

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN PROTOTIPO DE SISTEMA DISTRIBUIDO PARA LA GESTIÓN DE INFORMACIÓN DE UNA EMPRESA DE TRANSPORTE

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

PATRICIO XAVIER GARCIA CORONEL

DIRECTOR: DANNY SANTIAGO GUAMÁN LOACHAMÍN. PhD.

Quito, 2022

AVAL

Certifico que el presente trabajo fue desarrollado por Patricio Xavier García Coronel, bajo mi supervisión.

ING. DANNY GUAMAN
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

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

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Patricio Xavier Garcia Coronel

DEDICATORIA

Dedico el presente proyecto:

A mi madre, por ser las personas que me dio la vida, y quien ha permanecido junto a mí.
Sin ella nada de esto sería posible.

A mis familiares, por su apoyo incondicional.

A mis amigos.

AGRADECIMIENTO

A mi familia, por todo el amor y confianza que me brindan siempre, dándome fuerza en cada paso que doy en mi vida.

A mis amigos de toda la vida, por cada momento y experiencia compartida, y que aunque ya no los vea mucho siempre estarán en mi corazón.

A mis poliamigos, conocerlos hizo que cada decisión tomada en mi vida valga la pena. En especial a ti Erika, aprovecha todo ese potencial que tienes y nunca cambies, eres una excelente persona.

A mis maestros: Quienes me exigían lo mejor de mí y me demostraron el por qué sentirse orgulloso de llamarse politécnico.

Un agradecimiento especial a Danny Guamán por su paciencia y tiempo invertido en este proyecto.

ÍNDICE DE CONTENIDO

AVAL.....	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA.....	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS.....	VIII
ÍNDICE DE CÓDIGOS	IX
RESUMEN.....	X
ABSTRACT.....	XI
1 INTRODUCCIÓN.....	12
1.1 OBJETIVO GENERAL	12
1.2 OBJETIVOS ESPECÍFICOS.....	12
1.3 ALCANCE.....	13
1.4 MARCO TEÓRICO.....	15
1.4.1 SISTEMA DISTRIBUIDO [4], [5]	15
1.4.1.1 SISTEMAS CLIENTE-SERVIDOR	15
1.4.2 APLICACIONES WEB.....	17
1.4.3 SERVICIOS WEB.....	17
1.4.4 APLICACIONES MÓVILES	17
1.4.5 TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PRESENTE PROYECTO.....	18
1.4.5.1 MICROSOFT SQL SERVER [10].....	18
1.4.5.2 WINDOWS COMMUNICATION FOUNDATION (WCF) [8], [11]	18
1.4.5.3 ASP .NET [3].....	18
1.4.5.4 XAMARIN.FORMS [2]	19
1.4.5.5 MICROSOFT AZURE.....	19
1.4.5.6 METODOLOGÍA KANBAN [13].....	20
2 METODOLOGÍA	22
2.1 ANÁLISIS DE REQUISITOS.....	23
2.1.1 RESULTADOS DE ENCUESTAS DE REQUERIMIENTOS FUNCIONALES	24
2.1.2 RESULTADOS DE ENCUESTA DE REQUERIMIENTOS NO FUNCIONALES	25
2.1.3 HISTORIAS DE USUARIO	26
2.1.4 DIAGRAMA DE CASOS DE USOS.....	27
2.1.5 CASOS DE USO	28
2.2 DISEÑO.....	30
2.2.1 ARQUITECTURA DEL PROTOTIPO	30
2.2.2 CAPA DE DATOS.....	31
2.2.3 CAPA DE LÓGICA DE NEGOCIO	35
2.2.4 CAPA DE PRESENTACIÓN.....	44
2.3 IMPLEMENTACIÓN.....	49
2.3.1 CAPA DE DATOS.....	52
2.3.2 CAPA LOGICA DE NEGOCIO	54
2.3.3 CAPA DE PRESENTACIÓN.....	56
2.3.3.1 APLICACIONES MÓVILES.....	56

2.3.3.2	APLICACIÓN WEB	59
3	PRUEBAS Y RESULTADOS DEL PROTOTIPO	62
3.1	PRUEBAS UNITARIAS	62
3.2	PRUEBAS DE INTEGRACIÓN.....	63
3.3	PRUEBAS DE ACEPTACIÓN	63
3.3.1	REQUISITOS FUNCIONALES	64
3.3.2	REQUISITOS NO FUNCIONALES.....	64
3.4	VERIFICACIÓN Y VALIDACIÓN	64
4	CONCLUSIONES Y RECOMENDACIONES	69
4.1	CONCLUSIONES.....	69
4.2	RECOMENDACIONES	70
5	REFERENCIAS BIBLIOGRÁFICAS.....	71
6	ANEXOS	73

ÍNDICE DE FIGURAS

FIGURA 1.1 PRESENTACIÓN DISTRIBUIDA [5], [6].	16
FIGURA 1.2 PRESENTACIÓN REMOTA [6].	16
FIGURA 1.3 BASE DE DATOS DISTRIBUIDAS [5], [6].	16
FIGURA 1.4 CLIENTE SERVIDOR A TRES NIVELES [5], [6].	17
FIGURA 2.1 RESPUESTA PREGUNTA 6 CONDUCTOR.	24
FIGURA 2.2 DIAGRAMA DE CASO DE USO GENERAL.	28
FIGURA 2.3 ARQUITECTURA DEL SISTEMA DISTRIBUIDO.	31
FIGURA 2.4 MODELO ENTIDAD-RELACIÓN.	32
FIGURA 2.6 DIAGRAMA DE CLASES UML.	35
FIGURA 2.8 SOLICITUD DE RECORRIDO.	42
FIGURA 2.9 RECEPCIÓN DE RECORRIDO.	43
FIGURA 2.10 DIAGRAMA DE ESTADOS.	44
FIGURA 2.12 SCKETCH DE LA INTERFAZ PRINCIPAL DE UN CLIENTE.	45
FIGURA 2.13 SCKETCH DE LA INTERFAZ PRINCIPAL DE UN CONDUCTOR.	46
FIGURA 2.14 DIAGRAMA DE DESPLIEGUE DEL PROTOTIPO.	50
FIGURA 2.15 COMUNICACIÓN DE COMPONENTES DEL PROTOTIPO.	51
FIGURA 2.16 ESTRUCTURA DE LA SOLUCIÓN EN VISUAL STUDIO.	52
FIGURA 2.17 ESTRUCTURA DEL PROYECTO BASEDEDATOS.	53
FIGURA 2.19 ESTRUCTURA DEL PROYECTO WCFSERVICEAPP.	55
FIGURA 2.20 ESTRUCTURA DEL PROYECTO APPTESISTESTCLIENT.	56
FIGURA 2.21 ESTRUCTURA DEL PROYECTO ASPAPP.	60
FIGURA 2.22 PÁGINAS DE ASPAPP.	60
FIGURA 3.1 SERVICIO EN LA NUBE.	62
FIGURA 3.2 PRUEBA UNITARIA EXITOSA.	62

ÍNDICE DE TABLAS

TABLA 1.1 EJEMPLO TABLERO KANBAN.....	20
TABLA 2.1 ACTIVIDADES TABLERO KANBAN.....	22
TABLA 2.2 PREGUNTAS DE ENCUESTA A CONDUCTORES.....	23
TABLA 2.2 NECESIDADES FUNCIONALES.....	25
TABLA 2.3 NECESIDADES NO FUNCIONALES.....	26
TABLA 2.4 FORMATO HISTORIA DE USUARIO.....	26
TABLA 2.5 HISTORIA DE USUARIO AUTENTICACIÓN APLICACIÓN CLIENTE.....	27
TABLA 2.6 FORMATO DE CASO DE USO.....	28
TABLA 2.7 CASO DE USO SOLICITAR RECORRIDO.....	29
TABLA 2.8 MÉTODOS DEL SERVICIO WEB WCF.....	37
TABLA 2.9 MÉTODOS DESENCADENADOS EN CADA INTERFAZ DE USUARIO.....	47
TABLA 2.10 RESUMEN DE HERRAMIENTAS.....	49
TABLA 3.1 RESULTADOS DE ENCUESTAS DE SATISFACCIÓN DE REQUISITOS FUNCIONALES.....	64
TABLA 3.2 CRITERIOS DE ACEPTACIÓN DE HISTORIA DE USUARIO.....	65
TABLA 3.3 RESULTADOS DE ENCUESTAS DE SATISFACCIÓN DE REQUISITOS NO FUNCIONALES.....	68

ÍNDICE DE CÓDIGOS

CÓDIGO. 0.10 CONEXIÓN A BASE DE DATOS.	53
CÓDIGO. 0.11 CLASE ADMIN BASE DE DATOS.	54
CÓDIGO. 0.12 MÉTODO LISTARCLIENTES BASE DE DATOS..	54
CÓDIGO. 0.13 INTERFAZ DEL SERVICIO.	55
CÓDIGO. 0.14 IMPLEMENTACIÓN DE LA INTERFAZ DEL SERVICIO.....	55
CÓDIGO. 0.15 CONFIGURACIÓN DEL SERVICIO.	56
CÓDIGO. 0.16 ASIGNAR DATOS A INTERFAZ GRÁFICA.	57
CÓDIGO. 0.17 VINCULACIÓN DE DATOS A INTERFAZ GRÁFICA.	57
CÓDIGO. 0.18 SOLICITAR RECORRIDO.	58
CÓDIGO. 0.19 RECEPCIÓN DE SOLICITUDES DE RECORRIDO.....	58
CÓDIGO. 0.20 ESCUCHA DE CAMBIO DE ESTADOS APLICACIÓN CLIENTE.....	59
CÓDIGO. 0.21 GRIDVIEW EN APLICACIÓN ADMINISTRADOR.	61
CÓDIGO. 0.22 MÉTODO CLIC BOTÓN EDITAR USUARIO ADMINISTRADOR.	61
CÓDIGO. 0.23 PRUEBA UNITARIA DE AGREGAR CLIENTE EXITOSA.	62
CÓDIGO. 0.24 PRUEBA DE INTEGRACIÓN DE AGREGAR CLIENTE EXITOSA.	63

RESUMEN

El propósito de este proyecto de titulación es la elaboración de un prototipo de sistema distribuido para proveer y gestionar la información de una compañía de transporte. El sistema distribuido consta de cinco elementos de software y posee una arquitectura de tres capas: capa de datos, capa de lógica de negocio y capa de presentación. Así, la primera capa consta de una base de datos relacional desplegada en SQL Server. La capa de lógica de negocio posee un servicio web WCF (*Windows Communication Foundation*), que publica los métodos de la lógica de negocio para que sean consumidos por la capa de presentación. Esta última capa engloba 3 aplicaciones: dos aplicaciones móviles y una aplicación web. Las aplicaciones móviles han sido desarrolladas con el IDE Visual Studio y el *framework* de *Xamarin.Forms* para aplicaciones nativas de *Android*. Una aplicación móvil está destinada al cliente, quien puede realizar solicitudes de un recorrido. La segunda aplicación móvil está destinada al conductor, quien puede aceptar o no un recorrido. La tercera es una aplicación web destinada al administrador, quien podrá ver el registro de las solicitudes de recorrido realizadas por los clientes, además de administrar la información de los clientes, conductores y vehículos registrados.

En el primer capítulo se hace un breve resumen de las tecnologías y herramientas aplicadas para el desarrollo del sistema distribuido. El segundo capítulo describe el desarrollo completo del sistema, dividido en cuatro fases: análisis de requisitos, diseño, implementación y pruebas. La fase de análisis de requisitos comprende las entrevistas a usuarios para la toma de requerimientos del sistema, así como la creación de historias de usuario. Las historias de usuario se usaron como base para identificar los casos de uso del sistema. En la fase de diseño, se presenta la arquitectura del prototipo, el modelo de datos, el modelamiento de los procesos o funciones principales y el diseño de las interfaces para cada aplicación. La fase de implementación agrupa la configuración del equipo para poder desarrollar todo el prototipo, la codificación de las distintas capas del sistema, y además se describe el proceso de publicación de las aplicaciones. La fase de pruebas, que se presenta en el capítulo 3, resume los resultados de las pruebas realizadas. Se realizaron pruebas unitarias, de integración y aceptación para comprobar que cumple con los requerimientos especificados. Finalmente, el capítulo 4 muestran las conclusiones del trabajo y algunas recomendaciones para aquellos interesados en proyectos afines.

PALABRAS CLAVE: Aplicación web, servicio web, transporte, aplicación móvil, sistema distribuido.

ABSTRACT

The purpose of this project is elaborating a prototype of a distributed system, which provides and manages the information of a transport company. The distributed system consists of five software elements organized by following a three-tier architecture, data, business logic, and presentation. The data layer consists of a relational database deployed in a SQL Server. The business logic layer has a WCF (Windows Communication Foundation) web service that publishes the business logic methods to be consumed by the final presentation layer. The latter layer encompasses three applications: two mobile apps and one web application. The mobile applications have been developed in the Visual Studio IDE and Xamarin.Forms framework for Android native applications. The former mobile application targets a client, who will be able to request a vehicle. The second mobile application targets a driver, who may or may not accept a vehicle request. The third is a web application, which targets an administrator, who will be able to manage the information of the clients, drivers and vehicles registered.

The first chapter presents a brief summary of the technologies and tools used for developing the distributed system. The second chapter describes the overall development of the system, broken down into four phases: requirements analysis, design, implementation, and testing. The requirements analysis phase includes interviews with users to gather system requirements, as well as the creation of user stories. The user stories were used as a basis to identify the use cases of the system. In the design phase, the architecture of the prototype, the data model, the modeling of the main processes and the design of the interfaces for each application are presented. The implementation phase encompasses the environment settings to develop the entire prototype, the coding of the system, and also describes the publication process of the applications. The testing phase, presented in Chapter 3, summarizes the results of the tests performed. Unit, integration, and acceptance tests were carried out to verify that the prototype meets the specified requirements. Finally, chapter 4 shows the conclusions and some recommendations for those interested in related projects.

KEYWORDS: Web application, web service, transportation, mobile application, distributed system.

1 INTRODUCCIÓN

Actualmente, debido a la constante evolución de las tecnologías de la información, la mayoría de las empresas han optado por migrar de procesos manuales de gestión de la información a procesos automatizados. Esto, con el objetivo de brindar mayor competitividad en sus operaciones e impulsarse a crecer en conjunto con las nuevas tecnologías. A pesar de ello, aún existen compañías que no han logrado modernizar sus servicios. Este es el caso de la compañía de transporte *Inercita S.A.*, la cual actualmente no cuenta con un sistema informático que permita a su personal registrar a sus clientes, realizar peticiones de vehículos de transporte, llevar un historial de los viajes realizados por los vehículos y mantener la información de sus conductores de forma digital. Todas estas actividades se registran a mano en hojas de papel y son almacenadas en un archivador por el dueño de la compañía. Además, los registros de viajes realizados son escritos a mano y no tiene un formato concreto, lo que puede originar errores en el control de la información[1].

Por otro lado, los clientes realizan peticiones de vehículos de transporte vía llamadas telefónicas hacia el dueño y administrador, esto resulta ineficiente puesto que registrar manualmente datos como la ubicación de inicio y de fin del viaje a realizar, datos del cliente, y posteriormente contactar a un conductor que lleve a cabo dicho viaje puede tomar un tiempo considerable. Esto limita la cantidad de solicitudes de viaje que pueden ser atendidas. En este contexto, en el presente proyecto se propone desarrollar un prototipo de sistema distribuido que a futuro permita a la compañía de transporte *Inercita S.A.* gestionar de manera organizada y automática la información correspondiente a cada cliente y conductor, así como realizar una solicitud de viaje en tiempo real registrando la ubicación de partida y de destino.

1.1 OBJETIVO GENERAL

Desarrollar un prototipo de un sistema distribuido para proporcionar servicios de transporte y administrar su información.

1.2 OBJETIVOS ESPECÍFICOS

- Analizar los fundamentos teóricos de las herramientas necesarias para el desarrollo de este trabajo de titulación.
- Realizar un análisis de requerimientos del sistema distribuido.
- Diseñar la arquitectura y componentes del sistema distribuido que satisfagan los requerimientos previamente definidos.

- Implementar el sistema previamente diseñado.
- Validar el correcto funcionamiento de las aplicaciones respecto a los requisitos especificados.

1.3 ALCANCE

En este trabajo se plantea el desarrollo de un prototipo de un sistema distribuido que permita gestionar la información de la compañía de transporte *Inercita S.A.* El prototipo cuenta con 3 aplicaciones. La primera una aplicación nativa Android, desarrollado con *Xamarin*, que permite a los clientes realizar solicitudes de vehículos de transporte, la segunda una aplicación para que los conductores puedan atender las solicitudes de los clientes, y la tercera una aplicación web, desarrollada con *ASP .NET (Active Server Pages)*, para la administración de la información de los viajes, clientes y conductores [2], [3].

Cada aplicación tendrá un rol y diferentes funcionalidades como se describe a continuación:

Aplicación de cliente

El cliente fundamentalmente se registrará y realizará solicitudes de viaje. Específicamente, los módulos que integran la aplicación cliente son:

- Módulo de registro: Permite registrarse a los clientes que deseen usar el sistema. Para ello, un usuario ingresa cierta información de su perfil, además proporciona una contraseña para su posterior inicio de sesión.
- Módulo de autenticación: Permite a los clientes ingresar a la aplicación mediante su correo y contraseña.
- Módulo de viajes: Permite a los clientes realizar una solicitud de viaje. También permite cancelar solicitudes de viaje.
- Módulo de administración de viajes: Permite a los clientes ver los viajes realizados.
- Módulo de administración de perfil de cliente: Permite a un cliente editar sus datos de perfil.
- Módulo de reinicio de contraseña: La contraseña puede ser reestablecida por el usuario en caso de haberla olvidado, esto se realiza mediante el correo electrónico ingresado.

Aplicación de conductor

El conductor podrá responder a solicitudes de viajes de clientes. Los módulos que integrarán la aplicación del conductor son:

- Módulo de registro: Permite a los conductores que desean usar el sistema registrarse, se debe ingresar su correo y proporcionar una contraseña para su posterior inicio de sesión. Serán validadas por un administrador existente.
- Módulo de autenticación: Permite a los conductores ingresar a la aplicación mediante su correo y contraseña para su posterior uso.
- Módulo de viajes: Permite a los conductores realizar comenzar a escuchar solicitudes de viajes. Permite a los conductores atender a las solicitudes de viaje realizadas por los clientes.
- Módulo de administración de viajes: Permite ver a los conductores los viajes realizados.
- Módulo de administración de usuarios: Permite a los conductores modificar su información y actualizar la información de sus vehículos.
- Módulo de reinicio de contraseña: La contraseña podrá ser reestablecida por el usuario en caso de haberla olvidado, esto se hará mediante el correo electrónico ingresado.

Aplicación de administrador:

El rol administrador tendrá permitido realizar las operaciones *CRUD (Create, Read, Update and Delete)* de los datos del sistema. Los módulos que integrarán la aplicación de administrador son:

- Módulo de registro: Permite a los administradores que deseen usar el sistema registrarse, debe ingresar su correo y proporcionar una contraseña para su posterior inicio de sesión.
- Módulo de autenticación: Permite a los administradores ingresar a la aplicación mediante su correo y contraseña para su posterior uso de la aplicación.
- Módulo de viajes: Permite a los administradores ver todas las solicitudes de viaje realizadas.

- Módulo de administración de viajes: Permite a los administradores ver los viajes realizados.
- Módulo de administración de usuarios: Permite el administrador realizar el CRUD de los clientes y conductores.
- Módulo de inicio de sesión: Como vista de inicio las aplicaciones tienen un inicio de sesión para permitir el ingreso al sistema. El usuario inicia sesión con las credenciales registradas nombre de usuario y contraseña.

1.4 MARCO TEÓRICO

1.4.1 SISTEMA DISTRIBUIDO [4], [5]

Un sistema distribuido puede entenderse como un conjunto de computadores independientes que ejecutan una única tarea de forma transparente al usuario. Este sistema se basa en repartir la información a través de Internet mediante un software, previamente descargado por el usuario, a diferentes ordenadores, que van resolviendo los cálculos o realizando ciertas operaciones. Una vez obtenido el resultado, se lo envía a un servidor de almacenamiento de información.

1.4.1.1 SISTEMAS CLIENTE-SERVIDOR

Los sistemas cliente-servidor se ajustan a un patrón de arquitectura, constituido fundamentalmente de dos componentes de software. El cliente es un componente de software que solicita un determinado servicio, apuntando a una dirección y un puerto previamente definido. Esta dirección y puerto le pertenecen al servidor, el segundo componente de software que responde las peticiones del cliente. El servidor responde a un cliente mediante: (i) la ejecución de un determinado programa o (ii) el acceso a un determinado banco de información.

1.4.1.1.1 CLASIFICACIÓN DE LOS SISTEMAS CLIENTE-SERVIDOR

- **Presentación distribuida.** Toda la aplicación se encuentra del lado del servidor. La información es procesada del lado del servidor, El terminal de usuario solo muestra la información.



Figura 1.1 Presentación distribuida [5], [6].

- **Presentación Remota.** La lógica de la aplicación y la base de datos se encuentran en el servidor. En el cliente se encuentra una interfaz gráfica de usuario que fundamentalmente se asegura que el formato de los datos a enviar al servidor sea correcto.



Figura 1.2 Presentación Remota [6].

- **Base de Datos Distribuidas.** Toda la aplicación se encuentra del lado del usuario. Es decir, toda la información se procesa en el equipo terminal y almacena en un equipo diferente.

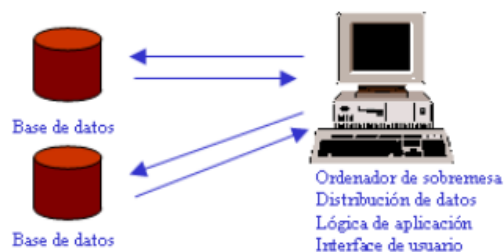


Figura 1.3 base de Datos Distribuidas [5], [6].

- **Cliente servidor a tres niveles.** El usuario utiliza una interfaz para conectarse a un servidor que procesa la información y luego la guarda en una base de datos.



Figura 1.4 cliente servidor a tres niveles [5], [6].

1.4.2 APLICACIONES WEB

A diferencia de las aplicaciones de escritorio¹, las aplicaciones web son diseñadas para ejecutar su lógica en un servidor web, al cual se accede a través de un navegador. Estas aplicaciones son diseñadas usando *HTML (Hypertext Transfer Protocol)*, *CSS (Cascading Style Sheets)* y *JavaScript*, los mismo que son interpretados por el cliente web para mostrar el contenido del lado del usuario. Las peticiones enviadas desde un cliente son recibidas a través de un servidor web como Apache o Nginx, entre otros. Para la comunicación entre un cliente y un servidor web se utiliza comúnmente el protocolo *HTTP/HTTPS (HyperText Transfer Protocol Secure)*. [7]

1.4.3 SERVICIOS WEB

Un servicio web puede entenderse como un conjunto de funciones (“servicios”) diseñadas para consumirse desde otras aplicaciones. Estos programas se comunican utilizando formatos de notación abierta como XML (Extensible Markup Language) o *JSON (JavaScript Object Notation)*. [8]

1.4.4 APLICACIONES MÓVILES

Son aplicaciones desarrolladas para ejecutarse exclusivamente en un dispositivo móvil. Al encontrarse optimizadas para trabajar con un tipo de específico de dispositivo permiten un mayor control de los recursos de hardware [9]. Por lo general estas aplicaciones son distribuidas por plataformas gestionadas por los propietarios del sistema operativo.

¹ Las aplicaciones de escritorio son el producto de software que se instala y usa en un dispositivo.

1.4.5 TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PRESENTE PROYECTO

1.4.5.1 MICROSOFT SQL SERVER [10]

Microsoft SQL Server es un software propietario de gestión de bases de datos relacionales que soporta una amplia variedad de aplicaciones de procesamiento de transacciones. Un base de datos en SQL Server está conformado de tablas relacionadas, evitando así la necesidad de almacenar datos de forma redundante.

El componente principal de Microsoft SQL Server es el motor de base de datos de SQL Server, que controla el almacenamiento, el procesamiento y la seguridad de los datos. Debajo del motor de base de datos se encuentra el sistema operativo de SQL Server, o SQLOS. SQLOS se encarga de las funciones de bajo nivel, como la gestión de la memoria y la E/S, la programación de trabajos y el bloqueo de datos para evitar conflictos de ejecución. Por encima del motor de base de datos se encuentra una capa de interfaz de red que utiliza el protocolo Tabular Data Stream de Microsoft para facilitar las interacciones de solicitud y respuesta con los servidores de base de datos. Y a nivel de usuario se escriben sentencias T-SQL para construir y modificar estructuras de bases de datos.

1.4.5.2 WINDOWS COMMUNICATION FOUNDATION (WCF) [8], [11]

Windows Communication Foundation (WCF) es un framework que permite crear aplicaciones orientadas a servicios. Este framework permite enviar datos como mensajes asíncronos o síncronos desde un punto de servicio a un cliente. Un punto de servicio puede ser parte de un servicio disponible de forma continua y hospedado por los IIS (Internet Information Services) o puede ser un servicio hospedado en una aplicación. Por otro lado, el cliente es un punto de red que solicita datos de un punto de servicio. Los mensajes pueden ser tan simples como un solo carácter o palabra enviada como XML, o tan complejos como un flujo de datos binarios que represente un objeto de una clase determinada.

1.4.5.3 ASP .NET [3]

ASP.NET es un entorno para el desarrollo de aplicaciones web, cuyo propietario es Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web basado en XML. ASP.NET está construido sobre el *Common Language Runtime*, permitiendo a los programadores escribir código ASP.NET con cualquier lenguaje admitido por el *framework* .NET. Además, ofrece tres *frameworks* para crear aplicaciones web: formularios web, ASP.NET MVC y páginas web ASP.NET.

1.4.5.3.1 MODELOS DE PROGRAMACIÓN EN ASP.NET

- **ASP.NET Web Forms** proporciona un alto nivel de abstracción con un modelo de programación basado en eventos y controles. Este favorece la productividad mediante la programación declarativa, reduciendo la cantidad de código necesaria para implementar una determinada funcionalidad.
- **ASP.NET MVC** se concibió como alternativa a Web Forms y proporciona un modelo de programación basado en el patrón de arquitectura MVC. Entre sus principales características destacan su completa integración con pruebas unitarias y la separación entre la lógica de presentación, la lógica de negocio y la lógica de acceso a datos.
- **ASP.NET WEB PAGES** es el más reciente de los tres modelos de programación, y fue creado como respuesta a una creciente demanda de desarrolladores web sin experiencia previa con ASP.NET. Dado que ASP.NET Web Forms o MVC les suponía una inversión inicial de tiempo demasiado grande, Web Pages proporciona un modelo de programación más simple y rápido de aprender, sin renunciar a toda la funcionalidad y flexibilidad de ASP.NET.

1.4.5.4 XAMARIN.FORMS [2]

Xamarin.Forms es un framework de código abierto que permite el desarrollo de aplicaciones para Android, iOS y Windows, a partir de una única base de código compartida. El framework permite a los desarrolladores crear interfaces de usuario en XAML con código subyacente en C#. Estas interfaces se representan como controles nativos de alto rendimiento en cada plataforma.

1.4.5.5 MICROSOFT AZURE

Azure es un conjunto de herramientas en la nube que permite compilar, implementar y administrar aplicaciones en una red global de recursos de Microsoft Azure. Azure es capaz de proporcionar software como servicio (SaaS), plataforma como servicio (PaaS) e infraestructura como servicio (IaaS), y es compatible con muchos lenguajes, herramientas y marcos de programación diferentes, incluidos software y sistemas específicos de Microsoft y de terceros[12].

1.4.5.6 METODOLOGÍA KANBAN [13]

Kanban es una metodología de gestión de proyectos que se basa en una filosofía de mejora continua, donde las tareas se asignan de una lista de pendientes, en un flujo de trabajo constante. Esta metodología se implementa mediante tableros Kanban, proveyendo una alternativa visual para la gestión de proyectos. Esto facilita a los equipos visualizar sus flujos y cargas de trabajo. La Tabla 1.1 es un ejemplo de un tablero Kanban donde se muestran 3 tareas que se realizarán al escribir este documento y tres columnas que indican el estado de las tareas, las tareas avanzan de izquierda a derecha conforme se realizan.

Tabla 1.1 Ejemplo tablero Kanban.

To do	Doing	Done
		Consultar acerca de Kanban.
	Escribir un resumen.	
Aplicar la metodología.		

Kanban provee las siguientes ventajas:

- Permite visualizar el flujo de todo el trabajo a realizar: La principal ventaja de Kanban es que puedes visualizar cómo el trabajo “avanza” a través de las etapas. Una tarea en Kanban inicia en el lado izquierdo de tu tablero y, a medida que tu equipo trabaja en ella, recorrerá lentamente las siguientes etapas hasta que aterrice en la columna de tareas realizadas. Esta práctica proporciona una idea general de cómo el trabajo avanza a través de las etapas y permite obtener información en tiempo real y apreciar de un vistazo el estado de los proyectos.
- Permite dividir el trabajo en tareas pequeñas: El trabajo se divide en tareas pequeñas, así se priorizarán y se colocarán en el panel de manera ordenada.
- Limitar el trabajo en curso: Para que Kanban funcione es necesario poner un límite al número de ítems en cada columna, a fin de evitar colapsos y eliminar prontamente los impedimentos para un ritmo de trabajo adecuado y sostenible. El límite también debe ser conocido por todos.

Kanban no proporciona un conjunto completo de reglas y procedimientos para la planificación y ejecución de proyectos, sino que se centra más en la visualización y el control del flujo de

trabajo. Por este motivo no puede ser considerada como una metodología completa sino como un enfoque para la gestión de proyectos y el flujo de trabajo, la cual se puede utilizar como parte de otras metodologías.

2 METODOLOGÍA

En el presente proyecto, se han seguido la metodología de desarrollo de software estándar. Se ha realiza el análisis de requisitos, diseño, implementación y pruebas. Además, se han adoptado las buenas prácticas de Kanban, que nos permite visualizar los flujos y la carga de trabajo del proceso de desarrollo. Mientras el tablero Kanban completo se encuentra en el Anexo A, la Tabla 2.1 muestra, a manera de ejemplo, un extracto del registro de las actividades realizadas durante el desarrollo del proyecto.

Tabla 2.1 Actividades Tablero Kanban.

To do	Doing	Done		
		Estudiar el framework Xamarin	10-Jan-2022	25-Jan-2022
		Estudiar el framework Windows Communication Foundation	10-Feb-2022	24-Feb-2022
		Describir las herramientas Xamarin	25-Feb-2022	10-Mar-2022
		Listar las tareas a realizar (fase visualización kanban)	11-Mar-2022	11-Mar-2022
Entrevistar al administrador y 10 conductores de la compañía			12-Mar-2022	15-Mar-2022

2.1 ANÁLISIS DE REQUISITOS

Para el análisis de requisitos se utilizó entrevistas no estructuradas con los conductores de la empresa Interciti S.A. También se realizó encuestas a 10 clientes, 10 conductores y 2 administradores con el objetivo de analizar las respuestas y legitimar los requisitos de usuario.

Primero, se realizó una entrevista no estructurada con el objetivo de comprender el funcionamiento de la empresa. Luego se procedió a realizar un conjunto de preguntas, en forma de entrevista estructurada, con el objetivo de recopilar información más precisa y organizada. De este modo, se obtuvo una lista de necesidades de los usuarios para, en base a ellos, definir las historias de usuario.

Las preguntas de la encuesta fueron realizadas posterior a la entrevista. La Tabla 2.2 muestra las preguntas realizadas en la encuesta. Por ejemplo, al analizar las respuestas de la pregunta 6 mostradas en la Figura 2.1 indica que la mayoría de los conductores requieren una aplicación móvil Android para la recepción de solicitudes de transporte. Los Anexo B y Anexo C contienen el análisis de las preguntas realizadas en las encuestas clientes, administradores y conductores.

Tabla 2.2 Preguntas de encuesta a conductores.

