Prototipo de un sistema de vehículo compartido (*carpooling*) basado en Sistemas Inteligentes de Transporte (ITS), utilizando autenticación con *Blockchain*

Universidad del Cauca

Tesis de Trabajo de Grado

Modalidad: Trabajo de Investigación

Lina Sofía Cardona Martínez

Código: 100316020640

César Andrés Sandoval Muñoz

Código: 100616021761

Director: MSc. Ricardo Salazar Cabrera

Co-Directora: MSc. Mary Cristina Carrascal

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Telemática

Línea de Investigación: Aplicaciones sobre Internet

Popayán, noviembre 2023

Quiero agradecer a mi padre, por ser mi motivación y fortaleza, por sus sabios consejos y sacrificios. A mi madre por su amor incondicional y por brindarme un plato de comida en el día a día. A mis profesores por sus enseñanzas y ejemplo. A mis familiares y amigos por el inmenso apoyo y mantener su confianza en mí.

César Andrés Sandoval Muñoz

Quiero agradecer a mis padres por su esfuerzo que me permitió estudiar y a mis hermanos por su constante aliento. A mis profesores por los conocimientos brindados. También, agradezco a mis amigos y compañeros por enriquecer mi experiencia académica con su colaboración y amistad.

Lina Sofía Cardona Martínez

Agradecimientos

Agradecemos a la Universidad del Cauca, especialmente a todos los profesores que durante nuestro camino por el Alma Mater, aportaron en nuestra formación profesional, así como en nuestro crecimiento personal.

Por último, agradecemos inmensamente a nuestros familiares, quienes nos han acompañado durante toda nuestra estadía universitaria, brindándonos su apoyo emocional y económico, siendo así partícipes de este logro.

Abstract

In today's context, transportation plays a fundamental role in economic development and the establishment of strong social connections. The massive use of public transportation and carpooling are fundamental to solve the main mobility problems in cities, which are the high level of traffic and accidents. The improvement of road infrastructure and the technological improvement of traffic light intersections are not enough to solve these problems; it is very important to use strategies to reduce the number of private vehicles on the city's roads. Ridesharing, specifically carpooling, is an appropriate strategy to achieve such vehicle reduction by increasing the number of people carried in each vehicle. However, safety issues in the use of carpooling tools is a considerable barrier to achieve growth in the use of this type of transportation. Several digital platforms have tried to address the aforementioned security issues by adding functionalities to their mobile applications. However, the security implemented in this type of platforms, in many cases does not use the appropriate technology for the authentication process, which is key to improve security for both the passenger and the driver.

The proposed work intends to investigate how to develop a prototype of a carpooling transportation system that improves the authentication process of passengers and drivers, in order to minimize information security risks. For which it proposes a prototype of a carpooling transportation system that improves the user authentication process, based on Intelligent Transportation Systems (ITS) and using Blockchain technology, known for its security and data integrity.

The proposed development of the prototype of the carpooling type transportation system managed to implement the main functionality of this type of tools, considerably improving the registration and authentication process, making use of Block chain technology. Additionally, by seeking as a reference what is proposed by ITS architectures at international level and previous related works, in this type of applications, it is guaranteed that the interoperability and integration with related mobility applications is facilitated. To validate the proposed prototype, security tests were designed and executed, obtaining very good results, which guarantees an improvement of the critical processes of this type of carpooling tools, mainly in the registration and authentication of users. The potential impact of this project is both social and economic. It has the capacity to encourage the use of carpooling and decrease carbon emissions, which would positively affect mobility and reduce environmental concerns.

Resumen

En la actualidad el transporte es un pilar fundamental para el desarrollo económico y la construcción de sólidas relaciones sociales. El uso masivo del transporte público y el transporte compartido son fundamentales para solucionar los inconvenientes principales de movilidad en las ciudades, que son el alto nivel de tráfico y la accidentalidad. El mejoramiento de la infraestructura vial y el mejoramiento tecnológico de los cruces semaforizados no son suficientes para solucionar estos problemas, es muy importante el uso de estrategias de reducción de vehículos particulares en las vías de la ciudad. El transporte compartido, específicamente el *carpooling*, es una estrategia adecuada para lograr dicha reducción de vehículos, incrementando el número de personas que son transportadas en cada vehículo. Sin embargo, los problemas de seguridad en el uso de herramientas de *carpooling* es una barrera considerable para lograr un crecimiento en el uso de este tipo de transporte. Varias plataformas digitales, han intentado atacar los problemas de seguridad anteriormente mencionados agregando funcionalidades a sus aplicaciones móviles. Sin embargo, la seguridad implementada en este tipo de plataformas, en muchos casos no utiliza la tecnología adecuada para el proceso de autenticación, la cual es clave para mejorar la seguridad tanto para el pasajero, como para el conductor.

El trabajo propuesto intenta investigar sobre cómo desarrollar un prototipo de un sistema de transporte de tipo *carpooling*, que mejore el proceso de autenticación de pasajeros y conductores, para minimizar los riesgos de seguridad de la información. Para lo cual propone un prototipo de un sistema de transporte *carpooling* que mejore el proceso de autenticación de usuarios, basándose en Sistemas Inteligentes de Transporte (ITS) y utilizando la tecnología *Blockchain*, conocida por su seguridad y la integridad de los datos.

El desarrollo propuesto del prototipo del sistema de transporte de tipo *carpooling* logro implementar la funcionalidad principal de este tipo de herramientas, mejorando considerablemente el proceso de registro y autenticación, haciendo uso de la tecnología *Blockchain*, Adicionalmente, al buscar como referencia lo propuesto por las arquitecturas ITS a nivel internacional y los trabajos previos relacionados, en este tipo de aplicaciones, la interoperabilidad e integración con aplicaciones de movilidad relacionadas son garantizadas. Para validar el prototipo propuesto fueron diseñadas y ejecutadas pruebas de seguridad, obteniendo muy buenos resultados, con lo cual podría ser posible un mejoramiento de los procesos críticos de este tipo de herramientas de *carpooling*, principalmente en el registro y autenticación de usuarios. El impacto potencial de este proyecto abarca tanto el ámbito social como el económico. Tiene la capacidad de fomentar el uso del *carpooling* y disminuir las emisiones de carbono, lo que afectaría positivamente la movilidad y reduciría las preocupaciones ambientales.

Tabla de contenido

Lista de Figuras

Lista de Tablas

Lista de acrónimos

ABI	Application Binary Interface
API	Application Programming Interface
ARC-IT	Architecture Reference for Cooperative and Intelligent Transportation
CORS	Cross Origin Resource Sharing
Dapp's	Decentralized Applications
DIMS	Digital Identity Management System
ETH	Ether
EVM	Ethereum Virtual Machine
FAME	Face Authentication for Mobile Encounter
FRAME	Framework Made For Europe
HSTS	HTTP Strict Transport Security
HTTP	Hypertext Transfer Protocol
HU	Historia de Usuario
IoT	Internet of Things
IoV	Internet of Vehicles
ITS	Intelligent Transportation System
JAD	Joint Application Design
JPA	Java Persistence API
KYC	Know Your Customer
NFT	Non-Fungible Token
NSVRC	National Sexual Violence Resource Center
OBE	On-Board Equipment
OMS	Organización mundial de la salud
PBFT	Practical Byzantine Fault Tolerance
PIN	Personal Identification Number
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PUF	Physical Unclonable Functions
REST	Representational State Transfer
RSU	Road Side Unit
SHA	Secure Hash Algorithm
SQL	Structured Query Language
TIC	Transportation Information Center
URL	Uniform Resource Locator
UX	User Experience
WP	Work Package
ZAP	Zed Attack Proxy

ZKP	Zero Knowledge Protocol

Capítulo 1.

1. Introducción

1.1. Planteamiento del problema

El transporte es una actividad fundamental para el desarrollo económico y la construcción adecuada de las relaciones sociales. De ahí que cada vez más el transporte sea considerado como una industria estratégica que demanda mayores facilidades, las cuales han requerido avances tecnológicos. Estos avances pueden traducirse en sistemas inteligentes y aplicaciones software que garanticen la eficiencia y la seguridad en el intercambio de información entre usuarios y sistemas de transporte, con el fin de mejorar el control, la gestión y el seguimiento por parte de los usuarios y diferentes organismos de control [1].

El transporte no solo demanda mejoras en términos de eficiencia, seguridad y gestión, sino que también es necesario reducir su impacto ambiental. En años recientes, la contaminación del aire ha aumentado significativamente, superando las pautas establecidas por la Organización Mundial de la Salud (OMS) [2]. Uno de los ámbitos más afectados por este incremento es el sector del transporte, el cual ha experimentado un aumento significativo en el número de vehículos, lo que ha generado una mayor congestión y, como consecuencia, un aumento en las emisiones de dióxido de carbono (CO_2) [2].

Para mejorar la eficiencia en el transporte y disminuir el impacto al medio ambiente han sido propuestas varias alternativas, una de ellas es el uso de transporte compartido, conocido como *carpooling*, en el cual un propietario de un vehículo particular puede ofrecer los asientos disponibles al usuario que los requiera [3]. Esto implica que los vehículos particulares, que utilicen este tipo de servicio, viajen con una mayor ocupación y por ende reduce la cantidad de vehículos en carretera. El uso de este servicio ayuda a disminuir el tráfico vehicular y el impacto ambiental a causa de la emisión de gases de efecto invernadero [3].

Al utilizar un servicio de *carpooling*, los usuarios emplean comúnmente una aplicación web, lo cual en algunos casos genera inseguridad tanto para el conductor, como para los pasajeros. Esto debido a que la información brindada por los usuarios (pasajeros

y conductores) al momento de registrarse es desconocida, lo que los expone a peligros como suplantación de identidad, hurto, situaciones de abuso y homicidio [4]. Los servicios de transporte existentes como taxi, Uber, entre otros, no generan confianza como opciones de transporte seguras.

En particular, la empresa *Transport for London* (Agencia reguladora inglesa de transporte) mostró interés por la falta de información acerca de situaciones ocurridas dentro de los vehículos [5]. Las situaciones presentadas incluyen desde quejas simples, hasta agresiones físicas, mala conducta, y agresiones sexuales. El Centro Nacional de Recursos de Violencia Sexual (*National Sexual Violence Resource Center* NSVRC) ha estudiado este tipo de comportamientos en Estados Unidos, centrándose en las agresiones sexuales dentro de los vehículos que prestan servicios de transporte y las clasifica en dos categorías generales: agresión sexual y mala conducta sexual [6]. Las agresiones sexuales a su vez están divididas en 10 categorías dependiendo de su gravedad. En un reporte de El Diario NY afirma que entre los años 2019 y 2020 hubo 3824 denuncias en las cinco categorías más graves de agresión sexual y mala conducta por parte de los conductores [7]. Además, de todos estos incidentes sexuales que ocurren dentro de los vehículos, sólo entre el 15.8% y el 35% son reportados a la policía [8]. Con respecto a los casos de muerte laboral, un estudio revela que el conductor que presta un servicio de transporte, ha estado históricamente más expuesto, debido a que los conductores generalmente son objeto de seguimiento y control, sin embargo, no existe información disponible sobre los pasajeros que utilizan el servicio. El Censo de Lesiones Laborales de Estados Unidos muestra que los conductores tienen una probabilidad más alta de morir de manera violenta. Específicamente de las 100 mil muertes de conductores entre los años 2006 al 2012 en dicho país, de 15 a 20 son homicidios; mientras que de 100 mil muertes de trabajadores en general, sólo de 3 a 5 son homicidios [9].

Servicios de transporte como “Uber” e “In Driver”, que hacen uso de plataformas digitales como modelo de negocio, han intentado atacar los problemas de seguridad anteriormente mencionados agregando funcionalidades a sus aplicaciones móviles. En el caso de Uber, ha implementado un botón de emergencia, además de campañas de concienciación [10]. “In Driver”, en cambio, solicita a los conductores las credenciales necesarias para verificar su identidad y la de su vehículo, el pasajero puede elegir el conductor que desee y añade a la aplicación un botón de seguridad que permite a los pasajeros compartir sus datos de viaje con sus familiares y amigos

[11]. En Colombia, la Superintendencia de Industria y Turismo, ha exigido a Uber implementar medidas para evitar incidentes de seguridad como el ocurrido en el año 2016, en donde los datos de 57 millones de cuentas de usuario alrededor del mundo (267 mil en Colombia) estuvieron expuestos debido a fallas en los protocolos de seguridad de Uber [12].

El problema de las soluciones propuestas por estas plataformas, es que los pasajeros no tienen la certeza de que el conductor sea la misma persona registrada en la plataforma. Además, estas soluciones hacen énfasis en la seguridad de los pasajeros y no en la del conductor (quien en muchos casos es el actor vulnerable), ya que no es solicitada la información suficiente a un pasajero para ingresar a la plataforma. Finalmente, cabe anotar que la seguridad implementada en este tipo de plataformas, en muchos casos no utiliza la tecnología adecuada para el proceso de autenticación, la cual es clave para mejorar la seguridad tanto para el pasajero, como para el conductor. Algunos trabajos encontrados en la literatura, proveen algunas opciones, como, por ejemplo, proveer privacidad de la ubicación de un servicio de "*carpooling*" [13], sin embargo, lo que más preocupa en estos casos es la privacidad de las personas.

Lo expuesto anteriormente muestra la importancia de la seguridad en los sistemas de transporte, haciendo indispensable que el manejo de los datos y el proceso de autenticación de los usuarios sea privado y seguro. Respecto al servicio de *carpooling* en particular, algunos inconvenientes de seguridad son la velocidad del vehículo y el comportamiento del conductor, puesto que es importante que los pasajeros tengan la sensación de seguridad al usar un servicio de *carpooling* y los aspectos mencionados no pueden ser controlados por ellos. Además, al igual que en los servicios de transporte convencionales mencionados anteriormente, existe un riesgo debido al desconocimiento de la información tanto del conductor como de los demás pasajeros, que genera desconfianza al desconocer la identidad tanto del viajero como del conductor [14].

Además, los servicios de *carpooling* utilizan tecnologías convencionales para el almacenamiento de los datos de los usuarios. Esto hace que el sistema sea susceptible a ataques de terceros y que la información quede expuesta [15]. Por ello es necesario identificar nuevas tendencias tecnológicas que puedan brindar una solución en cuanto a la seguridad del almacenamiento de datos, la verificación de ellos y un proceso seguro y privado de autenticación.

El uso de arquitecturas ITS, reconocidas internacionalmente, implementadas mediante la tecnología de *Blockchain* pueden ser útiles para mejorar el proceso de autenticación, mejorando la seguridad en el almacenamiento de datos y la verificación de los mismos. Con una mejora en la seguridad del servicio (de *carpooling*), puede mejorar la percepción del usuario a este servicio e incrementar su uso.

Blockchain, es una tecnología disruptiva e innovadora desde que apareció Bitcoin como una solución de finanzas descentralizadas [16]. Además, tiene actualmente un alto reconocimiento y ha experimentado un rápido desarrollo, con alto potencial en diversos ámbitos como los Sistemas de Transporte Inteligente (Intelligent Transport System, ITS) [17] y, que al ser usada en campos tan importantes y delicados como las finanzas (criptomonedas), ha demostrado tener éxito en cuanto a su seguridad y por ello, puede ser una posible solución para evitar problemas de suplantación de identidad en la autenticación de usuarios.

Los *Smart Contracts* o "contratos inteligentes", utilizados previamente a *Blockchain* (surgieron desde 1990) pueden considerarse un elemento fundamental de esta tecnología, y que ha dado en gran parte el potencial necesario a *Blockchain*. Un smart contract es un programa informático que ejecuta un acuerdo entre dos partes en un sistema no controlado por ninguno de los dos. Son resaltables las características adicionales, en comparación con un contrato inteligente, que *Blockchain* puede proporcionar, como inmutabilidad y trazabilidad, específicamente para el caso de mejoramiento del proceso de autenticación en un servicio de *carpooling*. Por lo cual esta tecnología es considerada para la solución del problema planteado [18].

Algunas de las aplicaciones que han empleado *Blockchain* enfocadas en ITS, propuestas por algunos autores, incluyen redes Ad-Hoc vehiculares (VANET), unidades de carretera (Road Side Unit, RSU), gestión de tráfico (*priority scheduling*) y sistemas de alarmas conectados a redes de información a nivel nacional [19]. Esta tecnología les brinda seguridad en el manejo de información, haciendo que los datos no sean fácilmente alterables.

Las soluciones que han sido propuestas a la problemática planteada incluyen sistemas de transporte compartido *carpooling* que no están basados en *Blockchain* y, además, no toman la seguridad en el sistema como un aspecto prioritario [20]. Una de las soluciones encontradas basada en *Blockchain*, aborda el transporte compartido con base a la renta de vehículos particulares y dispositivos de internet de las cosas (Internet of Things, IoT) [21].

1.2. Pregunta de investigación e hipótesis

Con base a todo lo expuesto anteriormente, fue planteada la siguiente pregunta de investigación: ¿Cómo implementar un prototipo de un sistema de transporte de tipo *carpooling*, que mejore el proceso de autenticación de pasajeros y conductores, para minimizar los riesgos de seguridad de la información?

La hipótesis propuesta para dar solución a la pregunta de investigación es la siguiente: Es posible mejorar el proceso de autenticación de pasajeros y conductores, en un prototipo de un sistema de transporte *carpooling*, basándose en Sistemas Inteligentes de Transporte (ITS) utilizando la tecnología *Blockchain*.

1.3. Motivación

El desarrollo de esta propuesta busca usar la tecnología *Blockchain* y los ITS para brindar una solución que permita proteger la información del usuario (pasajero o conductor) y el proceso de autenticación en un servicio de transporte compartido, *carpooling*.

El factor más importante del proyecto es la seguridad de la información del usuario (pasajero y/o conductor) en los sistemas de transporte compartido (*carpooling*), interactuando por medio de una aplicación web con el sistema propuesto, en el cual es posible garantizar que la información tanto de conductores como de pasajeros ha sido verificada al momento del registro y por lo tanto es confiable. Además, la *Blockchain* implementada, garantiza que los datos sean almacenados de forma descentralizada y privada, haciéndola de difícil acceso a terceros.

Una solución como la planteada es necesaria, si es considerada la importancia de la movilidad en cualquier ciudad (grande o intermedia) de todo el mundo; particularmente en un país en desarrollo como Colombia, en donde en los últimos años ha sido evidente una preocupante tasa de muertes a causa de accidentes de tránsito, así como niveles elevados de tráfico y contaminación [22]. Es importante aportar desde el área de la ingeniería a un mejoramiento de la movilidad en las ciudades. Por ese motivo es importante fomentar la investigación en esta área, tanto a nivel local (desde los grupos de investigación de la Universidad del Cauca), como a nivel Nacional (en grupos de investigación de otras Universidades del país, e

instituciones relacionadas como el Ministerio de Transporte, y las Secretarías de Tránsito y/o Transporte y/o movilidad de los diferentes municipios).

Desde la ingeniería y ehan sido propuestas diferentes opciones para mejorar la movilidad en las ciudades, como el mejoramiento de la infraestructura vial, la gestión automatizada de semáforos, el control de vehículos de servicio público, sistemas de transporte masivo y ayudas tecnológicas en los vehículos. Sin embargo, este tipo de estrategias deben ir de la mano con soluciones que fomenten el uso de vehículos particulares compartidos por más de una persona [22]. Si no es posible aumentar el número de personas que usan un mismo vehículo particular, la infraestructura vial no será suficiente para la cantidad de vehículos que transitan diariamente por las vías de las diferentes ciudades. Esto provocará un aumento del tráfico, la contaminación y el riesgo de accidentes de tránsito.

Por tal motivo el servicio de "*carpooling*" es un tema relevante de investigación para el mejoramiento de la movilidad, ya que fomenta el uso compartido del vehículo por parte de personas que tengan trayectos similares. Este servicio ("*carpooling*") tiene como mayor limitación la percepción de inseguridad tanto de pasajeros, como de conductores. Por lo cual, este trabajo, está centrado en buscar el mejoramiento de la seguridad percibida por conductores y pasajeros al utilizar un servicio de este tipo. Una de las formas de mejorar la percepción de la seguridad en el uso del servicio "*carpooling*" es mejorar la seguridad del sistema de información que utiliza este servicio.

La seguridad es actualmente uno de los puntos más críticos en cualquier sistema de información. Sin la seguridad propuesta, los sistemas de información son accesibles más fácilmente por terceros, lo que puede evidenciarse en la mayoría de aplicaciones actuales, específicamente en los sistemas de transporte. Los sistemas de carpooling han descuidado preocupantemente la seguridad de los datos, lo que expone a pasajeros y conductores a los riesgos mencionados en el planteamiento del problema.

Por lo ya expuesto, es importante centrar la investigación en tecnologías que brinden seguridad en la información, dado que *Blockchain* es la tecnología usada en aplicaciones que requieren alta seguridad como las finanzas descentralizadas,

demuestra ser una buena solución para brindar seguridad en otras aplicaciones, como la del sistema propuesto.

El cliente objetivo del proyecto realizado es cualquier usuario que desee compartir su vehículo (conductor), por razones como: movilidad sin restricción, una remuneración financiada por parte del estado o como trabajo comunitario para pago de infracciones. Además, también son clientes objetivo las personas (pasajeros) que necesiten transportarse en la ciudad y que estén interesados en utilizar este tipo de servicio, el cual es planteado con forma de acceso gratuito.

El impacto social y económico que tiene el proyecto está relacionado indirectamente con el posible incremento de la prestación y uso del *carpooling*, lo que podría contribuir a facilitar la movilidad de las personas en una ciudad. Con lo anterior, la congestión vehicular podría verse reducida, debido a que los automóviles estarían viajando con su capacidad máxima de pasajeros, lo que impacta también de manera positiva en los tiempos de desplazamiento. Por otra parte, al reducir el tráfico en las carreteras de las ciudades, la emisión de dióxido de carbono a la atmósfera podría ser menor, disminuyendo los niveles de contaminación ambiental.

1.4. Objetivos

1.4.1. Objetivo general

Proponer un prototipo de un sistema de vehículo compartido (*carpooling*) basado en sistemas inteligentes de transporte (ITS), utilizando un sistema de autenticación y gestión de información de usuario (conductor o pasajero) seguro y privado mediante la tecnología *Blockchain*.

1.4.2. Objetivos específicos

- Implementar un prototipo de un sistema de vehículo compartido, con un mejoramiento en la seguridad en el proceso de autenticación de los usuarios, utilizando *Blockchain*.
- Generar una base de conocimiento de la aplicación de la tecnología *Blockchain* que sirva de apoyo a futuras investigaciones en esta temática.

- Determinar una forma de evaluación de los beneficios obtenidos con la implementación de seguridad en el servicio, en el proceso de autenticación.

1.5. Metodología

Para desarrollar los objetivos mencionados fueron realizadas una serie de actividades según las directrices detalladas en el Cuerpo de Conocimientos de Gestión de Proyectos (*Project Management Body of Knowledge*, PMBOK), desarrollado por el Instituto de Gestión de Proyectos (*Project Management Institute*, PMI) [23]. El desarrollo del prototipo del sistema fue desarrollado utilizando la metodología *Scrum*, principalmente lo relacionado a Historias de Usuario (HU).

1.5.1. Cuerpo de conocimientos de gestión de proyectos

El Cuerpo de Conocimientos de Gestión de Proyectos o PMBOK por sus siglas en inglés es la publicación insignia de PMI y es un recurso para la dirección de proyectos a desarrollar en cualquier industria [24]. El PMBOK describe una serie de prácticas y conocimientos, y documenta la información necesaria para iniciar, planificar, ejecutar, supervisar, controlar y cerrar un proyecto individual. Basado en paquetes de trabajo (*Work Package*, WP) que son un grupo de tareas relacionadas de un proyecto [24].

Este proyecto está segmentado en 5 paquetes de trabajo (WP):

- Identificación de métodos y herramientas de autenticación.
- Implementación de módulo de autenticación basado en *Blockchain.*
- Desarrollo del prototipo.
- Pruebas de seguridad.
- Preparación de documentos y otros entregables.

1.5.2. Framework *Scrum*

En la actualidad, más del 70% de los equipos de desarrollo de las empresas implementan Scrum, ya que su productividad y eficiencia mejoran de 4 a 10 veces más respecto a otros *frameworks* y/o metodologías [25].
Scrum es un marco de trabajo ágil que facilita el desarrollo de los productos garantizando el cumplimiento de los requisitos y necesidades exigidas por los clientes, promueve el trabajo en equipo y la comunicación entre sus miembros [25].

En el desarrollo de los proyectos, los equipos entregan al cliente de forma periódica entregas funcionales parciales para obtener su retroalimentación y permitirle hacer uso parcial de la solución. Estas entregas son denominadas *sprints* y están dividas en tareas que pueden resumirse en las siguientes actividades: planificación, ejecución, revisión y retrospectiva [25].

1.6. Contenido de la monografía

La redacción de la monografía está basada en el cumplimiento de los objetivos específicos del proyecto, a través de paquetes de trabajo propuestos para su desarrollo. El Capítulo 2 presenta conceptos a tener en cuenta para el trabajo presentado. El Capítulo 3 reporta la revisión literaria realizada de acuerdo con la propuesta. El capítulo 4 contiene los aportes de investigación del presente proyecto. El Capítulo 5 presenta la implementación de Blockchain en el proceso de autenticación de la aplicación. El Capítulo 6 muestra la implementación del prototipo realizado. El Capítulo 7 presenta las pruebas de seguridad realizadas en el prototipo de la aplicación. El Capítulo 8 presenta la discusión de los resultados obtenidos en el desarrollo del trabajo. Finalmente, el Capítulo 9 presenta las conclusiones y trabajo futuro de este trabajo de grado.

Capítulo 2.

2. Marco Teórico

A continuación, son presentados algunos conceptos importantes para el desarrollo de la propuesta.

2.1. Web 3.0

La problemática de la centralización del Internet es principalmente que un grupo de grandes compañías realizan un monopolio del internet y deciden las reglas por las cuales la red opera. Web 3.0 nace con el objetivo de darle prioridad al usuario, quien es el que construye, opera y permanece como dueño de sus datos [26].
Como una idea general, puede describirse la evolución del internet como:

- En la era de la Web 1.0, las páginas web eran propiedad de grandes empresas y su contenido estaba basado en objetos estáticos, limitando la interacción entre usuarios. Esta etapa, conocida como la web de solo lectura, abarcó desde 1990 hasta 2004.
- *Web 2.0,* con la llegada de las redes sociales, el usuario empieza a crear contenido, lo que implica que no sólo sean las empresas quienes lo generan, haciendo que la web evolucione a ser de lectura y escritura, esta etapa inició en el año 2004 hasta la actualidad.
- *Web 3.0,* el término fue usado por primera vez por Gavin Wood, el cofundador de Ethereum, quien, en una conferencia en 2014, manifestó que la web necesitaba ser confiable para que los usuarios puedan realizar transacciones con criptomonedas. El dilema es que, si bien en la web 2.0 los usuarios pueden crear contenido y verse beneficiados por las utilidades que este genere, no dejan de ser las grandes empresas las que imponen las reglas y son dueñas de todo el contenido que genere la aplicación.

La solución que propone la Web 3.0 implica darle importancia al usuario por medio de *Blockchain*, las criptomonedas y los *Non Fungible Token* (*NFT*, Tokens no fungibles), convirtiéndose en una red de lectura, escritura y propiedad.

Las características principales de la web 3.0 son: la descentralización, donde los usuarios y constructores comparten la propiedad en igualdad de condiciones; la carencia de permisos, pues todos tienen la misma posibilidad de participar y tomar decisiones en la web; los pagos nativos, que son directamente en criptomonedas desde las billeteras de los usuarios, y no tienen que estar procesados por bancos tradicionales; y la ausencia de la necesidad de confianza, pues con la implementación de los contratos inteligentes para realizar las transacciones, no es requerido un tercero [26].

2.2. Blockchain

Es una base de datos pública, privada o híbrida actualizable, que comparte una serie de ordenadores. Su significado viene de "*Block*", haciendo referencia a datos almacenados en secuencia y "*chain*" que indica que los bloques están encadenados, donde cada bloque hace referencia criptográficamente a su antecesor [27].

Las aplicaciones que implementan *Blockchain* como solución para llevar un registro seguro, descentralizado y distribuido, son conocidas como *Decentralized Applications* (Dapp's, Aplicaciones Descentralizadas), las cuales pueden estar alojadas en diferentes cadenas de bloques. Así que, la elección de *Blockchain* es un aspecto importante dentro de las fases iniciales de desarrollo de la aplicación [27].

Para el proyecto, consideró el uso de *Ethereum*, una *Blockchain* de capa uno, de tipo pública, que es pionera en la implementación de contratos inteligentes.

2.3. Ethereum

Ethereum es una *Blockchain* de capa uno que permite la construcción de aplicaciones y organizaciones descentralizadas de todo tipo. A diferencia de *Bitcoin*, que solo es una red de transacciones financieras*, Ethereum* es una red programable, que puede ser usada por las aplicaciones, para el almacenamiento de datos o para el control de la lógica de negocio de la aplicación [28].

La gobernanza de *Ethereum* es descentralizada, lo que implica que es la comunidad quien crea la red, esto es posible gracias a los nodos. Un nodo es un computador que tiene una copia de la cadena de bloques y sirve como validador de cada transacción que sea ejecutada. Existen miles de nodos distribuidos en todo el mundo que son dirigidos por usuarios y empresas, esto le brinda solidez a la *Blockchain* y la hace

menos susceptible a ataques informáticos y a posible censura por parte de los gobiernos.

Por medio de la criptomoneda Ether (ETH), es posible realizar transacciones dentro de la red de *Ethereum*, es una moneda completamente digital y no existe un ente gubernamental que la controle. Ether es usado también para pagar el impuesto conocido como gas fee, que es un valor a pagar por el poder computacional que requiere la red *Ethereum* para dar respuesta a una solicitud [28].

2.4. Smart Contract

Ethereum necesita de *Smart Contracts* (contratos inteligentes) que vivan en su cadena de bloques, estos contratos inteligentes son programas informáticos que son ejecutados al activarse una transacción. Estos contratos contienen todo el conjunto de leyes o funciones que gobiernan la aplicación, por medio de una estructura condicional que no puede ser modificada. Además, almacenan datos (estado del contrato) que apuntan a una dirección específica dentro de la cadena de bloques [29].

Algunas características importantes de los contratos inteligentes son: ejecución automática, esto elimina la necesidad de confianza entre las partes, puesto que el contrato es ejecutado automáticamente al cumplirse las condiciones; resultados predecibles, puesto que no hay ambigüedad en la interpretación del contrato, este funciona de acuerdo con las condiciones escritas en el código; el registro público, cada transacción es almacenada en la *Blockchain* y cualquier persona en el planeta con acceso a Internet puede hacer seguimiento de cada movimiento de activos e información.

