# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

**TITULO DEL TRABAJO DE TITULACIÓN**

Desarrollo de una aplicación web con tecnología blockchain para la gestión de transacciones financieras descentralizadas entre usuarios a través de un prototipo de criptomoneda.


**AUTOR**

**Reyes Santos Yandry Efrain**


EXAMEN COMPLEXIVO

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

**TUTOR**

**Ing. Coronel Suárez Iván Alberto Mgt.**

**Santa Elena, Ecuador**

**Año 2025**

UPSE

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

**TRIBUNAL DE SUSTENTACIÓN**

Ing. José Sánchez Aquino, Mgt.
**DIRECTOR DE LA CARRERA**

Ing. Iván Coronel Suárez, Mgt.
**DOCENTE TUTOR**

Ing. Carlos Castillo Yagual, Mgt.
**DOCENTE ESPECIALISTA**

Ing. Marjorie Coronel Suárez, Mgt.
**DOCENTE GUÍA UIC**

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

## CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **REYES SANTOS YANDRY EFRAIN**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 20 días del mes de junio del año 2025

**TUTOR**

Firmado electrónicamente por:
IVAN ALBERTO
CORONEL SUAREZ
Validar únicamente con FirmaEC

**Ing. Coronel Suarez Iván Alberto Mgt.**

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, **REYES SANTOS YANDRY EFRAIN**

**DECLARO QUE:**

El trabajo de Titulación, implementación de una blockchain y una criptomoneda para aplicaciones financieras y tecnológicas previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 20 días del mes de junio del año 2025

**EL AUTOR**

**Yandry Efrain Reyes Santos**

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

## CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Desarrollo de una aplicación web con tecnología blockchain para la gestión de transacciones financieras descentralizadas entre usuarios a través de un prototipo de criptomoneda, presentado por el estudiante, REYES SANTOS YANDRY EFRAIN fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 6 %, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.


**TUTOR**

Firmado electrónicamente por:
IVAN ALBERTO CORONEL SUAREZ
Validar únicamente con FirmaEC

**Ing. Coronel Suarez Iván Alberto Mgt.**

# UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
# FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

## AUTORIZACIÓN

Yo, **YANDRY EFRAIN REYES SANTOS**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del presente trabajo de titulación con fines de difusión pública, además apruebo la reproducción dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 20 días del mes de junio del año 2025

**EL AUTOR**

**Yandry Efrain Reyes Santos**

## AGRADECIMIENTO

En primer lugar, agradezco a Dios por permitirme llegar a dar este importante paso en mi vida, brindándome salud, inteligencia y dándome las fuerzas necesarias para continuar hasta el final.

En segundo lugar, agradezco a mi familia por su respaldo constante, su comprensión y por motivarme a seguir adelante.

Y, por último, a compañeros, amigos, docentes, tutores, dirigentes y todos quienes forman parte de la comunidad universitaria, los cuales fueron parte esencial en mi formación académica.

*Yandry Efrain, Reyes Santos*

**DEDICATORIA**

Le dedico este trabajo a mi familia por creer en mí, por su comprensión y paciencia que me permitieron seguir adelante día a día y no rendirme en el camino.

Una dedicación especial para mis padres Estelly Santos y Gerson Reyes por su apoyo incondicional a lo largo de toda mi carrera, también por ser mi motivación diaria para conseguir mis metas.

*Yandry Efrain, Reyes Santos*

**ÍNDICE GENERAL**

## ÍNDICE DE TABLAS

## ÍNDICE DE FIGURAS

**RESUMEN**

Este proyecto, que lleva por tema “Desarrollo de una aplicación web con tecnología blockchain para la gestión de transacciones financieras descentralizadas entre usuarios a través de un prototipo de criptomoneda”, tiene como objetivo crear una blockchain y una criptomoneda ambas simuladas, así como el desarrollo de un aplicativo web que interactúe con el sistema, con esto se pretende comprender el funcionamiento de estas nuevas tecnologías. Mediante revisión documental y la observación participante, se busca indagar a fondo los conceptos clave y tecnologías para el desarrollo del proyecto. Los resultados muestran las fortalezas del uso de blockchain en el sistema financiero actual, así como la capacidad de implementar esta tecnología en otras áreas gracias a los principios que posee, tales como son la descentralización, transparencia, inmutabilidad y seguridad. En conclusión, el uso de tecnología blockchain es fundamental en el desarrollo de nuevas tecnologías, no solo en el área financiera, sino en diversos campos.

**Palabras claves:** Blockchain, Criptomonedas, Descentralización.

**ABSTRACT**

This project, entitled “Development of a web application with blockchain technology for the management of decentralized financial transactions between users through a cryptocurrency prototype”, aims to create a blockchain and a cryptocurrency both simulated, as well as the development of a web application that interacts with the system, with this is intended to understand the operation of these new technologies. Through documentary review and participant observation, we seek to thoroughly investigate the key concepts and technologies for the development of the project. The results show the strengths of the use of blockchain in the current financial system, as well as the ability to implement this technology in other areas thanks to its principles, such as decentralization, transparency, immutability and security. In conclusion, the use of blockchain technology is fundamental in the development of new technologies, not only in the financial area, but in various fields.

**Keywords:** Blockchain, Cryptocurrencies, Decentralization.

# INTRODUCCIÓN

El dinero es una herramienta indispensable para el ser humano y cada vez se vuelve más esencial tener un grado de educación financiera para saber administrarlo y no caer en ciclos de deudas por tomar malas decisiones. El dinero en efectivo siempre ha sido la forma tradicional desde sus orígenes como método de pago, pero desde un tiempo se ha vuelto conocido el uso de dinero digital de diversos tipos. En sus inicios, las criptomonedas no llamaban tanto la atención, pero desde la popularización de Bitcoin allá por el año 2020, se han convertido en una alternativa viable al uso de dinero tradicional, aunque esto a su vez trajo consigo un sinnúmero de problemas relacionados con casos de estafas, es por esta razón que la autoeducación en temas financieros juega un papel fundamental.

Más allá de temas de criptomonedas que sí, es un tema innovador y llamativo, es crucial conocer la tecnología que esta por detrás, la cuál es la Blockchain también conocida como cadena de bloques es una tecnología en auge en la actualidad debido a sus beneficios y aplicabilidad en diversas áreas. La cadena de bloques no es más que un registro de información tipo libro diario de contabilidad, y sus principales beneficios son la descentralización, seguridad, inmutabilidad y transparencia.

En el presente proyecto se aborda la creación de una blockchain simulada con código Python que permita la sincronización entre los nodos de la red para llevar un correcto conceso y garantizar la integridad de las transacciones y de los bloques de la cadena ante posibles intentos de alteraciones. Además de esto, se llevará a cabo la simulación de una criptomoneda volátil esto mediante el desarrollo una aplicación web que permita el envío, recepción y visualizar el historial de transacciones simulando una dirección de billetera y saldos ficticios.

Es primordial considerar que este proyecto no se implementará en el ecosistema real de blockchain, además la criptomoneda es una simulación y no tendrá un valor real en el mercado. El objetivo del proyecto pretende brindar los conocimientos básicos de blockchain, criptomonedas y aplicarlos mediante la simulación de una cadena de bloques interconectada a una aplicación web con el papel de billetera de criptomonedas aprovechando los beneficios de este sistema descentralizado, sentando las bases para una futura implementación.

# CAPÍTULO 1. FUNDAMENTACIÓN

## 1.1. Antecedentes

La blockchain también conocida en español como cadena de bloques es una lista de récords que crece continuamente, vinculada y asegurada criptográficamente [1]. La forma como almacena la información es parecida a un libro de registro contable pudiendo ser estos datos de cualquier tipo como, transacciones, NFT o contratos inteligentes [1].

El boom de las criptomonedas tuvo lugar en el año 2020 con la repentina subida del valor del Bitcoin, una de las criptodivisas más grandes por su capitalización de mercado, llegando hasta alrededor de los sesenta mil dólares, popularizándose así de manera instantánea y siendo noticia mundial [2].

Aparte de este principal referente también existían otras criptomonedas como Ethereum y Solana, así mismo hay un tipo de monedas llamadas meme coins que básicamente son producto de bromas o chistes de internet que no tienen respaldo e invertir en ellas conlleva un riesgo altísimo. [2]. Gracias a que Bitcoin se popularizó en su tiempo todas las demás monedas también tuvieron un aumento en su precio, pero las meme coins como era de esperarse no duraron mucho y en un corto plazo perdieron su valor, esto debido a que no contaban con un proyecto por detrás que las respaldara y por ende estas meme coins tampoco tenían una utilidad clara [2].

Actualmente cuando las personas escuchan hablar de criptomonedas piensan y las asocian inmediatamente con estafas, esto se debe al desconocimiento del tema y a la cantidad de fraudes relacionados [3]. No es secreto para nadie que esto tiene algo de veracidad, debido a que la gente desconoce cómo es que funcionan este tipo de criptodivisas y los delincuentes informáticos o estafadores se aprovechan de esto para venderles promesas falsas, ya sean de rendimientos absurdamente altos en poco tiempo o incitarles a comprar alguna moneda, para posteriormente los dueños que están por detrás de ese proyecto procedan a vender todas las criptomonedas o tokens y se quedan con todo el dinero, lo que se conoce como "rug pull" en el mundo cripto [3].

Una de las estafas más sonadas en los últimos tiempos fue la que tuvo lugar en Argentina según la BBC News Mundo, se trata de una criptomoneda llamada $LIBRA la cual habría sido promocionada por el presidente Javier Milei, debido a esto la criptomoneda habría tenido una subida drástica en cuestión de minutos, para posteriormente desplomarse el valor de la misma un 90%, esto sucedió porque los dueños del proyecto hicieron ventas

millonarias robándose así aproximadamente 100 millones de dólares, esto afecto a todo tipo de inversores que habrían confiado en este activo digital, el presidente negó vínculos con el proyecto y ahora afronta demandas penales en su contra [4].

El presente proyecto busca crear una blockchain y una criptomoneda mediante código Python, dejando como base un producto funcional con potencial de crecimiento a largo plazo, con la posibilidad de lanzar esta criptomoneda al mercado en el futuro, asignándole una utilidad concreta, lo cual se logra con la integración de un proyecto sólido por detrás de esa criptomoneda.

La blockchain al ser descentralizada no depende de las entidades financieras tradicionales, esto da pie a reducir costos y aumentar la velocidad de las transacciones [5]. La blockchain debido a como está estructurada brinda integridad y seguridad de las operaciones lo que es crucial para un entorno financiero confiable [6].

### 1.2. Descripción del Proyecto

El presente proyecto tiene como finalidad el desarrollo e implementación de una blockchain y sobre esta misma crear un prototipo de criptomoneda, esto mediante el uso del lenguaje de programación Python, uno de los más usados actualmente debido a su sintaxis sencilla, versatilidad y disponibilidad de herramientas para realizar distintos tipos de proyecto en diversas ramas [7], para posteriormente poder hacer pruebas intercambiando la criptomoneda creada entre un grupo limitado de usuarios mediante el uso de una aplicación web simulando una billetera de criptomonedas [8].

Cabe aclarar que la parte práctica se hará de manera local en la PC, para la creación de la blockchain y la criptomoneda primero se estudiarán los conceptos básicos que hay por detrás de estas tecnologías como, por ejemplo, la descentralización, la transparencia, la inmutabilidad y demás términos que se detallan en el capítulo II [9].

Las pruebas se realizarán en una red local, simulando el funcionamiento de una blockchain y manteniendo en ejecución en segundo plano los conceptos clave para su correcto funcionamiento. Esto incluye la simulación de nodos en terminales dentro de una misma computadora o en varias computadoras. Del mismo modo, se simulará el minado de bloques y las transacciones entre usuarios de la red local.

Para la realización de este proyecto, se contará con distintas fases, cada una de ellas compuesta por varios módulos como se muestra a continuación:

**Fase de Análisis y Planificación**

Análisis de requisitos

- Definición de funcionalidades básicas:
    - Enviar y recibir criptomonedas.
    - Validación de transacciones.
    - Consulta de historial de transacciones.
    - Generación y escaneo de direcciones de billeteras.
    - Autenticación de usuarios.
    - Recuperación de contraseñas.
    - Propagación de transacciones entre nodos de la red.
    - Propagación de cadena de bloques válida entre nodos.
    - Corrección de cadena alterada o vulnerada.
- Roles y elementos del sistema:
    - Usuarios: Realizan transacciones desde la app web.
    - Nodos: Validan transacciones, almacenan la cadena y dependiendo del tipo, pueden minar nuevos bloques.
    - API: Encargada de enlazar la aplicación web con la blockchain, el backend y con la base de datos.
- Tecnologías a utilizar:
    - Backend: Flask y Python.
    - Blockchain: Estructura con proof of work.
    - Web: Angular.
    - Comunicación: API REST y JSON.

**Fase de diseño**

Diseño de la Arquitectura del sistema

- Comunicación directa: Usuarios > API > Blockchain en los nodos.
- Minería automática en segundo plano.

Rutas esenciales de la API REST

- Endpoints:
    - POST: enviar información de la transacción a la blockchain, insertar transacciones y enviar credenciales de usuario a la base de datos.
    - GET: historial (usuario) > Ver historial de transacciones.

- GET: minar > Minar bloques.

Diseño de la criptomoneda

- Minería automática después de cada transacción.
- Nombre y símbolo de la moneda.

**Fase de desarrollo**

Desarrollo de la blockchain

- Implementación de bloques (hash, timestamp, datos de transacción).
- Minado automático de bloques con PoW.
- Validación de integridad (cada bloque enlazado correctamente).

Desarrollo de la API REST

- Implementación de los 3 endpoints principales.
- JSON para almacenar datos.
- Seguridad (tokens simples o clave secreta para validar transacciones).

Desarrollo de la aplicación web

- Pantalla para enviar dinero (Formulario simple: destinatario y cantidad).
- Pantalla de historial (Consulta la API y muestra transacciones).
- Pantalla para registro e inicio de sesión.
- Pantalla para intercambiar criptomonedas.

**Fase de Pruebas y Validación**

Pruebas de Funcionalidad

- Prueba de transacciones básicas (enviar, recibir, ver historial).
- Validación de bloques generados correctamente.

Pruebas de Seguridad

- Evitar transacciones duplicadas (control de hashes).
- Validación de usuario antes de transacción (clave o token básico).

### 1.3. Objetivos del Proyecto

#### 1.3.1. Objetivo General

Implementar una simulación blockchain con código Python mediante una aplicación web para gestión de transacciones, permitiendo evaluar su funcionamiento e integridad en un entorno controlado.

### 1.3.2. Objetivos específicos

- Implementar una blockchain en Python simulando nodos de una red descentralizada mediante consolas, para validar, almacenar transacciones y minar bloques.
- Simular el desarrollo de una criptomoneda sobre la blockchain para realizar transacciones de envío, recepción e intercambio de monedas en un entorno controlado mediante una API y una aplicación web.
- Evaluar el sistema implementado mediante las pruebas de integridad de la blockchain.

## 1.4. Justificación del Proyecto

Hoy en día, las criptomonedas y la tecnología blockchain se han vuelto cada vez más reconocidas y utilizadas en diversas áreas, especialmente en el sector financiero, gracias a su enorme potencial para transformar el modo en que se llevan a cabo transacciones [10]. A su vez, el desconocimiento de estas nuevas e innovadoras tecnologías ha traído consigo grandes desafíos como la desconfianza y la especulación. Retos que se buscan abordar en este proyecto para demostrar que en buenas manos se pueden lograr soluciones innovadoras [11].

Algunas de las ventajas más importantes de la blockchain son la descentralización y la seguridad, debido a que no es necesario un intermediario financiero tradicional (bancos) para poder realizar transacciones de manera segura y eficiente [12]. Además de las criptomonedas, en una blockchain se pueden realizar diversas aplicaciones, tales como contratos inteligentes o gestión de datos [12].

El enfoque de pruebas a nivel local nos brinda la capacidad de evaluar y mejorar la funcionalidad en un entorno controlado, pudiendo así identificar y corregir posibles errores, mejorando así su seguridad y eficiencia antes de considerar su implementación en un entorno real [13].

Aunque el proyecto se llevará a cabo en un entorno local, su potencial de crecimiento es amplio; por ejemplo, se podrían agregar funciones como contratos inteligentes o gestión de activos digitales a futuro [14]. Así mismo, la posibilidad de contar con colaboraciones de entidades financieras o tecnológicas, personas de alto valor o con un gran poder adquisitivo, que a su vez apoyen estas tecnologías o que respalden la criptomoneda para que el proyecto tenga credibilidad.

## 1.5. Alcance del Proyecto

En este proyecto se creará una blockchain siguiendo los principios más importantes para la correcta funcionalidad de la cadena de bloques; así mismo, se creará un prototipo de una criptomoneda [14]. Por último, las pruebas se realizarán con un grupo pequeño de usuarios en un entorno local.

El lenguaje de programación principal a usar es Python debido a su variedad de librerías, funcionalidades y aplicaciones en diferentes ámbitos, además de la gran comunidad que hay por detrás, que nos sirve para tener soporte en caso de problemas en el desarrollo [7].

Las pruebas se realizarán de manera local, ya sea en una o más computadoras que ejecutarán los nodos de la blockchain, lo que simulará la descentralización, creación de bloques, validación de transacciones y dará seguridad a la red [13].

El proyecto no se implementará en una red pública en internet; con respecto a la seguridad, tendrá lo básico sin llegar a tomar medidas avanzadas como auditorías de código o pruebas de penetración [13]. Además, la criptomoneda no se integrará en exchanges reales; solo se trabajará de manera local.

Finalmente, este proyecto nos servirá de prueba para entender cómo funciona una blockchain y una criptomoneda por detrás, y así conocer los procesos que se deben seguir para que todos los principios se cumplan, aprovechando así las fortalezas que caracterizan a una cadena de bloques [12]. Se espera que al término del proyecto se cuente con una plataforma funcional que permita transacciones entre usuarios dentro de una red local.

# CAPÍTULO 2. MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO

## 2.1. Marco Conceptual

La implementación de una blockchain y una criptomoneda mediante el uso de código Python para fines tecnológicos y financieros conlleva la incorporación de múltiples tecnologías y herramientas que se desglosan a continuación.

### 2.1.1. Python

Es un considerado un lenguaje de muy alto nivel, debido a la simplicidad y legibilidad de su código y también conocido por su amplia gama de aplicaciones en la actualidad [15].

### 2.1.2. TypeScript

Es un lenguaje de programación pensado en aprovechar las mejores funcionalidades de JavaScrip solucionando inconvenientes que este presentaba, teniendo así un lenguaje más completo y mejorado que cada vez es más usado por los programadores [16]. Esta versión mejorada se ha popularizado más en los últimos tiempos.

### 2.1.3. HTML

HTML es el medio universal que entienden las computadoras y celulares para presentar su información en internet en diferentes plataformas, permitiendo crear vínculos que dan la funcionalidad al sistema requerido [17].

### 2.1.4. CSS

Prácticamente va de la mano de HTML, mientras éste le da la forma al contenido el CSS se encarga del estilo, como darle colores, formas, alineación, y otras características más [18].

### 2.1.5. Api REST

Es una API que permite la comunicación entre diferentes programas mediante solicitudes HTTP, también es usada cuando se necesita que la aplicación sea accesible desde internet [19].

