Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Computación

Aplicación web para la gestión de carteras de inversión usando técnicas de Inteligencia Artificial

Autor: Víctor Arranz Barcenilla

Tutor: Valentín Cardeñoso Payo

«Una mente necesita de los libros igual que una espada de una piedra de amolar.»

Tyrion Lannister

Agradecimientos

A mi tutor Valentín, por estar siempre disponible y dispuesto a ayudarme cuando lo he requerido, así como por aconsejarme y guiarme en aquellas partes en las que andaba más perdido. A los compañeros que he conocido durante estos años y que han pasado a ser mis amigos, por acompañarme y ayudarme en este viaje por la universidad. A mis amigos de siempre, por permanecer a mi lado. Y, por encima de todos, a mis padres, por dedicarme su vida y aguantarme en mis peores días.

Resumen

En este trabajo se desarrolla una aplicación web que aborda fundamentalmente los problemas de la optimización de carteras de inversión y de la predicción de valores de activos financieros, con el doble objetivo de proporcionar una herramienta que permita entender y comprender los principales aspectos de las técnicas utilizadas, así como ayudar al inversor en su toma de decisiones.

Se utilizan datos pertenecientes a algunos de los principales mercados financieros del mundo, con un total de 840 activos: bolsa, con los índices S&P 500, NASDAQ 100, IBEX 35 y EURO STOXX 50 junto a las empresas asociadas a ellos, criptodivisas, divisas y materias primas.

La aplicación web construida es completamente interactiva, permitiendo en todo momento al usuario elegir con qué datos desea trabajar y configurar libremente los principales parámetros de los elementos que componen las distintas secciones de la misma.

En primer lugar, la web proporciona herramientas para realizar un análisis descriptivo con herramientas propias de la Estadística aplicadas al análisis financiero, como las medias móviles o distintos indicadores técnicos. Se proporcionan distintos tipos de visualizaciones que permiten llevar a cabo el estudio de la serie temporal de un valor. A su vez, se incluyen secciones descriptivas que contienen *dashboards* con los precios de los activos de los distintos mercados considerados en el trabajo.

Desde el punto de vista de la optimización de carteras, se trabaja con el modelo de Markowitz construyendo carteras eficientes, carteras notables y realizando representaciones gráficas de las mismas, así como de la frontera eficiente. También se permite realizar análisis de sensibilidad con algunos importantes parámetros.

Relacionado con este problema también se construyen regresiones con los índices bursátiles para abordar el estudio del riesgo de las carteras obtenidas con el modelo de Markowitz.

En cuanto a la predicción de valores de los activos, se utilizan técnicas de *machine learning*, que van desde los métodos más empleados como las máquinas de vectores soporte o los ensembles de árboles de aleatorios, pasando por distintos modelos de redes neuronales englobados dentro de lo que se conoce como *deep learning*, como las redes recurrentes o las redes convolucionales. Para la construcción de estos modelos de inteligencia artificial se utiliza la librería *Keras*, que permite el uso de *Tensorflow* con el lenguaje *Python*.

Un último punto explorado, también en relación con el aprendizaje automático, es el análisis *cluster*, utilizado para tratar de buscar grupos entre activos con un comportamiento semejante en términos de rentabilidad.

Para la implementación de la aplicación se han utilizado herramientas de programación web como *html*, *css* y *javascript*, mientras que la totalidad de elementos interactivos (a excepción del menú de navegación de la web) que contienen las distintas secciones, así como las visualizaciones que contienen éstas, se han desarrollado utilizando la librería *Shiny* del lenguaje *R*, que a su vez se vale del *solver* de optimización *AMPL* para tratar ese problema y de la librería *reticulate* para enlazar con *Python* y poder utilizar los códigos de los modelos de aprendizaje automático comentados. *R* a su vez ha sido el lenguaje utilizado en la captación de datos a través de la API de *Yahoo Finance* y para el procesamiento y tratamiento de los mismos.

La web se desarrolló sobre una imagen *Docker*, utilizando una metodología ágil consistente en un *Scrum* adaptado.

Palabras clave: optimización, cartera de inversión, modelo de Markowitz, inversión, *machine learning*, *deep learning*, web, *R*, *shiny*, *Python*.

Abstract

This work develops a web application that fundamentally addresses the problems of investment portfolio optimization and financial asset value prediction, with the dual objective of providing a tool to understand and comprehend the main aspects of the techniques used, as well as to help the investor in his decision making.

The data used belong to some of the world's main financial markets, with a total of 840 assets: stock markets such as S&P 500, NASDAQ 100, IBEX 35 and EURO STOXX 50 indexes together with the companies associated with them, cryptocurrencies, currencies and commodities.

The web application is fully interactive, allowing the user to choose at any time which data to work with and to freely configure the main parameters of the elements that make up the different sections of the application.

Firstly, the website provides tools to perform a descriptive analysis with statistical tools applied to financial analysis, such as moving averages or different technical indicators. Different types of visualizations are provided to allow the study of the time series of an asset. At the same time, descriptive sections containing dashboards with the prices of the assets of the different markets considered in the work are included.

From the point of view of portfolio optimization, we work with the Markowitz model, constructing efficient portfolios, notable portfolios and making graphical representations of them, as well as of the efficient frontier. It is also possible to perform sensitivity analysis with some important parameters.

Related to this problem, regressions with stock market indexes are also constructed to address the study of the risk of the portfolios obtained with the Markowitz model.

As for the prediction of asset values, machine learning techniques are used, ranging from the most commonly used methods such as support vector machines or ensembles of random trees, to different neural network models included in what is known as deep learning, such as recurrent networks or convolutional networks. For the construction of these artificial intelligence models, the *Keras* library is used, which allows the use of *Tensorflow* with the *Python* language.

A final point explored, also in relation to machine learning, is cluster analysis, used to try to find groups among assets with similar behavior in terms of profitability.

Web programming tools such as *html*, *css* and *javascript* have been used to implement the application, while all the interactive elements (with the exception of the web navigation menu) contained in the different sections, as well as the visualizations they contain, have been developed using the *Shiny* library of the *R* language, which in turn uses the *AMPL* optimization solver to deal with this problem and the reticulate library to link with *Python* and use the codes of the machine learning models mentioned above. *R* in turn has been the language used to capture data through the Yahoo Finance API and to process and treat them.

The website was developed on a *Docker* image, using an agile methodology consisting of an adapted *Scrum*.

Keywords :optimization, investment portfolio, Markowitz model, investment, machine learning, deep learning, web, *R*, *shiny*, *Python*.

Índice general

Índice de cuadros v

Índice de figuras ix

I Objeto, Concepto y Método 1

1. Introducción 3
1.1. Introducción . . . 3
1.2. Motivación . . . 4
1.3. Objetivos . . . 5
1.4. Estructura . . . 6

2. Planificación 9
2.1. Metodología . . . 9
2.2. Actividades . . . 10
2.3. Recursos . . . 11
2.4. Planificación inicial . . . 11
2.5. Fases y costes . . . 11
2.6. Presupuesto inicial . . . 13
2.7. Desviaciones de la planificación inicial . . . 14
2.8. Coste final . . . 14
2.9. Análisis de riesgos . . . 15

II Marco Conceptual y Contextual 19

3. Marco Conceptual 21
3.1. Introducción económica . . . 21
3.2. Análisis de datos de activos . . . 23
3.2.1. Análisis fundamental . . . 23
3.2.2. Análisis técnico . . . 24
3.3. Bolsa . . . 35
3.3.1. S&P 500 . . . 37
3.3.2. NASDAQ . . . 38
3.3.3. IBEX 35 . . . 38
3.3.4. EURO STOXX 50 . . . 39
3.4. Criptodivisas . . . 39

3.4.1. Blockchain . . . 39
3.4.2. Bitcoin . . . 40
3.4.3. Otras criptodivisas . . . 41
3.5. Materias primas . . . 41
3.5.1. Oro . . . 42
3.5.2. Otras materias primas . . . 42
3.6. Divisas . . . 42
3.7. Optimización de carteras . . . 43
3.7.1. Modelo básico . . . 44
3.7.2. Modelo de Markowitz . . . 45
3.7.3. Modificaciones al modelo de Markowitz . . . 46
3.7.4. Riesgo específico y riesgo sistemático . . . 47
3.7.5. Simulación de carteras con método de Montecarlo . . . 48
3.8. Predicción de valores . . . 49
3.8.1. Estado del arte . . . 50
3.8.2. Modelos . . . 52
3.8.3. Metodología experimental . . . 52
3.8.4. Variables . . . 54
3.8.5. Preparación de datos de series temporales para regresión . . . 55
3.8.6. Modelo ARIMA . . . 55
3.8.7. Modelo Support Vector Regressor . . . 57
3.8.8. Modelo Random Forest . . . 59
3.8.9. Modelo XGBoost . . . 60
3.8.10. Modelo MLP . . . 61
3.8.11. Modelo Simple RNN . . . 66
3.8.12. Modelo LSTM . . . 67
3.8.13. Modelo GRU . . . 69
3.8.14. Modelo CNN . . . 69
3.9. Otras técnicas . . . 72
3.9.1. Clustering . . . 72

III Desarrollo del Sistema **77**

4. Fuentes de datos **79**

5. Análisis **81**
5.1. Descripción del sistema . . . 81
5.2. Historias de usuario . . . 81
5.3. Elicitación de requisitos . . . 85
5.3.1. Requisitos generales . . . 86
5.3.2. Inicio . . . 86
5.3.3. Dashboard de precios . . . 86
5.3.4. Análisis de valores . . . 87
5.3.5. Optimización de carteras . . . 88
5.3.6. Requisitos de “Evaluación de carteras” . . . 89
5.3.7. Requisitos de “Regresión de índices” . . . 90
5.3.8. Requisitos de “Simulación de carteras” . . . 90
5.3.9. Requisitos de “Modelos de predicción” . . . 91

5.3.10. Requisitos de “Análisis de grupos” . . . 92
5.4. Casos de uso . . . 92

6. Diseño **95**
6.1. Arquitectura del sistema . . . 95
6.2. Diseño de casos de uso . . . 97
6.3. Diseño interfaz . . . 100
6.3.1. Mockups . . . 100
6.3.2. Elementos interfaz . . . 105

7. Implementación **107**
7.1. Despliegue del servidor . . . 107
7.1.1. Docker . . . 107
7.2. Herramientas de Desarrollo . . . 109
7.2.1. Extracción de datos . . . 109
7.2.2. Diseño de la Interfaz . . . 109
7.2.3. Frontend . . . 109
7.2.4. CSS . . . 110
7.2.5. Backend . . . 110
7.2.6. AMPL . . . 115
7.3. Análisis y diseño . . . 115
7.3.1. Visual Paradigm . . . 115
7.4. Otras herramientas . . . 115
7.4.1. Overleaf . . . 115
7.4.2. Gantt Project . . . 115
7.4.3. Gitlab . . . 116
7.5. Componentes de GUI . . . 116
7.5.1. Componentes shiny . . . 116
7.6. Secciones web . . . 117
7.6.1. Inicio . . . 117
7.6.2. Mercados . . . 117
7.6.3. Análisis de valores . . . 118
7.6.4. Optimización de carteras . . . 118
7.6.5. Evaluación de carteras . . . 118
7.6.6. Regresión de índices . . . 118
7.6.7. Simulación de carteras . . . 118
7.6.8. Modelos de predicción . . . 118
7.6.9. Análisis de grupos . . . 119
7.6.10. Ayuda . . . 119
7.6.11. Algoritmos . . . 119

8. Ejemplos de Uso **121**
8.1. Mercados . . . 121
8.2. Análisis de valores . . . 121
8.3. Optimización de carteras . . . 125
8.4. Evaluación de carteras . . . 130
8.5. Regresión de índices . . . 130
8.6. Simulación de carteras . . . 131
8.7. Modelos de predicción . . . 132

8.8. Análisis de grupos . . . 135
8.9. Ayuda . . . 136
8.10. Pruebas . . . 137

IV Discusión y Conclusiones **139**

9. Discusión y Ampliaciones **141**
9.1. Mejoras de la interfaz gráfica . . . 141
9.2. Mejoras del código . . . 142
9.3. Comparativa Python - R . . . 142
9.4. Evaluación del rendimiento de la web . . . 142
9.5. Registro de usuarios . . . 142
9.6. Modelo de *Black-Litterman* . . . 142
9.7. Nuevos modelos de IA . . . 143
9.8. Aumento de la personalización . . . 143
9.9. Profundizar en el tema de *clustering* . . . 143
9.10. Creación de un *bot* asistente . . . 143

10. Conclusiones **145**

Bibliografía **147**

Apéndices **157**

Apéndice A. Manual de Usuario **157**
A.1. Web . . . 157
A.2. Acciones del usuario . . . 157

Apéndice B. Manual del Desarrollador **159**
B.1. Máquina virtual . . . 159
B.1.1. Conexión al servidor a través de Visual Studio Code . . . 161
B.2. Diagramas de secuencia . . . 162

Índice de cuadros

2.1. Sprints de desarrollo del proyecto previstos. 12
2.2. Costes del proyecto. 14
2.3. Coste final del proyecto. 14
2.4. Tabla probabilidades Barry Boehm. Tomado de [12]. 15
2.5. Tabla impactos Barry Boehm. Tomado de [12] 15
2.6. Tabla de riesgos del proyecto. 16
2.7. Tabla de planes de prevención y contingencia de riesgos. 17

3.1. Ejemplo de serie temporal convertida a problema de aprendizaje supervisado. 55
3.2. Comparativa GRU y LSTM. 69

8.1. Ejemplo de pruebas de integración del sistema. 137

Índice de figuras

1.1. Taxonomía de las Ciencias de la Computación. Tomado de [1]. 4

2.1. Diagrama de tareas. 10
2.2. Diagrama de Gantt de actividades. 13

3.1. Esquema análisis fundamental. Tomado de [18]. 23
3.2. Indicadores análisis fundamental. Tomado de [18]. 24
3.3. Esquema análisis técnico. Tomado de [18]. 25
3.4. Indicadores análisis técnico. Tomado de [18]. 26
3.5. Media móvil. Tomado de [20]. 27
3.6. Bandas de Bollinger. Tomado de [21]. 27
3.7. Gráfico de barras EUR/USD Yahoo Finance 12/02/2022. 31
3.8. Descripción de una barra. Tomado de [31]. 32
3.9. Gráfico de líneas EUR/USD. Yahoo Finance. 12/02/2022. 33
3.10. Gráfico de velas S&P 500. Yahoo Finance. 12/02/2022. 33
3.11. Gráficos de tendencias. Tomado de [32]. 34
3.12. Distintas tendencias. Tomado de [29]. 34
3.13. Apoyos y resistencias en mercado bajista. Tomado de [29]. 35
3.14. Apoyos y resistencias en mercado alcista. Tomado de [29]. 35
3.15. Participantes de la bolsa de valores. Tomado de [15]. 36
3.16. Funcionamiento de la blockchain. Tomado de [43]. 40
3.17. Logotipo de Bitcoin. 41
3.18. Logotipo de las principales criptomonedas de la actualidad. 42
3.19. Representación de la frontera eficiente. Tomado de [55]. 46
3.20. Variaciones de los títulos o carteras en función de β. Tomado de [56]. 48
3.21. Ejemplo de obtención de la frontera eficiente con simulación por método de Montecarlo. Tomado de [58]. 49
3.22. Taxonomía de las principales técnicas de predicción de valores. Tomado de [19]. 50
3.23. Ejemplo de separación en clases con *kernel*. Tomado de [63]. 57
3.24. Hiperplano separante SVR. Tomado de [64] 58
3.25. Influencia del parámetro C en los modelos SVM. Tomado de [63] 59
3.26. Comparación de la neurona humana con la neurona de McCullot-Pitts. Tomado de [69]. 61
3.27. Propagación hacia delante y hacia atrás. Tomado de [72]. 63
3.28. Topología de un perceptrón multicapa don dos capas ocultas. Tomado de [65]. 64
3.29. Comparación de optimizadores en términos de coste. Tomado de [73]. 65
3.30. Desenrolle a través del tiempo de una red recurrente. Tomado de [77]. 66
3.31. Estructuras de redes recurrentes. Tomado de [78]. 67
3.32. Celda de una red LSTM. Tomado de [80]. 68

3.33. Celda de una red GRU. Tomado de [82]. 69
3.34. Operación de convolución (1). Tomado de [83]. 70
3.35. Operación de convolución (2). Tomado de [83]. 71
3.36. Operación de convolución (3). Tomado de [83]. 71
3.37. Ejemplo de *max pooling*. Tomado de [84] 71
3.38. Ejemplo de arquitectura de CNN ResNet. Tomado de [85]. 72
3.39. Taxonomía de los métodos de *clustering* . Tomado de [86]. 73
3.40. Ejemplo algoritmo K medias. Tomado de [87]. 74
3.41. Análisis de Componentes Principales. Tomado de [89]. 75

4.1. Pantalla de inicio de la web de Yahoo Finance. 80

5.1. Estructura de las historias de usuario. Tomado de [95]. 82
5.2. Diagrama de casos de uso. Realizado con 7.3.1. 94

6.1. Diagrama de despliegue. Realizado con 7.3.1. 96
6.2. Mockup de la pantalla inicial. 101
6.3. Mockup de la pantalla inicial con el menú de mercados desplegado. 102
6.4. Mockup de la pantalla de análisis de valores con el menú lateral desplegado. 103
6.5. Mockup de la pantalla de análisis de valores con el menú lateral oculto. 104
6.6. Mockup de la pantalla de ayuda. 105
6.7. Página de aterrizaje. 106
6.8. Menú de navegación plegado. 106
6.9. Menú de navegación con el submenú de mercados desplegado. 106

7.1. Arquitectura Docker. Tomado de [100]. 108
7.2. Shiny widgets. Tomado de [110]. 112
7.3. Shiny widgets. Tomado de [109]. 112
7.4. Arquitectura de la API de AMPL. Tomado de [113]. 113
7.5. Ejemplo de *select box*. 116
7.6. Ejemplo de *date input*. 116
7.7. Ejemplo de *slider input*. 117
7.8. Ejemplo de *numeric input*. 117
7.9. Ejemplo de *action button*. 117
7.10. Ejemplo de *radio buttons*. 117
7.11. Diagrama de flujo de la sección de modelos de predicción. Realizado con 7.3.1. 119
7.12. Diagrama de flujo de la sección de optimización de carteras. Realizado con 7.3.1. 120

8.1. Elementos shiny sección de mercados. 122
8.2. Gráfico de la serie temporal del índice S&P 500 de los últimos 3 meses. 122
8.3. Gráfico de velas japonesas del índice S&P 500 de los últimos 3 meses. 123
8.4. Gráfico de barras del índice S&P 500 de los últimos 3 meses. 123
8.5. Gráfico de la serie de rendimientos del índice S&P 500 de los últimos 3 meses. 124
8.6. Gráfico boxplot de rendimientos del índice S&P 500 de los últimos 3 meses. 124
8.7. Gráfico de la serie de disminuciones del índice S&P 500 de los últimos 3 meses. 124
8.8. Composición de las carteras notables. 125
8.9. Frontera eficiente. 126
8.10. Composición cartera eficiente. 127
8.11. Composición de carteras notables en diagrama de sectores. 127
8.12. Composición de carteras notables en diagrama de barras. 128

8.13. Comparación de carteras notables. 128
8.14. Gráfico en función de mu. 129
8.15. Evaluación de cartera óptima. 130
8.16. Regresión de la cartera con máximo ratio Sharpe frente al IBEX 35. 131
8.17. Simulación de carteras. 132
8.18. Descripción de los modelos ajustados. 133
8.19. Ajuste del modelo Random Forest. 134
8.20. Comparación de modelos. 134
8.21. Resultado análisis *cluster* 135
8.22. Extracto de la sección de ayuda. 136

B.1. Logotipo de Visual Studio Code. Tomado de [134]. 161

Parte I

Objeto, Concepto y Método

Capítulo 1

Introducción

1.1 Introducción

En la sociedad de hoy en día resulta de vital importancia comprender el funcionamiento del dinero y manejar conceptos como el ahorro o la inversión. Múltiples son las técnicas y teorías que se pueden utilizar para tratar de explicar el funcionamiento de los flujos financieros y muchos los *gurús* que pretenden tener la solución definitiva sobre cómo utilizar el dinero.

Con el creciente desarrollo de la computación y la informática de los últimos tiempos, se han desarrollado nuevos procedimientos de estudio para estos mercados. Es aquí donde aparece la *inteligencia artificial*, hoy en día en boca de todos, en artículos, publicaciones y noticias de cualquier tipo de temática. En este trabajo se utilizan técnicas englobadas en este campo, concretamente dentro de lo que se conoce como *machine learning* (aprendizaje automático) y como *deep learning* (aprendizaje profundo), convenientemente desarrolladas en su sección correspondiente. Una breve descripción de estos conceptos puede ser:

- **Inteligencia artificial (IA):** disciplina de las Ciencias de la Computación que busca construir máquinas que presenten las mismas capacidades que los seres humanos, es decir, que sean capaces de pensar.
- **Aprendizaje automático:** tipo de inteligencia artificial que proporciona a las computadoras la capacidad de aprender sin ser programadas explícitamente. Este aprendizaje se realiza a través de ejemplos.
- **Aprendizaje profundo:** combinación de sistemas de aprendizaje automático en una estructura de capas, habitualmente de redes neuronales.

Los diferentes modelos que surgen de la aplicación de estas técnicas tienen utilidad en prácticamente cualquier campo que podamos imaginar, desde el estudio de pandemias como la reciente del SARS-COV-2, estudios biológicos o genéticos, procesamiento del lenguaje, análisis de textos, estudio de imágenes, estudio de comportamiento de consumidores o clientes, y un largo etcétera. En el presente proyecto, se propone el uso de estos paradigmas en el estudio de las series temporales de los activos pertenecientes a distintos mercados, con la idea de poder modelizarlos y predecir su comportamiento, con la enorme dificultad que esto conlleva debido a su variabilidad.

Sin embargo, pese a utilizar técnicas modernas en el estudio de los mercados, no se pueden obviar las metodologías clásicas, basadas en el conocimiento financiero y apoyadas por las Matemáticas y, sobre todo, por la Estadística. Uno de los muchos intentos llevados a cabo para tratar de entender los mercados financieros, fue llevado a cabo por Harry Markowitz en [2], donde, a mediados de siglo XX, sentó las bases de un modelo

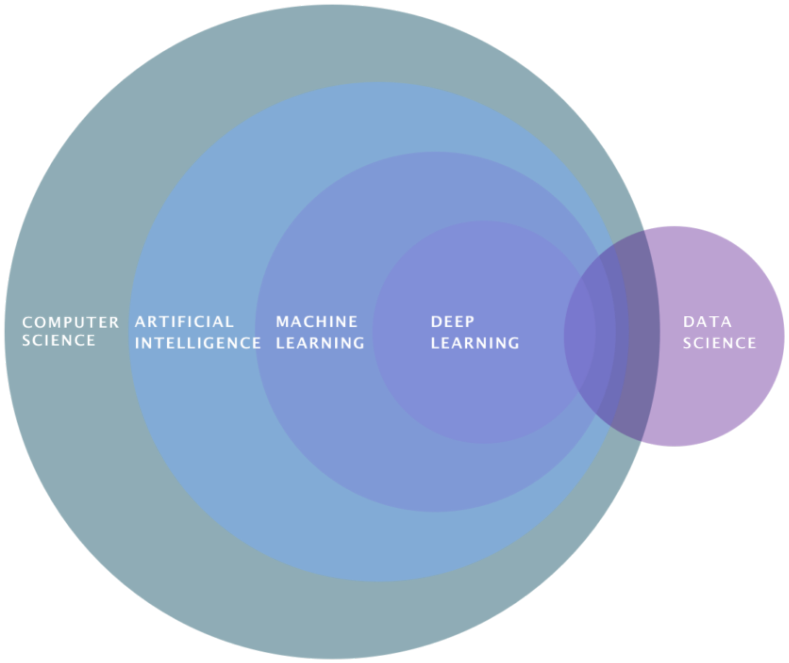


Figura 1.1: Taxonomía de las Ciencias de la Computación. Tomado de [1].

ampliamente utilizado y desarrollado hasta la actualidad. Este modelo, que pretende construir una cartera de inversión a partir de un conjunto de valores, mediante la diversificación del capital, se estudia en este trabajo utilizando como herramienta fundamental la Programación matemática. Este concepto hace referencia a un conjunto de técnicas del mundo de la Estadística, consistentes en la búsqueda del valor óptimo de una función denominada función objetivo, que se pretende maximizar o minimizar. Todo ello acompañado de otro tipo de restricciones que añaden complejidad y realismo al modelo.

Pese a todo lo que podamos intentar, los mercados financieros son algo muy complejo, sujeto a una gran variabilidad y movido, en muchas situaciones, por agentes externos a él como son las situaciones climáticas o políticas que se den en los distintos lugares del mundo. La suerte que tenemos a día de hoy, es disponer de una gran cantidad de datos y de métodos para trabajar con ellos, lo que nos permite, en bastantes ocasiones, tener una idea de cómo pueden moverse o comportarse.

1.2 Motivación

En el ámbito personal este trabajo surge con el objetivo de aumentar mis conocimientos en algunos de los campos que más interés me han generado a lo largo del PEC de Ingeniería Informática y Estadística, como son la Inteligencia Artificial y el estudio de series temporales. A su vez, la aplicación de estas técnicas al mundo financiero, es algo que resulta atractivo, ya que el dinero es un elemento fundamental de nuestra vida cotidiana.

Otro aspecto a destacar a la hora de afrontar este trabajo es el del aprendizaje de nuevas tecnologías, como es el caso de Docker, ampliamente utilizado en el mundo laboral actual y desconocido para mi antes de afrontar este proyecto. Del mismo modo, reforzar y ampliar los conocimientos en desarrollo web, poco enfocado en la

mención en Computación que he cursado, supuso un importante incentivo.

El trabajo desarrollado tiene como punto de partida otros trabajos realizados en la Universidad de Valladolid sobre el mismo tema, como es el caso del realizado por César Hernández sobre Predicción y Clasificación de series temporales bursátiles mediante Redes Neuronales Recurrentes [3] y los trabajos realizados por el propio César [4] y por Sonia Aparicio [5] sobre optimización de carteras. En este TFG se busca ampliar los objetivos llevados a cabo en esos proyectos, así como abordar alguna de las futuras líneas de trabajo que propusieron estos antiguos alumnos. De manera complementaria, el autor realiza su Trabajo de Fi de Grado del Grado en Estadística sobre el mismo tema, el cual puede verse en [6].

Existen múltiples herramientas para analizar activos financieros, tanto *online*, como programas de escritorio, tanto gratuitas como de pago. Sin embargo, no existen herramientas que presenten una diversidad de técnicas orientadas al estudio y predicción de mercados, por lo que esto supone a su vez un incentivo a la hora de afrontar este trabajo.

Un último aspecto motivador de este trabajo ha sido el trabajar dentro del marco de una metodología ágil, habilidad indispensable para trabajar en desarrollo de software en la actualidad.

1.3 Objetivos

Este proyecto persigue dos objetivos de carácter general. El primero de ellos es el desarrollo de una herramienta interactiva que permita comprender, entender y explicar algunos de los temas fundamentales relacionados con los activos financieros, así como de los principales modelos de optimización de carteras y de la aplicación de técnicas de *machine learning* al problema de la predicción de mercados. El segundo consiste en conseguir que la aplicación a su vez sirva de ayuda al inversor a la hora de tomar decisiones con su propio dinero, eso sí, ha de ser un inversor que tenga conocimientos técnicos sobre los modelos que se presentan en la aplicación ya que, de lo contrario, no podrá configurarlos correctamente. Todo ello, trabajando con activos pertenecientes a diversos mercados, y profundizando en algunas de las líneas de trabajo propuestas por otros compañeros en los trabajos comentados en el apartado anterior.

En lo relativo a los trabajos anteriores, los objetivos a mayores que se han planteado han sido:

- Inclusión de activos pertenecientes a diversos mercados.
- Mejora de los elementos software construidos con la librería shiny, así como aumento de las posibilidades proporcionadas con ellos para la interacción del usuario.
- Inclusión de restricciones de cardinalidad en el modelo de Markowitz de optimización de carteras, enlazando para ello con un *solver* de programación matemática a la hora de resolver el nuevo problema planteado.
- Inclusión de la posibilidad de utilizar corrección de estimadores con ponderación, de forma que las observaciones más recientes tengan un peso mayor a la hora de construir una cartera. También se afronta esto enlazando con un *solver*.
- Inclusión en la interfaz gráfica de la posibilidad de realizar análisis de sensibilidad con el parámetro de ponderación del modelo de Markowitz.
- Trabajo con otros tipos de arquitecturas de aprendizaje automático a la hora de enfrentar el problema de la predicción de valores.

En cuanto al desarrollo de la aplicación web, el objetivo es construir una herramienta que permita:

- Visualizar la información de algunos de los principales mercados del mundo.

- Construir carteras de optimización con el modelo de Markowitz de forma interactiva, pudiendo el usuario elegir los elementos que componen el modelo y su configuración.
- Evaluar una cartera construida en términos de rentabilidad.
- Realizar un análisis descriptivo de la serie temporal de un activo.
- Visualizar la serie temporal de un activo junto a los principales indicadores técnicos contemplados en el ámbito económico.
- Construir modelos de predicción de valores con técnicas de aprendizaje automático, permitiendo al usuario de manera interactiva elegir los principales elementos que lo componen y su configuración.
- Comparar distintos modelos de predicción de valores sobre la serie temporal de un mismo activo.
- Ver qué activos tienen un comportamiento semejante en términos de rentabilidad mediante *análisis cluster*, realizado tras reducir la dimensionalidad con un Análisis de Componentes Principales.
- Estudiar la relación entre un índice bursátil y una cartera compuesta por empresas pertenecientes a dicho índice, realizando una regresión entre la cartera y el índice y acompañando esto de la descomposición del riesgo de la cartera derivada de dicho modelo.

1.4 Estructura

La memoria del presente trabajo se estructura de la siguiente manera:

- **Capítulo 1 - Introducción:** planteamiento del trabajo, contextualización, motivación y objetivos del mismo.
- **Capítulo 2 - Planificación:** organización temporal del trabajo, riesgos y costes del mismo.
- **Capítulo 3 - Marco conceptual:** aspectos de economía en general y sobre activos financieros en particular, explicación sobre el modelo de Markwotitz, sobre la predicción de valores con técnicas de inteligencia artificial y sobre los distintos modelos de aprendizaje automático utilizados en el trabajo.
- **Capítulo 4 - Fuentes de datos:** origen y tratamiento de los datos utilizados.
- **Capítulo 5 - Análisis:** qué es lo que se necesita para llevar a cabo el desarrollo del sistema, requisitos que ha de cumplir el mismo y casos de uso que ha de satisfacer.
- **Capítulo 6 - Diseño:** cómo se planifica la solución software que se propone para cumplir los puntos detallados en análisis.
- **Capítulo 7 - Implementación:** cómo se lleva a cabo la solución software diseñada.
- **Capítulo 8 - Ejemplos de Uso:** resultado final de la aplicación web, con ejemplos del uso y visualización de las diferentes secciones.
- **Capítulo 9 - Discusión y Ampliaciones:** planteamiento de mejoras sobre el trabajo realizado, futuras ampliaciones del mismo y líneas de trabajo.
- **Capítulo 10 - Conclusiones:** punto donde se ha concluido el trabajo, con los principales hallazgos y logros conseguidos.
- **Apéndice A - Manual de Usuario:** cómo utilizar el sistema desarrollado.

- **Apéndice B - Manual del Desarrollador:** cómo modificar el sistema desarrollado, de cara a corregir problemas o imperfecciones y a añadir mejoras o incluir funcionalidades.
- **Apéndice C - Diagramas de secuencia:** enlaces al repositorio donde se encuentran los diagramas de secuencia de los casos de uso del sistema.

Capítulo 2

Planificación

2.1 Metodología

Todo trabajo requiere un método bien especificado y establecido. A lo largo del tiempo se han ido desarrollando diferentes metodologías de desarrollo con el objetivo de organizar el trabajo en equipo de un grupo de personas y de llevar a cabo las tareas requeridas en un proyecto de una manera productiva y eficaz [7].

Pese a que a lo largo de la carrera siempre se han utilizado metodologías tradicionales en los procesos de desarrollo de software, como son el desarrollo iterativo, en cascada o incremental, en el presente trabajo se ha optado por dar un enfoque más ágil. Esto suele ser más habitual en los equipos de trabajo de la industria, debido a la alta flexibilidad y agilidad que permiten este tipo de metodologías. Este enfoque tiene a su vez la ventaja de permitir la adaptación del producto y de los subproductos que se fabrican a las necesidades que van surgiendo durante el proceso de desarrollo, así como la construcción de equipos de trabajo autosuficientes e independientes, que se coordinan mediante reuniones periódicas.

Los métodos ágiles tienen como base el desarrollo incremental de las metodologías tradicionales, buscando agregar unas pocas nuevas funcionalidades al producto en cada ciclo de desarrollo, siendo estos de duración breve (como mucho de ocho semanas de duración, siendo dos o cuatro semanas la duración habitual). Las principales metodologías ágiles son las siguientes:

- **Kenban:** consiste en dividir las tareas en tres bloques: tareas finalizadas, tareas en curso y tareas pendientes, creando un flujo de trabajo muy claro que permite incrementar el valor del producto.

- SCRUM **:** similar al anterior, los ciclos de iteración son cortos y están fijados antes de comenzar el proyecto. Se introduce el concepto de *sprints* [8], que es la forma de denominar en esta metodología a cada iteración o ciclo de trabajo. En cada *sprint* se ha de generar lo que se denomina un entregable, es decir, un incremento que aporte valor al cliente. Es importante recalcar que en cada uno de ellos, el producto ha de ser funcional. Cada *sprint* de SCRUM está formado a su vez por varias fases:

  - **Planificación:** qué se va a hacer en el *sprint* y cómo se pretende eso llevar a cabo.
  - SCRUM **diario:** reuniones diarias de duración corta donde los miembros del equipo exponen sus progresos y dificultades.
  - **Revisión:** se acepta o no el *sprint* realizado.
  - **Retrospectiva:** se analiza cómo ha ido el *sprint* , qué problemas ha habido y cómo mejorarlos.

