el sitio web.

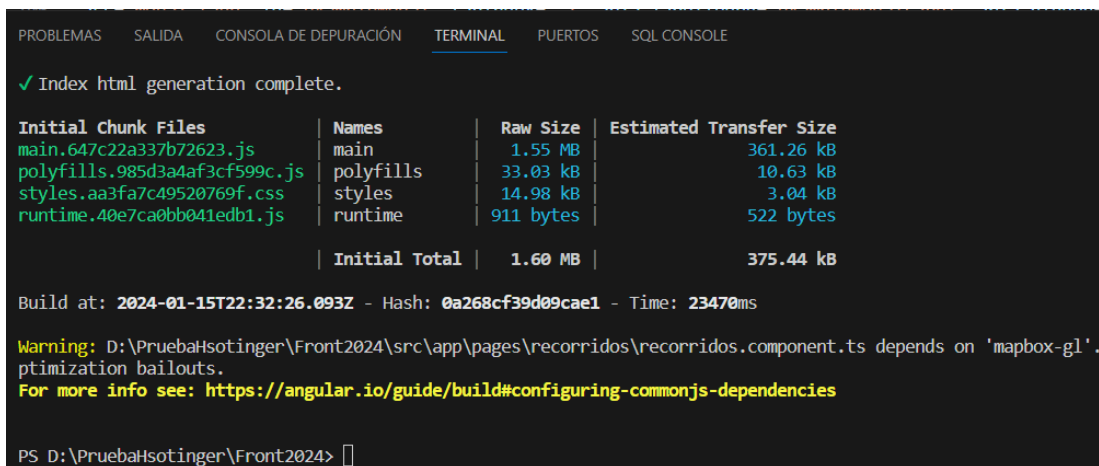


Figura 110. Crear carpeta dist para producción

En Hostinger se cargó los datos que generó la build en la carpeta del dominio contratado a través de Hpanel

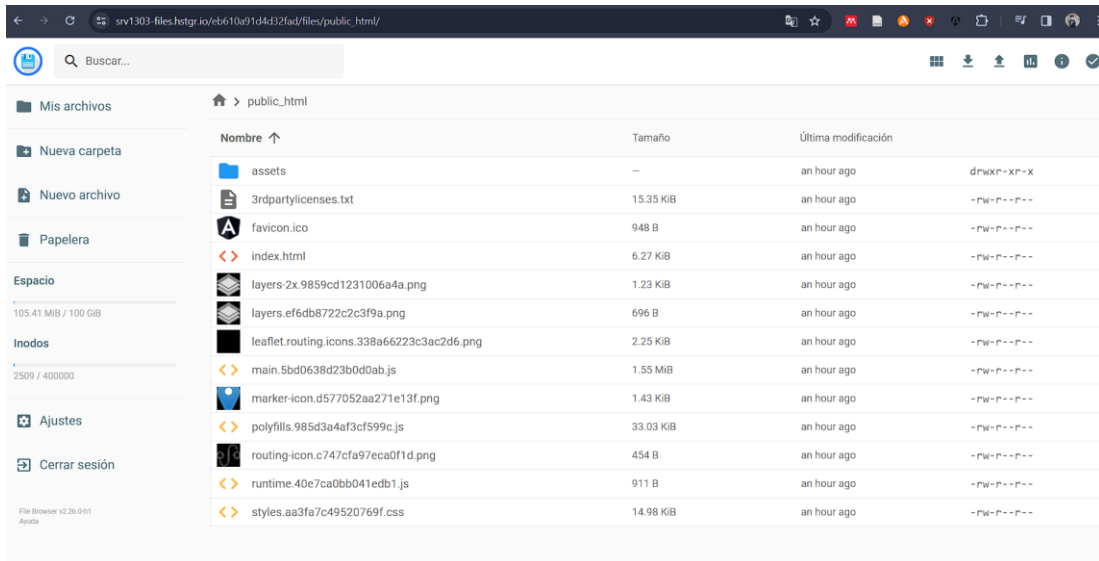


Figura 111. Configuración sitio web en HPanel

La aplicación implantada puede ser accedida desde cualquier navegador que soporte HTML y JavaScript. A través del dominio: <https://serviruta.eu/#/login>. La pantalla de inicio del sitio se observa en la figura:

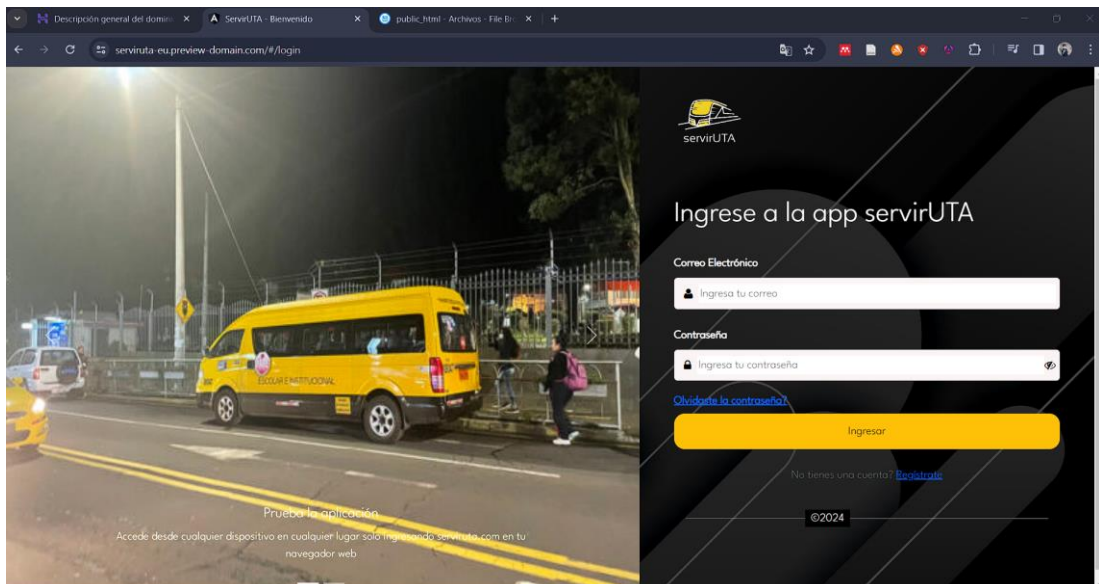


Figura 112. Pantalla de login en sitio web Hostinger



Figura 113. Pantalla home en sitio web Hostinger

CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Dentro del análisis de la investigación se abordó la necesidad de los estudiantes en el sector de transporte estudiantil y se logró identificar mediante los cuestionarios y entrevistas aplicadas que los principales inconvenientes que afrontan los estudiantes al acceder al servicio de transporte son: la descentralización de la información sobre los recorridos y la falta de herramientas destinadas a ese propósito. Las personas que brindan el servicio ponen a disposición las rutas que ofertan mediante el uso de redes sociales y los horarios que se ofrecen se cumplen a lo largo de la jornada educativa. Con esta información se definieron los requisitos funcionales y se establecieron los requisitos no funcionales.
- De acuerdo con los requisitos establecidos para el desarrollo se definió el framework de trabajo que solventó los requerimientos de diseño haciendo uso del diseño web responsivo. Para el uso de Bootstrap se hizo uso de sus clases en las etiquetas HTML para definir los elementos de las páginas y modificando las hojas de estilo CSS del proyecto se consiguió personalizarlas de acuerdo con los requisitos. También se utilizaron varios de los componentes del framework entre los que destacan cards, navbar, breadcrumbs, form y modal.
- La metodología OOHDM proporciona un modelado de objetos robusto y utilizando UML permitió la documentación de cada fase del desarrollo de la aplicación agilizando el proceso de construcción de la interfaz de usuario y su funcionamiento en general, aunque el seguimiento de la metodología dependerá de que tan bien estén planteadas los diagramas navegacionales y los cambios que se puedan realizar en caso de requerirlos.
- La implantación de la aplicación web en internet ayudará a los transportistas a promocionar de una mejor manera sus recorridos e información a los estudiantes que lo necesiten, sin costo y dentro de un entorno seguro. Este proceso se completó después de superar las pruebas correspondientes de funcionamiento del servicio y de la aplicación.

4.2. Recomendaciones

- Se recomienda una minuciosa recolección de información para el desarrollo del proyecto, plantear las preguntas correctamente al igual que los instrumentos de recolección de información, ya que esto permitirá que la aplicación solvete los inconvenientes que tenga el usuario, y que a futuro se pueda implementar nuevas funciones, a favor de lograr un mejor servicio a los estudiantes, donde se puede incorporar la función de rutas óptimas, geolocalización de busetas, experiencia de usuario mejorada entre otras funcionalidades que puedan surgir.
- Al ser una aplicación adaptable a los dispositivos se requiere utilizar una herramienta que haga uso del diseño responsivo. Bootstrap posee muchos componentes, usando grids para posición de estos en la pantalla, Por lo tanto, se recomienda que se realice una capacitación previa de la herramienta ya que esto evitará desfases en el cumplimiento del proyecto y ahorrará mucho tiempo de desarrollo una vez se definan los requisitos no funcionales.
- Se recomienda investigar sobre aplicaciones web que gestionen servicios de transporte ya que esto ayudará a tener una base para la creación de la interfaz de la aplicación. Es posible que surjan cambios al momento de desarrollar la aplicación con los mockups propuestos, por lo tanto, se sugiere que estos sean simples y puedan ser modificados más adelante.
- Se recomienda el uso de una herramienta dedicada a la modelación en UML ya que esto facilitará el diseño en la primera y segunda fase de la metodología OOADM. Es importante que la herramienta de modelado contenga plantillas para desarrollar cada una de las fases, para permitir acercar la metodología a desarrolladores quienes no la han implementado.
- Se recomienda una búsqueda previa de documentación del servicio de alojamiento al momento de implantar una aplicación en un sitio web, ya que diversos problemas puntuales pueden surgir al momento de escoger el sitio, uno de ellos es la seguridad. Se sugiere que antes de elegir una plataforma de hosting se verifique el uso de certificados ya que esto puede traer problemas cuando se trate de establecer conexión entre el cliente y el servidor.

