



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Desarrollo de una herramienta para la simulación y  
evaluación de estrategias de inversión en mercados de  
criptomonedas

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Pujol Martorell, Sergio

Tutor/a: Sánchez Anguix, Víctor

Cotutor/a: Alberola Oltra, Juan Miguel

CURSO ACADÉMICO: 2022/2023

# Resumen

El objetivo principal de este trabajo es desarrollar una herramienta enfocada en el trading automatizado de criptomonedas que simplifique al usuario la acción de realizar análisis del mercado mediante el uso de gráficas e indicadores. Para alcanzar este objetivo, se presentan una aplicación de escritorio y una aplicación web capaces de iniciar tareas gestionadas por un sistema de análisis de procesamiento que, mediante el uso de estrategias técnicas, realiza inversiones automáticas y simulaciones en mercados de criptomonedas.

Como resultado, la herramienta permite al usuario iniciar bots desde la aplicación con el fin de realizar trading automatizado en tiempo real y simulaciones utilizando históricos de datos y estrategias técnicas personalizables.

**Palabras clave:** criptomonedas, trading, automatización, aplicación, simulación.

# Abstract

The main goal of this work is to develop a tool focused on automated cryptocurrency trading that simplifies the user's action of performing market analysis using charts and indicators. To achieve this goal, a desktop application and a web application are presented that are able to start tasks managed by a processing analysis system that, through the use of technical strategies, performs automatic investments and simulations in cryptocurrency markets.

As a result, the tool allows the user to start bots from the application in order to perform automated trading in real time and simulations using customizable data histories and technical strategies.

**Keywords:** cryptocurrencies, trading, automation, application, simulation.



# Índice

<b>Resumen</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>Índice</b>	<b>4</b>
<b>Índice de figuras</b>	<b>6</b>
Figuras	6
Tablas	7
<b>Capítulo I</b>	
<b>Introducción</b>	<b>9</b>
1.1 Introducción	9
1.2 Motivación	9
1.3 Objetivos del proyecto	10
1.4 Estructura del proyecto	10
<b>Capítulo II</b>	
<b>Marco teórico</b>	<b>12</b>
2.1 Criptomonedas	12
2.1.1 Origen de las criptomonedas	12
2.1.2 Blockchain	13
2.1.3 Mercado actual	14
2.1.4 Criptomonedas Populares	15
2.2 Aplicaciones de las criptomonedas	15
2.2.1 Plataformas de intercambio de criptomonedas	16
2.3 Trading y Scalping	17
2.3.1 Automatización del trading	18
2.3.1.1 Funcionamiento	18
2.3.1.2 Ventajas y desventajas	19
2.3.1.3 Mercado	19
2.3.1.4 Aplicaciones de trading automatizado	20
<b>Capítulo III</b>	
<b>Desarrollo del proyecto</b>	<b>22</b>
3.1 Metodología	22
3.1.1 Metodología de trabajo	22
3.1.2 Roadmap	22
3.1.3 Control de versiones	23
3.2 Análisis de requisitos	24
3.3 Tecnologías de la aplicación	25
3.4 Diseño	25
3.4.1 Arquitectura del proyecto	26
3.4.2 Base de datos	28
3.4.3 Logotipo y Mockups	31

3.5 Implementación de la Aplicación	37
3.5.1 Front-end	38
3.5.2 Comunicación	38
3.6 Implementación del análisis de datos	39
3.6.1 APIs	40
3.6.2 Bots	40
3.6.3 Estrategias	42
3.6.4 Operaciones	44
3.6.5 Simulación	46
3.7 Tests	47
<b>Capítulo IV</b>	
<b>Resultados</b>	<b>48</b>
4.1 Resultados de trading	48
4.2 Resultados de simulación	49
4.3 Modelo de negocio	51
4.4 Manual de usuario	52
<b>Capítulo V</b>	
<b>Conclusión</b>	<b>53</b>
<b>Capítulo VI</b>	
<b>Referencias</b>	<b>55</b>

# Índice de figuras

## Figuras

- Figura 1: Representación gráfica del funcionamiento de la blockchain. Fuente: <https://www.investopedia.com/terms/b/blockchain.asp>.
- Figura 2: Gráfico de capitalización total del mercado de criptomonedas. Fuente: <https://coinmarketcap.com/es/charts/>.
- Figura 3: Listado de criptomonedas ordenadas por capitalización de mercado el 19-11-2022 a las 18:35. Fuente: <https://coinmarketcap.com/es/>.
- Figura 4: Roadmap.
- Figura 5: Arquitectura del proyecto de aplicación de escritorio.
- Figura 6: Arquitectura del proyecto de aplicación web.
- Figura 7: Diseño base de datos de autenticación de aplicación de escritorio.
- Figura 8: Diseño base de datos local de aplicación de escritorio.
- Figura 9: Diseño base de datos de aplicación web.
- Figura 10: Paleta de colores.
- Figura 11: Logotipo.
- Figura 12: Diagrama de navegación página inicio.
- Figura 13: Diagrama de navegación.
- Figura 14: Barra de navegación superior.
- Figura 15: Vista principal de Trading.
- Figura 16: Vista de Trading con popup de settings en aplicación de escritorio y web.
- Figura 17: Vista de Trading con dos gráficas.
- Figura 18: Vista de Trading con popup de crear un bot.
- Figura 19: Vista de Trading con un bot añadido en la segunda gráfica.
- Figura 20: Vista principal de Simulation.
- Figura 21: Vista de Simulation con popup de configuración del bot.
- Figura 22: Vista de Simulation con datos obtenidos.
- Figura 23: Diagrama de conexión entre servidor de Web y servidor de Base de datos.
- Figura 24: Diagrama de conexión entre servidor de Base de datos y servidor de procesamiento de datos.
- Figura 25: Esquema del objeto que almacena los bots en el sistema de procesamiento de datos.
- Figura 26: Esquema del proceso completo de un bot en el sistema de procesamiento de datos.
- Figura 27: Representación gráfica de estrategia DEMA. Fuente: <https://es.tradingview.com/chart/?symbol=ETHUSD>.
- Figura 28: Representación gráfica de estrategia MACD. Fuente: <https://www.investopedia.com/terms/m/macd.asp>.
- Figura 29: Representación gráfica de estrategia Bollinger. Fuente: <https://www.investopedia.com/trading/using-bollinger-bands-to-gauge-trends/>.
- Figura 30: Parámetros para new Order de Binance API. Fuente: <https://developers.binance.com/docs/binance-trading-api/spot#new-order--trade>.
- Figura 31: Ejemplo de respuesta de new Order de Binance API. Fuente: <https://developers.binance.com/docs/binance-trading-api/spot#new-order--trade>.

Figura 32: Esquema del proceso completo de un bot en el sistema de procesamiento de datos.

Figura 33: Gráfico de Bitcoin de 1m de Intervalo desde las 9:30 del 21/11/2022 hasta las 16:30 del 21/11/2022. Fuente: <https://tradingview.com/>.

Figura 34: Gráfica de porcentajes resultantes de la simulación con la estrategia DEMA.

Figura 35: Gráfica de porcentajes resultantes de la simulación con la estrategia MACD.

Figura 36: Gráfica de porcentajes resultantes de la simulación con la estrategia de Bandas de Bollinger.

Figura 37: Visualización de niveles de planes de suscripción del modelo de negocio.

## Tablas

Tabla 1: Exchanges más conocidas ordenadas por volumen de 24h el 25/07/2022. Fuente: <https://www.coingecko.com/en/exchanges/>.

Tabla 2: Pros y contras de los competidores de aplicaciones de trading automatizado.

Tabla 3: Resultados de estrategia 2EMA con diferentes intervalos.

Tabla 4: Resultados de estrategia MACD con diferentes intervalos.

Tabla 5: Resultados de estrategia Bollinger con diferentes intervalos.



# Capítulo I

## Introducción

### 1.1 Introducción

Las criptomonedas son un concepto revolucionario que ha cambiado el panorama y el rumbo de muchas de las tecnologías actuales. Estas consisten en activos digitales que emplean un cifrado criptográfico para garantizar su titularidad y asegurar la integridad de sus transacciones. Debido al gran interés que han generado en los últimos tiempos, se han desarrollado muchas de sus utilidades, entre ellas, el trading de monedas digitales.

El trading consiste en la acción de compraventa mediante la especulación de precios de activos cotizados y con suficiente liquidez en el mercado. Esta actividad se realiza mediante diferentes análisis, siendo el más común y efectivo el técnico, utilizando gráficas para predecir precios futuros o curvas de tendencia. Ante esta situación, surgió la posibilidad de realizar este proceso matemático automáticamente, originándose así el trading automatizado.

Como resultado, la propuesta de este proyecto consiste en la realización de un sistema capaz de invertir automáticamente, mediante la evaluación con diferentes estrategias técnicas. En consecuencia, el objetivo principal del trabajo es el desarrollo de una herramienta en forma de aplicación de escritorio o aplicación web, con un sistema de inversión de criptomonedas automatizado en tiempo real y para la simulación con históricos de datos mediante el uso de estrategias.

Este trabajo explica el proceso seguido para el desarrollo del proyecto, empezando por estudios teóricos y concepción de la idea hasta la comprobación del correcto funcionamiento de la herramienta diseñada.

### 1.2 Motivación

Como se ha comentado en la introducción, el trading es actualmente una actividad producida por la gran mayoría de usuarios con activos digitales, con la intención de obtener beneficios. Al tratarse de operaciones técnicas es posible la automatización de los procesos.

Ante esta nueva tecnología y el interés que genera, se planteó la opción de crear un sistema de trading que permita realizar operaciones según las condiciones del mercado. Como resultado, se decidió crear una herramienta con la capacidad de realizar inversiones de forma automática en tiempo real y efectuar simulaciones con históricos de datos, mediante un sistema de toma de decisiones aplicando estrategias técnicas.

Con el fin de ofrecer una herramienta totalmente completa, se ha diseñado una aplicación de escritorio ejecutada localmente y una aplicación web. De esta forma, el usuario podrá elegir que tipo de aplicación desea utilizar según sus necesidades.

Además, la principal motivación personal para realizar este proyecto es la posibilidad de poner en práctica todos los conocimientos adquiridos durante el transcurso del Grado de Tecnologías Interactivas aplicados a un tema tan interesante como son las criptomonedas. Al desarrollar el trabajo, se han utilizado muchos de los conocimientos de las asignaturas cursadas, y muy especialmente la programación, diseño de interfaces y experiencia de usuario, Big Data y álgebra matricial y geometría, o proyectos como el de desarrollo web.

### 1.3 Objetivos del proyecto

El principal objetivo de este trabajo es el desarrollo de una herramienta enfocada al trading de criptomonedas, con la capacidad de realizar trading automatizado y simulaciones con históricos de datos, ambos mediante el uso de estrategias basadas en análisis técnico.

Con el fin de alcanzar el objetivo principal, se deben cumplir los siguientes objetivos secundarios:

- Analizar el mercado actual de las criptomonedas, el trading y la automatización del mismo para validar las diferentes opciones y reforzar la idea del proyecto.
- Obtener información sobre las diferentes aplicaciones del trading automatizado y realizar un análisis de los mayores competidores.
- Realizar un estudio con los requisitos del proyecto para la correcta elección del tipo de aplicación a desarrollar.
- Estudiar las diferentes estrategias posibles dentro del mercado del trading de las criptomonedas.
- Implementar un sistema de análisis de toma de decisiones mediante la aplicación de estrategias técnicas.
- Realizar un sistema de compra y venta de activos digitales de forma automática mediante una plataforma de intercambio.
- Simular el trading mediante históricos de datos haciendo uso de estrategias técnicas.
- Crear y utilizar pruebas de validación para la comprobación del correcto funcionamiento del trabajo realizado.

### 1.4 Estructura del proyecto

El contenido restante de la memoria está estructurado de la siguiente manera:

## **Capítulo II: Marco teórico**

En el segundo capítulo, se realiza una recopilación de información e investigación respecto a la situación de los topics principales del proyecto. Entonces, se efectúa un análisis de la tecnología y de la situación de las criptomonedas, así como de sus posibles aplicaciones. De la misma forma, se indaga sobre el trading de los activos digitales, su automatización y la situación actual de herramientas del trading automatizado, diferenciando sus ventajas y desventajas.

## **Capítulo III: Desarrollo del proyecto**

En este capítulo se explica el proceso de creación del producto. Primeramente, se indica la metodología, el análisis de requisitos, la tecnología utilizada para el desarrollo y los diseños. Posteriormente, en cuanto al ámbito de creación, se diferencia entre la implementación de las aplicaciones y la implementación del sistema de análisis automatizado de los datos mediante diversas estrategias.

## **Capítulo IV: Resultados**

El cuarto capítulo se centra en el análisis de los resultados obtenidos en el sistema de simulación y trading mediante estrategias de inversión, permitiendo así determinar las mejores estrategias y comportamientos en función del momento y la configuración establecida. Una vez analizados los resultados, se ha realizado una propuesta de modelo de negocio para el producto final, teniendo en cuenta la situación del mercado de las criptomonedas y el trading.

## **Capítulo V: Conclusiones**

En el sexto capítulo, se expone una reflexión final del trabajo realizado. En ésta se indican los objetivos cumplidos, las posibles propuestas para la mejora del producto y, para finalizar, la relación del trabajo con los objetivos de desarrollo sostenible.

## **Capítulo VI: Referencias**

Finalmente, el último capítulo incluye las citas y documentos utilizados durante la creación de la memoria.

## Capítulo II

# Marco teórico

Para realizar un proyecto es necesario conocer de manera previa el ámbito en el cual se va a trabajar. De esta forma, el primer paso es realizar una investigación sobre aquellos conocimientos necesarios para poder desarrollar una idea. A continuación, se describen qué son las criptomonedas, su mercado actual y sus diferentes aplicaciones.

## 2.1 Criptomonedas

Durante los últimos años, se han generado nuevas tecnologías del mundo moderno y también se han desarrollado muchas otras, descubriéndose así nuevos conceptos. Sin embargo, muy pocas han conseguido destacar manteniendo a su vez la atención de millones de personas a lo largo del tiempo. Las criptomonedas son un ejemplo de tecnología digital que consigue seguir atrayendo y generando interés incluso 15 años después de su aparición.

Estas representan un bien digital y tienen el propósito de utilizarse como medio descentralizado de cambio de forma segura [1]. Al no tener un sistema de autoridad centralizado, se realizan y se registran las transacciones financieras mediante funciones criptográficas.

Las criptomonedas aprovechan la tecnología de blockchain para obtener la descentralización, transparencia e inmutabilidad. Una de las características más importantes de las criptomonedas y la blockchain, es la exclusión de intermediarios de instituciones financieras, por ello los costos de transacción se reducen [2].

### 2.1.1 Origen de las criptomonedas

La criptomoneda como concepto no se desarrolló de la noche a la mañana. El origen se remonta a los años 70 con el movimiento Cypherpunk. Previo a esa fecha, la encriptación únicamente se utilizaba con motivos militares.

Con la publicación de *“Security without Identification: Transaction Systems to make Big Brother Obsolete”* por el Dr. David Chaum en el año 1980 [3] se empezó a hablar sobre el dinero digital anónimo y sobre los sistemas de reputación seudónima. Este artículo supuso un gran avance para las actuales criptomonedas.

Fue en 1992 cuando un grupo de criptógrafos de San Francisco empezaron a trabajar en lo que llegaría a ser Cypherpunk. Se denomina Cypherpunk al movimiento social de individuos, instituciones, tecnologías y acciones políticas que con un enfoque descentralizado, defienden, codifican o recurren a sistemas de encriptación sólidos para reestructurar asimetrías sociales, políticas y económicas [4].

Cinco años más tarde, en 1997 se creó el HashCash como medida contra el spam. Y al año siguiente, se ideó el Proof of Work (PoW), concepto basado en la búsqueda de números primos que se introduciría posteriormente en el diseño de Peer-to-Peer [5]. Entonces, en 2008, surgió la idea de Bitcoin por Satoshi Nakamoto, publicando el libro blanco de Bitcoin, el cual se originó a causa de la unión de conceptos del movimiento Cypherpunk y la criptografía. En un futuro, surgirían nuevas criptomonedas, como Namecoin o Litecoin en 2011.

La razón principal que incitó el desarrollo de las criptomonedas se remonta a la grave crisis económica de 2008. La concesión de hipotecas por parte de los bancos afectó a las finanzas de Estados Unidos y se extendió por todo el mundo, llevando a la población a cuestionar la estabilidad y transparencia del sistema bancario [6].

