



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRONICA E
INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES E INFORMÁTICOS

Tema:

SISTEMA INFORMÁTICO BASADO EN TARJETAS INTELIGENTES PARA AUTOMATIZAR EL COBRO DE PASAJES, DE LA COOPERATIVA DE TRANSPORTE URBANO 'UNIÓN AMBATEÑA' EN LA CIUDAD DE AMBATO.

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de Software

AUTOR: Bryan Alexander Amaguaña Amaguaña

TUTOR: Ing. Edwin Hernando Buenaño Valencia, Mg.

Ambato –Ecuador

septiembre – 2021

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: SISTEMA INFORMÁTICO BASADO EN TARJETAS INTELIGENTES PARA AUTOMATIZAR EL COBRO DE PASAJES, DE LA COOPERATIVA DE TRANSPORTE URBANO 'UNIÓN AMBATEÑA' EN LA CIUDAD DE AMBATO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Bryan Alexander Amaguaña Amaguaña, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, 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 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2021.

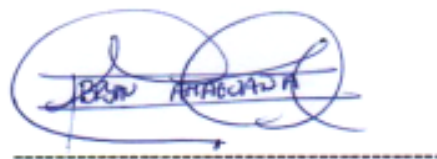
Ing. Edwin Hernando Buenaño Valencia, Mg.

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA INFORMÁTICO BASADO EN TARJETAS INTELIGENTES PARA AUTOMATIZAR EL COBRO DE PASAJES, DE LA COOPERATIVA DE TRANSPORTE URBANO 'UNIÓN AMBATEÑA' EN LA CIUDAD DE AMBATO. Es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2021.



Bryan Alexander Amaguaña Amaguaña

CC: 1804902904

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Bryan Alexander Amaguaña Amaguaña, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA INFORMÁTICO BASADO EN TARJETAS INTELIGENTES PARA AUTOMATIZAR EL COBRO DE PASAJES, DE LA COOPERATIVA DE TRANSPORTE URBANO ‘UNIÓN AMBATEÑA’ EN LA CIUDAD DE AMBATO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, septiembre 2021.

Ing. Pilar Urrutia, Mg.
PRESIDENTA DEL TRIBUNAL

Ing. Julio Balarezo, PhD
PROFESOR CALIFICADOR


Ing. Hernán Naranjo
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2021.



Bryan Alexander Amaguaña Amaguaña

CC: 1804902904

AUTOR

DEDICATORIA

A Dios por permitir iniciar y terminar una etapa de mi vida dándome bendiciones y apoyo en todo momento, sabiéndome guiar por el camino de la sabiduría y el conocimiento.

A mis padres que nunca me dejaron solo y siempre puedo contar con ellos a pesar de las dificultades que se presentaron en el camino, sé que puedo contar con ellos incondicionalmente.

A mis hermanos que supieron apoyarme en todo a su alcance, enseñándome el significado del trabajo y la perseverancia, guiando por el camino de la humildad y la educación.

Bryan Alexander Amaguaña Amaguaña

AGRADECIMIENTO

Agradezco a Dios y a mi familia por brindarme todo su apoyo y confianza, enseñándome el valor de la perseverancia y el trabajo, cuidándome y guiándome en el transcurso de mi vida.

A la Cooperativa de transporte Urbano “Unión Ambateña” por permitir realizar el presente proyecto de investigación en su institución, brindándome todo el apoyo e información necesaria para la finalización del presente proyecto de investigación.

A la facultad de Ingeniería en Sistemas Electrónica e Industrial por abrirme sus puertas y brindarme los conocimientos necesarios para hacer de mi un gran profesional, así también, a todos los profesores que compartieron sus conocimientos y experiencias en el transcurso de toda una vida universitaria.

A mi tutor Ing. Mg. Hernando Buenaño por guiarme en todo el desarrollo del presente proyecto con el fin de culminar mi etapa universitaria con nuevos conocimientos y sabiduría.

Bryan Alexander Amaguaña Amaguaña

ÍNDICE DE CONTENIDOS

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA DEL TRABAJO	iii
DERECHOS DE AUTOR	v
APROBACIÓN DEL TRIBUNAL DE GRADO	iv
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE DE CONTENIDOS	viii
ÍNDICE DE TABLAS	xv
ÍNDICE DE FIGURAS.....	xxii
RESUMEN EJECUTIVO	xxvii
ABSTRACT.....	xxviii
INTRODUCCIÓN	xxix
CAPITULO I.....	1
MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.2 Antecedentes Investigativos	1
• Contextualización del problema.....	1
• Delimitación	2
• Justificación.....	3
1.3 Antecedentes Investigativos	4
1.4 Fundamentación teórica	5
1.4.1 Aplicaciones Móviles.....	5
1.4.1.1 Aplicación Nativa	5
1.4.1.2 Aplicaciones Web Móvil	5
1.4.1.3 Aplicaciones híbrida	6

1.4.1.4	Comparación entre aplicaciones móviles.	7
1.4.2	Framework de desarrollo móvil	7
1.4.2.1	Flutter.....	8
1.4.2.2	Ionic	9
1.4.2.3	Xamarin	10
1.4.2.4	React Native.....	10
1.4.3	Base de Datos NoSQL	11
1.4.4	Software libre	12
1.4.5	MongoDB.....	12
1.4.5.1	Mongo Atlas	12
1.4.6	API REST	13
1.4.7	Node.js	13
1.4.8	Express.js	13
1.4.9	React.....	14
1.4.10	Visual Studio Code	14
1.4.11	GitHub.....	15
1.4.12	Heroku.....	15
1.4.13	Comunicación inalámbrica.....	16
1.4.14	RFID.....	16
1.4.14.1	Etiquetas RFID	17
1.4.14.2	Tipos de Tags RFID.....	18
1.4.14.3	Lector RFID	19
1.4.15	Tarjetas MIFARE.....	20
1.4.16	Tecnología NFC	20
1.4.16.1	NDEF	22
1.4.16.2	Protocolo NFCIP.....	23
1.4.17	Arduino	23

1.4.17.1	Arduino Mega	23
1.4.18	Sistema informático basado en tarjetas NFC.	24
1.4.19	Cobro de Pasaje.....	25
1.4.20	Transporte Público	25
1.4.21	Servicios Públicos	25
1.5	Objetivos	26
1.5.1	General	26
1.5.2	Específicos	26
CAPITULO II		27
METODOLOGÍA		27
2.1	Materiales	27
2.2	Métodos	30
2.2.1	Modalidad de la investigación	30
2.2.2	Población y Muestra.....	30
2.2.3	Recolección de la Información	32
2.2.3.1	Resultados de las encuestas.	33
2.2.3.2	Resultados de la entrevista.....	43
2.2.4	Procesamiento y análisis de datos	45
CAPITULO III.....		46
RESULTADOS Y DISCUSIÓN		46
3.1	Análisis y discusión de resultados.....	46
3.1.1	Metodología de desarrollo ágil.....	46
3.1.2	Metodología Scrum.....	46
3.1.2.1	Roles en Scrum	47
3.1.2.2	Hitos de la metodología Scrum.....	47
3.1.2.3	Herramientas de la metodología Scrum.....	48
3.1.2.4	Ventajas de la metodología Scrum	48

3.1.2.5	Desventajas de la metodología Scrum	48
3.1.3	Metodología Kanban	49
3.1.3.1	Reglas de la metodología Kanban	49
3.1.3.2	Roles de la metodología Kanban	49
3.1.3.3	Tablero Kanban.....	50
3.1.3.4	Ventajas de la metodología Kanban.	50
3.1.3.5	Desventajas de la metodología Kanban	51
3.1.4	Metodología Xtreme Programing	51
3.1.4.1	Historial de Usuarios	51
3.1.4.2	Roles de la metodología Xtreme Programing.....	52
3.1.4.3	Procesos XP	52
3.1.4.4	Ventajas de la metodología XP.....	53
3.1.4.5	Desventajas de la metodología XP	53
3.1.5	Comparación de las metodologías Agiles	54
3.1.6	Metodología Elegida	54
3.1.7	Comparación de tecnología de desarrollo Front-End	55
3.1.8	Tecnología de desarrollo Front-End elegida	58
3.1.9	Comparación de tecnología de desarrollo Back-End.....	58
3.1.10	Tecnología de desarrollo Back-End elegida	62
3.1.11	Comparación de Base de datos	63
3.1.12	Base de datos elegida	65
3.1.13	Materiales elegidos para la lectura y recarga de tarjetas NFC	66
3.1.14	Análisis de los requerimientos de la aplicación.	66
3.2	Desarrollo de la propuesta	67
3.2.1	Fase I: Exploración	67
3.2.1.1	Arquitectura de la aplicación.	68
3.2.1.2	Proceso normal del cobro del pasaje.....	68

3.2.1.3	Proceso automatizado del cobro del pasaje	69
3.2.1.4	Definición de roles	70
3.2.2	Fase II: Planificación de Proyecto.....	70
3.2.3	Historias de usuarios	71
3.2.4	Estimación de historias de usuario	87
3.2.5	Roles internos del sistema.	89
3.2.6	Tareas	89
3.2.3	Fase III: Iteraciones.....	120
3.2.3.1	Plan de entregas	120
3.2.3.2	Iteraciones	123
3.2.4	Fase IV: Producción.....	126
3.2.4.1	Iteración I.....	126
3.2.4.2	Iteración II.....	137
3.2.4.3	Iteración III	141
3.2.4.4	Iteración IV	148
3.2.4.5	Iteración V	154
3.2.4.6	Iteración VI.....	163
3.2.5	Codificación	169
3.2.5.1	Conexión con la base de datos y puerto.....	169
3.2.5.2	Método para crear el Token en la aplicación web	170
3.2.5.3	Método para refrescar un token en la aplicación web	171
3.2.5.4	Método de verificación del token	171
3.2.5.5	Método de verificación de rol.....	172
3.2.5.6	Estructura básica de una colección	172
3.2.5.7	Configuración del servidor	173
3.2.5.8	Método Get para la colección Buses.....	174
3.2.5.9	Método Post para la colección Buses	175

3.2.5.10	Método Put para la colección Buses	175
3.2.5.11	Método Delete para la colección Buses	176
3.2.5.12	Método para filtrar cobros de pasajes por unidad de transporte .	177
3.2.5.13	Conexión de Front-End con el Back-End	178
3.2.5.14	Método para iniciar sesión.....	178
3.2.5.15	Método GET para obtener usuarios de la base de datos	179
3.2.5.16	Método para desplegar los datos de los usuarios en un componente 180	
3.2.5.17	Método para crear reportes en el cobro de pasajes	181
3.2.5.18	Método para elaborar el grafico de barras de cobros de pasaje ..	184
3.2.5.19	Método para lista los cobros de pasajes.....	185
3.2.5.20	Variables para la conexión de los servicios a la aplicación móvil 185	
3.2.5.21	Método para el ScreenLoading	186
3.2.5.22	Método para registrar un pasajero en la App	187
3.2.5.23	Método para desplegar los datos del pasajero en el Home	189
3.2.5.24	Método para desplegar las recargas realizadas en la App	191
3.2.5.25	Menú inferior de la App.....	192
3.2.5.26	Método para cerrar sesión.....	193
3.2.5.27	Método Recargas de Tarjetas NFC Arduino.....	193
3.2.5.28	Método para crear una tarjeta Arduino	196
3.2.5.29	Método para realizar un cobro de pasaje	198
3.2.5	Fase V: Pruebas.....	201
CAPÍTULO IV		215
CONCLUSIONES Y RECOMENDACIONES		215
4.1	Conclusiones	215
4.2	Recomendaciones	217

BIBLIOGRAFÍA	218
--------------------	-----

ÍNDICE DE TABLAS

Tabla 1 Comparación de tipos de aplicaciones	7
Tabla 2 Características Arduino Mega.....	24
Tabla 3 Materiales para el lector y recargador de tarjetas NFC.....	29
Tabla 4 Total de personas que utilizan una unidad de transporte en la ciudad de Ambato	31
Tabla 5 Recolección de Información	32
Tabla 6 Frecuencia de uso del transporte publico.....	33
Tabla 7 Conformidad con el método de pago actual.....	34
Tabla 8 Comodidad con el intercambio de dinero	35
Tabla 9 Aceptación de un nuevo método de pago	36
Tabla 10 Aceptación de la utilización de una tarjeta como forma de pago	37
Tabla 11 Aceptación de lugares asignados para recargas de tarjetas.....	38
Tabla 12 Comodidad del nuevo método de pago.....	39
Tabla 13 Aceptación de la aplicación móvil.....	40
Tabla 14 Aceptación de la modalidad del cobro del pasaje.....	41
Tabla 15 Conformidad con la nueva modalidad de pago del pasaje.....	42
Tabla 16 Comparación de mitologías ágiles	54
Tabla 17 Ventajas y Desventajas de Ionic	56
Tabla 18 Ventajas y desventajas de Flutter.....	57
Tabla 19 Ventajas y desventajas de React	58
Tabla 20 Ventajas y desventajas de Laravel	60
Tabla 21 Ventajas y desventajas de Firebase.....	61
Tabla 22 Ventajas y desventajas de Node js	61
Tabla 23 Ventajas y desventajas de Redis	64
Tabla 24 Ventajas y desventajas de Cassandra.....	65
Tabla 25 Ventajas y desventajas de MongoDB	65
Tabla 26 Definiciones de Roles XP	70
Tabla 27 Historia de usuario	71
Tabla 28 Historia 001 – Estructura del proyecto	72
Tabla 29 Historia 002 – Definición y modelo de la base de datos.....	73
Tabla 30 Historia 003 – Definición y modelo de servicios para la página web.....	73
Tabla 31 Historia 004 – Despliegue de la base de datos.....	73

Tabla 32 Historia 005 – despliegue de los servicios	74
Tabla 33 Historia 006 – Diseño del layout Admin	74
Tabla 34 Historia 007 – Pagina Inicio de sesión.....	75
Tabla 35 Historia 008 – Diseño del MenuSider.....	75
Tabla 36 Historia 009 – Diseño del Menutop	76
Tabla 37 Historia 010 – Diseño del modal y vistas del componente Usuario	76
Tabla 38 Historia 011 – Diseño del modal y vistas del componente rol.....	76
Tabla 39 Historia 012 – Diseño del modal y vistas del componente Persona	77
Tabla 40 Historia 013 – Diseño del modal y vistas del componente Buses	77
Tabla 41 Historia 014 – Diseño del modal y vistas del componente Tarjetas	78
Tabla 42 Historia 015 – Diseño del modal y vistas del componente Tipo_Pasajero .	78
Tabla 43 Historia 016 – Diseño del modal y vistas del componente Pasajero	79
Tabla 44 Historia 017 – Diseño del modal y vistas del componente Cobro_Pasaje..	79
Tabla 45 Historia 018 – Diseño del modal y vistas del componente Recargas	79
Tabla 46 Historia 019 – Diseño del modal y vistas del componente Menu_Web.....	80
Tabla 47 Historia 020 – Diseño del modal y vistas del componente Contenido_Web	80
Tabla 48 Historia 021 – Diseño de la página de inicio de Contenido_Web	81
Tabla 49 Historia 022 – Diseño página de inicio del sistema.....	81
Tabla 50 Historia 023 – Definición y modelo de los servicios para el dispositivo lector Tarjeta NFC.....	82
Tabla 51 Historia 024 – Definición y modelo de los servicios para el dispositivo recargar Tarjetas NFC	82
Tabla 52 Historia 025 – Diseño del primer prototipo lector Tarjetas NFC	83
Tabla 53 Historia 026 – Diseño del primer prototipo Recargas Tarjetas NFC.....	83
Tabla 54 Historia 027 – Pruebas de funcionalidad del dispositivo recargas y Cobros	84
Tabla 55 Historia 028 – Inicio de sesión en la aplicación UnionApp.....	84
Tabla 56 Historia 029 – Diseño del componente Register de la App	85
Tabla 57 Historia 030 – Diseño del menú de la App	85
Tabla 58 Historia 031 – Diseño del componente home para la App	85
Tabla 59 Historia 032 – Diseño del componente Pagos para la App.....	86
Tabla 60 Historia 033 – Diseño del componente Recargas para la App.....	86

Tabla 61 Historia 034 – Diseño del componente Mi cuenta para la App	87
Tabla 62 Historia 035 – pruebas de funcionalidad de la App UnionApp	87
Tabla 63 Estimación de historias de usuarios	88
Tabla 64 Plantilla de tareas de la historia de usuario	90
Tabla 65 Tarea 1- Estructura del proyecto.....	90
Tabla 66 Tarea 2- Definición de la herramienta a utilizar para la base de datos.	91
Tabla 67 Tarea 3- Diseño de la base de datos.....	91
Tabla 68 Tarea 4- Definición de la herramienta a utilizar para la base de datos.	91
Tabla 69 Tarea 5- Diseño de los servicios.	92
Tabla 70 Tarea 6- Análisis de herramientas para el despliegue de la base de datos..	92
Tabla 71 Tarea 7- Despliegue de la base de datos.	93
Tabla 72 Tarea 8- análisis de herramientas para el despliegue del Back-End.....	93
Tabla 73 Tarea 9- Despliegue del Back-End.	93
Tabla 74 Tarea 10- Análisis de herramientas para el desarrollo Front-End.	94
Tabla 75 Tarea 11- Diseño del layout Admin.....	94
Tabla 76 Tarea 12- Diseño del formulario login.....	95
Tabla 77 Tarea 13- Configuración del AccessToken y RefreshToken.	95
Tabla 78 Tarea 14- Implementación de servicios con el login.	95
Tabla 79 Tarea 15 – Diseño del MenuSider.	96
Tabla 80 Tarea 16 – Diseño del MenuTop.	96
Tabla 81 Tarea 17 – Diseño del modal y vistas para el componente Usuario.	97
Tabla 82 Tarea 18 – Diseño del modal Agregar y Editar Usuarios.	97
Tabla 83 Tarea 19 – Implementación de los servicios en el componente Usuarios. .	97
Tabla 84 Tarea 20 – Diseño de modal y vistas para el componente Rol.	98
Tabla 85 Tarea 21 – Diseño del modal Agregar Rol.	98
Tabla 86 Tarea 22 – Implementación de los servicios en el componente Rol.....	99
Tabla 87 Tarea 23 – Diseño del modal y vistas para el componente Persona.	99
Tabla 88 Tarea 24 – Diseño del modal agregar y editar Persona.....	99
Tabla 89 Tarea 25 – Implementación de los servicios en el componente Persona. .	100
Tabla 90 Tarea 26 – Diseño de modal y vistas para el componente Buses.	100
Tabla 91 Tarea 27 – Diseño del modal agregar y editar Buses.....	101
Tabla 92 Tarea 28 – Implementación de los servicios en el componente Buses.	101
Tabla 93 Tarea 29 – Diseño del modal y vistas para el componente Tarjetas.	101

Tabla 94 Tarea 30 – Diseño del modal agregar y editar Tarjetas.	102
Tabla 95 Tarea 31 – Implementación de los servicios en el componente Tarjetas..	102
Tabla 96 Tarea 32 – Diseño de modal y vistas para el componente Tipo_Pasaje. ..	103
Tabla 97 Tarea 33 – Diseño del modal agregar y editar Tipo_pasaje.....	103
Tabla 98 Tarea 34 – Implementación de los servicios en el componente Tipo_Pasaje.	103
Tabla 99 Tarea 35 – Diseño de modal y vistas para el componente Pasajero.	104
Tabla 100 Tarea 36 – Diseño del modal agregar y editar Pasajeros.	104
Tabla 101 Tarea 37 – Implementación de los servicios en el componente Pasajeros.	105
Tabla 102 Tarea 38 – Diseño de vistas para el componente Cobro_Pasaje.....	105
Tabla 103 Tarea 39 - Diseño buscador y reportes del componente Cobro_Pasaje..	105
Tabla 104 Tarea 40 - Implementación de los servicios en el componente Cobro_Pasaje.	106
Tabla 105 Tarea 41 - Diseño de vistas para el componente Recargas.	106
Tabla 106 Tarea 42 - Diseño del buscador y reportes del componente Recargas....	107
Tabla 107 Tarea 43 – Implementación de los servicios en el componente Recargas.	107
Tabla 108 Tarea 44 – Diseño de modal y vistas para el componente Menu_Web..	107
Tabla 109 Tarea 45 – Diseño del modal agregar y Editar Menu_Web.....	108
Tabla 110 Tarea 46 – Implementación de los servicios en el componente Menu_Web.	108
Tabla 111 Tarea 47 - Diseño del modal para el componente Contenido_Web.	109
Tabla 112 Tarea 48 – Implementación de los servicios en el componente Contenido_Web.....	109
Tabla 113 Tarea 49 – Diseño de rutas y vistas de la página de inicio.	109
Tabla 114 Tarea 50 – Implementación de servicios en la página de inicio del contenido web.	110
Tabla 115 Tarea 51 – Diseño de grafico de barras representando los cobros de pasajes.	110
Tabla 116 Tarea 52 - Diseño de grafico de barras representando las recargas de tarjetas.	111

Tabla 117 Tarea 53 – Diseño de servicios para el dispositivo Lector Tarjetas NFC.	111
Tabla 118 Tarea 54 - Diseño de servicios para el dispositivo Recargas Tarjetas NFC.	111
Tabla 119 Tarea 55 – Adquisición de materiales para el dispositivo Lector Tarjetas NFC.	112
Tabla 120 Tarea 56 – Diseño y codificación del dispositivo lector de tarjetas NFC.	112
Tabla 121 Tarea 57 - Adquisición de materiales para el dispositivo Recargar Tarjetas NFC.	113
Tabla 122 Tarea 58 - Diseño y codificación del dispositivo Recargar Tarjetas NFC.	113
Tabla 123 Tarea 59 – Realizar pruebas de funcionalidad de ambos dispositivos....	113
Tabla 124 Tarea 60 – Diseño del componente móvil Login.....	114
Tabla 125 Tarea 61 – Configuración del AccessToken del login móvil.....	114
Tabla 126 Tarea 62 – Implementación de los servicios para el componente móvil login.	115
Tabla 127 Tarea 63 – Diseño del componente móvil Register.	115
Tabla 128 Tarea 64 - – Implementación de los servicios para el componente móvil Register.	115
Tabla 129 Tarea 65 – Diseño del menú inferior para la navegación de la aplicación.	116
Tabla 130 Tarea 66 – Diseño del componente móvil Home.....	116
Tabla 131 Tarea 67 – Implementación de los servicios para el componente móvil home.....	117
Tabla 132 Tarea 68 – Diseño del componente móvil Pagos.....	117
Tabla 133 Tarea 69 – Implementación de los servicios para el componente móvil Pagos.	117
Tabla 134 Tarea 70 – Diseño del componente móvil Recargas.....	118
Tabla 135 Tarea 71 – Implementación de los servicios para el componente móvil Recargas.	118
Tabla 136 Tarea 72 – Diseño del componente móvil Mi Cuenta.	119
Tabla 137 Tarea 73 – Diseño de los sub-ítems del componente móvil Mi Cuenta.	119

Tabla 138 Tarea 74- Implementación de los servicios para el componente móvil Mi Cuenta	119
Tabla 139 Tarea 75 – Pruebas de funcionalidad de los dispositivos NFC con la aplicación web.....	120
Tabla 140 Tarea 76 - Pruebas de funcionalidad de la aplicación móvil con la aplicación web	120
Tabla 141 Plan de entrega de la aplicación	123
Tabla 142 Planificación de iteraciones	126
Tabla 143 Iteración I	126
Tabla 144 Iteración II.....	137
Tabla 145 Iteración III	141
Tabla 146 Iteración IV	148
Tabla 147 Iteración V.....	154
Tabla 148 Iteración VI	163
Tabla 149 Prueba de aceptación 01 – Despliegue de la base de datos.....	202
Tabla 150 Prueba de aceptación 02 – Despliegue de los servicios.....	202
Tabla 151 Prueba de aceptación 03 – Diseño del layout Admin	203
Tabla 152 Prueba de aceptación 04 – Página de inicio de sesión.....	203
Tabla 153 Prueba de aceptación 05 – Diseño del MenuSider.....	204
Tabla 154 Prueba de aceptación 09 – Diseño del MenuTop.....	204
Tabla 155 Prueba de aceptación 07 – Diseño del Modal y vistas del componente usuario.....	204
Tabla 156 Prueba de aceptación 08 – Diseño del Modal y vistas del componente rol	205
Tabla 157 Prueba de aceptación 09 – Diseño del Modal y vistas del componente persona	205
Tabla 158 Prueba de aceptación 10 – Diseño del Modal y vistas del componente Buses	206
Tabla 159 Prueba de aceptación 11 – Diseño del Modal y vistas del componente Tarjetas.....	206
Tabla 160 Prueba de aceptación 12 – Diseño del Modal y vistas del componente Tipo_Pasajero.....	207

Tabla 161 Prueba de aceptación 13 – Diseño del Modal y vistas del componente Pasajero	207
Tabla 162 Prueba de aceptación 14 – Diseño del Modal y vistas del componente Cobro_Pasaje	208
Tabla 163 Prueba de aceptación 15 – Diseño del Modal y vistas del componente Recargas	208
Tabla 164 Prueba de aceptación 16 – Diseño del Modal y vistas del componente Menu_Web.....	208
Tabla 165 Prueba de aceptación 17 – Diseño del Modal y vistas del componente Contenido_Web.....	209
Tabla 166 Prueba de aceptación 18 – Diseño de la página de inicio de Contenido_web	209
Tabla 167 Prueba de aceptación 19 – Diseño de la página de inicio del sistema	210
Tabla 168 Prueba de aceptación 20 – Diseño del primer prototipo lector de tarjetas NFC.....	210
Tabla 169 Prueba de aceptación 21 – Diseño del primer prototipo Recargas Tarjetas NFC.....	211
Tabla 170 Prueba de aceptación 22 – Inicio de sesión en la aplicación UnionApp.	211
Tabla 171 Prueba de aceptación 22 – Diseño del componente Register de la App.	212
Tabla 172 Prueba de aceptación 24 – Diseño del menú de la App	212
Tabla 173 Prueba de aceptación 25 – Diseño del componente home para la App ..	213
Tabla 174 Prueba de aceptación 26 – Diseño del componente Pagos para la App..	213
Tabla 175 Prueba de aceptación 27 – Diseño del componente Recargas para la App	213
Tabla 176 Prueba de aceptación 28 – Diseño del componente Mi cuenta para la App	214

ÍNDICE DE FIGURAS

Figura 1: Arquitectura General de Flutter	8
Figura 2: Arquitectura General de Ionic	9
Figura 3: Arquitectura General de Xamarin.....	10
Figura 4: Arquitectura General de React Native.....	11
Figura 5: Comunicación RFID.....	16
Figura 6: Esquema de una etiqueta RFID	17
Figura 7: Etiqueta Pasiva	18
Figura 8: Etiqueta Activa	18
Figura 9: Etiqueta Semi-activa o semi-Pasiva	19
Figura 10: Tarjeta MIFARE.....	20
Figura 11: Estructura de un mensaje NFC	22
.....	24
Figura 12: Frecuencia de uso del transporte público	34
Figura 13: Conformidad con el método de pago actual	35
Figura 14: Comodidad con el intercambio de dinero.....	36
Figura 15: Aceptación de un nuevo método de pago.....	37
Figura 16: Aceptación de la utilización de una tarjeta como forma de pago.....	38
Figura 17: Aceptación de lugares asignados para recargas de tarjetas	39
Figura 18: Comodidad del nuevo método de pago	40
Figura 19: Aceptación de la aplicación móvil.....	41
Figura 20: Aceptación de la modalidad del cobro del pasaje.....	42
Figura 21: Conformidad con la nueva modalidad de pago del pasaje	43
Figura 22: Tablero Kanban	50
Figura 23: Tecnologías Híbridas más populares	56
Figura 24: Tecnologías Back-End más populares.....	59
Figura 25: Comparación de Tecnologías Back-End más populares	62
Figura 26: Comparación de Base de Datos NoSQL	63
Figura 27: Arquitectura de la aplicación.....	68
Figura 27.1: Proceso manual del cobro de pasaje	69
Figura 27.2: Proceso automatizado del cobro de pasaje	69
Figura 28: Comando para la instalación de yarn.....	128
Figura 29: Pagina administrado de Mongo Atlas.....	128

Figura 30: Comando para la instalación de React.....	128
Figura 31: Comando para la instalación de Express	129
Figura 32: Página oficial para la instalación de Visual Code	129
Figura 33: Extensiones de Visual Studio Code.....	131
Figura 34: Comando para la creación del proyecto Unión	131
Figura 35: Comando para ejecutar un proyecto en React	131
Figura 36: Estructura del proyecto Unión.....	132
Figura 37: Modelo de la base de datos.....	133
Figura 38: Mongo Atlas desplegado la base de datos	135
Figura 39: Panel de Actividades de Heroku.....	136
Figura 40: Sentencia GET de la colección Tarjetas en postman.....	136
Figura 41: Layout Admin.....	137
Figura 42: Página de Inicio de sesión	138
Figura 43: MenuSider	139
Figura 44: MenuTop	139
Figura 45: Componente Usuario	140
Figura 46: Modal para editar un usuario	140
Figura 47: Componente Rol	141
Figura 48: Modal para agregar un rol	142
Figura 49: Componente Persona	142
Figura 50: Modal para agregar una Persona	143
Figura 51: Componente Buses	143
Figura 52: Modal para agregar un Bus.....	144
Figura 53: Componente Tarjetas.....	144
Figura 54: Modal para agregar una tarjeta	145
Figura 55: Componente Tipo_Pasajero.....	145
Figura 56: Modal para agregar un Tipo_Pasajero.....	146
Figura 57: Componente Pasajero	146
Figura 58: Modal para agregar un Pasajero	147
Figura 59: Componente Cobro_Pasaje	147
Figura 60: Componente recargas	148
Figura 61: Componente Menu_Web.....	149
Figura 62: Modal para agregar un menú.....	149

Figura 63: Componente Contenido_Web.....	150
Figura 64: Layout publico	151
Figura 65: Componente home grafico de ingresos de Buses	151
Figura 66: Reporte generado por el grafico de barras de los ingresos de los buses. 152	
Figura 67: Componente home grafico de ingresos de Recargas	152
Figura 68: Reporte generado por el grafico de barras de los ingresos de las recargas	153
Figura 69: Diseño del primer prototipo lector de tarjetas NFC	154
Figura 70: Diseño del primer prototipo lector y escritura de tarjetas NFC.....	155
Figura 71: Implementación de un teclado mecánico al dispositivo	156
Figura 72: Diseño de la implementación en un bus de transporte urbano	156
Figura 73: Implementación del lector de tarjetas NFC en una unidad de Transporte	157
Figura 74: Primer mensaje de aviso del dispositivo NFC.....	157
Figura 75: Proceso del cobro del pasaje.....	158
Figura 76: Proceso del cobro del pasaje Exitoso	158
Figura 77: Proceso del cobro del pasaje con una tarjeta no registrada	159
Figura 78: Proceso del cobro del pasaje con una tarjeta modificada ilegalmente ...	159
Figura 79: Mensaje de bloqueo de tarjeta	160
Figura 80: Dispositivo para recargas de tarjetas NFC	160
Figura 81: Dispositivo para recargas de tarjetas NFC en una unidad de transporte	161
Figura 82: Creación exitosa de una tarjeta.....	161
Figura 83: Crear una tarjeta ya existente.....	162
Figura 84: Login de la Aplicación móvil	162
Figura 85: Registro de la Aplicación móvil	163
Figura 86: Menú de la Aplicación móvil	164
Figura 87: componente home de la Aplicación móvil	164
Figura 88: componente Pagos de la Aplicación móvil.....	165
Figura 89: componente Recargas de la Aplicación móvil	166
Figura 90: Componente Mi Cuenta de la Aplicación móvil	167
Figura 91: Componente para actualizar la información del usuario	167
Figura 92: Componente para Bloquear una Tarjeta	168
Figura 93: Componente para cambiar de tarjeta	168

Figura 94: Componente que despliega la información de la empresa	169
Figura 95: Conexión con la base de datos y configuración del puerto	170
Figura 96: Método para crear el token en la aplicación web	170
Figura 97: Método refrescar un token en la aplicación web	171
Figura 98: Método de verificación del token	171
Figura 99: Método de verificación de rol.....	172
Figura 100: Estructura básica de una colección.....	173
Figura 101: Configuración del servidor	174
Figura 102: Método GET para la colección Buses	174
Figura 103: Método POST para la colección Buses	175
Figura 104: Método PUT para la colección Buses.....	176
Figura 105: Método Delete para la colección Buses.....	177
Figura 106: Método para filtrar los cobros de pasaje por unidad de transporte.....	178
Figura 107: Conexión del Front-End con el Back-End.....	178
Figura 108: Método para iniciar sesión.....	179
Figura 109: Método GET para obtener usuarios de la base datos.....	180
Figura 110: Método para desplegar los datos de un componente	180
Figura 111: Método para elaborar reportes en el cobro de pasajes.....	184
Figura 112: Método para elaborar el grafico de barra del cobro de pasaje.....	184
Figura 113: Método para listar los cobros de pasaje.....	185
Figura 114: Variables para la conexión de los servicios a la aplicación móvil	185
Figura 115: Método para el ScreenLoading.....	186
Figura 116: Método para registrar un pasajero en la App.....	189
Figura 117: Método para desplegar los datos en el componente home	191
Figura 118: Método para desplegar las recargas realizadas	192
Figura 119: Menú de la App	193
Figura 120: Método para cerrar sesión.....	193
Figura 121: Método recargas para Tarjetas NFC Arduino.....	196
Figura 122: Método para crear una tarjeta Arduino.....	198
Figura 123: Método para realizar un cobro de pasaje	201
Figura 124: Página principal	230
Figura 125: Página de inicio de sesión.....	231
Figura 126: Página principal del Admin.	232

Figura 127: Componente recargas de tarjetas del Admin	233
Figura 128: Componente Cobro Pasaje.	234
Figura 129: Componente Recargas de Tarjetas NFC.....	235
Figura 130: Componente tarjetas activas	236
Figura 131: Componente Agregar Tarjetas.....	237
Figura 132: Componente Editar Tarjetas	237
Figura 133: Componente tarjetas inactivas	238
Figura 134: Componente Tipo de Pasajes activo	239
Figura 135: Componente Bus	240
Figura 136: Componente Bus	241
Figura 137: Componente Personas	242
Figura 138: Componente Usuarios	243
Figura 139: Componente Rol	244
Figura 140: Componente Usuarios	245
Figura 141: Componente Contenido Web.....	246
Figura 142: Componente Login y registro de pasajero	247
Figura 143: Componente principal de la aplicación.	248
Figura 144: Componente pago y recarga de tarjeta	249
Figura 145: Componente mi cuanta	250
Figura 146: Componente editar datos personales y bloquear tarjeta	251
Figura 147: Componente cambiar de tarjeta y página de contacto	252
Figura 148: Dispositivo lector de tarjetas NFC	253
Figura 149: Dispositivo lector y escritor de tarjetas NFC.....	254
Figura 152: Registro de una nueva tarjeta.....	255
Figura 151: Registro de una nueva tarjeta.....	256
Figura 152: Mensaje de interacción con la tarjeta	257

RESUMEN EJECUTIVO

Con la llegada del Coronavirus Disease 2019 (COVID-19) se ha paralizado toda actividad presencial a nivel mundial, implementando múltiples medidas de bioseguridad, y a medida que se ha expandido la vacuna del virus, las actividades presenciales se van normalizando poco a poco, dando como resultado la exposición de múltiples personas a ser contagiados o transporta el virus.

De acuerdo con la OMS (Organización Mundial de la Salud) es recomendable mantener la distancia de al menos 1 metro entre personas, tanto en lugares abiertos y cerrados de tal forma que no exista contacto entre personas debido a que el virus se puede propagar en una corta distancia en el aire o a través del intercambio de objetos.

A partir del problema mencionado se plantea el desarrollo de la automatización del cobro del pasaje en la cooperativa de transporte urbano “Unión Ambateña” en la ciudad de Ambato, el cual consiste en eliminar todo contacto que posee un usuario conductor de una unidad de transporte con los usuarios pasajeros que toman el servicio para el traslado de un lugar a otro dentro de la ciudad.

El sistema permitirá automatizar el cobro del pasaje en una unidad de transporte urbano a través de tarjetas NFC que permite el intercambio de información sin realizar ningún contacto con el dispositivo de lectura, evitando el contacto entre el usuario conductor y el usuario pasajero.

El sistema constará de una aplicación web que permita la administración de los cobros realizados en cada unidad de transporte, también constará de dos dispositivos: lector y escritor de información que permita la interacción de las tarjetas NFC con la aplicación Web y finalmente se llevará un control a través de una aplicación móvil donde los usuarios pasajeros podrán visualizar las acciones que se realiza sus respectivas tarjetas en cada unidad de transporte.

Palabras Clave: REACT, Node .js, mongoDB, NFC, TypeScript, XP.

ABSTRACT

With the arrival of coronavirus disease 2019 (COVID-19) all face-to-face activity worldwide has been paralyzed, implementing multiple biosecurity measures, and as the virus vaccine has expanded, face-to-face activities are gradually normalized, resulting in the exposure of multiple people to be infected or transport the virus.

According to the WHO (World Health Organization) it is advisable to maintain the distance of at least 1 meter between people, both in open and closed places so that there is no contact between people because the virus can spread in a short distance in the air or through the exchange of objects.

Based on the aforementioned problem, the development of the automation of fare charging in the urban transport cooperative "Unión Ambateña" in the city of Ambato arises, which consists of eliminating all contact that a user driver of a transport unit has with the passenger users who take the service for the transfer of one place or another within the city.

The system will automate the collection of the ticket in an urban transport unit through NFC cards that allows the exchange of information without making any contact with the reading device, avoiding contact between the driver user and the passenger user.

The system will consist of a web application that allows the administration of the charges made in each transport unit, it will also consist of 2 information reader and writer devices that allow the interaction of the NFC cards with the Web application and finally will carry out a control through a mobile application where the passenger users will be able to visualize the actions that their respective cards are performed in each transport unit.

Keywords: REACT, Node .js, mongoDB, NFC, TypeScript, XP.

INTRODUCCIÓN

Como estructura del proyecto de investigación “SISTEMA INFORMÁTICO BASADO EN TARJETAS INTELIGENTES PARA AUTOMATIZAR EL COBRO DE PASAJES, DE LA COOPERATIVA DE TRANSPORTE URBANO ‘UNIÓN AMBATEÑA’ EN LA CIUDAD DE AMBATO.” Consta de cuatro capítulos que se detallan a continuación.

Capítulo I: “MARCO TEORICO”, detalla el problema que presenta la empresa, justificando la razón y la necesidad del porque se realiza la investigación, estableciendo objetivos principales y específicos que serán fundamentales en el desarrollo del presente trabajo de investigación.

Capítulo II: “METODOLOGIA”, detalla los materiales, modalidad de investigación aplicable para el proyecto de investigación, su respectiva recolección de información y procesamiento, estableciendo las etapas de desarrollo del proyecto.

Capítulo III: “RESULTADOS Y DISCUSION”, detalla el desarrollo del proyecto de acuerdo con la metodología elegida, de una manera clara y precisa.

Capítulo IV: “CONCLUSIONES Y RECOMENDACIONES”, Define las conclusiones obtenidas en el desarrollo del proyecto redactando las recomendaciones para un mejor desarrollo futuro.

CAPITULO I

MARCO TEÓRICO

1.1 Tema de investigación

Sistema informático basado en tarjetas inteligentes para automatizar el cobro de pasajes, de la cooperativa de transporte urbano “Unión Ambateña” en la ciudad de Ambato

1.2 Antecedentes Investigativos

- **Contextualización del problema**

Con la llegada del Coronavirus Disease 2019 (COVID-19) se ha establecido múltiples medidas de bioseguridad en el mundo, desde seguimientos de la propagación del virus en los diferentes continentes, hasta trabajo a distancia o aislamiento, de tal forma que se disminuya las aglomeraciones o contacto entre personas, reduciendo la pandemia.[1]

De acuerdo con la OMS (Organización Mundial de la Salud) para detener la propagación del virus es recomendable mantener al menos un metro de distancia entre personas, ya sea en lugares abiertos o cerrados, de esta forma disminuye la propagación del virus.[1]

Estas medidas de bioseguridad han implementado cambios en el comportamiento financiero de múltiples empresas a nivel mundial, lo que aceleró el proceso de digitalización en todo el mundo. Ecuador al enfrentar la paralización de actividades presenciales en medio de la pandemia, optó por el comercio electrónico, lo que generó un nuevo medio de subsistencia. [2]

En el país existe pocas provincias que exigen el pago con tarjetas de contacto o dinero electrónico dentro de las actividades diarias como es compra de productos, pagos de servicios básico, o pagos en el medio de transporte público, lo que dificulta el

distanciamiento entre personas, esto fue demostrado en la conferencia virtual “Consumo masivo y medios de pago” donde se demostró que un ecuatoriano en promedio realiza siete pagos electrónicos en el transcurso del año, pero a partir del mes de abril estas cifras subieron considerablemente debido a la pandemia. [3]

Tungurahua es una de las provincias con mayor riesgo de contagio, ubicado entre las diez provincias con mayor caso de contagios, y al pasar el tiempo se va normalizando las actividades diarias como el transporte público, lo que aumenta el riesgo aglomeraciones entre los usuarios que se disponen a utilizar este medio de transporte. [4]

En la ciudad de Ambato, múltiples propietarios de buses del transporte urbano no cuentan con las normas de seguridad, que eviten la propagación del virus, y a pesar de una incorporación de medidas de seguridad que aislé al conductor de los pasajeros que utilizan este tipo de transporte, aún existe el contacto entre conductor y usuarios que es por medio del pago del pasaje. [5]

Existe múltiples cooperativas que brindan este tipo de transporte, una de ellas es la cooperativa “Unión Ambateña” que ha paralizado sus actividades diarias por el cumplimiento del aislamiento, lo que ha producido un congelamiento total en sus ingresos, y no poseen un plan de seguridad con respecto al aislamiento de los señores conductores. Los usuarios propietarios de una unidad de transporte han optado por cuenta propia incorporar una cabina de aislamiento para el conductor, pero el proceso de cobro de pasaje sigue siendo el mismo, interacción entre conductor y usuario, lo que descarta el objetivo del aislamiento del conductor. [6]

- **Delimitación**

Línea de investigación: Desarrollo de Software.

Sublínea de investigación: Aplicaciones web.

Delimitación Espacial

Cooperativa de Transporte Urbano “Unión Ambateña” de la ciudad de Ambato.

Delimitación Temporal

La presente investigación se llevará a cabo en el periodo académico Abril – septiembre 2021.

- **Justificación**

La importancia de esta investigación se enfoca en un tema actual que es el distanciamiento social, para disminuir el número de contagios de COVID-19, y crear una zona segura para los señores conductores de la cooperativa “Unión Ambateña”, que prestan servicio a diario en la ciudad de Ambato.[7]

El aporte que brinda la automatización del cobro de pasaje al momento de subir a un medio de transporte es disminuir el contacto entre pasajero y transportista de tal forma que no se cree un medio de transferencia del virus asegurando el aislamiento de los conductores y evitando el posible contagio de los conductores a múltiples pasajeros.[7]

Al desarrollar un sistema que automatice el cobro del pasaje en el transporte urbano se cumplirá una parte del distanciamiento social, el cual es el contacto físico con personas, generando un número reducido de probabilidad de contagio. [7]

Los beneficiarios del sistema de cobro de pasajes son las personas que necesitan usar el transporte público en el cantón Ambato, así como los señores transportistas que ofrecen el servicio de transporte diariamente, donde se pretende disminuir con contacto entre personas para no propagar el virus. [7]

El presente proyecto no solo posee un alto interés en el cuidado de la salud de la ciudadanía, también abre múltiples implementaciones para mejorar el servicio de transporte urbano como es un conteo de personas que han ingresado a la unidad de transporte que posteriormente se puede dar múltiples utilidades financieras.[7]

1.3 Antecedentes Investigativos

El presente trabajo de investigación parte de investigaciones previas referentes al tema a tratar como es la tecnología NFC y la automatización del sistema de cobros de pasajes del transporte público, obtenidos de repositorios digitales.[7]

De acuerdo con la tesis realizada por Ochoa Ochoa Sergio Rafael y Pinos Rodríguez Cristhian Paul de título “Estudio de Factibilidad para la implementación de un sistema de cobro electrónico en las unidades de transporte urbano en la ciudad de Azogues” (2018). Menciona aspectos relevantes sobre el tema donde se determinó la ineficacia del cobro de pasaje sin la implementación de un cobro electrónico haciendo referencia al cobro manual, aprobando la automatización para el pago al subir a un medio de transporte, también sustenta su conclusión con pruebas de rendimiento basado en tiempos de cobro de pasajes y cantidades de pasajeros. [7] [8]

De acuerdo con la tesis de Bajaña Franco Rafael Ernesto de título “Estudio de factibilidad para la aplicación de tarjeta electrónica de transporte público, con la finalidad de mejorar el rendimiento administrativo, financiero y operativo de la cooperativa intraprovincial “Rutas milagreñas del cantón milagro” (2014). Afirma que la implementación de un método de pago electrónico usando tarjetas electrónicas, optimiza los procesos con los usuarios y mejora el cobro de la tarifa del pasaje además de ayudar al rendimiento administrativo de la empresa. [7] [9]

Según la tesis realizado por Yacelga Lita Inti Atahualpa de título “Sistema de recaudo en base a recargas de dinero mediante el uso de tecnología NFC (Near Field Communication) para la cooperativa de transporte “Flota Antaña” de la ciudad de Atuntaqui” (2016). Donde enfatiza que las tarjetas de contacto que utiliza la tecnología NFC son muy poco exploradas en el país, también argumenta que su propuesta es aceptada por los transportistas de esta cooperativa de acuerdo con las entrevistas realizadas. [7] [10]

1.4 Fundamentación teórica

1.4.1 Aplicaciones Móviles

Se denomina aplicación móvil al programa o software diseñado para dispositivos como celulares, tabletas, televisores, smartwatch, etc. Las aplicaciones móviles son diseñadas de acuerdo con las necesidades de los usuarios y a las limitaciones de los dispositivos para los que son creados, estas aplicaciones pueden ser de ámbito profesional, acceso a servicios, informativo o de entretenimiento. [11]

1.4.1.1 Aplicación Nativa

Aplicación nativa es aquella que es desarrollada con el software que ofrece cada sistema operativo a los desarrolladores o llamados también SDK (Software development kit). Windows Phone, Android y iOS poseen diferentes SDK y las aplicaciones nativas se programan y diseñan específicamente para cada plataforma en el propio lenguaje del SDK. [12]

Al ser aplicaciones propias de cada sistema operativo posee una mayor coherencia y consistencia comparada a otras aplicaciones, dando como resultado una mayor usabilidad y manteniendo los diseños del propio sistema operativo, de esta forma el usuario final encuentra una aplicación con interfaces familiares. [12]

Su principal ventaja es no requerir internet para poner en marcha su funcionalidad, dando como resultado una experiencia más fluida, sin mencionar que puede utilizar todas las características del hardware como sensores, acelerómetro, cámara, GPS (Global Positioning System), entre otros. [12]

1.4.1.2 Aplicaciones Web Móvil

Aplicaciones Web Móvil o también llamado webapps, son aplicaciones desarrolladas en HTML (HyperText Markup Language), adicionalmente con JavaScript y CSS (Cascading Style Sheets), lenguajes utilizados para desarrollar páginas web, al ser

aplicaciones web no es necesario ser instalado en el dispositivo, debido a que puede ser visualizado en un navegador, de la misma forma no puede ser distribuido en una tienda de aplicaciones. [12]

Para el desarrollo de aplicaciones Web Móviles no se emplea SDK, lo que permite programar independientemente del sistema operativo donde será ejecutado, lo que facilita la utilización en diferentes plataformas desarrollando en un mismo código toda la aplicación. [12]

La principal ventaja al ser una aplicación web, es no necesitar actualizaciones debido a que siempre estará utilizando la última versión. De la misma forma, requiere una conexión a internet para su correcto funcionamiento. [12]

1.4.1.3 Aplicaciones híbrida

Las aplicaciones híbridas son la combinación de aplicaciones nativas y aplicaciones web móvil debido a que su forma de desarrollar es muy similar al de una aplicación web (HTML, CSS y JavaScript), al terminar su desarrollo, es compilada y empaquetada de tal forma que su resultado es muy similar a una aplicación nativa. [12]

Las aplicaciones híbridas al no ser nativas no poseen un diseño que se identifique con el sistema operativo, pero posee controles y botones nativos de cada plataforma para diseñar una interfaz familiar para el usuario. [12]

La principal ventaja de las aplicaciones híbridas es su código, debido a que puede ejecutarse en diferentes plataformas utilizando el mismo código, para la distribución de la aplicación en las diferentes tiendas de cada plataforma, accediendo a librerías, y componentes del teléfono. [12]

1.4.1.4 Comparación entre aplicaciones móviles.

✓ : Bueno. ⊖ : Regular ✗ : Malo

	Nativa	Web	Híbrida
Lenguaje	.NET, C, Java	Html, Css, JavaScript	Html, Css, JavaScript
Interfaz de Usuario	✓	⊖	✓
Rendimiento	✓	✗	⊖
Costo de Desarrollo	✗	✓	⊖
Multiplataforma	✗	✓	✓
Tiempo de Desarrollo	✗	✓	⊖
App Store	✓	✗	✓
Mantenimiento	✗	✓	⊖
Experiencia de Usuario	✓	⊖	✓
Funcionalidad Offline	✓	✗	⊖
Seguridad	✓	⊖	⊖

Tabla 1 Comparación de tipos de aplicaciones

Elaborado por: Bryan Amaguaña

1.4.2 Framework de desarrollo móvil

Framework es un marco de referencia que ayuda a la programación, estructurando código, optimizando tiempo y esfuerzo a los programadores. El Framework facilita la organización de la estructura del código de una aplicación, gestionando las operaciones, datos e interfaces. [13]

Los Framework son utilizados para programar en más de un lenguaje de programación, lo que facilita la elaboración de aplicaciones sin cambiar de entorno de desarrollo, ofreciendo las herramientas necesarias para controlar y organizar el código en general. [13]

1.4.2.1 Flutter

Flutter, desarrollado por google como un framework de código abierto, publicado en el 2018, basado en lenguaje Dart con el fin de sustituir a JavaScript, también desarrollado por Google para la creación de aplicaciones nativas de una forma rápida, fácil y sencilla para los programadores, optimizando el tiempo de desarrollo para las diferentes plataformas.[14]

Su principal venta es la generación de código nativo para ser ejecutado en diferentes plataformas móviles de forma nativa, mejorando su rendimiento y creando interfaces idénticas a las aplicaciones nativas, de tal forma que pueden ser distribuidas en las App store.[14]

Su estructura de programación es orientada a objetos. Su interfaz está basada en widgets animados entre ellos, cada componente, texto o botón son widgets modificables de acuerdo con las necesidades de cada programador, reaccionando a cambios en la aplicación, o interactuando entre widgets, adicional a esto se puede crear widgets propios y combinarlos con los widgets existentes. [14]

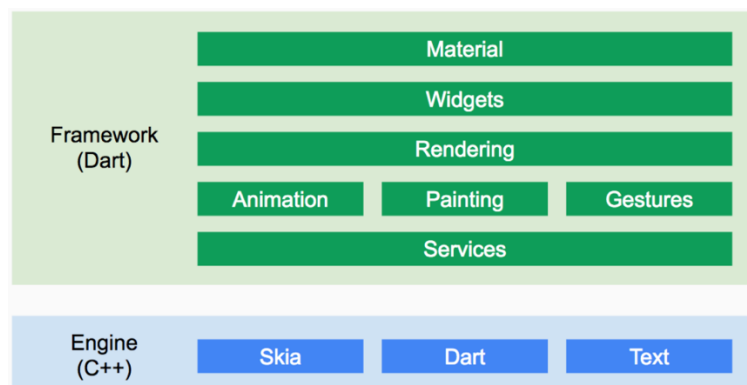


Figura 1: Arquitectura General de Flutter

Elaborado por: [15]

1.4.2.2 Ionic

Ionic Framework es un SDK diseñado para Front-End de código abierto para desarrollar aplicaciones híbridas, ejecutables en iOS nativo, Android y web desde un solo código, basado en tecnología web (HTML, CSS, JavaScript). Con la implementación de Cordova e Ionic Native es posible desarrollar con componentes híbridos y diseñar interfaces amigables con el sistema operativo donde será ejecuto la aplicación. [16]

Fue creado en el 2013 por Max Lynch, Ben Sperry, y Adam Bradley de Drifty basado en Angular JS y Cordova, con la intención de que desarrolladores crearan aplicaciones móviles híbridas sin preocuparse de distintos SDK para los que fueron creados. [17]

Diseñado bajo una base de tecnología web estables y API estandarizadas de alto rendimiento y fluidez, perfecto para desarrollar aplicaciones para todo tipo de dispositivos independientemente de las actividades que puede realizar, con un diseño responsive limpio y entendible para todo tipo de programador. [18]