Los contratos inteligentes son absolutamente descentralizados, por lo que no tienen “contacto” con el mundo real, por ello son necesarios puentes que les puedan brindar los datos necesarios para ejecutarse. Es debido a esto, que existen los oráculos, que son programas híbridos que cuentan con contratos inteligentes, pero también tienen elementos *off chain* (fuera de la *Blockchain*) como interfaces de aplicaciones (Applications Interfaces, *APIs*, por sus siglas en inglés) que traen diferentes datos necesarios para el contrato [29].

Programar un contrato inteligente requiere disponer de una copia de la máquina virtual de *Ethereum* (EVM) o una máquina virtual de prueba similar a la EVM, de manera que, puedan hacerse pruebas del contrato inteligente sin tener que pagar el gas fee.

Además, es necesario de un lenguaje de programación, cada *Blockchain* puede contener su propio lenguaje de programación o una variante de estos, en el caso de Ethereum, pueden ser usados dos lenguajes de programación que son *Solidity* y *Vyper*. De los dos, fue seleccionado *Solidity* para el desarrollo del proyecto, por ser más accesible en cuanto a la similitud con lenguajes de programación conocidos, el acceso a librerías de otros lenguajes, y las características de la programación orientada a objetos que pueden ser implementadas [29].

2.5. *Intelligent Transportation Systems* (ITS)

2.5.1. Aspectos importantes

El término ITS (Sistemas de Transporte Inteligentes) surgió en 1991, cuando los profesionales del área de transporte observaron que la tecnología podría desempeñar un papel fundamental en la optimización de los desplazamientos terrestres. Fue entonces cuando el Congreso Nacional de los Estados Unidos inauguró el Programa Nacional ITS. Desde ese punto en adelante, comenzaron a desarrollarse aplicaciones y sistemas tecnológicos con el objetivo de potenciar diversos aspectos del transporte, como su seguridad y eficiencia, al simplificar el control y la administración de los componentes relacionados con el transporte [30].
Algunas de las ventajas que aporta el uso de ITS son [31]:

- Aumenta la seguridad de los conductores
- Mejora de la eficiencia del tráfico
- Control de elementos de carreteras
- Control automático de vehículos

2.5.2. Arquitectura ITS

En el campo de los sistemas de transporte son importantes las arquitecturas ITS de referencia a nivel mundial. Las arquitecturas que tienen un alto nivel de uso a nivel internacional tomadas como referencia fueron: la arquitectura europea (FRAME) y la arquitectura americana (ARC-IT).

2.5.2.1. Arquitectura Europea

Mejor conocida como arquitectura FRAME, de las palabras *Framework Made For Europe*, fue creada para proporcionar un marco estable, necesario para sistemas de transporte integrados en la Unión Europea [32].

La arquitectura FRAME posee diez grupos funcionales, que son:

- Grupo 1. Genérico y no funcional.
- Grupo 2. Planificación y mantenimiento de infraestructuras.
- Grupo 3. Cumplimiento de la ley.
- Grupo 4. Transacciones financieras.
- Grupo 5. Servicios de emergencia.
- Grupo 6. Guía e información del viaje.
- Grupo 7. Tráfico, incidencias, gestión de la demanda y sistemas cooperativos.
- Grupo 8. Sistemas de vehículos inteligentes.
- Grupo 9. Gestión de flotas.
- Grupo 10. Administración de transporte público.

Los grupos identificados con características comunes a las requeridas en el presente proyecto, dichas categorías son [33]:

- **Sistemas de vehículos inteligentes.** En esta categoría corresponde a las funciones obtenidas por un vehículo, tales como prevención de colisiones longitudinales y laterales, mantenimiento de carril, pelotones de vehículos, control de velocidad y estado de alerta del conductor [34].
- **Gestión del transporte público.** En este grupo corresponde a las actividades asociadas al transporte público, tales como el transporte público compartido, la información durante el viaje y seguridad del viajero [34].

En cuanto al primer grupo de interés (Sistemas de vehículos inteligentes), este contiene algunas características a implementar en un servicio relacionado con seguridad en vehículos, pero en cuanto a la conducción, por lo cual no está relacionado directamente con el objeto del trabajo de investigación. El segundo grupo identificado (Gestión del transporte público) concuerda más con las ideas y características del presente proyecto, pero con un enfoque informativo al viajero, sin embargo, la sección de la seguridad del viajero es considerada, ya que en este proyecto es relevante.

2.5.2.2. Arquitectura Americana ITS (*Architecture Reference for Cooperative and Intelligent Transportation*, ARC-IT)

Esta arquitectura presenta un marco común para definir e integrar ITS, pero no exige ninguna implementación en particular [35].

Esta arquitectura está dividida en paquetes de servicios que representan segmentos de la vista física, la cual reúne varios objetos físicos, objetos funcionales y flujos de información. Estos paquetes de servicios están agrupados en 10 áreas de servicio, que son:

- Operaciones de vehículos comerciales.
- Gestión de datos.
- Mantenimiento y construcción.
- Gestión de estacionamiento.
- Seguridad pública.
- Transporte público.
- Soporte.
- Viajes sostenibles.
- Gestión de tráfico.
- Información del viajero.
- Seguridad del Vehículo.
- Clima.

Se consultaron algunos paquetes de servicio de algunas áreas relacionadas con el tema de la investigación como, por ejemplo: transporte público, seguridad del vehículo e información del viajero. Las dos primeras áreas no contienen paquetes de servicios específicos relevantes para el proyecto. Por el contrario, en el área de servicio de información del viajero fue identificado un paquete de servicio que contiene características similares a las requeridas en este proyecto.

El área de información del viajero brinda información estática y dinámica sobre la red de transporte a los usuarios. Cuenta con 6 paquetes de servicio que son: difundir información del pasajero; información personalizada del viajero; guía de ruta dinámica; planificación de viajes y guía de ruta proporcionada por la infraestructura; información y reserva de servicios turísticos, viajes compartidos dinámicos, y transporte de uso compartido; y señalización en vehículos [36].

El paquete de servicio que más similitud tiene con el proyecto presentado es el que corresponde a viajes compartidos dinámicos y transporte de uso compartido, el cual es explicado a continuación.

El paquete de servicio denominado “Viajes compartidos dinámicos y transporte de uso compartido (*Dynamic ridesharing and shared use transport – TI06*)” [37], permite a los viajeros organizar viajes compartidos a través de un dispositivo personal con conexión a un sistema de comparación de viajes. Las entradas de dicho servicio son los pasajeros y conductores (antes, durante y después del viaje), quienes obtienen emparejamientos óptimos para brindarles una ruta conveniente entre sus dos ubicaciones (destino y origen). Luego del viaje, brinda información al paquete de servicios para mejorar la experiencia del usuario en futuros viajes.

El paquete de servicio aborda tres tipos de uso compartido:

- Un viajero organiza el uso temporal de un vehículo
- Un viajero contrata un vehículo para que lo lleve de un lugar a otro.
- Compartir bicicletas.

La Figura 1 muestra la arquitectura base del paquete de servicio seleccionado, la cual sirvió como guía para la elaboración del prototipo del presente trabajo.

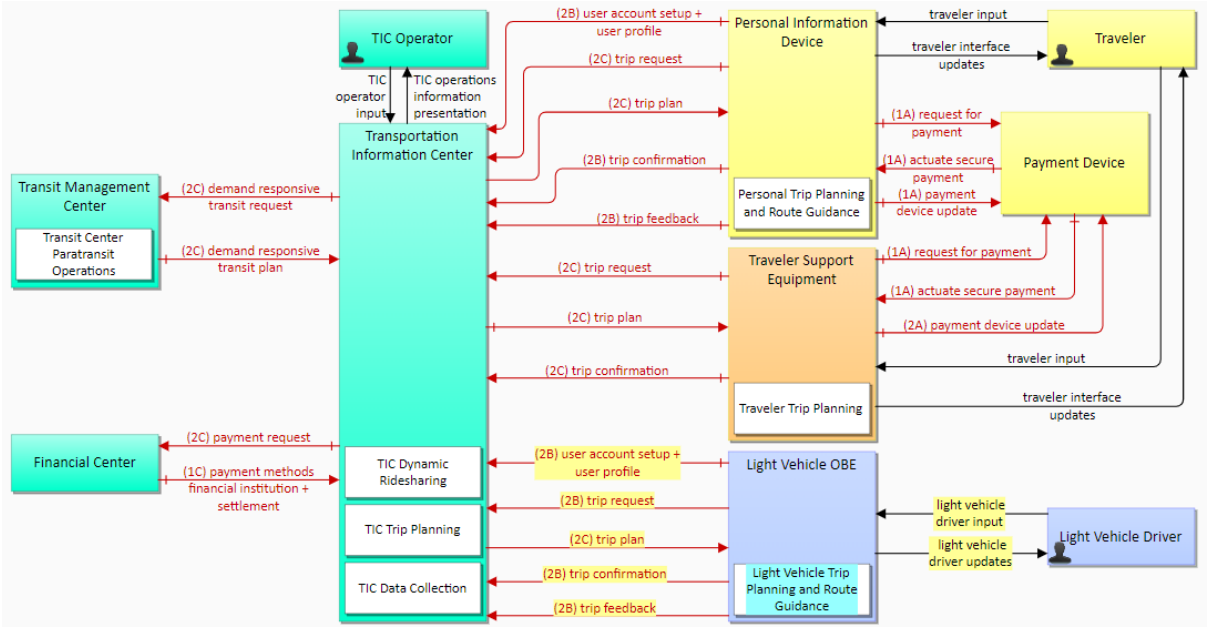


Figura 1. Diagrama físico del servicio de viajes compartidos dinámicos y transporte compartido, propuesto por ARC-IT. Fuente: [38].

La Figura 1 muestra el diagrama físico del servicio seleccionado que propone la arquitectura, compuesto por cuatro tipos de objetos físicos, que son: personal (amarillo), vehículo (azul), campo (naranja), y centro de gestión (verde). Además, el diagrama contiene objetos funcionales (recuadros blancos), incluidos en los objetos físicos las diferentes solicitudes propuestas por ARC-IT entre los diferentes objetos, las cuales están presentes en la Figura 1 con líneas rojas y negras.

A continuación, son detallados los objetos físicos presentados en la Figura 1.

- Conductor (*Driver*). Del tipo de objeto Vehículo (color azul). Actor que interactúa con la arquitectura, conduciendo el vehículo.
- Equipo a bordo del vehículo (*On-Board Equipment* u OBE). Del tipo de objeto Vehículo (color azul). Equipo encargado de las funciones de sensado, de procesamiento y de comunicación basada en el vehículo.
- Centro de información de transporte (*Transportation Information Center o TIC*). Del tipo de objeto Centro de Gestión (color verde). Encargado de distribuir, procesar y recopilar la información.
- Centro financiero (*Financial Center*). Del tipo de objeto Centro de Gestión (color verde). Organización que maneja las solicitudes de transacciones electrónicas de fondos del usuario de servicio al proveedor de servicio.
- Centro de gestión de transporte público (*Transit Management Center*). Del tipo de objeto Centro de Gestión (color verde). Gestiona flotas de vehículos y coordina con otros modos y servicios de transporte.
- Operador TIC (*TIC Operator*). Del tipo de objeto Centro de Gestión (color verde). Persona o conjunto de personas que monitorean y administran los servicios de información del viajero dados por el Centro de Información de Transporte (TIC).
- Dispositivo de información personal (*Personal Information Device*). Del tipo de objeto Personal (color amarillo). Proporciona información al viajero, planificación de viajes y guía de ruta.
- Equipo de apoyo al viajero (*Traveler Support Equipment*). Del tipo de objeto Campo (color naranja). Brinda acceso a la información del viajero en ubicaciones principales, administra la inscripción al servicio y al pago electrónico de tarifas.

- Dispositivo de pago (*Payment Device*). Del tipo de objeto Personal (color amarillo). Transferencia electrónica de fondos del usuario al proveedor del servicio, comúnmente usado mediante tarjetas.
- Viajero (Traveler). Del tipo de objeto Personal (color amarillo).

De los objetos funcionales, es importante resaltar los siguientes:

- Planeación de viajes del Centro de Información de Transporte (*TIC Trip planning*), Ubicado en el objeto físico TIC.
- Almacenamiento de Datos del Centro de Información de Transporte (*TIC Data collection*). Ubicado en el objeto físico TIC.
- Planeación de viaje personal y guía de ruta (*Personal Trip planning and route guidance*). Ubicado en el objeto físico Dispositivo de Información Personal.
- Planeación de uso compartido personal (*Personal shared use planning)*, Ubicado en el objeto físico Dispositivo de Información Personal.

El paquete de servicios TI-06 es aplicable principalmente a dos funcionalidades.

- Movilidad de uso compartido (TI06.1): Esta implementación puede aplicarse a cualquier vehículo de uso compartido, incluido el "*car sharing*".
- Viajes compartidos dinámicos (TI06.2): Cubre ambas implementaciones para viajes compartidos dinámicos y viajes compartidos estilo Uber.

Estos objetos físicos, objetos funcionales, relaciones y demás, propuestos en la arquitectura *ARC-IT* fueron considerados en el diseño del prototipo de este trabajo.

2.6. Carpooling

Carpooling es el uso compartido de un vehículo durante un trayecto definido. Existen tres tipos principales de viajes *carpooling* que son [39]:

- *Casual Carpooling,* es el uso compartido de automóvil informal, en donde no existe el cobro a los pasajeros.
- *Real-Time Carpooling,* es conocido como el uso compartido de vehículos basado en aplicaciones. Permite a las personas organizar viajes por medio de una aplicación.
- *Vanpooling,* es el uso compartido de autos grandes, generalmente puede tener de 7 a 15 pasajeros que comparten los costos del viaje.

Capítulo 3.

3. Revisión de literatura

3.1. Revisión sistemática

En primer lugar, fueron seleccionadas las bases de datos de referencia para realizar la búsqueda de los artículos de interés, las cuales fueron Scopus y ScienceDirect, a las cuales la Universidad del Cauca brinda acceso. La revisión sistemática fue realizada utilizando la metodología *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* versión 2020, conocida por su sigla en inglés con el nombre de PRISMA.

La metodología PRISMA es usada para documentar revisiones sistemáticas, fue lanzada en el año 2009 haciendo referencia a cuatro fases: Identificación, presentación, elegibilidad e inclusión. En el caso de esta investigación, fue utilizada la versión de PRISMA 2020 que, a diferencia de la versión anterior, une las fases de presentación y elegibilidad en una sola denominada fase de detección. Como resultado, esta nueva versión cuenta con tres fases, las cuales son: fase de identificación, fase de detección y fase de inclusión [23].

A continuación, son detalladas las fases de la metodología PRISMA versión 2020 aplicadas en la investigación realizada. En cada fase, son detallados los métodos de inclusión y exclusión de artículos, con el objetivo de obtener un conjunto reducido de artículos relevantes para la investigación.

3.1.1. Fase de identificación

Se seleccionaron dos cadenas para realizar la búsqueda en la base de datos, las cuales contienen palabras clave que fueron consideradas importantes en la investigación. Debido a que el proyecto está enfocado en el transporte compartido usando *Blockchain,* la primera cadena fue la siguiente: ***"Blockchain AND transportation AND system AND shared".*** Por otra parte, el segundo enfoque del proyecto es la autenticación de usuarios, por lo que la segunda cadena seleccionada

fue: ***"Blockchain AND identity AND authentication".*** **En el Anexo A. Revisión sistemática,** puede encontrarse mayor información sobre esta fase de la metodología PRISMA.

3.1.2. Fase de detección

El descarte por título fue el primer criterio de elegibilidad, ya que brinda una idea general del contenido de cada artículo.

El segundo criterio de elegibilidad consistió en la revisión del resumen de cada artículo seleccionado. Para obtener más información sobre esta fase de la metodología PRISMA, consulte el **Anexo A. Revisión sistemática.**

3.1.3. Fase de inclusión

Posteriormente, fue realizada una lectura detallada de los artículos seleccionados hasta esta etapa (25 en total), los cuales resultaron categorizados en grupos de acuerdo con la temática que abarcan. Los grupos fueron: medicina, movilidad y autenticación.

De acuerdo con los resultados obtenidos en esta fase, los trabajos más relevantes de cada grupo identificado (11 trabajos en total) fueron revisados detalladamente. **En el Anexo A. Revisión sistemática,** puede encontrar más información sobre esta fase.

La Tabla 1 muestra un resumen de los criterios considerados importantes respecto a los trabajos relacionados seleccionados en la investigación usando la metodología PRISMA. Sólo tres de las soluciones de los trabajos relacionados están enfocadas en transporte, pero ninguno está enfocado en el *carpooling*. Aunque la mayoría de los artículos realiza una autenticación de usuarios (solo tres realizan autenticación de dispositivos), no realizan autenticación o manejo de datos biométricos. Solo un artículo está enfocado en transporte y realiza autenticación de usuarios y es el único amigable con el medio ambiente, pero no realiza autenticación o manejo de datos biométricos ni pruebas de seguridad.

Criterio / Propuesta	[40]	[41]	[42]	[43]	[44]	[21]	[45]	[46]	[47]	[48]	[49]
Enfocado en transporte					✔	✔	✔				
Autenticación d e usuarios	✔	✔	✔			✔		✔	✔	✔	✔
Autenticación de dispositivos				✔	✔		✔				
Manejo de datos biométricos		✔	✔								✔
Amigable con el medio ambiente						✔					
Pruebas de seguridad	✔	✔			✔		✔	✔	✔		✔
Uso de métodos de encriptación	✔	✔	✔	✔	✔		✔	✔	✔	✔	✔

Tabla 1. Resumen de evaluación de los trabajos relacionados.

3.2. Revisión del estado del arte e identificación de métodos y herramientas de autenticación

La revisión del estado del arte fue realizada con base en los artículos encontrados en la revisión sistemática. 11 artículos fueron seleccionados, los cuales podían aportar al proyecto en cuanto a métodos de autenticación, métodos de encriptación, pruebas de seguridad y manejo de la información.

Primero, fueron revisados los artículos en cuanto a métodos de autenticación, de encriptación y las pruebas de seguridad implementadas en cada caso (si aplica).

- *Towards blockchain-IoT based shared mobility: Car-sharing and leasing as a case study [21].*
    - Como método de autenticación o reconocimiento de usuarios, este artículo maneja *Know your customer* (KYC, conozca su cliente).
    - Uso de punteros hash como técnica criptográfica implementada que brindan derecho al olvido.
- *A Blockchain Based Decentralized Identifiers for Entity Authentication in Electronic Health Records [40].*

    - La autenticación es realizada mediante credenciales verificables e identificadores descentralizados generados por un algoritmo.
- *A Permissioned Blockchain-Based Identity Management and User Authentication Scheme for E-Health Systems [41].*
    - La autenticación de usuarios es realizada a partir de la toma de datos biométricos.
    - La encriptación es realizada por medio de llaves y punteros hash.
    - Las pruebas de seguridad realizadas fueron: suplantación de identidad, mostrando el cumplimiento del anonimato, la autenticación mutua, la trazabilidad condicional y acuerdos de clave de sesión.
- *Health-ID: A Blockchain-Based Decentralized Identity Management for Remote Healthcare [42].*
    - El sistema de autenticación propuesto utiliza identificadores de salud denominados *HEALTH-ID.*
    - El cifrado es realizado por medio de llaves privadas y es almacenado en la nube.
- *A decentralized framework for device authentication and data security in the next generation internet of medical things [43].*
    - La autenticación realizada en este artículo no es compatible con la mencionada en este proyecto, ya que está enfocada en dispositivos.
    - Es importante el método de encriptación utilizado, ya que, aunque es autenticación de dispositivos, su método de encriptación denominado *Physical Unclonable Functions* (*PUF*) es clonable y usa claves que son generadas de acuerdo con la distancia *hamming* entre nodos.
    - Las pruebas de seguridad están enfocadas a ataques de reproducción, suplantación de identidad y sybil.
- *EASBF: An Efficient Authentication Scheme over Blockchain for Fog computing-enabled internet of vehicles" [44].*
    - La autenticación del artículo está enfocada a vehículos.
    - El manejo de los datos es realizado hace mediante métodos de encriptación de curva elíptica, función hash unidireccional, y un algoritmo de consenso que es tolerante a fallos Bizantinos. Además, realiza una codificación de alto nivel en los datos presentados al momento de autenticar dispositivos.

- Las pruebas de seguridad están centradas en ataques de repetición y man-in-the-middle, los cuales son validados con la herramienta de Validación Automatizada Protocolos y Aplicaciones de Seguridad de Internet (AVISPA).

- *Blockchain Empowered Cooperative Authentication with Data Traceability in Vehicular Edge Computing," in IEEE Transactions on Vehicular Technology [45]*.
    - Autenticación para vehículos por medio de proxy dinámico que es compartido entre la red de vehículos.
    - Rendimiento sujeto al poder de procesamiento del dispositivo en el vehículo.
    - Enfoque en seguridad, privacidad y descentralización debido a que promueve una autenticación colaborativa.
    - Cada vehículo es un nodo que verifica la autenticidad de los demás nodos.
- *A zero-knowledge-proof-based digital identity management scheme in blockchain [46].*
    - Zero-knowledge es una técnica de criptografía que consigue ser programada mediante un contrato inteligente implementado en *Blockchain* de *Ethereum*.
    - Las claves generadas son almacenadas en la cadena de bloques y son verificadas por el contrato inteligente cada vez que son consultadas.
    - El hash es generado por medio del Algoritmo de Hash Seguro de 256 bits (Secure Hash Algorithm 256-bit), SHA 256.
- *PTAS: Privacy-preserving Thin-client Authentication Scheme in blockchain-based PKI [47].*
    - Autenticación por medio del protocolo PTAS que ofrece llaves públicas que permiten al usuario interactuar con la *Blockchain*. Esta interacción es realizada sin tener que hacer una descarga de la cadena de bloques en el dispositivo final, pero conservando la seguridad y privacidad gracias al sistema propuesto, que mejora de acuerdo con el número de nodos que contenga la *Blockchain*.

    - Este sistema está pensado para usarse en dispositivos IoT, que normalmente no cuentan con la capacidad de almacenamiento para una cadena de bloques.
    - Se sacrifica rendimiento por seguridad.
- *AuthChain: A Decentralized Blockchain-based Authentication System [48].*
    - Autenticación por medio de *tokens* generados por un contrato inteligente programado en el lenguaje Solidity y ejecutado en el *Blockchain* de *Ethereum*.
    - Los *tokens* son encriptados tipo hash generados por el protocolo SHA256, este hash es almacenado en la cadena de bloques.
    - No son necesarias contraseñas, ni datos biométricos, el *token* genera un código QR que permite ingresar a las aplicaciones compatibles con el sistema AUTH CHAIN.
- *A secure end-to-end verifiable e-voting system using Blockchain and cloud server [49].*
    - La autenticación de los usuarios del sistema de votación fue realizada por medio de datos biométricos, en este caso la huella dactilar.
    - Los datos biométricos son encriptados por un hash generado automáticamente por el sistema y almacenado en la cadena de bloques.
    - Se genera una llave única para cada huella, que es verificada para que no coincida con nuevos usuarios.

3.3. Conjunto de Tecnologías y herramientas a utilizar en el proyecto

Un análisis de las tecnologías implementadas en los artículos sugiere que el siguiente conjunto de tecnologías puede ser una buena opción para diseñar un proyecto *Blockchain* que utiliza datos biométricos encriptados y los almacena de manera segura:

- ***Ethereum**,* como plataforma de contratos inteligentes para implementar la lógica del negocio y almacenar los datos biométricos encriptados en una *Blockchain* de tipo pública [50]. Fue utilizada una *Blockchain* ya creada, debido a que una red ya construida es más segura, puesto que ha sido revisada y validada por múltiples desarrolladores. Además, es una red que está

establecida, lo que facilita el manejo de algunas herramientas y recursos necesarios para desarrollar una aplicación.

- ***React*** fue el *"framework"* (librería de *javascript* para el desarrollo *frontend*) seleccionado para crear interfaces de usuario de forma rápida y eficiente. Fueron considerados otros *"frameworks"* como *Angular* y *Vue.js*, los cuales son buenas opciones para el desarrollo frontend, sin embargo, fue seleccionado *React* por la practicidad y el conocimiento previo adquirido. *React* está basado en una librería de *javascript* que brinda las herramientas necesarias para desarrollar una aplicación web. Ofrece una serie de ventajas por las cuales ha sido elegida como *"framework"* para desarrollar las interfaces de usuario. Su eficiencia y rapidez al "renderizar" (proceso de actualizar la interfaz de usuario en función de los cambios en los datos o en el estado del componente), así como su enfoque basado en componentes y la reutilización de los mismos, permiten el desarrollo de aplicaciones web escalables y mantenibles. Además, *React* cuenta con una gran comunidad de desarrolladores, una amplia variedad de recursos en línea, y la posibilidad de integración con otras herramientas y tecnologías [51].
- **Modelo de encriptación**, fue conveniente utilizar el *SHA256*, el cual es un método de encriptación utilizado para proteger la información y garantizar que solo pueda ser accedida por las personas autorizadas. La función de hash SHA256 es ampliamente utilizada en la criptografía y en la seguridad de la información para verificar la integridad y autenticidad de los datos, así como para encriptar contraseñas y otros datos sensibles, esta función criptográfica toma una entrada y devuelve un valor de hash de 256 bits único e irreproducible. La función hash es determinista, lo que significa que siempre producirá el mismo valor de hash para una entrada dada. Además, la función es unidireccional, lo que significa que no es posible obtener la entrada original a partir del valor de hash [52].
- **Solidity**, es un lenguaje de programación de alto nivel que permite desarrollar contratos inteligentes en la plataforma Ethereum. Fue seleccionado ya que es el principal lenguaje de programación utilizado para el desarrollo de contratos inteligentes, debido a su facilidad de uso, su popularidad y la capacidad de desplegar contratos de manera segura, rápida y eficiente, tanto en la red de principal de Ethereum como en redes de prueba [53].

- **Ether.js**, es una biblioteca completa y compacta para interactuar con *Ethereum Blockchain* y su ecosistema, fue seleccionada por su interfaz de programación de aplicaciones intuitiva para trabajar con contratos inteligentes, además de que es ideal para trabajar con contratos inteligentes [54].
- **Spring Boot**, es un proyecto de código abierto basado en Spring *framework*, fue creado para facilitar la creación de aplicaciones java, reduciendo la complejidad de la configuración [55]. Fue escogido por los conocimientos adquiridos previamente sobre esta herramienta, específicamente sobre la creación de *APIs* de *Representational State Transfer* (REST, Transferencia de Estado Representacional).
- Una *API* es un conjunto de protocolos, rutinas y herramientas que son utilizados para interactuar con un software o plataforma. Básicamente, una *API* define la forma en que otros programas pueden comunicarse con el software o plataforma para acceder a sus funciones y datos [56].
- **PostgreSQL**, es un sistema de base de datos relacional de código abierto que soporta gran parte del estándar *Structured Querry Language (SQL)*, sus características principales por las que fue escogido para el proyecto fueron su flexibilidad y compatibilidad con estándares técnicos [57].

Capítulo 4.

4. Sistema de vehículo compartido (carpooling)

El presente capítulo expone las contribuciones de investigación derivadas de este proyecto. Está estructurado en tres subsecciones que examinan detalladamente cada una de los principales aportes.

La subsección 4.1, contiene los componentes del módulo de registro y autenticación basado en *Blockchain.* La subsección 4.2, detalla el método de autenticación seleccionado para el desarrollo del proyecto. La subsección 4.2 aborda específicamente el método de encriptación implementado en el prototipo, ofreciendo una perspectiva detallada sobre su aplicación y funcionalidad. Finalmente, la subsección 4.4 destaca el empleo de la arquitectura ITS, ARC-IT, dentro del marco de este proyecto, resaltando su importancia y relevancia en la implementación exitosa de la propuesta investigativa.

4.1. Componentes del módulo de registro y autenticación

El diagrama de componentes es una herramienta visual que modela la estructura de un sistema de software en términos de componentes e interfaces. Permite representar cómo los diferentes módulos y componentes del sistema interactúan entre sí para cumplir con los objetivos del sistema [58].

Para realizar el diagrama de componentes fue utilizado un diagrama de Lenguaje de Modelado Unificado (UML), ya que es un lenguaje estándar para modelar sistemas orientados a objetos y es ampliamente utilizado en la industria del software. La Figura 2 exhibe el diagrama de componentes propuesto, el cual presenta una aproximación de los componentes utilizados y su funcionalidad dentro de un proceso de autenticación con *Blockchain*. Es importante destacar que este módulo inicial, está enfocado en realizar el proceso de registro y autenticación, de cualquier tipo de aplicación, no solamente una aplicación relacionada con movilidad. Por tal motivo, en esta etapa del desarrollo del prototipo, aún no fue considerado lo propuesto por la arquitectura ITS denominada ARC-IT (Figura 1, que fue la arquitectura internacional con el servicio más aproximado a lo requerido), porque de otra forma el módulo

quedaría desarrollado exclusivamente para aplicaciones de tipo ITS, y el objetivo es crear un módulo general que sea fácilmente adaptable a cualquier entorno de aplicación.

Fueron encontrados diferentes artículos con distintas propuestas de componentes utilizados. La propuesta del presente proyecto está basada en el artículo *BlockAPP: Using Blockchain for Authentication and Privacy Preservation in IoV,* en donde los usuarios realizan un proceso de registro asociado a un servidor de autenticación, para luego hacer una petición a la *Blockchain* y que esta almacene la información. Por otro lado, la autenticación de un usuario es realizada por medio de un proveedor de servicio, el cual verifica la información con ayuda de la *Blockchain* [59].

Como indica el párrafo anterior, el artículo muestra que los dos procesos (registro y autenticación) son realizados de forma independiente, utilizando cada uno componentes diferentes. En el presente proyecto considera que los componentes en el proceso de registro y autenticación pueden ser los mismos con el fin de ahorrar recursos, tiempos de ejecución y respuestas en la aplicación. Los componentes de la aplicación fueron definidos, lo cual permitió dividir su lógica en partes más pequeñas. Cada componente desempeña una tarea específica que puede reutilizarse en diferentes partes de código. Los componentes identificados para la aplicación fueron los siguientes:

- ***Aplicación Descentralizada.*** La aplicación descentralizada (*DApp*) se compone de la base de datos *Blockchain* y del contrato inteligente (*Smart Contract*).
- **Base de datos *Blockchain.*** Almacena los datos biométricos del usuario de manera segura en una *Blockchain*.
- ***Smart Contract*.** El contrato inteligente es el programa que tiene la responsabilidad obtener y escribir información en la base de datos Blockchain.
- **Base de datos *SQL.*** Usada para almacenar los datos de los usuarios no biométricos.
- **App de autenticación.** Este componente contiene la lógica de la aplicación referente al registro y autenticación del usuario.

- **Cámara.** Es la herramienta utilizada en la app de autenticación y la interfaz de usuario. Por medio de la cámara un puede autenticarse o registrarse.
- **Interfaz de usuario.** Es la vista que tiene el usuario de la aplicación web, en particular las vistas de inicio, autenticación y registro.

La Figura 2 presenta la interacción entre los componentes mencionados.