Esta necesidad de dotar de independencia financiera y de esta forma, idear una moneda descentralizada, llevó a cabo la creación de una nueva forma de dinero, las denominadas criptomonedas. Esta nueva moneda garantizaría la seguridad mediante la criptografía, asegurando el tráfico digital en la red de los usuarios y consiguiendo reemplazar el modelo basado en intervención de terceros. Esta seguridad y la eliminación de la necesidad de un intermediario se consigue mediante la tecnología blockchain.

## 2.1.2 Blockchain

La blockchain es un libro digital de transacciones económicas que se puede utilizar para registrar no solo transacciones financieras, sino cualquier objeto con un valor intrínseco [7]. Es decir, la blockchain consiste en una serie de registros inmutables con marcas de tiempo, las cuales se administran por un cluster de máquinas sin identidad. Cada bloque de datos está protegido criptográficamente y conectado uno a otro mediante una cadena [8].

En este sentido, la blockchain permite crear un almacenamiento distribuido de documentos con marcas de tiempo donde ninguna parte puede manipular el contenido. Los problemas de autenticación e integridad se resuelven mediante las firmas digitales, entonces cuando una parte crea una firma digital para un documento, se establece un vínculo verificable entre esta y el documento [9]. En la Figura 1, se puede observar el proceso de la blockchain al realizar una nueva transacción.

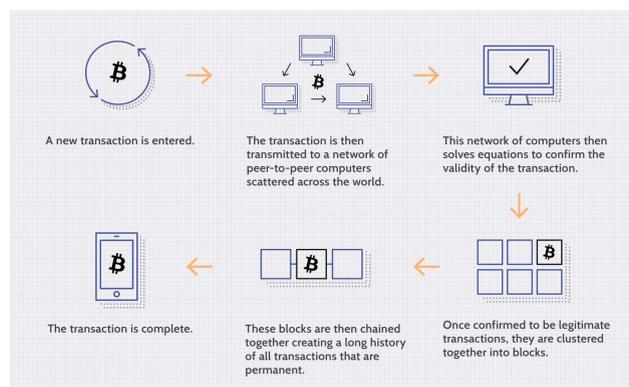


Figura 1: Representación del funcionamiento de la blockchain. Fuente: <https://www.investopedia.com/terms/b/blockchain.asp/>.

Esta tecnología fue utilizada por primera vez por Satoshi Nakamoto en el artículo de “*Bitcoin: A Peer-to-Peer Electronic Cash System*” [10]. El problema que resuelve Nakamoto con la blockchain es la falta de confianza en un sistema distribuido.

Gracias a esta tecnología, se han desarrollado más activos digitales con diferentes funcionalidades, favoreciendo que el mercado aumente drásticamente durante los últimos años.

### 2.1.3 Mercado actual

En noviembre de 2022, existen más de 20.000 criptomonedas y casi 500 Mercados/Bolsas de criptomonedas. La capitalización del mercado es de 831.07 billones de dólares estadounidenses, con un volumen actual de operaciones de 128 billones de dólares en un rango de 24 horas [11]. En la Figura 2, se puede observar el gráfico de la capitalización de las criptomonedas desde el 2014 hasta la actualidad.



Figura 2: Gráfico de capitalización total del mercado de criptomonedas. Fuente: <https://coinmarketcap.com/es/charts/>.

El mercado de las criptomonedas es muy amplio, pero hay determinadas monedas que destacan de la gran mayoría por su capitalización de mercado y su volumen. En este instante, 19 de Noviembre de 2022, las monedas más populares y con mayor capitalización de mercado según CoinMarketCap son:

1. Bitcoin, 2. Ethereum, 3. Tether, 4. USD Coin, 5. BNB, 6. Binance USD, 7. XRP, 8. Cardano, 9. Dogecoin y 10. Polygon. En la Figura 3 aparece el listado de criptomonedas ordenadas por capitalización.

#	Nombre	Precio	Cap. de Mercado	#	Nombre	Precio	Cap. de Mercado
1	Bitcoin BTC	\$16,621.53	\$319,327,062,064	6	Binance USD BUSD	\$1.00	\$22,881,281,146
2	Ethereum ETH	\$1,208.46	\$147,883,882,719	7	XRP XRP	\$0.3813	\$19,177,388,269
3	Tether USDT	\$0.9992	\$65,889,418,777	8	Cardano ADA	\$0.3266	\$11,240,496,874
4	USD Coin USDC	\$1.00	\$44,397,248,811	9	Dogecoin DOGE	\$0.08404	\$11,149,397,498
5	BNB BNB	\$271.06	\$43,362,117,250	10	Polygon MATIC	\$0.8691	\$7,591,238,957

Figura 3: Listado de criptomonedas ordenadas por capitalización de mercado el 19-11-2022. Fuente: <https://coinmarketcap.com/es/>.

## 2.1.4 Criptomonedas Populares

Como se ha comentado en el apartado anterior, actualmente existen una serie de activos digitales que dominan el mercado, debido principalmente a su tecnología utilizada y sus aplicaciones. Durante este apartado se van a describir unas de las monedas principales y más utilizadas hasta el momento. Estas son:

1. Bitcoin o **BTC**: es el efectivo electrónico más conocido creado en el año 2008. Consiste en una moneda digital que permite enviar pagos online directamente sin pasar a través de una institución financiera, siendo las firmas digitales una parte de la solución.
2. Ethereum o **ETH**: fue lanzada en 2015 en base a la tecnología utilizada en Bitcoin. Se desarrolló como plataforma de código abierto para construir aplicaciones y organizaciones, tener activos, hacer transacciones y comunicarse sin ser controlada por una autoridad central.
3. Binance Coin o **BNB**: es la moneda oficial del mercado de criptomonedas Binance, el cual se introducirá en el siguiente apartado. Esta moneda nace para soportar transacciones dentro de su propia plataforma.
4. Litecoin o **LTC**: es una moneda digital que permite realizar pagos instantáneos y con un costo casi cero a cualquier parte del mundo. Surge como proyecto de software de código abierto publicado bajo la licencia MIT.
5. **TRX**: es el activo que pertenece a la plataforma digital Tron, la cual tiene como objetivo albergar un sistema de entretenimiento global para compartir contenido digital de manera rentable.
6. XRP Ledger o **XRP**: es un tipo de criptomoneda digital utilizada para transacciones entre pares, con el objetivo principal de ofrecer intercambios de baja comisión.
7. Tether o **USDT**: el Tether consiste en una moneda estable, es decir, que mantiene un tipo de cambio fijo frente a otra criptomoneda, como es el euro o el dólar estadounidense.

Una criptomoneda, al fin y al cabo, es un bien digital, y por ello tiene un valor. De esta forma, para conocer el valor de una moneda digital, es necesaria la cotización de dos activos, uno frente a otro, denominados pares o símbolos. Lo más común es la comparación de un activo digital frente a una moneda estable, como es el USDT. Como son por ejemplo: BTC-USDT, ETH-USDT, BNB-USDT, LTC-USDT, TRX-USDT y XRP-USDT.

Con la gran demanda de esta nueva tecnología y su popularidad, las empresas se han visto con la obligación de evolucionar y utilizar estos activos a su favor.

## 2.2 Aplicaciones de las criptomonedas

En la sección anterior se ha profundizado en los fundamentos de las criptomonedas, pero estas monedas digitales no son solo un concepto teórico o una tecnología sin uso práctico, sino que cada vez se van descubriendo nuevas utilidades, lo que ha generado un creciente interés. Por ello, las empresas empiezan a adoptar estos activos digitales como posibles métodos de pago. Además, surgen nuevas formas de hacer dinero apareciendo

nuevos mercados o plataformas, para la distribución e intercambio de estas monedas.

Debido al aumento en el interés de la población en cuanto a esta nueva forma de dinero electrónico, muchas empresas han empezado a adaptarse y a permitir pagos con Bitcoin u otro tipo de criptomoneda popular. Algunos ejemplos de empresas reconocidas mundialmente que aceptan pagos con criptomonedas según Binance [12] son:

- Overstock: se convirtió en el primer minorista importante en aceptar bitcoin como forma de pago en enero de 2014.
- Paypal: en septiembre de 2014 cuando comenzó a permitir transacciones de bitcoin con empresas enfocadas en criptografía.
- Microsoft: acepta criptomonedas como pago por aplicaciones, juegos y otros productos y servicios en las tiendas de Xbox y Windows en 2014.
- AT&T: en 2019, se convirtió en el primer operador de telefonía móvil importante en aceptar pagos con criptomonedas . Los clientes pueden pagar sus facturas en línea utilizando criptomonedas como opción de pago.
- Starbucks: la cadena internacional de cafeterías acepta bitcoin en su aplicación desde 2020.
- Travala: unas de las mayores webs de viajes, permite pagar reservas de hotel, vuelos, tours y actividades con criptomonedas.

Estas empresas representan únicamente una pequeña parte del total de compañías que actualmente permiten realizar pagos con criptomonedas. No obstante, las criptomonedas son activos con valor, y por ello son necesarias las plataformas en las cuales se pueden realizar intercambios.

### 2.2.1 Plataformas de intercambio de criptomonedas

Como se ha comentado en el apartado anterior, las criptomonedas son un activo con valor. En consecuencia, como con todo bien, es necesario disponer de un mercado para comprar y vender. Para obtener criptomonedas a cambio de dinero fiat (moneda que se usa tradicionalmente para adquirir bienes y servicios, como el euro, el dólar, etc.), es necesario un *exchange* de criptomonedas o plataformas de intercambio de criptomonedas. Estas funcionan como punto de encuentro entre compradores y vendedores donde se realizan los intercambios de dinero fiat a criptomonedas y viceversa.

El objetivo de estas plataformas es permitir al usuario participar en el mercado de las criptomonedas, en el cual es posible obtener ganancias debido a las variaciones de precio en cualquier moneda digital.

Existen dos tipos de mercados dependiendo de cómo se realice la operación de cambio: exchanges centralizados y exchanges descentralizados. El mercado más común actualmente es el centralizado, en el cual se realizan las operaciones de forma muy similar a la de la bolsa de valores internacionales. Dicho de otro modo, las plataformas actúan como intermediarios de las transacciones sin revelar la identidad de los compradores o vendedores, permitiendo una mayor regulación.

En cambio, en las plataformas de intercambio descentralizado, estos se realizan directamente entre los compradores y vendedores. Este tipo de intercambio ha tenido un crecimiento considerable desde el verano del 2020, debido al desarrollo de los contratos inteligentes que permiten realizar operaciones entre dos usuarios de igual a igual [13]. En la Tabla 1 aparecen las plataformas más conocidas y más utilizadas.

Tabla 1: Exchanges más conocidas ordenadas por volumen de 24h el 25/07/2022. Fuente: <https://www.coingecko.com/en/exchanges/>.

#	Exchange	Centralizada o Descentralizada	Volumen 24h	Monedas	Pares
1	Binance	Centralizada	\$12,424,708,590	362	1456
2	OKX	Centralizada	\$1,611,178,225	352	603
3	FTX	Centralizada	\$1,525,165,101	346	530
4	Coinbase Exchange	Centralizada	\$1,360,052,622	205	510

Como se puede observar, los exchanges más utilizados son los mercados centralizados. En los principales se da soporte a menos de 400 monedas, ya que únicamente se aceptan aquellas con una mínima capitalización. Binance es el exchange con mayor volumen diario y alcanzando una cifra de usuarios a finales de 2021 de 28.6 millones [14]. Esta empresa dota a sus usuarios de una gran seguridad a la hora de almacenar los activos digitales y datos personales en su plataforma.

La causa principal de que las plataformas de intercambio comentadas anteriormente (Binance, OKX, etc.) tengan una capitalización tan elevada es el trading, o compraventa, concepto que se definirá en el siguiente apartado.

## 2.3 Trading y Scalping

El trading es una actividad llevada a cabo por la mayor parte de los usuarios con activos digitales que consiste en el acto de comprar y vender criptomonedas con el objetivo de obtener un beneficio. Esta acción ha llevado a muchas personas a dedicarse profesionalmente al mundo de las criptomonedas, siendo esta una actividad que conlleva un gran conocimiento sobre análisis técnico, análisis fundamental y aplicación de estrategias.

Existen dos tipos de trading. El primero se caracteriza por realizar operaciones de compra y venta de monedas con el objetivo de obtener un beneficio comprando cuando el precio está bajo y vendiendo cuando está alto. El segundo tipo se centra en el trading de contratos por diferencia (CFD), lo que permite especular sobre el precio de un activo en ambas direcciones, ya sea comprando (ir en largo) si se espera que el precio suba o vendiendo (ir en corto) si se espera que baje.

Hay muchas maneras de realizar operaciones de trading para buscar obtener beneficios, pero una de las formas más utilizadas por los usuarios experimentados en este tipo de mercado es el **scalping**. Esta técnica consiste en realizar muchas operaciones en

un periodo de tiempo muy corto, con el objetivo de obtener muchos pequeños beneficios. El scalping es una de las formas de trading de criptomonedas más utilizadas por las herramientas de trading automatizadas.

Para realizar todo tipo de tradings es necesario seguir una estrategia o múltiples a la vez. Estas estrategias se pueden dividir en dos grandes categorías: **técnicas**, cuando se analiza el precio observando los datos para predecir precios futuros; y **fundamentales**, cuando se observan los factores financieros y económicos del negocio. En el momento que se desea realizar un análisis del mercado financiero, los traders utilizan un tipo de estrategia o ambas a la vez [15].

El análisis técnico consiste en estudiar los movimientos de las cotizaciones a través de gráficos e indicadores matemáticos basados en los precios. Por ello, muchos de los usuarios se percataron de que la automatización en este mercado era algo muy beneficioso.

### 2.3.1 Automatización del trading

Observar gráficas con indicadores y estrategias técnicas puede ser repetitivo y mecánico para un usuario. Por ello, al tratarse de indicadores calculados mediante fórmulas matemáticas, la automatización permite realizar operaciones sobre diferentes gráficas al mismo tiempo y además, con mayor velocidad.

Actualmente, existen bots de intercambio de criptomonedas desarrollados a partir de software automatizado que ayudan a comprar o vender en el momento más adecuado. Estos programas siguen estrategias e indicadores técnicos para predecir futuros precios.

#### 2.3.1.1 Funcionamiento

La automatización mediante bots consiste en el análisis en profundidad de los datos sobre el mercado de las criptomonedas. Estos bots son los que se encargan de tomar las decisiones de comprar o vender en torno a las fluctuaciones del mercado según la estrategia o la configuración que se utilice.

En la mayoría de las herramientas del mercado, los bots son configurables con las especificaciones del usuario, utilizando una estrategia específica, unos parámetros, un intervalo, una moneda, etc.

Entonces, una vez configurado el bot con la información que el usuario desea, este se va a encargar de interactuar directamente con las plataformas de intercambio de criptomonedas, realizando compras o ventas dependiendo de las condiciones e indicadores técnicos que se utilicen. Posteriormente, se comentarán algunas de las diferentes opciones que se pueden encontrar en el mercado en cuanto a aplicaciones de trading automático.

A continuación, se analizarán las ventajas y desventajas que conlleva el hecho de realizar la compraventa mediante herramientas de trading automatizadas.

### 2.3.1.2 Ventajas y desventajas

La automatización en el mercado de las criptomonedas tiene puntos fuertes y débiles los cuales el usuario debe conocer y valorar antes de utilizar este tipo de herramienta. Vamos a comentar las ventajas y desventajas principales del trading de criptomonedas con bots según Nunes [16]:

Ventajas:

- **Ahorro de tiempo.** La automatización permite realizar acciones de forma paralela, sin necesidad de atender 24 horas al día a los cambios del mercado.
- **Reduce decisiones emocionales.** Al ser un mercado tan volátil, es arriesgado lidiar con los apegos emocionales. Entonces, al ser un bot de trading el que toma decisiones automáticas siguiendo un modelo matemático, el aspecto emocional no influye.
- **Aumenta la eficiencia.** Permite realizar procesos más eficientes debido a la posibilidad de realizar funcionalidades multitarea, como analizar múltiples criptomonedas a la vez.
- **Simplifica el trading.** El mercado de las criptomonedas es complicado de entender para gente nueva en el trading. En este sentido, un bot programado evita que el usuario tenga que comprender las gráficas y los datos de los indicadores, lo cual simplifica la inversión.
- **Tests.** Al ser un programa automatizado, este permite realizar tests 24 horas al día para, a partir de los resultados obtenidos, configurar de manera más óptima el bot.

