



TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

TradingAPP

Aplicación web para trading algorítmico

Autor

Manuel Carmona Pérez

Director

José Manuel Benítez Sánchez



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Granada, mes de 2021

TradingAPP

Aplicación web para trading algorítmico

Autor

Manuel Carmona Pérez

Director

José Manuel Benítez Sánchez

TradingAPP: Aplicación web para trading algorítmico

Manuel Carmona Pérez

Palabras clave: trading algorítmico, trading automático, *Wyckoff*, mercados financieros

Resumen

El volumen de datos involucrado y la velocidad a la que se realizan las operaciones de compra y venta en los mercados financieros ha hecho que la intervención de procedimientos automatizados juegue un papel fundamental en las actuales acciones de compra y venta de distintos operadores en los mercados. En estos operadores encontramos grandes inversores como entidades bancarias o instituciones.

El desarrollo de algoritmos de trading es una pieza clave en la creación de estos métodos automáticos para operar. Estos algoritmos explotan técnicas de análisis de la acción de mercados financieros ya conocidas junto con métodos basados en aprendizaje automático, entre otros, para maximizar los beneficios en las compras o ventas realizadas.

El objetivo de este proyecto es desarrollar una aplicación que implemente técnicas para realizar trading mediante algoritmos basados en métodos de análisis técnico ya conocidas. Esta aplicación tendrá una interfaz web para que sea usada por un operador humano.

TradingAPP: Algorithmic trading web application

Manuel Carmona Pérez

Keywords: algorithmic trading, automic trading, *Wyckoff*, financial markets

Abstract

The large volume of data envolved and the speed at which the buy and sell operations are done in the financial markets have made the intervention of automated procedures a vital role in how operators buy and sell in the markets nowadays. Large investors like banks or other institutions can be found among these operators.

The trading algorithms development plays a key role in these automatic methods creation. These algorithms exploit known techniques that analyze the financial markets actions and machine learning nethids, among others, to maximize benefits in the sales and purchases done.

The main purpose of this project is to develop an application that implements techniques to trade by using algorithms based on already known technical analysis methods. This application will have a web interface as it will be used by a human operator.

Yo, **Manuel Carmona Pérez**, alumno del grado en ingeniería informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 17482989E, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Manuel Carmona Pérez

Granada, septiembre de 2021.

D. **José Manuel Benítez Sánchez(tutor)**, Profesor del Área de XXXX del Departamento YYYY de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***TradingAPP, Aplicación web para trading algorítmico***, ha sido realizado bajo su supervisión por **Manuel Carmona Pérez(alumno)**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada, septiembre de 2021 .

Los directores:

José Manuel Benítez Sánchez(tutor)

Agradecimientos

Poner aquí agradecimientos...

Índice general

1. Introducción	17
1.1. Trading y otros conceptos	17
1.2. Trading algorítmico	18
1.2.1. Ventajas	18
1.2.2. Desventajas	19
1.3. Objetivos	19
2. Planificación	23
2.1. Planificación temporal	23
2.1.1. Fases del proyecto	23
2.1.2. Diagrama de Gantt	25
2.2. Presupuesto	26
3. Análisis	27
3.1. Descripción de los implicados	27
3.2. Especificación de requisitos	27
3.2.1. Requisitos Funcionales	27
3.2.2. Requisitos No Funcionales	28
3.2.3. Requisitos de Información	29
3.3. Diagramas de casos de uso	29
3.3.1. Gestión de sesiones de la APP	29
3.3.2. Gestiones de la plataforma de trading (MT5)	29
3.3.3. Visualización de datos	30
3.3.4. Funcionalidades de trading	30
4. Diseño	33
4.1. Arquitectura del software	33
4.2. Primer diseño de la interfaz	34
5. Implementación	39
6. Pruebas	1
7. Conclusiones	1

Índice de figuras

2.1. Diagrama de Gantt del proyecto	25
3.1. Diagrama de casos de uso referente al sistema de gestión de usuarios de la APP	30
3.2. Diagrama de casos de uso referente a la gestión de la plataforma de trading (MT5)	31
3.3. Diagrama de casos de uso referente a la visualización de datos de mercados financieros	32
3.4. Diagrama de casos de uso referente a las funcionalidades de trading	32
4.1. Arquitectura general del software	33
4.2. Primer diseño del menú principal de la APP	35
4.3. Primer diseño del formulario de login en <i>MT5</i>	36
4.4. Primer diseño del menú principal (usuario identificado en <i>MT5</i>)	37

Capítulo 1

Introducción

1.1. Trading y otros conceptos

El trading es una práctica aplicada a los mercados financieros. Realizar trading implica comprar o vender dentro de los distintos mercados financieros para vender o comprar de nuevo al cabo de un tiempo y obtener beneficio de la acción realizada. A grandes rasgos y con un ejemplo, realizar trading beneficiosamente podría ser comprar una acción de *Apple* por 100 dólares y venderla en cierto tiempo por 150 dólares, obteniendo por tanto un beneficio de 50 dólares.

¿Cuándo se debe comprar o vender un activo? Los traders u operadores trabajan en torno a dos principales tipos de análisis:

- **Análisis fundamental:** El análisis fundamental consiste en operar en base a las distintas noticias que ocurren en el mundo diariamente. Estas noticias sirven de fundamento para actuar de una forma u otra en el mercado.
- **Análisis técnico:** En este tipo de análisis, el operador o trader opera puramente en base al precio: su acción y movimientos. En este caso, no se tienen en cuenta las noticias. Es este el tipo de análisis interesante a la hora de automatizar. Esto ocurre porque en este método entran en juego cálculos matemáticos como medias, medias móviles, zonas de probabilidad estadística, etc. A partir de este análisis surge el trading algorítmico.
Este análisis, aunque se pueda automatizar, no te asegura que el precio vaya en la dirección resultado del análisis. A pesar de esto, si trabajamos con probabilidades y obtenemos probabilidades de ganar superiores al 50

1.2. Trading algorítmico

En pocas palabras, el trading algorítmico es implementar un sistema de trading que opere de forma automática.