- **Lean:** busca que pequeños equipos de trabajo con alta capacitación desarrollen cualquier tipo de tarea en poco tiempo. Se centra en las personas, dejando en segundo plano el tiempo y los costes.
- **Programación Extrema (XP):** centrado en las relaciones interpersonales donde habitualmente se realiza programación por parejas. Se basa en principios como: diseño sencillo, testeo, refactorización, integración continua y entregas semanales entre otros.

En el proyecto llevado a cabo en este trabajo se plantea un enfoque similar a SCRUM con *sprints* de dos semanas pero con matices, ya que al ser un equipo de desarrollo compuesto únicamente por una persona, ciertos elementos de la metodología como las reuniones entre miembros carecen de sentido. A su vez, se incluye en el proyecto cierto carácter de metodología tradicional al llevarse a cabo un desarrollo web con prototipos, lo que no aleja la metodología del enfoque ágil incremental planteado por SCRUM . Del mismo modo, al tratarse de un proyecto académico donde parte del trabajo es de investigación y aprendizaje, las primeras semanas del mismo no seguirán una metodología ágil como tal al no producirse entregables, puesto que estas primeras semanas se destinarán al despliegue del servidor y al análisis del problema y las posibles soluciones.

2.2 Actividades

Una manera habitual de definir las tareas que requiere un sistema software consiste en la creación de un diagrama de descomposición de tareas (WBS). Este enfoque implica identificar las principales tareas requeridas para llevar a cabo la construcción del sistema (tareas de alto nivel) para, posteriormente, descomponer cada tarea en subtareas de nivel más bajo. El modo de proceder consiste en añadir tareas en cada "rama" únicamente si están directamente relacionadas en la consecución de la tarea "padre". Cada rama debe descomponerse, al menos, hasta un punto en el cual la "hoja" pueda asignarse a un único individuo o sección de una organización. En el presente trabajo, todo será llevado a cabo por un mismo individuo pero se muestra la descomposición que habría que hacer en el ámbito profesional de una organización.

Un aspecto importante a considerar en este tipo de diagramas es el nivel de detalle que se quiere mostrar, ya que demasiada profundidad puede dar lugar a un número de tareas que sea complicado de gestionar, mientras que un diagrama demasiado superficial proporciona un nivel de detalle demasiado escaso para el necesario control de proyecto. El diagrama de tareas de este proyecto se muestra en la Figura 2.1.

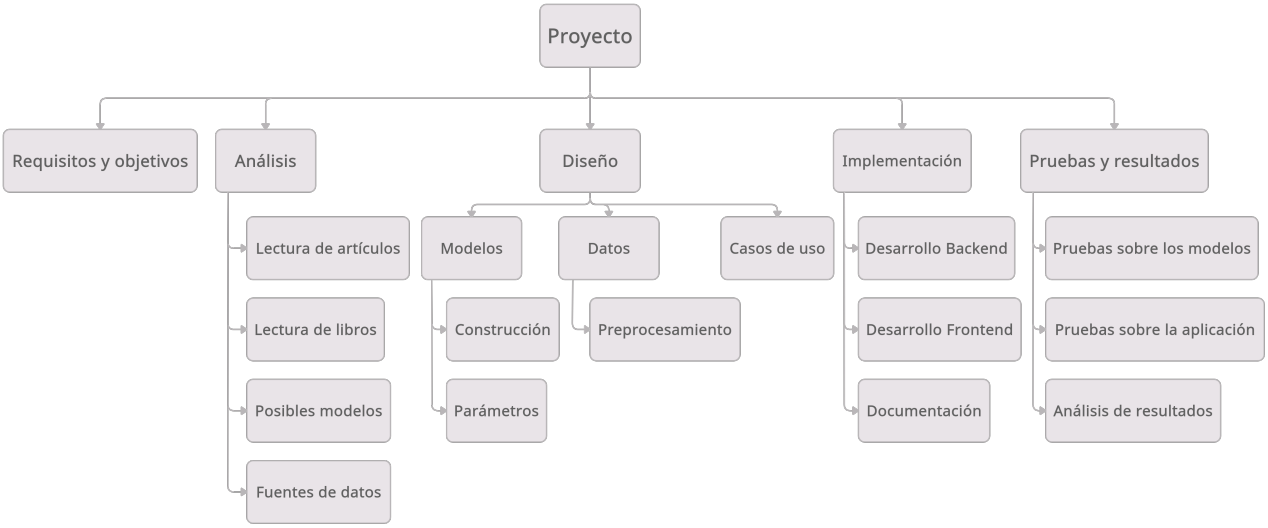


Figura 2.1: Diagrama de tareas.

2.3 Recursos

Los recursos requeridos en este proyecto software son únicamente de dos tipos:

- **Humanos:** el alumno encargado de desarrollar el proyecto en su totalidad.
- **Tecnológicos:** el equipo de trabajo que utilizará el desarrollador, la máquina virtual que alojará la aplicación (proporcionada por la Escuela de Ingeniería Informática de la Universidad de Valladolid), conexión a internet para poder acceder a dicha máquina y para la comunicación con el tutor del trabajo y luz corriente para poder utilizar los equipos informáticos.

2.4 Planificación inicial

De cara a la realización de este proyecto y conforme a la metodología anteriormente explicada, podemos agrupar las tareas que conformarán el trabajo en cuatro grandes bloques:

- **Análisis del problema:** en esta fase se estudiará el problema a resolver en este trabajo, tanto desde el punto de vista conceptual como desde el punto de vista práctico. Se llevarán a cabo tareas de aprendizaje, despliegue del servidor que alojará la aplicación web, lectura de diversos artículos y libros que permitirán trazar la hoja de ruta sobre los modelos a implementar en las fases posteriores. También se analizarán las tecnologías a utilizar en el proyecto y el modo de utilizarlas para la implementación del sistema.
- **Diseño del sistema:** se llevarán a cabo tareas de ingeniería de software que tratarán de definir y especificar claramente la estructura y funcionamiento de la aplicación web. Para ello se planteará un diseño con *mockups* y los diagramas necesarios para explicar todo lo necesario del sistema.
- **Implementación del sistema:** programación de los diferentes módulos que componen la aplicación web diseñada en la fase anterior.
- **Elaboración de la memoria:** escritura final de la memoria, revisando todo lo redactado anteriormente en las tareas de documentación.

Para adaptar estas tareas a la metodología explicada en el punto 2.1 se ha de tener en cuenta que la primera fase de desarrollo de este trabajo englobará toda la parte de análisis, que no se incluirá como tal en el marco ágil elegido para el desarrollo del trabajo. El resto del trabajo seguirá el enfoque comentado, teniendo en cuenta que las tareas de diseño e implementación se realizarán en cada *sprint* generando entregables de manera incremental y que la tarea de escritura de la memoria constituirá en sí el último *sprint* donde el entregable producido es la memoria final del proyecto.

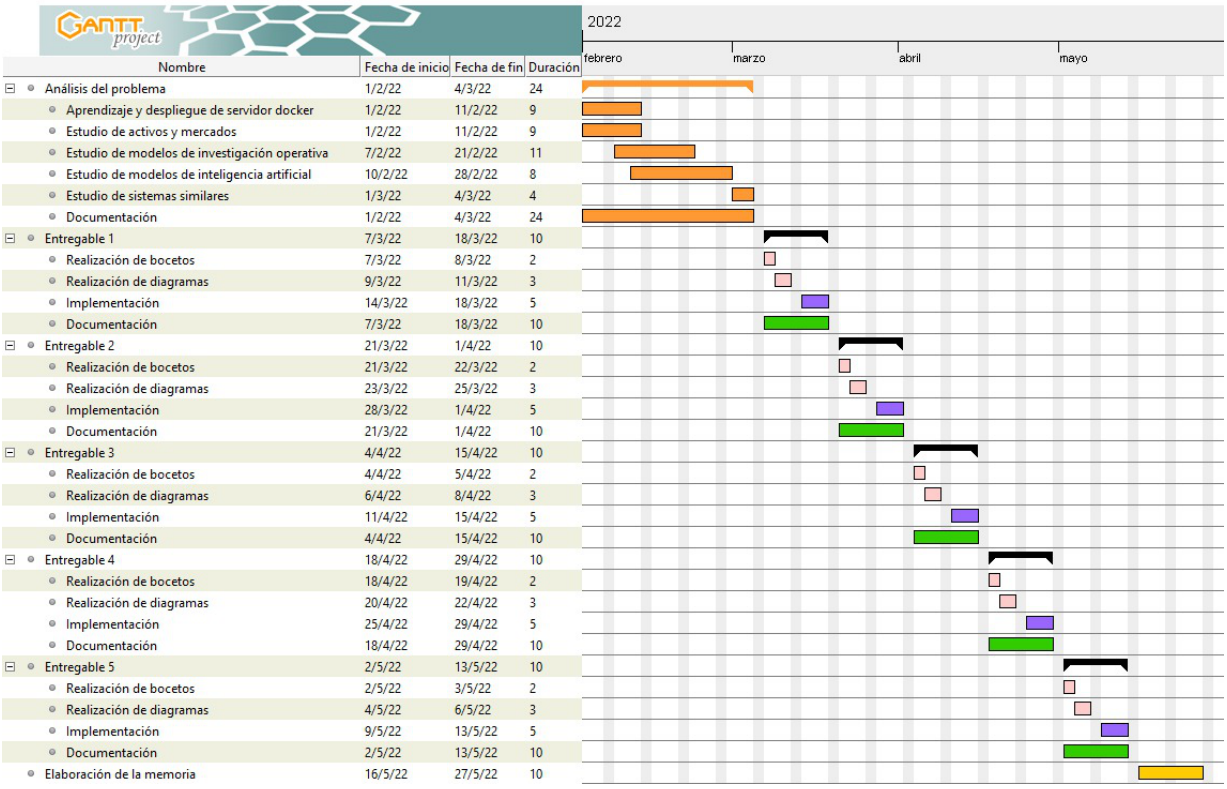
El diagrama de Gantt de la Figura 2.2 muestra el diagrama de tareas del proyecto, teniendo en cuenta que el color naranja identifica a la fase inicial de análisis, el color rosa identifica las tareas de diseño, el color morado las tareas de implementación, el verde la documentación que se va realizando en paralelo al resto de actividades de cada *sprint* y, finalmente, el color amarillo indica la escritura final de la memoria.

2.5 Fases y costes

Los *sprints* planteados en este trabajo son quinquenales y se resumen en la tabla 2.1.

sprint	**Nombre de actividad**	**Semanas**
sprint 1	Análisis y aprendizaje de las herramientas a utilizar para el despliegue del servidor web	1 - 2
sprint 1	Despliegue del servidor web	1 - 2
sprint 1	Estudio de activos y mercados	1 - 2
sprint 1	Estudio de modelos de investigación operativa	1 - 2
sprint 1	Estudio de modelos de inteligencia artificial	1 - 2
sprint 2	Estudio de modelos de investigación operativa	3 - 4
sprint 2	Estudio de modelos de inteligencia artificial	3 - 4
sprint 3	Elaboración documento de requisitos y casos de uso	5 - 6
sprint 3	Elaboración básica de la web	5 - 6
sprint 3	Elaboración sección inicial de la web	5 - 6
sprint 3	Elaboración *scripts* de pruebas de modelos	5 - 6
sprint 3	Elaboración *scripts* de pruebas de extracción de datos	5 - 6
sprint 4	Re-elaboración de la estructura de la web	7 - 8
sprint 4	Elaboración documento de historias de usuario	7 - 8
sprint 4	Elaboración inicial de la sección de análisis de valores	7 - 8
sprint 4	Elaboración inicial de la sección de optimización	7 - 8
sprint 4	Elaboración inicial de la sección de modelos de predicción	7 - 8
sprint 5	Elaboración de los primeros modelos de optimización	9 - 10
sprint 6	Elaboración completa de la sección de *clustering* de valores	11 - 12
sprint 6	Elaboración completa de la sección de simulación de modelos	11 - 12
sprint 6	Elaboración completa de la sección de evaluación de carteras	11 - 12
sprint 6	Inclusión de nuevos gráficos y funcionalidades en la sección de an	11 - 12
sprint 6	Elaboración de nuevos modelos de predicción	11 - 12
sprint 6	Elaboración completa de la sección de optimización de modelos	11 - 12
sprint 7	Re-elaboración de la web para darle un aspecto más profesional	13 - 14
sprint 7	Pruebas y solución de errores de las diferentes secciones	13 - 14
sprint 7	Mejoras en el código de diferentes secciones	13 - 14
sprint 7	Inclusión de funcionalidades menores	13 - 14
sprint 7	Documentación	13 - 14
sprint 8	Retoques finales de la web	15 - 16
sprint 8	Elaboración de la sección de modelos de predicción	15 16
sprint 8	Pruebas y solución de errores de las diferentes seccions	15 - 16
sprint 8	Mejoras en el código de diferentes secciones	15 - 16
sprint 8	Documentación análisis	15 - 16
sprint 8	Documentación diseño	15 - 16
sprint 9	Documentación completa	17 - 18
sprint 9	Retoques, pruebas y solución de últimos errores	17 - 18
sprint 9	Escritura final de la memoria del TFG	17 - 18

Cuadro 2.1: Sprints de desarrollo del proyecto previstos.



Nombre	Fecha de inicio	Fecha de fin	Duración
Análisis del problema	1/2/22	4/3/22	24
Aprendizaje y despliegue de servidor docker	1/2/22	11/2/22	9
Estudio de activos y mercados	1/2/22	11/2/22	9
Estudio de modelos de investigación operativa	7/2/22	21/2/22	11
Estudio de modelos de inteligencia artificial	10/2/22	28/2/22	8
Estudio de sistemas similares	1/3/22	4/3/22	4
Documentación	1/2/22	4/3/22	24
Entregable 1	7/3/22	18/3/22	10
Realización de bocetos	7/3/22	8/3/22	2
Realización de diagramas	9/3/22	11/3/22	3
Implementación	14/3/22	18/3/22	5
Documentación	7/3/22	18/3/22	10
Entregable 2	21/3/22	1/4/22	10
Realización de bocetos	21/3/22	22/3/22	2
Realización de diagramas	23/3/22	25/3/22	3
Implementación	28/3/22	1/4/22	5
Documentación	21/3/22	1/4/22	10
Entregable 3	4/4/22	15/4/22	10
Realización de bocetos	4/4/22	5/4/22	2
Realización de diagramas	6/4/22	8/4/22	3
Implementación	11/4/22	15/4/22	5
Documentación	4/4/22	15/4/22	10
Entregable 4	18/4/22	29/4/22	10
Realización de bocetos	18/4/22	19/4/22	2
Realización de diagramas	20/4/22	22/4/22	3
Implementación	25/4/22	29/4/22	5
Documentación	18/4/22	29/4/22	10
Entregable 5	2/5/22	13/5/22	10
Realización de bocetos	2/5/22	3/5/22	2
Realización de diagramas	4/5/22	6/5/22	3
Implementación	9/5/22	13/5/22	5
Documentación	2/5/22	13/5/22	10
Elaboración de la memoria	16/5/22	27/5/22	10

Figura 2.2: Diagrama de Gantt de actividades.

2.6 Presupuesto inicial

Teniendo en cuenta los recursos indicados en 2.3 y estimando una duración de unas 300 horas para la realización total del trabajo (12 créditos) repartidas en un total de unos 4 meses, podemos estimar el coste del proyecto. Para ello tenemos que hacer varias consideraciones:

- Estimamos el salario del desarrollador como el salario medio de un ingeniero informático en España que, de acuerdo a [9], es de unos 36500€ brutos al año, es decir, unos $18{,}71€/hora$ (aunque esto es muy variable en función de la empresa, años de experiencia, etcétera).

- Estimamos el coste de la luz durante los próximos 4 meses en España utilizando el promedio del coste de la luz a lo largo de 2021, que fue de $111,38€/MWh$ de media de acuerdo a [10]. Teniendo en cuenta una utilización aproximada de 300 horas del equipo de trabajo, que la batería está al 96 %, lo que hace un total de $4542mAh$ que, teniendo en cuenta que el voltaje en España es de $230V$ da un valor de $1044{,}66Wh$ tras hacer la conversión (para cada carga). El total de cargas completas es de unas 100 ($300h/3h$ cada carga), por lo que el total de gasto en luz en MWh se estima en unos $0{,}104466$ WMh.

- Ignoramos el consumo eléctrico imputable a la máquina virtual, pues no existe forma de acceder al consumo eléctricos de centros o departamentos en la universidad y, por tanto, podría sólo estimarse un consumo nominal por servidor. Aún en ese caso, sería necesario conocer el número de máquinas y la carga promedio del virtualizador para poder estimar el consumo por VM. Por ello, hemos ignorado este elemento de coste.

- El equipo utilizado para el desarrollo tiene un tiempo de vida menor a un año y es un ordenador MSI GL65

Leopard con un precio actual de unos 1200€ según distintas páginas de venta de productos informáticos.

Esto da como resultado un coste resumido en la siguiente tabla:

Concepto	**Coste**
Horas de trabajo del desarrollador	5613€
Equipamiento	1200€
Desgaste de equipamiento	10€
Luz	1,16€
Total	6824,16€

Cuadro 2.2: Costes del proyecto.

2.7 Desviaciones de la planificación inicial

En cuanto a la planificación inicial, se han producido las siguientes desviaciones:

- Alargamiento del periodo inicial considerado para la realización del trabajo, ya que inicialmente se planeó finalizar el día 27 de mayo y finalmente se ha finalizado el 6 de junio (10 días después).
- Inicialmente se planteó un enfoque basado en entregables, donde en cada uno de ellos se generaba la documentación correspondiente. Esto no se ha cumplido de este modo y la documentación ha sido generada, en su mayor parte, en las semanas finales del proyecto.
- Inicialmente se estimaba una duración de unas $300h$ para la realización del trabajo, que, junto a las $150h$ necesarias para realizar el TFG de estadística, hacen un total de $450h$ y, puesto que los trabajos en este caso eran conjuntos, esta era la duración estimada en horas. La duración final ha sido mayor, como es lógico al haber excedido el número de días. En total han sido unas 500 horas para la realización de los dos proyectos.

2.8 Coste final

Puesto que el coste de la luz en los últimos meses ha aumentado, con una media en el periodo de duración de este TFG de 215,96€ (elaboración propia con datos tomados de [11]), aproximadamente el doble del utilizado en los cálculos iniciales de 2.6, y que en ese cálculo no se incluyeron las horas adicionales resultantes de trabajar en conjunto con los dos proyectos, se estima un coste en luz de en torno a 2,5 veces el estimado. Del mismo modo, el aumento en la duración del trabajo implica un aumento en el coste del desgaste del equipamiento y del coste del desarrollador. El resultado final pues queda como indica la tabla 2.3. (**Nota:** las horas del desarrollador, inicialmente tomadas como 300 se calculan ahora teniendo en cuenta los dos TFGs realizados al no ser posible la distinción entre horas dedicadas a uno y otro por separado, ya que ambos se han realizado en conjunto. Esto hace que el coste en horas calculado en este apartado sea mucho mayor).

Concepto	**Coste**
Horas de trabajo del desarrollador	9355€
Equipamiento	1200€
Desgaste de equipamiento	12€
Luz	2.9€
Total	10569,9€

Cuadro 2.3: Coste final del proyecto.

2.9 Análisis de riesgos

En esta sección se tratará de realizar un análisis de los posibles riesgos a los que se encuentra expuesto este proyecto software. Los únicos que se van a considerar son los riesgos de proyecto, ignorándose los posibles riesgos de negocio. Esto se debe a que la aplicación web diseñada en este TFG no se va a crear con la intención de comercializarla.

Para poder realizar un correcto análisis y evaluación de los riesgos de un proyecto, es necesario asignar a cada uno de ellos una probabilidad de ocurrencia y una medida numérica del impacto que tendría el mismo en el caso de materializarse. El enfoque elegido para llevar a cabo esto, es el propuesto por Barry Boehm [12], consistente en analizar cualitativamente tanto las probabilidades de ocurrencia como los impactos de los riesgos.

Nivel de probabilidad	**Rango**
Alto	Más de un $50\,\%$ de probabilidad de ocurrencia
Significativo	$30 - 50\,\%$ de probabilidad de ocurrencia
Moderado	$10 - 29\,\%$ de probabilidad de ocurrencia
Bajo	Menos de un $10\,\%$ de probabilidad de ocurrencia

Cuadro 2.4: Tabla probabilidades Barry Boehm. Tomado de [12].

Nivel de impacto	**Rango**
Alto	Más de un $30\,\%$ sobre el gasto presupuestado
Significativo	$20 - 29\,\%$ sobre el gasto presupuestado
Moderado	$10 - 19\,\%$ sobre el gasto presupuestado
Bajo	Menos de un $10\,\%$ sobre el gasto presupuestado

Cuadro 2.5: Tabla impactos Barry Boehm. Tomado de [12]

Una vez estudiadas las probabilidades e impactos de los riesgos de un proyecto, conviene definir los planes de actuación frente a los mismos, los cuales se engloban en dos categorías principales:

- **Plan de prevención o protección:** acciones llevadas a cabo con el objetivo de reducir la probabilidad de que un riesgo se manifieste [13].
- **Plan de contingencia:** acciones llevadas a cabo con el objetivo de que los perjuicios causados por la materialización de un riesgo sean lo menos graves posibles. Constituye una guía de actuación a seguir cuando un riesgo se manifiesta y trata de reducir su impacto todo lo posible.

Así pues, los riesgos encontrados en el presente proyecto junto a sus consiguientes planes de prevención y contingencia, quedan resumidos en la siguiente tabla:

Id	**Descripción**	**Probabilidad**	**Impacto**
01	Retraso en la planificación de cualquiera de las tareas.	Significativa	Alto
02	Enfermedad del desarrollador del proyecto	Moderado	Alto
03	Incumplimiento de la planificación debido a un mal planteamiento inicial de la misma	Significativa	Alto
04	Falta de conocimiento de tecnologías	Significativa	Significativo
05	Problemas en la máquina virtual que aloje la aplicación	Baja	Alto
06	Problemas en el hardware del equipo utilizado para el desarrollo	Baja	Alto
07	Errores en el diseño de la aplicación	Moderada	Moderado
08	Falta de potencia computacional para desarrollar y probar los modelos generados	Baja	Alto
09	Errores en la implementación de la aplicación	Baja	Alto
10	Problemas en las webs utilizadas para la obtención de datos en tiempo real	Baja	Alto

Cuadro 2.6: Tabla de riesgos del proyecto.

Id	**Plan de prevención**	**Plan de contingencia**
01	Planificación y calendarización cuidadosa y con cierta holgura.	Replanificación de las tareas e incremento del número de horas invertidas en el proyecto.
02	Mantener hábitos de vida saludable y tener precaución con la situación asociada a la pandemia del COVID-19.	Aceptación y replanificación de tareas para ajustar el tiempo perdido por la indisposición.
03	Planificación y calendarización cuidadosa y con cierta holgura.	Replanificación de las tareas e incremento del número de horas invertidas en el proyecto.
04	Formación y fase de aprendizaje.	Pedir ayuda al tutor del trabajo.
05	Tener todos los archivos y programas en mi equipo personal para no perderlos en caso de fallo en la máquina virtual y para poder trabajar y probar cosas desde mi equipo.	Ponerse en contacto con los técnicos de la escuela.
06	Tener todos los archivos y programas copiados en algún dispositivo externo como un disco duro o alojados en algún servidor de nube. Hacer un uso cuidadoso del equipo utilizado para el desarrollo.	Contactar con algún amigo o familiar que pueda prestarme un equipo para continuar con el desarrollo del proyecto.
07	Realización cuidadosa y supervisada por el tutor del trabajo.	Re-elaboración de las partes mal diseñadas y replanificación en caso necesario.
08	Solicitar una máquina virtual con la potencia adecuada y minimización de los recursos utilizados por el ordenador personal en el caso de ejecutarse en este la aplicación.	Solicitar una nueva máquina virtual que cumpla con los requerimientos necesarios.
09	Desarrollo con prototipos, incremental y cuidadoso.	Re-elaboración de las partes mal implementadas y replanificación en caso necesario.
10	Búsqueda previa de varias fuentes de datos que proporcionen los mismos datos para tener una alternativa en caso de caída o problemas en alguna de ellas.	Uso de webs alternativas y replanificación en caso necesario.

Cuadro 2.7: Tabla de planes de prevención y contingencia de riesgos.

Parte II

Marco Conceptual y Contextual

Capítulo 3

Marco Conceptual

A lo largo de este capítulo se presentará el marco teórico sobre el que se desarrolla este trabajo. Se explicarán brevemente conceptos básicos sobre activos y mercados, sobre los activos elegidos en este proyecto y sobre los diferentes modelos que en él aparecen, con el fin de que todo lo que se muestra en la aplicación web construida sea comprensible.

3.1 Introducción económica

El sistema financiero de un país es el sistema constituido por entidades, mercados y medios que canalizan el ahorro generado por las unidades de gasto con superávit hacia las unidades de gasto con déficit, otorgando así seguridad al movimiento del dinero y al sistema de pagos [14].

Algunos de los instrumentos que componen este complejo sistema son los productos bancarios (cuentas, depósitos), los planes de pensiones, los productos de seguros y los productos de inversión (acciones, bonos). Estos últimos son lo que se conocen como activos financieros [15].

Los **activos financieros** se definen como las herramientas que utiliza el sistema financiero para facilitar la movilidad de recursos. Son emitidos por una institución y adquiridos por otras instituciones o por personas particulares. Con ellos, el comprador adquiere el derecho a recibir un ingreso futuro por parte del vendedor [16]. Los activos financieros se diferencian de los activos reales (un coche o una casa son ejemplos de activos reales) en que no suelen poseer un valor físico. Una segunda diferencia frente a otros tipo de activos, es que los activos económicos no incrementan la riqueza de un país (no influyen en el PIB), aunque sí contribuyen al crecimiento económico del mismo al impulsar la movilización de recursos.

Las principales características que permiten definir los distintos activos financieros son las siguientes:

- **Liquidez:** es la capacidad de transformar un activo en dinero en un espacio breve de tiempo, sin sufrir pérdidas y sin afectar a la rentabilidad. La liquidez depende tanto de la facilidad para realizar la conversión como del grado de certeza de que esa conversión se va a poder llevar a cabo sin pérdidas. El activo más líquido de todos es el dinero, aunque otros activos como los fondos bancarios también gozan de gran liquidez.

- **Riesgo:** es una medida de las garantías que ofrece el vendedor, es decir, de su solvencia. Depende de la probabilidad de que este, a su vencimiento, cumpla con lo pactado con el comprador. El riesgo de un activo será mayor cuanto mayor sea la incertidumbre ofrecida por el vendedor para la devolución de la inversión. Un ejemplo de inversión de bajo riesgo sería la deuda pública (donde el vendedor es el Estado) y ejemplos de riesgo alto podrían ser determinadas acciones de bolsa o las inversiones en criptomonedas.

- **Rentabilidad:** se define como el interés o la plusvalía que obtiene el comprador a cambio de aceptar el riesgo de ceder su capital. Puede reflejarse en términos de intereses, beneficios u otros. Por lo general, cuanto mayor es la rentabilidad, mayor es el riesgo asociado al activo y menor es su liquidez. Por el contrario, cuanto menor es el riesgo que ofrece un emisor y mayor es la liquidez, menor es también el beneficio que se espera obtener.

Otro aspecto importante a tratar a la hora de definir los activos financieros es su clasificación. Usualmente esta se divide en dos categorías:

- **Activos primitivos:** aquellos cuyo valor depende de las rentas futuras esperadas y de su riesgo asociado. A su vez se dividen en otras dos categorías:
  - **Renta fija:** emitidos por entidades públicas o empresas, que previamente fijan el pago de un interés (de una cantidad y en un plazo establecido), de forma que éste no depende de la evolución ni de los resultados de la compañía emisora y quedando siempre garantizado. Algunos ejemplos de este tipo de activos son los bonos de deuda emitidos por los Estados o los pagarés emitidos por las empresas.
  - **Renta variable:** se describen así los activos cuyo rendimiento es variable en función de la marcha de la entidad emisora. En este caso, ni la rentabilidad ni la recuperación del capital están garantizados. El ejemplo más importante de estos activos financieros son las acciones.
- **Activos derivados:** reciben este nombre porque su valor depende del valor de un activo primitivo denominado *activo subyacente*. El activo derivado es la *opción* que depende del activo primitivo, como, por ejemplo, una opción sobre una acción. Este tipo de activos permiten a las empresas protegerse frente a fluctuaciones. Tipos de activos derivados son:
  - **Contratos a plazo *forward*:** contratos de compraventa negociados directamente entre comprador y vendedor (sin un mercado organizado como, por ejemplo, el mercado de valores) en los que se establece el precio a pagar, en una fecha futura, ante la entrega de un activo.
  - **Contratos futuros:** contrato que obliga a las partes a la compraventa de bienes o activos con una fecha y precio previamente establecidos. Se diferencian de los contratos *forward* en que se negocian en un mercado organizado que hace las veces de intermediario y en que, en general, pueden transmitirse a terceros.
  - **Opción:** derecho a vender un activo (el activo subyacente) en una fecha futura, a un precio fijado. Se distingue entre opciones de compra (*call*) y opciones de venta (*put*).
  - ***Swaps:*** acuerdos para el intercambio futuro de flujos monetarios según unas reglas previamente determinadas.

Una última clasificación sobre activos financieros que es importante comentar, es aquella determinada por los plazos de vencimiento. Distinguiéndose de este modo entre:

- **Activos a corto plazo:** se amortizan en un plazo corto de tiempo, generalmente inferior a 12 meses. Suelen ofrecer rentabilidades más bajas y algunos ejemplos son los pagarés y las letras del tesoro.
- **Activos a medio y largo plazo:** su amortización es en un plazo superior a 12 meses. Se negocian en los mercados de capitales y su riesgo es mayor (y, por tanto, mayor también su rentabilidad en términos generales) debido a la mayor fluctuación y volatilidad que puede darse en los valores y en los tipos de interés al ampliarse el plazo temporal. Algunos ejemplos de estos activos son las acciones o los bonos.

Finalmente, comentar que existen otras maneras de clasificar los activos financieros, como la liquidez, la valoración, el tipo de entidad emisora, el plazo de vencimiento o las características legales de la relación vendedor-comprador, aunque no se comentará más acerca de las mismas.

3.2 Análisis de datos de activos

A lo largo de los años se han ido desarrollando multitud de métodos destinados al análisis de los mercados y de los activos financieros que se ponen en juego en ellos. El método más habitual llevado a cabo para analizar este tipo de escenarios consiste en la división entre análisis fundamental y análisis técnico. Estos dos análisis son complementarios y siempre será recomendable tener en cuenta la información que aportan los dos a la hora de realizar cualquier tipo de movimiento con nuestro dinero.

3.2.1 Análisis fundamental

El análisis fundamental [17] es el análisis de los mercados en el que se tiene en cuenta información relativa a aspectos sobre los fundamentos de la empresa. Este tipo de aspectos son los referidos a la macroeconomía, la microeconomía, la estrategia empresarial, la contabilidad y la valoración empresarial entre otros; aunque aspectos ajenos a la propia empresa como el contexto político de los lugares donde la empresa esté establecida o incluso situaciones climáticas son también considerados en el análisis fundamental. Las herramientas que proporciona este tipo de análisis resultan de utilidad a la hora de juzgar si determinado valor está valorado por encima de su valor real (sobrevalorado) o si está valorado por debajo (infravalorado).

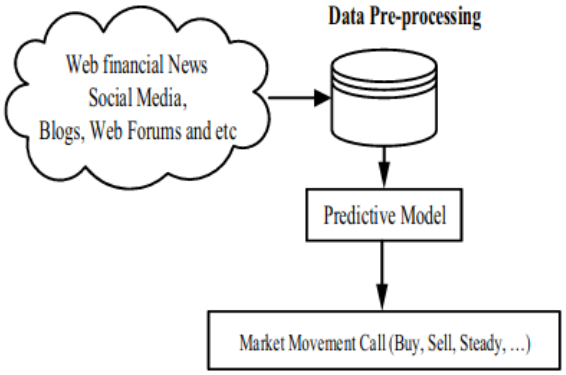


Figura 3.1: Esquema análisis fundamental. Tomado de [18].

Relacionado con esto último surge un concepto importante relativo al análisis fundamental, que es la idea de *mercado eficiente*. Se dice que un mercado presenta esta característica cuando el precio de cualquier acción refleja su verdadero valor, es decir, toda la información se encuentra disponible. Se distinguen tres hipótesis relativas a este concepto, en función de lo que se pueda entender por *información disponible*:

- **Eficiencia débil:** los precios reflejan la información que se puede extrapolar de la información histórica de las cotizaciones de un determinado valor. El análisis fundamental resulta de utilidad en este caso.
- **Eficiencia semifuerte:** los precios reflejan toda la información pública disponible. Lo útil en este caso sería la información privilegiada, conocida como *insider trading*.
- **Eficiencia fuerte:** los precios incorporan toda la información tanto pública como privilegiada. Nada es útil en este supuesto.

El analista fundamental requiere de un gran número de herramientas y de conocimientos para llevar a cabo sus predicciones. Algunas de las alteraciones del mercado que este puede detectar son las siguientes:

- **Efecto sobre-reacción:** el mercado tiende a sobre-reaccionar ante las nuevas noticias económicas, tendiéndose a que la información más reciente se sobrevalore. Estrategias que vayan en contra del concepto de eficiencia débil pueden ser rentables en estos casos.

- **Efecto enero:** indica que una parte importante de los rendimientos anuales derivados de la inversión se concentran en el mes de enero. La reestructuración de carteras a su vez es habitual que se realice en enero.
- **Efecto fin de semana:** el mercado tiende a tener un comportamiento más impulsivo y positivo los viernes, teniendo los lunes un comportamiento frecuentemente opuesto.