Desventajas:

- **Fallos en el bot.** Al ser un programa informático, es posible que este presente problemas imprevisibles que podrían ser arriesgados para el usuario al estar invirtiendo dinero digital.
- **Decisiones imprevistas.** Los bots trabajan automáticamente las 24 horas del día y es posible que este tome decisiones equivocadas, provocando que puedan desencadenarse eventos inesperados.
- **La probabilidad de utilizar bots fraudulentos.** Es posible encontrarse con algún bot en determinadas páginas web que puedan robar información personal e incluso robar todos los fondos de las carteras digitales.

### 2.3.1.3 Mercado

Debido a la cantidad de usuarios que se ha unido a los mercados de criptomonedas los últimos años y la cantidad de dinero que mueve y genera esta actividad, el mercado de los bots de trading es cada vez más amplio lo que supone una mayor competencia.

En este sentido, en el mercado de la automatización, al ser tan popular, han aparecido diferentes formas del trading automatizado. Vamos a diferenciar entre dos categorías, bots de *exchanges* y aplicaciones de automatización externas.

Los bots de exchanges consisten en software implementado en los grandes exchanges, como Binance, Kucoin, Bybit, etc., enfocado al trading automático sin tener que hacer ningún tipo de esfuerzo. Es un tipo de bot dirigido a principiantes con pocas opciones configurables y sencillo de utilizar.

Por otra parte, las aplicaciones externas de trading automatizado utilizan las apis y claves de exchanges centralizados para poder realizar operaciones. Este tipo de aplicaciones están enfocadas principalmente a usuarios con un mayor conocimiento; permitiendo configurar bots a tu elección, con indicadores y estrategias que el usuario seleccione. En el siguiente apartado, se muestran algunos ejemplos de aplicaciones externas de trading automático, para poder comparar sus funcionalidades entre ellas.

#### 2.3.1.4 Aplicaciones de trading automatizado

El mercado del software o aplicaciones para realizar trading de criptomonedas automatizado es cada vez más grande, y con ello los usuarios disponen de más opciones para elegir. De este modo, es importante destacar las funcionalidades disponibles.

A pesar de que la mayoría de las aplicaciones de trading disponen de una base general y similar en la cual se permite realizar operaciones de forma automática, cada software tiene diferentes funcionalidades y herramientas que le hacen destacar en el mercado. A continuación se presentan tres ejemplos de aplicaciones disponibles actualmente. Estas son:

- **Quance:** Es una compañía de software enfocada al mercado de activos digitales, especializada en trading algorítmico automatizado [17]. La aplicación consiste en una web con un sistema de automatización similar a un gestor de fondos tradicional, la cual tiene como objetivo configurar carteras de forma dinámica para adaptarse a la situación del mercado.
- **Tuned:** Es una empresa de software que administra las carteras de los activos digitales utilizando un sistema de comercio automatizado mediante el trading con estrategias o mediante algoritmos [18]. Ofrece a los usuarios dos productos diferentes: una aplicación móvil sencilla para inversores en la cual puedes automatizar sencillamente con algoritmos ya configurados y una página web avanzada para poder realizar trading o tests mediante estrategias configurables, e incluso la creación de estas mediante programación.
- **Haasonline:** Empresa de software de soluciones de comercio automatizado para criptomonedas [19]. Proporciona a los usuarios dos aplicaciones, una de ellas ejecutada de forma local, y otra de ellas, en la nube. Ambas aplicaciones ejecutan automatización de trading mediante estrategias totalmente configurables. Además, permiten la creación de estrategias mediante scripts.

Estas tres herramientas están enfocadas al trading automático del mercado de criptomonedas, pero a pesar de ello, se diferencian en algunas de sus características. Por este motivo, se ha decidido realizar un análisis de ventajas y desventajas de cada una de

ellas, con el fin de obtener una visión clara de las mejores funcionalidades. En la Tabla 2 se muestran los resultados de este análisis.

Tabla 2: Pros y contras de los competidores de aplicaciones de trading automatizado.

	Quance	Tuned	Haasonline
<b>Ventajas</b>	<p>No requiere conocimiento sobre trading avanzado</p> <p>Sencillo de utilizar</p>	<p>Aplicación web y móvil</p> <p>Ejecución de la aplicación en la nube</p> <p><b>Creación de estrategias mediante scripts</b></p> <p><b>Ejecución de múltiples bots al mismo momento</b></p> <p>Elección entre automatización sencilla o avanzada</p> <p><b>Visualización de los trades realizados y gráficas en tiempo real</b></p> <p><b>Posibilidad de realizar tests</b></p>	<p><b>Ejecución de la aplicación en la nube</b></p> <p><b>Ejecución de la aplicación en local</b></p> <p><b>Creación de estrategias mediante scripts</b></p> <p><b>Ejecución de múltiples bots al mismo momento</b></p> <p><b>Visualización de los trades realizados y gráficas en tiempo real</b></p> <p><b>Posibilidad de realizar tests</b></p>
<b>Desventajas</b>	<p>Únicamente 3 algoritmos disponibles</p> <p>Muy poco configurable</p> <p>No ofrece la posibilidad de visualizar los trades realizados y las gráficas</p>	<p>Aplicaciones separadas</p> <p>No es posible visualizar múltiples gráficas al mismo instante</p>	<p>Requiere conocimiento sobre trading avanzado</p> <p>Aplicación compleja</p> <p>No es posible visualizar múltiples gráficas al mismo instante</p>

Una vez analizados las ventajas y desventajas de cada una de las aplicaciones, se puede determinar que sería muy interesante disponer de una herramienta que ofrezca las funcionalidades siguientes:

1. Ejecución de forma local o en la nube.
2. Ejecución de múltiples bots simultáneamente.
3. Posibilidad de configurar los bots mediante diferentes estrategias y valores.
4. Visualización de las gráficas a tiempo real.
5. Visualización de los trades/operaciones realizados al instante.
6. Posibilidad de añadir estrategias totalmente personalizadas mediante programación.
7. Realización de tests con las estrategias.

Tras esta conclusión, a continuación se presenta el desarrollo del proyecto que da cabida a estas necesidades.

## Capítulo III

# Desarrollo del proyecto

En el capítulo anterior se ha realizado una investigación sobre las criptomonedas, sus posibles aplicaciones y el trading. Al obtener información respecto a esta acción, se ha indagado sobre su posible automatización, surgiendo así la idea del proyecto realizado. Esta idea se desarrolla a continuación y junto a ello, todos los pasos que se han seguido hasta llegar a la producción de **una herramienta para la simulación y evaluación de estrategias de inversión en mercados de criptomonedas**.

### 3.1 Metodología

#### 3.1.1 Metodología de trabajo

Durante la realización del proyecto se ha utilizado la metodología de trabajo *Waterfall*, permitiendo desarrollar el software en distintas fases. En este modelo, cada fase empieza cuando ha terminado la anterior. De esta forma, al desarrollar un proyecto de manera individual, todas las ideas deben quedar claras antes de pasar a la siguiente etapa. En el siguiente apartado, se mostrará la hoja de ruta con los pasos que se han seguido durante la construcción del proyecto.

#### 3.1.2 Roadmap

Como se ha mencionado en el anterior apartado, antes de empezar a desarrollar la aplicación, era necesario tener claro todos los pasos que se iban a realizar durante el proceso. Por ello, se ha elaborado un *Roadmap*, u hoja de ruta. Esto consiste en una guía objetiva que señala los caminos que se deben tomar para lanzar el nuevo producto. Los diferentes pasos seguidos se pueden visualizar en la Figura 4 y son comentados a continuación.

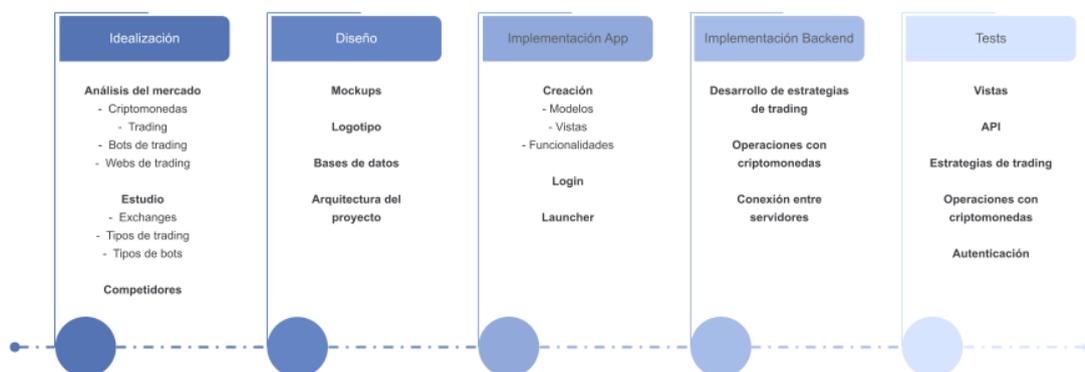


Figura 4: Roadmap.

El primer paso para realizar el proyecto era **concebir la idea** y realizar un análisis del mercado con el propósito de obtener un planteamiento sólido. Por ello, era necesario llevar a cabo un estudio de aquello en lo que iba a estar involucrada la aplicación, siendo en este caso las criptomonedas, los exchanges, el trading y los bots. No obstante, no podía faltar el análisis de la situación de los competidores, haciendo referencia a aquellas herramientas de trading automatizado presentes en el mercado.

A continuación, se realizó el **diseño** del proyecto, incluyendo la parte visual como son los mockups, con las vistas y los modelos. Después, se diseñó la arquitectura del proyecto, en cuanto a cómo se iba a dividir y cómo se iban a obtener los datos en cada caso. Por último, se elaboró la estructura de la base de datos. De este modo, con la realización de estos pasos con antelación, se podrían anticipar los posibles problemas o errores futuros.

Una vez realizado el diseño, se comenzaron a **implementar las vistas** en ambas aplicaciones, las cuales serían el elemento que iniciaría los procesos posteriores, por este motivo es lo primero que se debía desarrollar. De igual manera, para las aplicaciones, sería necesaria la implementación de un sistema de autenticación. Por ello durante este proceso, se añadió un login para el usuario. Una vez finalizado el desarrollo de la parte visual de la aplicación, se implementó todo el sistema para el procesamiento de datos.

La **implementación del backend** consiste principalmente en el sistema de trading automatizado, y para el correcto funcionamiento de este, era necesaria una conexión entre servidores, un sistema de trading mediante estrategias y la posibilidad de realizar operaciones con criptomonedas.

Finalmente, la última fase consistía en el **periodo de pruebas**, en la cual se verifica al completo el correcto funcionamiento de la aplicación y sus diferentes funcionalidades. De este modo, se han realizado tests de trading a tiempo real y simulaciones con históricos de datos, ambos con diferentes estrategias y diferentes configuraciones.

Una vez teníamos claro el proceso que se iba a seguir para desarrollar el proyecto, el siguiente paso consistía en realizar un análisis de los requisitos que había de tener la aplicación. De esta forma, antes de empezar a implementar el diseño, la idea estaba totalmente concebida y esclarecida.

### 3.1.3 Control de versiones

Para mantener un registro de los cambios realizados durante la elaboración del proyecto, se ha utilizado un control de versiones. Este sistema te permite regresar a versiones anteriores de los archivos, pudiendo sustituir el código por el antiguo y comparar cambios ya realizados.

El control de versiones que se ha utilizado en este proyecto es Git, utilizando la rama *main/develop* como la de desarrollo diario, pudiendo tener versiones no funcionales, y la rama *master* para las versiones funcionales.

Para almacenar estas versiones y poder trabajar desde varios dispositivos, se ha utilizado una plataforma de alojamiento de código, *GitHub* [20]. Pudiendo almacenar los proyectos de forma segura, visualizar los repositorios de cada uno de ellos, los commits realizados y las ramas de forma más visual.

Se han utilizado 2 repositorios diferentes para cada una de las aplicaciones desarrolladas en el proyecto, con la intención de mantener un registro individual en cada una de ellas.

Aplicación web: [https://github.com/SergioPujol/TrabajoFinGrado\\_CandIV-Web](https://github.com/SergioPujol/TrabajoFinGrado_CandIV-Web)

Aplicación de escritorio: [https://github.com/SergioPujol/TrabajoFinGrado\\_CandIV-App](https://github.com/SergioPujol/TrabajoFinGrado_CandIV-App)

## 3.2 Análisis de requisitos

Tal y como se ha comentado anteriormente, para tener la idea del proyecto que se va a llevar a cabo, es importante analizar previamente los requisitos que se desean implementar. De esta forma, se pueden realizar los diseños y la arquitectura del proyecto acorde a las funcionalidades que se desean añadir. Por ello, después del estudio presentado en el Capítulo 1, se realizó el siguiente listado de todos los requisitos necesarios para la herramienta.

1. Autenticación de usuario.
2. Creación de gráficas con opciones de símbolo e intervalo.
3. Visualización de múltiples gráficas al mismo momento.
4. Creación de bots de trading automatizados configurables con diferentes estrategias y valores.
5. Posibilidad de elección entre inversión fija o inversión porcentual en cada bot.
6. Configuración de claves de Binance desde la aplicación.
7. Visualización de listado de operaciones ya realizadas.
8. Visualización de operaciones corrientes en la aplicación.
9. Empezar o parar operaciones de forma manual desde la aplicación.
10. Almacenamiento de datos respecto a las gráficas, bots, estrategias por usuario.
11. Ejecución de la automatización en la nube.
12. Ejecución de la automatización en local.
13. Implementación de sistema de análisis técnico mediante estrategias.
14. Creación de estrategias mediante scripts.
15. Ejecución de múltiples bots al mismo momento .
16. Realización de operaciones mediante API de Binance.
17. Realizar operaciones con red de pruebas de Binance.
18. Realizar operaciones con red principal de Binance.
19. Simulación de trading con estrategias utilizando históricos de datos mediante diferentes símbolos e intervalos.
20. Cálculo ficticio del beneficio tras una simulación.

Realizado el análisis de los requisitos de la aplicación, se definieron las tecnologías que se iban a utilizar durante su desarrollo, las cuales se describen en el siguiente apartado.

### 3.3 Tecnologías de la aplicación

Para el desarrollo del proyecto se hace uso de diferentes tecnologías, las cuales han hecho posible la creación de la aplicación. A continuación, se enumeran cada una de ellas y su respectiva función durante la implementación.

1. **Javascript.** Es un lenguaje de programación interpretado, utilizado para secuencia de comandos para páginas web [21]. Nos permite añadir características interactivas a las vistas de la aplicación.
2. **HTML.** Es un lenguaje de etiquetas de hipertexto. Define el significado y la estructura del contenido web. [22]
3. **CSS.** Es un lenguaje de estilos, utilizado para describir la presentación de documentos HTML. Permite modificar el estilo del contenido de la página web.
4. **Bootstrap.** Consiste en una biblioteca multiplataforma de código abierto para diseños de aplicaciones web. Permite construir y customizar con Sass, utilizando cuadrículas y compuestos prediseñados [23].
5. **Electron.** Es un framework de código abierto que funciona para crear aplicaciones multiplataforma con tecnologías web como Javascript, HTML y CSS [24]. Permite visualizar la web mediante un launcher de escritorio.
6. **Nodejs.** Es un entorno en tiempo de ejecución multiplataforma basado en el lenguaje de programación Javascript [25]. Permite desarrollar el servidor web y los servidores de la base de datos.
7. **Nodejs.** Es un motor de ejecución de TypeScript (lenguaje de programación tipado que se basa en Javascript) y REPL, o bucle de Lectura-Evaluación-Impresión, para Nodejs [30]. Utilizado para desarrollar el sistema de procesamiento de datos.
8. **MongoDB.** Es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, así como un modelo de consultas e indexación avanzado [26]. Utilizado como base de datos NoSQL en el proyecto.
9. **Mongoose.** Es una biblioteca de programación orientada a objetos de Javascript que crea una conexión entre MongoDB y el entorno de Nodejs [27]. Su principal uso consiste en la definición de esquemas, normalizando la información de MongoDB.

Una vez claros todos los objetivos y funcionalidades que la herramienta debía cumplir, junto a la tecnología que se va a utilizar, se realizó el diseño para unificar todos los conceptos y crear una planificación del proyecto antes de su implementación.

### 3.4 Diseño

En función de los requisitos y tecnologías definidos en los apartados anteriores, se decidió realizar dos aplicaciones: una aplicación web ejecutada en la nube y una aplicación de escritorio ejecutada de forma local. De este modo, el usuario podría elegir qué aplicación utilizar, dependiendo de sus necesidades.