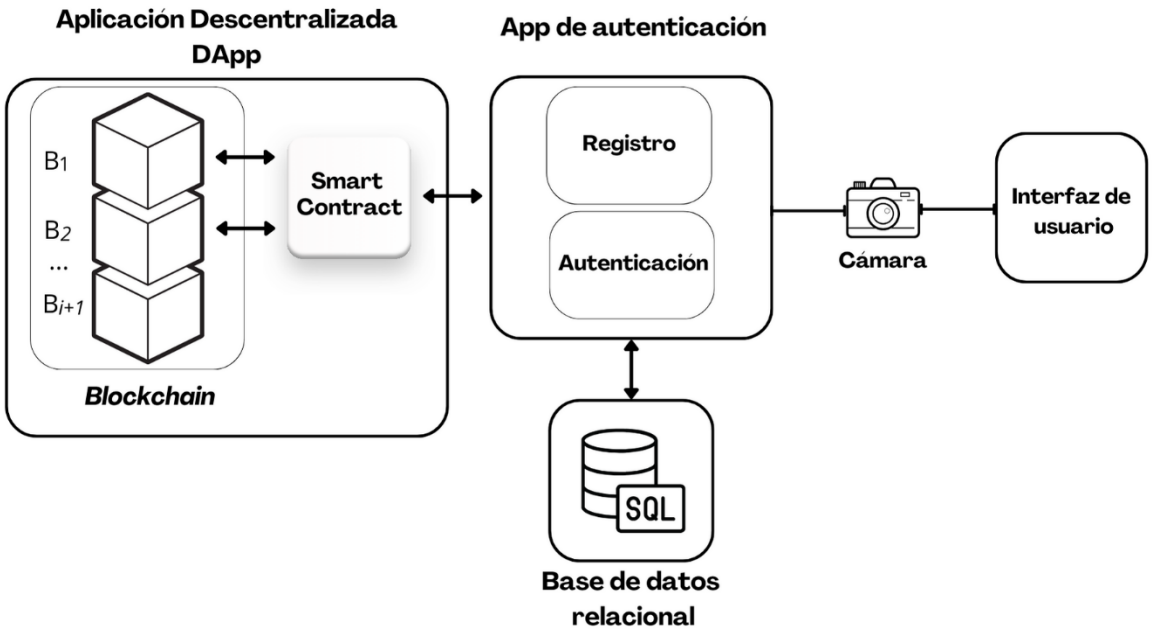


Figura 2. Diagrama de componentes UML del modelo de registro y autenticación propuesto.

4.2. Método de autenticación seleccionado

La revisión de literatura realizada, explicada en el capítulo 3, permitió identificar que la mayoría de los prototipos de sistemas de gestión de acceso remoto utilizan autenticación biométrica basada en huellas dactilares. Sin embargo, esta técnica presenta algunas limitaciones, como la necesidad de que el usuario tenga las manos limpias y secas para que la identificación sea exitosa, en algunos casos, las huellas dactilares son susceptibles al desgaste, en donde pueden cambiar con el tiempo debido a factores como el envejecimiento, lesiones o trabajos manuales [60].

Por ello, el proyecto propuso utilizar una técnica de autenticación biométrica basada en reconocimiento facial. Esta técnica tiene la ventaja de que no requiere que el usuario tenga contacto físico con el dispositivo, lo que la hace más segura y conveniente.

Adicionalmente, el articulo *A Review on Authentication Methods* [61] afirma que el uso de autenticación multifactor y el uso de autenticación biométrica son dos tendencias importantes en el desarrollo de métodos de autenticación.

En el caso de la autenticación multifactor (MFA), como su nombre lo dice, combina dos o más métodos de autenticación para proporcionar un mayor nivel de seguridad. Por ejemplo, un usuario puede usar una contraseña y un token para iniciar sesión en un sistema. MFA es más difícil de piratear que los métodos de autenticación de un solo factor, lo que hace que el sistema sea más robusto y seguro [61]. Para el caso del prototipo del presente documento, fue seleccionada la autenticación multifactor, integrando la autenticación biométrica a partir de datos faciales y un PIN de seguridad de más de 4 dígitos para brindar mayor seguridad a la aplicación.

El desarrollo de esta técnica de autenticación es un aporte significativo al estado del arte en sistemas de gestión de acceso remoto. El proyecto, por lo tanto, contribuye al avance de la investigación en esta área.

4.3. Método de encriptación

En relación con la encriptación, la elección de emplear el algoritmo de hash *SHA-256* para cifrar la información biométrica suministrada por el *API* de *FaceIO* está fundamentada en su robustez y amplia aceptación en el ámbito de la seguridad informática. Esta decisión surge del hecho de que, si bien *FaceIO* proporciona un *API* externo seguro, la seguridad de la información biométrica que suministra no está garantizada. Por ende, la decisión fue cifrar dicha información antes de almacenarla en la cadena de bloques.

SHA256, perteneciente a la familia de algoritmos *SHA-2*, destaca por su resistencia a colisiones y su capacidad para generar un hash único de 256 bits, asegurando así la integridad y autenticidad de los datos. La implementación de *SHA-256* en el contrato inteligente proporciona una capa adicional de seguridad al aplicar esta función hash a los datos biométricos previos a su almacenamiento en la cadena de bloques. Esta medida previene posibles ataques maliciosos al garantizar que cualquier intento de alterar la información almacenada sea detectado [62].

La principal ventaja de utilizar *SHA-256* para cifrar la información biométrica de los usuarios reside en su capacidad para generar un resumen único e irreversible de los datos, dificultando la reversión del proceso de hash y garantizando la privacidad del usuario. Además, la implementación de *SHA-256* facilita la comparación eficiente de identificadores biométricos sin necesidad de almacenar información sensible directamente en la cadena de bloques. Esta estrategia resulta fundamental para cumplir con estándares de privacidad y regulaciones asociadas con la protección de datos biométricos [62].

4.4. Arquitectura ARC-IT

Respecto a las dos arquitecturas ITS investigadas, fue seleccionada la arquitectura americana *ARC-IT* por su similitud en cuanto a funcionamiento respecto al prototipo propuesto.

En primer lugar, cabe aclarar que la arquitectura de la Figura 1 representa el funcionamiento de un servicio completamente centralizado y de la intervención de diferentes actores en él.

La Figura 1 cuenta con componentes como el centro de gestión de tránsito, el centro de información de tránsito y el operador SUTC, que hacen que el sistema tenga más verificaciones, por lo cual la privacidad de la información sea reducida. Por ello, en la propuesta los objetos mencionados anteriormente no son tenidos en cuenta.

Como segunda medida, el sistema propuesto incorpora funcionalidades avanzadas como la autenticación basada en *Blockchain* y la gestión de datos biométricos, lo cual es considerado en la arquitectura seleccionada de base. Por ello, son necesarios algunos ajustes en la arquitectura mostrada en la Figura 1 para cumplir con las necesidades del proyecto. Los ajustes fueron los siguientes:

1. Fue agregado un nuevo componente físico, denominado *Blockchain Service* (Servicio *Blockchain*), que proporciona la funcionalidad de *Blockchain* al sistema. Este componente está encargado de almacenar la información biométrica de los pasajeros y conductores.

2. El prototipo propuesto no cuenta con sistemas de pagos, por lo que los componentes relacionados a esta tarea no son tenidos en cuenta.

Los demás componentes fueron ubicados en los elementos físicos existentes de la arquitectura ARC-IT según la correspondencia:

- Componente *React*. *Personal Information Device* (Dispositivo de información personal) que es el componente usado para realizar el *Frontend* de la aplicación web para interactuar con los usuarios finales.
- Componente *Backend java*. *Shared Use Transportation Center* (Centro de transporte de uso compartido) que es el componente encargado de gestionar los viajes, y servicios de transporte. Proporciona funciones de operación, información, planificación y gestión de transporte compartido.

Los ajustes realizados a la arquitectura ARC-IT amplían las capacidades de la misma para satisfacer las necesidades específicas del proyecto. Los componentes añadidos logran integrarse de manera coherente con los elementos existentes de la arquitectura ARC-IT, sin comprometer su integridad.

La Figura 35 muestra los componentes físicos mencionados anteriormente que participan en la propuesta del presente proyecto dentro de la arquitectura seleccionada ARC-IT. En la Figura mencionada cuenta con dos actores, pasajero y conductor, los cuales interactúan con el dispositivo de información personal (presentado en la figura con fondo amarillo, tal como sugiere la Arquitectura ARC-IT), el cual es equivalente al componente *React* del prototipo de *Carpooling* que tiene comunicación con el componente Java y el servicio *Blockchain*. El componente Java que tiene un comportamiento semejante al centro de transporte de uso compartido en la arquitectura americana (presentado en la figura con fondo verde, tal como sugiere la arquitectura de referencia) y, el componente del servicio *Blockchain* el cual es un componente nuevo respecto a la Figura 1, pero es un componente fundamental para el prototipo (este último componente representado con un color diferente a los utilizados en la arquitectura ARC-IT).

En la Figura 35 visualiza también, el flujo de información entre los diversos elementos mencionados en el párrafo precedente. Este flujo de información, representado por líneas rojas y negras, se extrajo de la Figura 1. Asimismo, el recién incorporado componente de la aplicación, denominado Servicio Blockchain, exhibe su propio

esquema de flujo de información en interacción con el dispositivo de información personal, de manera análoga a lo descrito anteriormente.

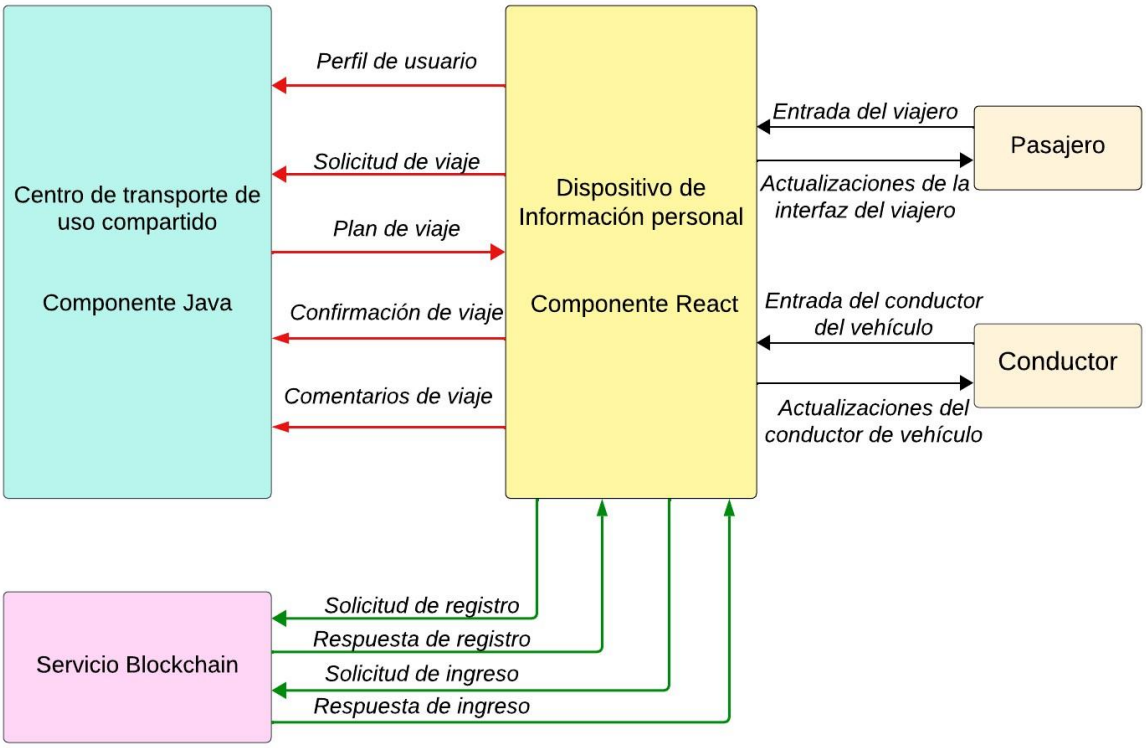


Figura 3. Componentes de prototipo basados en la arquitectura ARC-IT.

La Figura 4, refleja las funcionalidades de los usuarios de la aplicación *carpooling*. Cada funcionalidad parte de una investigación de literatura realizada mostrada a continuación.

En el artículo *An Android based application: Cab pooling* [63], los autores realizan una aplicación de *carpooling* que cuenta con un usuario y un administrador. Entre las funcionalidades del usuario está permitir gestionar el perfil, en donde el usuario puede modificar sus datos y recuperar su contraseña, y gestionar reservas en donde el usuario selecciona los taxis de acuerdo con sus requerimientos. Para el usuario administrador, su funcionalidad principal es gestionar taxis, en donde el administrador actualiza la información del vehículo. Cuando el usuario encuentra una ruta conveniente, puede enviar un mensaje con la información correspondiente del viaje,

el cual es enviado a la policía y a los contactos elegidos por el usuario si está en peligro [63].

El siguiente artículo encontrado fue *PoliUniPool: a carpooling system for universities* [76], que propone un sistema de transporte compartido entre las universidades *Università Statale* y *Politecnico di Milano*, el cual es un sistema restringido al uso de estudiantes, administrativos y docentes dentro del campus universitario o a las estaciones de transporte como tren o metro. Permite a los usuarios pasajeros proporcionar el horario esperado de sus viajes y ver la información de los mismos, ya sean cambios o retrasos en el servicio. De igual manera, Dentro de sus funcionalidades también están estimar un costo por persona, comunicación entre conductores y pasajeros, y también incluir las personas con las quiere o no viajar. Para los conductores, permite postular su ruta, indicando origen, destino y días en los que va a compartir el servicio, así como la disponibilidad y capacidad del automóvil [64].

El artículo *Car Pool'up – Real-time Carpooling using GPS* [65], los autores proponen un sistema *carpooling* en donde los usuarios pueden reservar su viaje con una persona que viaja en su misma ruta. Entre sus funcionalidades están: los usuarios pueden registrarse por medio del sitio web de la aplicación, obteniendo información del automóvil más cercano a su ubicación; tiene servicio de *schedule drive* en donde el usuario puede reservar su viaje.

Luego de realizar la reserva, puede verificar el horario del automóvil reservado y rastrear su ubicación; el usuario puede publicar su oferta económica para que los usuarios interesados puedan verla; permite al usuario consultar las tarifas a un destino en particular; y es posible consultar los perfiles de las personas con las que comparte el vehículo por motivos de seguridad.

El artículo *A Permissioned Blockchain-Based Identity Management and User Authentication Scheme for E-Health Systems* [66], propone un sistema de autenticación y gestión de identidad para sistemas de salud electrónicos (*e-health*) basado en *Blockchain* que trata de solventar los problemas de seguridad y

privacidad que introducen los sistemas de información en línea. El artículo propone un sistema de autenticación a través de datos biométricos, los cuales son almacenados en la *Blockchain.*

El artículo *FAME: Face Authentication for Mobile Encounter* [67], describe una aplicación desarrollada denominada FAME para dispositivos móviles, la cual proporciona identificación y verificación. La aplicación incluye la gestión de identidad para respaldar actividades sociales, por ejemplo, encontrar personas similares en una red social. Estas funcionalidades son implementadas teniendo en cuenta la privacidad y la seguridad.

La implementación de autenticación biométrica ofrece una opción de seguridad necesaria para la aceptación y adopción de una aplicación por parte de los usuarios, tal como presenta el artículo *A Permissioned Blockchain-Based Identity Management and User Authentication Scheme for E-Health Systems.* Por otra parte, la facilidad de uso y comodidad que brinda la autenticación facial, en comparación con otros métodos, la hace una opción interesante a considerar para el proyecto de investigación. El artículo "*FAME: Face Authentication for Mobile Encounters*" presenta varios aspectos interesantes que justifican el uso de una autenticación de tipo facial. Este tipo de autenticación, al aprovechar la tecnología biométrica, ofrece una alta precisión y robustez en la verificación de identidad. Debido a lo anterior, la implementación de la autenticación facial resulta ser la opción más adecuada, como menciona el capítulo anterior.

Aunque los tres primeros artículos mencionados anteriormente ([65-67]) están enfocados de manera distinta en implementaciones relacionadas al transporte compartido, tienen algunas funcionalidades en común, tales como la programación de viajes que es indispensable en este tipo de servicios. El primer artículo ([63]) está dirigido a la seguridad de los usuarios pasajeros al momento de usar la aplicación, ya que su aporte principal es el envío de mensajes de alerta a contactos y policía en caso de que el usuario esté en peligro. En este trabajo, los pasajeros son los que tienen el control de decidir sobre el vehículo y las características de este de acuerdo con sus requerimientos.

El segundo artículo ([64]) es más específico en cuanto a los usuarios a los que va dirigido, ya que fue diseñado para dos universidades específicamente. Al igual que en el artículo anterior, este permite programar un conjunto de viajes a un usuario pasajero, pero dicho usuario solo envía su solicitud, es el conductor quien ingresa su ruta, su hora de salida y también puede decidir, al igual que el usuario pasajero respecto a las personas con las desea compartir su viaje.

Finalmente, del tercer artículo mencionado en esta subsección ([65]), cabe destacar el seguimiento que un pasajero puede realizar al vehículo antes del viaje. Por lo cual, la ubicación de los vehículos participantes en esta aplicación siempre está comprometida, sin brindar mucha seguridad al conductor a excepción de poder observar los perfiles de las personas con las que compartirá el vehículo.

Con respecto a los artículos mencionados, la funcionalidad principal seleccionada para la aplicación de *carpooling* de esta investigación, la programación de viajes, debido a que, en una aplicación de este tipo, los usuarios deben poder organizar y planificar de manera eficiente sus viajes compartidos, decidiendo horarios, confirmación, recordatorios y otros ajustes. Además, es conveniente darle más libertad a los conductores para que decidan el viaje que quieren realizar, las personas a las que desean transportar, y permitirle calificarlas al terminar el viaje.

Para el usuario pasajero, la funcionalidad de postular la ruta que desea y calificar al conductor luego de un viaje fue incluida, con el fin de que otros usuarios puedan observar las calificaciones entregadas y que dicha calificación funcione como un parámetro de selección. En la siguiente subsección, la funcionalidad de la aplicación propuesta es explicada en detalle.

Cabe resaltar que la funcionalidad de la aplicación de *carpooling,* en este proyecto, es limitada, ya que el objetivo principal de la aplicación es **añadir seguridad de los usuarios haciendo uso de la aplicación**, más no prestar un servicio con una amplia gama de funcionalidades ya que esto incrementa el alcance del proyecto considerablemente. A continuación, son mostradas las funcionalidades seleccionadas.

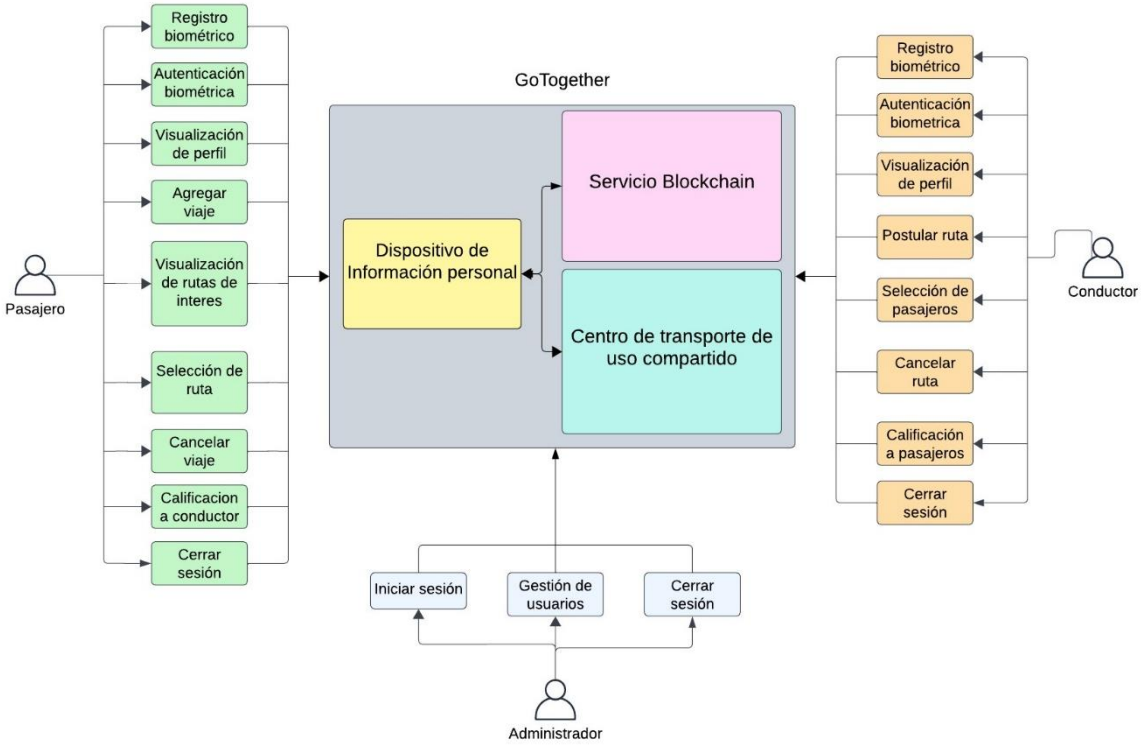


Figura 4. Funcionalidades de la aplicación *carpooling*.

Capítulo 5.

5. Aplicación de autenticación con *Blockchain*

5.1. Introducción

Este capítulo describe la implementación de un módulo de autenticación basado en *Blockchain* mediante el marco de trabajo *Scrum*. Este módulo es crucial para el prototipo a implementar. En consecuencia, el equipo lo diseñó e implementó de manera independiente para luego adaptarlo a su aplicación final, que está relacionada con el uso en transporte compartido (*carpooling*). El documento detalla los componentes principales del sistema y las HU necesarias para el desarrollo del módulo de autenticación y registro. También proporciona la documentación del proceso de desarrollo del módulo. El objetivo es implementar una solución de autenticación y registro segura y descentralizada que utilice la tecnología Blockchain y otras herramientas relevantes.

Este capítulo cumple con el objetivo estratégico de implementar un módulo de autenticación basado en *Blockchain*. En este sistema, los usuarios pueden autenticarse y registrarse utilizando información biométrica proporcionada a la aplicación. Un contrato inteligente gestiona esta información, actuando como intermediario entre la interfaz de la aplicación y la plataforma *Blockchain*. Este capítulo cumple con el objetivo estratégico dos de este trabajo, relacionado con la generación de una base de conocimiento sobre la aplicación de la tecnología *Blockchain*, que servirá de apoyo a futuras investigaciones en esta temática.

5.2. Desarrollo de un prototipo inicial de la aplicación web

5.2.1. HU del módulo de registro y autenticación

Las HU que tendría el módulo de autenticación son identificadas a partir del diagrama de componentes.

Las HU son una explicación general e informal de una función del software desarrollada desde la perspectiva del usuario final o cliente, en un lenguaje sencillo [60]. Cabe mencionar, que quien diligencia las HU no son únicamente los clientes o

usuarios finales externos, sino que también pueden ser clientes internos o colegas dentro del equipo a cargo del software. Las HU son herramientas valiosas dentro de los marcos ágiles como *Scrum*, ya que estas pueden añadirse a los *Sprints* y desarrollar a lo largo del *Sprint* [68].

Las HU suelen ir acompañadas de diseños de vistas o *"mockups"*, pues estas son importantes en el desarrollo de un prototipo de un módulo, ya que definen la apariencia visual y la interacción de los usuarios con la aplicación. En otras palabras, el diseño de vistas está enfocado en la presentación de la información al usuario y en la interacción de este con la interfaz [69].

La importancia del diseño de vistas radica en que tiene un impacto directo en la experiencia del usuario y en la percepción que este tenga de la calidad y la utilidad de la aplicación. Un buen diseño de vistas puede hacer que la aplicación sea más fácil de usar, más intuitiva y más atractiva visualmente, lo que brinda mayor satisfacción al usuario y en un mayor éxito del proyecto [69]. Para el módulo de autenticación basado en Blockchain, fueron identificadas dos HU: registro de usuario e inicio de sesión de usuario. Las tablas 2 y 3 detallan estas HU.

5.2.1.1 Historia de usuario registro de usuario

ID HISTORIA	HU1
NOMBRE	Registro de usuario
DESCRIPCIÓN	**Como** usuario **Quiero** registrar mis datos personales (nombre, cédula, correo) y biométricos (reconocimiento facial por medio de la cámara web) **Para** guardar esta información y después utilizarla para iniciar sesión.
CRITERIOS DE ACEPTACIÓN	**Escenario 1:** Registro exitoso **Dado** que los datos biométricos no han sido registrados anteriormente **Cuando** el formulario de registro ha sido completado **Entonces** es mostrado un mensaje de registro finalizado. **Escenario 2:** Registro fallido **Dado** que los datos biométricos ya han sido registrados **Cuando** el formulario de registro ha sido diligenciado **Entonces** es mostrado un mensaje de usuario ya registrado.
PRIORIDAD	1
ESTIMACIÓN	20 días

Tabla 2. Historia de usuario registro de usuario.

5.2.1.2. Historia de usuario iniciar sesión

ID HISTORIA	HU2
NOMBRE	Iniciar sesión
DESCRIPCIÓN	**Como** usuario **Quiero** iniciar sesión **Para** acceder a la aplicación web
CRITERIOS DE ACEPTACIÓN	**Escenario 1:** Inicio de sesión exitoso **Dado** que la aplicación detecta similitudes faciales **Cuando** usa la autenticación facial **Entonces** es mostrado un mensaje de mensaje inicio de sesión exitoso **Escenario 2:** Inicio de sesión fallido **Dado** que la aplicación no detecta similitudes faciales **Cuando** usa la autenticación facial **Entonces** es mostrado un mensaje de error al iniciar sesión
PRIORIDAD	2
ESTIMACIÓN	20 días

Tabla 3. Historia de usuario inicio de sesión de usuario.

A continuación, es presentado el diseño inicial de las vistas de cada una de las HU identificadas.

5.2.2. Vistas de las HU identificadas

5.2.2.1. Diseño de vista inicial (inicio de las HU1 y HU2)

La Figura 5 exhibe un diseño de la primera vista de la aplicación web, específicamente del proceso de autenticación, puede observarse el botón de inicio de sesión y un enlace para registrarse.

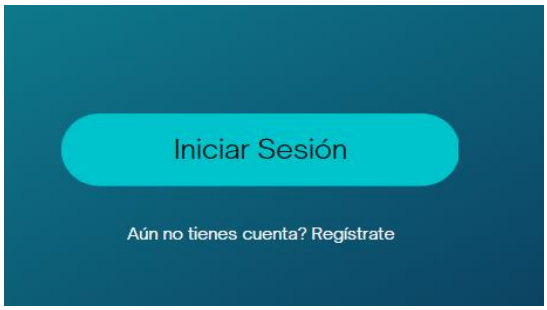


Figura 5. Inicio de la aplicación.

5.2.2.2. Diseño de vista 1 de la HU1 (inicio del registro de usuario)

Al ingresar al enlace “registrarse”, aparece un formulario en el que el usuario debe proporcionar información básica, como nombre y correo electrónico, para iniciar el

proceso de registro, como indica la Figura 6. Los botones tienen fondo azul y los campos a diligenciar tienen fondo blanco. Además, solicita tomar una foto del rostro de la persona.

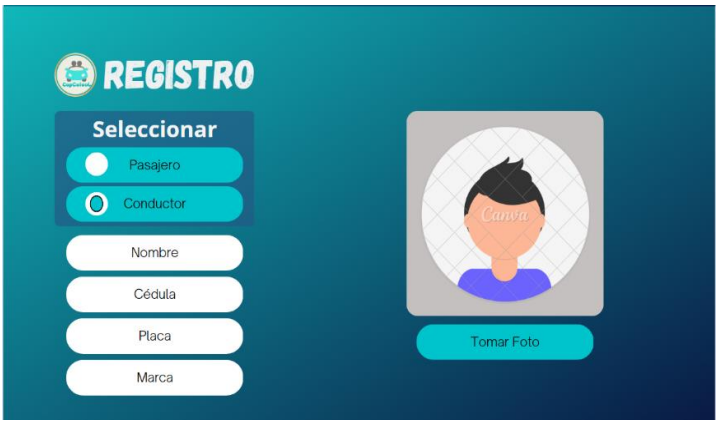


Figura 6. Vista registro de usuario.

El formulario de registro puede generar dos respuestas: una si el formulario no ha sido diligenciado correctamente, y otra si el registro fue exitoso. Estos escenarios son descritos a continuación.

5.2.2.3. Diseño de vista 2 de la HU1 (registro exitoso de usuario)

Un registro exitoso sucede cuando el formulario es diligenciado correctamente y en la foto tomada el rostro del usuario es identificado correctamente. Si es así, aparece un pequeño recuadro con la información brindada por el usuario, dicha información es la almacenada, como presenta la Figura 7.

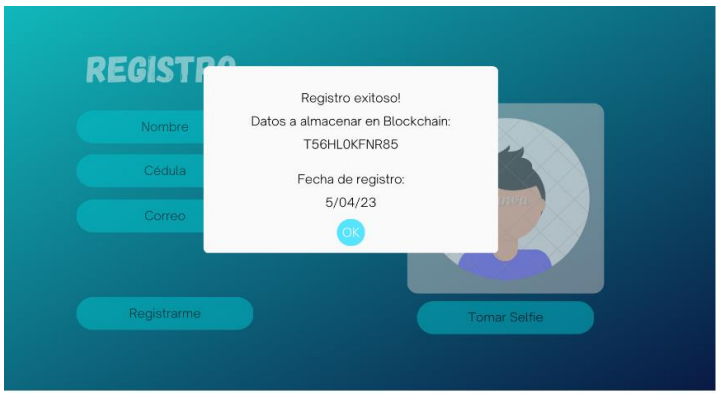


Figura 7. Vista registro exitoso.

5.2.2.4. Diseño de vista 3 de la HU1 (registro fallido de usuario)

Cuando el formulario no es correctamente diligenciado, el campo que falta por llenar es resaltado. Si la foto brindada no es clara, aparece un mensaje de error en la vista, indicando que el registro fue erróneo, tal como muestra la Figura 8.

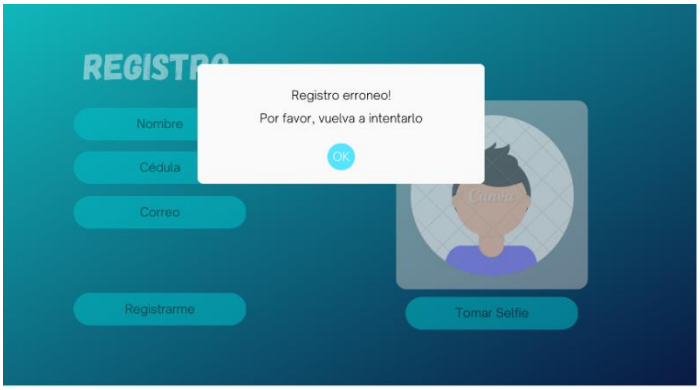


Figura 8. Vista registro fallido.