El trading algorítmico analiza gráficos de precios de acuerdo a unos criterios preestablecidos, que dependerán del análisis y del propio algoritmo. Cuando el mercado se ajusta a los criterios mencionados, el algoritmo que se ha diseñado para hacer trading ejecutará una acción de compra o venta de manera automática.

Aparte del ahorro de tiempo y esfuerzo, que es la obvia ventaja de usar trading algorítmico, podemos encontrar otros puntos a favor que hacen de las operaciones más eficientes si las comparamos con cómo las haría un operador humano.

1.2.1. Ventajas

- **Diversificación:** existe la posibilidad de aplicar un mismo análisis a distintos mercados financieros, aunque puede existir la posibilidad de que en ciertos mercados obtengamos beneficios con una técnica específica que no obtenemos en otro mercado distinto. Esto es más complicado para un operador humano ya que los cambios de acciones de precios en los mercados financieros hacen que un análisis técnico no sea sencillo para un trader acostumbrado a ciertos mercados financieros.
- **Evaluación de técnicas usadas:** debido a que el trading algorítmico automatiza la acción de realizar compras y ventas, podemos evaluar de forma fácil cuándo cierta técnica de análisis es más o menos eficiente en uno u otro mercado. Aquí podemos hablar de distintas horas del día, diferentes temporadas, mercados financieros, etc.
- **Evitar las emociones:** al automatizar un sistema para comprar y vender acciones, el operador humano evita dejarse guiar por las emociones. Esto puede parecer una ventaja simbólica, pero es bastante importante ya que el mercado suele estar sujeto a estadísticas, probabilidades, etc. Si el trader opera suponiendo que cierta vez ocurrirá algo distinto, acaba dejando de lado el análisis puramente técnico. En resumen, evitamos la principal razón por la cual la mayoría de personas que empiezan a dedicarse al trading fracasan, psicología y emociones.
- **Capacidad para desplegar en la nube:** al ser un algoritmo que puede ser desarrollado en un producto software, es posible desplegar o hacer deploy del mismo en un servidor, de manera que el algoritmo

desarrollado siempre está conectado al mercado y aplicando reglas para comprar o vender según los criterios mencionados.

- **Precisión y capacidad para realizar compras y ventas de forma simultánea:** el programa sería siempre más preciso que un humano a la hora de realizar entradas y salidas al mercado. Además, puede hacer esto de forma simultánea y operar para distintos mercados financieros a la vez.
- **Posibilidad de aplicar aprendizaje automático:** no sólo podemos centrarnos en desarrollar un algorítmico basado en reglas o análisis del mercado actual sino que también es posible entrenar algoritmos con modelos usando datos históricos de los mercados financieros. Aquí destaca el uso de herramientas de BI y Big Data.

1.2.2. Desventajas

- **Dependencia de noticias:** como se ha mencionado en anteriores apartados de esta memoria, el trading algorítmico surge principalmente del análisis técnico. Aquí destacamos una desventaja del mismo. En el trading algorítmico es muy difícil tener en cuenta noticias que afecten o puedan afectar al comportamiento del mercado en cuestión. Si se decide implementar lógica para estudiar noticias, la complejidad aumenta bastante.
- **Dificultad para detectar la acción del precio:** debido a que el análisis del trading algorítmico es mayoritariamente técnica y basado en cálculos matemáticos, estadística, etc, es complejo detectar la existencia de patrones en el gráfico de precios. Esto sí es más sencillo de realizar por parte de un trader. Entre estos patrones podemos ver soportes, resistencias, líneas de tendencia, niveles, etc. En resumen, es complicado programar una detección de patrones ya que no es algo numérico sino que el operador los identifica a simple vista.
- **Complejidad en la programación:** es realmente complejo programar un algoritmo de trading.

1.3. Objetivos

El objetivo principal del proyecto es desarrollar una aplicación web para realizar trading algorítmico basándose en técnicas de análisis de mercados financieros clásicas. La aplicación se desarrollará en el lenguaje de programación *Python* y usando el framework *Django*.

El desarrollo principal de la aplicación se encontrará en la capacidad para comprar y vender de forma automática usando una cuenta comercial real tal y como lo haría un usuario humano. Estas operaciones se realizarán según lo indique el algoritmo que elijamos, dentro de una lista de algoritmos que encontramos en la propia aplicación.

Para hacer uso de la aplicación, primero deberemos iniciar sesión en la misma, con un usuario y contraseñas que conocemos de antemano y que nos proporciona el administrador. La aplicación permitirá al usuario identificarse con su cuenta comercial o demo (de prueba) de *MetaTrader5*, que será la aplicación externa que realizará las compras y ventas en el mercado financiero seleccionado. **Ref.: OBJ 1, OBJ 2.**

Una vez un usuario está identificado y ha iniciado sesión en *MT5*, podrá escoger un algoritmo para realizar Trading automático en tiempo real o probar las técnicas a modo de Backtesting. La aplicación también proporcionará la posibilidad de ver el histórico de operaciones realizado y el balance actual de la cuenta de *MT5* en la que se ha identificado el usuario. **Ref.: OBJ 5, OBJ 6, OBJ 7, OBJ 8.**

Además de poder realizar operaciones de manera automática, la aplicación dispondrá de una interfaz propia para ver los datos de mercado en tiempo real o antiguos, utilizando gráficas interactivas. **Ref.: OBJ 3, OBJ 4.**

Más formalmente, podemos definir los objetivos del producto software de la siguiente forma:

- **OBJ 1:** La aplicación tendrá un sistema de gestión de usuarios.
- **OBJ 2:** El sistema conectará con la cuenta del usuario de la plataforma de trading en cuestión.
- **OBJ 3:** El sistema permitirá a los usuarios ver gráficos en tiempo real del mercado que se quiera visualizar.
- **OBJ 4:** El sistema permitirá a los usuarios ver gráficos de datos antiguos de precios del mercado que se quiera visualizar.
- **OBJ 5:** El sistema desarrollará varios algoritmos usados para predecir el comportamiento de los mercados y hacer compras o ventas. La aplicación permitirá a los usuarios elegir entre uno de estos algoritmos para ser usado en el resto de funciones de la APP.