Figura 2: Arquitectura General de Ionic

Elaborado por:[19]

1.4.2.3 Xamarin

Xamarin, diseñado para la creación de aplicaciones de diferentes plataformas como iOS, Windows y Android basado en .NET y reutilizando más del 90% de código en cada plataforma, fue presentado en el 2011 hasta el 2016 donde formo parte de Microsoft. [20]

Su principal objetivo es facilitar la optimización de código a tal punto de crear aplicaciones iOS y Android en un mismo lenguaje como es C#, Xamarin nos permite unificar estas diferencias de lenguaje y unirlos en Xamarin Studio para ser programado en C# para la creación de aplicaciones multiplataforma de forma nativa. [20]

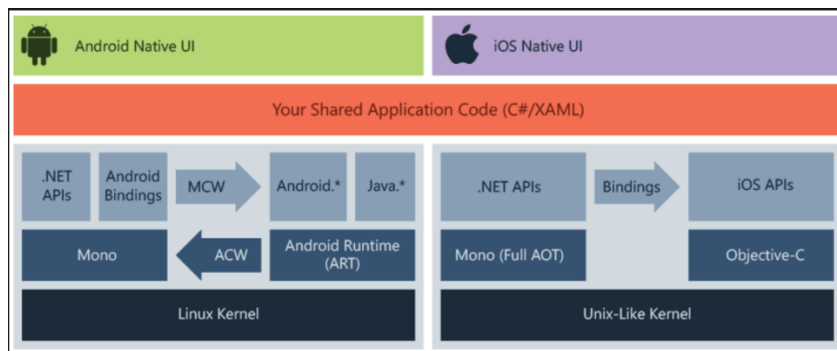


Figura 3: Arquitectura General de Xamarin

Elaborado por: [20]

1.4.2.4 React Native

React Native fue creado por Jordan Walke, Ingeniero de software en Facebook, permite la creación de aplicaciones nativas utilizando JavaScript, para luego convertirlos en código nativo ofreciendo una optimización considerable. [21]

Su principal cualidad es el Cross-Platform, lo que permite escribir el código en JavaScript y React Native lo ejecuta en iOS y Android de forma nativa sin implementaciones de funcionalidades de iOS o Android, adicional a esto los módulos creados en React Native pueden ser implementados en aplicaciones ya existentes. [21]

La comunicación de los diferentes hilos se los realiza a través del bridge de React Native lo que permite la comunicación en JavaScript con la parte nativa, para la transferencia de información o la funcionalidad de cualquier componente del dispositivo. De esta forma obtenemos un rendimiento de ejecución bastante considerable.[22]



Figura 4: Arquitectura General de React Native

Elaborado por: [22]

1.4.3 Base de Datos NoSQL

NoSQL (Not only SQL) hace referencia a un sistema de gestión de datos que no está vinculado con el modelo RDBMS (Relational Database Management System) o sistema de gestión de base de datos relacionales, y que no tiene necesidad de aplicar un lenguaje SQL para realizar consultas o procedimientos. [23]

Se originó en el año 1998 creado por un grupo de empresas que buscaban solucionar las necesidades que el modelo RDBMS carecía como es la escalabilidad, flexibilidad, alta disponibilidad y el rendimiento. [23]

Las bases de Datos NoSQL están diseñadas para entornos distribuidos de tal forma que no dificulte la tarea de agregar más nodos o instancias, debido a que no están atados a un esquema predefinido ofreciendo más flexibilidad, y escalabilidad permitiendo su ejecución en caso de originarse un error de software o hardware, permitiendo que su tiempo de respuesta sea superior al modelo RDBMS. [23]

1.4.4 Software libre

Se denomina software libre a la representación de libertad de un programa o software hacia los usuarios y la comunidad, manteniendo la ideología de que los usuarios pueden mejorar, modificar, copiar, distribuir o ejecutar un software gratuitamente, de tal forma que los usuarios pueden contralar su código y realizar pruebas libremente. [24]

1.4.5 MongoDB

MongoDB (Del inglés humongous: enorme) orientado a documentos de código abierto y desarrollado en C++, es un sistema de base de datos NoSQL, que a diferencia de base de datos SQL que almacena los datos en tablas, MongoDB guarda sus datos con esquemas dinámicos en estructura BSON (muy similar a JSON). [25]

MongoDB ofrece consultas ad hoc lo que permite realizar todo tipo de consultas, búsquedas por campo, expresiones regulares y consultas de rangos, lo que permite obtener un campo específico de la información, adicional a esto MongoDB permite escalar la carga de trabajo de manera simultánea en múltiples servidores, de tal forma que el sistema permanece conectado a la base de datos todo el tiempo. [25]

1.4.5.1 Mongo Atlas

Mongo Atlas es un servidor en la nube global gratuito, administrado por AWS, Azure y GCP, permitiendo administrar nuestra base de datos desde un navegador, desechando la idea de instalar en una computadora, y agilizando el trabajo para los programadores. Permite la distribución de sus datos en múltiples clústeres facilitando la accesibilidad de sus datos en una o más nubes al mismo tiempo. [26]

1.4.6 API REST

Un API de transferencia de estado representacional (REST) o API REST, fue creado por Roy Fielding como una interfaz de programación de aplicaciones el cual se ajusta a la arquitectura REST permitiendo la interacción con los servicios web. [27]

Se considera Api al conjunto de protocolos y definiciones utilizados para integrar y desarrollar un software. Cuando se solicita información a través de una API de REST full la información se entrega por medio de HTTP (Hypertext Transfer Protocol) en formato: HTML (HyperText Markup Language), XLT, PHP, Python o texto sin formato. [27]

1.4.7 Node.js

Node.js está diseñado para la creación de aplicaciones network escalables como un entorno para ejecutar JavaScript con eventos asíncronos. Desarrollado por programadores de JavaScript, paso por muchos cambios desde su ejecución, estaba limitado a ser ejecutado en un navegador, pero con el paso del tiempo se ha podido ejecutar en un ordenador como un programa independiente. [28]

Node.js utiliza el motor JavaScript V8 (toma el código JavaScript y lo ejecuta mientras utilizamos nuestro navegador), la funcionalidad de JavaScript V8 es convertir el código JavaScript en código de máquina que un código de bajo nivel aumentado la velocidad de ejecución. [29]

1.4.8 Express.js

Express.js es un Framework de desarrollo de aplicaciones web minimalistas para node.js, permite crear aplicaciones web y APIs con mayor facilidad, provee de manejo de rutas(direccionamiento), uso de motor de plantillas, manejo de errores, integración de base de datos, middlewares etc. [30]

Posee una variedad de métodos para ser utilizados en la programación de utilidad HTTP, así también es considerado como su fuerte la programación de APIs y posee middleware a su disposición, proporcionando una delgada capa de particularidades de aplicación web básicas. [31]

1.4.9 React

React es desarrollada por Facebook como una biblioteca de código abierto escrita en JavaScript, fue diseñada para facilitar la creación de componentes reutilizables e interactivos para diseños de interfaces de usuario, también puede ser programado como servidor de tal forma que pueda trabajar conjuntamente cliente y servidor. [32]

React.js posee un algoritmo que distingue una presentación virtual de una página actual de tal forma que realiza un conjunto mínimo de cambios en el DOM (Modelo de objeto de documento), evitando recargar una página completa y solo actualizando el componente actualizado. [32]

1.4.10 Visual Studio Code

Visual Studio Code fue lanzado en el 2015 por Microsoft para diferentes sistemas operativos como macOS, Microsoft y Linux como un editor de código fuente, compatible con diferentes lenguajes de programación como TypeScript, Node.js y JavaScript, adicional a dichos lenguajes posee extensiones para múltiples lenguajes como Python, Java, C#, C++, Go y PHP. [33]

Visual Studio Code implementa un framework que se utiliza para Node.js y Chromium (Navegador web de código abierto desarrollado por google) llamado Electron (Framework de código abierto), de tal forma que sean ejecutados como aplicaciones de escritorio utilizando el motor Blink (Motor de renderizado desarrollado por google para navegadores). [33]

1.4.11 GitHub

Git tuvo su inicio en el 2005 como un proyecto de código abierto, tratándose de un sistema controlador de versiones distribuidas, donde los desarrolladores alojan los repositorios de código brindando herramientas y fomentando el trabajo en equipo. [34]

Git además de ofrecer herramientas de control de versiones posee ramas de cambios, lo que permite que miembros del equipo puedan realizar cambios en el proyecto sin afectar al proyecto original, de tal forma que, si los cambios no son aceptados por el equipo de trabajo, se pueda regresar a la versión original del proyecto. [35]

GitHub es una plataforma de organización y gestión de proyectos almacenados en la nube que incorpora múltiples funciones de Git, una de ellas en el controlador de versiones. Dispone de una interfaz de usuarios fácil de entender lo que hace accesible a todo tipo de persona, mejorando el trabajo en equipo y la optimización de tiempo. [34]

1.4.12 Heroku

Heroku conocido como plataforma as a Service o PaaS es una empresa que ofrece servicios de plataforma en la nube, ofreciendo servicios de redes administrativas y servidores, donde se puede alojar aplicaciones de distintos lenguajes de programación como Python, PHP, Java, entre otros. [36]

Fue fundada en el 2007 hasta el 2010 donde fue adquirida por salesforce.com, diseñada para desarrolladores, pequeñas empresas y startups, utilizando contenedores Linux (dynos) para el alojamiento de aplicaciones permitiendo instalar add-ons para agregar distintas funcionalidades a estos contenedores como administrador de base de datos, mensajero o almacenamiento en la nube. [37]

1.4.13 Comunicación inalámbrica

La comunicación Inalámbrica es un sistema eléctrico que no utiliza alambres conductores, la comunicación se realiza sin cables que interconecten físicamente los equipos o dispositivos, conectando diversos nodos y permitiendo la comunicación entre dispositivos mediante ondas electromagnéticas, La recepción y la transmisión de la información entre dispositivos requiera que actúen como puertos. [38]

1.4.14 RFID

RFID (Radio Frequency Identification) es la comunicación inalámbrica entre un lector y un emisor utilizando ondas de radio de tal forma que el receptor envía señales continuas manteniendo el alcance concreto de lectura, hasta entrar en contacto con el emisor que debe incorporar un microchip y una antena de radio reducida para enviar información, este microchip está incorporados por etiquetas o tags RFID. [39]

Las etiquetas o tags RFID poseen una capacidad de Almacenamiento desde 1 bit hasta 8 KB (Kilobytes) este valor varía dependiendo del sistema acumulador de la tarjeta RFID, la información guardada en la tarjeta RFID debe ser detectado por un lector especial para las tarjetas RFID, que debe poseer una o varias antenas que emiten ondas de radio y receptan señales devueltas por las tarjetas RFID. [40]

Esta tecnología es utilizada para detectar cuando un objeto esta fuera del rango permitido, evitando robos o pérdidas, sustituyendo al código de barras debido a que posee un mayor alcance y sus datos son leídos con mayor rapidez. [39]

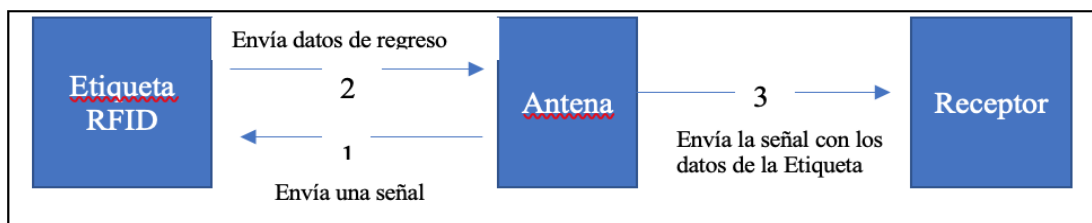


Figura 5: Comunicación RFID

Elaborado por: Bryan Amaguaña

1.4.14.1 Etiquetas RFID

Una etiqueta RFID está compuesta por diferentes materias como el sustrato (Material del cual este fabricado la etiqueta), Tipo IC (Circuito Integrado), antena (Permite enviar datos al lector). [41]

Material/Sustrato (Plástico, cartón, metal etc.), el componente de fabricación de la etiqueta RFID depende del ambiente donde se establecerá su uso, tomando en cuenta que la densidad el material también afecta la intensidad del distanciamiento del lector RFID. [42]

IC (Circuito integrado), compuesto por la parte lógica de la etiqueta RFID, donde se establece la memoria y la seguridad de la información que será gravado en la tarjeta. [42]

Antena, suministra la energía para la etiqueta (en algunos casos), permitiendo captar las señales enviadas por el lector dando como resultado una respuesta por parte del IC. [40]

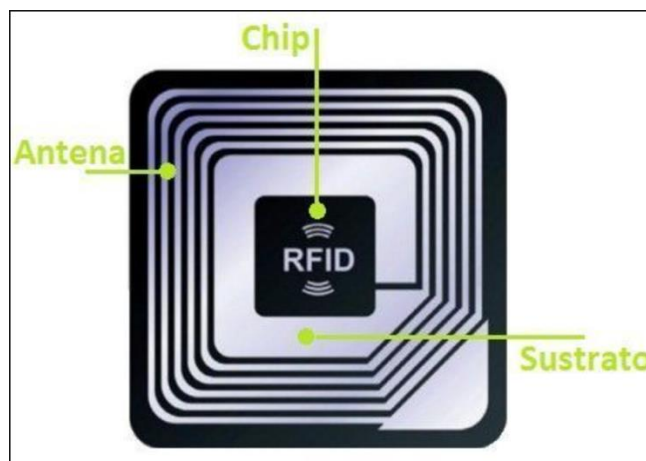


Figura 6: Esquema de una etiqueta RFID

Elaborado por: [43]

1.4.14.2 Tipos de Tags RFID

Etiquetas Pasivas.

Extraen energía del receptor, por lo que no lleva una batería interna, de esta forma el receptor envía energía que activa al emisor de esta forma transfiere la información requerida. Este tipo de etiquetas posee recursos limitados y una baja capacidad de almacenamiento por lo que es más utilizado como etiquetas de solo lectura, pero al poseer un bajo costo comercial es también una de las utilizadas.[40]

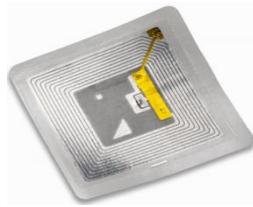


Figura 7: Etiqueta Pasiva

Elaborado por: [42]

Etiquetas Activas.

Poseen una fuente de alimentación interna aumentado su memoria y su tamaño, lo que permite enviar información a una distancia de 100 metros, son utilizados como etiquetas de lectura y escritura en diferentes ámbitos, pero su costo elevado limita su uso dando prioridad a etiquetas pasivas. [40]



Figura 8: Etiqueta Activa

Elabora por: [44]

Etiquetas semi-Activas o semi-Pasivas

Poseen una batería interna que alimenta al chip de la etiqueta, pero al igual que las etiquetas pasivas extraen energía del receptor para generar comunicación, aunque poseen una mayor velocidad de procesamiento y de respuesta. [40]



Figura 9: Etiqueta Semi-activa o semi-Pasiva

Elaborado por: [45]

1.4.14.3 Lector RFID

Lector RFID o interrogador es un dispositivo que crea campos magnéticos de corto alcance u ondas de radio de alta frecuencia, con el fin de contactar a un tag de RFID para recibir información y transmisión y convertirlo en formato legible para un ordenador. [46]

Al contactar a un tag RFID se crea un acoplamiento e intercambian información, dependiendo de las funciones que se le asignaron al lector RFID puede leer y sobrescribir información en el tag RFID, en un corto periodo de tiempo sin necesidad de tener contacto entre ambos dispositivos. [46]

Los lectores RFID más sofisticados pueden leer varias RFID al mismo tiempo, desarrollando un nuevo método anticolidión, asignando horarios o frecuencias de acceso evitando que las señales se colapsen, agilizando el registro o lectura de información de tag RFID. [46]

1.4.15 Tarjetas MIFARE

Son Tarjetas de Memoria protegida de no contacto, divididas en bloques, sectores y mecanismos simples para el control de acceso, a pesar de tener un alto nivel de seguridad no pueden realizar operaciones criptográficas debido a que están destinadas o control de accesos, transporte urbano, tarjetas de presentación corporativas, etc. [47]

Cada sector posee 4 bloques, el primer bloque posee información de la tarjeta (Código único), los restantes 3 bloques son para el almacenamiento de información por parte del usuario. [48]

Se establece un canal de cifrado en la tarjeta MIFARE para él envío del código de identificación con el receptor de tal forma que tenga acceso a los diferentes sectores de la tarjeta para posteriormente poder leerlos o modificarlos. [49]



Figura 10: Tarjeta MIFARE

Elaborado por: [50]

1.4.16 Tecnología NFC

La Tecnología NFC (Near Field Communication) es un sistema de comunicación inalámbrico que funciona en la banda de los 13,56 MHz caracterizado por ser de corto alcance, también es conocido como protocolo wireless en bidireccional de corto alcance, esta tecnología está integrado a Smartphone tarjetas, llaveros entre otros objetos que permite el intercambio de información entre dispositivos de manera

cómoda y eficaz. No está diseñado para transferir información de gran tamaño o información muy larga. [51]

La principal diferencia entre el sistema RFID y la tecnología NFC es el rango de alcance a diferencia de RFID que permite el intercambio de información de una distancia entre 1 a 4 cm, NFC intercambia información a 1 cm de distancia, lo cual puede ser una gran desventaja con respecto a distanciamiento, pero al tener un menor rango de distancia es menos propenso a tener fallos de comunicación convirtiendo la desventaja en ventaja. [52]

Los dispositivos NFC tienen una comunicación bidireccional, lo que permite que interactúen entre si enviando y recibiendo información. El protocolo NFC explica la comunicación entre dispositivos, de la misma forma el dispositivo que inicia la comunicación monitorea el comportamiento el otro dispositivo, este comportamiento puede tocar el otro dispositivo igualmente. En los dispositivos NFC de estándar NFCIP-1 existen 2 modos de comunicación que son el modo pasivo y activo. [40]

Modo Activo

Se requiere que los 2 dispositivos sean independientes, disponiendo de su propio campo electromagnético para la transferencia de datos, caracterizado por la comunicación Peer to Peer (igual a igual) entre dispositivos NFC. [42]

Modo Pasivo

Solo es requerido que un dispositivo genere el campo electromagnético para la comunicación y el intercambio de información, el dispositivo pasivo aprovecha el campo electromagnético para activar el chip y enviar los datos al dispositivo que inicio la comunicación. [42]

1.4.16.1 NDEF

NDEF o también conocido como NFC Data Exchange Format registrado por el NFC Forum, es un formato de compartición de datos entre dispositivos NFC tanto en pasivos como en activos, este formato permite la lectura o escritura de información en los tags RFID. Las etiquetas que son leídas o escritas en este formato se les conocen como etiquetas NDEF o NFC. [42]

NDEF crea la idea de organizar los datos y almacenados en bytes para el intercambio de información, su principal característica es la utilización de una cabecera NDEF de datos, donde se encuentra los bloques de datos, cada bloque de datos está unidos en registros que contienen mensajes NDEF que están caracterizados por un MIME (MUltipurpose Internet Mail Extensions) predeterminado. [42]

El formato NDEF posee una clave de acceso a la información que es por defecto (en hexadecimal FF FF FF FF FF FF), muy común en todo los dispositivos NFC lo que hace vulnerable a que otros dispositivos tengan acceso a los bloque de información, esta desventaja hace que una etiqueta RFID pueda ser sobrescrito o borrado la información en cualquier momento por otro dispositivo no permitido, pero este problema puede ser resuelto combinando las ventajas de NDEF y MIFARE donde se establecería una clave de acceso. [40]

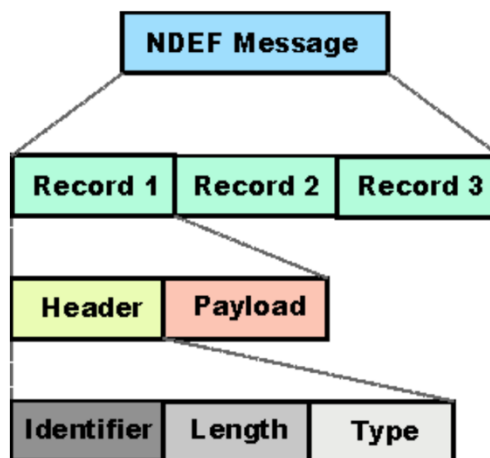


Figura 11: Estructura de un mensaje NFC

Elaborado por: [53]

1.4.16.2 Protocolo NFCIP

NFCIP (Near Field Communication Interface and Protocol) se define como el protocolo de interfaz inalámbrica entre 2 entidades, donde el dispositivo que inicia la comunicación es el encargado de vigilar el comportamiento, este rol puede cambiar de dispositivo, operando en una banda de 13,56 MHz con un alcance de 20 cm. [40]

1.4.17 Arduino

Arduino se define como una plataforma de desarrollo basado en hardware libre incorporada por una placa electrónica con microcontroladores reprogramables y múltiples entradas de pines hembra para la conexión de diversos dispositivos que actúan de una forma sencilla. [54]

1.4.17.1 Arduino Mega

Considerado como el Arduino más potente de toda la gama de microprocesadores Arduino mega posee 54 pines que cumplen con la función de entrada y salida de información, posee una conexión USB, una entrada de alimentación de la placa Arduino y 16 entradas analógicas. [55]

Para su correcto funcionamiento el Arduino mega posee una entrada USB para la alimentación, adicional a esto posee una entrada de alimentación especial para una fuente externa de energía, que obligatoriamente deber se implantado con un convertidos que regular el voltaje de 7v hasta los 12v. [55]

Arduino Mega posee 256 kb de memoria Flash para almacenar código de programas sketches, de esta capacidad de memoria 8 kb es utilizado para el gestor de arranque y kb son utilizados para el EEPROM (ROM programable) y 8 kb para SRAM (Static Random Access Memory). [55]

Características	
Microcontrolador:	ATmega2560
Voltaje Operativo:	5V
Voltaje de Entrada:	7-12V
Voltaje de Entrada(límites):	6-20V
Pines digitales de Entrada/Salida:	54 (de los cuales 15 proveen salida PWM)
Pines análogos de entrada:	16
Corriente DC por cada Pin Entrada/Salida:	40 mA
Corriente DC entregada en el Pin 3.3V:	50 mA
Memoria Flash:	256 KB (8KB usados por el bootloader)
SRAM:	8KB
EEPROM:	4KB
Clock Speed:	16 MHz

Tabla 2 Características Arduino Mega

Elaborado por: [55]

1.4.18 Sistema informático basado en tarjetas NFC.

Las tarjetas NFC o etiquetas son tarjetas de plástico similares a tarjetas de crédito integrado con un chip RFID que permite la comunicación entre dispositivos utilizando un protocolo NFCIP, de ahí su nombre tarjeta NFC, el cual permite realizar tareas de comunicación (almacenar, escribir o leer datos). [56]

Los sistemas que hace uso de las tarjetas inteligentes básicamente son para el intercambio de información, acceso, lectura a datos, identificación etc. Para que un sistema tenga acceso a dicha información debe poseer un lector de tarjetas NFC. [56]

1.4.19 Cobro de Pasaje

Es el cobro de un servicio de transporte previamente establecido y controlado por una entidad reguladora. En la cooperativa de transporte urbano Unión Ambateña se rige a las leyes ya impuestas desde años atrás donde el pasaje está definido a 0.30 centavos de dólar para los adultos y 0.15 centavos de dólar para los estudiantes, niños y tercera edad, de tal forma que no pueden ser manipulados ni expuestos a nuevas tarifas en los cobros de los pasajes. [7]

1.4.20 Transporte Público

El transporte público es un sistema de transportación que se desplazan en rutas fijas en determinados horarios para el servicio de todas las personas a cambio de un determinado pago de una tarifa previamente establecida y aceptada por las personas. [7] [57]

1.4.21 Servicios Públicos

Los servicios públicos son recursos humanos calificados que dan servicios personales a la administración central, institucional o seccional para satisfacer necesidades generales. Los servidores públicos llamados también “agentes públicos” se los ha llegado a identificar como “funcionarios” y “empleados”. [7] [58]

1.5 Objetivos

1.5.1 General

Implementar un sistema informático basado en tarjetas inteligentes para automatizar el cobro de pasajes de la Cooperativa de Transporte Urbano “Unión Ambateña” en la ciudad de Ambato.

1.5.2 Específicos

- Establecer un método de automatización de cobro de dinero de manera rápida y eficaz.
- Investigar tecnologías que implemente el uso de tarjetas inteligentes para el pago de pasajes.
- Desarrollar una aplicación que automatice el cobro de pasajes en la Cooperativa de Transporte Urbano “Unión Ambateña” en la ciudad de Ambato.




CAPITULO II

METODOLOGÍA

2.1 Materiales

Para el desarrollo del presente proyecto de investigación se optó por adquirir información de libros, artículos científicos, revistas y tesis, como fuentes digitales, de igual forma se realizó investigaciones previas y cursos para el diseño y desarrollo de la página web que gestionara las actividades y cambios que realicen los usuarios.

Para la elaboración del lector y recargador de tarjetas NFC se utilizó diferentes materiales electrónicos que ayudaran en la implementación del prototipo, detallados en la **Tabla 3**, con sus respectivas definiciones.

Nombre	Grafico	Definición
Arduino Mega		Se define como una placa de desarrollo llamado así por su microcontrolador ATmega2560. [55]
Cable Macho Hembra 20cm		Son cables de soldar que permiten cerrar circuitos formando una conexión, poseen entrada y salida.
Cable Macho Macho 20cm		Son cables de soldar que permiten cerrar circuitos formando una conexión Poseen solo salida.

<p>Adaptador 9v 1A</p>		<p>Convierte la alimentación externa en 9v y consume hasta 1 amperio.</p>
<p>Turnigy 2200 mAh 3S 30C</p>		<p>Batería recargable de 2200 mAh equipada con cables de descarga de alta resistencia.</p>
<p>Cargador lipo b3</p>		<p>Cargador de baterías Turnigy, equipado con 2 y 3 celdas Lipo de carga fluida.</p>
<p>Modulo Fuente Step Down LM2596</p>		<p>Regula la cantidad de voltaje entrante obteniendo un voltaje de salida inferior y constante.</p>
<p>Conector Plug CCTV hembra</p>		<p>Conector DC hembra, permite el ingreso de alimentación de la batería Turnigy.</p>
<p>Screw Proto Shield</p>		<p>Facilita la conexión de componentes a una placa Arduino, especializada para implementaciones.</p>
<p>Pantalla LCD 20x4</p>		<p>Pantalla LCD de 20 caracteres y 4 filas, con retro iluminación.</p>






Teclado 4x4 Matricial		Teclado matricial de plástico compuesto de 16 teclas.
Módulo NFC Rc522		Lectura-Escritura RFID de 13.56 MHz con comunicación SPI.
Módulo SIM800C		Dispositivo quad-band GSM/GPRS, permite realizar llamadas mensajes o conexión a internet desde el dispositivo Arduino.
Chip Claro		Tarjeta SIM de uso obligatorio para acceder a las redes GSM.
Módulo I2C		Facilita la utilización de pantallas LCD reduciendo las conexiones a 4 entradas

Tabla 3 Materiales para el lector y recargador de tarjetas NFC

Elaborado por: Bryan Amaguaña

2.2 Métodos

2.2.1 Modalidad de la investigación

Investigación Bibliográfica

La investigación bibliográfica posee un aporte fundamental en lo referente a fundamentación teórica, tomando como punto de referencia tesis, proyectos, investigaciones y artículos, de tal forma que se profundice los temas y subtemas relacionados a la automatización de pagos con el uso de tarjetas NFC ampliando los conocimientos del proyecto.[7]

Investigación de campo

Mediante la aplicación de las investigaciones se recopilará información y datos para analizar los problemas que se presenten en la realización del proyecto, así también se pretende solucionar y mejorar la estructura de la investigación, de tal forma que cumpla con sus requisitos de funcionamiento.[7]

2.2.2 Población y Muestra

Población

La cooperativa de transporte Unión Ambateña dispone de contadores en cada unidad de transporte y cada usuario tiene acceso a esta información mediante un aplicativo que refleja el total de usuarios que utilizan este medio de transporte, por lo que se consideró promedio de 2626 usuarios a la semana, esta cifra es determinada mediante el conteo de 7 días en cada línea de trabajo que realiza una unidad de transporte. [7]

La población se determinará mediante el cálculo de un 1 día laborable, del promedio semanal del total de usuarios que hacen uso del medio de transporte en la ciudad de Ambato, dando como resultado 375 personas, tal como se detalla en la **Tabla 4**. [7]

Fecha Inicio	Fecha Fin	Línea	Total, de personas
10/5/21	16/5/21	Pinllo	3426
17/5/21	23/5/21	Picaihua	1978
24/5/21	30/5/21	San juan	1989
31/5/21	6/6/21	Joya	2916
7/6/21	13/6/21	Totoras	2822
Promedio de todas las líneas en 7 días			2626
Promedio 1 día			375

Tabla 4 Total de personas que utilizan una unidad de transporte en la ciudad de Ambato

Elaborado por: Bryan Amaguaña

Muestra

Dado el caso que la población es un numero considerables se llega a la necesidad de aplicar una muestra donde se examinara a 375 personas que hace uso de un medio de transporte, estableciendo un error de muestro del 9% y una confiabilidad del 95%, aplicando la siguiente fórmula para obtener una muestra significativa para la investigación. [7]

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

Donde :

n = Resultado de la muestra.

N = 375 Usuarios que hacen uso de un medio de transporte

Z = Nivel de confianza 95% equivalente a 1,96 (como más usual).

e = Limite aceptable de error muestral 9% (0,09).

Aplicando la formula se ingresarían los siguientes datos:

$$x = \frac{(375)(0.5)^2(1.96)^2}{(375 - 1)(0.09)^2 + (0.5)^2(1.96)^2}$$

Resultado obtenido X= 90

Al conocer el número exacto de usuarios que hacen uso de este medio de transporte se considera trabajar con 90 personas para la investigación

2.2.3 Recolección de la Información

Preguntas Básicas	Explicación
¿Para qué?	- Para alcanzar los objetivos de la investigación
¿De qué personas u objetos?	- Personas que hacen uso del transporte público -Departamento de Sistemas
¿Sobre qué aspectos?	- Automatización - Cobro de pasajes
¿Quién, Quiénes?	- Investigador: Bryan Alexander Amaguaña Amaguaña
¿Cuándo?	- Periodo académico abril – septiembre 2021
¿Dónde?	- Cooperativa de Transporte Urbano “Unión Ambateña”
¿Cuántas veces?	- Una
¿Qué técnicas de recolección?	- Encuesta - Entrevistas
¿Con qué?	- Cuestionarios - Preguntas preparadas
¿En que situación?	- Normales

Tabla 5 Recolección de Información

Elaborado por: Bryan Amaguaña

Para la recolección de información se optó por herramientas digitales que ayudan en la obtención de material teórico fundamental para facilitar el desarrollo y orientar al objetivo del presente trabajo, también fue fundamental la utilización de encuestas donde se analizó la aceptación de los usuarios que hacen uso del transporte público sobre la automatización del cobro de pasaje en la cooperativa Unión Ambateña como se adjunta en el **Anexo A**. Adicional a esto también fue de suma importancia la intervención del personal de sistemas que apporto con información fundamental sobre los recursos disponibles en la institución y de la situación actual que está pasando la empresa como se adjunta en el **Anexo B**.

2.2.3.1 Resultados de las encuestas.

Una vez aplicado las encuestas dirigidas a las personas que hacen uso del transporte público se obtuvo los siguientes resultados, los cuales están interpretados de forma estadística en gráficos y tablas.

Pregunta 1 ¿Con la llegada de la pandemia con qué frecuencia utiliza el transporte urbano?

Alternativas	Frecuencia	Porcentaje %
1-2 veces al día	49	54.4%
3-5 veces al día	30	33.3%
Más de 5 veces al día	11	12.2%
Total	90	100%

Tabla 6 Frecuencia de uso del transporte publico

Elaborado por: Bryan Amaguaña

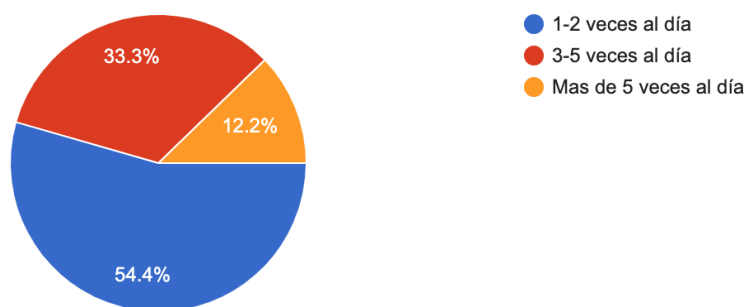


Figura 12: Frecuencia de uso del transporte público

Elaborado por: [59]

Análisis e interpretación:

De acuerdo con los datos representados en la **figura 12**, del 100% de encuestados el 12.2 % de usuarios que utilizan el transporte público, tiende a subirse a un bus más de 5 veces al día, el 33.3 % utiliza el transporte público de 3 a 5 veces al día, y el 54.4% de los encuestados utiliza el transporte público de 1 a 2 veces al día.

Basado en los datos adquirimos en la pregunta 1, podemos decir que después de la llegada de la pandemia aún existen personas que utilizan el transporte público más de 5 veces en el día, a pesar de ser la minoría representa un porcentaje significativo con respecto al riesgo de contagio que existe entre personas, sin contar también que existe personas que utilizan el transporte público más de 3 veces al día.

Pregunta 2 ¿Está conforme con el método de pago que realiza al subirse al transporte público?

Alternativas	Frecuencia	Porcentaje %
Si	31	34.4%
No	59	65.6 %
Total	90	100%

Tabla 7 Conformidad con el método de pago actual

Elaborado por: Bryan Amaguaña

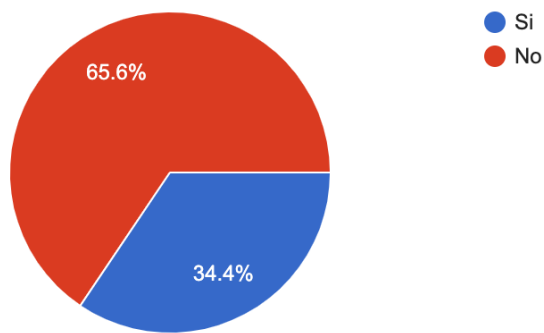


Figura 13: Conformidad con el método de pago actual

Elaborado por: [59]

Análisis e interpretación

De acuerdo con la figura 13 del 100 % de encuestados el 65.6 % de los usuarios que hacen uso del transporte público no están conforme con el método actual de pago, dando a entender que, si existe un déficit en el método actual, mientras que el 34.4 % de usuarios si están conforme con el método actual.

Pregunta 3 ¿Le resulta cómodo el intercambio de dinero entre usted y el señor conductor?

Alternativas	Frecuencia	Porcentaje %
Si	29	32.2 %
No	61	67.8 %
Total	90	100%

Tabla 8 Comodidad con el intercambio de dinero

Elaborado por: Bryan Amaguaña

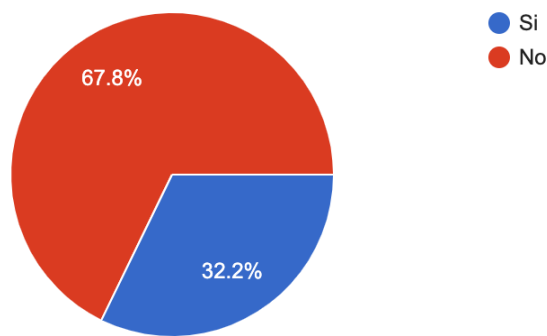


Figura 14: Comodidad con el intercambio de dinero

Elaborado por: [59]

Análisis e interpretación

De acuerdo con la **figura 14**, del 100 % de encuestados el 67.8 % de usuarios que hacen uso del medio de transporte no están cómodos con el intercambio de dinero con el señor conductor, dando a entender que existe incomodidad por parte de los usuarios al entregar y recibir dinero en el medio de transporte, de la misma forma el 32.2 % de usuarios manifiestan que si están conformes con el intercambio de dinero siendo el porcentaje mínimo en la pregunta realizada.

Pregunta 4 ¿Preferiría un método de pago del pasaje donde no tenga contacto alguno con el conductor existiendo un distanciado considerable?

Alternativas	Frecuencia	Porcentaje %
Si	67	74.4 %
No	23	25.6 %
Total	90	100 %

Tabla 9 Aceptación de un nuevo método de pago

Elaborado por: Bryan Amaguaña

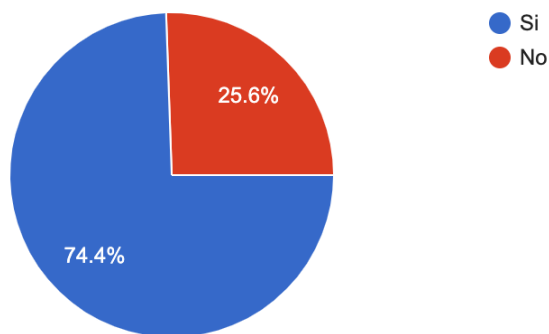


Figura 15: Aceptación de un nuevo método de pago

Elaborado por: [59]

Análisis e interpretación

De acuerdo con la figura 15, del 100 % de encuestados el 74.4 % de usuarios que hacen uso del medio de transporte público manifestó que, si acatan la idea de mantener un distanciamiento considerable con el señor conductor dando paso a un nuevo método de pago del pasaje en el transporte urbano, de igual manera el 25.6 % de los usuarios mantienen la idea de mantener el método actual de pago de pasaje en las unidades de transporte urbano.

Pregunta 5 ¿Le resultaría más fácil realizar el pago del transporte público utilizando una tarjeta recargable sin preocuparse de llevar el dinero justo o preocuparse del vuelto?

Alternativas	Frecuencia	Porcentaje %
Si	63	70 %
No	27	30 %
Total	90	100%

Tabla 10 Aceptación de la utilización de una tarjeta como forma de pago

Elaborado por: Bryan Amaguaña

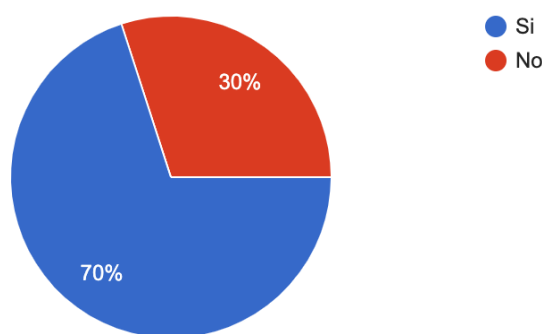


Figura 16: Aceptación de la utilización de una tarjeta como forma de pago

Elaborado por: [59]

Análisis e interpretación

De acuerdo con la **figura 16**, del 100 % de encuestados el 70 % de usuarios que hacen uso del transporte urbano afirma que, utilizar un método donde se utilice tarjetas recargables es más fácil de manejar, que mantener el método actual donde se intercambie dinero, de igual forma el 30 % de los usuarios encuestados mantienen el método actual como eficaz y más fácil de manejar que la utilización de tarjetas recargables.

Pregunta 6 ¿Está de acuerdo con la idea de acercarse a un lugar asignado para la recarga de su tarjeta para el pago del transporte público?

Alternativas	Frecuencia	Porcentaje %
Si	68	75.6 %
No	22	24.4 %
Total	90	100 %

Tabla 11 Aceptación de lugares asignados para recargas de tarjetas

Elaborado por: Bryan Amaguaña

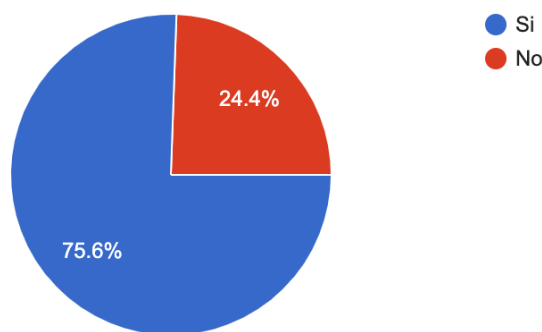


Figura 17: Aceptación de lugares asignados para recargas de tarjetas

Elaborado por: [59]

Análisis e interpretación

De acuerdo con la figura 17, del 100 % de encuestados el 75.6 % de usuarios que hacen uso del transporte urbano afirman que están de acuerdo con la idea de acercarse a puntos asignados de recarga de su respectiva tarjeta para la utilización del transporte urbano aceptando los requisitos para la correcta utilización de las tarjetas en los medios de transporte, de igual forma el 24.4 % de los usuarios encuestados manifiestan su informalidad negando la idea de acercarse a puntos específicos de recarga de tarjeta para poder utilizar el servicio de transporte urbano.

Pregunta 7 ¿Le resultaría cómodo realizar el pago del transporte público acercando una tarjeta a un lector de tarjetas sin la necesidad de realizar algún contacto?

Alternativas	Frecuencia	Porcentaje %
Si	61	67.8 %
No	29	32.2 %
Total	90	100 %

Tabla 12 Comodidad del nuevo método de pago

Elaborado por: Bryan Amaguaña

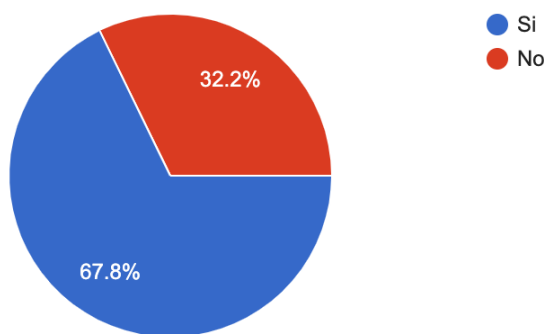


Figura 18: Comodidad del nuevo método de pago

Elaborado por: [59]

Análisis e interpretación

De acuerdo con la figura 18, del 100 % de encuestados el 67.8 % de usuarios que hacen uso del transporte urbano afirman la comodidad al realizar el pago del pasaje del transporte urbano mediante el uso de tarjetas inteligentes con la ayuda de un lector instalado en cada unidad de transporte evitando el contacto al intercambio de dinero con el señor conductor, de igual manera el 32.2 % de usuarios muestran su incomodidad al pago del pasaje del transporte urbano por medio de una tarjeta y lector de tarjetas.

Pregunta 8 ¿Le resultaría útil la utilización de una aplicación móvil donde pueda verificar los pagos y recargas que realice con su tarjeta?

Alternativas	Frecuencia	Porcentaje %
Si	69	76.7 %
No	21	23.3 %
Total	90	100 %

Tabla 13 Aceptación de la aplicación móvil

Elaborado por: Bryan Amaguaña

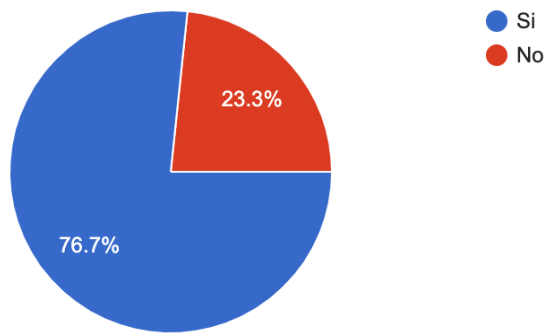


Figura 19: Aceptación de la aplicación móvil

Elaborado por: [59]

Análisis e interpretación

De acuerdo con la **figura 19**, del 100 % de encuestados el 76.6% de usuarios que hacen uso del medio de transporte urbano manifiestan que la utilización de una aplicación móvil resultaría de suma importancia para verificar todas las actividades que se realiza con la tarjeta del usuario de igual forma el 23.3 % de los usuarios encuestados no están de acuerdo con la utilización de un aplicativo móvil

Pregunta 9 ¿Está de acuerdo con acatar la nueva modalidad sobre el correcto funcionamiento del cobro de pasajes con tarjetas inteligentes?

Alternativas	Frecuencia	Porcentaje %
Si	64	71.1 %
No	26	28.9 %
Total	90	100 %

Tabla 14 Aceptación de la modalidad del cobro del pasaje

Elaborado por: Bryan Amaguaña

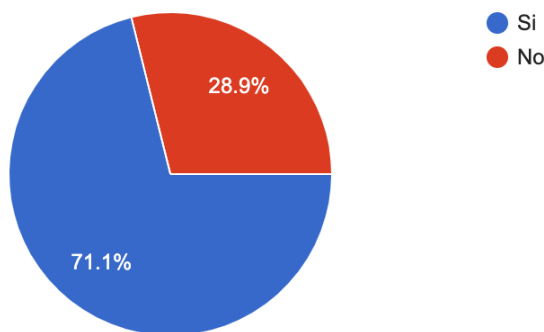


Figura 20: Aceptación de la modalidad del cobro del pasaje

Elaborado por: [59]

Análisis e interpretación

De acuerdo con la **figura 20**, del 100% de encuestados el 71.1% de usuarios de hacen uso del transporte urbano, acataran la nueva modalidad cobre el correcto funcionamiento de la utilización de tarjetas inteligentes para el pago de la tarifa del transporte urbano de igual forma el 28.9 % de usuarios encuestados manifiestan que no están dispuesto acatar la nueva modalidad del pago del pasaje del transporte urbano.

Pregunta 10 ¿Qué tan conforme estaría con la automatización del cobro del pasaje en el transporte público?

Alternativas	Frecuencia	Porcentaje %
Totalmente de acuerdo	31	34.4 %
De acuerdo	38	42.2 %
Ni de acuerdo ni desacuerdo	10	11.1 %
Desacuerdo	6	6.7 %
Totalmente en desacuerdo	5	5.6 %
Total	90	100 %

Tabla 15 Conformidad con la nueva modalidad de pago del pasaje

Elaborado por: Bryan Amaguaña

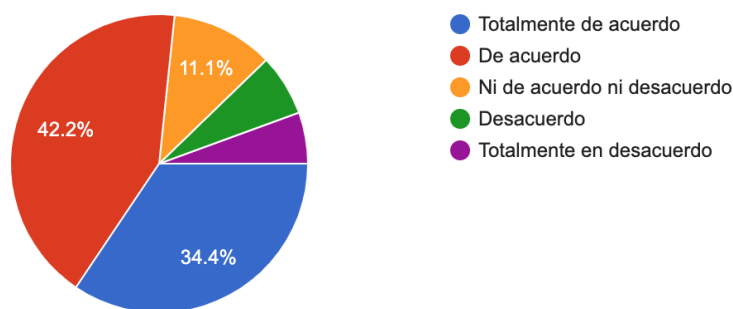


Figura 21: Conformidad con la nueva modalidad de pago del pasaje

Elaborado por: [59]

Análisis e interpretación

De acuerdo con la **figura 21**, del 100 % de encuestados el 34.4 % de usuarios que hacen uso del transporte urbano, manifiesta que está totalmente de acuerdo con la automatización del cobro del transporte urbano, el 42,2 % está de acuerdo, el 11.1 % manifiesta tu criterio como neutral, mientras que el 12.3 % manifiesta su desacuerdo con la automatización del cobro del pasaje, dando como resultado más de la mitad de las usuarios aceptan que la automatización del cobro de pasaje sería una gran ayuda para las personas, manteniendo el distanciamiento social entre usuarios y conductores.

2.2.3.2 Resultados de la entrevista

La entrevista realizada al jefe del departamento de sistemas tuvo un aporte fundamental para verificar el estado de la cooperativa y tener un punto de partida para la realización del presente proyecto, así también para cubrir las necesidades que posee la cooperativa con respecto a la automatización.

Pregunta 1 ¿Considera usted que los ingresos de la cooperativa bajaron con la llegada de la pandemia?

Con respecto a esta pregunta el Ing. Manifestó que los ingresos de la cooperativa han bajado más del 90% debido al confinamiento y con la aprobación de la normalización periódica de las actividades han ido creciendo poco a poco pero todavía está por debajo

del 50% de ingresos que se obtenía antes de la pandemia, también aclaro que no todas las unidades de transporte posee una cabina de seguridad para los señores conductores ya que este método de seguridad no es obligatorio para los señores conductores.

Pregunta 2 ¿Es recomendable que se normalice las jornadas diarias de los buses de la cooperativa?

Con respecto a esta pregunta el Ing. Manifestó lo siguiente, desde el primer día que se dio paso a salir a laborar en los buses de transporte urbano, se dio la orden a los señores propietarios a salir a trabajar con normalidad respetando el límite máximo de usuarios dentro de las unidades, así también se abasteció a cada bus con alcohol, desinfectante y trajes de bioseguridad para los señores conductores, pero la decisión de salir a laborar con normalidad está en los propietarios de cada unidad ya que no se puede obligar a laborar sabiendo la situación sanitaria que se encuentra nuestra provincia.

Pregunta 3 ¿le resultaría factible la idea de automatizar el cobro del pasaje?

Con respecto a esta pregunta el Ing. Manifestó que la automatización ya se ha venido pensando desde años atrás y se ha tomado una iniciativa implementando contadores en cada unidad de transporte, también manifestó que los contadores ayudan a tener un control con respecto a los ingresos que posee cada unidad de transporte pero no refleja un valor correcto debido a que posee un margen de error, de la misma forma manifestó que para automatizar el cobro de pasaje se necesita contratar un sistema que controle todos estos parámetros.

Pregunta 4 ¿Disponen de un sistema que contabilice los ingresos de la cooperativa?

Con respecto a esta pregunta el Ing. Manifestó que la cooperativa no dispone de un sistema en sí que controle todos los ingresos diarios que posee cada unidad de transporte, pero si controlan los ingresos que pagan a diario por la tarjeta de trabajo, también agrego que no se puede establecer una tarifa a cada línea de trabajo debido a que los ingresos de cada unidad de transporte no son estables y pueden variar.

Pregunta 5 ¿La cooperativa dispone de servidores o una base de datos donde respalden información de los ingresos y egresos diarios?

Con respecto a esta pregunta el Ing. Manifestó que no disponen de servidores, ni de una base de datos, todos los ingresos, egresos y datos de los usuarios están respaldados en documentos físicos y digitales, y manifestó que sería buena idea implementar respaldos digitales o contar con un espacio especial para guardar toda la información de la empresa de una forma segura y accesible.

2.2.4 Procesamiento y análisis de datos

De acuerdo con los resultados obtenidos en las encuestas realizadas a los usuarios que hacen uso del medio de transporte urbano, se llega a la conclusión que no están totalmente de acuerdo con el sistema actual de cobro de pasaje aceptando una nueva forma de cobro de pasaje, de la misma manera aceptan la idea de la utilización de una tarjeta inteligente para realizar el pago del pasaje y monitorear las actividades que realizar con sus tarjetas desde un aplicativo móvil.

De acuerdo con los resultados obtenidos en la entrevista realizado al jefe del departamento de sistemas se llegó a la conclusión que la cooperativa ha bajado sus ingresos considerablemente con la llegada del confinamiento adicional a esto se ha optado por implementar métodos de seguridad para la máxima protección de cada señor conductor. Pero no disponen de un método para aislar totalmente al señor conductor de los usuarios que suben a la unidad de transporte afirmando la necesidad de automatizar el cobro del pasaje.

CAPITULO III

RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de resultados.