5.2.2.5. Diseño de vista 1 de la HU2 (inicio de sesión de usuario)

En caso de indicar el inicio de sesión en el diseño de vista inicial (presentado en 5.2.2.1) la cámara del dispositivo es desplegada para hacer un reconocimiento biométrico como muestra en Figura 9.

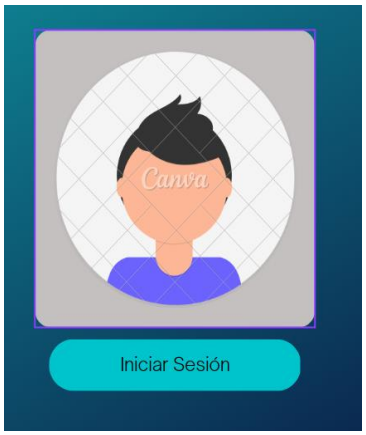


Figura 9. Vista de inicio de sesión.

Al intentar iniciar sesión, existen dos opciones, un inicio de sesión fallido o un inicio de sesión exitoso, los cuales son detallados en las siguientes subsecciones.

5.2.2.6. Diseño de vista 2 de la HU2 (inicio de sesión de usuario fallido)

La Figura 10 muestra el inicio de sesión fallido, cuando la aplicación produce un error al reconocer el rostro o si no encuentra similitudes faciales con un usuario registrado.

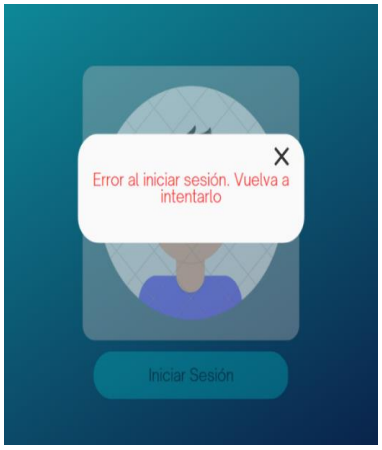


Figura 10. Vista de inicio de sesión de usuario fallido.

5.2.2.7. Diseño de vista 3 de la HU2 (inicio de sesión de usuario exitoso)

Un inicio de sesión correcto ocurre cuando la aplicación determina que el rostro del usuario coincide con el rostro almacenado, La Figura 11 muestra el resultado de un inicio de sesión exitoso.

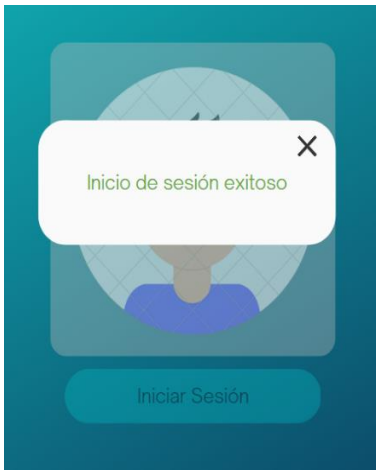


Figura 11. Vista de inicio de sesión exitoso.

5.3. Documentación del proceso de desarrollo del módulo de registro y autenticación

5.3.1. Implementación de la *Blockchain* a usar

En primer lugar, implementar una *Blockchain* de forma completa, ampliará significativamente el alcance del proyecto. Esto debido a que el proceso de diseñar y desarrollar una *Blockchain* completamente nueva requiere una gran cantidad de tiempo, esfuerzo y recursos para investigar, crear y optimizar la tecnología. Además, implica la necesidad de garantizar la seguridad de la red y la prevención de ataques

malintencionados, lo que añade un nivel adicional de complejidad y costo al proyecto [70].

Por lo anterior, fue considerado conveniente utilizar una *Blockchain* ya implementada (en lugar de desarrollar una nueva), aprovechando alguna opción probada y segura que hubiese sido optimizada durante años, lo que permitiría reducir significativamente los costos y los tiempos de desarrollo.

La revisión sistemática realizada previamente identificó proyectos relacionados desarrollados con redes *Blockchain*, como *Hyperledger* y *Ethereum. Ethereum* fue la *Blockchain* seleccionada, teniendo en cuenta aspectos como: seguridad, recursos disponibles, interoperabilidad y eficiencia [71].

Los aspectos más importantes de *Ethereum* son:

- Seguridad. *Ethereum* es una de las redes *Blockchain* más seguras y confiables disponibles. Los contratos inteligentes que son ejecutados en *Ethereum* están protegidos contra ataques malintencionados y vulnerabilidades de seguridad comunes [63].
- Recursos disponibles. *Ethereum* tiene una gran comunidad de desarrolladores y usuarios que están trabajando constantemente para mejorar la plataforma y crear nuevas aplicaciones y herramientas. Además, la mayoría de los contratos inteligentes desplegados en *Ethereum* son escritos en *Solidity,* que es un lenguaje de programación de alto nivel que permite desarrollar contratos inteligentes en la plataforma *Ethereum* [72].
- Interoperabilidad. *Ethereum* es compatible con una amplia gama de otros sistemas y plataformas, lo que permite integrar fácilmente el sistema de autenticación con otras aplicaciones y servicios. Esto es de gran utilidad cuando el contrato inteligente debe interactuar con otros componentes del sistema [71].
- Eficiencia. *Ethereum* es una plataforma *Blockchain* establecida y madura que ha sido probada y optimizada durante años. Como resultado, su arquitectura es altamente eficiente y escalable [71].

Luego de establecer la *Blockchain* a utilizar para el desarrollo del proyecto, fue realizado un primer prototipo de la aplicación web de autenticación biométrica utilizando la *Blockchain* de *Ethereum*. Este prototipo permite a los usuarios autenticarse y registrarse mediante características físicas únicas, en este caso el

reconocimiento facial, y almacenar dicha información biométrica en la *Blockchain* de *Ethereum*, garantizando así un alto nivel de seguridad.

5.3.2. Interfaces de Usuario

React fue el "framework" (librería de *javascript* para el desarrollo *frontend*) seleccionado para crear interfaces de usuario de forma rápida y eficiente. Fueron considerados otros *"frameworks*" como *Angular* y *Vue.js*, los cuales son buenas opciones para el desarrollo *frontend*, sin embargo, fue seleccionado *React* por la practicidad y el conocimiento previo adquirido**.**

El resultado del primer prototipo de la interfaz de usuario implementado en *React* puede observarse en las Figuras 12 y 13, siendo las vistas de "Registro" e Inicio de sesión" respectivamente.

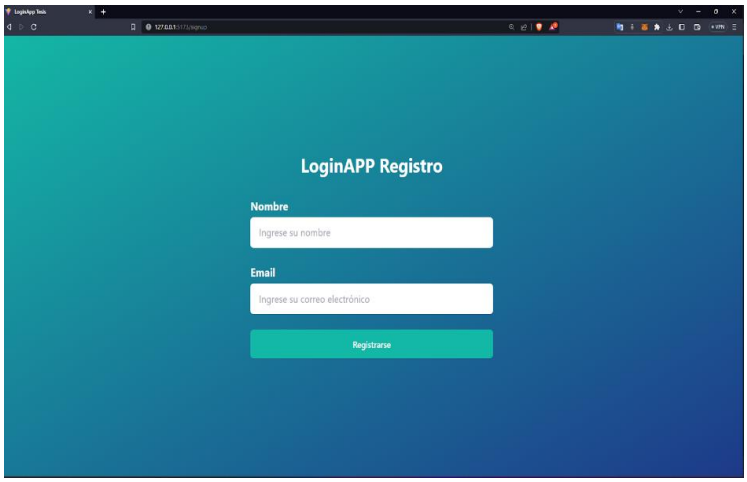


Figura 12. Interfaz de Registro de Usuario.

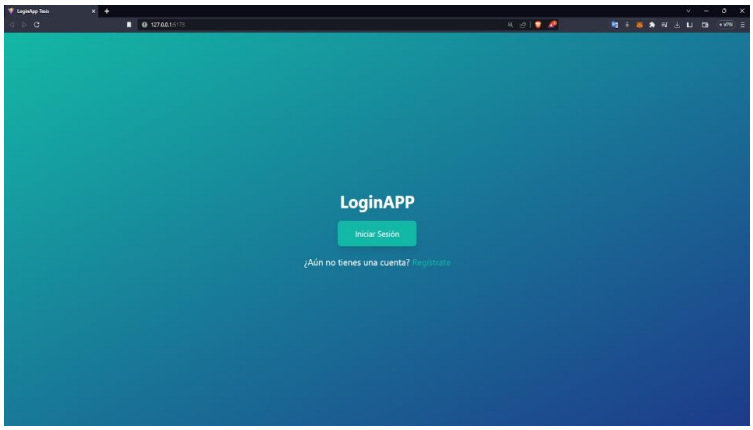


Figura 13. Interfaz de Inicio de Sesión de Usuario.

5.3.3. Toma de datos biométricos

Para capturar la información biométrica del usuario, que es usada como credencial para la autenticación, fue utilizado el *API* denominado *FaceIO*.

Específicamente, *FaceIO* proporciona una interfaz de programación de aplicaciones para interactuar con la plataforma de reconocimiento facial. Esta *API* permite a los desarrolladores integrar fácilmente la funcionalidad de reconocimiento facial en sus aplicaciones, lo que les permite identificar y autenticar a los usuarios a través de imágenes faciales [73]. *FaceIO* es un software creado por la empresa *FaceIO Inc*. que proporciona soluciones de reconocimiento facial a través de la inteligencia artificial y el aprendizaje automático.

5.3.4. Almacenamiento de los datos de usuario y encriptación

Para garantizar la seguridad y privacidad de los datos almacenados en la *Blockchain* de *Ethereum*, fue necesario desarrollar un contrato inteligente con *Solidity* para almacenar el código único generado por *FaceIO*. Para conseguir integridad y autenticidad, fue utilizado el algoritmo *SHA256*, que implementa una función de hash *SHA256* para encriptar los datos biométricos antes de almacenarlos en el contrato inteligente.

La función *hash SHA256* es una función criptográfica que toma una entrada (en este caso, el código único generado por *FaceIO*) y devuelve un valor de hash de 256 bits único e irreproducible. La función *hash* es determinista, lo que significa que siempre producirá el mismo valor de *hash* para una entrada dada. Además, la función es unidireccional, lo que significa que no es posible obtener la entrada original a partir del valor de *hash*.

La función de *hash SHA256* es ampliamente utilizada en la criptografía y en la seguridad de la información para verificar la integridad y autenticidad de los datos, así como para encriptar contraseñas y otros datos sensibles [56]. En este caso, *SHA256* es utilizado para encriptar los códigos generados por *FaceIO* antes de almacenarlos en la *Blockchain* de *Ethereum*.

5.3.5. Implementación del contrato inteligente

El desarrollo del contrato inteligente fue realizado en la plataforma de desarrollo de contratos inteligentes denominada *Hardhat*, la cual proporciona herramientas para compilar, probar y desplegar el contrato inteligente. La razón principal para utilizar

Hardhat fue la practicidad para proporcionar un entorno de desarrollo de contratos inteligentes rápido y eficiente. Además, esta plataforma ofrece herramientas integradas de auditoría de seguridad y una amplia gama de funciones que mejoran significativamente el flujo de trabajo de desarrollo de contratos inteligentes. Las otras alternativas revisadas similares a *Hardhat* fueron:*Truffle, Ganache, Remix y Brownie*. Para evitar costos adicionales asociados a la ejecución de transacciones durante la etapa de pruebas y depuración, el contrato inteligente fue elaborado inicialmente en una red de prueba. Esto ayuda a reducir errores y mejorar la seguridad de los contratos inteligentes antes de implementarlos en la red principal [74]. Las redes disponibles de *Ethereum* son: *Ropsten y Mainnet*, Fue seleccionada *Ropsten* debido a que es la red de pruebas. La red *Mainnet* es la red principal de *Ethereum*, la cual exige costos adicionales para desplegar un contrato inteligente.

Para integrar el módulo de autenticación *Blockchain* a la aplicación "*frontend"* implementada utilizando la librería *React,* fue necesario en primera medida instalar la herramienta *Hardhat*, la cual permite instalar un nodo de la *Blockchain* de *Ethereum* de manera local para ello fueron utilizados los siguientes comandos:

npx hardhat init

npm install --save-dev "hardhat@^2.18.0" "@nomicfoundation/hardhat-toolbox@^3.0.0"

El primer comando inicia un proyecto *Hardhat,* seleccionando los parámetros del proyecto con un menú de consola, como indica la Figura 14. la aplicación de *Hardhat* es ejecutada en el puerto 8545, que sirve para interactuar con el proyecto desde la aplicación desarrollada con *React*.

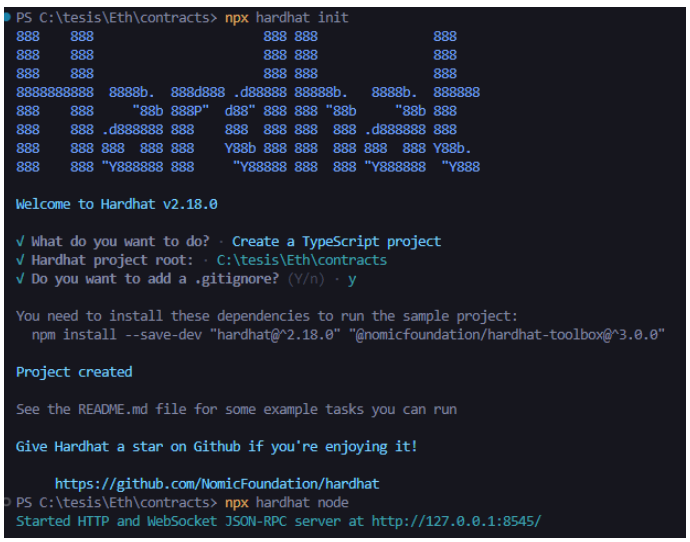
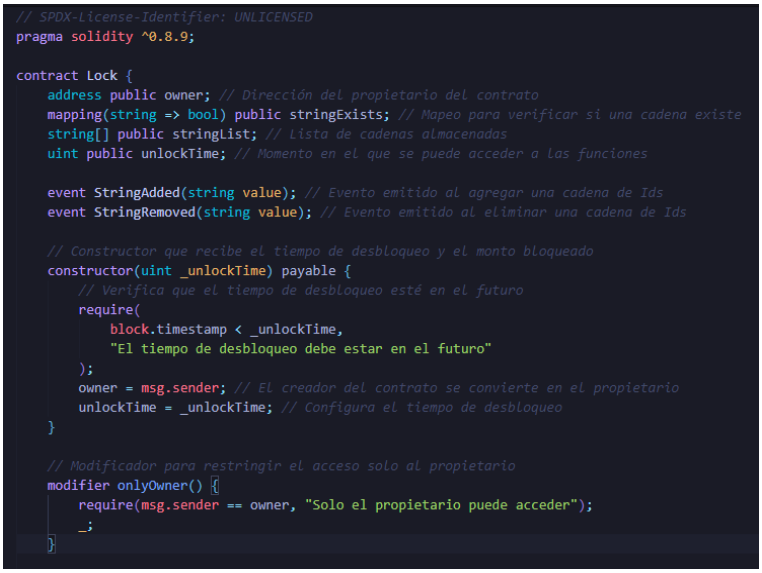


Figura 14. Creación de un proyecto en *Hardhat*.

El segundo comando agrega las herramientas necesarias para que *Hardhat* pueda compilar contratos inteligentes e instale las dependencias necesarias. Una vez creado el entorno de desarrollo, el contrato inteligente es agregado a la carpeta *Contracts,* las Figuras 15, 16 y 17 muestran la versión final del contrato inteligente.

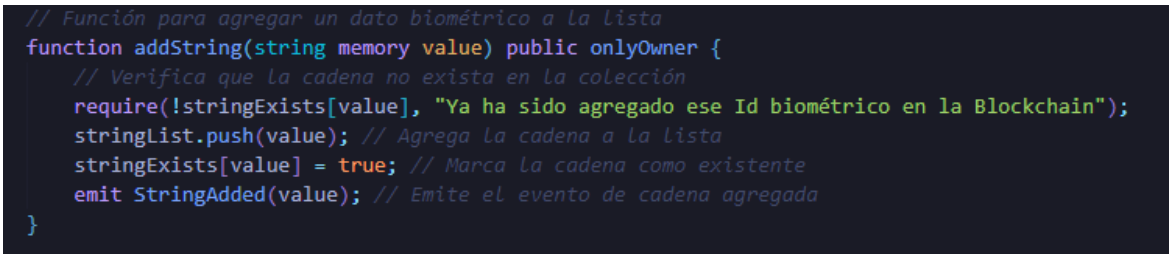


```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.9;

contract Lock {
    address public owner; // Dirección del propietario del contrato
    mapping(string => bool) public stringExists; // Mapeo para verificar si una cadena existe
    string[] public stringList; // Lista de cadenas almacenadas
    uint public unlockTime; // Momento en el que se puede acceder a las funciones

    event StringAdded(string value); // Evento emitido al agregar una cadena de Ids
    event StringRemoved(string value); // Evento emitido al eliminar una cadena de Ids

    // Constructor que recibe el tiempo de desbloqueo y el monto bloqueado
    constructor(uint _unlockTime) payable {
        // Verifica que el tiempo de desbloqueo esté en el futuro
        require(
            block.timestamp < _unlockTime,
            "El tiempo de desbloqueo debe estar en el futuro"
        );
        owner = msg.sender; // El creador del contrato se convierte en el propietario
        unlockTime = _unlockTime; // Configura el tiempo de desbloqueo
    }

    // Modificador para restringir el acceso solo al propietario
    modifier onlyOwner() {
        require(msg.sender == owner, "Solo el propietario puede acceder");
        _;
    }
```

Figura 15. Contrato Inteligente, parte 1.

La Figura 15 especifica la licencia del contrato, la versión de *Solidity* y define la dirección pública del contrato inteligente, que es la dirección donde es desplegado, en este caso de manera local. Además, establece el método constructor que configura una variable de tiempo *_unlockTime* que le indica al contrato inteligente en qué momento puede acceder a las funciones del mismo.

La Figura 16 presenta la definición de la función más importante que es denominada *addString,* la cual recibe el Id biométrico en formato *String* y verifica si este ya existe previamente, antes de ejecutar el proceso de almacenamiento en el nodo local de la *Blockchain* de *Ethereum.* Este método es ejecutado cada vez que un usuario pasajero o conductor realiza el proceso de registro en la aplicación.



```
// Función para agregar un dato biométrico a la lista
function addString(string memory value) public onlyOwner {
    // Verifica que la cadena no exista en la colección
    require(!stringExists[value], "Ya ha sido agregado ese Id biométrico en la Blockchain");
    stringList.push(value); // Agrega la cadena a la lista
    stringExists[value] = true; // Marca la cadena como existente
    emit StringAdded(value); // Emite el evento de cadena agregada
}
```

Figura 16. Contrato Inteligente, parte 2.

La Figura 17 presenta dos funciones, la función *checkString* es la más importante porque verifica la existencia de un dato biométrico en la *Blockchain* de *Ethereum.* Lo anterior posibilita que al momento en que un usuario intenta ingresar a la aplicación, esta última llame este método y verifique que este usuario efectivamente este registrado y que sus datos biométricos están almacenados en la *Blockchain* de *Ethereum.*

```
    // Función para verificar si una cadena existe en la colección
    function checkString(string memory value) public view returns (bool) {
        return stringExists[value];
    }

    // Función para obtener la lista de cadenas almacenadas
    function getStringList() public view returns (string[] memory) {
        return stringList;
    }
}
```

Figura 17. Contrato Inteligente, parte 3.

Para implementar el algoritmo de encriptación *SHA256* en el contrato inteligente y utilizar su función de *hash256*, fue importada la biblioteca *SHA256* de *OpenZeppelin* como muestra la Figura 18. Luego la función *storeCode* es implementada, que utiliza la variable *hashedCode* para almacenar el resultado de la función de *hash SHA256,* la cual es aplicada al código único generado por *FaceIO* como indica la Figura 19.

```
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/utils/cryptography/SHA256.sol";
```

Figura 18. Importación algoritmo de encriptación SHA256.

```
    function storeCode(bytes32 code) public {
        bytes32 hashedCode = sha256(code);
        _codes[msg.sender] = hashedCode;
    }
```

Figura 19. Función de encriptación.

Posteriormente el contrato inteligente es compilado con el siguiente comando:

npx hardhat compile.

Si este proceso es satisfactorio, la aplicación puede iniciarse en el nodo local con el comando *npx hardhat node,* lo cual arroja una lista de direcciones que pueden ser usadas para desplegar contratos inteligentes. Además, despliega el contrato inteligente en la dirección que fue configurado en las propiedades del código *deploy.ts.*

Es importante destacar que el archivo *deploy.ts* es un componente esencial en este proceso. Aunque es generado automáticamente al crear el proyecto de *Hardhat*, puede ser modificado para adaptarse a necesidades específicas. En este caso, la

dirección en la cual el contrato inteligente fue desplegado es definida. Algunas de estas direcciones y el resultado del despliegue del contrato son presentados en la Figura 20.

Figura 20. Despliegue de contrato inteligente.

Una vez los pasos anteriormente mencionados fueron realizados, fue necesario instalar la librería *ethers.js,* para poder interactuar con el contrato inteligente desde la aplicación elaborada con *React*. Para ello fue ejecutado el siguiente comando:

npm install ethers

A continuación, para importar la librería al código fuente fue utilizada siguiente línea:

import { ethers } from "ethers".

Finalmente, el código de los métodos de registro y autenticación es modificado para permitir la integración con *Blockchain*. La Figura 21 presenta el ejemplo para la HU de registro de un usuario, creando una instancia de la librería *ether.js* para el puerto 8545, es posible comunicarse con el nodo *Blockchain* que está ejecutándose en dicho puerto.

También es necesario especificar la dirección del contrato inteligente y la variable en formato JSON ABI (*Application Binary Interface,* es decir, Interfaz de Aplicación Binaria) que resume en formato JSON cuáles son las funciones que contiene el contrato inteligente. Finalmente, el método *addString* presente en el contrato inteligente que almacena los datos biométricos codificados en formato *String* en el nodo *Blockchain* de *Ethereum* es llamado de forma asíncrona.

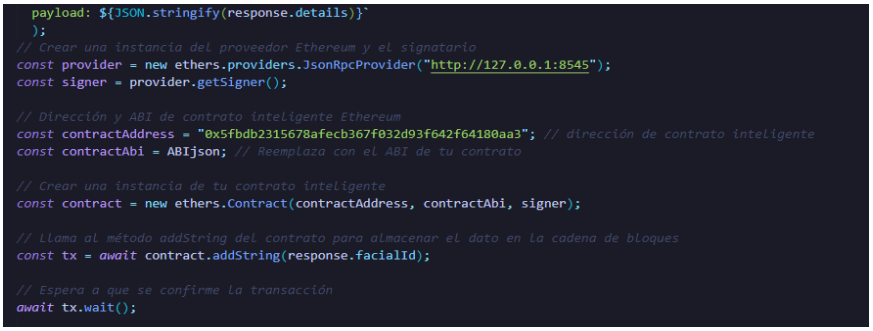


Figura 21. Llamado al contrato inteligente desde la aplicación utilizando React.

5.4. Pruebas de funcionamiento

Fueron diseñadas cuatro tipos de pruebas para la autenticación de usuario.

En la Tabla 4 son mostrados los tipos de pruebas, el resultado esperado y el resultado obtenido.

Nombre del tipo de prueba	**Resultado esperado**	**Resultado obtenido**
Registro exitoso	Mensaje de registro correcto. Redireccionamiento a la vista de inicio de sesión.	Mostró un mensaje de registro correcto y posteriormente el correcto direccionamiento a la vista de inicio de sesión.
Registro no exitoso	Mensajes de alerta para completar campos de formulario de registro.	Muestra un mensaje de alerta bajo cada campo del formulario que era obligatorio llenar.
	Si existe registrada una persona con los mismos datos, aparece una alerta y redirecciona al formulario de registro.	Mostró una alerta que indica que el usuario ya existe y posteriormente redirecciona nuevamente al formulario de registro.
Inicio de sesión exitoso	Mensaje de alerta indicando la correcta identificación. Redireccionamiento a la ventana de Bienvenida.	Mostró un mensaje indicando la identificación es exitosa y apareció la ventana de bienvenida.
Inicio de sesión no exitoso	Mensaje de inicio de sesión fallido y redireccionamiento al inicio de sesión de nuevo.	Mostró un mensaje de alerta indicando que el no inicio de sesión y posteriormente la aplicación redireccionó al inicio de sesión, para realizar el ingreso nuevamente.

Tabla 4. Pruebas de funcionamiento del módulo de autenticación.

A continuación, las pruebas realizadas son descritas detalladamente.

5.4.1. Registro exitoso

Un proceso de registro exitoso, comienza por la vista presentada en la Figura 22, en la cual, al seleccionar el enlace “Regístrate” lleva a la vista de registro de la Figura 23, en la cual, los campos correspondientes a nombre y correo electrónico deben ser completados.

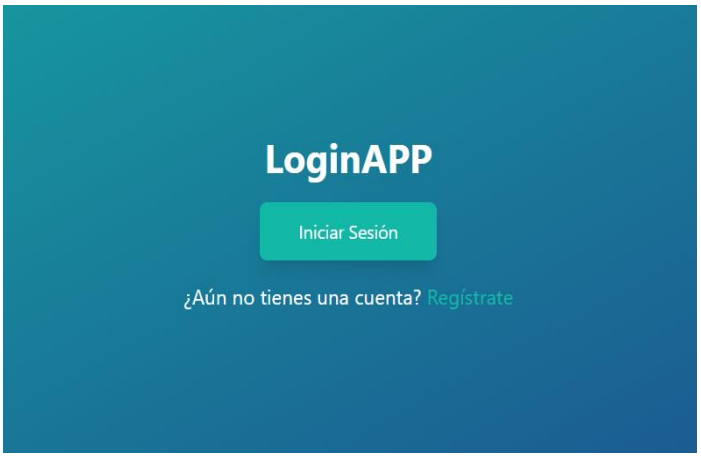


Figura 22. Vista inicial del módulo de autenticación y registro.

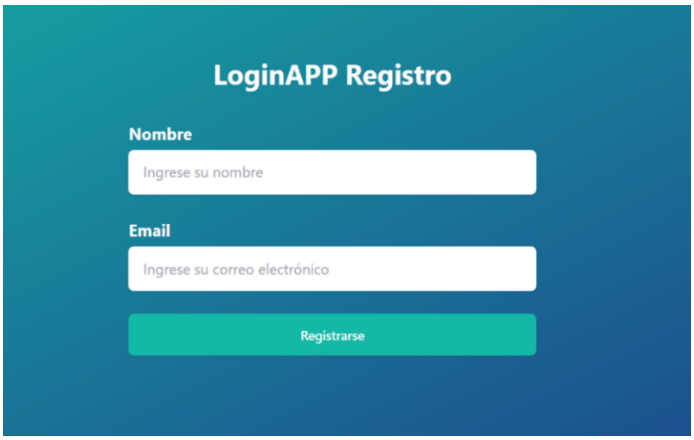


Figura 23. Vista inicial de registro.

Cuando los campos son ingresados de manera correcta y el *API de FaceIO* detecta que tanto el correo electrónico como el rostro no han sido ingresados anteriormente, una ventana emergente de alerta es desplegada en la interfaz, mostrando el nombre y el email que van a ser registrados, como lo indica Figura 24.

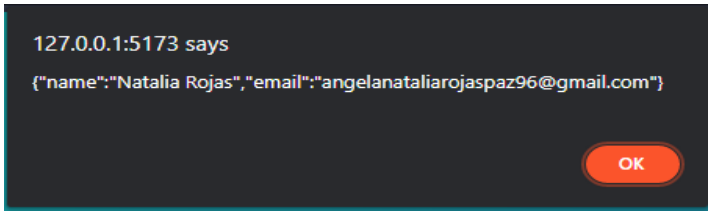


Figura 24. Ventana Emergente. Datos a registrar.

Luego, el dispositivo solicita permiso para utilizar la cámara web como presenta la Figura 25. Cuando la solicitud es aceptada, el *API* de *FaceIO* pide los datos biométricos como muestra la Figura 26, indicando al usuario que mire la cámara web y que ingrese un PIN de seguridad para completar el registro. A continuación, el usuario debe aceptar los términos y condiciones para poder usar *FaceIO*, como indica la Figura 27.

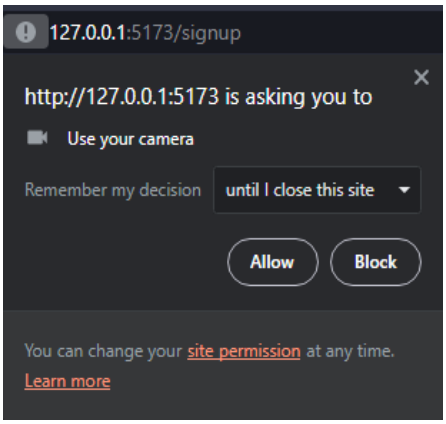


Figura 25. Solicitud de permiso para usar la cámara web del dispositivo.

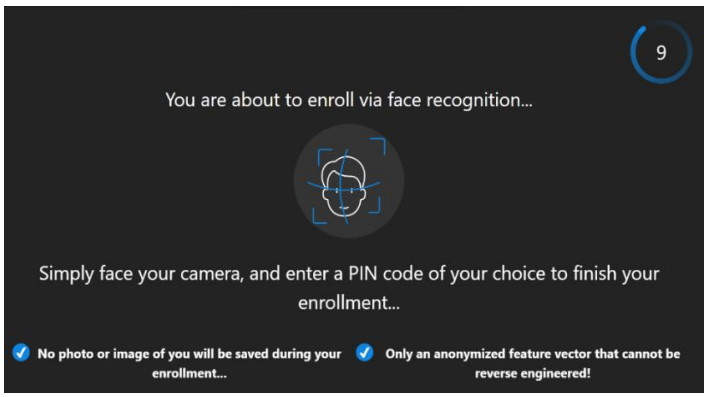


Figura 26. Ejecución Inicial de API de *FaceIO*.

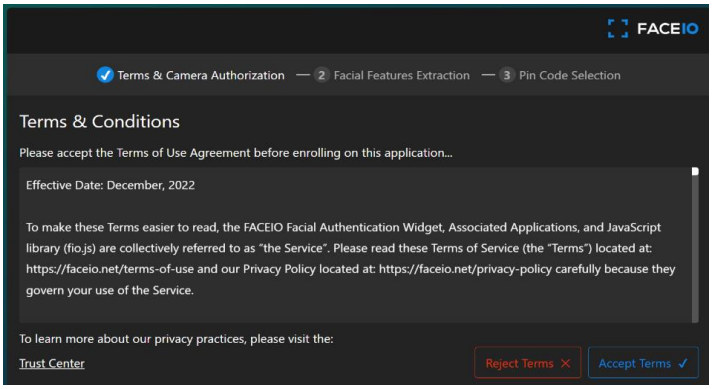


Figura 27. Ventana de términos y condiciones de API de FaceIO.