#	Preguntas	Opciones
1	¿Como recibe usted las solicitudes para realizar el servicio de transporte?	Llamada
		Mensajes de texto
		Otros(Ingresa texto)
2	¿Presenta algún inconveniente al receptor la solicitud de transporte?	Sin inconvenientes
		Llamada muy tardadas
		Direcciones erróneas
		Horarios inapropiados
		Otros(Ingresa texto)
3	¿Como le gustaría receptor las solicitudes de su servicio?	Aplicación móvil
		Aplicación web
		Aplicación PC
		Otros(Ingresa texto)
4	¿Tiene usted un teléfono móvil?	Si

		No
5	¿Tiene su teléfono móvil acceso a internet?	Si, wifi
		Si, Plan celular
		No
6	¿Qué marca es su teléfono móvil?	Xiaomi
		Apple
		Samsung
		Nokia
		Motorola
		Huawei
		POCO
		OPPO
		Otros(Ingresar texto)

6. ¿Qué marca es su teléfono móvil?

[Más detalles](#)

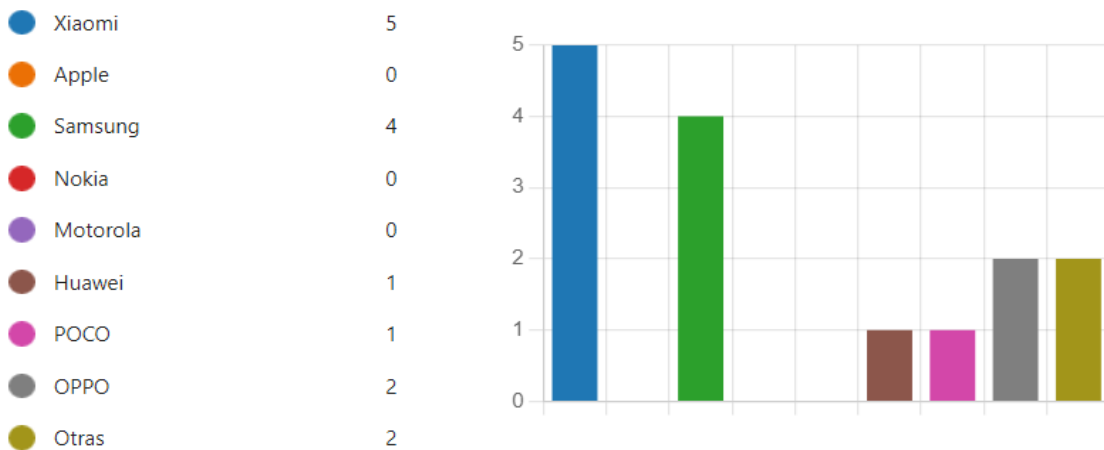


Figura 2.1 Respuesta pregunta 6 Conductor.

2.1.1 RESULTADOS DE ENCUESTAS DE REQUERIMIENTOS FUNCIONALES

Como resultado de la información proporcionada de las entrevistas y encuestas se definió una lista de necesidades mostradas en la Tabla 2.2. El Anexo B muestra los resultados y análisis de las encuestas de requerimientos funcionales.

Tabla 2.2 Necesidades funcionales.

CÓDIGO	TÍTULO	DESCRIPCIÓN
RF001	Registro	El usuario deberá ingresar la información necesaria para su registro como un número de teléfono, correo electrónico, fecha de nacimiento, nombre y apellido.
RF002	Autenticación	Para el ingreso al sistema se utilizará número de cédula y contraseña
RF003	Actualización de datos	El usuario podrá modificar sus datos.
RF004	Solicitar recorrido	El usuario solicitará un vehículo
RF005	Calificar recorrido	Después de solicitar un recorrido y concluirlo se tendrá la opción de calificar el mismo.
RF006	Mis ubicaciones	Se podrá guardar ubicaciones de los clientes.
RF007	Mis recorridos	Muestra la información de los recorridos de un usuario determinado.
RF008	Receptar recorridos	El usuario podrá recibir solicitudes de servicio por un tiempo limitado de tiempo.
RF009	Vehículos	El usuario podrá ver todos los vehículos de la empresa y agregar su propio vehículo.
RF010	Seleccionar vehículo	El usuario podrá cambiar el vehículo seleccionado.
RF011	Gestionar vehículos	El usuario podrá realizar un CRUD de los vehículos de la empresa.
RF012	Gestionar recorridos	El usuario podrá ver la información de los recorridos realizados en la empresa.
RF013	Gestionar clientes	El usuario podrá realizar un CRUD de los clientes de la empresa.
RF014	Gestionar conductores	El usuario podrá realizar un CRUD de los conductores de la empresa.

2.1.2 RESULTADOS DE ENCUESTA DE REQUERIMIENTOS NO FUNCIONALES

Las necesidades no funcionales del sistema distribuido se muestran en la Tabla 2.3. El Anexo C muestra el análisis de la encuesta de requerimientos no funcionales.

Tabla 2.3 Necesidades no funcionales.

Código	Título	Descripción
RNFU001	Autenticación	El sistema debe proveer autenticación de usuarios.
RNFU002	Compatibilidad con la plataforma	El prototipo debe ser ejecutable en un teléfono móvil con sistema operativo Android.
RNFU003	Datos	Se usará una base de datos relacional para la persistencia de datos del sistema.

2.1.3 HISTORIAS DE USUARIO

Una vez definidos las necesidades de los usuarios, estos se tradujeron a historias de usuario que denotan de forma concisa los requisitos y criterios de aceptación que debe cumplir el sistema distribuido. La Tabla 2.4 indica el formato de historias de usuario utilizado.

Tabla 2.4 Formato historia de usuario.

Nombre:	
CÓDIGO:	Asignada:
Prioridad: Alta, media, baja	
Descripción: Como usuario de la aplicación (cliente, administrador, conductor) requiero <Funcionalidad>, de esta manera poder cumplir con <Objetivo>	
Criterios de aceptación: Debe cumplir <condiciones>	

Las historias de usuario contienen:

- Nombre: nombre de la historia de usuario.
- Código: código asignado a la historia de usuario.
- Asignada: persona a quien se le asigna la historia de usuario.
- Prioridad: alta, media y baja.

- Descripción: se divide en:
 - Como usuario de la herramienta: describe el rol del usuario en el sistema distribuido.
 - Requero esta funcionalidad: indica lo que se debe implementar en la herramienta.
 - De esta manera poder cumplir el objetivo: Indica el porqué de la historia de usuario.
- Criterios de aceptación: establece los parámetros para validar si se ha cumplido el requerimiento.

Como un ejemplo del uso de la historia de usuario tenemos la Tabla 2.5 que indica la historia de usuario de autenticación de la aplicación cliente. El Anexo D muestra las historias de usuario de todo el sistema distribuido.

Tabla 2.5 Historia de usuario autenticación aplicación cliente.

Nombre: Autenticación.	
CÓDIGO: HUCL-02	Asignada: Xavier Garcia
Prioridad: Alta	
Descripción: Como usuario de la aplicación cliente requiero que cuente con un mecanismo para el inicio de sesión, de esta manera se evitará accesos no autorizados.	
Criterios de aceptación: Como cliente requiero estar en capacidad de: <ul style="list-style-type: none"> • Iniciar sesión al ingresar mi usuario y contraseña. • Cerrar sesión. 	

2.1.4 DIAGRAMA DE CASOS DE USOS

Con base en las historias de usuario, se ha identificado los casos de uso del prototipo, utilizando para ello diagramas de casos de uso. El diagrama de caso de uso general, que se muestra en la Figura 2.2, indica las interacciones de los diferentes usuarios con los módulos del sistema. El Anexo E muestra los diagramas de casos de uso individuales de cada módulo.

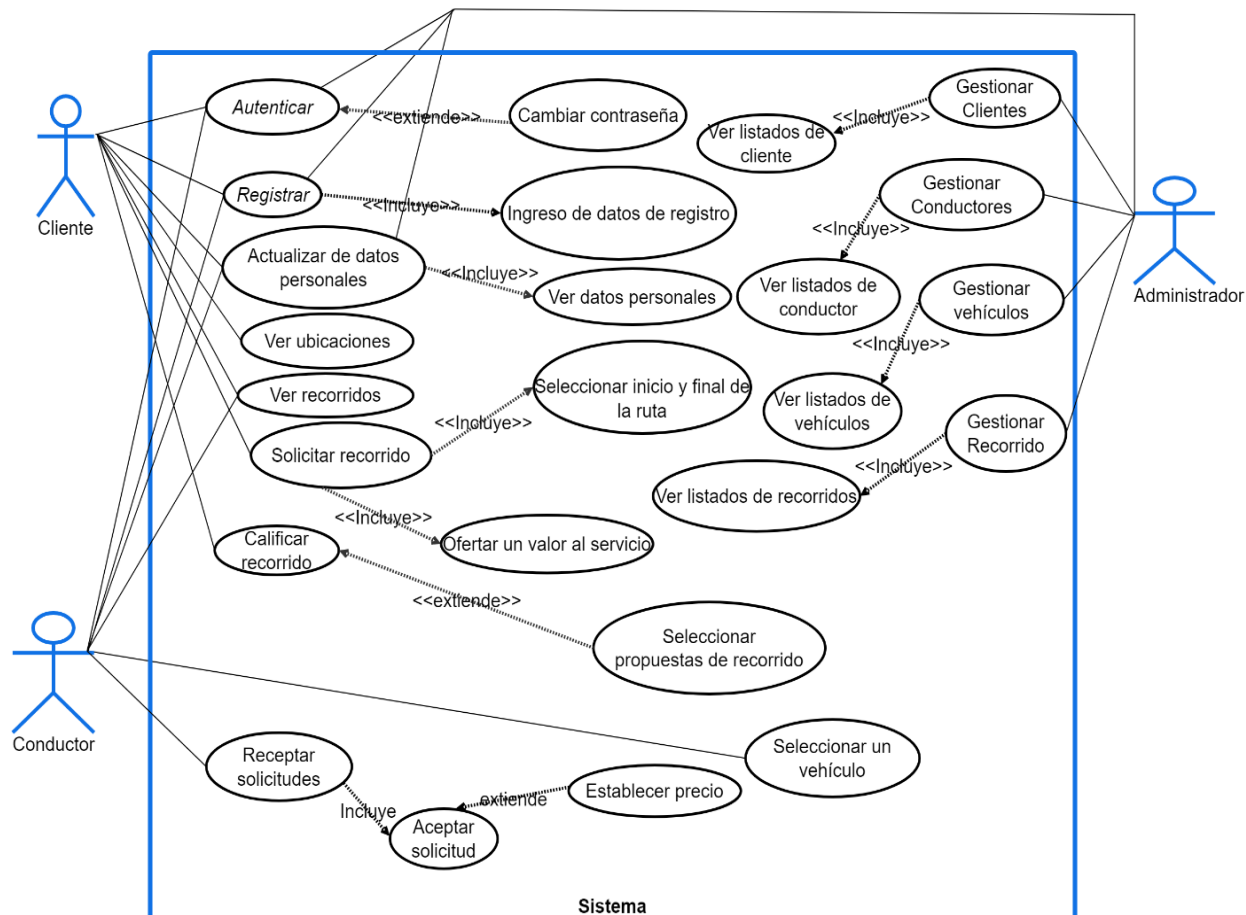


Figura 2.2 Diagrama de caso de uso general.

La Figura 2.2 indica los diferentes casos de uso que requieren los distintos usuarios con el sistema. En la siguiente sección se detalla cada uno de ellos.

2.1.5 CASOS DE USO

Para especificar cada caso de uso se utilizó el formato mostrado en la Tabla 2.6. La Tabla 2.7 muestra el caso de uso CU004, a manera de ejemplo, en el que se detalla el actor del caso de uso (en este caso el usuario Cliente), las precondiciones, y la secuencia de pasos que debe ejecutar el sistema para ser capaces de garantizar este caso de uso. El Anexo F muestra los casos de uso de todos los módulos del sistema.

Tabla 2.6 Formato de caso de uso.

Código CU		NOMBRE	Nombre del caso de uso
-----------	--	--------	------------------------

CU Relacionados	Códigos de Casos de uso relacionados con este.	
Código Requisito	Código de requisitos relacionados con este caso de uso.	
Actor/es Participantes	Entidades que interactúan con el caso de uso.	
Descripción CU	Breve descripción del caso de uso	
Precondiciones	Condiciones relevantes que se deben dar en el sistema antes de ponerse en marcha el caso de uso.	
Secuencia de pasos	PASO	Descripción
	Número de paso	Descripción de las acciones que se llevan a cabo durante el caso de uso.
Postcondiciones	Condiciones que se debe dar en el sistema después de finalizar el caso de uso	

Tabla 2.7 Caso de uso Solicitar Recorrido.

Código CU	CU004	NOMBRE	Solicitar Recorrido
CU Relacionados	CU001		
Código Requisito	RF004		
Actor/es Participantes	Cliente		
Descripción CU	Permite a un cliente realizar el proceso de solicitud e inicio de un recorrido.		
Precondiciones	Sesión iniciada en la aplicación.		
Secuencia de pasos	PASO	Descripción	
	1	El cliente ingresa el punto de partida, destino del recorrido y oferta para un recorrido.	
	2	La aplicación habilita el botón "Solicitar" cuando los datos son completados.	
	3	La aplicación registra el recorrido con estado de "Solicitado".	
	3.1	Si la solicitud es aceptada por un conductor, a aplicación muestra todas las respuestas dadas por los conductores.	
	3.1.1	El cliente selecciona una respuesta a su solicitud.	
	3.1.2	La aplicación inicia el recorrido y cambia el estado del recorrido.	
	3.1.3	La aplicación muestra la ubicación del conductor y notifica cuando llegue al	

		punto de partida.
	3.1.4	La aplicación muestra la ubicación del conductor y notifica al cliente cuando llegue al punto de destino.
	3.1.5	La aplicación muestra al cliente un formulario para calificar la aplicación.
	3.1.5.1	El cliente puede presionar uno de los emojis con caras que indican satisfacción o desaprobación de servicio. Además de enviar un comentario.
	3.1.5.2	El cliente puede omitir el enviar una calificación.
	3.2	Si la solicitud no es aceptada, se muestra una notificación y retorna a la vista principal.
Postcondiciones		La aplicación retorna a la vista principal.

2.2 DISEÑO

2.2.1 ARQUITECTURA DEL PROTOTIPO

La arquitectura del proyecto, que se muestra en la Figura 2.3, sigue el patrón de diseño de tres capas:

- **Capa de datos:** Encargada de almacenar y administrar los datos de los usuarios para el adecuado funcionamiento del sistema completo. Esta capa estará constituida fundamentalmente del servidor de base de datos.
- **Capa de Lógica de Negocio:** Esta capa procesa la información. Aquí se encuentra la funcionalidad principal de todo el sistema, implementada a través de métodos que permiten gestionar los datos. Esta capa estará constituida de un servicio web.
- **Capa de Presentación:** Gestiona las interfaces de usuario; es decir, cómo se comunican los usuarios con el sistema distribuido.

En los siguientes apartados se detalla el diseño de cada una de estas capas.

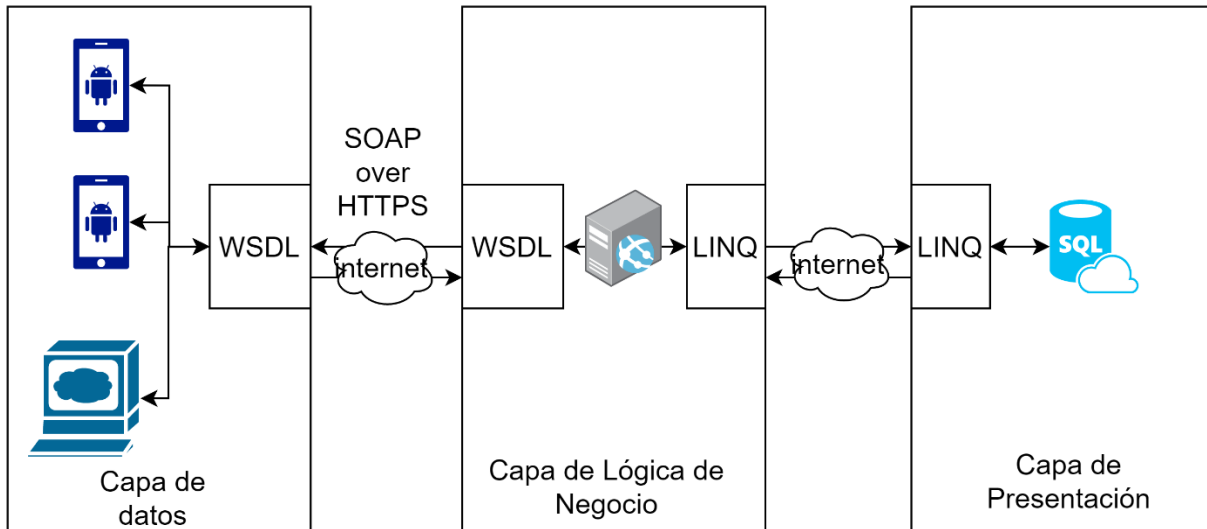


Figura 2.3 Arquitectura del sistema distribuido.

2.2.2 CAPA DE DATOS

En la Figura 2.4 se presenta el modelo entidad-relación de la base de datos que permite identificar las distintas entidades y sus relaciones [15]. A continuación, se detalla cada una de ellas:

- Cliente: Representa a los clientes.
- Conductor: Representa a los conductores.
- Administrador: Representa a los administradores.
- Vehículo: Representa a los vehículos.
- Tipo: Representa a los tipos de vehículos.
- Marca: Representa a la marca de un vehículo.
- Modelo: Representa al modelo de un vehículo.
- Recorrido: Representa a los recorridos.
- Estado: Representa a los estados de un recorrido.

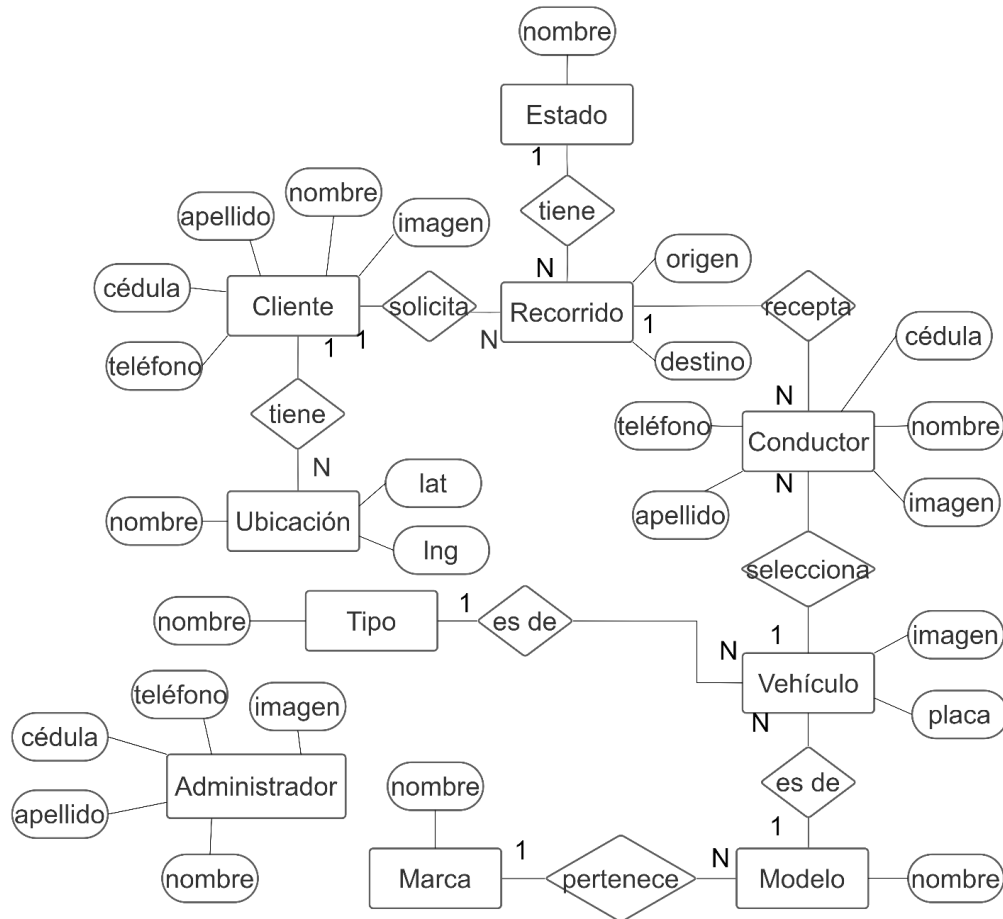


Figura 2.4 Modelo entidad-relación.

Con el objetivo de comprender el modelo entidad-relación, a continuación se describe la entidad *Vehículo*, ya que contiene varias relaciones con otras entidades. Los atributos de la entidad son:

- **idVehiculo**: representa el identificador único de vehículo.
- **Placa**: representa la placa correspondiente a cada Vehículo.
- **Imagen**: representa una fotografía del vehículo.

Las relaciones (caracterizadas por su cardinalidad) que tiene esta entidad son las siguientes:

- Un Vehículo pertenece a un Modelo.
- Un Conductor puede tener solo un Vehículo.
- Un Vehículo puede ser de un solo Tipo.

Del mismo modo, en la Figura 2.5 se muestra el diagrama relacional. El diagrama relacional describe los datos y relaciones entre los datos mediante un conjunto de tablas. Cada tabla contiene un conjunto de atributos.

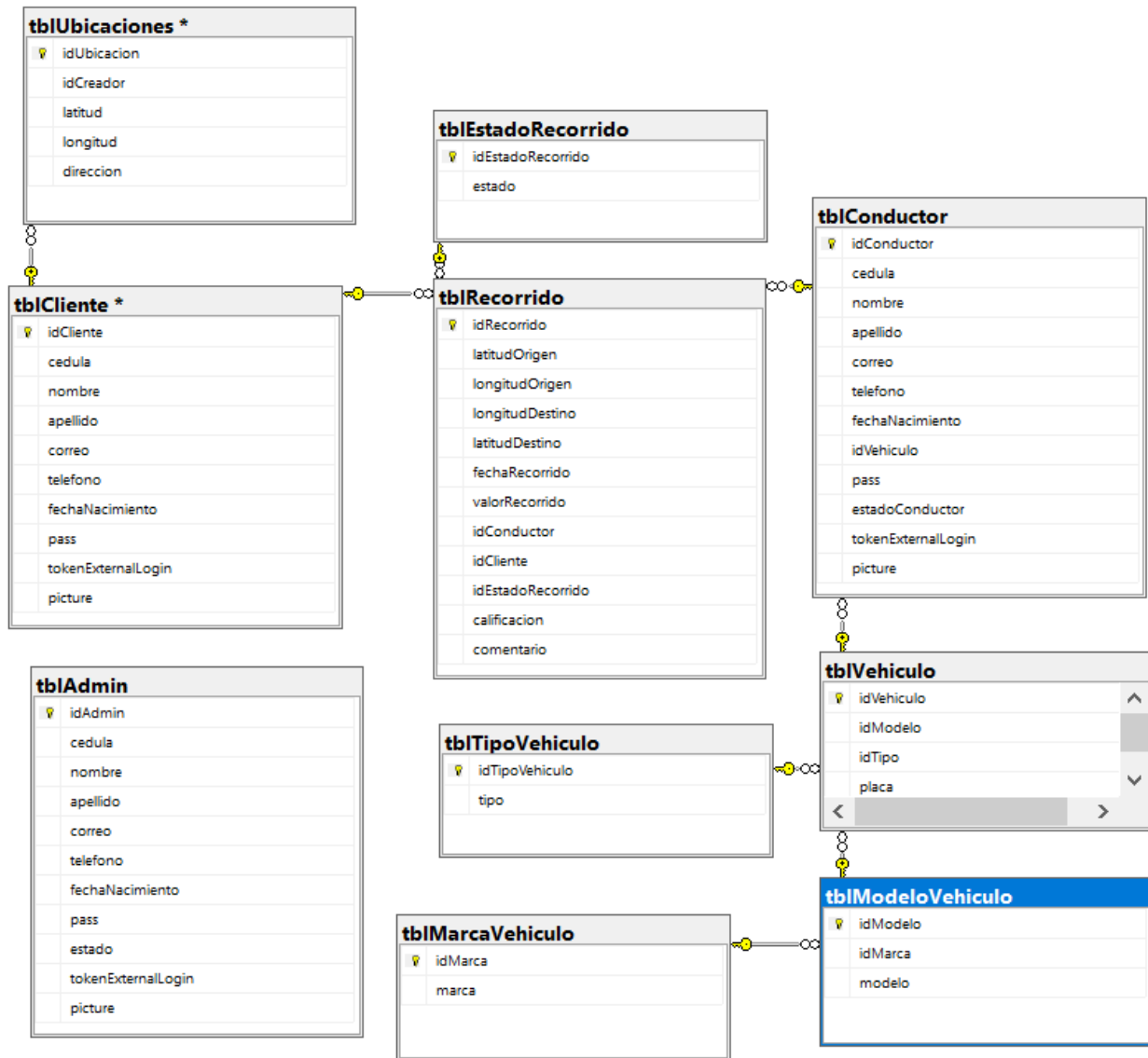


Figura 2.5 Diagrama relacional de la base de datos.

A continuación, se realiza una sucinta descripción de cada tabla que conforma el diagrama:

- **tblAdmin**: Almacena a todos los usuarios administradores. Este tipo de usuario no tiene relaciones, pero puede modificar registros de todas las tablas excepto los recorridos ni ubicaciones guardadas por los clientes.

- tblCliente: Almacena a todos los usuarios clientes. Un cliente puede guardar ubicaciones y solicitar de recorridos.
- tblConductor: Almacena a todos los usuarios conductores. Un conductor puede tener un vehículo y receptor solicitudes de recorridos.
- tblVehiculo: Almacena todos los vehículos de la empresa o conductor. Un vehículo es de un tipo y un modelo.
- tblModelo: Almacena todos los modelos de vehículos registrados en el sistema. Un modelo pertenece a una marca.
- tblTipo: Almacena todos los tipos de vehículos registrados en el sistema. Un tipo puede pertenecer a varios vehículos.
- tblMarca: Almacena todas las marcas de vehículos registrados en el sistema. Una marca puede tener varios modelos.
- tblUbicaciones: Almacena todas las ubicaciones de usuarios clientes registrados en el sistema. Una ubicación pertenece a un usuario cliente.
- tblEstadoRecorrido: Almacena todos los estados de un recorrido registrados en el sistema. Los estados que puede tener un recorrido son:
 - Solicitado: Al registrar la solicitud de un recorrido desde la aplicación cliente.
 - Respondido: Cuando un conductor ha seleccionado un recorrido solicitado.
 - Iniciado: Cuando el cliente selecciona una solicitud respondida por algún conductor.
 - Punto de partida: Cuando el conductor está en el punto de partida de la ruta.
 - Destino: Cuando el conductor llega al punto de destino de la ruta.
 - Finalizado: Cuando termina un recorrido sin ser calificado.
 - Calificado: Cuando envía una calificación del recorrido.
 - Cancelado: Cuando se cancela la solicitud.
- tblRecorrido: Almacena todos los recorridos realizados por los usuarios. Un recorrido puede tener un conductor, un cliente, un estado y varios mensajes.

- tblMensaje: Almacena todos los mensajes de los recorridos. Un mensaje pertenece a un recorrido.

2.2.3 CAPA DE LÓGICA DE NEGOCIO

A continuación, se presentan los artefactos diseñados en la capa de lógica del negocio, incluyendo el diagrama de clases, los diagramas de actividades y diagramas de estados.

La Figura 2.6 presenta el diagrama de clases del sistema. El diagrama de clases representa los aspectos estáticos de las clases que conforman el sistema distribuido. Las clases son abstracciones que representan objetos del mundo real. Cada clase posee atributos y métodos que determinan su comportamiento.

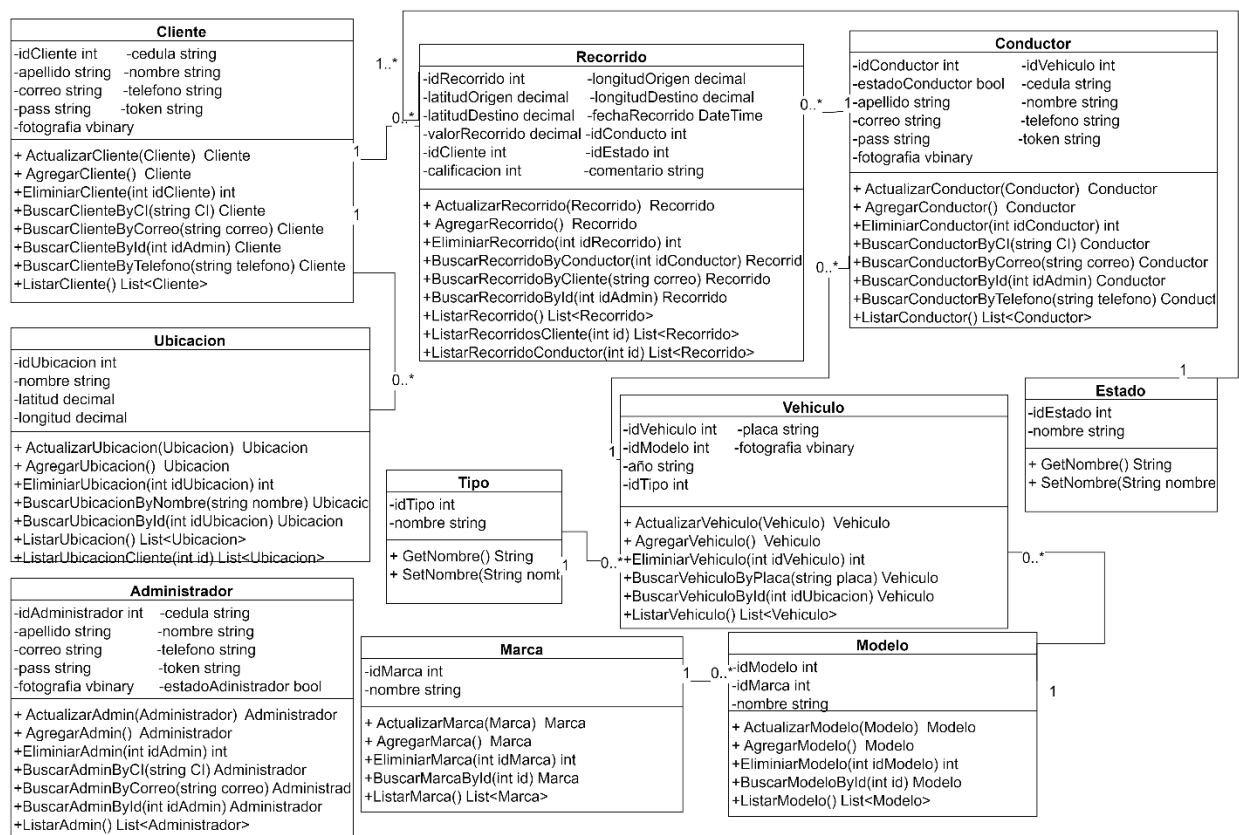


Figura 2.6 Diagrama de clases UML.

Las clases fueron nombradas del mismo modo que las entidades presentadas en la Figura 2.4 del diagrama entidad-relación, los atributos de cada clase coinciden con las columnas en la base de datos.

Por ejemplo, la clase Conductor tiene los siguientes atributos:

- idConductor: Corresponde a la llave primaria y es de valor entero.
- Cédula: Corresponde a la identificación de un conductor y es de tipo texto.
- Nombre: Corresponde al nombre de un conductor y es de tipo texto.
- Apellido: Corresponde al apellido de un conductor y es de tipo texto.
- Correo: Corresponde al correo de un conductor y es de tipo texto.
- Teléfono: Corresponde al teléfono de un conductor y es de tipo texto.
- fechaNacimiento: Corresponde a la fecha de nacimiento de un conductor y es de tipo date time.
- pass: Corresponde a la contraseña de un conductor y es de tipo texto. Se encripta utilizando SHA.
- estadoConductor: Corresponde al estado de un conductor y es de tipo booleano.
- fotografía: Corresponde a la foto de un conductor es de tipo Image.

Los métodos implementados para la clase Vehículo son los siguientes:

- + ActualizarConductor(Conductor) Conductor: Permite actualizar la información de un conductor.
- + AgregarConductor() int : Permite agregar un conductor.
- +EliminarConductor(int idConductor) int: Permite eliminar un conductor.
- +BuscarConductorByCI(string CI) Conductor: Permite buscar un conductor por su cédula.
- +BuscarConductorByCorreo(string correo) Conductor: Permite buscar un conductor por su correo.
- +BuscarConductorById(int idAdmin) Conductor: Permite encontrar u conductor por su ID.
- +BuscarConductorByTelefono(string teléfono) Conductor: Permite buscar un conductor por su teléfono.