3.1.1 Metodología de desarrollo ágil

La aplicación de metodologías ágiles en el desarrollo de aplicaciones informáticas posee un aporte fundamental pretendiendo mejorar el resultado final, siguiendo una secuencia de pasos sucesivos adaptando la forma de trabajo, a las condiciones del proyecto y el desarrollo a su entorno de ahí su nombre, pretendiendo agilizar el desarrollo de un software con iteraciones, mejorando el tiempo de desarrollo y la estructura de esta. De esta manera consigue gestionar todo tipo de proyecto de una forma flexible, eficaz y autónoma incrementando la productividad y reduciendo el costo de desarrollo. [60]

3.1.2 Metodología Scrum

Scrum es un framework utilizado en equipos de trabajo de proyectos complejos basado en 3 pilares: la transparencia, inspección y adaptación. Se fundamenta en aspectos como la flexibilidad al implementar o realizar variantes durante el desarrollo de un proyecto complejo, cuenta con un factor humano donde existe la colaboración e interacción con los clientes dando como resultado un desarrollo iterativo para obtener mejores resultados. [61]

Transparencia: Representa la forma de trabajo de todo el grupo, implicando que todos los integrantes tienen conocimiento de lo que ocurre en el proyecto fomentando el entendimiento y la visión global. [61]

Inspección: Representa la inspección del progreso del proyecto analizando el flujo de trabajo y el funcionamiento del equipo de forma auto-organizada. [61]

Adaptación: Si el proyecto presenta cambios, el equipo de trabajo se ajusta a realizarlos en el sprint de tal forma que no presenten inconvenientes en los requisitos que son vulnerables a cambios. [61]

3.1.2.1 Roles en Scrum

Product Owner: Solo puede ser una persona en todo el grupo y a su vez puede formar parte del equipo de desarrollo, posee muchos conocimientos del negocio y lleva las ideas del cliente al equipo de desarrollo a través de product backlog. [62]

Scrum Máster: Responsable de que las técnicas Scrum sean aplicadas y por ende comprendidas en el equipo de desarrollo, encargado de liderar y eliminar inconvenientes en el equipo de desarrollo a través del sprint. [62]

Equipo de desarrollo: Encargados de realizar las tareas asignadas por el Product Owner, transmitiendo la responsabilidad compartida entre todos los integrantes de tal forma que funciones como un equipo multifuncional y auto organizado. [62]

3.1.2.2 Hitos de la metodología Scrum

Sprint: Representa el cerebro de la metodología Scrum, conteniendo los hitos del proceso, avances por entregar, tiempo de entrega, etc. [61]

Sprint planning: Es una reunión de equipo donde se define las tareas y objetivos del sprint comprometiéndose el equipo a cumplir con la entrega de dicha tarea al final del sprint, dando como resultado el sprint goal y el sprint backlog. [61]

Daily meeting: Es una reunión diaria de máximo 15 minutos dentro del sprint donde participa el Scrum Máster y el equipo de desarrollo, aclarando las actividades hechas y por hacer resolviendo impedimentos y agilitando el flujo de trabajo. [61]

Sprint review: Es una reunión donde se revisa los avances del proyecto donde participa el Product Owner, el cliente y el equipo de desarrollo. El cliente analiza el

funcionamiento de los avances y asigna nuevos cambio o tareas para el equipo de desarrollo. [61]

Sprint retrospective: Es una reunión del equipo de desarrollo donde se analiza la implementación de la metodología Scrum en el último spint dando como resultado una lista de mejoras para el siguiente día dando lugar a un nuevo sprint. [61]

3.1.2.3 Herramientas de la metodología Scrum

Product Backlog: Listado de pendientes que forman un proyecto organizado por prioridad. [61]

Sprint Backlog: Grupo de tareas que forman en product backlog conocido por todo el equipo de desarrollo. [61]

3.1.2.4 Ventajas de la metodología Scrum

- Posee una estructura simple relacionada a trabajo diario. [61]
- Agilización de procesos por medio de entregas de valores muy frecuentes al cliente. [61]
- Facilidad a cambios o imprevisto del proyecto por parte del equipo de trabajo. [61]

3.1.2.5 Desventajas de la metodología Scrum

- Difícil de implementar debido a la alta organización y preparación desde los alto mandos hasta el cliente. [61]
- Todo el desarrollo del proyecto se lo realiza en equipo de trabajo. [61]
- Tratándose de cumplimientos diarios de tareas dificulta el mejoramiento de la calidad el proyecto tratándose de conseguir el camino más corto para el cumplimiento del sprint. [61]

3.1.3 Metodología Kanban

Tiene como objetivo principal la determinación de tareas por realizar así también puede cambiar su prioridad de acuerdo con los acontecimientos, tomando en cuenta que la cadena de trabajo está a disposición del grupo de trabajo facilitando la identificación de atascos o errores. [63]

3.1.3.1 Reglas de la metodología Kanban

Visualización del flujo de trabajo.

El objetivo primordial de esta regla es la visualización del trabajo por cada integrante del equipo tomando en cuenta las prioridades de cada tarea, el desarrollo del trabajo es dividido en partes, escritas y colocadas en una columna de un tablero Kanban, estas columnas o estados pueden no ser limitadas dependiendo del equipo de trabajo. [63]

Determinar el límite del trabajo en curso:

Su prioridad es detectar cuellos de botella con la colaboración de del equipo de trabajo ayudando a resolver problemas, asignando nuevos ítems a personas que tengan procesos libres. Analiza cada estado del flujo de trabajo limitando el número de tareas a realizar, tomando en cuenta una cantidad óptima para cada flujo. [63]

Controlar el tiempo de desarrollo de una actividad:

Mide el rendimiento del desarrollo del proyecto, tiene su inicio desde la petición del cliente hasta su entrega a diferencia del cycle-time que inicia su periodo en cada actividad, se recomienda optimizar el tiempo de desarrollo para cada actividad mejorando el control y el flujo de trabajo. [63]

3.1.3.2 Roles de la metodología Kanban

Service Request Manager

Define las políticas del sistema alineándolas con las expectativas de los clientes basados en aspectos como clases de servicios, evaluación de riesgos, priorización de los ítems de acuerdo con la importancia dentro del proyecto. [64]

Service Delivery Manager

Gestiona las peticiones del workflow vigiando los servicios y eliminando el trabajo obstaculizado agilitando su entrega dentro de los niveles de servicio objetivo. También facilita la planificación de entrega de avances mediante reuniones diarias. [64]

3.1.3.3 Tablero Kanban

Permite asignar tareas a todos los integrantes del equipo de trabajo, está abierto a todo tipo de opiniones, comentarios enlaces, descripciones, archivos, etc. Se basa en columnas donde cada columna representa el proceso de cada tarea y su funcionalidad depende del inicio proceso y finalización de cada actividad, de tal forma que se pueda llevar un control grafico del progreso del proyecto. [65]

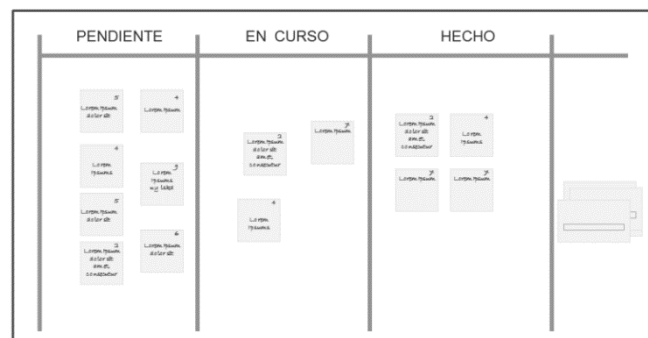


Figura 22: Tablero Kanban

Elaborado por: [66]

3.1.3.4 Ventajas de la metodología Kanban.

- Permite medir el rendimiento de cada integrante del equipo. [66]
- Organiza el flujo de trabajo de tal forma que cada fase se cumpla satisfactoriamente.[66]

- Permite cumplir con el tiempo de trabajo acordado con el cliente. [66]
- Visualización constante del progreso del proyecto de acuerdo con el tablero Kanban. [66]
- Mejora la calidad del proyecto con la ayuda del control de flujo de trabajo y solución de problemas. [66]
- Gracias a su organización evita la acumulación de trabajo durante su desarrollo

3.1.3.5 Desventajas de la metodología Kanban

- Diseñado para proyectos pequeños y medianos debido al almacenamiento del sistema, mientras más grande el sistema más costoso. [66]
- Al limitar sus tareas en un proyecto se puede dar el caso de presentarte trabajadores desocupados. [66]
- Al asumir sistemas cíclicos, no es recomendable adicionar cambios debido a que podrá afectar al proyecto negativamente. [66]
- Al existir cambios de gestión debido al incremento de tareas puede provocar un desbordamiento del trabajo. [66]

3.1.4 Metodología Xtreme Programing

XP es una metodología de desarrollo ágil que tiene como meta el desarrollo de un proyecto con la mejor eficiencia del equipo de trabajo dando como resultado un software de mayor calidad. XP está diseñado para ayudar a los desarrolladores a acoplarse a cambios en el proyecto de acuerdo con el gusto del cliente haciendo más énfasis en la adaptabilidad que en la previsibilidad. [67]

3.1.4.1 Historial de Usuarios

Es la técnica fundamental de XP donde se redacta los requisitos de usuarios. Basado en tarjetas de papel donde se describe las características del sistema ya sean requisitos funcionales o no funcionales, estas historias de usuario pueden ser rotas, modificadas o reemplazadas de acuerdo con los cambios o disposición del cliente. [68]

3.1.4.2 Roles de la metodología Xtreme Programing

Clientes: Establecen las necesidades, prioridades y marca del proyecto a desarrollar, son los usuarios finales. [67]

Programadores: Es el pilar del equipo se encarga de producir código y desarrollar el Xtreme Programing. [67]

Testers: Interpreta las peticiones del cliente y es el responsable de ayudar al equipo de desarrollo con las pruebas funcionales proporcionando herramientas de soporte de pruebas. [67]

Coach: Guía a todo el equipo marcando el rumbo del desarrollo del proyecto. [67]

Manager: Encargado de los recursos y quien coordina las actividades del equipo de trabajo siendo responsable de la comunicación externa. [67]

3.1.4.3 Procesos XP

Fase I: Exploración

Los clientes plantean las primeras historias de usuario de tal forma que proporcione información necesaria para que el equipo de trabajo se familiarice con el proyecto y con las herramientas, prácticas y tecnologías de desarrollo para el proyecto. [68]

Fase II: Planificación de Entrega

En esta fase el cliente posee un papel importante debido a que establece la prioridad de cada historia de usuario, de igual forma el equipo de trabajo analiza cada historia de usuario y estima el esfuerzo y tiempo necesario para realizar cada una de ellas, determinando un cronograma de entrega de las primeras tareas. [68]

Fase III: Iteraciones

Incluye varias iteraciones relacionadas al desarrollo del sistema antes de ser entregado. Se obtiene todos los entregables ya realizados y de igual forma se especifica las historias de usuario no abordadas, no superadas y tareas no terminadas. [68]

Fase IV: Producción o Codificado

El equipo de trabajo comienza a transformar las historias de usuarios a código, al mismo tiempo se toma decisiones de adición de características al proyecto en el estado actual. [68]

Fase V: Mantenimiento

Por cada versión del proyecto, el sistema debe mantenerse funcionando de tal forma que se desarrollen nuevas iteraciones. Esta fase puede requerir nuevas personas y cambios en la estructura del proyecto. [68]

Fase VI: Muerte del proyecto

Esta fase es la culminación del proyecto cuando el cliente ya no dispone de más historias de usuarios y el sistema cumpla con la confiabilidad y rendimiento que el cliente espera. [68]

3.1.4.4 Ventajas de la metodología XP

- Posee una relación cercana con el cliente, trabajando conjuntamente. [69]
- Cada iteración o entrega de avances está sometido a pruebas de funcionalidad.
- Fomenta la programación en parejas. [69]
- Adaptable a cambios en el sistema con mayor facilidad. [69]

3.1.4.5 Desventajas de la metodología XP

- Posee un mayor esfuerzo de desarrollo. [69]
- La intervención del cliente no siempre es comprensible. [69]
- Requiere de mucho tiempo de desarrollo. [69]
- Posee un mayor costo de desarrollo. [69]

3.1.5 Comparación de las metodologías Ágiles

	XP	KANBAN	SCRUM
Numero de integrantes	2-10 personas	indefinido	5-10 personas
Magnitud de proyecto	Pequeños proyectos	Pequeños proyectos	Para todo tipo de proyectos
Roles	- Clientes - Programadores - Testers - Coach - Manager	- Service Request Manager -Service Delivery Manager	- Product Owner - Scrum Máster - Equipo de desarrollo
Uso	Basado en la comunicación, retroalimentación y la simplicidad	Destinado a dar prioridades a tareas de trabajo	Se establece la estructura del proyecto al inicio limitando cambios
Cambios	Flexible a cambios durante el mantenimiento	Flexible a cambios en todo momento	Flexible a cambios al final del ciclo
Flujo de trabajo	Todo el proyecto esta dividido en iteraciones	Todo el proyecto esta dividido en tareas	Todo el proyecto dividido en Sprint
Entregas	Al final de cada iteración	Continuas, definidas como prioridad	Al final de cada Sprint

Tabla 16 Comparación de mitologías ágiles

Elaborado por: Bryan Amaguaña

3.1.6 Metodología Elegida

Una vez analizado las metodologías ágiles más populares en el desarrollo de software, se consideró que la metodología XP es la más adecuada para la realización del presente proyecto de investigación basado en los siguientes aspectos:

- Una de las principales razones por la que XP es la metodología ágil elegida es por el número reducido de participantes que dispone el presente proyecto, y XP es la única metodología ágil que permite un número reducido de participantes en el desarrollo de un proyecto.

- Permite la interacción con los clientes de manera continua detallando los cambios o tareas en iteraciones, reduciendo el número de errores y agilizando el desarrollo del proyecto.
- XP destaca por su diseño sencillo y por su forma de trabajar comprensible facilitando la comprensión del cliente con respecto al desarrollo y pruebas del sistema.

3.1.7 Comparación de tecnología de desarrollo Front-End

Existe múltiples tecnologías de desarrollo front-end para aplicaciones híbridas el cual es fundamental para el desarrollo de la presente aplicación, entre las alternativas más populares se tomó en cuenta React, Ionic y Flutter.

Para la selección del front-end se analizará múltiples aspectos que se adapten al desarrollo del proyecto manejando el flujo de trabajo y adaptándose a las necesidades del programa.

Tecnologías de desarrollo más populares.

Con la ayuda de Google Trends se realizó la comparación de las aplicaciones híbridas más populares en la actualidad, de los cuales vamos a descartar React, Ionic y Flutter como alternativas para el desarrollo de presente proyecto, tomando en cuenta un tiempo de comparación de 12 meses.



Figura 23: Tecnologías Híbridas más populares

Elaborado por: [70]

De acuerdo con la **figura 23**, React posee un alto nivel de popularidad en el desarrollo de aplicaciones híbridas en el mundo dejando por atrás a tecnologías como Ionic y Flutter, dando a entender que React ha evolucionado notablemente en la aceptación de las personas para el desarrollo de aplicaciones.

Ventajas y desventajas

Ionic

Ionic Framework basado en código abierto, es un SDK front-end para el desarrollo de aplicaciones híbridas utilizando tecnología web como HTML, CSS y JS. Permite el despliegue de aplicaciones híbridas con una única base de código ofrecido un diseño limpio, sencillo y funciona, empleando un capacitor (Córdova) que ayuda en a la aplicación a ejecutarse en un navegador web de forma nativa.[16]

Ventajas	Desventajas
<ul style="list-style-type: none">- Básicamente basado en Angular- De fácil aprendizaje y utilización- Integra componentes reutilizables- Posee documentación detallada y respaldo de la comunidad- Reduce los costos de desarrollo de proyectos.	<ul style="list-style-type: none">- Rendimiento bajo en aplicaciones nativas- Posee mucha dependencia de plugin- Produce aplicaciones mucho más pesadas que las aplicaciones nativas- Requiere de un alto dominio de aprendizaje

Tabla 17 Ventajas y Desventajas de Ionic

Elaborado por: Bryan Amaguaña

Flutter

Framework elaborado por Google basado en el lenguaje de programación Dart y publicado por primera vez en el 2018 para el desarrollo de aplicaciones para diferentes

plataformas ofreciendo un gran número de bibliotecas multiplataforma. Su principal característica es que posee interfaces similares a las aplicaciones nativas tanto para Android y para iOS. [14]

Ventaja	Desventaja
<ul style="list-style-type: none"> - Posee una única base de código de compilación. - Implementa un lenguaje fácil de entender, Dart. - Ofrece múltiples posibilidades por la utilización de widget. - Compila aplicaciones nativas fluidas - Amplia variedad de bibliotecas con elementos gráficos prefabricados 	<ul style="list-style-type: none"> - Dificultad de implementación de widgets por la complejidad del código. - Documentación reducida al tratarse un nuevo lenguaje de programación. - Comunidad de desarrolladores reducida. - Al actualizar un diseño se debe actualizar los módulos y compilarlos

Tabla 18 Ventajas y desventajas de Flutter

Elaborado por: Bryan Amaguaña

React

React se ha convertido una de las plataformas de desarrollo más populares en la actualidad basada en JavaScript posee una amplia documentación y una comunidad activa en todo el mundo. React Aprovecha la combinación de JSX, JavaScript, XML y muchas cosas más, también posee un diseño basado en material IU lo que permite la creación de interfaces muy llamativas en el desarrollo web y diseños nativos basados en plataformas como iOS y Android. [71]

Ventajas	Desventajas
<ul style="list-style-type: none"> - Desarrollo rentable de bajo costo - Reutiliza componentes prefabricados para un rápido desarrollo. - Utiliza JavaScript como lenguaje de desarrollo. 	<ul style="list-style-type: none"> - En algunos casos presenta problemas de depuración. - No todas las funciones nativas están disponibles en React Native.

<ul style="list-style-type: none"> - Lenguaje especializado para desarrollo multiplataforma - Permite el desarrollo de aplicaciones a través de bibliotecas y marcos. - Permite la actualización de componentes en tiempo real. - Posee múltiple documentación y comunidad activa. - Diseño similares a aplicaciones nativas 	<ul style="list-style-type: none"> - Gestión de memoria no optimizada para sus aplicaciones - Para aplicaciones grandes presente problemas de rendimiento. - Sus componentes son desarrollados por programadores independientes.
---	---

Tabla 19 Ventajas y desventajas de React

Elaborado por: Bryan Amaguaña

3.1.8 Tecnología de desarrollo Front-End elegida

Una vez analizado todos los aspectos necesarios, así como las ventajas y desventajas de cada Framework de desarrollo de aplicaciones híbridas se ha llegado a la conclusión de realizar el presente proyecto en React debido a la rentabilidad del framework así también por sus múltiples bibliotecas y documentación actualizada para el desarrollo y la implementación de la aplicación en la web y en los dispositivos móviles tanto en Android como en iOS.

React al poseer diseños en material IU permite la creación de interfaces intuitivas y agradables para los usuarios así también cuentan con librerías de componentes de código abierto para el desarrollo y modificación de los desarrolladores. La principal característica de React es el DOM Virtual lo que facilita la carga de datos y componentes sin recargar toda la página, agilizando la carga de información y dan un aspecto constante al manejo del usuario.

3.1.9 Comparación de tecnología de desarrollo Back-End

Existe múltiples tecnologías para el desarrollo Back-End, pero tomaremos en cuenta la mejor opción para combinar con el desarrollo de React, el cual compararemos

tecnologías como Laravel, Node js, y firebase destacando sus características ventajas y desventajas.

Tecnologías Back-End de desarrollo más populares.

Con la ayuda de Google Trends se comparó las tecnologías de Back-End mencionada destacando el número de búsquedas que existe en la web dando como resultado la **figura 24**, donde detalla el número de búsquedas de acuerdo con los últimos 12 meses, de esta forma comprobamos si existe una comunidad activa de las tecnologías que vamos a elegir.



Figura 24: Tecnologías Back-End más populares

Elaborado por: [72]

Ventajas y desventajas

Laravel

Laravel es un framework de tipo MVC (Modelo-Vista-Controlador) basado el PHP, tiene su funcionalidad desde la línea de comandos con Artisan (interfaz por comando) para ejecutar múltiples líneas de comando, a pesar de aun estar en etapa de evolución la nueva versión Laravel 8 posee cualidades que facilitan el desarrollo de proyectos como sistemas de rutas, landing de inicio y namespace. [73]

Ventajas	Desventajas
<ul style="list-style-type: none"> - Fácil de usar y aprender (Framework PHP). - Gran extensión de usabilidad en el mercado. - Documentación extensa y comunidad activa. - Utiliza un motor de plantillas que acelera las tareas de compilación. - Incorpora enrutamiento inverso 	<ul style="list-style-type: none"> - Algunas Librerías dependen de Symfony - El enrutamiento inverso resulta complejo. - Posee una ejecución lenta.

Tabla 20 Ventajas y desventajas de Laravel

Elaborado por: Bryan Amaguaña

Firestore

Creado como una plataforma de desarrollo móvil y web, en la actualidad perteneciente a google, cuenta con múltiples herramientas de desarrollo, análisis y mantenimiento de aplicaciones híbridas. Está constituido por Firestore Analytics, base de datos en tiempo real, notificaciones tipo Push, mensajería y múltiples funciones tanto de pago como de forma gratuita. [74]

Ventajas	Desventajas
<ul style="list-style-type: none"> - Facilidad de utilización - Posee una documentación extensa y comunidad activa. - Posee un alto rendimiento en sus servicios - Posee herramientas para aplicaciones en tiempo real - Posee soporte multiplataforma - Compatibilidad con múltiples software Back-End 	<ul style="list-style-type: none"> - Pose un plan gratuito limitado - Complejidad en consultas complejas y búsquedas de datos - Funciones no optimizadas en lo referente a consultas y búsquedas de datos

Tabla 21 Ventajas y desventajas de Firebase

Elaborado por: Bryan Amaguaña

Node js

Node js se define como un entorno de ejecución JavaScript de parte del servidor construido con el motor V8 de Chrome, basado en un modelo E/S lo que resulta liviano y eficiente, orientado a eventos y sin bloqueos. Considera como una de las mejores opciones para la creación de juegos. [75]

Ventajas	Desventajas
<ul style="list-style-type: none"> - Basado en Open Source - Esta optimizada para recursos de forma asíncrona - Mayor facilidad de desarrollo ágil - FullStack, puede ser utilizado tanto en el lado del servidor como del cliente - Modularidad, permite la utilización de módulos explícitamente hechos para su necesidad sin requerir módulos adicionales. 	<ul style="list-style-type: none"> - Posee un nivel de Arquitectura muy simple. - Posee librerías estándar redundando su funcionalidad - Baja confiabilidad, y muy vulnerables a caídas del sistema por un fallo. - Su programación es asíncrona dificultando su entendimiento para diferentes programadores

Tabla 22 Ventajas y desventajas de Node js

Elaborado por: Bryan Amaguaña

Comparación de tecnologías Back-End

Con la ayuda de G2, una página web que permite comparar tecnologías de desarrollo de software se realizó un análisis de las tecnologías de Back-End donde se especificó diferentes parámetros a comparar como los requisitos.

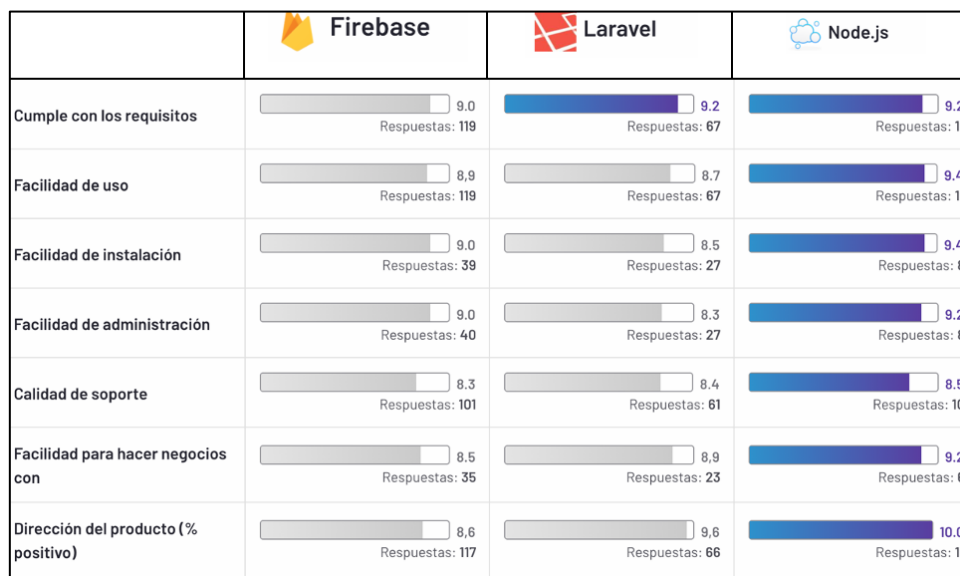


Figura 25: Comparación de Tecnologías Back-End más populares

Elaborado por: [76]

3.1.10 Tecnología de desarrollo Back-End elegida

Node js es la tecnología elegida para la realización de este proyecto no solo por su facilidad y su trabajo asíncrono, también posee cualidades y librerías muy útiles para la realización de la aplicación. Al ser un lenguaje fácil de implementar basado en JavaScript y compatible con múltiples tecnologías Front-End facilita la implementación con React que también utiliza JavaScript.

Las principales cualidades que hacen de Node js la mejor opción para la implementación en el presente proyecto son:

- Posee una licencia de código abierto.
- El servidor puede llegar a ser altamente escalable por poseer un solo hilo con bucles de eventos ayudando al servidor a responder sin bloqueos.
- No posee almacenamiento de bufer, generando sus datos en fragmentos.
- Posee un motor V8 que agiliza la ejecución de código.
- Posee una programación asíncrona, sin bloqueos ejecutando la API sin esperar una petición anterior

3.1.11 Comparación de Base de datos

Para el almacenamiento de información para el presente proyecto se optó utilizar una base de datos NoSQL debido a la cantidad de información que va a manejar, dejando a un lado los problemas de escalabilidad y rendimiento que posee una base de datos SQL.

Base de datos NoSQL más populares.

Gracias a la ayuda de Google Trends se realizó la comparación de Base de datos no SQL destacadas por el número de búsquedas en la web, las bases de datos que fueron tomados en cuenta son: Casandra, Redis y MongoDB detallando los resultados en la **figura 26**

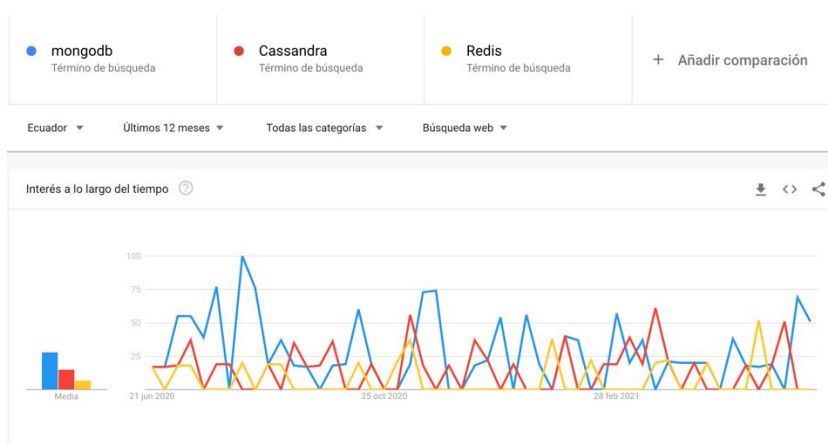


Figura 26: Comparación de Base de Datos NoSQL

Elaborado por: [77]

Ventajas y Desventajas

Redis

Fue creado por la necesidad de mejorar los tiempos de respuestas con respecto a consultas simples, basado en una estructura de tablas hash (Clave-Valor), abriendo múltiples posibilidades a aplicaciones de negocios. Dispone de 5 estructuras de datos: String, Lists, Sets, Sorted Sets y Hashes. [78]

Ventajas	Desventajas
<ul style="list-style-type: none"> - Posee una velocidad de respuesta muy rápido por su almacenamiento en memoria. - Permite el almacenamiento de datos en disco para recuperarlos posteriormente. - Facilidad de configuración. - Posee de alta disponibilidad. - Utiliza una estructura maestro-esclavo mejorando el desempeño de lectura - Compatible con múltiples lenguajes de programación. 	<ul style="list-style-type: none"> - No recomendable para búsquedas de datos complejas. - Requiere de mucha capacidad de memoria principal. - Compatible con estructura de datos limitada.

Tabla 23 Ventajas y desventajas de Redis

Elaborado por: Bryan Amaguaña

Cassandra

Cassandra es una base de datos NoSQL lanzado en el 2008, creado inicialmente por Facebook con la finalidad de obtener datos desestructurados o semiestructurados más rápido que las bases de datos SQL. Es una base de datos distribuida con escalas lineales siguiendo un patrón P2P (peer to peer), muy distinto al patrón maestro-esclavo. Gracias a la replicación de datos es tolerante a fallos ya que, si se cae un nodo, el nodo replicado puede continuar su trabajo. [79]

Ventajas	Desventajas
<ul style="list-style-type: none"> - Dispone de una alta disponibilidad, evitando que el sistema colapse. - Posee estándares bien definidos. - Posee un gran método de escritura 	<ul style="list-style-type: none"> - Para la conexión de nuevos nodos es necesario un amplio conocimiento de Cassandra - A medida que crece la base de datos los costos van creciendo

- Posee una gran variedad de documentación	- Posee mayor complejidad de instalación que otras bases de datos
--	---

Tabla 24 Ventajas y desventajas de Cassandra

Elaborado por: Bryan Amaguaña

MongoDB

MongoDB permite realizar todo tipo de consultas, búsquedas de campo, expresiones regulares y consultas de rangos, con la posibilidad de devolver campos específicos de documentos. Posee múltiples funciones como Replicación, Indexación, Balanceo de Carga y ejecución de código del lado del servidor. [25]

Ventajas	Desventajas
<ul style="list-style-type: none"> - No requiere de muchos recursos para su funcionamiento - Posee un costo bajo. - Amplia documentación y comunidad activa. - Dispone de múltiples complementos para JavaScript. 	<ul style="list-style-type: none"> - Al ser NoSQL no es recomendable utilizar para consultas complejas - No posee un tiempo considerable en el mercado. - No dispone de Joins para la realización de consultas

Tabla 25 Ventajas y desventajas de MongoDB

Elaborado por: Bryan Amaguaña

3.1.12 Base de datos elegida

La base de datos elegida para el presente proyecto es MongoDB debido a su amplia documentación, y por las herramientas disponibles para JavaScript. Cuenta con un espacio en la nube llamado mongo Atlas que permite el almacenamiento de información de forma gratuita.

MongoDB cuenta con múltiples características que fueron fundamental para seleccionar una base de datos entre las más importantes:

- Al ser una base de datos descentralizada permite la incorporación y el soporte de estructuras distribuidas, facilitando la incorporación de nuevos nodos.
- No requiere de muchos recursos para su correcto funcionamiento, sin mencionar que posee una herramienta que permite el almacenamiento de datos en la nube
- Optimización de consultas de datos para bases grandes.
- Posee una mayor velocidad de inserción de datos que otras bases NoSQL

3.1.13 Materiales elegidos para la lectura y recarga de tarjetas NFC

Para la automatización del cobro del pasaje en la cooperativa de transporte urbano “Unión Ambateña” es necesario la utilización de un lector de tarjetas NFC que registren la información por medio de servicios a la base de datos, de igual forma es necesario un dispositivo que permita la lectura y escritura de tarjetas NFC para la creación y recargas de tarjetas. Estos dispositivos serán elaborados como prototipos de prueba que complementen al sistema informático.

Los materiales para la elaboración tanto del lector como para las recargas de tarjetas NFC fueron elegidos de acuerdo con la disponibilidad de materiales existentes dentro del cantón Ambato detallados en la **Tabla 3**, diseñando un prototipo que cumpla con sus funciones.

3.1.14 Análisis de los requerimientos de la aplicación.

Una vez analizado los requerimientos de la empresa, se llega a la conclusión, que la aplicación estará compuesta de una aplicación web que será administrada por una o varias personas de la empresa, así también constara con una aplicación móvil que será desarrollado en React Native, donde los usuarios puedan realizar consultas de recargas y saldos, y por último constara de un dispositivo escritor y lector de tarjetas NFC para

los usuarios dueños de una unidad de transporte, y un dispositivo escritor y lector de tarjetas NFC para los usuarios asignados para realizar recargas.

El desarrollo de la aplicación estará basado en MERN Stack diseñado de la combinación de tecnologías utilizadas para el desarrollo de aplicaciones web Premium, con la ayuda de JavaScript, compuesta de componentes de código abierto como MongoDB, Express, React y Node.js.

La aplicación será implementada a través de un host gratuito debido a que es un proyecto de investigación y la empresa requiere realizar pruebas de campo, el host que se utilizara para el despliegue de la aplicación será Heroku, donde se alojara tanto la aplicación web como los servicios que se ocuparan. Para la base de datos se utilizará las herramientas de MongoDB donde los datos serán almacenados en la nube de mongo Atlas.

El lector de tarjetas NFC para el cobro del pasaje será fijado en una unidad de transporte donde se realizará las pruebas de funcionalidad tratando de imitar la acción de un pasajero al momento de ingresar a una unidad de transporte, de la misma forma el dispositivo para realizar recargas de las tarjetas será puesto a pruebas de funcionalidad.

3.2 Desarrollo de la propuesta

3.2.1 Fase I: Exploración

En la primera fase de la metodología XP (Xtreme Programming) el cliente define los requerimientos que serán de suma importancia para la elaboración de la aplicación, de la misma forma todos los requerimientos serán transformados a historias de usuarios donde se establecerá las actividades a realizar con las herramientas ya definidas.

3.2.1.1 Arquitectura de la aplicación.

El diseño de la arquitectura de la aplicación está basado en MERN Stack combinados por MongoDB para la base de datos Express y Node js como Back-End y en el lado del cliente React para la administración de la aplicación. También se creará una aplicación móvil para los usuarios que estará desarrollado en React Native y un dispositivo prototipo de lectura y escritura para el cobro del pasaje a los usuarios.

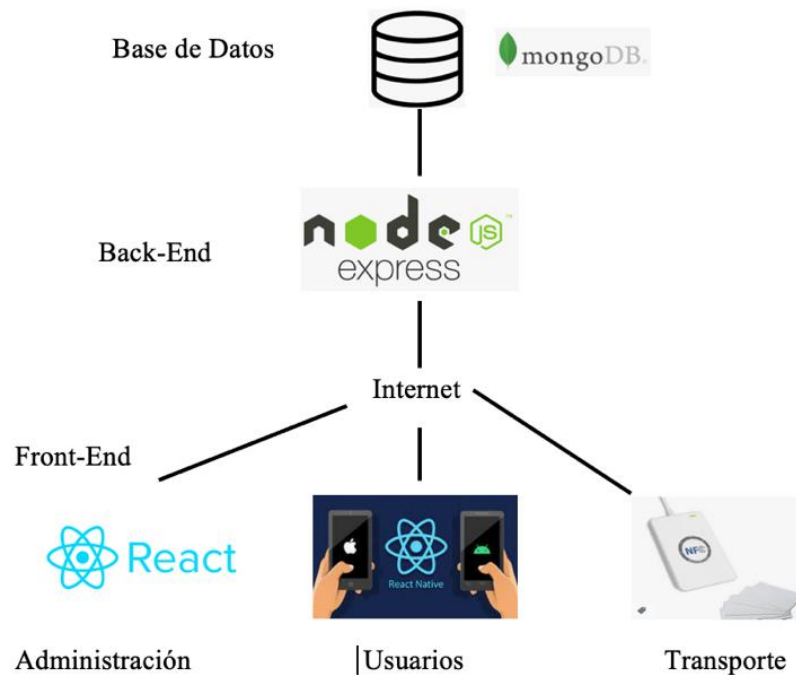


Figura 27: Arquitectura de la aplicación

Elaborado por: Bryan Amaguaña

3.2.1.2 Proceso normal del cobro del pasaje

El proceso que realiza normalmente una persona para realizar el pago del pasaje al momento de subirse a un bus de transporte es totalmente manual, manteniendo en contacto entre el conductor y el pasajero, identificando los dos únicos usuarios que intervienen en todo el proceso.

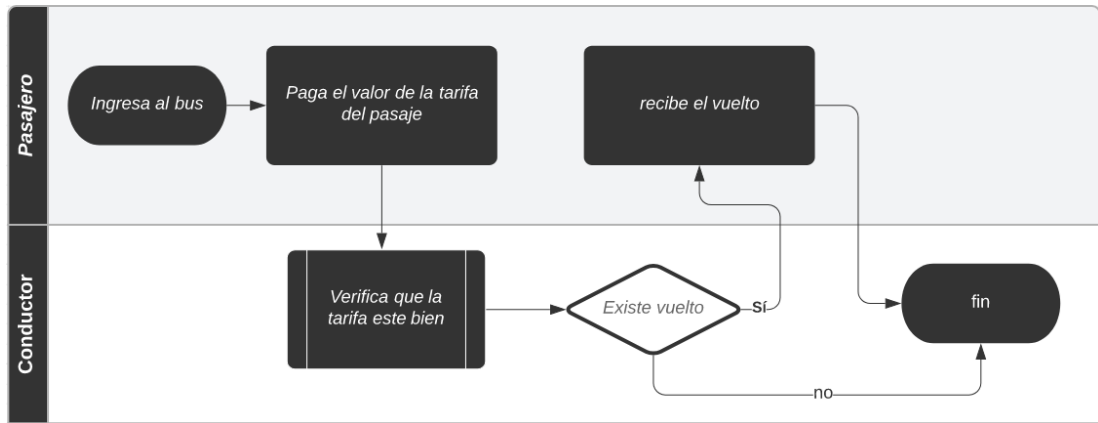


Figura 27.1: Proceso manual del cobro de pasaje

Elaborado por: Bryan Amaguaña

3.2.1.3 Proceso automatizado del cobro del pasaje

El proceso que realiza una persona al momento de ingresar a un medio de transporte para realizar el pago del pasaje una vez automatizado los procesos manuales está compuesto por usuario pasajero y la maquina lectora de tarjetas NFC, eliminando la interacción del conductor con el pasajero.

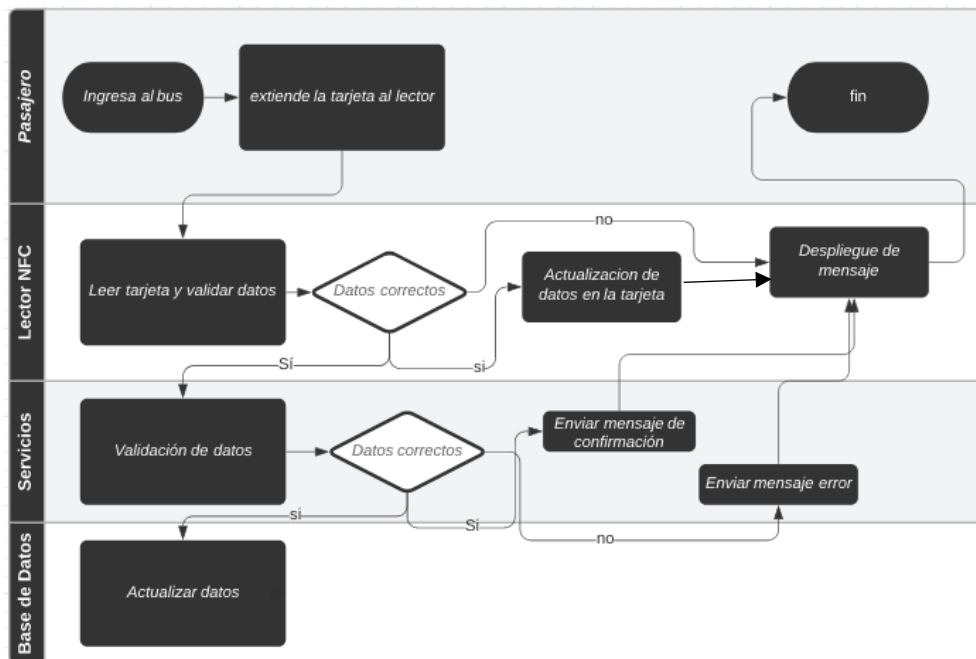


Figura 27.2: Proceso automatizado del cobro de pasaje

Elaborado por: Bryan Amaguaña

3.2.1.4 Definición de roles

Los roles cumplen una función muy importante en la organización de todas las personas involucradas para el desarrollo del presente proyecto, definiendo sus aportes para un desarrollo exitoso, al tratarse de un proyecto de investigación no se contará con la totalidad de roles que existen en la metodología XP.

Nombre	Nombre Rol	Descripción	Rol XP
Bryan Amaguaña	Tesista	Persona responsable de planificar, diseñar, desarrollar y realizar pruebas en el sistema	Programador, Tester
Ing. Mg. Hernando Buenaño	Tutor del proyecto de investigación	Persona responsable de las evaluaciones mensuales y del cumplimiento del cronograma para el desarrollo del proyecto	Consultor Coach
Ing. Ricardo Gavilanes	Jefe del departamento de sistemas	Persona que requiere el software, conoce los requerimientos y el proceso a manejar sobre el cobro del pasaje	Cliente

Tabla 26 Definiciones de Roles XP

Elaborado por: Bryan Amaguaña

3.2.2 Fase II: Planificación de Proyecto

Al utilizar la metodología XP, la planificación del proyecto de investigación estará basada en historias de usuario donde se organizará las prioridades y las actividades a realizar, desarrollando en diferentes iteraciones

3.2.3 Historias de usuarios

Las historias de usuarios están estructuradas de acuerdo con la **tabla 27** y hacen referencia a los requisitos y actividades que debe cumplir el software. Una vez ya establecido las historias de usuario, son divididas en actividades que se deben cumplir en un periodo de tiempo.

Historia de usuario	
Número:	Usuario:
Nombre de la historia:	
Prioridad en el negocio:	Riesgo en el desarrollo:
Puntos estimados:	Iteración asignada:
Programador responsable:	
Descripción:	
Observación:	

Tabla 27 Historia de usuario

Elaborado por: Bryan Amaguaña

Elementos de una plantilla de una historia de usuario

- **Número:** Índice o identificador obligatorio en una historia de usuario.
- **Usuario:** Usuario al que se asigna la historia de usuario.
- **Nombre de la historia:** Título asignado a la historia de usuario.
- **Prioridad en el negocio:** Prioridad de la historia de usuario de acuerdo con la necesidad del usuario (Baja, Media, Alta).
- **Riesgo en el desarrollo:** Riesgo de desarrollo al desarrollar la historia de usuario (Baja, Media, Alta).
- **Puntos estimados:** Tiempo estimado en días para el desarrollo de la historia de usuario.
- **Iteración asignada:** Iteración que pertenece a la historia de usuario.

- **Programador responsable:** Persona responsable para el desarrollo de la historia de usuario.
- **Descripción:** Espacio disponible para el cliente donde expresara las peticiones de validaciones o procesos.
- **Observación:** Espacio disponible para redactar procesos relacionados a la historia de usuario

Historia de usuario	
Número: 001	Usuario: Programador
Nombre de la historia: Estructura del proyecto	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Bryan Amaguaña	
Descripción: Es fundamental definir la estructura del proyecto con el fin de realizar mantenimientos futuros sin impedir el funcionamiento del software.	
Observación:	

Tabla 28 Historia 001 – Estructura del proyecto

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 002	Usuario: Programador
Nombre de la historia: Definición y modelo de la base de datos	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: Bryan Amaguaña	
Descripción: La estructura de la base de datos ayudara a la conexión del sistema Web con la aplicación móvil y los lectores de tarjetas NFC.	
Observación: La Cooperativa de transporte urbano “Unión Ambateña” no cuenta con un servidor.	

Tabla 29 Historia 002 – Definición y modelo de la base de datos

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 003	Usuario: Programador
Nombre de la historia: Definición y modelo de servicios para la página web	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: Bryan Amaguaña	
Descripción: Los servicios para la página web serán establecidos de acuerdo con las actividades que se incorporarán a la aplicación web para el almacenamiento y consumo de datos de la base de datos.	
Observación:	

Tabla 30 Historia 003 – Definición y modelo de servicios para la página web

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 004	Usuario: Programador, Cliente
Nombre de la historia: Despliegue de la base de datos	
Prioridad en el negocio: Medio	Riesgo en el desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Bryan Amaguaña	
Descripción: La base de datos será desplegado en una herramienta gratuita que proporciona MongoDB (Mongo Atlas) para la realización de pruebas de funcionalidad.	
Observación:	

Tabla 31 Historia 004 – Despliegue de la base de datos

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 005	Usuario: Programador, Cliente
Nombre de la historia: Despliegue de los servicios	
Prioridad en el negocio: Medio	Riesgo en el desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Bryan Amaguaña	
Descripción: Los servicios serán desplegados en una plataforma gratuita (Heroku) para su respectivo consumo y pruebas de funcionalidad.	
Observación:	

Tabla 32 Historia 005 – despliegue de los servicios

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 006	Usuario: Programador, Cliente
Nombre de la historia: Diseño del layout Admin	
Prioridad en el negocio: Medio	Riesgo en el desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Bryan Amaguaña	
Descripción: El diseño del Layout Admin se realizará de forma general para todo el programa, para la incorporación de los componentes futuros.	
Observación: Se tomará como referencia los colores de la empresa para la elaboración del Layout Admin.	

Tabla 33 Historia 006 – Diseño del layout Admin

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 007	Usuario: Todos
Nombre de la historia: Pagina inicio de sesión	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto

Puntos estimados: 8	Iteración asignada: 2
Programador responsable: Bryan Amaguaña	
Descripción: Se visualizará la página de inicio de sesión para los usuarios de la cooperativa de transporte “Unión Ambateña” puedan acceder al sistema.	
Observación: Solo los usuarios registrados en el sistema por medio de un usuario y contraseña tendrán acceso al sistema.	

Tabla 34 Historia 007 – Pagina Inicio de sesión

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 008	Usuario: Todos
Nombre de la historia: Diseño del MenuSider	
Prioridad en el negocio: Medio	Riesgo en el desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Bryan Amaguaña	
Descripción: Visualización de un menú lateral izquierdo donde se encuentre todos los módulos del sistema de fácil acceso.	
Observación:	

Tabla 35 Historia 008 – Diseño del MenuSider

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 009	Usuario: Todos
Nombre de la historia: Diseño del Menutop	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Bryan Amaguaña	
Descripción: Visualización de un menú superior donde se pueda cerrar sesión y desplegar el MenuSider.	

Observación:

Tabla 36 Historia 009 – Diseño del Menutop

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 010	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Usuario	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 5	Iteración asignada: 2
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar o buscar los datos de un usuario, solo el usuario con el rol de administrador podrá modificar o desactivar a un usuario.	
Observación:	

Tabla 37 Historia 010 – Diseño del modal y vistas del componente Usuario

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 011	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente rol	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar los datos de un rol, solo el usuario con el rol de administrador podrá modificar un rol.	
Observación:	

Tabla 38 Historia 011 – Diseño del modal y vistas del componente rol

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 012	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Persona	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar o buscar los datos de una persona, solo el usuario con el rol de administrador podrá modificar una persona.	
Observación:	

Tabla 39 Historia 012 – Diseño del modal y vistas del componente Persona

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 013	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Buses	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar o buscar los datos de un bus, solo el usuario con el rol de administrador podrá modificar o desactivar un bus.	
Observación:	

Tabla 40 Historia 013 – Diseño del modal y vistas del componente Buses

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 014	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Tarjetas	

Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar o buscar los datos de una tarjeta, solo el usuario con el rol de administrador podrá modificar o desactivar una tarjeta.	
Observación:	

Tabla 41 Historia 014 – Diseño del modal y vistas del componente Tarjetas

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 015	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Tipo_Pasajero	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar los datos de un Tipo_Pasajero, solo el usuario con el rol de administrador podrá modificar o desactivar un Tipo_Pasajero.	
Observación:	

Tabla 42 Historia 015 – Diseño del modal y vistas del componente Tipo_Pasajero

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 016	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Pasajero	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Bryan Amaguaña	

Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar o buscar los datos de un Pasajero, solo el usuario con el rol de administrador podrá modificar o desactivar un Pasajero.
Observación:

Tabla 43 Historia 016 – Diseño del modal y vistas del componente Pasajero

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 017	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Cobro_Pasaje	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar o buscar los datos de un Cobro_Pasaje.	
Observación:	

Tabla 44 Historia 017 – Diseño del modal y vistas del componente Cobro_Pasaje

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 018	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Recargas	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar o buscar los datos de una Recarga.	
Observación:	

Tabla 45 Historia 018 – Diseño del modal y vistas del componente Recargas

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 019	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Menu_Web	
Prioridad en el negocio: Medio	Riesgo en el desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar los datos de un Menu_Web, solo el usuario con el rol de administrador podrá modificar un Menu_Web.	
Observación:	

Tabla 46 Historia 019 – Diseño del modal y vistas del componente Menu_Web

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 020	Usuario: Todos
Nombre de la historia: Diseño del modal y vistas del componente Contenido_Web	
Prioridad en el negocio: Medio	Riesgo en el desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Bryan Amaguaña	
Descripción: Los usuarios que inicien sesión en el sistema podrán visualizar los datos de un Contenido_Web, solo el usuario con el rol de administrador podrá modificar un Contenido_Web.	
Observación:	

Tabla 47 Historia 020 – Diseño del modal y vistas del componente Contenido_Web

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 021	Usuario: Todos
Nombre de la historia: Diseño de la página de inicio de Contenido_Web	
Prioridad en el negocio: Medio	Riesgo en el desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Bryan Amaguaña	
Descripción: Pagina publica para todas las personas que tengan el enlace del sistema, se visualizara los datos de la empresa junto a los Menu_web.	
Observación: La página de inicio de Contenido_Web tendrá una ruta diferente para que no tengan acceso al sistema por medio del login.	

Tabla 48 Historia 021 – Diseño de la página de inicio de Contenido_Web

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 022	Usuario: Todos
Nombre de la historia: Diseño página de inicio del sistema	
Prioridad en el negocio: Bajo	Riesgo en el desarrollo: Bajo
Puntos estimados: 4	Iteración asignada: 4
Programador responsable: Bryan Amaguaña	
Descripción: Pagina que se mostrara al momento de ingresar al sistema, conformado por un gráfico de barras donde se mostrara el ingreso por mes de las unidades de transporte	
Observación: Se representará el cobro del pasaje y las recargas realizadas por mes.	

Tabla 49 Historia 022 – Diseño página de inicio del sistema

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 023	Usuario: Todos
Nombre de la historia: Definición y modelo de los servicios para el dispositivo lector Tarjetas NFC	

Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 4
Programador responsable: Bryan Amaguaña	
Descripción: Los servicios para el dispositivo lector Tarjetas NFC serán establecidos de acuerdo con las actividades que se incorporen.	
Observación:	

Tabla 50 Historia 023 – Definición y modelo de los servicios para el dispositivo lector Tarjeta NFC

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 024	Usuario: Todos
Nombre de la historia: Definición y modelo de los servicios para el dispositivo recargar Tarjetas NFC	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 4
Programador responsable: Bryan Amaguaña	
Descripción: Los servicios para el dispositivo recargar Tarjetas NFC serán establecidos de acuerdo con las actividades que se incorporen.	
Observación:	

Tabla 51 Historia 024 – Definición y modelo de los servicios para el dispositivo recargar Tarjetas NFC

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 025	Usuario: Todos
Nombre de la historia: Diseño del primer prototipo lector Tarjetas NFC	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 6	Iteración asignada: 5
Programador responsable: Bryan Amaguaña	

Descripción: El dispositivo será elaborado de acuerdo con las funcionalidades expresadas por el cliente, y será ubicado en una unidad de transporte para la prueba de funcionalidad
Observación: Es necesario el asesoramiento de un Ing. Electrónico para el diseño y codificación del dispositivo.

Tabla 52 Historia 025 – Diseño del primer prototipo lector Tarjetas NFC

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 026	Usuario: Todos
Nombre de la historia: Diseño del primer prototipo Recargas Tarjeta NFC	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 6	Iteración asignada: 5
Programador responsable: Bryan Amaguaña	
Descripción: El dispositivo será elaborado de acuerdo con las funcionalidades expresadas por el cliente, y serán asignadas a una persona responsable.	
Observación: Es necesario el asesoramiento de un Ing. Electrónico para el diseño y codificación del dispositivo.	

Tabla 53 Historia 026 – Diseño del primer prototipo Recargas Tarjetas NFC

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 027	Usuario: Todos
Nombre de la historia: Pruebas de funcionalidad del dispositivo recargas y Cobro	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 5
Programador responsable: Bryan Amaguaña	
Descripción: Cada dispositivo será evaluado en su ámbito de trabajo para simular el trabajo que ejercerá y eliminar errores.	

Observación: Las pruebas de realizará en los edificios de la cooperativa de transporte urbano “Unión Ambateña” con presencia del gerente y el jefe del departamento de sistemas

Tabla 54 Historia 027 – Pruebas de funcionalidad del dispositivo recargas y Cobros

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 028	Usuario: Todos
Nombre de la historia: Inicio de sesión en la aplicación UnionApp	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 6	Iteración asignada: 5
Programador responsable: Bryan Amaguaña	
Descripción: Se visualizará la página de inicio de sesión en la aplicación móvil para los usuarios pasajeros registrados en el programa	
Observación: Para el ingreso al programa será necesario la cedula de la persona y el código de la tarjeta como contraseña.	

Tabla 55 Historia 028 – Inicio de sesión en la aplicación UnionApp

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 029	Usuario: Todos
Nombre de la historia: Diseño del componente Register de la App	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 6
Programador responsable: Bryan Amaguaña	
Descripción: Se visualizará un formulario donde el pasajero se registre en la aplicación con su información personal, adicional a eso debe ingresar el código de su tarjeta y realizar su respectiva validación.	
Observación: El registro de la tarjeta estará basado en la edad del pasajero y será validado por la aplicación.	

Tabla 56 Historia 029 – Diseño del componente Register de la App

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 030	Usuario: Todos
Nombre de la historia: Diseño del menú de la App	
Prioridad en el negocio: Medio	Riesgo en el desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 6
Programador responsable: Bryan Amaguaña	
Descripción: Se visualizará un menú inferior donde estén ubicados todas las funciones disponibles para los usuarios pasajeros	
Observación: cada ítem estará compuesto de un icono.	