El análisis fundamental [18] utiliza información disponible esencialmente sobre tres aspectos: economía, la industria del activo y la propia empresa. Para ello se construyen diversos indicadores o *ratios*, algunos de los cuales se muestran reflejados en el siguiente esquema resumen del análisis fundamental:

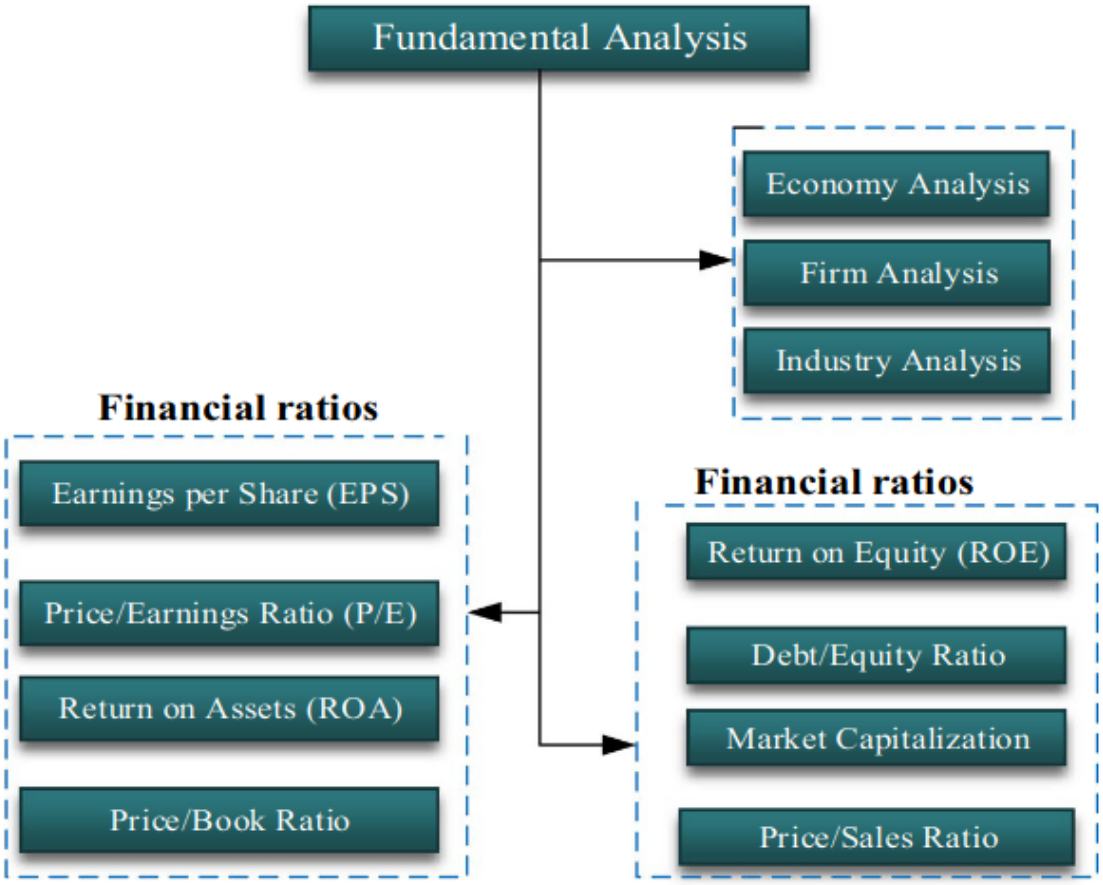


Figura 3.2: Indicadores análisis fundamental. Tomado de [18].

3.2.2 Análisis técnico

A diferencia del análisis fundamental, el análisis técnico se centra en predecir la conducta de los inversores a través de los movimientos de las acciones, analizándose el volumen y el precio, ignorándose los aspectos observados por el análisis fundamental. Algunos conceptos importantes a la hora de entender este tipo de análisis son [19]:

- **Sentimiento:** refleja el comportamiento de los participantes del mercado.
- **Flujo financiero (*flow-of-funds*):** indicador utilizado para representar el estatus de varios inversores en lo relativo a su fortaleza en términos de capacidad de compra y de venta. En base a esto se pueden adoptar diferente estrategias de inversión.

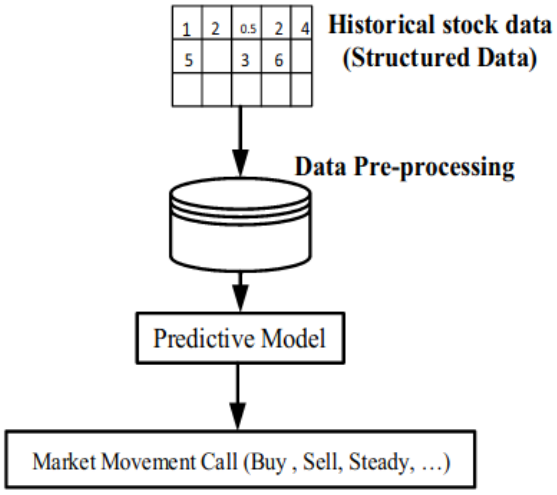


Figura 3.3: Esquema análisis técnico. Tomado de [18].

- **Datos *crudos*:** series de precios, patrones de precios y gráficos.
- **Tendencia:** indicador que hace referencia al patrón de movimiento que siguen una serie de observaciones de una serie temporal.
- **Volumen:** es un indicador que refleja el entusiasmo tanto de compradores como de vendedores, ya que hace referencia al total de ingresos recibidos en el total de transacciones económicas sobre un valor en un periodo de tiempo.
- **Volatilidad:** es la variación o el cambio de tendencia en el tiempo.

Al igual que en el análisis fundamental, existen diversos indicadores, algunos de los cuales se muestran recogidos en la siguiente Figura:

Medias móviles

Las medias móviles son uno de los indicadores más versátiles y de uso más común, consistiendo la base de muchos sistemas de seguimiento de tendencias utilizados en la actualidad. Pese a que su análisis gráfico resulta complejo, son fácilmente programables, de forma que se pueden crear programas que, de modo automatizado, detecten puntos de compra o de venta, independientemente de lo que el análisis gráfico pueda indicar.

Como su propio nombre indica, las medias móviles son un promedio. El objetivo de este tipo de funciones matemáticas es suavizar el comportamiento de la serie de un valor, de manera que se describa el comportamiento de la misma eliminando el ruido y facilitándose la visión de la tendencia subyacente en una serie.

Las medias móviles son esencialmente una forma de seguir una tendencia y el objetivo de las mismas es identificar los cambios en las mismas. Sin embargo, debido a que avisan de un cambio en la tendencia una vez este se ha producido, no resultan de utilidad para pronosticar el comportamiento del mercado.

Existen tres tipos de medias móviles:

- **Media móvil simple:** es la media aritmética. Es la más frecuentemente utilizada en análisis técnico. Sin embargo, aspectos como que da el mismo peso a todas las observaciones y que solo toma en consideración el periodo de tiempo cubierto por la media, hacen que otros tipos de medias móviles también resulten de interés.

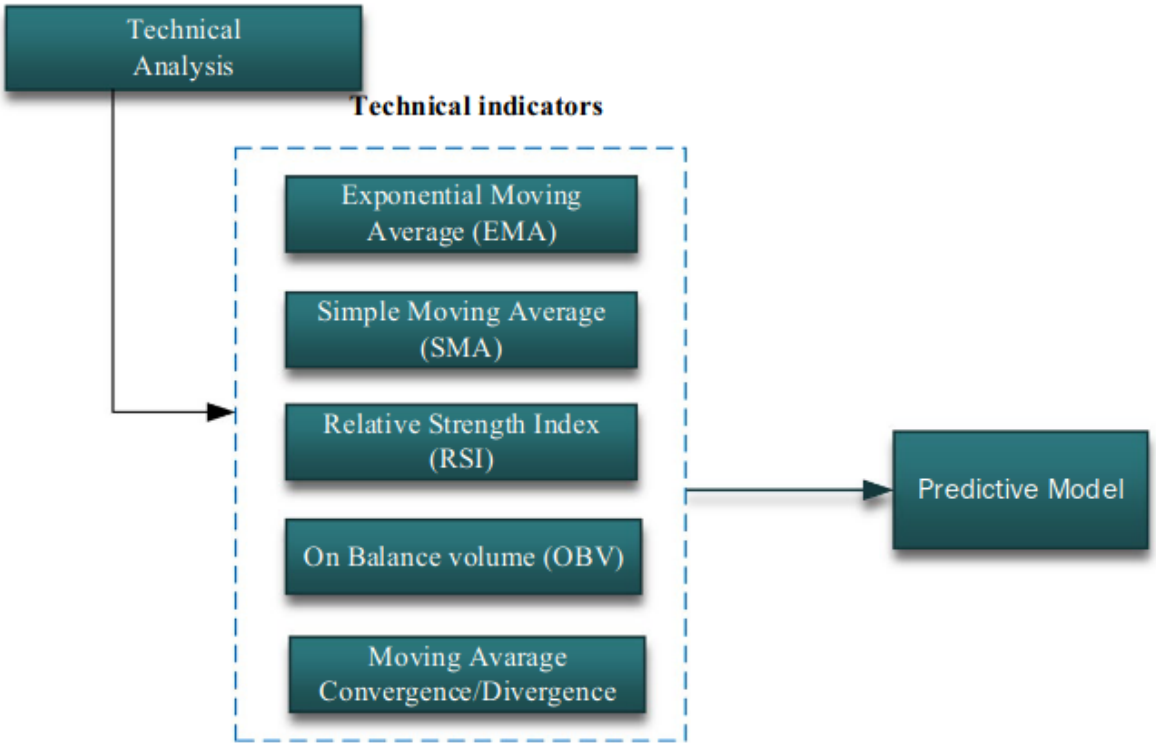


Figura 3.4: Indicadores análisis técnico. Tomado de [18].

Si denotamos como p_t al precio de un valor en un día t y como n al número de observaciones elegido para el cálculo de la media móvil, tenemos que:

$$MMS = \frac{\sum_{i=0}^{k} p_{t-i}}{n}$$

- **Media móvil ponderada:** media ponderada. Se da distintos pesos a las observaciones de manera que las más recientes tengan un mayor impacto en el cálculo de la media móvil.

$$MMP = \frac{\sum_{i=0}^{n-1} p_{t-i}(n-i)}{(n+1)\frac{n}{2}}$$

- **Media móvil exponencial:** tipo de media móvil ponderada que posee un multiplicador acotado entre 0 y 1 que depende del número de periodos de la media móvil. Se diferencian de las medias ponderadas en que este multiplicador hace que sean más sensibles a la hora de detectar cambios de tendencia.

  Si denotamos dicho multiplicador como α y definimos la media móvil exponencial del periodo anterior como MME^{t-1}, tenemos que:

$$MME = \alpha(p_t) + (1-\alpha)(MME^{t-1})$$

En relación con las medias móviles es habitual construir otro indicador conocido como **bandas de Bollinger**[21], que se colocan por encima y por debajo de la media móvil, a dos desviaciones típicas de distancia, siendo tres el total de bandas representado. El espacio entre las bandas varía en función de la volatilidad de manera que, en periodos donde esta es alta, las bandas se distancian, acercándose en caso contrario.

Figura 3.5: Media móvil. Tomado de [20].

Figura 3.6: Bandas de Bollinger. Tomado de [21].

Otros indicadores

Más allá de los indicadores comentados anteriormente, hoy en día se utilizan multitud de indicadores técnicos. Algunos de ellos, incluidos en la aplicación web construida como objetivo de este trabajo, son los siguientes [22]:

- **Momento:** indica la velocidad media de cambio de precio durante un periodo de tiempo, permitiendo anticipar un posible cambio futuro en el movimiento del mercado. Es un indicador sencillo, versátil y de gran popularidad debido a su carácter predictivo del mercado, ya que no solo reacciona a la dirección en la que se mueven los precios, sino que también cambia su dirección antes de que estos lo hagan.

$$momento = (cierre(actual) - cierre(NperiodosAtras)$$

- **MACD (*Moving Average Convergene Divergence)*:** [23] mide la convergencia y divergencia en el tiempo de dos medias móviles del precio de un activo, es decir, indica la separación entre dos medias móviles calculadas sobre distinto periodo de tiempo. Lo habitual es emplear dos medias móviles exponenciales, donde una sea de un corto periodo de cálculo y otra sea de un periodo medio (habitualmente 12 y 26 periodos respectivamente). Cuanto más corto es el periodo de cálculo, más sensible es la media móvil a la variación del precio. La importancia de este indicador se debe a su capacidad informativa sobre la fortaleza de los movimientos de los precios, indicando cómo de fiable es un movimiento y permitiendo realizar acciones en consecuencia.

$$MACD = EMA(periodo1) - EMA(periodo2)$$

- **RSI (*Relative Strength Index)*:** también sirve para analizar la fortaleza de un movimiento de un precio, ya que evalúa si un activo está siendo vendido o comprado en demasía para su valor actual, dando la idea de si un valor está *barato* o *caro*. Es un indicador de tipo oscilador, cuyo valor se encuentra entre 0 y 100. Un RSI inferior a 30 indica que el activo está siendo sobre-vendido (activo infravalorado), mientras que un RSI superior a 70 indica lo contrario, una sobre-compra (activo sobrevalorado). En su cálculo aparecen medias móviles, habitualmente de 14 días.

  $$RSI = 100 - \frac{100}{1 + RS}$$

  Donde $RS = \frac{MME(NperiodosAlcistas)}{MME(NperiodosBajistas)}$

- **ROC (*Rate of Change*):** [24] mide la tasa de cambio del precio entre periodos. Muy relacionado con el indicador *momento*, aporta una información similar, aunque expresada como porcentaje.

  $$\frac{cierre(actual) - cierre(NperiodosAtras)}{cierre(NperiodosAtras)} * 100$$

- ***Stochastic*:** sirve para evaluar el *momento*, ya que antes de que se produzca un cambio de tendencia, es habitual que el precio muestre señales de agotamiento. Se compone de dos líneas $\%K$ y $\%D$, que miden la capacidad de un precio de cerrar su valor cerca del máximo o mínimo del día:

  $$\%K = 100 * \frac{C - Ln}{Hn - Ln}$$

  donde C es el precio de cierre actual, Ln es el precio más bajo de los últimos n periodos y Hn es el precio más alto de los últimos n periodos.

  $\%D$ se define como la media móvil de $\%K$ durante N periodos.

- **A/D (*Accumulation/Distribution*):** utiliza el volumen y el precio para evaluar cuándo un valor está en fase de acumulación o de distribución. Trata de identificar divergencias entre el precio y la dirección del volumen, proporcionando información sobre cómo de fuerte es una tendencia. Siguiendo la notación anterior

  $$A/D = \frac{(C - Ln) - (Hn - C)}{Hn - Ln}$$

- **CCI (*Commodity Channel Indicator*):** indicador multifunción que puede ser empleado para identificar nuevas tendencias. De forma generalizada, se dice que mide el nivel actual del precio en relación a un nivel de precio promedio durante un periodo de tiempo determinado. Tendrá un valor alto si los precios están relativamente por encima de su promedio y bajo en el caso opuesto.

  $$CCI = \frac{precioTipico - promedioMovil}{0{,}015 * desviacionMedia}$$

  Donald Lambert (el creaor de este indicador) utilizó la constante $0{,}015$ con la intención de que, entre el $70\,\%$ y el $80\,\%$ de los valores de CCI se encuentren entre $+100$ y -100, siendo los valores que no entran en este intervalo considerados como desviaciones inusuales a la media.

- **%B:** [25] cuantifica el precio de un valor en relación a las bandas de Bollinger:
  - Inferior a 0 cuando el valor está por debajo de la banda inferior.
  - Igual a 0 cuando el valor está en la banda inferior.
  - Entre 0 y $0{,}5$ cuando el valor está entre la banda inferior y la banda media.
  - Entre $0{,}5$ y 1 cuando el valor está entre la banda media y la banda superior.
  - 1 cuando el valor está en la banda superior.
  - Superior a 1 cuando el valor está por encima de la banda superior.

$$\%B = \frac{precio - bandaInferior}{bandaSuperior - bandaInferior}$$

- **OBV (*On Balance Volume*):** [26] utiliza el volumen para predecir cambios en los precios.

$$OBV = OBV_{previo} + \begin{cases} volumen & si \quad cierre > cierre_{previo} \\ 0 & si \quad cierre = cierre_{previo} \\ -volumen & si \quad cierre < cierre_{previo} \end{cases}$$

- **ATR (*True Range/Average True Range*):** [27] mide la volatilidad del mercado descomponiendo el rango del precio de un activo en un periodo determinado. Siguiendo la notación de otros indicadores:

$$ATR = \left(\frac{1}{n}\right)\sum_{i=1}^{n} TR_i$$

  donde $TR_i = Max[(Hn - Ln), Abs(Hn - C), Abs(Ln - C)]$

- **ADX (*Welles Wider's Directional Movement Index*):** [28] oscilador que se mueve entre 0 y 100 y que da una idea de la fortaleza de una tendencia, por lo que se utiliza para determinar si un mercado se encuentra en situación lateral, si está oscilando en un rango o si está comenzando una nueva tendencia.
  - $0 - 25$: ausencia de tendencia.
  - $25 - 50$: tendencia fuerte.
  - $50 - 75$: tendencia muy fuerte.
  - $75 - 100$: tendencia extremadamente fuerte.

$$ADX = \frac{+DI - (-DI)}{+DI + (-DI)}$$

  Donde $+DI$ se define como la línea del indicador de dirección positiva y $-DI$ como la línea del indicador de dirección negativa, calculándose ambas como: $+DI = \frac{+DM}{TR}$ y $-DI = \frac{-DM}{TR}$ Donde $+DM$ es la suma de los movimientos positivos, $-DM$ es la suma de los movimientos negativos y TR el *true range* (rango verdadero), que es en el caso de movimientos positivos, el mayor de estos tres:
  - El máximo del día menos el mínimo.
  - El máximo del día menos el cierre del día anterior.
  - El cierre del día anterior menos el mínimo del día.

Teoría de Dow

Si hablamos de análisis técnico resulta obligado comentar, aunque sea brevemente, las teorías expuestas por Charles Dow, ya que la mayor parte del conocimiento actual sobre análisis técnico tiene sus bases en las teorías que este estadounidense publicó en el *Wall Street Journal* en el siglo XIX [29].

Las ideas principales que presentó Dow fueron las siguientes:

- **Las medias lo descuentan todo:** la suma y tendencia de las transacciones sobre un valor representan el total de todo el conocimiento pasado sobre dicho valor, el inmediato y el futuro. Por esto, no hay necesidad de añadir elaborados artificios matemáticos a las medias. Esta afirmación se traduce en que los mercados reflejan en sus precios todo factor que afecte de uno u otro modo a la oferta y la demanda.
- **El mercado tiene tres tendencias:** en primer lugar, cabe decir que Dow definía una tendencia ascendente como una situación en la que una recuperación cierra con un valor más alto que el nivel más alto de la recuperación previa. Del mismo modo, el nivel bajo también cierra por encima del nivel más bajo de la recuperación previa. Dow también afirmaba que las leyes de acción reacción del mundo físico se aplicaban a los mercados financieros y describía la tendencia dividiéndola en tres partes [30]:
  - **Primaria:** duración superior a un año, generalmente de entre uno y tres. Es la dirección principal del mercado.
  - **Secundaria:** duración de entre tres semanas y tres meses. Se mueve dentro de la primaria pero va en contra de ella, retrocediendo entre un tercio y dos tercios del movimiento anterior a la tendencia, frecuentemente en torno a un $50\,\%$. Se la define como correcciones a la tendencia principal.
  - **Menor:** duración inferior a tres semanas. Representa las fluctuaciones de la tendencia secundaria y se mueve en contra de ella.
- **Las tendencias principales tienen tres fases:** Dow consideraba que la tendencia principal se desarrolla en tres fases claramente diferenciables:
  - **Fase de acumulación:** la tendencia anterior era descendente y lo sigue siendo pero de manera estable, el mercado no bajará más. En esta fase los inversores más astutos realizan sus compras.
  - **Fase de participación pública:** los valores comienzan a subir rápidamente. La mayoría de inversores (que siguen tendencias) entran a jugar en esta fase.
  - **Fase de distribución:** las noticias sobre el mercado son alcistas y las económicas son positivas. Se incrementa el volumen especulativo y la participación pública. Los inversores que comenzaron a *acumular* en las fases de acumulación comienzan a *distribuir* antes de que el resto de inversores comiencen a vender.
- **Las medias deben confirmarse entre ellas:** no existe un cambio de tendencia hasta que dos índices (uno del sector y otro del mercado concreto) van en la misma dirección. Dow considera que si uno cambia de tendencia pero el otro la mantiene, la tendencia anterior todavía permanece.
- **El volumen debe confirmar la tendencia:** el volumen debe moverse en la dirección de la tendencia principal. En tendencia ascendente el volumen se incrementa conforme los valores suben y en una tendencia descendente el volumen aumenta a medida que las cotizaciones caen. De este modo, el volumen debe aumentar cuando los precios van en la dirección de la tendencia y disminuir en caso contrario. Para Dow, el volumen era un indicador secundario, siendo el principal los precios de cierre de los valores.
- **Una tendencia está en vigor hasta que otra la sustituye:** entiende que las tendencias funcionan de manera análoga a la *primera ley de Newton*. Este principio trata de explicar las transiciones entre una tendencia alcista y una bajista, con especial énfasis en la capacidad de reconocer estos cambios frente a correcciones.

Aunque la teoría de Dow estableció las bases del análisis técnico actual, no fue ni mucho menos perfecta. Una crítica realizada a esta teoría se basa en que Dow trató de reconocer la aparición de tendencias, pero nunca anticiparlas

Construcción de gráficos

Las herramientas gráficas se consideran de gran utilidad a la hora de realizar un análisis técnico de un valor. El estudio de las distintas representaciones gráficas aporta información de gran utilidad que, explotada por un experto (habitualmente denominados *chartistas* en la jerga) puede ser ampliamente valorado a la hora de realizar movimientos.

El tipo de gráfico realizado depende de si el inversor pretende operar a corto, medio o largo plazo, ya que las periodicidades empleadas pueden variar desde datos horarios a incluso anuales. Habitualmente [15] en la inversión a corto plazo se usan marcas temporales horarias, en la inversión a medio plazo se utilizan mediciones diarias y en la inversión largoplacista se utilizan datos semanales o mensuales.

Los gráficos más habitualmente utilizados en análisis técnico son los siguientes:

- **Gráfico de barras:** aporta información del precio de apertura, el precio máximo, el mínimo y el de cierre de un valor en un tiempo establecido. Es habitual acompañarlo de los volúmenes.

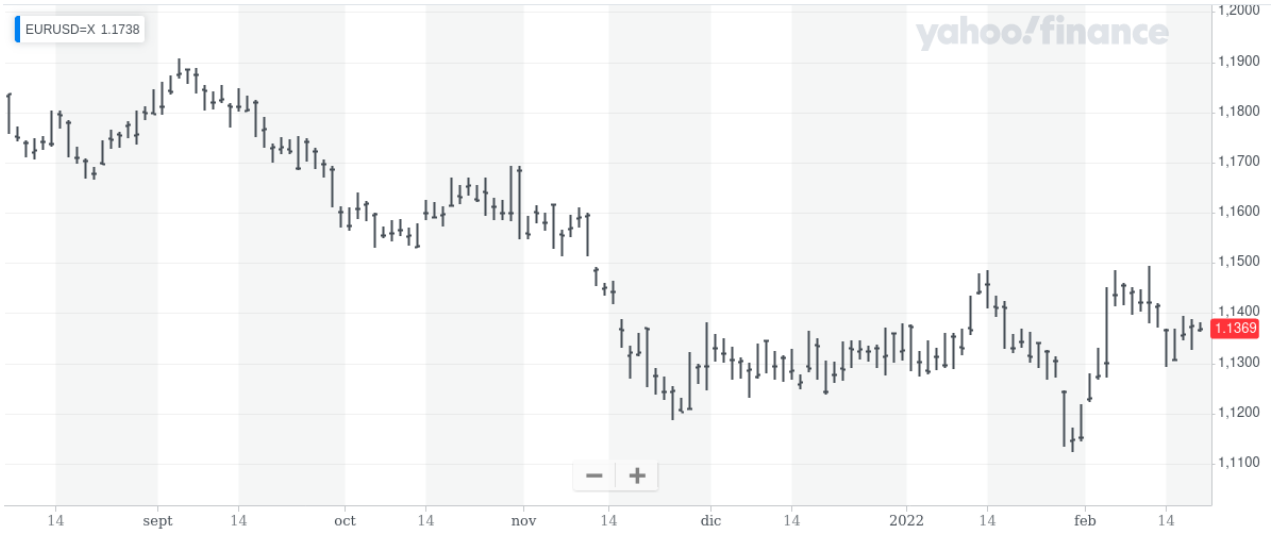


Figura 3.7: Gráfico de barras EUR/USD Yahoo Finance 12/02/2022.

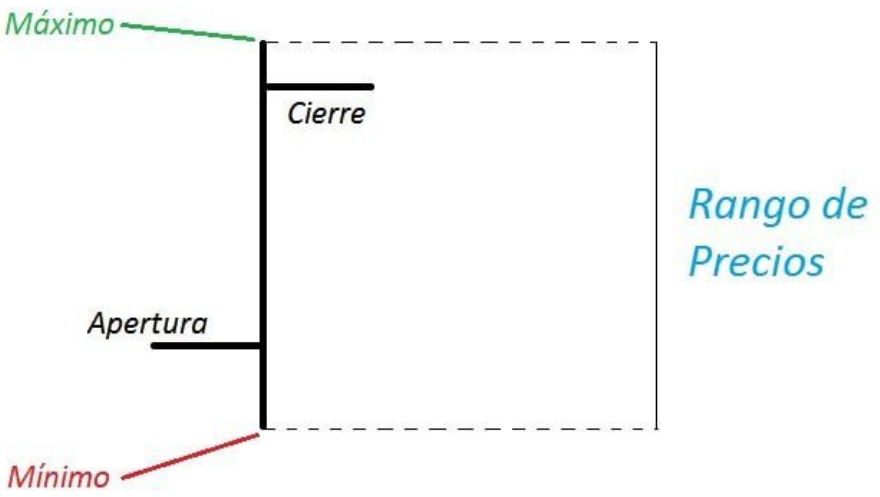


Figura 3.8: Descripción de una barra. Tomado de [31].

- **Gráfico de líneas:** representa las variaciones en el precio de cierre de un valor durante un periodo concreto, con una periodicidad establecida. Ver Figura 3.9.
- **Gráfico de velas:** versión japonesa de los gráficos de barras, los más utilizados a día de hoy por los chartistas. Muestran la misma información que los gráficos de barras, aunque de un modo diferente. Una línea delgada denominada *sombra* representa el alcance de un precio desde su mínimo hasta el máximo y una parte más ancha de la barra (*cuerpo*) mide la distancia entre el precio de apertura y el de cierre. En función de si el cierre es más alto que la apertura (precio sube) o si es más bajo (precio baja) la parte gruesa se colorea de un color u otro (habitualmente negro o rojo para las bajadas y blanco o verde para las subidas). Esta es la clave de la popularidad de este tipo de gráficos. Ver Figura 3.10.

Figura 3.9: Gráfico de líneas EUR/USD. Yahoo Finance. 12/02/2022.

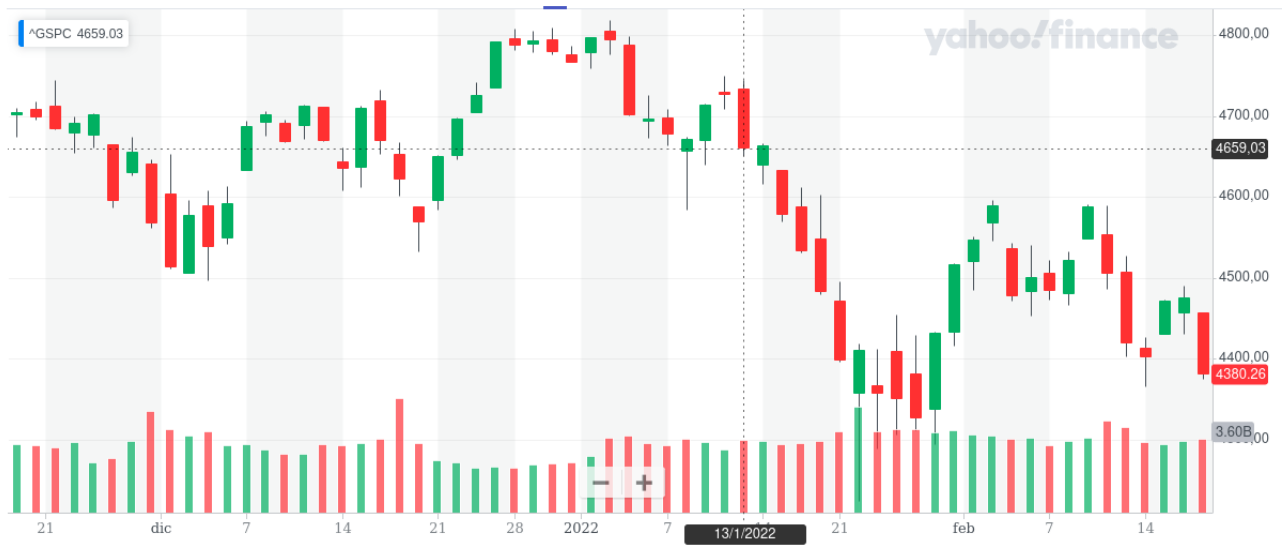


Figura 3.10: Gráfico de velas S&P 500. Yahoo Finance. 12/02/2022.

Análisis e interpretación de gráficos

A la hora de analizar gráficos en análisis técnico, el concepto que resulta principal observar es la tendencia [29]. Redefiniendo este concepto de modo un poco más preciso de lo realizado anteriormente, se puede considerar que expresa la dirección del mercado en base a los movimientos de *zigzag* que este realiza, de manera semejante a una ola con *picos* y *valles*. Es la dirección de estos picos y estos valles la que indica la tendencia del mercado, de forma que una sucesión de estos, cada vez más elevada representa una tendencia alcista y una sucesión cada vez más baja, una tendencia bajista. Los picos y valles horizontales identifican una tendencia lateral. Es importante destacar que, para que una tendencia se considere como tal, ha de mantenerse al menos en tres puntos consecutivos.

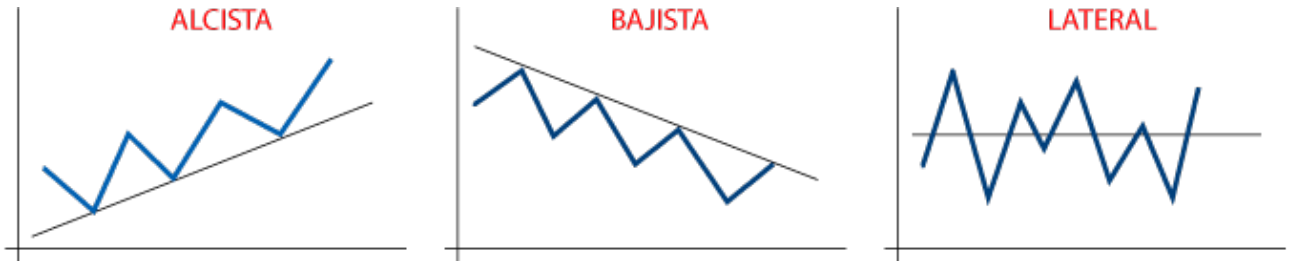


Figura 3.11: Gráficos de tendencias. Tomado de [32].

Es importante describir la tendencia en base a estas tres direcciones mencionadas (ver 3.11), ya que, pese a lo que habitualmente se tiende a pensar, los mercados no se mueven únicamente hacia arriba o hacia abajo, los desplazamientos laterales son muy importantes y constituyen una banda de fluctuación. Este tipo de movimientos reflejan un periodo de equilibrio en los precios, donde las fuerzas de la oferta y la demanda se enfrentan con relativa igualdad.

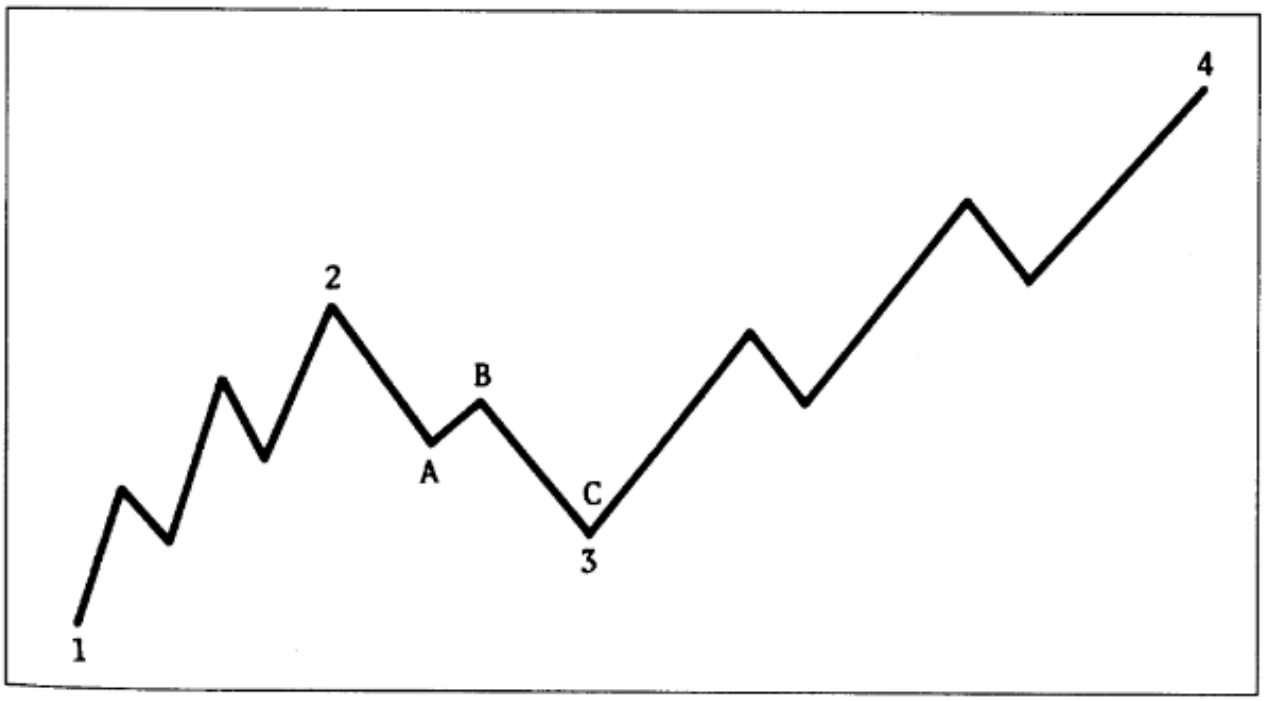


Figura 3.12: Distintas tendencias. Tomado de [29].