- +ListarConductor() List<Conductor>: Obtiene todos los conductores registrados en el prototipo.

En cuanto a las relaciones con la clase vehículo:

- Un Vehículo tiene un modelo.
- Un vehículo tiene un tipo.

La Tabla 2.8 nos indica los métodos expuestos por el contrato de nuestro servicio WCF con una descripción corta de cada método.

Tabla 2.8 Métodos del servicio web WCF.

Métodos del contrato	Descripción
http://tempuri.org/IService/ActualizarAdmin	Permite actualizar la información de un administrador.
http://tempuri.org/IService/ActualizarCliente	Permite actualizar la información de un conductor.
http://tempuri.org/IService/ActualizarConductor	Permite actualizar la información de un cliente.
http://tempuri.org/IService/ActualizarMarca	Permite actualizar la información de una marca.
http://tempuri.org/IService/ActualizarModelo	Permite actualizar la información de un modelo.
http://tempuri.org/IService/ActualizarRecorrido	Permite actualizar la información de un recorrido.
http://tempuri.org/IService/ActualizarTipo	Permite actualizar la información de un tipo de vehículo.
http://tempuri.org/IService/ActualizarUbicaciones	Permite actualizar la información de una ubicación.
http://tempuri.org/IService/ActualizarVehiculo	Permite actualizar la información de un vehículo.
http://tempuri.org/IService/AgregarAdmin	Permite agregar un administrador.
http://tempuri.org/IService/AgregarCliente	Permite agregar un cliente.

http://tempuri.org/IService/AgregarConductores	Permite agregar un conductor.
http://tempuri.org/IService/AgregarMarcaVehiculo	Permite agregar una marca de vehículo.
http://tempuri.org/IService/AgregarModeloVehiculo	Permite agregar un modelo de un modelo de vehículo.
http://tempuri.org/IService/AgregarRecorrido	Permite agregar un recorrido.
http://tempuri.org/IService/AgregarTipoVehiculo	Permite agregar un tipo de vehículo.
http://tempuri.org/IService/AgregarUbicaciones	Permite agregar una ubicación.
http://tempuri.org/IService/AgregarVehiculo	Permite agregar un vehículo.
http://tempuri.org/IService/EliminarAdmin	Permite eliminar un administrador.
http://tempuri.org/IService/EliminarCliente	Permite eliminar un cliente por su ID.
http://tempuri.org/IService/EliminarConductor	Permite eliminar un conductor por su ID.
http://tempuri.org/IService/EliminarMarcaVehiculo	Permite eliminar una marca por su ID.
http://tempuri.org/IService/EliminarModeloVehiculo	Permite eliminar un modelo por ID.
http://tempuri.org/IService/EliminarRecorrido	Permite eliminar un recorrido por ID.
http://tempuri.org/IService/EliminarTipoVehiculo	Permite eliminar un tipo de vehículo por ID.
http://tempuri.org/IService/EliminarUbicaciones	Permite eliminar una ubicación por ID.
http://tempuri.org/IService/EliminarUbicacionesTemporales	Permite eliminar ubicaciones temporales que se crean en el proceso de solicitud de recorrido.
http://tempuri.org/IService/EliminarVehiculo	Permite eliminar un vehículo por ID.
http://tempuri.org/IService/BuscarAdminByCI	Permite buscar un administrador por cédula.
http://tempuri.org/IService/BuscarAdminByCorreo	Permite buscar un administrador por correo electrónico.
http://tempuri.org/IService/BuscarAdminByID	Permite buscar un administrador por ID.
http://tempuri.org/IService/BuscarClienteByCI	Permite buscar un cliente por cédula.
http://tempuri.org/IService/BuscarClienteByCorreo	Permite buscar un cliente por correo.
http://tempuri.org/IService/BuscarClienteByID	Permite buscar un cliente por ID.
http://tempuri.org/IService/BuscarClienteByTelefono	Permite buscar un cliente por teléfono.
http://tempuri.org/IService/BuscarConductorByCI	Permite buscar un conductor por cédula.
http://tempuri.org/IService/BuscarConductorByID	Permite buscar un conductor por ID.

http://tempuri.org/IService/BuscarConductorByMail	Permite buscar un conductor por correo.
http://tempuri.org/IService/BuscarConductorByTelefono	Permite buscar un conductor por teléfono.
http://tempuri.org/IService/BuscarMarcaVehiculoById	Permite buscar una marca por ID.
http://tempuri.org/IService/BuscarModeloVehiculoById	Permite buscar un modelo por ID.
http://tempuri.org/IService/BuscarModeloVehiculoByMarca	Permite buscar los modelos por marca.
http://tempuri.org/IService/BuscarRecorridoById	Permite buscar un recorrido por ID.
http://tempuri.org/IService/BuscarRecorridosByCliente	Permite buscar los recorridos de un cliente.
http://tempuri.org/IService/BuscarRecorridosByConductor	Permite buscar los recorridos de un conductor.
http://tempuri.org/IService/BuscarSolicitudRecorrido	Permite buscar los recorridos con estado solicitado.
http://tempuri.org/IService/BuscarTipoVehiculoById	Permite buscar un tipo por ID.
http://tempuri.org/IService/BuscarTipoVehiculoByTipo	Permite buscar un vehículo por su tipo.
http://tempuri.org/IService/BuscarUbicacionById	Permite buscar una ubicación por ID.
http://tempuri.org/IService/BuscarUbicacionByNombre	Permite buscar una ubicación por nombre.
http://tempuri.org/IService/BuscarVehiculoById	Permite buscar un vehículo por ID.
http://tempuri.org/IService/GenerarPass	Permite generar una contraseña de 8 dígitos.
http://tempuri.org/IService/ListarAdmin	Devuelve una lista de todos los administradores.
http://tempuri.org/IService/ListarAdminApellido	Devuelve una lista de todos los administradores por apellido.
http://tempuri.org/IService/ListarAdminCédula	Devuelve una lista de todos los administradores por cédula.
http://tempuri.org/IService/ListarClientes	Devuelve una lista de todos los clientes.
http://tempuri.org/IService/ListarClientesCédula	Devuelve una lista de todos los clientes por número de cédula.
http://tempuri.org/IService/ListarConductorCédula	Devuelve una lista de todos los conductores por número de cédula.
http://tempuri.org/IService/ListarConductores	Devuelve una lista de todos los

	conductores.
http://tempuri.org/IService/ListarConductorNombre	Devuelve una lista de todos los conductores por nombre.
http://tempuri.org/IService/ListarMarcaVehiculo	Devuelve una lista de todas las marcas de vehículos.
http://tempuri.org/IService/ListarModeloVehiculo	Devuelve una lista de todos los modelos de vehículos.
http://tempuri.org/IService/ListarRecorrido	Devuelve una lista de todos los recorridos.
http://tempuri.org/IService/ListarRecorridoCliente	Devuelve una lista de todos los recorridos de un cliente.
http://tempuri.org/IService/ListarRecorridoConductor	Devuelve una lista de todos los recorridos de un conductor.
http://tempuri.org/IService/ListarTipoVehiculo	Devuelve una lista de todos los tipos de vehículos.
http://tempuri.org/IService/ListarUbicaciones	Devuelve una lista de todas las ubicaciones de un cliente.
http://tempuri.org/IService/ListarVehiculo	Devuelve una lista de todos los vehículos.
http://tempuri.org/IService/SendMail	Envía un correo electrónico.
http://tempuri.org/IService/sha256_hash	Permite obtener un hash SHA256 de un texto.
http://tempuri.org/IService/sha512_hash	Permite obtener un hash SHA512 de un texto.

Por otro lado, los diagramas de actividades modelan los principales algoritmos, que proveen la funcionalidad requerida por los diferentes usuarios. A continuación, a manera de ejemplo, se presentan los diagramas de actividades de las siguientes funcionalidades:

- Registro de usuario (Figura 2.7).
- Solicitud de recorrido por parte de un usuario cliente (Figura 2.8).
- Recepción de solicitud por parte de un conductor (Figura 2.9).

La Figura 2.7 presenta el flujo de actividades a realizar durante el proceso de registro de un usuario; este incluye la validación de los datos ingresados.

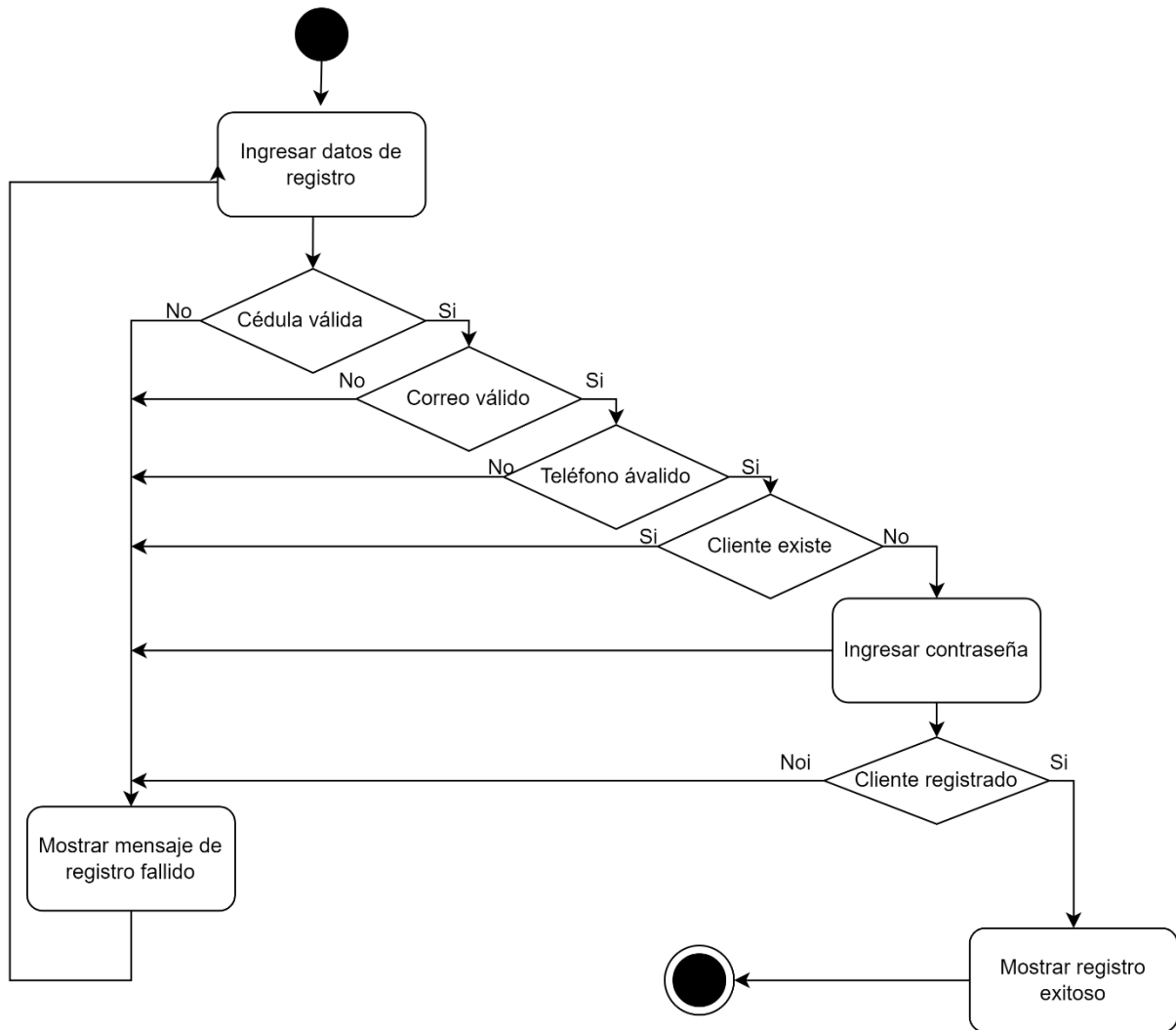


Figura 2.7 Registro de cliente.

La Figura 2.8 presenta el algoritmo para la solicitud de un recorrido. Esta característica de la aplicación del cliente permite llenar un formulario de datos para solicitar un recorrido posteriormente se selecciona una respuesta y, finalmente, inicia el viaje. Una vez terminado el recorrido se procederá a mostrar un formulario para calificar el servicio y termina el proceso.

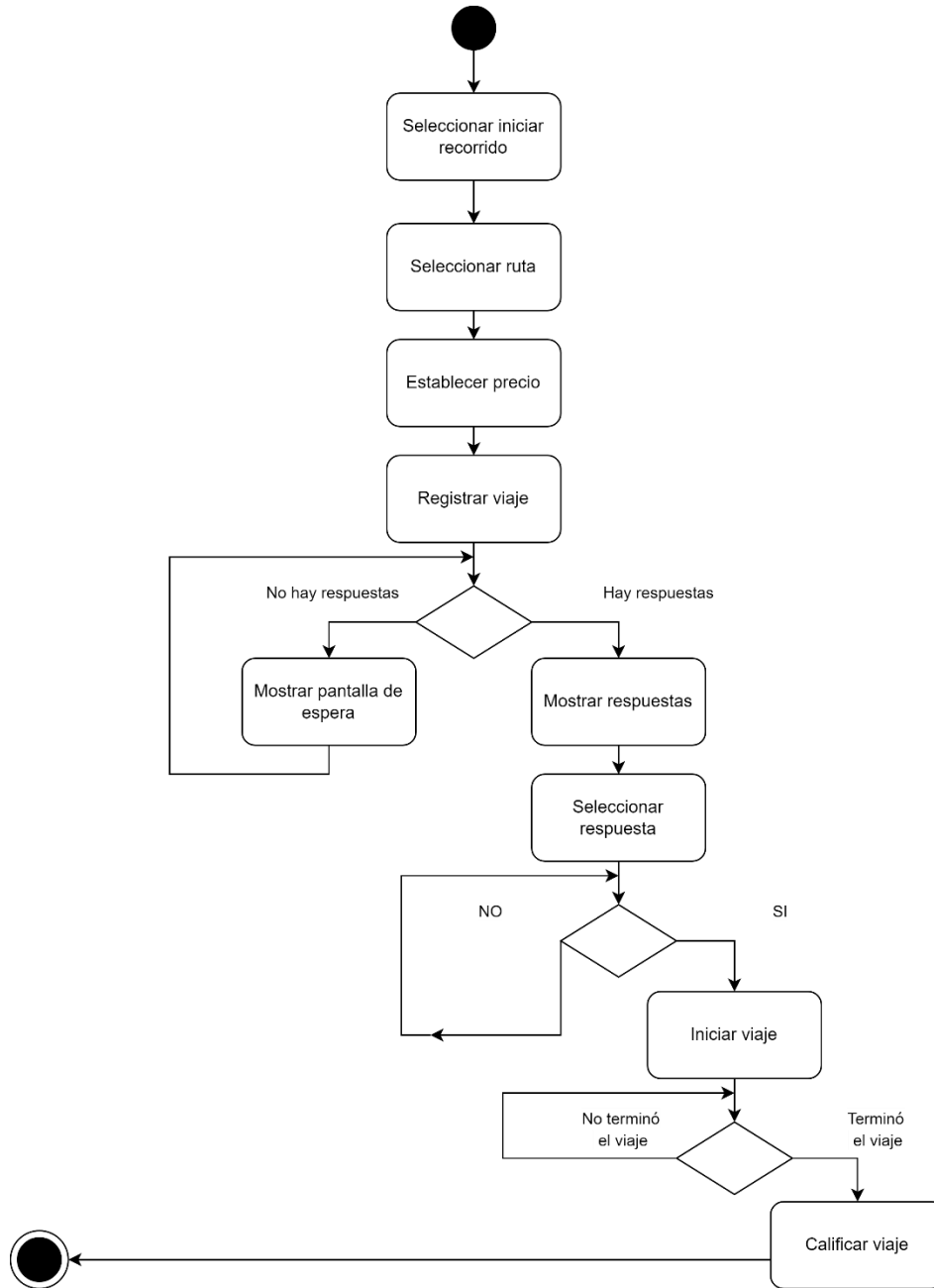


Figura 2.8 Solicitud de recorrido.

En la Figura 2.9 se presenta el algoritmo para la recepción de solicitudes en la aplicación cliente. El conductor recibe continuamente las solicitudes de recorridos realizadas por los clientes. Al seleccionar una solicitud, se enviará una notificación al cliente; si la notificación es aceptada, se inicia el recorrido. El Anexo G presenta los diagramas de actividades de todo el prototipo.

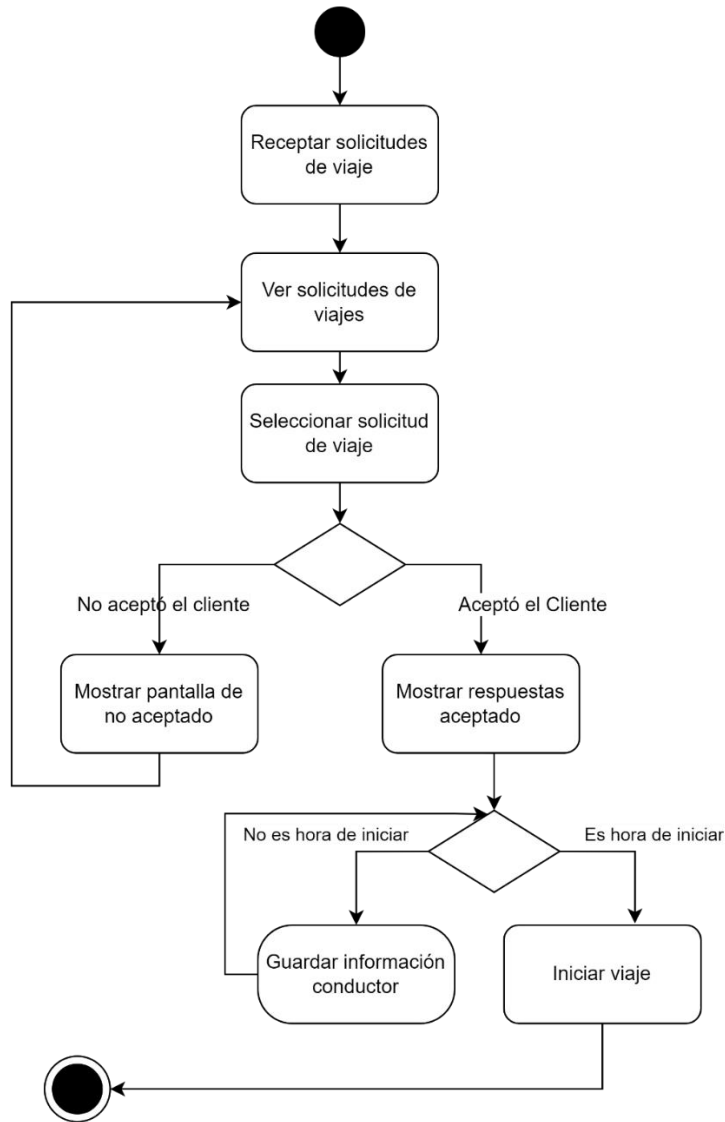


Figura 2.9 Recepción de recorrido.

Finalmente, el diagrama de estados fue usado para modelar los estados que un recorrido puede tener. La Figura 2.10 presenta los estados que puede tener un recorrido durante el proceso de solicitud en la aplicación cliente. Los estados se detallan a continuación:

- Solicitado: Al registrar la solicitud de un recorrido desde la aplicación cliente.
- Respondido: Cuando un conductor ha seleccionado un recorrido solicitado.
- Iniciado: Cuando el cliente selecciona una solicitud respondida por algún conductor.
- Punto de partida: Cuando el conductor está en el punto de partida de la ruta.

- Destino: Cuando el conductor llega al punto de destino de la ruta.
- Finalizado: Cuando termina un recorrido sin ser calificado.
- Calificado: Cuando envía una calificación del recorrido.
- Cancelado: Cuando se cancela la solicitud.

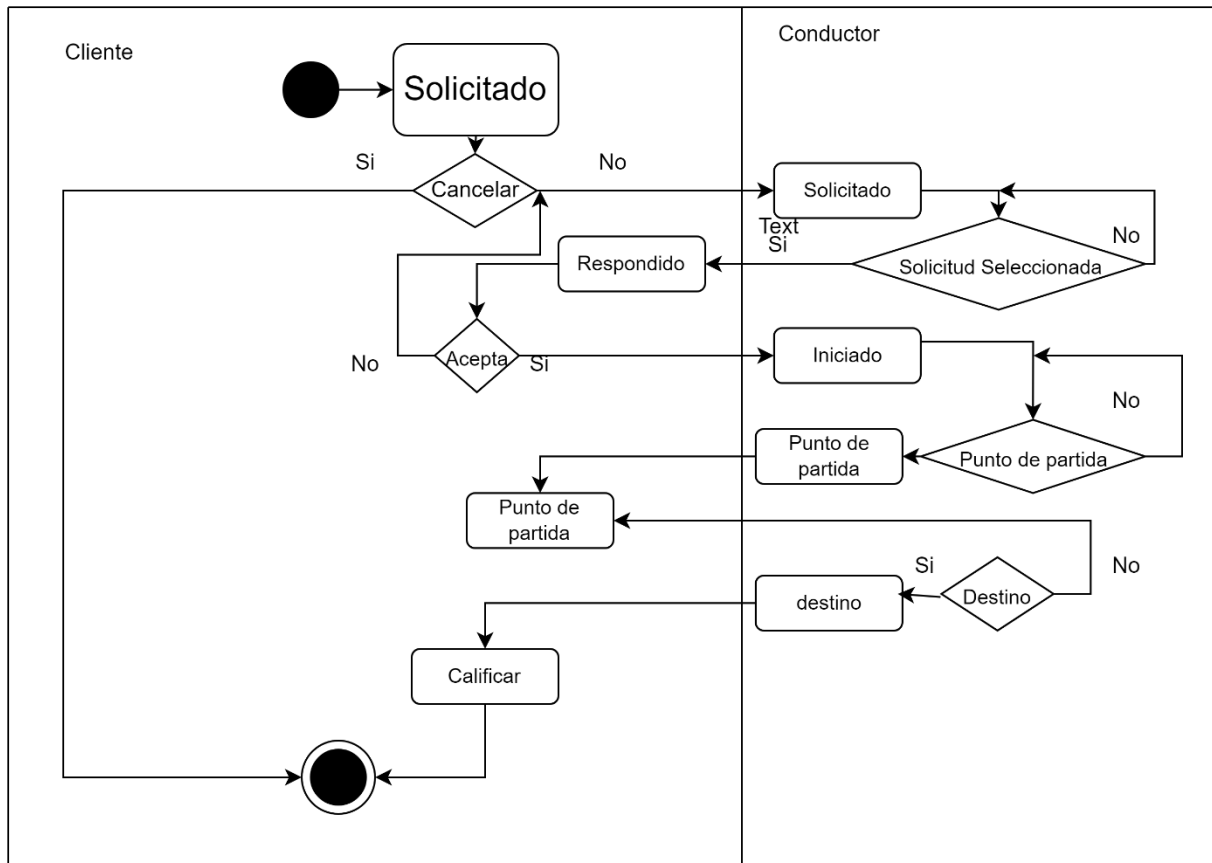


Figura 2.10 Diagrama de estados.

2.2.4 CAPA DE PRESENTACIÓN

Esta sección se muestra los sketches² realizados para la modelación de las distintas interfaces de usuario. Estos sketches nos ayudan a comprender de manera más clara las necesidades del usuario y a tomar decisiones en cuanto al diseño de la interfaz. A continuación, se muestran algunos ejemplos de los sketches realizados:

² El Sketch es boceto aproximado que representa las características principales de una interfaz gráfica en la aplicación y, a menudo, se realiza como estudio preliminar.

Inicio de Sesión



Correo electrónico	patricio.garcia@epn.edu.ec
Contraseña
Iniciar de Sesión	

No tienes cuenta? [Regístrate Aquí!](#)

Figura 2.11 Scketch del inicio de sesión de un administrador.



Figura 2.12 Scketch de la interfaz principal de un cliente.



Figura 2.13 Scketch de la interfaz principal de un conductor.

Por ejemplo, en la Figura 2.11, se encuentra la interfaz de autenticación de un usuario, la Figura 2.12 corresponde a la interfaz de solicitud de recorrido en la aplicación cliente, y la Figura 2.13 presenta la ventana de recepción de solicitudes en la aplicación conductor.

En la Figura 2.11 se presenta el formulario de autenticación de un usuario en la aplicación administrador. Se muestra las opciones que puede realizar un usuario que son 1) registrarse 2) iniciar sesión y 3) restablecer una contraseña.

En la Figura 2.12 se presenta la interfaz de usuario de solicitud de recorrido en la aplicación cliente. El usuario deberá ingresar algunos datos necesarios para poder solicitar el recorrido. También se muestra la parte superior izquierda un botón que indicará el menú de opciones que tiene un usuario en la aplicación.

En la Figura 2.13 se presenta la interfaz de recepción de solicitudes de un usuario en la aplicación conductor. El usuario inicia la recepción de solicitudes mediante una switch en la interfaz. Este despliega una lista de solicitudes y las muestra en la aplicación como se indica.

Además, esta aplicación cuenta con un botón que indica el menú de opciones que tiene un usuario en la aplicación. En el Anexo H y Anexo I se encuentran los interfaces de todas las interfaces de las aplicaciones.

La Tabla 2.9 muestra los métodos de la capa de lógica del negocio que son desencadenados desde los diferentes eventos desencadenados por los elementos gráficos de las interfaces.

Tabla 2.9 Métodos desencadenados en cada interfaz de usuario.

Aplicación cliente	
Interfaz	Métodos
LoginAppPage	sha256_hash
	BuscarClienteByCI
	BuscarClienteByCorreo
RegisterPage	BuscarClienteByCI
	BuscarClienteByCorreo
	BuscarClienteByTelefono
	AgregarCliente
MapPage	ListarTipoVehiculo
	ActualizarRecorrido
	FindRecorridoById
	FindRecorridosByCliente
ProfilePage	FindClienteById
	ActualizarCliente
InfoPage	ListarUbicaciones
	FindRecorridosByCliente
CRUDPage	AgregarUbicaciones
	EliminarUbicaciones
Aplicación Conductor	
LoginAppPage	sha256_hash
	BuscarConductorByCI
	BuscarConductorByCorreo
RegisterPage	BuscarConductorByCI

	BuscarConductorByCorreo
	BuscarConductorByTelefono
	AgregarConductor
MapPage	ActualizarRecorrido
	FindRecorridoById
	FindRecorridosByConductor
ProfilePage	FindConductorByID
	ActualizarConductor
InfoPage	ListarVehiculos
	FindRecorridosByCliente
CRUDPage	AgregarVehiculo
	SeleccionarVehiculo
Aplicación Administrador	
frmLogin	sha256_hash
	BuscarAdminByCI
	BuscarAdminByCorreo
frmRegistro	BuscarAdminByCI
	BuscarAdminByCorreo
	BuscarAdminByTelefono
	AgregarAdmin
frmPerfil	BuscarClienteByID
	ActualizarCliente
frmCliente	ListarClientes
frmMarca	ListarMarca
frmModelo	ListarModelos
frmTipo	ListarTipos
frmConductor	ListarConductores
frmAdmin	ListarAdministradores
frmClienteCRUD	BuscarClienteByID
	EliminarCliente
	ActualizarCliente

frmMarcaCRUD	BuscarMarcaByID
	EliminarMarca
	ActualizarMarca
frmModeloCRUD	BuscarModeloByID
	EliminarModelo
	ActualizarModelo
frmTipoCRUD	BuscarTipoByID
	EliminarTipo
	ActualizarTipo
frmConductorCRUD	BuscarConductorByID
	EliminarConductor
	ActualizarConductor
frmAdminCRUD	BuscarAdminByID
	EliminarAdmin
	ActualizarAdmin

2.3 IMPLEMENTACIÓN

Esta sección describe la estructura y codificación de proyecto. Los componentes del proyecto se realizaron utilizando las versiones de los lenguajes de programación mostrados en la Tabla 2.10. Para la interacción entre la capa de datos y las vistas se utiliza un servicio WCF.

Tabla 2.10 Resumen de herramientas.

COMPONENTE		Lenguaje	Versión	Framework	Versión	IDE
CAPA DE DATOS	Base de datos	SQLServer	12.0.2000.8	.NET	SQLAzure RTM	SQL Server Management Studio (SSMS)
	Acceso a datos	C#	5	.NET	4.8	Visual Studio
CAPA LÓGICA DE NEGOCIO	Servicio WCF	C#, xml	5	.NET	4.8	Visual Studio
CAPA PRESENTA-	Aplicaciones	C#, xaml	5	.NET Standard	2.1	Visual Studio
	Xamarin	C#, xaml	5	Android	4.4 - 11.0	Visual Studio

CIÓN	Aplicación ASP	C#, html,css	5	.NET	4.8	Visual Studio
------	----------------	--------------	---	------	-----	---------------

Un servicio WCF trabaja del modo cliente servidor y sobre diferentes protocolos de transporte de datos como HTTP. Los mensajes entre clientes y servidores se envían entre extremos de conexión. Un extremo de conexión define la información requerida para la comunicación con un servicio. Un servicio puede exponer uno o más puntos de conexión para una aplicación. El cliente genera un extremo de conexión compatible con los del servicio para lograr una comunicación de mensajes. La Figura 2.14 muestra el diagrama de despliegue del prototipo, el cual muestra cómo interactúan todos los componentes a través de Internet.

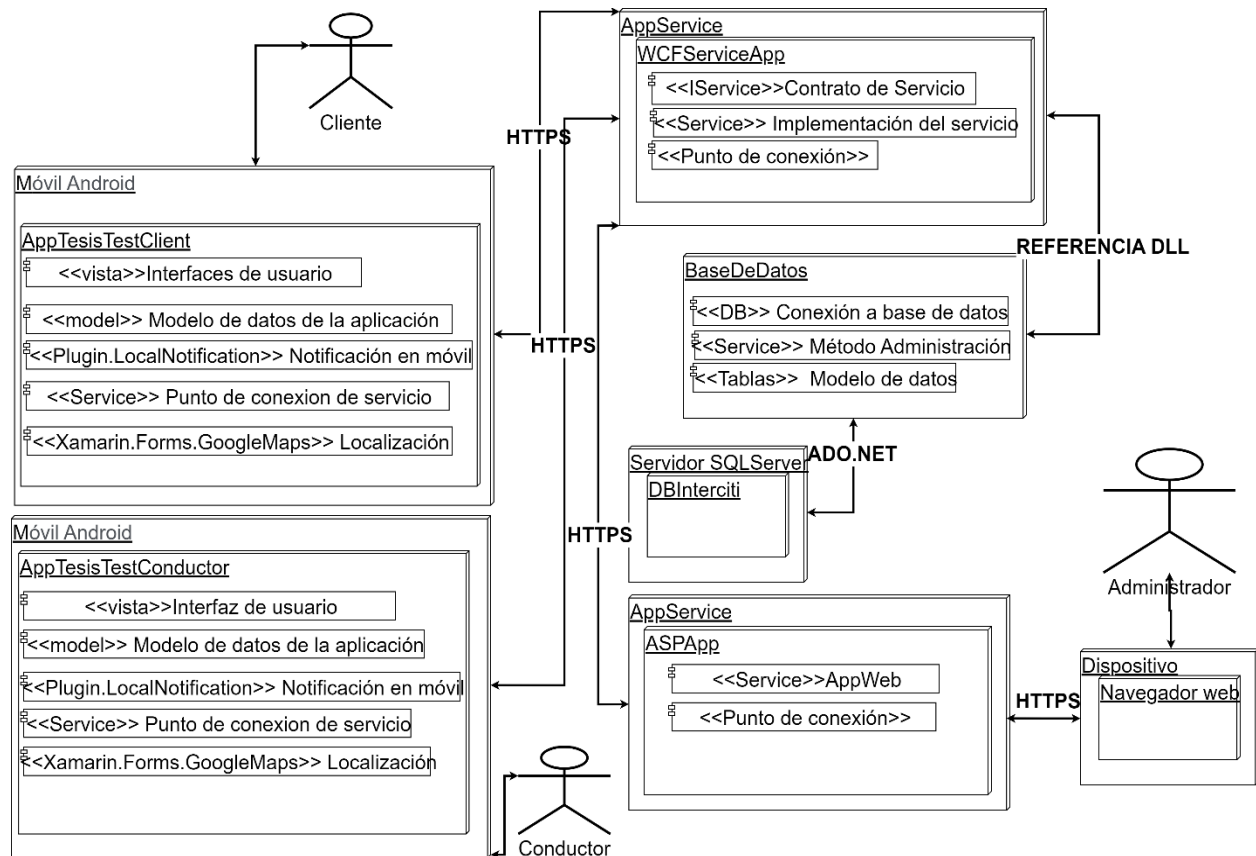


Figura 2.14 Diagrama de despliegue del prototipo.