Bibliografía

- [1] C. Mariño, “Entendiendo las preferencias y actitudes al escoger el medio de transporte en estudiantes universitarios,” 2014.
- [2] A. Agurto and J. Hurtado, “Proyecto de inversión para la creación de una empresa de transporte interprovincial privado de puerta a puerta para los estudiantes de la ESPOL,” 2012.
- [3] T. K. Pizarro Ochoa, “Análisis, diseño e implementación de una aplicación web que gestione los servicios de Radio Candela 90.7 FM empleando Responsive Design aplicando la metodología Scrum.,” Feb. 2015, Accessed: Mar. 05, 2023. [Online]. Available: <http://repositorio.utmachala.edu.ec/handle/48000/6042>
- [4] P. A. Calle, “Desarrollo del sistema web aplicando tecnología Responsive Web Design para el control de información de un Broker de la ciudad de Riobamba,” 2019.
- [5] M. Pintag and Naranjo Hernán, “Sistema web para la gestión georreferenciada de los recorridos y horarios del servicio de buses de la Universidad Técnica de Ambato,” 2021.
- [6] O. González, A. Inés, O. Jácome, and M. Isabel, “Movilidad cotidiana de estudiantes universitarios: caso de la Facultad de Arquitectura de la Universidad Central de Ecuador,” *XIII CTV 2019 Proceedings: XIII International Conference on Virtual City and Territory: “Challenges and paradigms of the contemporary city”*: UPC, Barcelona, October 2-4, 2019, May 2019, doi: 10.5821/CTV.8692.
- [7] J. C. Caballero Rueda, N. A. Vergara Harris, J. D. Coley Vives, P. A. Pérez Almario, I. C. Asprilla Riascos, and F. J. Maza Ávila, “Nivel de satisfacción y factores que inciden en la elección del servicio de transporte entre estudiantes de la Universidad de Cartagena, sede Piedra de Bolívar,” *Revista de Jóvenes Investigadores Ad Valorem*, ISSN-e 2711-3485, Vol. 3, N°. 2, 2020 (Ejemplar dedicado a: Revista de Jóvenes Investigadores Ad Valorem), págs. 84-99, vol. 3, no. 2, pp. 84–99, 2020.

- [8] D. F. Hurtado, “Aplicación web administrativa para reserva de servicios de transporte y envío de encomiendas para la empresa Romero y Asociados (AMBASEUR) de la ciudad de Ambato,” 2019.
- [9] W. Orrala and R. Castro, “Desarrollo De Una Aplicación Web Responsive Para sistematizar La Gestión De Planificación Y Seguimiento del Proceso De Visitas Áulicas Para La Carrera De ingeniería En Sistemas computacionales de La Facultad De Ciencias Matemáticas Y Físicas De La Universidad De Guayaquil,” 2018.
- [10] Y. Ricardo Malagón Hernández, C. Eduardo Niño Silva, D. Alejandro Silva Torres Santiago Vásquez Romero, and N. Duan Barbosa Castro, “Desarrollo de un prototipo de aplicación web para las empresas del servicio de transporte especial terrestre,” May 2021.
- [11] N. A. Silva and C. Torres, “Calidad del servicio de transporte urbano en la ciudad de Cuenca,” 2017.
- [12] I. Thomson and Bull Alberto, “La congestión del transito urbano,” *Revista de la CEPAL*, no. 76, pp. 109–121, 2002.
- [13] “Servicio de transporte: ¿Cómo elegir el que mejor se adapte a tus necesidades? - Logistorage.” Accessed: Jan. 14, 2024. [Online]. Available: <https://www.logistorage.com/servicio-de-transporte-como-elegir-el-que-mejor-se-adapte-a-tus-necesidades/>
- [14] “Ruta de transporte: tipos, diseño y planificación de rutas logísticas.” Accessed: Dec. 05, 2022. [Online]. Available: <https://www.beetrack.com/es/blog/ruta-de-transporte-dise%C3%B1o-y-planificaci%C3%B3n-de-rutas-log%C3%ADsticas/>
- [15] C. Mayorga, M. Ruiz, and D. Aldas, “Percepciones acerca de la contaminación del aire generada por el transporte urbano en Ambato,” 2020.
- [16] A. Rodriguez, S. Flavio, R. Cabrera, M. Hermogenes, M. Rivera Prieto, and H. Ascensión, “Uso de las Tecnologías de Información y Comunicación en la Gestión de la empresa de transporte de servicios público Ícaro,” 2017.

- [17] J. Morales-Carrillo *et al.*, “Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática,” 2021.
- [18] E. García, “La Web 2.0,” España, Sep. 2006.
- [19] A. R. Rocío, M. V. Pablo, Martín Ariel Ramírez, G. A. Claudia, and G. D. Mariano, “Vista de Análisis del Diseño Web Adaptativo. Caso de estudio: Universidades Argentinas.”
- [20] C. Mateu, “Software libre - Desarrollo de aplicaciones web,” *Eureca Media SL*, vol. 1, Mar. 2004.
- [21] L. Tayupanda, “Desarrollo de una interfaz web, que permita mejorar la seguridad en la transferencia de estados de servicios web, basado en autenticación y autorización mediante el estándar Json Web Token,” Sep. 2022.
- [22] E. Bernardis, H. Bernardis, M. Berón, and G. Montejano, “Seguridad en Servicios Web,” 2020.
- [23] L. Alberto, C. Santillán, M. Gibert, G. Óscar, and P. Mora, “Bases de datos en MySQL,” *Uoc*, 2009.
- [24] A. Silva, E. Ledezma, J. Castoreña, and Dominguez Alma, “Utilidad del Lenguaje Unificado de Modelado (UML) en el desarrollo de software profesional dentro del sector empresarial y educativo,” *CienciAcierta*, Sep. 2018.
- [25] B. I. Chaucha De, L. A. Cruz, E. Ruidson, B. I. C. Rodrigo, and J. Luis, “Arquitectura cliente servidor en una red de telecomunicaciones para la actualización de la base de datos del registro de atención del centro de salud del distrito de Colcabamba.” Universidad Nacional de Huancavelica, Jun. 16, 2021. Accessed: Jan. 14, 2024. [Online]. Available: <http://repositorio.unh.edu.pe/handle/UNH/4178>
- [26] “¿Qué son los diseños web fluidos, adaptativos y responsivos?” Accessed: Jan. 14, 2024. [Online]. Available: <https://blog.ida.cl/disenio/diferencias-diseno-web-fluido-adaptativo-responsivo/>

- [27] K. Ramírez-Acosta, “Interfaz y experiencia de usuario: parámetros importantes para un diseño efectivo User experience and user interface: important parameters for an effective design,” *Tecnología en marcha*, 2017.
- [28] “¿Qué es el Diseño Responsive? Ejemplos y Ventajas | aeuroweb.” Accessed: Nov. 30, 2022. [Online]. Available: <https://www.aeuroweb.com/que-es-diseno-responsive/>
- [29] J. Waranashiwar and M. Ukey, “Ionic Framework with Angular for Hybrid App Development,” 2018.
- [30] O. C. Novac, D. E. Madar, C. M. Novac, G. Bujdoso, M. Oproescu, and T. Gal, “Comparative study of some applications made in the Angular and Vue.js frameworks,” *2021 16th International Conference on Engineering of Modern Electric Systems, EMES 2021 - Proceedings*, Jun. 2021, doi: 10.1109/EMES52337.2021.9484150.
- [31] J. Jimenez, “Descubre React,” 2015, Accessed: Jan. 14, 2024. [Online]. Available: <http://leanpub.com/descubre-react>
- [32] D. Ovando, “Bootstrap y Laravel, herramientas para el desarrollo de aplicaciones web,” Dec. 2019.
- [33] F. Reynders, “Modern API Design with ASP.NET Core 2: Building Cross-Platform Back-End Systems,” pp. 1–236, Jan. 2018, doi: 10.1007/978-1-4842-3519-5/COVER.
- [34] E. Haro, T. Guarda, A. O. Zambrano Peñaherrera, and G. Ninahualpa Quiña, “Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot,” *Risti*, Dec. 2018.
- [35] A. Calderón and S. Dámaris Valverde Rebaza Jorge Carlos, “Universidad Nacional de Trujillo Metodologías Ágiles,” 2007.
- [36] “Vista de Metodologías ágiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel.” Accessed: Jan. 14, 2024. [Online]. Available: <https://revistas.unamad.edu.pe/index.php/rad/article/view/168/275>

- [37] S. Darjeling, “OOHDM (Método de diseño hipermedia objeto orientado) y normativa ISO 9126,” *Unellez*, Mar. 2014.
- [38] L. A. Guachamin Yumbo, “Implementación de plataforma web mediante la aplicación de la metodología object oriented hypermedia design method (OOHDM), para apoyo a micro, pequeñas y medianas empresas (MIPYMES) afectadas en sus ventas ante la emergencia sanitaria en la provincia de Napo,” 2022, Accessed: Jan. 14, 2024. [Online]. Available: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/34809>
- [39] S. Silva Ascuntar and Y. G. Carolina García, ““Documentación de la metodología Iconix a través del desarrollo del caso Oriéntate Cali,”” 2018.

ANEXOS

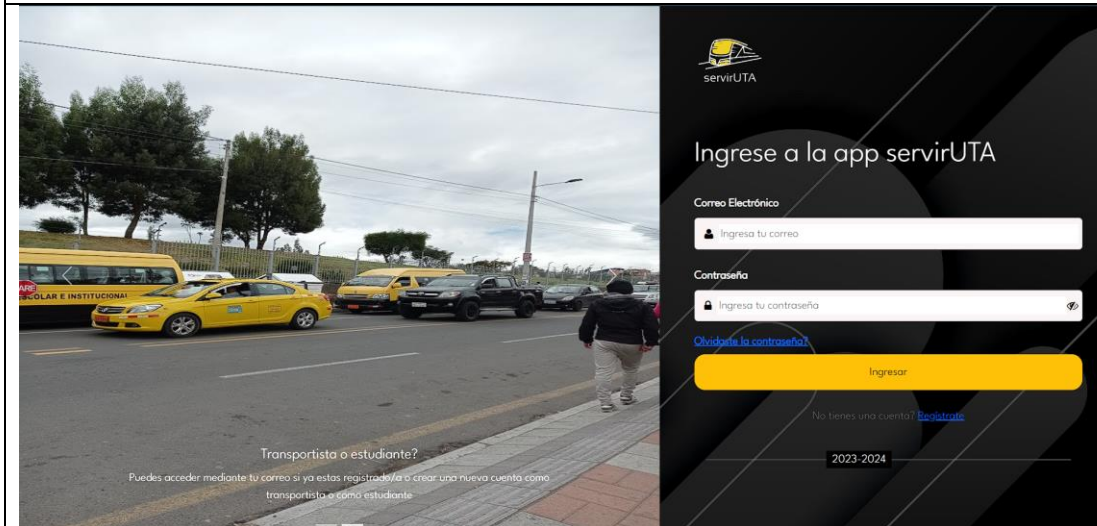
Anexo 1 – Manual de usuario

El presente manual es una guía sobre cómo funciona la aplicación cuando un usuario accede al sitio web.