A continuación, el *API* de *FaceIO* realiza la extracción de las características faciales identificando el rostro del usuario y tomando una foto automáticamente, como en la Figura 28. Posteriormente, la aplicación pide el PIN de seguridad antes mencionado, que es de cuatro dígitos, el cual debe ingresarse dos veces, como indica la Figura 29.

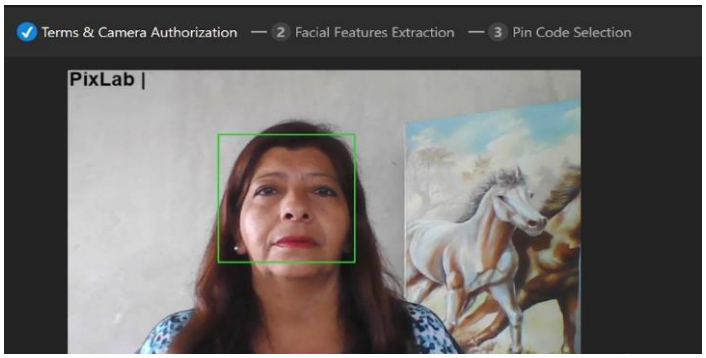


Figura 28. Toma de datos biométricos por medio de cámara web.

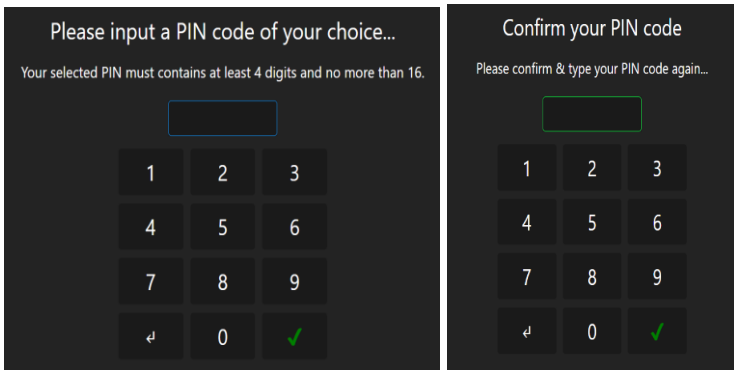


Figura 29. Ingreso de PIN de seguridad.

Una vez es ingresado el PIN en los campos solicitados, la aplicación muestra una ventana de alerta con los datos de las características biométricas detectadas que indican que el registro ha sido exitoso. De los datos presentados, solamente interesa el ID único que genera el *API* de *FaceIO*, que es el dato a ser almacenado en la Blockchain, como muestra la Figura 30.

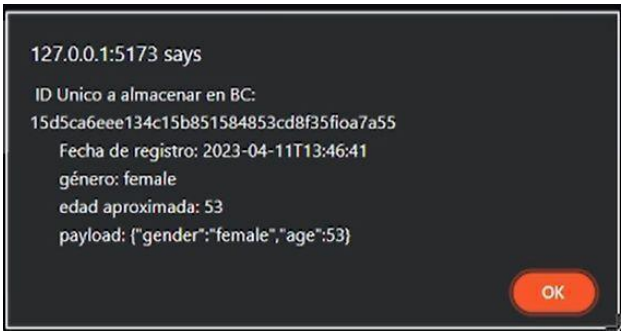


Figura 30. Ventana emergente de registro exitoso.

5.4.2 Registro no exitoso

Un registro no exitoso puede ocurrir en dos casos. El primer caso sucede cuando hay algún problema con alguno de los campos del formulario de registro, tales como dejar un campo vacío o ingresar un correo electrónico no válido, y presionar el botón "Registrarse". En estos casos la aplicación muestra mensajes en rojo indicando que los campos deben diligenciarse correctamente, como indica la Figura 31.

Figura 31. Vista de Registro, campos requeridos.

Cuando un correo electrónico que ya ha sido registrado es ingresado o cuando el *API* de *FaceIO* detecta un rostro ya registrado, es el segundo caso no exitoso. Cuando el email ya ha sido registrado, aparece una alerta como la presentada en la Figura 32 indicando que el correo electrónico ha sido registrado anteriormente y retorna a la vista de inicio.

Cabe mencionar que, hasta esta etapa de desarrollo, la alerta emergente para correo electrónico ya registrado y para rostro ya registrado, es la misma.

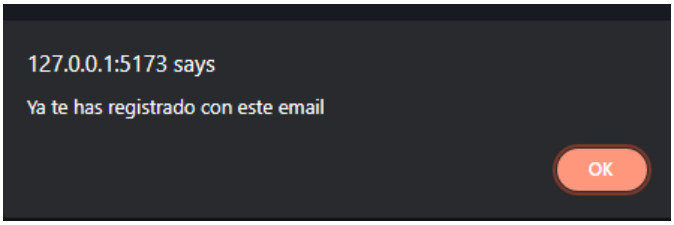


Figura 32. Vista de Registro, usuario ya existente.

5.4.3. Inicio de sesión exitoso

El flujo de un proceso de inicio de sesión exitoso, empieza por la vista presentada en la Figura 22, en la cual, al seleccionar el botón “Iniciar sesión”, hace el llamado al *API* de *FaceIO*, la cual toma una fotografía por medio de la cámara web (pidiendo permisos si no han dado anteriormente). Posteriormente es realizada la verificación de la información biométrica almacenada y el inicio de sesión mostrando un mensaje en modo ventana de alerta que indica que la identificación ha sido correcta, tal como muestra la Figura 33. Luego, la aplicación redirige al usuario a una vista de inicio la cual está implementando el módulo de autenticación, como presenta la Figura 34.

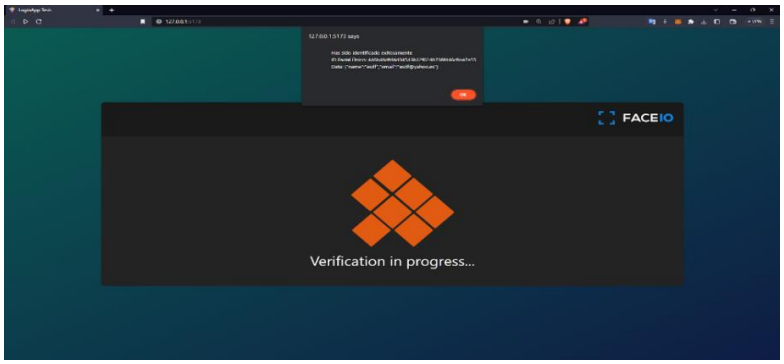


Figura 33. Vista de Inicio de sesión, inicio de sesión exitoso

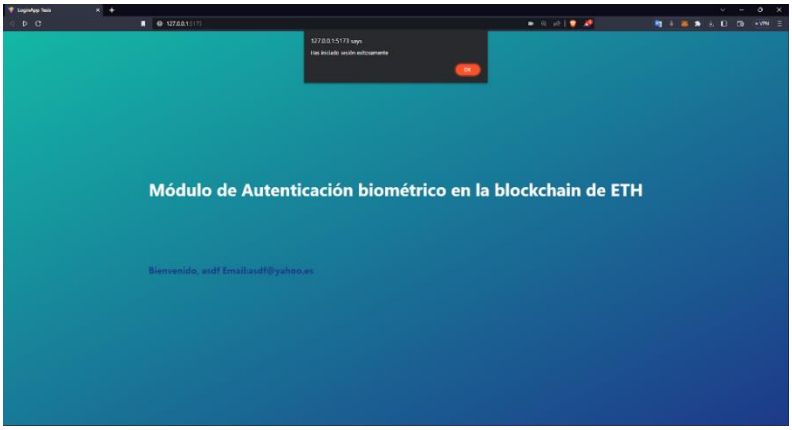


Figura 34. Vista de Inicio luego de inicio de sesión exitoso.

5.4.4. Inicio de sesión no exitoso

En caso de el inicio de sesión no sea exitoso, por diferentes motivos como falta de iluminación, posición incorrecta al momento de tomar la fotografía, usuario no existente, entre otros, aparece un mensaje de alerta como el que muestra en la Figura 35 y la aplicación redirige al usuario a la vista de inicio de sesión.

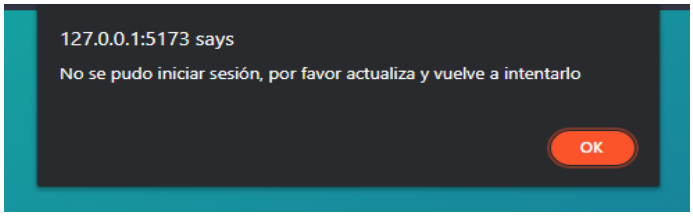


Figura 35. Vista de Inicio de sesión, inicio de sesión no exitoso.

Las pruebas de funcionamiento son finalizadas con resultados satisfactorios para el módulo de autenticación propuesto e implementado.

5.5. Costos de despliegue y consumo del contrato inteligente

Teniendo en cuenta que *Hardhat* fue utilizado de manera local para simular un nodo de la *Blockchain* de *Ethereum*, fue necesario encontrar una herramienta que nos permitiera simular el despliegue del contrato inteligente en un entorno más cercano al real, *Remix*, es una herramienta integral de desarrollo de contratos inteligentes en *Ethereum*, que ha ganado prominencia en la comunidad de *Blockchain* debido a su interfaz amigable y sus capacidades de simulación de despliegue de contratos inteligentes. La utilidad principal de Remix radica en su capacidad para proporcionar un entorno de desarrollo en línea, eliminando la necesidad de configuraciones locales complejas y permitiendo a los desarrolladores escribir, probar y desplegar contratos inteligentes de manera eficiente [75].

La simulación de despliegue de contratos inteligentes en *Remix* es una práctica común para comprender y evaluar el rendimiento de los contratos antes de su implementación en la cadena de bloques. La ventaja crítica de esta simulación radica en la capacidad de estimar con precisión los costos asociados con la ejecución y la transacción del contrato inteligente.

Para realizar el análisis, fue estimada una cantidad de 500 usuarios, considerando que es un escenario representativo que refleja una carga de trabajo significativa.

Para determinar el costo aproximado de ejecución y transacción para 500 usuarios que intenten registrarse, dados los valores de gas de 6589 y 28241 respectivamente, como presenta la Figura 36, (la cual es generada a partir de Remix, para simular y analizar el rendimiento del sistema), un valor de gas de US $0.00025 por unidad, podemos calcular el costo total en dólares.

Para la ejecución por usuario, el costo sería de 6589 * US $0.00025, y para la transacción, el costo sería de 28241 * US $0.00025. Sumando ambos valores, obtenemos el costo total de operación por usuario, que asciende a US $0.01644 aproximadamente. Para el caso de los 500 usuarios considerados, el costo total de operación aproximado es US $ 8,22.

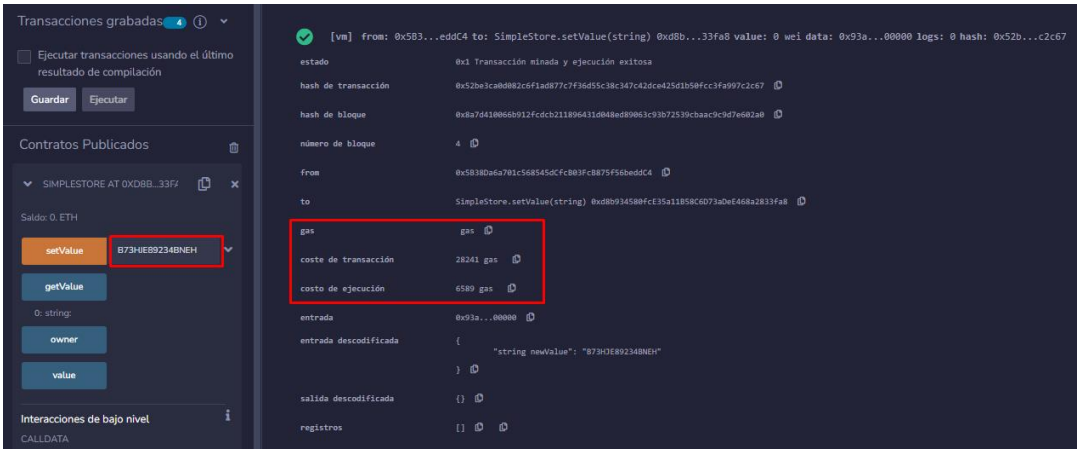


Figura 36. Costos de transacción y ejecución para el registro en la Blockchain de Ethereum.

Capítulo 6.

6. Prototipo de un sistema de vehículo compartido, que utiliza la aplicación de autenticación con *Blockchain*

Este capítulo presenta el proceso de implementación de una aplicación web de carpooling y su integración con el módulo de autenticación *Blockchain*.
Este capítulo está relacionado con el logro del objetivo específico uno, enfocado a la implementación de un prototipo de un sistema de vehículo compartido, con un mejoramiento en la seguridad en el proceso de autenticación de los usuarios, utilizando *Blockchain*. De esta forma el prototipo contiene características inherentes de la *Blockchain* como descentralización, privacidad y seguridad.

6.1. Estudio de requisitos

El estudio de requisitos es una etapa fundamental para desarrollar una aplicación web, ya que permite identificar y documentar de manera precisa las necesidades y las expectativas de los usuarios al usar la aplicación. Esto es fundamental para que el producto final de la aplicación cumpla con las expectativas y necesidades de los usuarios finales [76].

La ingeniería de requisitos utiliza dos medios de comunicación para recoger los requisitos, los cuales son la comunicación verbal y la comunicación escrita, cada uno de ellos tiene diferentes ventajas. La comunicación verbal es fácilmente adaptable a nuevos desarrollos y permite recibir un *feedback* de forma inmediata, mientras que, la comunicación escrita registra la información de forma permanente y puede compartirse de forma más sencilla con grupos, lo que permite su completa revisión [77].

Existen diferentes metodologías para realizar el estudio de requisitos, las más conocidas son:

- **Entrevistas con los usuarios.** La entrevista es un proceso que consiste en un intento sistemático de recolectar información de otra persona a través de una comunicación interpersonal estructurada en forma de conversación.
  En una entrevista, el entrevistador debe seguir una estrategia para obtener información relevante y útil. Para ello, puede hacer uso de preguntas abiertas y cerradas, así como de técnicas de escucha activa [78].
- **Desarrollo conjunto de aplicaciones.** Más conocido como JAD por sus siglas en inglés, *Joint Application Design.* Es una metodología enfocada en la colaboración entre el equipo de desarrollo y los usuarios finales. En un proceso típico de *JAD* los desarrolladores y los usuarios realizan reuniones en una serie de sesiones de trabajo estructuradas [78].
- **Desarrollo de prototipos.** Consiste en versiones reducidas de la aplicación que no son necesariamente funcionales en su totalidad. Permiten al usuario experimentar con dicho prototipo para ver sus posibles funcionalidades [76].
- **Lluvia de ideas.** Basado en reuniones con un grupo de personas, en donde proponen ideas sin juzgar su validez, y luego de compilar dichas ideas, para continuar con un análisis detallado de cada una de ellas [78].
- **HU.** Son utilizadas en el desarrollo ágil de software para identificar y documentar los requisitos de una aplicación. Está enfocado en describir las necesidades y objetivos de los usuarios de una forma clara y fácil de entender, además, combina las ventajas de los dos tipos de comunicaciones (verbal y escrita) mencionadas anteriormente [77].

La metodología en cascada fue la seleccionada para desarrollar el proyecto, desarrollando los paquetes de trabajo definidos, pero algunos paquetes de trabajo involucraron el desarrollo de prototipos o aplicaciones en particular, en los cuales fue utilizado el marco de trabajo (o *framework)* de *Scrum.*

En cuanto a los procesos de la gestión de un proyecto, Scrum propone cinco fases: inicio, planeación y estimación, implementación, revisión y retrospectiva, y lanzamiento [79]. En la fase de planeación y estimación Scrum propone el uso de HU.

Fue conveniente seleccionar las HU para el estudio de requisitos de la aplicación web del proyecto, considerando que el *framework* Scrum las utiliza y adicionalmente, debido a las siguientes razones:

- Agiliza la administración de requisitos, reduciendo la cantidad de documentos formales necesarios.
- Están centradas en el usuario, existe mayor comprensión de lo que el usuario realmente necesita.
- Son fáciles de comprender, ya que son presentadas en un lenguaje natural.
- Son iterativas y evolutivas, ya que son ajustables a medida que avanza el desarrollo del proyecto, lo que permite una mayor flexibilidad y adaptación a los cambios.
- Pueden priorizarse en función de su importancia para el usuario y su valor para el negocio.

La aplicación a desarrollar fue denominada “GoTogether”. Es una aplicación web que presta un servicio de carpooling implementando un registro y una autenticación de usuario segura. Los usuarios conductores pueden compartir su ruta y los usuarios pasajeros pueden seleccionar la ruta que más les convenga.

6.2. Descripción de la funcionalidad seleccionada

La aplicación implementada en el presente documento es una aplicación de transporte compartido tipo *carpooling* que permite a las personas utilizar un medio de transporte compartido. Este medio de transporte busca minimizar problemas como congestión y contaminación ambiental, intentando reducir el número de vehículos que transitan las vías de una ciudad. En primer lugar, la aplicación proporciona un sistema de registro y autenticación facial para los conductores y pasajeros.

Para un usuario pasajero, su interacción con la aplicación de *carpooling* inicia con el proceso de registro, para esto, el usuario pasajero debe completar un formulario con su información, además de proporcionar una fotografía frontal y un pin de seguridad. Si el sistema acepta la información brindada por el usuario, éste podrá iniciar sesión y hacer uso de la aplicación.

Para el inicio de sesión de un usuario pasajero, la aplicación pide al usuario la toma de una fotografía, con la cual el sistema verifica la identidad de la persona que intenta ingresar a la aplicación. En caso de que la información biométrica suministrada

coincida con la almacenada, el sistema le pide al usuario el pin de seguridad y si este último es correcto, el usuario finaliza correctamente el proceso de inicio de sesión.

Una vez el pasajero ingresa al sistema, mediante la autenticación descrita, procede a agregar su requerimiento de viaje, proporcionando la ubicación del lugar inicial y final del viaje, así como la hora aproximada en la que desea realizarlo. Al ingresar dicha información, la aplicación realiza las consultas respectivas y luego procede a informar de los posibles conductores que pueden atender el servicio del pasajero, dependiendo tanto de la ubicación inicial como del destino.

Posteriormente, el pasajero selecciona el servicio del conductor que mejor considere y que tenga compatibilidad con su viaje (si existe alguno), con lo cual envía una solicitud al conductor elegido para realizar el viaje. Si la solicitud es aceptada por el conductor, el pasajero recibe a través de la aplicación la aceptación de su solicitud (indicando lugar de recogida, y lugar de destino), con lo cual quedaría programado el servicio. Está disponible para el usuario la opción de cancelar viaje, en caso de que ya no desee tomar el servicio.

Al terminar el viaje, el pasajero puede calificar al conductor según su experiencia en el trayecto. Adicionalmente, el usuario puede observar la información de su perfil y modificar algunos datos que no sean relevantes a la seguridad de la aplicación. Por último, el pasajero puede cerrar la cesión cuando termine de usar la aplicación.

Para un usuario conductor, luego de realizar el registro, proporcionando la fotografía solicitada, su nombre, la placa y modelo de su vehículo, puede iniciar sesión cuando quiera interactuar en la aplicación, realizando la autenticación facial respectiva. El conductor puede postular una ruta, ingresando un punto inicial, un punto final de su recorrido y puntos de referencia dentro de este. Dicha ruta es mostrada a los pasajeros para seleccionar un viaje.

El conductor es quien toma la decisión final de las personas a las que desea transportar, aceptando o denegando las solicitudes según sea el caso, en el momento que le aparezca cada una de las solicitudes provenientes de los usuarios. En cualquier momento, el conductor puede cancelar la ruta, es decir, su ruta

anteriormente postulada es eliminada. Si el conductor cancela una ruta, y ya hay pasajeros en dicha ruta, estos son informados de dicho evento. En caso de que la ruta no sea cancelada y él viaje sea realizado, el conductor puede brindar una calificación a los pasajeros de acuerdo con su experiencia en el trayecto. Además, la aplicación brinda la opción de gestionar perfil, en donde el conductor puede editar información que no sea relevante para la seguridad de la aplicación. Por último, el conductor puede cerrar sesión cuando termine la interacción en la aplicación.

También, la aplicación cuenta con un usuario administrador, el cual no va a poder registrarse ya que la información de ingreso es única. Para iniciar sesión como administrador, el usuario debe ingresar el correo y la contraseña. Al ingresar, este tiene dos opciones, visualización de los perfiles de los usuarios y cerrar sesión.

6.3. HU de la aplicación

De la funcionalidad descrita en el punto 6.2, fueron identificados tres tipos de usuario que son: pasajero, conductor y administrador, cada uno de ellos cuenta con las siguientes HU:

HU del pasajero

- Registro del pasajero
- Inicio de sesión del pasajero
- Visualización de perfil del pasajero
- Calificación al conductor.
- Cerrar sesión del pasajero.
- Agregar viaje como pasajero.
- Visualización de rutas de interés por parte del pasajero.
- Selección de ruta del pasajero.
- Cancelar viaje como pasajero.

HU del conductor

- Registro del conductor
- Inicio de sesión del conductor.
- Visualización de perfil del conductor.
- Calificación a pasajeros.

- Cerrar sesión del conductor.
- Postular ruta por parte del conductor.
- Selección de pasajeros.
- Cancelar ruta de usuario conductor.

HU del administrador

- Inicio de sesión del administrador.
- Gestión de usuarios
- Cerrar sesión del administrador

Algunas de las HU más relevantes son detalladas a continuación en las Tablas 5, 6, 7, 8, 9, 10, y 11. Las HU completas están disponibles en el **Anexo B. Desarrollo de un prototipo de una aplicación web de *carpooling.***

6.3.1. HU del pasajero

6.3.1.1. Agregar viaje como pasajero

ID HISTORIA	HU5
NOMBRE	Agregar viaje como pasajero
DESCRIPCIÓN	**Como** usuario pasajero **Quiero** poder agregar mi lugar de inicio y destino **Para** poder solicitar el servicio de transporte según mis necesidades.
CRITERIOS DE ACEPTACIÓN	**Escenario 1:** Ingresar punto inicial y final del recorrido **Dado** que un usuario agrega ubicación inicial y final **Cuando** desea realizar un viaje **Entonces** el pasajero postula el viaje para observar las opciones brindadas en la aplicación.
PRIORIDAD	2
ESTIMACIÓN	5 días

Tabla 5. Historia de usuario – agregar viaje como pasajero.

6.3.1.2. Visualización de rutas de interés por parte del pasajero

ID HISTORIA	HU6
NOMBRE	Visualización de rutas de interés por parte del pasajero

DESCRIPCIÓN	**Como** usuario pasajero **Quiero** poder observar todas las rutas disponibles dependiendo de mi ubicación actual y mi destino **Para** poder escoger una ruta y el conductor que realice dicho viaje.
CRITERIOS DE ACEPTACIÓN	**Escenario 1:** Existen rutas para las ubicaciones ingresadas. **Dado** que un usuario ingresa la información de ubicación inicial y de destino y existen rutas que coinciden con dicha información **Cuando** desea realizar un viaje **Entonces** el pasajero observa todas las rutas de interés que están disponibles. **Escenario 2:** No existen rutas disponibles para las ubicaciones ingresadas. **Dado** que un usuario ingresa la información de ubicación inicial y de destino y la aplicación no encuentra rutas que coinciden con dicha información **Cuando** desea realizar un viaje **Entonces** la aplicación muestra una notificación al usuario manifestando que no existen rutas disponibles para la ubicación ingresada.
PRIORIDAD	3
ESTIMACIÓN	5 días

Tabla 6. Historia de usuario - visualización de rutas de interés por parte del pasajero.

6.3.1.3. Selección de ruta del pasajero

ID HISTORIA	HU7
NOMBRE	Selección de ruta del pasajero
DESCRIPCIÓN	**Como** usuario pasajero **Quiero** poder seleccionar una ruta que cubra algún conductor para solicitar cupo en el automóvil. **Para** poder enviar la solicitud al conductor relacionado.
CRITERIOS DE ACEPTACIÓN	**Escenario 1:** Mensaje de confirmación de ruta. **Dado** que el usuario acepta una ruta seleccionada. **Cuando** el conductor registra un punto de partida y un punto final

	Entonces aparece el recorrido del conductor con la información correspondiente al viaje e información del conductor que sea relevante. **Escenario 2:** No hay confirmación de ruta **Dado que** el pasajero no es aceptado **Cuando** la solicitud es enviada al chofer **Entonces** aparece una notificación indicando que el conductor rechazó el viaje.
PRIORIDAD	4
ESTIMACIÓN	5 días

Tabla 7. Historia de usuario - selección de ruta del pasajero.

6.3.2. HU del conductor

6.3.2.1. Postular ruta por parte del conductor

ID HISTORIA	HU14
NOMBRE	Postular ruta por parte del conductor
DESCRIPCIÓN	**Como** usuario conductor **Quiero** poder postular una ruta de mi interés. **Para** ofrecer cupos en mi automóvil a otras personas que compartan la misma ruta.
CRITERIOS DE ACEPTACIÓN	**Escenario 1:** Ruta correctamente postulada **Dado** que el conductor ingresa información referente a su ubicación inicial, final y la hora en que realizará el viaje **Cuando** desea compartir un viaje **Entonces** la información es agregada a una nueva ruta.
PRIORIDAD	1
ESTIMACIÓN	6 días

Tabla 8. Historia de usuario - postular ruta por parte del conductor.

6.3.2.2. Selección de pasajeros

ID HISTORIA	HU15

NOMBRE	Selección de pasajeros
DESCRIPCIÓN	**Como** usuario conductor **Quiero** poder aceptar o rechazar solicitudes de pasajeros **Para** tener control de las personas con las que compartiré mi ruta.
CRITERIOS DE ACEPTACIÓN	**Escenario 1:** solicitud de pasajero aceptada. **Dado** que el conductor acepta una solicitud de un pasajero **Cuando** postula un viaje **Entonces** aparece información relevante del pasajero junto con su lugar de recogida y llegada. **Escenario 2:** solicitud de pasajero rechazada. **Dado** que el conductor rechaza una solicitud de un pasajero **Cuando** postula un viaje **Entonces** la información de la solicitud del pasajero desaparece en la pantalla del dispositivo del conductor.
PRIORIDAD	5
ESTIMACIÓN	8 días

Tabla 9. Historia de usuario - selección de pasajeros.

6.3.3. HU del administrador

6.3.3.1. Inicio de sesión del administrador

ID HISTORIA	HU18
NOMBRE	Inicio de sesión del administrador
DESCRIPCIÓN	**Como** usuario administrador **Quiero** iniciar sesión **Para** acceder a la aplicación web
CRITERIOS DE ACEPTACIÓN	**Escenario 1:** Inicio de sesión exitoso **Dado** que la información coincide con la almacenada **Cuando** el administrados ingresa la contraseña **Entonces** aparece un mensaje inicio de sesión exitoso **Escenario 2:** Inicio de sesión fallido **Dado** que la información no coincide con la almacenada

	Cuando el administrador ingresa la contraseña **Entonces** la aplicación muestra un mensaje error al iniciar sesión, vuelva a intentarlo
PRIORIDAD	1
ESTIMACIÓN	3 días

Tabla 10. Historia de usuario - inicio de sesión del administrador.

6.3.3.2. Gestión de usuarios

ID HISTORIA	HU19
NOMBRE	Gestión de usuarios
DESCRIPCIÓN	**Como** usuario administrador **Quiero** poder administrar información de usuarios en caso de alguna eventualidad. **Para** garantizar la seguridad y la eficacia de la aplicación.
CRITERIOS DE ACEPTACIÓN	**Escenario 1:** el usuario observa la información básica de los usuarios en caso de alguna eventualidad que así lo requiera. Como administrador, puede habilitar o deshabilitar un usuario si la situación lo requiere.
PRIORIDAD	2
ESTIMACIÓN	4 día

Tabla 11. Historia de usuario - gestión de usuarios.

6.4. Vistas de las HU

En el **Anexo B. Desarrollo de un prototipo de una aplicación web de *carpooling*,** pueden visualizarse la totalidad de vistas propuestas como interfaz de usuario tanto para usuario pasajero como usuario conductor. Las Figuras 37, 38, 39 y 40 son las vistas de usuario más relevantes, para presentar en este documento.

La vista de la Figura 37 es el inicio de la aplicación. Los usuarios pasajero y conductor, que requieren autenticación biométrica, inician sesión ingresando a la opción respectiva presentada en la Figura 37. Mientras que el usuario administrador, que no requiere autenticación biométrica, tiene una opción para ingresar con el botón

denominado Administrador. No hay posibilidad de registrarse como usuario de tipo administrador, por restricciones del alcance de la aplicación.

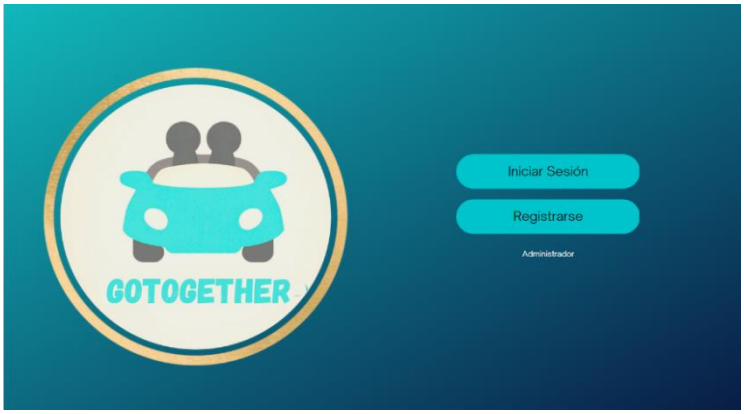


Figura 37. Vista de inicio.

La Figura 38 muestra la vista de registro, en la cual los usuarios pueden seleccionar entre pasajero y conductor, luego completan el formulario con datos básicos y finalmente, la cámara web es habilitada para realizar una fotografía del usuario.

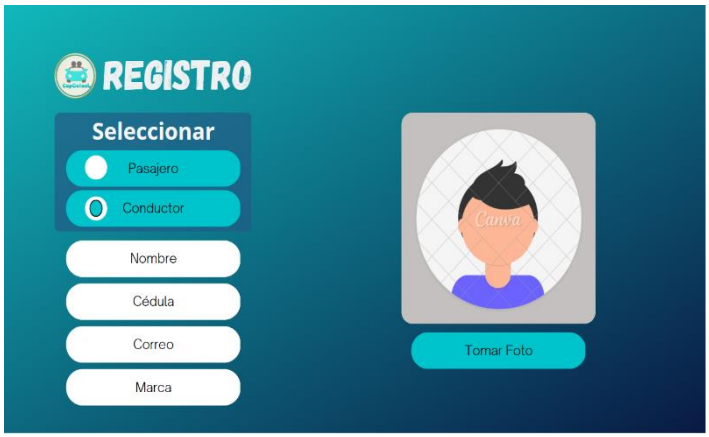


Figura 38. Vista de registro de usuarios.