El punto de conexión detalla donde se deben enviar los mensajes y el formato que deben tener. El servicio expone esta información como metadatos que los clientes pueden procesar para agregar clientes WCF y comunicaciones exitosas.

La Figura 2.15 indica como se comunican todos los componentes del prototipo.

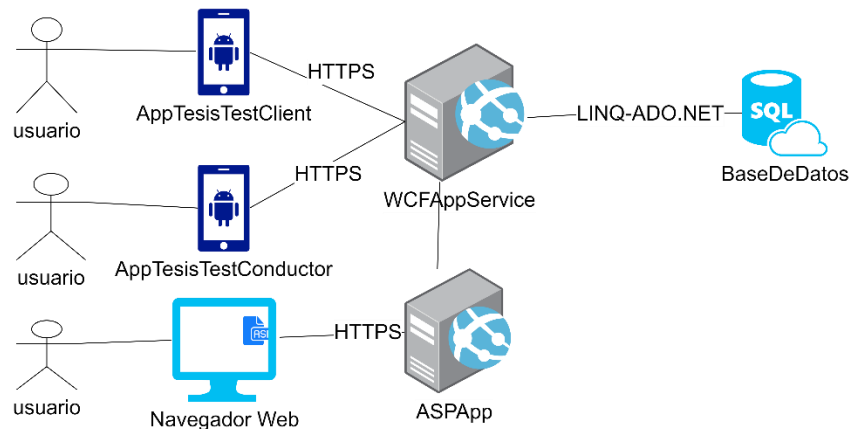


Figura 2.15 Comunicación de componentes del prototipo.

La solución de Visual Studio contiene distintos proyectos, tal como se muestra en la Figura 2.16. Cada proyecto cumple una función concreta como se describe a continuación:

- AppTesisTestClient: Es el proyecto Xamarin.Forms el cual contiene todas las interfaces y lógica de la aplicación cliente.
- AppTesisTestClient.Android: Proyecto que contiene todos los componentes necesarios para interpretar la aplicación general de AppTesisTestClient y crear una aplicación nativa de Android.
- AppTesisTestClient.Test: Contiene las pruebas realizadas mediante código para la aplicación AppTesisTestClient.
- AppTesisTestConductor: Es el proyecto Xamarin.Forms el cual contiene todas las interfaces y lógica de nuestra aplicación conductor.
- AppTesisTestConductor.Android: Proyecto que contiene todos los componentes necesarios para interpretar la aplicación general de AppTesisTestConductor y crear una aplicación nativa de Android.
- AppTesisTestConductor.Test: Contiene las pruebas realizadas mediante código para la aplicación AppTesisTestConductor.
- ASPApp: Es el proyecto ASP .NET el cual contiene todas las interfaces y lógica de nuestra aplicación administrador.

- BaseDeDatos: Contiene los métodos de acceso a base de datos.
- WCFServiceApp: Se trata de un proyecto del tipo aplicación de servicio. Contiene los métodos que se publican en el servicio web.
- WCFServiceApp.Test: Contiene las pruebas realizadas mediante código para la aplicación WCFServiceApp.

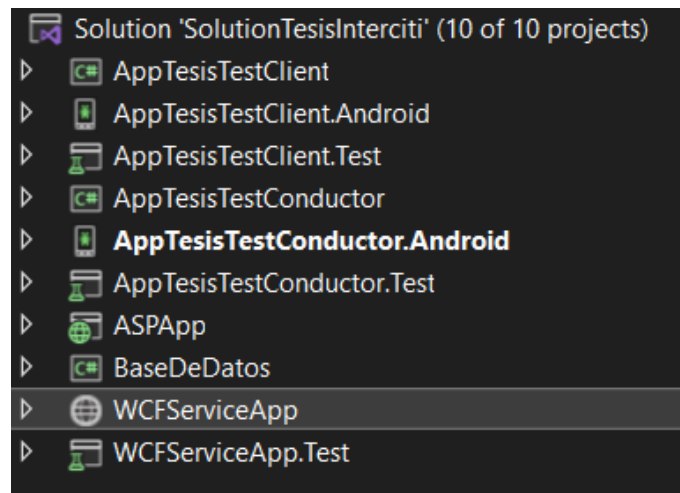


Figura 2.16 Estructura de la solución en Visual Studio.

A continuación, se describen algunos detalles de la implementación de cada capa de la arquitectura antes descrita.

2.3.1 CAPA DE DATOS

La capa de acceso a datos se ha implementado mediante el ORM (Object Relational Mapping) LINQ (Language-Integrated Query), que permite la conexión a la base de datos y la ejecución de consulta en C#.

El proyecto BaseDeDatos tiene la estructura presentada en la Figura 2.17. Este contiene la clase DB, que contiene la cadena de conexión a la base de datos, las clases que hacen referencia a las entidades de la base de datos, y la clase Métodos que contiene los métodos necesarios para modificar los registros en la base de datos.

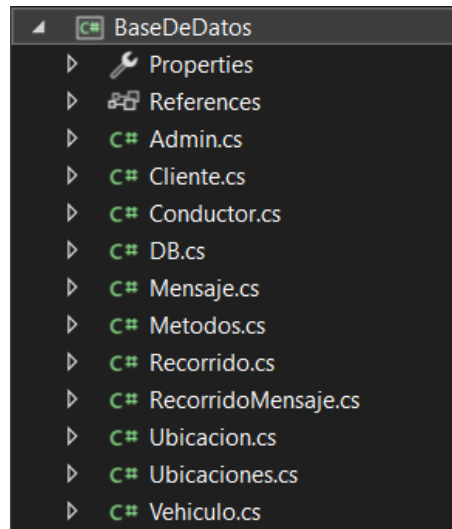


Figura 2.17 Estructura del proyecto BaseDeDatos.

La conexión con la base de datos se realiza mediante la clase DB que hereda de DataContext. La clase tiene como atributo todos los modelos de datos relacionados a la base. Es necesaria una cadena de conexión como se indica en el Código 2.1.

```
using System.Data.Linq;

namespace BaseDeDatos
{
    3 references
    public class BD : DataContext
    {
        //Constructor de la clase que contiene el string para la conexión a la base de datos DBIntercitiPrueba
        1 reference
        public BD() : base(@"Data Source=sqlserverprototipo.database.windows.net;Initial Catalog=DBInterciti;
            Persist Security Info=True;User ID=Patoadmin;Password=.Patin12345;") { }
        public Table<Vehiculo> tblVehiculo;
        public Table<Conductor> tblConductor;
    }
}
```

Código. 2.1 Conexión a la base de datos.

LINQ permite mapear las tablas de una base de datos en objetos basados en un modelo de clase. Por ejemplo, en el Código 2.2 se muestra la clase Admin. Esta clase posee la etiqueta Table(Name = "tblAdmin") que indica con que tabla se realiza el mapeo de información. Además, cada atributo debe tener el mismo nombre y tipo de dato si se desea realizar una correcta comunicación entre el ORM y el servidor de base de datos.

```

6      using System.Data.Linq.Mapping;
7      using System.Runtime.Serialization;
8
9      namespace BaseDeDatos
10     {
11         [Table(Name = "tblAdmin")]
12         public class Admin
13         {
14             [Column(IsPrimaryKey = true, IsDbGenerated = true)]
15             private int idAdmin;
16             [Column]
17             private string cedula;

```

Código. 2.2 Clase Admin Base de datos.

Los métodos de administración de la base de datos utilizan expresiones propias del ORM para acceder y modificar los datos. Por ejemplo, el Código 2.3 presenta el método ListarClientes, en el que el código de las líneas 47 y 48 realiza una selección de los registros en la tabla tblCliente.

```

44     public List<Cliente> ListarClientes()
45     {
46         List<Cliente> listaPaketes = new List<Cliente>();
47         var paquetes = from iterPaquete in baseDatos.tblCliente
48                       select iterPaquete;
49         //lista
50         foreach (Cliente iter in paquetes)
51         {
52             listaPaketes.Add(iter);
53         }
54         if (listaPaketes.Count > 0)
55         {
56             return listaPaketes;
57         }
58         else
59         {
60             return null;
61         }
62     }

```

Código. 2.3 Método ListarClientes Base de datos.

2.3.2 CAPA LOGICA DE NEGOCIO

El servicio web WCF tiene la estructura que se indica en la Figura 2.19. Sus componentes principales son: 1) una interfaz del servicio contiene los métodos y clases que se publicarán, 2) una clase que consiste en la implementación lógica del servicio y 3) un archivo XML de configuración web.config.

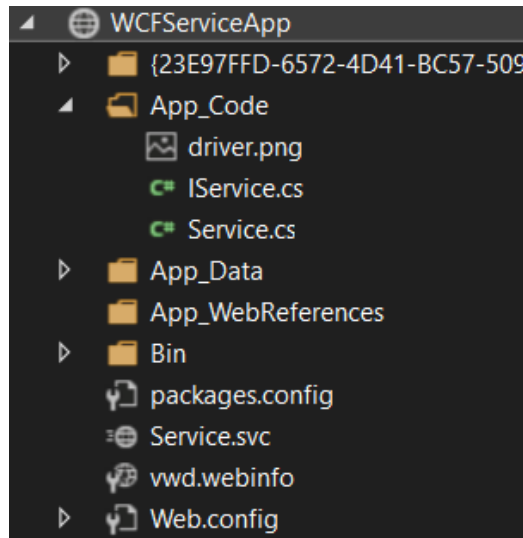


Figura 2.19 Estructura del proyecto WCFServiceApp.

La interfaz del servicio muestra la etiqueta ServiceContract, el cual indica que se trata del contrato de servicio; es decir, todos los métodos que puede ejecutar el consumidor del servicio. La etiqueta OperationContract permite exponer los métodos, tal como se indica en el Código 2.4.

```

11  [ServiceContract]
    1 reference
12  public interface IService
13  {
14      [OperationContract]
    1 reference
15      Admin FindAdminByID(int ci);

```

Código. 2.4 Interfaz del servicio.

El Código 2.5 muestra la implementación de la interfaz de servicio en la cual se utilizan los métodos creados en el proyecto BaseDeDatos.

```

18  public class Service : IService
19  {
20      Metodos metodos = new Metodos();
    1 reference
21      public Admin FindAdminByID(int ci)
22      {
23          return AdminAApp(metodos.FindAdminByID(ci));
24      }
25  }

```

Código. 2.5 Implementación de la interfaz del servicio.

La Configuración del servicio se muestra en el Código 2.6. El cual se establece el tipo de protocolo para el punto de conexión que crea el servicio.

```
<system.serviceModel>
  <bindings>
    <basicHttpsBinding>
      <binding name="myBinding" maxReceivedMessageSize="2147483647" sendTimeout="10:10:
    </binding>
  </basicHttpsBinding>
</bindings>
```

Código. 2.6 Punto de conexión del servicio.

2.3.3 CAPA DE PRESENTACIÓN

2.3.3.1 APLICACIONES MÓVILES

Los proyectos de las aplicaciones móviles contienen una estructura similar (Figura 2.20). Esta contiene todas las interfaces de usuario desarrolladas en XML Views, una carpeta que contiene los modelos de datos de la aplicación Model, métodos de validaciones y lógica de la aplicación Services, y una carpeta que contiene un archivo de fuentes que contiene los iconos de la aplicación Fuentes. Además, contiene una carpeta con los metadatos necesarios para utilizar el servicio WCF.

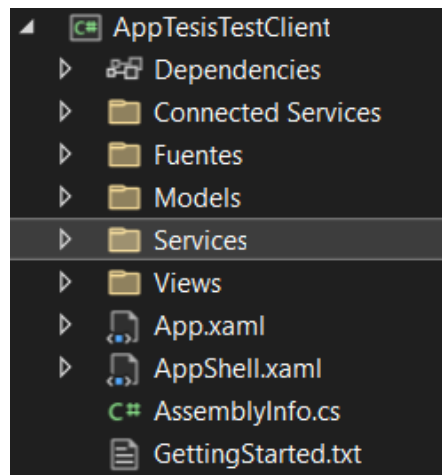


Figura 2.20 Estructura del proyecto AppTesisTestClient.

Por ejemplo, el Código 2.7 muestra el método LlenaListView, el mismo que desencadena los métodos de la capa aplicación ListarUbicaciones y BuscarRecorridoByCliente. El resultado de estos métodos es enviado a una lista visual en la interfaz de usuario. El Código 2.8 indica como se vinculan los modelos de datos a un control visual. El método WCFToAppList se encarga de pasar los datos obtenidos al modelo de datos de la aplicación.

```

public async void LlenarListView()
{
    var x = Servicio.WCFToAppList(Servicio.client.ListarUbicaciones(Servicio.usua
    var y = Servicio.WCFToAppList(Servicio.client.FindRecorridosByCliente(Servici
    await Device.InvokeOnMainThreadAsync(() => {
        if (x != null)
        {
            lstUbicaciones.ItemsSource = x;
        }
        if (y != null)
        {
            lstRecorridos.ItemsSource = y;
        }
    });
}

```

Código. 2.7 Asignación de datos a interfaz gráfica.

```

94 <ListView x:Name="lstRecorridos" RowHeight="180" ItemsSource="{Binding Models.Recorrido}"
95     VerticalOptions="FillAndExpand" HorizontalOptions="FillAndExpand">
96     <ListView.ItemTemplate>
97     <DataTemplate >
98     <ViewCell>
99     <Frame BackgroundColor="White" BorderColor="#F0F0F0" HasShadow="False">
100     <Grid HorizontalOptions="FillAndExpand" VerticalOptions="Start">
101     <Grid.ColumnDefinitions>
102     <ColumnDefinition Width="*" />
103     <ColumnDefinition Width="60" />
104     </Grid.ColumnDefinitions>
105     <StackLayout Orientation="Vertical" Margin="0" HorizontalOptions="Star
106
107     <Label Text="{Binding IdConductor.Nombre}" TextColor="ForestGree
108     <Label Text="Punto de partida:" VerticalOptions="Center" Horizontal
109     <Label Text="{Binding Origen}" TextColor="Black" FontSize="10" V
110     <Label Text="Destino:" VerticalOptions="Center" HorizontalOptions=
111     <Label Text="{Binding Destino}" TextColor="Black" FontSize="10"
112     </StackLayout>
113     <StackLayout Grid.Column="1" Orientation="Vertical" Margin="0" Horiz
114     <Label Text="Datos de Conductor:" FontSize="12"></Label>
115     <Label Text="{Binding IdConductor.Nombre}" TextColor="Black" For

```

Código. 2.8 Vinculación de datos a la interfaz gráfica.

La función principal de las aplicaciones cliente y servidor es la realización de un recorrido. Este proceso se detalla a continuación:

- La solicitud de un recorrido solo puede ser realizada desde una aplicación cliente. Una vez completados todos los datos se puede realizar una solicitud. Esto agrega un nuevo recorrido, que no tiene un conductor asignado y un estado de 1 (Solicitado), tal como se muestra en el Código 2.9. Por otro lado, el conductor puede habilitar el conmutador para escuchar las solicitudes de transporte disponibles según lo indicado en el Código 2.10.

```

private async void SolicitarRecorrido(object sender, EventArgs e)
{
    layoutDirecciones.IsVisible = false;
    layoutBarraBu.IsVisible = false;
    layoutRecorrido.IsVisible = true;
    if (Servicio.client.AgregarRecorrido((decimal)ubicacionInicial.Latitude, (decimal)ubicacionIn:
        await DisplayAlert("Error!", "No se logro crear la solicitud. Intente nuevamente", "OK");
        return;
    }
    thread.RunWorkerAsync();
    timer.Start();
}

```

Código. 2.9 Solicitar Recorrido.

```

private async void ActivarEscuchaSolicitudes(object sender, ToggledEventArgs e)
{
    if (e.Value)
    {
        await Device.InvokeOnMainThreadAsync(async () => {
        });
        if (Servicio.usuarioConectado.Vehiculo.IdVehiculo>0)
        {
            if (!thread.IsBusy)
            {
                start = true;
                thread.RunWorkerAsync();
                timer.Start();
                lstListaRecorrido.IsVisible = true;
                layoutListaRecorrido.IsVisible = true;
            }
        }
        else
    }
}

```

Código. 2.10 Recepción de solicitudes de Recorrido.

- Después de registrar la solicitud de transporte, el conductor puede seleccionar un recorrido y realizar una contraoferta al precio ofertado. El estado del recorrido pasa a 2 (Respondido).
- Las respuestas son recibidas por el cliente. El cliente puede seleccionar una de las respuestas enviadas por los conductores. El estado pasa a ser 3 (Iniciado).
- El conductor se mueve hacia el punto de partida. Se toma un radio de 50 metros debido al margen de error del GPS (Sistema de Posicionamiento Global). El estado del recorrido pasa a 4 (Punto de partida).
- El conductor se mueve hacia el destino. Se toma un radio de 50 metros debido al margen de error del GPS. El estado del recorrido pasa a 6 (Destino).

- El cliente puede evaluar el recorrido realizado. El estado cambia a 8 (Calificado).
- Si el cliente no califica el recorrido el estado pasa a 0 (Finalizado).
- Si del lado del cliente o conductor cancelan el recorrido, pasa su estado a -1 (Cancelado).

Las aplicaciones pueden notar el cambio de estado ya que cuentan con un hilo de ejecución que se encuentra validando en cuál de los posibles estados se encuentra un determinado recorrido, tal como se muestra en el Código 2.11.

```

359 switch (iter.IdEstadoRecorrido)
360 {
361     case 2:
362         notificacion.Title = "Respuestas!";
363         notificacion.Description = $"{iter.IdConductor.M
364         notificacion.NotificationId = iter.IdConductor.I
365         notificacion.BadgeNumber = iter.IdConductor.IdCo
366         await NotificationCenter.Current.Show(notificaci
367         recorridosAux.Add(iter);
368         break;
369     case 4:
370         recorrido = iter;
371         notificacion.Title = "Llego el conductor!";
372         notificacion.Description = $"{iter.IdConductor.M
373         notificacion.NotificationId = iter.IdConductor.I
374         notificacion.BadgeNumber = iter.IdConductor.IdCo
375         await NotificationCenter.Current.Show(notificaci
376         // recorridosAux.Add(iter);
377         ServiceReferenceInterciti.Recorrido recorridoAux
378         recorridoAux.EstadoRecorrido = 5;
379         inicio = false;
380         if (!threadTracking.IsBusy)
381         {
382             Servicio.client.ActualizarRecorrido(recorrid
383             threadTracking.RunWorkerAsync();
384         }
385         break;

```

Código. 2.11 Escucha de cambio de estados en la aplicación cliente.

2.3.3.2 APLICACIÓN WEB

El proyecto de la aplicación web posee la estructura que se indica en la Figura 2.18. Este proyecto está conformado por la carpeta Content y Scripts que almacenan código CSS y JavaScript utilizado por Bootstrap, y la carpeta fonts que contiene archivos de fuente de FontAwesome³ utilizados para los iconos de la aplicación. En la carpeta Sources están los

³ FontAwesome es una librería de íconos disponible en internet.

recursos como las imágenes, interfaces de usuario y una clase que realiza las validaciones dentro de la aplicación web.

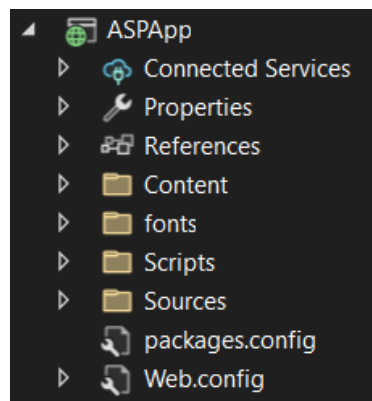


Figura 2.21 Estructura del proyecto ASPApp.

La aplicación está formada por una página maestra MPInterciti.Master y varias páginas hijas, que muestran la información de los diferentes datos que forman parte del prototipo, tal como se indica en la Figura 2.22.

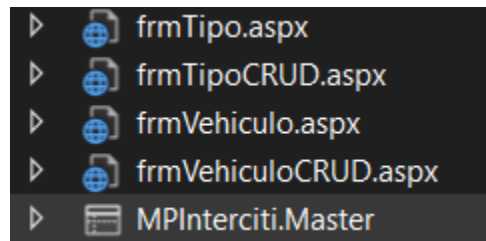


Figura 2.22 Páginas de ASPApp.

La aplicación web realiza tareas de visualización y edición de información. Por ejemplo, Código 2.12 ilustra un control GridView que permite mostrar un origen de datos en forma de una tabla. Este control define una etiqueta Columns y dentro de esta pueden ir todos los atributos de un origen de datos que se seleccionen. También se pueden agregar columnas adicionales como, en este caso, para mostrar ciertas opciones de administración.

```

44 <asp:GridView ID="grdVCliente" runat="server" AllowPaging="True" Au
45 <AlternatingRowStyle BackColor="White" ForeColor="#284775" Hori
46 <Columns>
47 <asp:TemplateField HeaderText="Opciones de administrador">
48 <ItemTemplate>
49
50 <button id="bntSelect" onserverclick="bntSelect_Cli
51 <button id="btnEliminar" onclick="return confirm('A
52 <button id="btnModificar" onserverclick="btnModific
53 </ItemTemplate>
54 </asp:TemplateField>
55 <asp:TemplateField HeaderText="Foto">
56 <ItemTemplate>
57 <asp:Image id="Img1" runat="server" Width="65"
58
59 </ItemTemplate>
60 </asp:TemplateField>

```

Código. 2.12 GridView en la aplicación del administrador.

Además, cada página tiene código en C#, tal como se indica en el ejemplo de Código 2.13. El método obtiene el identificador único del registro seleccionado y redirecciona a la página frmAdminCRUD, concatenando la opción seleccionada (U: Update) y el identificador único del administrador.

```

42 protected void btnModificar_Click(object sender, EventArgs e)
43 {
44     var btn = (System.Web.UI.HtmlControls.HtmlButton)sender;
45     GridViewRow seleccion = (GridViewRow)(btn.NamingContainer);
46     var ci = seleccion.Cells[2].Text;
47     Response.Redirect("/Sources/Pages/frmAdminCRUD.aspx?op=U&id="
48 }
49

```

Código. 2.13 Método clic botón editar del usuario administrador.

3 PRUEBAS Y RESULTADOS DEL PROTOTIPO

Esta sección tiene como objetivo presentar los resultados de las pruebas de funcionamiento del sistema distribuido. Se han realizado pruebas unitarias, de integración y de aceptación.

3.1 PRUEBAS UNITARIAS

Se realizó pruebas unitarias de los métodos implementados con el objetivo de validar el correcto funcionamiento de los componentes de forma individual. Por ejemplo, el proyecto BaseDeDatos contiene una única clase, tal como se indica en la Figura 3.1. La clase contiene todos los métodos de administración de la base de datos.

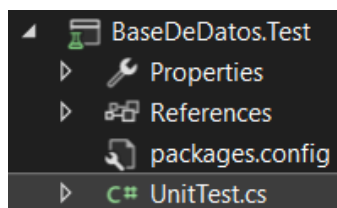


Figura 3.1 Servicio en la nube.

Para la codificación de las pruebas unitarias se utilizó el framework y la guía de Microsoft *Microsoft.VisualStudio*. Además, se usó *TestTools* [21], que permite conservar el estado del código y detectar errores y fallos de forma anticipada.

Las pruebas se realizaron comparando la salida del método ejecutado con la salida esperada. Por ejemplo, el Código 3.14 indica un ejemplo de una prueba unitaria exitosa del método *AgregarCliente*. Se ejecuta el método que devuelve 0 en caso de éxito y -1 en caso de error. Finalmente, la clase *Assert* permite comparar que la salida de la ejecución con otro valor. Si el valor es igual se marca como una prueba exitosa como muestra la Figura 3.2. Las pruebas unitarias forman parte de la solución del proyecto presente en el Anexo K.

```
[TestMethod]
✓ | 0 references
public void AgregarClienteTest()
{
    valorEsperado = client.AgregarCliente("ClienteServicio", c
    Assert.AreEqual(0, valorEsperado);
}
```

Código. 3.14 Prueba unitaria de Agregar Cliente Exitosa.

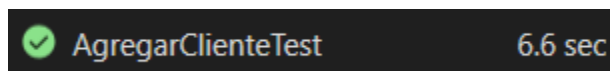


Figura 3.2 Prueba unitaria exitosa.

3.2 PRUEBAS DE INTEGRACIÓN

Del mismo modo se realizaron pruebas de integración entre los distintos componentes del prototipo con el objetivo de validar el funcionamiento en conjunto de todo el sistema distribuido. Para ello, se han creado los proyectos WCFServiceApp,Test, AppTesisTest.Test, AppTesisTest.Test, ASPApp.Test. La estructura de los proyectos es similar al realizado en las pruebas unitarias, una clase que contiene las pruebas de los distintos métodos.

Las pruebas de integración se realizaron de la siguiente manera: 1) servicio WCF – Base de datos, 2) aplicación cliente servicio WCF, 3) aplicación conductor servicio WCF y 4) aplicación administrador servicio WCF.

En todas las pruebas de integración se utilizó el framework xUnit. A diferencia del TestTool, xUnit permite realizar pruebas en proyectos .Net Standard como las aplicaciones móviles de Xamarin.Forms.

Al igual que las pruebas unitarias, los métodos de pruebas unitarias se realizan verificando la salida del método ejecutado con la salida esperada.

Por ejemplo, el Código 2.15 muestra dos pruebas de integración para la autenticación. Esta muestra si la autenticación fue exitosa o no. Este método retorna null cuando la autenticación es errónea y un objeto cuando es exitosa. Las pruebas de integración forman parte de la solución del proyecto presente en el Anexo K.

```
[Fact]
0 references
public void LoginCI()
{
    var distancia = Servicio.GetLoginCI("1724718182", "12345678");
    Assert.NotNull(distancia);
}

[Fact]
0 references
public void LoginCIError()
{
    var distancia = Servicio.GetLoginCI("1724718182", "123456789");
    Assert.Null(distancia);
}
```

Código. 3.15 Prueba de integración de Agregar Cliente Exitosa.

3.3 PRUEBAS DE ACEPTACIÓN

Esta sección muestra las pruebas de aceptación del prototipo mostrando el cumplimiento de los requisitos establecidos por los usuarios.

Al terminar las pruebas de funcionamiento se realizó pruebas servicios de recorrido con el objetivo de validar el correcto funcionamiento del sistema.

3.3.1 REQUISITOS FUNCIONALES

Con el objetivo de ratificar los requisitos funcionales se realizó una encuesta de satisfacción mediante formularos de office. La encuesta contiene preguntas para: 1) validar las historias de usuario y 2) preguntas de opinión.

Se realizó una encuesta por aplicación, las encuestas se aplicaron a los diferentes conductores, clientes y administradores que utilizaron la aplicación en al menos una ocasión. Las preguntas de estas encuestas forman parte del Anexo J.

3.3.2 REQUISITOS NO FUNCIONALES

Finalmente se realizó una encuesta de satisfacción para la validación de los requisitos no funcionales. Las preguntas de esta encuesta se encuentran en el Anexo J.

3.4 VERIFICACIÓN Y VALIDACIÓN

Al realizar la encuesta de satisfacción de requisitos funcionales a 10 clientes, 10 conductores y 2 administradores se obtuvo el resultado que se presenta en la Tabla 3.1.

Tabla 3.1 Resultados de encuestas de satisfacción de requisitos funcionales.

CLIENTE			
PREGUNTA	Si[%]	No[%]	Código
¿La aplicación permite el registro de usuarios?	100%	0%	RFCL001
¿La aplicación permite el autenticase como usuario?	100%	0%	RFCL002
¿Ha logrado actualizar su información después de registrarse?	100%	0%	RFCL003
¿Ha logrado realizar la solicitud de un recorrido mediante la aplicación?	100%	0%	RFCL004
¿Ha logrado guardar y visualizar ubicaciones en la aplicación?	100%	0%	RFCL005
¿Ha logrado calificar un recorrido realizado previamente?	100%	0%	RFCL006
CONDUCTOR			
¿La aplicación permite el registro de usuarios?	100%	0%	RFCO001
¿La aplicación permite el autenticase como usuario?	100%	0%	RFCO002
¿Ha logrado actualizar su información después de registrarse?	100%	0%	RFCO003
¿Ha logrado receptor las solicitudes de un recorrido mediante la	100%	0%	RFCO004

aplicación?			
¿Ha logrado visualizar los vehículos de la empresa en la aplicación?	100%	0%	RFCO005
¿Ha logrado seleccionar un vehículo en la aplicación?	100%	0%	RFCO006
ADMINISTRADOR			
¿La aplicación permite el registro de usuarios?	100%	0%	RFAD001
¿La aplicación permite el autenticase como usuario?	100%	0%	RFAD002
¿Ha logrado actualizar su información después de registrarse?	100%	0%	RFAD003
¿Ha logrado administrar los vehículos(Agregar, Editar, Eliminar y visualizar) mediante la aplicación?	100%	0%	RFAD004
¿Ha logrado administrar los recorridos(Visualizar) mediante la aplicación?	100%	0%	RFAD005
¿Ha logrado administrar los clientes(Agregar, Editar, Eliminar y visualizar) mediante la aplicación?	100%	0%	RFAD006
¿Ha logrado administrar los conductores (Agregar, Editar, Eliminar y visualizar) mediante la aplicación?	100%	0%	RFAD007

Se validó que las historias de usuario fueron implementadas de manera exitosa, ya que cada pregunta corresponde a una historia de usuario. También en cada prueba se establecieron los siguientes criterios de aceptación presentados en la Tabla 3.2.

Tabla 3.2 Criterios de aceptación de historia de usuario.