- **OBJ 6:** El sistema permitirá a los usuarios hacer operaciones de compra y venta de manera automatizada en un periodo de tiempo y mercado concretos, eligiendo los modelos de predicción mencionados.
- **OBJ 7:** El sistema permitirá a los usuarios probar cada uno de los algoritmos en un periodo de tiempo fijo, a modo de backtesting.
- **OBJ 8:** El sistema permitirá a los usuarios ver un histórico de operaciones realizadas así como el balance actual de la cuenta a la que se ha conectado.

Capítulo 2

Planificación

2.1. Planificación temporal

2.1.1. Fases del proyecto

En este apartado se mencionan cada una de las fases en las que se ha dividido el desarrollo del proyecto. Con cada fase se incluye una explicación de los temas desarrollados. En el siguiente apartado de la sección se encuentra el *diagrama de Gantt* correspondiente.

Fase inicial o de investigación

En cuanto a la primera fase del desarrollo del proyecto, se comenzó con una investigación teórica sobre lo que es el trading, trading algorítmico y demás conceptos de economía que entran en juego en el tema. Aquí se realizó también una investigación sobre dos de los principales referentes del análisis de mercados: *Charles Dow* y *Richard Wyckoff*.

En esta fase también se hicieron programas de ejemplo usando las distintas *APIs* de conocidas plataformas de trading como son *MetaTrader* o *Binance* (específica para criptomonedas).

Durante este período también se estuvo estudiando las posibles herramientas a usar para el proyecto. Se siguieron tutoriales de *Django* ya que se decidió que la APP fuera implementada usando este framework y por tanto, programada con *Python*. También se estudió cómo usar las herramientas necesarias para el proyecto en *Windows*, ya que librerías como la de *MetaTrader5* para *Python* sólo están disponibles en este sistema operativo.

Podemos llamar a esta fase inicial o de investigación y ocurrió desde la propuesta del proyecto, noviembre de 2020, hasta marzo de 2021.

Fase de recopilación de datos

En esta segunda fase de desarrollo del proyecto, y tras varias charlas con el tutor del mismo, el Prof. José Manuel Benítez, se decidió investigar qué volumen de datos históricos de mercados financieros podríamos recopilar de internet u otras fuentes. Esta acción era necesaria previa al análisis de lo que iba a ser el producto software, ya que en todo caso, se necesitaría cierto volumen de datos para poner a funcionar los algoritmos de trading, ya fuesen de análisis puro, o de aprendizaje automático.

En la primera aproximación, se implementaron una serie de scripts en *Python* para obtener datos de mercados a través de *MT5*. Con estos scripts, se consiguió un volumen de datos de 1 ó 2 años atrás para 179 mercados financieros distintos.

Debido a la clara limitación en espacio que esto suponía y tras acordar que lo ideal sería un volumen de datos más grande aún suponiendo limitar el número de mercados financieros a tratar, se decidió finalmente obtener un mayor volumen de datos de 5 mercados diferentes entre sí.

Esta fase de recopilación de datos históricos supuso desde marzo hasta mayo de 2021.

Fase de análisis

En la fase de análisis, se realiza la especificación de requisitos funcionales, no funcionales y de información. En esta fase también se describen los implicados en la aplicación.

Esta fase transcurre entre la segunda y la tercera semana del mes de mayo de 2021.

Fase de diseño

En la cuarta fase del desarrollo del proyecto, se propone el diseño a alto nivel de la aplicación. En esta fase de diseño, se estudia la arquitectura del software a desarrollar con un diagrama de paquetes o módulos en los que se dividirá la aplicación.

En este punto se propone también el primer boceto de diseño de la interfaz presente en la aplicación web.

Esta fase transcurre entre la tercera y la cuarta semana del mes de mayo de 2021.

Fase de desarrollo

En la fase de desarrollo de código del proyecto es donde se comienza con la implementación del producto software.

Este período se describe de manera más detallada en el capítulo 5. En dicho capítulo se explica la metodología ágil usada para el seguimiento del desarrollo y cada unas de las issues de *GitHub* creadas para implementar la aplicación.

Esta fase se desarrolla desde junio hasta la tercera semana de agosto de 2021.

Fase de pruebas

La última parte del proyecto recae en una fase de pruebas. En esta fase se realizan pruebas y se incluyen posibles fixes de última hora para bugs encontrados. Esta fase también sirve como retroalimentación para las conclusiones y para ver algunos resultados del algoritmo en funcionamiento.

Este último período del proyecto ocurre en la última semana de agosto y primera de septiembre de 2021.

2.1.2. Diagrama de Gantt

A continuación se muestra el diagrama de Gantt correspondiente con las fases del proyecto anteriormente mencionadas.