Tabla 57 Historia 030 – Diseño del menú de la App

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 031	Usuario: Todos
Nombre de la historia: Diseño del componente home para la App	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 6
Programador responsable: Bryan Amaguaña	
Descripción: Se visualizará la pantalla de inicio donde se muestre la información de la tarjeta y los datos del pasajero	
Observación: Cuando la tarjeta este bloqueada se mostrada en los datos de la tarjeta	

Tabla 58 Historia 031 – Diseño del componente home para la App

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 032	Usuario: Todos
Nombre de la historia: Diseño del componente Pagos para la App	

Prioridad en el negocio: Alto	Riesgo en el desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 6
Programador responsable: Bryan Amaguaña	
Descripción: Se visualizará la pantalla de vistas de todos los pagos realizados con la tarjeta registrada.	
Observación: Si el pasajero cambia de tarjeta se mostrará todos los pagos realizados con la tarjeta actual.	

Tabla 59 Historia 032 – Diseño del componente Pagos para la App

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 033	Usuario: Todos
Nombre de la historia: Diseño del componente Recargas para la App	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 6
Programador responsable: Bryan Amaguaña	
Descripción: Se visualizará la pantalla de vistas de todas las recargas realizados con la tarjeta registrada.	
Observación: al cambia de tarjeta se mostrará todas las recargas realizadas con la tarjeta actual.	

Tabla 60 Historia 033 – Diseño del componente Recargas para la App

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 034	Usuario: Todos
Nombre de la historia: Diseño del componente Mi cuenta para la App	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 4	Iteración asignada: 6
Programador responsable: Bryan Amaguaña	

Descripción: Se visualizará la pantalla Mi cuenta con múltiples funciones donde el usuario pueda visualizar sus datos, editarlos, bloquear una tarjeta, cambiar una tarjeta o cerrar sesión de la aplicación.

Tabla 61 Historia 034 – Diseño del componente Mi cuenta para la App

Elaborado por: Bryan Amaguaña

Historia de usuario	
Número: 035	Usuario:
Nombre de la historia: Pruebas de funcionalidad de la App UnionApp	
Prioridad en el negocio: Alto	Riesgo en el desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 6
Programador responsable: Bryan Amaguaña	
Descripción: Se realizará pruebas de funcionalidad de la aplicación móvil donde se registre, inicie sesión, cambie de tarjeta, bloquee una tarjeta y modifique los datos del usuario registrado	
Observación:	

Tabla 62 Historia 035 – pruebas de funcionalidad de la App UnionApp

Elaborado por: Bryan Amaguaña

3.2.4 Estimación de historias de usuario

Para la estimación del desarrollo del proyecto se estableció un tiempo de trabajo de 5 horas al día, tomando como referencia los días laborables, dando como resultado la tabla de evaluación de las historias de usuario detallados en la **tabla 63**

Iteración	N°	Historia de usuario	Tiempo estimado	
			Días	Horas
1	001	Estructura del proyecto	4	20
1	002	Definición y modelo de la base de datos	6	30
1	003	Definición y modelo de servicios para la página web	6	30
1	004	Despliegue de la base de datos	2	10
1	005	Despliegue de los servicios	2	10
2	006	Diseño del layout Admin	3	15

2	007	Página inicio de sesión	8	40
2	008	Diseño del MenuSider	2	10
2	009	Diseño del Menutop	2	10
2	010	Diseño del modal y vistas del componente usuario	5	25
3	011	Diseño del modal y vistas del componente rol	3	15
3	012	Diseño del modal y vistas del componente Persona	3	15
3	013	Diseño del modal y vistas del componente Buses	3	15
3	014	Diseño del modal y vistas del componente Tarjetas	3	15
3	015	Diseño del modal y vistas del componente Tipo_Pasajero	3	15
3	016	Diseño del modal y vistas del componente Pasajero	3	15
3	017	Diseño del modal y vistas del componente Cobro_Pasaje	2	10
4	018	Diseño del modal y vistas del componente Recargas	3	15
4	019	Diseño del modal y vistas del componente Menu_Web	3	15
4	020	Diseño del modal y vistas del componente Contenido_Web	3	15
4	021	Diseño de la página de inicio de Contenido_Web	3	15
4	022	Diseño página de inicio del sistema	4	20
4	023	Definición y modelo de los servicios para el dispositivo lector Tarjetas NFC	2	10
4	024	Definición y modelo de los servicios para el dispositivo recargador de tarjetas NFC	2	10
5	025	Diseño del primer prototipo Lector de tarjetas NFC	6	30
5	026	Diseño del primer prototipo Recargas Tarjeta NFC	6	30
5	027	Pruebas de funcionalidad del dispositivo recargas y Cobro de pasajes	2	10
5	028	Inicio de sesión en la aplicación UnionApp	6	30
6	029	Diseño del componente Register de la App	3	15
6	030	Diseño del menú de la App	2	10
6	031	Diseño del componente home para la App	3	15
6	032	Diseño del componente Pagos para la App	3	15
6	033	Diseño del componente Recargas para la App	3	15
6	034	Diseño del componente Mi cuenta para la App	4	20
6	035	Pruebas de funcionalidad de la App UnionApp	2	10
Total, del tiempo estimado (Días, Horas)			120	600

Tabla 63 Estimación de historias de usuarios

Elaborado por: Bryan Amaguaña

3.2.5 Roles internos del sistema.

La seguridad por parte del sistema con los usuarios que manipularan la información ingresada en la base de datos está basado en roles asignados cuidadosamente por la persona de mayor jerarquía o usuario administrador, definiendo los siguientes roles para cada persona que realizara una actividad dentro del sistema.

- **Administrador:** Rol de mayor jerarquía, capaz de realizar todo tipo de acción en el sistema, posee permisos para agregar, desactivar o editar usuarios creados dentro del sistema, así también posee permisos para modificar o eliminar información de la empresa registrada en la base de datos.
- **Sistemas:** Rol de jerarquía intermedia, capaz de realizar modificaciones menores en las personas y cuentas registradas en la base de datos, posee permisos para realizar consultas de información o reportes de los ingresos registrados en el sistema.
- **Secretario/a:** Rol de jerarquía menor, no posee permisos para realizar cambios en la información del sistema, posee permisos para realizar consultas de información, generar reportes y visualizar información de las personas registradas en el sistema.
- **Roles creados por el administrador:** Roles creados por el usuario de mayor jerarquía, poseen una jerarquía de menor prioridad, dando permisos de ingresar al sistema para visualizar información, realizar consultas, generar reportes y visualizar información de las personas registradas en el sistema.

3.2.6 Tareas

Cada historia de usuario se divide en múltiples tareas que son asignados una persona responsable de culminar exitosamente el desarrollo, limitándose a un periodo de tiempo.

Tarea	
Numero:	Numero de historia:
Nombre:	
Tipo de Tarea	Puntos Estimados:
Fecha de Inicio	Fecha Fin
Programador responsable	
Descripción	

Tabla 64 Plantilla de tareas de la historia de usuario

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 1	Numero de historia: 001
Nombre: Estructura del proyecto	
Tipo de Tarea: Análisis de Arquitectura	Puntos Estimados: 4
Fecha de Inicio: 21/12/2020	Fecha Fin: 24/12/2020
Programador responsable: Bryan Amaguaña	
Descripción: Análisis del estado actual de la empresa y definición de herramientas a utilizar para el desarrollo del proyecto	

Tabla 65 Tarea 1- Estructura del proyecto.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 2	Numero de historia: 002
Nombre: Definición de la herramienta a utilizar para la base de datos	
Tipo de Tarea: Análisis de bases de Datos	Puntos Estimados: 2
Fecha de Inicio: 25/12/2020	Fecha Fin: 26/12/2020
Programador responsable: Bryan Amaguaña	

Descripción: Análisis de las herramientas actuales para el desarrollo de la base de datos que más se adapte al desarrollo del proyecto.

Tabla 66 Tarea 2- Definición de la herramienta a utilizar para la base de datos.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 3	Numero de historia: 002
Nombre: Diseño de la base de datos	
Tipo de Tarea: Diseño de la base de datos	Puntos Estimados: 4
Fecha de Inicio: 28/12/2020	Fecha Fin: 31/12/2020
Programador responsable: Bryan Amaguaña	
Descripción: Diseño del modelo de la base de datos de acuerdo con los requerimientos	

Tabla 67 Tarea 3- Diseño de la base de datos.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 4	Numero de historia: 003
Nombre: Definición de la herramienta a utilizar para el desarrollo del Back-End	
Tipo de Tarea: Análisis de Back-End	Puntos Estimados: 3
Fecha de Inicio: 04/01/2021	Fecha Fin: 06/01/2021
Programador responsable: Bryan Amaguaña	
Descripción: Análisis de las herramientas actuales para el desarrollo del Back-End que más se adapte al desarrollo del proyecto.	

Tabla 68 Tarea 4- Definición de la herramienta a utilizar para la base de datos.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 5	Numero de historia: 003
Nombre: Diseño de los servicios	
Tipo de Tarea: Diseño del Back-End	Puntos Estimados: 3
Fecha de Inicio: 07/01/2021	Fecha Fin: 11/01/2021
Programador responsable: Bryan Amaguaña	
Descripción: Diseño del modelo de los servicios de acuerdo con los requerimientos	

Tabla 69 Tarea 5- Diseño de los servicios.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 6	Numero de historia: 004
Nombre: Análisis de herramientas para el despliegue de la base de datos	
Tipo de Tarea: Análisis de herramientas de host	Puntos Estimados: 1
Fecha de Inicio: 12/01/2021	Fecha Fin: 12/01/2021
Programador responsable: Bryan Amaguaña	
Descripción: Análisis de las herramientas actuales para el despliegue de la base de datos	

Tabla 70 Tarea 6- Análisis de herramientas para el despliegue de la base de datos.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 7	Numero de historia: 004
Nombre: Despliegue de la base de datos	
Tipo de Tarea: Despliegue de la base de datos	Puntos Estimados: 1
Fecha de Inicio: 13/01/2021	Fecha Fin: 13/01/2021
Programador responsable: Bryan Amaguaña	

Descripción: Despliegue de la base de datos en la herramienta previamente analizada (Mongo Atlas) para la realización de pruebas

Tabla 71 Tarea 7- Despliegue de la base de datos.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 8	Numero de historia: 005
Nombre: Análisis de herramientas para el despliegue del Back-End.	
Tipo de Tarea: Análisis de herramientas de host	Puntos Estimados: 1
Fecha de Inicio: 14/01/2021	Fecha Fin: 14/01/2021
Programador responsable: Bryan Amaguaña	
Descripción: Análisis de las herramientas actuales para el despliegue del Back-End	

Tabla 72 Tarea 8- análisis de herramientas para el despliegue del Back-End.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 9	Numero de historia: 005
Nombre: Despliegue del Back-End	
Tipo de Tarea: Despliegue del Back-End	Puntos Estimados: 1
Fecha de Inicio: 15/01/2021	Fecha Fin: 15/01/2021
Programador responsable: Bryan Amaguaña	
Descripción: Despliegue del Back-End en la herramienta previamente analizada (Heroku) para la realización de pruebas	

Tabla 73 Tarea 9- Despliegue del Back-End.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 10	Numero de historia: 006
Nombre: Análisis de herramientas para el desarrollo Front-End	
Tipo de Tarea: Análisis de herramientas Front-End	Puntos Estimados: 1
Fecha de Inicio: 18/01/2021	Fecha Fin: 18/01/2021
Programador responsable: Bryan Amaguaña	
Descripción: Análisis de herramientas Front-End para el desarrollo del proyecto	

Tabla 74 Tarea 10- Análisis de herramientas para el desarrollo Front-End.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 11	Numero de historia: 006
Nombre: Diseño del layout Admin	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 19/01/2021	Fecha Fin: 20/01/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del layout Admin para la incorporación de componentes	

Tabla 75 Tarea 11- Diseño del layout Admin.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 12	Numero de historia: 007
Nombre: Diseño del formulario login	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 21/01/2021	Fecha Fin: 22/01/2021
Programador responsable: Bryan Amaguaña	

Descripción: Construcción del diseño de la página de inicio para los usuarios registrados en la aplicación.

Tabla 76 Tarea 12- Diseño del formulario login.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 13	Numero de historia: 007
Nombre: Configuración del AccessToken y RefreshToken	
Tipo de Tarea: Desarrollo Back-End	Puntos Estimados: 4
Fecha de Inicio: 25/01/2021	Fecha Fin: 29/01/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción de la estructura del AccessToken para el ingreso de los usuarios y del RefreshToken para refrescar la página.	

Tabla 77 Tarea 13- Configuración del AccessToken y RefreshToken.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 14	Numero de historia: 007
Nombre: Implementación de servicios con el login	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 01/02/2021	Fecha Fin: 03/02/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al login para la generación del token e ingreso al sistema	

Tabla 78 Tarea 14- Implementación de servicios con el login.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 15	Numero de historia: 008
Nombre: Diseño del MenuSider	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 04/02/2021	Fecha Fin: 08/02/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del MenuSider donde se incorporará los componentes del sistema	

Tabla 79 Tarea 15 – Diseño del MenuSider.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 16	Numero de historia: 009
Nombre: Diseño del MenuTop	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 09/02/2021	Fecha Fin: 11/02/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del MenuTop donde se incorporar el despliegue del MenuSider y el enlace para cerrar sesión	

Tabla 80 Tarea 16 – Diseño del MenuTop.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 17	Numero de historia: 010
Nombre: Diseño de modal y vistas para el componente Usuario	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 12/02/2021	Fecha Fin: 15/02/2021

Programador responsable: Bryan Amaguaña
Descripción: Construcción del modal y vistas de todos los usuarios registrados en el programa

Tabla 81 Tarea 17 – Diseño del modal y vistas para el componente Usuario.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 18	Numero de historia: 010
Nombre: Diseño del modal Agregar y Editar Usuarios	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 16/02/2021	Fecha Fin: 18/02/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal agregar y editar Usuarios integrados al componente Usuarios	

Tabla 82 Tarea 18 – Diseño del modal Agregar y Editar Usuarios.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 19	Numero de historia: 010
Nombre: Implementación de los servicios en el componente Usuarios	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 19/02/2021	Fecha Fin: 19/02/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al momento de agregar, editar o desactivar un usuario, tomando en cuenta el rol del usuario que realiza la acción.	

Tabla 83 Tarea 19 – Implementación de los servicios en el componente Usuarios.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 20	Numero de historia: 011
Nombre: Diseño de modal y vistas para el componente Rol	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 22/02/2021	Fecha Fin: 22/02/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal y vistas de todos los roles registrados en el programa.	

Tabla 84 Tarea 20 – Diseño de modal y vistas para el componente Rol.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 21	Numero de historia: 011
Nombre: Diseño del modal Agregar Rol	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 23/02/2021	Fecha Fin: 23/02/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal agregar rol integrados al componente Rol	

Tabla 85 Tarea 21 – Diseño del modal Agregar Rol.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 22	Numero de historia: 011
Nombre: Implementación de los servicios en el componente Rol	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 24/02/2021	Fecha Fin: 24/02/2021

Programador responsable: Bryan Amaguaña
Descripción: Implementación de los servicios al momento de agregar un rol, tomando en cuenta el rol del usuario que realiza la acción.

Tabla 86 Tarea 22 – Implementación de los servicios en el componente Rol.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 23	Numero de historia: 012
Nombre: Diseño de modal y vistas para el componente Persona	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 25/02/2021	Fecha Fin: 25/02/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal y vistas de todas las personas registrados en el programa.	

Tabla 87 Tarea 23 – Diseño del modal y vistas para el componente Persona.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 24	Numero de historia: 012
Nombre: Diseño del modal agregar y editar Persona	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 26/02/2021	Fecha Fin: 26/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal agregar y editar Personas integrados al componente Persona	

Tabla 88 Tarea 24 – Diseño del modal agregar y editar Persona.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 25	Numero de historia: 012
Nombre: Implementación de los servicios en el componente Persona	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 01/03/2021	Fecha Fin: 01/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al momento de agregar, editar o desactivar una persona, tomando en cuenta el rol del usuario que realiza la acción.	

Tabla 89 Tarea 25 – Implementación de los servicios en el componente Persona.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 26	Numero de historia: 013
Nombre: Diseño de modal y vistas para el componente Buses	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 02/03/2021	Fecha Fin: 02/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal y vistas de todos los buses registrados en el programa.	

Tabla 90 Tarea 26 – Diseño de modal y vistas para el componente Buses.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 27	Numero de historia: 013
Nombre: Diseño del modal agregar y editar Buses	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 03/03/2021	Fecha Fin: 03/03/2021

Programador responsable: Bryan Amaguaña
Descripción: Construcción del modal agregar y editar un bus integrado al componente Buses

Tabla 91 Tarea 27 – Diseño del modal agregar y editar Buses.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 28	Numero de historia: 013
Nombre: Implementación de los servicios en el componente Buses	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 04/03/2021	Fecha Fin: 04/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al momento de agregar, editar o desactivar un bus, tomando en cuenta el rol del usuario que realiza la acción.	

Tabla 92 Tarea 28 – Implementación de los servicios en el componente Buses.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 29	Numero de historia: 014
Nombre: Diseño de modal y vistas para el componente Tarjetas	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 05/03/2021	Fecha Fin: 05/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal y vistas de todas las tarjetas registrados en el programa.	

Tabla 93 Tarea 29 – Diseño del modal y vistas para el componente Tarjetas.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 30	Numero de historia: 014
Nombre: Diseño del modal agregar y editar Tarjetas	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 08/03/2021	Fecha Fin: 09/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal agregar y editar tarjetas integrado al componente Tarjetas	

Tabla 94 Tarea 30 – Diseño del modal agregar y editar Tarjetas.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 31	Numero de historia: 014
Nombre: Implementación de los servicios en el componente Tarjetas	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 10/03/2021	Fecha Fin: 10/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al momento de agregar, editar o desactivar una tarjeta, tomando en cuenta el rol del usuario que realiza la acción.	

Tabla 95 Tarea 31 – Implementación de los servicios en el componente Tarjetas.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 32	Numero de historia: 015
Nombre: Diseño de modal y vistas para el componente Tipo_Pasaje	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 11/03/2021	Fecha Fin: 11/03/2021

Programador responsable: Bryan Amaguaña
Descripción: Construcción del modal y vistas de todos los tipos de pasajes registrados en el programa.

Tabla 96 Tarea 32 – Diseño de modal y vistas para el componente Tipo_Pasaje.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 33	Numero de historia: 015
Nombre: Diseño del modal agregar y editar Tipo_Pasaje	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 12/03/2021	Fecha Fin: 12/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal agregar y editar tipos de pasajes integrado al componente Tipo_Pasaje	

Tabla 97 Tarea 33 – Diseño del modal agregar y editar Tipo_pasaje.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 34	Numero de historia: 015
Nombre: Implementación de los servicios en el componente Tipo_Pasaje	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 15/03/2021	Fecha Fin: 15/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al momento de agregar, editar o desactivar un tipo de pasaje, tomando en cuenta el rol del usuario que realiza la acción.	

Tabla 98 Tarea 34 – Implementación de los servicios en el componente Tipo_Pasaje.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 35	Numero de historia: 016
Nombre: Diseño de modal y vistas para el componente Pasajero	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 16/03/2021	Fecha Fin: 16/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal y vistas de todos los pasajeros registrados en el programa.	

Tabla 99 Tarea 35 – Diseño de modal y vistas para el componente Pasajero.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 36	Numero de historia: 016
Nombre: Diseño del modal agregar y editar Pasajeros	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 17/03/2021	Fecha Fin: 17/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal agregar y editar pasajeros integrado al componente Pasajeros	

Tabla 100 Tarea 36 – Diseño del modal agregar y editar Pasajeros.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 37	Numero de historia: 016
Nombre: Implementación de los servicios en el componente Pasajeros	
Tipo de Tarea:	Puntos Estimados: 1
Fecha de Inicio: 18/03/2021	Fecha Fin: 18/03/2021

Programador responsable: Bryan Amaguaña
Descripción: Implementación de los servicios al momento de agregar, editar o desactivar un pasajero, tomando en cuenta el rol del usuario que realiza la acción.

Tabla 101 Tarea 37 – Implementación de los servicios en el componente Pasajeros.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 38	Numero de historia: 017
Nombre: Diseño de vistas para el componente Cobro_Pasaje	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 19/03/2021	Fecha Fin: 19/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción de vistas de todos los cobros de pasajes registrados en el programa.	

Tabla 102 Tarea 38 – Diseño de vistas para el componente Cobro_Pasaje.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 39	Numero de historia: 017
Nombre: Diseño del buscador y reportes del componente Cobro_Pasaje	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 22/03/2021	Fecha Fin: 22/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del buscador Cobro_Pasaje filtrados por el número del bus y la fecha, desplegando los datos en pantalla y a través de un reporte	

Tabla 103 Tarea 39 - Diseño buscador y reportes del componente Cobro_Pasaje.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 40	Numero de historia: 017
Nombre: Implementación de los servicios en el componente Cobro_Pasaje	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 23/03/2021	Fecha Fin: 23/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al momento de visualizar un cobro, no se tomará en cuenta el rol del usuario.	

Tabla 104 Tarea 40 - Implementación de los servicios en el componente Cobro_Pasaje.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 41	Numero de historia: 018
Nombre: Diseño de vistas para el componente Recargas	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 24/03/2021	Fecha Fin: 24/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción de vistas de todas las recargas de pasajes registrados en el programa.	

Tabla 105 Tarea 41 - Diseño de vistas para el componente Recargas.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 42	Numero de historia: 018
Nombre: Diseño del buscador y reportes del componente Recargas	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 25/03/2021	Fecha Fin: 25/03/2021

Programador responsable: Bryan Amaguaña
Descripción: Construcción del buscador recargas filtrados por el nombre del usuario y la fecha, desplegando los datos en pantalla y a través de un reporte

Tabla 106 Tarea 42 - Diseño del buscador y reportes del componente Recargas.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 43	Numero de historia: 018
Nombre: Implementación de los servicios en el componente Recargas	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 26/03/2021	Fecha Fin: 26/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al momento de visualizar una recarga, no se tomará en cuenta el rol del usuario.	

Tabla 107 Tarea 43 – Implementación de los servicios en el componente Recargas.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 44	Numero de historia: 019
Nombre: Diseño de modal y vistas para el componente Menu_Web	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 29/03/2021	Fecha Fin: 29/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal y vistas de todos menús registrados en el programa.	

Tabla 108 Tarea 44 – Diseño de modal y vistas para el componente Menu_Web.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 45	Numero de historia: 019
Nombre: Diseño del modal agregar y editar Menu_Web	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 30/03/2021	Fecha Fin: 30/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del modal agregar y editar menús integrado al componente Menu_Web	

Tabla 109 Tarea 45 – Diseño del modal agregar y Editar Menu_Web.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 46	Numero de historia: 019
Nombre: Implementación de los servicios en el componente Menu_Web	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 31/03/2021	Fecha Fin: 31/03/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al momento de agregar, editar o desactivar menú, tomando en cuenta el rol del usuario que realiza la acción.	

Tabla 110 Tarea 46 – Implementación de los servicios en el componente Menu_Web.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 47	Numero de historia: 020
Nombre: Diseño del modal para el componente Contenido_Web	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 01/04/2021	Fecha Fin: 03/04/2021
Programador responsable: Bryan Amaguaña	

Descripción: Construcción del modal de todos los datos que contiene el Contenido_Web ingresados en el programa.

Tabla 111 Tarea 47 - Diseño del modal para el componente Contenido_Web.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 48	Numero de historia: 020
Nombre: Implementación de los servicios en el componente Contenido_Web	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 04/04/2021	Fecha Fin: 04/04/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios al momento de agregar contenido, tomando en cuenta el rol del usuario que realiza la acción.	

Tabla 112 Tarea 48 – Implementación de los servicios en el componente Contenido_Web.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 49	Numero de historia: 021
Nombre: Diseño de rutas y vistas de la página de inicio	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 05/04/2021	Fecha Fin: 07/04/2021
Programador responsable: Bryan Amaguaña	
Descripción: Creación de rutas y vistas para la implementación del contenido web, como página principal del sistema al publico	

Tabla 113 Tarea 49 – Diseño de rutas y vistas de la página de inicio.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 50	Numero de historia: 021
Nombre: Implementación de servicios en la página de inicio del contenido web	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 1
Fecha de Inicio: 08/04/2021	Fecha Fin: 09/04/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los datos ingresados a través del componente Contenido_Web, para la visualización en el sistema sin necesidad de iniciar sesión	

Tabla 114 Tarea 50 – Implementación de servicios en la página de inicio del contenido web.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 51	Numero de historia: 022
Nombre: Diseño de gráfico de barras representando los cobros de pasajes	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 12/04/2021	Fecha Fin: 13/04/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción de gráficos de barras representando los cobros por mes de cada unidad de transporte, exportando como reportes a través de búsquedas	

Tabla 115 Tarea 51 – Diseño de gráfico de barras representando los cobros de pasajes.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 52	Numero de historia: 022
Nombre: Diseño de gráfico de barras representando las recargas de tarjetas	
Tipo de Tarea: Desarrollo Front-End	Puntos Estimados: 2
Fecha de Inicio: 14/04/2021	Fecha Fin: 15/04/2021

Programador responsable: Bryan Amaguaña
Descripción: Construcción de gráficos de barras representando las recargas por mes de cada usuario, exportando como reportes atreves de búsquedas

Tabla 116 Tarea 52 - Diseño de grafico de barras representando las recargas de tarjetas.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 53	Numero de historia: 023
Nombre: Diseño de servicios para el dispositivo Lector Tarjetas NFC	
Tipo de Tarea: Desarrollo Back-End	Puntos Estimados: 2
Fecha de Inicio: 16/04/2021	Fecha Fin: 19/04/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción de todos los servicios para el almacenamiento de datos de los cobros de pasajes.	

Tabla 117 Tarea 53 – Diseño de servicios para el dispositivo Lector Tarjetas NFC.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 54	Numero de historia: 024
Nombre: Diseño de servicios para el dispositivo Recargas Tarjetas NFC	
Tipo de Tarea: Desarrollo Back-End	Puntos Estimados: 2
Fecha de Inicio: 20/04/2021	Fecha Fin: 21/04/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción de todos los servicios para el almacenamiento de datos de las recargas realizadas por los usuarios asignados.	

Tabla 118 Tarea 54 - Diseño de servicios para el dispositivo Recargas Tarjetas NFC.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 55	Numero de historia: 025
Nombre: Adquisición de materiales para el dispositivo Lector Tarjeta NFC	
Tipo de Tarea: Desarrollo Dispositivo NFC	Puntos Estimados: 1
Fecha de Inicio: 22/04/2021	Fecha Fin: 23/04/2021
Programador responsable: Bryan Amaguaña	
Descripción: Adquisición de componentes compatibles con Arduino para la lectura y escritura de datos en tarjetas NFC con conexión GSM	

Tabla 119 Tarea 55 – Adquisición de materiales para el dispositivo Lector Tarjetas NFC.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 56	Numero de historia: 025
Nombre: Diseño y codificación del dispositivo lector de Tarjetas NFC	
Tipo de Tarea: Desarrollo Dispositivo NFC	Puntos Estimados: 5
Fecha de Inicio: 26/04/2021	Fecha Fin: 03/05/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del diseño y codificación simultanea del prototipo lector de tarjetas NFC	

Tabla 120 Tarea 56 – Diseño y codificación del dispositivo lector de tarjetas NFC.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 57	Numero de historia: 026
Nombre: Adquisición de materiales para el dispositivo Recargar Tarjeta NFC	
Tipo de Tarea: Desarrollo Dispositivo NFC	Puntos Estimados: 1

Fecha de Inicio: 04/05/2021	Fecha Fin: 05/05/2021
Programador responsable: Bryan Amaguaña	
Descripción: Adquisición de componentes compatibles con Arduino para la lectura y escritura de datos en tarjetas NFC con conexión GSM	

Tabla 121 Tarea 57 - Adquisición de materiales para el dispositivo Recargar Tarjetas NFC.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 58	Numero de historia: 026
Nombre: Diseño y codificación del dispositivo Recargar Tarjetas NFC	
Tipo de Tarea: Desarrollo Dispositivo NFC	Puntos Estimados: 5
Fecha de Inicio: 06/05/2021	Fecha Fin: 13/05/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del diseño y codificación simultanea del prototipo Recargar tarjetas NFC	

Tabla 122 Tarea 58 - Diseño y codificación del dispositivo Recargar Tarjetas NFC.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 59	Numero de historia: 027
Nombre: Realizar pruebas de funcionalidad de ambos dispositivos	
Tipo de Tarea: Desarrollo Dispositivo NFC	Puntos Estimados: 2
Fecha de Inicio: 14/05/2021	Fecha Fin: 17/05/2021
Programador responsable: Bryan Amaguaña	
Descripción: Se realizará pruebas de funcionalidad a los 2 dispositivos en su respectivo campo de trabajo.	

Tabla 123 Tarea 59 – Realizar pruebas de funcionalidad de ambos dispositivos.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 60	Numero de historia: 028
Nombre: Diseño del componente móvil Login.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 2
Fecha de Inicio: 18/05/2021	Fecha Fin: 20/05/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del diseño y rutas de los componentes de la aplicación móvil.	

Tabla 124 Tarea 60 – Diseño del componente móvil Login.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 61	Numero de historia: 028
Nombre: Configuración del AccessToken del login móvil	
Tipo de Tarea: Desarrollo Back-End	Puntos Estimados: 2
Fecha de Inicio: 21/05/2021	Fecha Fin: 24/05/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del AccessToken por parte del Back-End para el ingreso a la aplicación móvil.	

Tabla 125 Tarea 61 – Configuración del AccessToken del login móvil.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 62	Numero de historia: 028
Nombre: Implementación de los servicios para el componente móvil Login	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 2
Fecha de Inicio: 25/05/2021	Fecha Fin: 26/05/2021

Programador responsable: Bryan Amaguaña
Descripción: Implementación de los servicios para el ingreso a la aplicación móvil.

Tabla 126 Tarea 62 – Implementación de los servicios para el componente móvil login.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 63	Numero de historia: 029
Nombre: Diseño del componente móvil Register.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 1
Fecha de Inicio: 27/05/2021	Fecha Fin: 27/05/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del diseño para el registro de pasajeros con sus respectivas validaciones.	

Tabla 127 Tarea 63 – Diseño del componente móvil Register.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 64	Numero de historia: 029
Nombre: Implementación de los servicios para el componente móvil Register.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 2
Fecha de Inicio: 28/05/2021	Fecha Fin: 31/05/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios para el registro de pasajeros con el ingreso de datos personales y el código de la tarjeta previamente validado.	

Tabla 128 Tarea 64 - - Implementación de los servicios para el componente móvil Register.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 65	Numero de historia: 030
Nombre: Diseño del menú inferior para la navegación de la aplicación	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 2
Fecha de Inicio: 01/06/2021	Fecha Fin: 02/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: construcción del menú inferior donde se incorporar ítems de navegación de la aplicación.	

Tabla 129 Tarea 65 – Diseño del menú inferior para la navegación de la aplicación.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 66	Numero de historia: 031
Nombre: Diseño del componente móvil Home.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 1
Fecha de Inicio: 03/06/2021	Fecha Fin: 03/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del diseño para desplegar la información del usuario juntamente con los datos de la tarjeta registrada	

Tabla 130 Tarea 66 – Diseño del componente móvil Home.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 67	Numero de historia: 031
Nombre: Implementación de los servicios para el componente móvil Home.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 2
Fecha de Inicio: 04/06/2021	Fecha Fin: 07/06/2021

Programador responsable: Bryan Amaguaña
Descripción: Implementación de los servicios para el despliegue de información de la base de datos.

Tabla 131 Tarea 67 – Implementación de los servicios para el componente móvil home.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 68	Numero de historia: 032
Nombre: Diseño del componente móvil Pagos.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 1
Fecha de Inicio: 08/06/2021	Fecha Fin: 08/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del diseño para desplegar la información de los pagos realizados filtrados por fechas con la tarjeta registrada	

Tabla 132 Tarea 68 – Diseño del componente móvil Pagos.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 69	Numero de historia: 032
Nombre: Implementación de los servicios para el componente móvil Pagos.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 2
Fecha de Inicio: 09/06/2021	Fecha Fin: 10/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios para el despliegue de información de la base de datos.	

Tabla 133 Tarea 69 – Implementación de los servicios para el componente móvil Pagos.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 70	Numero de historia: 033
Nombre: Diseño del componente móvil Recargas.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 1
Fecha de Inicio: 11/06/2021	Fecha Fin: 11/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del diseño para desplegar la información de las recargas realizados filtrados por fechas con la tarjeta registrada	

Tabla 134 Tarea 70 – Diseño del componente móvil Recargas.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 71	Numero de historia: 033
Nombre: Implementación de los servicios para el componente móvil Recargas.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 2
Fecha de Inicio: 14/06/2021	Fecha Fin: 15/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios para el despliegue de información de la base de datos.	

Tabla 135 Tarea 71 – Implementación de los servicios para el componente móvil Recargas.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 72	Numero de historia: 034
Nombre: Diseño del componente móvil Mi Cuenta.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 1
Fecha de Inicio: 16/06/2021	Fecha Fin: 16/06/2021

Programador responsable: Bryan Amaguaña
Descripción: Construcción del diseño móvil Mi Cuenta donde se desplegará los sub-ítems que conforman el componente.

Tabla 136 Tarea 72 – Diseño del componente móvil Mi Cuenta.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 73	Numero de historia: 034
Nombre: Diseño de los sub-ítems del componente móvil Mi Cuenta.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 1
Fecha de Inicio: 17/06/2021	Fecha Fin: 17/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: Construcción del diseño de los sub-ítems del componente Mi cuenta (Actualizar información, Bloquear tarjeta, Cambiar de tarjeta, cerrar sesión, información).	

Tabla 137 Tarea 73 – Diseño de los sub-ítems del componente móvil Mi Cuenta.

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 74	Numero de historia: 034
Nombre: Implementación de los servicios para el componente móvil Mi Cuenta.	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 2
Fecha de Inicio: 18/06/2021	Fecha Fin: 19/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: Implementación de los servicios para el despliegue de información de la base de datos.	

Tabla 138 Tarea 74- Implementación de los servicios para el componente móvil Mi Cuenta

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 75	Numero de historia: 035
Nombre: Pruebas de funcionalidad de los dispositivos NFC con la aplicación web	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 1
Fecha de Inicio: 21/06/2021	Fecha Fin: 21/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: Se realizará pruebas en los dispositivos NFC tanto de cobro de pasaje como en el dispositivo de recargas, verificando datos en la aplicación web	

Tabla 139 Tarea 75 – Pruebas de funcionalidad de los dispositivos NFC con la aplicación web

Elaborado por: Bryan Amaguaña

Tarea	
Numero: 76	Numero de historia: 035
Nombre: Pruebas de funcionalidad de la aplicación móvil con la aplicación web	
Tipo de Tarea: Desarrollo Front-End Móvil	Puntos Estimados: 1
Fecha de Inicio: 22/06/2021	Fecha Fin: 22/06/2021
Programador responsable: Bryan Amaguaña	
Descripción: Se realizará pruebas en la aplicación móvil, verificando datos en la aplicación web.	

Tabla 140 Tarea 76 - Pruebas de funcionalidad de la aplicación móvil con la aplicación web

Elaborado por: Bryan Amaguaña

3.2.3 Fase III: Iteraciones

3.2.3.1 Plan de entregas

De acuerdo con la metodología XP, el plan de entregas del presente proyecto fue establecido tomando en cuenta los 20 días hábiles que posee un mes, y estableciendo una jornada de trabajo de 5 horas diarias.

N°	Historias de Usuario	Tiempo estimado		Iteración Asignada						Entrega Asignada					
		Días	Horas	1	2	3	4	5	6	1	2	3	4	5	6
001	Estructura del proyecto	4	20	X						X					
002	Definición y modelo de la base de datos	6	30	X						X					
003	Definición y modelo de servicios para la página web	6	30	X						X					
004	Despliegue de la base de datos	2	10	X						X					
005	Despliegue de los servicios	2	10	X						X					
006	Diseño del layout Admin	3	15		X					X					
007	Página inicio de sesión	8	40		X					X					
008	Diseño del MenuSider	2	10		X					X					
009	Diseño del Menutop	2	10		X					X					
010	Diseño del modal y vistas del componente usuario	5	25		X					X					
011	Diseño del modal y vistas del componente rol	3	15			X					X				
012	Diseño del modal y vistas del componente Persona	3	15			X					X				
013	Diseño del modal y vistas del componente Buses	3	15			X					X				
014	Diseño del modal y vistas del componente Tarjetas	3	15			X					X				

	dispositivo recargas y Cobro de pasajes																
028	Inicio de sesión en la aplicación UnionApp	6	30					X									X
029	Diseño del componente Register de la App	3	15						X								X
030	Diseño del menú de la App	2	10						X								X
031	Diseño del componente home para la App	3	15						X								X
032	Diseño del componente Pagos para la App	3	15						X								X
033	Diseño del componente Recargas para la App	3	15						X								X
034	Diseño del componente Mi cuenta para la App	4	20						X								X
035	Pruebas de funcionalidad de la App UnionApp	2	10						X								X

Tabla 141 Plan de entrega de la aplicación

Elaborado por: Bryan Amaguaña

3.2.3.2 Iteraciones

Ya establecido las historias de usuario con sus respectivas iteraciones se presenta un plan los estados y prioridades del proceso de desarrollo de la aplicación presentado en la **tabla 142**.

	N°	Historia de Usuario	Prioridad	Riesgo	Estado de desarrollo	Prueba
Primera Iteración	001	Estructura del proyecto	Alto	Alto	Completada	Aprobada
	002	Definición y modelo de la base de datos	Alto	Alto	Completada	Aprobada

	003	Definición y modelo de servicios para la página web	Alto	Alto	Completada	Aprobada
	004	Despliegue de la base de datos	Medio	Medio	Completada	Aprobada
	005	Despliegue de los servicios	Medio	Medio	Completada	Aprobada
Segunda Iteración	006	Diseño del layout Admin	Medio	Medio	Completada	Aprobada
	007	Página inicio de sesión	Alto	Alto	Completada	Aprobada
	008	Diseño del MenuSider	Medio	Medio	Completada	Aprobada
	009	Diseño del Menutop	Alto	Medio	Completada	Aprobada
	010	Diseño del modal y vistas del componente usuario	Alto	Alto	Completada	Aprobada
Tercera iteración	011	Diseño del modal y vistas del componente rol	Alto	Alto	Completada	Aprobada
	012	Diseño del modal y vistas del componente Persona	Alto	Alto	Completada	Aprobada
	013	Diseño del modal y vistas del componente Buses	Alto	Alto	Completada	Aprobada
	014	Diseño del modal y vistas del componente Tarjetas	Alto	Alto	Completada	Aprobada
	015	Diseño del modal y vistas del componente Tipo_Pasajero	Alto	Alto	Completada	Aprobada
	016	Diseño del modal y vistas del componente Pasajero	Alto	Alto	Completada	Aprobada
	017	Diseño del modal y vistas del componente Cobro_Pasaje	Alto	Alto	Completada	Aprobada

Cuarta Iteración	018	Diseño del modal y vistas del componente Recargas	Alto	Alto	Completada	Aprobada
	019	Diseño del modal y vistas del componente Menu_Web	Medio	Medio	Completada	Aprobada
	020	Diseño del modal y vistas del componente Contenido_Web	Medio	Medio	Completada	Aprobada
	021	Diseño de la página de inicio de Contenido_Web	Medio	Medio	Completada	Aprobada
	022	Diseño página de inicio del sistema	Bajo	Bajo	Completada	Aprobada
	023	Definición y modelo de los servicios para el dispositivo lector Tarjetas NFC	Alto	Alto	Completada	Aprobada
	024	Definición y modelo de los servicios para el dispositivo recargador de tarjetas NFC	Alto	Alto	Completada	Aprobada
	Quinta iteración	025	Diseño del primer prototipo Lector de tarjetas NFC	Alto	Alto	Completada
026		Diseño del primer prototipo Recargas Tarjeta NFC	Alto	Alto	Completada	Aprobada
027		Pruebas de funcionalidad del dispositivo recargas y Cobro de pasajes	Alto	Medio	Completada	Aprobada
028		Inicio de sesión en la aplicación UnionApp	Alto	Alto	Completada	Aprobada
Sexta Iteración	029	Diseño del componente Register de la App	Alto	Alto	Completada	Aprobada

	030	Diseño del menú de la App	Medio	Medio	Completada	Aprobada
	031	Diseño del componente home para la App	Alto	Alto	Completada	Aprobada
	032	Diseño del componente Pagos para la App	Alto	Medio	Completada	Aprobada
	033	Diseño del componente Recargas para la App	Alto	Medio	Completada	Aprobada
	034	Diseño del componente Mi cuenta para la App	Alto	Alto	Completada	Aprobada
	035	Pruebas de funcionalidad de la App UnionApp	Alto	Alto	Completada	Aprobada

Tabla 142 Planificación de iteraciones

Elaborado por: Bryan Amaguaña

3.2.4 Fase IV: Producción

Basado en la planificación de iteraciones se desarrolla el proyecto de acuerdo con las historias de usuario que pertenece a cada iteración.

3.2.4.1 Iteración I

	N°	Historia de Usuario	Tiempo estimado	
			Días	Horas
Primera Iteración	001	Estructura del proyecto	3	15
	002	Definición y modelo de la base de datos	8	40
	003	Definición y modelo de servicios para la página web	2	10
	004	Despliegue de la base de datos	2	10
	005	Despliegue de los servicios	5	25

Tabla 143 Iteración I

Elaborado por: Bryan Amaguaña

Estructura del proyecto

Para la estructura del proyecto se estableció los recursos tanto del software como del hardware describiéndolos detalladamente a continuación.

Hardware

- 1 laptop MacBook pro 16gb de memoria RAM, core i7 de séptima generación, Sistema operativo macOS Big Sur versión 11.4.
- 1 Smartphone Samsung s6 edge plus, 4gb de Memoria RAM (Random Access Memory), version de android 7.0

Software

El framework para el desarrollo del presente proyecto de investigación es React basado en MERN Stack, dado así por los nombres de los componentes que fueron seleccionados para la creación del software, cada componente será detallado los pasos de instalación a continuación.

Instalación Node js

Para la instalación de Node js se debe realizar desde la página oficial de Node js <https://nodejs.org/es/> donde se descargar todos los recursos necesarios y procederá a instalador en el dispositivo, como complemento para la continuación de la instalación también se incluye el paquete npm que será tomado en cuenta para la instalación de los complementos adicionales.

Instalación yarn

Yarn es un gestor dependiente de JavaScript, se caracteriza por la fluidez y rapidez que otros gestores como npm, pero para su instalación es necesario realizarlo a través de npm por medio de la consola de comando.


```
npm install --global yarn
```

Figura 28: Comando para la instalación de yarn

Elaborado por: Bryan Amaguaña

▪ Instalación de MongoDB

Para la base de datos se puede optar por 2 opciones, gracias a las herramientas de MongoDB podemos realizar nuestra base de datos de forma local o desde la nube, para crear nuestra base de forma local debemos descargar los archivos necesarios desde la página oficial de MongoDB <https://www.mongodb.com/try/download>, si deseamos conectar nuestro proyecto desde la nube lo ponemos realizar atreves de mongo atlas, donde crearemos nuestro proyecto y desplegaremos nuestra base de datos

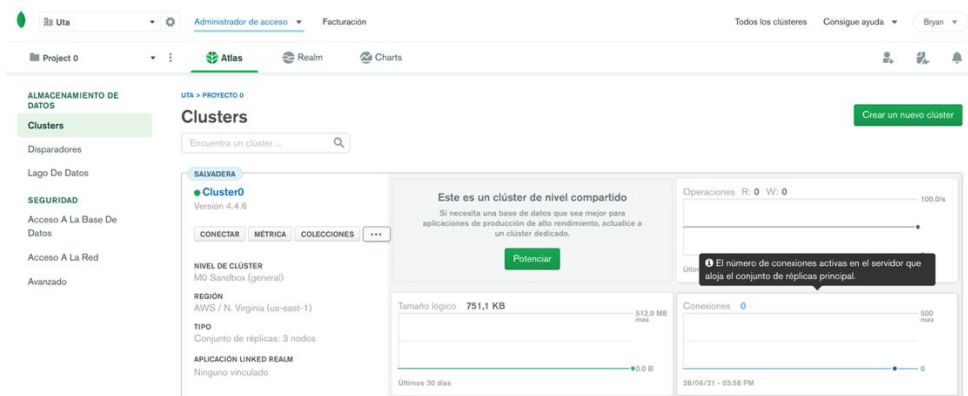


Figura 29: Pagina administrado de Mongo Atlas

Elaborado por: Bryan Amaguaña

▪ Instalación de React

Para la instalación de React es necesario la ejecución en consola del respectivo comando con la utilización del gestor de paquetes yarn.

```
yarn add create-react-app
```

Figura 30: Comando para la instalación de React

Elaborado por: Bryan Amaguaña

- **Instalación de Express**

Para la instalación de Express se debe ejecutar el siguiente comando en la consola de nuestro terminal con la ayuda del gestor de paquetes yarn, las versiones pueden variar de acuerdo con las actualizaciones de Express.

```
yarn add express
```

Figura 31: Comando para la instalación de Express

Elaborado por: Bryan Amaguaña

Editor de código fuente

Para el desarrollo del proyecto se necesita de un editor de código que permita la implementación de librerías para un entorno de trabajo más confortable, es por esta razón que el editor de código elegido es Visual Code.

Visual Code permite la gestión de librerías, componentes y extensiones que brinda un entorno de trabajo entendible y manejable para el programador, su instalación se lo realiza por la página oficial de Visual Code “<https://code.visualstudio.com/download>”

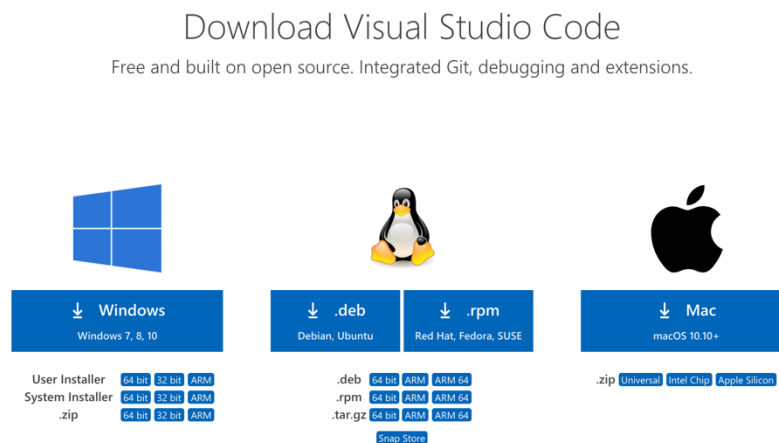


Figura 32: Página oficial para la instalación de Visual Code

Elaborado por: Bryan Amaguaña

Ya instalado el editor de texto procedemos a implementar las extensiones necesarias para configurar nuestro entorno de trabajo, entre ellas están:

- Bracket Pair Color
- ES7 React/Redux snippets
- ES7 React/Redux/GraphQL/React-Native snippets
- HTML CSS Support
- HTML snippets
- IntelliSense for CSS class names in HTML
- JavaScript (ES6) code snippets
- JavaScript and TypeScript Nightly
- Material Icon Theme
- Path IntelliSense
- Rainbow CSV
- React Native Snippets
- Simple React Snippets
- Todo tree
- TypeScript Importer

Todas las extensiones mencionadas pueden ser instaladas en el apartado de extensiones de visual code como se puede apreciar en la **figura 33**

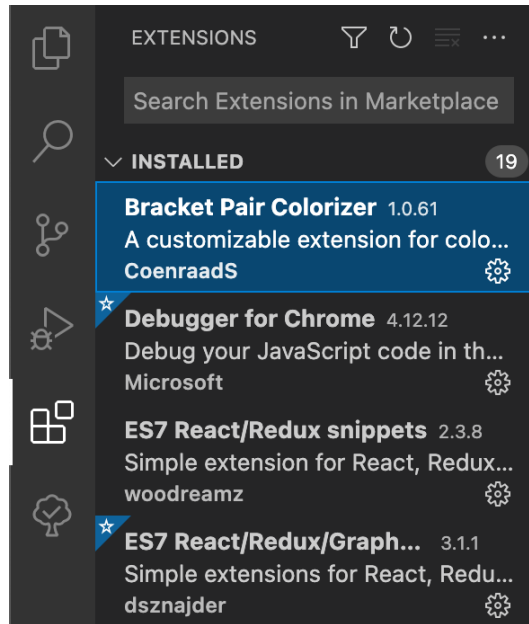


Figura 33: Extensiones de Visual Studio Code

Elaborado por: Bryan Amaguaña

Creación del proyecto.



Figura 34: Comando para la creación del proyecto Unión

Elaborado por: Bryan Amaguaña

Al ejecutar esta línea de comando en nuestro editor de código nos desplegará múltiples opciones como qué tipo de proyecto debemos crear, el nombre del autor, entre otras preguntas de información con el fin de completar la creación del proyecto, posteriormente ya creado el proyecto exitosamente es recomendable ejecutarlo antes de iniciar con nuestra programación para comprobar que nuestro proyecto se ejecute correctamente, cabe mencionar que primero debemos dirigirnos a la dirección de nuestra carpeta donde se encuentra nuestro proyecto.



Figura 35: Comando para ejecutar un proyecto en React

Elaborado por: Bryan Amaguaña

En la carpeta de nuestro proyecto se presenta múltiples carpetas con diferentes funciones dentro del proyecto como la **figura 36**

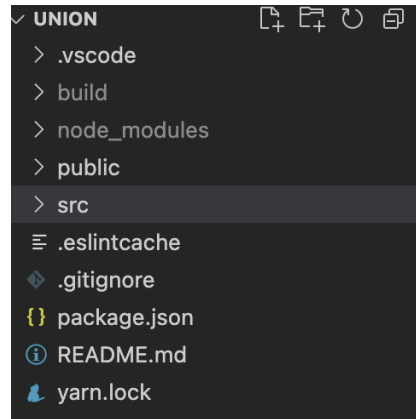


Figura 36: Estructura del proyecto Unión

Elaborado por: Bryan Amaguaña

- **.vscode:** Contiene el archivo launch.json, que son las configuraciones creadas automáticamente cuando se realiza un debug en nuestro proyecto guardando su versión y configuraciones del programa.
- **Node_modules:** Es el espacio donde se instala todas las dependencias de manera temporal, declaradas en el package.json.
- **Public:** Posee los archivos estáticos de la aplicación como el logo, icono entre otros archivos que pueden ser modificados de acuerdo con las exigencias de la aplicación.
- **Src:** Posee toda la aplicación, es el lugar donde el programador implementa todos los componentes, métodos o rutas de la aplicación.
- **.gitignore:** Es un archivo de texto donde se describe todos los componentes de nuestra aplicación que no deben ser tomados en cuenta para el despliegue en una plataforma o cuando deseemos realizar respaldos.
- **Package.json:** Posee el nombre de la aplicación, versión y todas las dependencias con las que trabaja la aplicación.
- **README.md:** Posee la información que se desplegará en la pantalla principal de nuestro repositorio al momento de subirlo a una plataforma
- **Yarn.lock:** Posee todas las dependencias de la aplicación.

Definición y modelo de la base de datos

MongoDB al ser una base no relacionada no permite las relaciones entre tablas, pero adiciona un campo para poder agregar objetos direccionados simulando una relación entre tablas de esta forma permite obtener objetos completos por parte del .id de cada colección.

En la **figura 37** se aprecia el modelo de colecciones que se aplicó en el presente proyecto de investigación, al ser NoSql no se puede representar como tablas entidad relación sino como un campo de objeto dentro de cada colección.

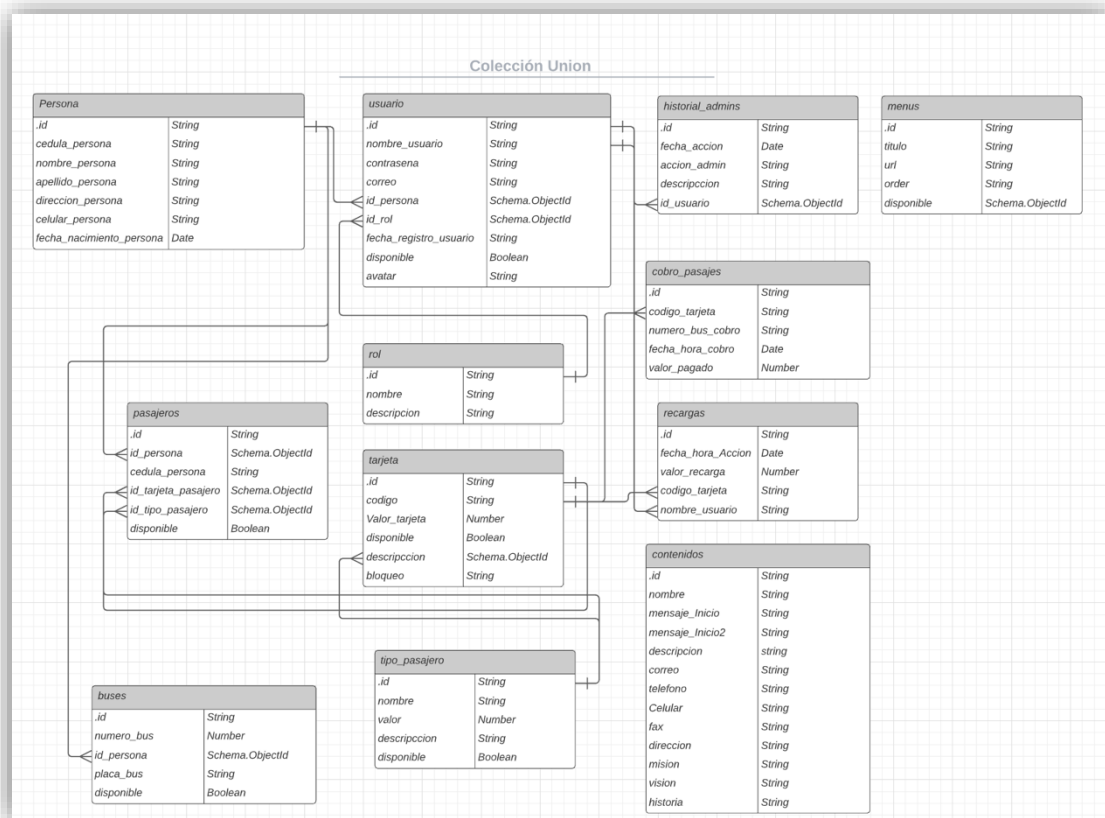


Figura 37: Modelo de la base de datos

Elaborado por: Bryan Amaguaña

Definición y modelo de servicios para la página web

Ya definido la base de datos se establece la conexión con el back-End que será desarrollado en Node js juntamente con Express, donde se realizará todas las peticiones, inserciones o supresiones de información por parte del Front-End, adicional a estos métodos se establecerá la creación de tokens para el ingreso de usuarios a la aplicación, así también este compuesto por método lógicos que deberá realizar la aplicación devolviendo datos en json para la interpretación del Font-End.