Observando a su vez la Figura 3.12 se aprecia que los puntos 1, 2, 3 y 4 describen la tendencia principal, la cual es ascendente. Los puntos 2 y 3 representan una *corrección* (cambio brusco de tendencia provocado cuando el mercado aumenta de manera muy grande y se produce una fuerte caída que equilibra los precios sobrevalorados) secundaria dentro de esta tendencia principal, dividiéndose a su vez cada una de estas *ondas* secundarias en tendencias de menor duración.

Otros dos conceptos importantes a la hora de describir los gráficos de tendencia son el de *apoyo* o *soporte* y el de *resistencia*. Describiendo con más precisión a los *picos* y *valles* antes mencionados como mínimo y máximos, podemos a su vez nombrarlos como *apoyos* y *resistencias* respectivamente.

- **Apoyo:** área del gráfico por debajo del mercado. Hay una bajada de precios que se detiene y estos vuelven a subir. Suelen venir precedidos de un mínimo previo.

- **Resistencia:** área del gráfico por encima del mercado. El precio vuelve a retroceder y del mismo modo que los soportes, suele venir precedido de un pico anterior.

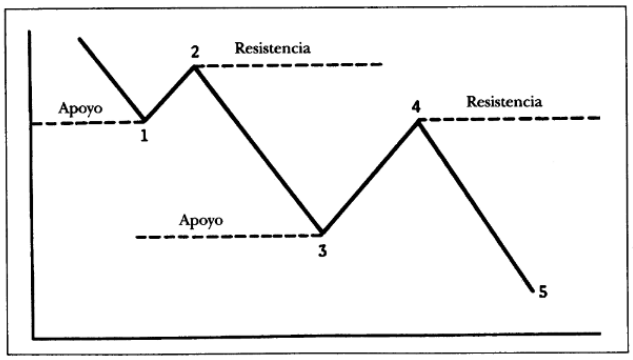


Figura 3.13: Apoyos y resistencias en mercado bajista. Tomado de [29].

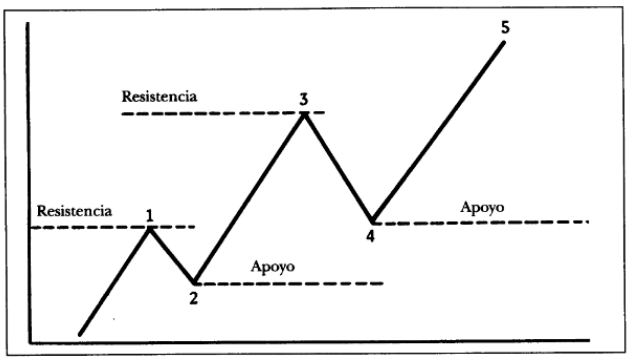


Figura 3.14: Apoyos y resistencias en mercado alcista. Tomado de [29].

Finalmente, un último concepto a mencionar en relación con esto último sería el de *ruptura*, que se produce cuando el precio de un valor rompe, o bien un apoyo, o bien una resistencia, indicando un movimiento en la misma dirección.

3.3 Bolsa

La bolsa de valores [33] es una organización pública o privada (autorizada por el gobierno de los países) que brinda las facilidades necesarias para la realización de negociaciones de compra y venta de valores. Algunos de los instrumentos de inversión que se negocian en la bolsa son las acciones de empresas o sociedades, los bonos públicos o los títulos de participación. La bolsa presta el servicio de poner en contacto a empresas que requieren de capital y a personas o empresas que desean invertir su dinero a cambio de recibir una retribución representada en la parte que les corresponde de las utilidades de la empresa.

Las bolsas de valores [34] impulsan el desarrollo económico y financiero de la mayoría de países del mundo. Su origen se encuentra en Amberes, Bélgica, en el año 1460, aunque una familia de banqueros de Brujas organizaba en su palacio reuniones donde se transaban activos y se hacían operaciones comerciales en el siglo $XIII$.

Función económica

Algunas de las funciones económicas que cumplen las bolsas de valores son las siguientes [15]:

- Canalizan el ahorro hacia la inversión, contribuyendo de esa manera al desarrollo económico del país.
- Ponen en contacto a las empresas y entidades estatales que necesitadas de recursos de inversión con los ahorradores dispuestos a invertir su capital.
- Aportan liquidez a las inversiones, de manera que los poseedores de títulos de bolsa pueden convertir en dinero sus acciones con facilidad.
- Certifican precios de mercado.
- Favorecen la asignación eficiente de recursos.
- Contribuyen a la correcta valoración de los activos financieros.

Participantes

En la bolsa de valores intervienen los siguientes agentes:

- **Intermediarios:** agentes que ponen en contacto a vendedores y compradores. Son las casas de bolsa, agentes de bolsa, corredores de bolsa, agencias y sociedades de valores. Los corredores de bolsa se encargan de negociar por los intermediarios a cambio de una comisión y solo aquellos que están autorizados pueden desempeñar tal labor.
- **Inversionistas:** personas o entidades que desean canalizar su inversión hacia los títulos de valores. Los inversionistas a corto plazo arriesgan en mayor medida su capital que los inversionistas a largo plazo, que buscan obtener rentabilidad a través de dividendos (proporción de beneficios que la empresa reparte entre sus accionistas) o ampliaciones de capital (incremento de los recursos de una empresa para acometer nuevas inversiones o como modo de financiación).
- **Emisores:** empresas o entes públicos que buscan financiar su actividad a través de la emisión de títulos de valores en el caso de las primeras o de deuda en caso de los segundos. Una empresa ha de estar inscrita en la bolsa para poder vender sus valores, para lo cual necesita acreditar solvencia y seriedad.

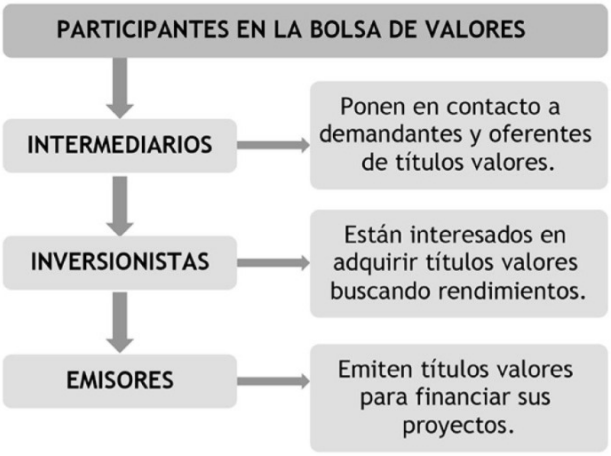


Figura 3.15: Participantes de la bolsa de valores. Tomado de [15].

3.3.1 S&P 500

Un índice bursátil es un registro estadístico que trata de reflejar las variaciones y rentabilidad promedio de las acciones que lo componen. Estas suelen presentar características comunes como pertenecer a la misma bolsa o sector de negocio.

El *Standard and Poor's 500* [35] es uno de los índices más importantes de Estados Unidos y del mundo, siendo considerado como el índice más representativo de la situación real del mercado.

Este índice actualmente se basa en las capitalizaciones bursátiles de 500 empresas estadounidenses y captura aproximadamente el $80\,\%$ de toda la capitalización del mercado estadounidense, aunque en su origen en 1923 únicamente estaba compuesto por 233 compañías.

La ponderación de las empresas de este índice está determinada por *S&P Dow Jones Indices* y se diferencia de otros índices de Estados Unidos en la metodología utilizada para ello.

Las empresas que forman parte de este índice se seleccionan en base a criterios como la capitalización bursátil (que ha de superar los 4200 millones), la liquidez, el capital flotante, la clasificación del sector, la viabilidad económica o el tiempo que ha cotizado en bolsa entre otros. Según el sector en el que desarrollan su actividad económica [36] las empresas se agrupan en 11 sectores, debido a que es habitual que cada sector evolucione de manera diferente en función del contexto económico:

- **Sector Consumo Discrecional (*Consumer Discretionary*):** compuesto por empresas cuya demanda fluctúa dependiendo de la situación económica general, incluye el comercio minorista, hoteles, cadenas de restaurantes o fabricantes de automóviles entre otros. Algunos ejemplos de empresas son *Amazon, Netflix, Nike* y *General Motors*.

- **Sector Consumo Básico (*Consumer Staples*):** compuesto por empresas de productos básicos para el consumidor, tales como comida, bebidas y productos de higiene personal. Algunos ejemplos de empresas son *Coca-Cola* y *Costco*.

- **Sector Energía (*Energy*):** compuesto por empresas que exploran y explotan hidrocarburos como el petróleo y el gas natural, así como las refinerías que los tratan y otras compañías que les suministran equipamiento. Algunos ejemplos de empresas son *Chevron* y *ExxonMobil*.

- **Sector Financiero (*Financials*):** compuesto por bancos comerciales, banca de inversión y compañías aseguradoras. Algunos ejemplos de empresas son *Bank of America* y *JPMorgan Chase*.

- **Sector Salud (*Healthcare*):** compuesto por hospitales, compañías farmacéuticas, de equipamiento médico y empresas de biotecnología. Algunos ejemplos de empresas son *Pfizer* y *Merck*.

- **Sector Industria (*Industrials*):** compuesto por fabricantes de diversos tipos como aerolíneas, aeroespaciales, construcción, maquinaria,etc. Algunos ejemplos de empresas son *Boeing* y *General Electric*.

- **Sector Tecnología de la Información (*Technology*):** compuesto por fabricantes de hardware, software, semiconductores y equipamiento tecnológico. Algunos ejemplos de empresas son *Apple, IBM, Cisco* y *Microsoft*.

- **Sector Materiales (*Materials*):** compuesto por empresas metalúrgicas, de minería, químicas y relacionadas con materias primas. Algunos ejemplos de empresas son *Newmont Corp* y *DuPont*.

- **Sector Inmobiliario (*Real State*):** compuesto por empresas relacionadas de una otra forma con el mercado de inmuebles. Algunos ejemplos de empresas son *Public Storage* y *Simon Property Group*.

- **Sector Servicios de Comunicación (*Communication services*):** compuesto por compañías telefónicas, proveedores de servicios de Internet y medios de comunicación tradicionales, así como por compañías de redes sociales y medios digitales. Algunos ejemplos de empresas son *Alphabet* y *Facebook*.

- **Sector Servicios Públicos (*Utilities*):** compuesto por compañías de electricidad, agua, gas y energías renovables. Algunos ejemplos de empresas son *Pacific Gas & Electric* y *American Electric Power*.

3.3.2 NASDAQ

National Association of Securities Dealers Automated Quotation [37] es el segundo mercado de valores y bolsa de valores automatizada y electrónica más grande de los Estados Unidos después de la Bolsa de Nueva York. Es el mercado mundial con mayor intercambio por hora del mundo. Se caracteriza por comprender las principales empresas de alta tecnología en electrónica, informática, telecomunicaciones y biotecnología del país. Sus índices más representativos son el Nasdaq-100 y el Nasdaq-Composite.

NASDAQ-100

Este índice se compone por las 100 empresas más grandes en términos de capitalización, de entre las que se comercializan en la bolsa de valores NASDAQ.

Algunas de las empresas que componen este índice son: *Amazon, Apple, Alphabet, Cisco, Facebook, eBay, Microsoft, Nvidia, Netflix, PayPal* o *Tesla*.

3.3.3 IBEX 35

El IBEX 35 [38] es el índice bursátil de referencia en España, elaborado por *Bolsas y Mercados Españoles (BME)*. Está formado por las 35 empresas con mayor liquidez de las que cotizan en el *Sistema de Interconexión Bursátil Español (SIBE)* en las cuatro bolsas españolas (Madrid, Barcelona, Bilbao y Valencia). Al igual que el S&P 500 es un índice ponderado, donde no todas las empresas tienen el mismo peso y también se divide en sectores [39]:

- **Petróleo y energía:** compuesto por empresas que exploran y explotan hidrocarburos como el petróleo y el gas natural, así como aquellas dedicadas a la electricidad o el agua. Algunos ejemplos de empresas son *Iberdrola*, *Endesa* y *Repsol*.

- **Materiales básicos, industria y construcción:** compuesto por empresas dedicadas a los minerales, metales, las actividades de la construcción y las relaionadas con la ingeniería, la química y la industria aeroespacial. Algunos ejemplos de empresas son *Acerinox, Acciona, Abengoa* y *ACS*.

- **Bienes de consumo:** compuesto por empresas dedicadas a productos básicos para el consumidor, incluyendo las relacionadas con la actividad agrícola, la ganadera y la industria textil. Algunos ejemplos de empresas son *Pescanova* e *Inditex*.

- **Servicios de consumo:** compuesto por compañías dedicadas al ocio, relacionadas con el arte, los espectáculos o la gestión de eventos e instalaciones deportivas entre otros. También se incluyen en este sector los medios de comunicación y las empresas de transporte y distribución. Algunos ejemplos de empresas son *Atresmedia*, *Mediaset* y *Prosegur*.

- **Servicios financieros:** compuesta por empresas dedicadas a la economía y los seguros, así como aquellas dedicadas a la inversión. Algunos ejemplos de empresas son *Banco Santander, BBVA* y *Mapfre*.

- **Tecnología y telecomunicaciones:** compuesto por empresas dedicadas a las telecomunicaciones: telefonía, redes e infraestructuras de comunicaciones. Algunos ejemplos de empresas son *Indra* y *Telefónica*.

- **Servicios inmobiliarios:** compuesto por empresas dedicadas a la gestión y promoción de bienes inmuebles. Algunos ejemplos de empresas son *Montebalito* y *Realia Business*.

3.3.4 EURO STOXX 50

El *EURO STOXX 50* [40] es un índice compuesto por las 50 principales empresas de la eurozona en términos de capitalización bursátil. Es un índice diseñado por *Stoxx Ltd* y creado e1 1998 que está compuesto por empresas de 8 países: España, Francia, Alemania, Bélgica, Irlanda, Italia, Países Bajos y Finlandia. Algunas empresas que forman parte de este índice son *Adidas*, *Inditex* y *Philips*.

3.4 Criptodivisas

Las criptodivisas (o criptomonedas) [41] son nuevos tipos de divisa o moneda de carácter digital que utilizan criptografía (matemáticas avanzadas) y avanzadas técnicas informáticas con el fin de no depender de las entidades centralizadas (bancos y gobiernos) que emiten y controlan el dinero.

Este tipo de dinero *P2P (Peer to peer)* es totalmente digital, lo que permite envíos a cualquier parte del mundo en cuestión de segundos sin necesidad de la intervención de intermediarios como los bancos. Las cripotomonodedas pueden ser intercambiadas por bienes y servicios como habitualmente se hace con el dinero Fiat, al menos en aquellas entidades que permiten operar con estos nuevos tipos de divisas.

Además de la descentralización comentada, las criptomonedas surgen con el objetivo de proporcionar seguridad, transparencia y privacidad a las transacciones financieras, características sustentadas en la tecnología *blockchain*.

Un último aspecto a comentar en esta breve introducción sobre criptodivisas es el concepto de capitalización del mercado (*market cap*) [42]. Este término hace referencia a un indicador que sigue el valor de mercado de una criptomoneda, midiendo su dominio y su popularidad. El valor de este indicador se obtiene multiplicando el precio por la cantidad de activo en circulación, por lo que se puede decir que mide el valor total de una criptomoneda. A su vez, conocer la capitalización del mercado de una criptomoneda permite realizar una nueva clasificación que pretende ayudar a la toma de decisiones del inversor:

- ***Large-cap*:** superior 10 billones de dólares americanos. Se consideran inversiones de bajo riesgo al tener un historial de crecimiento demostrado y, a menudo, presentar mayor liquidez. Esto se traduce en una mayor capacidad de soportar que un gran número de personas vendan sus activos sin que se produzca un desplome en el precio.

- ***Mid-cap*:** entre 1 y 10 billones de dólares americanos. Se considera que tienen mayor potencial sin explotar pero más riesgo en contraposición.

- ***Small-cap*:** menos de 1 billón de dólares americanos. Muy volátiles y arriesgadas ante los posibles cambios del mercado y en el comportamiento de los inversores.

3.4.1 Blockchain

La blockchain (cadena de bloques) es una red de datos sostenida por una red de ordenadores descentralizada global que se basa en técnicas de criptografía asimétrica avanzada, haciendo uso de sistemas de *clave pública - clave privada*.

Sostenida por miles de ordenadores en todo el mundo (nodos), funciona de forma descentralizada y por consenso. La blockchain consiste una forma de registro de operaciones, llevándose en ella la cuenta de todas las unidades de una criptodivisa que hay en circulación y todas las operaciones realizadas. Por ello se puede decir que funciona a modo de libro contable, teniendo cada criptodivisa el suyo propio y estando estos a su vez enlazados y cifrados. Ese libro de cuentas [43] no solo está distribuido y es seguro: los bloques enlazados (de ahí lo de cadena de bloques) cuentan con un puntero *hash* (codificado) que enlaza al bloque anterior, además de una marca de tiempo y los datos de la transacción, y esa información es pública

El hecho de que la cadena de bloques se diga que está distribuida se debe a que en toda blockchain existe una gran red de ordenadores conectados por todo el mundo (los nodos antes mencionados) que se encargan de almacenar y validar la información de la misma. Gracias a esto hay millones de copias de seguridad sincronizándose en todo momento para salvaguardar la información al instante. Así, la información de la creación y distribución del dinero mediante las transacciones, queda registrada de forma inmutable. La Figura 3.16 muestra el funcionamiento de la blockchain.

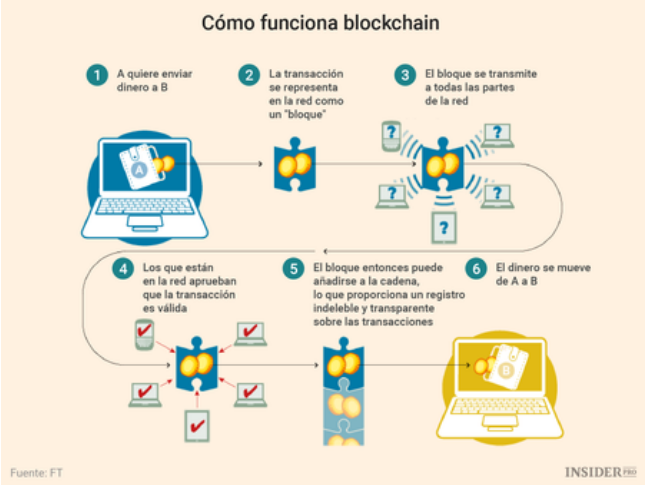


Figura 3.16: Funcionamiento de la blockchain. Tomado de [43].

Cuando un bloque ya no admite más transacciones, llega un momento importante: el de validarlo o sellarlo, que es lo que los usuarios hacen cuando hacen *minería*. La minería es un proceso compuesto por complejos cálculos matemáticos que requieren grandes inversiones de tiempo y de electricidad y que tiene como resultado la realización de un registro permanente en la cadena de bloques, lo que impide que este sea modificado sin alterar todos los bloques que están enlazados con él, operación imposible en la práctica al necesitarse que la mayoría de nodos la validen (siendo esto extremadamente costoso). Como los sistemas blockchain son sistemas *P2P*, los mineros reciben avisos de las nuevas transacciones. En este punto, se ponen a competir para ver quién es el primero que logra crear un bloque válido y sellarlo, recibiendo una recompensa a cambio.

Debido al uso de una cadena común sincronizada entre nodos se logra la irreversibilidad de las transacciones, lo que evita el fraude y hace que el sistema sea muy seguro.

3.4.2 Bitcoin

En cuanto a lo que criptomonedas se refiere, es obligatorio hablar sobre Bitcoin (BTC) [44] (su logotipo se muestra en la Figura 3.17), la primera que surgió en el mundo y la más importante de todas.

Tomando como base el proyecto *DigiCash* promulgado por *David Chaum*, en el año 2008 una entidad (hasta la fecha desconocida) conocida como *Satoshi Nakamoto* propuso una serie de mejoras que establecieron los principios de los sistemas blockchain y plasmó un nuevo proyecto conocido como Bitcoin que, sin embargo, no vio la luz en su primera versión hasta el año 2009.

Con el paso de los años, el proyecto original de *Nakamoto* fue sufriendo mejoras y actualizaciones, constituyendo la piedra angular de todo el ecosistema de criptomonedas que se ha ido desarrollando en la última década y siendo, sin duda, el proyecto de criptodivisa más fuerte y con más futuro, habiendo llegado a alcanzar una cotización de más de $50{,}000$ dólares y habiendo llegado a ser establecida como moneda oficial en países como El Salvador [45].

La emisión máxima del Bitcoin no puede superar los 21 millones de unidades. Además, tiene la capacidad de subdividirse hasta los ocho decimales. De esta manera, un Bitcoin puede fraccionarse hasta en $100{,}000{,}000$

unidades más pequeñas que reciben el nombre de *satoshis* y que equivalen a $0{,}00000001$ BTC.

En contraposición a todos sus aspectos positivos, Bitcoin aún tiene importantes retos que resolver antes de consolidarse como un nuevo sistema económico en sí mismo, como son el gran consumo energético que requiere la *prueba de trabajo (proof of work (PoW)* necesaria para sellar un Bitcoin o el exceso de especulación que conlleva que sea un sistema cuyo valor depende de la aceptación del mismo, al no estar respaldado por metales preciosos como el dinero *fiat*.

Figura 3.17: Logotipo de Bitcoin.

3.4.3 Otras criptodivisas

Pese a que Bitcoin sea la criptomoneda más importante y sobre la que gira todo el ecosistema *cripto*, existen multitud de proyectos que también se basan en este tipo de tecnologías, las denominadas *altcoins* o monedas alternativas a Bitcoin, que surgen con el objetivo de diversificar y mejorar el ecosistema *cripto* (aunque problemas como la especulación y la duplicidad entre proyectos son claramente patentes en este tipo de criptodivisas).

Ethereum (ETH) [46] es otro importante proyecto que también está claramente consolidado a día de hoy. Diferenciándose de *Bitcoin* en su precio (muy inferior), en su nivel de descentralización, en su escalabilidad y en su proceso de minería, parte con la intención de ser utilizado en la programación de contratos inteligentes (*smart contracts*), que son programas informáticos que facilitan, aseguran, hacen cumplir y ejecutan acuerdos registrados entre dos o más partes (personas o entidades).

Además de las criptomonedas comentadas, existen otros importantes proyectos como son **Cardano (ADA)**, **Polkadot (DOT)**, **Ripple (XRC)** o **Litecoin (LTC)** entre muchos otros. Algunos de sus logotipos des muestran en la Figura 3.18.

3.5 Materias primas

El mercado de materias primas [47] conocidas como *commodities* es una rama de la inversión que permite operar con productos físicos, a diferencia de otros mercados como la bolsa.

El funcionamiento de este mercado es muy similar al de los mercados de renta variable, ya que, si bien se opera con bienes físicos, no es necesario adquirirlos de manera directa. Los precios varían también en los mismos periodos de tiempo y se compran y venden como si fueran acciones. En cuanto a la inversión, la principal diferencia radica en que en ocasiones la inversión en materias primas incluye un sobre-coste debido al transporte, seguro y almacenamiento del activo, aunque lo más habitual es invertir a través de lo que se conoce como *ETFs*, que son conjuntos diversificados de activos que que cotizan en bolsa.

En el mercado de materias primas, el modo de negociar es mediante lo que se conoce como *contratos de futuros*, donde cada contrato equivale a una cantidad de producto especificada.

Figura 3.18: Logotipo de las principales criptomonedas de la actualidad.

En el mundo [48] hay unos 50 mercados que cotizan materias primas, siendo los más importantes la *Bolsa de Metales de Londres (LME)*, la *Chicago Board of Trade (CBOT)* y la *New York Merchantile Exchange (NYMEX)*.

3.5.1 Oro

El oro ha sido utilizado como dinero desde hace más de $5{,}000$ años [49], siendo la actualidad la única época de la historia en la que el oro no es directamente utilizado como dinero en ningún país del mundo, aunque sigue siendo utilizado como reserva de valor por parte de bancos centrales. Esto es así porque, a diferencia de las monedas fiduciarias (y como ocurre con el resto de metales preciosos), el oro tiene un valor más allá de la convicción de que vaya a ser aceptado en el futuro.

Debido a su carácter improductivo *per se*, el oro no compite con las acciones que cotizan en bolsa, sino que compite con las antes mencionadas monedas fiduciarias (las divisas nacionales) y con los bonos gubernamentales.

El oro es habitualmente referido como un activo refugio [50] . Esto es así por su valor tanto para la fabricación de joyas, como por su carácter no corrosivo (que le permite mantener su calidad durante periodos muy prolongados de tiempo), como por su fácil conversión en *dinero fiat*.

3.5.2 Otras materias primas

Otros importantes productos que forman parte de este mercado son los productos agrícolas, los metales, hidrocarburos o minerales. Algunos ejemplos pueden ser metales como la plata, el platino o el cobre; hidrocarburos como el petróleo o derivados de este como la gasolina. Otros productos que entran dentro de este mercado podrían ser el café, el azúcar o el algodón y dentro de los agrícolas algunos a mencionar podrían ser el trigo, el maíz o la cebada.

3.6 Divisas

El mercado de divisas [51] o mercado *Forex* (abreviatura del término inglés *Foreign Exchange*) es un mercado en el que se negocia con las monedas de los distintos países del mundo. Este mercado es el que mayor flujo tiene a nivel mundial, llegando a superar en varios billones estadounidenses ($1{,}000{,}000{,}000$ dólares) el flujo económico de todas las bolsas del mundo combinadas.

Una característica importante de este mercado es que, a diferencia de los mercados bursátiles, [52] este es un mercado libre no reglado. Esto quiere decir que no existe un órgano de compensación y liquidación que intermedie entre las partes y garantice el cumplimiento de las obligaciones convenidas por las mismas. Esto tiene como consecuencia que cada operación se cierre de manera particular entre las partes.

Otra característica importante de este mercado es que está en funcionamiento 24 horas al día durante $5{,}5$ días a la semana, comenzando a operar los domingos por la tarde con la apertura del mercado en Australia y finalizando los viernes con el cierre del mercado en Estados Unidos, concretamente en Nueva York.

En este mercado intervienen tanto inversores particulares como importantes organismos como los bancos centrales. El principal activo que se negocia es el dólar americano, puesto que esta divisa constituye el $60\,\%$ de las reservas de los bancos centrales. Le sigue el euro con en torno a un $24\,\%$.

El mercado de divisas cumple importantes funciones en cuanto a la situación económica mundial. Algunas de ellas son: fijar los precios de cambio entre las distintas divisas, favorecer el intercambio de divisas entre los diferentes países o regular el comercio internacional.

Un último punto a comenta sobre este mercado es la distinción entre mercado mayorista y minorista.

- **Mercado mayorista:** entre entidades financieras, empresas o instituciones.
- **Mercado minorista:** intercambio de billetes físicos en casas de cambio o bancos, realizado por particulares habitualmente para cubrir sus necesidades ante un viaje.

Algunos ejemplos de monedas que cotizan en este mercado son el dólar estadounidense (USD), el euro (EUR), la libra esterlina (GBP) o el yen japonés (JPY).

3.7 Optimización de carteras

En este trabajo se afrontará el problema de la optimización de carteras, encuadrado dentro del campo de la *Investigación Operativa* (disciplina Estadística encargada de resolver problemas de planificación de toma de decisiones en base a la evaluación y comparación de distintas estrategias). Para ello, se expondrán los principales modelos utilizados para realizar la selección de valores de una cartera de inversión y se comentarán sus principales características.

En primer lugar, cabe mencionar los componentes de un modelo de investigación operativa: alternativas, restricciones y función objetivo. Las alternativas del problema son lo que también se conoce como las **variables** desconocidas, que a su vez son los elementos con los que se construyen en forma de funciones matemáticas tanto las **restricciones** como la **función objetivo** del problema. La conjunción de estos tres elementos constituye un modelo matemático, cuya solución consiste en resolver un problema de **optimización**, es decir, en la **maximización** o **minimización** de la función objetivo satisfaciendo las restricciones del problema.

Algunos ejemplos de problemas conocidos que se resuelven utilizando este tipo de metodologías son los problemas de asignación y distribución de recursos, problemas de inventarios, problemas de colas, problemas de rutas o problemas de secuenciación de actividades.

El problema de la optimización de carteras puede verse desde la perspectiva de dos objetivos: la maximización de la rentabilidad (o ganancia esperada) y la minimización del riesgo. El planteamiento de un modelo con cualquiera de estas dos funciones objetivo constituye lo que denominamos **modelo básico** y el planteamiento de un modelo **biobjetivo** que pretende cumplir ambos propósitos de manera simultánea recibe el nombre de **modelo de Markowitz** en honor a *Harry Markowitz*, quien lo formuló a mediados del siglo XX. Ambos modelos pueden incluir restricciones como limitaciones en las cantidades de inversión mínima o máxima en un valor, o restricciones de cardinalidad que limiten el número de valores con los que componer la cartera.

Las variables de decisión del problema de optimización que se plantea son las proporciones de capital que se invierte en cada valor o activo j, denotadas como x_j y siendo $x = (x_1, ..., x_n)$ la cartera óptima construida.

Las hipótesis básicas del problema son:

$$\sum_{j=1}^{n} x_j = 1,$$

$$x_j \geq 0, j = 1, ..., n$$

Siguiendo [53], el rendimiento de una cartera se define para un periodo de T días como $R(x) = \sum_{j=1}^{n} x_j R_j$, donde R_j es el rendimiento del activo j. El rendimiento esperado es:

$$r(x) = E(R(x)) = \sum_{j=1}^{n} x_j r_j,$$

donde $r_j = E(R_j)$ Y el riesgo:

$$\sigma^2(x) = Var(R(x)) = E(R(x) - r(x))^2 = \frac{1}{T} \sum_{t=1}^{T} (\sum_{j=1}^{n} (R_j(t) - r_j))^2$$

También podría expresarse de forma cuadrática, que quizás sea más simple:

$$\sigma^2(x) = Var(R(x)) = E(R(x) - r(x))^2 = \sum_{i,j=1}^{n} \sigma_{ij} x_i x_j,$$

donde $\sigma_{ij} = cov(R_i R_j)$

Un tercer concepto importante dentro del marco de este problema es el de **ratio Sharpe** [54], desarrollado por el Premio Nóbel William Sharpe de la Universidad de *Stanford* y que mide la relación entre la rentabilidad y la volatilidad de una inversión, comparándola con una realizada con un riesgo muy bajo o nulo (habitualmente un fondo de inversión). El ratio Sharpe de una cartera x es:

$$SR(x) = \frac{r(x) - r_0}{\sigma(x)}$$

donde r_0 es el tipo interés del fondo de inversión sin riesgo tomado como referencia. En el caso de empresas pertenecientes a un índice bursátil se toma como referencia un fondo sin riesgo de dicho índice, pero debido al carácter de este trabajo, donde es posible construir carteras con activos pertenecientes a diversos mercados y al carácter interactivo de la misma, el usuario podrá elegir este valor, siendo 0 el valor por defecto.

Cabe destacar también que, a la hora de comparar diferentes posibilidades de inversión, aquella con un ratio Sharpe más alto proporciona una mayor rentabilidad para niveles iguales de riesgo, por lo que es habitual decantarse por inversiones con un valor elevado de este indicador.

3.7.1 Modelo básico

Existen dos modelos básicos de optimización de carteras:

- **Modelo de minimización del riesgo:** busca minimizar el riesgo fijando una rentabilidad mínima k. Se formula como:

  $$minf(x) = \sigma^2(x) = \frac{1}{T} \sum_{t=1}^{T} (\sum_{j=1}^{n} x_j (R_j(t) - r_j))^2$$

  o, de forma cuadrática:

  $$minf(x) = \sigma^2(x) = \sum_{i,j=1}^{n} \sigma_{ij} x_i x_j,$$

sujeto a:

$$\sum_{j=1}^{n} x_j = 1,$$
$$\sum_{j=1}^{n} r_j x_j \geq k,$$
$$x_j \geq 0, j = 1, ..., n.$$

- **Modelo de maximización de la rentabilidad:** busca maximizar la rentabilidad fijando un riesgo máximo k. Se formula como:

$$max f(x) = r(x) = \sum_{j=1}^{n} r_j x_j$$

o, de forma cuadrática:

$$max f(x) = r(x) = \sum_{j=1}^{n} r_j x_j,$$

sujeto a:

$$\sum_{j=1}^{n} x_j = 1,$$
$$\frac{1}{T} \sum_{t=1}^{T} (\sum_{j=1}^{n} x_j (R_j(t) - r_j))^2 \leq k,$$
$$x_j \geq 0, j = 1, ..., n.$$

3.7.2 Modelo de Markowitz

Harry **Markowitz** en [2] se planteó que un inversor debe considerar el retorno esperado de una inversión como algo deseable y la varianza de los mismos como algo indeseable, por lo que planteó que este debe anticipar dicha variabilidad implicando una concesión de riesgo. Por ello rechazó el paradigma habitual donde se consideraba que el inversor debe maximizar los rendimientos, proponiendo en contra realizar una diversificación de la inversión que permitiera combatir la variabilidad de los mercados. Desarrolló pues un modelo **biobjetivo** donde se busca maximizar la rentabilidad y minimizar el riesgo de manera simultánea, algo imposible de alcanzar. Es por eso que la solución de este modelo se plantea en términos de **eficiencia**, al no ser posible la obtención de una solución óptima como tal. Se considera pues que una cartera es eficiente si, para un riesgo dado, obtiene la máxima rentabilidad posible y viceversa, si, para una rentabilidad dada, obtiene el menor riesgo posible. La formulación del modelo con el enfoque de las ponderaciones es la siguiente:

$$min f(x) = \mu\sigma^2(x) - r(x) = \mu \frac{1}{T} \sum_{t=1}^{T} (\sum_{j=1}^{n} x_j (R_j(t) - r_j))^2 - \sum_{j=1}^{n} x_j r_j,$$

que, en forma cuadrática se expresaría como:

$$min f(x) = \mu\sigma^2(x) - r(x) = \mu \sum_{i,j=1}^{n} \sigma_{ij} x_i x_j - \sum_{j=1}^{n} x_j r_j,$$

sujeto a:

$$\sum_{j=1}^{n} x_j = 1,$$

$$x_j \geq 0, j = 1, ..., n,$$

donde μ es un parámetro, $0 \leq \mu \leq \infty$ que "regula" el énfasis puesto en cumplir cada uno de los dos objetivos (el primer término de la formulación hace referencia al riesgo y el segundo a la rentabilidad), de forma que un valor grande de μ centra el modelo en la minimización del riesgo y un valor pequeño en la maximización de la rentabilidad.