### 2.1.6. Frameworks

En programación un framework es un conjunto de buenas prácticas estandarizadas que siguen ciertas reglas, además cuentan con funciones ya hechas y listas para ser usadas al crear proyectos, logrando así ahorrar tiempo y trabajo a los desarrolladores [20]. Se podría decir que son herramientas clave para el desarrollo de proyectos tecnológicos.

#### 2.1.6.2. Angular

Angular es un framework compuesto por el lenguaje de programación TypeScript [21]. El objetivo primordial de este framework es producir aplicaciones web y sus principales características son: interfaz fluida, velocidad, rendimiento y escalabilidad [21].

#### 2.1.6.3. Flask

Es un Microframework de Python pensado para desplegar aplicaciones web con todo lo primordial y necesario, perfecto para prototipos o proyectos pequeños. Además, a pesar de constar solo con funcionalidades básicas, posee la capacidad y compatibilidad para integrar diversas extensiones o librerías para escalar un proyecto [22].

### 2.1.11. Postman

Esta aplicación nos permite probar que nuestro Frontend y Backend se comuniquen como se espera, esto mediante solicitudes http, siendo así una de las mejores herramientas para el testing de APIs en la actualidad [23].

### 2.1.12. Visual Studio Code

Es un editor de código fuente basado en la simplicidad y eficiencia para mantener una experiencia de programación efectiva [24]. La edición y desarrollo de código, la identificación y corrección de errores, la gestión estructurada, son unas de las funciones más importantes que ofrece de esta aplicación [24].

### 2.1.13. Spyder (Anaconda Navigator)

Spyder es un entorno de desarrollo integrado (IDE) para Python. Cuenta con un editor de código, consola integrada, explorador de variables, así como también ya trae preinstaladas ciertas librerías. Además, nos brinda la capacidad de poder interactuar con entornos de trabajo.

## 2.2. Marco Teórico

### 2.2.1. Blockchain

Gracias a la fama de Bitcoin, la blockchain se dio a conocer como un mecanismo de seguridad y privacidad [25]. Es como un registro inmutable y transparente de información parecido a un libro diario de contabilidad [25]. Cada página de este libro es equivalente a un bloque en blockchain, y cada bloque enlazado a otros gracias a el hash del bloque más el hash del bloque anterior, constituyendo así la cadena de bloques.

#### 2.2.1.1 Tipos de Blockchain

##### 2.2.1.1.1 Blockchain pública

Es un tipo de blockchain donde no se requiere de ningún tipo de permiso o verificación para unirse a la cadena de bloques, esto se debe a su nivel de descentralización alto [26]. Este tipo de blockchain son las más populares actualmente y podemos verlo claro con ejemplos como, Bitcoin y Ethereum, dos monedas digitales referentes en el mundo cripto que usan esta tecnología.

##### 2.2.1.1.2 Blockchain privada

Es todo lo contrario a la cadena de bloques pública, debido a que no es de código abierto y para poder entrar a la blockchain los usuarios deben tener una invitación para participar de la red por parte de la empresa o entidad dueña, siendo esta la que brinda el acceso o no a los nuevos usuarios, brindando así una capa extra de seguridad en toda la red [26].

##### 2.2.1.1.3 Blockchain híbrida

Es una mezcla de los dos tipos anteriores, siendo así que la entidad sigue siendo la encargada de conceder el acceso a los usuarios para formar parte del sistema, pero que a su vez la información de la blockchain es accesible para cualquier persona, tal y como funcionan las cadenas de bloques públicas [26].

##### 2.2.1.1.4 Blockchain de consorcio

Este tipo de blockchain extrae características de otros modelos vistos anteriormente como son las públicas y privadas. Se trata de un esquema integrado por varias entidades u organizaciones, para ser parte de la red hay que ser miembro de una de las entidades, también cumple con el papel de la descentralización, pero la información solo es publica y accesible para los que forman parte de la blockchain más no para cualquier persona [27].

#### 2.2.1.2 Principales Blockchain en la Actualidad

##### 2.2.1.2.1 Bitcoin

Bitcoin es la cadena de bloques más conocida en el mundo principalmente debido a la popularización de su moneda digital que lleva el mismo nombre, cabe aclarar que esta blockchain es única y exclusivamente para la criptomoneda Bitcoin [28]. Esta cadena de bloques usa el mecanismo de consenso llamado Prood of Work (PoW).

#### 2.2.1.2.2 Ethereum

Como la anterior cadena, ésta también es pública, pero con la diferencia de que su uso no está restringido a una sola moneda sino más bien puede ser utilizada por cualquier criptomoneda que sea compatible con las tecnologías de Ethereum [29].

### 2.2.1.3 Aplicaciones de la Blockchain

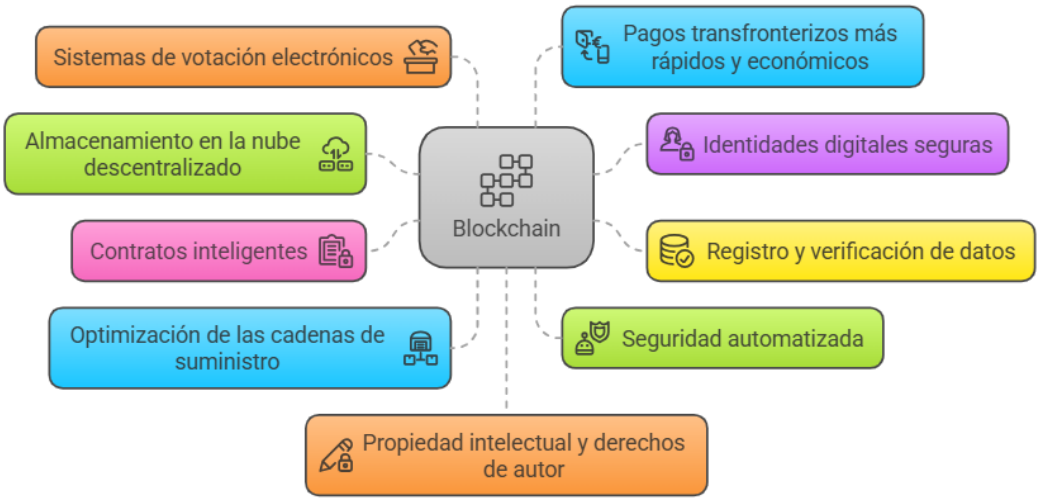


*Figura 1: Aplicaciones de la Blockchain.*

### 2.2.1.4 Casos de uso de la blockchain a nivel local

Esta tecnología a dado paso a la creación de nuevas aplicaciones, las cuales son implementadas a situaciones reales. Existen muchos casos en los que se puede aplicar la tecnología blockchain, y cada vez se descubren nuevos casos. En la siguiente figura podremos observar varios de estos.

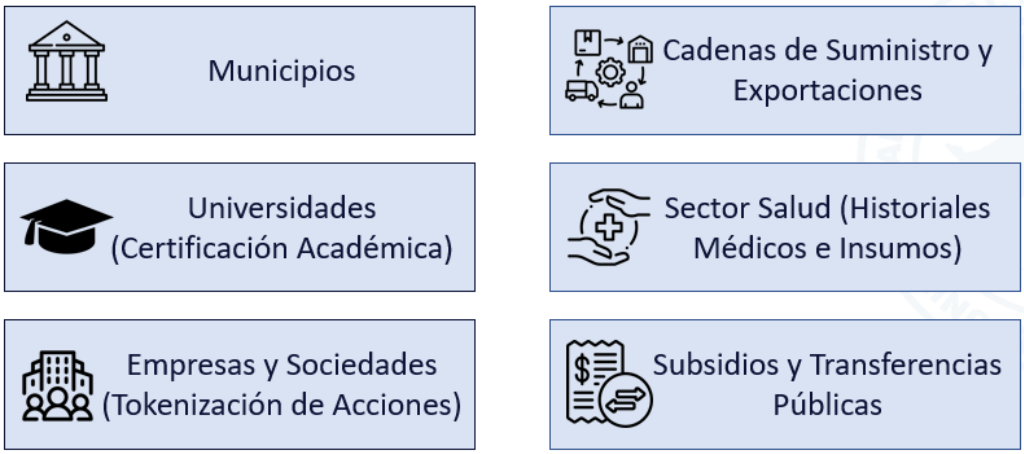


*Figura 2: Casos de uso de la blockchain*

#### 2.2.1.4 Ventajas y Desventajas

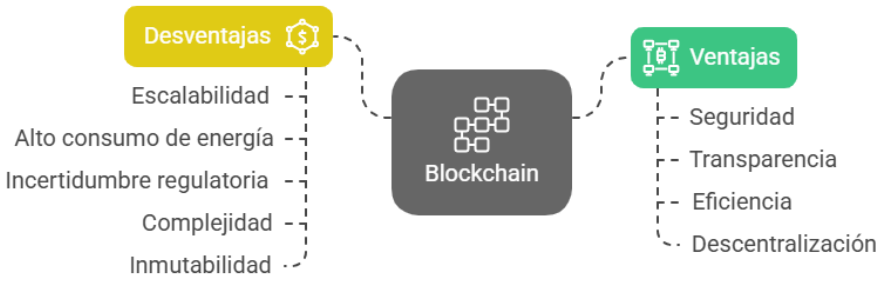


*Figura 3: Ventajas y Desventajas de la Blockchain.*

### 2.2.2. Criptomonedas

Son activos digitales que no dependen ni están controladas por ningún organismo o entidad, estas monedas están cifradas criptográficamente para asegurar su pertenencia hacia los usuarios y así prevenir riesgos de seguridad [6]. Es necesario que estas se almacenen en algún lugar, normalmente en una billetera digital y para transaccionar de forma segura se usa una blockchain [6].

Existen cuatro grupos principales de criptomonedas que se describen en la siguiente tabla:

| | **Definición** | **Ejemplos** |
|---|---|---|
| **Criptomonedas de valor** | Activos digitales que sirven para almacenar, enviar y recibir dinero. | Bitcoin (BTC), Litecoin (LTC), Bitcoin Cash (BCH) |
| **Criptomonedas de plataforma** | Criptomonedas que permiten crear contratos inteligentes y aplicaciones descentralizadas. | Ethereum (ETH), Cardano (ADA), Solana (SOL) |
| **Tokens estables (Stablecoins)** | Tokens cuyo valor está vinculado a monedas fiat u otros activos para mantener estabilidad. | USDT, USDC, BUSD |
| **Tokens y activos** | Monedas con valor solo dentro de un ecosistema. | BAT, UNI, AAVE, CryptoPunks, DOGE |

*Tabla 1: Tipos de Criptomonedas.*

### 2.2.2.1 Diferencia entre token y criptomoneda

La principal diferencia es que una criptomoneda tiene su propia blockchain en cambio un token usa una ya existente. A continuación, en la siguiente figura se muestra una comparación entre ambas:

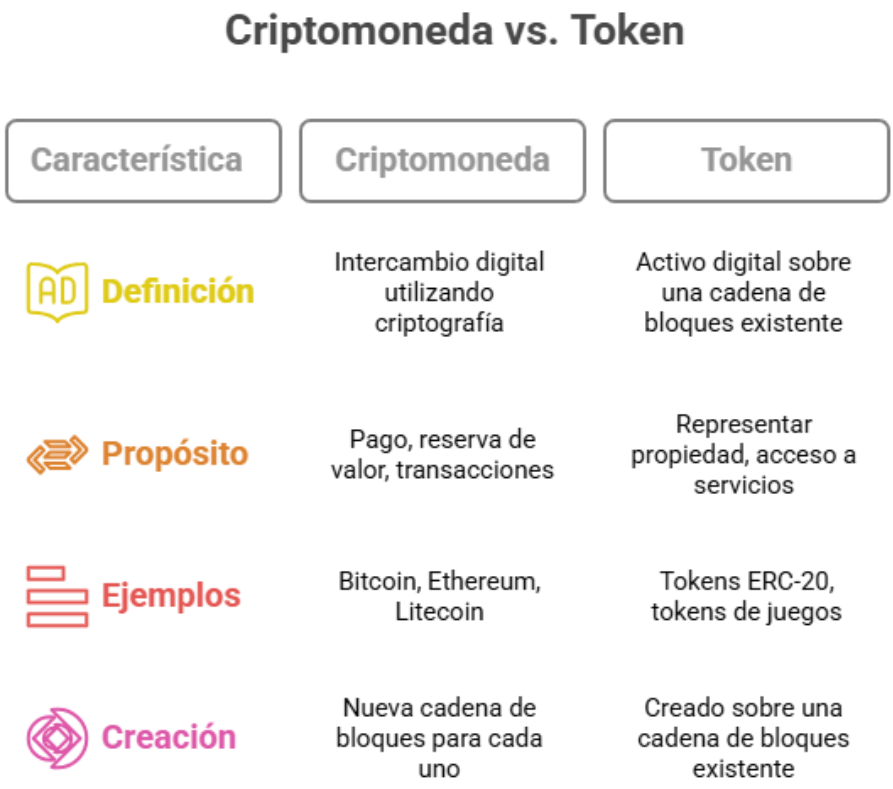


*Figura 4: Diferencia entre Criptomoneda y Token.*

### 2.2.2.2 Importancia de proyectos que respaldan a una criptomoneda

Es necesario comprender la importancia del proyecto que esta por detrás de una criptomoneda debido a que estos aportan confiabilidad, utilidad y sostenibilidad a la moneda, variables importantes para su acogimiento y triunfo en el mercado cripto mundial [30]. Hay personas de alto valor, ya sean por reconocimiento o por poder monetario, estas personas en la mayoría de los casos también respaldan proyectos cripto, y en algunas veces aportan liquidez a la ronda de financiación.

### 2.2.2.3 Monetización de criptomonedas o proyectos cripto

Para los dueños y comunidad de proyectos cripto un tema de relevancia es la forma de monetizar, lo cual es uno de los objetivos por los cuales se trabaja e invierte tiempo y recursos en dicho proyecto cripto, a continuación, en la figura 4 se enumeran ciertas formas de monetización:


*Figura 5: Métodos de Monetización*

## 2.3. Metodología del Proyecto

### 2.3.1. Metodología de Investigación

El método cualitativo tiene su foco en buscar y comprender aspectos sociales o humanos, este enfoque se concentra en el análisis de datos, los mismos que pueden ser recopilados desde entrevistas, observaciones hasta el análisis de documentos [31]. En el presente proyecto se utilizará la revisión documental y la observación participante para analizar la información existente sobre blockchain y criptomonedas, reconociendo ventajas, desafíos y casos de uso.

La metodología de investigación descriptiva la usaremos para saber por qué muchas criptomonedas o tokens fracasan así estén respaldadas con proyectos, además hablaremos sobre monedas estables, así como también como afectan las estafas cripto de tipo rug pulls a la confianza del usuario.

Este proyecto es del tipo de metodología aplicada ya que se enfoca en desarrollar e implementar una tecnología funcional y al final se tendrá un producto terminado.

### 2.3.2. Técnicas e instrumentos de recolección de datos

Se usó es la revisión documental digital, extrayendo información de artículos científicos, libros, revistas científicas y páginas web para documentar los conceptos y teorías necesarias sobre blockchain, criptomonedas y billeteras digitales profundizando así los conceptos básicos para cada apartado.

La observación participante se centra en participar activamente desde dentro de un sistema o aplicación para obtener más de cerca información clave que sirve como guía para el desarrollo de un proyecto [32]. En este caso se usó la observación participante para hacer pruebas dentro de Exchanges, billeteras de criptomonedas y sitios web de seguimiento de precios de criptomonedas. Explorando así las principales funciones y características de estas aplicaciones.

### 2.3.3. Metodología de desarrollo

Para el desarrollo implementación de una bockchain y una criptomoneda se utilizará una metodología por fases similar como se plantea en el trabajo de B. Ocaña Valdez en la Universidad Politécnica Salesiana [33], pero con ciertas variaciones ya que no se usará la

metodología XP (Extreme Programming) [34]. A continuación, se presentan las fases del proyecto con una breve descripción.

**Fase de análisis y planificación:** Para comenzar se analizan los requisitos del proyecto como, la definición de funcionalidades básicas como las transacciones, los roles y elementos del sistema, así como también las tecnologías y herramientas a utilizar.

**Fase de diseño:** En esta fase de define el esquema de la arquitectura del sistema, de la API REST, de la criptomoneda y se diseña la lógica del minado de bloques

**Fase de desarrollo:** Aquí se llevará a cabo la creación de la blockchain con su lógica de bloques, minado y validaciones. La API integrará los endpoints esenciales y por último se desarrollará la aplicación web que permitirá a los usuarios hacer transacciones y ver el historial.

**Fase de pruebas y validación:** Para comprobar que todo el sistema esta funcional se llevarán a cabo una serie de pruebas de funcionalidad y de seguridad como la prevención de transacciones duplicadas para garantizar la integridad y confiabilidad del sistema.

# CAPÍTULO 3. PROPUESTA

## 3.1. Requerimientos

### 3.1.1. Requerimientos Funcionales

| Identificador | Detalle | Categoría |
|---|---|---|
| **RF - 01** | La aplicación permitirá a los nuevos usuarios registrarse mediante su nombre de usuario, correo y contraseña. | Registro de usuarios |
| **RF - 02** | El usuario podrá recuperar su contraseña mediante su dirección de correo electrónico. | Recuperación de cuenta |
| **RF - 03** | La aplicación permitirá a los usuarios iniciar sesión con su nombre de usuario y contraseña. | Inicio de sesión |
| **RF - 04** | El botón cerrar sesión borrará el token JWT del localStorage. | Cierre de sesión |

| | | |
|---|---|---|
| **RF - 05** | La página principal permitirá a los usuarios intercambiar sus dólares en USDT a UPSX. | Intercambio de monedas |
| **RF - 06** | Los usuarios podrán generar el código QR de su dirección de billetera. | Generar código QR |
| **RF - 07** | Los usuarios podrán escanear los códigos QR de otros usuarios para agilizar la realización de transacciones. | Escanear código QR |
| **RF 08** | El botón validar confirmará la existencia de la dirección. | Validar direcciones de billeteras |
| **RF - 09** | La criptomoneda simulada será volátil, es decir, su valor varía dentro de un rango, no es fija. | Volatilidad de moneda |
| **RF - 10** | Los usuarios podrán enviar criptomonedas simuladas a otros usuarios, esto sabiendo la dirección de billetera del usuario receptor mediante el escáner QR o de forma manual. | Enviar transacción |
| **RF - 11** | El saldo recibido se almacenará en la billetera o cuenta del usuario que recibe la transacción. | Recibir transacción |
| **RF - 12** | Los usuarios podrán ver el historial de transacciones recibidas, enviadas e intercambios. | Ver historial de transacciones |
| **RF - 13** | Los nodos validadores y nodos mineros deben poder almacenar temporalmente las transacciones válidas en la mempool antes de ser confirmadas en un bloque. | Gestión de mempool |
| **RF - 14** | El nodo validador que recibe la transacción debe propagar las transacciones a los demás nodos de la red de forma eficiente. | Propagación de transacciones |

| | | |
|---|---|---|
| **RF - 15** | El nodo minero no debe iniciar el minado de un bloque si no hay transacciones disponibles en la mempool. | Validación previa de mempool |
| **RF - 16** | El nodo minero debe ser capaz de minar bloques resolviendo el algoritmo de consenso y agregando transacciones válidas. | Minería de bloques |
| **RF - 17** | El nodo minero propagará la cadena de bloques válida hacia los demás nodos de la red. | Propagación de cadena de bloques |
| **RF - 18** | La validación de transacciones se hará mediante el PoS. | Consenso entre nodos |
| **RF - 19** | El sistema debe deshabilitar los botones tras ser presionados para evitar múltiples clics y solicitudes duplicadas. | Desactivación de botón tras clic |
| **RF - 20** | El sistema debe validar los campos de entrada en los formularios para asegurar que los usuarios ingresen datos correctos, completos y en el formato requerido antes de permitir el envío de la información. | Validaciones |

*Tabla 2: Requerimientos funcionales.*

### 3.1.2. Requerimientos no Funcionales

| Identificador | Detalle | Categoría |
|---|---|---|
| **RF - 21** | La aplicación validará el inicio de sesión mediante la base de datos. | Seguridad |
| **RF – 22** | La comunicación entre frontend, backend y nodos usará protocolo HTTP seguro. | Seguridad |

| | | |
|---|---|---|
| **RF – 23** | Las transacciones se guardan en la blockchain. | Almacenamiento |
| **RF – 24** | La aplicación guardará la lógica del inicio de sesión en una base de datos. | Almacenamiento |
| **RF – 25** | El sistema deberá responder a las solicitudes del usuario en menos de 2 s. | Rendimiento |
| **RF - 26** | La aplicación debe ser compatible con navegadores modernos. | Compatibilidad |

*Tabla 3: Requerimientos no funcionales.*

## 3.2. Componente de la Propuesta

### 3.2.1. Arquitectura del Sistema

La arquitectura de este sistema plantea varios conectores: El usuario, mediante la aplicación web, se registra e inicia sesión. Una vez dentro, para que la app funcione correctamente, se comunica con el backend mediante rutas preestablecidas que ejecutan funciones específicas; en este caso serían las transacciones que se envían hacia la base de datos y a la vez hacia los nodos para posteriormente pasarlas desde los nodos a la blockchain simulada con terminales de consola. La implementación de la base de datos se debe a fines de respaldo y velocidad en consultas.