Con la aplicación web, el usuario podría acceder a ella desde cualquier parte, ya que un servidor externo sería el encargado de almacenar la información de todos los usuarios, y

de procesar los datos y operaciones. De esta forma, la automatización estaría siempre en funcionamiento, y el usuario podría dejar bots ejecutándose sin la necesidad de tener un dispositivo encendido.

Para la aplicación de escritorio, el usuario tendría la posibilidad de ejecutar un launcher de forma local, realizando la automatización y todas las operaciones en el mismo dispositivo. De igual manera, el almacenamiento de los datos se efectuaría localmente. Consecuentemente, se mejoró el rendimiento del procesamiento automático de los datos. Además, al ejecutar la aplicación en un dispositivo local, el usuario tiene la opción de implementar estrategias mediante código typescript y realizar simulación con históricos de datos.

Una vez claro el proyecto que se iba a desarrollar, a continuación se muestra la arquitectura del proyecto, los diseños de las bases de datos y por último, los mockups.

### 3.4.1 Arquitectura del proyecto

El primer paso del diseño del producto fue realizar una planificación de la arquitectura general. De esta manera, es posible entender cómo va a funcionar el software de la aplicación al completo. Durante el proyecto, como se muestra en la Figura 5, se diferencia entre diferentes sistemas: Aplicación de escritorio / web, Servidor web, Servidor de autenticación, Servidor de base de datos y Servidor de procesamiento.

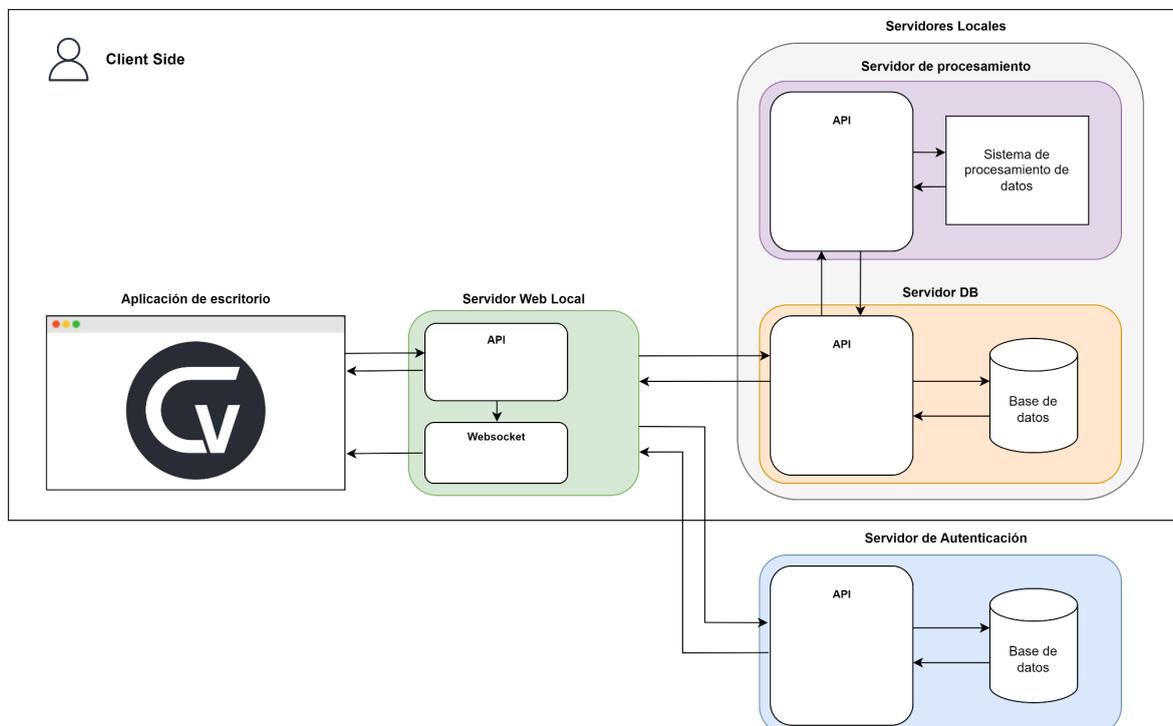


Figura 5: Arquitectura del proyecto de aplicación de escritorio.

Primero de todo, se encuentra la **visualización de la aplicación**, que como se ha comentado anteriormente, se lleva a cabo mediante una página web, o un launcher de escritorio.

- La **aplicación web**, carga las siguientes vistas en un navegador web: la página inicio con la autenticación y la página de trading.
- La **aplicación de escritorio** consiste en un launcher donde carga la url de la web ejecutada de forma local. Esta tiene las dos vistas de la página web y una más para la simulación de histórico de datos.

En ambos casos hay un **servidor web** que funciona de la misma manera, el cual se encarga de alojar la web y las vistas de la aplicación. Este servidor tiene una API y un websocket. La función de la API consiste principalmente en la comunicación entre la web y el servidor, donde a partir de la web se envían instrucciones para obtener o crear gráficas y bots en la base de datos, iniciar o parar una operación, etc. Es decir, todas aquellas funcionalidades posibles desde la web.

La función principal del websocket en este servidor web es mantener un canal de comunicación full-duplex a través de una conexión TCP, permitiendo una comunicación en ambos sentidos y de forma simultánea [29]. Este websocket está conectado también entre el cliente y el servidor web, pero se utiliza para enviar información sobre trades realizados, la actualización continua de las operaciones en curso y los errores por parte de los servidores.

Los servidores de procesamiento de datos y base de datos se distribuyen de diferentes maneras en las dos aplicaciones.

Por una parte, en la **aplicación de escritorio**, se encuentra un **servidor de autenticación**, el cual es un servidor remoto con una base de datos, utilizado para verificar la autenticación de los usuarios en su aplicación de escritorio. A este servidor se le hacen peticiones por parte de todos los clientes. Por otra parte, se encuentran los **servidores alojados localmente** en cada uno de los dispositivos de los clientes. La idea principal de establecer estos servidores de forma local es agilizar el análisis de los datos y el almacenamiento y obtención de estos. El esquema en cuestión se muestra en la Figura 5.

Y en el caso de la **aplicación web**, se gestiona mediante **servidores en la nube**. Por ello, la información de todos los usuarios se almacena en una misma base de datos. De esta manera se junta lo que en el diagrama anterior era, por una parte, el servidor de base de datos y, por otra, el servidor de autenticación. Finalmente, se encuentra el servidor de procesamiento, el cual procesa los datos respecto al trading de todos los usuarios desde el mismo sistema. La arquitectura de la página web se puede encontrar en el diagrama de la Figura 6.

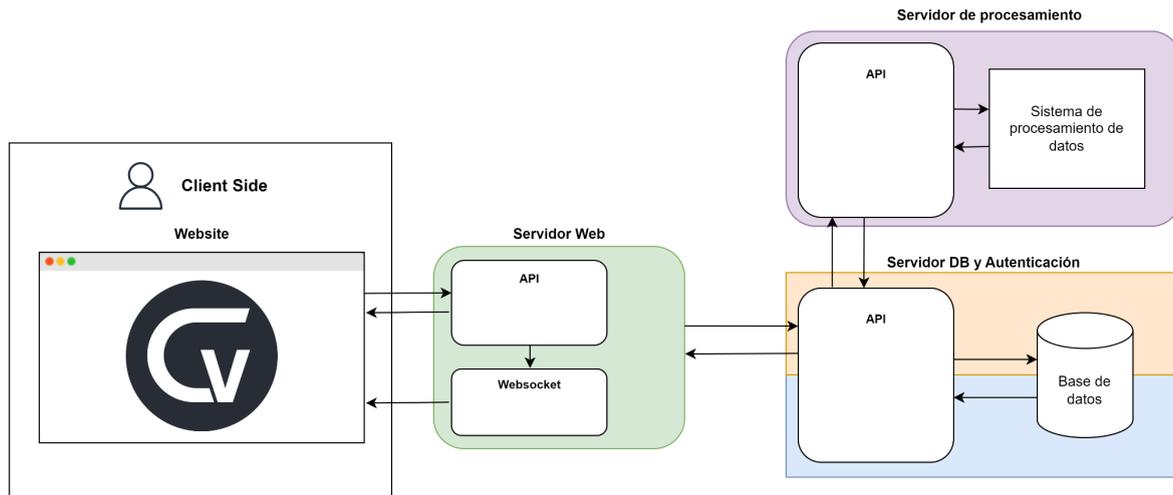


Figura 6: Arquitectura del proyecto de página web.

Aunque estos servidores se encuentren alojados localmente o en un servidor externo, realizan las mismas funciones.

El servidor de Base de datos, tiene como objetivos principales el almacenar la información y funcionar como comunicador entre el Servidor Web y el Servidor de procesamiento de datos. Este servidor de base de datos almacena las diferentes gráficas, bots, trades, estrategias personalizadas y la configuración de la aplicación. En el caso del proyecto de la aplicación web, la base de datos de información y autenticación es común, por ello, también se almacenan los datos de autenticación de usuarios.

Este servidor mantiene una comunicación con el servidor de procesamiento, indicando cuándo crear bots y cuándo eliminarlos, de la misma forma que cuando iniciar operaciones o detenerlas.

En definitiva, este último servidor de procesamiento de datos es el encargado de realizar todas las operaciones y análisis de los datos. En este sistema se ejecutarán las conexiones con la API de Binance, tanto para obtener información de criptomonedas, como también para realizar trades de compra o venta [30]. Este servidor está desarrollado en Node Typescript.

### 3.4.2 Base de datos

Tal y como se ha planteado en la arquitectura del proyecto, hay dos tipos de estructuras dependiendo de la aplicación. En el caso de la aplicación de escritorio, serán necesarias dos bases de datos, una de ellas alojada en un servidor en la nube y utilizada para la autenticación de todos los usuarios, y otra alojada en un servidor local y utilizada para almacenar la información respecto a las gráficas, bots, trades, etc. Al ser un servidor local, los datos únicamente pertenecen al usuario que inicie la aplicación en cuestión.

Por otra parte, para la aplicación web, únicamente se requiere un servidor de base de datos remoto, donde la información de todos los usuarios respecto a la autenticación y las gráficas, bots, etc. se encuentra almacenada de forma común.

Se ha utilizado una base de datos no relacional ya que estas ofrecen un rendimiento y escalabilidad superiores en comparación con las bases de datos relacionales, lo que es esencial para el almacenamiento de bots, gráficas y trades.

En el caso de la arquitectura de la aplicación de escritorio, las bases de datos se distribuyen de la siguiente manera:

La base de datos de autenticación tiene un solo esquema, el de User. Este únicamente dispone de 3 atributos los cuales aparecen en la Figura 7: la id, un email y una key. La id, la key primaria de cada usuario, es asignada automáticamente con Mongoose. Y también dispone de un email y una key, donde ambas son strings utilizadas para la autenticación.

### Public Database

User	
id	PK - mongoose.Schema.ObjectId
email	String
key	String

Figura 7: Diseño base de datos de autenticación de aplicación de escritorio.

Además, hay una base de datos local encargada de almacenar colecciones respecto al uso de la aplicación. Es decir, todo lo relacionado con su configuración, las gráficas creadas, los bots, las operaciones y las estrategias. El diseño de esta base de datos local se muestra en la Figura 8.

### Local Database of each User

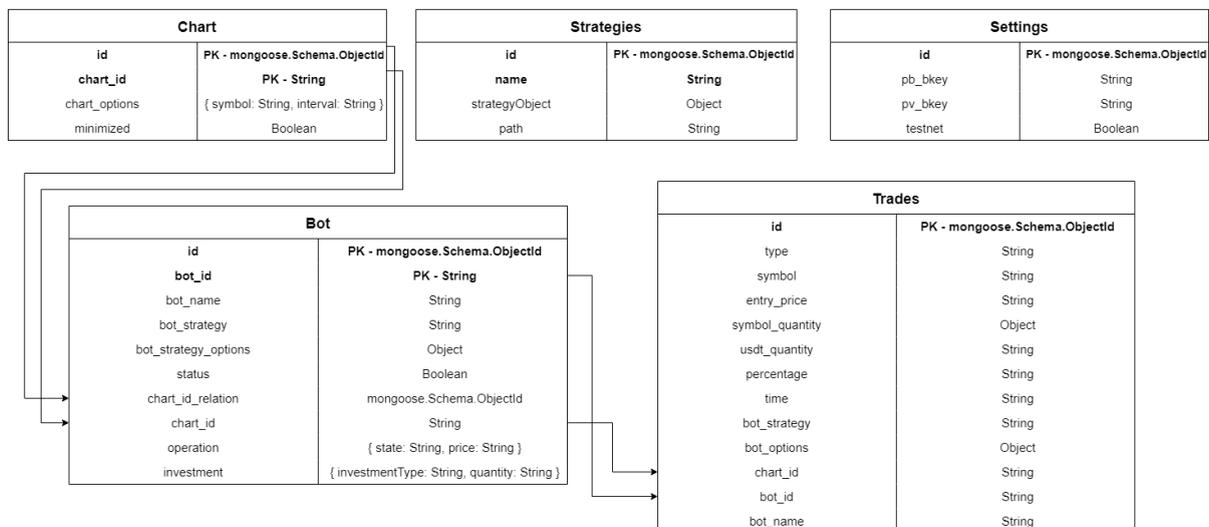


Figura 8: Diseño base de datos local de aplicación de escritorio.

En primer lugar están los “settings” donde se almacenan las keys de binance para realizar trades y un boolean para indicar si se realizan las operaciones en la red de pruebas o la red principal de Binance. Las keys, se almacenan encriptadas mediante una cifrado AES.

A continuación, tenemos las gráficas o “charts” con sus diferentes opciones. Después se encuentran los bots, cada uno relacionado con una chart. Así pues, una chart puede tener más de 1 bot asignado. La colección de “bot” contiene la información de las estrategias e inversiones, al igual que la operación actual.

Por último, quedarían los trades y las estrategias. La colección de trades, almacena cada una de las operaciones realizadas con su información al respecto, con el símbolo utilizado, la cantidad invertida, la resultante, el porcentaje, etc. Cada trade pertenece a un bot, y este, a una gráfica. Finalmente, tenemos las estrategias personalizadas, donde cada una de ellas tiene un nombre, un objeto con la información a completar a la hora de crear un bot y un path, que es la localización del archivo typescript en el dispositivo.

En este caso, la base de datos local no tiene ninguna relación con la id del usuario en cuestión, ya que está almacenada en el ordenador del cliente, accesible únicamente desde ese dispositivo. En cambio, el diagrama de la base de datos para la aplicación web, el cual se puede observar en la Figura 9, es diferente, ya que existe una relación entre las id de los usuarios para poder acceder a la información correcta.

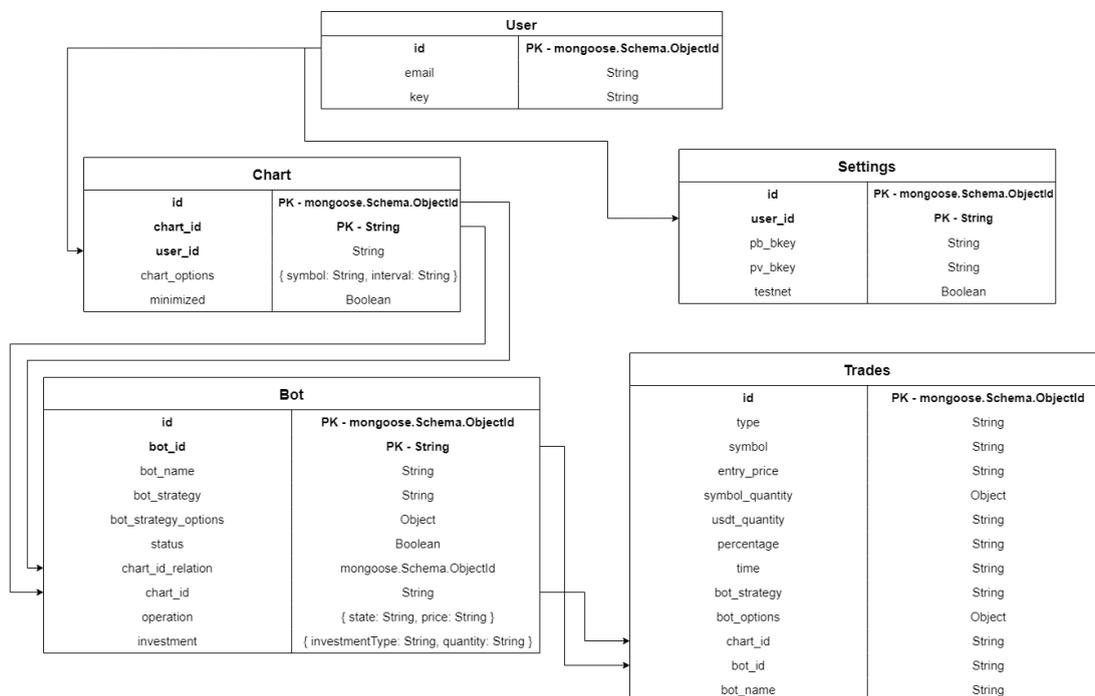


Figura 9: Diseño base de datos de aplicación web.