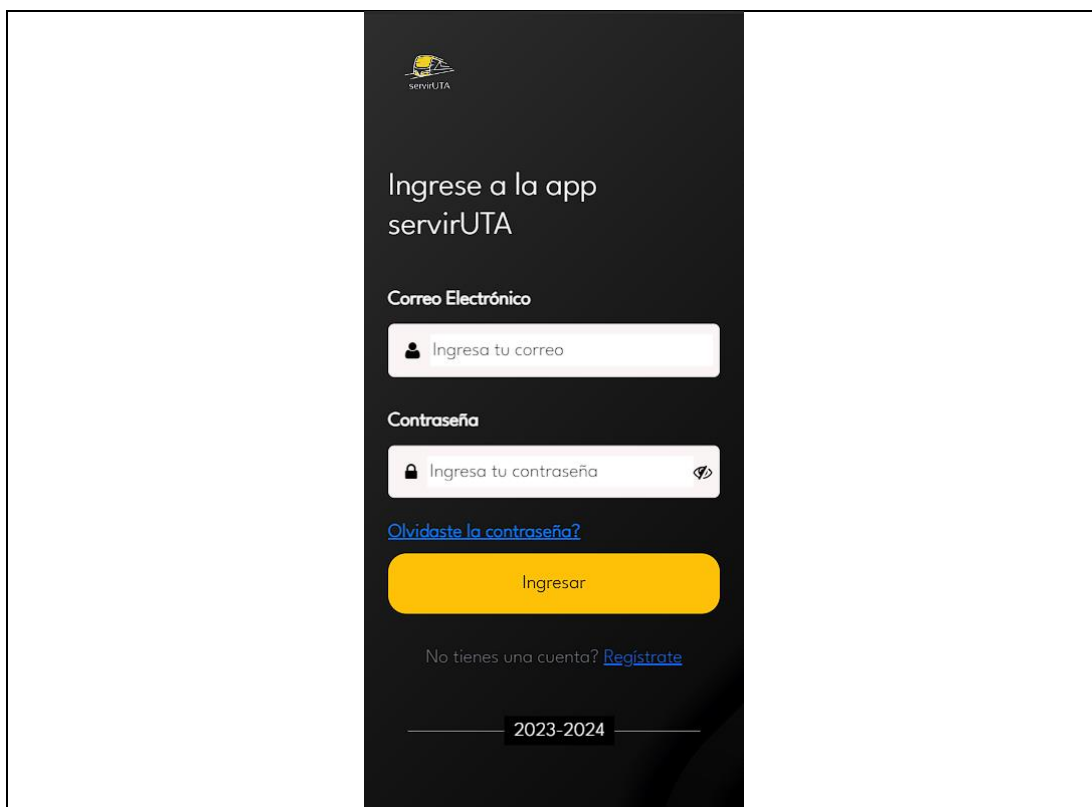
Página de inicio

Cuando el usuario accede a la aplicación mediante el navegador se muestra una página de login, esta se adapta al dispositivo que tenga el usuario

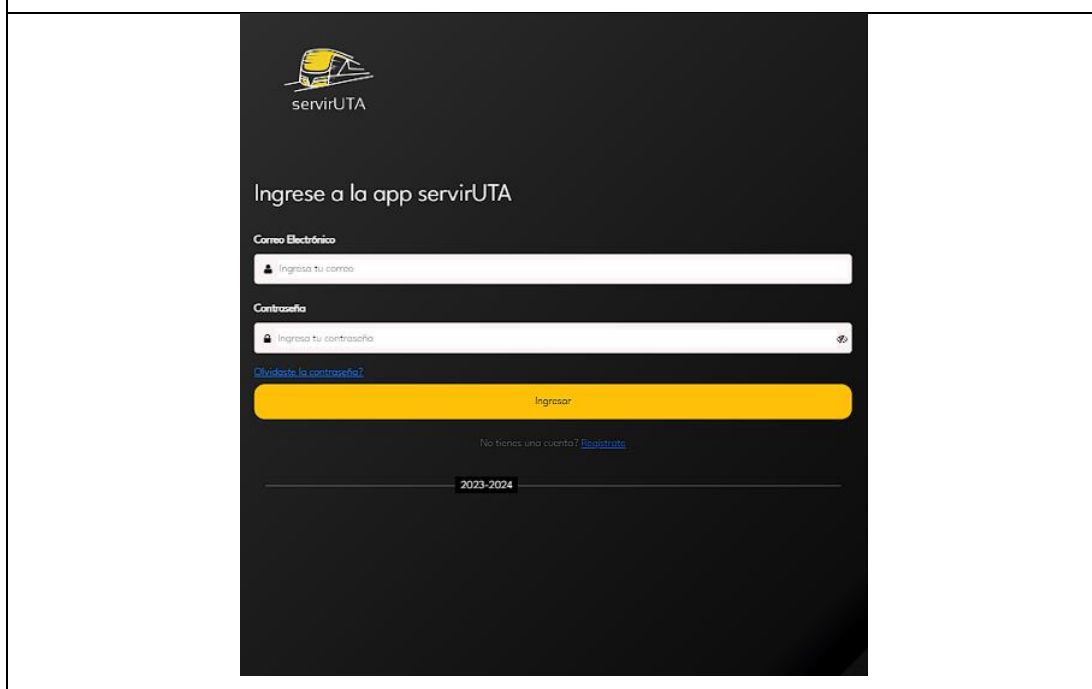
Login – aquí se observa un carrusel de imágenes junto con un formulario de ingreso de correo y contraseña, el botón de ingresar, un botón de registrarse y la opción de



Vista desde móvil



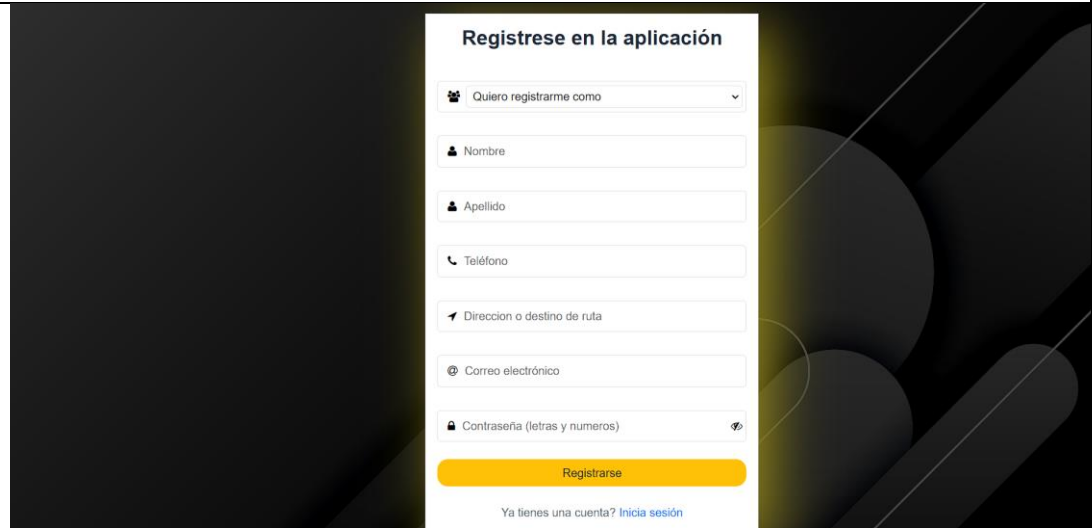
Vista desde tableta



Si se da clic en la opción crear una cuenta aparecerá otra página donde se ingresan los datos de registro

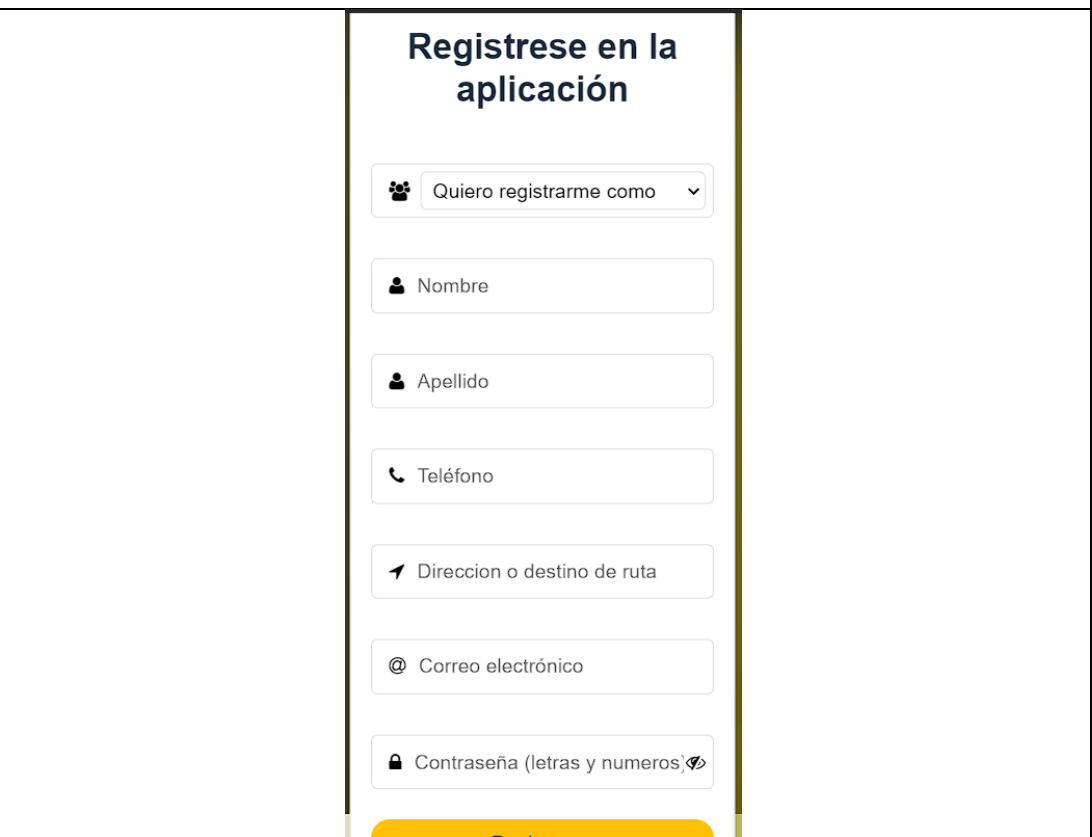
Registro- aquí se llenan los datos escogiendo primero el rol que se va a utilizar en la aplicación.

Vista desde pc



The screenshot shows a registration form titled "Regístrate en la aplicación" on a PC screen. The form is centered on a dark background with abstract shapes. It includes a dropdown menu for "Quiero registrarme como", followed by input fields for "Nombre", "Apellido", "Teléfono", "Direccion o destino de ruta", "Correo electrónico", and "Contraseña (letras y numeros)". A yellow "Registrarse" button is at the bottom, with a link "Ya tienes una cuenta? Inicia sesión" below it.