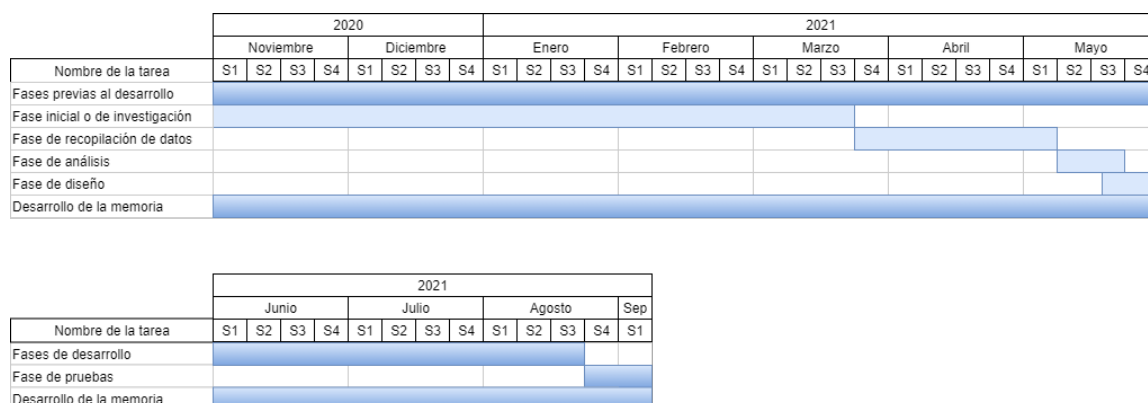


Figura 2.1: Diagrama de Gantt del proyecto

2.2. Presupuesto

Capítulo 3

Análisis

3.1. Descripción de los implicados

En esta aplicación, destacamos dos principales implicados: el administrador de la aplicación y el usuario final de la aplicación.

- **Desarrollador y administrador del sistema:** La responsabilidad del implicado será la de realizar las distintas actividades de desarrollo de la aplicación: corregir errores o añadir nuevas features, entre otras actualizaciones, que garantizan el correcto funcionamiento del sistema y su mantenimiento. Se encargará también de las tareas de gestión de la base de datos como puede ser añadir nuevos usuarios.
- **Usuario de la aplicación:** Este implicado representa al cliente que usa la aplicación. Este implicado hace uso de la aplicación como usuario final.

3.2. Especificación de requisitos

3.2.1. Requisitos Funcionales

Descripción de los requisitos más importantes a nivel de funciones que debe incluir el sistema, realizando una clasificación en categorías, a cada uno de los requisitos se le ha asignado un código y un nombre, con el fin de identificarlos fácilmente a lo largo de todo el proyecto.

- **RF-1. Iniciar sesión en la aplicación.** El usuario de la aplicación deberá iniciar sesión en la APP para hacer uso de la misma. El sistema por tanto incluirá un **sistema de gestión de usuarios**.
- **RF-2. Gestiones de la plataforma de trading.**

- **RF-2.1. Login en la plataforma de trading.** El usuario de la aplicación podrá conectarse con su cuenta de trading, comercial o demo, para poder hacer el uso completo de la APP.
- **RF-2.2. Ver capital disponible.** El usuario podrá ver el capital disponible en su cuenta de trading.
- **RF-2.3. El usuario podrá ver información de las operaciones** realizadas y de operaciones que en ese momento aún no se han cerrado.
- **RF-3. Visualización de datos.** El usuario de la aplicación podrá ver información de precios de un mercado financiero específico.
 - **RF-3.1. Ver datos de mercado en rango de tiempo específico.** El usuario de la aplicación podrá ver información de precios entre dos fechas específicas.
 - **RF-3.2. Ver datos de mercado con un marco de tiempo específico en tiempo real.** El usuario de la aplicación podrá ver información de precios en tiempo real con un marco de tiempo específico.
- **RF-4. El usuario podrá elegir un modelo y realizar trading algorítmico.** También podrá parametrizarlo según modelo y elegir tiempo en el que quiere dejar haciendo las operaciones automáticas.
- **RF-5. El usuario podrá elegir un modelo y realizar trading algorítmico a modo de backtesting.** De esta forma podrá probar cada uno de los modelos en un mercado y periodo de tiempo prefijados.

3.2.2. Requisitos No Funcionales

- **RNF-1.** La plataforma de trading que usará la aplicación será *MetaTrader5*.
- **RNF-2.** La aplicación permitirá el uso de cualquier bróker aceptado por *MetaTrader5*.
- **RNF-3.** Para la visualización de datos, el usuario podrá elegir un marco de tiempo de entre m1, m3, m5, m15, m30 ó m45; h1, h2, h3 ó h4; d1 (minutos, horas o días, respectivamente).
- **RNF-4.** La aplicación responderá a las peticiones de los usuarios en un tiempo determinado, mostrando un aviso de error si el tiempo de respuesta es superior al establecido.
- **RNF-5.** La aplicación deberá funcionar computadoras mediante navegador web.

- **RNF-6.** Para la implementación de la aplicación, se utilizará *Python* y su framework *Django*.
- **RNF-7.** La interfaz debe ser sencilla e intuitiva.

3.2.3. Requisitos de Información

- **RI-1.** El sistema gestor de bases de datos utilizado será *SQLite3*.

3.3. Diagramas de casos de uso

En esta sección, muestro los distintos diagramas de cada caso de uso que implementa la aplicación. Con estos diagramas se completa la fase de análisis del proyecto y con ellos vemos los roles que tienen los implicados y cómo interactúan con el sistema. Véase punto 3.1, descripción de actores o implicados.

3.3.1. Gestión de sesiones de la APP

En el diagrama 3.1 se representan los casos de uso referentes al sistema de gestión de usuarios de la aplicación.

El usuario de la aplicación deberá identificarse y cerrar sesión en *TradingAPP*. Estas acciones de Login y Logout interactúan internamente con el módulo de *Django*, que es el que implementa el sistema de sesiones.

A su vez, el administrador de la APP será quién administre los usuarios, a través del módulo de *Django*.

3.3.2. Gestiones de la plataforma de trading (MT5)

En el diagrama 3.2 se representan los casos de uso referentes a la gestión de la plataforma de trading. Como ya hemos mencionado en los requisitos no funcionales y en otras ocasiones, esta plataforma será *MetaTrader5*. Se mencionará *MetaTrader5* a lo largo del proyecto con la abreviatura *MT5*.