La diferencia que existe en cuanto al anterior esquema es la unificación de bases de datos. De esta forma, cada “Chart” y cada “Settings” está relacionada con un usuario, por ello se le añade un atributo de “user\_id”. Entonces, cada usuario tendrá sus propias gráficas, en la que cada gráfica tendrá sus bots, y cada bot, tendrá sus trades.

### 3.4.3 Logotipo y Mockups

Como último apartado de diseño, se encuentra la realización de los mockups, que consiste en la creación de modelos de la aplicación: la comunicación entre páginas y diferentes opciones y también, el diseño del logotipo del proyecto. Para la realización de los mockups se ha utilizado *Figma* y para el diseño del logotipo, *Photoshop*.

Para facilitar el proceso de desarrollo de la aplicación, es importante tener los mockups de todas las vistas disponibles. Antes de comenzar, es crucial elegir una paleta de colores coherente con la idea de la aplicación. Muchas aplicaciones relacionadas con el trading y las criptomonedas utilizan colores oscuros y azules. Por lo tanto, como se muestra en la Figura 10, para este proyecto se han seleccionado colores oscuros para el fondo, y azul claro y blanco para los detalles importantes.

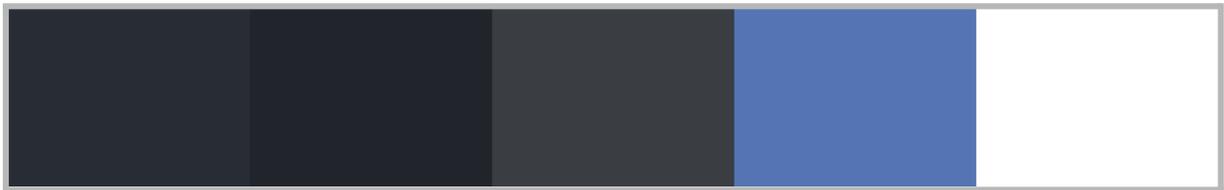


Figura 10: Paleta de colores.

Una vez clara la paleta de colores que se va a utilizar, es importante diseñar el logotipo. El nombre de la aplicación es CandIV, una abreviación a Candle View, que hace referencia a la visión de las candles o velas. El gráfico de candles es el tipo de gráfico utilizado en el análisis técnico [31]. De esta forma, el logotipo representa sus iniciales principales y se muestra en la Figura 11.



Figura 11: Logotipo.

En la Figura 12 se muestra el diagrama de navegación de la aplicación al completo. En este se muestra la arquitectura de la web, con sus diferentes vistas y las posibles interacciones de cada página.

Primero de todo, se muestra el inicio de la aplicación para la autenticación, en la cual, se introduce el email y la clave, y seguidamente se clicca sobre “Authenticate”. Si este acceso es válido, te enviará a la página principal de la aplicación, la vista de Trading.



Figura 12: Diagrama de navegación página inicio.

En la Figura 13 y en el anexo “Diagrama de navegación”, se muestra el diagrama de navegación de la aplicación al completo con las funcionalidades principales. En esta, se incluyen las vistas de la aplicación, la de trading y la de simulación. En el caso de la página web, la vista de simulación no forma parte de la aplicación. A continuación, se explicaran uno a uno los componentes y sus funcionalidades.

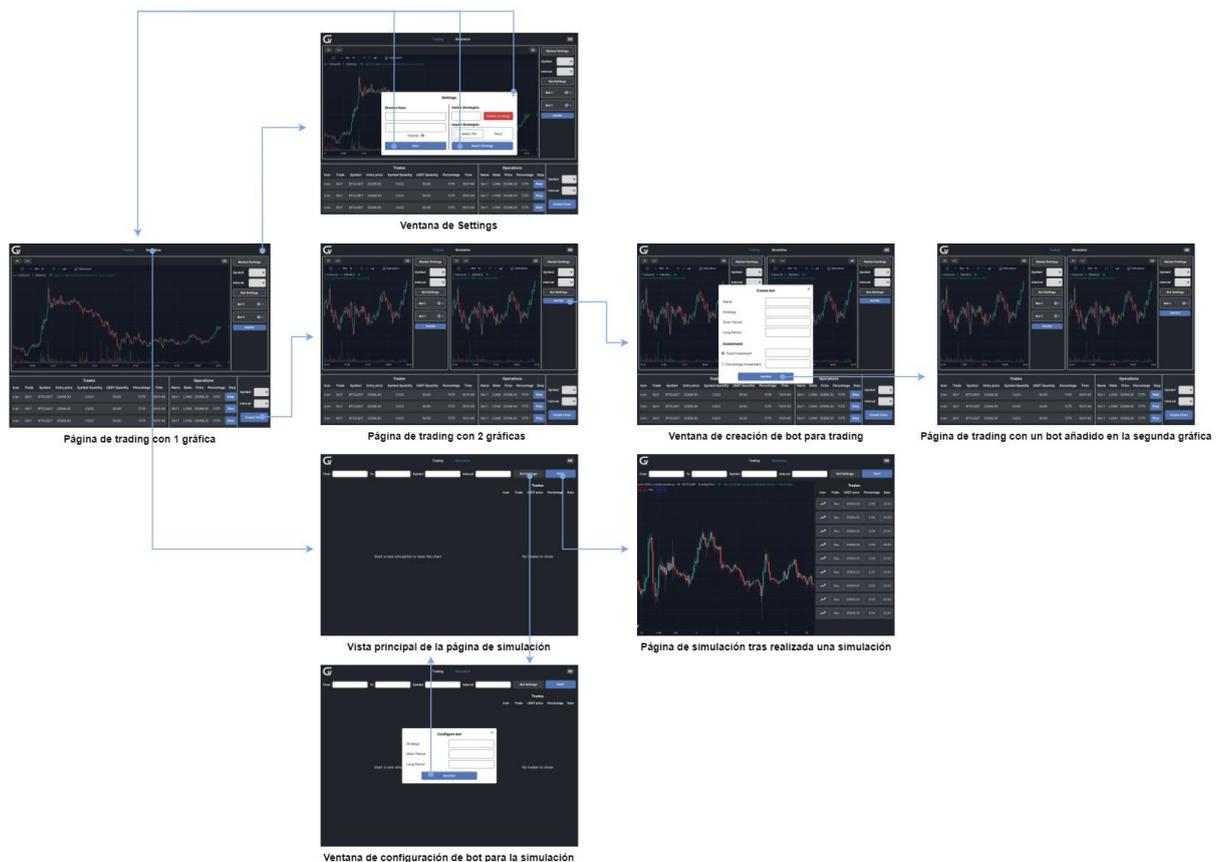


Figura 13: Diagrama de navegación.

El navbar o barra de navegación aparece en la parte superior de la página, y su diseño se puede visualizar en la Figura 14. El componente muestra a la izquierda el logotipo de la aplicación. Después, en el centro, se muestran los textos de Trading y Simulation, al clicar sobre cada uno de ellos estos redireccionan al usuario a sus respectivas páginas (aquella que sea seleccionada aparecerá de color azul). A la derecha, se muestra un botón con un emoticono de configuración. Al clicar sobre este, aparecerá un popup con las opciones de guardar las claves de Binance, seleccionar entre *Mainnet* o *Testnet* y gestionar las estrategias. En la Barra de navegación de la página web únicamente aparecerá la opción Trading.



Figura 14: Barra de navegación superior.

La página principal de Trading, que se muestra en la Figura 15, está compuesta por las gráficas o charts, donde cada una tiene las opciones de: eliminar, en el botón de la izquierda, al clicar sobre este aparecerá un popup para confirmar si se desea eliminar o no; minimizar, botón el cual minimizará la gráfica para que no ocupe espacio y aparecerá una pequeña pestaña en la parte inferior para volverla a maximizar o eliminar; y finalmente, el tercer botón con icono de hamburger menu, te permite mostrar/ocultar el menú de la gráfica.

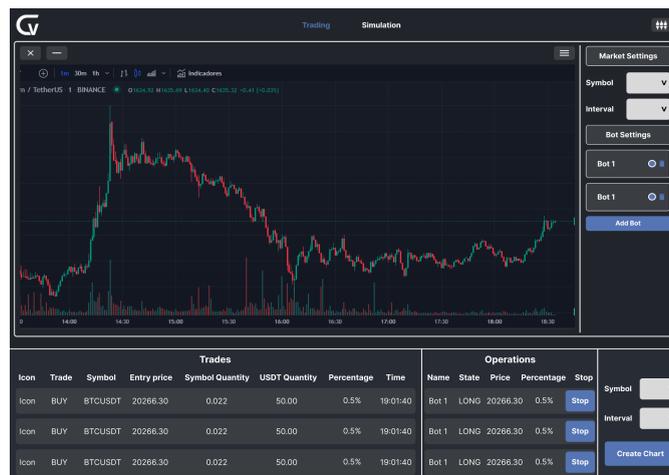


Figura 15: Vista principal de Trading.

En la parte derecha de la gráfica se encuentra su configuración, con la sección de los "Market Settings", donde se puede cambiar el símbolo y el intervalo utilizado en la gráfica. Y en la sección "Bot Settings" se encuentran los ajustes de los bots, donde se pueden agregar, modificar y eliminar bots. Una vez agregados, cada bot tiene la opción de ser modificado.

En la parte inferior de la pantalla, aparecen a la izquierda los trades. Este apartado consiste en un historial de operaciones ya realizadas con la siguiente información: tipo de operación ("BUY" o "SELL"), el símbolo de la operación, el precio de compra del símbolo en USDT, la cantidad de símbolo adquirido o vendido, la cantidad en USDT, el porcentaje, y finalmente la fecha en la que se ha realizado la operación.

En operations se muestran las operaciones en curso con cada bot, incluyendo su estado (si se ha comprado o si se recomienda esperar para realizar la operación), el precio de la operación en la que se inició, el porcentaje en tiempo real de la operación y la opción de detener la operación en cualquier momento, y en el caso de no haber ninguna operación en marcha, iniciarla. Por último, aparece la opción de crear más gráficas, introduciendo un símbolo y un intervalo, y posteriormente clicando sobre el botón “Create Chart” se agrega una gráfica a la derecha.

Cuando se hace clic sobre el botón de la parte superior, se emerge un popup con opciones. Estas opciones **no son las mismas** en la aplicación de escritorio que en la página web. En el caso del launcher, las opciones están divididas **en dos apartados**. A la izquierda se pueden rellenar los valores de las claves de Binance y a la derecha se pueden gestionar las estrategias personalizadas del usuario, tanto para importar más estrategias como para eliminarlas. En cuanto a la página web, únicamente aparece la opción de Binance Keys. Ambos popups se pueden visualizar en la Figura 16.

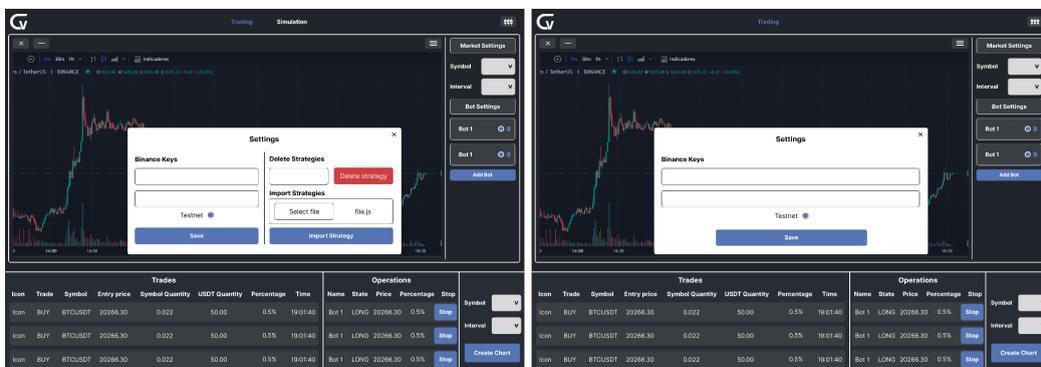


Figura 16: Vista de Trading con popup de settings en aplicación de escritorio y web.

Cuando se hace clic sobre “Create Chart” con las opciones seleccionadas, la página se muestra tal y como aparece en la Figura 17. Entonces, cada una de las gráficas tiene sus propios datos y sus propios bots, en el caso de que se quieran añadir. Una vez una de ellas se elimine o se minimice, la otra queda al completo, utilizando al máximo el espacio de la página. Si se desean añadir más gráficas, el tamaño queda dividido entre la cantidad de gráficas totales.

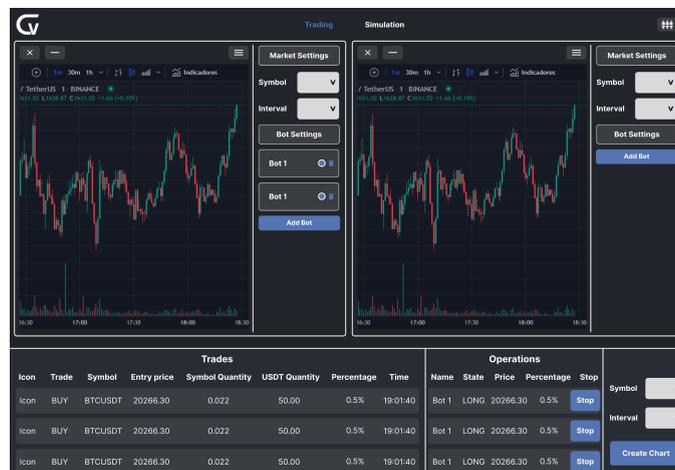


Figura 17: Vista de Trading con dos gráficas.

Cuando en una gráfica se desea añadir un bot, al clicar sobre el botón de “Add Bot” emerge un popup con las posibles opciones, donde se podrán completar los valores de name, strategy y las opciones características de cada estrategia. Después, se tendrá que seleccionar entre uno de los dos tipos de inversiones en el bot: inversión fija, para utilizar una cantidad fija de dólares, o inversión porcentual, el cual utilizará el porcentaje indicado de dólares del portafolio del usuario. En la Figura 18 se puede observar la vista de la página Trading con la ventana emergente para crear un bot.

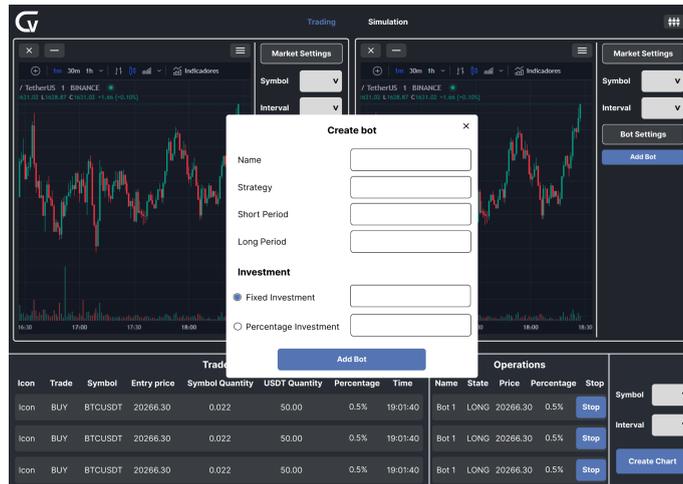


Figura 18: Vista de Trading con popup de crear un bot.

Cuando todos los datos sean completados, se podrá clicar en “Add Bot” para finalizar este proceso. Una vez clicado y verificado en los servidores de procesamiento que este bot puede ser añadido, aparecerá debajo de bot settings como otro contenedor tal y como se muestra en la Figura 19.