El modelo tiene pues, infinitas soluciones, una para cada posible valor de μ, son las carteras eficientes descritas anteriormente, que con la notación utilizada se pueden describir como:

$x* = (x_1^*, ..., x_n^*)$ tal que no existe otra cartera x' tal que $r'(x) \geq r(x^*)$ y a la vez $\sigma^2(x') \leq \sigma^2(x^*)$ con al menos una desigualdad estricta.

Todas esas carteras son razonables e interesantes para el inversor, aunque no es posible que una sea a la vez la que tiene menor riesgo y máxima rentabilidad. El conjunto de todas ellas recibe el nombre de **frontera eficiente** y se representa como una curva $riesgo - rendimiento$ en los ejes $0X - 0Y$ respectivamente con los puntos $r(x^*)$ y $\sigma^2(x^*)$. Un ejemplo se puede ver en la Figura 3.19.

Figura 3.19: Representación de la frontera eficiente. Tomado de [55].

Según la teoría de carteras existe interés por 3 carteras, denominadas **carteras notables**:

- **Máximo rendimiento:** $\mu = 0$
- **Mínimo riesgo:** $\mu = \infty$
- **Máximo ratio Sharpe.**

Una cuarta cartera que será considerada de interés en este trabajo pese a no pertenecer al conjunto de carteras eficientes es la cartera **equiponderada**, reparte el capital de manera equitativa entre todos los componentes de la cartera.

El artículo original de Markowitz [42] y el capítulo 23 del libro [53] ofrecen una visión más completa y matemática de este modelo.

3.7.3 Modificaciones al modelo de Markowitz

Una vez presentados los principales aspectos del modelo se comentan las posibles modificaciones que se le pueden añadir a este:

1. Como en cualquier problema de **programación lineal** se pueden incluir restricciones, que en este caso pueden ir referidas tanto a una inversión en particular $0 \leq x_i \leq k_i$, como a la limitación del número de total de valores en los que invertir (**restricción de cardinalidad**) $\#\{i \in \{1,2,...,n\} \, | x_i \neq 0\} \leq K$. También pueden incluir cotas inferiores y superiores de la inversión en cada valor, $x_i \leq U$ y $x_i \geq L$ respectivamente.

   En este trabajo se contemplan todas las restricciones comentadas a excepción de la que limita la inversión en un valor en particular.

2. En lugar de utilizar la media como estimación del rendimiento esperado, puede resultar más interesante **ponderar**, ya que es lógico pensar que las observaciones más lejanas tendrán una menor influencia en lo que ocurrirá en el futuro. Para ello [53] propone estimar el rendimiento utilizando **corrección de estimadores**, con la fórmula siguiente:

$$r_j = \frac{\sum_{t=1}^{T} p^{T-t} R_j(t)}{\sum_{t=1}^{T} p^{T-t}},$$

   donde p es un parámetro denominado **factor de descuento** que va entre 0 y 1 de manera que cuanto más se acerque a 1, más peso tendrán las observaciones de los periodos recientes.

   Esta modificación también será considerada en este trabajo.

3. También se puede plantear el riesgo como la desviación media absoluta respecto del rendimiento del modo siguiente:

$$\sigma_j^2 = E(|R_j - r_j|),$$

$$\sigma^2 = E|R - r(x)| = \frac{1}{T} \sum_{t=1}^{T} \left| \sum_{j=1}^{n} x_j (R_j(t) - r_j) \right|$$

   Esta modificación **no** se incluye en este trabajo.

3.7.4 Riesgo específico y riesgo sistemático

El riesgo de una cartera puede descomponerse en dos componentes:

- **Riesgo sistemático:** corresponde al riesgo del mercado. Es un riesgo que siempre está ahí y que, por tanto, no puede reducirse mediante la diversificación de la inversión. En el caso de los índices bursátiles la forma habitual de medirlo es con el riesgo del valor del propio índice.

- **Riesgo específico:** es propio de los valores que componen la cartera de inversión y, por tanto, puede reducirse mediante la diversificación del capital. Depende de las fluctuaciones de cada activo independientemente del contexto de mercado en el que se encuentran. Es el riesgo que interesa analizar.

En esta parte del trabajo, únicamente se permitirá trabajar con carteras pertenecientes a un mismo índice bursátil, ya que descomponer el riesgo de esta manera permite estudiar la relación entre dichos elementos.

Para poder llevar a cabo esta descomposición, es necesario calcular la **regresión lineal simple** de los retornos (rendimientos) de una cartera frente a los del índice deseado [56], ya que el coeficiente β de la regresión informa de cómo varía el rendimiento de la cartera al aumentar en 1 el valor del índice. El rendimiento de un valor expresado en estos términos queda como:

$$R_j = \alpha_j + \beta_j R_M + e_j,$$

donde R_j es el rendimiento del activo, R_M la rentabilidad del mercado en función del índice, β_j el parámetro que mide los cambios en R_j en función de R_M, α_j una constante (ordenada en el origen) y e_j el error residual. La pendiente de la recta de regresión obtenida viene determinada por el β obtenido, por lo que mide la **sensibilidad** del rendimiento del título o cartera al rendimiento del índice y puede considerarse un indicador del riesgo sistemático. Permite distinguir entre tres tipos de títulos o carteras:

1. **Títulos neutros** ($\beta = 1$): el título o cartera tiende a fluctuar en sintonía con el mercado.
2. **Títulos agresivos** ($\beta > 1$): el título o cartera tiende a fluctuar más que el mercado.
3. **Títulos defensivos** ($\beta < 1$): el título o cartera tiende a fluctuar menos que el mercado.

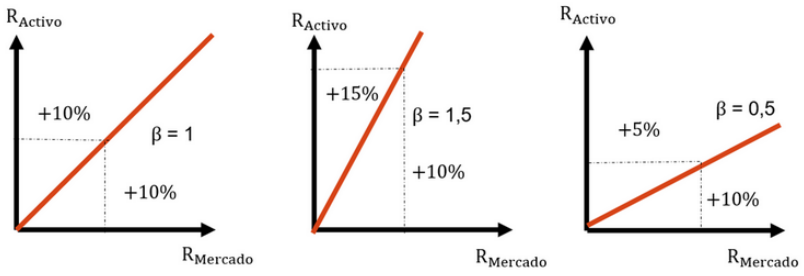


Figura 3.20: Variaciones de los títulos o carteras en función de β. Tomado de [56].

Finalmente podemos expresar el riesgo como:

$$\sigma_j^2 = \beta_j^2 \sigma_M^2 + \sigma_{e_j}^2,$$

donde σ_j^2 es el riesgo del título o cartera, σ_M^2 la varianza del mercado y $\sigma_{e_j}^2$ la varianza del propio título o cartera.

En la aplicación de este trabajo habrá una sección dedicada al estudio del riesgo.

3.7.5 Simulación de carteras con método de Montecarlo

El método de Montecarlo [57] es un método estadístico numérico utilizado para resolver problemas matemáticos complejos mediante la generación de variables aleatorias. Recibe ese nombre en referencia al casino de dicha ciudad, ya que la ruleta se trata de un generador simple de números aleatorios.

Los estudios llevados a cabo con este tipo de simulaciones se vieron impulsados enormemente con el desarrollo de los ordenadores y son hoy en día la base para la resolución aproximada de múltiples problemas, utilizados en campos como la física o la ingeniería.

En cuanto a la obtención de carteras, en este trabajo se utiliza el método de Montecarlo para obtener de manera aproximada la frontera eficiente comentada en 3.7.2. Para ello se llevan a cabo los siguientes pasos:

1. Se divide inicialmente el capital entre todos los activos considerados en la cartera, un número grande de veces.
2. Obtener el riesgo y el rendimiento de cada una de las carteras obtenidas mediante el reparto aleatorio del capital.
3. Representación $riesgo - rendimiento$ en los ejes $0X - 0Y$ de todas las carteras obtenidas mediante un gráfico de nube de puntos o *scatter plot*. El color lo indica el ratio Sharpe de cada una de las carteras obtenidas.

La frontera eficiente sería pues, la envolvente convexa de la nube de puntos representada.Para que la simulación de un resultado que verdaderamente aproxime a la realidad ha de utilizarse un número grande de carteras, con el fin de representar todos los escenarios posibles, ya que la *random-walk hypothesis* [18] indica que los rendimientos de una inversión son impredecibles y aleatorios.

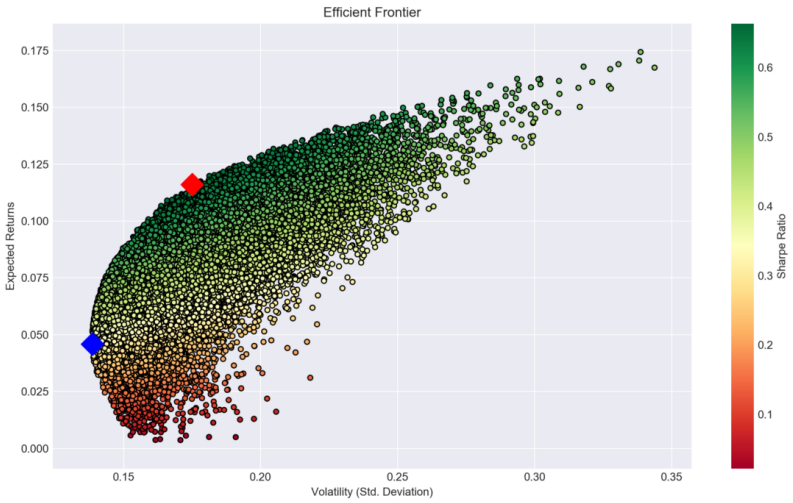


Figura 3.21: Ejemplo de obtención de la frontera eficiente con simulación por método de Montecarlo. Tomado de [58].

En la aplicación de este trabajo habrá una sección dedicada a la obtención de carteras con simulación por método de Montecarlo.

3.8 Predicción de valores

La predicción del precio de los valores es un problema que lleva mucho tiempo tratándose de diversas maneras. Los enfoques principales siempre han sido mediante la realización de análisis fundamental y de análisis técnico, junto al estudio de los indicadores que estos proporcionan.

El desarrollo de las matemáticas, la algoritmia y el incremento de la capacidad computacional ha ido dirigiendo este problema hacia un enfoque cada vez más basado en el uso de tecnologías complejas y novedosas como las relacionadas con la inteligencia artificial.

El estudio realizado durante muchos años por multitud de autores ha llevado a concluir que el mercado es es no lineal, incierto, estocástico y no estacionario. Esto ha llevado a tomar por válidas dos importantes hipótesis [18] que pretenden ser puestas en jaque por los nuevos modelos de predicción de valores:

- ***The random walk hypothesis (RWH)*:** el precio de los valores es fundamentalmente estocástico. Esto quiere decir que cualquier intento de predecir los precios resulta inútil, debido a la gran aleatoriedad intrínseca del problema y al movimiento impredecible de los precios. La consecuencia de esta hipótesis es, por tanto, que no se pueden utilizar valores pasados para predecir valores futuros.
- ***The efficient market hyphotesis (EMH)*:** se basa en la premisa de que el mercado es *informativamente eficiente*, lo que quiere decir que el mercado es eficiente a la hora de establecer el precio correcto de un valor y este recoge toda la información posible sobre el mismo. La consecuencia de esta hipótesis es que los valores siempre tienen un precio justo y, por tanto, no es posible *batir al mercado* comprando valores infravalorados o vendiendo valores sobrevalorados.

En las siguientes secciones de este trabajo se mostrarán y explicarán algunos de los intentos realizados en los últimos tiempos para afrontar el problema de la predicción de datos de valores de los distintos mercados. A su vez, se explicarán los modelos implementados en la aplicación web construida como objeto de este trabajo.

3.8.1 Estado del arte

Las contribuciones realizadas hasta la fecha en cuanto a la predicción de valores, se basa en cuatro categorías principales [19]:

- **Enfoque estadístico.**
- **Reconocimiento de patrones.**
- **Aprendizaje automático .**
- **Análisis de sentimientos.**

Un breve resumen de la taxonomía de esta clasificación puede verse en la Figura 3.22.

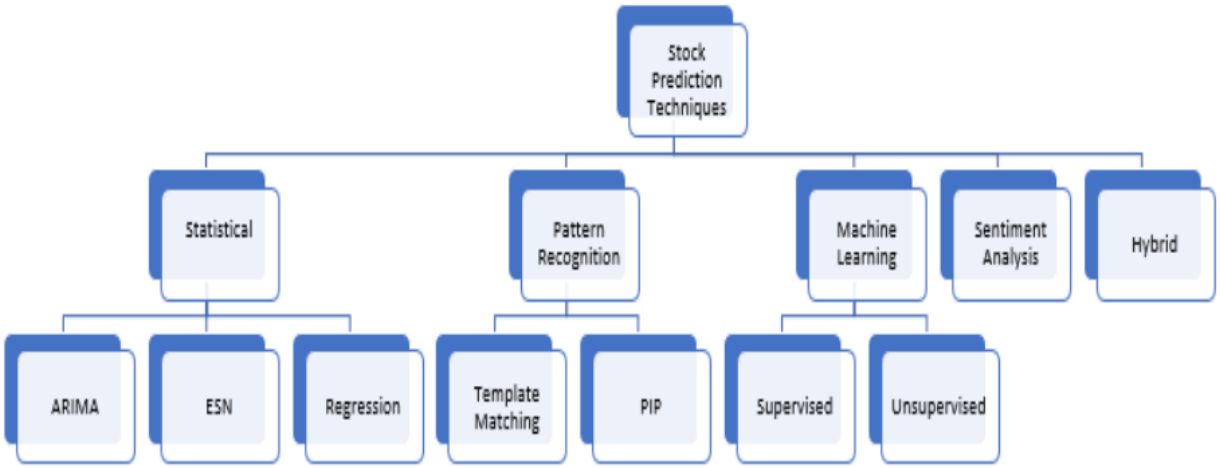


Figura 3.22: Taxonomía de las principales técnicas de predicción de valores. Tomado de [19].

Las primeras aproximaciones al problema se llevaron a cabo desde el punto de vista de la **Estadística**. Este enfoque parte de la asunción de que el mercado es lineal, estacionario y normal (cuestiones comúnmente asumidas en la Estadística). Las variables de entrada en este caso son series temporales y los modelos de análisis más utilizados son los univariantes: modelos ARMA (*Auto-Regressive Moving Average*) y ARIMA (*Auto-Regressive Integrated Moving Average*). Otros modelos como el GARCH (*Generalized Auto-Regressive Conditional Heteroskedastic*) y el STAR (*Smooth Transition Average*) también son empleados.

Los modelos principales tienen una parte autorregresiva (AR) que trata de explicar la media de las series temporales de los mercados y una parte de media móvil (MA) que trata de capturar los cambios en el mercado. Sin embargo, algo que estos modelos no tienen en cuenta es el fenómeno conocido como agrupamiento de la volatilidad (*volatility clustering*) que se reiere a que los grandes cambios en el mercado (sean del signo que sean) tienden a ir seguidos de grandes cambios y los pequeños cambios tienden a ir seguidos de pequeños cambios. Los modelos ARIMA surgen en este contexto como una mejora natural de los modelos ARMA, permitiendo convertir series no estacionarias en estacionarias (una serie es estacionaria cuando su media y varianza son constantes en el tiempo).

Otro modelos utilizado es el ESM (*Exponential Smoothing Model*) que suaviza la serie temporal utilizando una función de ventana exponencial y que ha dado buenos resultados en múltiples estudios.

Dentro de los modelos multivariantes caben destacar los LDA (*Linear Discriminant Analysis*), los QDA (*Quadratic Discriminant Analysis*) y los modelos de regresión.

Otra aproximación que se ha tratado de llevar a cabo para la predicción de valores es el uso de algoritmos de **reconocimiento de patrones**. Este término resulta equivalente al de aprendizaje automático (*machine learning*) aunque en la predicción de valores de mercados se utilizan de manera diferente. El reconocimiento de patrones se utiliza para identificar patrones y tendencias en los datos y acostumbran a estar formados por secuencias recurrentes que pueden observarse en gráficos como los de velas y ser tomadas como señales de compra o venta.

Muy ligado al análisis gráfico, algunos importantes métodos utilizados son el PIP (*Perceptually Important Points*) relacionado con la reducción de la dimensionalidad de la serie temporal, el DTW (*Dynamic Time Warping*)el *template matching* y el *flag pattern*, estos dos últimos en ocasiones combinados con otro tipo de algoritmos como los algoritmos genéticos.

En lo relativo a los métodos de **aprendizaje automático** hay que hablar de dos categorías: el aprendizaje *supervisado* y el aprendizaje *no supervisado*. El primero de ellos utiliza entradas etiquetadas para, tras la realización del proceso de entrenamiento, predecir un valor futuro, mientras que el segundo utiliza entradas sin etiquetar con el objetivo de encontrar relaciones o agrupaciones en los datos.

Dentro de los métodos **supervisados**, se han utilizado todo tipo de algoritmos para tratar la predicción de valores. Las SVM (*Support Vector Machines*) y los árboles de decisión fueron algunos de los primeros en ser empleados con ese fin, aunque otros como el algoritmo de los K vecinos más próximos (*K-nearest neighbors*), el de *Naive Bayes* o la regresión logística también se han explorado (aunque en menor medida) en algunos estudios. Otros métodos como los *ensembles* (métodos que entrenan varios algoritmos de aprendizaje y posteriormente combinan sus resultados) también han gozado de popularidad. Ejemplos de estos son los árboles aleatorios (*random forest*), el algoritmo *AdaBoost* o el *Kernek Factory*.

Sin embargo, los métodos que más han destacado en los últimos tiempos son los que emplean el algoritmo XGBoost (*Extreme Gradient Boosting*) y los basados en redes neuronales. Dentro de estos últimos se ha utilizado una gran variedad de técnicas como son las redes neuronales artificiales (ANN) o las basadas en *deep learning* . Las redes neuronales recurrentes son el enfoque principal dentro de este grupo de técnicas debido a sus características que le permiten modelar la variable tiempo. Los principales modelos de redes recurrentes son las redes simples (RNN), las LSTM (*Long Short Term Memory*) y las GRU (*Gated Recurrent Units*) son a día de hoy muy ampliamente estudiados, con multitud de variantes que pretenden mejorarlos. Otras técnicas de *deep learning* muy utilizadas son las CNN (*Convolutional Neural Networks*) cuyas principales aplicaciones están relacionadas con el reconocimiento de imágenes y de textos, las DBN (*Deep Belief Networks*) o las RBM (*Restricted Boltzmann Machines*); aunque prácticamente cualquier tipo de construcción de redes neuronales existente se ha adaptado para su utilización en la predicción de valores, debido al enorme interés del tema.

En cuanto a los métodos **no supervisados** cabe destacar la utilización del algoritmo de las K-medias, en ocasiones combinado con algoritmos de reducción de dimensionalidad como el análisis de componentes principales (PCA). Otros algoritmos de *clustering* (construcción de agrupaciones) como el HAC (*Hierachical Agglomerative Clustering*) o el HRK (*Hierachical Reverse K-means*) también han sido explorados. También hay que mencionar el uso de métodos basados en reglas como el *AprioriAll*.

Posteriormente, con el objetivo de incluir componentes del análisis fundamental en los métodos de predicción de valores basados en el uso de computadoras, surgió lo que se conoce como ***sentiment analysis***. Estos métodos tratan de recoger la variabilidad asociada al comportamiento de los inversores como reacción ante las diferentes noticias que surjan relacionadas con el mercado, tanto a través de los medios tradicionales como de las redes sociales. Los algoritmos utilizados en este enfoque sirven de apoyo a los modelos de aprendizaje automático antes comentados, introduciendo en estos nuevas variables que tratan de identificar el sentimiento y comportamiento de los inversores. Algunos de los algoritmos de procesamiento de texto utilizados para estos fines son el algoritmo BoW (*Bag of Words*) o el *Word2vec*, habitualmente empleado para analizar la polaridad de sentimientos ante el mercado (dividiendo entre favorables y no favorables al mismo).

Finalmente, enfoques modernos están empleando lo que se conoce como **enfoque híbrido**, que consiste en la combinación de distintos modelos de los comentados con el fin de mejorar los resultados obtenidos de cara a la predicción.

3.8.2 Modelos

En las siguientes secciones se presentarán los modelos empleados en el presente trabajo para tratar el problema de la predicción de valores, pero antes conviene definir qué es lo que se entiende por modelo en el contexto en el que nos encontramos. Un **modelo** es la definición de la relación entre las variables de un problema y la respuesta. Se puede entender como un conjunto de reglas que, para un determinado conjunto de características, determina la correspondiente salida. El término de modelo está referido a una representación abstracta, que puede ser de carácter conceptual, gráfico o visual, pero siempre tiene la finalidad de obtener un resultado a partir de unos datos de entrada, sea cual sea el contexto en el que se esté trabajando.

Dentro de los modelos de inteligencia artificial, se distinguen dos tipos [59]:

- **Modelos de clasificación:** la salida del modelo es una clase (categoría), obtenida de entre un número limitado de las mismas. Un ejemplo de problema de clasificación puede ser determinar si el animal que aparece en una imagen es un perro o un gato. Una variante de estos modelos son los que utilizan probabilidades en la salida. El enfoque habitual en estos casos es determinar como salida la clase que se presente con una probabilidad más alta.
- **Modelos de regresión:** la salida del modelo es un valor numérico obtenido de un conjunto infinito de posibles resultados. Un ejemplo de problema de regresión puede ser determinar el precio de una vivienda a partir de unos datos.

Aunque hay algoritmos específicos para un tipo de problema u otro, la gran mayoría de técnicas como las SVM, la regresión logística, los árboles de decisión o las redes neuronales (en todas sus variante, incluidas las de *deep learning*) funcionan en ambos tipos de problema.

Los modelos construidos en este trabajo serán todos modelos de regresión.

3.8.3 Metodología experimental

La metodología experimental consiste en un conjunto de procedimientos destinados a evaluar los modelos utilizados en la resolución de un problema. Habitualmente este conjunto de técnicas tienen la finalidad de medir la **tasa de error**, lo cual se lleva a cabo mediante tests de hipótesis (en este contexto entendemos que la hipótesis es el modelo generado) realizados sobre los datos.

La metodología habitual utilizada para evaluar correctamente un modelo generado en un problema consiste en la división en 2 o 3 del conjunto de datos:

- **Conjunto de entrenamiento (T):** subconjunto de los datos con el que se crea la hipótesis.
- **Conjunto de validación (V):** subconjunto de los datos con el que se ajustan diversos parámetros del modelo.
- **Conjunto de prueba (P):** subconjunto de los datos con el que se calcula la tasa de error.

En ocasiones el conjunto de validación se omite y únicamente se utilizan los subconjuntos de entrenamiento y prueba, enfoque que se utilizará en este trabajo por simplicidad.

Los 3 (o 2) subconjuntos han de ser seleccionados de manera aleatoria, de forma que los elementos de P no se utilicen en T ni V. A su vez, es conveniente que estos conjuntos sean lo más grandes posibles. Una vez evaluado el modelo, el **conjunto de datos al completo** puede utilizarse para crear el modelo. Separar parte

de los datos para evaluar el modelo se dice que da lugar a una estimación optimista, es decir, la tasa de error obtenida tiene un valor superior al error verdadero que estima. Esto es así debido a que la tasa de error se obtiene con el subconjunto T y a que la tasa de error disminuye al aumentar el tamaño del conjunto de datos sobre el que se calcula.

Existen dos formas principales de realizar la división del conjunto de datos:

- **Hold out:** consiste en dividir el conjunto de datos en T y P de manera aleatoria, de forma que el conjunto de entrenamiento sea de tamaño mayor, habitualmente con una proporción de, al menos, el $60\,\%$ de los datos. Tiene el inconveniente, sobre todo en problemas de clasificación, de que las muestras de alguno de los dos subconjuntos pueden no ser representativas (por ejemplo, alguna clase podría no estar presente). Para solucionar esto es habitual utilizar estratificación en ese tipo de problemas, asegurando así una distribución aproximadamente igual de las clases en los subconjuntos. Es posible repetir el proceso k veces realizando lo que se conoce como *hold out repetido*, obteniendo la tasa de error como el promedio de las tasas obtenidas en cada experimento base (cada una de las realizaciones del hold out). Debido a que el intento de mejorar mediante la repetición del experimento tiene la consecuencia del solapamiento entre los conjuntos utilizados y, por tanto, la ruptura de la independencia entre los mismos, se propone un nuevo método: la validación cruzada.

- **Validación cruzada:** el modo de proceder en este método consiste en la división inicial del conjunto de datos en k subconjuntos de igual tamaño (de manera aproximada). Posteriormente, se utilizan todos los subconjuntos creados menos uno como conjunto de entrenamiento, quedando el restante como conjunto de prueba. Este proceso se repite k veces obteniéndose k tasas de error, que permiten estimar el error verdadero promediándolas. Al igual que el *hold out* tiene variante estratificada y puede realizarse de manera repetida, que tiene el efecto de reducir la variabilidad a costa de aumentar el coste.

En este trabajo se utilizará *hold out* adaptado a series temporales: se separará el porcentaje elegido de datos de la serie temporal, pero no de manera aleatoria, sino que se mantendrá el orden de las observaciones. El porcentaje de división de los datos será elegido por el usuario de manera interactiva. Esta elección se debe a la complejidad y alto coste que conlleva utilizar validación cruzada en algunos de los modelos que se incluirán, como es el caso de aquellos basados en redes neuronales.

Evaluación de modelos

Se utilizarán distintas métricas para evaluar los modelos construidos en este trabajo y poder ver la calidad de los mismos y comparar unos con otros sobre un mismo conjunto de datos.

Siendo A_t el valor observado, F_t el valor predicho, M la media y n n el número de muestras, definimos las métricas empleadas:

- **RMSE (*Root of Mean Squared Error*):** raíz del error cuadrático medio, indica cómo de ajustados están los datos al modelo.

$$\sqrt{\frac{1}{n}\sum_{t=1}^{n}(A_t - F_t)^2}$$

- **MSE (*Mean Squared Error*):** error cuadrático medio, momento de segundo orden del error. Incorpora tanto la varianza como el sesgo y da una idea del error general que comete un modelo a la hora de hacer predicciones. Sus unidades están elevadas al cuadrado.

$$\frac{1}{n}\sum_{t=1}^{n}(A_t - F_t)^2$$

- **ME (*Mean Error*):** error medio, indica que el modelo sobreestima si su valor es negativo y que subestima si es positivo.

$$\frac{1}{n}\sum_{t=1}^{n}(A_t - F_t)$$

- **MAE (*Mean Absolute Error*):** da una medida de la diferencia media entre los valores reales y los predichos por el modelo.

$$\frac{1}{n}\sum_{t=1}^{n}|A_t - F_t|$$

- **MEDAE (*Median Absolute Error*):**

$$median(|A_t - F_t|)$$

- **R2 (*R squared*):** desviación media de los puntos del conjunto de datos con respecto de la media. Da una idea de cómo de bien se ajusta el modelo a los datos: cuanto más cercano sea su valor a 1, mejor es el ajuste.

$$1 - \frac{\sum_{i=0}^{n-1}(A_t - F_t)^2}{\sum_{i=0}^{n-1}(A_t - M)^2}$$

  Aunque parezca antinatural, el R^2 puede ser negativo, lo que se entiende como si fuera 0, indicando una falta total de ajuste en el modelo.

- **MAPE (*Mean Absolute Percentage Error*):** da una medida del *accuracy* como porcentaje.

$$\frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \times 100$$

- **Varianza explicada:** ratio entre la varianza del error y la varianza real de los datos. Mide cómo de bien ajusta el modelo las variaciones en los datos.

$$1 - \frac{Var(A_t - F_t)}{Var(A_t)}$$

La evaluación de los distintos modelos implementados se realizará con *Python*, utilizando diversas funciones contenidas en la librería *sklearn* 7.2.5 dentro del paquete *metrics*.

3.8.4 Variables

Las variables que se tendrán en consideración en las distintas secciones de la web serán aquellas proporcionadas por la *API de Yahoo Finance* 4. Aquellas derivadas de la obtención de los distintos indicadores técnicos explicados en 3.2.2 serán utilizadas en la sección donde se podrán analizar los activos de manera descriptiva.

Las variables extraídas de *Yahoo Finance* son:

- Precio de apertura.
- Precio de cierre.
- Precio máximo.
- Precio mínimo.
- Volumen.

- Precio de cierre ajustado.

El precio que se va a utilizar a lo largo de este trabajo de manera central es el **precio de cierre ajustado** (*Adjusted closing price*) [60], que se diferencia del precio de cierre convencional en que el valor se modifica levemente teniendo en cuenta cualquier cosa que pueda afectar el precio de los valores después del cierre del mercado. Esta será la variable a predecir en todos los modelos que se propondrán y la única que se utilizará en su construcción a excepción del caso del modelo 3.8.6 y los modelos de redes recurrentes 3.8.11, 3.8.12 y 3.8.13, que permitirán la inclusión del resto de variables.

3.8.5 Preparación de datos de series temporales para regresión

De cara a poder utilizar algunos de los algoritmos de *machine learning* propuestos en este trabajo, que originalmente no estaban pensados para trabajar con **series temporales**, se necesitará realizar un pequeño procesamiento previo en los datos consistente en transformar la serie temporal en un problema de aprendizaje supervisado ([61]). Este procesamiento debe hacerse ya que las series temporales no tienen como tal una salida, es decir, no tenemos una variable o conjunto de variables de entrada y una variable respuesta. El proceso tiene como resultado una transformación en los datos que se muestra en el siguiente ejemplo, donde de una serie temporal definida como $X = [10, 20, 30, 40, 50]$ se pasa a:

X	**y**
?	10
10	20
20	30
30	40
40	50
50	?

Cuadro 3.1: Ejemplo de serie temporal convertida a problema de aprendizaje supervisado.

Lo que se ha hecho es utilizar el paso (*step*) previo para predecir el valor en el paso siguiente. Este método, conocido como ***ventana deslizante***, recibe su nombre del hecho de que la ventana de entradas y salidas esperadas se desplaza hacia adelante en el tiempo para crear nuevas "muestras". Tiene como resultado un nuevo conjunto de datos donde cada variable tiene dos columnas y donde las filas primera y última no pueden utilizarse para entrenar el modelo.

3.8.6 Modelo ARIMA

Dentro del enfoque estadístico de predicción de valores los principales modelos a considerar son los modelos **ARIMA**, procedentes de la parte estadística conocida como *Análisis de series temporales* y encuadrados dentro de lo que se conoce como la metodología de ***Box-Jenkins*** [62].

Los modelos ARIMA permiten describir un valor como una función lineal de los datos anteriores y de errores debidos al azar. Además, puede incluir un componente cíclico o estacional. Para pode aplicar su metodología, Box y Jenkins recomiendan como mínimo 50 observaciones en la serie temporal.

ARIMA es un acrónimo que proviene de ***Autorregresive*** y de ***Integrated Moving Average*** y que describe los aspectos clave que componen el modelo. Una breve descripción de estos términos podría ser:

- **AR:** *Autorregresión*, modelo que utiliza relaciones de dependencia entre observaciones con sus observaciones pasadas (conocidas como retardos).

- **I:** uso de diferenciación (construcción de una nueva serie temporal suponiendo que la tendencia en un instante es muy próxima a la tendencia en el instante anterior) para transformar la serie y hacerla estacionaria, es decir, para eliminar su componente de tendencia (componente que modela el comportamiento de la serie a largo plazo). La operación de diferenciación se describe del siguiente modo:

$$y_t = x_t - x_{t-1}$$

  Y puede realizarse en repetidas ocasiones, dando lugar a diferenciaciones de orden 1, 2, etc.

- **MA:** *Media móvil*, modelo que utiliza relaciones de dependencia entre las observaciones y el error residual modelados como medias móviles sobre los retardos.

Cada uno de los componentes del modelo se describe mediante un parámetro, y se distinguen a su vez dos partes: regular y estacional. Los primeros 3 términos identifican la parte regular (modela la dependencia entre observaciones consecutivas) y los restantes describen la parte estacional (modela la variación periódica y predecible de la serie con un periodo inferior o igual al de un año). Un modelo ARIMA pues, queda descrito de la siguiente forma:

$$ARIMA(p, d, q) \times (P, D, Q)_s$$

- **p:** número de retardos del modelo en su parte regular.
- **d:** grado de diferenciación de la parte regular (número de veces que se realiza la operación de diferenciación sobre este componente del modelo).
- **q:** tamaño de la ventana temporal de la media móvil de la parte regular, conocido como orden de media móvil.
- **P:** número de retardos del modelo en su parte estacional.
- **D:** grado de diferenciación de la parte estacional.
- **Q:** orden de media móvil de la parte estacional.
- **s:** longitud del periodo de la estacionalidad.

Con estos elementos se construye un modelo de *regresión lineal*, donde los parámetros descritos pueden tomar el valor 0 indicando la no utilización de dicho parámetro, lo que da lugar a otros tipos de modelos como los AR(p), MA(q) o ARMA(p, q).

La escritura matemática del modelo es:

$$Y_t = -(\bigtriangledown^d Y_t - Y_{t-1}) + \phi_0 + \sum_{i=1}^{p} \phi_i \bigtriangledown^d Y_{t-i} - \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t$$

Donde d corresponde al orden de diferenciación, ϕ denota los parámetros de la parte autorregresiva del modelo, θ los parámetros de la parte de media móvil y ϵ es el término del error. A su vez, se tiene que tener en cuenta que $\bigtriangledown^d Y_t = Y_t - Y_{t-1}$.

Hay diversos procedimientos para determinar los valores adecuados de los parámetros a la hora de modelar una serie temporal, siendo los principales los que implican analizar gráficos como los *periodogramas* y los *correlogramas*, que representan la aportación a la varianza total de las componentes estacionales de la serie y las correlaciones de una observación con las observaciones anteriores respectivamente.