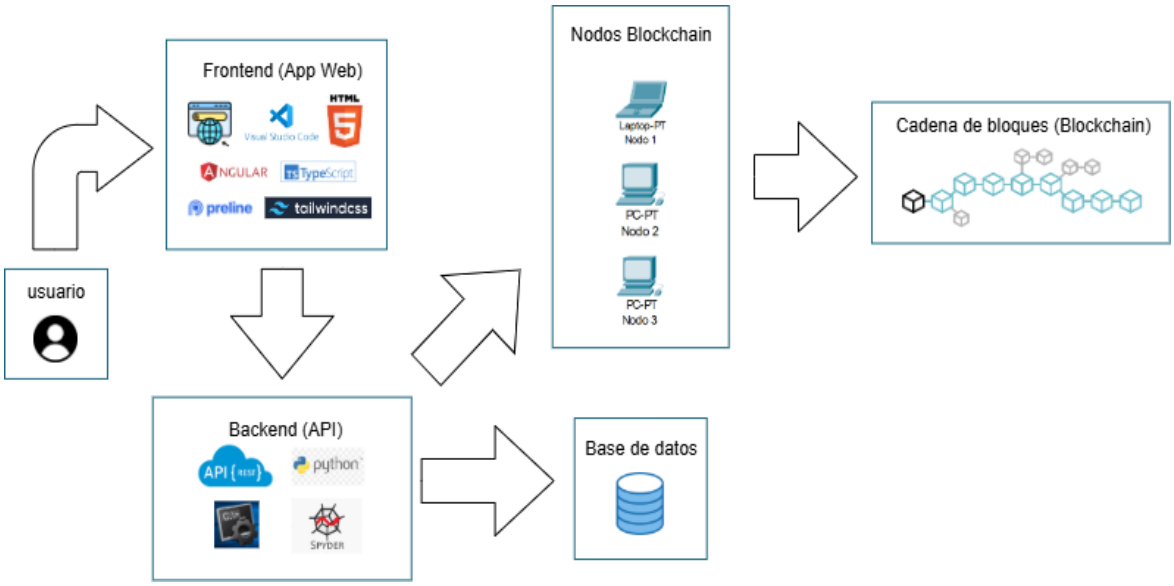


*Figura 6: Arquitectura del sistema.*

### 3.2.2. Diagramas de casos de uso

| Caso de Uso | Registro de usuarios |
|---|---|
| **Actor(es)** | Usuario, sistema. |
| **Descripción** | Se pueden registrar con un nombre de usuario, un correo y una contraseña. |
| **Flujo inicial** | El usuario accede a la opción de registro desde la página principal. |
| **Pasos** | • El usuario accede al formulario de registro.<br>• Ingresa nombre de usuario, correo electrónico y contraseña.<br>• El sistema valida los datos.<br>• Se crea la cuenta del usuario.<br>• El sistema notifica que el registro fue exitoso. |
| **Requisitos previos** | El usuario no debe tener una cuenta previamente registrada con el mismo correo. |
| **Resultado esperado** | El usuario puede iniciar sesión correctamente. |

*Tabla 4: Caso de uso – Registro de usuarios.*

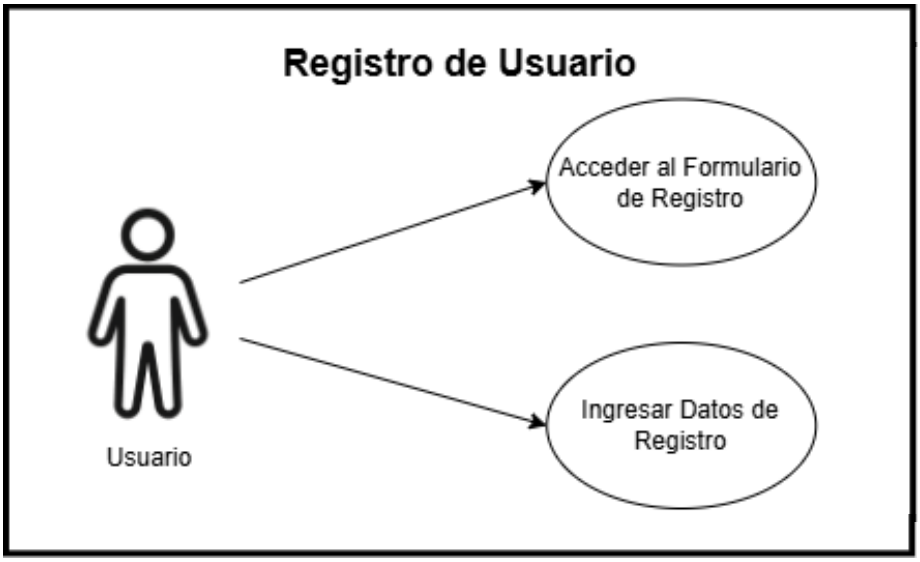


*Figura 7: Caso de uso – Registro de usuarios_1.*

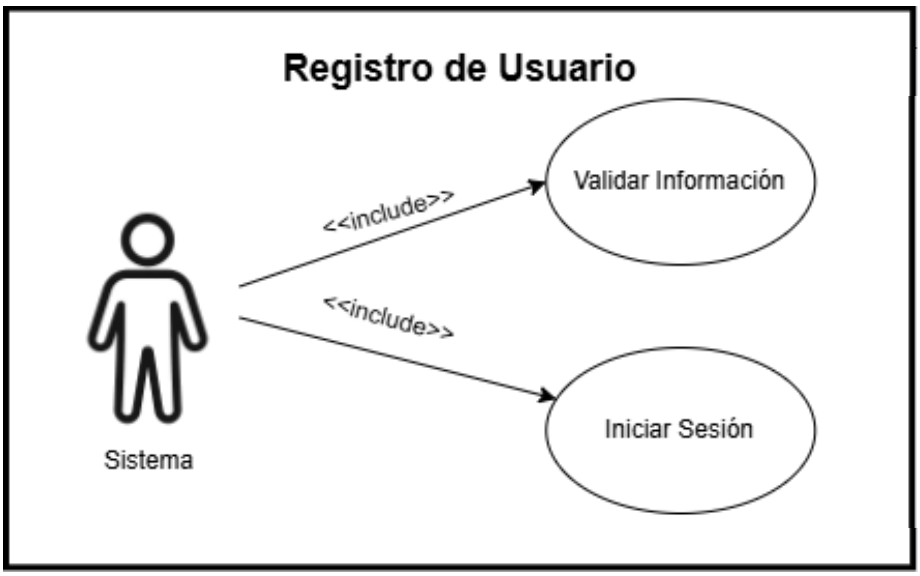


*Figura 8: Caso de uso – Registro de usuarios_2.*

| Caso de Uso | Recuperar contraseña |
|---|---|
| **Actor(es)** | Usuario, sistema. |
| **Descripción** | Los usuarios podrán acceder su cuenta de nuevo gracias a un código de recuperación enviado al correo. |
| **Flujo inicial** | El usuario selecciona la opción "¿Olvidaste tu contraseña?". |
| **Pasos** | • Ingresa su correo electrónico.<br>• El sistema valida el correo.<br>• Se envía un código al correo.<br>• El usuario ingresa el código.<br>• El sistema valida el código.<br>• El usuario ingresa la nueva contraseña.<br>• El sistema actualiza la contraseña. |
| **Requisitos previos** | El correo debe estar registrado en el sistema. |

| | |
|---|---|
| **Resultado esperado** | La contraseña se actualiza y puede iniciar sesión. |

*Tabla 5: Caso de uso - Recuperar contraseña.*

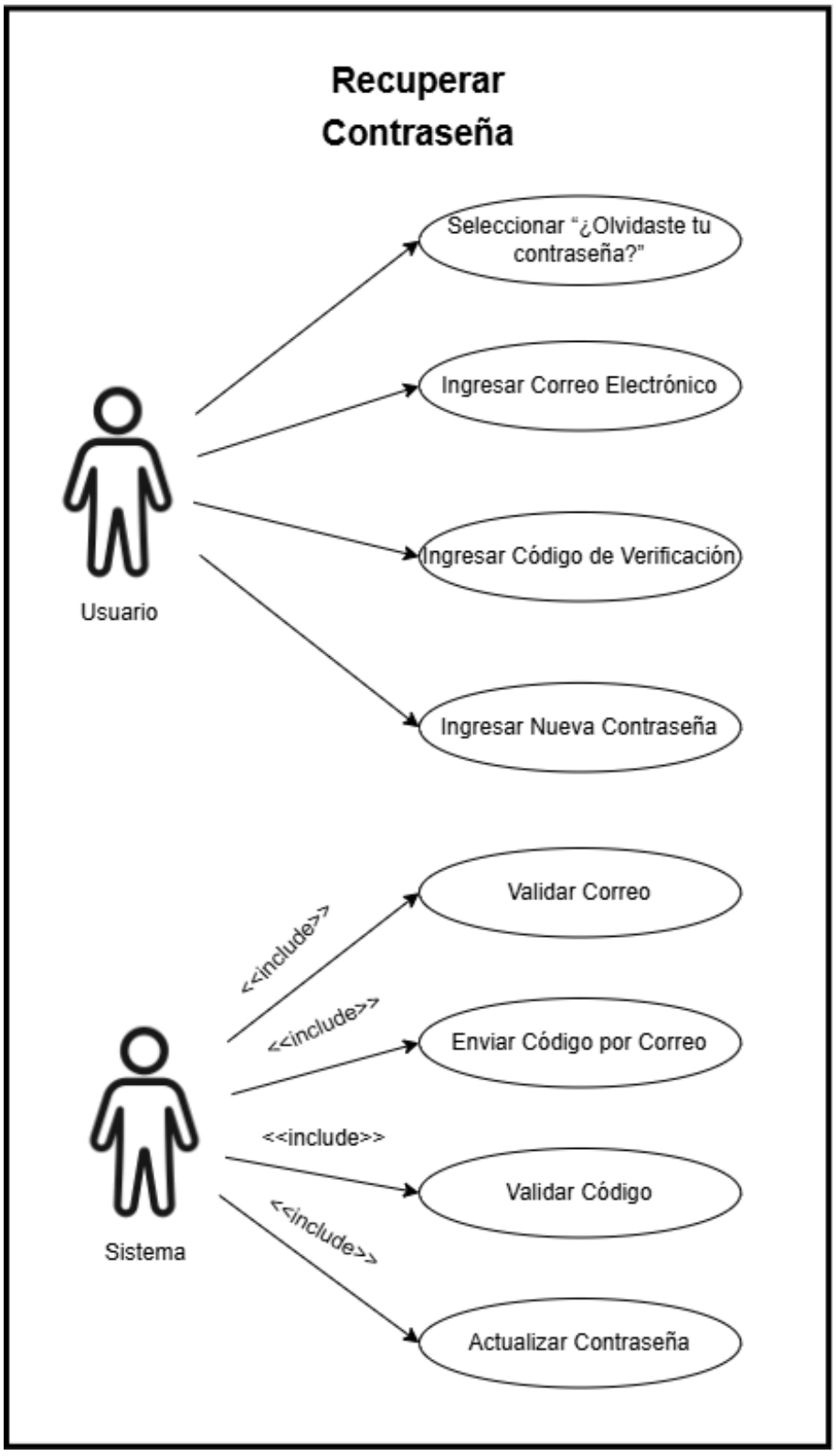


*Figura 9: Caso de uso - Recuperar contraseña.*

| Caso de Uso | Acceso al sistema |
|---|---|
| **Actor(es)** | Usuario, sistema |
| **Descripción** | Permite a los usuarios ingresar al sistema mediante credenciales (usuario y contraseña). |
| **Flujo inicial** | El usuario accede a la página de login e ingresa sus credenciales. |
| **Pasos** | • El usuario accede al formulario de login.<br>• Ingresa usuario y contraseña.<br>• El sistema valida las credenciales. |
| **Requisitos previos** | El usuario debe estar registrado y tener credenciales válidas. |
| **Resultado esperado** | El usuario accede a la página principal de la aplicación. |

*Tabla 6: Caso de uso - Acceso al sistema.*

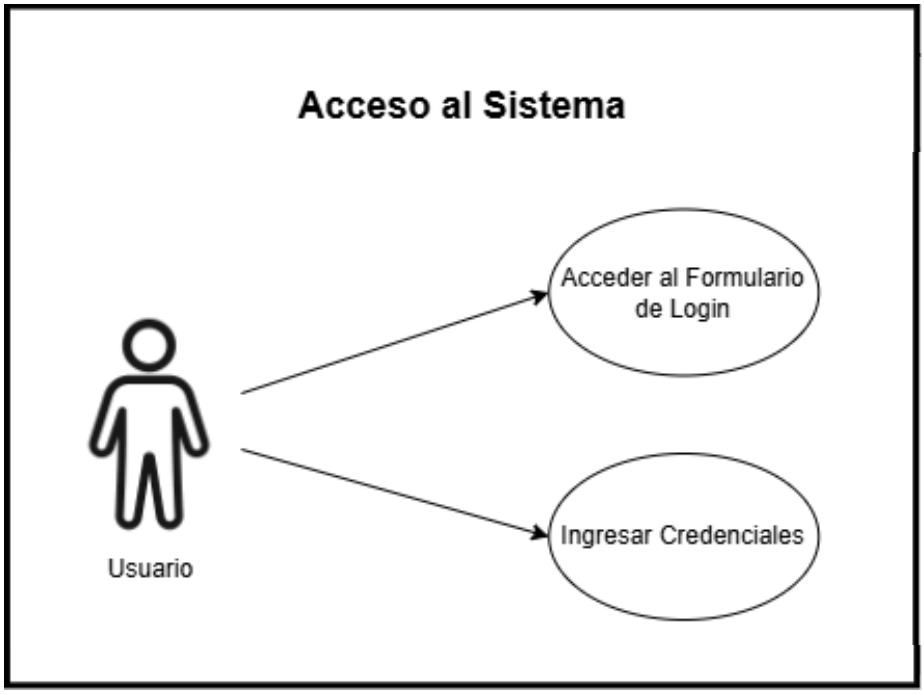


*Figura 10: Caso de uso - Acceso al sistema_1.*

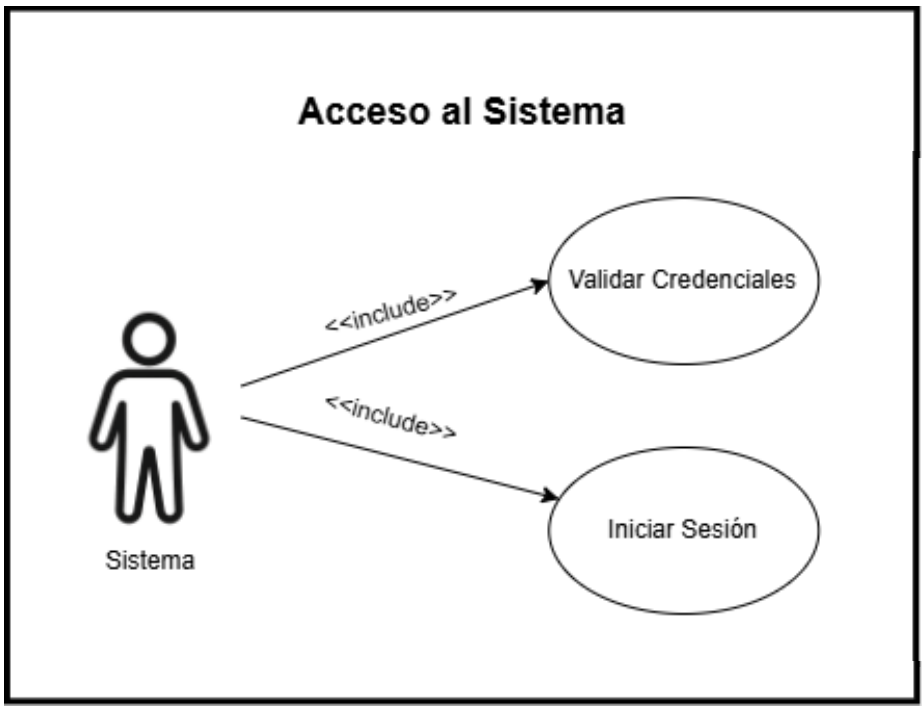


*Figura 11: Caso de uso - Acceso al sistema_2.*

| Caso de Uso | Intercambiar USDT por UPSX |
|---|---|
| **Actor(es)** | Usuario, sistema |
| **Descripción** | El usuario puede intercambiar la cantidad deseada de USDT por la criptomoneda simulada UPSX.<br>Esta criptomoneda tendrá un valor simulado de alrededor de 3 dólares, su valor sara volátil. |
| **Flujo inicial** | El usuario accede a la opción de intercambio y define la cantidad.<br>El sistema muestra el valor equivalente en UPSX. |
| **Pasos** | • El usuario ingresa monto en USDT.<br>• El sistema calcula el equivalente en UPSX.<br>• Confirma el intercambio.<br>• El sistema actualiza los saldos. |
| **Requisitos previos** | El usuario debe tener saldo USDT disponible. |

| | |
|---|---|
| **Resultado esperado** | El saldo en UPSX aumenta y el saldo en USDT disminuye. |