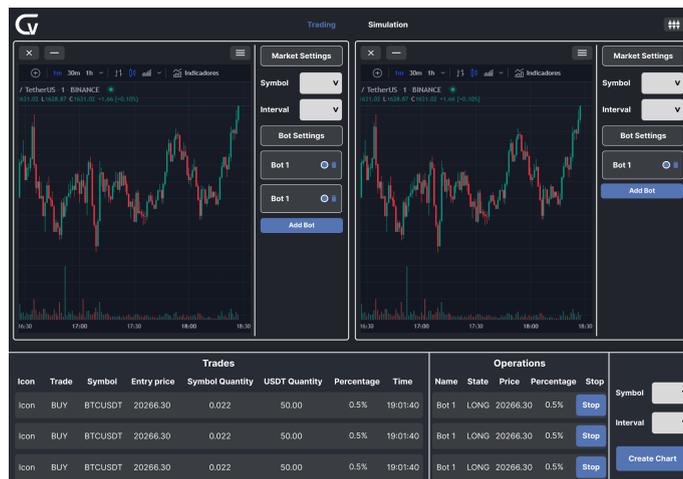


Figura 19: Vista de Trading con un bot añadido en la segunda gráfica.

La segunda página de la aplicación consiste en la **simulación**, que como se ha comentado anteriormente, únicamente está disponible para la aplicación de escritorio. Como se puede observar en la Figura 20, la página tiene también la barra superior de navegación con las mismas opciones, pero en este caso al estar en el modo de simulación, el texto azul indicará la palabra “Simulation”.

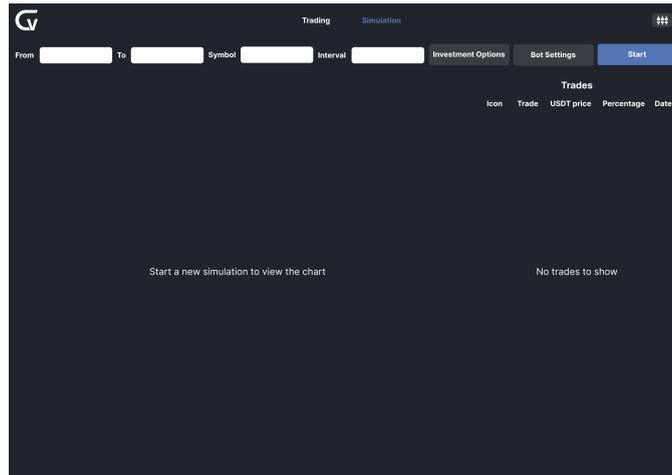


Figura 20: Vista principal de Simulation.

Bajo la barra superior, se muestran las opciones para ejecutar una simulación. Para ello, se tendrán que rellenar los inputs de “From”, el cual se refiere a la fecha de inicio para la simulación, y “To”, referido a la fecha final. El “Symbol” e “Interval” son los valores correspondientes a la criptomoneda, completándolos de la misma forma que en Trading.

A continuación, aparecen 3 botones diferentes. El primero de todos es “Investment Options”, que al clicarse abre un popup para completar las opciones de inversión para el resultado de la simulación. Después, al clicar en “Bot Settings” se abre otro popup para rellenar los valores de los bots. La funcionalidad del botón de “Start” inicia la simulación y si al clicar sobre este, alguno de estos valores no está completado o no es válido, aparece un mensaje indicando que no es posible iniciar la simulación y se deben corregir los errores.

Como se ha comentado anteriormente, el botón de bot settings abre una ventana emergente, donde se tiene que seleccionar una estrategia y rellenar sus valores específicos, de la misma forma que se ha explicado anteriormente a la hora de añadir un bot en Trading. En la Figura 21 se muestra la vista de la página de simulación con el popup de configuración del bot.

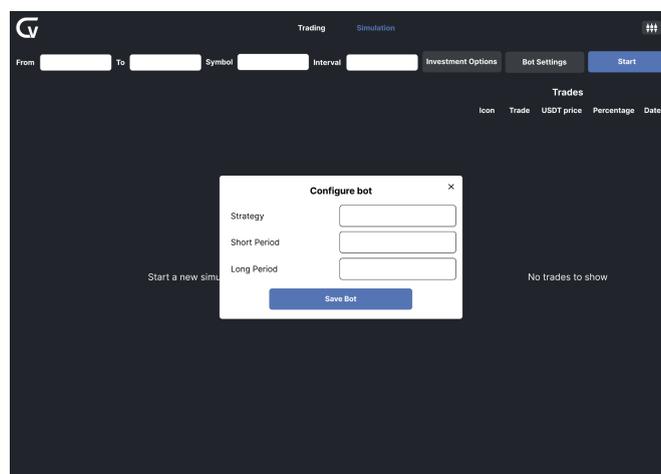


Figura 21: Vista de Simulation con popup de configuración del bot.

Una vez que se han completado todas las opciones necesarias y se hace clic en el botón "Start", la simulación se realiza en el servidor de procesamiento con la configuración introducida en la aplicación. De esta manera, se carga una gráfica de TradingView, una plataforma de gráficos enfocada en traders [32], en la parte izquierda de la página, especificando en ella los instantes de compra y venta. Luego, se van añadiendo los valores de los trades en la parte derecha de la página, indicando cinco valores: un icono, el tipo de trade (compra o venta), el precio, el porcentaje y la fecha en que se tomaron esos valores. En la Figura 22 se puede ver la página después de realizar la simulación.

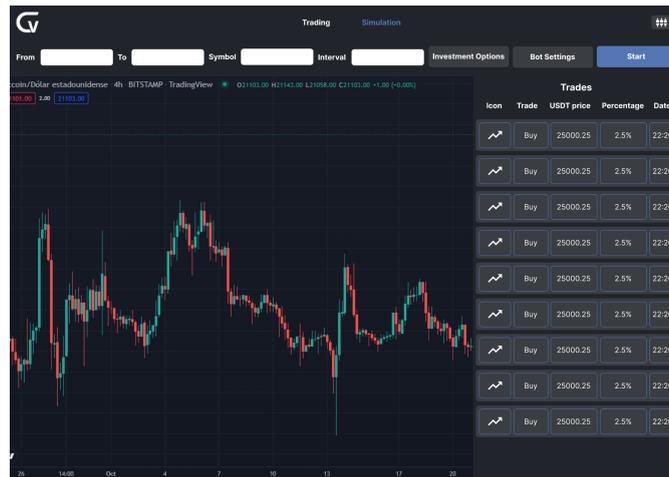


Figura 22: Vista de Simulation con datos obtenidos.

Y por último, justo después de haberse cargado todos los resultados, aparecerá finalmente una ventana emergente indicando que cantidad de dinero se hubiese obtenido con la configuración indicada y los valores de inversión rellenos.

Una vez realizado el diseño del proyecto, la idea a desarrollar se encontraba concluida. En ese momento, el siguiente paso era la implementación del mismo.

### 3.5 Implementación de la Aplicación

Cuando la tecnología a utilizar y los diseños estaban seleccionados, fue momento de implementar la aplicación. Durante este apartado se comentará el proceso seguido para la realización de la aplicación en su totalidad, incluyendo sus vistas y la comunicación necesaria con los servidores.

Es importante que durante el desarrollo se pueda trabajar con un producto mínimo viable. Por ello, se realiza un proyecto progresivamente en base a pequeños avances en las vistas de la aplicación. De esta forma, se puede construir un proyecto de forma global e ir añadiendo las diferentes funcionalidades de la página.

Como se ha comentado anteriormente, este proyecto está compuesto por dos aplicaciones diferentes, una web y una de escritorio. Aun así, comparten una estructura muy similar, en la que el front end es una página web, pero en la aplicación de escritorio se aloja

localmente y se visualiza en un launcher de escritorio; y en la aplicación web, se visualiza mediante un navegador web, y esta se encuentra alojada en un servidor remoto.

Entonces, ambas utilizan el mismo modelo, de la misma forma que la comunicación con las bases de datos y las APIs aunque éstas pertenezcan a servidores diferentes.

### 3.5.1 Front-end

El primer paso para implementar las aplicaciones es desarrollar el front-end, es decir, la interfaz gráfica para que el usuario pueda ver e interactuar con la información de forma digital. Entonces, durante este apartado, se describe la implementación de los mockups realizados durante el diseño.

Para la creación de la aplicación web, se ha hecho uso de un único archivo css, común para todas las vistas. Y en el caso del js y del html, cada página tiene un archivo para su estructura y sus funciones.

En la página de Simulation, se utiliza un librería de *TradingView*, empresa que proporciona software para la implementación de gráficas de trading en aplicaciones web. Esta permite mostrar las gráficas de los activos digitales y modificar su apariencia.

Finalmente, también se encuentra un archivo común de js para todas las páginas, dedicado a los métodos comunes de interacciones con el servidor web. Estas interacciones se realizan mediante un conexión entre la aplicación y el servidor web, la cual se comentará en el siguiente apartado.

### 3.5.2 Comunicación

Es necesaria una comunicación entre la aplicación y el servidor web, independientemente de si el servidor está alojado localmente o de forma externa. Como se puede observar en la Figura 23, se hace uso de una API y un websocket.

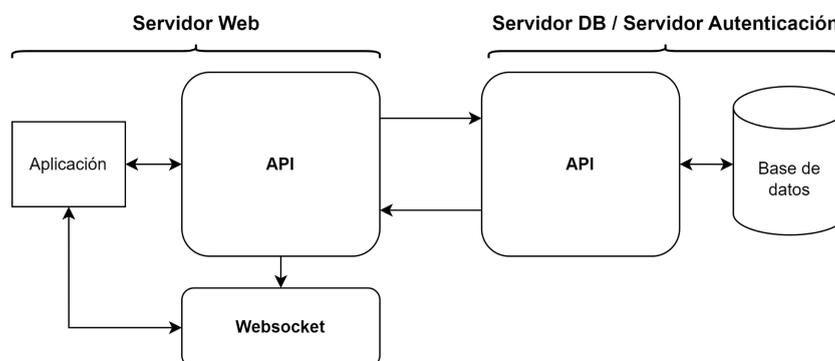


Figura 23: Diagrama de conexión entre servidor de Web y servidor de Base de datos.

La funcionalidad principal de este API es mantener una comunicación directa entre la aplicación y el servidor de base de datos. De esta forma, desde la web se mandan

directrices al servidor web para que este envíe solicitudes al servidor de base de datos, que en función de la solicitud recibirá unos datos u otros.

En cambio, el websocket sirve para mantener una comunicación bidireccional. Entonces el servidor web puede enviar mensajes a la aplicación sin esta solicitarlos. El servidor web mantiene una conexión bidireccional con cada cliente conectado mediante websocket. Cada conexión se almacena en un objeto y se asigna a una identidad única de usuario, para poder acceder a esa conexión cuando sea necesario. Estos mensajes enviados a la aplicación web o de escritorio son utilizados para enviar actualizaciones de operaciones, nuevos trades realizados y errores procedentes del servidor de procesamiento de análisis de datos.

Como se ha comentado anteriormente, es necesaria una API para la comunicación entre el servidor web y el servidor de la base de datos para poder acceder a la información almacenada. En este caso, en las dos aplicaciones no está presente la misma estructura. En el caso de la aplicación de escritorio, se presentan 2 bases de datos, una local y otra remota, y por este motivo es necesario el desarrollo de 2 APIs y conexiones diferentes. En cambio, la aplicación web tiene una base de datos unificada externa.

El objetivo de mantener este servidor de base de datos separado del servidor web es principalmente aumentar la seguridad de los datos. De esta manera, no es tan fácil conocer la IP y puerto del servidor directamente desde la aplicación web. Además, se agiliza la ejecución de los procesos del sistema de almacenamiento, lo que permite que el sistema realice las tareas de manera más efectiva.

Una vez finalizada la implementación de las aplicaciones, el producto ya era mínimamente viable, con sus vistas funcionales y su conexión a la base de datos. Entonces, era momento de realizar el sistema de procesamiento de datos para completar el desarrollo del proyecto.

### 3.6 Implementación del análisis de datos

Para la aplicación de escritorio, se ha implementado un sistema de análisis y procesamiento de datos en un servidor local, que se encarga de procesar los datos únicamente en el dispositivo del usuario. Por otro lado, la aplicación web cuenta con un servidor remoto dedicado que se encarga de procesar los datos de todos los usuarios que accedan a la aplicación a través de internet.

En este sistema, se inician las automatizaciones de los bots, se toman las decisiones mediante las estrategias y se realizan las operaciones. Por consiguiente, es necesaria una conexión con el servidor de la base de datos, que ejerce también como comunicador entre la web y el sistema de procesamiento de los datos.

### 3.6.1 APIs

Como se ha comentado anteriormente y como se puede observar en las figuras respecto a la arquitectura del proyecto, hay una conexión entre el servidor de análisis de datos y el servidor de base de datos, sean locales o externos. Entonces, para mantener una comunicación entre estos dos servidores se ha hecho uso de dos APIs, una alojada en cada servidor.

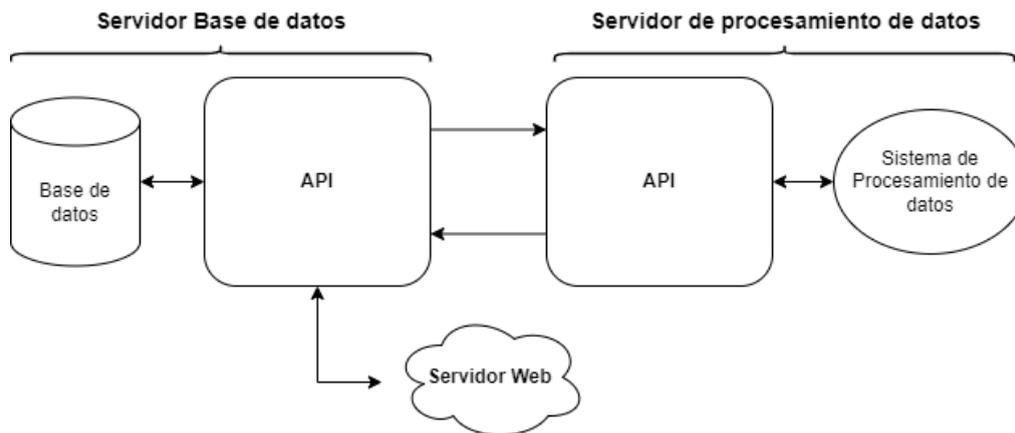


Figura 24: Diagrama de conexión entre servidor de Base de datos y servidor de procesamiento de datos.

La función de la API del servidor de procesamiento de datos es principalmente la recepción de órdenes por parte del servidor de la base de datos en cuanto a los bots. De esta forma, el sistema inicia o elimina un bot conforme la API haya recibido las peticiones.

La API del servidor de base de datos, como se ha comentado en el apartado anterior de Implementación de la Aplicación y que se puede observar en la Figura 24, mantiene una conexión con el servidor web. Pero esta no es la única, porque también necesita de una comunicación directa con el servidor de procesamiento, dado que una vez analizados los datos mediante los bots, es necesario enviar peticiones al servidor de base de datos para que estos se almacenen y, seguidamente, se envíen a la aplicación para ser vistos por el usuario.

### 3.6.2 Bots

La automatización es la base principal del proyecto, encargada de realizar de forma periódica todo el proceso desde el inicio de un bot hasta la realización de una decisión. Esta automatización la realizan los bots del sistema de procesamiento de análisis. Cada uno de los bots mediante su configuración realiza el análisis periódico de la criptomoneda que se haya especificado e indica qué tipo de decisión se va a tomar.

Estos bots se mantienen almacenados en el servidor de procesamiento, en un objeto diferenciándolos por ids. Este objeto es diferente en cada una de las aplicaciones, tal y como aparece en la Figura 25, ya que en el caso de la aplicación web, es necesario diferenciar los bots de cada usuario, y en el caso de la aplicación de escritorio, al ser un servidor local, únicamente se almacenan los de un único usuario. Gracias al uso de un

objeto y su diferenciación por ids, es más sencillo identificarlo, para iniciar o detener el proceso del bot en específico. La estructura del objeto se indica en la Figura 25.

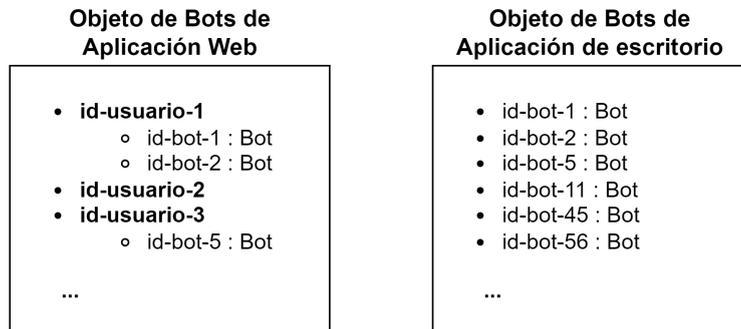


Figura 25: Esquema del objeto que almacena los bots en el sistema de procesamiento de datos.