En este trabajo, el ajuste de modelos ARIMA para la predicción de series temporales de valores se realizará con *Python* utilizando la función *ARIMA* del paquete *statsmodels* 7.2.5. Los parámetros que se podrán ajustar de manera interactiva en la web son los 7 parámetros que describen el modelo ARIMA (p, d, q, P, D, Q, s), el

porcentaje de datos utilizados para entrenamiento y para evaluación del modelo construido y el tamaño de la ventana temporal utilizada en el modelo.

En este modelo se permitirá incluir la totalidad de las variables descritas en 3.8.4 y utilizar todas las métricas descritas en 3.8.3.

3.8.7 Modelo Support Vector Regressor

El primer modelo de *aprendizaje automático* que presentamos en este trabajo es el modelo ***Support Vector Regressor (SVR)***. Este modelo, constituye una variante del conocido modelo *Support Vector Machines (SVM)*, habitualmente utilizado en problemas de clasificación, pero debidamente adaptado para su utilización en problemas de regresión como el que aquí se aborda.

La idea básica del algoritmo SVM es representar los puntos de la muestra de datos del problema en el espacio, llevando a cabo posteriormente una separación entre las clases mediante un ***hiperplano*** *separante* que divide la región espacial en secciones lo más grandes posible. Este hiperplano se define como un *subespacio afín plano* de dimensión $p-1$, lo que se traduce en que en un subespacio plano será una línea y en un subespacio de dimensión $p = 3$ será un plano bidimensional. Las observaciones de cada clase que se encuentran a menor distancia del hiperplano dan forma a lo que se conoce como **vectores soporte**, que son los vectores paralelos al hiperplano separante que pasan por dichos puntos.

El algoritmo SVM puede emplearse en problemas de alta dimensionalidad, construyéndose no un único hiperplano, sino un conjunto de hiperplanos. A la variable predictora se le llama atributo y los atributos utilizados para definir el hiperplano se denominan características. La elección de la representación más adecuada del universo estudiado se lleva a cabo mediante un proceso llamado selección de características, que da como resultado una representación conocida como *espacio de características*.

Este tipo de algoritmos están relacionados con los de *redes neuronales*, ya que, al igual que estas, hacen uso de una función núcleo conocida como ***kernel***, que sirve para realizar la separación en clases cuando estas no son linealmente separables. Estas funciones núcleo lo que hacen es "inventar" una nueva dimensión que permite llevar a cabo la separación (ver Figura 3.23).

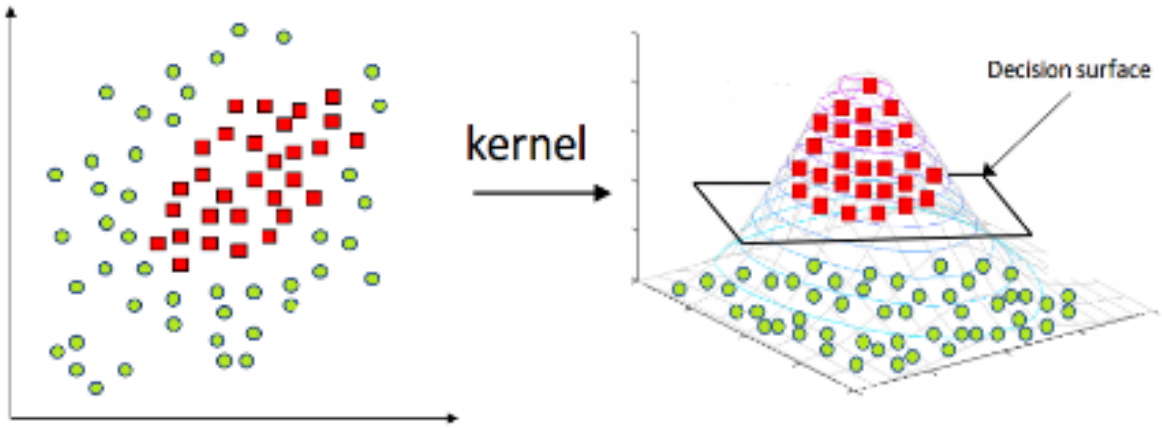


Figura 3.23: Ejemplo de separación en clases con *kernel*. Tomado de [63].

El modo de proceder en los modelos SVR es análogo al de las SVM [64] puesto que se tratará de encontrar un hiperplano separante del mismo modo que se hacía en estos modelos. Para ello se establecen dos "límites" entre los cuales procede a realizarse la búsqueda del plano separante entre todos los puntos situados a una distancia $+\epsilon$ o $-\epsilon$ (los límites mencionados) del plano buscado. El hiperplano elegido será aquel que pase por un mayor número de puntos, de forma que se minimice la *norma l2* del vector de coeficientes del modelo,

definida a su vez como:

$$|\mathbf{X}| = \sqrt{\sum_{k=1}^{n} |X_k|^2}$$

Este margen se modela con un nuevo término denominado ***tolerancia***, que hace referencia a "cuánto error estamos dispuestos a asumir en nuestro modelo de regresión". A su vez se tiene la restricción de que $|y_i - w_i x_i| \leq \epsilon$. Lo podemos ver en 3.24.

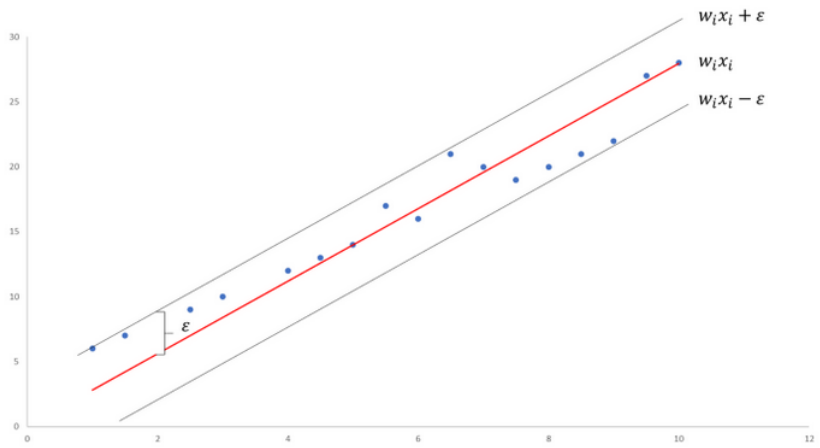


Figura 3.24: Hiperplano separante SVR. Tomado de [64]

Otro parámetro fundamental en los modelos SVM y SVR es el **parámetro de regularización C**, que establece un compromiso entre el error de entrenamiento y la complejidad del modelo.

- Un C bajo lleva a un modelo sencillo, con un mayor error de entrenamiento, con suavidad en la frontera de decisión.
- Un C alto por contra lleva a un modelo más complejo, con poca suavidad en la frontera de decisión y mayor tendencia al sobre-ajuste.

Esto se puede ver claramente en la Figura 3.25.

Dentro de los *kernels* antes mencionados podemos encontrar multitud de variantes diferentes. Sin entrar más en detalle podemos decir que los principales son los siguientes:

- **Lineal:**

$$k(x_i, x_j) = < x_j, x_i^t >$$

- **Polinómico:**

$$k(x_i, x_j) = (\gamma < x_j, x_i^t > + r)^d$$

  Donde d es el grado del polinomio.

- **Gaussiano:**

$$k(x_i, x_j) = exp(-\frac{||x_j - x_i^t||^2}{2\sigma^2})$$

- **Redes de base radial (RBF):**

$$k(x_i, x_j) = exp(-\gamma ||x_j - x_i^t||^2)$$

  Donde γ hace referencia a cuánta influencia tiene cada ejemplo de entrenamiento. Cuanto mayor sea γ, más cercanos han de estar los ejemplos para verse afectados.

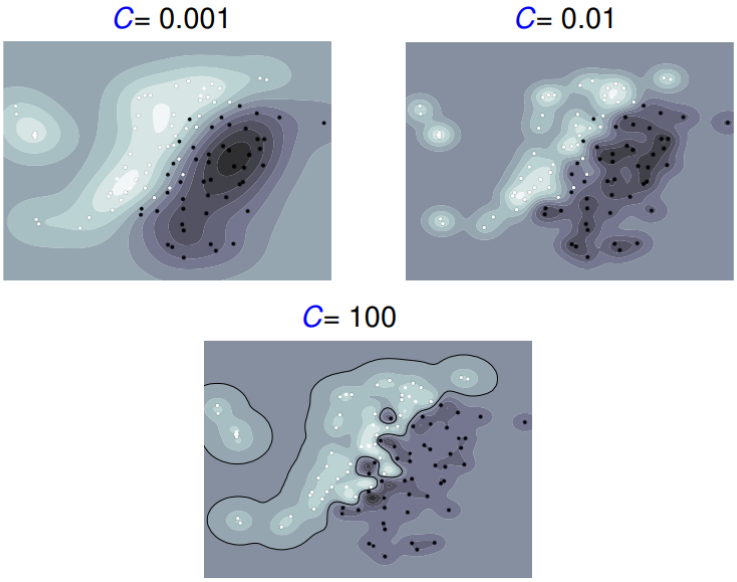


Figura 3.25: Influencia del parámetro C en los modelos SVM. Tomado de [63]

- **Sigmoide:**

$$k(x_i, x_j) = tanh(\gamma < x_j, x_i^t > +r)$$

El método para transformar una serie temporal de forma que podamos utilizar el modelo SVR es el expuesto en 3.8.5.

En este trabajo, el ajuste de *SVR* para la predicción de series temporales de valores se realizará con *Python* utilizando la función *SVR* del paquete *sklearn* 7.2.5. Los parámetros que se podrán ajustar de manera interactiva en la web son el tipo de kernel a utilizar, el grado del kernel polinómico en caso de haber elegido este núcleo, la forma de elegir el coeficiente γ en los *kernels* no lineales, el valor del parámetro de regularización C y el uso o no de una heurística de regularización que en ocasiones tiene el efecto de reducir el tiempo de entrenamiento cuando el número de iteraciones es elevado. En casos en los que usemos una tolerancia grande, no utilizar esta heurística será más rápido.

En este modelo se permitirá incluir únicamente la variable del precio de cierre ajustado y utilizar todas las métricas descritas en 3.8.3.

3.8.8 Modelo Random Forest

Uno de los modelos más importantes dentro del *aprendizaje automático* por su versatilidad y eficiencia es el modelo de ***Random Forest***. Este tipo de modelo pertenece a la categoría de modelos de *ensembles* (ver 3.8.1) y constituye un tipo de modelo de ***bagging*** de árboles aleatorios. Los algoritmos de *bagging* (*bootstrap aggregating*) utilizan algoritmos simples en **paralelo**, con el fin de aprovechar la independencia existente entre estos para reducir el error (la varianza) y proporcionar una salida obtenida como el fruto de la "votación" de los distintos algoritmos empleados (en el caso de los problemas de clasificación) o como el promedio (en los problemas de regresión como el de este trabajo).

El algoritmo de *Random Forest* [65] introduce diversidad (aleatoriedad) utilizando muestreo con reemplazamiento en el conjunto de entrenamiento y muestreo sin reemplazamiento en el conjunto de atributos utilizado para realizar la separación en los nodos de los árboles de decisión y es uno de los mejores métodos de *ensembles* que se puede construir al ser muy precisos y eficientes gracias a la paralelización.

El algoritmo procede creando un gran número de **árboles de decisión** (de ahí el nombre de bosque) y promediando las salidas de todos ellos para dar la salida final del algoritmo. Típicamente, la construcción de

un árbol de decisión incluye evaluar con alguna métrica cada variable en un nodo para elegir en base a qué realizar la división en ese nodo. En los bosques aleatorios, reduciendo el conjunto de variables considerado en cada punto de separación, se fuerza a que cada árbol de decisión que compone el *ensemble* sea muy diferente. Esto tiene como consecuencia que las predicciones son diferentes y los errores proporcionados por cada árbol están menos correlados, dando como resultado habitualmente un gran rendimiento.

De cara a utilizar el algoritmo *Random Forest* con series temporales, se realiza previamente el procesamiento descrito en 3.8.5 para las distintas variables incluidas en el modelo.

En este trabajo, el ajuste de *Random Forest* para la predicción de series temporales de valores se realizará con *Python* utilizando la función *RandomForestRegressor* del paquete *sklearn* 7.2.5. Los parámetros que se podrán ajustar de manera interactiva en la web son el número de árboles de decisión que componen el bosque, el número mínimo de muestras que contendrá un nodo hoja, el número mínimo de muestras necesario para realizar una separación, la función utilizada para evaluar una división en un nodo y el número de variables a considerar a la hora de hacer una separación.

En este modelo se permitirá incluir la totalidad de las variables descritas en 3.8.4 y utilizar todas las métricas descritas en 3.8.3.

3.8.9 Modelo XGBoost

Otro modelo de *ensembles* muy importante es el modelo **XGBoost** (*eXtreme Gradient Boost*), de gran popularidad hoy en día debido a su facilidad de implementación y buenos resultados, siendo famoso por resultar vencedor en un gran número de competiciones algorítmicas en plataformas como *Kaggle*. Es un tipo de ***boosting*** [66], que, a diferencia del *bagging* de los árboles aleatorios, basan la construcción del *ensemble* en la **secuenciación** de algoritmos. Al igual que el modelo *Random Forest* combina árboles de decisión, aunque en este caso no son árboles aleatorios.

XGBoost procede generando múltiples modelos “débiles” e introduciendo los resultados de un modelo como entrada en el siguiente. Ajustando los parámetros y buscando mejorar una función objetivo se consigue generar un modelo final más fuerte y con una mayor estabilidad en sus resultados. La forma de conseguir esto es utilizando un algoritmo de optimización denominado **descenso del gradiente** [67], que, sin profundizar matemáticamente, podemos decir que estima los mínimos locales de una función únicamente utilizando salidas numéricas (no funciones).

Durante el proceso, cada modelo es comparado con el anterior de forma que, si obtiene mejores resultados en la función de evaluación utilizada, se toma dicho modelo como base para realizar modificaciones. Si, por el contrario, proporciona unos resultados peores, se vuelve al modelo anterior y se modifican los parámetros de manera diferente. Los criterios de parada del algoritmo son los habituales: número máximo de iteraciones o diferencia no significativa en el resultado de dos modelos consecutivos.

Otra característica importante de este algoritmo es que puede optimizarse utilizando la GPU del sistema para ejecutarlo, ya que las unidades gráfica suelen explotar la paralelización de mejor manera que las CPUs. Frente a otros modelos de *ensembles* de árboles, XGBoost presenta una importante diferencia: considera la potencial pérdida de las divisiones realizadas para crear las ramas observando la distribución de las características en todos los elementos de una hoja y utilizando esta información para reducir el espacio de búsqueda de posibles divisiones de características [68].

Nuevamente, para utilizar el algoritmo *Random Forest* con series temporales, se realiza previamente el procesamiento descrito en 3.8.5 para las distintas variables incluidas en el modelo.

En este trabajo, el ajuste de XGBoost para la predicción de series temporales de valores se realizará con *Python* utilizando la función *XGBRegressor* del paquete *xgboost* 7.2.5. Los parámetros que se podrán ajustar de manera interactiva en la web son el número de árboles utilizado y la política de crecimiento de los árboles (favorecer las separaciones en nodos cercanos al nodo base o favorecer las separaciones en los nodos con mayor ratio de perdida).

En este modelo se permitirá incluir únicamente la variable del precio de cierre ajustado y utilizar todas las

métricas descritas en 3.8.3.

3.8.10 Modelo MLP

Redes neuronales

Dentro de los modelos de inteligencia artificial, aquellos basados en **redes neuronales** ocupan un lugar muy importante [69]. Partiendo de la idea de tratar de replicar el comportamiento del sistema nervioso humano, estos modelos computacionales tuvieron un gran apogeo en las décadas de los 80's y 90's, con un reciente repunte de popularidad con el crecimiento del ***deep learning*** . Esta idea de imitar el funcionamiento biológico del cerebro humano, tiene sus inicios en el **modelo de McCullot-Pitts**, que se muestra en la Figura 3.26.

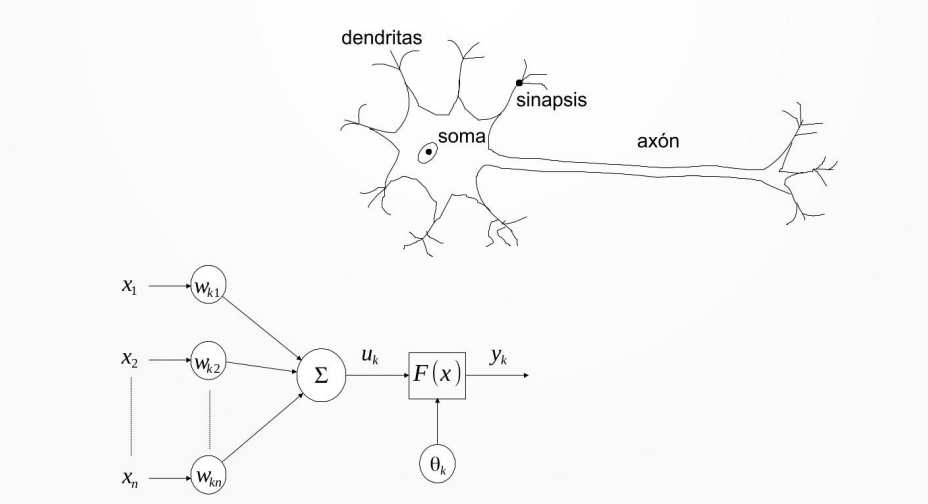


Figura 3.26: Comparación de la neurona humana con la neurona de McCullot-Pitts. Tomado de [69].

La neurona biológica recibe descargas eléctricas llamadas señales desde otra neurona y cuando recibe un suficiente número de ellas, dispara su propia señal en un proceso denominado sinapsis. Tienen un comportamiento simple, pero combinando un gran número de ellas se puede crear una red muy compleja.

Tras pasar por una época de pérdida de popularidad debido a la falta de potencia computacional para continuar su desarrollo, en 1957 surgió el modelo de redes neuronales más simple, el **perceptrón simple**, basado en la idea original comentada anteriormente. En este modelo, las entradas son números y la salida son dichos números ponderados por unos **pesos**, aunque previamente ha de pasar un filtro, que se conoce como **función de activación**. Esta función sirve para acotar los valores de modo análogo a como actúan las neuronas del cerebro, donde solo se captan señales eléctrica comprendidas en un rango.

El funcionamiento básico de una neurona, el cual puede verse en la Figura 3.26 es el siguiente:

$$u = w_0 + \sum_{i=1}^{n} w_i \times x_i$$

$$y = F(u + b)$$

Donde u se denomina **salida analógica** e y es la **salida digital** (la que se busca) obtenida tras aplicar la función de activación y de incluir el término b que hace referencia al **bias** o **umbral** (*threshold*), término que hace referencia al grado de inhibición de la neurona y que ayuda a que el rango numérico de los valores se sitúe en torno al 0.

El entrenamiento de una red neuronal busca ajustar el conjunto de pesos w_i y en él se utilizan diversas funciones de activación, siendo las principales las siguientes [70]:

- **Identidad:** no introduce ninguna transformación.

$$F(u) = u$$

- **Relu:** rectificador lineal. Anula los valores negativos y deja los positivos tal y como entran.

$$F(u) = max(0, u) = \begin{cases} 0 & \text{si} \quad u < 0 \\ u & \text{si} \quad u \geq 0 \end{cases}$$

- **Sigmoide:** transforma la entrada al intervalo $(0, 1)$, donde devuelve un valor cercano a 1 para los valores altos y un valor cercano a 0 para los valores bajos.

$$F_\beta(u) = \frac{1}{1 + e^{-\beta u}}, \beta = 1, 2, ...$$

- **Tanh:** tangente hiperbólica. Transforma la entrada al intervalo $(-1, 1)$, donde devuelve un valor cercano a 1 para los valores altos y un valor próximo a -1 para los valores bajos.

$$F(u) = \frac{2}{1 + e^{-2x}} - 1$$

- **Softmax:** transforma las salidas en una distribución de probabilidad, por lo que está en el rango (0, 1). Habitualmente se utiliza en la última capa de los problemas de clasificación.

$$F(z_j) = \frac{e^{z_j}}{\sum_{i=1}^{k} e^{z_i}}$$

- **Softplus:** versión suavizada del rectificador lineal. Es derivable en todo su dominio, a diferencia del relu, que presenta una discontinuidad en el origen.

$$F(u) = ln(e^u + 1)$$

Otras funciones de activación posibles sobre las que no se profundizará son *softsign*, *selu*, *elu* o *exponencial*.

Para la obtención de los pesos y el entrenamiento de la red se utiliza el algoritmo de **retropropagación** del error o *backpropagation*, propuesto en 1986 por Rumenhart en [71]. Este algoritmo consiste en un procedimiento de cálculo del gradiente donde los errores se propagan hacia atrás desde la capa de salida, de forma que a cada neurona le llega únicamente una fracción del mismo proporcional a la contribución de cada una de ellas a la salida y actualizándose el valor de los pesos en consecuencia. Esto permite que, mediante un proceso iterativo, las neuronas que forman parte de las distintas capas de la red vayan "aprendiendo" las características del espacio de entrada.

Frente a esta propagación hacia atrás está la propagación hacia delante, consistente en la obtención de la salida de la red de neuronas a partir de los pesos y los valores de entrada.

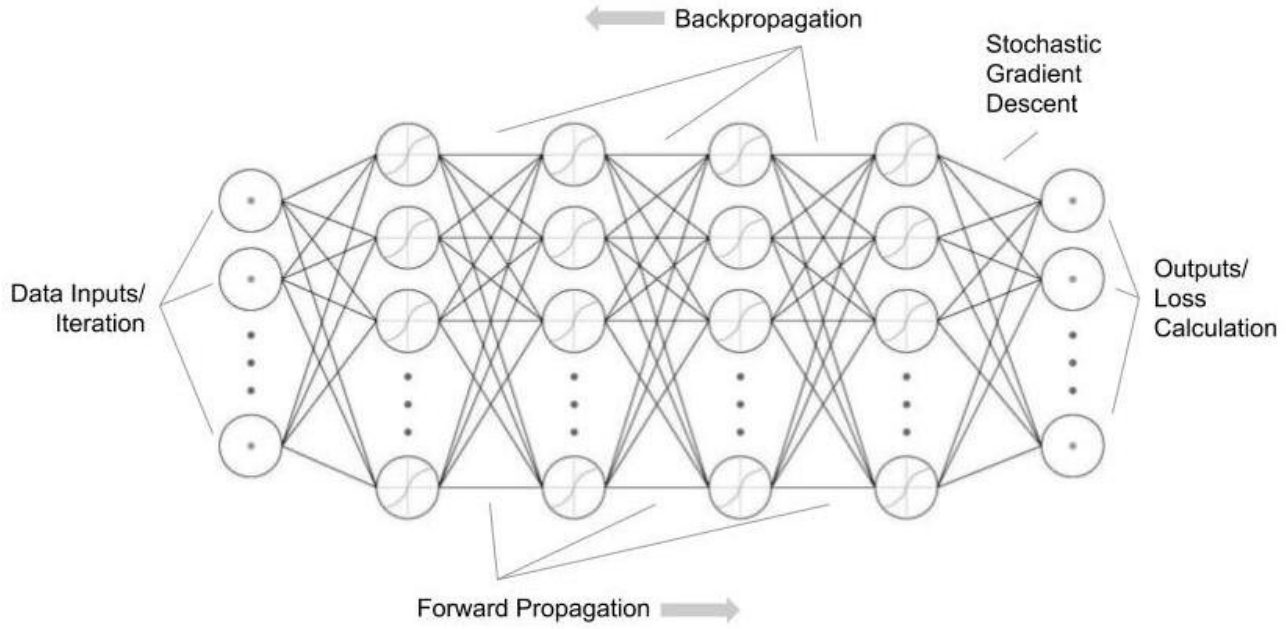


Figura 3.27: Propagación hacia delante y hacia atrás. Tomado de [72].

MLP

Debido a las limitaciones del perceptrón simple, únicamente útil en problemas linealmente separables, se propuso un modelo consistente en la combinación de un número grande de estas unidades, el **perceptrón multicapa** (*Multi Layer Perceptron*). Más allá de incrementar el número de neuronas frente al perceptrón simple, esta nueva arquitectura incluye como novedad la aparición de **capas ocultas** entre la entrada y la salida. En un esquema de capas, la salida de una neurona es la entrada de la siguiente, distinguiéndose a su vez entre arquitectura local o totalmente conectada. En caso de las primeras, la salida de una neurona de la capa i es entrada de un grupo (o unidad) de la capa $i+1$, mientras que en el segundo caso, la salida de cada neurona de la capa i es entrada de todas las neuronas de la capa $i+1$. Estas últimas suelen se denominadas **redes neuronales artificiales (ANN)** en la literatura.

Este nuevo planteamiento permite afrontar problemas mucho más complejos, donde la no linealidad no supone un problema. Cuando el perceptrón multicapa contiene más de una capa oculta, podemos hablar de una arquitectura de ***deep learning*** .

Un problema al que se enfrentan habitualmente las redes neuronales es el de la **evanescencia del gradiente**, consistente en que los pesos de las neuronas de las capas profundas a penas se modifican debido a que el gradiente que les llega es muy pequeño. Algunas de las arquitecturas que se comentarán posteriormente tienen entre sus objetivos lidiar con este problema y el uso de funciones de activación como el relu tratan de dar también una solución al mismo.

Nuevamente, para utilizar el algoritmo *Multi Layer Perceptron* con series temporales, se realiza previamente el procesamiento descrito en 3.8.5 para las distintas variables incluidas en el modelo.

En este trabajo, el ajuste de MLP para la predicción de series temporales de valores se realizará con *Python* utilizando la arquitectura proporcionada por *Keras* 7.2.5 para la utilización de *Tensorflow* 7.2.5. Los parámetros que se podrán ajustar de manera interactiva en la web son el número de capas ocultas (será obligatorio añadir al menos una), junto con el número de neuronas que componen cada una de ellas y la correspondiente función de activación de dicho nivel, el tamaño de la ventana temporal utilizada por el modelo, el optimizador a utilizar, la función de pérdida, el tamaño de los lotes, el número de época y el uso o no de *dropout*. Estos últimos parámetros se describen de la siguiente manera:

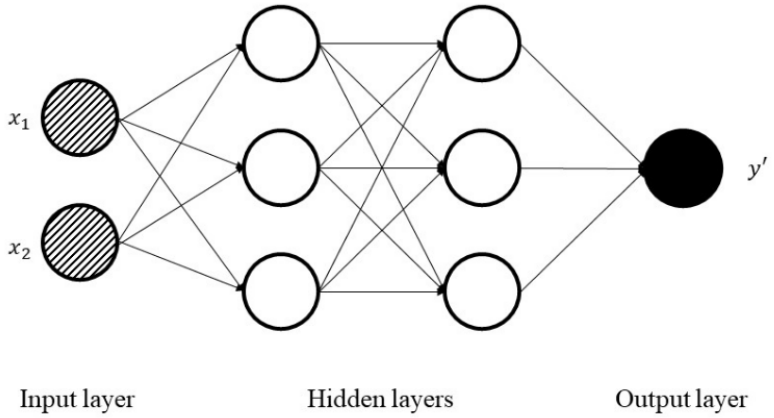


Figura 3.28: Topología de un perceptrón multicapa don dos capas ocultas. Tomado de [65].

- **Optimizador:** función que optimiza los parámetros de la red neuronal (pesos, tasa de aprendizaje). Hay múltiples:
  - **SGD:** *Stochastic Gradient Descent*. Utiliza el descenso del gradiente estocástico.
  - **RMSProp:** mantiene tasas de aprendizaje por parámetro que se adaptan en función de la media de las magnitudes recientes de los gradientes para el peso. Esto hace que el algoritmo funcione bien en problemas ruidosos.
  - **Adagrad:** mantiene una tasa de aprendizaje por parámetro que mejora el rendimiento en problemas con gradientes dispersos.
  - **Adam:** [73] extensión del SGD surgida en 2015 que combina las mejoras propuestas por los algoritmos RMSProp y Adagrad. En lugar de adaptar las tasas de aprendizaje de los parámetros basándose en el momento de primer orden (la media) como en RMSProp, Adam también hace uso de los momentos de segundo orden de los gradientes (la varianza no centrada). En concreto, el algoritmo calcula una media móvil exponencial del gradiente y del gradiente al cuadrado, y controla las tasas de descenso de estas medias móviles con los parámetros β_1 y β_2. Estos parámetros tienen un valor inicial cercano a 1, que da lugar a un sesgo hacia 0, que se supera calculando primero las estimaciones sesgadas antes de calcular después las estimaciones corregidas por el sesgo. El optimizador Adam habitualmente es considerado como la mejor opción por su rapidez y bajos requerimientos de memoria.
  - **Adadelta:** método de descenso de gradiente estocástico que se basa en la tasa de aprendizaje adaptativa por dimensión para resolver dos inconvenientes: el continuo descenso de las tasas de aprendizaje a lo largo del entrenamiento y la necesidad de una tasa de aprendizaje global seleccionada manualmente.
  - **Adamax:** variante del Adam basado en la norma infinito.
  - **Nadam:** RMSProp con *momentum*.
  - **Ftrl:** utiliza penalización l2, propia de la *Regresión Ridge*.

    Una comparación entre algunos de los distintos optimizadores comentados puede verse en la Figura 3.29.
- **Función de pérdida:** [74] Normalmente, con las redes neuronales, buscamos minimizar el error (función objetivo utilizada en la evaluación de una solución candidata, es decir, un conjunto de pesos). Como tal,

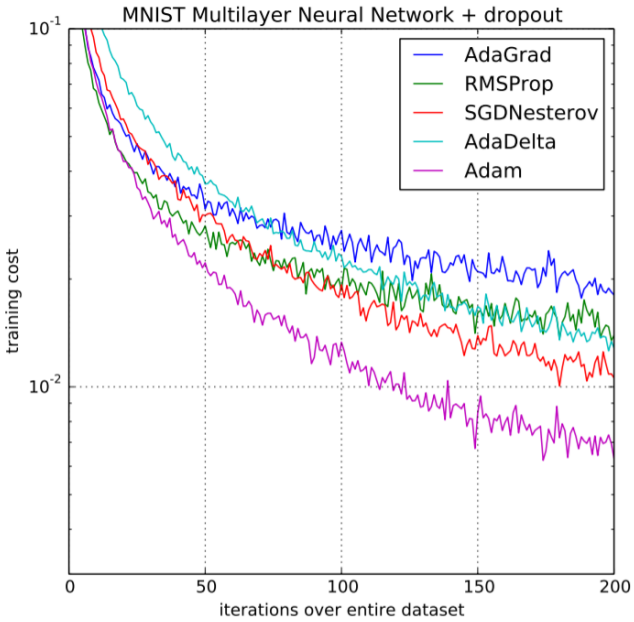


Figura 3.29: Comparación de optimizadores en términos de coste. Tomado de [73].

la función objetivo suele denominarse función de coste o función de pérdida y el valor calculado por la función de pérdida se denomina simplemente "pérdida". Existen multitud de funciones de pérdida, las consideradas en este trabajo, con el enfoque de regresión utilizado son:

- ***Mean squared error**:* se calcula como la media de las diferencias al cuadrado entre los valores previstos y los reales
- ***Mean absolute error**:* se calcula como la media de la diferencia absoluta entre los valores reales y los previstos.

- **Tamaño de los lotes:** *batch size* es el número de muestras que pasan por la red neuronal de una sola vez, es decir, el número de muestras que se procesan antes de actualizar el modelo. Su valor es un número entre 1 y el número de muestras del conjunto de entrenamiento.

- **Épocas:** *epochs* número de pasadas que el algoritmo de aprendizaje automático realiza sobre el conjunto de datos de entrenamiento. Su valor es cualquier número igual o mayor que 1.

  Como ejemplo de estos dos conceptos: supongamos que se tiene un conjunto de datos con 200 muestras y se elige un tamaño de lotes de 5 y 1000 épocas. Esto indica que el conjunto de entrenamiento se dividirá en 40 porciones de 5 muestras cada una, por lo que el modelo se actualizará un total de 40 veces. Como se realizan 1000 pasadas por el total del conjunto, el algoritmo trabajará con un total de 40000 lotes en total.

- ***Dropout*:** [75] uso o no de *dropout* en el modelo, consistente en aplicar un método de "desactivación" aleatoria de neuronas en el proceso de entrenamiento, de forma que dicha neurona no se tenga en cuenta ni en la obtención de la salida ni en el *backpropagation*. Esto ayuda a prevenir el **sobreajuste**, ya que las neuronas cercanas tienden a aprender patrones relacionados que pueden llegar a formar patrones muy específicos con los datos de entrenamiento. El parámetro de *dropout* en *Keras* hace referencia al porcentaje de neuronas que se desactivan. Puede incluirse de manera diferente en las distintas capas de una red neuronal, siendo más habitual utilizar una probabilidad de permanencia mayor en las primeras capas. En este trabajo se utilizará una única tasa para toda la red neuronal, por simplicidad.

En este modelo se permitirá incluir únicamente la variable del precio de cierre ajustado y utilizar todas las métricas descritas en 3.8.3.

3.8.11 Modelo Simple RNN

Redes neuronales recurrentes

Las redes neuronales recurrentes son un tipo de redes neuronales de *deep learning* que surgen del trabajo realizado por David Rumelhart en [76] con la idea de proponer un nuevo modelo que incluyera conexiones de una neurona consigo misma, directa o indirectamente. Esto aumenta enormemente las posibilidades de la red, ya que al haber más conexiones aumenta la capacidad de representación y el uso de estructuras recurrentes permite modelizar el uso de la variable **tiempo**. Este tipo de redes son muy populares hoy en día y tienen aplicaciones diversas que van desde la predicción de series temporales de activos financieros como en este trabajo, hasta la predicción de trayectorias en vehículos autónomos, el procesamiento del lenguaje o la clasificación de vídeos y textos.