*Tabla 7: Caso de uso - Intercambiar USDT por UPSX.*

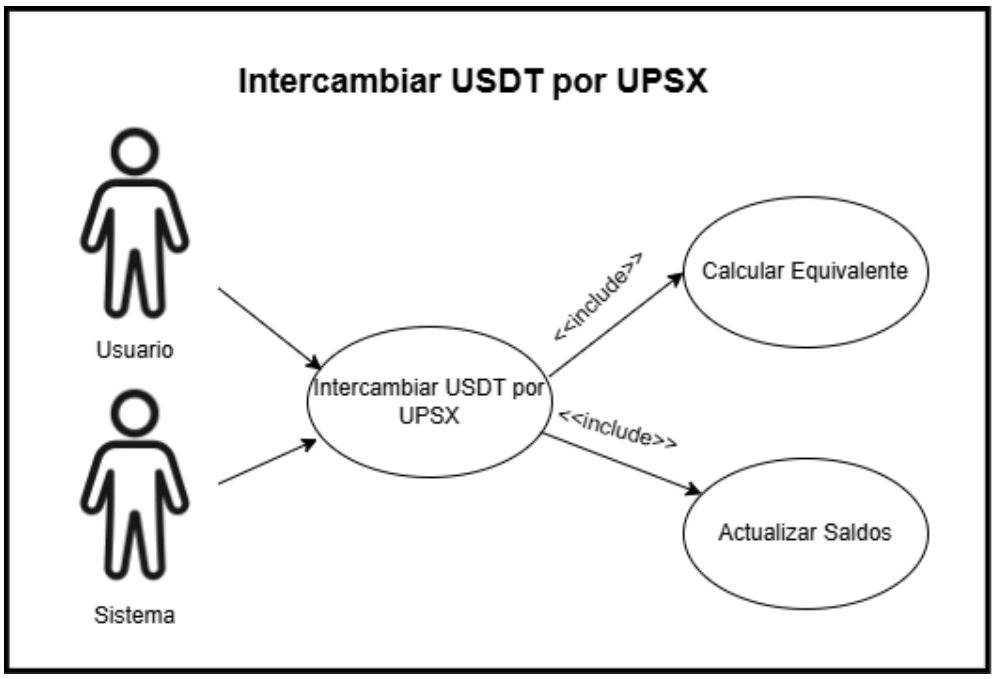


*Figura 12: Caso de uso - Intercambiar USDT por UPSX.*

| Caso de Uso | Compartir dirección de billetera |
|---|---|
| **Actor(es)** | Usuario |
| **Descripción** | Los usuarios podrán compartir su dirección con otros para recibir criptomonedas UPSX.<br>Los usuarios podrán escanear la dirección de otros para enviar criptomonedas UPSX. |
| **Flujo inicial** | El usuario accede a las opciones de generación y escaneo de códigos QR desde la pantalla de envíos, además lo muestra al usuario que le va a enviar criptomonedas. |
| **Pasos** | • El usuario selecciona "Mostrar mi QR". |

| | |
|---|---|
| | • El sistema obtiene la dirección del usuario y genera un código QR.<br>• El código QR se muestra en pantalla.<br>• El usuario selecciona "Escanear QR".<br>• El sistema activa la cámara y escanea el QR de otro usuario.<br>• Se extrae la dirección desde el código escaneado.<br>• La dirección escaneada queda lista para realizar una transacción. |
| **Requisitos previos** | Ambos usuarios deben estar autenticados en la aplicación. |
| **Resultado esperado** | La dirección del usuario se muestra como un código QR para ser escaneada, y también se puede capturar la dirección de otro usuario escaneando su QR, dejándola lista para el envío o recepción de fondos. |

*Tabla 8: Caso de uso - Compartir dirección de billetera.*

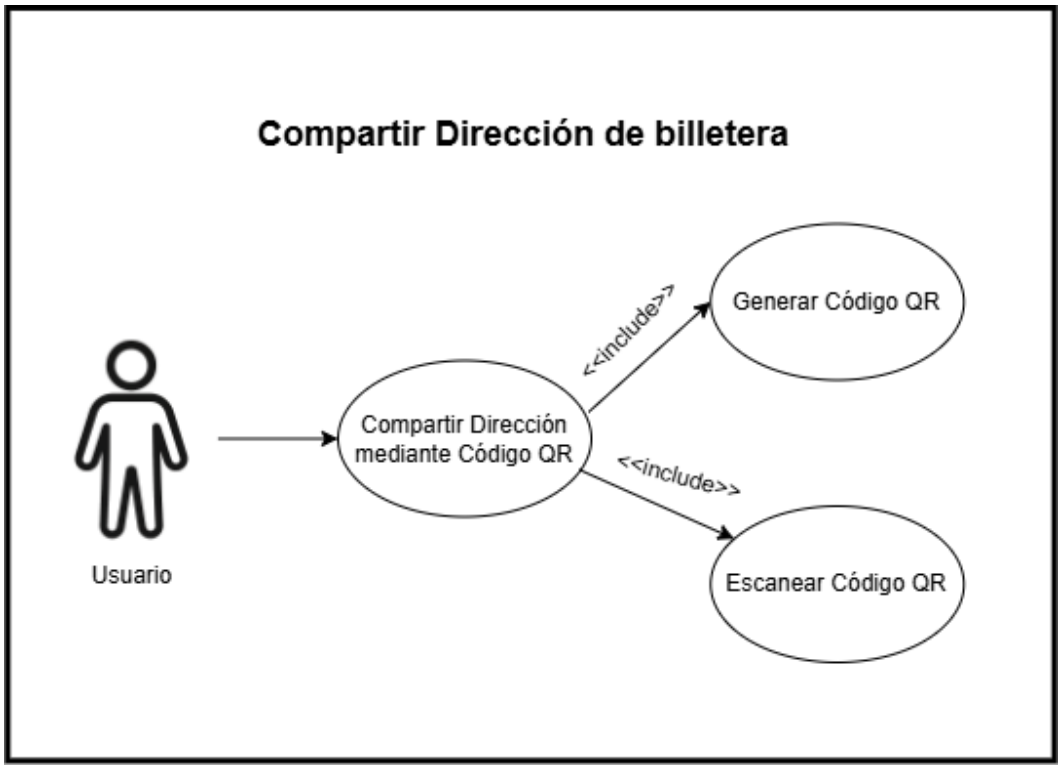


*Figura 13: Caso de uso - Compartir dirección de billetera.*

| Caso de Uso | Enviar criptomoneda |
|---|---|
| **Actor(es)** | Usuario, sistema, backend, nodo validador, nodo minero. |
| **Descripción** | Permite al usuario enviar la criptomoneda UPSX a otra dirección, ingresando manualmente la dirección o escaneando un código QR. El sistema procesa la transacción, la guarda en la base de datos mediante un backend y la propaga a la red blockchain. |
| **Flujo inicial** | El usuario accede a la opción de envío<br>El usuario ingresa la información y presiona "Enviar". |
| **Pasos** | • El usuario ingresa o escanea la dirección destino.<br>• El usuario especifica la cantidad a enviar.<br>• El sistema valida los datos ingresados.<br>• El sistema envía la información de la transacción al backend.<br>• El backend guarda la transacción en la base de datos.<br>• El backend propaga la transacción al nodo validador.<br>• El nodo validador coloca la transacción en su mempool.<br>• La transacción se propaga a los demás nodos, incluyendo el nodo minero.<br>• El nodo minero incluye la transacción en un bloque y lo mina.<br>• El sistema informa al usuario que la transacción fue procesada.<br>• El nodo minero propaga la transacción. |

| | |
|---|---|
| **Requisitos previos** | El usuario debe estar autenticado.<br>El usuario debe tener saldo suficiente. |
| **Resultado esperado** | La transacción queda registrada en la base de datos.<br>La transacción es propagada y minada por la red.<br>El saldo del usuario se actualiza. |

*Tabla 9: Caso de uso - Enviar criptomoneda.*

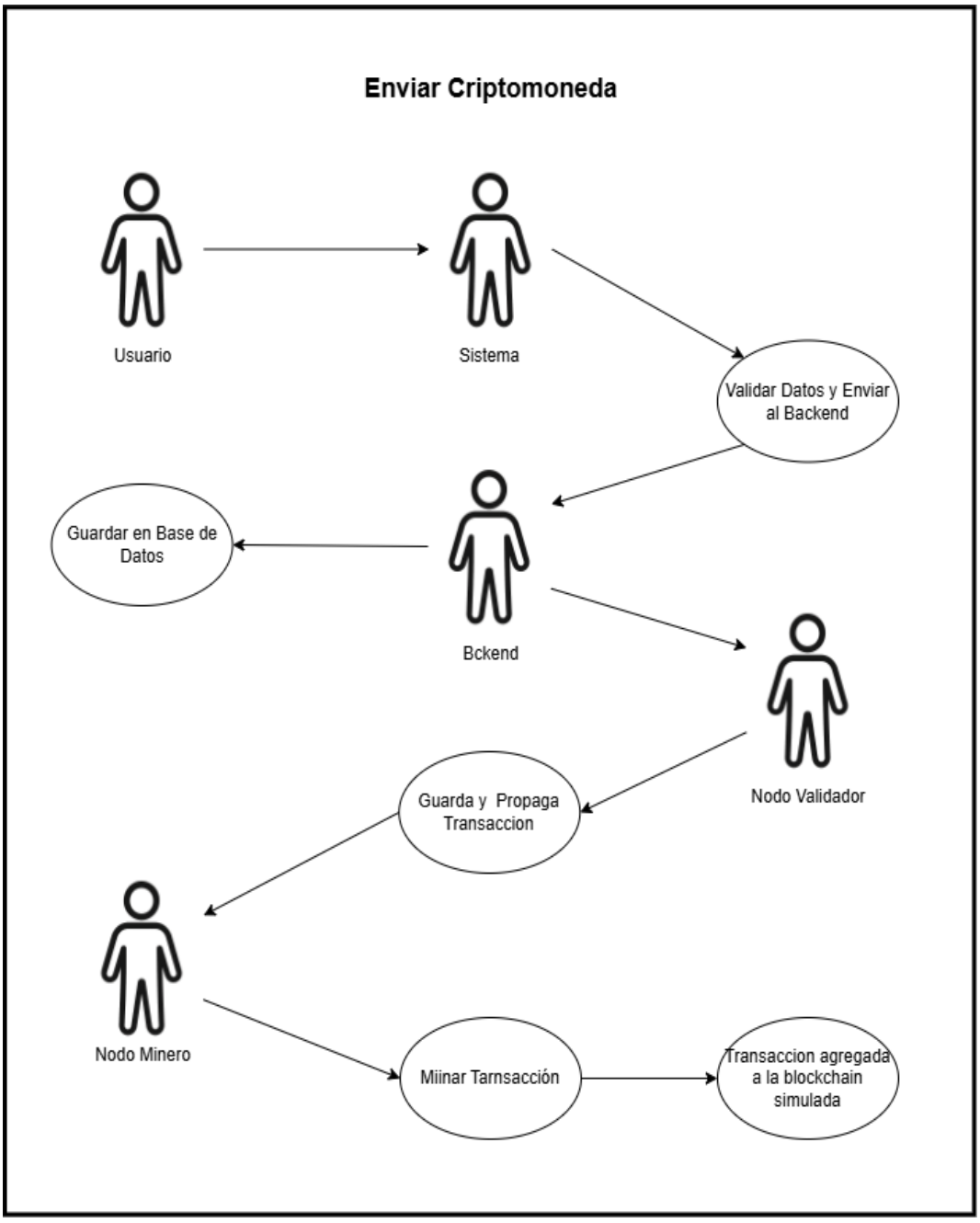


*Figura 14: Caso de uso - Enviar criptomoneda.*

| Caso de Uso | Visualizar historial |
|---|---|
| **Actor(es)** | Usuario |
| **Descripción** | Los usuarios pueden ver sus movimientos a lo largo del tiempo como intercambios, criptomonedas enviadas y así mismo recibidas. |
| **Flujo inicial** | El usuario accede a la sección de historial desde el menú principal. |
| **Pasos** | • El usuario accede a la opción "Historial".<br>• El sistema consulta los movimientos del usuario en la base de datos. |
| **Requisitos previos** | El usuario debe haber iniciado sesión. |
| **Resultado esperado** | Se muestra el historial completo de movimientos del usuario. |

*Tabla 10: Caso de uso - Visualizar historial de movimientos.*

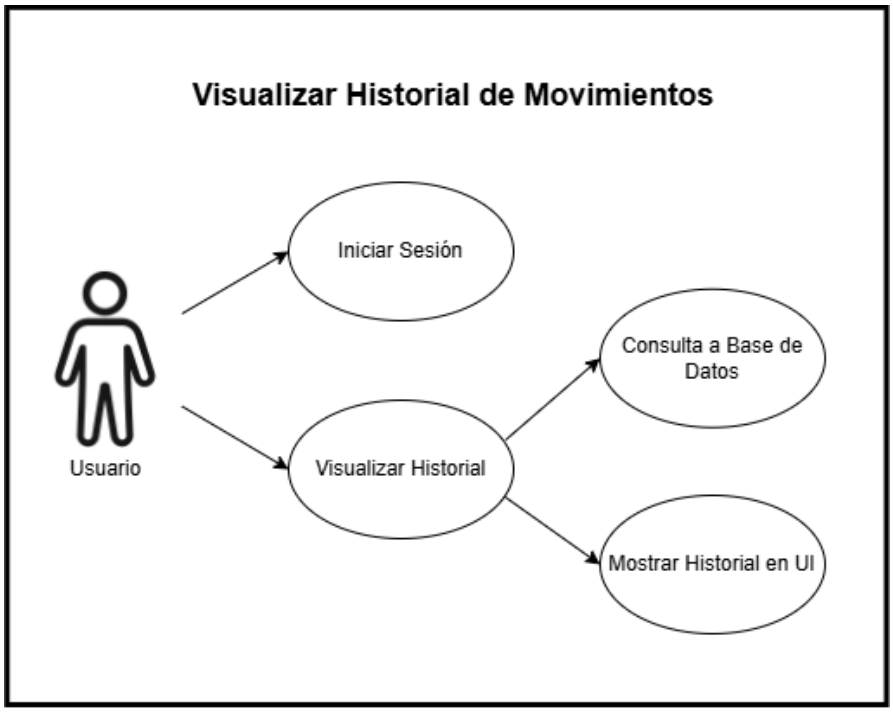


*Figura 15: Caso de uso - Visualizar historial de movimientos.*

### 3.2.3. Modelado de Datos

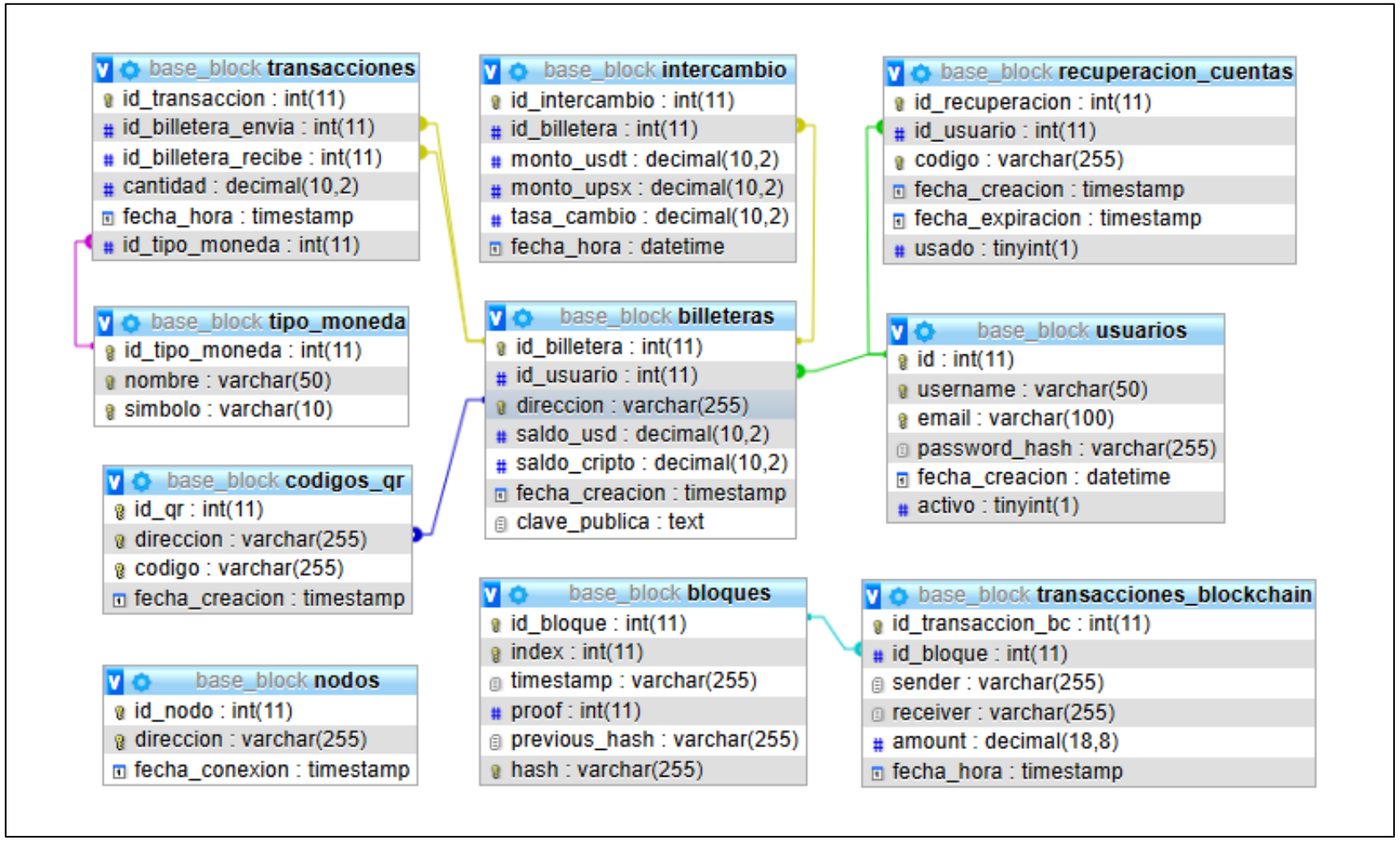


*Figura 16: Modelado de datos.*

### 3.3. Diseño de Interfaces

#### Interfaz de bienvenida

En esta página el usuario podrá elegir dos opciones: una de iniciar sesión en caso de ya tener una cuenta creada o, de lo contrario, crear una nueva cuenta.