Cada módulo de colección estará conformado por una clase donde se definan las sentencias CRUD así también como las rutas y los métodos necesarios para la obtención de datos de la colección.

Despliegue de la base de datos

Ya estructura la base de datos de forma local, se tomó la decisión de realizar pruebas del comportamiento que tendrá en la nube por lo cual se optó por la utilización del Mongo Atlas, una plataforma gratuita para el alojamiento.

Se establece el modelo de la colección en los servicios y al momento de realizar la conexión se establece el enlace que nos refleja mongo Atlas, de tal forma que se establezca la conexión con el servicio, adicional a este proceso de debe dar acceso a la dirección ip de donde se va a realizar la conexión este paso se puede configurar en mongo Atlas, de tal forma que quede habilitado la ip de la computadora donde vamos a realizar la conexión.

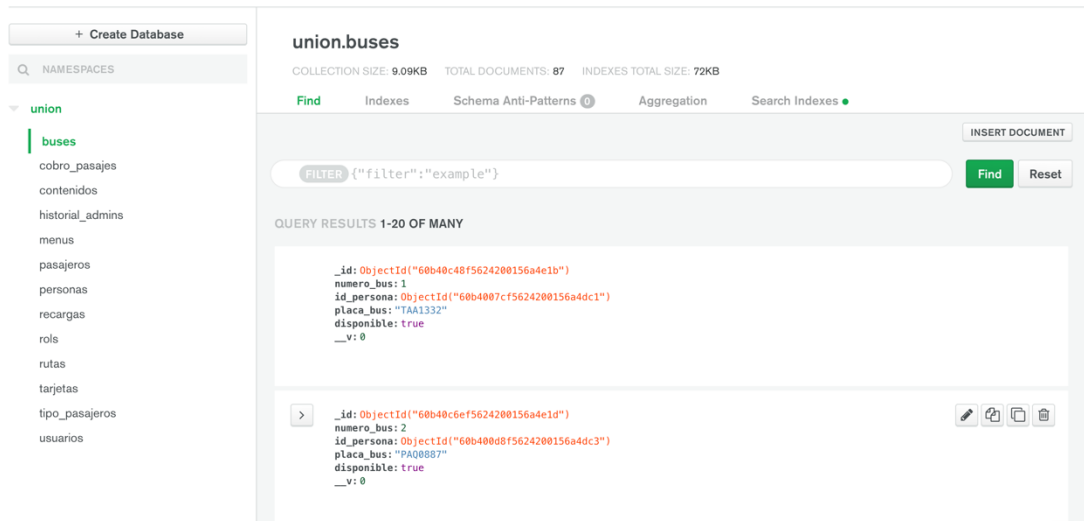


Figura 38: Mongo Atlas desplegado la base de datos

Elaborado por: Bryan Amaguaña

Despliegue de los servicios

Ya establecido los servicios en Node js de forma local se procede a subir a un host gratuito con la finalidad de realizar pruebas CRUD de la obtención de datos, como está en etapa de desarrollo la plataforma seleccionada es Heroku.

Heroku es una plataforma gratuita muy utilizada por desarrolladores para el alojamiento de aplicaciones tanto web como servicios, su diseño y su interfaz es muy sencilla lo que facilito la implementación de los servicios en la web como se puede apreciar en la **figura 39**

Personal > servicios-union

GitHub BryanAmaguaña/Servicios main

Overview Resources Deploy Metrics Activity Access Settings

Get a complete visualization of your app in a team-based continuous delivery environment with [Heroku Pipelines](#). Hide Create a Heroku Pipeline

Installed add-ons 50.00/month [Configure Add-ons](#)

There are no add-ons for this app. You can add add-ons to this app and they will show here. [Learn more](#)

Dyno formation 50.00/month [Configure Dynos](#)

This app is using free dynos

web npm start ON

Collaborator activity [Manage Access](#)

fol.brn@gmail.com 54 deploys

Latest activity [All Activity](#)

- fol.brn@gmail.com: Deployed 728ddc4a May 31 at 2:54 PM · v56 · [Compare diff](#)
- fol.brn@gmail.com: Build succeeded May 31 at 2:54 PM · [View build log](#)
- fol.brn@gmail.com: Deployed 728ddc4a May 31 at 2:47 PM · v55 · [Compare diff](#)
- fol.brn@gmail.com: Build succeeded May 31 at 2:46 PM · [View build log](#)
- fol.brn@gmail.com: Deployed a0cb8f10 May 27 at 12:45 PM · v54 · [Compare diff](#)
- fol.brn@gmail.com: Build succeeded May 27 at 12:45 PM · [View build log](#)
- fol.brn@gmail.com: Deployed a0cb8f10 May 27 at 12:42 PM · v53 · [Compare diff](#)

Figura 39: Panel de Actividades de Heroku.

Elaborado por: Bryan Amaguaña

Untitled Request BUILD

GET <https://servicios-union.herokuapp.com/Tarjeta> Send Save

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (12) Test Results Status: 200 OK Time: 539 ms Size: 891 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "tarjeta": [
3     {
4       "_id": "60b5508549a33400152136bb",
5       "codigo": "0aa6b535",
6       "valor_tarjeta": 1,
7       "disponible": false,
8       "bloqueo": "Tarjeta Modificada, Valor de la Tarjeta: 0.70",
9       "descripcion": "604e6c86e20cbe0f62d9d35d",
10      "__v": 0
11    },
12    {
13      "_id": "60b550ed49a33400152136bf",
14      "codigo": "d04fc270",
15      "valor_tarjeta": 32.92,
16      "disponible": true,
17      "bloqueo": "",
18      "descripcion": "604e6c86e20cbe0f62d9d35d",
19      "__v": 0
20    }
21  ]
22 }

```

Figura 40: Sentencia GET de la colección Tarjetas en postman.

Elaborado por: Bryan Amaguaña.

3.2.4.2 Iteración II

	N°	Historia de Usuario	Tiempo estimado	
			Días	Horas
Segunda	006	Diseño del layout Admin	4	20
	007	Página inicio de sesión	6	30
	008	Diseño del MenuSider	6	30
	009	Diseño del Menutop	2	10
	010	Diseño del modal y vistas del componente usuario	2	10

Tabla 144 Iteración II

Elaborado por: Bryan Amaguaña

Diseño del layout Admin

La aplicación está compuesta por 2 layout, uno diseñado para la administración de la aplicación y otro layout diseñado para las personas al público donde estará la información de la cooperativa de transporte urbano Unión Ambateña.

El layout de la administración del sistema está compuesto por 3 partes que es MenuSider, Menutop y un espacio diseñado para las vistas de datos como se puede apreciar en la **figura 41**

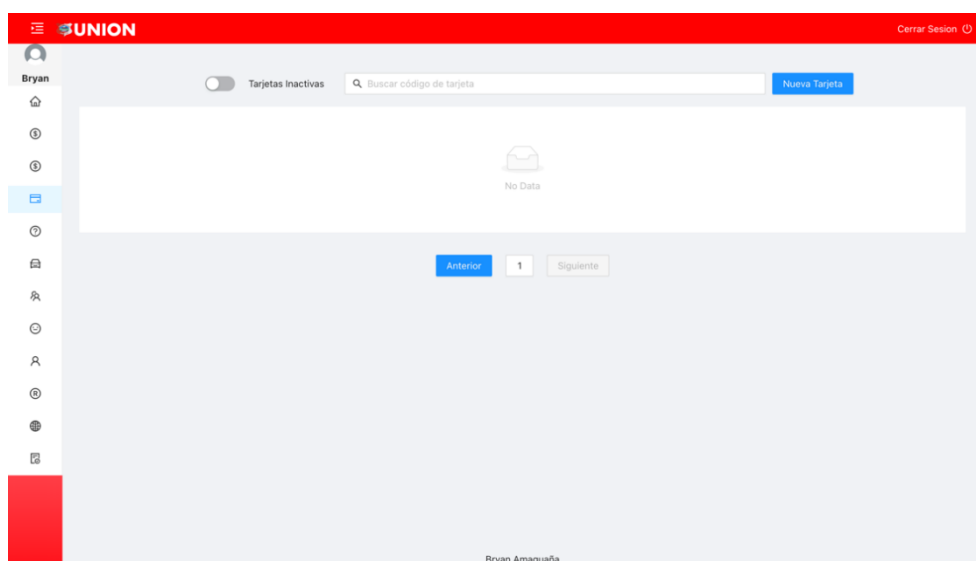


Figura 41: Layout Admin

Elaborado por: Bryan Amaguaña

Página de Inicio de sesión

Página de inicio de sesión para usuarios ya autenticados en el sistema, la forma de ingreso al sistema es a través del nombre de usuario y una contraseña, si el usuario o la contraseña no coinciden con las credenciales de la base de datos se desplegará un mensaje informando al usuario que credencial es incorrecto de la misma forma si el usuario no está activo se le informara al usuario.

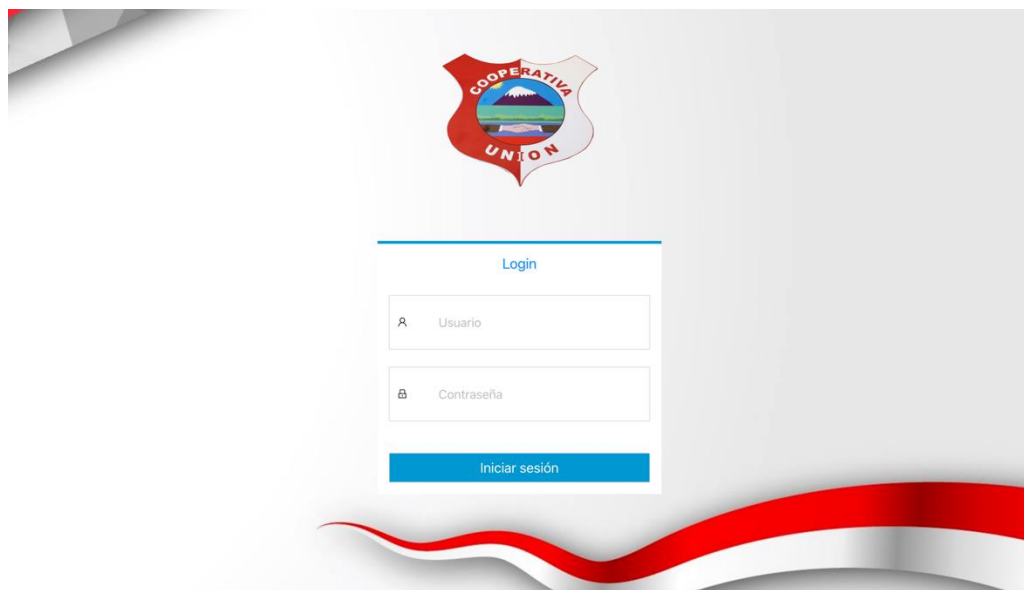


Figura 42: Página de Inicio de sesión

Elaborado por: Bryan Amaguaña

Diseño del MenuSider

Es el menú donde se despliega todos los componentes del sistema, diseñado de forma responsiva adaptándose a las necesidades del cliente, cuenta con un apartado para mostrar el nombre del usuario autenticado así también dispone de un espacio para la foto del usuario o del avatar, modificable desde el componente usuarios.

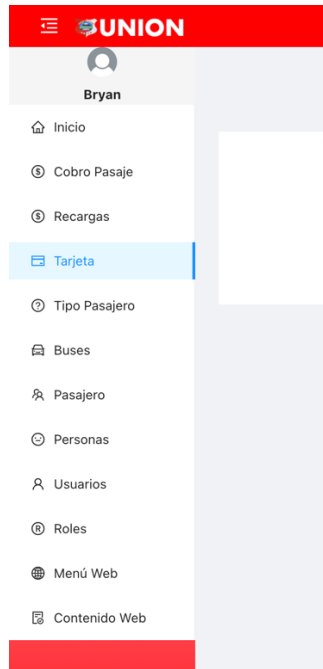


Figura 43: MenuSider

Elaborado por: Bryan Amaguaña

Diseño del Menutop

Localizado en la parte superior del layout Admin, está compuesto por 3 apartados que cumplen una función, el primer apartado es para el despliegue del MenuSider, para una mejor navegación entre componentes, el segundo apartado es para el icono de la empresa que a su vez también redirección a la página principal de la aplicación, y el tercer apartado es para cerrar sesión que cuenta con un icono para una mejor apariencia y a su vez regresa a la página de inicio de sesión.



Figura 44: MenuTop

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Usuario

La vista de usuarios está compuesto por un ListItem que despliega la información de cada usuario así también incorpora opciones para editar o desactivar un usuario, está diseñado para desplegar 4 usuarios a la vez, paginado desde el Back-End y manejado por un apartado de navegación en la parte inferior indicando en que pestaña se encuentra, adicional a esto cuenta con una búsqueda dinámica basado en el nombre del usuario y con un espacio dedicado para los usuarios que fueron desactivados por el administrador como se muestra en la **figura 45**

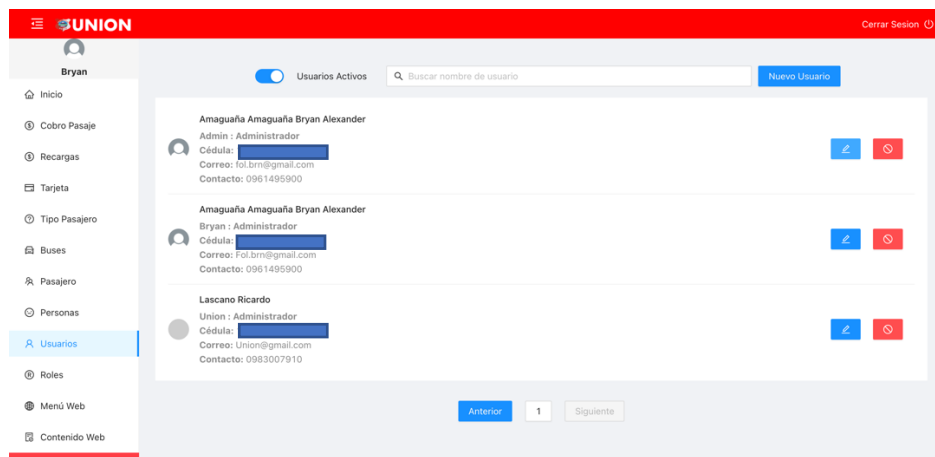


Figura 45: Componente Usuario

Elaborado por: Bryan Amaguaña

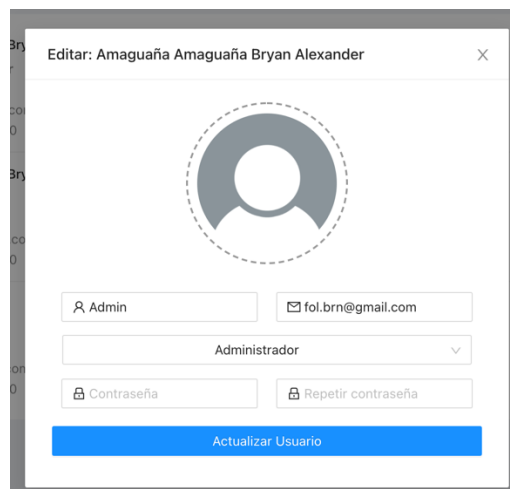


Figura 46: Modal para editar un usuario

Elaborado por: Bryan Amaguaña

3.2.4.3 Iteración III

	N°	Historia de Usuario	Tiempo estimado	
			Días	Horas
tercera Iteración	011	Diseño del modal y vistas del componente rol	3	15
	012	Diseño del modal y vistas del componente Persona	3	15
	013	Diseño del modal y vistas del componente Buses	3	15
	014	Diseño del modal y vistas del componente Tarjetas	3	15
	015	Diseño del modal y vistas del componente Tipo_Pasajero	3	15
	016	Diseño del modal y vistas del componente Pasajero	3	15
	017	Diseño del modal y vistas del componente Cobro_Pasaje	2	10

Tabla 145 Iteración III

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente rol

La vista de rol está compuesta por un ListItem que despliega toda la información de los roles registrados incorporando opciones para editar un rol tal como lo muestra en la **figura 47**

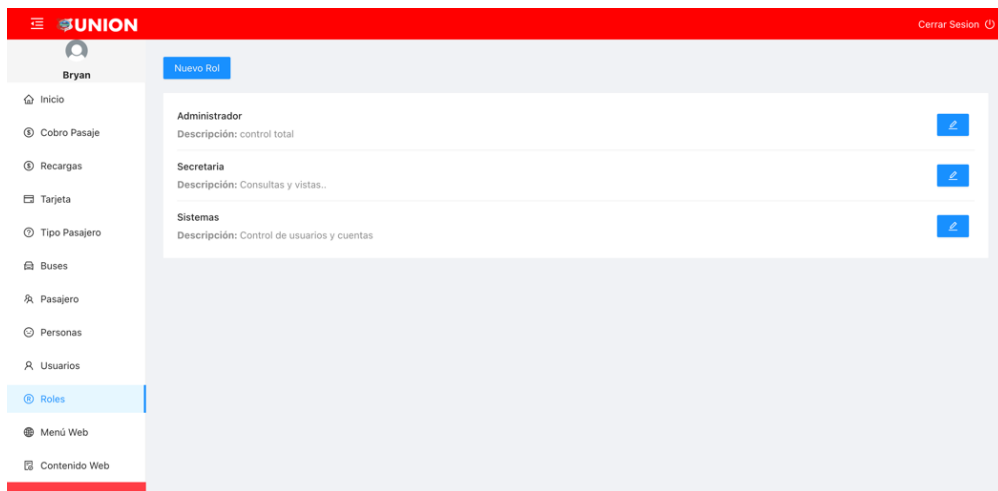


Figura 47: Componente Rol

Elaborado por: Bryan Amaguaña

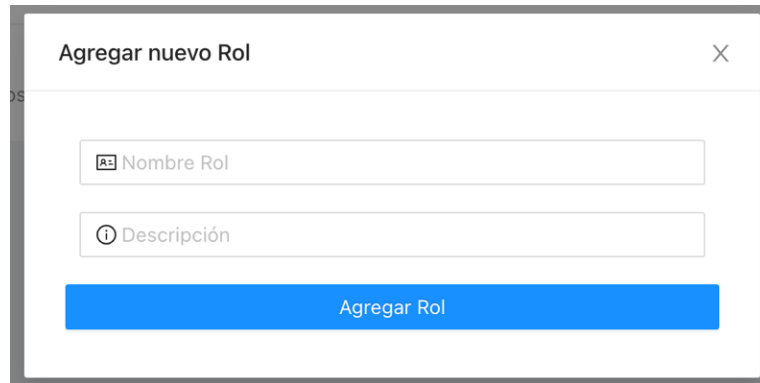


Figura 48: Modal para agregar un rol

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Persona

La vista de personas está compuesta por un ListItem que despliega la información de cada persona así también incorpora opciones para editar, está diseñado para desplegar 4 personas a la vez, paginado desde el Back-End y esta manejado por un apartado de navegación en la parte inferior indicando en que pestaña se encuentra, adicional a esto cuenta con una búsqueda dinámica basado en la cedula tal como se muestra en la **figura 49**

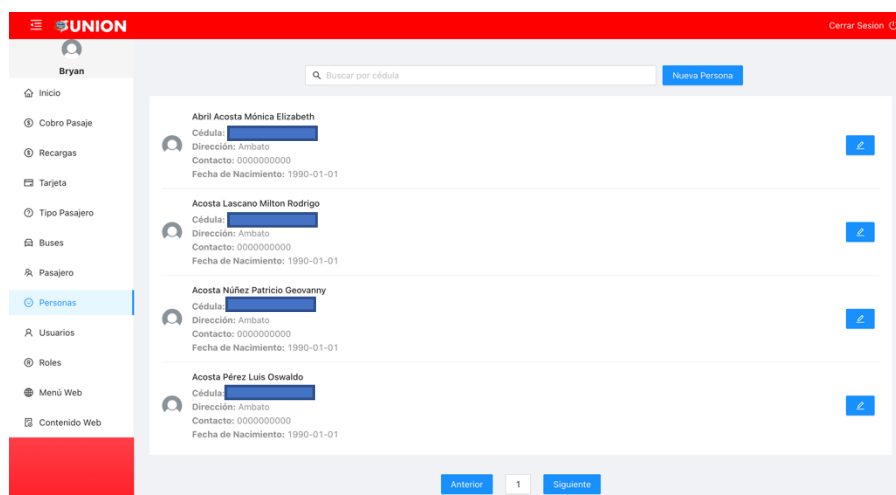


Figura 49: Componente Persona

Elaborado por: Bryan Amaguaña

Modal titled "Agregar nueva persona" with a close button (X) in the top right corner. The form contains the following fields:

- Cédula (with a person icon)
- Fecha de Nacimiento (with a calendar icon)
- Nombre (with a person icon)
- Apellido (with a person icon)
- Dirección (with a location pin icon)
- Celular (with a phone icon)

Below the fields, there is a red link labeled "Verificar Cédula" and a blue button labeled "Agregar Persona".

Figura 50: Modal para agregar una Persona

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Buses

La vista de Buses está compuesta por un ListItem que despliega la información de cada bus así también incorpora opciones para editar y desactivar un bus, está diseñado para desplegar 4 buses a la vez, paginado desde el Back-End y esta manejado por un apartado de navegación en la parte inferior indicando en que pestaña se encuentra, adicional a esto cuenta con una búsqueda dinámica basado en el número del bus tal como se muestra en la **figura 51**

The screenshot shows the "UNION" web application interface. The top navigation bar is red with the "UNION" logo and a "Cerrar Sesión" button. Below it, there's a user profile for "Bryan" and a "Buses Activos" toggle. A search bar labeled "Buscar numero de Bus" is present, along with a "Nuevo Bus" button. The main content area displays a list of 4 active buses, each with a bus icon, a title "Bus Número: X", and detailed information: owner name, ID (Cédula), address (Dirección), phone number (Celular), and license plate (Placas). Each bus entry has a blue edit icon and a red delete icon. The sidebar on the left contains navigation items: Inicio, Cobro Pasaje, Recargas, Tarjeta, Tipo Pasajero, Buses (highlighted), Pasajero, Personas, Usuarios, Roles, Menú Web, and Contenido Web. At the bottom, there are pagination controls: "Anterior", "1", and "Siguiente".

Figura 51: Componente Buses

Elaborado por: Bryan Amaguaña



Figura 52: Modal para agregar un Bus

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Tarjetas

La vista de Tarjetas está compuesta por un ListItem que despliega la información de cada tarjeta así también incorpora opciones para editar y desactivar una tarjeta, está diseñado para desplegar 4 tarjetas a la vez, paginado desde el Back-End y esta manejado por un apartado de navegación en la parte inferior indicando en que pestaña se encuentra, adicional a esto cuenta con una búsqueda dinámica basado en el código de la tarjeta tal como se muestra en la **figura 53**

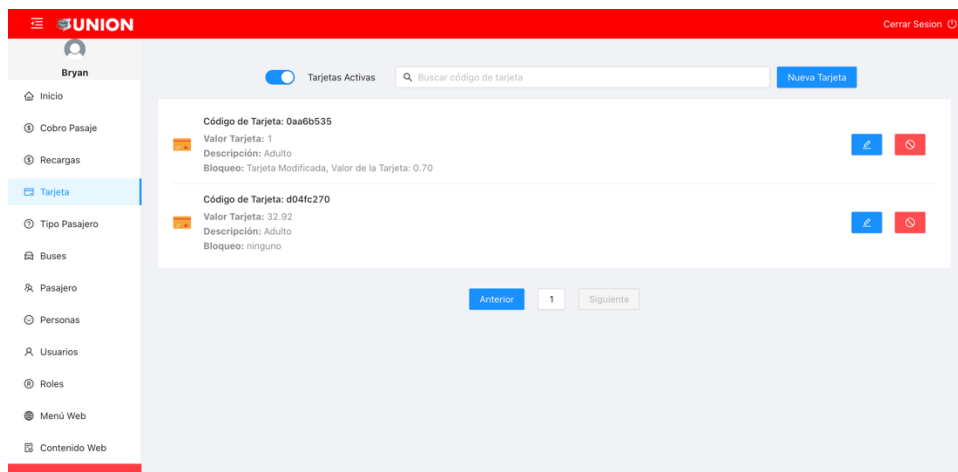


Figura 53: Componente Tarjetas

Elaborado por: Bryan Amaguaña



Figura 54: Modal para agregar una tarjeta

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Tipo_Pasajero

La vista Tipo_Pasajero está compuesto por un ListItem que despliega la información de cada tipo de pasajero, así también incorpora opciones para editar y desactivar un tipo de pasajero tal como se muestra en la **figura 55**

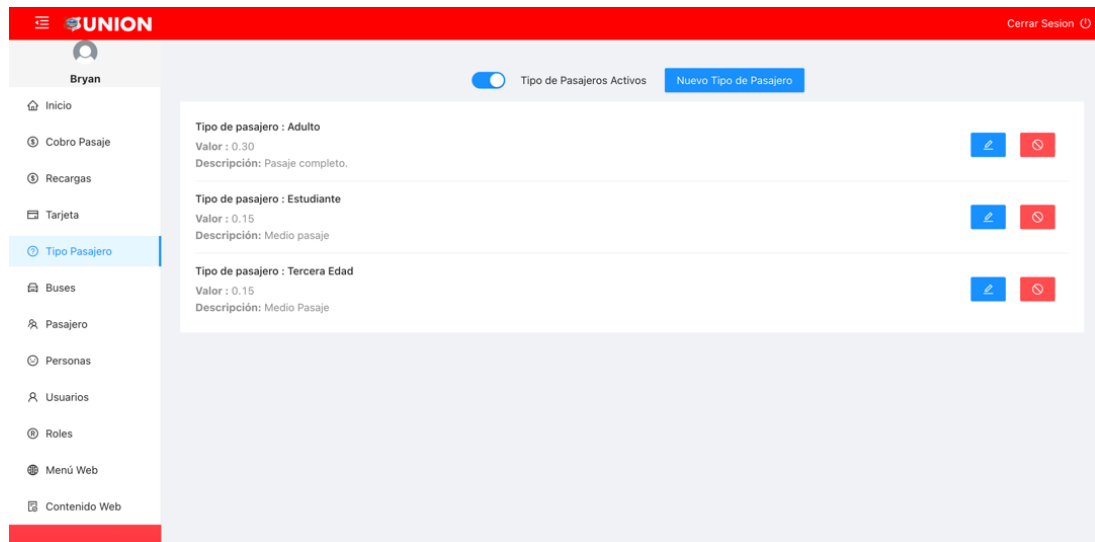


Figura 55: Componente Tipo_Pasajero

Elaborado por: Bryan Amaguaña

Figura 56: Modal para agregar un Tipo_Pasajero

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Pasajero

La vista de Pasajeros está compuesto por un ListItem que despliega la información de cada pasajero así también incorpora opciones para editar y desactivar una pasajero, está diseñado para desplegar 4 pasajeros a la vez, paginado desde el Back-End y esta manejado por un apartado de navegación en la parte inferior indicando en que pestaña se encuentra, adicional a esto cuenta con una búsqueda dinámica basado en el código de la tarjeta y en la cedula de la persona tal como se muestra en la **figura 57**

Figura 57: Componente Pasajero

Elaborado por: Bryan Amaguaña

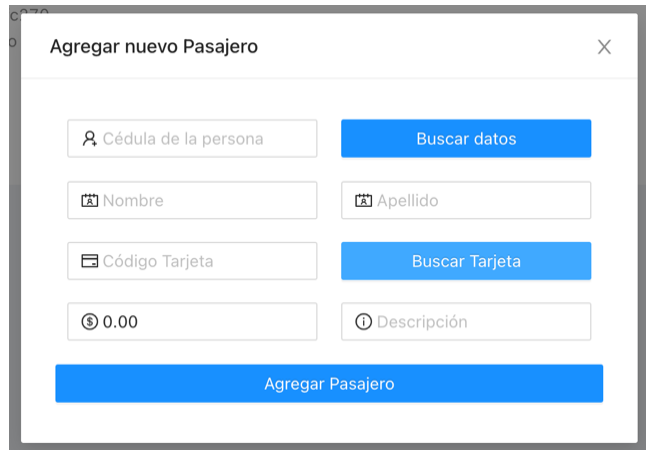


Figura 58: Modal para agregar un Pasajero

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Cobro_Pasaje

La vista Cobro_Pasaje está compuesto por un ListItem que despliega la información de cada cobro de pasaje realizado desde el dispositivo lector de tarjetas NFC, cuenta con una búsqueda basado en la fecha de inicio, de fin y el número de bus tal como se muestra en la **figura 59**

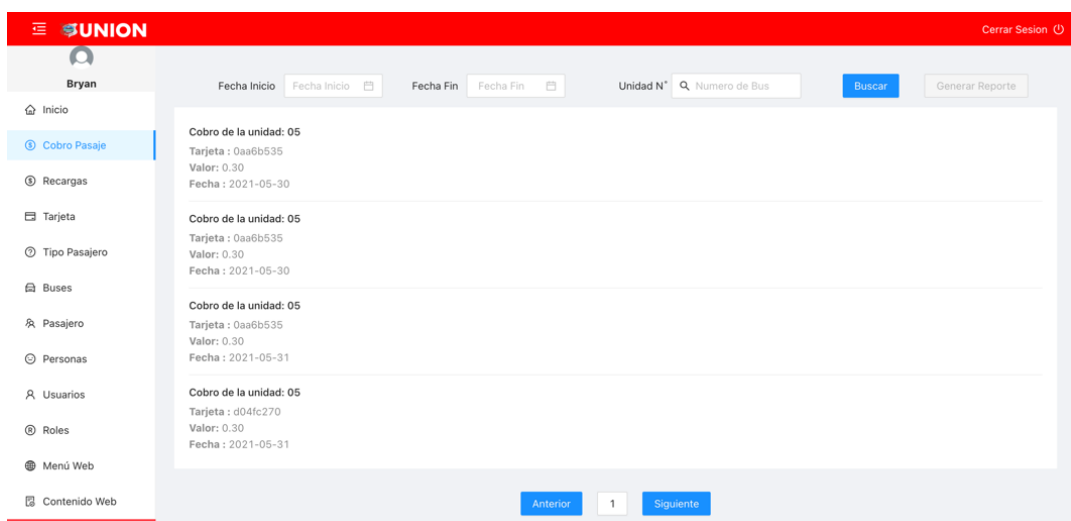


Figura 59: Componente Cobro_Pasaje

Elaborado por: Bryan Amaguaña

3.2.4.4 Iteración IV

	N°	Historia de Usuario	Tiempo estimado	
			Días	Horas
Cuarta Iteración	018	Diseño del modal y vistas del componente Recargas	3	15
	019	Diseño del modal y vistas del componente Menu_Web	3	15
	020	Diseño del modal y vistas del componente Contenido_Web	3	15
	021	Diseño de la página de inicio de Contenido_Web	3	15
	022	Diseño página de inicio del sistema	4	20
	023	Definición y modelo de los servicios para el dispositivo lector Tarjetas NFC	2	10
	024	Definición y modelo de los servicios para el dispositivo recargador de tarjetas NFC	2	10

Tabla 146 Iteración IV

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Recargas

La vista Recargas está compuesto por un ListItem que despliega la información de cada recarga de tarjetas realizado desde el dispositivo lector y escritor de tarjetas NFC, cuenta con una búsqueda basado en la fecha de inicio, de fin y el nombre del usuario que realizo la recarga, tal como se muestra en la **figura 60**

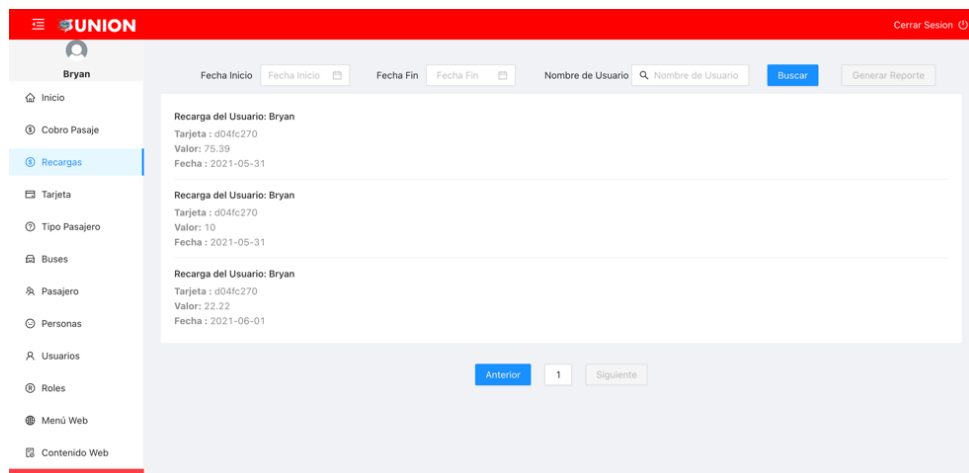


Figura 60: Componente recargas

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Menu_Web

La vista Menu_Web está compuesto por un ListItem que despliega la información de cada menú que compone el layout público, incorporando opciones para desactivar un menú o para editar el menú, tal como se muestra en la **figura 61**

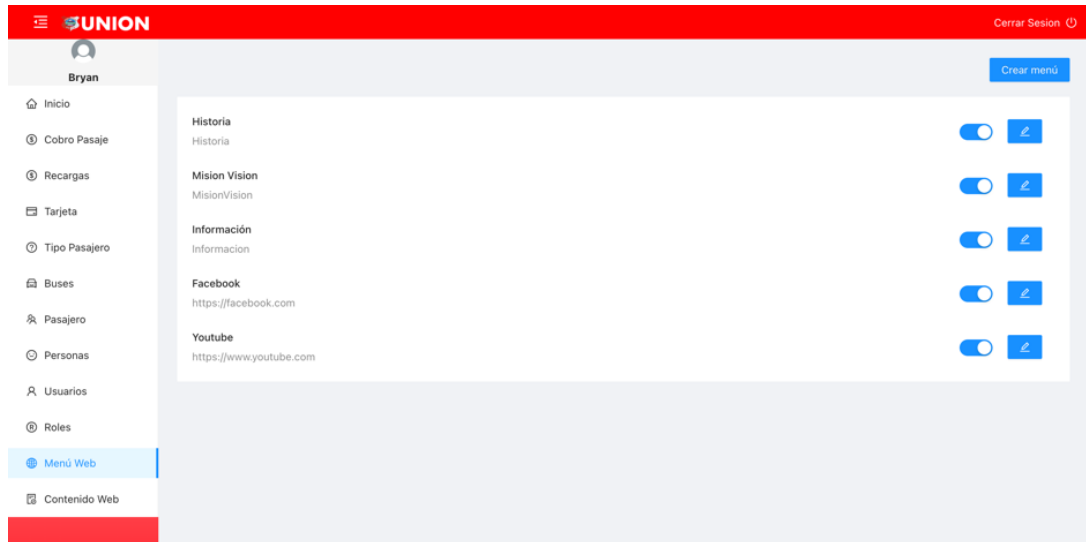


Figura 61: Componente Menu_Web

Elaborado por: Bryan Amaguaña

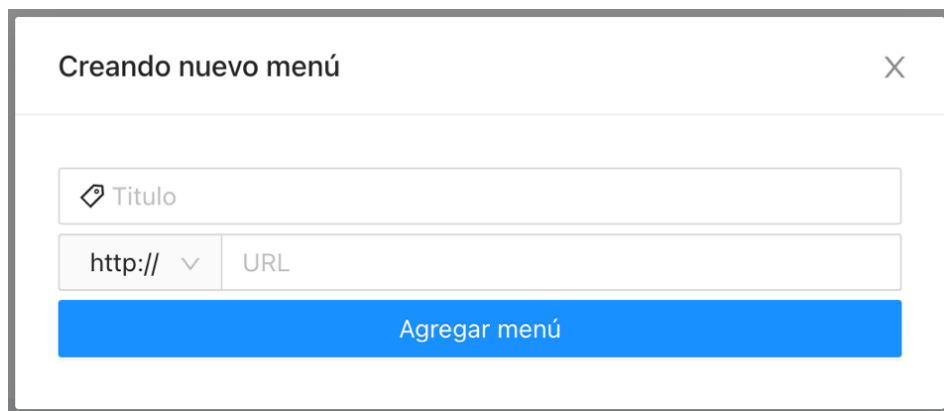


Figura 62: Modal para agregar un menú

Elaborado por: Bryan Amaguaña

Diseño del modal y vistas del componente Contenido_Web

La vista Contenido_Web está compuesto por cuadros de texto donde se encuentra la información que se despliega en el layout público, esta información es editable solo por el usuario que posee el rol de administrador, de igual forma al realizar algún cambio se despliega un mensaje de confirmación tal como se muestra en la **figura 63**

The screenshot shows a web application interface for 'UNION'. On the left is a sidebar menu with the user's name 'Bryan' at the top. The menu items are: Inicio, Cobro Pasaje, Recargas, Tarjeta, Tipo Pasajero, Buses, Pasajero, Personas, Usuarios, Roles, Menú Web, and Contenido Web (highlighted in blue). The main content area is a form for editing website content. It has a red header with the 'UNION' logo and a 'Cerrar Sesión' button. The form fields are: Cooperativa (Cooperativa de Transporte Urbano Union Am), Correo (Union0000@gmail.com), Dirección (Huachi Grande), Teléfono (034125423), Celular (0000000000), Fax (55555555), Mensaje de Inicio (La union hace la fuerza), Mensaje Secundario (Bienvenido), Descripción (Trabajamos de Domingo a Domingo), Misión (MISIÓN, la cooperativa de transporte urbano "UNIÓN AMBATEÑA", es una organización que realiza en forma permanente el servicio Urbano de transporte de pasajeros, en el cantón Ambato, y sus parroquias con unidades motorizadas en perfectas condiciones mecánicas, con confort y seguridad, manteniendo como principios entre sus asociados la IGUALDAD, SOLIDARIDAD, COMPAÑERISMO, y DEMOCRACIA), and Visión (VISIÓN, ser una operadora de Transporte Urbano de pasajeros, competitiva por excelencia, poniendo al servicio de la ciudadanía a la que sirve, sus modernas unidades, cómodas y confortables, dando un óptimo servicio a la colectividad, con RESPETO, SEGURIDAD, CONFORT, y EFICIENCIA, de acuerdo con los planes y políticas institucionales y disposiciones de los Organos de Tránsito competentes).

Figura 63: Componente Contenido_Web

Elaborado por: Bryan Amaguaña

Diseño de la página de inicio de Contenido_Web

Toda la información ingresada en el contenido web se despliega en el layout público, o en la página de inicio del sistema, que se muestra antes de ingresar sesión en la aplicación, de igual forma los ítems ingresados en el menú web se despliegan en el Menutop del layout público.



Figura 64: Layout publico

Elaborado por: Bryan Amaguaña

Diseño página de inicio del sistema

Compuesta por un gráfico de barras donde despliega todos los ingresos de los cobros de pasajes de los buses filtrados por fechas y disponible para el despliegue del reporte de acuerdo con los datos filtrados en el gráfico.



Figura 65: Componente home grafico de ingresos de Buses

Elaborado por: Bryan Amaguaña

COOPERATIVA DE TRANSPORTE URBANO UNION AMBATEÑA
Correo: union@uni.com
 Teléfono: 034125423 Celular: 093431276

Reporte mensual de Ingresos de Buses
 Del 01-01-2021 Al 16-05-2021

Total de ingresos \$ 22.80

Unidad #	Total pasajeros	Total Ingreso
01	2	\$ 0.30
02	2	\$ 0.30
03	1	\$ 0.15
04	2	\$ 0.30
05	3	\$ 0.45
06	27	\$ 6.45
07	1	\$ 0.15
08	1	\$ 0.15
09	1	\$ 0.15
10	3	\$ 0.75
11	1	\$ 0.15
12	1	\$ 0.15
13	1	\$ 0.15
14	1	\$ 0.15
15	1	\$ 0.15
16	1	\$ 0.30
17	1	\$ 0.15
18	1	\$ 0.15
19	1	\$ 0.15
20	3	\$ 0.60
21	1	\$ 0.15
22	1	\$ 0.15
23	1	\$ 0.15
24	1	\$ 0.15
25	1	\$ 0.15
26	1	\$ 0.15
27	1	\$ 0.15
28	1	\$ 0.15

Figura 66: Reporte generado por el grafico de barras de los ingresos de los buses

Elaborado por: Bryan Amaguaña

En el mismo componente se despliego un segundo grafico de barras donde se visualiza los ingresos de las recargas realizadas por los usuarios asignados, de igual forma está disponible la opción para generar reportes de todos los ingresos filtrados por fechas.

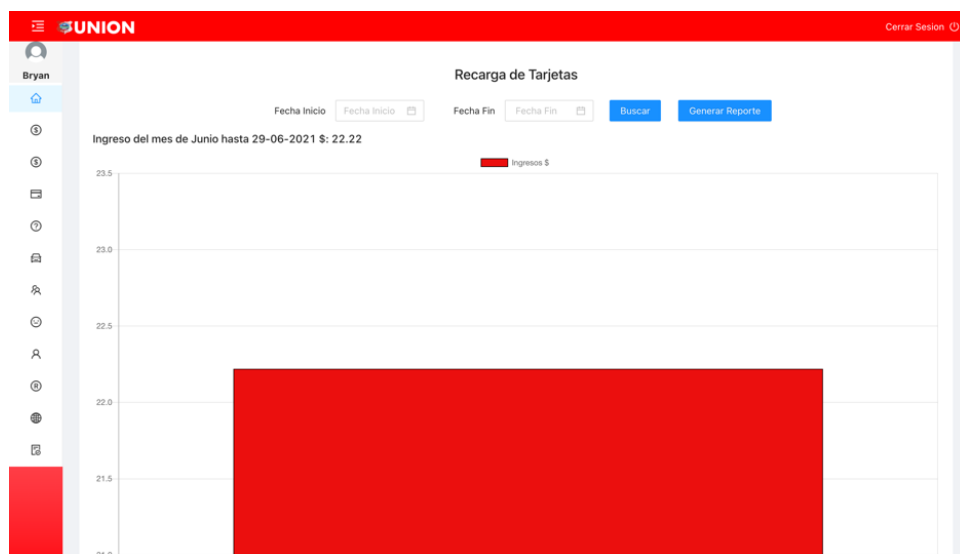


Figura 67: Componente home grafico de ingresos de Recargas

Elaborado por: Bryan Amaguaña

COOPERATIVA DE TRANSPORTE URBANO UNION AMBATEÑA		
Correo: union@gmail.com		
Teléfono: 034125423 Celular: 093451276		
Reporte mensual de Recargas		
Del 01/04/2021 Al 19/04/2021		
Total de ingresos \$ 95.93		
Usuario	Numero de Recargas	Total Ingresos
Usuario	2	\$ 1.30
bryan	79	\$ 94.63

Figura 68: Reporte generado por el grafico de barras de los ingresos de las recargas

Elaborado por: Bryan Amaguaña

Definición y modelo de los servicios para el dispositivo lector Tarjetas NFC

Los servicios para la interacción entre el dispositivo lector de tarjetas NFC con la base de datos está desarrollado en Node js, juntamente con Express que permite la conexión con la base de datos desarrollada en MongoDB, las peticiones de los servicios estarán basado a través del método GET, que permite enviar paramentos través del enlace con el fin de no enviar archivos json reduciendo el tiempo de envió de datos.

Al tratarse de dispositivos con conexión GSM su conexión no es muy estable por lo cual es necesario que las peticiones a la base de datos sean rápidas y no sobrecargadas de información innecesarias, partiendo de este concepto se optó por realizar todos los procesos lógicos en el dispositivo lector de tarjetas NFC y en los servicios.

Definición y modelo de los servicios para el dispositivo recargador de tarjetas NFC

Basado en los servicios del lector de tarjetas NFC se crea los servicios para realizar recargas a través del dispositivo lector y escritor de tarjetas NFC, de igual forma se toma como base los métodos GET que permite enviar parámetros a través del enlace reduciendo el tamaño de información de envió, al tratarse de un dispositivo multifuncional que permite las recargas de tarjetas, y permite inicializar tarjetas por un código, valor y tipo de tarjeta necesita más tiempo de respuesta que es definido en el dispositivo a diferencia del dispositivo lector de tarjetas que será utilizado solo para la realización del cobro del pasaje.

El dispositivo lector de tarjetas NFC cuenta con un Arduino mega, y una pantalla lcd de 4x20 donde despliega los mensajes de la base de datos al momento de realizar un cobro, adicional a estos componentes dispone de un lector de tarjetas NFC que detecta una tarjeta cercana y realizar el cobro sin necesidad de realizar otra acción.

De acuerdo con la funcionalidad del primer prototipo se adicionará elementos que complementen el dispositivo tomando en cuenta la conectividad, el tiempo de respuesta y la intensidad de señal que existe para su ejecución.

Diseño del primer prototipo Recargas Tarjeta NFC

Basado en los materiales descritos en la **tabla 3** y con la ayuda de un Ing. Electrónico se ensambló un prototipo para el manejo de las tarjetas NFC al momento de realizar una recarga en las tarjetas NFC, comenzando con un diseño de pruebas para la conexión de componente como se aprecia en la **figura 70**, adicional a estos componentes se llegó a la necesidad de implementar un teclado para el ingreso de información tal como se aprecia en la **figura 71**

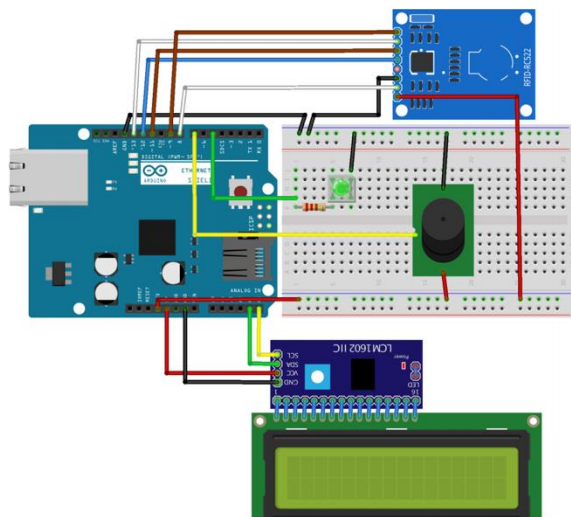


Figura 70: Diseño del primer prototipo lector y escritura de tarjetas NFC

Elaborado por: [80]

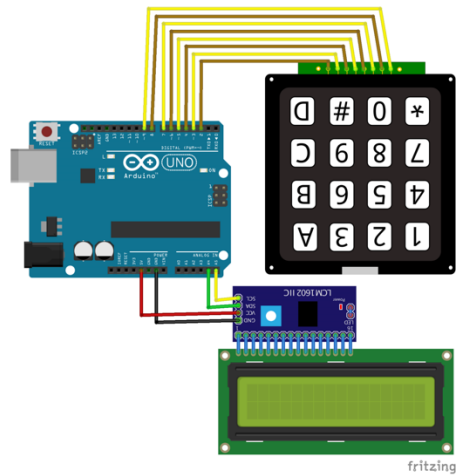


Figura 71: Implementación de un teclado mecánico al dispositivo

Elaborado por: [80]

Pruebas de funcionalidad del dispositivo recargas y Cobro de pasajes

Las pruebas de funcionalidad se las realizo en el campo donde desarrollaran su función, especificando 2 campos de pruebas, el primero es una unidad de transporte donde el dispositivo lector de tarjetas NFC realice la acción de cobro de pasaje como se puede apreciar en la **figura 72**.

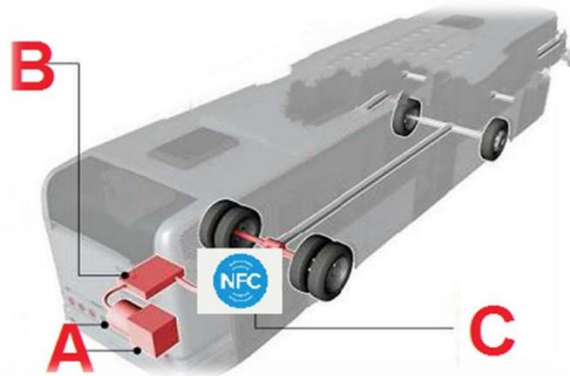


Figura 72: Diseño de la implementación en un bus de transporte urbano

Elaborado por: [10]

La instalación eléctrica del dispositivo lector de tarjetas NFC en una unidad de transporte Urbano “Tipo” se lo realiza de acuerdo con la Figura 71, el punto A es la fuente de alimentación del bus, normalmente una batería donde se toma 12V con la

ayuda de un regulador de voltaje que es el punto B, hasta llegar al dispositivo lector de tarjetas NFC que es el punto C.

Ya establecido la fuente de alimentación, se fija el dispositivo a la vista de los señores pasajeros con el fin de tener una interacción cómoda como se ve en la **figura 73**



Figura 73: Implementación del lector de tarjetas NFC en una unidad de Transporte

Elaborado por: Bryan Amaguaña

Ya instalado el dispositivo, nos refleja un mensaje el cual es muy claro para el señor pasajero el cual menciona que acerque la tarjeta para realizar el pago, tal como se presenta en la **figura 74**



Figura 74: Primer mensaje de aviso del dispositivo NFC

Elaborado por: Bryan Amaguaña

Al acercar una tarjeta NFC el dispositivo comienza a realizar el cobro y registrar el valor en la base de datos tomando un periodo de tiempo de 3 a 5 segundos dependiendo

de la intensidad de señal que exista para la conexión de internet por medio del Chip Sim800c, **Figura 75**



Figura 75: Proceso del cobro del pasaje

Elaborado por: Bryan Amaguaña

Al realizar el cobro del pasaje exitosamente el dispositivo emitirá un sonido y un mensaje informando al señor pasajero que el cobro se realizó exitosamente permitiendo el pase al interior del bus de transporte, **Figura 76**



Figura 76: Proceso del cobro del pasaje Exitoso

Elaborado por: Bryan Amaguaña

Al intentar realizar el pago con una tarjeta que no está registrado dentro del programa el dispositivo nos despliega un mensaje de advertencia emitiendo un sonido de alerta, **figura 77**



Figura 77: Proceso del cobro del pasaje con una tarjeta no registrada

Elaborado por: Bryan Amaguaña

Al intentar realizar un pago con una tarjeta que fue modificada ilegalmente el dispositivo emite un mensaje de advertencia con un sonido que alerta del estado de la tarjeta, mencionando que intente realizar el pago nuevamente para la verificación de la tarjeta, como se aprecia en la **figura 78**, Si se confirma la modificación ilegal de la tarjeta el dispositivo emite un mensaje mencionando que la tarjeta fue bloqueada como se aprecia en la **figura 79**



Figura 78: Proceso del cobro del pasaje con una tarjeta modificada ilegalmente

Elaborado por: Bryan Amaguaña



Figura 79: Mensaje de bloqueo de tarjeta

Elaborado por: Bryan Amaguaña

El siguiente campo de pruebas no está definido estáticamente por el cual se lo puede realizar en cualquier parte del Ambato debido a que el dispositivo de recargas de tarjetas NFC debe estar en múltiples lugares a la vista de las personas, por lo cual se estableció las pruebas en las oficinas de la cooperativa unión Ambateña como un lugar seguro de recargas tal como se puede apreciar en la **figura 80**



Figura 80: Dispositivo para recargas de tarjetas NFC

Elaborado por: Bryan Amaguaña

También se tomó como punto de referencia la unidad de transporte como lugar de recargas como se puede apreciar en la **figura 81**



Figura 81: Dispositivo para recargas de tarjetas NFC en una unidad de transporte

Elaborado por: Bryan Amaguaña

El dispositivo para realizar recargas en las tarjetas NFC presenta una interfaz con 2 opciones, la primera opción permite realizar una recarga a una tarjeta existente solo acercando la tarjeta la segunda opción permite crear una tarjeta, integrando los datos necesarios para que el lector acepte la tarjeta y realice el cobro respectivo del pasaje como se puede apreciar en la **figura 82**.



Figura 82: Creación exitosa de una tarjeta

Elaborado por: Bryan Amaguaña



Figura 83: Crear una tarjeta ya existente

Elaborado por: Bryan Amaguaña

Inicio de sesión en la aplicación UnionApp

Permite al usuario el ingreso a la aplicación móvil por medio de la cédula y del código de la tarjeta, si en el caso de no estar registrado en el sistema también presenta una opción donde direcciona a la pestaña de registro.

A screenshot of the UnionApp login screen. At the top center is the logo of "COOPERATIVA UNION". Below the logo are two input fields: "Cédula" and "Código de la tarjeta". Below these fields are two buttons: a blue button labeled "ENTRAR" and a black button labeled "REGISTRARSE".

Figura 84: Login de la Aplicación móvil

Elaborado por: Bryan Amaguaña

3.2.4.6 Iteración VI

	N°	Historia de Usuario	Tiempo estimado	
			Días	Horas
Cuarta Iteración	029	Diseño del componente Register de la App	3	15
	030	Diseño del menú de la App	2	10
	031	Diseño del componente home para la App	3	15
	032	Diseño del componente Pagos para la App	3	15
	033	Diseño del componente Recargas para la App	3	15
	034	Diseño del componente Mi cuenta para la App	4	20
	035	Pruebas de funcionalidad de la App UnionApp	2	10

Tabla 148 Iteración VI

Elaborado por: Bryan Amaguaña

Diseño del componente Register de la App

Permite a las personas registrarse de tal forma que ingresen sus datos personas y el código de la tarjeta que será tomado como clave de ingreso a la aplicación.

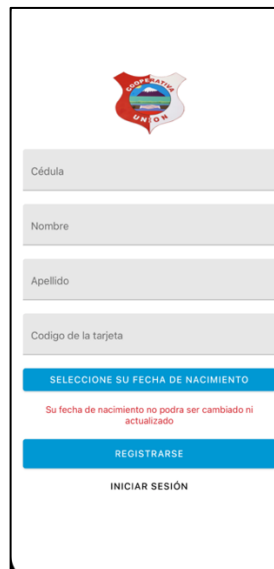


Figura 85: Registro de la Aplicación móvil

Elaborado por: Bryan Amaguaña

Diseño del menú de la App

Para la navegación una vez ingresado en la aplicación se implementó un menú inferior donde se despliega todas las opciones que posee la aplicación de igual forma cada menú lleva a un nuevo componente que despliega los datos de acuerdo con los datos del pasajero.



Figura 86: Menú de la Aplicación móvil

Elaborado por: Bryan Amaguaña

Diseño del componente home para la App

Despliega todos los datos del pasajero y la tarjeta, informando el estado en el que se encuentra la tarjeta, su saldo, su tipo de tarjeta y si posee algún bloqueo por el uso inadecuado que se le dé.

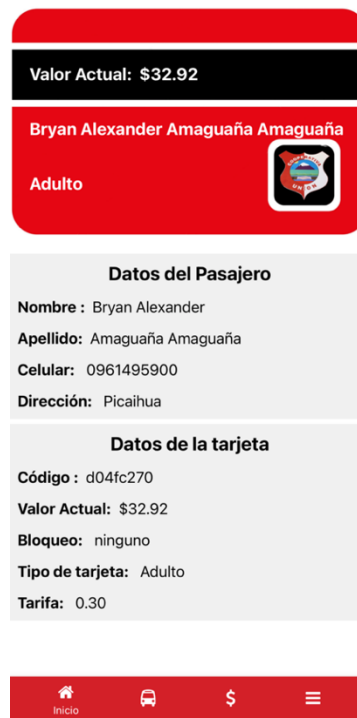


Figura 87: componente home de la Aplicación móvil

Elaborado por: Bryan Amaguaña

Diseño del componente Pagos para la App

Despliega todos los pagos realizados con la tarjeta actual, si se cambia de tarjeta por alguna razón los datos desplegados en este componente se actualizarán a la nueva tarjeta de tal forma que se pueda rastrear una tarjeta desde su inicio hasta que el usuario desee cambiarla.



Figura 88: componente Pagos de la Aplicación móvil

Elaborado por: Bryan Amaguaña

Diseño del componente Recargas para la App

Despliega todas las recargas realizadas con la tarjeta actual, si se cambia de tarjeta por alguna razón los datos desplegados en este componente se actualizarán a la nueva tarjeta de tal forma que se pueda rastrear una tarjeta desde su inicio hasta que el usuario desee cambiarla.

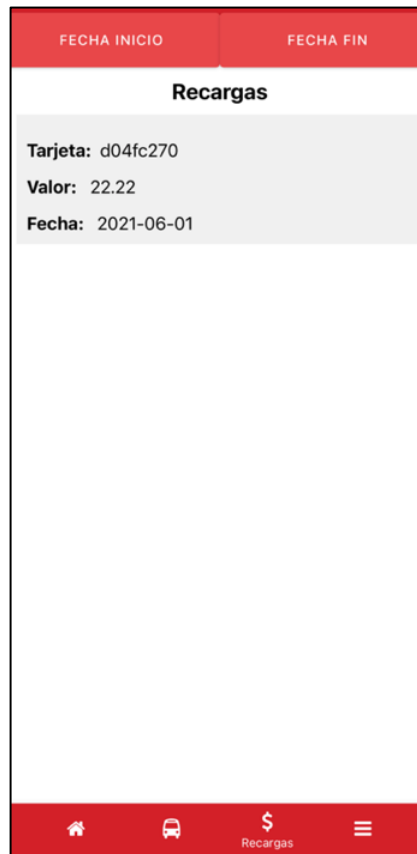


Figura 89: componente Recargas de la Aplicación móvil

Elaborado por: Bryan Amaguaña

Diseño del componente Mi cuenta para la App

Despliega múltiples componentes donde el usuario puede actualizar su información como se puede apreciar en la **figura 91**, de igual forma el usuario puede cambiar de tarjeta como se aprecia en la **figura 93**, o puede bloquear la tarjeta actual como se puede apreciar en la **figura 92**, de igual forma se implementó un apartado para los datos de contacto de la cooperativa de transporte urbano Unión Ambateña como se muestra en la **figura 94**

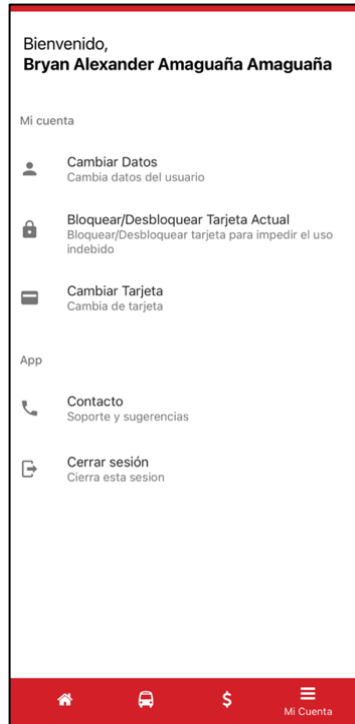


Figura 90: Componente Mi Cuenta de la Aplicación móvil

Elaborado por: Bryan Amaguaña

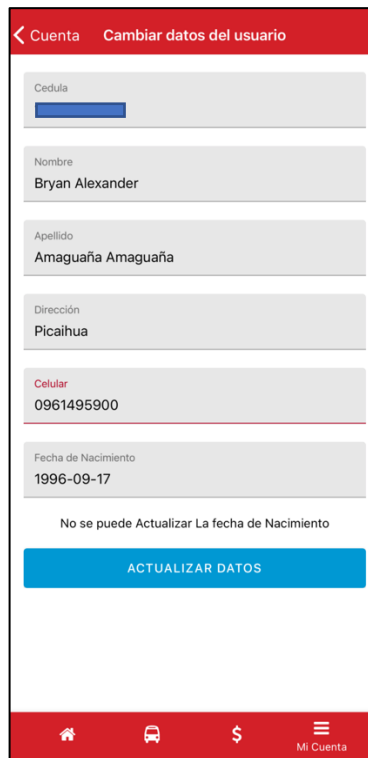


Figura 91: Componente para actualizar la información del usuario

Elaborado por: Bryan Amaguaña

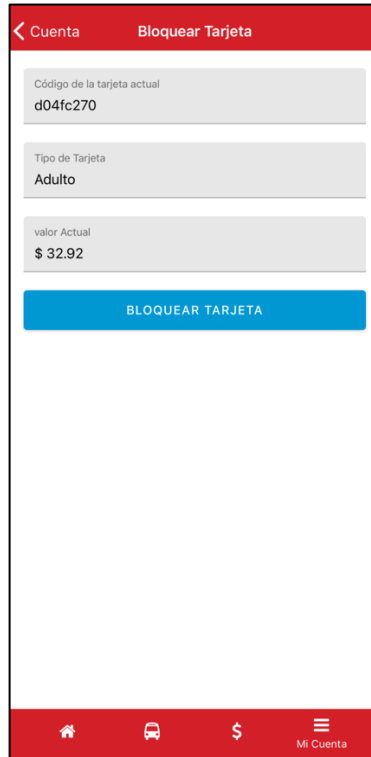


Figura 92: Componente para Bloquear una Tarjeta

Elaborado por: Bryan Amaguaña

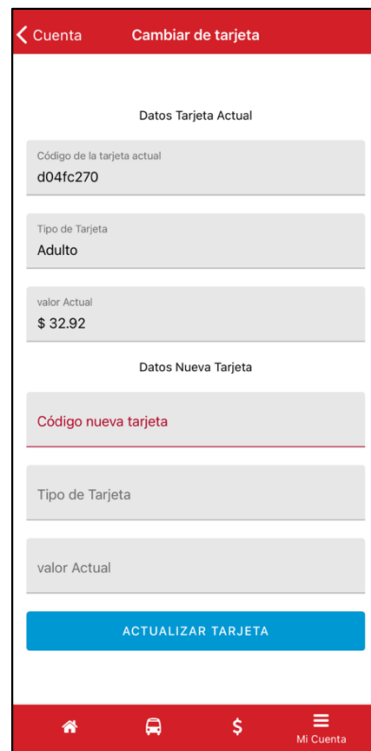


Figura 93: Componente para cambiar de tarjeta

Elaborado por: Bryan Amaguaña

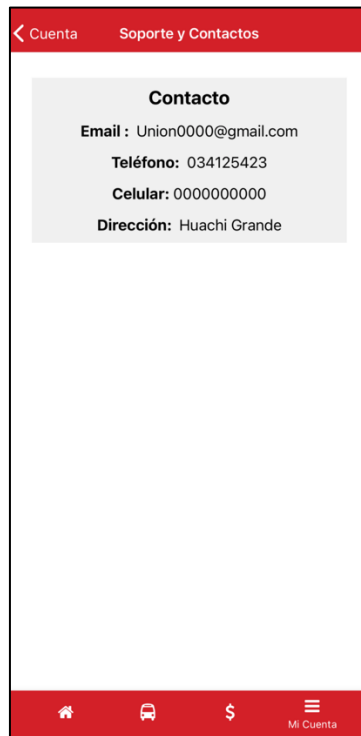


Figura 94: Componente que despliega la información de la empresa

Elaborado por: Bryan Amaguaña

Pruebas de funcionalidad de la App UnionApp

Las pruebas de funcionalidad de la aplicación móvil para la gestión de las tarjetas por parte del pasajero fueron realizados por el personal de sistemas de la cooperativa de transporte urbano “Unión ambateña”, simulando múltiples situaciones donde la tarjeta puede ser modificada o situaciones donde la tarjeta puede extraviarse, quedando solucionado todos los inconvenientes que se pueden presentar a un pasajero durante su experiencia utilizando una tarjeta inteligente para el pago del pasaje en el transporte urbano.

3.2.5 Codificación

3.2.5.1 Conexión con la base de datos y puerto

Conexión a la base de datos por medio del enlace proporcionado por mongo atlas y definición del puerto a utilizar.