El usuario deberá identificarse con su cuenta, contraseña y servidor en la plataforma en cuestión, MT5. La cuenta podrá ser comercial o demo; y podrá ser creada por cualquier bróker aceptado en MT5. A este caso de uso se le suma el de desconectarse de la cuenta en la que el usuario se ha identificado previamente.

El usuario podrá ver el capital disponible y las operaciones realizadas y en curso. Estos dos casos de uso interactúan internamente con MT5, que

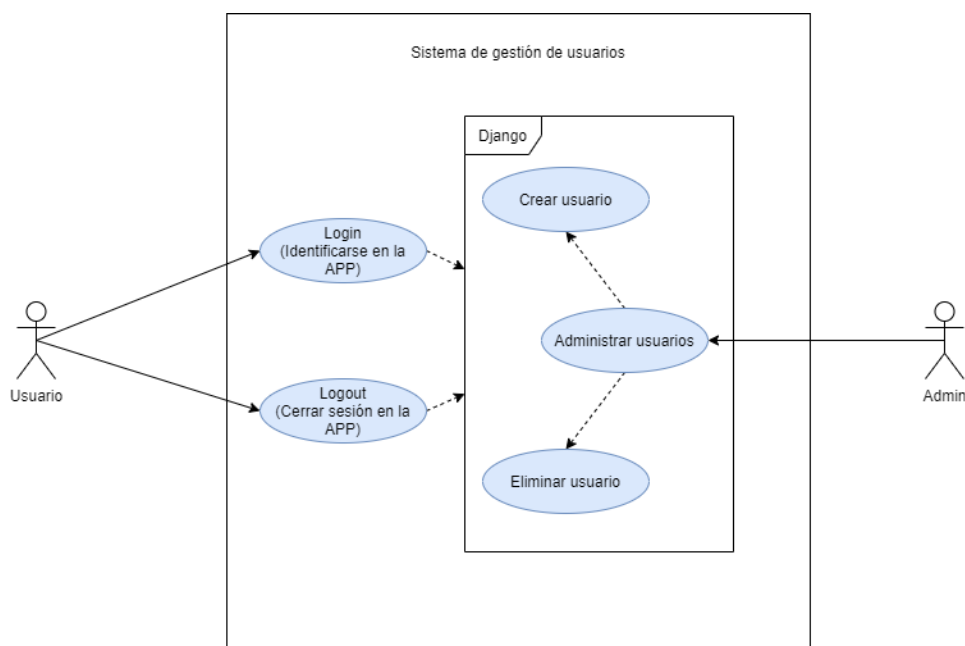


Figura 3.1: Diagrama de casos de uso referente al sistema de gestión de usuarios de la APP

es el módulo que nos provee dicha información. La previa identificación del usuario con su cuenta de MT5 será obligatoria para los otros casos de uso.

3.3.3. Visualización de datos

El diagrama 3.3 muestra los casos de uso referentes a la visualización de datos de mercados financieros. Dichos datos serán mostrados mediante un gráfico que muestra precios por medio de velas siguiendo el formato *OHLC*.

El usuario podrá ver datos históricos de mercado en un rango de tiempo específico. Este caso de uso interactúa con los datos proporcionados previamente por el usuario.

El usuario de la APP podrá ver datos en tiempo real con un marco de tiempo específico, es decir, con velas generadas cada minuto, cada hora, cada día, etc. Este *CU* usa el módulo de *MT5* para obtener los datos y luego procesa dichos datos para ajustarlos al formato *OHLC*.

3.3.4. Funcionalidades de trading

Finalmente, el diagrama 3.4 representa los casos de uso referentes al módulo de trading y elección de algoritmos.

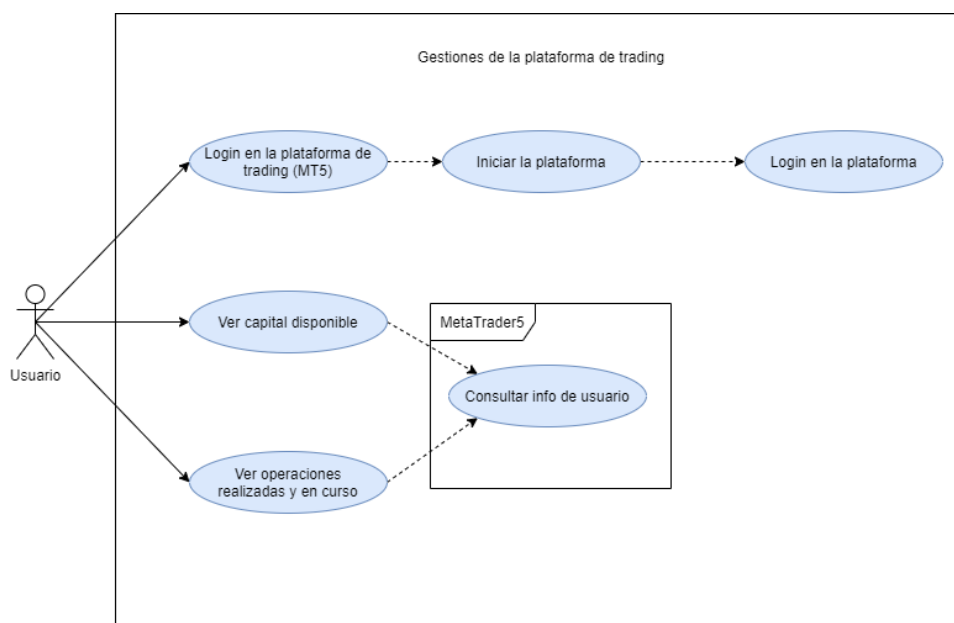


Figura 3.2: Diagrama de casos de uso referente a la gestión de la plataforma de trading (MT5)

El usuario podrá elegir un algoritmo de trading. A la hora de operar, el usuario podrá usar el algoritmo elegido para operar en tiempo real o realizar backtesting. El usuario también podrá operar manualmente, siguiendo su propio análisis en tiempo real, sin algoritmo.