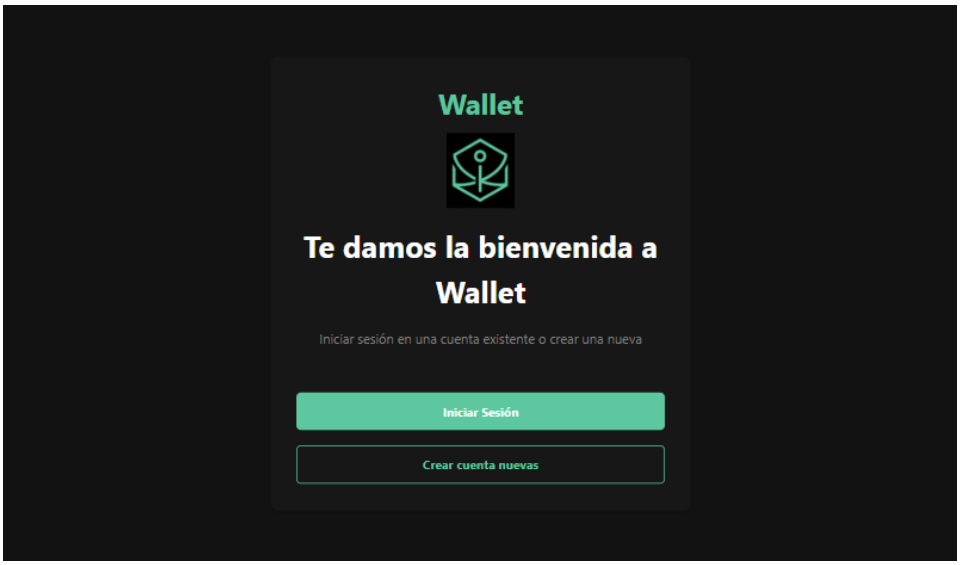


*Figura 17: Interfaz de bienvenida.*

#### Interfaz de registro de nuevos usuarios

En esta página los usuarios podrán registrarse con 3 campos: nombre de usuario, correo electrónico válido y una contraseña.

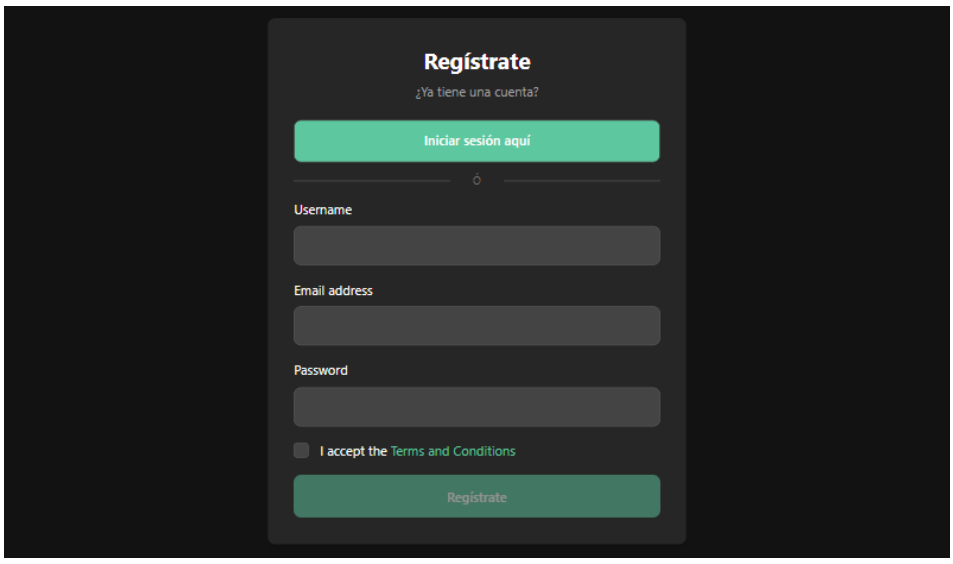


*Figura 18: Interfaz de registro.*

### Interfaz de ingreso a la aplicación

En esta interfaz los usuarios ingresan su username y su password para que el sistema valide el ingreso a la aplicación web.

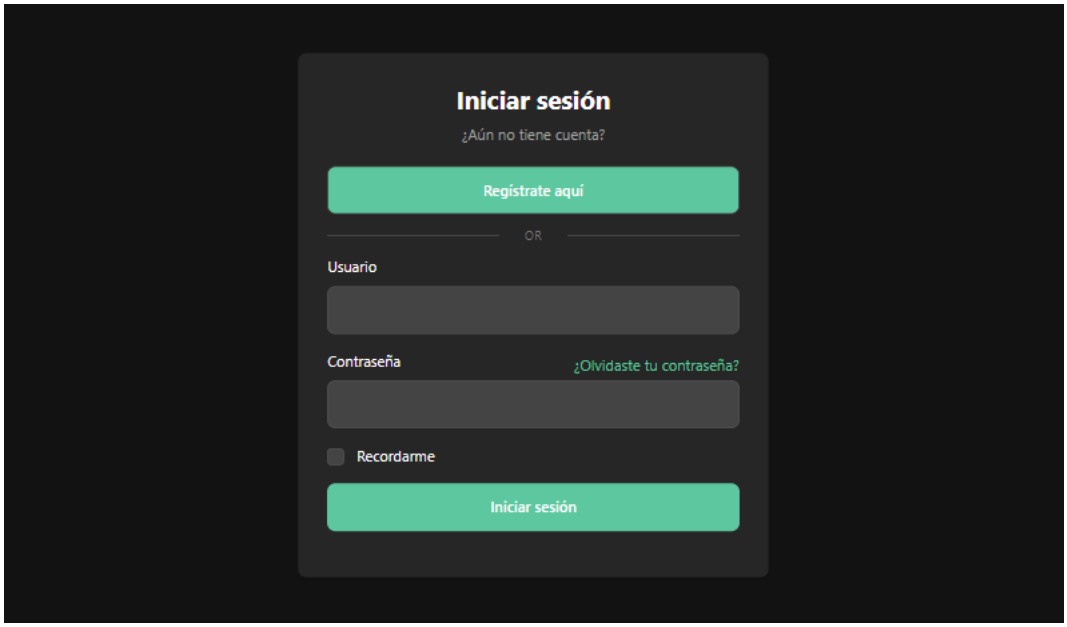


*Figura 19: Interfaz de inicio de sesión.*

### Interfaz de recuperación de contraseña

En caso de pérdida de contraseña, el usuario deberá ingresar la dirección de correo validada en la base de datos para poder recuperar su cuenta.

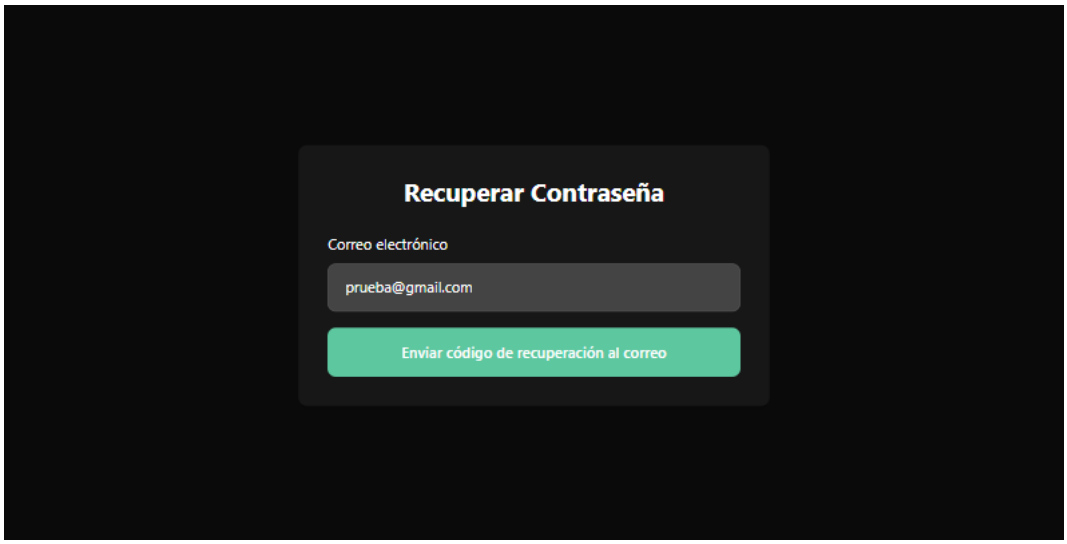


*Figura 20: Interfaz de recuperación de contraseña.*

Se envía un código al correo del usuario y el usuario procede a colocarlo en la pantalla correspondiente.


*Figura 21: Validación de código.*

Una vez que el código es validado correctamente por el sistema, el usuario podrá ingresar una nueva contraseña.

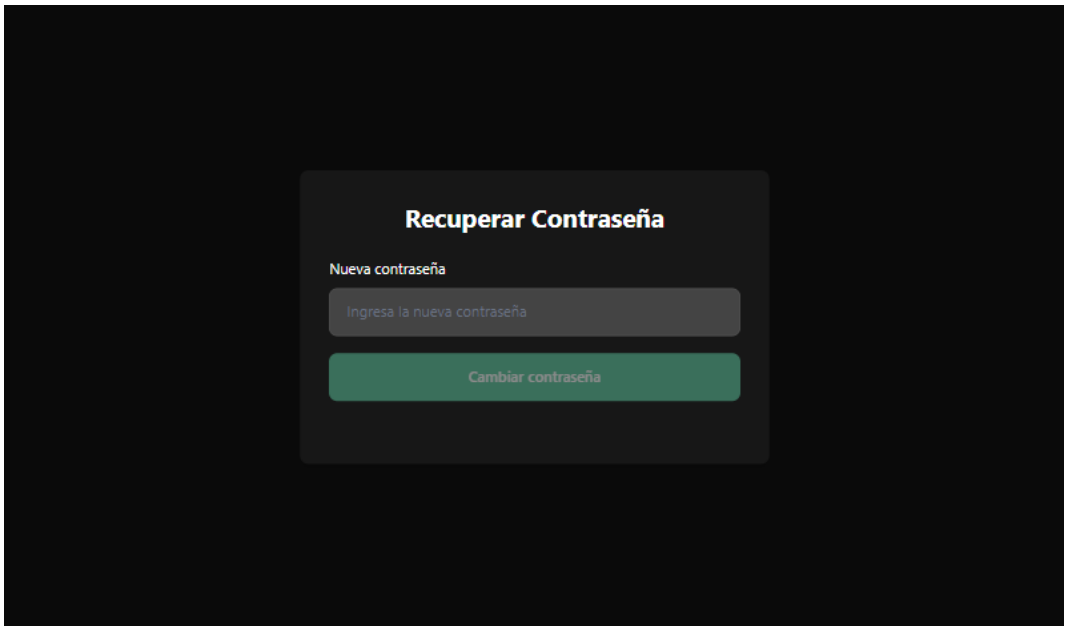


*Figura 22: Ingreso de nueva contraseña.*

### Interfaz principal

Esta es la página principal de la aplicación web.

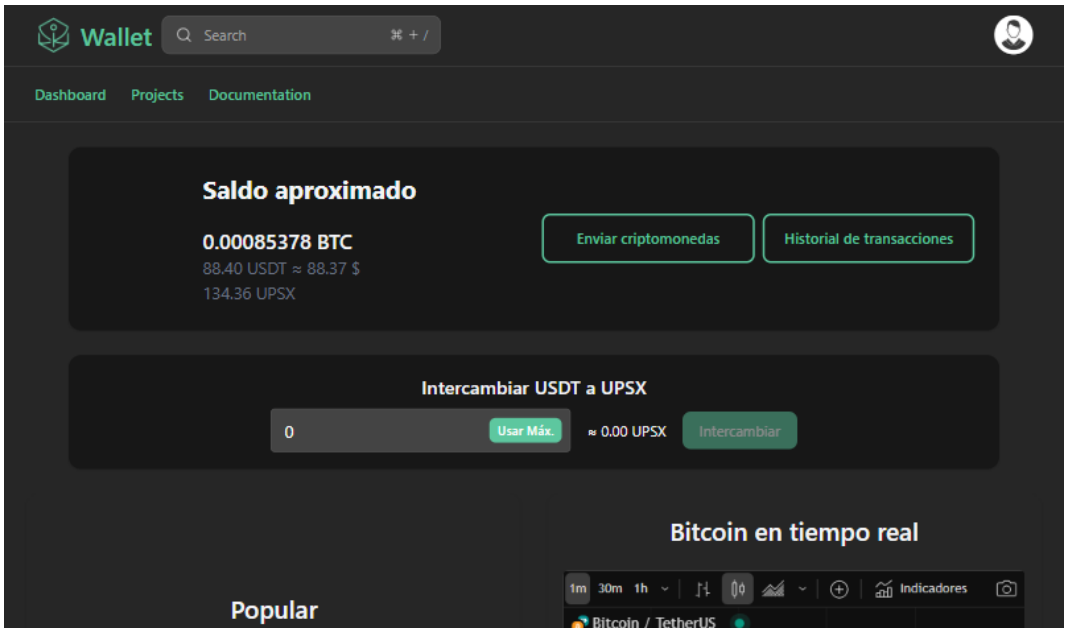


*Figura 23: Página principal.*

En este apartado se muestra el saldo del usuario en Bitcoin, USDT y en UPSX.

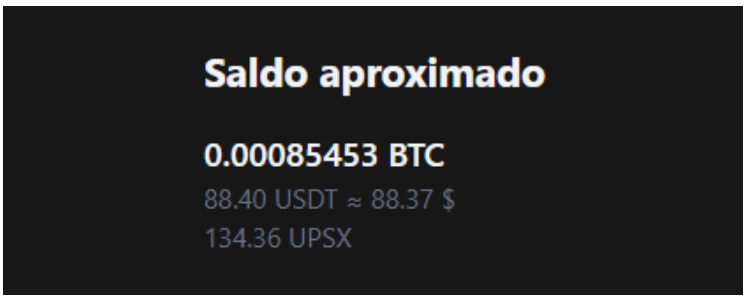


*Figura 24: Saldos del usuario.*

Aquí se presentan los botones para realizar transacciones y ver el historial de movimientos del usuario.

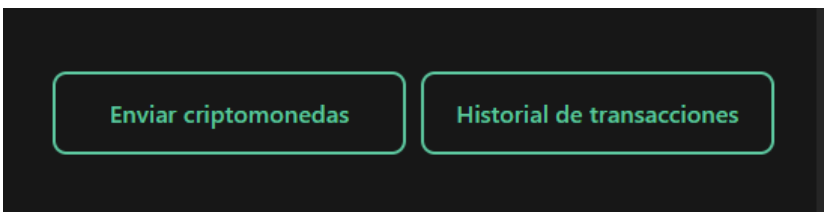


*Figura 25: Enviar e historial de transacciones.*

### Interfaz de intercambio de criptomonedas

El usuario podrá intercambiar sus USDT por la criptomoneda UPSX.

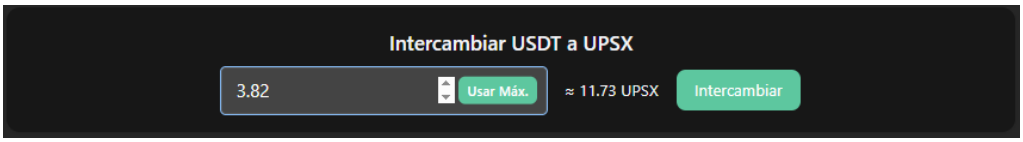


*Figura 26: Intercambio de criptomonedas.*

### Interfaz de información

Los usuarios podrán ver en la parte izquierda el top 5 de criptomonedas populares actualmente en el mercado.

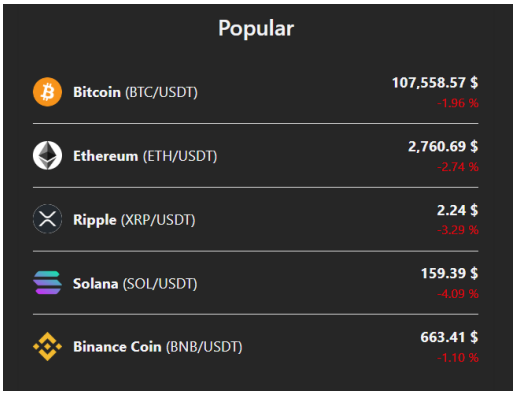


*Figura 27: Criptomonedas populares.*

Los usuarios verán la gráfica de Bitcoin en tiempo real.


*Figura 28: Gráfica de Bitcoin.*

### Interfaz de cierre de sesión

En esta sección el usuario verá su perfil asociado con datos como usuario y correo; también podrá cerrar la sesión activa.

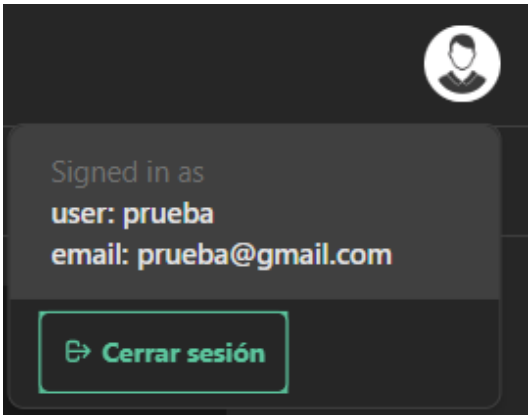


*Figura 29: Cierre de sesión.*

### Interfaz de envío

En esta página el usuario tendrá acceso a todas las funciones para realizar transacciones.

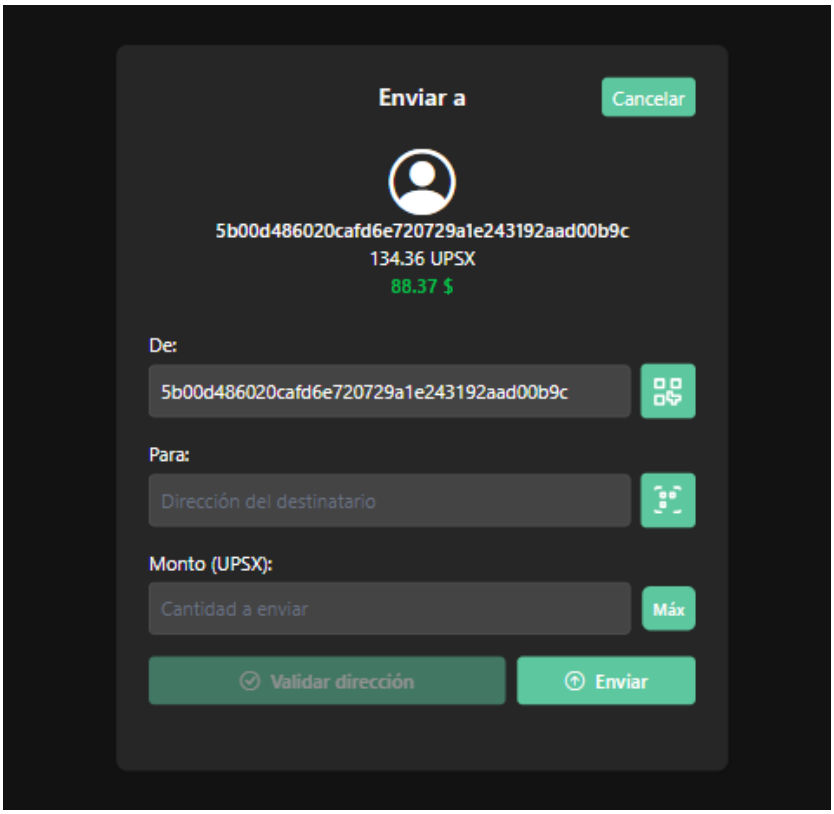


*Figura 30: Realizar transacciones.*

Aquí los usuarios podrán compartir su dirección de billetera hacia otros usuarios de forma fácil.


*Figura 31: Código QR de la dirección de billetera.*

Por otro lado, el usuario podrá escanear los códigos QR de los usuarios a quienes les va a enviar alguna transacción.