Una vez iniciado un bot, este va a empezar un proceso periodico cada cierto tiempo, el cual viene determinado por el intervalo, que se ha establecido en la aplicación. El proceso se detendrá cuando el bot en cuestión se apague desde la vista de la aplicación.

El proceso se comentará a continuación y se puede observar de forma esquemática en la Figura 26.

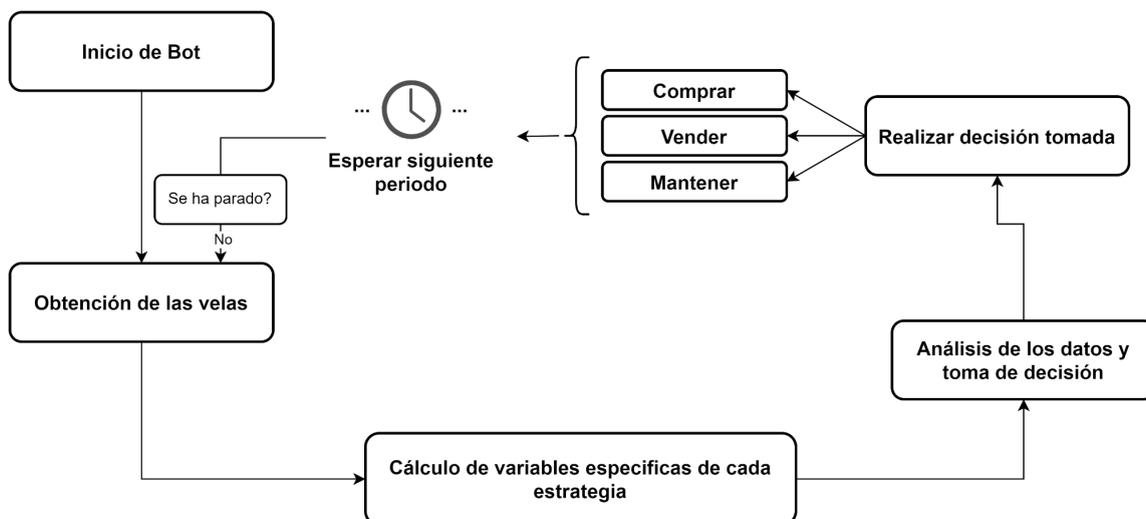


Figura 26: Esquema del proceso completo de un bot en el sistema de procesamiento de datos.

Después de iniciar el Bot, el primer paso es obtener los datos de las velas de la criptomoneda. Esto implica recoger la información que se muestra en el gráfico para cada período de tiempo determinado. Cada vela en el gráfico representa un período de tiempo, que puede ser un minuto, una hora o un día, dependiendo del tipo de gráfico que se esté utilizando. Cada vela contiene información sobre el precio de apertura, el precio de cierre, el precio más alto y el precio más bajo para el período de tiempo representado por la vela.

Con estos datos se realizan los cálculos pertinentes de la estrategia que se vaya a utilizar, como por ejemplo, calcular la Media Móvil Exponencial.

Una vez calculados todas las variables o datos, se realiza el análisis con esa información. El análisis depende de la estrategia que se vaya a utilizar. Una vez finalizado, se toma una decisión. Hay 3 tipos de decisiones que se pueden tomar: la compra del activo, la venta del activo o mantener la operación (no realizar ningún movimiento).

Finalmente, una vez tomada la decisión a realizar, y que ésta consista en compra o venta, se realiza la operación en cuestión. A continuación, todos los datos serán enviados a los otros servidores para almacenar la información y también para mostrarla al usuario.

Este proceso se va a realizar durante todo el tiempo que el bot se mantenga encendido y únicamente finalizará cuando lo indique el usuario.

En el siguiente apartado se comentan las estrategias técnicas, encargadas de tomar la decisión en cada proceso.

### 3.6.3 Estrategias

Las estrategias son algo esencial a la hora de realizar un análisis. Una estrategia de trading consiste en un conjunto de reglas y comportamientos que orientan a un inversor, o en este caso a un automatización, para lograr seleccionar momentos idóneos en los cuales realizar compraventas de activos financieros.

Estas estrategias se centran en aspectos como los indicadores técnicos, la capitalización del mercado, los riesgos, el sector de la industria, etc. A la hora de automatizar procesos de análisis es importante tener en cuenta que no es posible incorporar decisiones humanas, puesto que se hace uso únicamente de operaciones matemáticas.

En la aplicación, se emplean estrategias técnicas mediante indicadores matemáticos para ayudar a decidir en qué momento se deberían comprar o vender activos digitales. Es en la aplicación donde el usuario tendrá la opción de importar estrategias mediante scripts u optar por las estrategias predeterminadas de la aplicación, y estas son:

**DEMA** o Double Exponential Moving Average es una estrategia que consiste principalmente en un indicador técnico, utilizado para indicar la tendencia en el precio de una criptomoneda. Su proceso consiste en rastrear el precio del activo a lo largo del tiempo, detectando si la tendencia es alcista o bajista.

El cálculo de esta estrategia se basa en la diferencia de dos Medias Móviles Exponenciales (EMA), una de corto periodo y otra de largo periodo. En la cual:

$$\text{EMA} = \text{Closing price} \times \text{multiplier} + \text{EMA (previous day)} \times (1 - \text{multiplier}) \quad [33]$$

De esta forma, cuando la EMA de periodo largo > EMA corto periodo, es señal de venta, y cuando EMA corto periodo > EMA largo periodo, es señal de compra. Esto se puede ver con facilidad en la gráfica de la Figura 27.



Figura 27: Representación gráfica de estrategia DEMA. Fuente: <https://es.tradingview.com/chart/?symbol=ETHUSD/>.

**MACD** o Moving Average Convergence/Divergence es también una estrategia de impulso de seguimiento de tendencia. Esta tendencia muestra la relación entre dos medias móviles exponenciales. La línea MACD se calcula restando la EMA de un periodo mayor de la de uno menor, donde los periodos habituales son 26 y 12 respectivamente.

Después tenemos una línea de señal, que consiste en una EMA de, por lo general, periodo 9. De esta forma, cuando la línea de señal y la línea MACD se cruzan, representan una señal de compra o venta. Cuando la línea MACD cruza por encima de la línea de señal indica tendencia alcista, y cuando cruza por debajo, indica tendencia bajista [34]. Esto se puede observar en la Figura 28.



Figura 28: Representación gráfica de estrategia MACD. Fuente: <https://www.investopedia.com/terms/m/macd.asp/>.

**Bollinger** es otra estrategia que utiliza las bandas de bollinger, un tipo de indicador técnico muy empleado en el análisis de todos los tipos de mercados. Las bandas de bollinger se componen de tres líneas:

- La **banda media** utiliza un promedio móvil simple del periodo que se indique, normalmente 20.
- La **banda superior** se calcula sumando el doble de la desviación estándar diaria a la banda media.
- La **banda inferior** se calcula restando dos veces la desviación estándar diaria a la banda media.

Cuando el precio de la vela actual es inferior o igual al valor de la banda inferior de bollinger se trataría de un indicador de compra, y cuando el precio es mayor o igual al valor de la banda superior de bollinger, se trataría de un indicador de venta del activo [35]. Los indicadores de compra o venta se muestran en la Figura 29 mediante una representación gráfica de la estrategia de Bollinger.



Figura 29: Representación gráfica de estrategia Bollinger. Fuente: <https://www.investopedia.com/trading/using-bollinger-bands-to-gauge-trends/>.

Para implementar esta estrategia se ha hecho uso de la librería *Bollinger Bands*, que consiste en una utilidad matemática para calcular las bandas de Bollinger [36].

Por último, y como ya se ha comentado anteriormente, está la opción de importar estrategias personalizadas por parte del usuario. Estas estrategias necesitan ser desarrolladas en Typescript, siguiendo unas normas preestablecidas indicadas en el anexo "Manual de usuario".

### 3.6.4 Operaciones

Una vez se ha tomado una decisión mediante las estrategias y esta consiste en **compra** o **venta**, es el momento en el que se realiza una operación. Para realizar una operación hay que tener unos puntos claros. Primero de todo, se necesita de una cuenta de Binance con sus claves que permiten realizar operaciones mediante su API. Se trata de dos claves: una pública y una privada.

Después, es necesario tener activos en la cuenta, en otras palabras un balance con el que poder interactuar. De todas formas, la aplicación permite utilizar la *testnet* como red de Binance, que consiste en una red de pruebas diseñada para experimentar. Esta proporciona un portafolio virtual de prueba con diferentes activos.

Las operaciones que se quieran realizar a la mainnet, la red principal de Binance, requieren una URL base de la petición, "https://api.binance.com/api/". En cambio, para realizar las peticiones a la testnet, la red para pruebas, se utiliza como URL base,

“https://testnet.binance.vision/”. Cabe destacar que el usuario puede elegir, desde la aplicación, que red utilizar.

Para realizar un nuevo orden a la API, se realizará una petición **POST /api/v3/order**. Para la query de esta petición son necesarios una serie de parámetros. Primero de todo, es imprescindible obtener una firma digital mediante la clave secreta. Esta firma es estrictamente necesaria para poder confirmar esta nueva orden en la cuenta de Binance. Pero además, son necesarios otros datos para esta operación, reflejados en la Figura 30.

Name	Type	Mandatory	Description
symbol	STRING	YES	
side	ENUM	YES	
type	ENUM	YES	
timeInForce	ENUM	NO	
quantity	DECIMAL	NO	
quoteOrderQty	DECIMAL	NO	
price	DECIMAL	NO	
newClientOrderId	STRING	NO	A unique id among open orders. Automatically generated if not sent.
stopPrice	DECIMAL	NO	Used with <code>STOP_LOSS</code> , <code>STOP_LOSS_LIMIT</code> , <code>TAKE_PROFIT</code> , and <code>TAKE_PROFIT_LIMIT</code> orders.
icebergQty	DECIMAL	NO	Used with <code>LIMIT</code> , <code>STOP_LOSS_LIMIT</code> , and <code>TAKE_PROFIT_LIMIT</code> to create an iceberg order.
newOrderRespType	ENUM	NO	Set the response JSON. <code>ACK</code> , <code>RESULT</code> , or <code>FULL</code> ; <code>MARKET</code> and <code>LIMIT</code> order types default to <code>FULL</code> , all other orders default to <code>ACK</code> .
recvWindow	LONG	NO	The value cannot be greater than <code>60000</code> .
timestamp	LONG	YES	

Figura 30: Parámetros para new Order de Binance API. Fuente: <https://developers.binance.com/docs/binance-trading-api/spot#new-order--trade/>.

Después de realizar la petición, si se recibe una respuesta correcta con un código de estado 200, se proporcionará un cuerpo con información sobre la orden. La respuesta incluirá datos como la cantidad de usdt gastada, la cantidad de activo digital comprado con esa cantidad de usdt, el momento de la transacción y otros detalles, como se puede ver en la Figura 31.

```
{
  "symbol": "BTCUSDT",
  "orderId": 28,
  "orderListId": -1, //Unless OCO, value will be -1
  "clientOrderId": "6gCrw2kRUAf9CvJDGP16IP",
  "transactTime": 1507725176595,
  "price": "0.00000000",
  "origQty": "10.00000000",
  "executedQty": "10.00000000",
  "cumulativeQuoteQty": "10.00000000",
  "status": "FILLED",
  "timeInForce": "GTC",
  "type": "MARKET",
  "side": "SELL"
}
```

Figura 31: Ejemplo de respuesta de new Order de Binance API. Fuente: <https://developers.binance.com/docs/binance-trading-api/spot#new-order--trade/>.

### 3.6.5 Simulación

La simulación utiliza un proceso muy similar a la hora de realizar el análisis, pero su proceso es algo diferente, ya que no se ejecuta ninguna orden. Únicamente se procede a guardar las decisiones tomadas en una lista, que posteriormente es enviada a la aplicación.

El proceso se puede observar en el esquema de la Figura 32, y se encuentra explicado paso por paso a continuación.

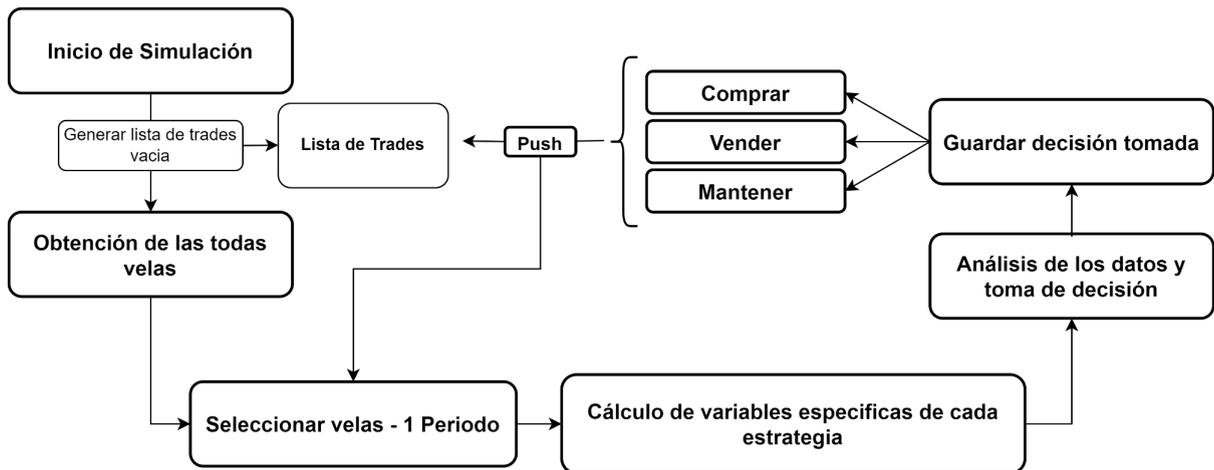


Figura 32: Esquema del proceso completo de un bot en el sistema de procesamiento de datos.

Al inicio de la simulación, se genera una lista de trades vacía, la cual se rellenará con las decisiones. A continuación, se obtiene el valor de todas las velas desde el inicio del periodo hasta el final mediante los valores indicados en la aplicación.

De esta forma, se inicia un bucle en el cual se analizan bloques de velas, restando cada vez un periodo. El bucle finaliza cuando no hay suficientes velas para realizar un análisis completo.

Seguidamente, durante el bucle, con cada bloque de velas se calcularán los valores específicos de la estrategia definida y se realizará un análisis de los datos. Se tomará la decisión, y acto seguido, no se ejecutará ninguna orden, y se guardará la decisión que haya sido tomada como un objeto con valores de porcentaje, precio y tiempo. Este objeto se insertará en la lista de trades.

Finalmente, serán enviados de vuelta a la aplicación. Y allí, con la configuración elegida de inversión, será calculado el beneficio que se hubiera obtenido durante ese periodo y este será mostrado al usuario.

## 3.7 Tests

Para realizar una verificación del trabajo realizado se han desarrollado unos tests unitarios, los cuales consisten en una forma efectiva de comprobar el correcto funcionamiento del programa informático, en este caso del sistema de procesamiento de los datos.

Para desarrollar estos tests unitarios se ha utilizado *Mocha*, *Chai* y *Sinon*, marcos de prueba de *JavaScript* enfocados para programas de *Nodejs* [37] [38] [39]. Estos paquetes permiten realizar pruebas para comprobar la ejecución correcta del código, incluyendo pruebas asíncronas, el uso de librerías, etc.

Se han realizado diferentes comprobaciones divididas en dos partes, el funcionamiento del sistema del trading y el de la simulación.

En primer lugar, para la verificación del preciso funcionamiento del sistema de trading, se incluyen los tests de: creación de clientes con claves de Binance, el uso de esos clientes para realizar operaciones de compra o venta, la creación de bots, su inicio y su eliminación, el correcto momento de inicio del trading según su intervalo y el comportamiento de las estrategias.

En la segunda parte, se ha comprobado el sistema de simulación mediante diferentes tipos de tests. Primeramente, se ha comprobado la correcta creación de un bot para simulación. Posteriormente, se ha verificado si los resultados obtenidos mediante una simulación con una estrategia predeterminada (o importada) y una configuración específica han sido los esperados.

Por último, se han realizado comprobaciones con dos simulaciones, con la intención de obtener el mismo resultado si las estrategias y configuraciones eran iguales, u obteniendo diferente resultado si se utilizaban diferentes estrategias o mismas estrategias pero con diferentes configuraciones. De igual manera, se ha comprobado la simulación con una configuración en la cual el resultado debía ser una lista vacía.