Las Figuras 39 y 40, presentan las vistas iniciales propuestas para pasajero y conductor respectivamente cuando han ingresado. En estas vistas pueden observarse opciones como perfil, cerrar sesión, agregar viaje o postular ruta dependiendo del tipo de usuario y la posibilidad de visualizar un mapa que muestra las posibles rutas.

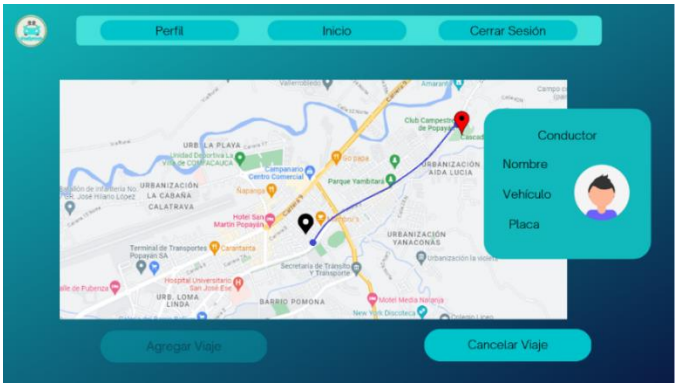


Figura 39. Vista de usuario pasajero.

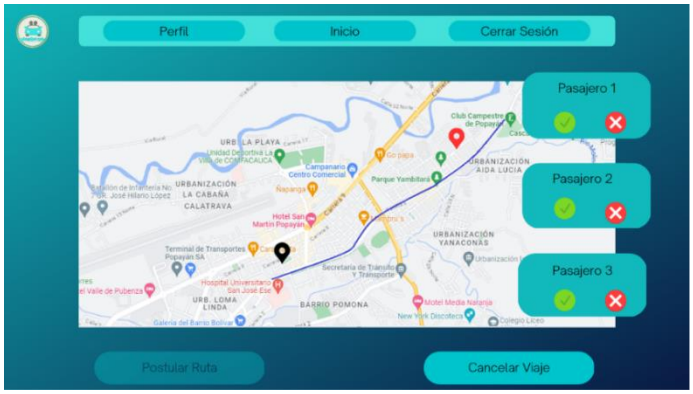


Figura 40. Vista de usuario conductor.

6.5. Diseño de la experiencia de usuario

El diseño de la experiencia de usuario (UX *design* por sus siglas en inglés), es el proceso de diseñar la interacción que los usuarios tienen con una aplicación, sitio web, producto o servicio, con el objetivo de crear una experiencia satisfactoria y efectiva para el usuario. El UX *design* está enfocado en la experiencia del usuario en todas las etapas del proceso de interacción, desde la primera impresión al descubrir la aplicación hasta la realización de las tareas que desea [80].

El UX *design* también está enfocado en factores como la arquitectura de la información, el diseño visual, la interacción y la usabilidad, todo ello con el objetivo de crear una experiencia coherente y atractiva para el usuario [80].

Para este proyecto, los siguientes criterios de UX fueron tenidos en cuenta:

- Usabilidad. Es la facilidad con la que los usuarios pueden utilizar una aplicación para lograr sus objetivos. Los criterios de usabilidad incluyen la claridad del diseño, la facilidad de navegación y la eficiencia en la realización de tareas.

- Atractivo visual. Es la apariencia estética y la calidad del diseño de la aplicación. Los criterios de atractivo visual incluyen el diseño coherente, la consistencia en la marca, la calidad de las imágenes y la simplicidad visual.
- Funcionalidad. Es la capacidad de la aplicación para cumplir con las necesidades y objetivos de los usuarios. Los criterios de funcionalidad incluyen la capacidad para realizar tareas específicas, la rapidez y la capacidad de respuesta de la aplicación.

En el **Anexo B. Desarrollo de un prototipo de una aplicación web de carpooling** están de forma detallada las vistas modificadas teniendo en cuenta los criterios de experiencia de usuario.

6.6. Desarrollo de la aplicación web de *Carpooling*

La aplicación web de *carpooling* fue desarrollada usando *React.js y Spring Boot,* para las aplicaciones *frontend* y *backend* respectivamente, también el *API de FaceIO* tuvo un papel importante para obtener la información biométrica del usuario.

La Figura 40 muestra el diagrama de arquitectura funcional del prototipo realizado, basado en la arquitectura ITS tal como la Figura 4, ya que proponía el tipo de servicio más relacionado con el prototipo desarrollado, respecto a las otras arquitecturas ITS revisadas.

La parte central de la Figura 41 exhibe la aplicación implementada, detallando el frontend construido con React que aborda el componente de información personal. Conforme fue expuesto en el capítulo anterior, este frontend ha sido desarrollado mediante Typescript, HTML y CSS, constituyendo el espacio donde las interfaces visuales de la aplicación web son configuradas. Este componente esta enlazado al módulo de autenticación descrito en el Capítulo 5 a través de la librería ether.js (Servicio Blockchain) y al backend elaborado con Spring Boot, específicamente en Java, que forma parte del Centro de Transporte de Uso Compartido, como indica el capítulo 4.

El *Backend* establece conexión con la base de datos relacional utilizada, que en este caso es *PostgreSQL*, mediante *Java Persistence API* (JPA). En dicha base de datos,

son almacenados datos cruciales como la información de viajes, rutas, calificaciones de conductores y pasajeros, entre otros. Además, como indica la Figura 41, son descritos los distintos tipos de usuarios presentes en la aplicación, como administrador, conductor y pasajero. Cada uno de estos perfiles es definido minuciosamente en función de las Historias de Usuario (HU) desarrolladas.

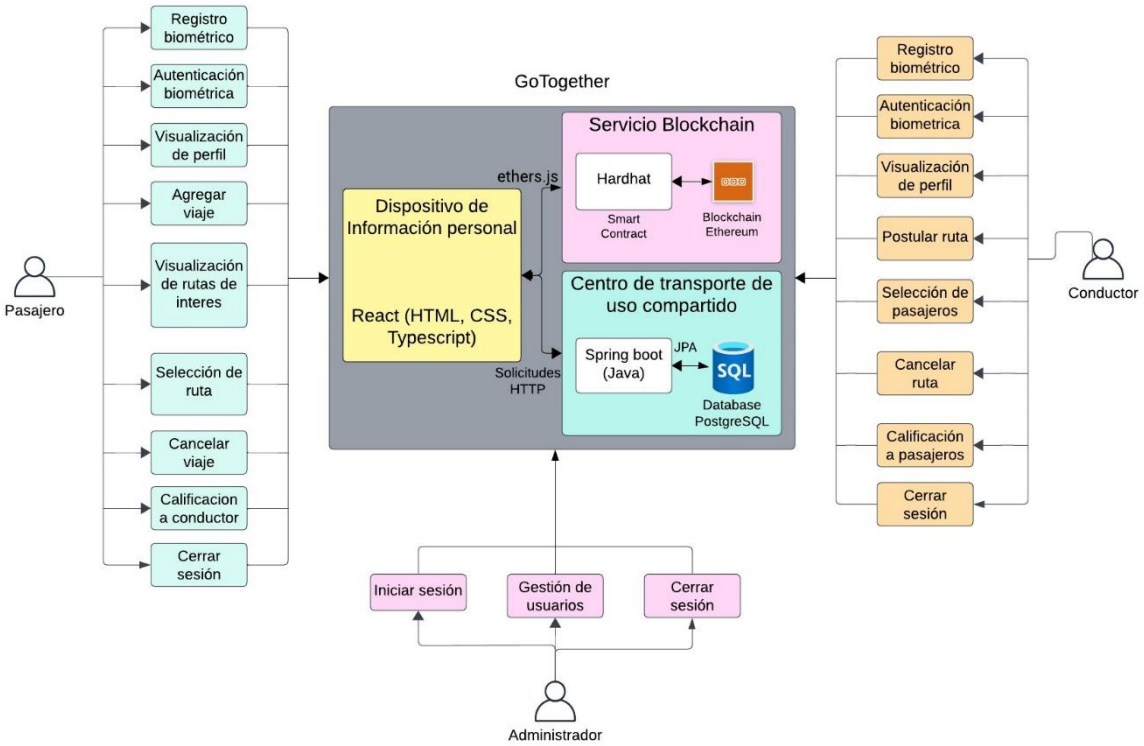


Figura 41. Arquitectura de prototipo implementada.

Este proceso de desarrollo, así como el resultado final de las vistas de usuario son detallados en el **Anexo B. Desarrollo de un prototipo de una aplicación web de *carpooling*,** sin embargo, el resultado de las interfaces más importantes es mostrado a continuación.

La Figura 42, muestra la pantalla de inicio de la aplicación, que contiene diferentes opciones. El botón principal (“Iniciar sesión”) permite a los pasajeros y conductores iniciar sesión en la aplicación por medio de datos biométricos. Si aún los usuarios no tienen cuenta, aparecen dos opciones (“Registrarse como conductor” o “Registrarse como pasajero”) y, por último, para el administrador del sistema, aparece la opción de iniciar sesión como administrador.

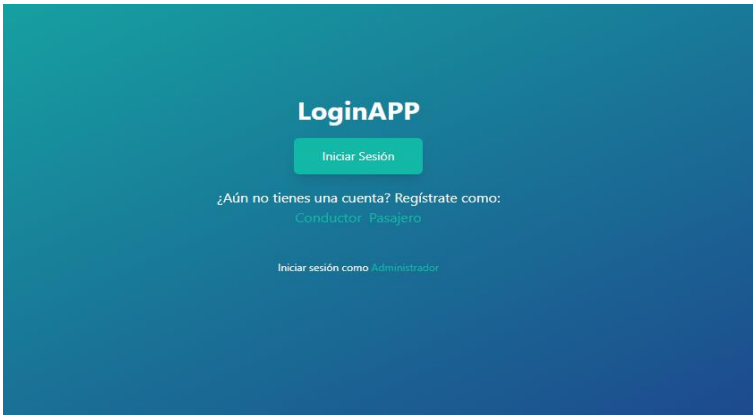


Figura 42. Inicio de la aplicación.

Tomando como ejemplo un usuario pasajero, presionando en el botón de “pasajero” de la Figura 42. Posteriormente, la aplicación redirige al usuario a un formulario mostrado en la Figura 43, en donde el usuario pasajero debe diligenciar cada uno de los campos requeridos que para su caso son: Nombre, Número de identificación y Correo electrónico.

Figura 43. Registro de usuario pasajero.

Luego de que el usuario diligencie el formulario y selecciona la opción “Registrarse”, la aplicación realiza la petición al *API* de *FaceIO* que es la encargada del proceso de tomar los datos biométricos del usuario y la asignación de un *Personal Identification Number* (PIN, número de identificación personal), como indican las Figuras 44 y 45.

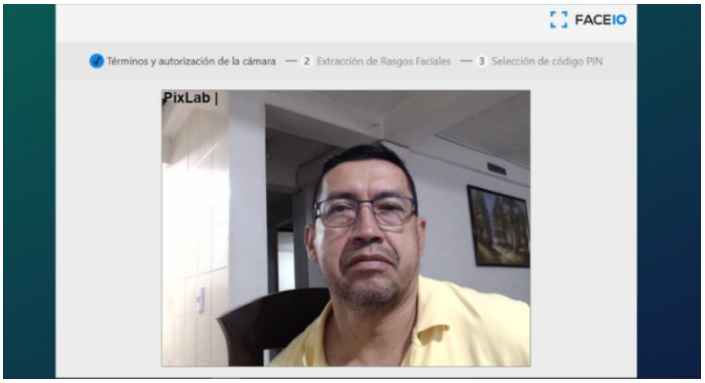


Figura 44. Toma de datos biométricos.

Figura 45. PIN solicitado.

Es importante tener en cuenta que los datos adicionales de los usuarios, que no son biométricos, fueron almacenados en una base de datos relacional *SQL* gestionada por la aplicación *backend* que fue implementada en *Spring Boot.*

El proceso de registro como usuario conductor es similar al realizado con el usuario pasajero, con la diferencia de que el conductor debe especificar algunos datos extra como placa y modelo del vehículo.

Cuando el proceso de registro es exitoso es posible seleccionar la opción Iniciar Sesión en la vista principal de la Figura 42, esto realiza nuevamente una petición al *API* de *FaceIO*, quién está encarga de autenticar el usuario que intenta ingresar. Además, la aplicación *backend* identifica si el usuario que intenta acceder es un usuario pasajero o un usuario conductor. La Figura 46 muestra la vista de un usuario conductor que ha propuesto una ruta y puede consultar qué pasajeros han solicitado unirse a su viaje, además tiene opciones como Perfil, Cerrar sesión, Generar ruta, Cancelar Ruta. Estas y las demás características desarrolladas, tanto para usuario

conductor, como para usuario pasajero, son descritas de forma detallada en el **Anexo B. Desarrollo de un prototipo de una aplicación web de *carpooling.***

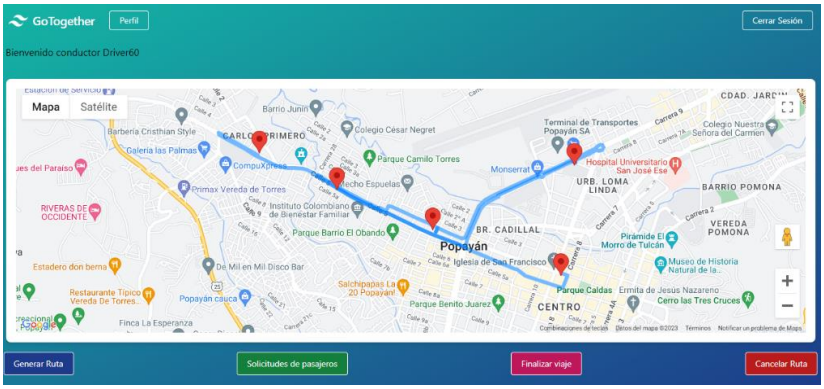


Figura 46. Vista del usuario conductor.

El usuario administrador no cuenta con autenticación y en cambio, utiliza un método más clásico de autenticación como lo es usuario y contraseña, este tipo de usuario tiene acceso a la lista de usuarios conductores y a la lista de pasajeros, además de la opción de deshabilitar o habilitar dichos usuarios. El resultado de la vista de administrador puede observarse en la Figura 47.

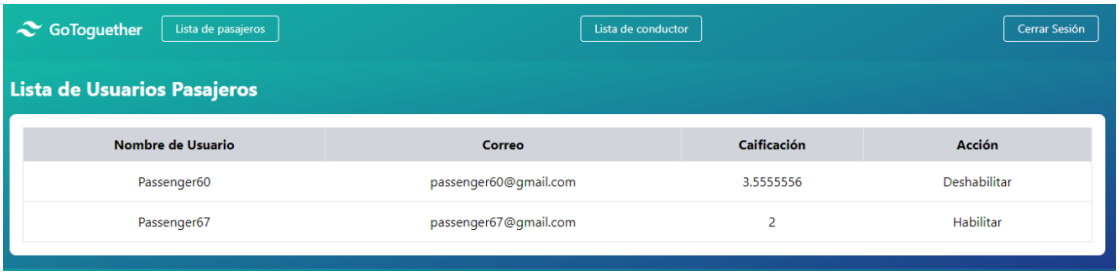


Figura 47. Vista de usuario administrador.

6.7. Integración módulo de autenticación *Blockchain* con la aplicación web de *Carpooling*

Para que la aplicación web de carpooling pudiera integrar el módulo de autenticación Blockchain fue necesario importar la librería ether.js a la aplicación *frontend* de *React.js* como indica el capítulo 5, con la diferencia que ahora el método que realiza la autenticación y el registro en el *frontend* debe pasar por tres capas: la interacción con el módulo de autenticación *Blockchain* a través de la invocación del contrato inteligente desplegado en el nodo de *Ethereum* para la gestión de datos biométricos, la verificación de los datos no biométricos de usuario en la aplicación *backend* y la verificación de identidad del usuario por parte del *API* de *FaceIO.*

A continuación, los diagramas de secuencia que ilustran el proceso de registro e inicio de sesión son mostrados (Figuras 48 y 49 respectivamente). Estas representaciones gráficas permiten una comprensión más clara de la interacción entre las aplicaciones *Frontend*, *Backend* y *Blockchain.* El código completo de la aplicación desarrollada es detallado en el Anexo **C. Repositorios de GitHub.**

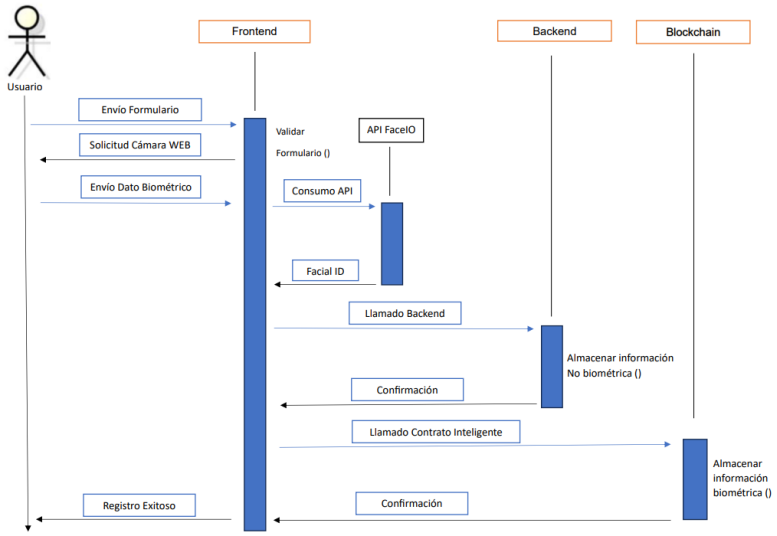


Figura 48. Diagrama de secuencia de registro de usuario.

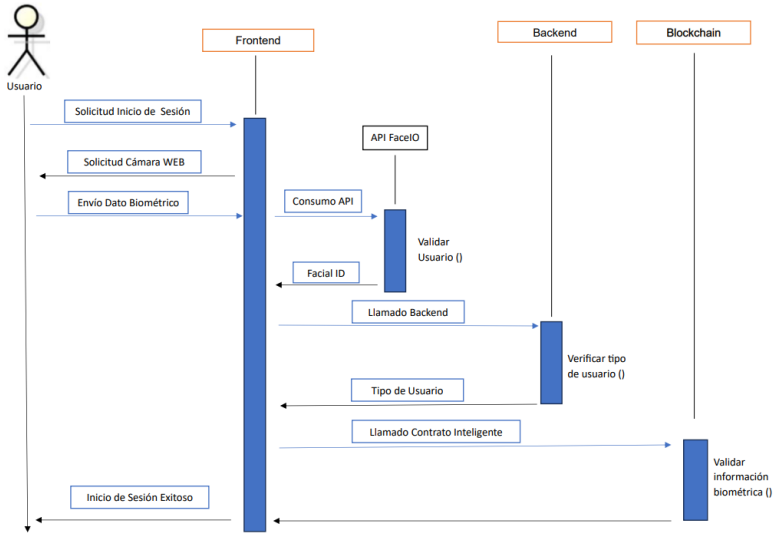


Figura 49. Diagrama de secuencia inicio de sesión de usuario.

6.8. Pruebas de funcionamiento del prototipo desarrollado

Las pruebas de funcionamiento realizadas sobre la aplicación web de *carpooling* integrando el módulo de autenticación *Blockchain* son detalladas en el **Anexo B. Desarrollo de un prototipo de una aplicación web de *carpooling.*** Es importante destacar que todos los casos de prueba propuestos fueron superados de manera

exitosa, las Tablas 12 y 13 muestran los casos que son considerados de mayor relevancia.

Nombre del tipo de prueba	**Resultado esperado**	**Resultado obtenido**
Registro exitoso	Mensaje de registro correcto. Redireccionamiento a la vista de inicio de sesión.	Mostró un mensaje de registro correcto y posteriormente redirecciona a la vista de inicio de sesión.
Registro no exitoso	Mensajes de alerta para completar campos de formulario de registro.	Mostró un mensaje de alerta bajo cada campo del formulario que era obligatorio diligenciar.
	Si un pasajero ha sido registrado anteriormente, aparece una alerta y redirecciona al formulario de registro de pasajero.	Fue obtenida una alerta que indicaba que el usuario ya existía y posteriormente redirecciona al usuario nuevamente al formulario de registro.
Inicio de sesión exitoso	Mensaje de alerta indicando la correcta identificación. Redireccionamiento a la ventana de bienvenida.	Mostró un mensaje indicando que el proceso de identificación adecuada y apareció la ventana de bienvenida.
Inicio de sesión no exitoso	Mensaje de inicio de sesión fallido y redireccionamiento al inicio de sesión de nuevo.	Mostró un mensaje de alerta indicando que no fue posible iniciar sesión y posteriormente la aplicación redireccionó al inicio de sesión, para realizar el ingreso nuevamente.
Encontrar una ruta exitosa	Al seleccionar dos puntos para inicio y final de viaje, la fecha y la hora, posteriormente aparecen las rutas que puedan cumplir con las especificaciones ingresadas por el pasajero	Fueron seleccionados dos puntos en el mapa, fecha y hora y fueron mostrados los botones que muestran las rutas disponibles
Encontrar una ruta no exitosa	Al seleccionar más de dos puntos en el mapa, no debe permitir la selección de fecha y hora, los puntos marcados en el mapa desaparecen y deben ser seleccionados de nuevo.	Fueron seleccionados tres puntos en el mapa y apareció un mensaje que indica que hay que seleccionar exactamente dos puntos en el mapa, los puntos marcados ya no son visibles y mantuvo en la vista del pasajero.
	Al seleccionar dos puntos en el mapa, sin embargo, no hay rutas que	Fueron seleccionados dos puntos en el mapa, la fecha y la hora, seguido la aplicación mostró un

	cumplan con las especificaciones dadas por el pasajero. El usuario es informado diciéndole que no hay rutas disponibles y las marcas en el mapa son borradas, manteniéndose en la vista de pasajero	mensaje que indica que no había rutas disponibles, los puntos marcados son borrados manteniéndose en la vista del pasajero.
Seleccionar una ruta exitosa	Dado que el pasajero visualiza las rutas disponibles, cuando el pasajero selecciona su ruta, puede visualizar los datos del conductor y presionar el botón aceptar, con lo cual, ya no es posible visualizar las rutas disponibles y es posible visualizar los botones Estado de ruta y Finalizar viaje.	Fueron visualizadas las rutas disponibles, al seleccionar en el ícono de la lupa fueron visualizados los datos del conductor que corresponde a la ruta seleccionada, al presionar en “aceptar”, ya no es posible visualizar las rutas encontradas y es posible visualizar los botones Estado de ruta y Finalizar viaje.
Seleccionar una ruta no exitosa	Al seleccionar una ruta y eliminar la sesión de manera manual por base de datos, es esperado que la consola muestre un mensaje indicando que no ha sido posible seleccionar una ruta. Este caso es posible únicamente en el caso de que exista algún problema de comunicación entre la aplicación *frontend* y la aplicación *backend*.	Fueron visualizadas las rutas disponibles, al presionar el ícono de la lupa y fue posible visualizar los datos del conductor que corresponde a la ruta seleccionada, al presionar en “aceptar”. Pudo visualizarse un mensaje en consola que indica que no fue posible seleccionar la ruta.
	Dado que el pasajero visualiza los datos del conductor y decide no seleccionar dicha ruta, presiona en el botón cancelar y vuelve a visualizar las rutas disponibles	Fueron visualizados los datos de un conductor, al presionar en cancelar mostró la vista de pasajero, en la cual persisten las rutas disponibles.

Tabla 12. Pruebas a HU de pasajero.

Nombre del tipo de prueba	**Resultado esperado**	**Resultado obtenido**
Registro exitoso como conductor	Mensaje de registro correcto. Redireccionamiento a la vista de inicio de sesión.	La aplicación mostró el mensaje de registro correcto y posteriormente redireccionó a la vista de inicio de sesión.

Registro no exitoso como conductor	Mensajes de alerta para completar campos de formulario de registro.	La aplicación mostró los mensajes de alerta bajo cada campo del formulario que era obligatorio diligenciar.
	Si ya ha registrado una persona con los mismos datos, aparece una alerta y redirecciona al formulario de registro.	Fue obtenida una alerta que indica que el usuario ya existe y posteriormente redireccionó nuevamente al formulario de registro.
Inicio de sesión exitoso	Mensaje de alerta indicando la correcta identificación. Redireccionamiento a la ventana de bienvenida.	Se realizó un correcto inicio de sesión, apareció un mensaje indicando una identificación adecuada y apareció la ventana de bienvenida.
Inicio de sesión no exitoso	Mensaje de inicio de sesión fallido y redireccionamiento al inicio de sesión de nuevo.	Se recibió un mensaje de alerta indicando que no ha sido posible iniciar sesión y posteriormente la aplicación redireccionó al inicio de sesión, para realizar el ingreso.
Selección de puntos de ruta exitoso	Al dibujar la ruta que pasa por los puntos del mapa seleccionados del conductor e inmediatamente despliega un formulario de fecha y hora.	El mapa la ruta del conductor fue visible en pantalla y posteriormente es mostrado un formulario para escoger la fecha y la hora de la ruta.
Selección de puntos de ruta no exitoso	Alerta indicando que los puntos proporcionados fueron insuficientes.	La aplicación mostró una alerta en la pantalla indicando que deben ser seleccionados dos o más puntos.
Selección de fecha y hora del viaje exitoso.	El formulario es cerrado y la ruta del conductor puede observarse en el mapa	El formulario desapareció y fue posible observar la ruta del conductor en el mapa.
Selección de fecha y hora del viaje no exitoso.	Aparece una alerta en la pantalla del usuario, que indica que los datos ingresados fueron inválidos	Apareció un mensaje, indicando que deben proporcionarse una fecha y hora válidas, manteniendo abierto el formulario.

Tabla 13. Pruebas HU de conductor.

Capítulo 7.

7. Evaluación de seguridad del prototipo del sistema

Este capítulo muestra las pruebas de seguridad a la aplicación realizada. Este procedimiento contribuye en el cumplimiento del objetivo específico 3, el cual corrobora los beneficios del sistema propuesto en cuanto a seguridad.

7.1. Diseño de pruebas

Para realizar las pruebas de seguridad del prototipo del sistema desarrollado, varios aspectos que garantizaran una mayor cobertura en el desarrollo de las pruebas fueron tenidos en cuenta.

Primero, fue establecida la arquitectura de las pruebas, instalando una máquina virtual de Linux en nuestros equipos con sistema operativo Windows. La elección de Linux como entorno de pruebas estuvo basada en la presencia mayoritaria de herramientas de simulación de ataques en este sistema operativo.

Aunque existen varias distribuciones de Linux, tales como *Ubuntu*, *CentOS*, *RedHat*, entre otras. La que fue seleccionada para la realización de las pruebas fue Kali Linux, la cual es una distribución de Linux basada en Debian que es utilizada principalmente para tareas de seguridad. Esta versión contiene múltiples herramientas que ayudan con tareas como pruebas de penetración, investigación de seguridad, informática forense y gestión de vulnerabilidades [81].

Dado que el ambiente de desarrollo del prototipo fue desarrollado en el sistema operativo *Windows*, fue necesario utilizar una herramienta que permita utilizar la aplicación desarrollada desde una máquina virtual e incluso también desde un dispositivo diferente. Por lo anterior, herramientas como *Forward, TunnelBear y Ngrok* fueron tenidas en cuenta. Dichas herramientas crean un túnel seguro que permite acceder a una aplicación desde otros dispositivos.

Para la selección de esta herramienta fueron considerados diversos aspectos, tales como la facilidad de uso, el soporte multiplataforma y que no genere ningún costo su instalación.

Según lo anterior, la herramienta escogida fue *Ngrok*, considerando que es fácil de usar en cualquier sistema operativo, existe una gran cantidad de documentación sobre su uso, tiene soporte en el sistema operativo *Windows* y, en su versión gratuita permite el acceso a la aplicación desarrollada.

Ngrok, es una herramienta que permite exponer servicios locales temporalmente a internet, es decir, brinda acceso a servicios o aplicaciones que están ejecutándose en un entorno local o en una red privada para poder ser accesibles desde cualquier lugar a través de internet [82].

La herramienta *Ngrok* permite eludir la necesidad de transferir el entorno de desarrollo a *Linux*. *Ngrok* expone la aplicación a dispositivos remotos, lo que proporciona flexibilidad en las pruebas de seguridad.

Luego de seleccionar las herramientas, fue realizado un análisis del tipo de aplicación que utiliza el prototipo del sistema, para escoger las pruebas de seguridad más convenientes a ejecutar.

En la revisión de la literatura realizada en el Capítulo 3, los artículos seleccionados mostrados en la Tabla 1, realizan diferentes pruebas de seguridad en las aplicaciones propuestas, tales como ataques *man in the middle,* suplantación de identidad, ataques de reproducción, *sybil,* entre otros. En estos artículos son encontrados algunos prototipos que utilizan la tecnología *Blockchain,* aunque no centran las pruebas de seguridad específicamente en dicha tecnología, sino en la seguridad de la aplicación completa, los artículos citados concluyen, con base de las pruebas realizadas, que *Blockchain* es lo suficientemente segura y esto sucede debido a características como la descentralización, inmutabilidad de datos y algoritmos de consenso distribuidos. En consecuencia, las pruebas realizadas en las investigaciones respaldan la definición de *Blockchain*.

El prototipo presentado en este documento enfrentó desafíos considerables al intentar simular ataques, especialmente los mencionados en el contexto de *Blockchain*. Estos desafíos son derivan de la arquitectura y el enfoque del prototipo.

En primer lugar, es fundamental destacar que nuestra aplicación no está basa exclusivamente en la tecnología *Blockchain*, sino que integra diversas herramientas y componentes para conformar una solución integral. Esto agrega una capa adicional de complejidad al realizar simulaciones de ataques, ya que no estamos tratando únicamente con la seguridad de la cadena de bloques, sino con un ecosistema más amplio que incluye otros componentes.

En segundo lugar, es importante subrayar que la tecnología *Blockchain* es empleada en el sistema para la gestión de datos de usuarios, específicamente en relación con el proceso de autenticación. En otras palabras, la *Blockchain* puede ser vista como un repositorio seguro de los datos biométricos proporcionados por *FaceIO*. Esto introduce un nivel significativo de seguridad y filtros para detectar posibles intentos de fraude en la aplicación, particularmente en lo que respecta a la autenticación de usuarios y pasajeros.

La Figura 50 presenta un diagrama general de las pruebas realizadas.