Vista desde móvil



The screenshot shows the registration form on a mobile screen. The form is titled "Regístrate en la aplicación" and is centered. It includes a dropdown menu for "Quiero registrarme como", followed by input fields for "Nombre", "Apellido", "Teléfono", "Direccion o destino de ruta", "Correo electrónico", and "Contraseña (letras y numeros)". A yellow "Registrarse" button is at the bottom.

Vista desde tableta

Regístrese en la aplicación

Quiero registrarme como

Nombre

Apellido

Teléfono

Dirección o destino de ruta

Correo electrónico

Contraseña (letras y números)

Registrarse

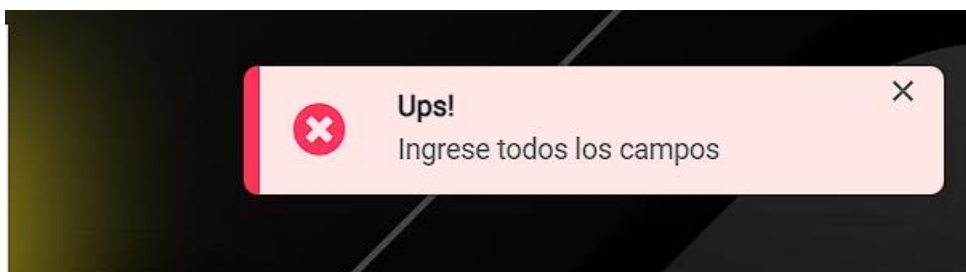
Ya tienes una cuenta? [Inicia sesión](#)

Si no se ingresa datos validos aparecerá el mensaje de error donde se registre el error

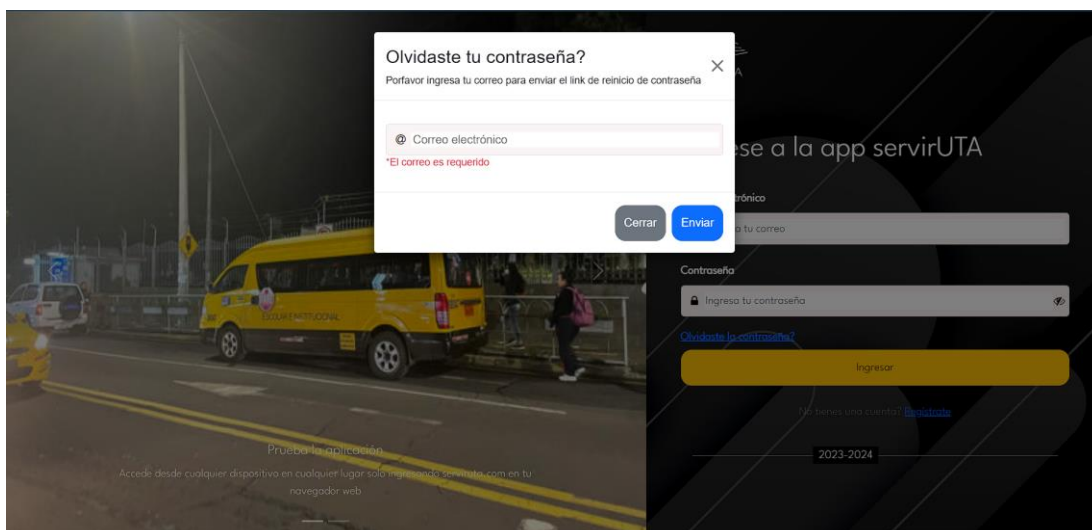
Dirección o destino de ruta

*Ingrese su domicilio

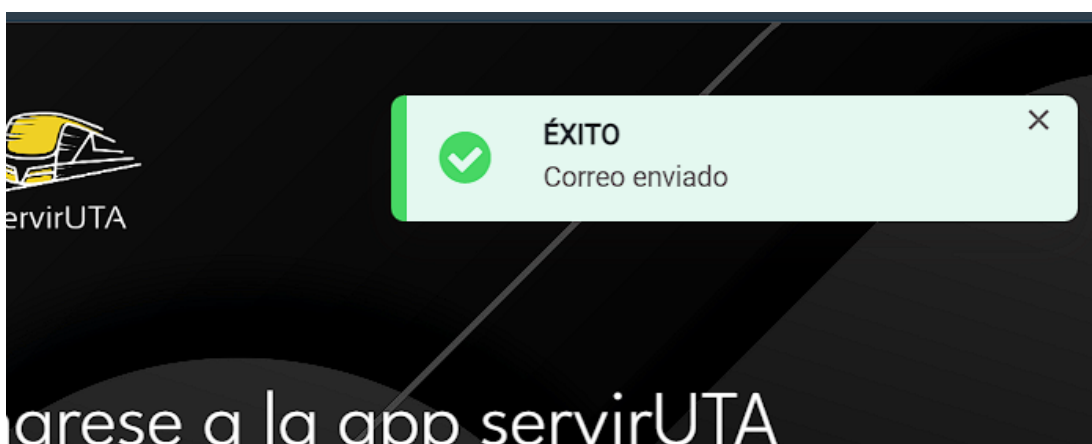
Y notificaciones tipo toast con el error



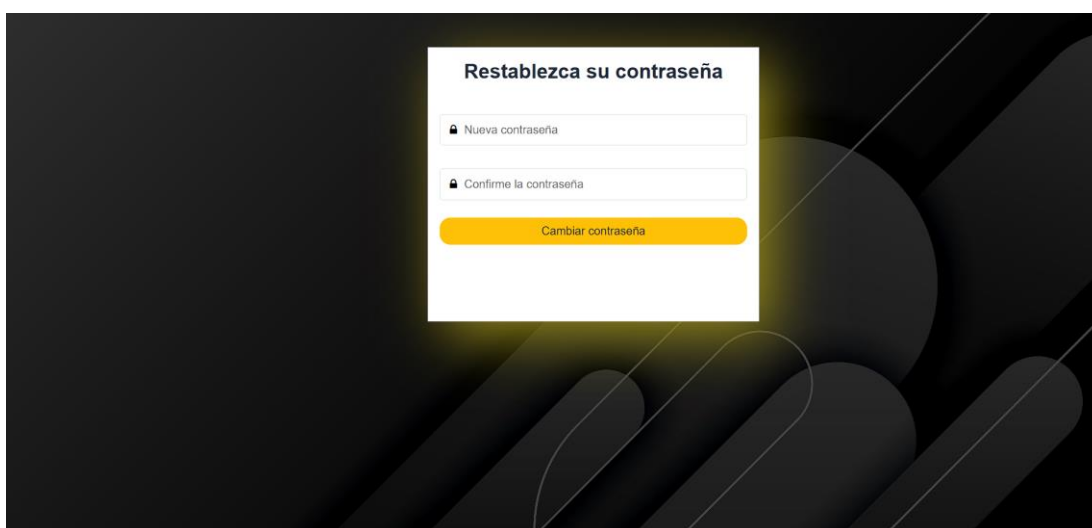
Si el usuario olvidó su contraseña se puede abrir un modal para enviar el correo de reinicio de contraseña



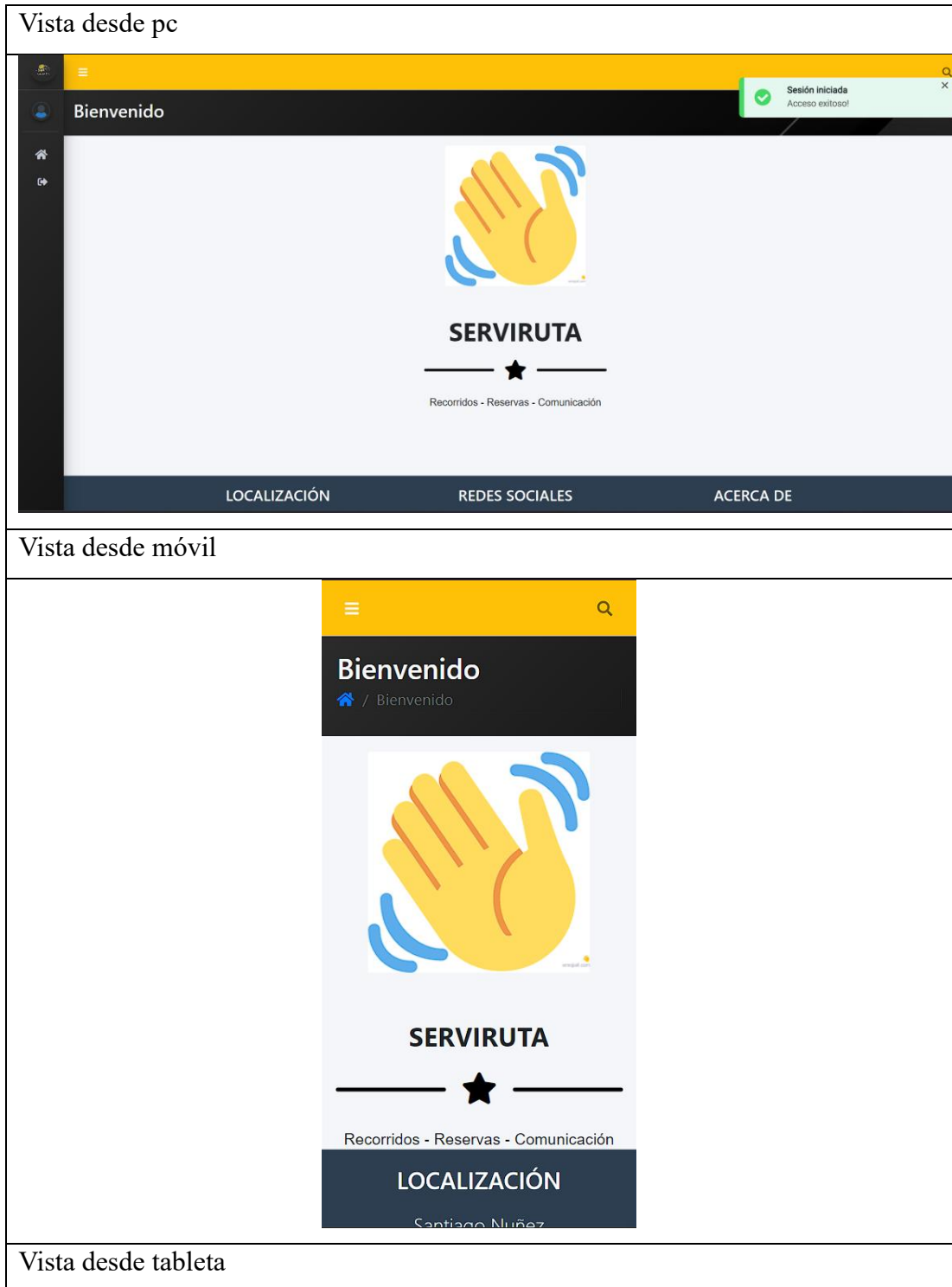
El correo enviará un mensaje de confirmación al correo registrado para que se abra otra página donde se ingresa la nueva contraseña