Anteriormente, en el perceptrón, vimos que las funciones de activación de las redes únicamente actuaban hacia delante, sin recordar valores previos. En el caso de las redes recurrentes [77] esto cambia y existen conexiones hacia atrás, de forma que una neurona recibe la entrada x_t correspondiente a un momento de tiempo t y la salida del paso anterior y_{t-1}. Esto se conoce como **desenrolle a través del tiempo** y puede verse en la Figura 3.30.

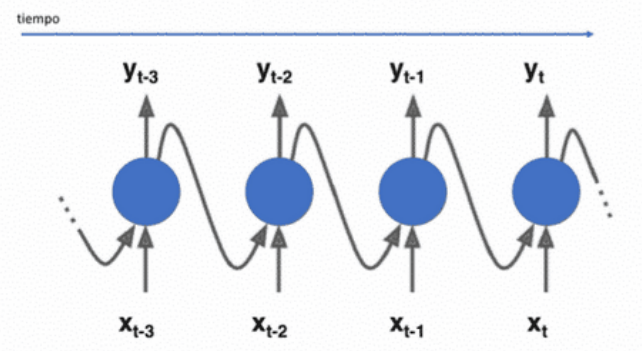


Figura 3.30: Desenrolle a través del tiempo de una red recurrente. Tomado de [77].

Con esta idea es posible construir capas de neuronas donde cada unidad reciba las dos entradas comentadas en cada instante de tiempo. Esto tiene la consecuencia de que cada neurona tiene dos conjuntos de parámetros, uno para la entrada de datos que recibe de la capa anterior y un segundo para la entrada que recibe del instante $t-1$. Por tanto, la neurona tiene dos conjuntos de pesos, W_x y W_y respectivamente, de forma que la salida podría expresarse como:

$$y_t = F(W_x x_t^T + W_y y_{t-1}^T + b)$$

Debido a que las redes recurrentes utilizan información de instantes anteriores, se dice que tienen "memoria". El elemento de las mismas encargado de conservar el estado de instantes anteriores recibe el nombre de ***memory cell*** y es la clave de que estos modelos sean tan útiles a la hora de tratar datos secuenciales como las series de tiempo de este proyecto.

Existen multitud de posibilidades para crear redes recurrentes. Algunos ejemplos se pueden ver e la Figura 3.31.

Al igual que con el perceptrón multicapa, para utilizar los distintos modelos de redes recurrentes con series temporales, se realiza previamente el procesamiento descrito en 3.8.5 para las distintas variables incluidas en el modelo.

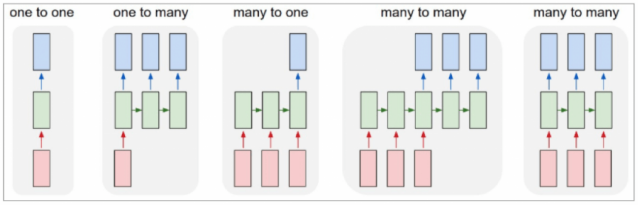


Figura 3.31: Estructuras de redes recurrentes. Tomado de [78].

En este trabajo, el ajuste de los modelos de redes recurrentes para la predicción de series temporales de valores se realizará con *Python* utilizando la arquitectura proporcionada por *Keras* 7.2.5 para la utilización de *Tensorflow* 7.2.5. Los parámetros que se podrán ajustar de manera interactiva en la web son el número de capas ocultas (será obligatorio añadir al menos una), junto con el número de neuronas que componen cada una de ellas y la correspondiente función de activación de dicho nivel, el tamaño de la ventana temporal utilizada por el modelo, el optimizador a utilizar, la función de pérdida, el tamaño de los lotes, el número de época y el uso o no de *dropout*. Existe una diferencia en el uso de este último frente al del modelo MLP, donde ahora se incluye de dos formas distintas:

1. ***Input dropout*:** aplicado a la conexión de entrada dentro de los nodos de la red.
2. ***Recurrent dropout*:** aplicado a la señal de entrada recurrente de los nodos de la red.

Los modelos de redes recurrentes permitirán incluir la totalidad de las variables descritas en 3.8.4 y utilizar todas las métricas descritas en 3.8.3.

Simple RNN

El modelo **Simple RNN** aplica las ideas de las redes recurrentes de manera básica. Mantiene la estructura de las redes neuronales habituales donde hay una capa de entrada, una de salida y una o varias capas ocultas con la diferencia de que cada capa tiene varias entradas del modo explicado en el apartado anterior. Únicamente recuerda la salida del instante $t-1$ y el resto, a través del estado. En estos modelos, la retropropagación se utiliza de manera diferente, ya que se utiliza a través del tiempo con el desenrolle a través del tiempo.

3.8.12 Modelo LSTM

Las redes *Long Short Term Memory* [79] son una variante de las redes neuronales recurrentes capaces de manejar dependencias a largo plazo y diseñado para manejar datos de secuencias temporales recogidos a intervalos iguales, como los datos de transacciones periódicas de este trabajo. Esto lo consiguen mediante un mecanismo de puerta (*gate mechanism*), una forma de permitir que la información fluya. Las puertas se componen de una capa de red neuronal sigmoidea y una operación de multiplicación puntual. Esto produce una salida acotada entre 0 y 1 que indica cuánto debe dejarse pasar de cada componente. Una unidad LSTM tiene tres de estas puertas. A su vez, la clave de estas redes es el estado de cada unidad (también llamadas células), que es referido en la literatura como una especie de "cinta transportadora", que permite recordar entradas anteriores, aunque estas estén alejadas en el tiempo. Esto último se ve en la línea superior de la Figura 3.32.

Los vectores h_t y c_t son los encargados de guardar el estado, a corto plazo en el primer caso y a corto y largo plazo en el segundo. c_{t-1} pasa en primer lugar por una *forget gate*, donde pierde parte de los datos que tenía guardados para incluir luego nueva información en una *input gate*. El resultado de c_t no sufre ninguna transformación. h_t convierte el estado de corto plazo aplicando la función *tangente hiperbólica*.

Tanto la entrada x_t como el vector h_{t-1} se conectan con 4 capas diferentes:

1. *Forget gate*: información a largo plazo a olvidar. Nos referiremos a ella como f_t.

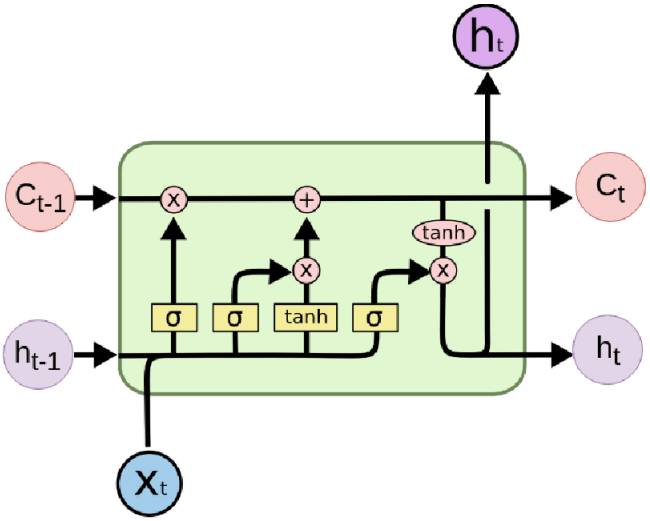


Figura 3.32: Celda de una red LSTM. Tomado de [80].

2. Da la salida, realizando una transformación con x_t y h_{t-1} para construir la información a recordar a largo plazo. Nos referiremos a ella como g_t.

3. *Input gate*: nueva información a largo plazo a recordar. Nos referiremos a ella como i_t.

4. *Output gate*: información a largo plazo proporcionada como salida (y_t y h_t). Nos referiremos a ella como o_t.

El símbolo $\oplus$ indica una operación de adición y el símbolo $\otimes$ indica la multiplicación elemento a elemento. Los elementos con los que se realizan los cálculos son los siguientes:

- Pesos de las matrices de las capas que se conectan con el vector de entrada x_t: W_{xf}, W_{xf}, W_{xi} y W_{xo}.
- Pesos de las matrices de las capas que se conectan con el vector de estado a corto plazo h_t: W_{hf}, W_{hf}, W_{hi} y W_{ho}.
- Bias de cada una de las capas: b_f, b_g, b_i y b_o.

Y los cálculos propiamente dichos son:

$$f_t = \sigma_g(W_f x_t + U_f c_{t-1} + b_f)$$

$$g_t = tanh(W_{xg}^T * x_t + W_{hg}^T * h_{t-1} + b_g)$$

$$i_t = \sigma_g(W_i x_t + U_i c_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o c_{t-1} + b_o)$$

$$c_t = f_t \otimes c_{t-1} \oplus i_t \otimes g_t$$

$$h_t = y_t = \sigma_h(o_t \otimes tanh(c_t))$$

Donde σ_g representa la función sigmoide.

3.8.13 Modelo GRU

Las redes *Gated Recurrent Unit* son otro tipo de redes recurrentes. Constituyen una versión simplificada de las redes LSTM y las diferencias con estas [81] se muestran en la tabla 3.2.

LSTM	**GRU**
Tres puertas	Dos puertas
Tiene memoria interna	No tiene memoria interna
Tiene *output gate*	No tiene *output gate*
Entrada y destino en puerta de actualización	Inicio directamente al estado oculto anterior.

Cuadro 3.2: Comparativa GRU y LSTM.

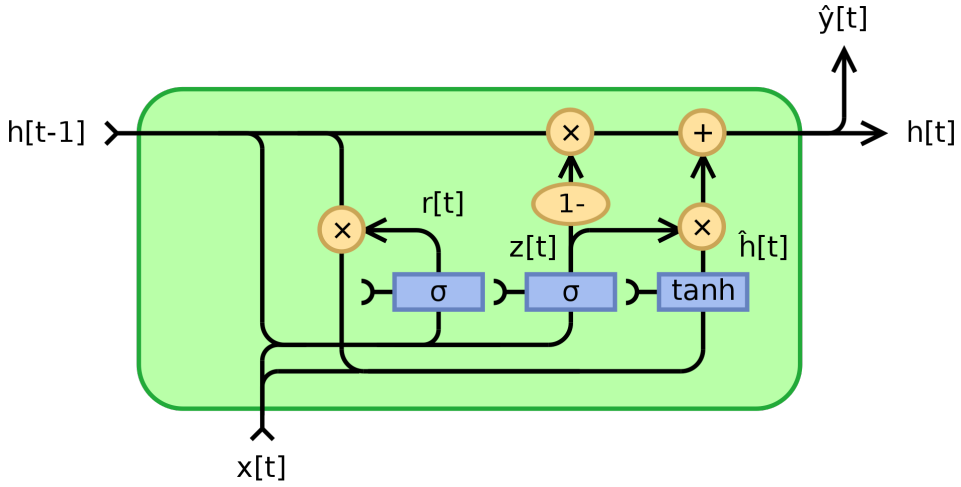


Figura 3.33: Celda de una red GRU. Tomado de [82].

Los cálculos ahora son:

$$z_t = \sigma_g(W_z x_t + U_z c_{t-1} + b_z)$$

$$r_t = \sigma_g(W_r x_t + U_r c_{t-1} + b_r)$$

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes tanh(W_h x_t + U_r(r_t \otimes h_{t-1} + b_h)$$

Donde z_t es la puerta de actualización y r_t es la puerta de *reset*.

3.8.14 Modelo CNN

Las redes neuronales convolucionales o *Convolutional Neural Networks* son otro tipo de redes neuronales de *deep learning* [83], habitualmente utilizadas en el procesamiento de datos que tienen "patrones de rejilla" como es el caso de las imágenes y que basan su funcionamiento en el aprendizaje jerárquico de características, para lo cual se valen de la operación de **convolución**. Las redes CNN se componen de tres tipos de capas:

- ***Convolutional layer:*** realiza la extracción de característica mediante operaciones de convolución y mediante operaciones no lineales.

- **Convolución:** utiliza una pequeña matriz de números denominada **kernel** que hace las veces de filtro y que se aplica sobre una matriz de entrada. Cada filtro utiliza un conjunto compartido de pesos para realizar la operación de convolución y estos se actualizan durante el proceso de de entrenamiento. Se calcula un producto elemento a elemento entre cada elemento del núcleo y la entrada en cada posición de la misma para obtener mediante una suma el valor de salida de cada posición correspondiente de la salida, denominada **mapa de características**. Este procedimiento se repite aplicando múltiples núcleos para formar un número arbitrario de mapas de características que representan diferentes características de la entrada. Los kernel suelen ser siempre matrices pequeñas y de dimensión impar: 3×3, 5×5 o 7×7.

  Mediante la realización iterativa de esta operación se reduce la dimensión de la entrada, realizando avances en la posición del kernel (*strides*) de tamaño habitualmente 1. Si en algún momento se llega a salir de la dimensión de la matriz de entrada, se añaden elementos artificialmente en lo que se conoce como ***padding*** o relleno. Las arquitecturas modernas de CNN suelen emplear relleno cero para retener las dimensiones en el plano con el fin de aplicar más capas. Sin el relleno cero, cada mapa de características sería sucesivamente más pequeño después de la operación de convolución.

  La característica principal de una operación de convolución es la compartición de pesos: los núcleos se comparten entre todas las posiciones de la imagen. Esto permite que:

  1. Las características locales extraídas por el kernel permanezcan invariantes mientras este se mueve por la entrada.
  2. Aprendizaje de jerarquías espaciales de patrones de características mediante muestreo junto con una operación de agrupación, lo que permite capturar un campo de visión cada vez más amplio.
  3. Aumenta la eficiencia del modelo al reducir los parámetros a aprender en comparación con otras arquitecturas de redes neuronales.

- **Operaciones no lineales:** una vez realizada la convolución, se aplica una función de activación no linear, del mismo modo que en modelos anteriores.

Las Figuras 3.34, 3.35 y 3.36 describen el funcionamiento de la capa convolucional.

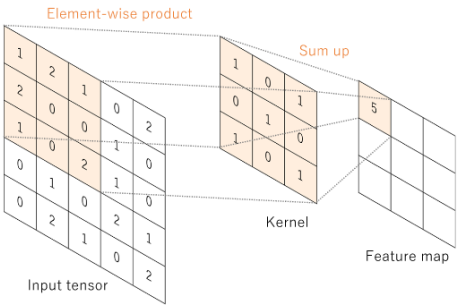


Figura 3.34: Operación de convolución (1). Tomado de [83].

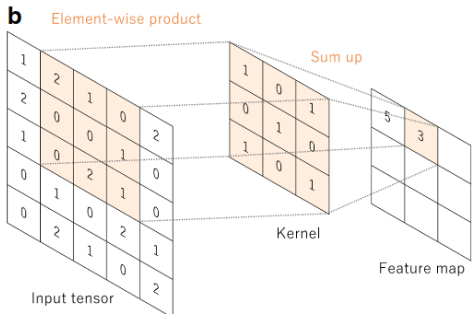


Figura 3.35: Operación de convolución (2). Tomado de [83].

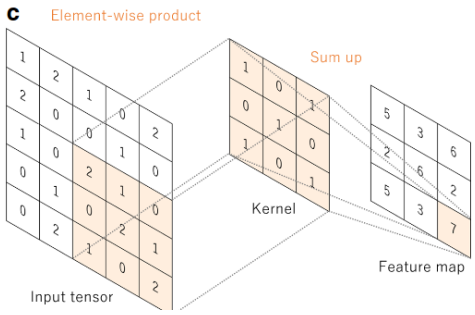


Figura 3.36: Operación de convolución (3). Tomado de [83].

- ***Pooling layer:*** operación de reducción de dimensionalidad (submuestreo) aplicada sobre el espacio de características, con el objetivo de reducir el número de parámetros de aprendizaje de la red y de lidiar con el posible sobreajuste. Todos valores dentro de una *pooling window*, que no es otra cosa que una matriz de dimensión pequeña no necesariamente impar como anteriormente, se convierten en un único valor. Pueden realizarse distintas operaciones en esta capa, aunque lo más habitual es quedarse con el máximo de la ventana en lo que se conoce como ***max pooling***, que es lo que se utilizará en este trabajo.

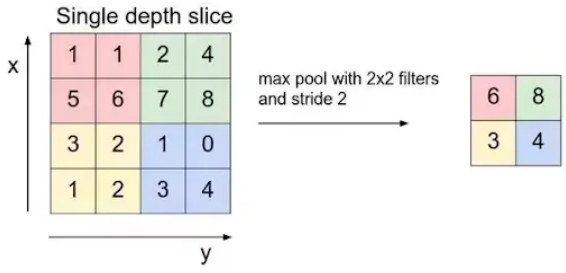


Figura 3.37: Ejemplo de *max pooling*. Tomado de [84]

- ***Fully-connected layer:*** la salida de la última capa de *pooling* habitualmente se aplana y se introduce en una red completamente conectada, que no es otra cosa que un perceptrón multicapa donde cada entrada se conecta con cada salida.

Existen múltiples arquitecturas de CNN, pero en este trabajo utilizaremos la arquitectura **ResNet**, consistente en alternar una capa convolucional con una capa de *pooling* de manera sucesiva, e incluir un MLP a modo de capa completamente conectada.

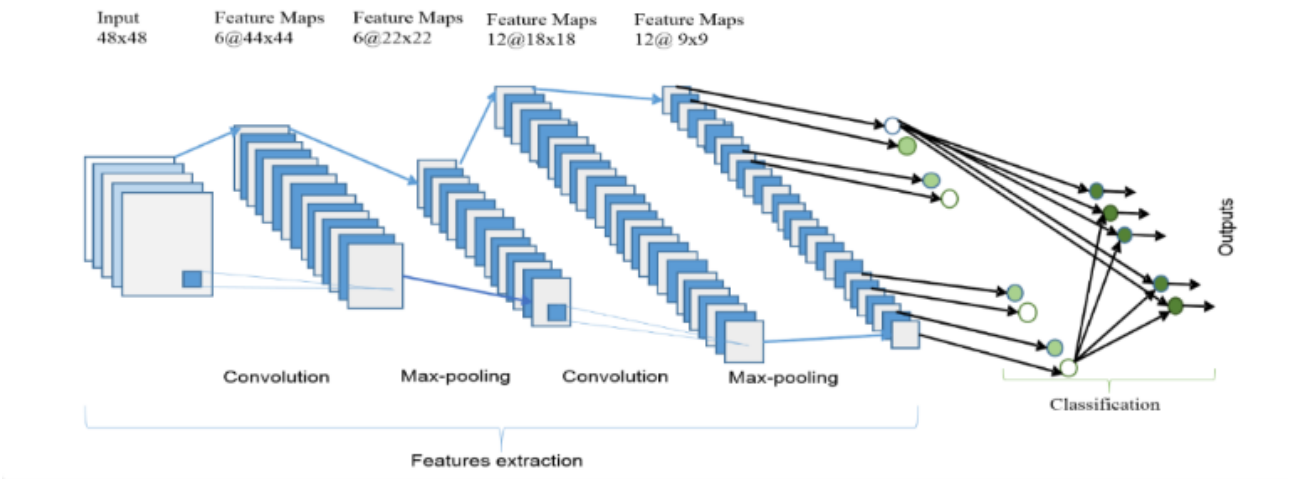


Figura 3.38: Ejemplo de arquitectura de CNN ResNet. Tomado de [85].

Al igual que en los modelos anteriores, para utilizar el modelo CNN, se realiza previamente el procesamiento descrito en 3.8.5 para las distintas variables incluidas en el modelo.

En este trabajo, el ajuste del modelo CNN para la predicción de series temporales de valores se realizará con *Python* utilizando la arquitectura proporcionada por *Keras* 7.2.5 para la utilización de *Tensorflow* 7.2.5. Los parámetros que se podrán ajustar de manera interactiva en la web son el número de filtros de salida de cada capa convolucional, así como su función de activación y el número de estas que se dese añadir (será obligatorio añadir al menos una), el número de neuronas que componen cada una de las capas del MLP y la correspondiente función de activación de cada nivel, el tamaño del kernel utilizado en el *pooling*, el tamaño de la ventana temporal utilizada por el modelo, el optimizador a utilizar, la función de pérdida, el tamaño de los lotes, el número de época y el uso o no de *dropout*. En este trabajo, el *dropout* de las redes CNN se incluye entre las distintas capas del MLP de la capa totalmente conectada, aunque podría incluirse entre las capas de la red resnet o dentro de la misma.

En este modelo se permitirá incluir únicamente la variable del precio de cierre ajustado y utilizar todas las métricas descritas en 3.8.3.

3.9 Otras técnicas

Otras técnicas de Inteligencia Artificial utilizadas en este trabajo son los **modelos de regresión**, comentados en 3.7.4, y el **análisis *cluster*** , empleado con el objetivo de ver qué activos, de un conjunto inicial, tienen un comportamiento similar en términos de rentabilidad a lo largo de un periodo temporal elegido por el usuario.

3.9.1 Clustering

En análisis *cluster* o análisis de grupos [82] es un conjunto de técnicas que tienen el objetivo de agrupar muestras o individuos de un conjunto de datos atendiendo a sus diferencias y semejanzas. Un *cluster* es, por tanto, cada uno de los conjuntos obtenidos como resultado de la aplicación del análisis. Tiene aplicaciones en múltiples campos, como la segmentación de clientes, el estudio demográfico o el estudio de patrones climáticos.

Puesto que no precisa de salida deseada, el análisis de grupos constituye un tipo de problema **no supervisado** (ver 3.8.1). Dentro de los métodos de *clustering* se realiza una división básica en dos categorías, cuya taxonomía desarrollada se muestra en la Figura 3.39:

- **Métodos jerárquicos:** la pertenencia a un *cluster* condiciona su posible pertenencia en niveles superiores.

Se parte de un único *cluster* que progresivamente se va dividiendo (métodos divisivos) o de tantos *clusters* como individuos, que iterativamente se van agrupando (métodos aglomerativos).

- **Métodos particionales:** se generan particiones disjuntas con un algoritmo y posteriormente se evalúan en base a algún criterio.

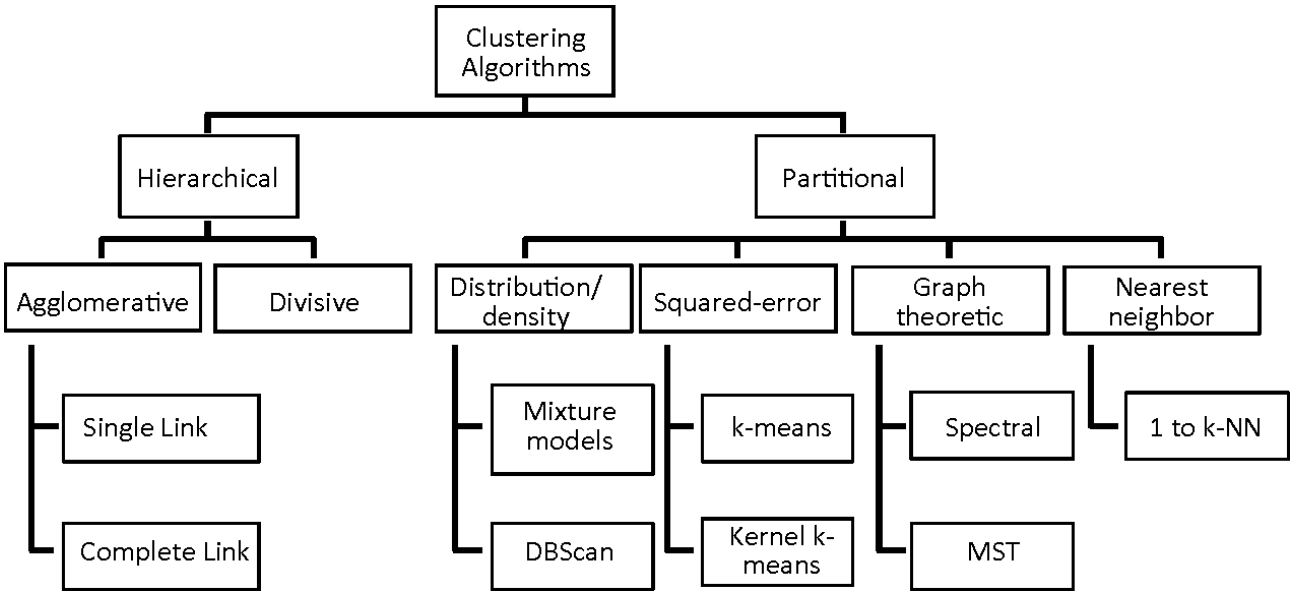


Figura 3.39: Taxonomía de los métodos de *clustering* . Tomado de [86].

En este trabajo el método utilizado a la hora de realizar las agrupaciones de activos es particional, siendo el algoritmo utilizado el de **K medias** o *K means*. Este método requiere de fijar previamente el número de grupos que se desean formar y el agrupamiento se realiza minimizando la distancia entre cada observación y el *centroide* de su *cluster* (el centro del mismo), utilizando una medida de distancia, que habitualmente es la distancia cuadrática. Se busca que los grupos sean lo más heterogéneos posible entre sí y que a su vez los individuos que lo componen sean lo más homogéneos posible. Es un algoritmo muy popular que destaca por su simpleza y eficiencia, aunque es muy sensible a la configuración de centros que se realice inicialmente y sus resultados se ven seriamente afectados en conjuntos de datos que presenten ruido o valores ausentes. Los pasos que sigue este algoritmo son:

1. **Inicialización:** fijado el número k de grupos, se escogen k *centroides* en el espacio de datos. Esta elección puede realizarse de manera aleatoria, de manera prefijada o utilizando algún tipo de heurística. Habitualmente se prefiere la inicialización aleatoria o utilizar una heurística que busque el baricentro de la muestra y escoja las k más alejadas del mismo, buscando maximizar la heterogeneidad entre los grupos.

2. **Asignación de individuos:** cada individuo del conjunto de datos se asigna al centroide más cercano.

3. **Actualización de *centroides*:** se actualiza la posición del *centroide* de cada grupo tomando como nuevo *centroide* la posición del promedio de los elementos pertenecientes a dicho grupo.

El algoritmo se repite iterativamente hasta que los *centroides* permanecen estables (la diferencia entre dos pasos consecutivos se encuentra por debajo de cierto umbral) o hasta que se alcanza un máximo de iteraciones fijado de antemano.

El algoritmo de K medias se considera un problema de optimización, donde la función objetivo consiste en minimizar la suma de las distancias al cuadrado de cada individuo a su centroide. Matemáticamente esto se puede expresar como:

$$min(E_{\mu_i}) = min\left(\sum_{i=1}^{k}\sum_{x_j \in S_i} ||x_j - \mu_i||^2\right)$$

Donde k es el número de grupos, el vector $x = (x_1, ..., x_n)$ los datos $d - dimensionales$, $S_1, ..., S_k$ los grupos y μ_i el *centroide* del grupo i.

Un ejemplo del funcionamiento del algoritmo se muestra en la Figura 3.40.

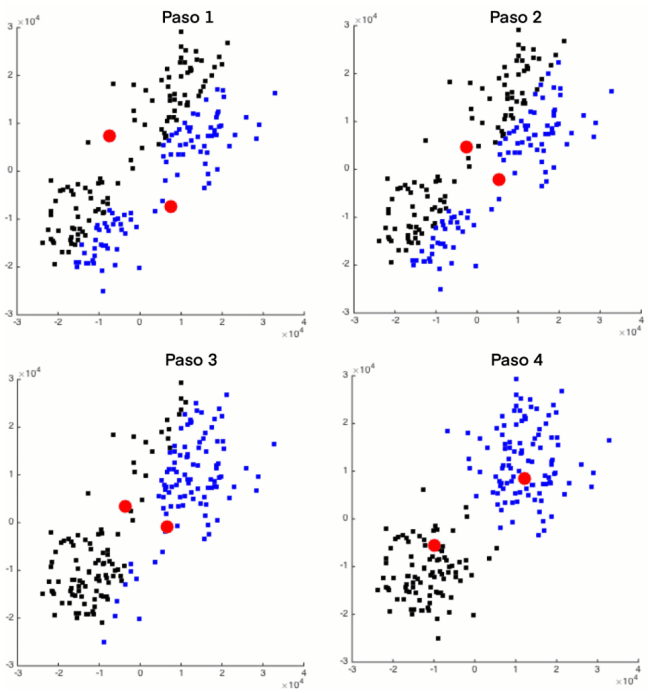


Figura 3.40: Ejemplo algoritmo K medias. Tomado de [87].

Para poder utilizar el algoritmo de K medias con datos de series temporales exiten múltiples métodos. Algunos de ellos se basan en la correlación entre las series temporales, otros utilizan transformaciones de Fourier, aunque en este trabajo se propone utilizar **Análisis de Componentes Principales (PCA)** como método de reducción de la dimensionalidad que permita representar los distintos activos en un espacio bidimensional donde se busquen los grupos.

Análisis de Componentes Principales

El análisis de Componentes Principales [88] consiste en un procedimiento multivariante de interdependencia, es decir, no hay variable respuesta. Se busca estudiar las relaciones entre las variables y los individuos del conjunto de datos y el objetivo fundamental es **reducir la dimensión** del mismo perdiendo la mínima cantidad de información posible. Para ello se construye un **nuevo conjunto de variables** a partir de combinaciones lineales de las variables originales, buscando recoger la mayor cantidad de información o variabilidad posible. Es un método diseñado para variables continuas que en muchas ocasiones se utiliza como paso previo de otras técnicas como le Regresión, el Análisis Discriminante o, como en este trabajo, el Análisis *Cluster*. El método busca encontrar la estructura subyacente de los datos, viendo en qué direcciones la varianza y, por tanto, la variabilidad, es mayor. Utilizando PCA se reducirá la dimensión de las series temporales con las que se trabaja en

este proyecto de forma que se tenga un conjunto bidimensional (con las dos primeras componentes principales) que permita la representación en un espacio bidimensional donde podamos realizar el *clustering* .

Para hacer todo lo anterior, utiliza transformaciones ortogonales y descompone el conjunto de datos en autovectores y autovalores, habiendo un par de estos últimos para cada nueva dimensión generada y denominándose componente principal al autovector con mayor autovalor de la dimensión. No profundizaremos más en este tema ya que matemáticamente tiene una complejidad que no se pretende abordar en este trabajo.

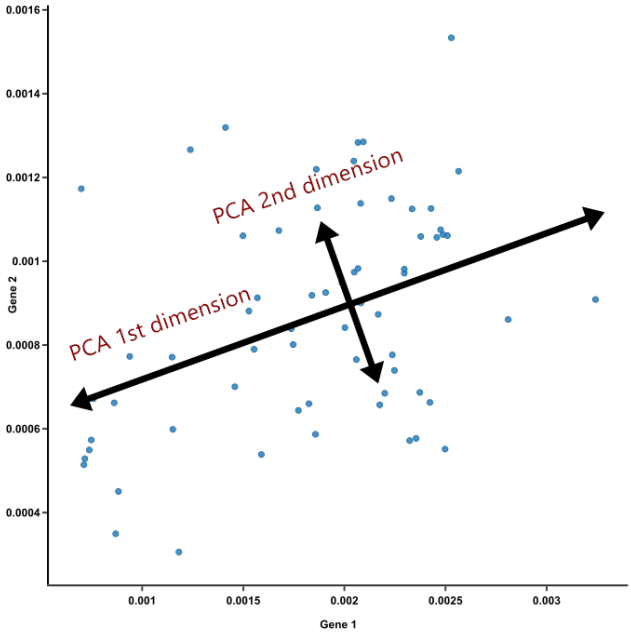


Figura 3.41: Análisis de Componentes Principales. Tomado de [89].

Parte III

Desarrollo del Sistema

Capítulo 4

Fuentes de datos

En el presente trabajo se ha decidido trabajar con los activos comentados en las secciones 3.3, 3.4, 3.5 y 3.6. Todos los datos se extraen en tiempo de ejecución de la web de ***Yahoo Finance*** [90] directamente con R utilizando la función *getSymbols* de la librería *tidyquant* [91]. La elección de esta web como fuente de datos única del trabajo se debe a la sencillez que proporciona para acceder a los datos a través de la *API* (*Application Programming Interface* [92] o interfaz de aplicación de aplicaciones dicho en español. Consiste en la especificación formal que establece cómo un módulo de un software se comunica o interactúa con otro para cumplir con diferentes funciones.) y a que dicho acceso puede realizarse en tiempo de ejecución (sin necesidad de descargar los datos) y de manera gratuita sin limitaciones (más allá de la disposición de valores que proporcione la web). Cabe mencionar que otras webs como *Quandl* o *Bloomberg* también podrían haberse utilizado, ya que permitían la extracción de datos a través de sus *APIs*, aunque por las razones expuestas *Yahoo Finance* se presentó como una mejor opción.

Comentar en este punto que, de cara a agilizar la sección de mercados de la web, donde el numeroso grupo de conjuntos de datos a obtener ralentizaba enormemente el servicio, con un rendimiento muy pobre, se optó por tener los datos utilizados en esas secciones ya descargados previamente antes de que el usuario acceda a ellas. El modo de conseguir esto es con la utilización de lo que se conoce como ***cron jobs*** [93]. *Cron* es una utilidad de los sistemas *Unix* que permite programar la ejecución automática de tareas, en este caso, la ejecución automática de un *script* de R que se encarga de descargar por la noche los datos necesarios par la sección comentada y depositarlos en documentos CSV, de forma que su lectura sea inmediata no comprometiendo el funcionamiento de la aplicación.

Los aspectos negativos y a mejorar de las fuentes de datos de este trabajo se comentan en el el capítulo 9.

De cara a facilitar la extracción de los datos, la cual ha de hacerse a través de los símbolos asociados a los valores, en primer lugar se procedió a la creación de un objeto *.RData* que contiene el listado completo de valores y símbolos que están disponibles en la web. Para ello, se hizo *scraping web* (proceso consistente en la extracción automática de información de una página web) con la herramienta ***ParseHub*** *7.2.1* en su versión gratuita y se consultaron diversas páginas de internet de las que se extrajeron los nombres de los valores y el símbolo que *Yahoo Finance* asigna a cada uno de ellos. Alguna correspondencia entre símbolos y valores tuvo que hacerse de forma manual y algunos valores tuvieron que modificarse por haber sufrido la empresa un cambio de nombre desde que la información fue escrita en la web elegida. Incluso una empresa del índice NASDAQ-100 3.3.2 como *Maxim Itegrated* tuvo que suprimirse al haber dejado de pertenecer al índice. Otros activos se suprimieron al ser excesivamente habitual que sus datos no pudieran descargarse correctamente de la API.