```

// puerto //
process.env.PORT = process.env.PORT || 3050;

// entorno

process.env.PORT = process.env.PORT || "dev";

//SEED de autentificación

process.env.SEED = process.env.SEED || "este-es-el-seed-desarrollo";

// base de datos

let urlDB;

if (process.env.PORT === "dev") {
  //urlDB = "mongodb://localhost:27017/union";
  urlDB = 'mongodb+srv://Credenciales@cluster0.vfala.mongodb.net/union'
}

process.env.URLDB = urlDB;

```

Figura 95: Conexión con la base de datos y configuración del puerto

Elaborado por: Bryan Amaguaña

3.2.5.2 Método para crear el Token en la aplicación web

Método que permite la creación del token de acceso a la aplicación web por medio del usuario.

```

exports.createAccessToken = function (user) {
  const payload = {
    id: user._id,
    id_persona: user.id_persona,
    Usuario: user.nombre_usuario,
    avatar: user.avatar,
    correo: user.correo,
    rol: user.id_rol,
    createToken: moment().unix(),
    exp: moment().add(1, "hours").unix(),
  };

  return jwt.encode(payload, SECRET_KEY);
};

```

Figura 96: Método para crear el token en la aplicación web

Elaborado por: Bryan Amaguaña

3.2.5.3 Método para refrescar un token en la aplicación web

Método que refresca el token cuando el token expira, pero todavía existe actividad por parte del usuario.

```
exports.createRefreshToken = function (user) {
  const payload = {
    id: user._id,
    exp: moment().add(1, "days").unix(),
  };

  return jwt.encode(payload, SECRET_KEY);
};
```

Figura 97: Método refrescar un token en la aplicación web

Elaborado por: Bryan Amaguaña

3.2.5.4 Método de verificación del token

Método que permite verificar que un usuario ingreso correctamente al sistema.

```
let verificaToken = (req, res, next) => {
  if (!req.headers.authorization) {
    return res
      .status(403)
      .send({ message: "La petición no tiene la cabecera de Autenticación." });
  }
  const token = req.headers.authorization.replace(/["]+/g, "");
  try {
    var payload = jwt.decode(token, SECRET_KEY);

    if (payload.exp <= moment().unix()) {
      return res.status(404).send({ message: "El token ha expirado." });
    }
  } catch (ex) {
    return res.status(404).send({ message: "Token inválido." });
  }
  req.user = payload;
  next();
};
```

Figura 98: Método de verificación del token

Elaborado por: Bryan Amaguaña

3.2.5.5 Método de verificación de rol

Método que verifica el rol de un usuario de tal forma que no pueda realizar acciones sin privilegios dentro del sistema.

```
let verificarRol = (req, res, next) => {
  let usuario = req.user;
  if (usuario.rol == '5ff62a2d9e8b3e14d287f16c') {
    next();
  } else {
    return res.status(404).send({ message: "Usted no es administrador." });
  }
};
```

Figura 99: Método de verificación de rol

Elaborado por: Bryan Amaguaña

3.2.5.6 Estructura básica de una colección

Estructura de una colección de datos, todas las colecciones poseen la misma estructura variando el tipo de dato que desee incorporar.

```
const mongoose = require('mongoose');
const uniqueValidator = require('mongoose-unique-validator');

let Schema = mongoose.Schema;

let BusSchema = new Schema({
  numero_bus: {
    type: Number,
    unique: true,
    required: [true, "Numero Obligatorio"]
  },
  id_persona: {
    type: Schema.ObjectId,
    ref: 'Persona',
    required: [true, "Dueño Obligatorio"]
  },
  placa_bus: {
    type: String,
```

```

    required: false
  },

  disponible: {
    type: Boolean,
    required: [true, "Disponibilidad Obligatorio"]
  },
});

BusSchema.plugin(uniqueValidator, { message: 'Numero de Bus ya registrado' });

module.exports = mongoose.model('Bus', BusSchema);

```

Figura 100: Estructura básica de una colección

Elaborado por: Bryan Amaguaña

3.2.5.7 Configuración del servidor

Clase principal donde se especifica las rutas, las librerías a utilizar, el puerto y la conexión con la base datos.

```

require('./config/config');

const express = require('express');
const mongoose = require('mongoose');

const app = express();

const bodyParser = require('body-parser');

//
app.use(bodyParser.urlencoded({ extended: false }));

//
app.use(bodyParser.json());

//configuración global de rutas
app.use(require('./routes/index'));

//conexión a mongo
mongoose.connect(process.env.URLDB, { useNewUrlParser: true, useCreateIndex: true,
useUnifiedTopology: true }, (err, res) => {
  if (err) throw err;
  console.log('Base de datos conectada');
});

```

```
//puerto
app.listen(process.env.PORT, () => {
  console.log('escuchando puerto: ', process.env.PORT);
});
```

Figura 101: Configuración del servidor

Elaborado por: Bryan Amaguaña

3.2.5.8 Método Get para la colección Buses

La sintaxis del método Get es la misma para cada colección, variando los requerimientos del Front-end.

```
/* Listado de buses paginada */
app.get('/ObtenerBus/:disponible/:desde/:limite', [verificaToken], (req, res) => {
  try {
    let disponible = req.params.disponible;
    const desde = req.params.desde;
    const limite = req.params.limite;
    Bus.find({ disponible: disponible }).skip(Number(desde)).limit(Number(limite)).sort({ numero_bus:
1 }).populate('id_persona').exec((err, bus) => {
      if (err) {
        return res.status(400).send({ message: "No se encontro ningun bus." });
      }
      res.json({
        bus: bus
      });
    });
  } catch (error) {
    console.log("Error: ObtenerBus")
    console.log(error);
  }
});
```

Figura 102: Método GET para la colección Buses

Elaborado por: Bryan Amaguaña

3.2.5.9 Método Post para la colección Buses

La sintaxis del método Post es la misma para cada colección, variando los requerimientos del Front-End.

```
/* Agregar un bus */
app.post('/AgregarBus', [verificaToken, verificarRol], function (req, res) {
  try {
    const bus = new Bus();
    let usuario = req.user;
    const { numero_bus, id_persona, placa_bus } = req.body;
    bus.numero_bus = numero_bus;
    bus.id_persona = id_persona;
    bus.placa_bus = placa_bus;
    bus.disponible = true;

    bus.save((err, BusStored) => {
      if (err) {
        res.status(500).send({ message: "El Bus ya existe." });
      } else {
        if (!BusStored) {
          res.status(404).send({ message: "Error al crear el Bus." });
        } else {
          res.status(200).send({ message: "Bus creado exitosamente." });
          AgregarHistorial(usuario.id, "Agrego el bus: " + bus.numero_bus);
        }
      }
    });
  } catch (error) {
    console.log("Error: AgregarBus")
    console.log(error);
  }
});
```

Figura 103: Método POST para la colección Buses

Elaborado por: Bryan Amaguaña.

3.2.5.10 Método Put para la colección Buses

La sintaxis del método Put es la misma para cada colección, variando los requerimientos del Front-End.


```

/* Activar/Desactivar Buses por el Id */
app.put('/ActivarBuses/:id', [verificaToken, verificarRol], function activateBus(req, res) {
  try {
    const { id } = req.params;
    const { disponible } = req.body;
    let usuario = req.user;
    Bus.findByIdAndUpdate({ _id: id }, { disponible }, (err, BusActivado) => {
      if (err) {
        res.status(500).send({ message: "Error del servidor." });
      } else {
        if (!BusActivado) {
          res.status(404).send({ message: "No se ha encontrado ningun Bus." });
        } else {
          if (disponible) {
            res.status(200).send({ message: "Bus activado correctamente." });
            AgregarHistorial(usuario.id, "Activo el bus id: " + id);
          } else {
            res.status(200).send({ message: "Bus desactivado correctamente." });
            AgregarHistorial(usuario.id, "desactivo el bus id: " + id);
          }
        }
      }
    });
  } catch (error) {
    console.log("Error: ActivarBus");
    console.log(error);
  }
});

```

Figura 104: Método PUT para la colección Buses

Elaborado por: Bryan Amaguaña

3.2.5.11 Método Delete para la colección Buses

La sintaxis del método Delete es la misma para cada colección, variando los requerimientos del Front-End.

```

/* Eliminar un Bus */
app.delete('/BorrarBus/:id', [verificaToken, verificarRol], function (req, res) {
  try {
    const { id } = req.params;
    let usuario = req.user;
    Bus.findByIdAndRemove(id, (err, BusDeleted) => {

```

```

if (err) {
  res.status(500).send({ message: "Error del servidor." });
} else {
  if (!BusDeleted) {
    res.status(404).send({ message: "Bus no encontrado." });
  } else {
    res.status(200).send({ message: "El Bus ha sido eliminado correctamente." });
    AgregarHistorial(usuario.id, "Borro el bus: " + id);
  }
}
});
} catch (error) {
  console.log("Error: BorrarBus");
  console.log(error);
}
});

```

Figura 105: Método Delete para la colección Buses

Elaborado por: Bryan Amaguaña

3.2.5.12 Método para filtrar cobros de pasajes por unidad de transporte

Método que permite realizar cálculos de los ingresos por fechas establecidas desde el Back-End, sumando todos los ingresos de cada bus de la base de datos.

```

function ResolverRepetidos(lista) {
  try {
    const buses = [];
    const busesTemp = [];
    const ListTemp = [];
    let cob = 0;
    var Total = 0;
    for (let i = 0; i < lista.length; i) {
      ListTemp.push(lista[i]);
      busesTemp.push(lista[i]);
      lista.splice(i, 1);
      for (let q = 0; q < lista.length; q++) {
        if (parseFloat(ListTemp[0].numero_bus_cobro) === parseFloat(lista[q].numero_bus_cobro)) {
          busesTemp.push(lista[q]);
          lista.splice(q, 1);
          q = q - 1;
        }
      }
    }
  }
}

```

```

for (let x = 0; x < busesTemp.length; x++) {
  Total += busesTemp[x].valor_pagado
  cob = cob + 1;
}
buses.push({ Numero_bus: ListTemp[0].numero_bus_cobro, total: Total, cobros: cob })
ListTemp.splice(0)
busesTemp.splice(0)
Total = 0;
cob = 0;
}
return buses;
} catch (error) {
  console.log("Error: Resolver Repetidos")
  console.log(error)
}
}
}

```

Figura 106: Método para filtrar los cobros de pasaje por unidad de transporte

Elaborado por: Bryan Amaguaña

3.2.5.13 Conexión de Front-End con el Back-End

En el proyecto de la aplicación web, se establece la conexión con los servicios de forma global como se representa en la **figura 107**.

```

//export const basePath = "http://localhost:3050";
export const basePath = "https://servicios-union.herokuapp.com";
export const apiVersion = "v1";

```

Figura 107: Conexión del Front-End con el Back-End

Elaborado por: Bryan Amaguaña

3.2.5.14 Método para iniciar sesión.

Petición al servidor del método Login a través de una sentencia Post.

```

export function signInApi(data) {
  const url = `${basePath}/login`;
  const params = {
    method: "POST",
    body: JSON.stringify(data),
    headers: {
      "Content-Type": "application/json"
    }
  }
}

```

```

    }
  };

  return fetch(url, params)
    .then(response => {
      return response.json();
    })
    .then(result => {
      return result;
    })
    .catch(err => {
      return err.message;
    });
}

```

Figura 108: Método para iniciar sesión.

Elaborado por: Bryan Amaguaña

3.2.5.15 Método GET para obtener usuarios de la base de datos

Método GET para la obtención de todos los usuarios de la base de datos por medio del token.

```

export function ObtenerTodosUsuarios(token) {
  const url = `${basePath}/ObtenerUsuario`;
  const params = {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
      Authorization: token
    }
  };
  return fetch(url, params)
    .then(response => {
      return response.json();
    })
    .then(result => {
      return result;
    })
    .catch(err => {
      return err.message;
    });
}

```

Figura 109: Método GET para obtener usuarios de la base datos

Elaborado por: Bryan Amaguaña

3.2.5.16 Método para desplegar los datos de los usuarios en un componente

Método GET para la obtención de todos los usuarios de la base de datos por medio del token.

```
export default function Users() {
  const [UsuariosActivos, setusuariosActivos] = useState([]);
  const [UsuariosInactivos, setusuariosInactivos] = useState([]);
  const [reloadUsers, setReloadUsers] = useState(false);
  const [rol, setRol] = useState('');
  const token = getAccessTokenApi();

  useEffect(() => {

    ObtenerUsuariosAl(token, true, 0, 4).then(response => {
      setusuariosActivos(response.usuario);
    });
    ObtenerUsuariosAl(token, false, 0, 4).then(response => {
      setusuariosInactivos(response.usuario);
    });
    ObtenerRol(token).then(response => {
      setRol(response.rol);
    });
    setReloadUsers(false);
  }, [token, reloadUsers]);

  return (
    <div className="users">
      <ListUsers usuarioActivo={UsuariosActivos} usuarioInactivo={UsuariosInactivos} rol={rol}
      setReloadUsers={setReloadUsers} setusuariosActivos={setusuariosActivos}
      setusuariosInactivos={setusuariosInactivos}/>
    </div>
  );
}
```

Figura 110: Método para desplegar los datos de un componente

Elaborado por: Bryan Amaguaña

3.2.5.17 Método para crear reportes en el cobro de pasajes

De acuerdo con los datos filtrados en los gráficos de barra se despliega el reporte.

```
const GenerarReporte = () => {
  const doc = new jsPDF("p", "pt", "a4"); // default values
  doc.setFontSize(16);
  doc.text(
    contenido[0].nombre.toUpperCase(),
    105 * 2.83, //horizontal
    15 * 2.83, //vertical
    null,
    null,
    "center"
  );
  doc.setFont("Times-Roman", "normal");
  doc.setFontSize(10);
  doc.text(
    "Correo: " + contenido[0].correo.toLowerCase(),
    105 * 2.83, //horizontal
    19 * 2.83, //vertical
    null,
    null,
    "center"
  );
  doc.setFont("Times-Roman", "normal");
  doc.text(
    "Teléfono: " + contenido[0].telefono.toLowerCase() + " Celular: " +
    contenido[0].Celular.toLowerCase(),
    105 * 2.83, //horizontal
    23 * 2.83, //vertical
    null,
    null,
    "center"
  );
  doc.setFont("Times-Roman", "normal");

  doc.setFontSize(14);
  doc.text(
    103 * 2.83, //posicion Horizontal
    26 * 2.83, //Posicion vertical
    null,
    null,
    "center"
  );
  doc.setFontSize(14);
  doc.text(
    "Reporte mensual por unidad de transporte",
    105 * 2.83, //horizontal
```

```

34 * 2.83, //vertical
null,
null,
"center"
);
doc.setFont("Times-Roman", "normal");
if (InicioRp === "" || InicioRp === " " || FinRp === "" || FinRp === " ") {
  notification["info"]({
    message: "Reporte generado del mes de " + mes[month]
  });

  doc.setFontSize(13);
  doc.text(
    // eslint-disable-next-line
    "Del " + `01/${month < 10 ? `0${month}` : `${month}`}/${year}` + " Al " + `${date}/${month <
10 ? `0${month}` : `${month}`}/${year}`,
    105 * 2.83, //horizontal
    42 * 2.80, //vertical
    null,
    null,
    "center"
  );
  doc.setFont("Times-Roman", "normal");
} else {
  notification["info"]({
    message: "Reporte generado del " + InicioRp + " Al " + FinRp
  });
  doc.setFontSize(13);
  doc.text(
    "Del " + InicioRp + " Al " + FinRp,
    105 * 2.83, //horizontal
    42 * 2.80, //vertical
    null,
    null,
    "center"
  );
  doc.setFont("Times-Roman", "normal");
}
if (BusquedaCobro) {
  let TotalB = 0;

  if (BusquedaCobro) {
    for (let i = 0; i < BusquedaCobro.length; i++) {
      TotalB += BusquedaCobro[i].valor_pagado;
    }
  }
  doc.setFontSize(13);
  doc.text(
    "Total de ingresos $" + TotalB.toFixed(2),
    14 * 2.83, //horizontal

```

```

48 * 2.83, //vertical
null,
null,
"left"
);
doc.setFont("Times-Roman", "normal");
const usersCol = [ " ", " " ];
let Pasajeros = 0;
let PasajerosA = 0;
let PasajerosAT = 0;
let PasajerosN = 0;
let PasajerosNT = 0;
if (BusquedaCobro) {
  for (let i = 0; i < BusquedaCobro.length; i++) {
    if (BusquedaCobro[i].valor_pagado > 0.15) {
      PasajerosA++;
      PasajerosAT += BusquedaCobro[i].valor_pagado;
    } else {
      PasajerosN++;
      PasajerosNT += BusquedaCobro[i].valor_pagado;
    }
    Pasajeros++;
  }
}
const usersRows = [
  ["Cédula del propietario ", Cedula],
  ["Nombre Apellido del propietario ", Dueño],
  ["Bus N. ", Nbus],
  ["Fecha Inicio ", InicioRp],
  ["Fecha Fin ", FinRp],
  ["Pasajes completos registrados ($ 0.30)", PasajerosA + " Adultos"],
  ["Total pasaje completos", "$ " + PasajerosAT.toFixed(2)],
  ["Niños o Tercera edad registrados ($ 0.15) ", PasajerosN + " Niños/Tercera edad"],
  ["Total medio pasaje", "$ " + PasajerosNT.toFixed(2)],
  ["Total Pasajeros ", Pasajeros + " Personas"],
  ["Ingresos Totales ", "$ " + TotalB.toFixed(2)];
const startY = 50 * 2.83;
doc.autoTable(usersCol, usersRows, {
  startY,
  theme: "striped",
  styles: {
    fontSize: 11
  }
});
} else {
doc.setFontSize(14);
doc.text(
  "Sin Conexión a la base de Datos",
  105 * 2.83, //horizontal
  55 * 2.83, //vertical

```



```

    null,
    null,
    "center"
  );
  doc.setFont("Times-Roman", "normal");
}
doc.save(`Reporte Cobros Bus N ${Nbus} del mes de ${mes[month]}.pdf`);
}

```

Figura 111: Método para elaborar reportes en el cobro de pasajes

Elaborado por: Bryan Amaguaña

3.2.5.18 Método para elaborar el grafico de barras de cobros de pasaje

De acuerdo con los datos filtrados en los servicios se despliega el grafico de barras.

```

function CreacionGrafico(props) {
  const { data, total, Busqueda, InicioR, FinR } = props;

  const opciones = {
    maintainAspectRatio: false,
    responsive: true
  }

  let newDate = new Date()
  const date2 = moment(newDate).format("DD-MM-YYYY");
  let mes = newDate.getMonth();
  // eslint-disable-next-line
  var meses = new Array("Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto",
  "Septiembre", "Octubre", "Noviembre", "Diciembre");
  return (
    <div>
      <h3>{Busqueda ? `${InicioR ? "Ingreso del " + InicioR : "Mes no seleccionado"} al ${FinR} $:
      ${total.toFixed(2)}`
      : `Ingreso del mes de ${meses[mes]} hasta ${date2} $: ${total.toFixed(2)}`}</h3>
      <Bar className="Gafico" data={data} options={opciones} />
    </div>
  );
}

```

Figura 112: Método para elaborar el grafico de barra del cobro de pasaje

Elaborado por: Bryan Amaguaña

3.2.5.19 Método para lista los cobros de pasajes

Todos los cobros registrados en la base de datos se despliegan en el componente cobros con su información

```
function ListaPersonas(props) {
  const { cobro } = props;
  return (
    <List.Item>
      <List.Item.Meta
        title={ `Cobro de la unidad:
        ${cobro.numero_bus_cobro ? cobro.numero_bus_cobro : '...'}`
      ` }
      description={
        <div>
          <b>Tarjeta :</b> {cobro.codigo_tarjeta ? cobro.codigo_tarjeta : '...'}
          <br />
          <b>Valor:</b> {cobro.valor_pagado ? Valor(cobro.valor_pagado) : '...'}
          <br />
          <b>Fecha :</b> {cobro.fecha_hora_cobro ? cobro.fecha_hora_cobro.substr(0, 10) : '...'}
        </div>
      }
    </List.Item>
  );
}
```

Figura 113: Método para listar los cobros de pasaje

Elaborado por: Bryan Amaguaña

3.2.5.20 Variables para la conexión de los servicios a la aplicación móvil

Ya en la aplicación móvil se declara las variables que son base para las peticiones a los servicios.

```
//export const API_URL = "http://localhost:3050";
export const API_URL = "https://servicios-union-app.herokuapp.com";
export const apiVersion = "v1";
export const TOKEN = "token";
```

Figura 114: Variables para la conexión de los servicios a la aplicación móvil

Elaborado por: Bryan Amaguaña

3.2.5.21 Método para el ScreenLoading

Cuanto una página se demora en desplegar datos aparece el ScreenLoading, indicando que se está cargando los datos.

```
export default function ScreenLoading(props) {
  const { text, size, color } = props;

  return (
    <SafeAreaView style={styles.containerLoading}>
      <ActivityIndicator size={size} color={color} style={styles.loading} />
      <Text style={styles.title}>{text}</Text>
    </SafeAreaView>
  );
}

var styles = StyleSheet.create({
  containerLoading: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center",
  },
  loading: {
    marginBottom: 10,
  },
  title: {
    fontSize: 18,
  },
});

ScreenLoading.defaultProps = {
  text: "Cargando...",
  color: "#000",
};
```

Figura 115: Método para el ScreenLoading

Elaborado por: Bryan Amaguaña

3.2.5.22 Método para registrar un pasajero en la App

Cuanto los datos fueron llenados correctamente se ejecuta la constante Registrar que ingresa los datos de la persona en la base de datos.

```
const Registrar = useFormik({
  initialValues: initialValues(),
  validationSchema: Yup.object(validationSchema()),
  onSubmit: async (formData) => {
    setLoading(true);
    var f = new Date();
    f.setDate(f.getDate() - 1800);
    /* ----- calculo de edad */
    const TipoPasajero = () => {
      var hoy = new Date();
      var cumpleaños = new Date(Fecha);
      var edad = hoy.getFullYear() - cumpleaños.getFullYear();
      var m = hoy.getMonth() - cumpleaños.getMonth();
      if (m < 0 || (m === 0 && hoy.getDate() < cumpleaños.getDate())) {
        edad--;
      }
      if (edad < 18) {
        return "Estudiante";
      } else if (edad >= 18) {
        return "Adulto";
      } else if (edad >= 60) {
        return "Tercera Edad";
      }
    }
  }
});
/* ----- */
if (validarCedula(formData.cedula_persona)) {
  setLoading(true);
  try {
    if (Fecha < f) {
      const mensaje = await registerApi(formData);
      if (
        mensaje === "Persona creada exitosamente." ||
        mensaje === "Numero de cédula ya registrado."
      ) {
        await ObtenerPersonaCedula(formData.cedula_persona).then(
          (response) => {
            try {
              if (response.persona.cedula_persona) {
                pasajeroData.id_persona = response.persona._id;
                pasajeroData.cedula_persona = response.persona.cedula_persona;
              }
            } catch (error) {
              setTimeout(function () {
```

```

        Toast.show("Intente nuevamente por favor", {
            position: Toast.positions.CENTER,
        });
        setLoading(false);
    }, 1000);
    }
}
);
await ObtenerTarjetaCodigoApp(formData.codigo, true).then(
    (response) => {
        try {
            if (response.tarjeta[0].codigo) {
                pasajeroData.id_tarjeta_pasajero = response.tarjeta[0]._id;
                pasajeroData.id_tipo_pasajero = response.tarjeta[0].descripcion._id;
                setId_Tipo_Tarjeta(response.tarjeta[0].descripcion.nombre);
            }
        } catch (error) {
            setTimeout(function () {
                Toast.show("Codigo de tarjeta incorrecto", {
                    position: Toast.positions.CENTER,
                });
                setLoading(false);
            }, 500);
        }
    }
);
if (TipoPasajero() === Id_Tipo_Tarjeta) {
    await CrearPasajero(pasajeroData).then((response) => {
        try {
            if (response === "Pasajero creado exitosamente.") {
                setTimeout(function () {
                    Toast.show(response, {
                        position: Toast.positions.CENTER,
                    });
                    showLogin();
                }, 1000);
            } else {
                setTimeout(function () {
                    Toast.show(response, {
                        position: Toast.positions.CENTER,
                    });
                }, 3000);
                setLoading(false);
            }
        } catch (error) {
            setLoading(false);
        }
    });
} else {
    setTimeout(function () {

```

```

    Toast.show("La tarjeta no es la correcta", {
      position: Toast.positions.CENTER,
    });
    setLoading(false);
  }, 3000);
}
} else {
  Toast.show(mensaje, {
    position: Toast.positions.CENTER,
  });
  setLoading(false);
}
} else {
  Toast.show("Ingrese su verdad fecha de nacimiento", {
    position: Toast.positions.CENTER,
  });
  setLoading(false);
}
} catch (error) {
  setLoading(false);
}
} else {
  Toast.show(Mensaje, {
    position: Toast.positions.CENTER,
  });
}
},
});

```

Figura 116: Método para registrar un pasajero en la App

Elaborado por: Bryan Amaguaña

3.2.5.23 Método para desplegar los datos del pasajero en el Home

Los datos registrados en la base de datos son desplegados en la pantalla home de la aplicación móvil.

```

export default function Datos(props) {
  const { Pasajero } = props;

  const B = (props) => <Text style={{ fontWeight: 'bold' }}>{props.children}</Text>

  const Valor = valor => {
    var cadena = valor;
    var separador = ".";
    var arregloDeSubCadenas = cadena.toString().split(separador, 3);
  }
}

```

```

    if (arregloDeSubCadenas[1] && arregloDeSubCadenas[1].length === 1) {
        cadena = arregloDeSubCadenas[0] + "." + arregloDeSubCadenas[1] + "0";
    }
    return cadena;
}

return (
    <View style={styles.container}>
        <View style={styles.containerProduct}>
            <View style={styles.product}>

                <ImageBackground
                    source={Logo}
                    style={styles.logo}
                >
                    <Text style={styles.Valor}> Valor Actual: ${Pasajero.id_tarjeta_pasajero.valor_tarjeta
? Valor(Pasajero.id_tarjeta_pasajero.valor_tarjeta) : ".."}</Text>
                    <Text style={styles.ValorNombre}> {Pasajero.id_persona.nombre_persona ?
Pasajero.id_persona.nombre_persona : ".."} {Pasajero.id_persona.apellido_persona ?
Pasajero.id_persona.apellido_persona : ".."}</Text>
                    <Text style={styles.ValorNombre}> {Pasajero.id_tipo_pasajero.nombre ?
Pasajero.id_tipo_pasajero.nombre : ".."}</Text>
                </ImageBackground >

            </View>

            <View>
                <Text></Text>
            </View>

            <View style={stylesDatos.containerProduct}>
                <View style={stylesDatos.product}>
                    <Text style={styles.title}>Datos del Pasajero</Text>
                    <Text style={stylesDatos.name} numberOfLines={1} ellipsizeMode="tail">
                        <B>Nombre : </B> {Pasajero.id_persona.nombre_persona}
                    </Text>
                    <Text style={stylesDatos.name} numberOfLines={1} ellipsizeMode="tail">
                        <B>Apellido: </B> {Pasajero.id_persona.apellido_persona}
                    </Text>
                    <Text style={stylesDatos.name} numberOfLines={1} ellipsizeMode="tail">
                        <B>Celular:</B> {Pasajero.id_persona.celular_persona}
                    </Text>
                    <Text style={stylesDatos.name} numberOfLines={1} ellipsizeMode="tail">
                        <B>Dirección: </B> {Pasajero.id_persona.direccion_persona}
                    </Text>
                </View>
            </View>

            <View style={stylesDatos.containerProduct}>

```

```

<View style={stylesDatos.product}>
  <Text style={styles.title}>Datos de la tarjeta</Text>
  <Text style={stylesDatos.name} numberOfLines={1} ellipsizeMode="tail">
    <B>Código : </B> {Pasajero.id_tarjeta_pasajero.codigo}
  </Text>
  <Text style={stylesDatos.name} numberOfLines={1} ellipsizeMode="tail">
    <B>Valor Actual: </B> ${Valor(Pasajero.id_tarjeta_pasajero.valor_tarjeta)}
  </Text>
  <Text style={stylesDatos.name} numberOfLines={1} ellipsizeMode="tail">
    <B>Bloqueo:</B> {Pasajero.id_tarjeta_pasajero.disponible ? "ninguno" :
Pasajero.id_tarjeta_pasajero.bloqueo}
  </Text>
  <Text style={stylesDatos.name} numberOfLines={1} ellipsizeMode="tail">
    <B>Tipo de tarjeta: </B> {Pasajero.id_tipo_pasajero.nombre}
  </Text>
  <Text style={stylesDatos.name} numberOfLines={1} ellipsizeMode="tail">
    <B>Tarifa: </B> {Valor(Pasajero.id_tipo_pasajero.valor)}
  </Text>
</View>
</View>
</View>
)
}

```

Figura 117: Método para desplegar los datos en el componente home

Elaborado por: Bryan Amaguaña

3.2.5.24 Método para desplegar las recargas realizadas en la App

Todas las recargas realizadas con la tarjeta se desplegarán con el siguiente método.

```

export default function RecargasFiltradas(props) {
  const { recargas } = props;

  const B = (props) => <Text style={{ fontWeight: 'bold' }}>{props.children}</Text>

  const Valor = valor => {
    var cadena = valor;
    var separador = ".";
    var arregloDeSubCadenas = cadena.toString().split(separador, 3);
    if (arregloDeSubCadenas[1] && arregloDeSubCadenas[1].length === 1) {
      cadena = arregloDeSubCadenas[0] + "." + arregloDeSubCadenas[1] + "0";
    }
  }
  return cadena;
}

```



```

}

return (
  <View style={styles.container}>
    {map(recargas, (recarga) => (
      <TouchableWithoutFeedback key={recarga._id}>
        <View style={styles.containerProduct}>
          <View style={styles.product}>
            <Text style={styles.name} numberOfLines={1} ellipsizeMode="tail">
              <B>Tarjeta: </B> {recarga.codigo_tarjeta}
            </Text>
            <Text style={styles.name} numberOfLines={1} ellipsizeMode="tail">
              <B>Valor:</B> {Valor(recarga.valor_recarga)}
            </Text>
            <Text style={styles.name} numberOfLines={1} ellipsizeMode="tail">
              <B>Fecha: </B> {recarga.fecha_hora_Accion.substr(0, 10)}
            </Text>
          </View>
        </View>
      </TouchableWithoutFeedback>
    ))}
  </View>
);
}

```

Figura 118: Método para desplegar las recargas realizadas

Elaborado por: Bryan Amaguaña

3.2.5.25 Menú inferior de la App

```

export default function AppNavigation() {
  return (
    <NavigationContainer>
      <Tab.Navigator
        barStyle={styles.navigation}
        screenOptions={({ route }) => ({
          tabBarIcon: (routeStatus) => {
            return setIcon(route, routeStatus);
          },
        })>
        <Tab.Screen
          name="home"
          component={Home}
          options={{
            title: "Inicio",
          }}
        />
      </Tab.Navigator>
    </NavigationContainer>
  );
}

```

```

        name="Historial"
        component={PagosStack}
        options={{
          title: "Pagos",
        }}
      />
      <Tab.Screen
        name="Recargas"
        component={RecargasStack}
        options={{
          title: "Recargas",
        }}
      />
      <Tab.Screen
        name="Configuraciones"
        component={AccountStack}
        options={{
          title: "Mi Cuenta",
        }}
      />
    </Tab.Navigator>
  </NavigationContainer>
)
}

```

Figura 119: Menú de la App

Elaborado por: Bryan Amaguaña

3.2.5.26 Método para cerrar sesión

```

const logoutAccount = () => {
  Alert.alert(
    "Cerrar sesión",
    "¿Deseas salir de tu cuenta?",
    [{ text: "NO" }, { text: "SI", onPress: logout }],
    { cancelable: false }
  );
};

```

Figura 120: Método para cerrar sesión

Elaborado por: Bryan Amaguaña

3.2.5.27 Método Recargas de Tarjetas NFC Arduino

Permite realizar una recarga a una tarjeta previamente registrada en la base de datos.

```

void recarga(float valorD) {
  String dato = "";
  String valorE = "";
  String valorU = "";
  float vT = 0;
  String recar = "https://servicio-union.herokuapp.com/AgregarRecarga/";
  recar += valorD;
  recar += "/";
  Serial.println("Recarga");
  while ( ! mfr522.PICC_IsNewCardPresent());

  // Select one of the cards
  while ( ! mfr522.PICC_ReadCardSerial());

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Procesando...");

  Serial.print("\n");
  Serial.println("***Card Detected***");
  // Print UID of the Card
  Serial.print(F("Card UID:"));
  for (byte i = 0; i < mfr522.uid.size; i++)
  {
    Serial.print(mfr522.uid.uidByte[i] < 0x10 ? "0" : "");
    Serial.print(mfr522.uid.uidByte[i], HEX);
    dato += String(mfr522.uid.uidByte[i] < 0x10 ? "0" : "");
    dato += String(mfr522.uid.uidByte[i], HEX);
    recar += String(mfr522.uid.uidByte[i] < 0x10 ? "0" : "");
    recar += String(mfr522.uid.uidByte[i], HEX);
    //
  }
  Serial.println("");
  Serial.println("ID:");
  Serial.println(dato);
  //nueva += tipo;
  Serial.print("\n");
  // Print type of card (for example, MIFARE 1K)
  Serial.print(F("PICC type: "));
  MFRC522::PICC_Type piccType = mfr522.PICC_GetType(mfr522.uid.sak);
  Serial.println(mfr522.PICC_GetTypeName(piccType));
  ReadDataFromBlock(blockNum, readBlockData);
  //recar += String(readBlockData[0] < 0x10 ? "0" : "");
  recar += "/Bryan";
  valorE += String(readBlockData[1], DEC);
  valorE += ".";
  valorE += String(readBlockData[2], DEC);
  Serial.print("\n");
  Serial.print("valor Anterior:");

```

```

Serial.print(valorE.toFloat());
Serial.print("\n");
Serial.print("valor A sumar:");
Serial.print(valorD);
Serial.print("\n");
Serial.print("valor Nuevo:");
vT = valorE.toFloat() + valorD;
Serial.println(vT);
Serial.println("valor String:");
Serial.print(String(vT, 2));
Serial.print("\n");
Serial.print("valor Entero:");
Serial.println(String(vT, 2).substring(0, String(vT, 2).indexOf('.')).toInt());
Serial.print("\n");
Serial.print("valor Decimal:");
Serial.println(String(vT, 2).substring(String(vT, 2).indexOf('.') + 1, String(vT, 2).indexOf('.') + 3).toInt());

readBlockData[1] = String(vT, 2).substring(0, String(vT, 2).indexOf('.')).toInt();
readBlockData[2] = String(vT, 2).substring(String(vT, 2).indexOf('.') + 1, String(vT, 2).indexOf('.') +
3).toInt();
/*Serial.print("\n");
  Serial.print("Bloque de dolar:");
  Serial.println(readBlockData[1], HEX);*/
byte escribir [] = {readBlockData[0], readBlockData[1], readBlockData[2]};
// Call 'WriteDataToBlock' function, which will write data to the block
Serial.print("\n");
Serial.println("Writing to Data Block...");
WriteDataToBlock(blockNum, escribir);
// Read data from the same block
Serial.print("\n");
Serial.println("Reading from Data Block...");
ReadDataFromBlock(blockNum, readBlockData);
// If you want to print the full memory dump, uncomment the next line
//mfr522.PICC_DumpToSerial(&(mfr522.uid));

// Print the data read from block
Serial.print("\n");
Serial.print("Data in Block:");
Serial.print(blockNum);
Serial.print(" --> ");

//printHex(readBlockData,2);

for (int j = 0 ; j < 16 ; j++)
{
  Serial.write(readBlockData[j]);
}
Serial.print("\n");

```

```

Serial.println(dato);
Serial.println(recar);
dato = "";
//dato = "";
// Halt PICC
mfrc522.PICC_HaltA();
// Stop encryption on PCD
mfrc522.PCD_StopCrypto1();
//Serial.println(nueva);
conexionR(recar);
delay(3000);
}

```

Figura 121: Método recargas para Tarjetas NFC Arduino

Elaborado por: Bryan Amaguaña

3.2.5.28 Método para crear una tarjeta Arduino

Permite crear una tarjeta en la base de datos para realizar un pago del pasaje o realizar una recarga.

```

void crear(byte data[], const String& tipo) {
  String dato = "";
  String nueva = "https://servicios-union.herokuapp.com/InicializarTarjeta/";
  String ID="";
  Serial.println("crear");
  while ( ! mfrc522.PICC_IsNewCardPresent());

  // Select one of the cards
  while ( ! mfrc522.PICC_ReadCardSerial());

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Procesando...");

  Serial.print("\n");
  Serial.println("***Card Detected***");
  // Print UID of the Card
  Serial.print(F("Card UID:"));
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    dato += String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
    dato += String(mfrc522.uid.uidByte[i], HEX);
    nueva += String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
  }
}

```

```

nueva += String(mfrc522.uid.uidByte[], HEX);
//Serial.println(nueva);
}
Serial.println("");
Serial.println("ID:");
Serial.println(dato);
ID=dato;
nueva += "/1/";
nueva += tipo;
Serial.print("\n");

// Print type of card (for example, MIFARE 1K)
Serial.print(F("PICC type: "));
MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
Serial.println(mfrc522.PICC_GetTypeName(piccType));

// Call 'WriteDataToBlock' function, which will write data to the block
Serial.print("\n");
Serial.println("Writing to Data Block...");
WriteDataToBlock(blockNum, data);

// Read data from the same block
Serial.print("\n");
Serial.println("Reading from Data Block...");
ReadDataFromBlock(blockNum, readBlockData);
// If you want to print the full memory dump, uncomment the next line
//mfrc522.PICC_DumpToSerial(&(mfrc522.uid));

// Print the data read from block
Serial.print("\n");
Serial.print("Data in Block:");
Serial.print(blockNum);
Serial.print(" --> ");
dato += String(readBlockData[1] < 0x10 ? "0" : "");
dato += String(readBlockData[1], HEX);
dato += ".";
dato += String(readBlockData[2] < 0x10 ? "0" : "");
dato += String(readBlockData[2], HEX);
dato += "F";

//printHex(readBlockData,2);

for (int j = 0 ; j < 16 ; j++)
{
  Serial.write(readBlockData[j]);
}
Serial.print("\n");
Serial.println(dato);
dato = "";

```

```

//dato = "";
// Halt PICC
mfrc522.PICC_HaltA();
// Stop encryption on PCD
mfrc522.PCD_StopCrypto1();
//Serial.println(nueva);
conexionR(nueva);
delay(3000);
lcd.clear();
lcd.setCursor(0, 1);
lcd.print(" ID de la Tarjeta ");
lcd.setCursor(2, 2);
lcd.print(ID);
delay(5000);
}

```

Figura 122: Método para crear una tarjeta Arduino

Elaborado por: Bryan Amaguaña

3.2.5.29 Método para realizar un cobro de pasaje

Permite realizar un cobro del pasaje a una tarjeta previamente registrado en la base de datos.