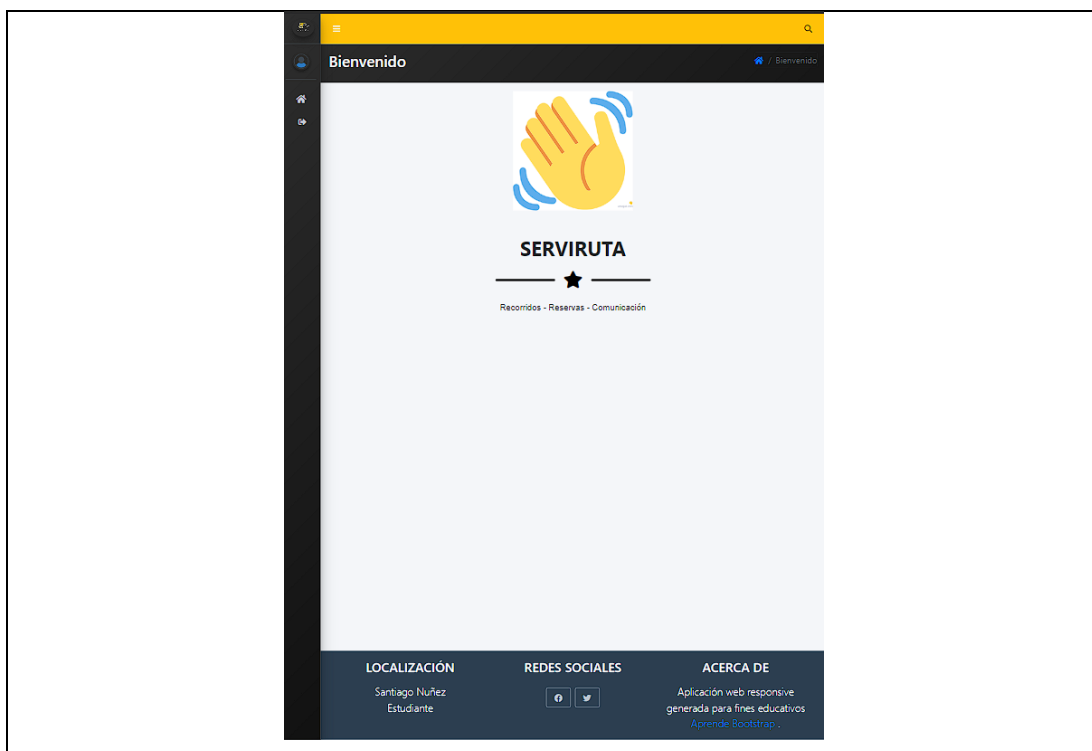


Aquí se ingresa la nueva contraseña



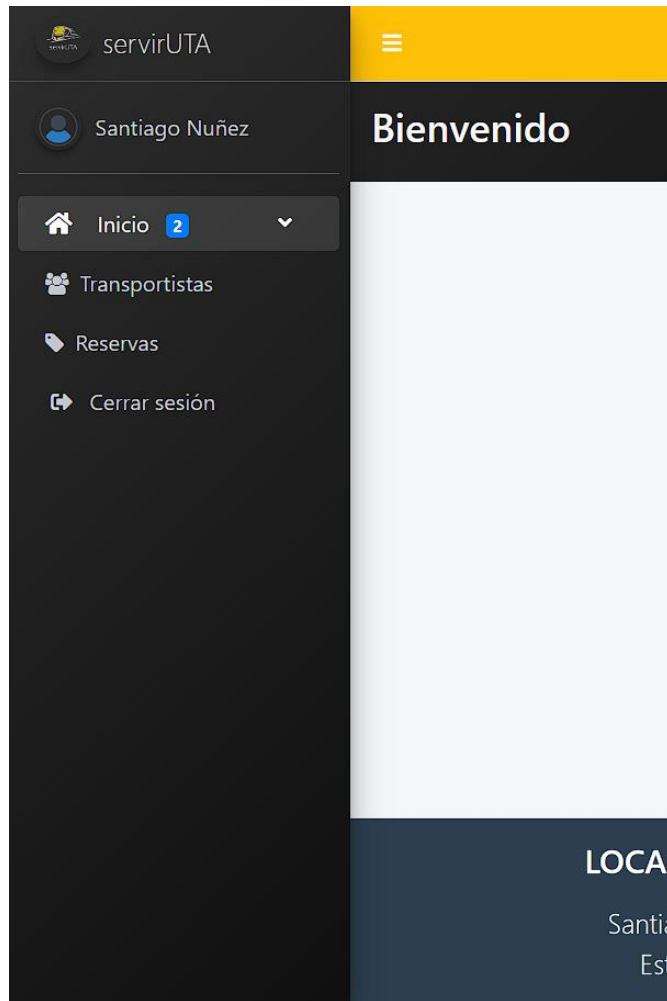
Una vez que se acceda a la aplicación la página principal mostrara un mensaje de acceso exitoso





ACCEDER COMO ESTUDIANTE

Si se accede como estudiante la barra lateral mostrará las opciones que brinda la aplicación para ese rol



En la pestaña transportistas se puede observar a todos los conductores que se han registrado en la aplicación, el estudiante debe escoger uno de ellos y al dar clic en enviar mensaje se le redirigirá a WhatsApp con el número del transportista tomado de la base de datos con un mensaje solicitando más información.

Vista desde pc




servirUTA

Santiago Nuñez

Transportistas

Contacto con transportistas

Envía un mensaje a los transportistas registrados en la aplicación


 <p>Dirección: Baños Nombre: Conductor1 Conductor1 Email: conductor1@gmail.com</p> <p>Enviar mensaje</p>	 <p>Dirección: Puyo Nombre: Conductor2 Conductor2 Email: conductor2@gmail.com</p> <p>Enviar mensaje</p>	 <p>Dirección: Benítez Nombre: conductor3 conductor3 Email: conductor3@gmail.com</p> <p>Enviar mensaje</p>
---	---	---

Vista desde móvil

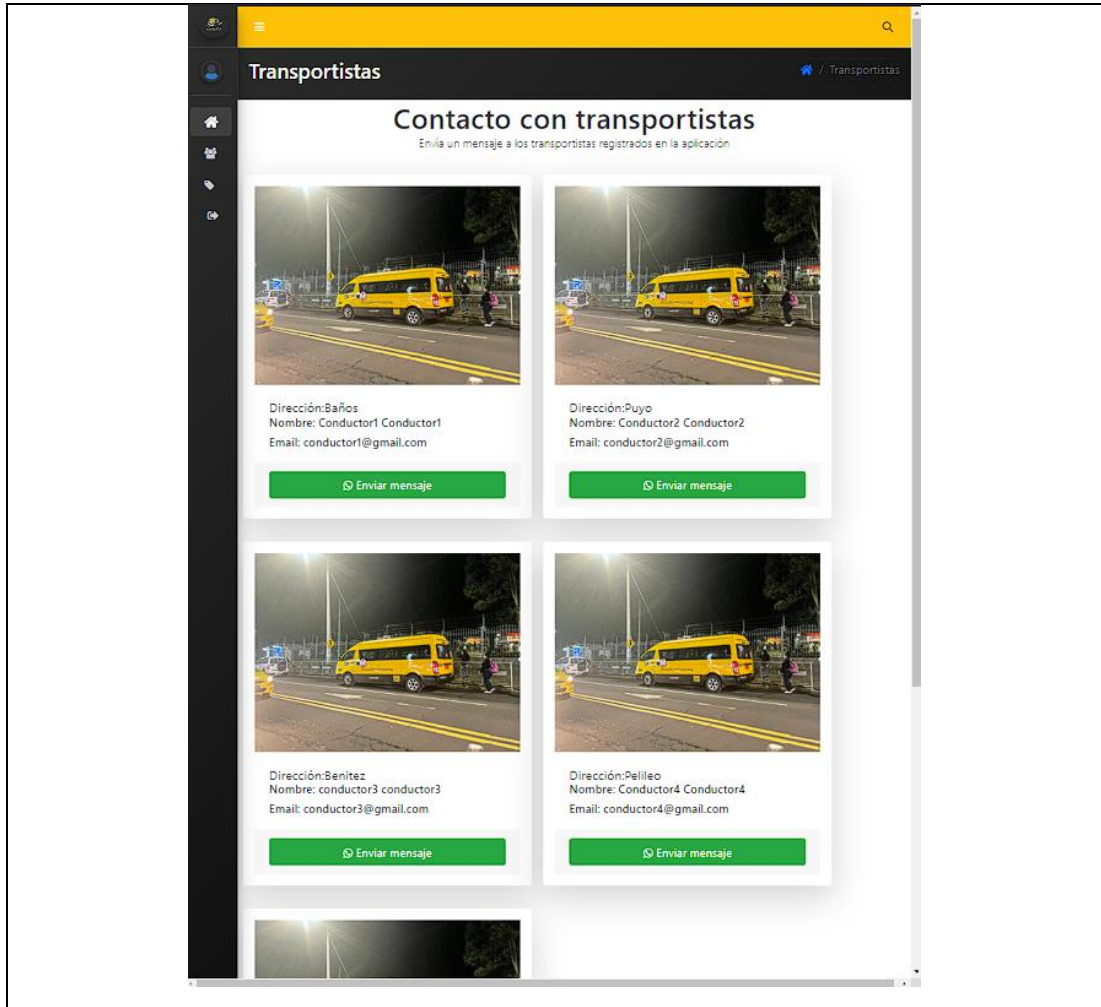
Transportistas

Contacto con transportistas

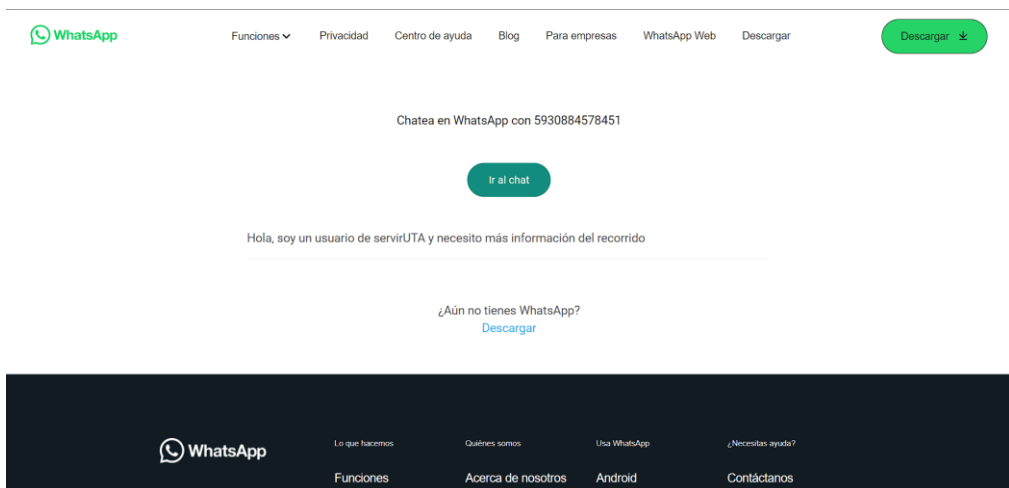
Envía un mensaje a los transportistas registrados en la aplicación