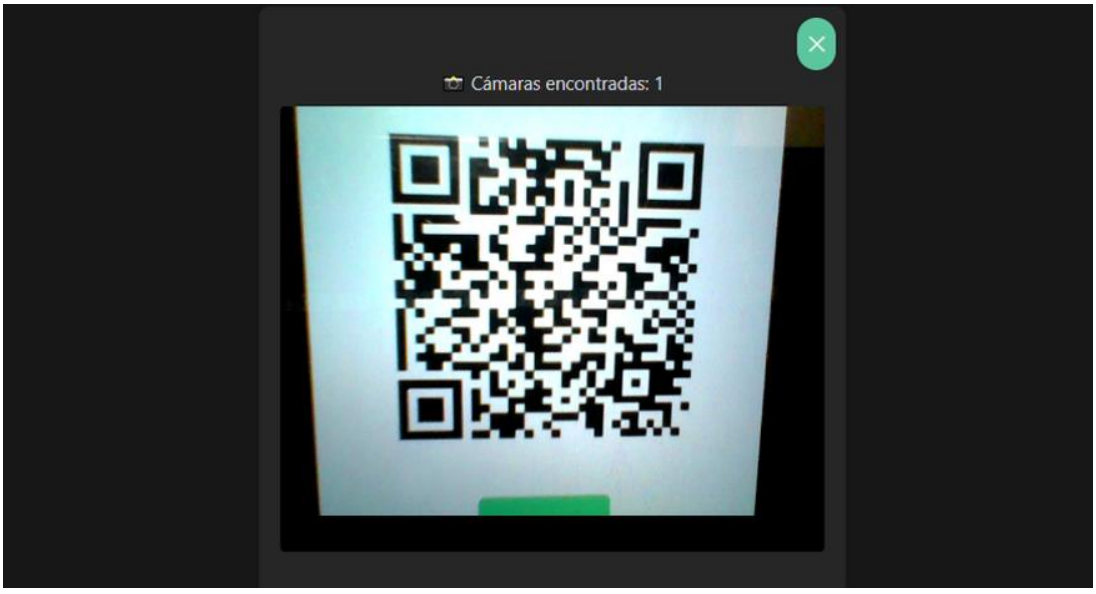


*Figura 32: Escáner de códigos QR.*

El sistema validará la dirección antes de poder realiza una transacción.

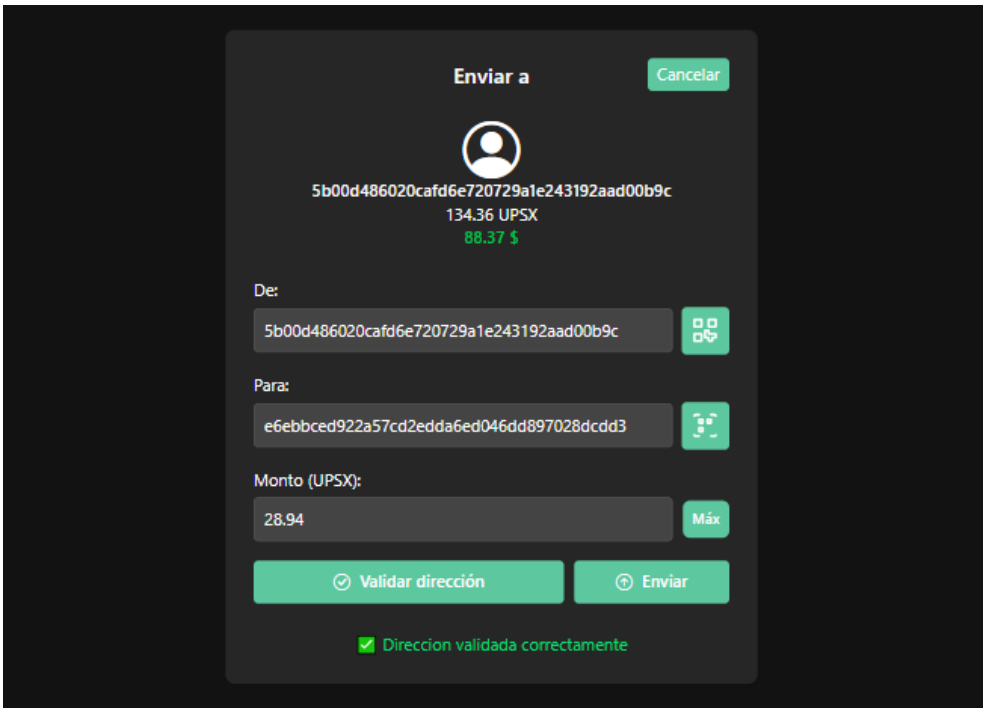


*Figura 33: Validar dirección de billetera.*

El usuario envía la transacción y ésta se guarda en la base de datos y en la blockchain.

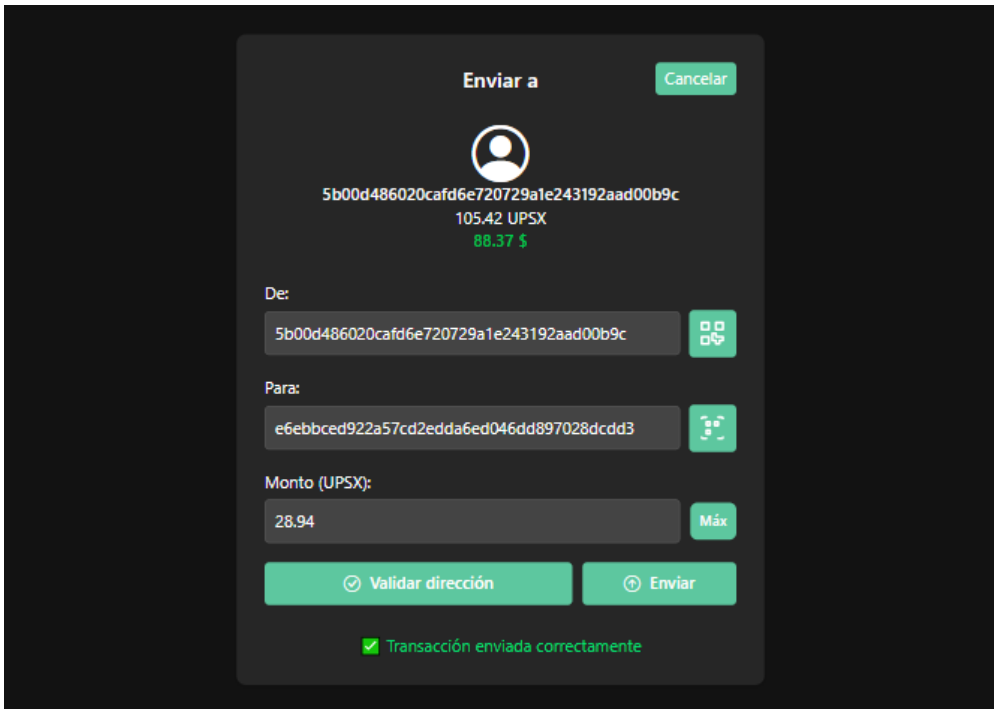


*Figura 34: Transacción realizada exitosamente.*

### Interfaz historial

El usuario podrá ver todos los movimientos de la cuanta como los canjes o intercambios, los envíos y los recibidos, todos ellos con su respectivo detalle,

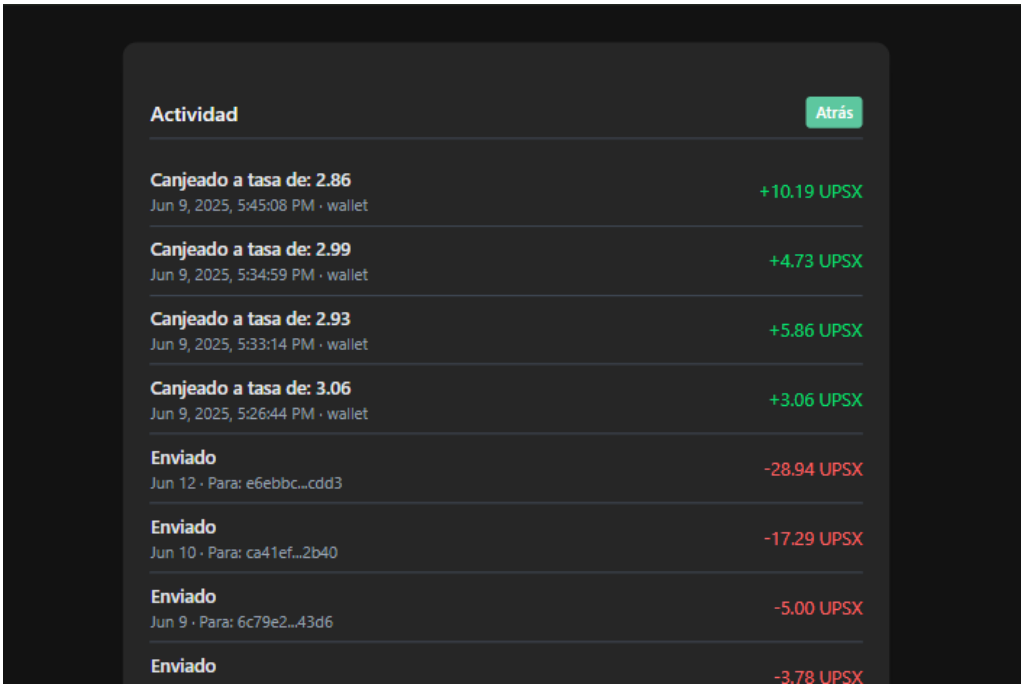


*Figura 35: Interfaz historial – 1.*

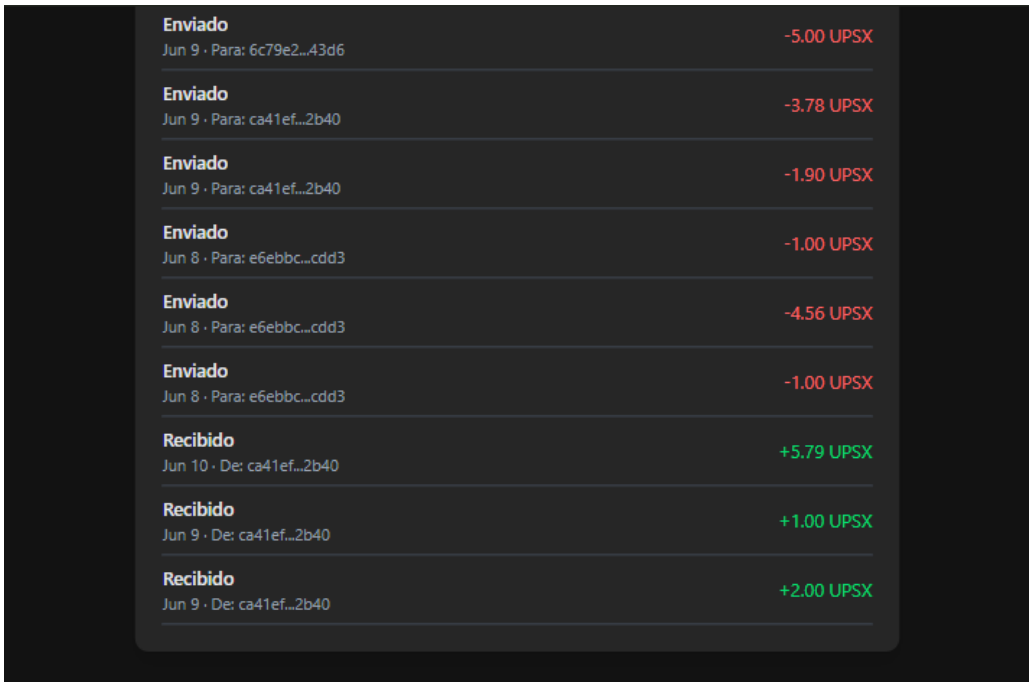


*Figura 36: Interfaz historial – 2.*

Una vez hecha una transacción se va a los nodos validadores y se almacena en su mempool para posteriormente propagarse hacia los demás nodos de la red.

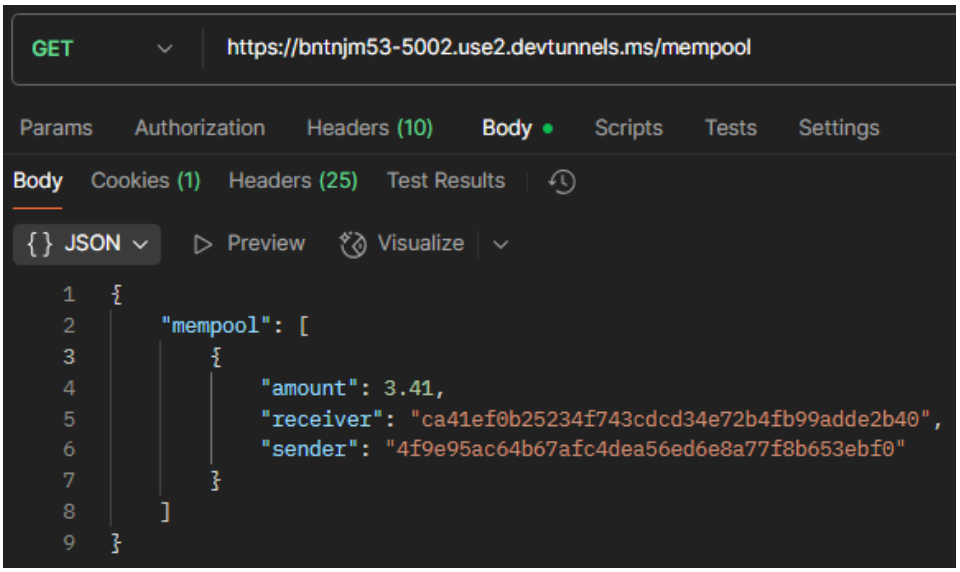


*Figura 37: Transacción almacenada temporalmente en la mempool del nodo 5002.*

**Nodos ejecutándose en los puertos simulando una blockchain.**

Nodo dos ejecutándose en el terminal de consola integrado en Spyder.

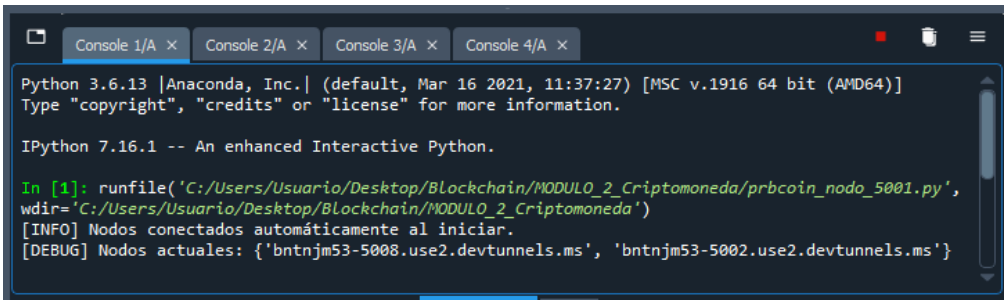


*Figura 38: Nodo Validador 1 ejecutándose en el puerto 5001.*

Nodo dos ejecutándose en el terminal de consola integrado en Spyder.

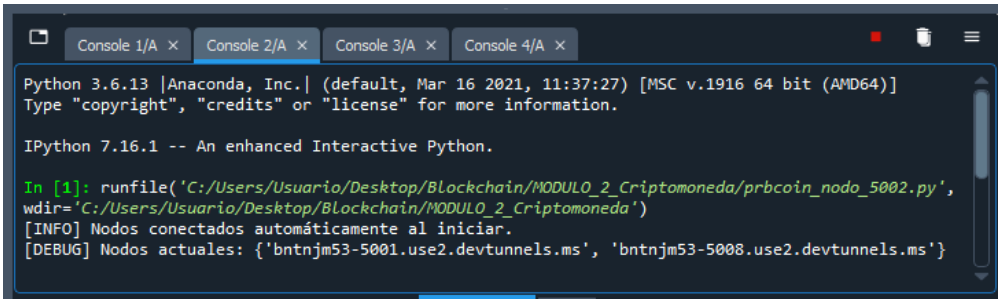


*Figura 39: Nodo Validador 2 ejecutándose en el puerto 5002.*

Nodo minero conectándose automáticamente a los nodos validadores.

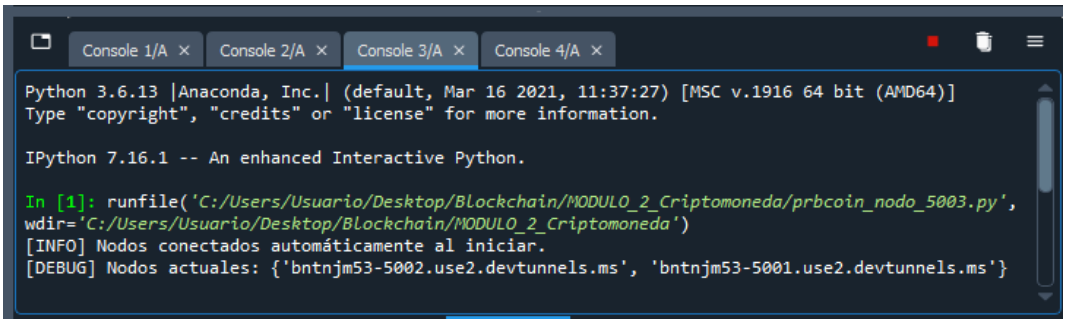


*Figura 40: Nodo Minero ejecutándose en el puerto 5008.*

El código de la aplicación más la lógica de la base de datos se ejecutan también en la consola de Spyder.