CÓDIGO	CRITERIOS DE ACEPTACIÓN	ESTADO
RFCL001	El usuario debe ingresar con su número de identificación como usuario.	Finalizado
	Los caracteres de contraseña deben tener al menos 8 dígitos.	Finalizado
	El usuario no podrá ver datos que no le pertenezcan.	Finalizado
RFCL002	Los números de teléfono deben contener 10 dígitos.	Finalizado
	La cédula debe ser un número de cédula ecuatoriano válido.	Finalizado
	El correo de usuario debe ser validado.	Finalizado
	Todos los datos de registro deben ser completados.	Finalizado
RFCL003	Se mostrará una vista de información de usuario.	Finalizado
	Al dar clic en el botón editar se habilitará controles para modificar la	Finalizado

	información.	
	Al dar clic en el botón guardar se actualiza la información.	Finalizado
RFCL004	Al dar clic en Punto de partida se muestra una ventana de selección de ubicación.	Finalizado
	Al dar clic en Destino se muestra una ventana de selección de ubicación.	Finalizado
	Al completar todos los datos necesarios se habilita un botón para realizar la solicitud.	Finalizado
	Se muestra la ubicación del conductor de forma concurrente.	Finalizado
RFCL005	Se muestra una vista con la ubicación del usuario.	Finalizado
	Al dar clic en agregar se puede seleccionar una ubicación.	Finalizado
	Al dar clic en guarda se agrega la información de la ubicación.	Finalizado
	Al dar clic en el boto eliminar se borra la ubicación de los registros.	Finalizado
RFCL006	Al finalizar un recorrido se muestra una ventana de calificación.	Finalizado
	Al escoger una calificación se habilita la opción de enviar la calificación.	Finalizado
	Al dar clic en cancelar el recorrido se registra sin calificación.	Finalizado
CÓDIGO	CRITERIOS DE ACEPTACIÓN	ESTADO
RFCL001	El usuario debe ingresar con su número de identificación como usuario.	Finalizado
	Los caracteres de contraseña deben tener al menos 8 dígitos.	Finalizado
	El usuario no podrá ver datos que no le pertenezcan.	Finalizado
RFCL002	Los números de teléfono deben contener 10 dígitos.	Finalizado
	La cédula debe ser un número de cédula ecuatoriano válido.	Finalizado
	El correo de usuario debe ser validado.	Finalizado
	Todos los datos de registro deben ser completados.	Finalizado
RFCL003	Se mostrará una vista de información de usuario.	Finalizado
	Al dar clic en el botón editar se habilitará controles para modificar la información.	Finalizado
	Al dar clic en el botón guardar se actualiza la información.	Finalizado
RFCL004	Al dar clic en un conmutador se habilita la escucha de solicitudes.	Finalizado
	Al seleccionar una solicitud se puede realizar una contraoferta.	Finalizado
	Al aceptar el cliente l contraoferta inicia el recorrido.	Finalizado

	Al está el cliente menos de 50 metros de distancia del punto de partida se notifica al usuario.	Finalizado
	Al está el cliente menos de 50 metros de distancia del destino se notifica al usuario de la llegada.	Finalizado
RFCO005	Se muestra una vista con los vehículos de la empresa.	Finalizado
	Al dar clic en agregar se puede agregar un vehículo.	Finalizado
RFCO006	El usuario no puede realizar la escucha de solicite sin un vehículo seleccionado.	Finalizado
	El vehículo se puede seleccionar desde la lista de vehículos disponibles.	Finalizado
CÓDIGO	CRITERIOS DE ACEPTACIÓN	ESTADO
RFAD001	El usuario debe ingresar con su número de identificación como usuario.	Finalizado
	Los caracteres de contraseña deben tener al menos 8 dígitos.	Finalizado
	El usuario no podrá ver datos que no le pertenezcan.	Finalizado
RFAD002	Los números de teléfono deben contener 10 dígitos.	Finalizado
	La cédula debe ser un número de cédula ecuatoriano válido.	Finalizado
	El correo de usuario debe ser validado.	Finalizado
	Todos los datos de registro deben ser completados.	Finalizado
RFAD003	Se mostrará una vista de información de usuario.	Finalizado
	AL dar clic en el botón editar se habilitará controles para modificar la información.	Finalizado
	AL dar clic en el botón guardar se actualiza la información.	Finalizado
RFAD004	Al dar clic en un agregar se muestra el formulario para agregar vehículos.	Finalizado
	Al dar clic en un editar se muestra el formulario para editar vehículos.	Finalizado
	Al dar clic en un ver se muestra el formulario para mostrar información de vehículos.	Finalizado
	Al dar clic en un eliminar se elimina el vehículo seleccionado.	Finalizado
RFAD004	Al dar clic en un agregar se muestra el formulario para agregar vehículos.	Finalizado
	Al dar clic en un editar se muestra el formulario para editar vehículos.	Finalizado
	Al dar clic en un ver se muestra el formulario para mostrar información de vehículos.	Finalizado

	Al dar clic en un eliminar se elimina el vehículo seleccionado.	Finalizado
RFAD004	Al dar clic en un agregar se muestra el formulario para agregar vehículos.	Finalizado
	Al dar clic en un editar se muestra el formulario para editar vehículos.	Finalizado
	Al dar clic en un ver se muestra el formulario para mostrar información de vehículos.	Finalizado
	Al dar clic en un eliminar se elimina el vehículo seleccionado.	Finalizado
RFAD004	Al dar clic en un agregar se muestra el formulario para agregar vehículos.	Finalizado
	Al dar clic en un editar se muestra el formulario para editar vehículos.	Finalizado
	Al dar clic en un ver se muestra el formulario para mostrar información de vehículos.	Finalizado
	Al dar clic en un eliminar se elimina el conductor seleccionado.	Finalizado

Por otro lado, los resultados de las encuestas de satisfacción de requisitos no funcionales se muestran en la Tabla 3.3.

Tabla 3.3 Resultados de encuestas de satisfacción de requisitos no funcionales.

PREGUNTA	Si	No	Código
Es necesario un usuario y contraseña para acceder a la aplicación.	X		RNFCL001
La aplicación está disponible en un teléfono móvil Android.	X		RNFCL002
La aplicación almacena la información de manera persistente.	X		RNFCL003
La aplicación está disponible en un teléfono móvil Android.	X		RNFCL004

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- El presente proyecto logró llevar a cabo el desarrollo de un prototipo de sistema distribuido para la empresa de transporte Interciti S.A. Este prototipo tiene la finalidad de facilitar el proceso de solicitudes de transporte y gestionar la información.
- Durante el desarrollo del presente trabajo de titulación, se ha logrado profundizar en los fundamentos teóricos de las tecnologías y herramientas necesarias para la gestión de bases de datos, servicios web, desarrollo web, y desarrollo móvil.
- Se realizó el análisis de requerimientos del sistema distribuido mediante la aplicación de entrevistas y encuestas a los conductores, clientes y administradores de la compañía. Se diseñó la arquitectura y componentes del sistema distribuido que satisfagan los requerimientos que se definieron mediante el análisis.
- Se implementó cada uno de los componentes del sistema. En la capa de datos se utilizó SSMS para la gestión de la base de datos. La lógica de negocio se realizó en un servicio WCF, las aplicaciones móviles se realizaron utilizando Xamarin.Forms y la aplicación web se realizó utilizando ASP.net. Todos estos componentes forman parte de una única solución de Visual Studio y fueron codificados en C#.
- Se validó el correcto funcionamiento del sistema mediante el uso de pruebas unitarias, de integración y de aceptación, con el propósito de validar y verificar que el prototipo funciona de acuerdo con las necesidades de los usuarios.
- Después de utilizar el framework de WCF en nuestra aplicación, encontramos algunos desafíos relacionados con las limitaciones de cuotas de llamada al servicio y latencias en la versión para estudiantes. Además, descubrimos que su implementación en un entorno de producción puede ser costosa. A pesar de estos inconvenientes, pudimos utilizar eficazmente las herramientas y alcanzar nuestros objetivos de desarrollo.

4.2 RECOMENDACIONES

- Las aplicaciones que consumen el servicio web no muestran latencias muy significativas en su comunicación; no obstante, el uso de métodos asíncronos podría mejorar la experiencia de usuario.
- Se recomienda tener en cuenta las versiones de los paquetes utilizados en el desarrollo de este prototipo, para evitar inconvenientes al trabajar desde diferentes entornos de trabajo.
- En futuro se podría incrementar la funcionalidad del prototipo para realizar solicitudes transporte con reservación anticipada para la aplicación cliente y la posibilidad de aumentar los puntos de parada en un recorrido.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] B. Toranza, “Vista de Las TIC en las empresas”. [En línea] Disponible en: <http://www.fitecvirtual.org/ojs-3.0.1/index.php/clic/article/view/279/245> (consultado may 28, 2021).
- [2] Microsoft, “What is Xamarin.Forms? - Xamarin”. [En línea] Disponible en: <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin-forms> (consultado abr. 17, 2022).
- [3] Microsoft, “ASP.NET overview | Microsoft Docs”. [En línea] Disponible en: <https://docs.microsoft.com/en-us/aspnet/overview> (consultado abr. 17, 2022).
- [4] Cinvestav, “Sistemas distribuidos ”. Disponible en: https://www.tamps.cinvestav.mx/~vjsosa/clases/sd/sistemas_distribuidos_panorama.pdf (consultado abr 28, 2021).
- [5] F. de Asís López Fuentes, “Sistemas distribuidos”, 2015. [En línea]. Disponible en: http://dccd.cua.uam.mx/libros/archivos/03IXStream_sistemas_distribuidos.pdf (Consultado: abr. 16, 2022)
- [6] Andrews, Gregory R, “Foundations of multithreaded, parallel, and distributed programming”. [En línea]. Disponible en: <https://archive.org/details/foundationsofmul0000andr> (consultado abr. 15, 2022).
- [7] IBM , “Componentes de aplicaciones basadas en la web”. [En línea]. Disponible en: <https://www.ibm.com/docs/es/db2-for-zos/11?topic=environment-components-web-based-applications> (consultado abr. 17, 2022).
- [8] Microsoft, “Web services protocols interoperability guide - WCF”. [En línea]. Disponible en: <https://docs.microsoft.com/en-us/dotnet/framework/wcf/feature-details/web-services-protocols-interoperability-guide> (consultado abr. 17, 2022).
- [9] IBM, “Servicios web ”. [En línea]. Disponible en: <https://www.ibm.com/docs/es/was-nd/9.0.5?topic=services-web> (consultado abr. 17, 2022).
- [10] Microsoft, “SQL Server technical documentation - SQL Server”. [En línea]. Disponible en: <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15> (consultado abr. 17, 2022).
- [11] Microsoft, “What Is Windows Communication Foundation - WCF”. [En línea]. Disponible en: <https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf> (consultado abr. 17, 2022).
- [12] Microsoft, “Qué es Azure: Servicios en la nube de Microsoft”. [En línea]. Disponible en: https://azure.microsoft.com/es-es/overview/what-is-azure/?ef_id=EA1aIQobChMII7_80qmR9gIVlaCGCh0htwGBEAAAYASABEgKHFvD_BwE%3AG%3As&OCID=AID2201052_SEM_EA1aIQobChMII7_80qmR9gIVlaCGCh0htwGBEAAAYASABEgKHFvD_BwE%3AG%3As&gclid=EA1aIQobChMII7_80qmR9gIVlaCGCh0htwGBEAAAYASABEgKHFvD_BwE (consultado abr. 17, 2022).

- [13] Asana, “¿Qué es la metodología Kanban y cómo funciona?”. [En línea]. Disponible en: <https://asana.com/es/resources/what-is-kanban> (consultado ago. 22, 2022).
- [14] Lucidchart, “¿Qué es un diagrama entidad-relación?”. [En línea]. Disponible en: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion> (consultado ago. 22, 2022).
- [15] Microsoft, “Introducción a las pruebas unitarias - Visual Studio (Windows)”. [En línea]. Disponible en: <https://docs.microsoft.com/es-es/visualstudio/test/getting-started-with-unit-testing?view=vs-2022&tabs=dotnet%2Cmstest> (consultado ago. 28, 2022).

6 ANEXOS

ANEXO A. TABLERO KANBAN

ANEXO B. REQUISITOS FUNCIONALES

ANEXO C. REQUISITOS NO FUNCIONALES

ANEXO D. HISTORIAS DE USUARIO

ANEXO E. DIAGRAMAS DE CASOS DE USO

ANEXO F. CASOS DE USO

ANEXO G. DIAGRAMAS DE ACTIVIDADES

ANEXO H. SCKETCHES DE APLICACIÓN WEB

ANEXO I. SCKETCHES DE APLICACIONES MÓVILES

ANEXO J. PRUEBAS ACEPTACIÓN

ANEXO K. CÓDIGO DEL PROYECTO

Anexo A

TABLERO KANBAN

En este Anexo se muestra la totalidad de las tareas realizadas durante todo el desarrollo del proyecto.

Tabla A.1 Tablero Kanban.

Done			
Tarea	Subtarea	Fecha inicio	Fecha Finalización
Estudiar el framework Xamarin		10-Jan-2022	25-Jan-2022
Estudiar el framework ASP		26-Jan-2022	9-Feb-2022
Estudiar el framework Windows Communication Foundation		10-Feb-2022	24-Feb-2022
Describir las herramientas Xamarin	Consulta	25-Feb-2022	3-Mar-2022
	Escritura	4-Mar-2022	10-Mar-2022
Listar las tareas a realizar (fase visualización Kanban)		11-Mar-2022	11-Mar-2022
Entrevistar al administrador y 10 conductores de la compañía	Entrevista con administradores	12-Mar-2022	12-Mar-2022
	Entrevista a conductores	13-Mar-2022	13-Mar-2022
	Creación de formularios	14-Mar-2022	14-Mar-2022
	Análisis de resultados	15-Mar-2022	15-Mar-2022
Realizar un tablero de desarrollo con las actividades a realizar		16-Mar-2022	16-Mar-2022
Crear los sketches para las vistas de la aplicación cliente		17-Mar-2022	17-Mar-2022
Crear los sketches para las vistas de la aplicación conductor		18-Mar-2022	18-Mar-2022

Crear los sketches para las vistas de la aplicación administrador		19-Mar-2022	19-Mar-2022
Realizar el diagrama relacional de la base de datos	Modelación de datos	20-Mar-2022	20-Mar-2022
	Creación del diagrama	21-Mar-2022	21-Mar-2022
Realzar el diagrama de clases		22-Mar-2022	23-Mar-2022
Realizar el diagrama de casos de uso		24-Mar-2022	25-Mar-2022
Realizar el diagrama de actividades		26-Mar-2022	27-Mar-2022
Definir funciones generales del servicio web		28-Mar-2022	29-Mar-2022
Hacer un cronograma de las actividades a realizar (fase de limitación y gestión de trabajo en curso Kanban)		29-Mar-2022	30-Mar-2022
Instalar el IDE Visual Studio 2019		31-Mar-2022	31-Mar-2022
Instalar las dependencias y paquetes necesarios para Xamarin		31-Mar-2022	31-Mar-2022
Instalar los paquetes necesarios para Windows Communication Foundation		31-Mar-2022	31-Mar-2022
Instalar los paquetes necesarios para ASP		31-Mar-2022	31-Mar-2022
Instalar SQL Server		31-Mar-2022	31-Mar-2022
Levantar un repositorio GIT para el control de versiones del código		1-Apr-2022	1-Apr-2022
Crear el proyecto en la API de mapas de Google		1-Apr-2022	1-Apr-2022
Codificar la base de Datos en SQL Server		2-Apr-2022	8-Apr-2022

Anexo B

REQUISITOS FUNCIONALES

Basados en las entrevistas con el administrador y los conductores de la empresa, se realizaron formularios de Microsoft con el fin de recabar información acerca de las funcionalidades requeridas para las distintas aplicaciones.

APLICACIÓN CLIENTE

Podemos ver que la encuesta fue a los clientes que ocupan frecuentemente los servicios de la empresa Interciti S.A., tal como se muestra en la Figura B.1.

1. ¿Con que frecuencia solicita un vehículo en la empresa?

[Más detalles](#)

[Información](#)

● 1-2 veces por semana	1
● 2-4 veces por semana	3
● 4-6 veces por semana	6
● Otras	0



FIGURA B.1 Pregunta 1 Clientes.

La siguiente pregunta mostrada en la Figura B.2 muestra que en su mayoría las solicitudes de los clientes son realizadas mediante llamadas telefónicas a la empresa.

2. ¿Mediante que mecanismo solicita el vehículo en la empresa?

[Más detalles](#)

[Información](#)

● Presencial	2
● Llamada telefonica	6
● Mensajes de Texto	0
● Otras	2

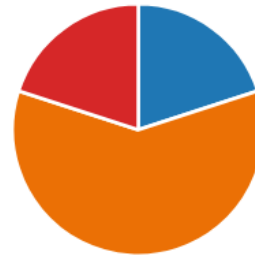


Figura B.2 Pregunta 2 Clientes.

Esta pregunta permite evidenciar que el mecanismo utilizado actualmente por los clientes no les agrada, en un acercamiento con el administrador de la empresa expresó que sus clientes han manifestado quejas al tener que realizar sus solicitudes mediante llamadas telefónicas.

3. ¿Considera usted eficiente el mecanismo utilizado para solicitar el vehículo?

[Más detalles](#)

Información

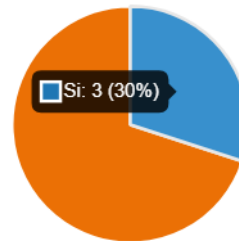


Figura B.3 Pregunta 3 Clientes.

APLICACIÓN CONDUCTOR

La siguiente pregunta, mostrada en la Figura B.4, indica que en su mayoría las solicitudes de servicio son realizadas mediante llamadas telefónicas desde la empresa al conductor.

1. ¿Como recibe usted las solicitudes para realizar el servicio de transporte?

[Más detalles](#)

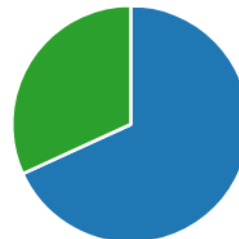
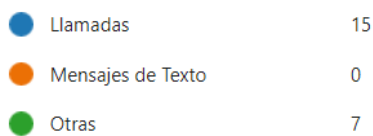


Figura B.4 Pregunta 1 Conductores.

En Figura B.5 podemos constatar que existen varios inconvenientes al realizar las solicitudes como se vienen realizando actualmente. Estos, en su mayoría, se deben a que no existe una estructura de información normalizada y a la demora en el proceso de brindar el servicio.

2. ¿Presenta algún inconveniente al recibir la solicitud de transporte?

[Más detalles](#)

● Sin inconvenientes	0
● Llamadas muy largas	7
● Direcciones erróneas	14
● Horarios inapropiados	8
● Otras	3



Figura B.5 Pregunta 2 Conductores.

APLICACIÓN ADMINISTRADOR

La empresa cuenta con dos usuarios administradores a los que se les realizó las siguientes preguntas. En la pregunta de la Figura B.6 podemos ver que llevan sus registros en papel y tablas de Excel generadas manualmente por los administradores.

1. ¿Como registra la información de viajes realizados por sus conductores?

[Más detalles](#)

● Papel	2
● Excel	1
● Otras	0



Figura B.6 Pregunta 1 Administrador.

Esta pregunta Figura B.7 pudimos ver que los vehículos en su mayoría son proporcionados por la empresa, pero también existen empleados de la empresa que utilizan sus propios vehículos para brindar servicio.

2. ¿La empresa provee los vehículos para el servicio de transporte?

[Más detalles](#)

● Si	0
● No	0
● Otras	2



Figura B.7 Pregunta 2 Administrador.

Al igual que los conductores, los administradores presentan inconvenientes al realizar la recepción de servicios, tal como se muestra en la Figura B.8.

3. ¿Presenta algún inconveniente al receptor la solicitud de transporte? Marque las casillas necesarias

[Más detalles](#)

● Sin inconvenientes	0
● Direcciones errneas	2
● Demoras en el servicio	1
● Llamadas muy tardadas por cad...	1
● Otras	2



Figura B.8 Pregunta 3 Administrador.

6. Observaciones. Si tiene alguna sugerencia puede escribirla aquí.

[Más detalles](#)

1
Respuestas

Respuestas más recientes

"La aplicacion web sole debe visualizar los recorridos. Tambien se manejan d..."

Figura B.9 Pregunta 6 Administrador.

De las respuestas a estas preguntas, se elicitaron los siguientes requisitos funcionales:

Tabla B.1 Requisitos funcionales por aplicación.

APP CLIENTE		
CÓDIGO	TÍTULO	DESCRIPCIÓN

RFCL001	Registro	El usuario deberá ingresar la información necesaria para su registro.
RFCL002	Autenticación	Para el ingreso al sistema se utilizará número de cédula y contraseña
RFCL003	Actualización de datos	El usuario podrá modificar sus datos.
RFCL004	Solicitar recorrido	El usuario solicitará un vehículo
RFCL005	Calificar recorrido	Después de solicitar un recorrido y concluirlo se tendrá la opción de calificar el mismo.
RFCL006	Mis ubicaciones	Se podrá guardar ubicaciones de los clientes.
APP CONDUCTOR		
CÓDIGO	TÍTULO	DESCRIPCIÓN
RFCO001	Registro	El usuario deberá ingresar la información necesaria para su registro.
RFCO002	Autenticación	Para el ingreso al sistema se utilizará número de cédula y contraseña
RFCO003	Actualización de datos	El usuario podrá modificar sus datos.
RFCO004	Receptar recorridos	El usuario podrá recibir solicitudes de servicio por un tiempo limitado de tiempo.
RFCO005	Vehículos	El usuario podrá ver todos los vehículos de la empresa y agregar su propio vehículo.
RFCO006	Seleccionar vehículo	El usuario podrá cambiar el vehículo seleccionado.
APP ADMINISTRADOR		
CÓDIGO	TÍTULO	DESCRIPCIÓN
RFAD001	Registro	El usuario deberá ingresar la información necesaria para su registro.
RFAD002	Autenticación	Para el ingreso al sistema se utilizará número de cédula y contraseña
RFAD003	Actualización de datos	El usuario podrá modificar sus datos.
RFAD004	Información vehículos	El usuario podrá realizar un CRUD de los vehículos de la empresa.
RFAD005	Recorridos	El usuario podrá ver la información de los recorridos realizados en la empresa.
RFAD006	Información clientes	El usuario podrá realizar un CRUD de los clientes de la empresa.
RFAD007	Información Conductores	El usuario podrá realizar un CRUD de los conductores de la empresa.

REQUISITOS FUNCIONALES GENERALES DEL SISTEMA

Tabla B.2 Necesidades funcionales del prototipo.

CÓDIGO	TÍTULO	DESCRIPCIÓN
RF001	Registro	El usuario deberá ingresar la información necesaria para su registro.
RF002	Autenticación	Para el ingreso al sistema se utilizará número de cédula y contraseña
RF003	Actualización de datos	El usuario podrá modificar sus datos.
RF004	Solicitar recorrido	El usuario solicitará un vehículo
RF005	Calificar recorrido	Después de solicitar un recorrido y concluirlo se tendrá la opción de calificar el mismo.
RF006	Mis ubicaciones	Se podrá guardar ubicaciones de los clientes.
RF007	Mis recorridos	Muestra la información de los recorridos de un usuario determinado.
RF008	Receptar recorridos	El usuario podrá recibir solicitudes de servicio por un tiempo limitado de tiempo.
RF009	Vehículos	El usuario podrá ver todos los vehículos de la empresa y agregar su propio vehículo.
RF010	Seleccionar vehículo	El usuario podrá cambiar el vehículo seleccionado.
RF011	Gestionar vehículos	El usuario podrá realizar un CRUD de los vehículos de la empresa.
RF012	Gestionar Recorridos	El usuario podrá ver la información de los recorridos realizados en la empresa.
RF013	Gestionar clientes	El usuario podrá realizar un CRUD de los clientes de la empresa.
RF014	Gestionar conductores	El usuario podrá realizar un CRUD de los conductores de la empresa.

Anexo C

REQUISITOS NO FUNCIONALES

APLICACIÓN CLIENTE

La Figura C.1 nos indica que los clientes están interesados en contar con una aplicación móvil para realizar la solicitud de recorridos.

1. ¿Cómo registra la información de viajes realizados por sus conductores?

[Más detalles](#)

● Papel	2
● Excel	1
● Otras	0

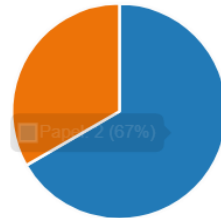


Figura C.1 Pregunta 4 Clientes.

La Figura C.2 indica que todos los clientes encuestados usan un teléfono móvil.

5. ¿Tiene usted un teléfono móvil?

[Más detalles](#)

[Información](#)

● Sí	10
● No	0



Figura C.2 Pregunta 5 Cliente.

Igualmente nos indica que todos los usuarios que realizaron la encuesta tienen acceso a internet y un celular con sistema operativo Android ver Figura C.3. y C.4.

6. ¿Tiene su teléfono móvil acceso a internet?

[Más detalles](#)

● Si	10
● No	0

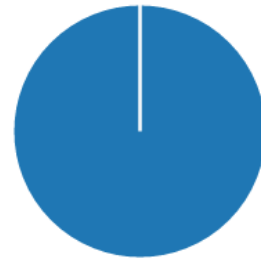


Figura C.3 Pregunta 6 Cliente.

7. ¿Qué marca es su teléfono móvil?

[Más detalles](#)

[Información](#)

● Xiaomi	3
● Apple	0
● OPPO	0
● Samsung	4
● Motorola	0
● Otras	3

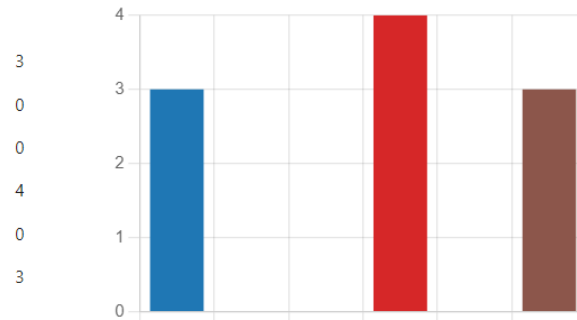


Figura C.4 Pregunta 7 Cliente.

APLICACIÓN CONDUCTOR

La Figura C.5 nos indica que los conductores están interesados en contar con una aplicación móvil para realizar la recepción de recorridos.

1. ¿Cómo registra la información de viajes realizados por sus conductores?

[Más detalles](#)

● Papel	2
● Excel	1
● Otras	0

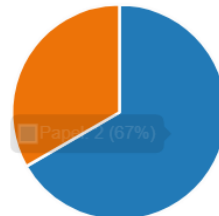


Figura C.5 Pregunta 3 Conductor.

Igualmente nos indica que todos los usuarios que realizaron la encuesta tienen acceso a internet y un celular con plan de datos móvil y con S.O. además Android (ver Figuras C.6, C.7 y C.8).

4. ¿Tiene usted un teléfono móvil?

[Más detalles](#)

Información

● Si	15
● No	0

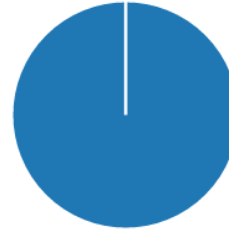


Figura C.6 Pregunta 4 Conductor.

5. ¿Tiene su teléfono móvil acceso a internet?

[Más detalles](#)

● Si, WIFI	0
● Si, Plan celular	15
● No	0



Figura C.7 Pregunta 5 Conductor.

6. ¿Qué marca es su teléfono móvil?

[Más detalles](#)

● Xiaomi	5
● Apple	0
● Samsung	4
● Nokia	0
● Motorola	0
● Huawei	1
● POCO	1
● OPPO	2
● Otras	2

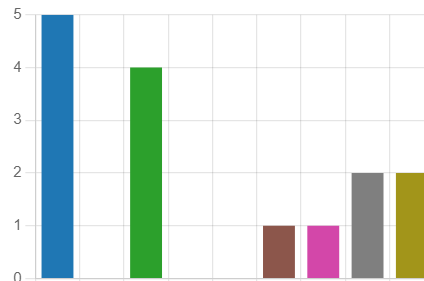


Figura C.8 Pregunta 6 Conductor.

APLICACIÓN ADMINISTRADOR

La empresa cuenta con dos usuarios administradores a los que se les realizó las siguientes preguntas. La Figura C.9 nos indica que los clientes están interesados en contar con una aplicación web para realizar la visualización y administración de los datos de la compañía.

4. Cómo le gustaría recibir las solicitudes de sus vehículos

[Más detalles](#)

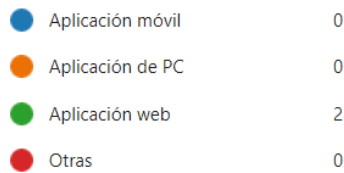


Figura C.9 Pregunta 4 Administrador.

Además, nos indica que todos los usuarios que la empresa poseen equipos para poder acceder a la información a través de Internet (ver Figura C.10).

5. ¿Cuenta la empresa con algún equipo tecnológico con acceso a internet?

[Más detalles](#)

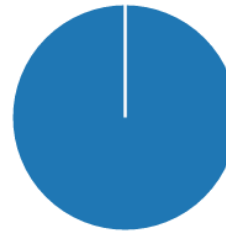


Figura C.10 Pregunta 5 Administrador.

De estas preguntas se obtuvieron los siguientes requisitos no funcionales:

Tabla C.1 Requisitos no funcionales de cada aplicación.

APP CLIENTE		
CÓDIGO	TÍTULO	DESCRIPCIÓN
RNFCL001	Autenticación	El ingreso al sistema estará restringido por un usuario y contraseña.
RNFCL002	Datos	Se usará una base de datos SQL Server para la persistencia de datos del sistema.
RNFCL003	Acceso desde móvil Android	El prototipo debe ser ejecutable en un teléfono móvil con sistema operativo Android.

APP CONDUCTOR		
CÓDIGO	TÍTULO	DESCRIPCIÓN
RNFCO001	Autenticación	El ingreso al sistema estará restringido por un usuario y contraseña.
RNFCO002	Datos	Se usará una base de datos SQL Server para la persistencia de datos del sistema.
RNFCO003	Acceso desde móvil Android	El prototipo debe ser ejecutable en un teléfono móvil con sistema operativo Android.
APP ADMINISTRADOR		
CÓDIGO	TÍTULO	DESCRIPCIÓN
RNFAD001	Autenticación	El ingreso al sistema estará restringido por un usuario y contraseña.
RNFAD002	Datos	Se usará una base de datos SQL Server para la persistencia de datos del sistema.
RNFAD003	Acceso desde móvil Android	El prototipo debe ser ejecutable en un teléfono móvil con sistema operativo Android.

Anexo D

HISTORIAS DE USUARIO

A continuación, se detallan todas las historias de usuario para cada aplicación.

APLICACIÓN CLIENTE

Tabla D.1 HUCL-01.

Nombre: Registro	
CÓDIGO: HUCL-01	Asignada: Xavier Garcia
Prioridad: Alta	
Descripción: Como usuario de la aplicación cliente requiero que cuente con un mecanismo para el registro de usuarios, así se evitarán accesos no autorizados a la información de la empresa.	
Criterios de aceptación: Como cliente requiero estar en capacidad de: <ul style="list-style-type: none">• Ingresar datos en formulario de registro.• Validación de usuario existente.	

Tabla D.2 HUCL-02.

Nombre: Autenticación	
CÓDIGO: HUCL-02	Asignada: Xavier Garcia
Prioridad: Alta	
Descripción: Como usuario de la aplicación cliente requiero que cuente con un mecanismo para el inicio de sesión, de esta manera se evitará accesos no autorizados.	
Criterios de aceptación: Como cliente requiero estar en capacidad de: <ul style="list-style-type: none">• Iniciar sesión al ingresar mi usuario y contraseña.• Cerrar sesión.	

Tabla D.3 HUCL-03.

Nombre: Actualización de datos	
CÓDIGO: HUCL-03	Asignada: Xavier Garcia
Prioridad: Media	
<p>Descripción:</p> <p>Como usuario de la aplicación cliente requiero que cuente con un mecanismo para actualización de datos personales, de esta manera poder mantener información actualizada de los usuarios.</p>	
<p>Criterios de aceptación:</p> <p>Como cliente requiero estar en capacidad de:</p> <ul style="list-style-type: none"> • Ver mi información actualizada. • Editar mi información personal. 	

Tabla D.4 HUCL-04.

Nombre: Solicitar Recorrido	
CÓDIGO: HUCL-04	Asignada: Xavier Garcia
Prioridad: Media	
<p>Descripción:</p> <p>Como usuario de la aplicación cliente requiero que cuente con un mecanismo para solicitar viajes a conductores, de esta manera se evitará que la parte administrativa se involucre en este procedimiento.</p>	
<p>Criterios de aceptación:</p> <p>Como cliente requiero estar en capacidad de:</p> <ul style="list-style-type: none"> • Marcar el punto de partida y destino de un recorrido. • Realizar una oferta de pago por el recorrido solicitado. 	

Tabla D.5 HUCL-05.

Nombre: Calificar Recorrido	
CÓDIGO: HUCL-05	Asignada: Xavier Garcia
Prioridad: Media	

<p>Descripción:</p> <p>Como usuario de la aplicación cliente requiero que cuente con un mecanismo para calificar el recorrido previamente realizado, de esta manera tener un estimado del nivel de servicio ofertado.</p>
<p>Criterios de aceptación:</p> <p>Como cliente requiero estar en capacidad de:</p> <ul style="list-style-type: none"> • Enviar una calificación de 1 a 5. • Enviar un comentario del servicio recibido.

Tabla D.6 HUCL-06.

Nombre: Mis ubicaciones	
CÓDIGO: HUCL-06	Asignada: Xavier Garcia
Prioridad: Baja	
<p>Descripción:</p> <p>Como usuario de la aplicación cliente requiero que cuente con un mecanismo para guardar ubicaciones, de esta manera tener un registro personal de las ubicaciones que requiera.</p>	
<p>Criterios de aceptación:</p> <p>Como cliente requiero estar en capacidad de:</p> <ul style="list-style-type: none"> • Agregar ubicaciones. • Ver mis ubicaciones guardadas. 	

APLICACIÓN CONDUCTOR

Tabla D.7 HUCO-01.

Nombre: Registro	
CÓDIGO: HUCO-01	Asignada: Xavier Garcia
Prioridad: Alta	
<p>Descripción:</p> <p>Como usuario de la aplicación CONDUCTOR requiero que cuente con un mecanismo para el</p>	

registro de usuarios, así se evitarán accesos no autorizados a la información de la empresa.