 <p>Dirección: Baños Nombre: Conductor1 Conductor1 Email: conductor1@gmail.com</p> <p>Enviar mensaje</p>
--

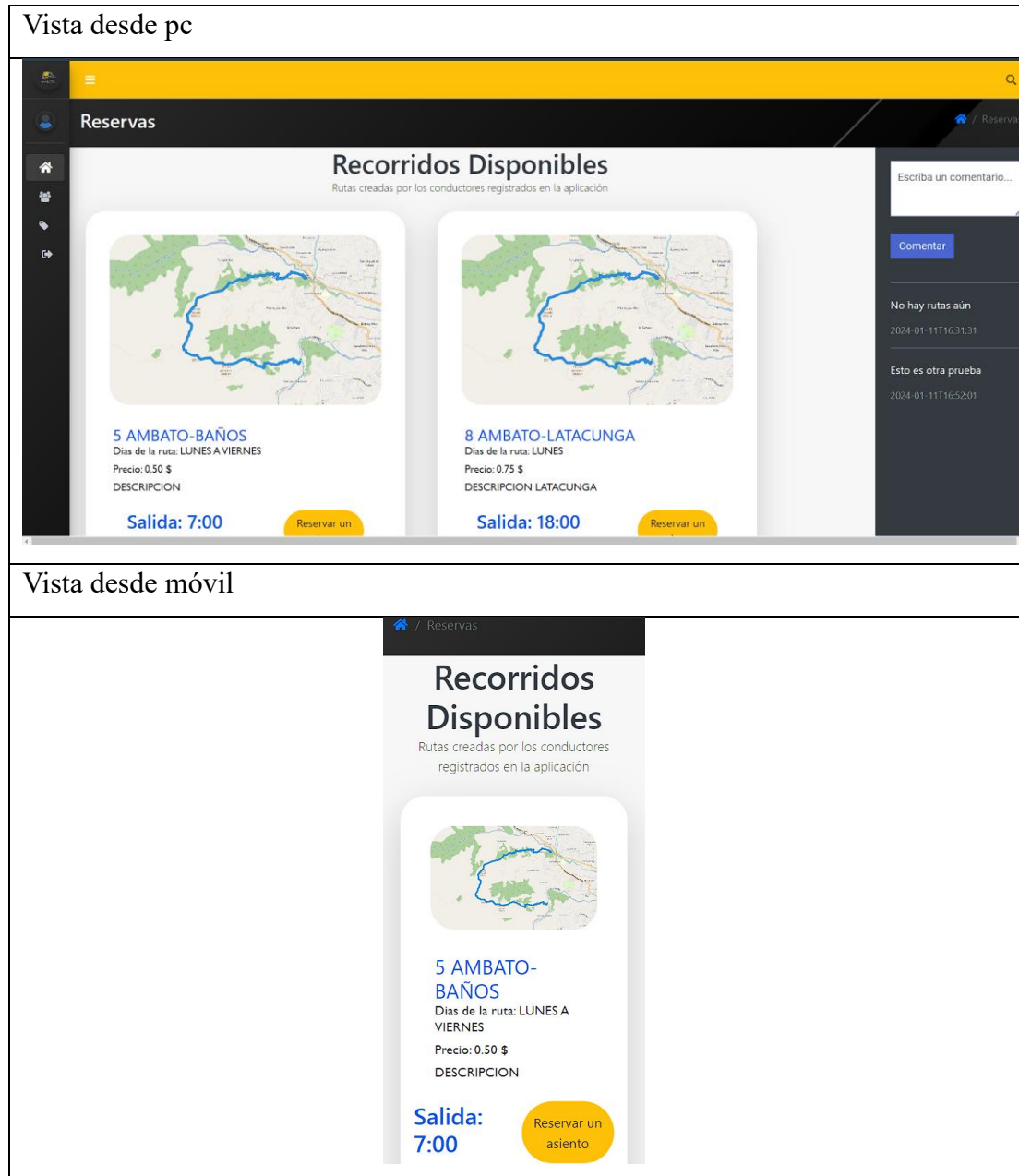
Vista desde tableta



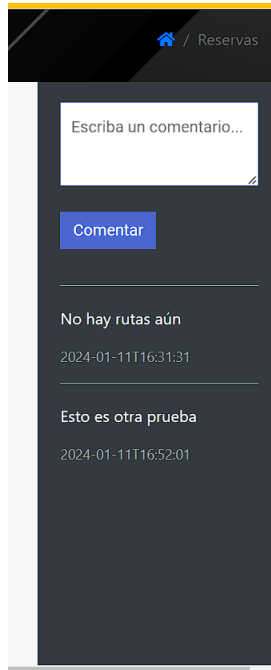
Aquí se envía el mensaje al número registrado



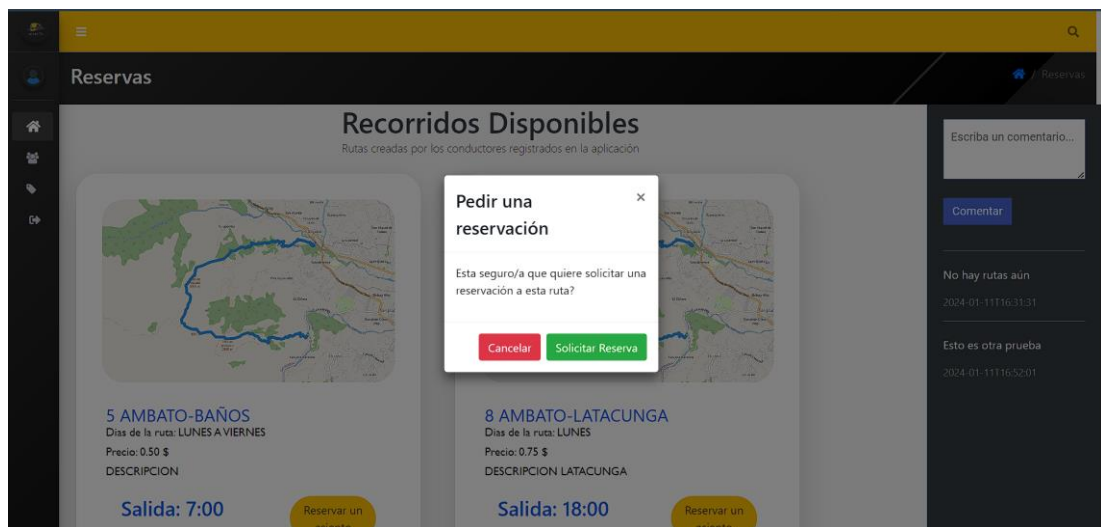
En la opción de reservas se puede observar tarjetas que contienen la información de las rutas creadas por los transportistas registrados en la aplicación, su horario, su costo y hora de salida. Cada ruta contiene un botón para solicitar una reserva a ese recorrido



En la parte de comentarios se pueden ver todos los comentarios que han hecho los usuarios de la aplicación y se puede agregar uno



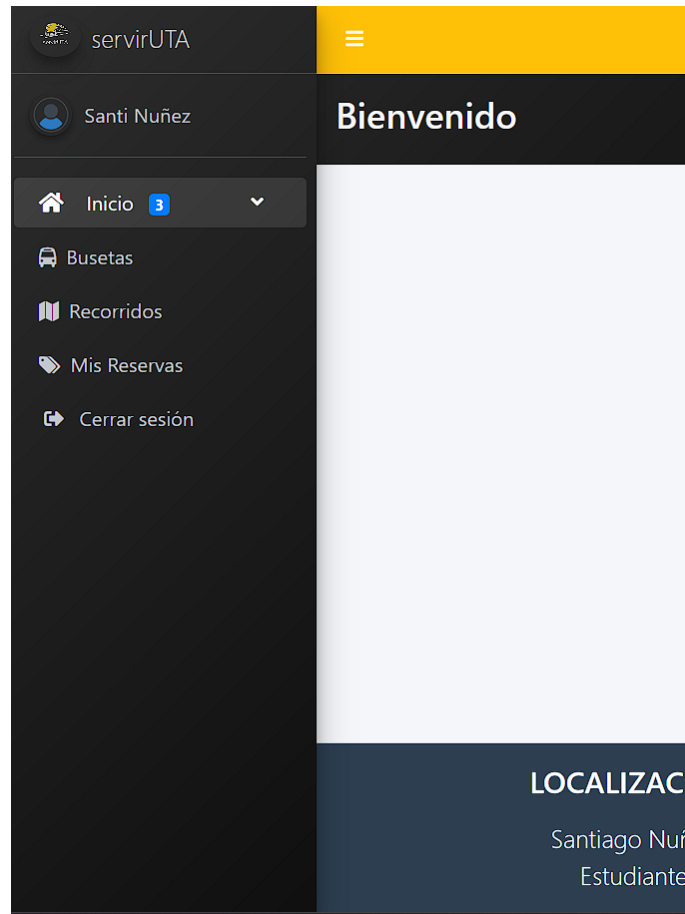
Al momento de seleccionar una ruta y pedir una reserva se abrirá un modal pidiendo confirmar la reservación.



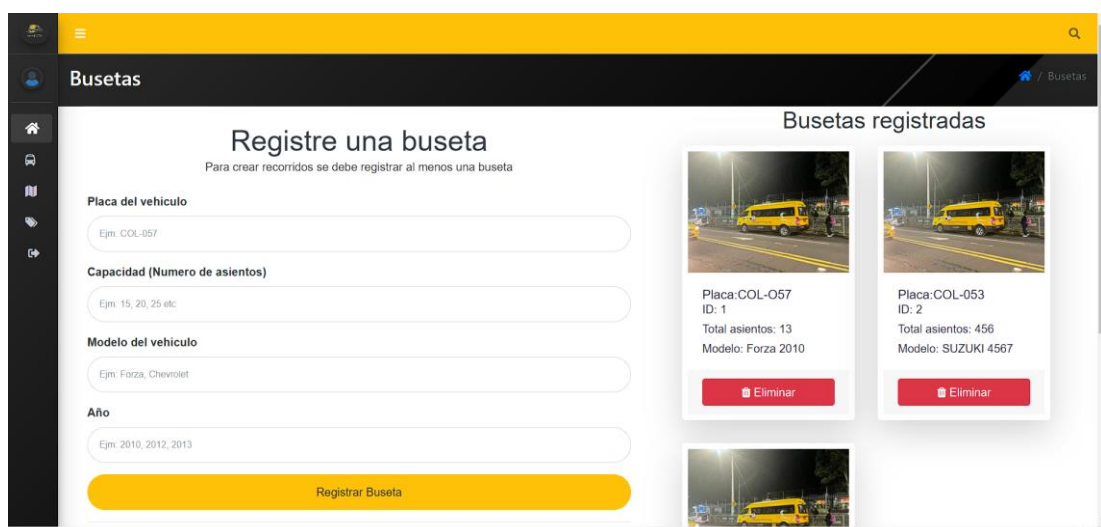
La reserva le llegará al chofer que creó la ruta.