```

void cobrar(float valorD) {
  String dato = "";
  String valorE = "";
  String valorU = "";
  String valorCobro = "";
  float vT = 0;
  float vC = 0;
  String recar = "https://servicios-union.herokuapp.com/AgregarCobro/";
  Serial.println("Recarga");
  while ( ! mfrc522.PICC_IsNewCardPresent());

  // Select one of the cards
  while ( ! mfrc522.PICC_ReadCardSerial());

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Procesando...");

  Serial.print("\n");
  Serial.println("**Card Detected**");
  // Print UID of the Card
  Serial.print(F("Card UID:"));

```

```

for (byte i = 0; i < mfrc522.uid.size; i++)
{
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    dato += String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
    dato += String(mfrc522.uid.uidByte[i], HEX);
    recar += String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
    recar += String(mfrc522.uid.uidByte[i], HEX);
    //
}
Serial.println("");
Serial.println("ID:");
Serial.println(dato);
//nueva += tipo;
Serial.print("\n");
// Print type of card (for example, MIFARE 1K)
Serial.print(F("PICC type: "));
MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
Serial.println(mfrc522.PICC_GetTypeName(piccType));
ReadDataFromBlock(blockNum, readBlockData);

recar += "/";
recar += String(readBlockData[1], DEC);
recar += ".";
recar += String(readBlockData[2], DEC);
recar += "/";
recar += "05/";

valorE += String(readBlockData[1], DEC);
valorE += ".";
valorE += String(readBlockData[2], DEC);
Serial.print("\n");
Serial.print("valor Anterior:");
Serial.print(valorE.toFloat());
Serial.print("\n");
Serial.print("valor A sumar:");
Serial.print(valorD);
Serial.print("\n");
Serial.print("valor Nuevo:");
//String(readBlockData[0],DEC); Tipo de tarjeta
switch (readBlockData[0]) {
    case 1:
        valorCobro = "0.30";
        vC = 0.30;
        break;
    case 2:
        valorCobro = "0.15";
        vC = 0.15;
        break;
    case 3:

```



```

    valorCobro = "0.15";
    vC = 0.15;
    break;
default:
    // if nothing else matches, do the default
    // default is optional
    break;
}
recar += valorCobro;
vT = valorE.toFloat() - vC; // Aqui se suma o se resta

Serial.println(vT);
Serial.println("valor String:");
Serial.print(String(vT, 2));
Serial.print("\n");
Serial.print("valor Entero:");
Serial.println(String(vT, 2).substring(0, String(vT, 2).indexOf('.')).toInt());
Serial.print("\n");
Serial.print("valor Decimal:");
Serial.println(String(vT, 2).substring(String(vT, 2).indexOf('.') + 1, String(vT, 2).indexOf('.') + 3).toInt());

readBlockData[1] = String(vT, 2).substring(0, String(vT, 2).indexOf('.')).toInt();
readBlockData[2] = String(vT, 2).substring(String(vT, 2).indexOf('.') + 1, String(vT, 2).indexOf('.') +
3).toInt();
/*Serial.print("\n");
Serial.print("Bloque de dolar:");
Serial.println(readBlockData[1], HEX);*/
if (vT >= 0) {
    byte escribir [] = {readBlockData[0], readBlockData[1], readBlockData[2]};
    // Call 'WriteDataToBlock' function, which will write data to the block

    Serial.print("\n");
    Serial.println("Writing to Data Block...");
    WriteDataToBlock(blockNum, escribir);
    // Read data from the same block
    Serial.print("\n");
    Serial.println("Reading from Data Block...");
    ReadDataFromBlock(blockNum, readBlockData);
    // If you want to print the full memory dump, uncomment the next line
    //mfr522.PICC_DumpToSerial(&(mfr522.uid));

    // Print the data read from block
    Serial.print("\n");
    Serial.print("Data in Block:");
    Serial.print(blockNum);
    Serial.print(" --> ");

    //printHex(readBlockData,2);

```

```

for (int j = 0 ; j < 16 ; j++)
{
  Serial.write(readBlockData[j]);