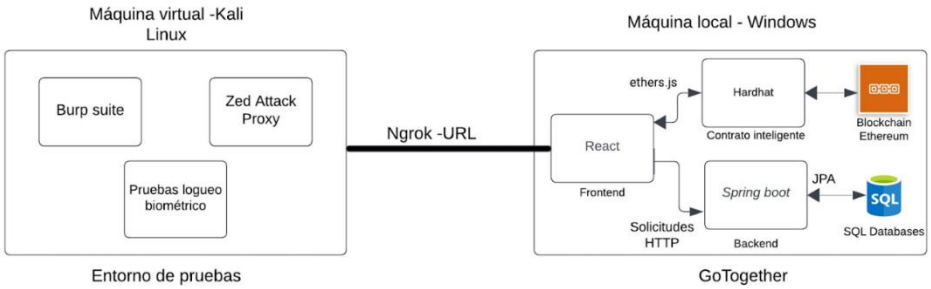


Figura 50. Diagrama de pruebas.

La Figura 50 muestra la arquitectura de pruebas. En primer lugar, es importante considerar que la aplicación de *carpooling* está corriendo en un entorno local en *Windows* (parte derecha de la Figura 50), mientras que las pruebas fueron desarrolladas en un entorno *Linux.* En la parte izquierda de la Figura 50 está alojada

la máquina virtual. La comunicación entre la aplicación en el entorno local y la máquina virtual para la realización de las pruebas es mediante la herramienta *Ngrok*, facilitando así una conexión fluida y segura. En la mencionada máquina virtual, están alojadas las diversas herramientas empleadas para llevar a cabo las pruebas, las cuales posteriormente serán explorarán detenidamente en las secciones subsiguientes.

La ejecución de la totalidad de los posibles ataques disponibles para la aplicación resulta una tarea inviable debido a su amplitud y diversidad. Por consiguiente, en aras de abordar de manera eficiente la evaluación de la seguridad de la aplicación, fueron seleccionadas cuatro tipos de pruebas centradas en puntos específicos y críticos. Esta estrategia está basada en la necesidad de enfocar los recursos disponibles de manera precisa en áreas de vital importancia, permitiendo así una evaluación exhaustiva y efectiva de la seguridad de la aplicación.

Teniendo en cuenta estas consideraciones, las pruebas de seguridad más convenientes utilizar en este caso, son: análisis de vulnerabilidades, inyección SQL, fuerza bruta, pruebas de inicio de sesión biométrico. A continuación, serán detalladas cada una de estas.

7.1.1. Análisis de vulnerabilidades

El análisis de vulnerabilidades es un proceso que ayuda a identificar y priorizar los problemas o debilidades de una aplicación, evaluando las posibles amenazas para poder reaccionar a ellas de manera oportuna y adecuada [83].

Existe una variedad de herramientas que realizan escaneo de vulnerabilidades, lo cual facilita el proceso de análisis, entre ellas estan *Nmap, Nikto y Zed attack proxy* [84]. La herramienta seleccionada para realizar las pruebas realizadas fue "*Zed Attack Proxy*" debido a que ofrece una amplia gama de funcionalidades específicas para el análisis de aplicaciones web y la detección de vulnerabilidades. Su enfoque es altamente especializado en este ámbito, lo que le permite identificar y evaluar de manera exhaustiva las posibles debilidades en aplicaciones web, además, automatiza

el proceso de escaneo de vulnerabilidades, lo que ahorra tiempo y recursos en comparación con enfoques manuales.

Zed Attack Proxy (ZAP). Es una herramienta gratuita de pruebas de penetración y es de código abierto**,** enfocada en probar aplicaciones web [85].

7.1.2. Inyección SQL

Es un ataque que consiste en insertar código SQL por medio de datos de entrada desde el cliente de la aplicación. El objetivo de este ataque es modificar las consultas originalmente realizadas por la aplicación (del prototipo propuesto) y crear otras totalmente diferentes para que el atacante obtenga información de otros usuarios o la modifique [86].

En el marco de esta investigación, fueron evaluadas diferentes herramientas de inyección SQL proporcionadas por el sistema operativo Kali Linux, ampliamente reconocido en la comunidad de seguridad informática.

Dentro de las opciones disponibles, las herramientas SQL: *Sqlmap y Burp Suite* sobresalieron por su relevancia en la detección de vulnerabilidades de inyección. La selección de la herramienta estuvo basó en una serie de consideraciones específicas, vinculadas tanto a la funcionalidad de la aplicación en cuestión como a la forma en que gestiona los recursos disponibles.

En un primer análisis, *Sqlmap* es una herramienta que automatiza el proceso de detección y explotación de vulnerabilidades de inyección SQL en aplicaciones web. Sin embargo, su funcionamiento requiere identificar el formulario a través del cual será realizada la inyección SQL. Lamentablemente, esta tarea fue un desafío insuperable en el contexto de la aplicación objeto de estudio. La razón subyace en la estructura de las *Uniform Resource Locator* (URL, Localizador de Recursos Uniformes) que conforman las rutas de la aplicación, las cuales no proporcionan información detallada acerca de la ubicación de los formularios vulnerables. Una URL es dirección exclusiva a un recurso que está disponible en internet [87].

Dadas las particularidades de la aplicación y las limitaciones identificadas en el proceso de identificación de formularios, fue seleccionada *Burp Suite* como la herramienta principal para llevar a cabo la detección y explotación de vulnerabilidades de inyección SQL en este estudio.

Burp Suite es una herramienta especializada en pruebas de penetración de aplicaciones web. Tiene dos versiones, una de ellas es gratuita y la otra requiere de un pago. Para las pruebas realizadas fue utilizada la versión gratuita, la cual ya está instalada por defecto en Kali Linux [88].

7.1.3. Fuerza bruta

Técnica que permite probar diferentes combinaciones posibles, usa técnicas de prueba y error para encontrar contraseñas, nombres de usuarios, claves cifradas e incluso páginas web ocultas [89].

Kali Linux ofrece herramientas como *Ncrack, Hydra, Wordlist* para realizar ataques de fuerza bruta. En el contexto de esta investigación, fue considerada la posibilidad de utilizar estas herramientas de Kali Linux para llevar a cabo ataques de fuerza bruta. No obstante, la decisión fue optar por utilizar una alternativa que ya había sido empleada anteriormente en el estudio, denominada *Burp Suite*. Esta selección está fundamentada en los conocimientos previamente adquiridos en el uso de "Burp Suite" durante la realización de un ataque previo.

Además, el hecho de que *Burp Suite* permite llevar a cabo ataques de fuerza bruta, al igual que las herramientas de Kali Linux mencionadas, hace que sea una opción conveniente y coherente con el enfoque de la investigación.

7.1.4. Pruebas de inicio de sesión biométrico

Son pruebas basadas en el uso de características físicas de los usuarios para acceder a una aplicación.

Al utilizar datos biométricos, el uso de ataques de “fuerza bruta” no es aplicable, ya que dichos ataques están relacionados principalmente a sistemas de autenticación

basados en usuario y contraseña, por ello, esta es la única prueba para la que no es utilizada ninguna herramienta, sino que son pruebas realizadas de forma manual, mostrando diferentes casos que un atacante puede usar para ingresar a la aplicación.

7.2. Recolección de datos

7.2.1. Análisis de vulnerabilidades

Para iniciar con las pruebas, fue necesario obtener una *URL* de la aplicación, que es proporcionada por *Ngrok* de manera temporal, como muestra la Figura 51.

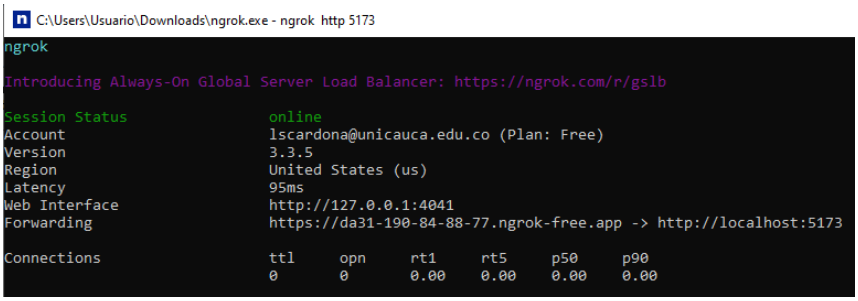


Figura 51. Aplicación lanzada usando *Ngrok.*

En primer lugar, es realizado un escaneo de vulnerabilidades en la aplicación (del prototipo propuesto), usando la herramienta denominada *ZAP* en la máquina virtual de *Kali Linux* y el enlace arrojado por la aplicación de *Ngrok* al momento de lanzar la aplicación, el escaneo es realizado de forma automática por la aplicación *ZAP* como muestra la Figura 52.

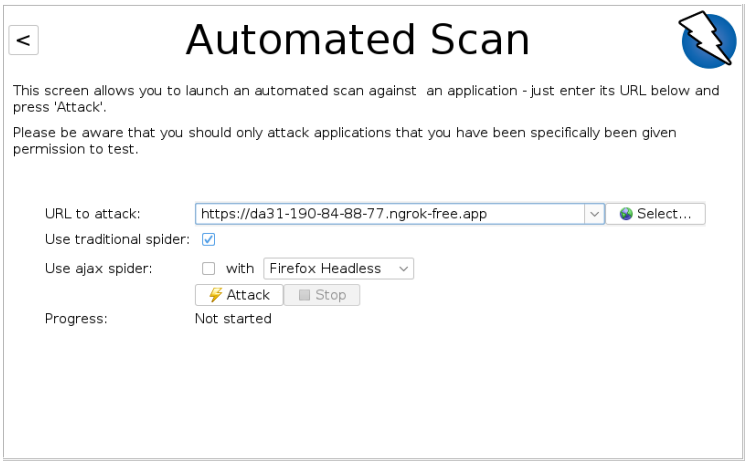


Figura 52. Escaneo automático de la aplicación ZAP.

Como resultado, la aplicación lanza alertas de acuerdo con las vulnerabilidades encontradas y las califica según su gravedad. La Figura 53 muestra un resumen de las vulnerabilidades encontradas, las cuales están descritas en la Tabla 14.

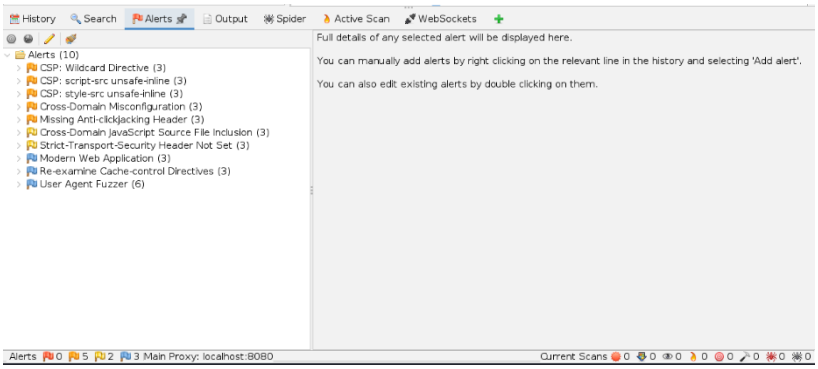


Figura 53. Alertas detectadas por ZAP.

Las vulnerabilidades detectadas por *ZAP* son mostradas en la parte izquierda de la Figura 53, mientras que la parte derecha de dicha figura, brinda una descripción de la vulnerabilidad y los detalles del ataque.

En la Figura 53, parte inferior izquierda, aparece un recuento de las alertas detectadas, marcadas de acuerdo con la categoría del riesgo, de la siguiente manera: bandera roja (alerta de alta prioridad), bandera naranja (alerta de prioridad media), bandera amarilla (alerta de prioridad baja), bandera verde (falso positivo) y bandera azul (alertas de prioridad informativa).

Tipo de vulnerabilidad	**Prioridad de alerta**	***Vulnerabilidad***	**Descripción**
Content-Security-Policy	Media	*Wildcard Directive*	La Política de Seguridad de Contenidos (CSP) es una capa adicional de seguridad que ayuda a detectar y mitigar ciertos tipos de ataques. Entre ellos, los ataques de secuencias de comandos en sitios cruzados (XSS) y de inyección de datos.
	Media	*script-src includes unsafe-inline.*	
	Media	*style-src includes unsafe-inline*	
Cross-Domain Misconfiguration	Media	*Cross-Domain Misconfiguration*	La carga de datos en el navegador web puede ser posible, debido a una mala configuración de Cross Origin Resource Sharing (CORS) en el servidor web.
Missing Anti-clickjacking Header	Media	*Missing Anti-clickjacking Header*	La respuesta no incluye ni *Content-Security-Policy* con la directiva *'frame-ancestors'* ni X-

			Frame-Options para protegerse de los ataques *'ClickJacking'*.
Cross-Domain JavaScript Source File Inclusion	Baja	*Cross-Domain JavaScript Source File Inclusion*	La página incluye uno o más archivos de script de un dominio de terceros.
Strict-Transport-Security Header Not Set	Baja	*Strict-Transport-Security Header Not Set*	HTTP *Strict Transport Security* (HSTS) es un mecanismo de política de seguridad web por el que un servidor web declara que los agentes de usuario que cumplan los requisitos (como un navegador web) deben interactuar con él utilizando únicamente conexiones HTTPS seguras (es decir, HTTP en capas sobre TLS/SSL).

Tabla 14. Alertas obtenidas en escaneo de vulnerabilidades.

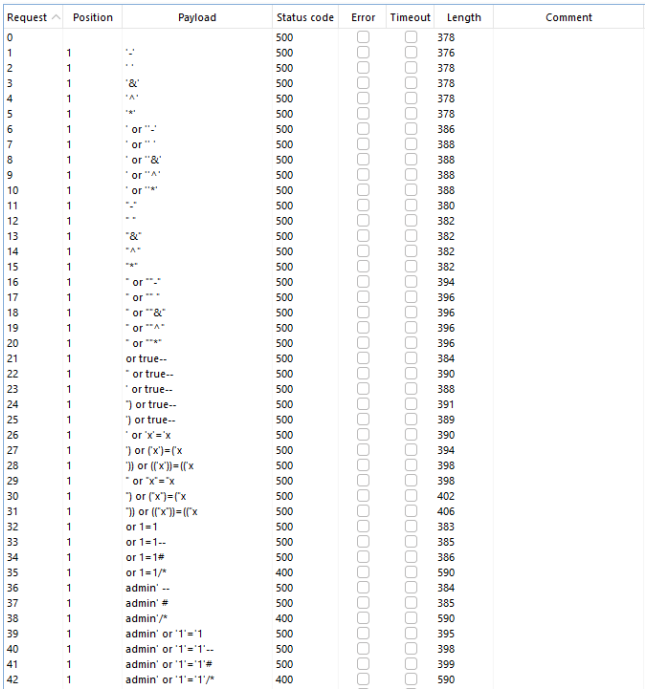
7.2.2. Prueba de inyección SQL

La prueba de inyección SQL fue realizada con *Burp Suite*.

Previo a la explicación de la ejecución de la prueba, resulta pertinente establecer la definición del término *payload,* el cual es utilizado en dicha prueba. Un *payload* es la carga que es ejecuta en una vulnerabilidad específica, en este caso, la prueba consiste en inyectar sentencias SQL a través de los formularios, los *payload* permiten que las herramientas de ataques de seguridad lo realicen de manera automática.

La prueba consistió en interceptar las peticiones HTTP relacionadas a los formularios implementados en la aplicación (del prototipo propuesto) e inyectar un intruso usando un *payload*.

La Figura 54 muestra algunos de los resultados obtenidos con las diferentes sentencias inyectadas.



Request	Position	Payload	Status code	Error	Timeout	Length	Comment
0			500			378	
1	1	'-'	500			376	
2	1	' '	500			378	
3	1	'&'	500			378	
4	1	'^'	500			378	
5	1	'*'	500			378	
6	1	' or ''-'	500			386	
7	1	' or ' '	500			388	
8	1	' or ''&'	500			388	
9	1	' or ''^'	500			388	
10	1	' or ''*'	500			388	
11	1	"-"	500			380	
12	1	" "	500			382	
13	1	"&"	500			382	
14	1	"^"	500			382	
15	1	"*"	500			382	
16	1	" or ""-"	500			394	
17	1	" or " "	500			396	
18	1	" or ""&"	500			396	
19	1	" or ""^"	500			396	
20	1	" or ""*"	500			396	
21	1	or true--	500			384	
22	1	" or true--	500			390	
23	1	' or true--	500			388	
24	1	") or true--	500			391	
25	1	') or true--	500			389	
26	1	' or 'x'='x	500			390	
27	1	') or ('x')=('x	500			394	
28	1	'])) or (('x'))=(('x	500			398	
29	1	" or "x"="x	500			398	
30	1	") or ("x")=("x	500			402	
31	1	"])) or (("x"))=(("x	500			406	
32	1	or 1=1	500			383	
33	1	or 1=1--	500			385	
34	1	or 1=1#	500			386	
35	1	or 1=1/*	400			590	
36	1	admin' --	500			384	
37	1	admin' #	500			385	
38	1	admin'/*	400			590	
39	1	admin' or '1'='1	500			395	
40	1	admin' or '1'='1'--	500			398	
41	1	admin' or '1'='1'#	500			399	
42	1	admin' or '1'='1'/*	400			590	

Figura 54. Resultados obtenidos en la prueba de inyección SQL.

El total de pruebas realizadas fueron 396 correspondientes a los casos identificados en el *payload* mencionado. Debido a la cantidad de resultados obtenidos, no fue posible detallar cada uno de ellos en este informe. En términos generales, es importante destacar que en ninguno de los casos fue recibida una respuesta del servidor en forma de 'OK' (*Status* 200). Las pruebas realizadas a la aplicación (del prototipo propuesto) demostraron que esta superó satisfactoriamente la prueba.

7.2.3. Ataque de fuerza bruta

Para los ataques de fuerza bruta son utilizados diccionarios los cuales tienen diferentes combinaciones, las cuales son intentos para ingresar a la aplicación.

Kali Linux cuenta con diferentes diccionarios predeterminados alojados en la ruta */usr/share/wordlists/,* de este directorio, el diccionario seleccionado fue *wifite.txt.*

Con ayuda de la herramienta *Burp Suite* fue realizado un ataque de fuerza bruta en el inicio de sesión del administrador. Cabe resaltar que este es el único lugar en la aplicación de *carpooling* desarrollada en el que un ataque de fuerza bruta puede ser realizado ya que los otros tipos de usuarios (conductor y pasajero) inician sesión en la aplicación por medio de datos biométricos.

La prueba fue ejecutada con 203807 opciones de posibles contraseñas con las que cuenta el diccionario seleccionado.

En la Figura 55 es posible apreciar el resultado de las pruebas realizadas. La columna resaltada en dicha Figura exhibe el valor del *payload* que fue sometido a evaluación, así como el código HTTP que generado como respuesta. Para la solicitud número 516, fue obtenido un estado HTTP 200. Este resultado es de suma importancia, ya que denota que fue posible vulnerar la seguridad del sistema, permitiendo el ingreso en calidad de usuario administrador.

Request	Payload	Status code	Error	Timeout	Length	Comment
501	veronika	500			383	
502	test1234	500			383	
503	teddybear	500			384	
504	sporting	500			383	
505	papillon	500			383	
506	nevermind	500			384	
507	marketing	500			384	
508	juliette	500			383	
509	gabrielle	500			384	
510	fuckyou2	500			383	
511	firewall	500			383	
512	evolution	500			384	
513	cristian	500			383	
514	cavalier	500			383	
516	admin123	200			400	
518	spongebob	500			384	
519	pa55w0rd	500			383	
520	halflife	500			383	
521	formula1	500			383	
522	dragonball	500			385	
523	thirteen	500			383	
524	stonecold	500			384	
525	rastaman	500			383	
526	mustang1	500			383	
527	cucumber	500			383	
528	skateboard	500			385	
529	sheridan	500			383	
530	qqqqqqqq	500			383	
531	punisher	500			383	
532	lovelife	500			383	
533	gretchen	500			383	
534	chevelle	500			383	

Figura 55. Resultado obtenido del ataque de fuerza bruta.

7.2.4. Pruebas de inicio de sesión biométrico

Las pruebas de autenticación biométrica fueron realizadas de forma manual. En este contexto, el proceso inició registrando a un usuario en la aplicación, tal como ilustra la Figura 56.

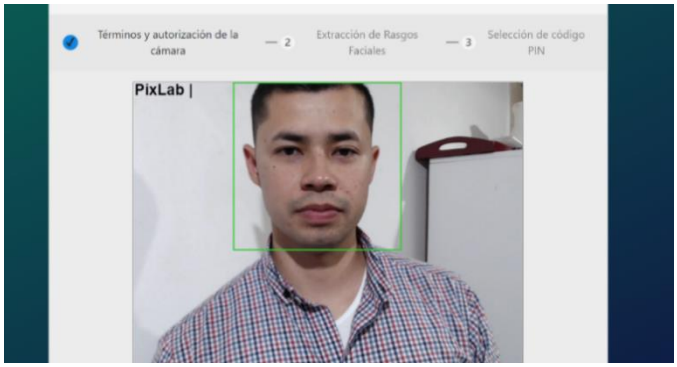


Figura 56. Registro de usuario de prueba.

Continuando con la prueba, procede a intentarse el acceso a la aplicación web (del prototipo propuesto) y utilizando la información previamente registrada, tal como exhibe la Figura 57. El resultado de esta operación de inicio de sesión fue exitoso,

puesto que la persona que había sido registrada previamente logró acceder exitosamente, como muestra la Figura 58.

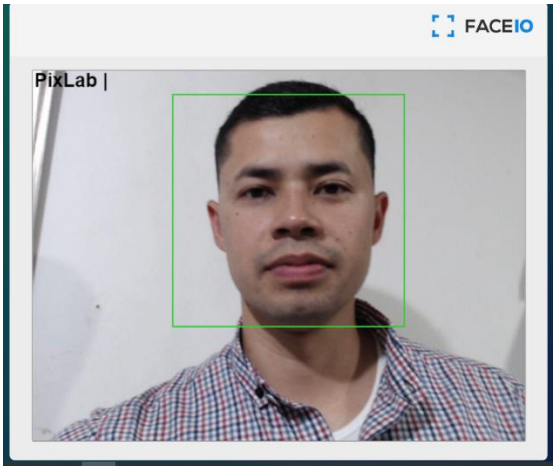


Figura 57. Intento de inicio de sesión con usuario de prueba.

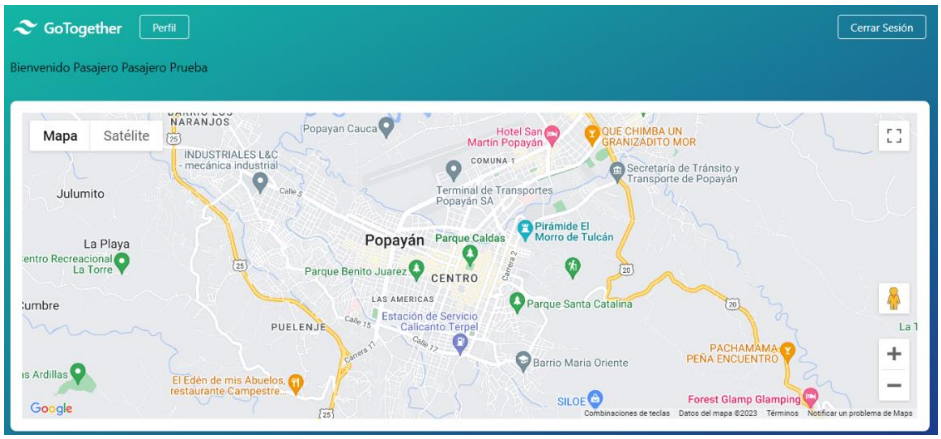


Figura 58. Inicio de sesión exitoso con usuario de prueba.

Otro escenario es representado en la Figura 59, donde es realizado un intento de inicio de sesión en la aplicación (del prototipo propuesto) por parte de una persona no registrada. El resultado obtenido es mostrado en la Figura 60, la cual evidencia que el usuario fue rechazado.

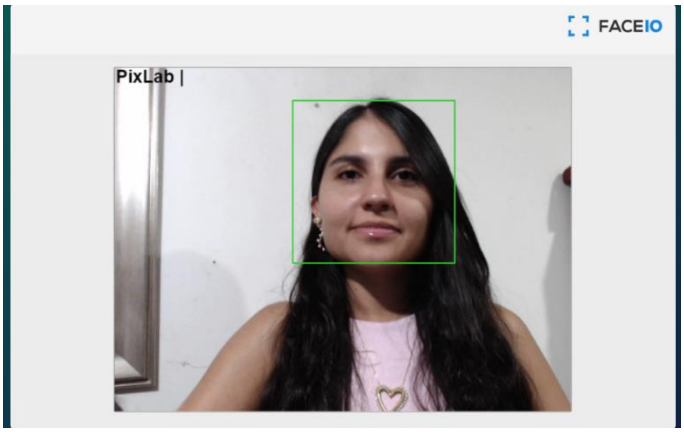


Figura 59. Intento de inicio de sesión con persona no registrada.

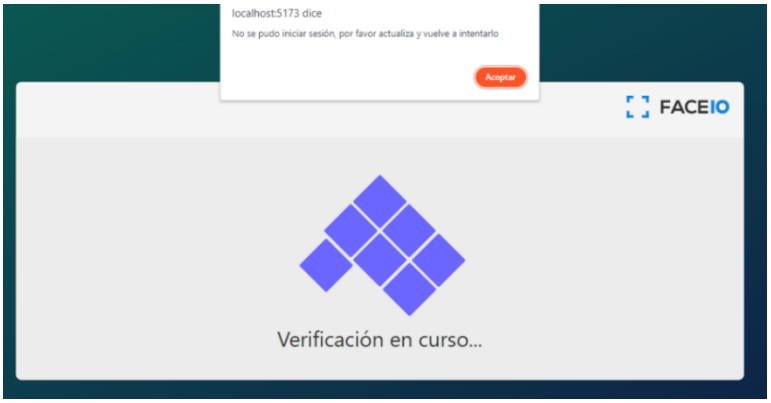


Figura 60. Resultado de inicio de sesión con persona no registrada.

La última prueba realizada consistió en utilizar una fotografía de la persona previamente registrada en la aplicación, tal como indica la Figura 61. Los resultados de esta evaluación están reflejados en la Figura 62, donde es evidente notar que la aplicación validó los datos biométricos presentados y permitió el acceso a la siguiente etapa en la aplicación, que corresponde al ingreso del PIN de seguridad. La herramienta *FaceIO* establece un límite de tiempo y un número específico de intentos para ingresar el código, lo que dificulta significativamente que los posibles atacantes accedan a la aplicación, ya sea como pasajeros o conductores.

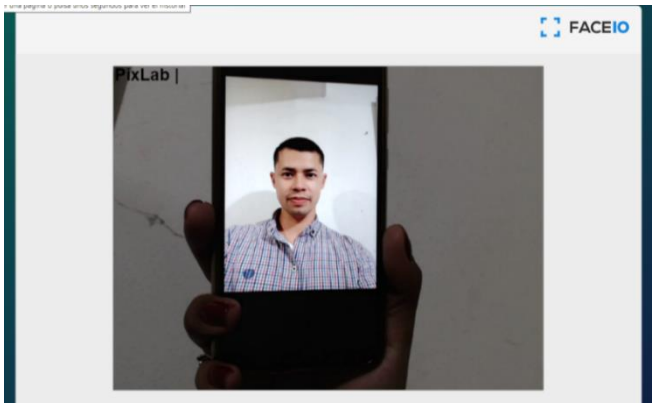


Figura 61. Intento de inicio de sesión con foto.

Figura 62. Resultado de inicio de sesión con foto.

7.3. Análisis y evaluación de resultados

7.3.1. Análisis de vulnerabilidades

Las pruebas de vulnerabilidades realizadas constatan que la aplicación no presenta vulnerabilidades significativas que puedan comprometer su seguridad, lo cual sucede por los niveles de protección implementados en la arquitectura. En primer lugar, la aplicación *Backend* de *Spring boot* proporciona una sólida capa de seguridad. Además, las dos capas adicionales que previenen la inyección de datos no autorizados.

La primera de estas capas (*Blockchain*) realiza una primera verificación para determinar si el usuario que intenta autenticarse posee datos biométricos válidos y almacenados previamente en la cadena de bloques, esto refuerza que sólo los usuarios que hayan hecho el registro de manera exitosa tengan acceso a la aplicación.

La segunda capa de seguridad es la autorización proporcionada por *FaceIO*, la cual está encarga de verificar la autenticidad de los datos biométricos, asegurando que provienen de una persona real que ha sido previamente registrado de manera exitosa en la aplicación.

7.3.2. Inyección de SQL

Las pruebas de inyección de SQL llevadas a cabo mediante la herramienta *Burp Suite* arrojaron resultados satisfactorios. En todos los casos sometidos a prueba, el servidor respondió con un código de error 500, lo que indica que la herramienta no logró efectuar peticiones exitosas al servidor, es decir un código de respuesta 200 (OK), el cual implicaría acceso a datos no autorizados. Por lo tanto, podemos concluir que esta prueba de seguridad fue superada con éxito. Este resultado refuerza la robustez de la aplicación frente a posibles ataques de inyección de SQL, asegurando así la integridad de los datos almacenados.

7.3.3. Ataque de fuerza bruta

Este escenario evidenció que la herramienta *Burp Suite* logró descifrar la contraseña "admin123", permitiendo así la obtención de una respuesta del servidor con un código

200 (OK). Esto sugiere que este tipo de ataques podría representar una amenaza potencial para la seguridad del módulo de autenticación del usuario administrador. No obstante, existen dos consideraciones a tener en cuenta:

- En primer lugar, es importante destacar que el módulo de autenticación diseñado para el usuario administrador no incluye autenticación biométrica y almacenamiento en la *Blockchain*, debido a que estas capas fueron implementadas únicamente para los usuarios conductores y pasajeros. Por lo tanto, si la autenticación biométrica y el almacenamiento en la cadena de bloques fuese utilizada también para el usuario administrador, los ataques de fuerza bruta carecerían de sentido debido a la imposibilidad de aplicar contraseñas generadas a un sistema que requiere datos biométricos para la autenticación.
- En segundo lugar, cabe señalar que la contraseña "admin123" es idéntica al nombre de usuario del administrador y contiene la palabra clave "admin". Esto indica que la contraseña utilizada posee un nivel de seguridad considerablemente bajo. Al proponer una contraseña alfanumérica con una combinación de caracteres aleatorios, el nivel de seguridad de la cuenta sería mayor.

7.3.4. Pruebas de inicio de sesión biométrico

Las pruebas realizadas en relación al proceso de autenticación biométrica evidenciaron que la aplicación es efectiva en prevenir intentos de suplantación de identidad. Al tratarse de una autenticación biométrica, que aprovecha la singularidad de las características de cada individuo y que limita significativamente la probabilidad de éxito de este tipo de ataques. No obstante, durante una de las pruebas, hubo una situación particular en la que la aplicación permitió el acceso al usuario utilizando una fotografía, lo que podría ser indicativo de una posible debilidad en la seguridad de la misma. Es relevante señalar que actualmente muchas empresas trabajan en la mejora de la autenticación biométrica, ya que este tipo de vulnerabilidad también ha sido identificado en algunos dispositivos móviles [90]. Sin embargo, es importante destacar que, la aplicación desarrollada, implementa un proceso de verificación de dos pasos, que requiere además de la identificación facial, la introducción de un PIN de seguridad. Esta característica adicional refuerza significativamente la seguridad de la aplicación.