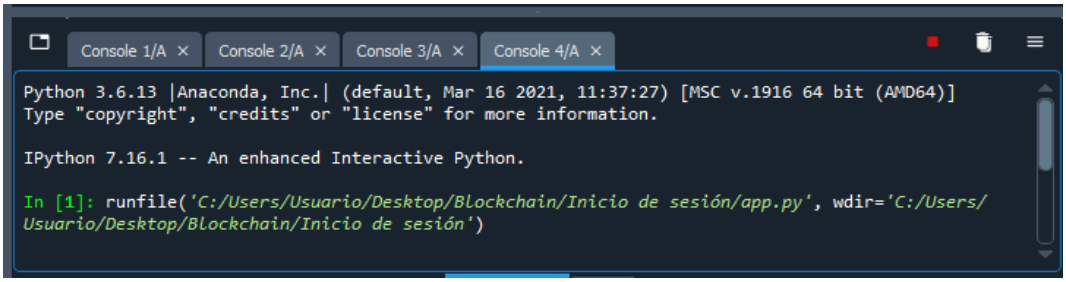


*Figura 41: Lógica de aplicación web ejecutándose en el puerto 5012.*

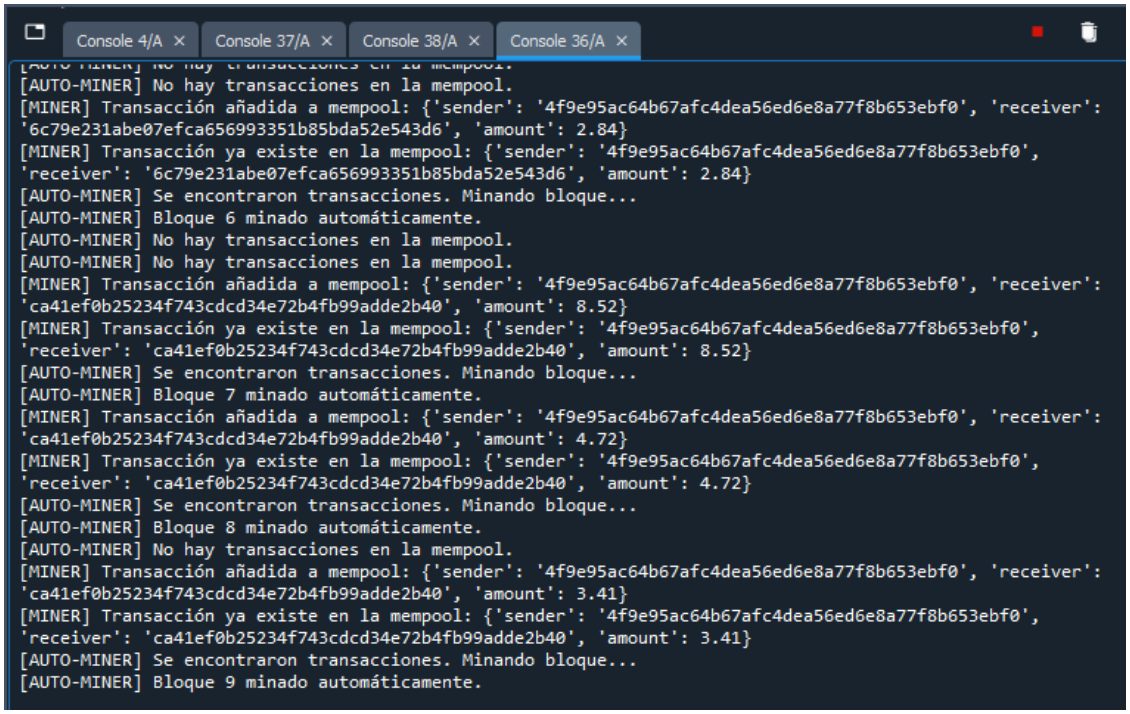


*Figura 42: Minado automático de bloques.*

Después de que el nodo minero cree un nuevo bloque mediante la simulación del método Proof of Work, el bloque es propagado a los otros dos nodos validadores, estos verifican y reemplazan su cadena a la correcta, teniendo así la cadena válida en todos los nodos, para así cumplir con el mecanismo de consenso.

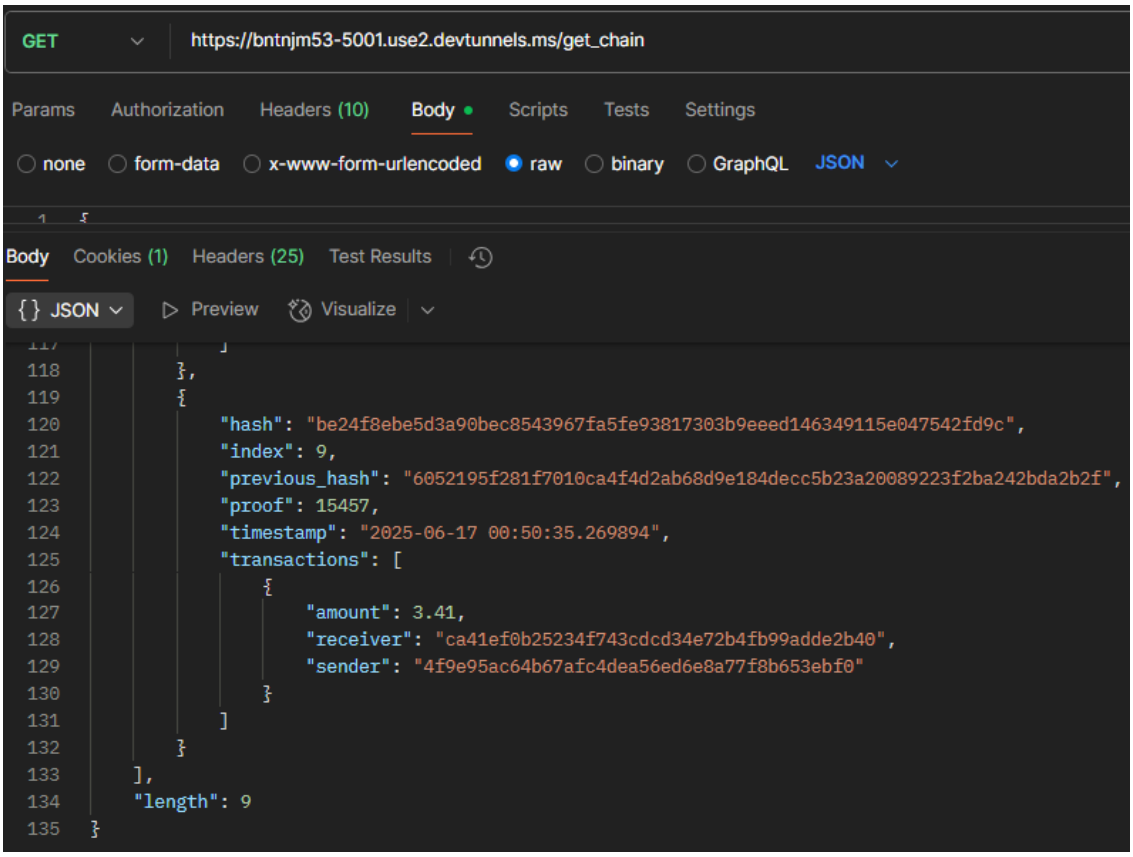


*Figura 43: Blockchain actualizada con el nuevo bloque minado.*

Escenario de alteración de la cadena de bloques modificando una transacción

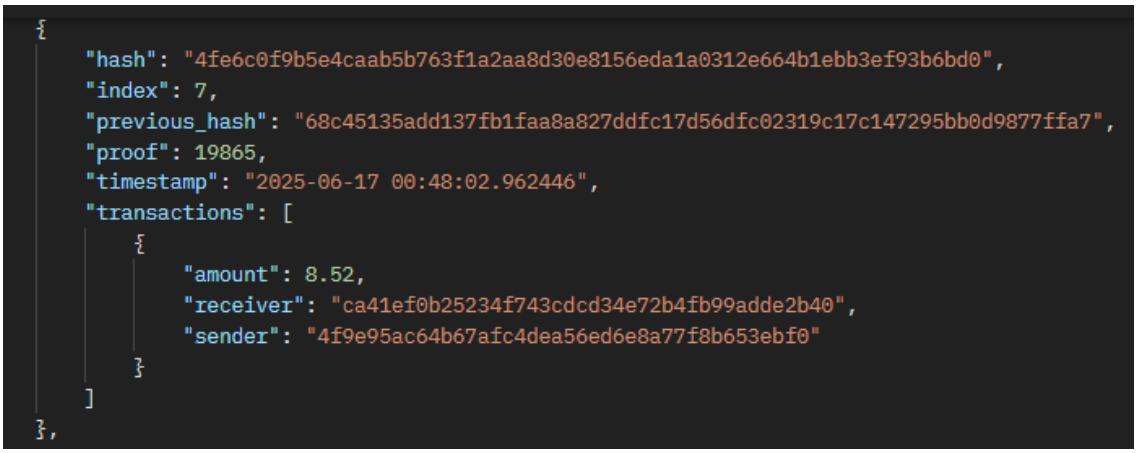


*Figura 44: Transacción antes de simulación de ataque.*

Simulamos un ataque mediante una solicitud POST y enviando parámetros en el cuerpo de la solicitud, esto lo hacemos con el fin de verificar que los nodos estén conectados correctamente, teniendo así comunicación para poder mantener la integridad del sistema en este caso cuando simulemos un ataque.

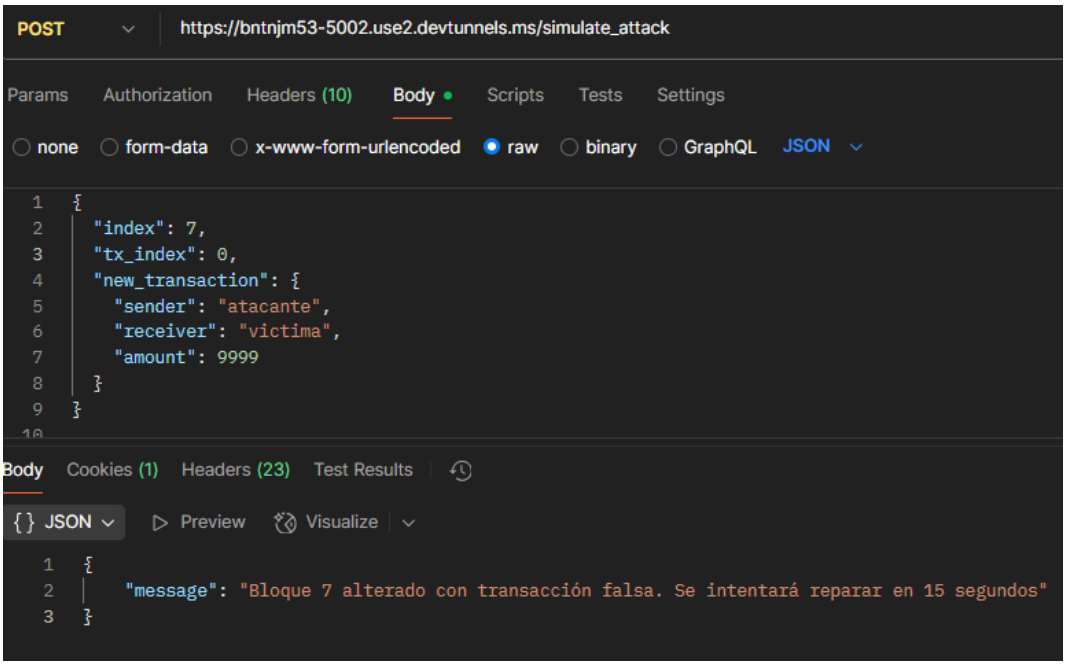


*Figura 45: Simulación de ataque.*

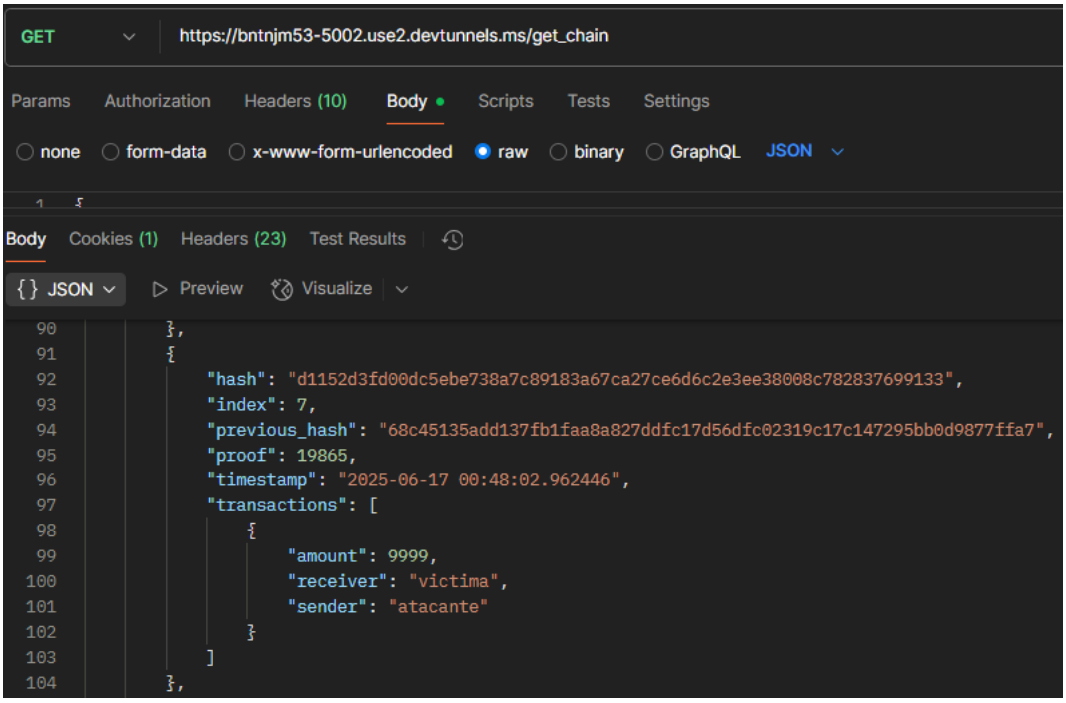


*Figura 46: Cadena alterada en el nodo 5002.*

En la consola integrada de Spyder vemos que el nodo afectado reemplaza la cadena alterada por la correcta

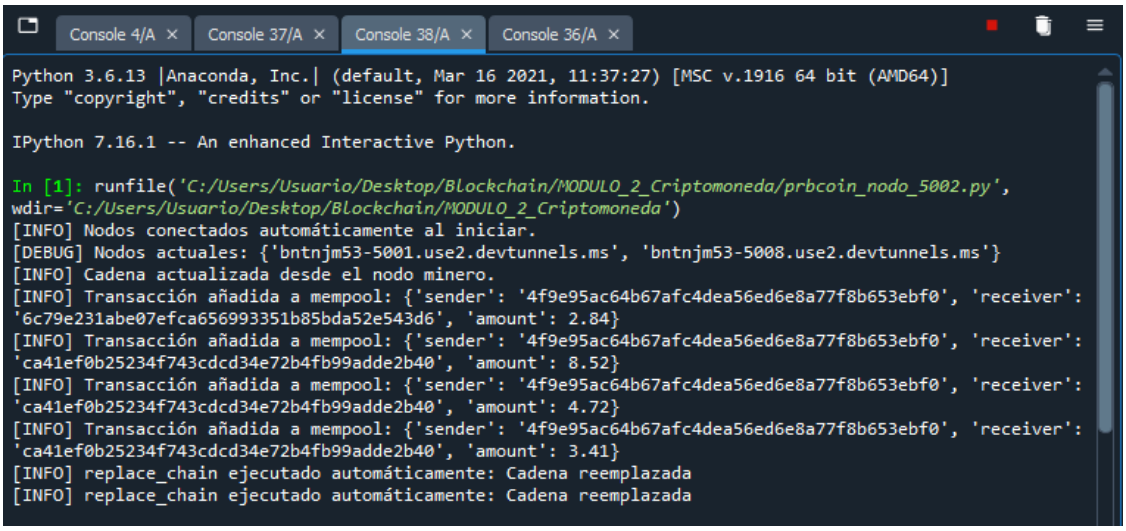


*Figura 47: Nodo afectado actualiza su cadena a la cadena válida.*

Verificamos mediante una consulta GET el estado de la cadena de bloques y podemos observar que tras pasar un periodo de tiempo muy corto la cadena fue reemplazada correctamente.

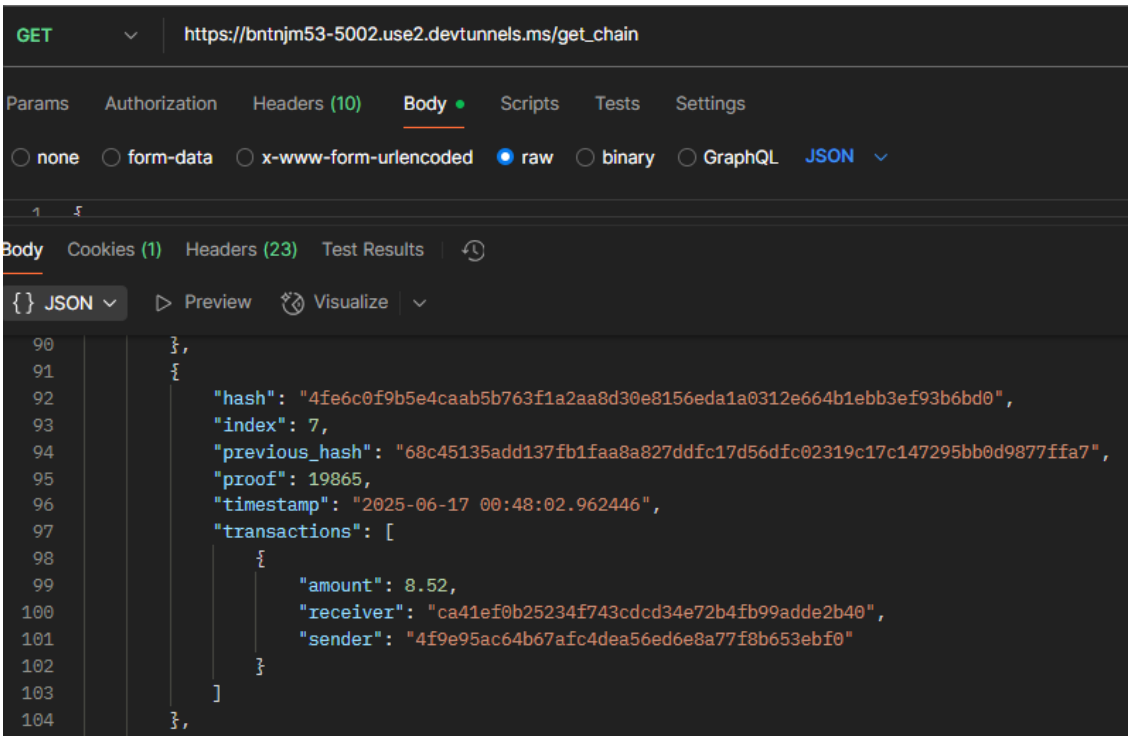


*Figura 48: Cadena actualizada correctamente.*

### 3.4. Pruebas

| Test 01 – Registro de usuario | | |
|---|---|---|
| **Descripción** | Los usuarios se registran con su nombre de usuario, correo y una clave. | |
| **Prerrequisitos** | El correo ingresado no debe haber sido usado por otro usuario.<br>La contraseña debe cumplir con los requisitos mínimos de seguridad. | |
| **Escenarios** | **Registro exitoso** | • El usuario ingreso todos los capos correctamente. |
| | **Registro fallido** | • El usuario intento registrarse con un correo en uso.<br>• El usuario no ingreso campos obligatorios.<br>• El usuario ingreso un correo no valido. |
| **Evaluación** | ✓ Exitosa ☐ Fallida | |

*Tabla 11: Prueba de registro de usuario.*

| Test 02 - Recuperación de cuenta | | |
|---|---|---|
| **Descripción** | Los usuarios reestablecen su cuenta, gracias al código de recuperación enviado al correo. | |
| **Prerrequisitos** | El correo debe constar en el sistema. | |
| **Escenarios** | **Recuperación exitosa** | • El usuario registró correctamente el código recibido por correo. |

| | | |
|---|---|---|
| | **Recuperación fallida** | • El correo no existe en la base de datos.<br>• El código de recuperación ingresado es incorrecto. |
| **Evaluación** | ✓ Exitosa ☐ Fallida | |

*Tabla 12: Prueba de recuperación de cuenta.*

| **Test 03 - Inicio de sesión** | | |
|---|---|---|
| **Descripción** | El usuario podrá entrar a la página principal si las credenciales son correctas. | |
| **Prerrequisitos** | Las credenciales del usuario deben existir en la base de datos. | |
| **Escenarios** | **Inicio de sesión exitosa** | • El usuario se logueó correctamente debido a credenciales válidas. |
| | **Inicio de sesión fallida** | • Credenciales incorrectas.<br>• Usuario no existe. |
| **Evaluación** | ✓ Exitosa ☐ Fallida | |