Para los casos de uso de operar manualmente o de forma automática en tiempo real, la APP se conectará al módulo de MT5 para ordenar las operaciones de compra o venta.

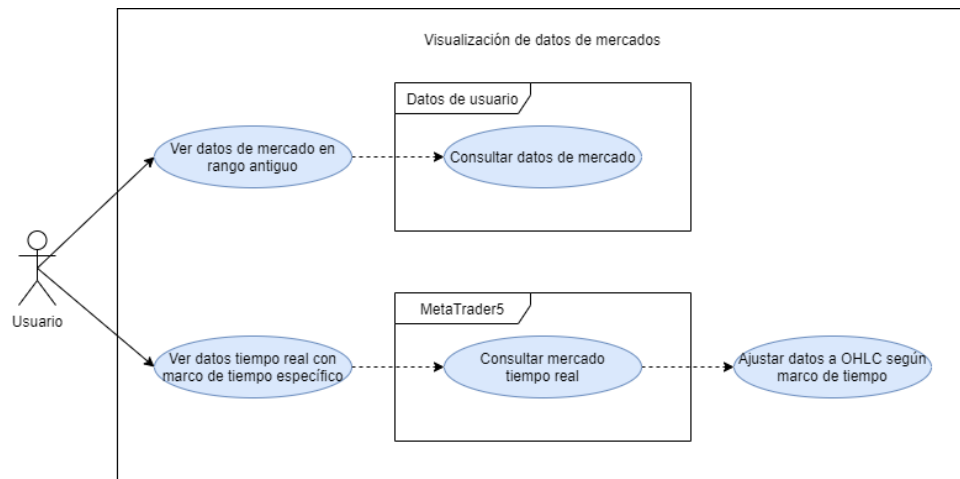


Figura 3.3: Diagrama de casos de uso referente a la visualización de datos de mercados financieros

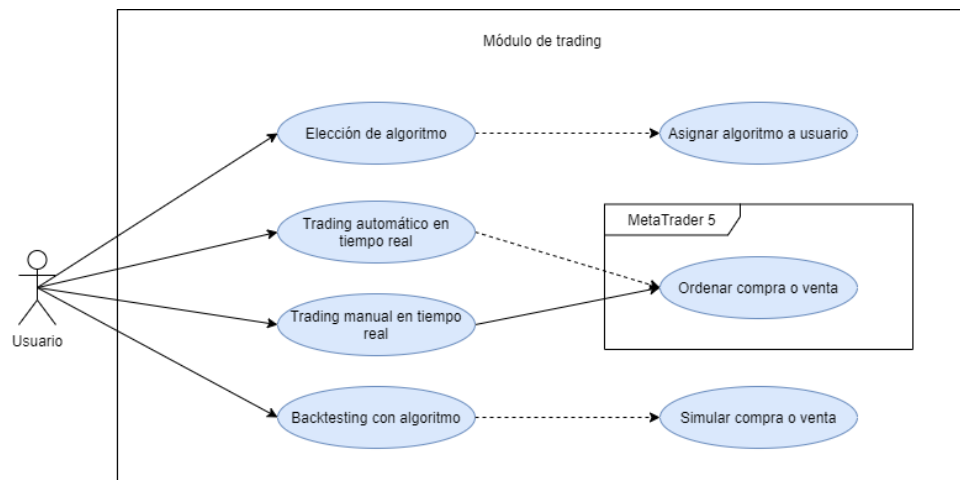


Figura 3.4: Diagrama de casos de uso referente a las funcionalidades de trading

Capítulo 4

Diseño

4.1. Arquitectura del software

A continuación muestro un esquema que representa la arquitectura general del software.