En resumen, la evaluación de seguridad reveló que la aplicación presenta una sólida resistencia ante ataques de inyección SQL. Sin embargo, hay aspectos que requieren mejoras en lo que respecta a la autenticación y la gestión de contraseñas, particularmente en el caso del usuario administrador.

Es importante destacar que la comparación del prototipo del sistema *carpooling propuesto* (el cual incorpora autenticación biométrica y *Blockchain)*, con otros prototipos o sistemas similares no fue posible, ya que las características de dichos sistemas son diferentes y no utilizan unas métricas estandarizadas que faciliten dicha comparación.

Para finalizar, una propuesta para trabajos futuros es ampliar la seguridad de la aplicación mediante la verificación de los datos almacenados en la *Blockchain* en todas las vistas y funciones de la aplicación, no limitándose únicamente al proceso de registro y autenticación. Esto fortalecería la protección de los datos en todos los aspectos de la aplicación y reduciría el riesgo de accesos no autorizados o alteraciones de datos y la implementación de autenticación biométrica para todos los tipos de usuarios.

Capítulo 8.

8. Discusión de resultados

En una fase inicial, fue realizada una revisión sistemática de diversos artículos relacionados en cierta medida con el trabajo propuesto en el presente documento. A partir de dichos artículos, fueron seleccionados mecanismos y tecnologías aplicables al prototipo realizado y permitió notar, que en el ámbito de la aplicación de la tecnología *Blockchain* en el transporte, no han sido implementadas aplicaciones web enfocadas en *carpooling* que implementen registro y autenticación biométricas y el uso de *Blockchain*, por lo cual fue el enfoque del trabajo presentado.

Posteriormente, con base a la revisión sistemática, fue diseñado un módulo de registro y autenticación basado en la tecnología *Blockchain*.

Además, fue exitosa la integración con el prototipo inicial de una aplicación de carpooling. Esto posibilitó a los usuarios, ya sean conductores o pasajeros, realizar registros y autenticarse en la plataforma mediante información biométrica adquirida a través del reconocimiento facial.

La exitosa incorporación del módulo de registro y autenticación basado en *Blockchain* en el prototipo de la aplicación de *carpooling* no solo valida su aplicabilidad, sino que también evidencia que es fácilmente adaptable. Este módulo podría ser integrado en una amplia variedad de aplicaciones, brindando flexibilidad para su implementación en diferentes contextos. Además, su capacidad para funcionar en múltiples entornos demuestra su escalabilidad, ya que puede integrarse tanto en pequeñas aplicaciones como en sistemas más complejos.

Una de las ventajas fundamentales de la integración del módulo de autenticación radica en la descentralización inherente a la tecnología *Blockchain*, lo que otorga a las aplicaciones una mayor resistencia a ataques maliciosos y garantiza la inmutabilidad de los datos. Asimismo, la implementación de *Blockchain* promueve la privacidad de los usuarios, al permitir que los datos sensibles, como la información

biométrica, que es almacenada de manera segura y controlada por los propios usuarios. Esto, a su vez, fortalece la confidencialidad y la seguridad de la información, aspectos cruciales en aplicaciones que manejan datos sensibles. En particular, en el sector de la movilidad, la seguridad es una de las principales preocupaciones al utilizar un servicio de transporte. Este tipo de autenticación brinda seguridad al usuario, mejorando la confianza en la aplicación y la experiencia del usuario.

Este tipo de aplicaciones *carpooling* no solo promueven la eficiencia en el transporte, sino que también contribuyen a la disminución de emisiones de carbono, fomentando la movilidad sostenible.

La segunda parte del estudio consistió en pruebas de seguridad del aplicativo, y los resultados presentados en el Capítulo 6 arrojaron datos significativos. En concreto, pudo observarse que los métodos clásicos de ataque, tales como la inyección SQL, el ataque de fuerza bruta y el análisis de vulnerabilidades, solamente pudieron ser aplicados con éxito al perfil del administrador de la plataforma. Sin embargo, estas pruebas no pudieron realizarse en los perfiles de los pasajeros y conductores. Esta limitación es atribuida al hecho de que estos usuarios (conductores y pasajeros) pasan por un proceso de autenticación seguro en la aplicación (basado en *Blockchain*), basado en datos biométricos, específicamente la identificación facial; mientras que, para el usuario de tipo Administrador, por limitaciones de tiempo en el desarrollo, no fue incluido para realizar este tipo de registro y autenticación segura.

Los resultados obtenidos ponen de manifiesto que la autenticación facial puede representar una alternativa de autenticación relativamente segura, especialmente cuando es empleado en conjunto con un proceso de autenticación en dos pasos, como en el prototipo desarrollado. En este enfoque, además de la verificación facial, es necesario que el usuario introduzca un PIN, lo que añade una capa adicional de seguridad al proceso de autenticación.

Con respecto a las limitaciones que surgieron durante la implementación del prototipo y su módulo de autenticación, es necesario destacar, en primer lugar, la importancia del hardware requerido. La cámara utilizada para el proceso de registro y acceso a la aplicación, tanto para pasajeros como para conductores, desempeñó un papel crítico

en el funcionamiento del sistema. La resolución de esta cámara tuvo un impacto directo en la capacidad de detección facial de la aplicación. En casos en los que la cámara no contaba con una resolución adecuada, la aplicación enfrentaba dificultades para identificar y autenticar los rostros de los usuarios, lo que, a su vez, resultaba en la imposibilidad de llevar a cabo las operaciones de manera efectiva.

La gestión del tiempo en el desarrollo del proyecto adquirió una importancia crítica, especialmente en lo que respecta a ciertas tareas inicialmente planificadas, como la fase de despliegue. Dado el carácter integral del proyecto, que involucraba la comprensión y aplicación de conceptos en áreas tan diversas como *Blockchain*, *React* y Java. El tiempo requerido fue mayor al inicialmente estimado para la ejecución de todas las actividades requeridas, por lo cual fue necesario ajustes en la planeación para asignar más tiempo a ciertas actividades.

Capítulo 9.

9. Conclusiones

Se implementó con éxito un módulo de registro y autenticación biométrica utilizando contratos inteligentes en la *Blockchain* de *Ethereum*. Este módulo gestiona la información biométrica de los usuarios, lo que les brinda la capacidad de registrarse y autenticarse en aquellas aplicaciones web en las que integre esta característica. Esta implementación aporta cualidades esenciales como descentralización, seguridad y privacidad a estos procesos de autenticación y registro.

La exitosa implementación de una aplicación web de *carpooling*, que integró con éxito el módulo de registro y autenticación basado en *Blockchain*, resalta la importancia de aplicar las características de *Blockchain* a la seguridad de un servicio de transporte compartido. Esta integración no solo demuestra la aplicabilidad de la tecnología *Blockchain* en entornos de movilidad, sino que también resalta su papel fundamental en fortalecer la seguridad y la confianza de los usuarios al garantizar la integridad de los datos y la privacidad en un servicio crítico como el *carpooling*.

Los resultados obtenidos en las pruebas de seguridad aplicadas a la aplicación desarrollada, reflejaron un desempeño altamente satisfactorio. Estos resultados son particularmente alentadores ya que apuntan a la garantía de privacidad y seguridad en el contexto de una aplicación de *carpooling*, un tema que en la actualidad genera una creciente inquietud en la sociedad, en especial en lo que respecta a la salvaguardia de los datos personales. Estos hallazgos respaldan el principio fundamental de la *Blockchain*, que es caracterizada por su naturaleza descentralizada, lo que a su vez consolida y refuerza los aspectos mencionados en términos de privacidad y seguridad.

En cuanto a las futuras líneas de investigación y desarrollo, puede plantearse la perspectiva de implementar una aplicación de *carpooling* basada exclusivamente en la tecnología de *Blockchain* como sistema de gestión de datos. Además, notar la necesidad de investigar en profundidad el impacto de la inteligencia artificial en las

pruebas biométricas aplicadas a los usuarios que desempeñan roles de conductor y pasajero en la plataforma. Esto adquiere relevancia en un contexto donde los avances tecnológicos suceden a un ritmo acelerado, y existe la amenaza potencial de que la inteligencia artificial pueda ser utilizada para comprometer la seguridad del sistema.

Por otro lado, como otra dirección de investigación prometedora, puede considerarse la implementación de un sistema de PIN dinámico durante el registro y autenticación de usuarios. Esta medida podría ofrecer un importante impulso en términos de seguridad al proceso de autenticación, lo que resulta fundamental en un entorno digital cada vez más vulnerable a amenazas cibernéticas.

Finalmente, como propuesta futura, la implementación de la tecnología *Blockchain* en otras aplicaciones relacionadas con movilidad, como por ejemplo el transporte público, el cual es clave en el logro de una mejor movilidad en las ciudades. Es conveniente la integración del módulo de registro y autenticación desarrollado con la aplicación que sea pertinente, el cual tiene la adaptabilidad y escalabilidad requeridas para llevar a cabo esta tarea.

Referencias

[1] B. Jadhav, G. Mujumdar, and N. Jadhav, "*Applications of Artificial Intelligence in Machine Learning: Review," International Journal of Advanced Research in Science, Communication and Technology*, pp. 19–23, Jun. 2022. doi: https://doi.org/10.48175/ijarsct 5104

[2] B. Pineda, C. H. Muñoz, and H. Gil, "*Relevant aspects of the mobility and its relation with environment in the Valle de Aburrá: a review*," Ingeniería y Desarrollo, vol. 36, no. 2, pp. 489–508, Jul. 2018. doi: https://doi.org/10.14482/inde.36.2.10403

[3] C. Bresciani, A. Colorni, F. Costa, A. Luè and L. Studer, "*Carpooling: facts and new trends," 2018 International Conference of Electrical and Electronic Technologies for Automotive*, 2018, pp. 1-4, doi: 10.23919/EETA.2018.8493206.

[4] C. Manjarrez, E. Mogollon, I. D. Cortés, and L. Dussan. "Identificación de riesgos en el tratamiento de datos personales a nivel de usuarios clientes de aplicaciones móviles en el sector del transporte público individual en Bogotá," trabajo de grado, Universidad Católica de Colombia, Bogotá, 2016. Disponible: http://hdl.handle.net/10983/7831

[5] S. Bevilacqua and J. E. Neira-Villena, "Percepción del riesgo en viajes compartidos. Efectos de la «Uber Economía» en el transporte de taxis", Revista Escuela de Administración de Negocios, n.º 90, mayo de 2021. Doi: https://doi.org/10.21158/01208160.n90.2021.2875

[6] C. Sniffen, J. Durnan, and J. Zweig. *Helping industries to classify reports of sexual harassment, sexual misconduct, and sexual assault. Harrisburg, PA: National Sexual Violence Resource Center*. (2018), disponible: https://www.nsvrc.org/sites/default/files/publications/2018-11/NSVRC_HelpingIndustries.pdf

[7] J. Vázquez, "Uber reporta casi 4,000 reclamos por agresión sexual y mala conducta en periodo de pandemia", El Diario NY, 2 de julio de 2022. [En línea]. Disponible: https://eldiariony.com/2022/07/02/uber-reporta-casi-4000-reclamos-por-agresion-sexual-y-mala conducta-en-periodo-de-pandemia/

[8] *Uber Under the Hood. "Counting it is the first step towards ending it". Medium*. Disponible en: https://medium.com/uber-under-the-hood/counting-it-is-the-first-step-towards-ending-it-3b63163e1330

[9] *Between Public and Private Mobility: Examining the Rise of Technology-Enabled Transportation Services. Washington, D.C.: Transportation Research Board*, 2016. doi: https://doi.org/10.17226/21875

[10] Uber. "Seguridad de las mujeres", 2022. Disponible: https://www.uber.com/us/es/safety/womens safety/

[11] inDriver.com. "inDriver. Más beneficios que otros servicios". Disponible: https://indriver.com/es/city/

[12] "Superindustria exige a Uber fortalecer medidas de seguridad para proteger datos personales de colombianos | Superintendencia de Industria y Comercio". Superintendencia de Industria y Comercio, 2019, Disponible: https://www.sic.gov.co/Superindustria-exige-a-Uber-fortalecer-medidas-de seguridad-para-proteger-datos-personales-de-colombianos

[13] J. Friginal, S. Gambs, J. Guiochet and M.-O. Killijian, *"Towards privacy-driven design of a dynamic carpooling system", Pervasive and Mobile Computing*, vol. 14, pp. 71–82, octubre de 2014. Doi: https://doi.org/10.1016/j.pmcj.2014.05.009

[14] L. &. B. Créno, Cahour, "*Perceived risks and trust experience in a service of Carpooling*", en 22nd ITS *World Congress*, Bordeaux, Francia, 5–9 de octubre de 2015.

[15] H. Sun, L. Wei, L. Wang, J. Yin and W. Ma, "*A Trusted and Privacy-Preserving Carpooling Matching Scheme in Vehicular Networks", Journal of Information Security*, vol. 13, n. º 01, pp. 1–22, 2022. Disponible: https://doi.org/10.4236/jis.2022.131001

[16] S. Squarepants, "*Bitcoin: A Peer-to-Peer Electronic Cash System", SSRN Electronic Journal*, 2008. Doi: https://doi.org/10.2139/ssrn.3977007

[17] Y. Yuan and F. -Y. Wang, *"Towards Blockchain-based intelligent transportation systems,"* 2016 IEEE 19th *International Conference on Intelligent Transportation Systems* (ITSC), 2016, pp. 2663-2668, doi: 10.1109/ITSC.2016.7795984.

[18] "¿Qué son los contratos inteligentes en Blockchain? | IBM". IBM - Deutschland | IBM. https://www.ibm.com/co-es/topics/smart-contracts.

[19] N. D. Janakbhai, M. J. Saurin, and M. Patel. (2020). "*Blockchain-Based Intelligent Transportation System with Priority Scheduling". Lecture Notes on Data Engineering and Communications Technologies*, vol 52 pp. 311–317, 2021 doi: https://doi.org/10.1007/978-981-15-4474-3_34

[20] C. Bresciani, A. Colorni, F. Costa, A. Lue and L. Studer, *"Carpooling: facts and new trends*", en *2018 International Conference of Electrical and Electronic Technologies for Automotive*, Milan, 9–11 de julio de 2018. IEEE, 2018. doi: https://doi.org/10.23919/eeta.2018.8493206

[21] S. Auer, S. Nagler, S. Mazumdar and R. R. Mukkamala, "*Towards Blockchain-IoT based shared mobility: Car-sharing and leasing as a case study", Journal of Network and Computer Applications*, vol. 200, p. 103316, abril de 2022. Doi: https://doi.org/10.1016/j.jnca.2021.103316

[22] R. Salazar-Cabrera, Á. Pachón de la Cruz and J. M. Madrid Molina*, "Sustainable transit vehicle tracking service, using intelligent transportation system services and emerging communication technologies: A review", Journal of Traffic and Transportation Engineering (English Edition),* vol. 7, n. º 6, pp. 729–747, diciembre de 2020. [En línea]. Disponible: https://doi.org/10.1016/j.jtte.2020.07.003

[23] M. J. Page, D. Moher, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, . . . J. E. McKenzie. "*PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews"*, BMJ, p. 160, marzo de 2021. [En línea]. Disponible: https://doi.org/10.1136/bmj.n160

[24] P. M. Institute and P. M. Institute, *A Guide to the Project Management Body of Knowledge, Third Edition (PMBOK Guides). Newton Square: Project Management Institute*, 2004.

[25] M. Torrado, "Estudio de metodologías ágiles. Aplicación SCRUM", Trabajo de grado, Universidad de Sevilla. Escuela Técnica Superior de Ingeniería, Sevilla, 2019. Disponible: https://idus.us.es/bitstream/handle/11441/101313/TFG 2513%20TORRADO%20NEVADO.pdf?sequence=1& isAllowed=y

[26] "*What is Web3 and why is it important*? | ethereum.org". ethereum.org. https://ethereum.org/en/web3/.

[27] J. Cook. *"Intro to Ethereum* | ethereum.org". ethereum.org. https://ethereum.org/en/developers/docs/intro-to-ethereum#what-is-a-Blockchain.

[28] Ethereum.org". *"What is Ethereum?"* Disponible: https://ethereum.org/en/what-is-ethereum/.

[29] *"Smart Contracts* | ethereum.org". ethereum.org. https://ethereum.org/en/smart-contracts/.
[30] S. Shaheen and R. Finson, *"Intelligent Transportation Systems," UC Berkeley: Transportation Sustainability Research Center,* 2013. [Online]. Disponible: https://escholarship.org/uc/item/3hh2t4f9
[31] L. Figueiredo, I. Jesus, J. A. T. Machado, J. R. Ferreira and J. L. Martins de Carvalho, "*Towards the development of intelligent transportation systems*", en 2001 *IEEE Intell. Transp. Systems*. Proc., Oakland, CA, USA. Disponible: https://doi.org/10.1109/itsc.2001.948835
[32] AustriaTech. *"FRAME ARCHITECTURE". FRAME ARCHITECTURE*. https://frame-online.eu/.
[33] AustriaTech. "*The FRAME model* | *FRAME ARCHITECTURE". FRAME ARCHITECTURE*. http://frame-online.eu/frame-architecture/detailed-information/the-frame-model.
[34] *FRAME User Needs*. [En línea]. Disponible: http://frame-online.eu/wp-content/uploads/2014/10/FRAME-User-Needs-V4.1-01.pdf
[35] U.S. *Department of Transportation. "Architecture Reference for Cooperative and Intelligent Transportation". Architecture Reference for Cooperative and Intelligent Transportation.* https://www.arc-it.net/.
[36] U.S. *Department of Transportation. "Service Packages". Architecture Reference for Cooperative and Intelligent Transportation.* https://www.arc-it.net/html/servicepackages/servicepackages-areaspsort.html
[37] U.S. Department of Transportation. "*Dynamic Ridesharing and Shared Use Transportation". Architecture Reference for Cooperative and Intelligent Transportation.* https://www.arc-it.net/html/servicepackages/sp17.html#Tab-3.
[38] U.S. Department of Transportation. (s.f.). *Shared Use Mobility and Dynamic Ridesharing. Architecture Reference for Cooperative and Intelligent Transportation.* https://www.arc-it.net/html/servicepackages/sp17.html#tab-3.
[39] S. Shaheen, A. Cohen, and A. Bayen, "*The Benefits of Carpooling," UC Berkeley: Transportation Sustainability Research Center*, 2018. [Online]. Available: https://escholarship.org/uc/item/7jx6z631
[40] M. T, K. Makkithaya and N. V G, *"A Blockchain Based Decentralized Identifiers for Entity Authentication in Electronic Health Records", Cogent Engineering*, vol. 9, n. º 1, marzo de 2022. Doi: https://doi.org/10.1080/23311916.2022.2035134
[41] X. Xiang, M. Wang and W. Fan, "*A Permissioned Blockchain-Based Identity Management and User Authentication Scheme for E-Health Systems*," in IEEE *Access*, vol. 8, pp. 171771-171783, 2020, doi: 10.1109/ACCESS.2020.3022429.
[42] I. T. Javed, F. Alharbi, B. Bellaj, T. Margaria, N. Crespi and K. N. Qureshi, "*Health-ID: A Blockchain Based Decentralized Identity Management for Remote Healthcare", Healthcare*, vol. 9, n. º 6, p. 712, junio de 2021. Doi: https://doi.org/10.3390/healthcare9060712
[43] K. P. Satamraju and B. Malarkodi, "*A decentralized framework for device authentication and data security in the next generation internet of medical things*", *Computer Communications*, vol. 180, pp. 146–160, diciembre de 2021. doi: https://doi.org/10.1016/j.comcom.2021.09.012
[44] M. S. Eddine, M. A. Ferrag, O. Friha and L. Maglaras, *"EASBF: An efficient authentication scheme over Blockchain for fog computing-enabled internet of vehicles", Journal of*

Information Security and Applications, vol. 59, p. 102802, junio de 2021. Doi: https://doi.org/10.1016/j.jisa.2021.102802

[45] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan and Y. Zhang, *"Blockchain Empowered Cooperative Authentication With Data Traceability in Vehicular Edge Computing," in IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4221-4232, April 2020, doi: 10.1109/TVT.2020.2969722.

[46] X. Yang and W. Li, *"A zero-knowledge-proof-based digital identity management scheme in Blockchain", Computers & Security*, vol. 99, p. 102050, diciembre de 2020. Doi: https://doi.org/10.1016/j.cose.2020.102050

[47] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong and X. Lin, "*PTAS: Privacy-preserving Thin-client Authentication Scheme in Blockchain-based PKI", Future Generation Computer Systems*, vol. 96, pp. 185–195, 2019. doi: https://doi.org/10.1016/j.future.2019.01.026

[48] S. Y. Lim, P. T. Fotsing, O. Musa and A. Almasri, "*AuthChain: A Decentralized Blockchain-based Authentication System", International Journal of Engineering Trends and Technology*, pp. 70–74, octubre de 2020. Doi: https://doi.org/10.14445/22315381/cati1p212

[49] S. Panja and B. Roy, "*A secure end-to-end verifiable e-voting system using Blockchain and cloud server", Journal of Information Security and Applications*, vol. 59, p. 102815, 2021. doi: https://doi.org/10.1016/j.jisa.2021.102815

[50] "Inicio | ethereum.org". ethereum.org. [En línea]. Disponible: https://ethereum.org/es/

[51] *React, "React documentation*," React, 2021. [Online]. Available: https://reactjs.org/docs/getting-started.html

[52] N. Azeez, and O. Chinazo." Achieving data authentication with Hmac-SHA256 algorithm". GESJ: Computer Science and Telecommunications, vol. 2, n° 54, pp. 34-4, 2018.

[53] V. Buterin, "*A Next-Generation Smart Contract and Decentralized Application Platform," Ethereum White Paper*, 2014. [En línea]. Disponible: https://ethereum.org/en/whitepaper/.

[54] "*Documentation*". *ethers*. [En línea]. Disponible: https://docs.ethers.org/v5/

[55] "*Spring Boot*". Spring Boot. [En línea]. Disponible: https://spring.io/projects/spring-boot

[56] M. Thakur, "*Understanding Application Programming Interface (API) and Its Role in Web Development," International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 11, no. 2, pp. 135-139, 2021.

[57] "PostgreSQL". PostgreSQL. [En línea]. Disponible: https://www.postgresql.org/

[58] S. V. Hurtado Gil, "Representación de la arquitectura de software usando UML", Sistemas & Telemática - Universidad ICESI, p. 65. [En línea]. Disponible: https://www.icesi.edu.co/contenido/pdfs/shurtado_repres-uml.pdf

[59] R. Sharma and S. Chakraborty, "*BlockAPP: Using Blockchain for Authentication and Privacy Preservation in IoV," 2018 IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, 2018, pp. 1-6, doi: 10.1109/GLOCOMW.2018.8644389.

[60] I. M. Alsaadi, "Physiological Biometric Authentication Systems, Advantages, Disadvantages and Future Development: A Review", Int. J. Scientific & Technol. Res., vol. 4, n.º 12, pp. 285–289, 2015.

[61] E. Cherrier, S. Z. Syed Idrus, C. Rosenberger and J. Schwartzmann, "A Review on Authentication Methods", Australian J. Basic Appl. Sci., vol. 7, n.º 5, pp. 99–107, 2013.

[62] R. Coyne, "Cryptographic City: Decoding the Smart Metropolis," The MIT Press, 2023, DOI: 10.7551/mitpress/14712.001.0001.

[63] C. Dangare and Akila, Ms. (2016). IJARCCE An Android based application: Cab pooling. International Journal of Advanced Research in Computer and Communication Engineering. 5. 10.17148/IJARCCE.2016.53138.

[64] M. Bruglieri, D. Ciccarelli, A. Colorni and A. Luè, "PoliUniPool: a carpooling system for universities", Procedia - Social Behav. Sci., vol. 20, pp. 558–567, 2011. [En línea]. Disponible: https://doi.org/10.1016/j.sbspro.2011.08.062
[65] B. Akshay, G. Asmita, J. Kshetrapal and W. Archana, "Car Pool'up – Real-time Carpooling using GPS,".Proceedings of National Conference on New Horizons in IT - NCNHIT, pp. 126-129, 2013.
[66] X. Xiang, M. Wang and W. Fan, "A Permissioned Blockchain-Based Identity Management and User Authentication Scheme for E-Health Systems," in IEEE Access, vol. 8, pp. 171771-171783, 2020, doi: 10.1109/ACCESS.2020.3022429.
[67] S. Barra, M. De Marsico, C. Galdi, D. Riccio and H. Wechsler, "FAME: Face Authentication for Mobile Encounter," 2013 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications, Napoli, Italy, 2013, pp. 1-7, doi: 10.1109/BIOMS.2013.6656140.
[68] A. Menzinsky, G. López, J. Palacio, M. Á. Sobrino, R. Álvarez, and V. Rivas. (2020). Historias de Usuario. Ingeniería de Requisitos Ágil (3a ed.). Scrum Manager.
[69] N. A. Shamat, S. Sulaiman, and J. S. Sinpang, *"A Systematic Literature Review on User Interface Design for Web Applications"*, JTEC, vol. 9, no. 3-4, pp. 57–61, Oct. 2017.
[70] F. Knirsch, A. Unterweger and D. Engel. *Implementing a Blockchain from scratch: why, how, and what we learned. EURASIP J. on Info. Security* 2019, 2 (2019). https://doi.org/10.1186/s13635-019-0085-3.
[71] S. Panda, D. Jena and P. Das. (2021). *A Blockchain-Based Distributed Authentication System for Healthcare. International Journal of Healthcare Information Systems and Informatics.* 16. 1-14. 10.4018/IJHISI.20211001.oa12
[72] V. Buterin, "*A Next-Generation Smart Contract and Decentralized Application Platform," Ethereum White Paper, 2014.* [En línea]. Disponible: https://ethereum.org/en/whitepaper/.
[73] FaceIO, *"FaceIO API documentation" FaceIO,* 2021. [Online]. Available: https://face.io/docs/api.
[74] "*Documentation | Ethereum development environment for professionals by Nomic Foundation". Hardhat | Ethereum development environment for professionals by Nomic* Foundation. https://hardhat.org/docs.
[75] Remix - Ethereum IDE 1 documentation". Welcome to Remix's documentation! — Remix - Ethereum IDE 1 documentation. [En línea]. Disponible: https://remix-ide.readthedocs.io/en/latest/
[76] C. M. Zapata, C. Palacio, N. Olaya. Unc-Analista: Hacia La Captura De Un Corpus De Requisitos A Partir De La Aplicación Del Experimento Mago De Oz. Rev.EIA.Esc.Ing.Antioq [En línea]. 2007, n.7 [cited 2023-04-25], pp.25-40. Disponible: <http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1794-12372007000100003&lng=en&nrm=iso>. ISSN 1794-1237.
[77] A. Menzinsky, G. López, J. Palacio, M. Á. Sobrino, R. Álvarez and V. Rivas, Historias de usuario - Ingeniería de requisitos ágil, 3a ed. Scrum Manager, 2022. [En línea]. Disponible: https://www.scrummanager.com/files/scrum_manager_historias_usuario.pdf
[78] S. Raghavan, G. Zelesnik and G. Ford, *"Lecture Notes on Requirements Elicitation"*, *Defense Technical Information Center, Fort Belvoir,* VA, marzo de 1994. [En línea]. Disponible: https://doi.org/10.21236/ada278536
[79] K. Schwaber and J. Sutherland, *"The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game,"* Scrum.org, Nov. 2020. [En línea]. Disponible: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

[80] C. Cordoba-Cely, "La experiencia de usuario: de la utilidad al afecto", iconofacto, vol. 9, n.º 12, p. 14, 2013.
[81] “Kali Linux | *Penetration Testing and Ethical Hacking Linux Distribution”*. Kali Linux. [En línea]. Disponible: https://www.kali.org/
[82] “ngrok | *Unified Ingress Platform for Developers”. ngrok* | *Unified Ingress Platform for Developers.* [En línea]. Disponible: https://ngrok.com/
[83] SayNet. (s.f.). ¿Qué es un análisis de vulnerabilidades? – SAYNET. SAYNET – Seguridad y Tecnología para tu Empresa. https://saynet.com.mx/que-es-un-analisis-de-vulnerabilidades/
[84] L. H. Beltrán and B. F. Ramirez Análisis de vulnerabilidades en aplicaciones WEB de una empresa de Automatización industrial e IOT, para la prevención de ataques Cibernéticos. [online]. Disponible en: http://hdl.handle.net/20.500.12495/8104.
[85] *ZAP. Open Web Application Security Project. “Zed Attack Proxy”.* [En línea]. Disponible: https://www.zaproxy.org/
[86] Hostalia. “Ataques de inyección SQL”. Hostalia – Pressroom | Casos de clientes hostalia y mucho más en prensa. [En línea]. Disponible: https://pressroom.hostalia.com/contents/ui/theme/images/inyeccion-sql-wp-hostalia.pdf
[87] S. White, T. Petersen, D. Coulter, D. Batchelor, M. Jacobs and M. Satran. “Control de localizadores uniformes de recursos - Win32 apps”. Microsoft Learn: Build skills that open doors in your career. [En línea]. Disponible: https://learn.microsoft.com/es-es/windows/win32/wininet/handling-uniform-resource-locators
[88] PortSwigger Ltd. *“Burp Suite - Application Security Testing Software”. Web Application Security, Testing, & Scanning* - PortSwigger. [En línea]. Disponible: https://portswigger.net/burp
[89] L. Bosnjak, J. Sres and B. Brumen, *“Brute-force and dictionary attack on hashed real-world passwords”, en 2018 41st Int. Conv. Inf. Communication Technology, Electron. Microelectronics* (MIPRO), Opatija, 21–25 de mayo de 2018. IEEE, 2018. [En línea]. Disponible: https://doi.org/10.23919/mipro.2018.8400211
[90] Samsung. “¿se puede desbloquear el dispositivo mediante imágenes o video teniendo bloqueado de reconocimiento facial?” Accedido el 13 de octubre de 2023. [En línea]. Disponible: https://www.samsung.com/cl/support/mobile-devices/can-the-device-be-unlocked-by-images-or-video-with-facial-recognition-locked/

Anexo A. Revisión sistemática

El Anexo A presenta en detalle las fases de la revisión sistemática de la literatura realizada.

Disponible en el siguiente enlace:
https://drive.google.com/file/d/1oRmXEyk7jZ85c8eWhxNGAg4WNHDh1UZD/view?usp=sharing

Anexo B. Desarrollo de un prototipo de una aplicación web de *carpooling*

El Anexo B consigna de forma detallada las HU de la aplicación de *carpooling,* el desarrollo del prototipo de la aplicación web y las pruebas de funcionamiento respectivas.

Disponible en el siguiente enlace:
https://drive.google.com/file/d/1gv-VKeDN52HAF-fOQQw88MDLufepZ6Yl/view?usp=drive_link