Criterios de aceptación:

Como CONDUCTOR requiero estar en capacidad de:

- Ingresar datos en Formulario de registro.
- Validación de usuario existente.

Tabla D.8 HUCO-02.

Nombre: Autenticación	
CÓDIGO: HUCO-02	Asignada: Xavier Garcia
Prioridad: Alta	
Descripción: Como usuario de la aplicación CONDUCTOR requiero que cuente con un mecanismo para el inicio de sesión, de esta manera se evitará accesos no autorizados.	
Criterios de aceptación: Como CONDUCTOR requiero estar en capacidad de:	
<ul style="list-style-type: none">• Iniciar sesión al ingresar mi usuario y contraseña.• Cerrar sesión.	

Tabla D.8 HUCO-03.

Nombre: Actualización de datos	
CÓDIGO: HUCO-03	Asignada: Xavier Garcia
Prioridad: Media	
Descripción: Como usuario de la aplicación CONDUCTOR requiero que cuente con un mecanismo para actualización de datos personales, de esta manera poder mantener información actualizada de los usuarios.	
Criterios de aceptación: Como CONDUCTOR requiero estar en capacidad de:	
<ul style="list-style-type: none">• Ver mi información actualizada.	

- Editar mi información personal.

Tabla D.9 HUCO-04.

Nombre: Receptar Recorridos	
CÓDIGO: HUCO-04	Asignada: Xavier Garcia
Prioridad: Media	
<p>Descripción:</p> <p>Como usuario de la aplicación CONDUCTOR requiero que cuente con un mecanismo para receptar solicitudes viajes de clientes, de esta manera se evitará que la parte administrativa se involucre en este procedimiento.</p>	
<p>Criterios de aceptación:</p> <p>Como CONDUCTOR requiero estar en capacidad de:</p> <ul style="list-style-type: none"> • Recibir notificaciones de las solicitudes de clientes. • Realizar una contra oferta de pago por el recorrido. 	

Tabla D.10 HUCO-05.

Nombre: Vehículos	
CÓDIGO: HUCO-05	Asignada: Xavier Garcia
Prioridad: Media	
<p>Descripción:</p> <p>Como usuario de la aplicación CONDUCTOR requiero que cuente con un mecanismo para visualizar un vehículo, de esta manera tener una administración global de los vehículos de la empresa.</p>	
<p>Criterios de aceptación:</p> <p>Como CONDUCTOR requiero estar en capacidad de:</p> <ul style="list-style-type: none"> • Permite visualizar los vehículos. • Permite el ingreso de vehículos. 	

Tabla D.11 HUCO-06.

Nombre: Seleccionar vehículo	
CÓDIGO: HUCO-06	Asignada: Xavier Garcia
Prioridad: Baja	
Descripción: Como usuario de la aplicación CONDUCTOR requiero que cuente con un mecanismo para seleccionar un vehículo, de esta manera poder atender una solicitud.	
Criterios de aceptación: Como CONDUCTOR requiero estar en capacidad de: <ul style="list-style-type: none">• La recepción se habilita tras la selección de un vehículo.• Se puede seleccionar un vehículo previamente ingresado.	

APLICACIÓN ADMINISTRADOR

Tabla D.12 HUAD-01.

Nombre: Registro	
CÓDIGO: HUAD-01	Asignada: Xavier Garcia
Prioridad: Alta	
Descripción: Como usuario de la aplicación ADMINISTRADOR requiero que cuente con un mecanismo para el registro de usuarios, así se evitará accesos no autorizados a la información de la empresa.	
Criterios de aceptación: Como ADMINISTRADOR requiero estar en capacidad de: <ul style="list-style-type: none">• Ingresar datos en Formulario de registro.• Validación de usuario existente.	

Tabla D.13 HUAD-02.

Nombre: Autenticación	
CÓDIGO: HUAD-02	Asignada: Xavier Garcia
Prioridad: Alta	

<p>Descripción:</p> <p>Como usuario de la aplicación ADMINISTRADOR requiero que cuente con un mecanismo para el inicio de sesión, de esta manera se evitará accesos no autorizados.</p>
<p>Criterios de aceptación:</p> <p>Como ADMINISTRADOR requiero estar en capacidad de:</p> <ul style="list-style-type: none"> • Iniciar sesión al ingresar mi usuario y contraseña. • Cerrar sesión.

Tabla D.14 HUAD-03.

Nombre: Actualización de datos	
CÓDIGO: HUAD-03	Asignada: Xavier Garcia
Prioridad: Media	
<p>Descripción:</p> <p>Como usuario de la aplicación ADMINISTRADOR requiero que cuente con un mecanismo para actualización de datos personales, de esta manera poder mantener información actualizada de los usuarios.</p>	
<p>Criterios de aceptación:</p> <p>Como ADMINISTRADOR requiero estar en capacidad de:</p> <ul style="list-style-type: none"> • Ver mi información actualizada. • Editar mi información personal. 	

Tabla D.15 HUAD-04.

Nombre: Información Vehículos	
CÓDIGO: HUAD-04	Asignada: Xavier Garcia
Prioridad: Media	
<p>Descripción:</p> <p>Como usuario de la aplicación ADMINISTRADOR requiero que cuente con un mecanismo para receptar solicitudes viajes de clientes, de esta manera se evitará que la parte administrativa se involucre en este procedimiento.</p>	
<p>Criterios de aceptación:</p>	

Como ADMINISTRADOR requiero estar en capacidad de:

- Recibir notificaciones de las solicitudes de clientes.
- Realizar una contra oferta de pago por el recorrido.

Tabla D.16 HUAD-05.

Nombre: Información Recorridos	
CÓDIGO: HUAD-05	Asignada: Xavier Garcia
Prioridad: Media	
Descripción: Como usuario de la aplicación ADMINISTRADOR requiero que cuente con un mecanismo para visualizar la información de los conductores, de esta manera poder tener un registro de los recorridos de la empresa.	
Criterios de aceptación: Como ADMINISTRADOR requiero estar en capacidad de: <ul style="list-style-type: none">• Permite visualizar los recorridos.• Permite editar los recorridos.	

Tabla D.17 HUAD-06.

Nombre: Información Clientes	
CÓDIGO: HUAD-06	Asignada: Xavier Garcia
Prioridad: Baja	
Descripción: Como usuario de la aplicación ADMINISTRADOR requiero que cuente con un mecanismo para visualizar la información de los conductores, de esta manera poder tener un registro de los clientes de la empresa.	
Criterios de aceptación: Como ADMINISTRADOR requiero estar en capacidad de: <ul style="list-style-type: none">• Ver los clientes registrados.• Editar información de los clientes.	

Tabla D.18 HUAD-07.

Nombre: Información Conductores	
CÓDIGO: HUAD-07	Asignada: Xavier Garcia
Prioridad: Baja	
Descripción: Como usuario de la aplicación ADMINISTRADOR requiero que cuente con un mecanismo para visualizar la información de los conductores, de esta manera tener un registro de los conductores de la empresa.	
Criterios de aceptación: Como ADMINISTRADOR requiero estar en capacidad de: <ul style="list-style-type: none">• Ver los conductores registrados.• Editar la información de los conductores.	

Anexo E

DIAGRAMAS DE CASOS DE USO

Se ha realizado diagramas de casos de uso, a partir de las historias de usuario antes descritas. De este modo, se puede identificar claramente las funciones que debe proveer el sistema a los diferentes usuarios.

APLICACIÓN CLIENTE

Inicialmente la aplicación contará con un módulo de registro ver Figura E.1, en el cual se solicitarán todos los datos que la empresa ha considerado necesarios para registrar un cliente en su sistema.

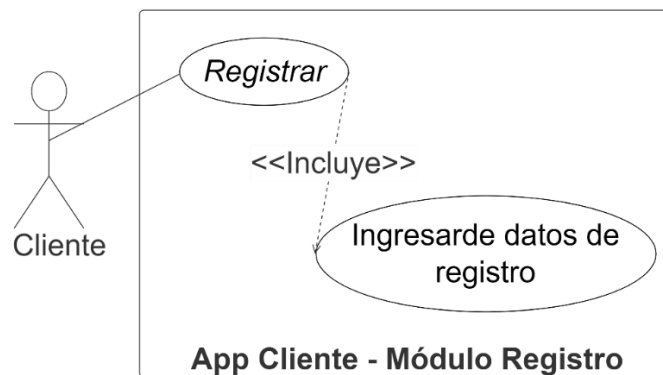


Figura E.1 Módulo de registro Cliente.

El módulo de autenticación permita restablecer a contraseña en caso de olvidarla, se recuperará mediante correo electrónico.

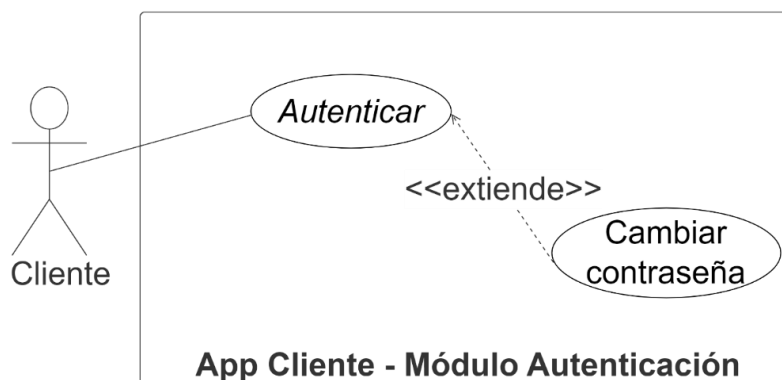


Figura E.2 Módulo autenticación Cliente.

El módulo de solicitud de recorrido permitirá indicar el punto de partida y el destino de la solicitud, una vez finalizado el recorrido existirá la posibilidad de calificar el recorrido.

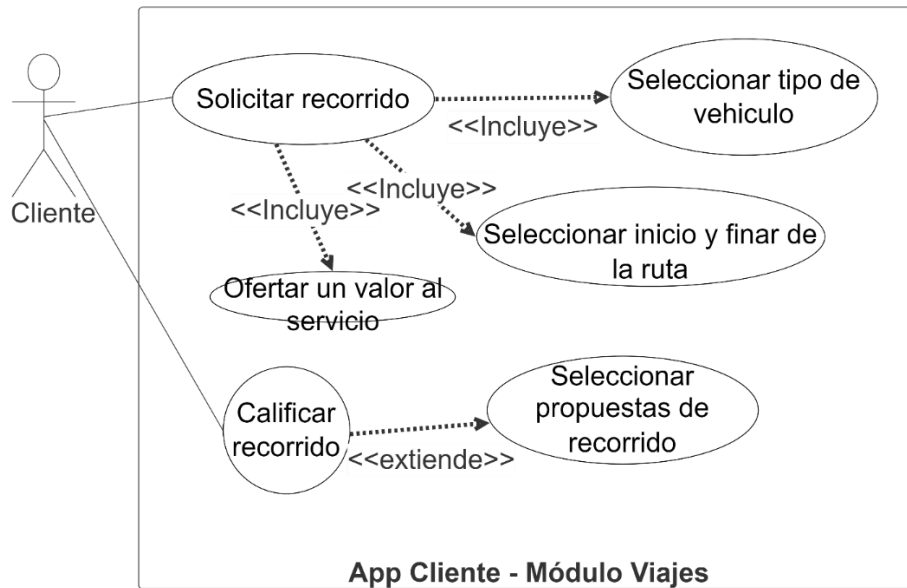


Figura E.3 Módulo Solicitud Recorrido Cliente.

El cliente tendrá la posibilidad de guardar ubicaciones en el sistema.

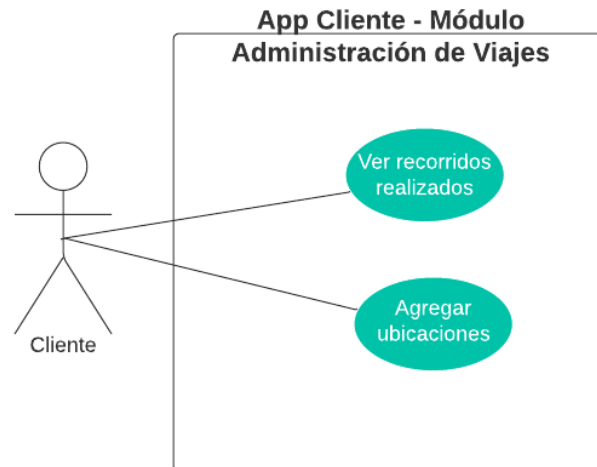


Figura E.4 Módulo Mis ubicaciones.

El usuario tendrá la posibilidad de cambiar su información en el sistema.

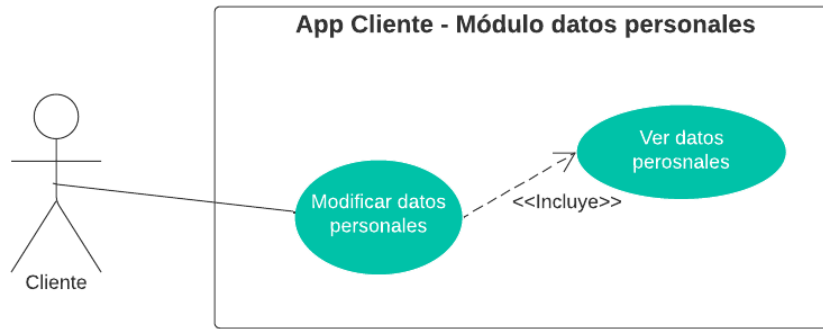


Figura E.5 Módulo actualización de datos.

APLICACIÓN CONDUCTOR

Inicialmente la aplicación contará con un módulo de registro (ver Figura E.6), en el cual se solicitarán todos los datos que la empresa ha considerado necesarios para registrar un conductor en su sistema.

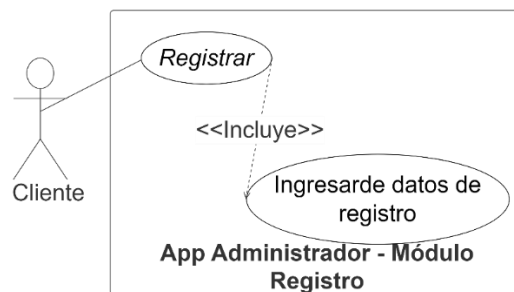


Figura E.6 Módulo Registro Conductor.

El módulo de autenticación permita restablecer a contraseña en caso de olvidarla, se recuperará mediante correo electrónico. No se permitirá el inicio de sesión hasta que se habilite el usuario conductor.

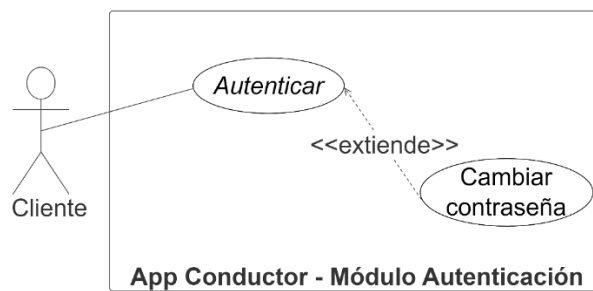


Figura E.7 Módulo autenticación Conductor.

El usuario podrá escuchar solicitudes del cliente mediante su aplicación, también podrá realizar una contra oferta.

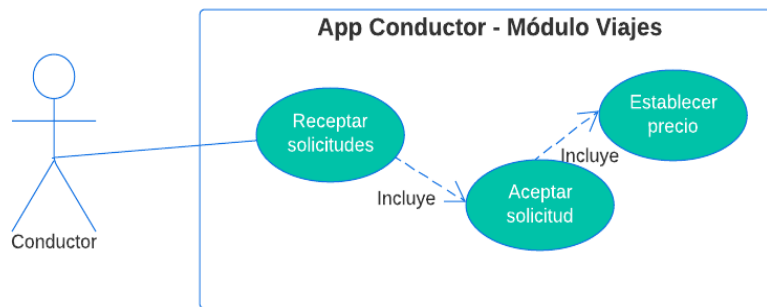


Figura E.8 Módulo recepción de Recorridos Conductor.

El módulo de viajes permitirá ver los recorridos realizados y ver los vehículos de la compañía.

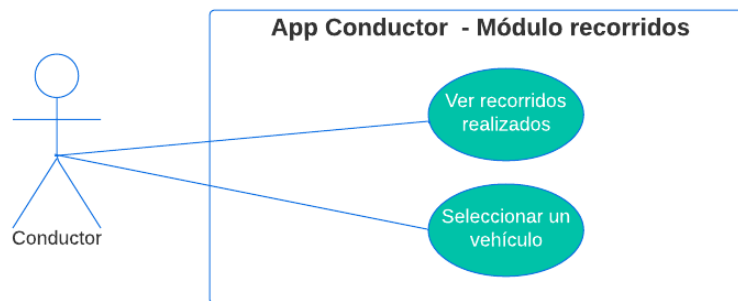


Figura E.9 Módulo Vehículos Conductor.

El usuario tendrá la posibilidad de cambiar su información en el sistema.

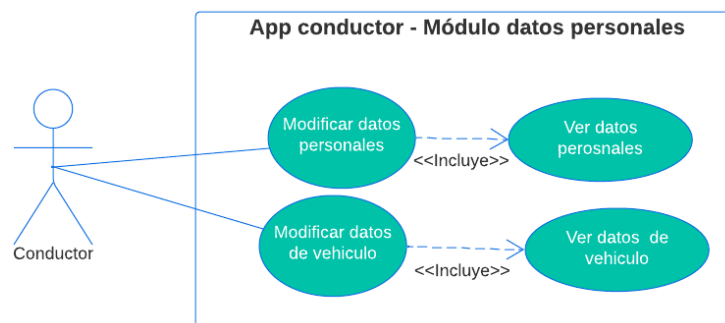


Figura E.10 Módulo actualización de datos Conductor.

APLICACIÓN ADMINISTRADOR

Inicialmente la aplicación contará con un módulo de registro (ver Figura E.13) en el cual se solicitarán todos los datos que la empresa ha considerado necesarios para registrar un administrador en su sistema.



Figura E.11 Módulo Registro Administrador.

El módulo de autenticación permita restablecer a contraseña en caso de olvidarla, se recuperará mediante correo electrónico. No se permitirá el inicio de sesión hasta que se habilite el usuario conductor.

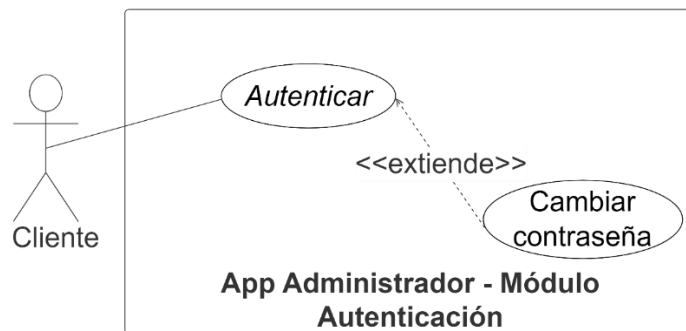


Figura E.12 Módulo Autenticación.

El usuario tendrá la posibilidad de ver todos los recorridos solicitados por los clientes.



Figura E.13 Módulos Recorridos.

El módulo de administración permitirá realizar un CRUD de los clientes, conductores y vehículos en el sistema.

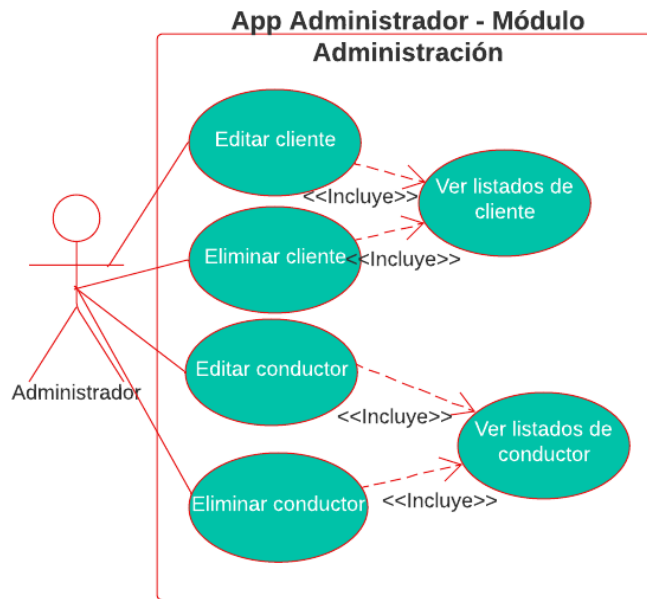


Figura E.14 Módulo Clientes y Conductores.

El usuario tendrá la posibilidad de cambiar su información en el sistema.

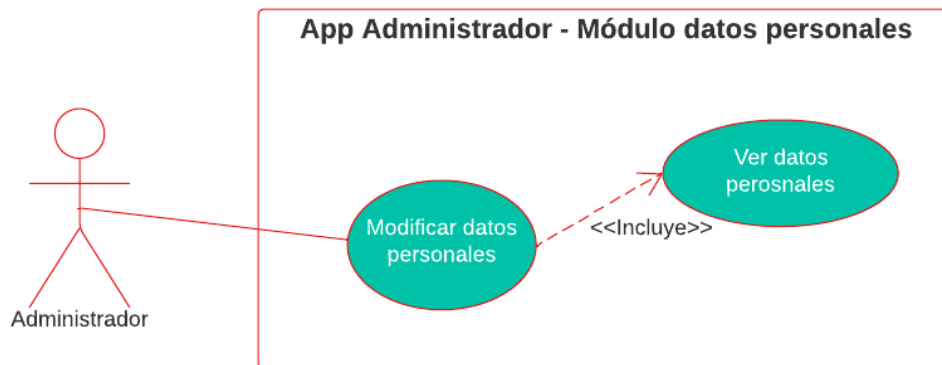


Figura E.15 Módulo Actualización de datos.

Anexo F

CASOS DE USO

Tabla F.1 CU001.

Código CU	CU001	NOMBRE	Registro
CU Relacionados	RF001		
Código Requisito			
Actor/es Participantes	Cliente, Conductor, Administrador		
Descripción CU	Permite a un usuario registrarse en el sistema. Para ello existe un botón que despliega un formulario de registro.		
Precondiciones	Haber ingresado a la aplicación. No haber sido registrado anteriormente.		
Secuencia de pasos	PASO	Descripción	
	1	El usuario presiona el botón de registro.	
	2	La aplicación muestra un formulario de registro.	
	3	El usuario debe ingresar todos los datos solicitados por la aplicación como cédula, teléfono, correo electrónico y fecha de nacimiento.	
	4	El usuario presiona el botón de guardar registro.	
	5	La aplicación valida los datos de cédula, teléfono y correo electrónico.	
	5.1	Si los datos son correctos. La aplicación crea un nuevo usuario y mediante la comunicación con el servicio lo guarda.	
	5.1.1	La aplicación vuelve a mostrar el formulario de autenticación	
	5.2	Si el registro es erróneo se muestra un mensaje de error.	
Postcondiciones	Se crea un nuevo registro en la tabla correspondiente al usuario.		

Tabla F.2 CU002.

Código CU	CU002	NOMBRE	Autenticación
CU Relacionados	CU001		
Código Requisito	RF002		
Actor/es Participantes	Cliente, conductor, administrador.		
Descripción CU	Permite al usuario autenticarse con un usuario y contraseña.		
Precondiciones	Iniciar la aplicación. Haber realizado el proceso de autenticación.		
Secuencia de pasos	PASO	Descripción	
	1	Ingresar el usuario y contraseña solicitadora en el formulario	
	2	Presionar el botón iniciar sesión.	
	2.1	Si la autenticación es exitosa se muestra la página principal de la aplicación.	
	2.2	Si la autenticación es errónea. Se muestra un mensaje de error.	
Postcondiciones	Se muestra una página principal y un menú de opciones.		

Código CU	CU003	NOMBRE	Actualización de datos personales
CU Relacionados	CU001, CU002		
Código Requisito	RF003		
Actor/es Participantes	Cliente, conductor, administrador		
Descripción CU	Permite a usuarios previamente registrados actualizar la información personal.		
Precondiciones	Usuario Previamente registrado. Presionar la opción de perfil dentro del menú de aplicación.		
Secuencia de pasos	PASO	Descripción	
	1	El usuario presiona el botón "Mi Perfil"	
	2	El usuario presiona el botón "Editar"	
	3	El usuario ingresa la información que desea cambiar.	
	4	El usuario presiona el botón "GUARDAR".	
	5	La aplicación valida la información ingresada.	
	5.1	Si la información es correcta se modifica mediante el servicio web.	
	5.2	Si la información es incorrecta. Se muestra un mensaje de error.	
	Postcondiciones	La aplicación mostrará el mismo formulario una vez realizada la actividad.	

Tabla F.3 CU003.

Código CU	CU004	NOMBRE	Solicitar Recorrido
CU Relacionados	CU001		
Código Requisito	RF004		
Actor/es Participantes	Cliente		
Descripción CU	Permite realizar el proceso de solicitud e inicio de un recorrido por parte de un cliente.		
Precondiciones	Sesión iniciada en la aplicación.		
Secuencia de pasos	PASO	Descripción	
	1	Ingresar el punto de partida, destino del recorrido y oferta para un recorrido.	
	2	La aplicación habilita el botón "Solicitar" cuando los datos son completados.	
	3	La aplicación registra el recorrido con estado "solicitado".	
	3.1	Si la solicita es respondida. La aplicación muestra las respuestas.	
	3.1.1	El usuario selecciona una respuesta a su solicitud	
	3.1.2	La aplicación inicia el recorrido y cambia el estado del recorrido.	
	3.1.3	La aplicación muestra la ubicación del conductor y notifica cuando llegue al punto de partida	
	3.1.4	La aplicación muestra la ubicación del conductor y notifica cuando llegue al punto de destino	
	3.1.5	La aplicación muestra un formulario para calificar la aplicación	
	3.1.5.1	El usuario puede presionar uno de los emojis con caras de satisfacción	

		o desaprobación de servicio. Además, debe enviar un comentario.
	3.1.5.2	El usuario puede omitir el enviar una calificación
	3.2	Si la solicitud no es aceptada la aplicación regresa a la vista principal.
Postcondiciones	La aplicación regresa a la vista principal.	

Tabla F.4 CU004.

Código CU	CU005	NOMBRE	Calificar servicio
CU Relacionados	CU001		
Código Requisito	RF005		
Actor/es Participantes	Cliente		
Descripción CU	Permite calificar un recorrido realizado previamente. Se utiliza una calificación que va desde 1 hasta 5 utilizando caras tristes (calificaciones bajas) y sonrientes (calificaciones altas).		
Precondiciones	Iniciar sesión en la aplicación. Haber concluido una solicitud de recorrido.		
Secuencia de pasos	PASO	Descripción	
	1	La aplicación muestra un conjunto de 5 caras diferentes que indican una calificación diferente. Además un cuadro de texto para ingresar un comentario del servicio.	
	1.1	Si el usuario selecciona una de las caras de calificación, se habilita el botón "Enviar"	
	1.1.1	Si el usuario presiona enviar. La aplicación guardará la calificación y comentario del recorrido.	
	1.2	Si el usuario no presiona ninguna cara, únicamente tendrá habilitado el botón de "OMITIR"	
Postcondiciones	La aplicación regresa a la vista principal.		

Tabla F.5 CU005.

Código CU	CU006	NOMBRE	Mis ubicaciones
CU Relacionados	CU001		
Código Requisito	RF006		
Actor/es Participantes	Cliente		
Descripción CU	Permite visualizar tu guardar direcciones a los clientes en su aplicación.		
Precondiciones	Sesión iniciada en la aplicación. Opción presionada la opción "Mis recorridos" en el menú de la aplicación.		
Secuencia de pasos	PASO	Descripción	
	1	El usuario presiona el botón "Mis lugares"	
	2	La aplicación tiene los lugares registradores por el usuario mediante el servicio web y los muestra en una lista.	
	3	El usuario puede presionar el botón "Agregar" o seleccionar un lugar	

		de la lista.
	3.1	Si el usuario presiona agregar. Se muestra una nueva vista con un mapa para realizar la búsqueda de la ubicación.
	3.1.1	El usuario selecciona el botón "Guardar" para registrar su nueva ubicación.
	3.2	Si el usuario presiona una ubicación. Se muestra una nueva vista con un mapa y la información de la ubicación. Además de un botón "Eliminar".
	3.2.1	Si el usuario presiona "Eliminar". La aplicación borra el registro mediante el servicio web.
Postcondiciones	La aplicación regresa a la vista anterior.	

Tabla F.6 CU006.

Código CU	CU007	NOMBRE	Mis Recorridos	
CU Relacionados	CU001			
Código Requisito	RF007			
Actor/es Participantes	Cliente, conductor			
Descripción CU	Permite visualizar la información de los recorridos realizados.			
Precondiciones	Sesión iniciada en la aplicación.			
Secuencia de pasos	PASO	Descripción		
	1	El usuario presiona la opción "Mis recorridos" en el menú de la aplicación.		
	2	La aplicación obtiene los recorridos realizados por el usuario mediante el servicio web y la muestra mediante una lista.		
	3	El usuario puede seleccionar un recorrido presionando sobre el mismo.		
	4	La aplicación muestra la información del recorrido de forma detallada en un nuevo formulario.		
Postcondiciones	La aplicación se mantiene en la vista del recorrido hasta presionar el botón de retroceder.			

Anexo G

DIAGRAMAS DE ACTIVIDADES

Se ha realizado diagramas de actividades de todos los requisitos funcionales de las distintas aplicaciones, esto con el objetivo de describir el algoritmo que se deberá implementar más adelante.

APLICACIÓN CLIENTE

Un cliente solo puede ingresar a la aplicación una vez que cuente con un usuario y contraseña. Con su usuario y contraseña previamente creados se realizará un inicio de sesión exitoso, tal como se indica en la Figura G.1.

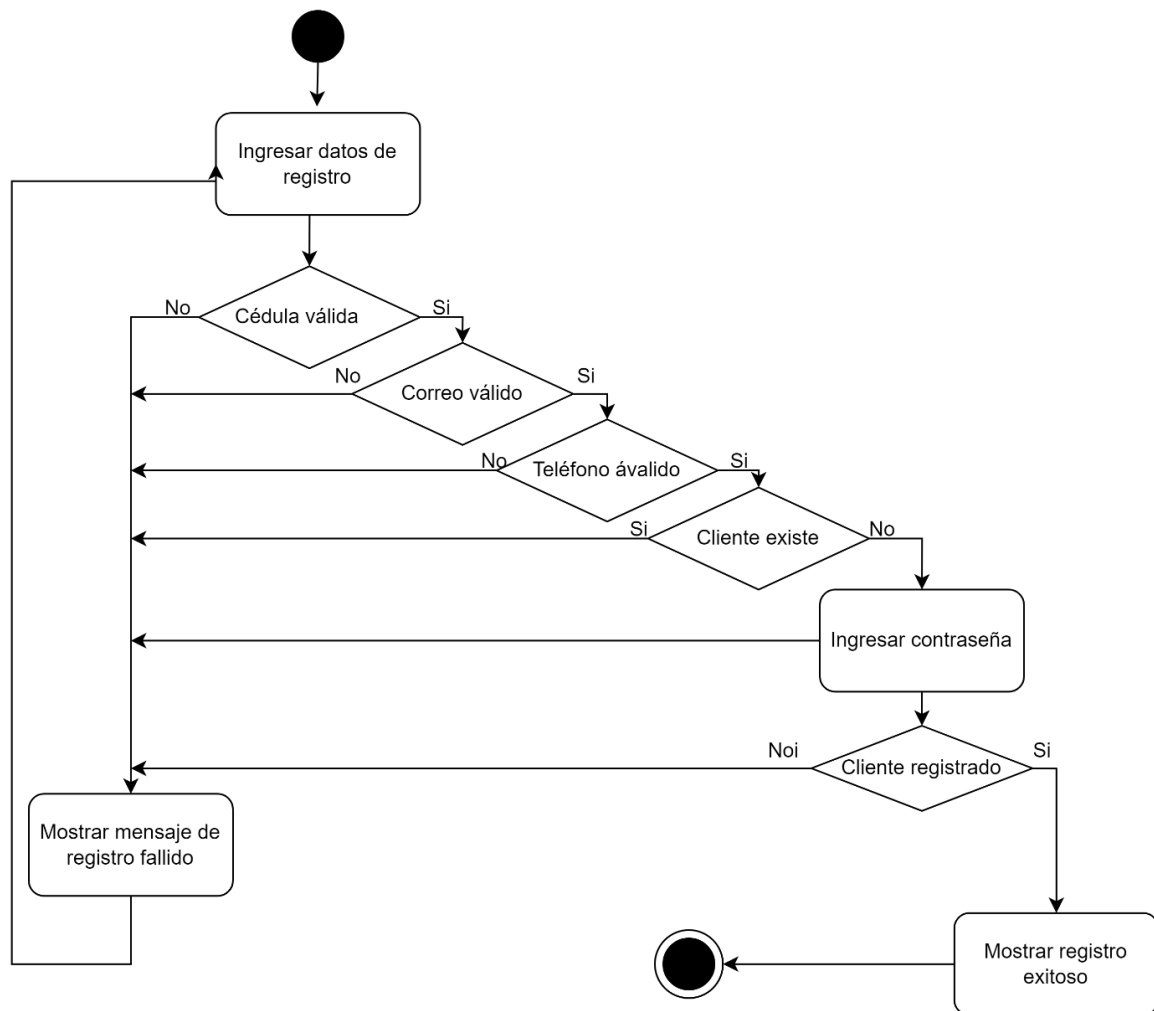


Figura G.1 Registro de usuario Cliente.