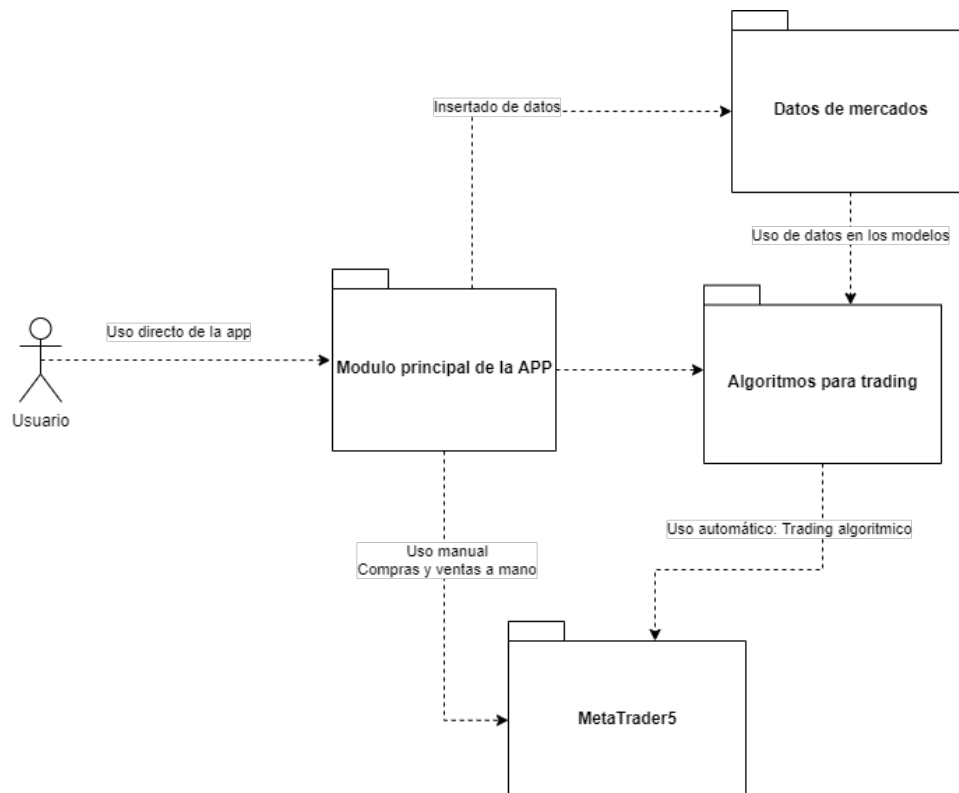


Figura 4.1: Arquitectura general del software

Como resumen, en el esquema podemos ver cuál es la idea principal del proyecto. El usuario de la aplicación (véase punto 3.1, segundo implicado) será el que haga un uso directo de la interfaz de la aplicación. A continuación presento cada uno de los módulos con sus funciones y demás utilidades:

- **Módulo principal de la APP:** este módulo se corresponderá con el controlador principal del software. Será usado por el usuario a través de una interfaz escrita en Django y se comunicará con el resto de módulos directamente.
- **Datos de mercados:** este módulo se corresponderá con la base de datos y las operaciones que hagamos con dichos datos (insertado de datos, adaptaciones a cada algoritmo, etc.) Será usado por la interfaz principal de la aplicación y mandará los inputs a los algoritmos.
- **Algoritmos para trading o backend:** se corresponde con el backend o código fuente de la aplicación. Aquí se encontrará cada uno de los algoritmos que usemos para trading algorítmico. Recibe datos de la BBDD y se comunica con el módulo de MetaTrader5 para indicar cada una de las operaciones decididas y obtener información en tiempo real.
- **MetaTrader5:** Módulo externo de la aplicación, se usará para mandar órdenes de operaciones y recibir resultados e información en tiempo real.

El software consistirá en una aplicación web con código escrito en Python, usando Django como Framework para la interfaz.

4.2. Primer diseño de la interfaz

La página web principal mostrará un menú con distintos botones que implementan funcionalidades diferentes. Previo a este menú tendremos un formulario para identificarnos como usuarios de la aplicación. A ambos lados del menú principal y durante todas las vistas de la APP tendremos hiperenlaces con accesos directos a distintas funcionalidades de la aplicación. A continuación muestro una serie de borradores con diseños de cada una de las secciones de la interfaz.

En este primer diseño que muestro en la siguiente imagen, se puede ver el menú principal de la aplicación.

En la imagen mostrada vemos los siguientes botones: *Login*, *Logout*, *Gestión de datos*, *Mi perfil*, *Ver datos de mercados*, *Operar*, *Estrategias de trading*, *Backtesting*, *Histórico de operaciones* y *Balance y operaciones en curso*. En los puntos siguientes explico qué realizaría cada botón y con qué

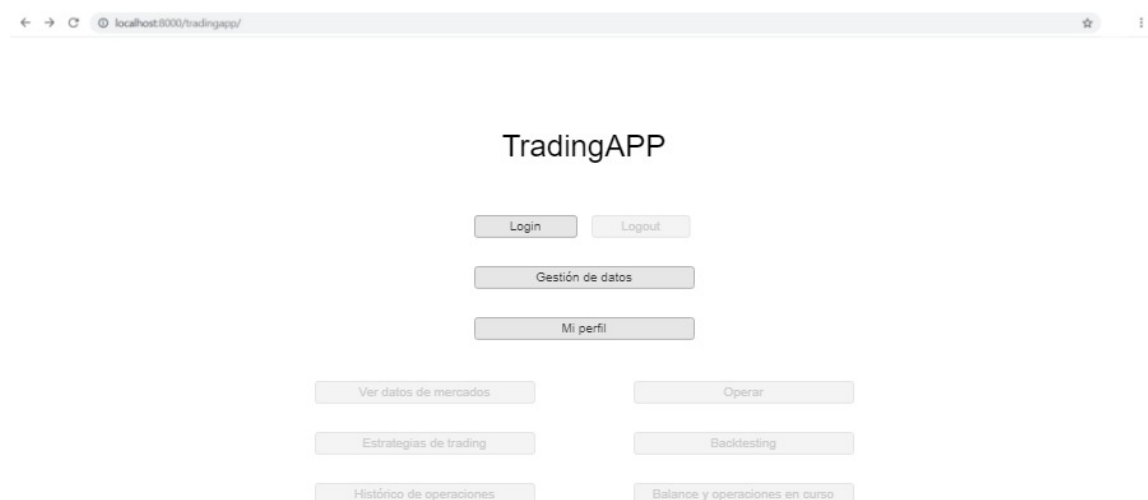


Figura 4.2: Primer diseño del menú principal de la APP

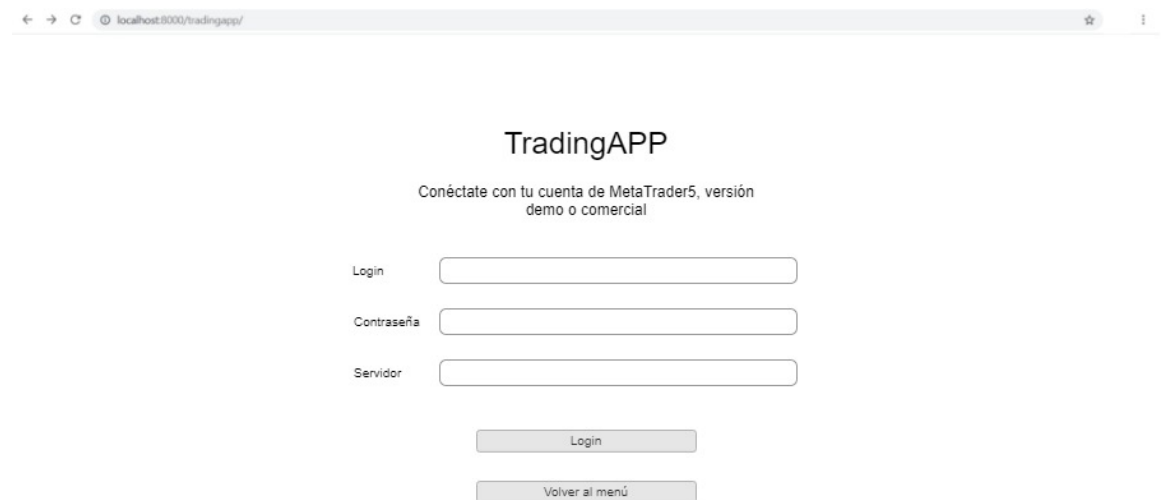
parte del software conectaría.