*Tabla 13: Prueba de inicio de sesión.*

| **Test 04 - Intercambio de criptomonedas (Swap)** | |
|---|---|
| **Descripción** | En la aplicación se podrá intercambiar el USDT equivalente al dinero real, por la criptomoneda UPSX. |
| **Prerrequisitos** | El usuario deberá contar con saldo USDT suficiente.<br>El usuario deberá ingresa una cantidad válida. |

| | | |
|---|---|---|
| **Escenarios** | **Intercambio exitoso** | • El usuario contaba con saldo necesario. |
| | **Intercambio fallido** | • Saldo insuficiente.<br>• Valor incorrecto.<br>• Campo vacío. |
| **Evaluación** | ✓ Exitosa ☐ Fallida | |

*Tabla 14: Prueba de intercambio de Criptomonedas.*

| **Test 05 - Envío de Criptomonedas** | | |
|---|---|---|
| **Descripción** | La aplicación permitirá el envío de UPSX hacia otros usuarios, registrando la transacción en la base de datos y en la cadena de bloques. | |
| **Prerrequisitos** | El usuario deberá contar con saldo UPSX suficiente.<br>El usuario deberá estar autenticado. | |
| **Escenarios** | **Envío exitoso** | • El usuario estaba autenticado y contaba con saldo necesario. |
| | **Envío fallido** | • Saldo insuficiente.<br>• Dirección de billetera incorrecta.<br>• Dirección vacía. |
| **Evaluación** | ✓ Exitosa ☐ Fallida | |

*Tabla 15: Prueba de envío de criptomonedas.*

| **Test 06 - Propagación de transacción hacia los nodos de la red** | |
|---|---|
| **Descripción** | La aplicación envía la transacción hacia los nodos validadores, y estos propagan la transacción a la red. |

| | | |
|---|---|---|
| **Prerrequisitos** | Los nodos de la red deben estar conectados y la blockchain funcional. | |
| **Escenarios** | **Envío exitoso** | • La transacción fue enviada correctamente a la base de datos y a la blockchain. |
| | **Envío fallido** | • Nodo validador caído.<br>• Propagación fallida.<br>• Error en el envío desde la aplicación web. |
| **Evaluación** | ✓ Exitosa ☐ Fallida | |

*Tabla 16: Prueba de propagación de transacciones a través de los nodos de la red.*

| **Test 07 - Minado de bloques y actualización de la cadena (Blockchain)** | | |
|---|---|---|
| **Descripción** | El nodo minero es el encargado de crear nuevos bloques y propagar la nueva cadena a la red de nodos. | |
| **Prerrequisitos** | El nodo minero debe estar conectado correctamente a los nodos validadores. | |
| **Escenarios** | **Envío exitoso** | • El nodo minero valida la transacción, crea el nuevo bloque, añade la transacción a ese bloque y propaga la nueva cadena. |
| | **Envío fallido** | • No hay transacciones en espera.<br>• Error de sincronización de nodos. |
| **Evaluación** | ✓ Exitosa ☐ Fallida | |

*Tabla 17: Prueba de minado de bloques y actualización de la blockchain.*

## CONCLUSIONES

Se logró la implementación de la simulación de una blockchain básica conformada por nodos validadores y mineros que simulan una red descentralizada, estos nodos están en puertos diferentes y se conectan automáticamente al ejecutarse replicando la lógica de una red distribuida real. La verificación de transacciones en espera se cumple y se ejecuta cada cierto tiempo, de cumplirse esta premisa se ejecuta el minado de bloques automático, proceso crucial para evitar problemas de generación de bloques vacíos. La sincronización entre nodos permite la correcta gestión de transacciones desde la mempool hasta que se minan nuevos bloques y se añaden a la cadena, garantizando así el consenso ente los nodos

Se desarrollo una aplicación web con la lógica para la simulación de transacciones con un prototipo de criptomoneda, así mismo se implementó autenticación mediante tokens JWT, esto permitió manejar correctamente los datos correspondientes a cada usuario. Se desarrollaron endpoints que además de encargarse del envío de transacciones, se encargan de la comunicación efectiva entre el frontend y el backend. La base de datos permite consultas de históricos de forma sencilla y rápida. Este formato combinado entre blockchain, base de datos y aplicación web no solo permite la gestión de transacciones, sino que representa un entorno seguro y funcional garantizando los principios de la Blockchain.

Las revisiones de integridad mostraron la aptitud del sistema para reaccionar ante fallos frente a posibles situaciones de hackeos o intentos de modificación en transacciones de la cadena de bloques. Los nodos están configurados para que, en caso de que se caigan al reinicio de sistema, se vinculen a la red y se actualicen de acuerdo a la cadena de bloques vigente, asegurando de esta manera la resiliencia del sistema. El nodo que tiene problemas necesita 15 segundos para verificar y actualizar su cadena a la cadena correcta, según lo que indican los otros nodos de la red. Esta acción mostró que el mecanismo de consenso utilizado es válido, ya que el nodo con problemas pudo arreglar su cadena de bloques garantizando la integridad de los datos.

## RECOMENDACIONES

En la actualidad sino estamos informados de las nuevas tecnologías, no podremos aprovechar las capacidades que éstas nos brindan, así como también es crucial tener cierto grado de educación financiera no solo para no sobre endeudarse sino para saber administrar el dinero, herramienta fundamental con la que se mueve el mundo. Es importante, además conocer el funcionamiento de las criptomonedas y blockchain, sobre todo para evitar ser estafado y perder dinero.

A parte del área de finanzas y pagos, la blockchain es aplicable a diversas áreas como la salud, identidad digital, cadena de suministro y logística, estas solo son unas pocas más usadas actualmente. Se recomienda indagar más sobre las distintas aplicaciones de la cadena de bloques en otras áreas y así aprovechar los beneficios más importantes tales como: descentralización, inmutabilidad y consenso.

Al tratarse de una aplicación que trata de temas financieros que maneja activos digitales y transacciones es fundamental contar con un sistema de validaciones robusto tanto en el frontend como del backend, así como contar con un sistema de autenticación de dos factores avanzado (2FA) para mejorar la seguridad del sistema.

## REFERENCIAS


[1] LISA Institute, «LISA Institute,» [En línea]. Available: https://www.lisainstitute.com/blogs/blog/que-es-blockchain-tipos-ejemplos-ventajas. [Último acceso: 22 marzo 2025].

[2] M. Javier, «Next Educación,» 29 agosto 2023. [En línea]. Available: https://www.nexteducacion.com/noticias/el-presente-de-las-criptomonedas/. [Último acceso: 10 marzo 2025].

[3] Coinbase, «Coinbase,» [En línea]. Available: https://www.coinbase.com/en-ca/learn/tips-and-tutorials/what-is-a-rug-pull-and-how-to-avoid-it. [Último acceso: 22 marzo 2025].

[4] C. J. Orgaz, «BBC News Mundo,» 20 febrero 2025. [En línea]. Available: https://www.bbc.com/mundo/articles/cj3n5gjd2dxo. [Último acceso: 7 marzo 2025].

[5] Banco Santander, «Banco Santander,» [En línea]. Available: https://www.bancosantander.es/glosario/criptomonedas. [Último acceso: 10 marzo 2025].

[6] Santander, «Santander,» 29 septiembre 2022. [En línea]. Available: https://www.santander.com/es/stories/guia-para-saber-que-son-las-criptomonedas. [Último acceso: 10 marzo 2025].

[7] Universidad Europea, «Universidad Europea,» 5 julio 2023. [En línea]. Available: https://universidadeuropea.com/blog/usos-python/. [Último acceso: 23 marzo 2025].

[8] Coinbase, «Coinbase,» [En línea]. Available: https://www.coinbase.com/es-la/learn/crypto-basics/what-is-a-crypto-wallet. [Último acceso: 23 marzo 2025].

[9] N. Thakur y Teccmark, «LinkedIn,» 5 octubre 2024. [En línea]. Available: https://www.linkedin.com/pulse/blockchain-development-using-python-comprehensive-guide-nizam-thakur-vs0vc. [Último acceso: 23 marzo 2025].

[10] J. M. García Hernandez, «Criptomonedas y aplicación en la economía,» Madrid, 2018.

[11] L. D. Laise y G. Manzo Ugas, «Bases para la interpretación y regulación razonable de las criptomonedas: naturaleza, dificultades y desafíos constitucionales,» *Cuadernos del CENDES*, vol. 36, nº 100, pp. 107-124, 2019.

[12] J. A. Corredor Higuera y D. Díaz Guzmán, «Blockchain y mercados financieros: aspectos generales del impacto regulatorio de la aplicación de la tecnología blockchain en los mercados de crédito de América Latina,» *Derecho PUCP*, nº 81, pp. 405-439, 2018.

[13] V. P. Castro-Rivera, R. A. Herrera Acuña y M. A. Villalobos Abarca, «Fundamentos de las pruebas continuas de software,» 2022.

[14] N. Salgado Reyes, «La tecnología Blockchain y su potencial para revolucionar la gestión de datos y la seguridad de las transacciones,» *Revista Científica FIPCAEC (Fomento De La investigación Y publicación científico-técnica multidisciplinaria)*, vol. 8, nº 2, pp. 546-562, 2023.

[15] A. Marzal y I. & Gracia, Introducción a la programación con Python, Universitat Jaume I., 2020.

[16] U. Revista, «UNIR La Universidad en Internet,» 7 octubre 2022. [En línea]. Available: https://www.unir.net/revista/ingenieria/que-es-typescript/. [Último acceso: 9 abril 2025].

[17] R. Casado Vara, Knowledge extraction and representation, Salamanca: Ediciones Universidad de Salamanca, 2019.

[18] J. Attardi, CSS moderno, Berkeley, CA: Apress, 1, 2020.

[19] Mailchimp, «Mailchimp,» [En línea]. Available: https://mailchimp.com/es/resources/rest-api/. [Último acceso: 9 abril 2025].

[20] Universidad Internacional de La Rioja (UNIR), 22 septiembre 2022. [En línea]. Available: https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/. [Último acceso: 21 abril 2025].

[21] A. R. Gudiño Quinteros, «Estudio de integración de los frameworks Angular 4 y Yii2, orientado a servicios REST, que permitan la gestión y control de inventarios para mejorar la productividad en la empresa Induxion,» *Universidad Técnica del Norte*, 2018.

[22] D. Ghimire, «Comparative study on Python web frameworks: Flask and Django,» Metropolia University of Applied Sciences., Helsinki, 2020.

[23] P. P. Kore, M. J. Lohar, M. T. Surve y S. Jadhav, «API Testing Using Postman Tool,» *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, nº 12, p. 841–843, 2022.

[24] Microsoft, 24 marzo 2025. [En línea]. Available: https://visualstudio.microsoft.com/es/. [Último acceso: 21 abril 2025].

[25] T. Huynh The, T. R. Gadekallu, W. Wang, G. Yenduri, P. Ranaweera, Q. V. Pham y M. Liyanage, «Blockchain for the metaverse: A review,» *Future Generation Computer Systems*, vol. 143, p. 401–419, 2023.

[26] E. J. Guaña Moya, H. N. Roa, F. R. Marcillo Vera, L. Ayavaca Vallejo, M. Chiluisa Chiluisa y B. Moya Carrera, «Tecnología Blockchain, qué es y cómo funciona,» *RISTI - Revista Ibérica de Sistemas e Tecnologias de Información*, vol. E54, pp. 101-114, 2022.

[27] S. Perera, S. Nanayakkara, M. N. N. Rodrigo, S. Senaratne y R. & Weinand, «Blockchain technology: Is it hype or real in the construction industry?,» *Journal of Industrial Information Integration*, vol. 17, p. 100125, 2020.

[28] K. John, M. O'Hara y F. Saleh, «Bitcoin and beyond,» *Annual Review of Financial Economics*, vol. 14, nº 1, p. 95–115, 2022.

[29] W. Metcalfe, «Ethereum, contratos inteligentes, DApps,» *Blockchain y criptomonedas*, p. 77–93, 2020.

[30] T. U. o. S. &. T. (MSMK), 9 enero 2025. [En línea]. Available: https://msmk.university/como-las-criptomonedas-y-proyectos-blockchain-construyen-su-identidad/. [Último acceso: 12 mayo 2025].

[31] P. I. Vizcaíno Zúñiga, R. J. Cedeño Cedeño y I. A. Maldonado Palacios, «Metodología de la investigación científica: guía práctica,» *Ciencia Latina*, vol. 7, nº 4, pp. 9723-9762, 2023.

[32] L. M. Retegui, «La observación participante en una redacción: Un caso de estudio,» *La trama de la comunicación*, vol. 24, nº 2, p. 103–119, 2020.

[33] B. F. Ocaña Valdez, «Repositorio Institucional de la Universidad Politécnica Salesiana,» febrero 2024. [En línea]. Available: http://dspace.ups.edu.ec/handle/123456789/26831. [Último acceso: 21 marzo 2025].

[34] P. A. Urrego Gordillo, «Implementación de un sistema de control de compras, ventas e inventarios en una microempresa ubicada en la ciudad de Bogota por medio de la metodología Extreme Programming (XP),» *Universidad Militar Nueva Granada*, 2023.

[35] Banco Santander, «Banco Santander,» 10 marzo 2025. [En línea]. Available: https://www.bancosantander.es/glosario/criptomonedas. [Último acceso: 10 marzo 2025].

[36] Vmark, «Vmark,» [En línea]. Available: https://vmark.eu/por-que-es-tan-importante-la-tecnologia-blockchain/. [Último acceso: 26 marzo 2025].

## ANEXOS

### Guía de herramientas y comandos por componente del sistema.

### Anexo 1: Frontend Angular.

| Herramienta / Librería | Comando de instalación | Etapa / Uso | Recomendación | Detalle |
|---|---|---|---|---|
| Angular CLI | npm install -g @angular/cli | Inicialización | Usar versión estable | Crear estructura del proyecto |
| Tailwind CSS | npm install -D tailwindcss postcss autoprefixer | Estilos UI | Usar ng-tailwindcss para integración rápida | Estilos modernos y rápidos |
| Preline | npm install preline | Componentes UI | Importar solo lo necesario | Mejorar apariencia con componentes |
| JWT para Angular | npm install @auth0/angular-jwt | Seguridad | No guardar tokens en localStorage sin cifrar | Leer tokens en interceptores |
| ngx-toastr (notificaciones) | npm install ngx-toastr | UX | Mostrar mensajes claros al usuario | Alertas visuales |
| sweetalert2 | npm install sweetalert2 | UX | Mensajes interactivos | Confirmaciones, alertas |
| Angular Forms | Incluido | Formularios | Validar bien antes de enviar | ReactiveForms o TemplateForms |

| | | | | |
|---|---|---|---|---|
| Angular Router | Incluido | Navegación | Rutas protegidas con guards | Manejo de rutas |
| Interceptor (token JWT) | Manual | Seguridad | Crear e inyectar en providers | Adjuntar token a peticiones |
| Environment vars | Manual (environment.ts) | Configuración | Separar dev y prod | URL backend, flags, etc. |
| Build producción | ng build --configuration=production | Deploy | Minimiza y optimiza | Listo para producción |
| Deploy en hosting | depende del hosting | Publicación | Usar HTTPS y configurar CORS | Subida del frontend |

*Tabla 18: Anexo 1; Guía - Frontend Angular.*

## Anexo 2: Backend Python

| Herramienta / Librería | Comando de instalación | Etapa / Uso | Recomendación | Detalle |
|---|---|---|---|---|
| Flask | pip install flask | API backend | Proyecto modularizado | Servidor principal |
| Flask-CORS | pip install flask-cors | CORS | Permitir solo origen Angular | Permite conexión frontend |
| Flask-JWT-Extended | pip install flask-jwt-extended | Seguridad | Tokens con expiración | Autenticación de usuarios |
| Flask-Bcrypt | pip install flask-bcrypt | Hashing | No guardar contraseñas planas | Hasheo de contraseñas |
| passlib | pip install passlib | Hashing | Verificación de contraseñas | Compatibilidad adicional |

| | | | | |
|---|---|---|---|---|
| pymysql | pip install pymysql | Base de datos | Usar try/except y cerrar conexiones | Conexión a MySQL |
| smtplib | Incluido en Python | Email | Correo seguro con app password | Envío de códigos de recuperación |
| uuid | Incluido en Python | Generación de códigos | Generar códigos únicos | Para recuperación de cuenta |
| waitress | pip install waitress | Producción | Mejor que Flask dev server | Servidor WSGI |
| requests | pip install requests | Conexión a nodos | Reintento si falla conexión | Llamadas HTTP a nodos |
| datetime | Incluido en Python | Fechas | Formatear bien fechas | Registro de eventos y logs |
| json / jsonify | Incluido en Flask | Respuestas | JSON claro y estructurado | Comunicación con frontend |
| Conexión con nodos (HTTP POST) | requests.post (url, json=data) | Blockchain | Manejar errores de conexión | Enviar datos a nodos |
| Config archivo .env | pip install python-dotenv (opcional) | Variables sensibles | No subir claves a repositorio | Manejo de claves, puertos, tokens |
| Logging (opcional) | import logging | Registro | Guardar errores y eventos | Facilita depuración |

*Tabla 19: Anexo 2; Guía - Backend Python.*

## Anexo 3: Nodos validadores y mineros

| Herramienta / Librería | Comando de instalación | Uso | Recomendación | Detalle |
|---|---|---|---|---|
| Flask | pip install flask | Servidor HTTP | Puerto diferente por nodo | Recibir transacciones / bloques |
| requests | pip install requests | Comunicación entre nodos | Manejar errores de red | Enviar bloques y transacciones |
| hashlib | Incluido en Python | Hash de bloques | Usar SHA-256 | Integridad de la cadena |
| json | Incluido en Python | Serializar datos | Usar .json() y json.dumps() | Manejo de info entre nodos |
| datetime | Incluido en Python | Tiempos en bloques | Usar isoformat() | Marcar tiempo de creación |
| uuid | Incluido en Python | ID único en transacciones | Para evitar duplicados | ID de transacción |
| threading | Incluido en Python | Paralelismo básico | Evitar bloqueos | Minado en segundo plano (nodo minero) |
| time | Incluido en Python | Simulación y delay | Para tiempo de espera en minado | Delay entre intentos de minado |
| Estructura por nodo | Manual | Organización | 1 archivo por nodo (ej: nodo1.py) | Más claro y fácil de correr |

| | | | | |
|---|---|---|---|---|
| Puerto específico | Manual (ej: 5000, 5001, 5002) | Flask API | Evitar colisiones | Cada nodo escucha en su puerto |
| Comunicación REST | requests.post/get() | Sincronización | Establecer dirección IP del otro nodo | Conexión simple entre nodos |
| Cadena blockchain propia | Manual (variable en Python) | Validación y replicación | Copiar cadena de nodo más largo | Mantener consistencia |
| Verificación de hash anterior | Manual (comparación simple) | Validación de bloques | Solo aceptar bloques válidos | Evitar manipulación |
| Validación en paralelo | threading + while | Minado continuo (nodo minero) | Usar control con flag | Simula minería sin parar |

*Tabla 20: Anexo 3; Guía - Nodos validadores y mineros.*