Al momento de solicitar un viaje se solicitará una ruta, el precio del viaje y se escuchará las respuestas de los conductores. Como se indica en la Figura G.2.

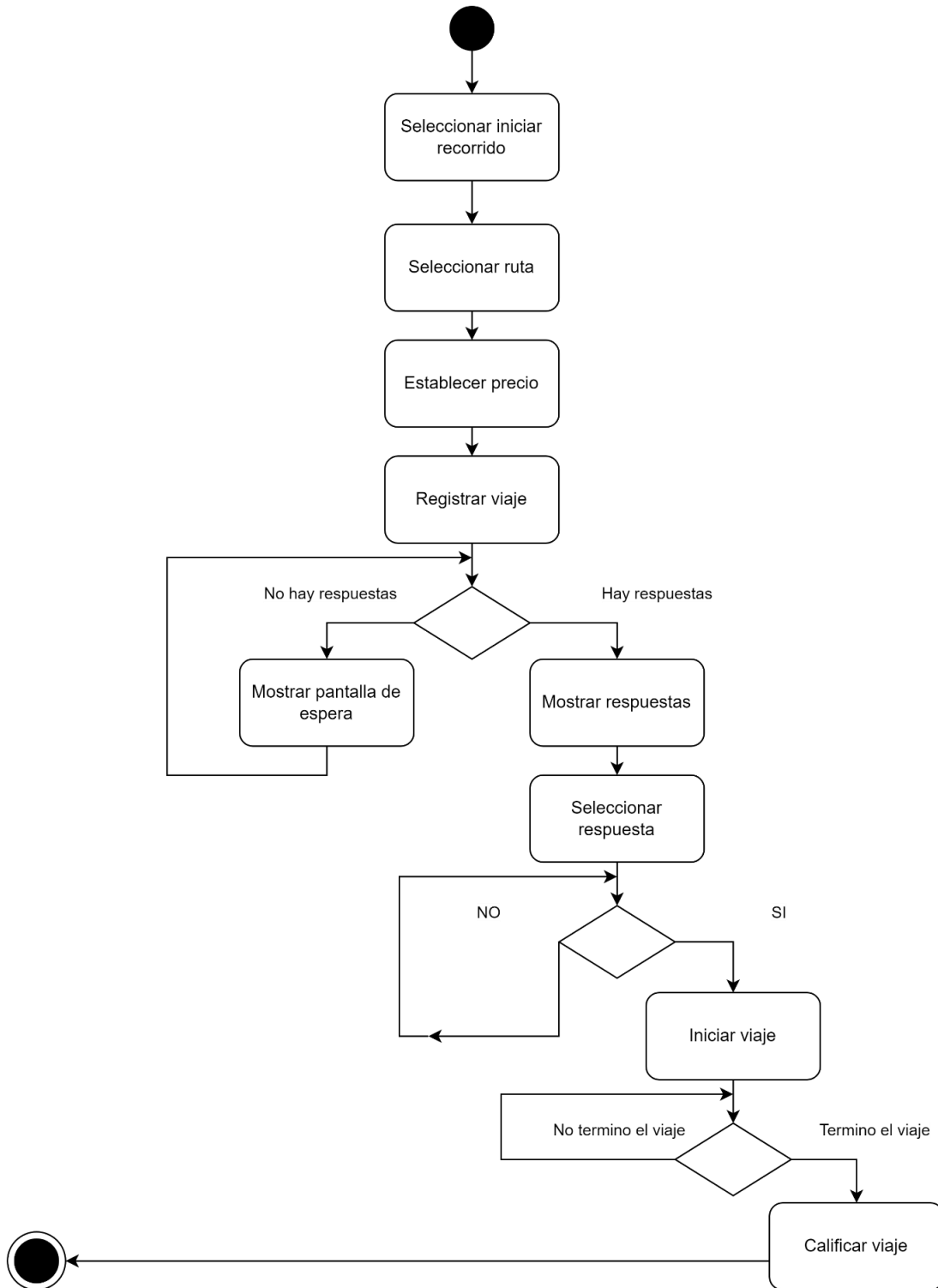


Figura G.2 Solicitud de Recorrido Cliente.

Con los datos creados en el módulo de registro se realizará la autenticación. El usuario una vez iniciado la sesión tendrá la posibilidad de editar su información de perfil.

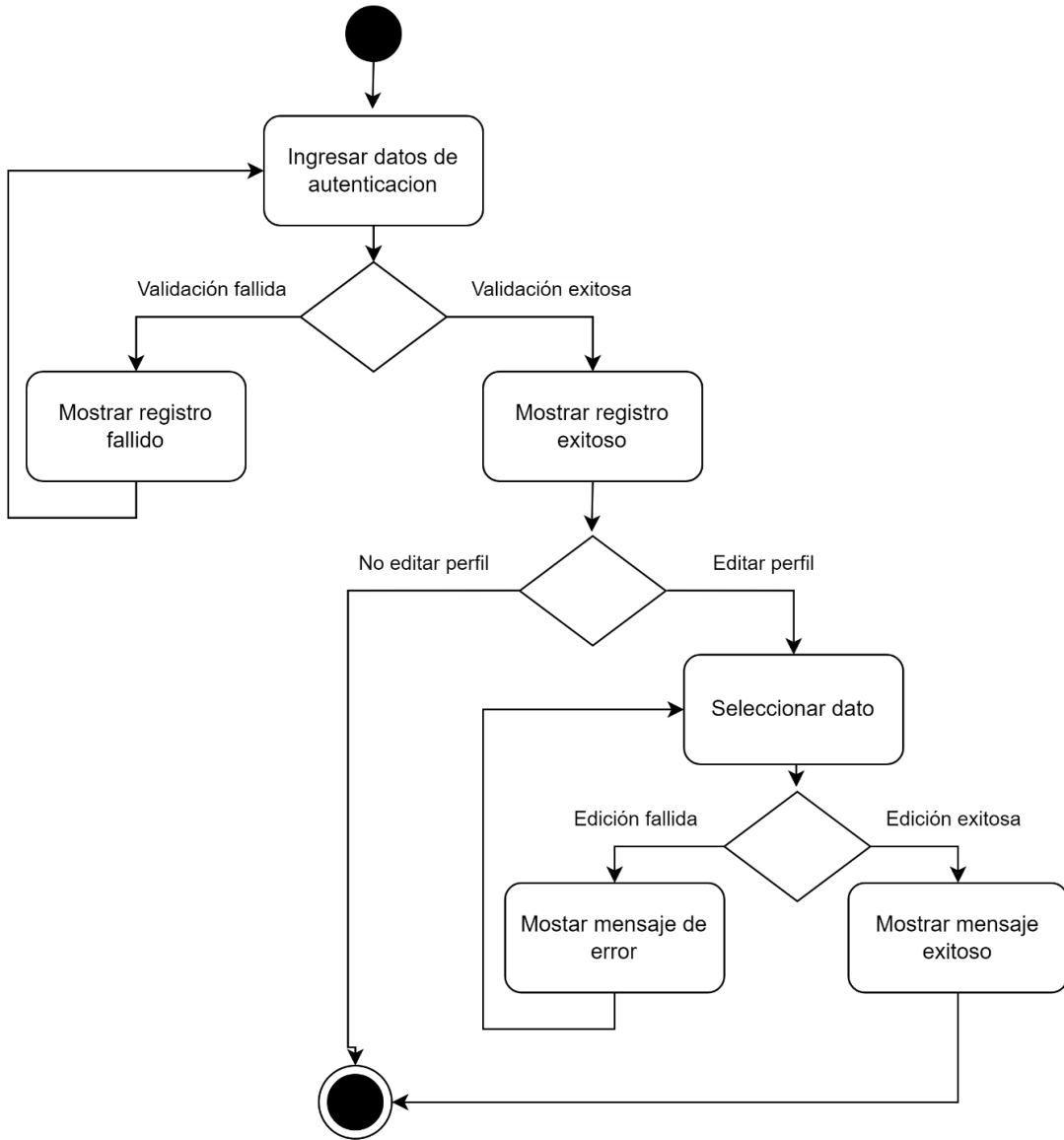


Figura G.3 Autenticación y Actualización de datos.

APLICACIÓN CONDUCTOR

Un conductor solo puede ingresar a la aplicación una vez que cuente con un usuario y contraseña. Con su usuario y contraseña previamente creados se realizará un logon exitoso como se indica en la Figura G.4.

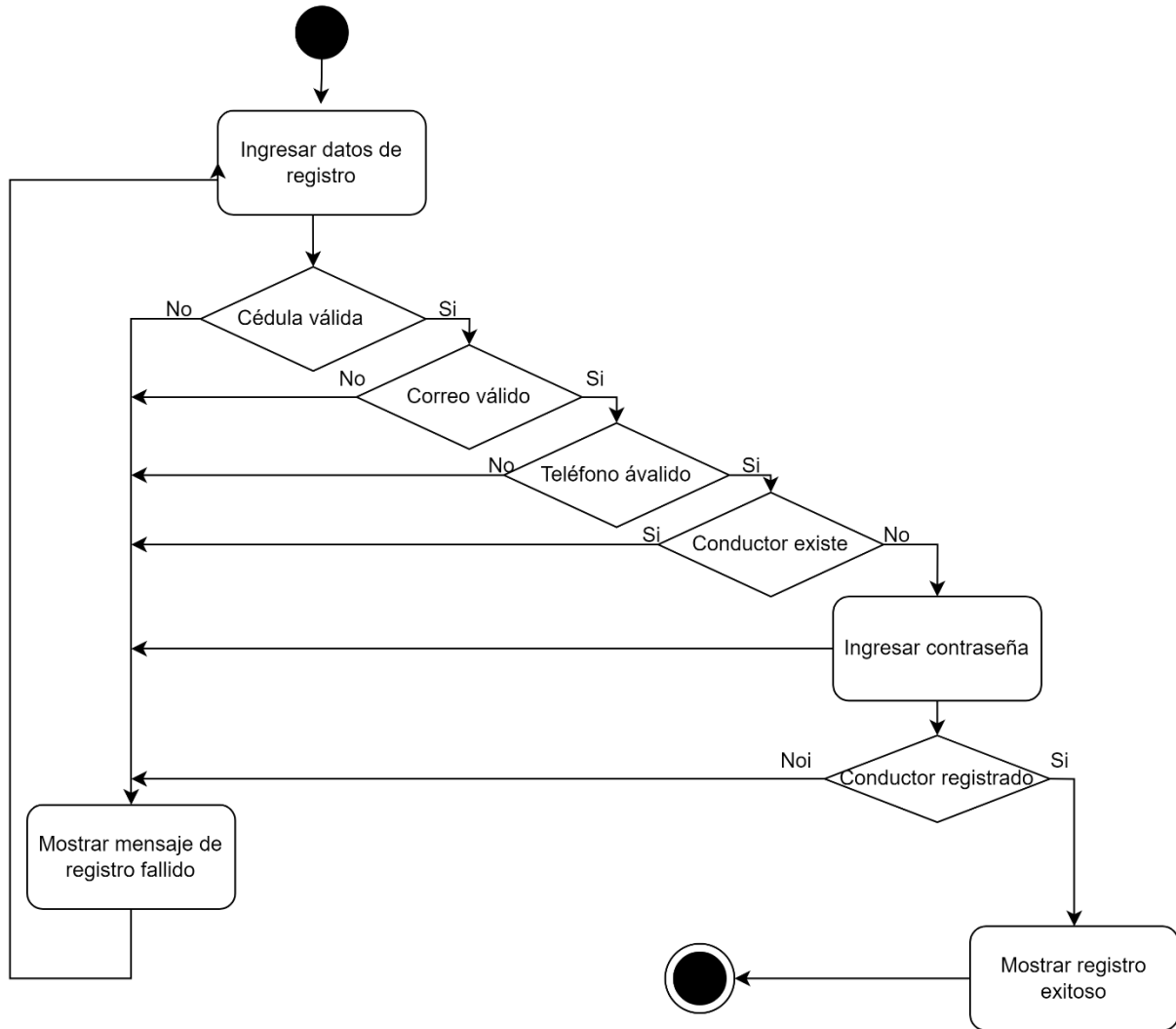


Figura G.4 Registro de usuario Conductor.

Al momento de recibir un viaje se podrá seleccionar la solicitud y realizar una contra oferta. Como se indica en la Figura G.5.

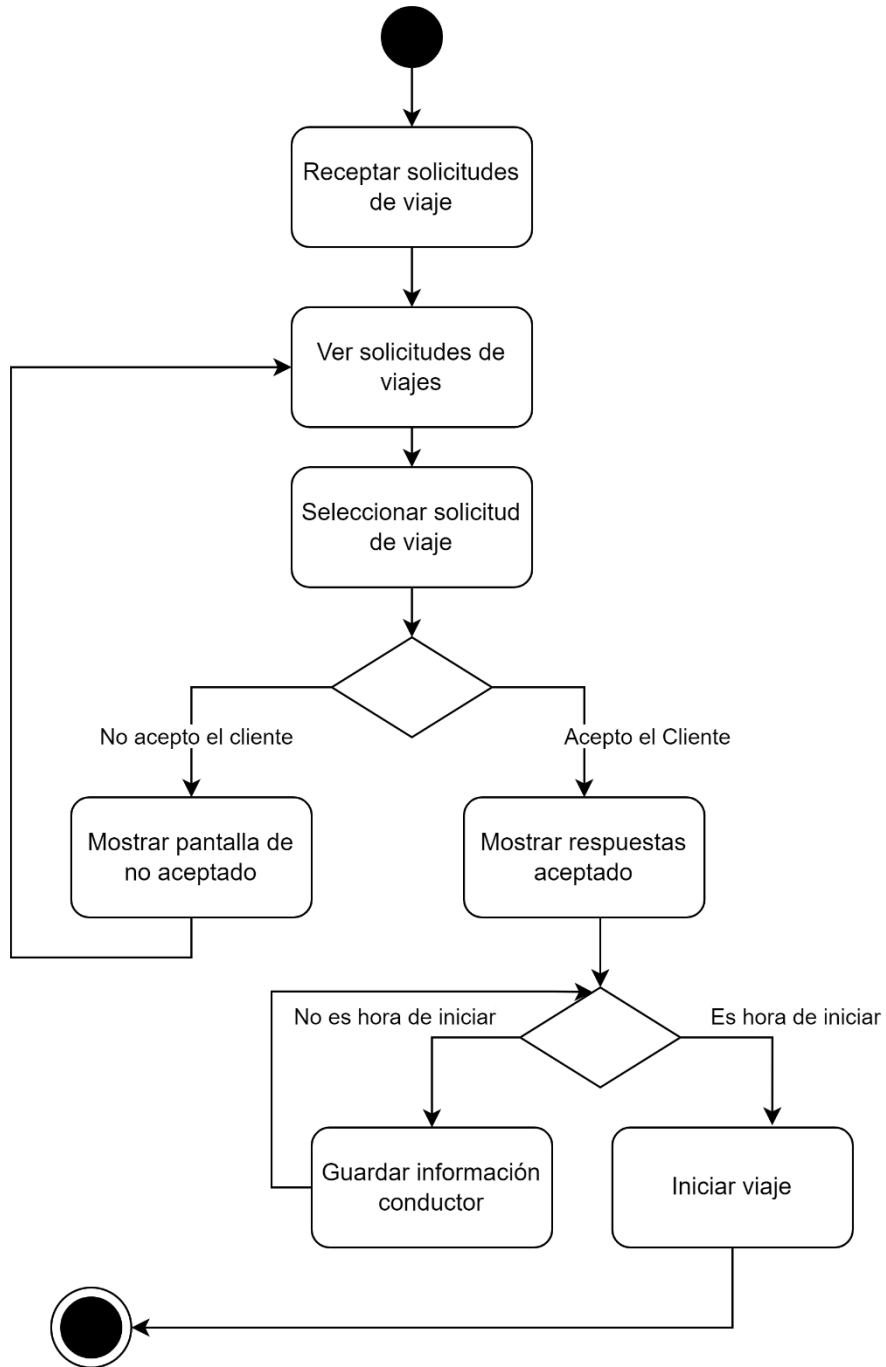


Figura G.5 Receptar Recorridos Conductor.

Con los datos creados en el módulo de registro se realizará la autenticación. El usuario una vez iniciado la sesión tendrá la posibilidad de editar su información de perfil.

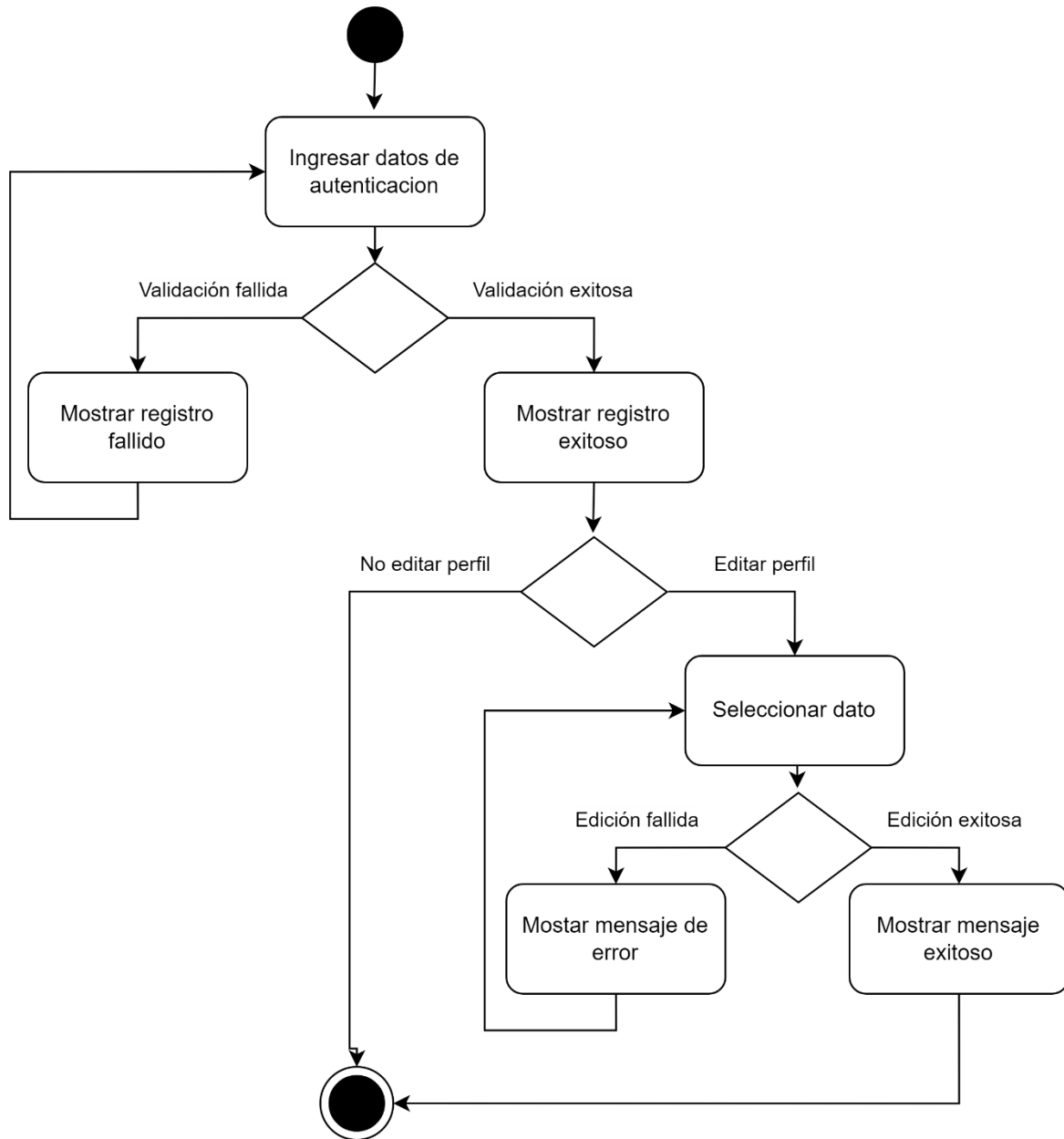


Figura G.6 Autenticación y Actualización de datos Conductor.

APLICACIÓN ADMINISTRADOR

Un conductor solo puede ingresar a la aplicación una vez que cuente con un usuario y contraseña. Con su usuario y contraseña previamente creados se realizará un inicio de sesión exitoso como se indica en la Figura G.7.

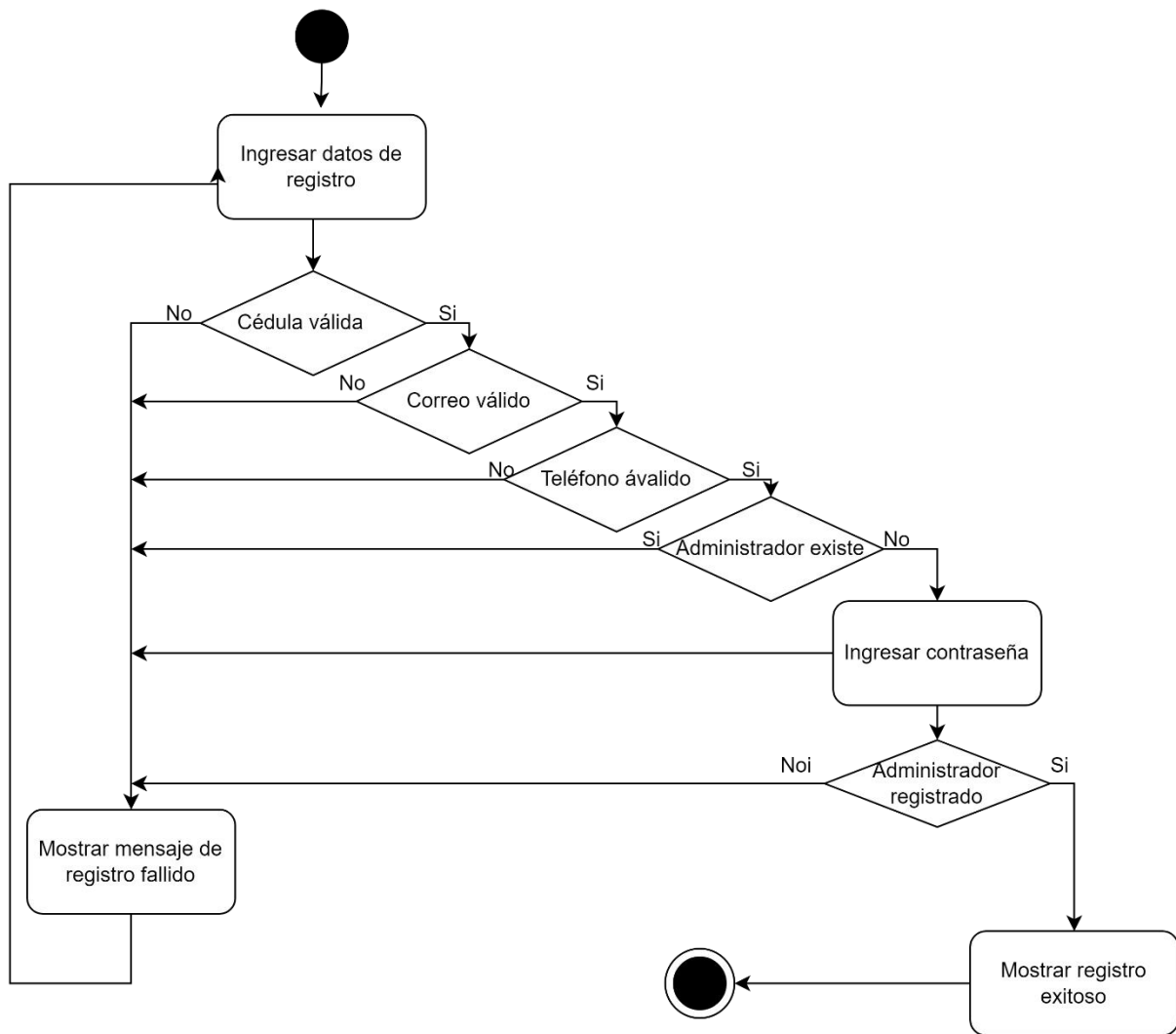


Figura G.7 Registro de usuario Administrador.

El módulo le permitirá realizar un CRUD de los Clientes de la compañía.

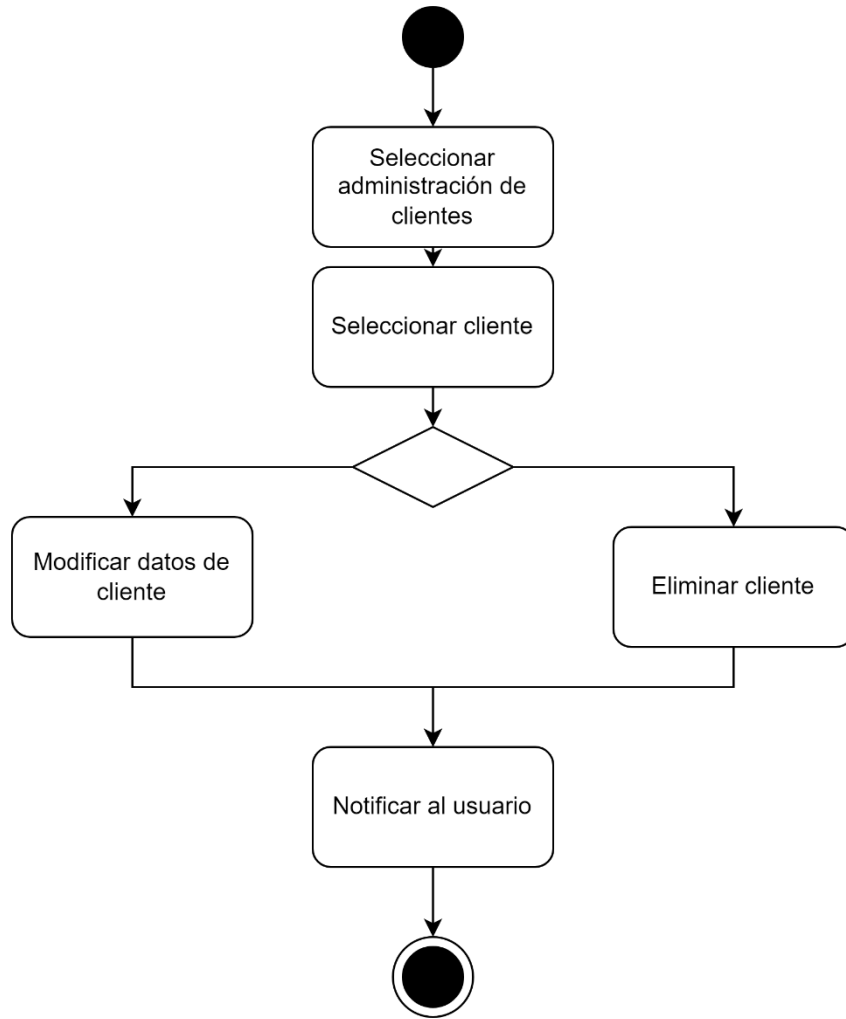


Figura G.8 Información Clientes Administrador.

El módulo le permitirá realizar un CRUD de los conductores de la compañía.

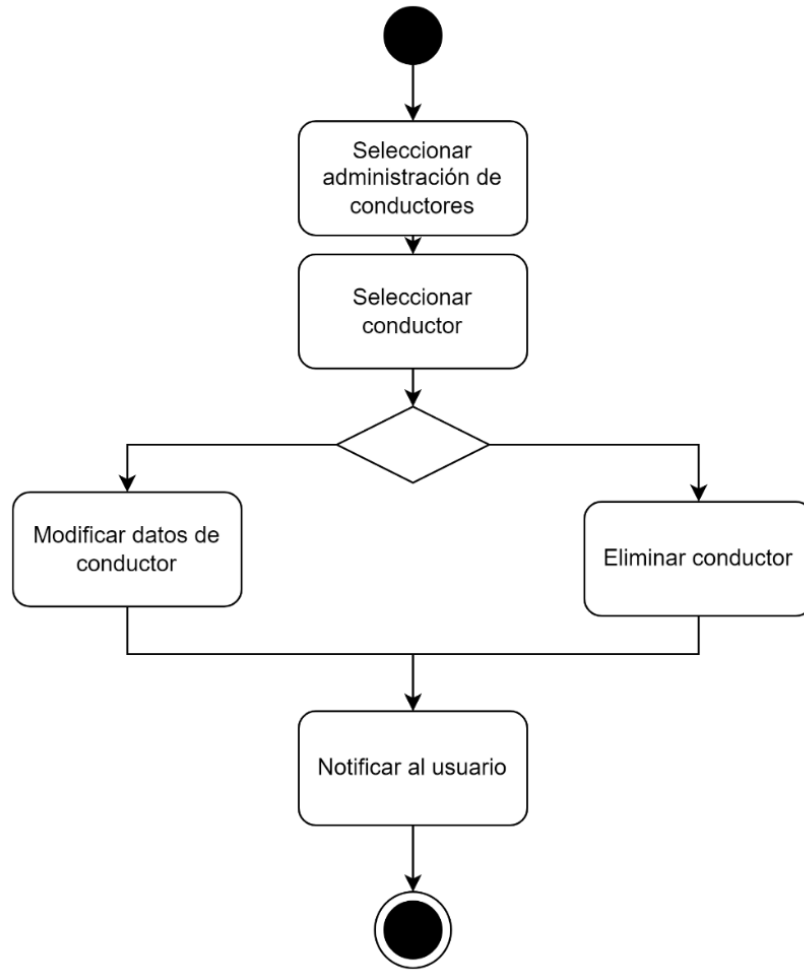


Figura G.9 Información Conductores Administrador.

El módulo le permitirá realizar un CRUD de los Vehículos de la compañía.

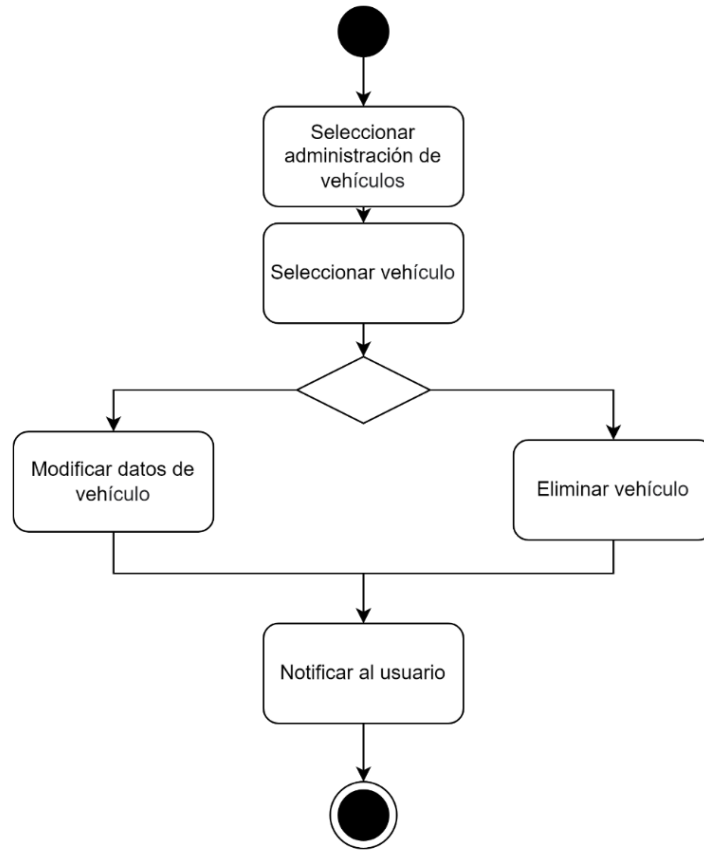


Figura G.10 Información Vehículos Administrador.