- **Login:** permite conectar a nuestra cuenta de MetaTrader5, comercial o demo. Véase la siguiente figura. *Referencia: login, RF-1.1.*
- **Logout:** permite cerrar la sesión iniciada anteriormente con el menú de Login. Este botón no estará habilitado hasta que hayamos iniciado sesión. *Referencia: login, RF-1.1.*
- **Gestión de datos:** permite el insertado y procesado de datos de mercado. El usuario podrá a través del menú correspondiente introducir datos al módulo que se encarga de gestionar la BBDD. Al ser un módulo totalmente independiente del módulo de MetaTrader5, no será necesario haber iniciado sesión anteriormente. *Referencia: procesado de datos, RF-2.*
- **Mi perfil:** permite indicar la información del bróker usado. En este menú indicamos información sobre comisiones u otros parámetros.
- **Ver datos de mercados:** permite ver datos de precios de un mercado elegido. Se permite ver datos en un marco de tiempo específico, en rango y de mercados específicos. *Referencia: visualización de datos, RF-3.*
- **Operar:** lleva al usuario a un menú donde podrá operar manualmente, es decir, gestionando las operaciones abiertas y/o abriendo nuevas; o

usando el trading algorítmico. **Referencia:** *operar manualmente, RF-4; trading algorítmico, RF-5.*

- **Estrategias de trading:** lleva a un menú en el que se puede obtener información de cada una de las estrategias de trading así como elegir una para que sea usada en las próximas operaciones. **Referencia:** *trading algorítmico, RF-5.*
- **Backtesting:** menú similar a Operar. En este caso el usuario podrá iniciar operaciones manualmente o usando trading algorítmico a partir de una fecha anterior a la actual, simulando el transcurso del tiempo. **Referencia:** *backtesting, RF-6.*
- **Histórico de operaciones:** permite al usuario ver un resumen del histórico de operaciones realizadas. **Referencia:** *información de operaciones, RF-1.3.*
- **Balance y operaciones en curso:** permite al usuario ver el capital disponible, así como un balance a tiempo real de pérdidas y ganancias y el resumen de las operaciones en curso. **Referencia:** *capital disponible, RF-1.2; información de operaciones, RF-1.3.*

A continuación muestro una imagen de lo que sería el menú de **Login**. En este menú se introducirán Login, Contraseña y Servidor para que nuestra aplicación inicie sesión en la cuenta demo o comercial de MetaTrader5.



TradingAPP

Conéctate con tu cuenta de MetaTrader5, versión demo o comercial

Login

Contraseña

Servidor

Login

Volver al menú

Figura 4.3: Primer diseño del formulario de login en MT5

Una vez estemos conectados en la cuenta de MT5, nuestro menú principal mostrará disponibles las funcionalidades que en la figura 4.2 no estaban habilitadas. Dichas funcionalidades dependen directamente del módulo de MT5, por lo que necesitan de conexión a la cuenta.

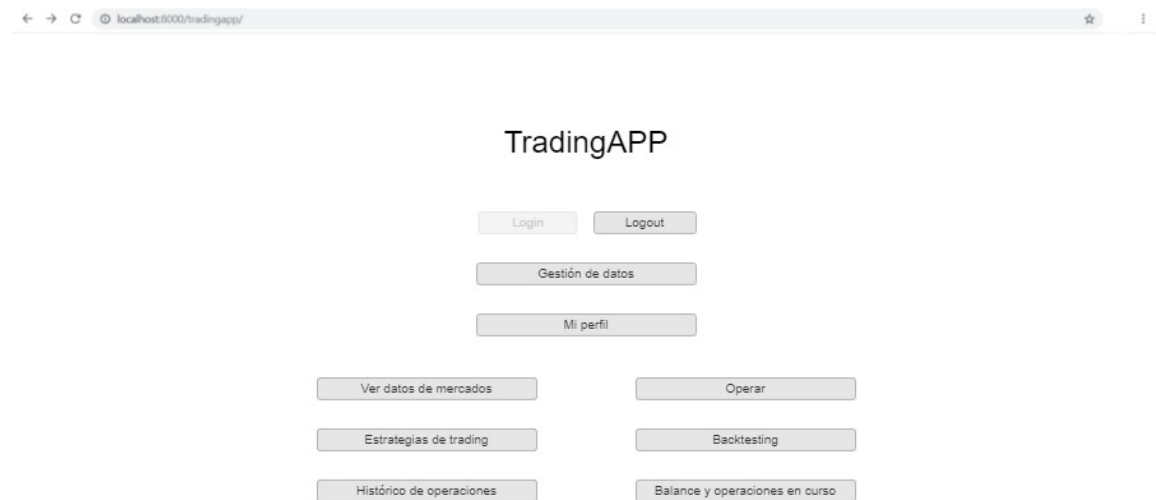


Figura 4.4: Primer diseño del menú principal (usuario identificado en *MT5*)

Capítulo 5

Implementación

Capítulo 6

Pruebas

Capítulo 7

Conclusiones