ACCEDER COMO CHOFER

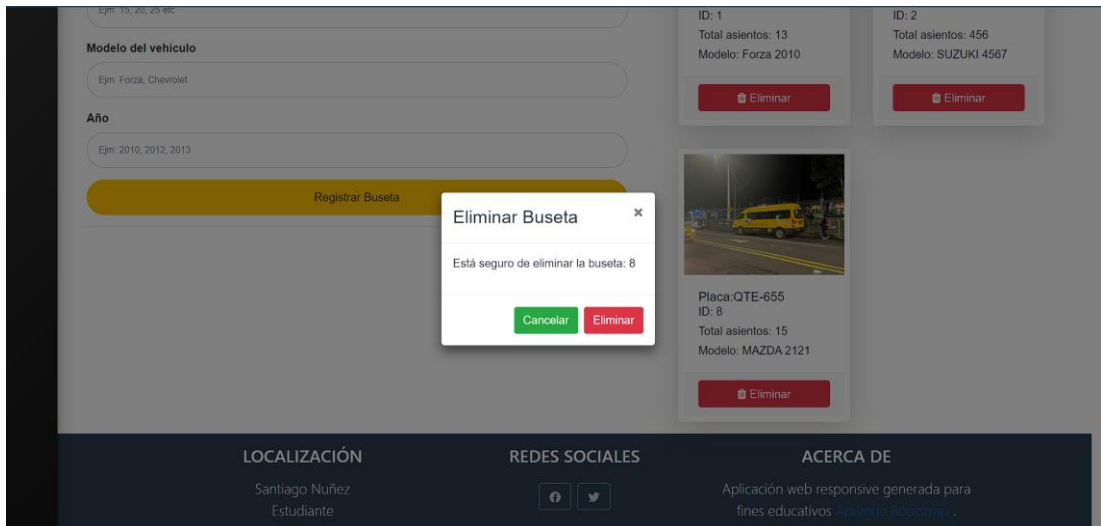
Al momento de registrarse si se inicia sesión como chofer se tienen las siguientes opciones: Busetas, Recorridos y MisReservas



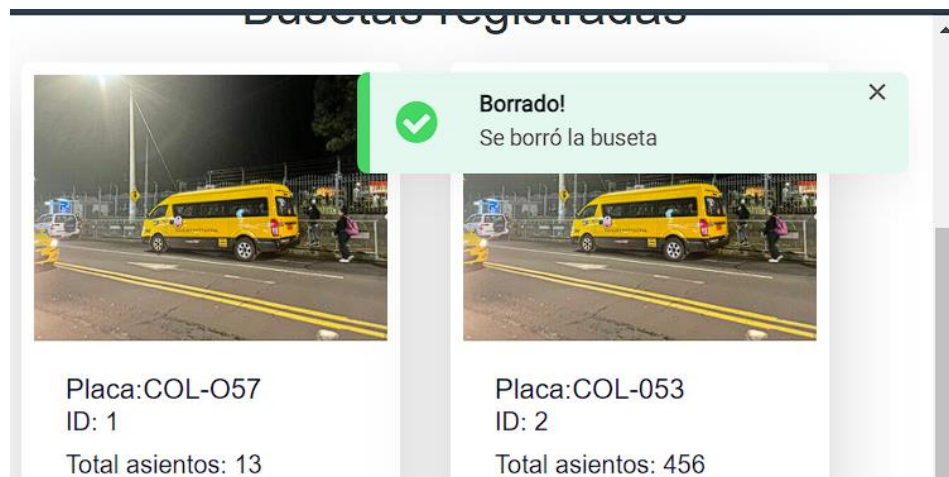
Si accedemos al menú busetas se desplegará la opción de registrar busetas para usarlas en los recorridos creados

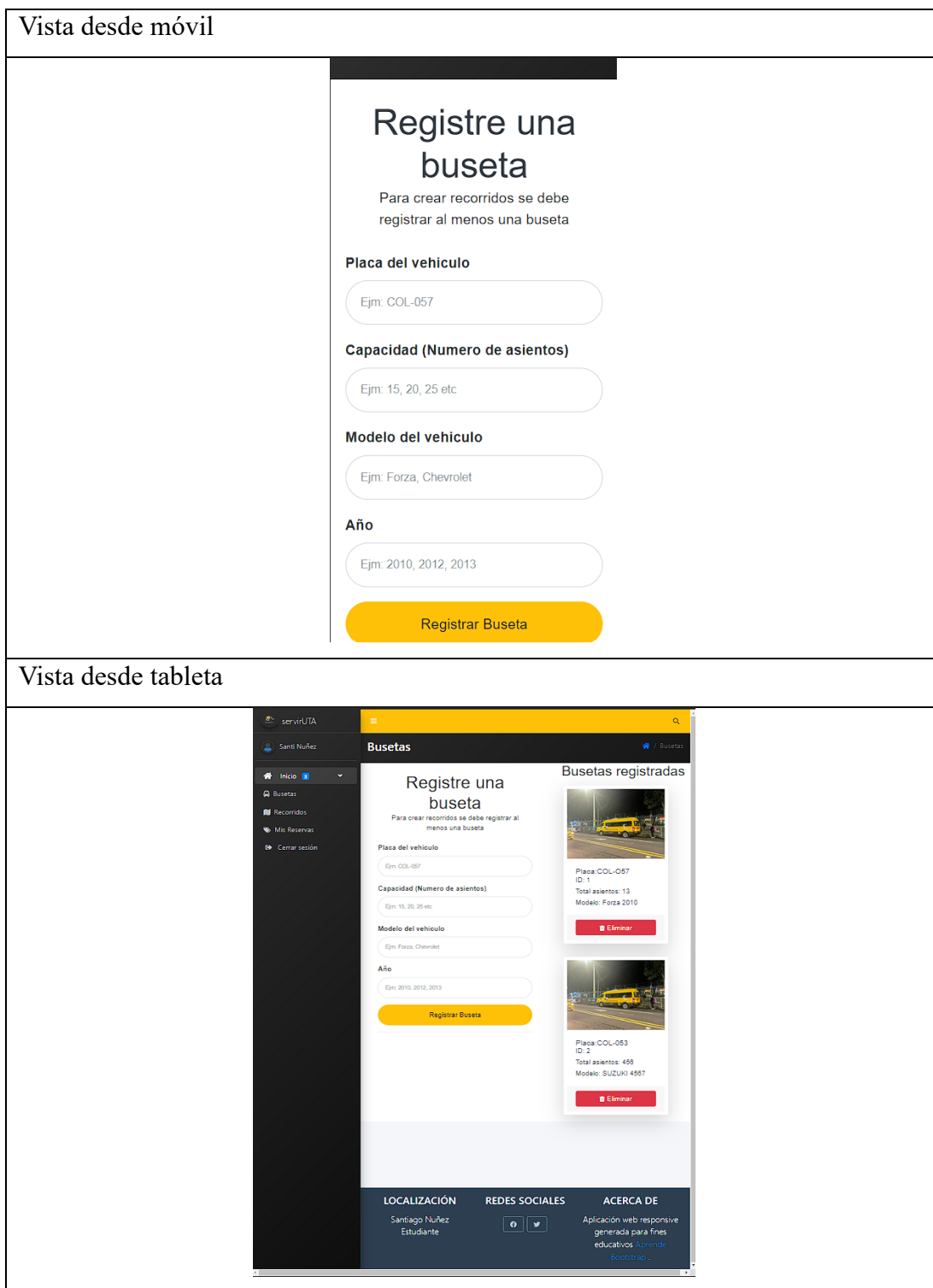


Aquí se puede observar las busetas registradas en forma de tarjetas. Estas busetas también se pueden eliminar de la base de datos