Con los datos creados en el módulo de registro se realizará la autenticación. El usuario una vez iniciado la sesión tendrá la posibilidad de editar su información de perfil.

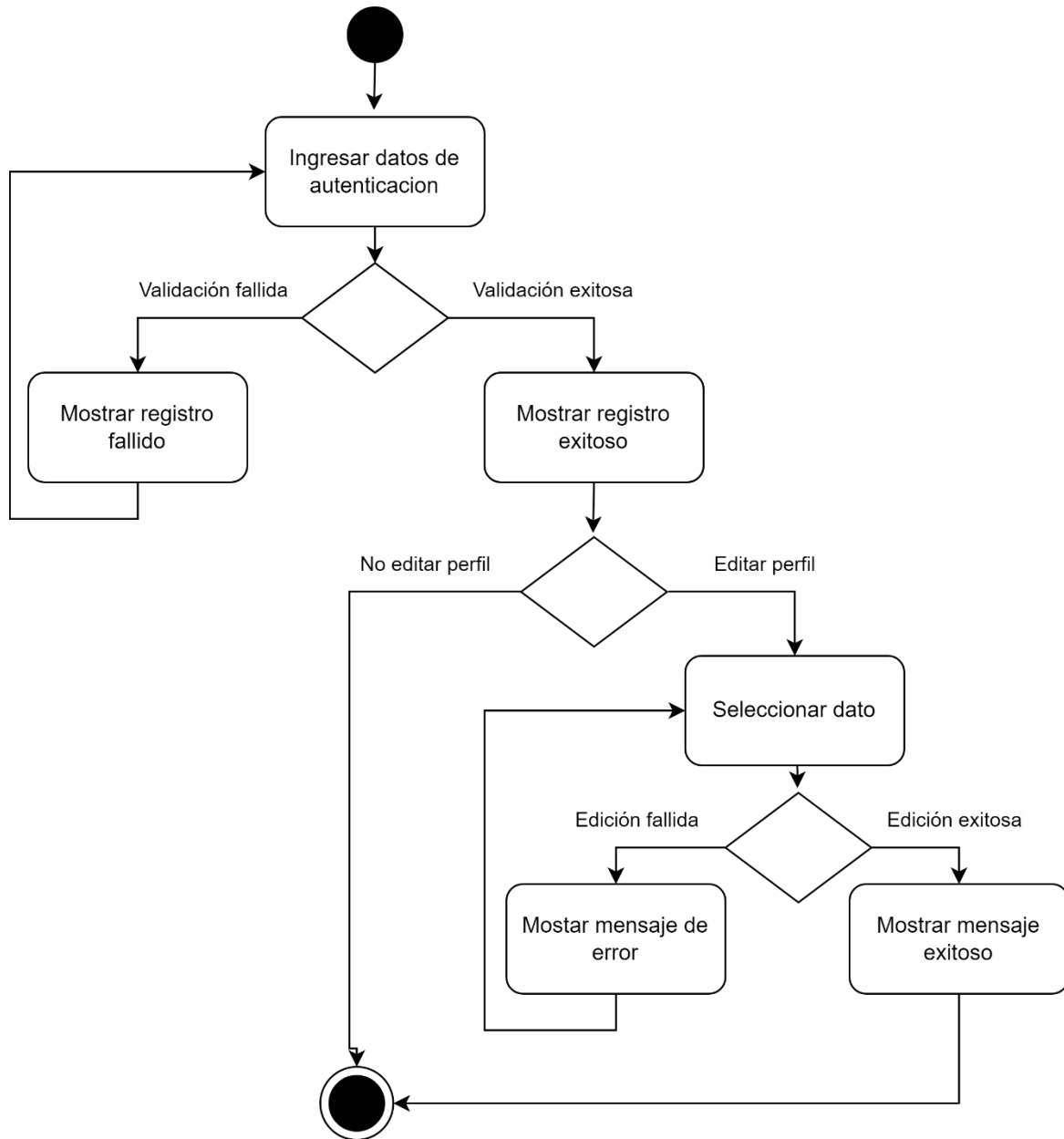


Figura G.11 Autenticación y actualización de datos Administrador.

Anexo H

SCKETCHES DE APLICACIÓN WEB

Inicio de Sesión



Correo electrónico

patricio.garcia@epn.edu.ec

Contraseña

.....

Iniciar de Sesión

No tienes cuenta? [Regístrate Aquí!](#)

Figura H.1 Login Administrador.

Lista de Clientes

Crear Nuevo

Opciones de administrador	image	Cedula	Nombre	Apellido	Correo	FechaNacimiento	Telefono
Ver Información Eliminar Editar		1724718182	cc	cc	ga-xavier@live.com	8/1/2004 12:00:00 AM	0960400373

Figura H.2 CRUD Cliente Administrador.

Lista de Recorridos

Opciones de administrador	Cliente	Conductor	Origen	Destino	Estado	FechaRecorrido	IdRecorrido	ValorRecorrido	Calificacion	Comentario
Ver Información	 cc cc	 ahsjs dbdvs	Hospital General de las Fuerzas Armadas del Ecuador, Queseras del Medio, Guangacalle, Itchimbia, Quito, Pichincha, 170413, Ecuador	Hospital General de las Fuerzas Armadas del Ecuador, Queseras del Medio, Guangacalle, Itchimbia, Quito, Pichincha, 170413, Ecuador	6	8/11/2022 5:00:00 AM	80	3.00	0	

Figura H.3 Información Recorridos Administrador.

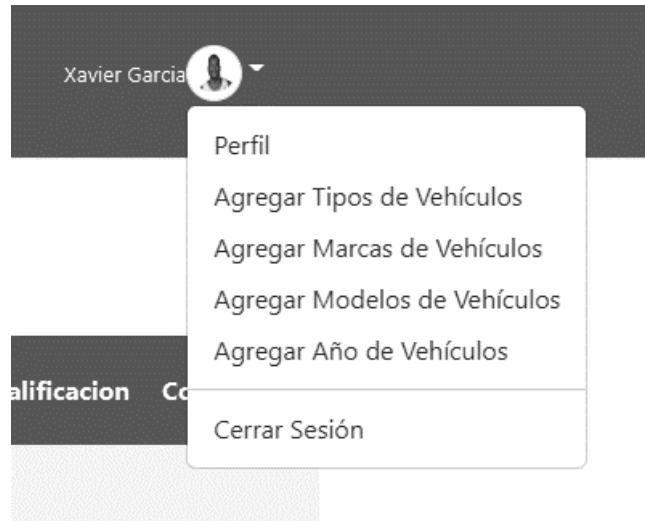


Figura H.4 Administración de datos de Vehículos.

A screenshot of a web browser displaying a registration form titled 'Registro de Usuarios' with the subtitle 'Datos Personales'. The browser's address bar shows 'localhost:63798/Sources/Pages/frmRegistro.aspx'. The form includes a 'Foto:' field with a 'Choose File' button and a 'No file chosen' message. Below this are input fields for 'Nombre:', 'Apellido:', 'Cédula:', 'Correo electrónico:', 'Fecha de Nacimiento:' (with a date picker set to 'mm/dd/yyyy'), 'Teléfono:' (with the value 'patricio.garcia@epn.edu.ec'), and 'Contraseña:' (with a masked password '.....') and 'Confirmar Contraseña:'. At the bottom of the form are two buttons: 'Registrar' and 'Cancelar'.

Figura H.5 Registro de usuario Administrador.

Registro de Usuarios

Datos Personales

Foto:



Choose File No file chosen

Nombre:	Xavier
Apellido:	Garcia
Cédula:	1724718182
Correo electrónico:	patricio.garcia@epn.edu.ec
Fecha de Nacimiento:	mm/dd/yyyy 📅
Teléfono:	0960400373

Guardar

Regresar

Cambiar contraseña:

Contraseña:		Confirmar Contraseña:	
-------------	--	-----------------------	--

Cambiar Contraseña

Figura H.6 Registro de usuarios Administrador.

Actualizar usuario

Datos Personales

Foto:



Choose File No file chosen

Nombre:	cc
Apellido:	cc
Cédula:	1724718182
Correo electrónico:	ga-xavier@live.com
Fecha de Nacimiento:	mm/dd/yyyy 📅
Teléfono:	0960400373

Guardar Cambios

Volver

Figura H.7 Actualización de usuarios Administrador.

Anexo I

SCKETCHES DE APLICACIONES MÓVILES

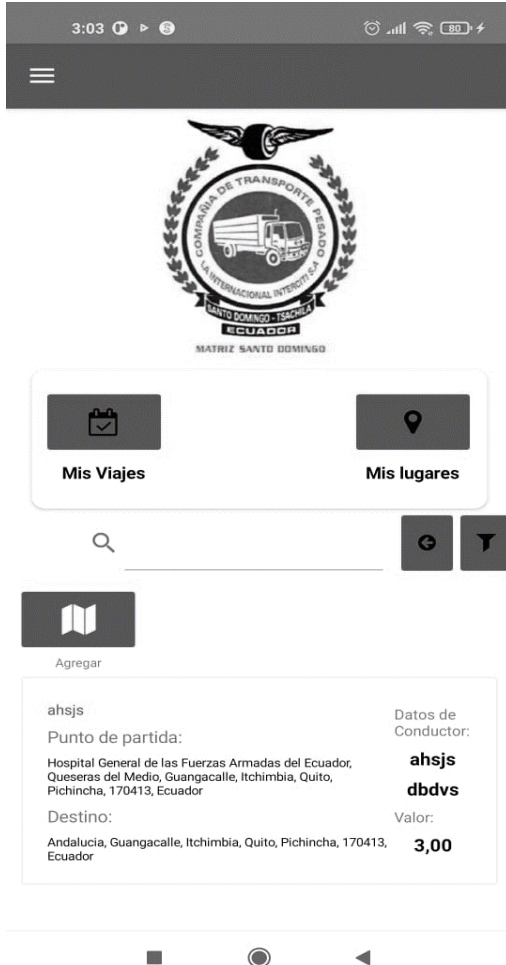


Figura I.1 Información Recorridos Cliente.

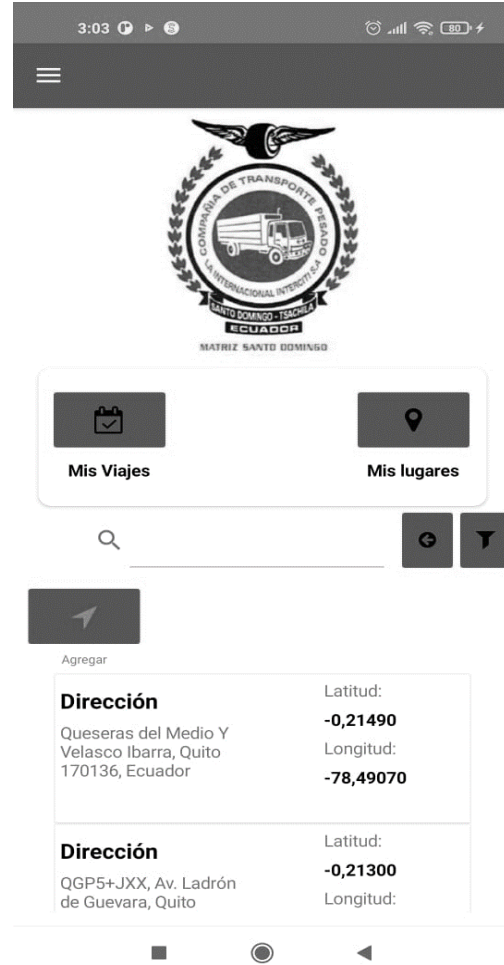


Figura I.2 Mis ubicaciones Cliente.

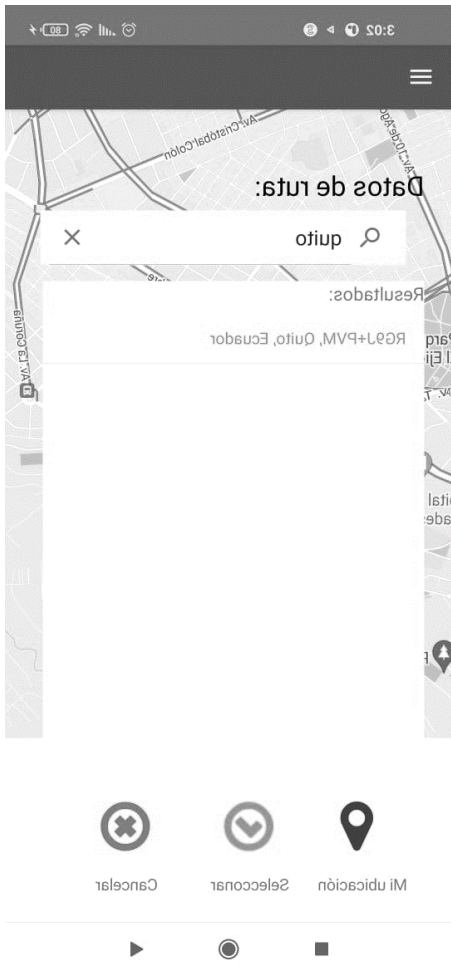


Figura I.3 Buscar ubicación Cliente.

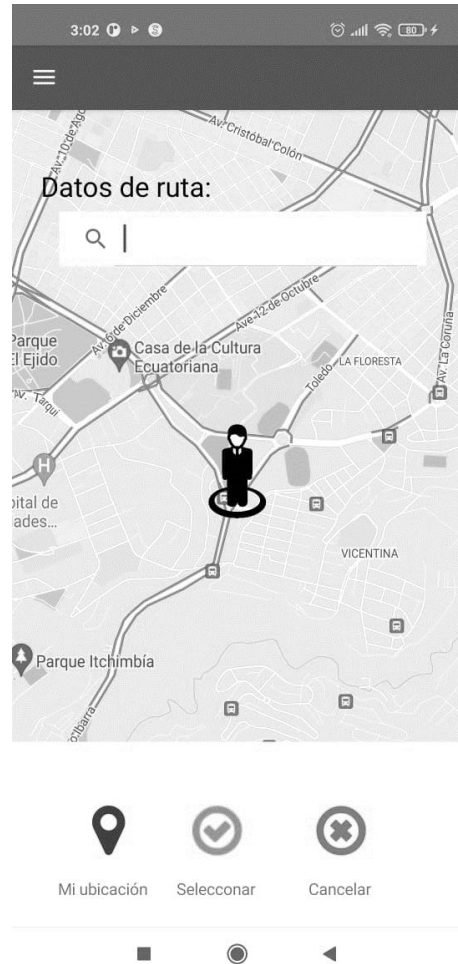


Figura I.4 Mostrar ubicación Cliente.



Figura I.5 Solicitar Recorrido Cliente.

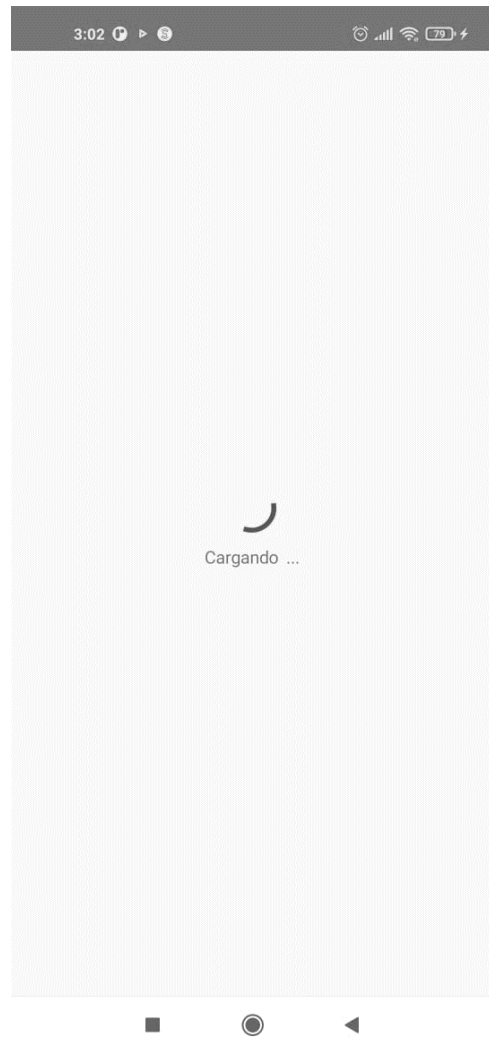


Figura I.6 Pantalla de carga.

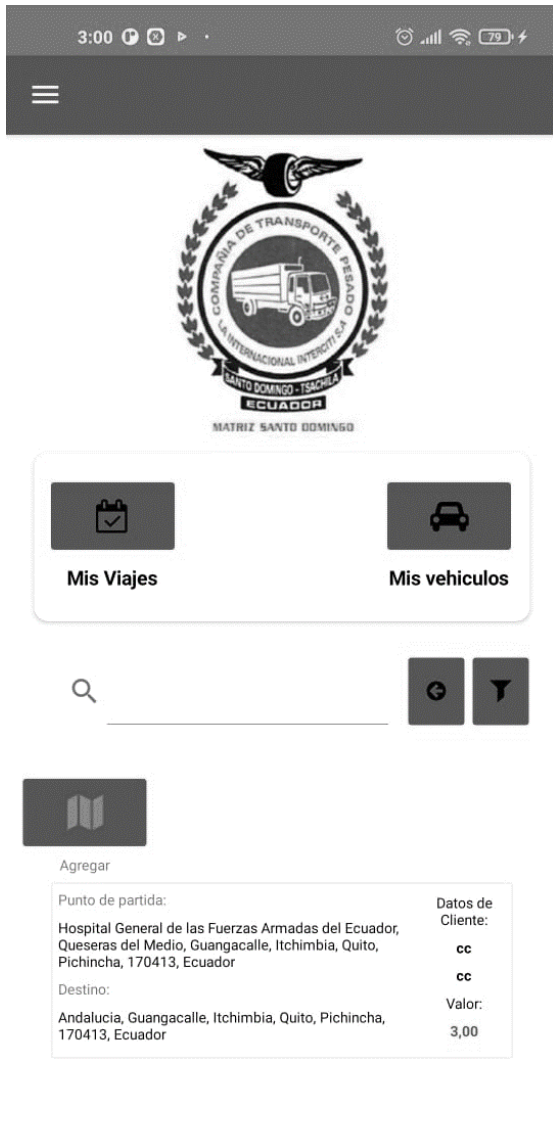


Figura I.7 Información Recorridos Conductor.

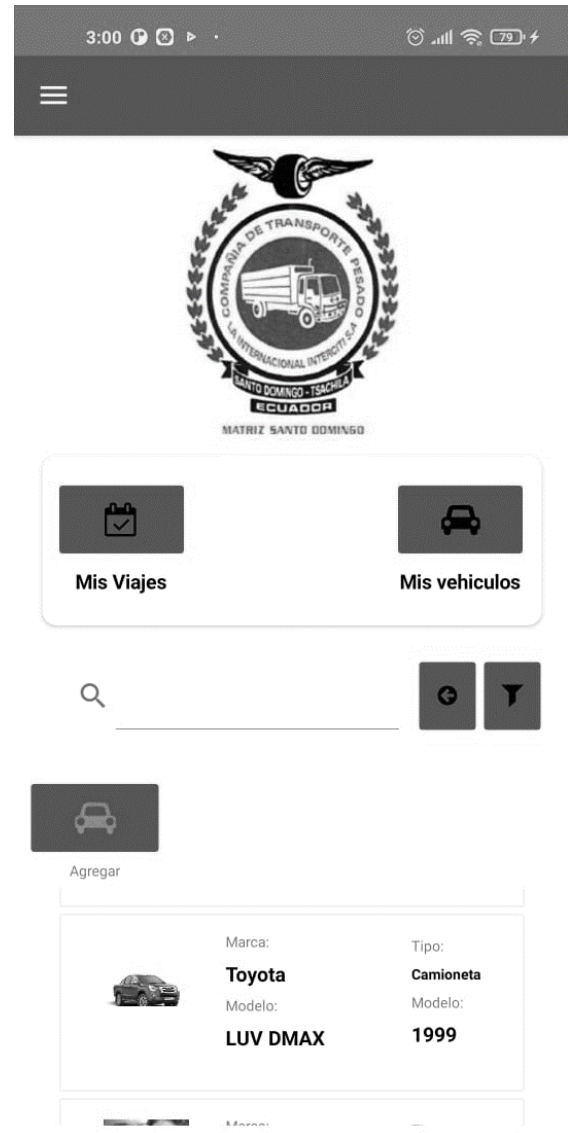


Figura I.8 Vehículos Conductor.

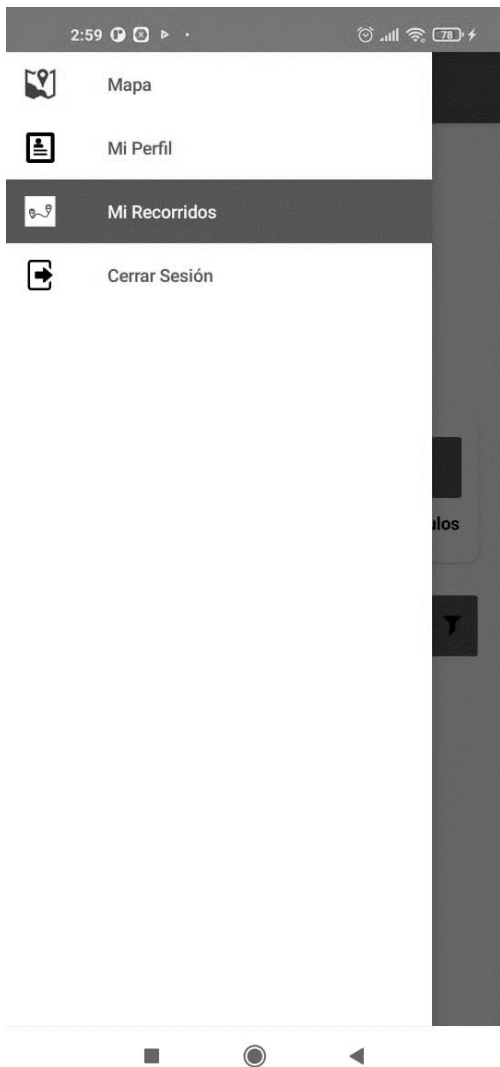


Figura I.9 Menú de aplicación.

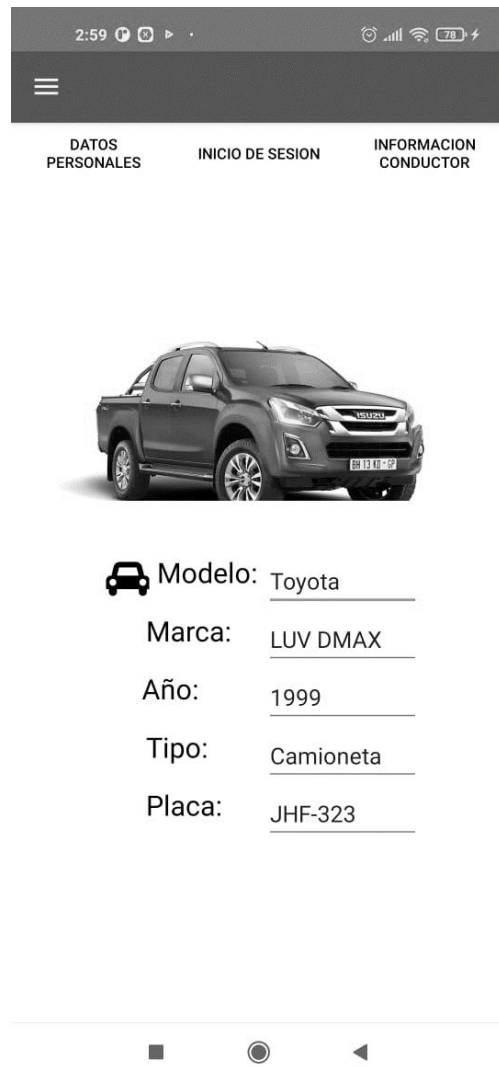


Figura I.10 Información Mi Vehículo Conductor.

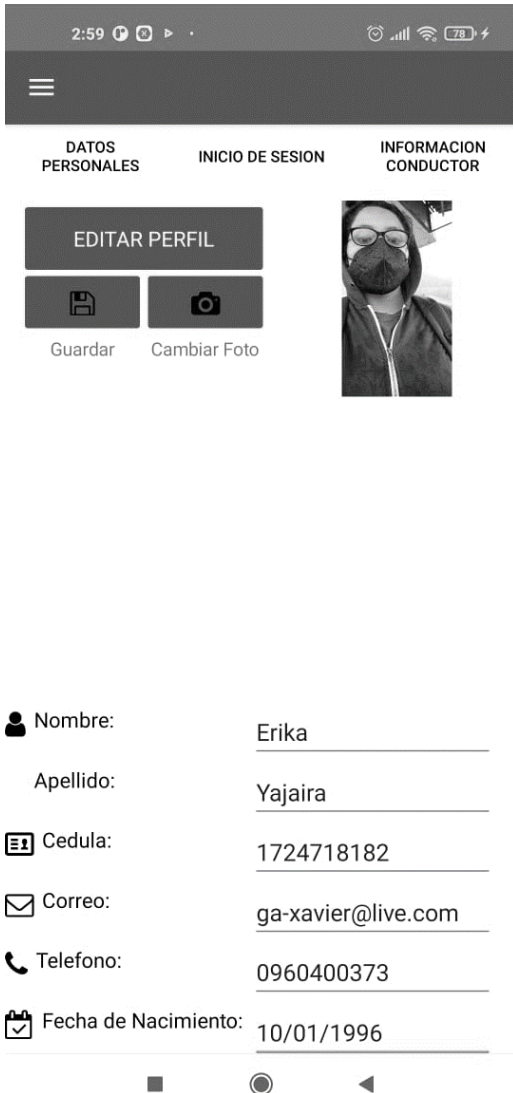


Figura I.11 Actualizar mis datos.



Figura I.12 Escuchar Solicitudes Conductor.

Anexo J

PRUEBAS DE ACEPTACIÓN

Se tiene los resultados de las preguntas realizadas en las distintas encuestas de satisfacción.

APLICACIÓN CLIENTE

1. ¿La aplicación permite el registro de usuarios?

[Más detalles](#)

[Información](#)

● Si	10
● No	0

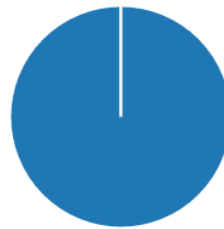


Figura J.1 Pregunta 1 Clientes.

2.

¿La aplicación permite el autenticarse como usuario?

[Más detalles](#)

[Información](#)

● Si	10
● No	0

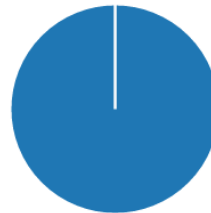



Figura J.2 Pregunta 2 Clientes.

3. ¿Ha logrado actualizar su información después de registrarse?

[Más detalles](#)

 Información

 Si	10
 No	0

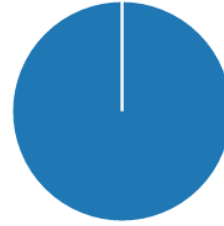



Figura J.3 Pregunta 3 Clientes.

4. ¿Ha logrado realizar la solicitud de un recorrido mediante la aplicación?

[Más detalles](#)

 Información

 Si	10
 No	0

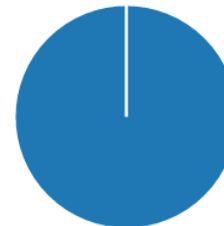



Figura J.4 Pregunta 4 Clientes.

5. ¿Ha logrado guardar y visualizar ubicaciones en la aplicación?

[Más detalles](#)

 Información

 Si	10
 No	0



Figura J.5 Pregunta 5 Clientes.

6. ¿Ha logrado calificar un recorrido realizado previamente?

[Más detalles](#)

● Si	10
● No	0

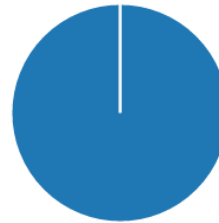


Figura J.6 Pregunta 6 Clientes.

APLICACIÓN CONDUCTOR

1. (0 punto)
¿La aplicación permite el registro de usuarios?

[Más detalles](#)

⚙ Información

● Si	10
● No	0



Figura J.7 Pregunta 1 Conductores.

2. (0 punto)
¿La aplicación permite el autenticarse como usuario?

[Más detalles](#)

⚙ Información


● Si	10
● No	0



Figura J.8 Pregunta 2 Conductores.

3. ¿Ha logrado actualizar su información después de registrarse? (0 punto)

[Más detalles](#)

 Información

● Si	10
● No	0

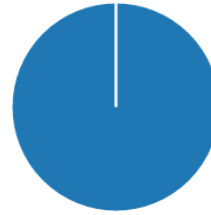



Figura J.9 Pregunta 3 Conductores.

4. (0 punto)

¿Ha logrado receptor las solicitudes de un recorrido mediante la aplicación?

[Más detalles](#)

 Información

● Si	10
● No	0




Figura J.10 Pregunta 4 Conductores.

5. (0 punto)

¿Ha logrado visualizar los vehículos de la empresa en la aplicación?

[Más detalles](#)

 Información

● Si	10
● No	0



Figura J.11 Pregunta 5 Conductores.

6. (0 punto)
¿Ha logrado seleccionar un vehículo en la aplicación?

[Más detalles](#)

● Si	10
● No	0



Figura J.12 Pregunta 6 Conductores.

APLICACIÓN ADMINISTRADOR

¿La aplicación permite el registro de usuarios?

[Más detalles](#)

[Información](#)

● Si	10
● No	0

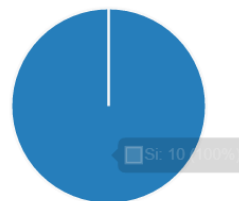


Figura J.13 Pregunta 1 Administradores.

2. ¿La aplicación permite el autenticase como usuario? (0 punto)

[Más detalles](#)

[Información](#)

● Si	10
● No	0

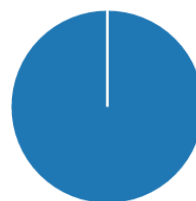


Figura J.14 Pregunta 2 Administradores.

3. ¿Ha logrado actualizar su información después de registrarse? (0 punto)

[Más detalles](#)

Información

● Si	10
● No	0

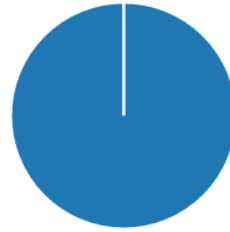


Figura J.15 Pregunta 3 Administradores.

4. ¿Ha logrado administrar los vehículos(Agregar, Editar, Eliminar y visualizar) mediante la aplicación?

[Más detalles](#)

Información

● Si	10
● No	0



Figura J.16 Pregunta 4 Administradores.

5. ¿Ha logrado administrar los recorridos(Visualizar) mediante la aplicación? (0 punto)

[Más detalles](#)

Información

● Si	10
● No	0



Figura J.17 Pregunta 5 Administradores.

6. ¿Ha logrado administrar los clientes(Agregar, Editar, Eliminar y visualizar) mediante la aplicación?

[Más detalles](#)

[Información](#)

● Si	10
● No	0

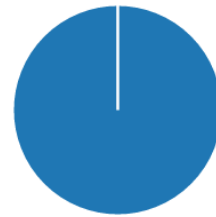


Figura J.18 Pregunta 6 Administradores.

7. ¿Ha logrado administrar los conductores (Agregar, Editar, Eliminar y visualizar) mediante la aplicación?

[Más detalles](#)

● Si	10
● No	0



Figura J.19 Pregunta 7 Administradores.

Anexo K

CÓDIGO DEL PROYECTO

<https://github.com/Xavier9610/TesisInterciti.git>