}

}

else {
  Serial.print("Saldo Insuficiente");
  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("Saldo Insuficiente");
  digitalWrite(49, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(300);             // wait for a second
  digitalWrite(49, LOW); // turn the LED off by making the voltage LOW
  delay(150);
  digitalWrite(49, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(300);             // wait for a second
  digitalWrite(49, LOW);
  delay(2000);
}
Serial.print("\n");
Serial.println(dato);
Serial.println(recar);
dato = "";
//dato = "";
// Halt PICC
mfrc522.PICC_HaltA();
// Stop encryption on PCD
mfrc522.PCD_StopCrypto1();
//Serial.println(nueva);
conexionR(recar);
delay(3000);

```

Figura 123: Método para realizar un cobro de pasaje

Elaborado por: Bryan Amaguaña

3.2.5 Fase V: Pruebas

El presente proyecto de investigación fue sometido a pruebas de aceptación con el fin de validar el correcto funcionamiento del sistema, dando como resultado la satisfacción del cliente con el funcionamiento del sistema, aprobando cada historia de usuario de una manera satisfactoria, aplicando pruebas de caja negra para evaluar cada historia de usuario con sus respectivos resultados.

Pruebas de Aceptación	Número: 01
N° de historia de usuario: 004	
Nombre: Despliegue de la base de datos	
Descripción: La base de datos debe estar disponible un host virtual con el fin de poder guardar y obtener información en cualquier momento y desde cualquier lugar.	
Condiciones de ejecución: La base de datos debe mostrar las colecciones de datos en la plataforma virtual y los datos almacenados de prueba.	
Entrada/Procesos de ejecución: El desarrollador debe desplegar las colecciones y los datos en la plataforma Heroku, de tal forma que puedan realizar la conexión con los servicios.	
Resultados esperados: El desarrollador debe observar las colecciones creadas en la base de datos, así también debe emitir un enlace para realizar las conexiones con los servicios	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 149 Prueba de aceptación 01 – Despliegue de la base de datos.

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 02
N° de historia de usuario: 005	
Nombre: Despliegue de los servicios	
Descripción: Los servicios deben estar disponible en un host virtual con el fin de poder acceder a los servicios para interactuar con la base de datos.	
Condiciones de ejecución: Los servicios debe mostrar los datos de la base de datos de acuerdo con las peticiones de la aplicación.	
Entrada/Procesos de ejecución: El desarrollador debe desplegar los datos de las colecciones de la base de datos con los enlaces de los servicios	
Resultados esperados: El desarrollador debe observar los datos desplegados de la base de datos al ejecutar una petición de los servicios.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 150 Prueba de aceptación 02 – Despliegue de los servicios.

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 03
N° de historia de usuario: 006	
Nombre: Diseño del layout Admin	
Descripción: Los usuarios registrados en la aplicación podrán visualizar el layout Admin	

Condiciones de ejecución: Ninguno
Entrada/Procesos de ejecución: El usuario debe iniciar sesión para ingresar al layout Admin
Resultados esperados: El usuario puede observar el diseño del layout Admin con los colores de la cooperativa “Unión Ambateña”
Resultado de la prueba: Prueba Satisfactoria.

Tabla 151 Prueba de aceptación 03 – Diseño del layout Admin

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 04
N° de historia de usuario: 007	
Nombre: Pagina inicio de sesión	
Descripción: El usuario para ingresar al sistema debe ingresar sus credenciales para autenticarse e ingresar a los módulos del sistema.	
Condiciones de ejecución: Disponer de sus credenciales para la autenticación en el sistema.	
Entrada/Procesos de ejecución: El usuario debe dirigirse a la página de login de la aplicación donde ingresara sus datos de autenticación para ingresar a la página principal del sistema.	
Resultados esperados: Ingresa al sistema y se despliega la página principal	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 152 Prueba de aceptación 04 – Página de inicio de sesión.

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 05
N° de historia de usuario: 008	
Nombre: Diseño del MenuSider	
Descripción: El usuario podrá visualizar un menú lateral izquierdo con todos los componentes del sistema	
Condiciones de ejecución: El usuario debe estar correctamente autenticado en el sistema.	
Entrada/Procesos de ejecución: El usuario debe desplegar el menú lateral izquierdo desde le MenuTop para visualizar los nombres de los componentes	
Resultados esperados: Visualización de los componentes de la aplicación con sus respectivos iconos y nombres mismos que están re direccionados a las pestañas correspondientes.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 153 Prueba de aceptación 05 – Diseño del MenuSider

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 06
N° de historia de usuario: 009	
Nombre: Diseño del MenuTop	
Descripción: El usuario podrá visualizar un menú superior donde se desplegará la opción de visualización del MenuSider, el logo de la cooperativa de transporte urbano “Unión Ambateña” y la opción de cerrar sesión.	
Condiciones de ejecución: Ingresar a la aplicación correctamente.	
Entrada/Procesos de ejecución: El MenuTop se desplegará inmediatamente después de ingresar a la aplicación	
Resultados esperados: Visualización del menú superior en el layout Admin.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 154 Prueba de aceptación 09 – Diseño del MenuTop

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 07
N° de historia de usuario: 010	
Nombre: Diseño del Modal y vistas del componente usuario	
Descripción: Los usuarios que ingresen en el sistema visualizaran los datos de los usuarios activos y desactivados del sistema, permitiendo agregar, editar, buscar o desactivar un usuario	
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar y buscar la información de otros usuarios, pero solo el usuario administrador podrá agregar, editar o desactivar un usuario	
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente usuario disponible en el MenuSider	
Resultados esperados: Visualización del componente usuario con las opciones de agregar, editar, buscar o desactivar un usuario	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 155 Prueba de aceptación 07 – Diseño del Modal y vistas del componente usuario

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 08
N° de historia de usuario: 011	

Nombre: Diseño del Modal y vistas del componente rol
Descripción: Los usuarios que ingresen en el sistema visualizarán los datos de los roles del sistema, permitiendo agregar o editar un rol
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar la información de los roles, pero solo el usuario administrador podrá agregar o editar un rol.
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente rol disponible en el MenuSider
Resultados esperados: Visualización del componente rol con las opciones de agregar o editar un rol
Resultado de la prueba: Prueba Satisfactoria.

Tabla 156 Prueba de aceptación 08 – Diseño del Modal y vistas del componente rol

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 09
N° de historia de usuario: 012	
Nombre: Diseño del Modal y vistas del componente persona	
Descripción: Los usuarios que ingresen en el sistema visualizarán los datos de las personas registradas en el sistema, permitiendo agregar, buscar o editar una persona	
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar y buscar la información de las personas, pero solo el usuario administrador podrá agregar o editar una persona.	
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente persona disponible en el MenuSider	
Resultados esperados: Visualización del componente persona con las opciones de agregar, buscar o editar una persona	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 157 Prueba de aceptación 09 – Diseño del Modal y vistas del componente persona

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 10
N° de historia de usuario: 013	
Nombre: Diseño del Modal y vistas del componente Buses	
Descripción: Los usuarios que ingresen en el sistema visualizarán los datos de los buses registrados en el sistema, permitiendo agregar, buscar o editar un bus	

Condiciones de ejecución: Todos los usuarios registrados podrán visualizar y buscar la información de las personas, pero solo el usuario administrador podrá agregar o editar un bus.
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente bus disponible en el MenuSider
Resultados esperados: Visualización del componente buses con las opciones de agregar, buscar desactivar o editar un bus
Resultado de la prueba: Prueba Satisfactoria.

Tabla 158 Prueba de aceptación 10 – Diseño del Modal y vistas del componente Buses

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 11
N° de historia de usuario: 014	
Nombre: Diseño del Modal y vistas del componente Tarjetas	
Descripción: Los usuarios que ingresen en el sistema visualizaran los datos de las tarjetas registradas en el sistema, permitiendo agregar, buscar o editar una tarjeta.	
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar y buscar la información de las tarjetas, pero solo el usuario administrador podrá agregar o editar una tarjeta.	
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente tarjeta disponible en el MenuSider	
Resultados esperados: Visualización del componente tarjeta con las opciones de agregar, buscar, desactivar o editar una tarjeta	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 159 Prueba de aceptación 11 – Diseño del Modal y vistas del componente Tarjetas

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 12
N° de historia de usuario: 015	
Nombre: Diseño del Modal y vistas del componente Tipo_Pasajero	
Descripción: Los usuarios que ingresen en el sistema visualizaran los datos de los tipos de pasajeros registrados en el sistema, permitiendo agregar, buscar o editar un tipo de pasajero.	
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar y buscar la información de los tipos de pasajeros, pero solo el usuario administrador podrá agregar, editar o desactivar un tipo de pasajero.	

Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente tipo pasajero disponible en el MenuSider
Resultados esperados: Visualización del componente tipo pasajero con las opciones de agregar, buscar, desactivar o editar una tarjeta.
Resultado de la prueba: Prueba Satisfactoria.

Tabla 160 Prueba de aceptación 12 – Diseño del Modal y vistas del componente Tipo_Pasajero

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 13
N° de historia de usuario: 016	
Nombre: Diseño del Modal y vistas del componente Pasajero	
Descripción: Los usuarios que ingresen en el sistema visualizaran los datos de todos los pasajeros registrados en el sistema, permitiendo agregar, buscar o editar un pasajero.	
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar y buscar la información de los pasajeros, pero solo el usuario administrador podrá agregar, editar o desactivar un pasajero.	
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente pasajero disponible en el MenuSider.	
Resultados esperados: Visualización del componente pasajero con las opciones de agregar, buscar o editar un pasajero.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 161 Prueba de aceptación 13 – Diseño del Modal y vistas del componente Pasajero

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 14
N° de historia de usuario: 017	
Nombre: Diseño del Modal y vistas del componente Cobro_Pasaje	
Descripción: Los usuarios que ingresen en el sistema visualizaran los datos de todos los cobros de pasajes registrados en el sistema, permitiendo buscar un cobro de pasaje por fechas y número de bus.	
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar y buscar la información de los cobros de pasajes.	
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente cobro pasaje disponible en el MenuSider	
Resultados esperados: Visualización del componente cobro pasajes con las opciones de buscar un cobro de pasaje.	

Resultado de la prueba: Prueba Satisfactoria.

Tabla 162 Prueba de aceptación 14 – Diseño del Modal y vistas del componente Cobro_Pasaje

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 15
N° de historia de usuario: 018	
Nombre: Diseño del Modal y vistas del componente Recargas.	
Descripción: Los usuarios que ingresen en el sistema visualizarán los datos de todas las recargas registradas en el sistema, permitiendo buscar una recarga de tarjetas por fechas y nombre del usuario.	
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar y buscar la información de las recargas.	
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente recargas disponible en el MenuSider	
Resultados esperados: Visualización del componente recargas con las opciones de buscar una recarga.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 163 Prueba de aceptación 15 – Diseño del Modal y vistas del componente Recargas

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 16
N° de historia de usuario: 019	
Nombre: Diseño del Modal y vistas del componente Menu_Web	
Descripción: Los usuarios que ingresen en el sistema visualizarán los datos de todos los menús registrados en el sistema, permitiendo agregar, desactivar o editar un menú web.	
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar la información de los menús, pero solo el usuario administrador podrá agregar, editar o desactivar un menú web.	
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente Menú Web disponible en el MenuSider.	
Resultados esperados: Visualización del componente menú web con las opciones de agregar, desactivar o editar un menú.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 164 Prueba de aceptación 16 – Diseño del Modal y vistas del componente Menu_Web

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 17
N° de historia de usuario: 020	
Nombre: Diseño del Modal y vistas del componente Contenido_Web	
Descripción: Los usuarios que ingresen en el sistema visualizaran la información editable de la cooperativa “Unión Ambateña”.	
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar la información de la cooperativa “Unión Ambateña”, pero solo el usuario administrador podrá editar dicha información.	
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente Contenido web disponible en el MenuSider	
Resultados esperados: Visualización del componente Contenido web con las opciones de editar información del contenido web.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 165 Prueba de aceptación 17 – Diseño del Modal y vistas del componente Contenido_Web

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 18
N° de historia de usuario: 021	
Nombre: Diseño de la página de inicio de Contenido_web	
Descripción: Los usuarios autenticados o no autenticados en el sistema visualizaran el layout público con toda la información de la cooperativa de transporte urbano “Unión Ambateña”.	
Condiciones de ejecución: Todas las personas que poseen el enlace de la página de inicio del sistema podrán visualizar la información de la cooperativa de transporte urbano “Unión ambateña”	
Entrada/Procesos de ejecución: El usuario debe poseer el enlace de la página de información de la cooperativa de transporte urbano “Unión ambateña”	
Resultados esperados: Visualización de la página de inicio con la información ingresada a través del componente Contenido_Web.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 166 Prueba de aceptación 18 – Diseño de la página de inicio de Contenido_web

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 19
N° de historia de usuario: 022	
Nombre: Diseño de la página de inicio del sistema	

Descripción: Los usuarios que ingresen en el sistema visualizarán los gráficos de barra de todos los ingresos de cobros de pasajes realizados por los buses, así también de las recargas realizadas por los usuarios asignados para realizar este trabajo.
Condiciones de ejecución: Todos los usuarios registrados podrán visualizar y crear reporte de todos los ingresos de la cooperativa de transporte urbano “Unión Ambateña”.
Entrada/Procesos de ejecución: El usuario deberá seleccionar el ítem del componente inicio disponible en el MenuSider
Resultados esperados: Visualización del gráfico de barras y realización de reportes de todos los ingresos de la cooperativa de transporte urbano “Unión Ambateña”.
Resultado de la prueba: Prueba Satisfactoria.

Tabla 167 Prueba de aceptación 19 – Diseño de la página de inicio del sistema

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 20
N° de historia de usuario: 025	
Nombre: Diseño del primer prototipo Lector de tarjetas NFC	
Descripción: Dispositivo diseñado para realizar cobros de pasajes en una unidad de transporte urbano	
Condiciones de ejecución: El dispositivo debe estar alimentado con 12V en una unidad de transporte con la ayuda de un transformador conectado a la batería del bus, adicional a esto, debe constar con un paquete de datos que permita la conexión a internet para el almacenamiento de datos	
Entrada/Procesos de ejecución: El usuario pasajero de una unidad de transporte urbano debe poseer una tarjeta registrada en el sistema, el cual debe acercarse al lector de tarjetas con el fin de realizar el pago del pasaje.	
Resultados esperados: Mensaje de confirmación por parte del dispositivo lector de tarjetas NFC, y registro del cobro en la base de datos.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 168 Prueba de aceptación 20 – Diseño del primer prototipo lector de tarjetas NFC

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 21
N° de historia de usuario: 026	
Nombre: Diseño del primer prototipo Recargas Tarjetas NFC	

Descripción: Dispositivo diseñado para registrar tarjetas en la base de datos y realizar recargas de tarjetas NFC
Condiciones de ejecución: El dispositivo incorpora una batería interna que facilita su movilidad, adicional a esto, debe constar con un paquete de datos que permita la conexión a internet para el almacenamiento de datos
Entrada/Procesos de ejecución: El usuario pasajero de una unidad de transporte urbano debe poseer una tarjeta registrada en el sistema, el cual debe acercarse al dispositivo recargador de saldo para aumentar el saldo de la tarjeta y poder realizar pagos en una unidad de transporte urbano
Resultados esperados: Mensaje de confirmación de éxito al momento de realizar una recarga y al momento de registrar una tarjeta
Resultado de la prueba: Prueba Satisfactoria.

Tabla 169 Prueba de aceptación 21 – Diseño del primer prototipo Recargas Tarjetas NFC

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 22
N° de historia de usuario: 028	
Nombre: Inicio de sesión en la aplicación UnionApp	
Descripción: El usuario pasajero para ingresar a la aplicación móvil debe ingresar sus credenciales para autenticarse (cedula, código de la tarjeta).	
Condiciones de ejecución: Disponer de sus credenciales para la autenticación en la aplicación.	
Entrada/Procesos de ejecución: El usuario debe descargar la aplicación móvil para ingresar a la app.	
Resultados esperados: Ingresar a la aplicación móvil y visualizar la página home de la app	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 170 Prueba de aceptación 22 – Inicio de sesión en la aplicación UnionApp

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 23
N° de historia de usuario: 029	
Nombre: Diseño del componente Register de la app	
Descripción: Para acceder a la aplicación móvil por medio del login, primero debe realizar el registro en la app, ingresando sus datos personales y el código de la tarjeta que realizará el cobro del pasaje	

Condiciones de ejecución: Disponer de sus datos personales y de una tarjeta NFC registrada en el sistema.
Entrada/Procesos de ejecución: el usuario debe dirigirte a la opción de Registrarse y llenar las casillas con sus datos personales y el código de la tarjeta para realizar un registro exitoso.
Resultados esperados: Si el registro fue exitoso recibirá un mensaje por parte de la app confirmando el registro y automáticamente se redireccionará al login de la aplicación.
Resultado de la prueba: Prueba Satisfactoria.

Tabla 171 Prueba de aceptación 22 – Diseño del componente Register de la App

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 24
N° de historia de usuario: 030	
Nombre: Diseño del menú de la App	
Descripción: Al realizar un ingreso exitoso a la App, en la parte inferior se desplegará un menú con todas las opciones que ofrece la aplicación móvil con un icono interactivo en cada opción	
Condiciones de ejecución: Realizar un ingreso exitoso a través del login de la aplicación.	
Entrada/Procesos de ejecución: El usuario debe ingresar a través del login.	
Resultados esperados: Al ingresar a la aplicación se despliega un login con las opciones de la aplicación.	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 172 Prueba de aceptación 24 – Diseño del menú de la App

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 25
N° de historia de usuario: 031	
Nombre: Diseño del componente home para la App	
Descripción: Componente principal donde despliega toda la información del usuario pasajero juntamente con la información de la tarjeta.	
Condiciones de ejecución: Realizar un ingreso exitoso a través del login de la aplicación.	
Entrada/Procesos de ejecución: El usuario debe ingresar a través del login.	
Resultados esperados: Al ingresar a la aplicación se despliega el componente home con la información del usuario y de la tarjeta registrada	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 173 Prueba de aceptación 25 – Diseño del componente home para la App

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 26
N° de historia de usuario: 032	
Nombre: Diseño del componente Pagos para la App	
Descripción: Componente diseñado para desplegar todos los pagos de pasajes realizados con la tarjeta registrada filtrados por fechas	
Condiciones de ejecución: Realizar un ingreso exitoso a través del login de la aplicación.	
Entrada/Procesos de ejecución: El usuario pasajero deberá seleccionar el ítem del componente Pagos disponible en el menú inferior	
Resultados esperados: Visualización del componente Pagos con todos los pagos realizados con la tarjeta registrada filtrada por fechas	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 174 Prueba de aceptación 26 – Diseño del componente Pagos para la App

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 27
N° de historia de usuario: 033	
Nombre: Diseño del componente Recargas para la App	
Descripción: Componente diseñado para desplegar todas las recargas realizadas con la tarjeta registrada, filtrados por fechas	
Condiciones de ejecución: Realizar un ingreso exitoso a través del login de la aplicación.	
Entrada/Procesos de ejecución: El usuario pasajero deberá seleccionar el ítem del componente Recargas disponible en el menú inferior	
Resultados esperados: Visualización del componente Recargas con todos los pagos realizados con la tarjeta registrada, filtrada por fechas	
Resultado de la prueba: Prueba Satisfactoria.	

Tabla 175 Prueba de aceptación 27 – Diseño del componente Recargas para la App

Elaborado por: Bryan Amaguaña

Pruebas de Aceptación	Número: 28
N° de historia de usuario: 034	

Nombre: Diseño del componente Mi cuenta para la App
Descripción: Componente diseñado para la configuración de la cuenta del usuario pasajero.
Condiciones de ejecución: Realizar un ingreso exitoso a través del login de la aplicación.
Entrada/Procesos de ejecución: El usuario pasajero deberá seleccionar el ítem del componente Mi cuenta disponible en el menú inferior
Resultados esperados: Visualización de las opciones como edición de la información del pasajero, cambio de tarjeta, bloqueo de tarjeta o cerrar sesión.
Resultado de la prueba: Prueba Satisfactoria.

Tabla 176 Prueba de aceptación 28 – Diseño del componente Mi cuenta para la App

Elaborado por: Bryan Amaguaña

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El desarrollo de un sistema informático basado en tarjetas inteligentes para la automatización del cobro del pasaje en las unidades de transporte urbano tubo gran acogida por las personas dueñas de las unidades de transporte así también por las personas que hacer uso de este medio de transporte.
- Se observó el correcto funcionamiento de los dispositivos lectores de tarjetas NFC en la implementación en una unidad de transporte, aprobando cada una de las pruebas expuestas por el gerente de la cooperativa, siendo verificado cada cobro a través del sistema web, y de la aplicación móvil afirmando el funcionamiento y las ventajas de la automatización.
- Al realizar las pruebas de funcionalidad de acuerdo con las exigencias del cliente se verifico la opción de aplicar nuevos métodos de cobros de pasaje que además de adicionar seguridad para el usuario conductor también contribuye con los registros de ingresos diarios que posee la cooperativa en cada unidad de transporte.
- La utilización de tarjetas NFC como método para realizar el pago del pasaje, posee una gran ventaja, gracias a la información que puede almacenar el tag de cada tarjeta aportando además de la tarifa del pasaje información como el tipo de pasaje, el valor y la fecha que se realizó el pago, aumentando la información de cada cobro con el fin de desplegar un informe completo de cada unidad de transporte que ayude con la contabilidad de la cooperativa.
- El uso de la tecnología RFID permite la comunicación inalámbrica entre chips NFC lo que permite el intercambio de información sin necesidad de realizar algún contacto entre dispositivos permitiendo realizar pagos manera segura.

- El cumplimiento de las medidas de bioseguridad al momento de utilizar el transporte público está siendo respetado por la ciudadanía y por los usuarios conductores, rigiéndose al límite máximo de capacidad en cada bus, pero continúan con el mismo sistema de cobro de pasaje lo que descarga las medidas de bioseguridad del medio de transporte. Al establecer un nuevo método de cobro del pasaje se elimina todo contacto entre usuarios manteniendo las medidas de bioseguridad entre personas.
- La aplicación web permite la gestión de todos los ingresos que realiza el dispositivo lector de tarjetas NFC generando reportes de cada unidad de transporte, manteniendo una contabilidad estable e informando a los respectivos dueños los ingresos diarios o mensuales de su unidad.
- Al ser un proyecto de investigación su desarrollo está limitado por los materiales disponibles al alcance del investigador con respecto a los lectores de tarjetas NFC, alcanzando un costo estimado de 150 dólares por cada lector implementado en cada bus, de igual manera los dispositivos recargadores están estimados en 150 dólares cada uno, estos valores están sujeto a la fecha que fue elaborado el presente proyecto, y a los materiales utilizados, dando un valor total de implementación de 15 mil dólares, tomando en cuenta los lectores de tarjeta NFC, recargadores y el sistema en sí.
- La aplicación móvil permite controlar las actividades que realiza la tarjeta registrada a un usuario, informando cada pago del pasaje realizado filtrado por fechas, de la misma forma permite visualizar todas las recargas realizadas en la tarjeta manteniendo un control de la tarjeta. Dispone de opciones que permite controlar la utilización de la tarjeta NFC, permitiendo bloquear o cambiar de tarjeta de una manera rápida y eficaz.

4.2 Recomendaciones

- Es recomienda cotizar el valor de un hosting para el despliegue de la aplicación y de la base de datos, debido a que permanecen en un host gratuito limitando su rendimiento, de tal forma que no tengo limitaciones y pueda mejorar su crecimiento a medida que pase el tiempo.
- Se recomienda innovar los dispositivos lectores y escritores de tarjetas NFC de acuerdo con las nuevas tecnologías que aparezcan en los años venideros.

BIBLIOGRAFÍA

- [1] “Orientaciones para el público.”
https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public?gclid=CjwKCAiA_eb-BRB2EiwAGBnXXqtay5LcAUdPdjo0F8ozP8Ibkd_R_3Qc5-1jBL_jkK7tBVUMMwI9zRoCCR0QAvD_BwE (accessed Jul. 09, 2021).
- [2] “*El Telégrafo* - Digitalización o mortalidad empresarial, el camino para las mipymes.”
<https://www.eltelegrafo.com.ec/noticias/tecnologia/1/digitalizacion-emprendimiento-paginaweb-kolau> (accessed Jul. 09, 2021).
- [3] “*El Telégrafo* - Pagos digitales aumentan en Ecuador aunque siguen por debajo de la media regional.”
<https://www.eltelegrafo.com.ec/noticias/economia/4/pagos-digitales-aumentan-en-ecuador-aunque-siguen-por-debajo-de-la-media-regional> (accessed Jul. 09, 2021).
- [4] “Este lunes 14 de diciembre Ecuador no reportó nuevas muertes por covid-19; cifra de contagios asciende a 202 180 - *El Comercio*.”
<https://www.elcomercio.com/actualidad/informe-contagios-muertes-coronavirus-ecuador.html> (accessed Jul. 09, 2021).
- [5] “Transporte público y COVID-19: Mensajes clave Mensajes clave.”
- [6] “Transportistas apoyan la campaña ‘Un pacto para vivir’, en Ambato - *El Comercio*.” <https://www.elcomercio.com/actualidad/transportistas-campana-ambato-covid-contagio.html> (accessed Jul. 09, 2021).
- [7] B. Amaguaña, “BryanAmaguañaPerfil.”
- [8] E. M. Calidad, “certificado de aceptación del tutor.”
- [9] Bajaña Franco Rafael Ernerto, “estudio de factibilidad para la aplicacion de tarjetas electronica de transporte publico, con la finalidad de mejorar el rendimiento administrativo, financiero y operativo de la cooperativa intraprovincial ‘rutas milagreñas del canton milagro,”” Apr. . [Online]. Available:
[http://repositorio.unemi.edu.ec/bitstream/123456789/2009/1/Estudio de factibilidad para la aplicación de tarjeta electrónica de transporte público%2C con la finalidad de mejorar el rendimiento administrativo%2C financiero y operativo de la cooperativa.pdf](http://repositorio.unemi.edu.ec/bitstream/123456789/2009/1/Estudio%20de%20factibilidad%20para%20la%20aplicaci%C3%B3n%20de%20tarjeta%20electr%C3%B3nica%20de%20transporte%20p%C3%BAblico%20con%20la%20finalidad%20de%20mejorar%20el%20rendimiento%20administrativo%20financiero%20y%20operativo%20de%20la%20cooperativa.pdf).

- [10] Yacelga Lita and Inti Atahualpa, “universidad técnica del norte ‘flota anteaña’ de la ciudad de atuntaqui proyecto previo a la obtención del título de ingeniero en.” [Online]. Available: [http://repositorio.utn.edu.ec/bitstream/123456789/7192/1/04 RED 100 TRABAJO GRADO.pdf](http://repositorio.utn.edu.ec/bitstream/123456789/7192/1/04_RED_100_TRABAJO_GRADO.pdf).
- [11] “Definición y cómo funcionan las aplicaciones móviles.” https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/#Que_es_una_aplicacion_movil (accessed Jun. 03, 2021).
- [12] “Capítulo 1: Las aplicaciones – Diseñando apps para móviles.” <https://appdesignbook.com/es/contenidos/las-aplicaciones/> (accessed Jun. 03, 2021).
- [13] “Qué es un framework y para qué se utiliza - *dipe*.” <https://blog.dipe.es/blog/que-es-un-framework-y-para-que-se-utiliza/> (accessed Jun. 04, 2021).
- [14] “Flutter: framework multiplataforma para desarrollar apps - *IONOS*.” <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-flutter/> (accessed Jun. 04, 2021).
- [15] “Flutter I: Introducción al desarrollo | by Oscar Caballero | Oscar | *Medium*.” <https://medium.com/maybeoscar/flutter-i-introducción-al-desarrollo-8c2250d60186> (accessed Jun. 04, 2021).
- [16] “Qué es Ionic: ventajas y desventajas de usarlo en apps móviles híbridas.” <https://profile.es/blog/que-es-ionic/> (accessed Jun. 04, 2021).
- [17] “✓Qué es Ionic | Quality Devs | Somos Desarrolladores.” <https://www.qualitydevs.com/2019/05/31/que-es-ionic-desarrollador-web/> (accessed Jun. 04, 2021).
- [18] “¿Qué es Ionic y por qué es tan conveniente para el desarrollo de Apps? - *Globalbit*.” <https://www.globalbit.co/2020/08/14/que-es-ionic-y-por-que-es-tan-conveniente-para-el-desarrollo-de-apps/> (accessed Jun. 04, 2021).
- [19] “¿Qué es iónico?” <https://ionicframework.com/what-is-ionic> (accessed Jun. 04, 2021).
- [20] “¿Qué es Xamarin? - *Xamarin | Documentos de Microsoft*.” <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>

(accessed Jun. 04, 2021).

- [21] “¿Qué es React Native?” <https://raullg.com/wiki/que-es-react-native/> (accessed Jun. 05, 2021).
- [22] “React Native: ¿Qué es y para que sirve? | *OpenWebinars*.” <https://openwebinars.net/blog/react-native-que-es-para-que-sirve/> (accessed Jun. 05, 2021).
- [23] “SQL, NoSQL, NewSQL. Qué son, historia y elección.” <https://www.ithinkupc.com/es/blog/sql-nosql-newsql-que-son-historia-y-eleccion/> (accessed Jun. 05, 2021).
- [24] “¿Qué es el software libre? - Proyecto GNU - *Free Software Foundation*.” <https://www.gnu.org/philosophy/free-sw.es.html> (accessed Jun. 06, 2021).
- [25] “Qué es MongoDB y características | *OpenWebinars*.” <https://openwebinars.net/blog/que-es-mongodb/> (accessed Jun. 05, 2021).
- [26] “Distribución de datos de múltiples nubes | *MongoDB*.” <https://www.mongodb.com/cloud/atlas/multicloud-data-distribution> (accessed Jun. 05, 2021).
- [27] “¿Qué es una API de REST?” <https://www.redhat.com/es/topics/api/what-is-a-rest-api> (accessed Jun. 05, 2021).
- [28] “Acerca | *Node.js*.” <https://nodejs.org/es/about/> (accessed Jun. 05, 2021).
- [29] “Qué es NodeJS y para qué sirve | *OpenWebinars*.” <https://openwebinars.net/blog/que-es-nodejs/> (accessed Jun. 05, 2021).
- [30] “Introducción a express js. Antes de comenzar a utilizar express se... | by *Aarón López Sosa* | Medium.” <https://medium.com/@aarnlpezsosa/introducción-a-express-js-a1ebe16dbcf4> (accessed Jun. 05, 2021).
- [31] “Introducción a Express.js (Framework Node.js) – *Programación*.” <https://www.programacion.com.py/web/javascript/introduccion-a-express-js-node-js> (accessed Jun. 06, 2021).
- [32] “React js - *EcuRed*.” https://www.ecured.cu/React_js (accessed Jun. 06, 2021).
- [33] “Documentación para Visual Studio Code.” <https://code.visualstudio.com/docs> (accessed Jun. 06, 2021).

- [34] “*GitHub: ¿Qué Es GitHub Y Cómo Utilizarlo?*” <https://www.hostinger.es/tutoriales/que-es-github> (accessed Jun. 08, 2021).
- [35] “*Qué es el control de versiones | Atlassian Git Tutorial.*” <https://www.atlassian.com/es/git/tutorials/what-is-version-control> (accessed Jul. 09, 2021).
- [36] “*Heroku - Wikipedia, la enciclopedia libre.*” <https://es.wikipedia.org/wiki/Heroku> (accessed Jul. 09, 2021).
- [37] “*que es heroku, para qué sirve, ventajas y desventajas – videlcloud.*” <https://videlcloud.wordpress.com/2018/12/22/que-es-heroku-para-que-sirve-ventajas-y-desventajas/> (accessed Jun. 08, 2021).
- [38] “*Definición de red inalámbrica - Qué es, Significado y Concepto.*” <https://definicion.de/red-inalambrica/> (accessed Jun. 21, 2020).
- [39] “*Rfid: qué es y cómo funciona | VIU.*” <https://www.universidadviu.com/pe/actualidad/nuestros-expertos/rfid-que-es-y-como-funciona> (accessed Jun. 08, 2021).
- [40] S. DE Recaudo En Base A Recargas De Dinero Mediante El and E. Y. Redes De Comunicación, “universidad técnica del norte ‘flota anteña’ de la ciudad de atuntaqui proyecto previo a la obtención del título de ingeniero en.” Accessed: Jun. 03, 2021. [Online]. Available: [http://repositorio.utn.edu.ec/bitstream/123456789/7192/1/04 RED 100 TRABAJO GRADO.pdf](http://repositorio.utn.edu.ec/bitstream/123456789/7192/1/04_RED_100_TRABAJO_GRADO.pdf).
- [41] “*Etiquetas RFID: qué son y qué aplicaciones tienen | Dipole.*” <https://www.dipolerfid.es/blog/etiquetas-rfid-y-aplicaciones> (accessed Jul. 09, 2021).
- [42] “*¿Qué son las etiquetas RFID UHF? | EAE.*” <https://retos-operaciones-logistica.eae.es/que-son-las-etiquetas-rfid-uhf-o-de-frecuencia-ultraelevada/> (accessed Jun. 10, 2021).
- [43] “*Chip RFID: qué es y cuál es su aplicación en la industria - Química.*” <https://iquimicas.com/chip-rfid-aplicacion-la-industria/> (accessed Jun. 10, 2021).
- [44] “*Etiqueta RFID activas - T2EB - Extronics Ltd - para batería / industrial.*” <https://www.directindustry.es/prod/extronics-ltd/product-118355->

2181839.html (accessed Jun. 10, 2021).

- [45] “Dlr-tl001 Semi-pasiva Rfid Uhf Registrador De Temperatura Etiqueta - Buy Dlr-tl001 Semi-passive,Uhf Rfid Temperature Logger Tag Product on *Alibaba.com*.” <https://spanish.alibaba.com/product-detail/dlr-tl001-semi-passive-uhf-rfid-temperature-logger-tag-62570746018.html> (accessed Jun. 10, 2021).
- [46] “RFID | Identificación por radiofrecuencia - *IONOS*.” <https://www.ionos.es/digitalguide/servidores/know-how/rfid/> (accessed Jun. 10, 2021).
- [47] “Características Técnicas Tarjetas de Proximidad Mifare® llave incopiable.” https://www.huellasdigitales.com.mx/tarjeta_proximidad_mifare.html (accessed Jul. 09, 2021).
- [48] “¿Qué es una tarjeta MIFARE? Fabricación y venta a precios de descuento.” <https://a3m.eu/es/tarjeta-mifare> (accessed Jun. 10, 2021).
- [49] “Mifare - *Wikipedia*, la enciclopedia libre.” <https://es.wikipedia.org/wiki/Mifare> (accessed Jun. 10, 2021).
- [50] “Tarjeta Mifare 1K IC S50 El mejor precio en línea BarMax Monterrey, Mexico.” <https://www.barmax.com/proddetail.asp?IDM=275&CatID=409&Prod=MifareCard> (accessed Jun. 10, 2021).
- [51] “Tecnología NFC: ¿para qué sirve?” <https://www.ocu.org/tecnologia/telefono/noticias/tecnologia-nfc#> (accessed Jun. 08, 2021).
- [52] “RFID: qué es, diferencias con NFC y aplicación en eventos | *CASFID*.” <https://www.casfid.es/es/blog/que-es-rfid-nfc> (accessed Jul. 09, 2021).
- [53] “Formato de intercambio de datos NFC NDEF »*Notas electrónicas*.” <https://www.electronics-notes.com/articles/connectivity/nfc-near-field-communication/data-exchange-format-ndef.php> (accessed Jun. 11, 2021).
- [54] “¿Qué es Arduino? | *Arduino.cl* - Compra tu Arduino en Línea.” <https://arduino.cl/que-es-arduino/> (accessed Jun. 11, 2021).
- [55] “▷ Arduino Mega 2560 Características, Especificaciones | *Proyecto Arduino*.” <https://proyectoarduino.com/arduino-mega-2560/> (accessed Jun. 11, 2021).

- [56] T. Inteligentes, “Tecnologías. Tarjetas Inteligentes,” *rnds, Argentina*, vol. 146, pp. 1–2, 1999.
- [57] L. Gustavo, “Medios De Transporte Urbano,” p. 40, 2017, Accessed: Jul. 06, 2020. [Online]. Available: <http://ingenieria.uncuyo.edu.ar/catedras/u1-medios-de-transporte-urbano.pdf>.
- [58] “Derecho Ecuador - *El Servicio Público*.” <https://www.derechoecuador.com/el-servicio-publico> (accessed Jul. 06, 2020).
- [59] “Formulario sin título - *Formularios de Google*.” <https://docs.google.com/forms/d/1rPiNq8gF5KgLQ5gBswjpbMoS3jDJf15VaEc99FebxNg/edit#responses> (accessed Jun. 17, 2021).
- [60] “Conoce las metodologías de desarrollo ágil más usadas.” <https://www.grupocibernos.com/blog/conoce-las-metodologias-de-desarrollo-agil-mas-usadas> (accessed Jun. 18, 2021).
- [61] “Metodología Scrum: qué es y cómo funciona.” <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html> (accessed Jun. 18, 2021).
- [62] “Scrum: roles y responsabilidades | *Deloitte España*.” <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html> (accessed Jul. 09, 2021).
- [63] C. DE Ingeniería En Sistemas Y Computación, E. Dayana Yépez Llerena Kleber Fabián Armijos Guillen, and I. Pamela Buñay MsC, “universidad nacional de chimborazo facultad de ingeniería.”
- [64] “¿Qué roles son necesarios para los equipos Kanban?” <https://itnove.com/blog/kanban/que-roles-son-necesarios-para-los-equipos-kanban/> (accessed Jun. 18, 2021).
- [65] “¿Qué es un diagrama de Gantt y para qué sirve? | *OBS Business School*.” <https://www.obsbusiness.school/blog/que-es-un-diagrama-de-gantt-y-para-que-sirve> (accessed Jul. 09, 2021).
- [66] “Tableros kanban: conceptos - *Scrum Manager BoK*.” https://www.scrummanager.net/bok/index.php?title=Tableros_kanban:_conceptos (accessed Jun. 18, 2021).

- [67] “Descubre qué es el Extreme Programming y sus características.” <https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum/> (accessed Jun. 18, 2021).
- [68] “CyTA.” <http://www.cyta.com.ar/ta0502/v5n2a1.htm> (accessed Jun. 19, 2021).
- [69] “Extreme Programming | Principios, Técnicas, Ventajas e Inconvenientes - IONOS.” <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/extreme-programming/> (accessed Jun. 19, 2021).
- [70] “React, Ionic, Flutter - Explorar - Google Trends.” https://trends.google.es/trends/explore?q=%2Fm%2F01211vxv,%2Fg%2F1q6l_n0n0,%2Fg%2F11f03_rzbg (accessed Jun. 20, 2021).
- [71] “React Native: Ventajas y desventajas reveladas | Back4App Blog.” <https://blog.back4app.com/es/react-native-ventajas-y-desventajas-reveladas/> (accessed Jun. 20, 2021).
- [72] “laravel, Node js, Firebase - Explorar - Google Trends.” [https://trends.google.es/trends/explore?geo=EC&q=laravel,Node js,Firebase](https://trends.google.es/trends/explore?geo=EC&q=laravel,Node%20js,Firebase) (accessed Jun. 21, 2021).
- [73] “Qué es Laravel: Características y ventajas | OpenWebinars.” <https://openwebinars.net/blog/que-es-laravel-caracteristicas-y-ventajas/> (accessed Jun. 21, 2021).
- [74] “Un vistazo a la plataforma de Firebase.” <https://www.clubdetecnologia.net/blog/2017/un-vistazo-a-la-plataforma-de-firebase/> (accessed Jun. 21, 2021).
- [75] “Introducción a NodeJS (JavaScript del lado del Servidor) - Oscar Blancarte - Software Architecture.” <https://www.oscarblancarteblog.com/2017/05/29/introduccion-a-nodejs-2/> (accessed Jun. 21, 2021).
- [76] “firebase vs. Laravel vs. Node.js | G2.” <https://www.g2.com/compare/firebase-vs-laravel-vs-node-js> (accessed Jun. 21, 2021).
- [77] “mongodb, Cassandra, Redis - Explorar - Google Trends.” <https://trends.google.es/trends/explore?geo=EC&q=mongodb,Cassandra,Redis> (accessed Jun. 21, 2021).
- [78] “Redis para principiantes - BI Geek Blog.” <https://blog.bi-geek.com/redis-para->

principiantes/ (accessed Jun. 21, 2021).

- [79] “Qué es Apache Cassandra | OpenWebinars.” <https://openwebinars.net/blog/que-es-apache-cassandra/> (accessed Jun. 21, 2021).
- [80] *M. Viñas Gutiérrez and O. J. Sellarès*, “Control de acceso mediante NFC con Arduino.”

ANEXOS Y APÉNDICES

Anexo A

Encuesta realizada a los usuarios que hacen uso del sistema de transporte urbano de la cooperativa “Unión Ambateña”

ENCUESTA REALIZADA A LOS USUARIOS QUE HACEN USO DEL SISTEMA DE TRANSPORTE URBANO DE LA COOPERATIVA “UNION AMBATEÑA”



Objetivo: Establecer las necesidades de los usuarios que utilizan el transporte urbano.

Instrucciones: Responda cada pregunta detallada a continuación.

Pregunta 1 ¿Con la llegada de la pandemia con qué frecuencia utiliza el transporte urbano?

- 1-2 veces al día
- 3-5 veces al día
- Más de 5 veces al día

Pregunta 2 ¿Está conforme con el método de pago que realiza al subirse al transporte público?

- Si
- No

Pregunta 3 ¿Le resulta cómodo el intercambio de dinero entre usted y el señor conductor?

- Si
- No

Pregunta 4 ¿Preferiría un método de pago del pasaje donde no tenga contacto alguno con el conductor existiendo un distanciado considerable?

- Si
- No

Pregunta 5 ¿Le resultaría más fácil realizar el pago del transporte público utilizando una tarjeta recargable sin preocuparse de llevar el dinero justo o preocuparse del vuelto?

- Si
- No

Pregunta 6 ¿Está de acuerdo con la idea de acercarse a un lugar asignado para la recarga de su tarjeta para el pago del transporte público?

- Si
- No

Pregunta 7 ¿Le resultaría cómodo realizar el pago del transporte público acercando una tarjeta a un lector de tarjetas sin la necesidad de realizar algún contacto?

- Si
- No

Pregunta 8 ¿Le resultaría útil la utilización de una aplicación móvil donde pueda verificar los pagos y recargas que realice con su tarjeta?

- Si
- No

Pregunta 9 ¿Está de acuerdo con acatar la nueva modalidad sobre el correcto funcionamiento del cobro de pasajes con tarjetas inteligentes?

- Si
- No

Pregunta 10 ¿Qué tan conforme estaría con la automatización del cobro del pasaje en el transporte público?

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- Desacuerdo
- Totalmente en desacuerdo

Anexo B

Entrevista dirigida al jefe del departamento de sistemas.

ENTREVISTA DIRIGIDO AL JEFE DEL DEPARTAMENTO DE SISTEMAS DE LA COOPERATIVA DE TRANSPORTE URBANO “UNION AMBATEÑA”



Objetivo: Conocer el estado actual de la cooperativa de transporte Unión Ambateña.

Instrucciones: Responda cada pregunta detallada a continuación.

Pregunta 1 ¿Considera usted que los ingresos de la cooperativa bajaron con la llegada de la pandemia?

Pregunta 2 ¿Es recomendable que se normalice las jornadas diarias de los buses de la cooperativa?

Pregunta 3 ¿le resultaría factible la idea de automatizar el cobro del pasaje?

Pregunta 4 ¿Disponen de un sistema que contabilice los ingresos de la cooperativa?

Pregunta 5 ¿La cooperativa dispone de servidores o una base de datos donde respalden información de los ingresos y egresos diarios?

Anexo C

C.1 Manual de Usuario

El siguiente manual detalla el correcto manejo de la aplicación web, móvil y de los dispositivos lectores de Tarjeta NFC.

Aplicación Web

- **Página principal**

Permite delegar toda la información de la cooperativa de transporte urbano “Unión Ambateña” dividida en secciones accesibles desde el MenuTop en la parte superior.

Enlace de Acceso: <https://coopunion.herokuapp.com>



Figura 124: Página principal

Elaborado por: Bryan Amaguaña

- **Página de acceso al sistema**

Para acceder al sistema es necesario autenticarse con el nombre de usuario y su respectiva contraseña.

Enlace de Acceso: <https://coopunion.herokuapp.com/admin>

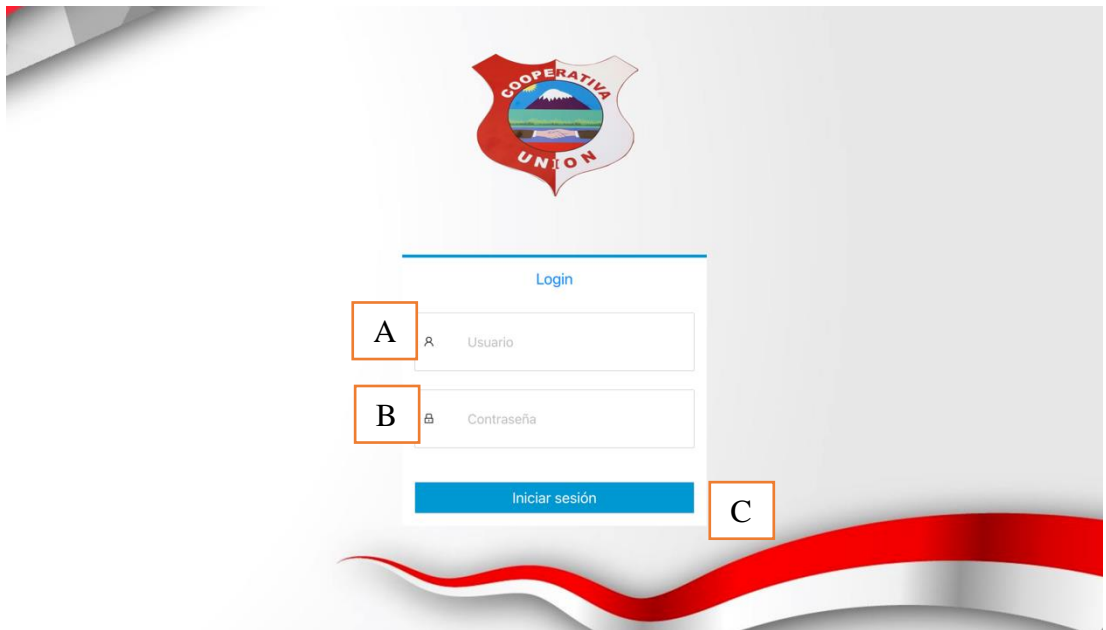


Figura 125: Página de inicio de sesión.

Elaborado por: Bryan Amaguaña

La página de inicio de sesión presenta un formulario con las siguientes opciones:

- A. Ingreso del nombre de usuario previamente registrado en el sistema
- B. Ingreso de la contraseña de autenticación
- C. Botón para verificar datos e ingresar al sistema

Al ingresar los datos de autenticación correctamente el sistema redireccionará la página al componente principal del programa.

- **Página principal del Admin**

Componente principal que desplegara un gráfico de barras de los ingresos al sistema de cada unidad de transporte y de cada recarga realizada por un usuario.



Figura 126: Página principal del Admin.

Elaborado por: Bryan Amaguaña

- A. Menú lateral izquierdo,** Contiene todos los componentes del sistema para una fácil navegación.
- B. Opciones de Búsquedas,** Incorpora 2 cuadros de texto para realizar una búsqueda por límites de fecha para desplegar en el gráfico de barras, de la misma forma incorpora una opción para generar un reporte con los datos filtrados.
- C. Gráfico de Barras,** Despliega todos los datos filtrados por la búsqueda, si no se realiza ninguna búsqueda el grafico desplegara los ingresos del mes actual hasta la fecha.

Componente Recargas de tarjetas incorporado en la página principal del Admin

En el mismo componente al deslizar la pestaña a la parte inferior se despliega un gráfico de barras con todas las recargas realizadas ordenadas por el nombre de usuario que realizo la recarga, permitiendo visualizar el usuario que más recargas a realizado en el transcurso del mes o del día.



Figura 127: Componente recargas de tarjetas del Admin

Elaborado por: Bryan Amaguaña

- A. Opciones de Búsquedas,** Incorpora 2 cuadros de texto para realizar una búsqueda por límites de fecha para desplegar en el gráfico de barras, de la misma forma incorpora una opción para generar un reporte con los datos filtrados.
- B. Gráfico de Barras,** Despliega todos los datos filtrados por la búsqueda, si no se realiza ninguna búsqueda el grafico desplegara los ingresos del mes actual hasta la fecha.

- **Componente Cobro Pasaje**

Componente diseñado para desplegar los datos detallados de cada cobro de pasaje realizado por los dispositivos lectores de tarjetas NFC incorporados en los buses de transporte urbano.

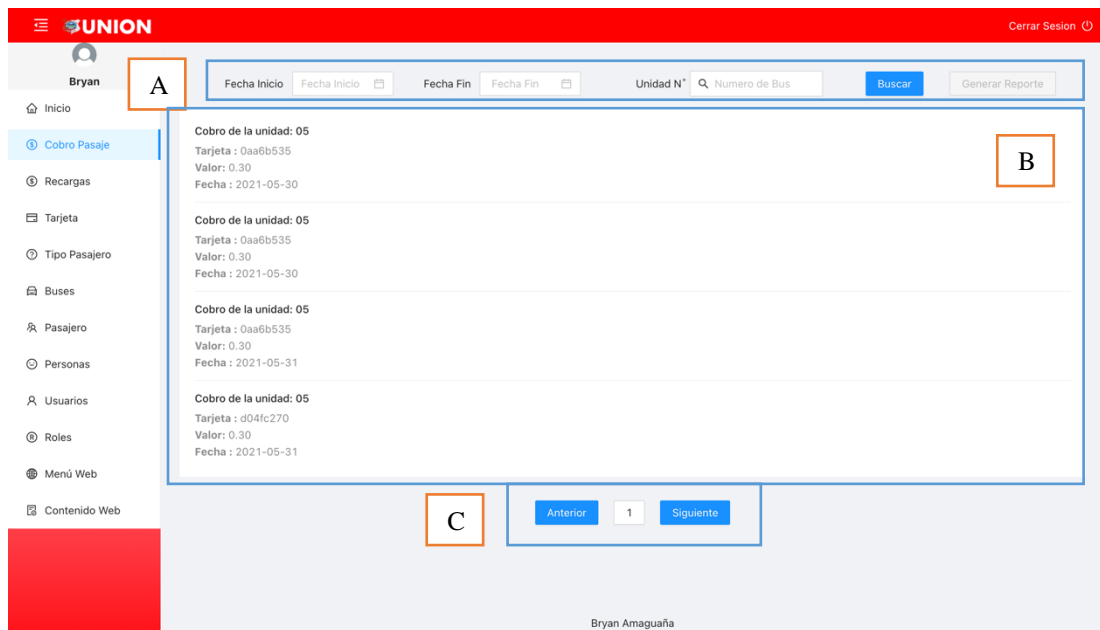


Figura 128: Componente Cobro Pasaje.

Elaborado por: Bryan Amaguaña

- A. Opciones de Búsquedas,** Incorpora 2 cuadros de texto para realizar una búsqueda por límites de fecha para desplegar la información en el apartado de información de cobros, de la misma forma incorpora una opción para generar un reporte con los datos filtrados.
- B. Información de cobros,** Despliega todos los datos filtrados por la búsqueda, si no se realiza ninguna búsqueda los datos desplegara los ingresos del mes actual hasta la fecha.
- C. Paginación,** Permite la navegación de información en pantalla, mostrando 4 cobros de pasajes a la vez.

- **Componente Recargas**

Componente diseñado para desplegar los datos detallados de cada recarga realizada por los dispositivos diseñados para recargar tarjetas NFC incorporados a las personas designadas.

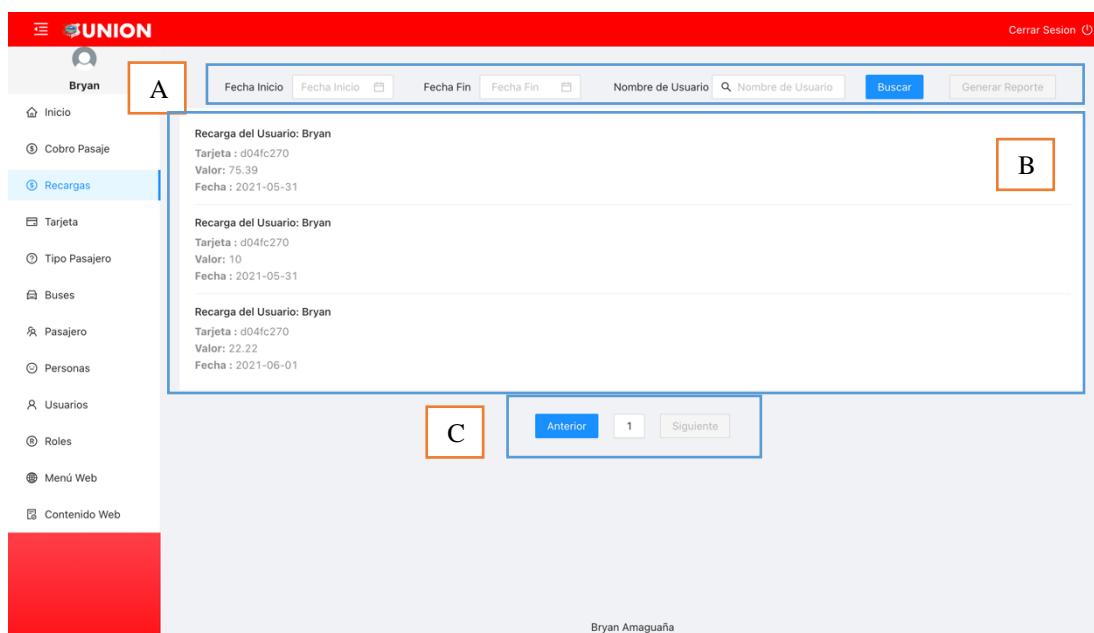


Figura 129: Componente Recargas de Tarjetas NFC.

Elaborado por: Bryan Amaguaña

- A. Opciones de Búsquedas,** Incorpora 2 cuadros de texto para realizar una búsqueda por límites de fecha para desplegar la información en el apartado de información de recargas, de la misma forma incorpora una opción para generar un reporte con los datos filtrados.
- B. Información de recargas,** Despliega todos los datos filtrados por la búsqueda, si no se realiza ninguna búsqueda los datos desplegara los ingresos del mes actual hasta la fecha.
- C. Paginación,** Permite la navegación de información en pantalla, mostrando 4 recargas de tarjetas a la vez.

- **Componente tarjetas activas**

Componente diseñado para desplegar los datos detallados de cada tarjeta registrada en el sistema que se encuentren en funcionamiento.

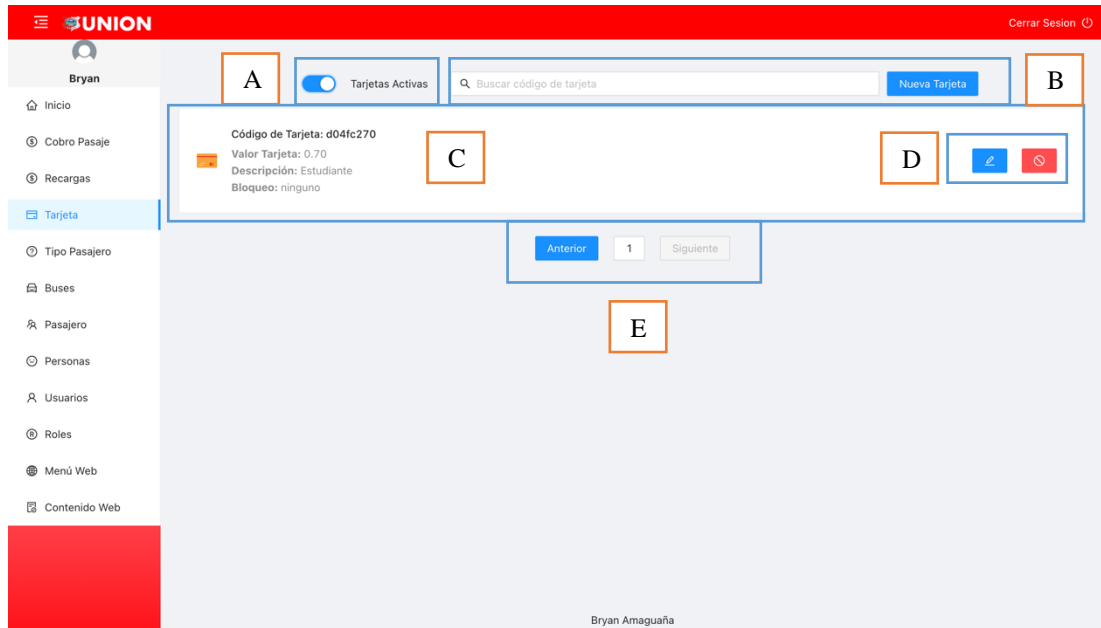


Figura 130: Componente tarjetas activas

Elaborado por: Bryan Amaguaña

- A. Cambio de formulario,** Permite visualizar el listado de tarjetas Activas e inactivas registrado en el sistema
- B. Opciones de búsqueda,** incorpora un cuadro de texto para ingresar el código de la tarjeta y realizar una búsqueda dinámica.
- C. Información de Tarjeta,** Despliega toda la información de cada tarjeta.
- D. Opciones de edición,** incorpora 2 opciones, edición y desactivar una tarjeta
- E. Paginación,** Permite la navegación de información en pantalla, mostrando 4 tarjetas a la vez.

- **Agregar una nueva tarjeta**

En el Componente tarjeta existe la opción de agregar Tarjeta donde nos permite adicionar una nueva tarjeta a la base de datos para ser utilizado en los pagos de pasaje mostrando el siguiente formulado donde deberes llenar los campos y validar la información.



El formulario 'Agregar nueva Tarjeta' contiene los siguientes elementos:

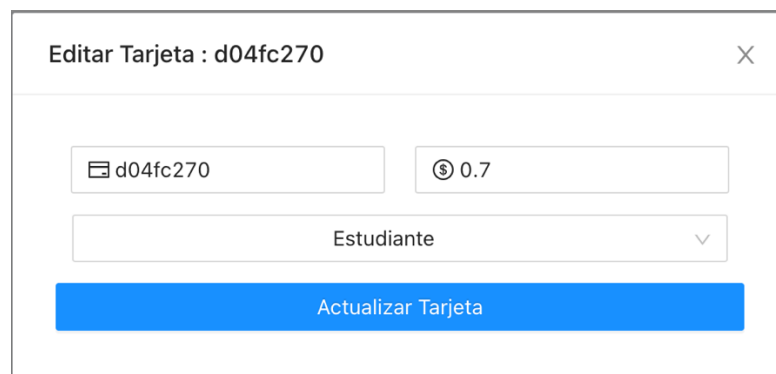
- Un campo de texto con un ícono de tarjeta y el texto 'Código'.
- Un campo de texto con un ícono de moneda y el texto 'Valor 00.00'.
- Un menú desplegable con el texto 'Seleccione tipo de Tarjeta'.
- Un botón azul con el texto 'Agregar Tarjeta'.

Figura 131: Componente Agregar Tarjetas

Elaborado por: Bryan Amaguaña

- **Editar una tarjeta existente**

En el componente tarjetas existe la opción para editar una tarjeta registrada en el sistema, donde nos permite editar el valor o el código de la tarjeta de tal forma que se puede corregir errores de inserción, o modificar tarjetas.



El formulario 'Editar Tarjeta : d04fc270' contiene los siguientes elementos:

- Un campo de texto con un ícono de tarjeta y el texto 'd04fc270'.
- Un campo de texto con un ícono de moneda y el texto '0.7'.
- Un menú desplegable con el texto 'Estudiante'.
- Un botón azul con el texto 'Actualizar Tarjeta'.

Figura 132: Componente Editar Tarjetas

Elaborado por: Bryan Amaguaña

- **Componente tarjetas inactivas**

Componente diseñado para desplegar los datos detallados de cada tarjeta inactiva registrada en el sistema.

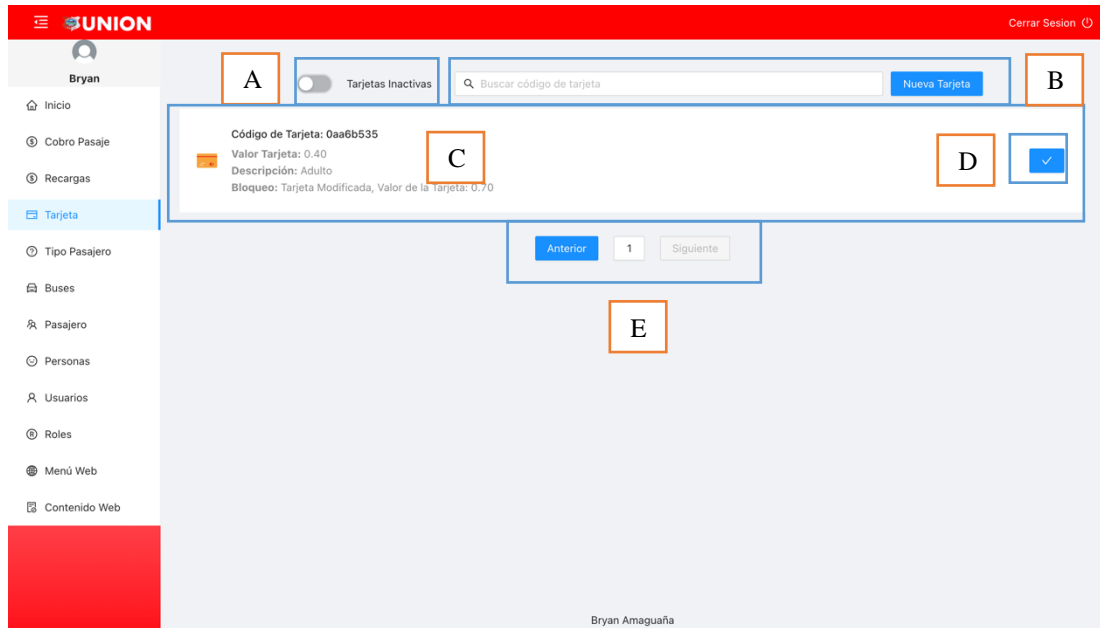


Figura 133: Componente tarjetas inactivas

Elaborado por: Bryan Amaguaña

- A. Cambio de formulario,** Permite visualizar el listado de tarjetas Activas e inactivas registrado en el sistema
- B. Opciones de búsqueda,** incorpora un cuadro de texto para ingresar el código de la tarjeta y realizar una búsqueda dinámica.
- C. Información de Tarjeta,** Despliega toda la información de cada tarjeta.
- D. Opciones de edición,** incorpora 1 opciones, activación de tarjeta
- E. Paginación,** Permite la navegación de información en pantalla, mostrando 4 tarjetas a la vez.

- **Componente Tipo de Pasaje**

Componente diseñado para desplegar los datos detallados de cada tipo de pasaje registrados en el sistema.

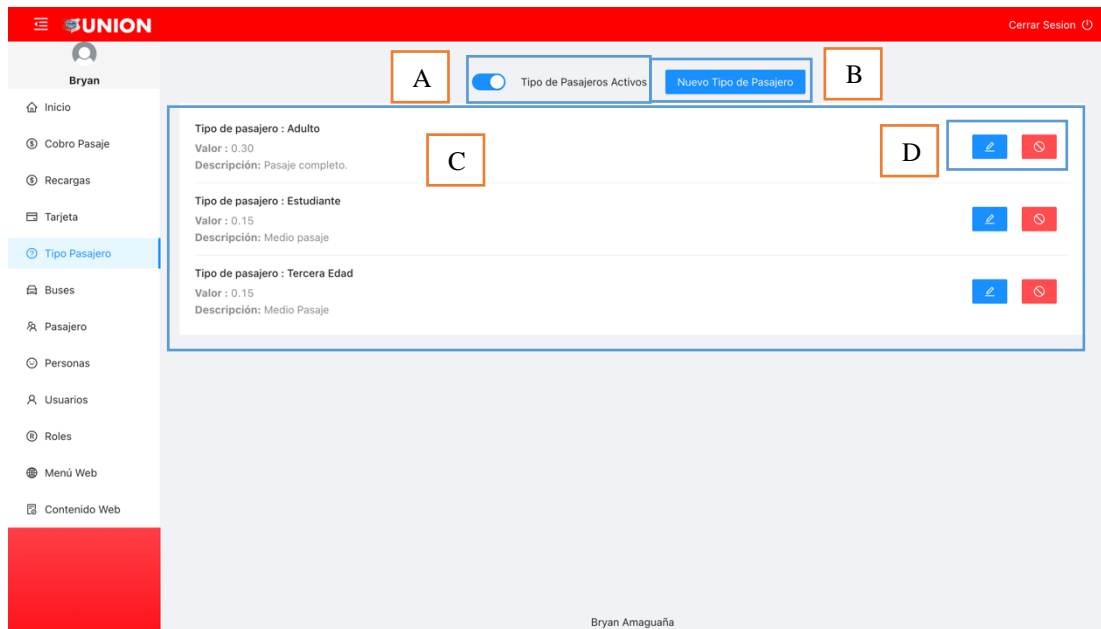


Figura 134: Componente Tipo de Pasajes activo

Elaborado por: Bryan Amaguaña

- A. Cambio de formulario,** Permite visualizar el listado de tipos de pasajeros activos e inactivas registrado en el sistema.
- B. Botón para agregar,** permite agregar un nuevo tipo de pasajero.
- C. Información,** Despliega toda la información de cada tipo de pasajero existente.
- D. Opciones de edición,** incorpora 2 opciones, desactivar tipo de pasajero y editar tipo de pasajero existente.

- **Componente Buses**

Componente diseñado para desplegar los datos detallados de cada bus registrado en el sistema.

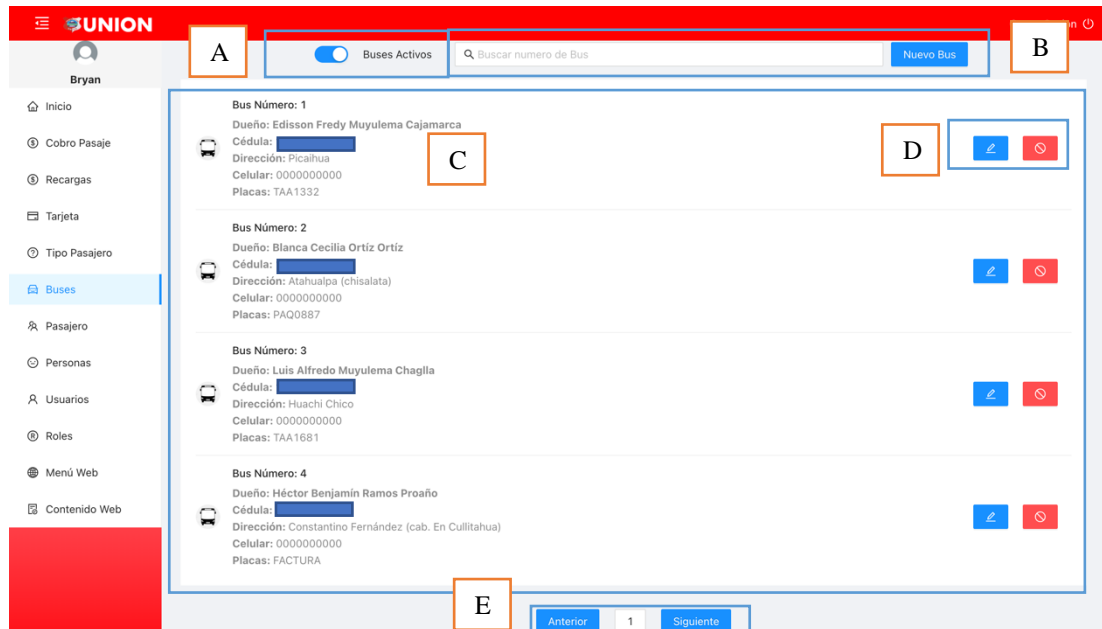


Figura 135: Componente Bus

Elaborado por: Bryan Amaguaña

- A. Cambio de formulario,** Permite visualizar el listado de buses Activos e inactivos registrado en el sistema
- B. Opciones de búsqueda,** incorpora un cuadro de texto para ingresar el número del bus y realizar una búsqueda dinámica, adicional incorpora un botón para agregar nuevos buses.
- C. Información de contenido,** Despliega toda la información de cada bus.
- D. Opciones de edición,** incorpora 2 opciones, desactivar y editar un bus
- E. Paginación,** Permite la navegación de información en pantalla, mostrando 4 buses a la vez.

- **Componente Pasajero**

Componente diseñado para desplegar los datos detallados de cada pasajero registrado en el sistema.

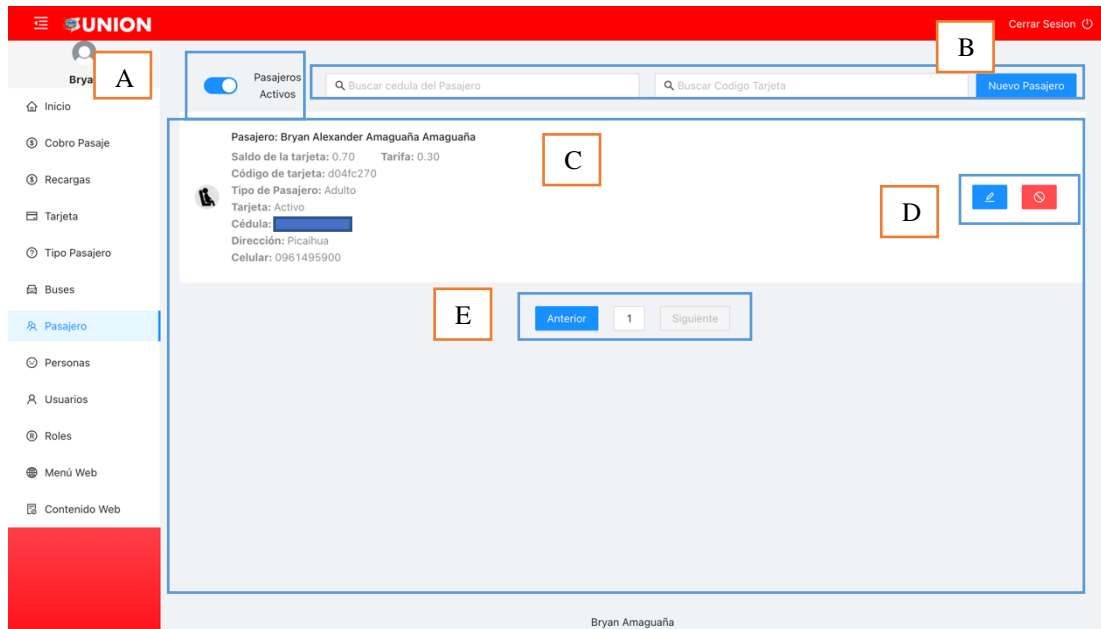


Figura 136: Componente Bus

Elaborado por: Bryan Amaguaña

- A. Cambio de formulario,** Permite visualizar el listado de pasajeros Activos e inactivos registrado en el sistema
- B. Opciones de búsqueda,** incorpora 2 cuadro de texto para ingresar el código de la tarjeta y la cedula del pasajero para realizar una búsqueda dinámica, adicional incorpora un botón para agregar nuevos pasajeros.
- C. Información de contenido,** Despliega toda la información de cada pasajero.
- D. Opciones de edición,** incorpora 2 opciones, desactivar y editar un pasajero
- E. Paginación,** Permite la navegación de información en pantalla, mostrando 4 buses a la vez.

- **Componente Personas**

Componente diseñado para desplegar los datos detallados de cada persona registrada en el sistema.

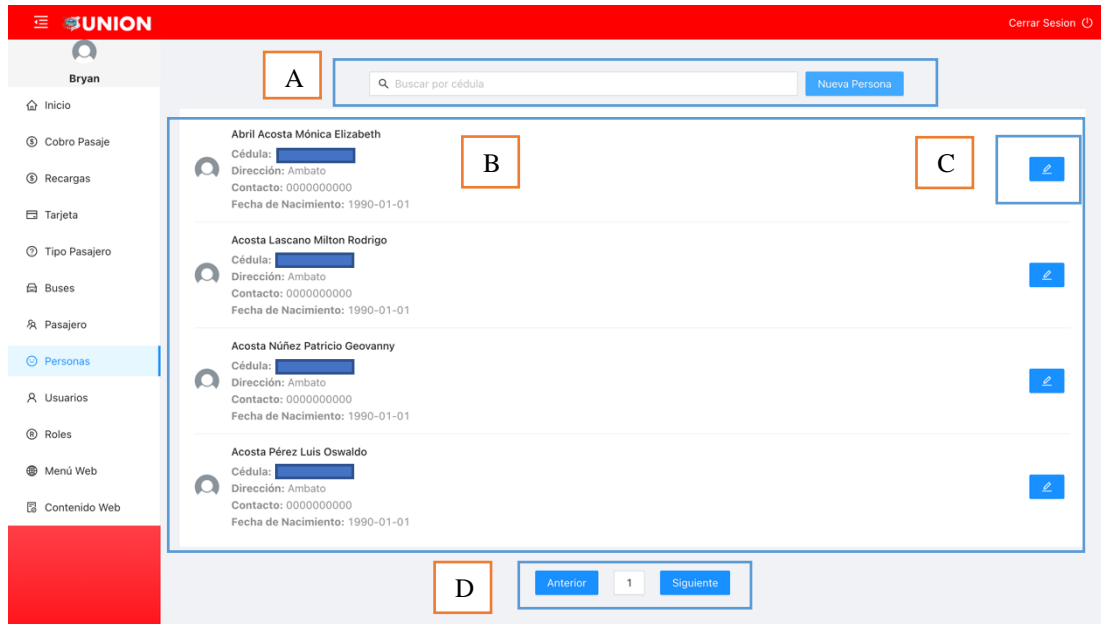


Figura 137: Componente Personas

Elaborado por: Bryan Amaguaña

- A. Opciones de búsqueda,** incorpora un cuadro de texto para ingresar la cedula de la persona para realizar una búsqueda dinámica, adicional incorpora un botón para agregar nuevas personas al sistema.
- B. Información de contenido,** Despliega toda la información de cada persona registrada.
- C. Opciones de edición,** incorpora una opción, editar una persona.
- D. Paginación,** Permite la navegación de información en pantalla, mostrando 4 buses a la vez.

- **Componente Usuarios**

Componente diseñado para desplegar los datos detallados de cada usuario registrado en el sistema.

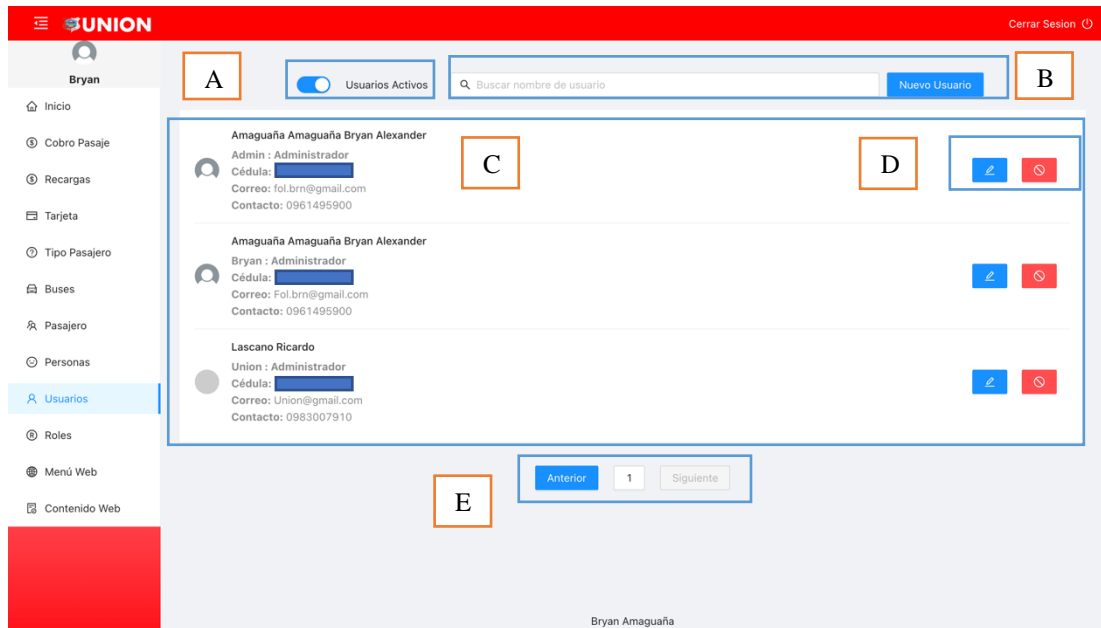


Figura 138: Componente Usuarios

Elaborado por: Bryan Amaguaña

- A. Cambio de formulario,** Permite visualizar el listado de usuarios activos e inactivos registrado en el sistema
- B. Opciones de búsqueda,** incorpora un cuadro de texto para ingresar el nombre de usuario y realizar una búsqueda dinámica, adicional incorpora un botón para agregar nuevos usuarios.
- C. Información de contenido,** Despliega toda la información de cada bus.
- D. Opciones de edición,** incorpora 2 opciones, desactivar y editar un usuario
- E. Paginación,** Permite la navegación de información en pantalla, mostrando 4 buses a la vez.

- **Componente Rol**

Componente diseñado para desplegar los datos detallados de cada rol registrado en el sistema.

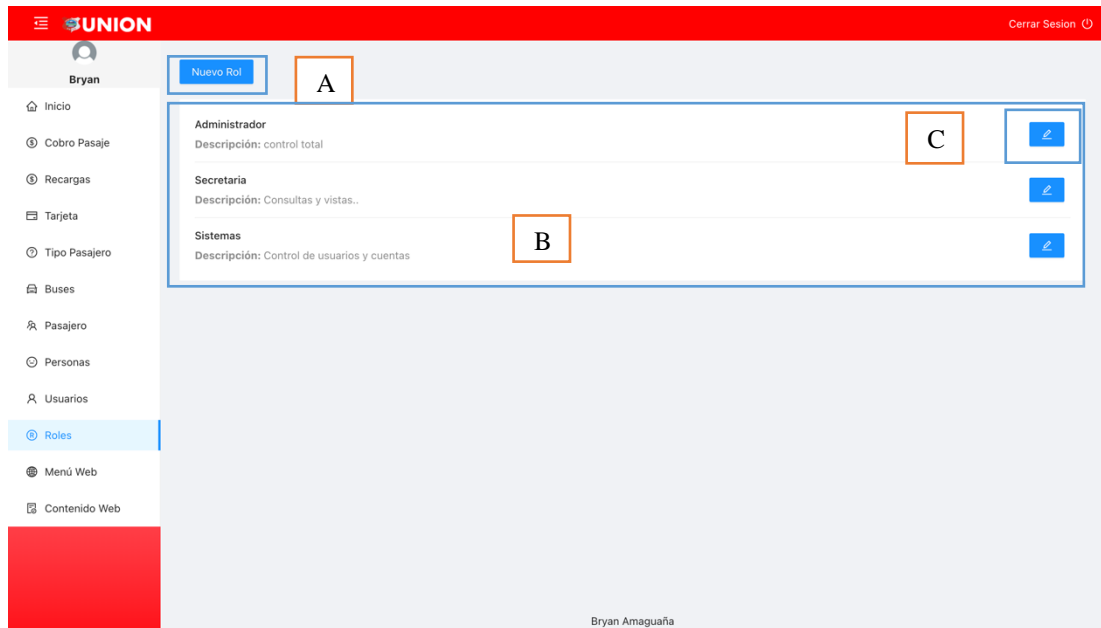


Figura 139: Componente Rol

Elaborado por: Bryan Amaguaña

- A. Agregar nuevo rol,** Permite agregar un nuevo rol al sistema
- B. Información de contenido,** Despliega toda la información de cada rol.
- C. Opciones de edición,** incorpora una opción, editar un rol

- **Componente Menú Web**

Componente diseñado para desplegar los datos detallados de cada menú que incorpora la página principal.

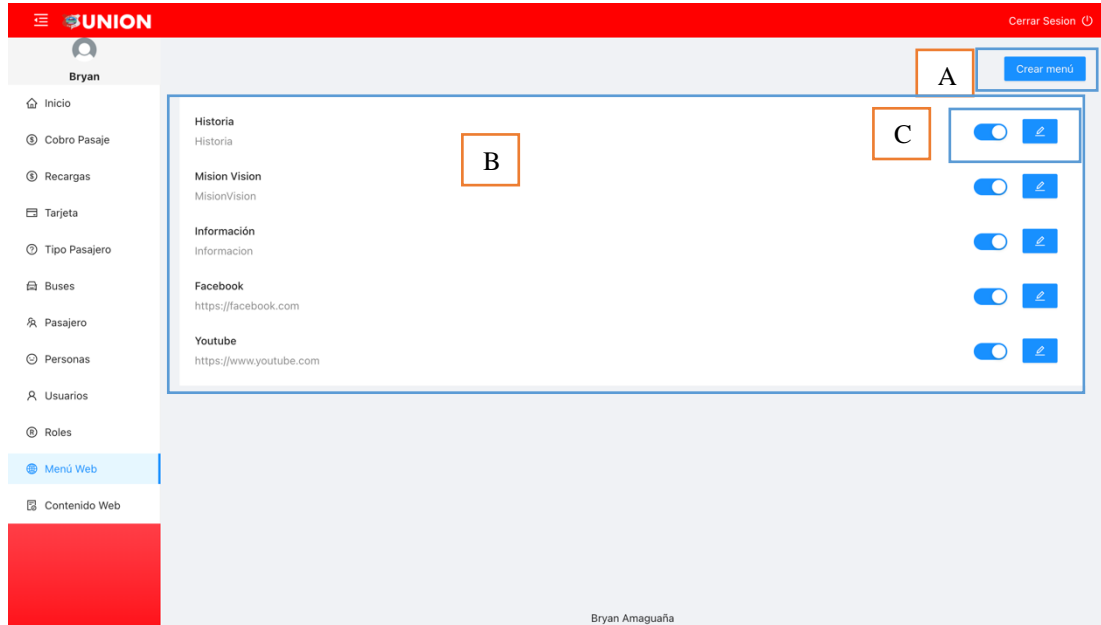


Figura 140: Componente Usuarios

Elaborado por: Bryan Amaguaña

- A. Agregar nuevo menú,** Permite agregar un nuevo menú a la página principal el sistema.
- B. Información de contenido,** Despliega toda la información de cada menú.
- C. Opciones de edición,** incorpora 2 opciones, editar un menú y desactivar un menú de la página principal del sistema.

- **Componente Contenido Web**

Componente diseñado para desplegar los datos detallados de cada información desplegada en la página principal del sistema disponible para el público.

The screenshot displays the 'UNION' web application interface. On the left, a sidebar lists navigation options: Inicio, Cobro Pasaje, Recargas, Tarjeta, Tipo Pasajero, Buses, Pasajero, Personas, Usuarios, Roles, and Menú Web. The 'Contenido Web' section is highlighted in blue. The main content area shows a form for editing website information. A red box labeled 'A' highlights the 'Cooperativa' field, which contains the text 'Cooperativa de Transporte Urbano Union Am'. Other fields include 'Correo' (Union0000@gmail.com), 'Dirección' (Huachi Grande), 'Teléfono' (034125423), 'Celular' (0000000000), 'Fax' (55555555), 'Mensaje de Inicio' (La union hace la fuerza), 'Mensaje Secundario' (Bienvenido), 'Descripción' (Trabajamos de Domingo a Domingo), 'Misión' (MISIÓN, la cooperativa de transporte urbano "UNIÓN AMBATEÑA", es una organización que realiza en forma permanente el servicio Urbano de transporte de pasajeros, en el cantón Ambato, y sus parroquias con unidades motorizadas en perfectas condiciones mecánicas, con confort y seguridad, manteniendo como principios entre sus asociados la IGUALDAD, SOLIDARIDAD, COMPAÑERISMO, y DEMOCRACIA), and 'Visión' (VISIÓN, ser una operadora de Transporte Urbano de pasajeros, competitiva por excelencia, poniendo al servicio de la ciudadanía a la que sirve, sus modernas unidades, cómodas y confortables, dando un óptimo servicio a la colectividad, con RESPETO, SEGURIDAD, CONFORT, y EFICIENCIA, de acuerdo con los planes). Below the form is a blue button labeled 'Actualizar Contenido' and a red box labeled 'B' highlights this button. The footer of the page shows the name 'Bryan Amaguaña'.

Figura 141: Componente Contenido Web

Elaborado por: Bryan Amaguaña

- A. Formulario,** Permite agregar información disponible para desplegar en la página principal del sistema.
- B. Botón de actualización,** al realizar un cambio en el formulario es necesario confirmar la actualización de los cambios en el sistema.

Aplicación Móvil.

- **Componente login y Registro de pasajero**

Componente diseñado para registrar un pasajero con sus datos personales y el código de la tarjeta, una vez registrado se direcciona al login para ingresar a la aplicación.

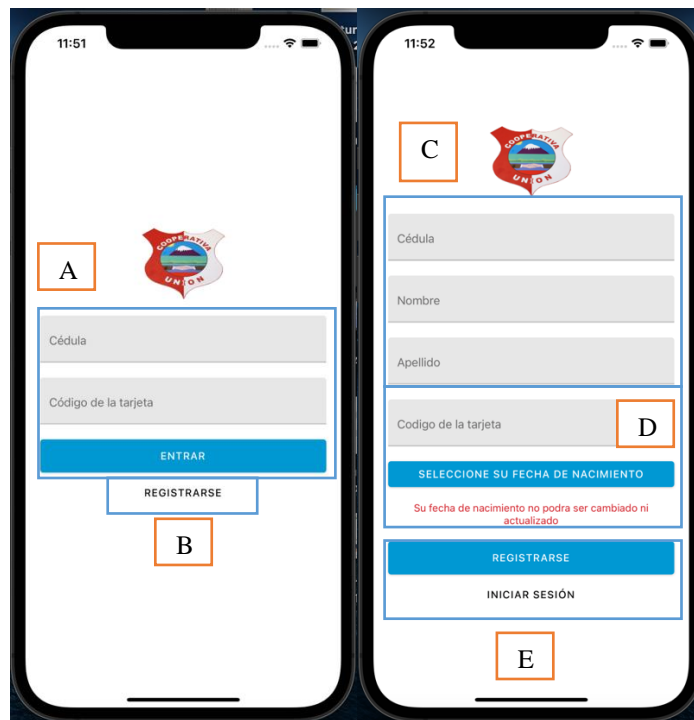


Figura 142: Componente Login y registro de pasajero

Elaborado por: Bryan Amaguaña

- A. Formulario para acceder al sistema**, requiere del número de cedula y el código de la tarjeta para acceder al sistema
- B. Opciones registro**, dirige al componente de registro de pasajero
- C. Formulario para registrar una persona**, requiere de información de la persona
- D. Formulario para registrar una tarjeta**, requiere del código de la tarjeta para completar el registro
- E. Botón de registro**, permite validar y registrar un pasajero en el sistema, cuenta con un enlace de “INICIAR SESION” de dirige a la página de login.

- **Componente principal al momento de ingresar a la aplicación**

Componente diseñado para desplegar toda la información del pasajero y de la tarjeta registrada en el sistema.



Figura 143: Componente principal de la aplicación.

Elaborado por: Bryan Amaguaña

- A. Imagen de presentación,** despliega el nombre del pasajero, e información de la tarjeta.
- B. Formulario de información del pasajero,** despliega información registrada por el usuario pasajero.
- C. Formulario de información de la tarjeta,** despliega información de la tarjeta registrada.
- D. Menú de navegación,** integra todas las opciones que posee la aplicación para una navegación agradable para el pasajero.

- **Componente de pagos y recargas de tarjetas**

Componente diseñado para desplegar los pagos y las recargas realizadas con la tarjeta registrada en el sistema filtrado por fechas.

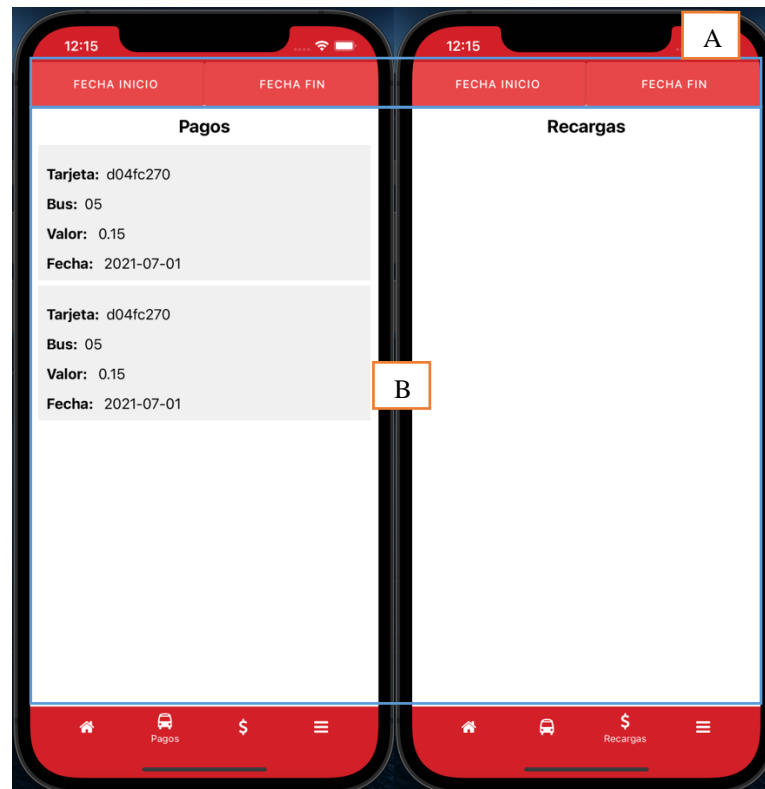


Figura 144: Componente pago y recarga de tarjeta

Elaborado por: Bryan Amaguaña

- A. Opciones de búsqueda de datos**, incorpora 2 opciones para filtrar los datos a mostrar de acuerdo con la fecha inicial y a una fecha final
- B. Formulario de información de datos**, despliega información registrada de las actividades que realiza la tarjeta como los pagos y las recargas de acuerdo con las fechas seleccionadas para la búsqueda.

- **Componente mi cuenta**

Componente diseñado para desplegar opciones de configuración del pasajero.

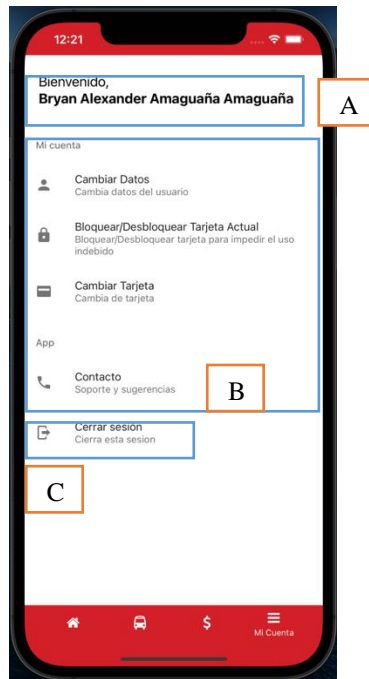


Figura 145: Componente mi cuenta

Elaborado por: Bryan Amaguaña

A. Información del pasajero, despliega un saludo para el usuario registrado.

B. Formulario de opciones, despliega múltiples opciones para la gestión de la información del usuario y de la tarjeta.

C. Cerrar Sesión. Cierra correctamente la sesión activada en la aplicación, regresando al login y permitiendo ingresar a otra cuenta.

- **Componente para editar datos personales y bloquear tarjeta**

Componente diseñado para editar los datos del pasajero y para bloquear la tarjeta actual registrada en el sistema.

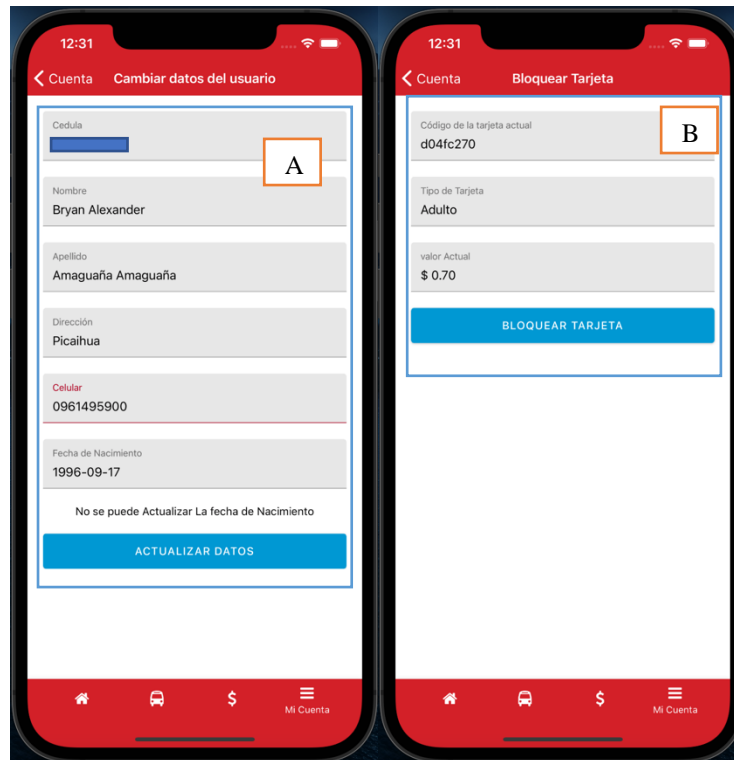


Figura 146: Componente editar datos personales y bloquear tarjeta

Elaborado por: Bryan Amaguaña

- A. Formulario Editar Pasajero**, despliega un formulario para editar la información del pasajero proporcionada el momento de realizar el registro en la aplicación.
- B. Formulario de Bloqueo de tarjetas**, despliega un formulario con los datos de la tarjeta para verificar la tarjeta que se va a bloquear para el uso indebido de la tarjeta en caso de pérdida o robo.

- **Componente para cambiar de tarjeta y página de contacto**

Componente diseñado para cambiar de tarjeta y página de contacto para informar cualquier inconveniente con la tarjeta o con la aplicación.

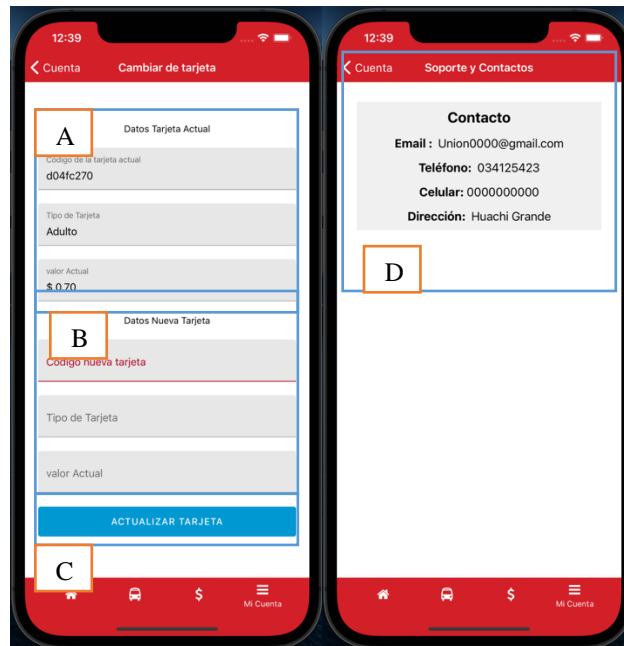


Figura 147: Componente cambiar de tarjeta y página de contacto

Elaborado por: Bryan Amaguaña

- A. Formulario datos de la tarjeta,** despliega un formulario con los datos de la tarjeta registrado actualmente.
- B. Formulario de ingreso de datos,** despliega un formulario para ingresar el código de la nueva tarjeta para validar datos, desplegar el tipo de tarjeta y valor de la nueva tarjeta.
- C. Botón de confirmación,** Permite confirmar el cambio de tarjeta.
- D. Formulario de información,** Despliega los datos de contacto de la empresa para informar cualquier inconveniente con la aplicación o con el sistema de cobros

Dispositivo Lector de tarjetas NFC

El dispositivo lector de tarjetas NFC será incorporado en la unidad de transporte urbano siendo instalado a una fuente de alimentación incorporada en el mismo bus desde la batería.

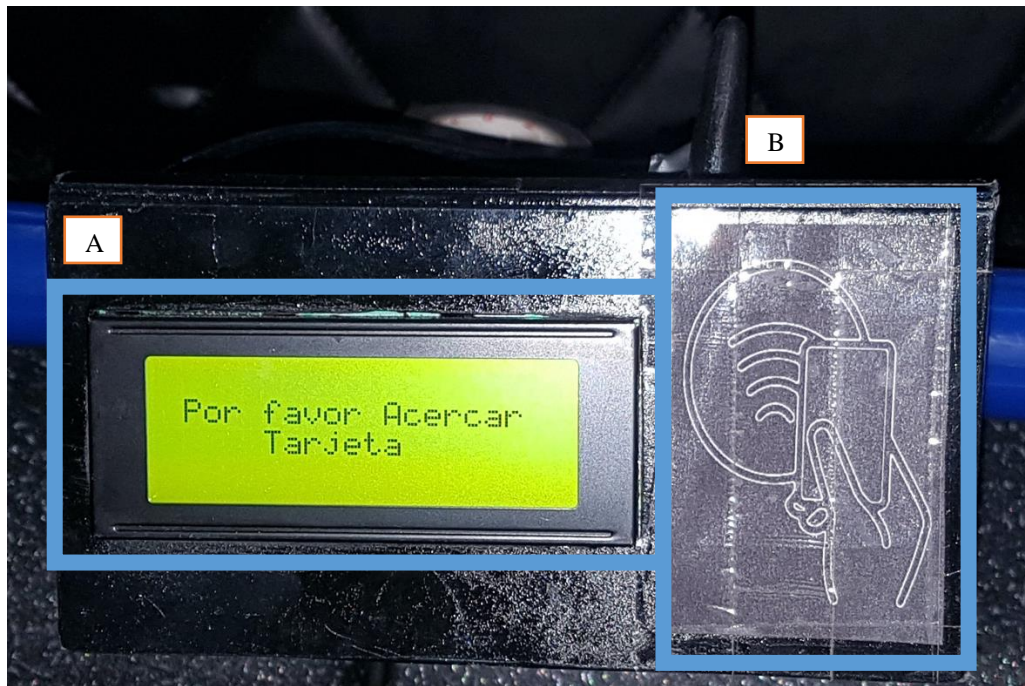


Figura 148: Dispositivo lector de tarjetas NFC

Elaborado por: Bryan Amaguaña

- A. Pantalla de despliegue de mensaje,** despliega un mensaje para que el usuario interactúe con el dispositivo, cuando se acerca una tarjeta desplegará un mensaje si el pago se realizó correctamente o si la tarjeta no posee saldo o fue bloqueada.
- B. Espacio disponible para acercar la tarjeta,** Espacio disponible para que el señor pasajero acerque la tarjeta para realizar el pago del pasaje.

Dispositivo lector y escritor de tarjetas NFC

El dispositivo lector y escritor de tarjetas NFC será entregado a una persona a cargo de realizar las recargas de las tarjetas de los pasajeros. El dispositivo cuenta con 2 opciones una que es para realizar una recarga a una tarjeta registrada en el sistema y otras opciones que es para registra una tarjeta en el sistema

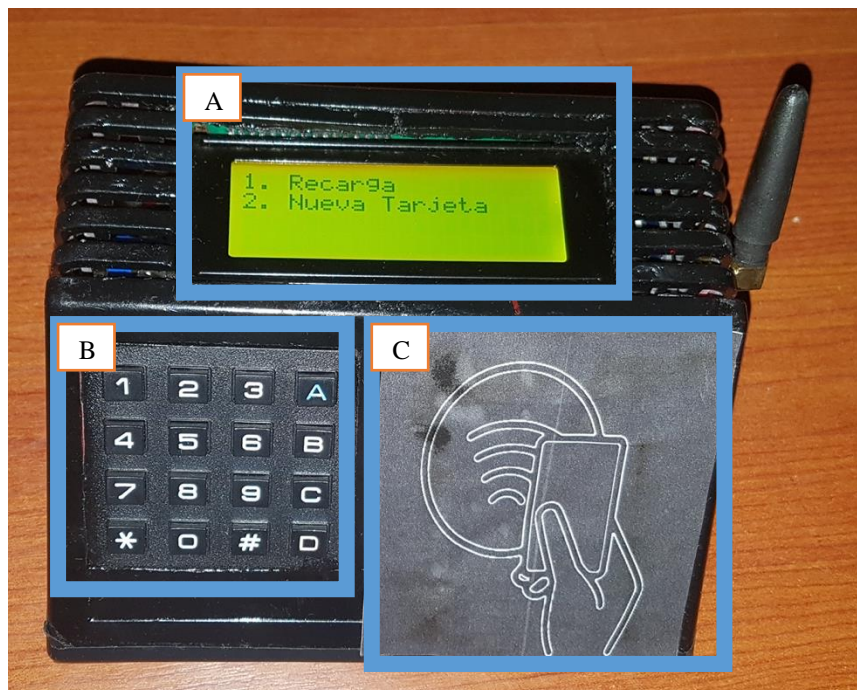


Figura 149: Dispositivo lector y escritor de tarjetas NFC

Elaborado por: Bryan Amaguaña

- A. Pantalla de despliegue de mensaje**, despliega un mensaje para que el usuario interactúe con el dispositivo, mostrando opciones para realizar una acción.
- B. Teclado mecánico**, Para realizar una acción con el dispositivo es necesario seleccionar una opción que se despliega en la pantalla.
- C. Espacio disponible para acercar la tarjeta NFC**

- **Realizar una recarga**

Al Seleccionar la opción 1 del dispositivo desplegara un mensaje para ingresar el valor de la tarjeta una vez ya ingresado se acerca la tarjeta y el dispositivo mostrara un mensaje de confirmación



Figura 152: Registro de una nueva tarjeta

Elaborado por: Bryan Amaguaña

En el teclado mecánico se observa letras que ayudan en la interacción del dispositivo.

- **A**, confirmar un proceso y continua con la recarga de la tarjeta.
- **B**, Borra el valor de la recarga en caso de error
- **C**, no cumple ninguna funcionalidad.
- **D**, Cancela la recarga de la tarjeta

- **Registro de una nueva tarjeta**

Al Seleccionar la opción 2 del dispositivo desplegara el siguiente menú de opciones donde seleccionaremos el tipo de tarjeta para el registro de en el sistema, mostrando un mensaje que informara cuando acercar la tarjeta para registrar y notificar por pantalla si la tarjeta fue registrada con éxito.



Figura 151: Registro de una nueva tarjeta

Elaborado por: Bryan Amaguaña

En el teclado mecánico se observa letras que ayudan en la interacción del dispositivo.

- **A**, confirmar un proceso y continua con el registro de la tarjeta.
- **B**, para la creación de tarjeta no cumple ninguna funcionalidad.
- **C**, no cumple ninguna funcionalidad.
- **D**, Cancela una creación de tarjetas.

- **Mensaje de interacción de la tarjeta**

Cuando uno de los 2 procesos mencionados se cumple correctamente el dispositivo despliega un mensaje de interacción con la tarjeta donde el usuario debe acercar la tarjeta para finalizar con el proceso.



Figura 152: Mensaje de interacción con la tarjeta

Elaborado por: Bryan Amaguaña