Con la realización de estos tests unitarios se ha podido constatar el correcto funcionamiento del código del sistema de procesamiento de los datos, utilizando diferentes estrategias y configuraciones.

## Capítulo IV

# Resultados

Durante este capítulo, se muestran los diferentes resultados de trading, con valores en tiempo real, y de simulación con históricos de datos, obtenidos al realizar los tests con la herramienta desarrollada. En este apartado se presentan únicamente algunas de las pruebas realizadas. En el documento “Resultados” encontrado en el anexo, se incluyen todas las pruebas realizadas y resultados obtenidos.

### 4.1 Resultados de trading

Para realizar la comprobación de resultados en el sistema de trading, se ha optado por realizar análisis con diferentes estrategias e intervalos. De esta forma, se han iniciado 9 bots diferentes entre sí, empezando todos a las 9:30 del 21/11/2022 y finalizando a las 16:30 del 21/11/2022. Durante este periodo, como se puede observar en la Figura 33, los valores han fluctuado entre 15957 USDT y 16248 USDT, con una diferencia de un 1.79%.



Figura 33: Gráfico de Bitcoin de 1m de Intervalo desde las 9:30 del 21/11/2022 hasta las 16:30 del 21/11/2022. Fuente: <https://tradingview.com/>.

Estas pruebas se han realizado con intervalos pequeños, de 1, 3 y 5 minutos. De esta forma, el porcentaje resultante es pequeño y se llevan a cabo más operaciones cuanto menor es el intervalo. Una vez obtenidos los trades realizados, se han dividido en 4 categorías, con resultados mayores a un 0.1%, entre 0% y 0.1%, entre 0% y -0.1% y finalmente menor a -0.1%.

También se ha plasmado el porcentaje resultante de todas las operaciones después de las 7 horas en cada intervalo. En todos los bots, se han utilizado los valores por defecto dados por la herramienta.

Los resultados de las estrategias 2EMA, MACD y Bollinger se presentan en las Tablas 3, 4 y 5, respectivamente.

Tabla 3: Resultados de estrategia 2EMA con diferentes intervalos.

Estrategia 2EMA						
Intervalo	> 0.1%	Entre 0% y 0.1%	Entre 0% y -0.1%	< -0.1%	Total	Porcentaje Resultante
1m	6	15	22	3	46	0.368%
3m	2	3	5	4	14	0.127%
5m	1	2	2	3	8	-0.058%

Tabla 4: Resultados de estrategia MACD con diferentes intervalos.

Estrategia MACD						
Intervalo	> 0.1%	Entre 0% y 0.1%	Entre 0% y -0.1%	< -0.1%	Total	Porcentaje Resultante
1m	6	4	7	7	25	0.199%
3m	1	1	1	4	7	0.021%
5m	2	0	0	1	3	0.315%

Tabla 5: Resultados de estrategia Bollinger con diferentes intervalos.

Estrategia Bollinger						
Intervalo	> 0.1%	Entre 0% y 0.1%	Entre 0% y -0.1%	< -0.1%	Total	Porcentaje Resultante
1m	1	0	1	0	2	0.093%
3m	1	0	0	0	1	0.772%
5m	0	0	0	0	0	0.000%

Como se puede observar en las tablas, la estrategia que genera más operaciones es la **DEMA**, porque utiliza periodos pequeños. De esta manera, cuando se utilizan intervalos menores a 10 minutos, suele generarse una gran cantidad de operaciones con porcentajes resultantes entre 0.1% y -0.1%.

En el caso de la estrategia **MACD** se genera también una gran cantidad de operaciones, pero suelen ser operaciones con un valor de porcentaje resultante más grande, entonces suelen ser mayores a 0.1% o menores a -0.1%. Finalmente, la estrategia de **bollinger** genera muy pocas operaciones, ya que es más complicado que con periodos cortos se ocasionen decisiones de compra.

Por los motivos expuestos, se puede determinar que en intervalos inferiores a 30 minutos la estrategia más adecuada es la **DEMA**.

## 4.2 Resultados de simulación

Por otra parte, para la comprobación de los resultados del sistema de simulación de históricos de datos, se ha realizado una simulación con cada una de las estrategias encontradas en la aplicación, con sus valores predeterminados, con la criptomoneda Bitcoin

y un intervalo de 5 minutos. Esta simulación se ha realizado desde el 03/11/2022 a las 00:00 hasta el 05/11/2022 a las 00:00.

Para cada estrategia, se ha elegido mostrar los resultados a través de una gráfica. La gráfica muestra todos los valores de porcentaje de cada una de las operaciones finalizadas.

En primer lugar, se encuentra la estrategia **DEMA** y el resultado se puede observar en la gráfica de la Figura 34.

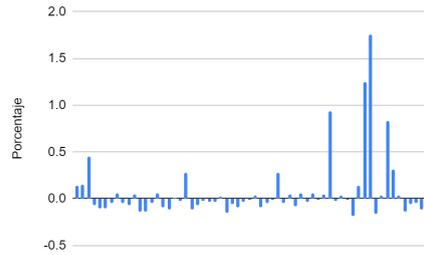


Figura 34: Gráfica de porcentajes resultantes de la simulación con la estrategia DEMA.

La cantidad total de operaciones son 62, con un porcentaje resultante de 4,356%.

A continuación, se encuentra la estrategia **MACD** y el resultado se puede observar en la gráfica de la Figura 35.

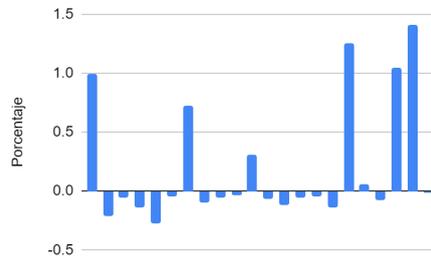


Figura 35: Gráfica de porcentajes resultantes de la simulación con la estrategia MACD.

La cantidad total de operaciones son 22, con un porcentaje resultante de 4,374%.

Finalmente, se encuentra la estrategia de **Bandas de Bollinger** y el resultado se puede observar en la gráfica de la Figura 36.

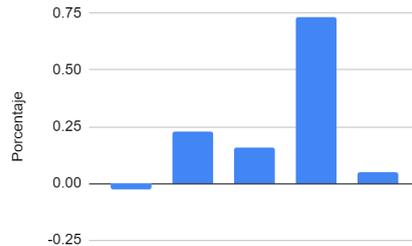


Figura 36: Gráfica de porcentajes resultantes de la simulación con la estrategia de Bandas de Bollinger.

La cantidad total de operaciones son 5, con un porcentaje resultante de 1,138%.

Como se puede observar en los datos, la estrategia que más operaciones genera es la estrategia DEMA, pero a su vez se trata de la más arriesgada, ya que trabaja con intervalos muy cortos y por lo tanto, presenta más volatilidad. En estas simulaciones, el resultado de la estrategia MACD y la estrategia DEMA tienen un mayor porcentaje resultante y similar, pero también presentan una mayor probabilidad de generar porcentajes negativos, aunque estos no sean inferiores a -0.22%.

Por el contrario, al aplicar la estrategia de Bollinger, solo 1 de los 5 resultados fue negativo y tuvo un valor de -0.03%, lo que es mucho menor en comparación con las otras operaciones.

En conclusión, se puede observar que todas las simulaciones han tenido un porcentaje mayor al final, debido a que el mercado experimentó un aumento en los precios durante ese tiempo.

### 4.3 Modelo de negocio

Una vez finalizado este proyecto se ha concebido una idea de negocio, dicho en otras palabras una forma de vender el producto generado.

En este mercado es importante dar la opción al cliente de poder prescindir del producto cuando lo considere. De esta forma, el modelo de negocio más adecuado es mediante una membresía o suscripción mensual. Ello permite que el usuario disponga de la posibilidad de probar el producto durante un mes, pudiendo continuar con la suscripción si se encuentra satisfecho o, en caso contrario, cancelarla sin ningún tipo de penalización.

Por otra parte, además de realizar una suscripción mensual se permitiría al cliente elegir entre diferentes planes de membresía. De esta forma, se limitan o proporcionan características exclusivas según el importe que desee abonar el usuario, es decir, el precio aumenta según más funcionalidades puedas utilizar dentro de la aplicación. En la Figura 37 se muestra una imagen de cómo podrían ofrecerse los diferentes planes.

Principiante	Normal	Avanzado
Max 10 bots activos Acceso a 10 Estrategias Acceso únicamente mediante página web Unlimited Trades	Max 20 bots activos Acceso a 15 Estrategias Acceso mediante App de escritorio o página web Unlimited Trades Acceso a simulación Acceso a testing trades	Ilimitados bots activos Acceso a todas las Estrategias Acceso mediante App de escritorio o página web Unlimited Trades Acceso a simulación Acceso a testing trades Importe de Scripts de estrategias custom
Select Plan	Select Plan	Select Plan

Figura 37: Visualización de niveles de planes de suscripción del modelo de negocio.

En el mundo del trading y las criptomonedas, la forma adecuada de pago es mediante activos digitales, como Bitcoin o Ethereum. De esta forma, el usuario trader puede optar a abonar la suscripción sin necesidad de realizar un pago con dinero *fiat*. Consecuentemente, en la mayoría de los casos, los usuarios estarían dispuestos a pagar el precio de la aplicación con menor preocupación.

## 4.4 Manual de usuario

El manual de usuario se encuentra en el anexo “Manual de usuario”, en el cual contiene instrucciones de cómo ejecutar cada uno de los proyectos, una guía del uso de las aplicaciones mediante imágenes del proyecto ejecutado, y finalmente una guía de la implementación de estrategias mediante scripts.

## Capítulo V

# Conclusión

En este proyecto se ha desarrollado una herramienta para la simulación y evaluación de estrategias de inversión en mercados de criptomonedas y, como se ha podido comprobar durante el trabajo, se han cumplido todos los objetivos secundarios planteados en la introducción.

Con el fin de conseguir los resultados obtenidos se han llevado a cabo una serie de procesos hasta llegar a completar la herramienta. En primer lugar, se ha realizado una investigación teórica sobre las criptomonedas y el trading, valorando las diferentes opciones y analizando los competidores dentro del mercado, pudiendo concebir la idea final. Posteriormente, se han implementado las dos opciones de aplicación, la aplicación web y la aplicación de escritorio, para continuar con el desarrollo del sistema de procesamiento de datos mediante el uso de estrategias técnicas.

Con el sistema automatizado junto a las aplicaciones se debía realizar una comprobación de que el resultado era el esperado. Por ello, se ejecutaron pruebas con el fin de comprobar el correcto funcionamiento de la aplicación, y para realizar un estudio sobre las mejores estrategias y sus configuraciones recomendadas.

Una vez finalizado el proyecto y los tests, se ha reflexionado sobre posibles aspectos de mejora y posibles implementaciones útiles en el producto, con el fin de ofrecer una herramienta más completa a los usuarios.

En primer lugar, uno de los objetivos principales para un trabajo futuro sería la creación de más estrategias técnicas e incluso la adición de estrategias fundamentales, mediante el estudio de factores financieros con inteligencia artificial (IA). Otro punto interesante, sería añadir una capa de IA en la toma de decisiones del sistema de procesamiento de datos.

Por último, otro punto muy beneficioso para el producto, sería la disponibilidad de la herramienta como aplicación móvil. De esta forma, el cliente podría acceder fácilmente al producto desde cualquier dispositivo.

Este proyecto está vinculado a los Objetivos de Desarrollo Sostenible (ODS), específicamente al objetivo número 9: Industria, Innovación e Infraestructuras. Ya que se trata de un producto innovador el cual promueve el uso de las nuevas tecnologías como son las monedas digitales.

De forma indirecta, durante el desarrollo de este nuevo producto he trabajado algunas de las competencias transversales tratadas y aprendidas durante el transcurso de la carrera. Entre ellas, destaco principalmente tres, las cuales han tenido un gran papel para la elaboración del proyecto. Estas son: **análisis y resolución de problemas**, a la hora de considerar y encontrar respuesta a los problemas que han aparecido durante el desarrollo;

**innovación, creatividad y emprendimiento**, debido a que se ha innovado a la hora de crear un nuevo producto para una cuestión tan actual como son las criptomonedas; y por último, **diseño y proyecto**, puesto que se ha realizado la evaluación y el diseño de una nueva idea y se ha concretado en un proyecto.

Al finalizar el trabajo, puedo concluir que su desarrollo me ha permitido reconocer y saber aplicar la gran cantidad de conocimientos que he aprendido a lo largo del grado. Asimismo, la implementación del proyecto me ha permitido profundizar en cuanto a lenguajes como Typescript y Node o el uso de APIs para la comunicación entre diferentes sistemas. En definitiva, la realización de este trabajo me ha aportado experiencia y seguridad para poder seguir desarrollando e innovando nuevos productos tecnológicos en el futuro.

## Capítulo VI

# Referencias

- [1] Milutinović, M. (2018). Cryptocurrency. *Ekonomika*, 64(1), 106.
- [2] Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., & Li, L. (2022). Cryptocurrency trading: a comprehensive survey. *Financial Innovation*, 8(1), 5.
- [3] Chaum, D. (1985). Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10), 1030-1044.
- [4] Ramiro, A., & de Queiroz, R. J. (2022). Cypherpunk. *Internet Policy Review*, 11(2), 2.
- [5] King, S. (2013). Primecoin: Cryptocurrency with prime number proof-of-work. *July 7th*, 1(6).
- [6] Romero Cubero, M. Á. (2020). Las criptomonedas, 11-12.
- [7] Tapscott, D., & Tapscott, A. (2016). *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. Penguin.
- [8] Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., & Li, L. (2022). Cryptocurrency trading: a comprehensive survey. *Financial Innovation*, 8(1), 4.
- [9] Di Pierro, M. (2017). What is the blockchain?. *Computing in Science & Engineering*, 19(5), 92.
- [10] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.
- [11] CryptoMarket. URL: <https://coinmarketcap.com/>
- [12] Binance (2022), *6 Big Brands That Accept BTC And Why*. URL: <https://www.binance.com/en/blog/payment/6-big-brands-that-accept-btc-and-why-421499824684903357/> .
- [13] Arslanian, H. (2022). The Book of Crypto. *Springer Books*, 337-345.
- [14] Estadísticas de Binance. URL: <https://www.businessofapps.com/data/binance-statistics/>
- [15] Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., & Li, L. (2022). Cryptocurrency trading: a comprehensive survey. *Financial Innovation*, 8(1), 9.
- [16] Nunes, M. D. A. (2021). *Automated Trading Systems VS Manual Trading in Forex Exchange Market* (Doctoral dissertation), 7.
- [17] Quance. URL: <https://quance.tech/>
- [18] Tuned. URL: <https://www.tuned.com/>
- [19] Haasonline. URL: <https://www.haasonline.com/>
- [20] GitHub. URL: <https://github.com/>
- [21] JavaScript. URL: <https://www.javascript.com/>
- [22] HTML. URL: <https://developer.mozilla.org/es/docs/Web/HTML>
- [23] Bootstrap. URL: <https://getbootstrap.com/>
- [24] Electron. URL: <https://electronjs.org/es/>
- [25] Nodejs. URL: <https://nodejs.org/>
- [26] Nodets. URL: <https://nodejs.dev/en/learn/nodejs-with-typescript/>
- [27] MongoDB. URL: <https://www.mongodb.com/>
- [28] Mongoose. URL: <https://mongoosejs.com/docs/>
- [29] Websocket. URL: <https://developer.mozilla.org/en-US/docs/Web/API/WebSocket/>
- [30] API de Binance. URL: <https://www.binance.com/en/binance-api/>
- [31] Candlestick Charts. URL: <https://www.investopedia.com/trading/candlestick-charting-what-is-it/>

- [32] TradingView. URL: <http://tradingview.com/>
- [33] Exponential Moving Average. URL: <https://www.investopedia.com/terms/e/ema.asp/>
- [34] Moving Average Convergence Divergence. URL: <https://www.investopedia.com/terms/m/macd.asp/>
- [35] Bollinger Bands. URL: <https://www.investopedia.com/trading/using-bollinger-bands-to-gauge-trends/>
- [36] Librería bollinger-bands. URL: <https://www.npmjs.com/package/bollinger-bands/>
- [37] Mocha. URL: <https://mochajs.org/>
- [38] Chai. URL: <https://www.chaijs.com/>
- [39] Sinon. URL: <https://sinonjs.org/>