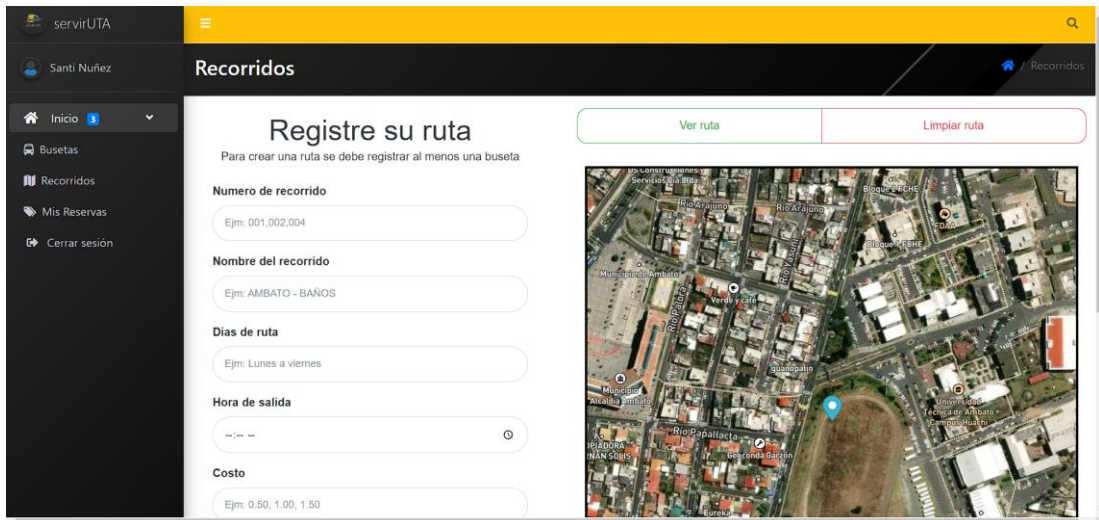


Mensaje de confirmación tipo toast de tarea exitosa

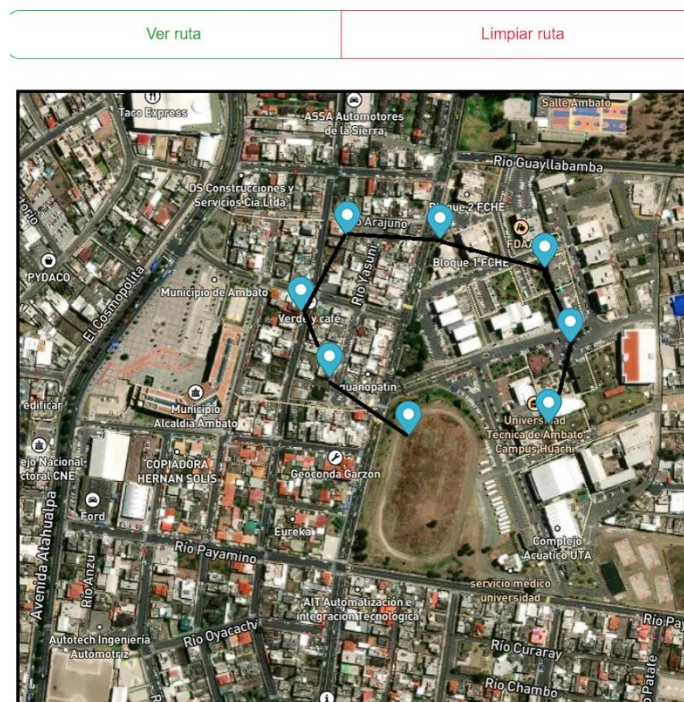




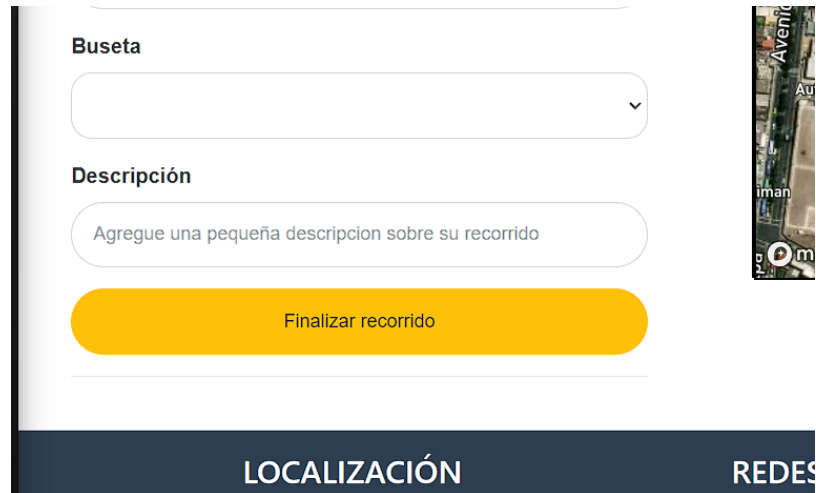
Si se ingresa a la opción recorridos se desplegará un formulario para ingresar los datos de la ruta a crearse junto con un mapa para seleccionar los puntos por donde pasará esa ruta. Los dos botones Ver ruta y limpiar Ruta sirven para crear la ruta y borrar la ruta antes de guardar el recorrido



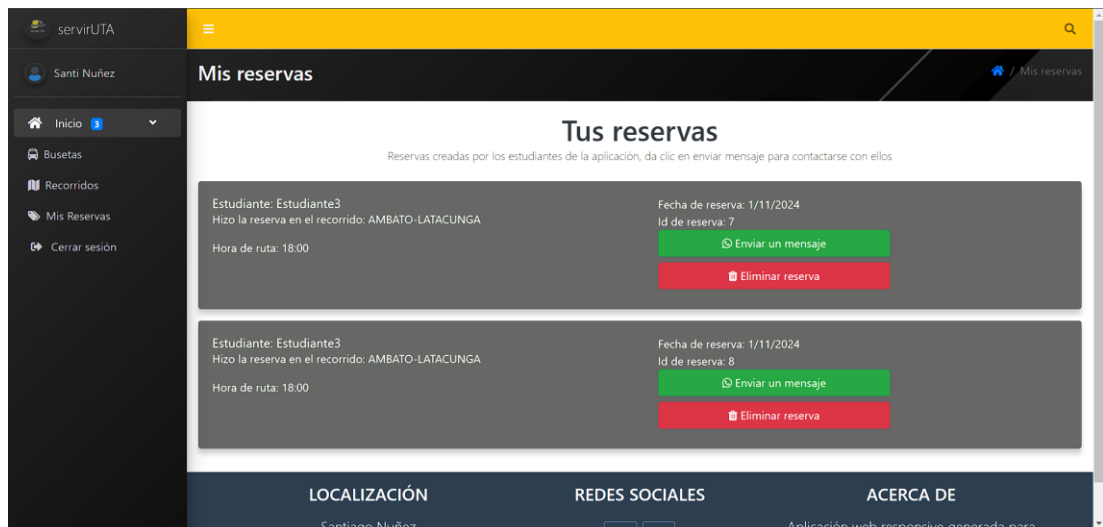
Al dar clic en la opción crear ruta se unen los puntos y se guardan las coordenadas



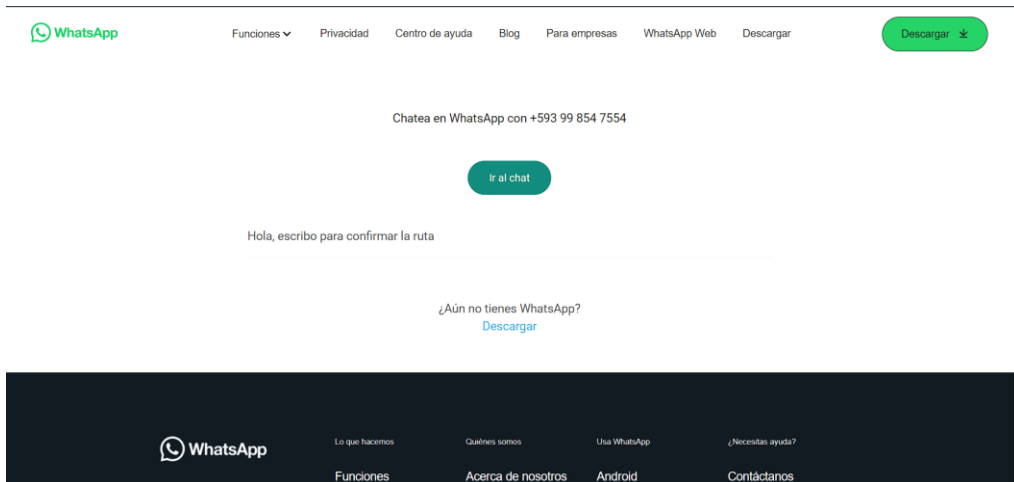
Después se da clic en finalizar recorrido y se guardan las rutas. En la opción Busetas se despliegan las busetas registradas en la aplicación



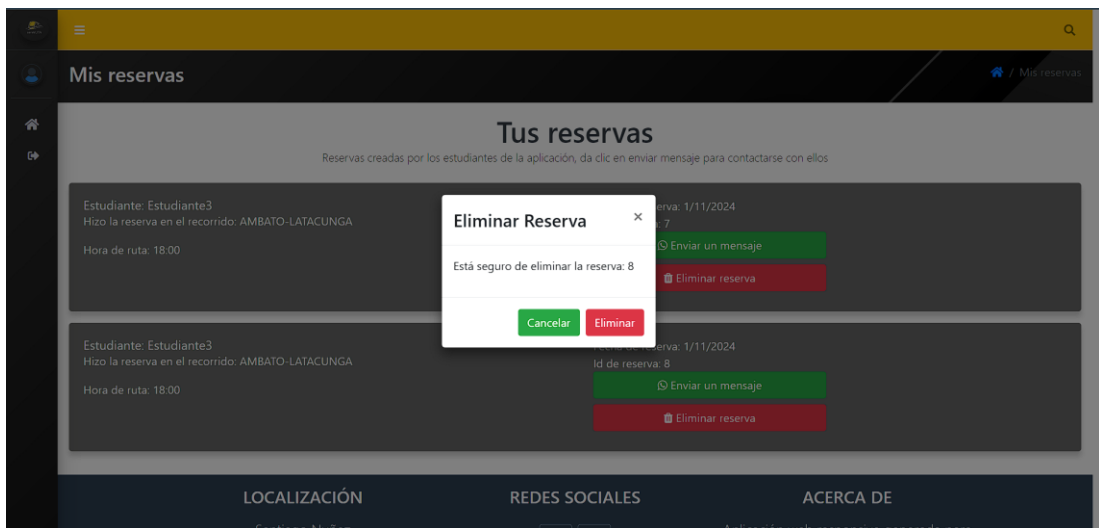
En la pestaña MisReservas se puede visualizar las reservas que han pedido al usuario que esta logueado. Las tarjetas de reservas muestran la información del estudiante y del recorrido que se ha pedido la reserva. El conductor puede contactarse con el estudiante directamente dando clic en enviar whatsapp para confirmar la reserva o puede eliminarla



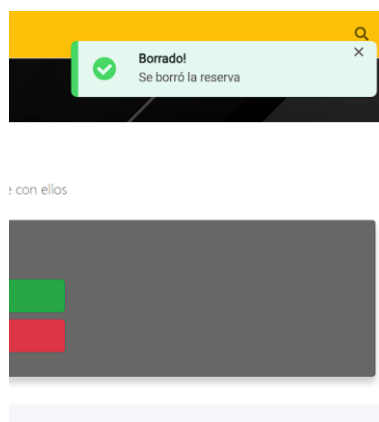
Se manda un mensaje al estudiante que hizo la reserva



O se puede eliminar la reserva



Mensaje tipo toast cuando se elimina una reserva



Anexo 2

Cuestionario de la evaluación heurística aplicando principios de Nielsen para evaluar la adaptabilidad y usabilidad de la aplicación

Preguntas	Puntuación (0-5)	Recomendaciones
1. Visibilidad de la aplicación		
¿El diseño de la interfaz de usuario es adecuado al contexto de la aplicación?		
2. Relación entre la aplicación y el usuario		
¿El contenido de la aplicación está acorde a las funciones que el usuario requiere?		
3. Control de libertad del usuario en la aplicación		
¿Puede navegar por todo el contenido de la aplicación sin que alguna opción lo limite?		
4. Consistencia de la aplicación		
¿Los elementos de la interfaz tienen un diseño coherente?		
5. Prevención de errores		
¿Se presentan mensajes de error que limitan cometer errores al usuario?		
6. Reconocer antes que recordar		
¿El usuario dispone de opciones para recuperar la contraseña en caso de pérdida?		
7. Flexibilidad y eficiencia de uso		
¿La aplicación se adapta al dispositivo que disponga el usuario?		
8. Diseño minimalista		
¿El diseño de la interfaz es fácil de entender?		
9. Reconocimiento de errores		
¿El usuario observa mensajes de error descriptivo cuando ingresa datos?		

10. Documentación y ayuda		
----------------------------------	--	--

¿El usuario tiene acceso a información relevante de acuerdo con su rol al utilizar la aplicación?		
---	--	--