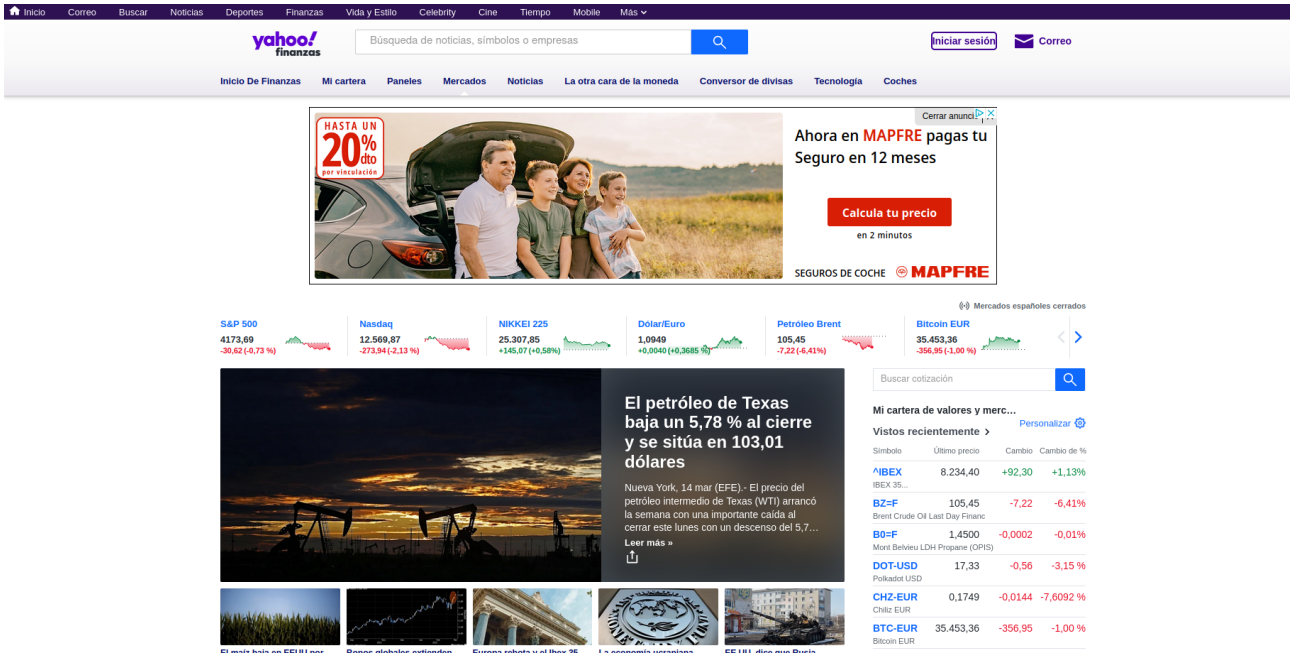


Figura 4.1: Pantalla de inicio de la web de Yahoo Finance.

Capítulo 5

Análisis

Algunos de los aspectos pertenecientes a esta sección y a la sección de diseño han tenido como trabajo de referencia [94], realizado por Javier Gatón en el curso 2020-2021.

5.1 Descripción del sistema

El proyecto consiste en la creación de una plataforma web que permita interactuar y realizar diferentes pruebas relacionadas con valores de activos económicos, permitiendo realizar actividades como optimizar carteras, predecir valores utilizando modelos de inteligencia artificial, realizar *clustering* de valores, regresiones sobre índices bursátiles, simulaciones de carteras con el método de Montecarlo o análisis descriptivos. Todo esto se hará dando un enfoque didáctico que permita al usuario entender en todo momento lo que está haciendo, valiéndose de una interfaz sencilla e intuitiva.

5.2 Historias de usuario

Dentro del marco de la metodología ágil en el que se encuadra este proyecto (ver 2.1), una forma muy interesante de enfocar la parte de análisis y de hablar sobre los requisitos es mediante la escritura de lo que se conoce como **historias de usuario** [95]. Estas no son más que descripciones breves y sencillas de las características o funcionalidades que se desean del sistema, vistas desde el punto de vista del usuario o cliente final. La estructura en la que se suelen redactar se puede observar en la Figura 5.1.

Cada historia debe ir a su vez acompañada de sus correspondientes **criterios de aceptación** (habitualmente entre 1 y 4), que son sentencias que serán ciertas una vez se complete la historia de usuario. A su vez, han de ser redactados en una frase que indique el contexto, el evento y el comportamiento esperado ante el evento.

Las historias de usuario se escriben a lo largo de todo el proyecto ágil y siempre resulta interesante que todos los miembros de un equipo de desarrollo formen parte activa de este proceso, permitiendo además discutir sobre las mismas en las reuniones periódicas realizadas en las metodologías ágiles, donde estas forman el denominado ***Product Backlog***, que es una lista priorizada de las funcionalidades a desarrollar.

En este trabajo, las historias se escriben al completo al comienzo del mismo, con el objetivo de poder definir posteriormente los requisitos del sistema de la forma habitual.

Figura 5.1: Estructura de las historias de usuario. Tomado de [95].

- **Historia de usuario 1:**
  - **Como** usuario
  - **Quiero** seleccionar un mercado
  - **Para** ver cuáles son los principales activos del mismo, conocer su valor y evolución en el tiempo.
  - **Criterios de aceptación:**
    - Se puede seleccionar un mercado y ver la información relativa al mismo.
    - Se muestran los títulos del mercado seleccionado acompañados de su valor y del porcentaje de subida o bajada con respecto al día anterior.
    - Se puede seleccionar un valor de un mercado y ver la gráfica de la serie temporal del mismo.
- **Historia de usuario 2:**
  - **Como** usuario
  - **Quiero** modificar la temporalidad de los datos de un gráfico
  - **Para** poder ver la información del activo en el periodo temporal deseado.
  - **Criterios de aceptación:**
    - Se puede modificar la temporalidad de un gráfico mediante un menú seleccionable que hace que el gráfico cambie de manera reactiva al modificar la selección del periodo temporal.
- **Historia de usuario 3:**
  - **Como** usuario
  - **Quiero** ver de manera conjunta los valores de los principales mercados
  - **Para** conocer el valor de los mismos.
  - **Criterios de aceptación:**
    - Se muestran los títulos de los mercados acompañados de su valor y del porcentaje de subida o bajada con respecto al día anterior.
- **Historia de usuario 4:**
  - **Como** usuario

- **Quiero** seleccionar un conjunto de valores
- **Para** analizar distintos gráficos sobre la evolución temporal de los valores.
- **Criterios de aceptación:**
  - Se pueden elegir distintos tipos de gráficos.
  - Se pueden superponer gráficos de series temporales de valores, del mismo modo que se puede eliminar un valor de la gráfica.
  - Se puede superponer la información de diversos indicadores técnicos en el gráfico de la serie temporal de un único valor.
  - Se puede modificar la temporalidad de los gráficos mostrados.

- **Historia de usuario 5:**

  - **Como** usuario
  - **Quiero** poder interactuar con los distintos gráficos con el ratón.
  - **Para** poder ver el valor de un gráfico en una zona concreta, hacer zoom o personalizar la información visualizada.
  - **Criterios de aceptación:**
    - Al pasar el ratón por cualquier zona de cualquier gráfico se muestra el valor del gráfico en dicha zona.

- **Historia de usuario 6:**

  - **Como** usuario
  - **Quiero** obtener la cartera óptima que maximiza el beneficio o minimiza el riesgo con un conjunto de valores seleccionados y una serie de restricciones
  - **Para** poder diversificar la inversión de modo adecuado.
  - **Criterios de aceptación:**
    - Se puede obtener la cartera óptima que maximiza el beneficio teniendo en cuenta las restricciones establecidas.
    - Se puede obtener la cartera óptima que minimiza el riesgo teniendo en cuenta las restricciones establecidas.
    - Se muestra la información de la cartera óptima de manera tanto tabulada como gráfica.

- **Historia de usuario 7:**

  - **Como** usuario
  - **Quiero** resolver el problema de de optimización de carteras que de manera simultánea maximiza la rentabilidad y minimiza el riesgo de un conjunto de valores seleccionados y una serie de restricciones establecidas
  - **Para** poder diversificar la inversión de modo adecuado.
  - **Criterios de aceptación:**
    - Se puede obtener la frontera eficiente de todas las carteras que constituyen una solución del problema de optimización planteado.
    - Se pueden obtener las carteras notables y ver su composición de manera gráfica y tabulada.
    - Se pueden obtener gráficas con la frontera eficiente y con las carteras notables.

- **Historia de usuario 8:**
  - **Como** usuario
  - **Quiero** evaluar el rendimiento de una cartera
  - **Para** conocer la rentabilidad obtenida con la misma.
  - **Criterios de aceptación:**
    - Se muestra de manera gráfica la rentabilidad proporcionada por una cartera.
- **Historia de usuario 9:**
  - **Como** usuario
  - **Quiero** estudiar la relación entre una cartera y un índice bursátil
  - **Para** ver cómo afecta el aumento en el índice al rendimiento del valor o cartera.
  - **Criterios de aceptación:**
    - Asegurarse de que se seleccionan valores pertenecientes a un único índice bursátil.
    - Se muestra la recta de regresión del valor o cartera frente al índice.
- **Historia de usuario 10:**
  - **Como** usuario
  - **Quiero** simular carteras
  - **Para** obtener carteras eficientes de forma no analítica.
  - **Criterios de aceptación:**
    - Se representan las carteras simuladas por el método de Montecarlo.
- **Historia de usuario 11:**
  - **Como** usuario
  - **Quiero** obtener la predicción de un valor para el día siguiente
  - **Para** tomar una decisión sobre la inversión a realizar.
  - **Criterios de aceptación:**
    - Se puede obtener la predicción utilizando distintos modelos de inteligencia artificial.
    - Se pueden comparar los distintos modelos utilizados.
- **Historia de usuario 12:**
  - **Como** usuario
  - **Quiero** seleccionar modelos de inteligencia artificial para realizar predicciones
  - **Para** poder configurar los distintos parámetros de los mismos, así como configurar su entrenamiento y evaluación; además de poder compararlos.
  - **Criterios de aceptación:**
    - Se puede obtener la predicción utilizando distintos modelos de inteligencia artificial.
    - Se pueden comparar los distintos modelos elegidos.
- **Historia de usuario 13:**

- **Como** usuario
- **Quiero** ver qué valores tienen un comportamiento semejante en términos de rentabilidad
- **Para** conocer qué inversiones pueden tener un resultado futuro parecido.
- **Criterios de aceptación:**
  - Se obtiene el resultado de un análisis *cluster* que muestra por colores los grupos de empresas con comportamientos similares en su serie temporal en términos de rentabilidad.

- **Historia de usuario 14:**
  - **Como** usuario
  - **Quiero** disponer de un botón de ayuda
  - **Para** poder consultar en cualquier momento la información sobre los distintos componentes y secciones de la web.
  - **Criterios de aceptación:**
    - Se muestra un panel que describe de manera breve las distintas funcionalidades de la web.

- **Historia de usuario 15:**
  - **Como** usuario
  - **Quiero** hacer zoom en un gráfico
  - **Para** poder visualizar mejor una parte del mismo.
  - **Criterios de aceptación:**
    - Se puede hacer *zoom in* y *zoom out* con distintas teclas o botones.

- **Historia de usuario 16:**
  - **Como** usuario
  - **Quiero** descargar un gráfico
  - **Para** poder trabajar con él fuera de la web.
  - **Criterios de aceptación:**
    - Se puede obtener una imagen `png` del gráfico que se descarga tras hacer click en un botón.

5.3 Elicitación de requisitos

La descripción de los requisitos del sistema surge del desarrollo detallado de los objetivos planteados en la sección 1.3 y de las historias de usuario de la sección 5.2. La división de los mismos se realizará en primer lugar en cuanto a la funcionalidad de la web a la que hacen referencia y, dentro de cada una de estas divisiones, se distinguirá entre requisitos funcionales y requisitos no funcionales. Los primeros de ellos harán referencia al planteamiento inicial general del trabajo, mientras que los de las distintas secciones irán directamente referidos a aquello que se desea conseguir en ellas.

5.3.1 Requisitos generales

Requisitos funcionales

- **RF101:** el sistema debe mostrar una descripción de cada una de las secciones que contiene.
- **RF102:** el sistema debe mostrar la información de contacto del autor del proyecto en cada una de las secciones que contiene.
- **RF103:** el sistema debe permitir al usuario navegar libremente entre todas las secciones de la web, pudiendo ir desde cada una de ellas a todas las demás.
- **RF104:** El sistema debe permitir al usuario hacer zoom en todos los gráficos que aparezcan en la web.
- **RF105:** El sistema debe permitir al usuario descargar todos los gráficos que aparezcan en la web como `png`.
- **RF106:** El sistema debe permitir al usuario seleccionar la temporalidad de los valores con los que desea trabajar en todos los apartados en los que se le permita seleccionar datos.
- **RF107:** El sistema debe indicar que un elemento gráfico está en proceso de carga mientras algún modelo o cálculo está realizándose.

Requisitos no funcionales

- **RNF101:** El sistema debe ser un entorno web.
- **RFN102:** El modo de indicar que un elemento está cargando será un icono dinámico de carga.

5.3.2 Inicio

Requisitos funcionales

- **RF201:** El sistema debe mostrar la autoría del trabajo.
- **RF202:** El sistema debe mostrar un resumen de las principales funcionalidades y contenidos de la web.

Requisitos no funcionales

- **RNF201:** La información de la autoría aparecerá en forma de título e irá acompañada de la imagen de la Universidad de Valladolid.
- **RNF202:** El resumen de las funcionalidades y contenidos de la web se mostrará en forma de texto.

5.3.3 Dashboard de precios

Requisitos funcionales

- **RF301:** El sistema debe mostrar información de los principales valores de cada uno de los tipos de activos considerados.
- **RF302:** El sistema debe mostrar la cotización de cada valor en el último día.
- **RF303:** El sistema debe mostrar el valor del cambio porcentual en el valor de cada valor en el último día.

- **RF304:** El sistema debe permitir al usuario seleccionar uno de los mercados disponibles en el menú lateral y mostrar la pantalla correspondiente a dicho mercado.
- **RF305:** El sistema debe mostrar una breve descripción del mercado en la pantalla de cada uno de los mismos.
- **RF306:** El sistema debe permitir al usuario seleccionar un valor en la pantalla de cada mercado y mostrar el gráfico de su serie temporal.

Requisitos no funcionales

- **RNF301:** Los porcentajes de subida o bajada de un valor con respecto al día anterior se mostrarán en rojo si el valor es una bajada y en verde si es una subida.
- **RNF302:** Los gráficos de las series de valores serán gráficos de líneas que permitirán la interacción del usuario con el ratón, mostrando el valor en un punto cuando este se pase por encima de la línea.

5.3.4 Análisis de valores

Requisitos funcionales

- **RF401:** El sistema debe permitir al usuario seleccionar todos los valores con los que quiere trabajar, de entre los valores disponibles en la aplicación.
- **RF402:** El sistema debe permitir al usuario variar la periodicidad de los datos mostrados en los distintos gráficos mostrados en esta pantalla.
- **RF403:** El sistema debe permitir al usuario representar la serie temporal de valores de un activo.
- **RF404:** El sistema deber permitir al usuario superponer los gráficos de las series temporales de tantos activos como desee de entre los seleccionados.
- **RF405:** El sistema debe permitir cambiar el valor que se muestra en todos los gráficos.
- **RF406:** El sistema debe permitir al usuario representar el gráfico de velas japonesas de un valor.
- **RF407:** El sistema deberá permitir al usuario mostrar el gráfico de barras del volumen acompañando al gráfico de velas japonesas.
- **RF408:** El sistema debe permitir al usuario representar el gráfico de barras de un valor.
- **RF409:** El sistema debe permitir al usuario representar la serie de rendimientos de un valor.
- **RF410:** El sistema debe permitir al usuario representar la serie de disminuciones de un valor.
- **RF411:** El sistema debe permitir al usuario representar el *boxplot* de rendimientos de un valor.
- **RF412:** El sistema debe permitir al usuario marcar y desmarcar los activos que desea representar en el gráfico de series temporales.
- **RF413:** El sistema deberá permitir al usuario representar la información de los distintos indicadores técnicos disponibles en la web en los gráficos de líneas.

Requisitos no funcionales

- **RNF401:** Toda la información se representará con el precio de cierre ajustado de los valores, salvo en el gráfico de líneas, donde se permitirá mostrar otras informaciones de acuerdo al requisito funcional del apartado anterior.
- **RNF402:** Cada gráfico de la serie temporal de un valor diferente se mostrará de un color distinto a los demás.
- **RNF403:** En los gráficos de velas, las velas que indiquen una subida de valor se rellenarán de color verde y las que indiquen una bajada de valor se rellenarán de color rojo.
- **RNF404:** En los gráficos de barras, cuando la barra indique una subida en el precio se dibujará de color verde y cuando indique una bajada de precio se dibujará de color rojo.
- **RNF405:** El gráfico de la serie temporal es el único que permite superponer los gráficos de varios activos.
- **RNF406:** Los gráficos irán acompañados de leyenda.
- **RNF407:** Todos los gráficos han de permitir al usuario la interacción con el ratón, mostrando o destacando los valores sobre los que mueva el mismo.

5.3.5 Optimización de carteras

Requisitos funcionales

- **RF501:** El sistema debe permitir al usuario seleccionar todos los valores con los que quiere trabajar, en caso de no haberlos seleccionado en la ventana de análisis de valores.
- **RF502:** El sistema debe permitir al usuario seleccionar el modelo de optimización con el que quiere trabajar, de entre los siguientes: modelo de maximización del rendimiento, modelo de minimización del riesgo, modelo bi-objetivo con maximización del rendimiento y minimización del riesgo (modelo de Markowitz).
- **RF503:** El sistema debe permitir al usuario seleccionar el tipo de rentabilidad con el que desea trabajar: media o media ponderada.
- **RF504:** El sistema debe permitir al usuario introducir la cantidad de dinero que desea invertir.
- **RF505:** El sistema debe permitir al usuario seleccionar una cota inferior de inversión en términos de porcentaje.
- **RF506:** El sistema debe permitir al usuario seleccionar una cota superior de inversión en términos de porcentaje.
- **RF507:** El sistema debe permitir al usuario seleccionar restricciones de cardinalidad que limiten el número de valores en los que invertir de entre el subconjunto de valores seleccionado.
- **RF508:** El sistema permitirá fijar el porcentaje de los datos seleccionados que se utilizarán para generar el modelo (cartera óptima), quedando el porcentaje restante reservado para la evaluación del mismo en la ventana de “Evaluación de cartera”.
- **RF509:** El sistema mostrará una tabla con la distribución de la cartera óptima generada por el modelo.

- **RF510:** En el caso del modelo de Markowitz el sistema permitirá al usuario representar la frontera eficiente rentabilidad-riesgo.

- **RF511:** En el caso del modelo de Markowitz el sistema permitirá al usuario representar las carteras notables (equiponderada, mínimo riesgo, máximo rendimiento y máximo ratio Sharpe).

- **RF512:** En el caso del modelo de Markowitz el sistema permitirá al usuario variar el valor del parámetro mu utilizado para variar el peso entre rendimiento y rentabilidad en la construcción de la frontera eficiente.

- **RF513:** El sistema debe permitir al usuario representar gráficamente la composición de las carteras que constituyen la frontera eficiente.

- **RF514:** El sistema debe permitir al usuario comparar las distintas carteras notables.

- **RF515:** En el caso del modelo básico (maximización de rentabilidad o minimización del riesgo), el sistema permitirá al usuario representar de manera tabulada y de manera gráfica la composición de la cartera óptima.

Requisitos no funcionales

- **RNF501:** La representación de las carteras notables se realiza mediante gráficos de barras, gráficos de sectores y *radar chart*, permitiendo al usuario seleccionar la representación que desee.

- **RFN502:** La representación de las carteras notables también se realizará de manera tabulada mostrando las medidas de rendimiento, riesgo y ratio Sharpe de cada una de ellas.

- **RNF503:** La comparación de carteras notables se realizará mediante gráficos de barras que muestren la composición de cada cartera.

- **RNF504:** La cartera equiponderada únicamente se representará en caso de no haberse incluido la restricción de cardinalidad en el modelo de Markowitz.

- **RNF505:** Las representaciones gráficas irán acompañadas de una leyenda.

- **RNF506:** Todos los gráficos han de permitir al usuario la interacción con el ratón, mostrando o destacando los valores sobre los que mueva el mismo.

5.3.6 Requisitos de "Evaluación de carteras"

Requisitos funcionales

- **RF601:** El sistema no permitirá llevar a cabo la evaluación de una cartera si esta no ha sido previamente optimizada en la ventana adecuada para ello.

- **RF602:** El sistema debe permitir al usuario representar la rentabilidad obtenida con las carteras eficientes en el caso de estar trabajando con el modelo de Markowitz.

- **RF603:** El sistema debe permitir al usuario representar la rentabilidad obtenida con la cartera óptima en caso de estar trabajando con el modelo básico.

Requisitos no funcionales

- **RNF601:** Las representaciones de la rentabilidad de las carteras se realizarán mediante un diagrama de líneas.
- **RNF602:** Las representaciones gráficas irán acompañadas de una leyenda.
- **RNF603:** Todos los gráficos han de permitir al usuario la interacción con el ratón, mostrando o destacando los valores sobre los que mueva el mismo.

5.3.7 Requisitos de "Regresión de índices"

Requisitos funcionales

- **RF701:** El sistema permitirá al usuario seleccionar el índice bursátil con el que desea trabajar.
- **RF702:** El sistema permitirá al usuario seleccionar las empresas del índice con las que desea trabajar.
- **RF703:** El sistema permitirá al usuario construir una cartera con empresas de un mismo índice y trabajar con la cartera óptima obtenida tras realizar la optimización de la misma en caso de emplearse un modelo básico.
- **RF704:** El sistema permitirá al usuario construir una cartera con empresas de un mismo índice y trabajar con las carteras notables obtenidas tras realizar la optimización de la misma en caso de emplearse un modelo de Markowitz.
- **RF705:** El sistema mostrará gráficamente el resultado de la regresión de la cartera o empresa elegida frente al índice bursátil.

Requisitos no funcionales

- **RNF701:** La representación de las regresiones se realizará mediante un gráfico de puntos sobre el que se superpondrá la recta de regresión calculada.
- **RNF702:** La representación de las regresiones irá acompañada del coeficiente R2 de la misma.
- **RNF703:** La representación gráfica ha de permitir al usuario la interacción con el ratón, mostrando o destacando los valores sobre los que mueva el mismo.

5.3.8 Requisitos de "Simulación de carteras"

Requisitos funcionales

- **RF801:** El sistema debe permitir al usuario seleccionar todos los valores con los que quiere trabajar.
- **RF802:** El sistema mostrará gráficamente el resultado de la simulación realizada.
- **RF803:** El sistema permitirá al usuario establecer el número de carteras que desea simular.

Requisitos no funcionales

- **RNF801:** La representación gráfica de las carteras simuladas se realizará en términos de la rentabilidad y el riesgo de las carteras, con un gradiente de color que variará en función del *ratio Sharpe* de la misma.

- **RNF802:** La representación gráfica se acompañará de una leyenda que ayude a comprender el gradiente de colores empleado.
- **RNF803:** La representación gráfica ha de permitir al usuario la interacción con el ratón, mostrando o destacando los valores sobre los que mueva el mismo.

5.3.9 Requisitos de "Modelos de predicción"

Requisitos funcionales

- **RF901:** El sistema debe permitir al usuario seleccionar el valor sobre el que desea realizar la predicción.
- **RF902:** El sistema debe permitir al usuario seleccionar la periodicidad de los datos con los que desea realizar la predicción.
- **RF903:** El sistema debe permitir al usuario seleccionar las variables con las que desea trabajar en los modelos que admitan más variables que la que se desea predecir (precio de cierre ajustado).
- **RF904:** El sistema debe permitir al usuario seleccionar los modelos con los que desee trabajar en este apartado.
- **RF905:** El sistema debe permitir al usuario seleccionar la configuración de parámetros que desee en cada uno de los modelos que quiera utilizar.
- **RF906:** El sistema debe permitir al usuario seleccionar las métricas con las que desea evaluar los modelos que desee utilizar.
- **RF907:** El sistema debe permitir al usuario fijar el porcentaje de datos que desea utilizar para entrenamiento y para prueba en la construcción y evaluación de los modelos seleccionados.
- **RF908:** El sistema debe permitir al usuario representar los modelos construidos.
- **RF909:** El sistema debe permitir al usuario representar la predicción de los modelos construidos para el siguiente periodo temporal.
- **RF910:** El sistema debe permitir al usuario superponer los modelos construidos con la representación de la serie temporal del valor seleccionado.
- **RF911:** El sistema debe permitir al usuario representar la información resumida de cómo está constituido cada modelo.
- **RF912:** El sistema debe permitir al usuario representar la información relativa a la evaluación de los modelos construidos.

Requisitos no funcionales

- **RNF901:** El usuario únicamente podrá seleccionar un valor para realizar sobre él la predicción.
- **RNF902:** El sistema obligará a que el precio de cierre ajustado sea una de las variables que el usuario ha de seleccionar.
- **RNF903:** Las representaciones de los modelos y de las series temporales de valores se realizarán mediante gráficos de líneas.

- **RNF904:** El valor predicho por cada modelo para el siguiente intervalo temporal se mostrará junto al gráfico.
- **RNF905:** La información de los modelos construidos se mostrará mediante cadenas de texto, con una entrada para cada modelo construido.
- **RNF906:** La información de las métricas de evaluación de los modelos construidos se mostrará de forma tabulada.
- **RNF907:** Las predicciones de los modelos para el siguiente periodo temporal se mostrarán en forma de texto.
- **RNF908:** Las representaciones gráficas irán acompañadas de una leyenda.
- **RNF909:** Todos los gráficos han de permitir al usuario la interacción con el ratón, mostrando o destacando los valores sobre los que mueva el mismo.

5.3.10 Requisitos de "Análisis de grupos"

Requisitos funcionales

- **RF1001:** El sistema debe permitir al usuario seleccionar la periodicidad de los datos con los que desea realizar la predicción.
- **RF1002:** El sistema debe permitir seleccionar el número de *clusters* a buscar.
- **RF1003:** El sistema debe permitir seleccionar el número máximo de iteraciones del algoritmo de K medias.
- **RF1004:** El sistema debe permitir mostrar o no los centroides utilizados por el algoritmo de K medias.
- **RF1005:** El sistema debe permitir al usuario representar los valores elegidos indicando en qué *cluster* está asignado cada uno.

Requisitos no funcionales

- **RNF1001:** La representación de los valores se realizará mediante un *scatter plot*.
- **RNF1002:** La forma de indicar a qué *cluster* pertenece un valor será mediante el color, quedando cada *cluster* identificado con un color diferente.
- **RNF1003:** La representación gráfica se acompañará de una leyenda que ayude a comprender el gradiente de colores empleado.
- **RNF1004:** La representación gráfica ha de permitir al usuario la interacción con el ratón, mostrando o destacando los valores sobre los que mueva el mismo.

5.4 Casos de uso

El único actor del sistema es el usuario que lo está utilizando y los principales casos de uso que puede llevar a cabo son los siguientes:

- **Consultar información de un mercado:** el actor podrá consultar la descripción de un mercado, así como las cotizaciones de los valores que lo componen y las series temporales de los mismos.

- **Configurar gráfico analítico:** el actor podrá seleccionar un valor, establecer la fecha inicial y final de los datos que desea visualizar y establecer la periodicidad de los mismos.
- **Visualizar gráfico analítico:** el actor podrá visualizar un gráfico en la ventana de análisis de valores.
- **Visualizar indicadores técnicos:** el actor podrá visualizar el gráfico de los indicadores técnicos acompañando al gráfico de la serie temporal de un único valor.
- **Configurar modelo de optimización:** el actor podrá seleccionar un conjunto de valores sobre el que construir una cartera, elegir el tipo de modelo que desea optimizar a la hora de realizar la optimización de la misma, así como elegir el tipo de rentabilidad con el que desea trabajar, establecer la cantidad de inversión, las cotas inferior y superior de inversión en cada valor y establecer el porcentaje de datos dedicados a entrenamiento y evaluación de la cartera.
- **Visualizar resultados de modelo de optimización:** el actor podrá ver distintas salidas den forma de gráficas y tablas con los resultados del modelo de optimización entrenado, siendo estas distintas si se trata de un modelo básico o si se trata del modelo bi-objetivo de Markowitz.
- **Visualizar variaciones en función del parámetro μ:** el actor podrá variar el valor del parámetro μ utilizado en la construcción del modelo de Markowitz, modificándose el gráfico resultante de manera reactiva.
- **Evaluar cartera:** el actor podrá evaluar la cartera óptima obtenida en términos de rendimiento.
- **Configurar modelo de predicción:** el actor podrá seleccionar un valor y un modelo de predicción, establecer los parámetros con los que desea construirlo y especificar las variables que desea incluir como *inputs*, las métricas que desea utilizar para evaluar el ajuste proporcionado y establecer el porcentaje de datos destinados a entrenamiento y evaluación.
- **Visualizar modelo de predicción:** el actor podrá visualizar un modelo de predicción construido, superponiéndose la serie temporal de los valores ajustados por el mismo con los datos originales. También podrá visualizar la evaluación del modelo, así como la predicción realizada para el periodo siguiente por el modelo de predicción construido.
- **Comparar modelos de predicción:** el actor podrá visualizar una tabla comparativa con las métricas obtenidas por los distintos modelos de predicción obtenidos.
- **Simular carteras:** el actor podrá simular un número determinado de carteras mediante el método de Montecarlo.
- **Ver la relación entre una cartera y un índice bursátil:** el actor podrá ejecutar una regresión lineal simple de una cartera óptima construida sobre valores pertenecientes a un único índice bursátil, frente a dicho índice.
- **Ver grupos de activos en términos de rentabilidad:** el actor podrá ejecutar un análisis *cluster* con el algoritmo de K medias para obtener agrupaciones de activos que tengan un comportamiento similar en términos de rentabilidad.

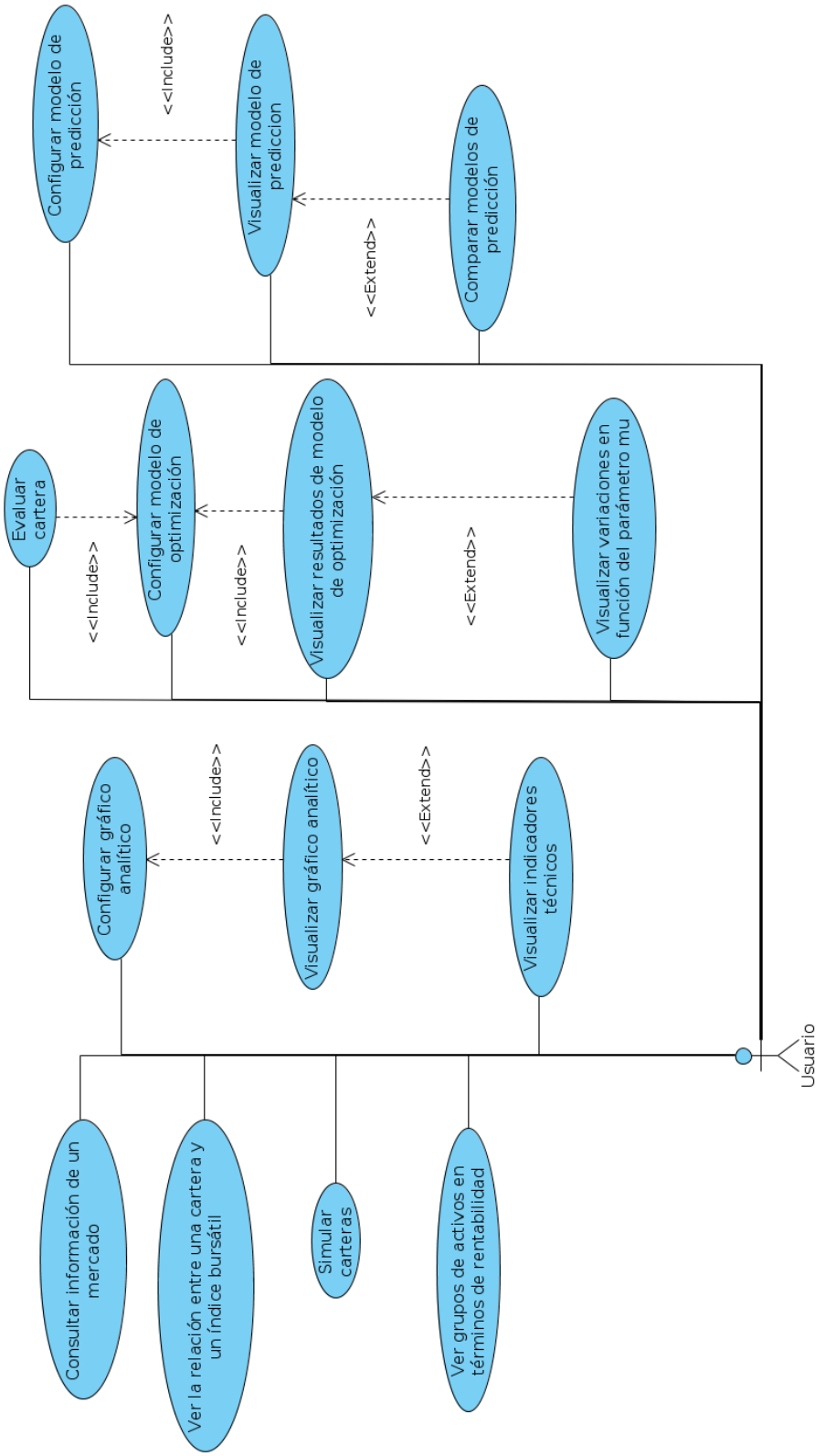


Figura 5.2: Diagrama de casos de uso. Realizado con 7.3.1.

Capítulo 6

Diseño

6.1 Arquitectura del sistema

Para representar la arquitectura del sistema se muestra un **diagrama de despliegue** [96], tipo de diagrama **UML** frecuentemente utilizado para ilustrar los componentes *hardware* y *software* de un sistema, así como el modo de relacionarse de estos elementos.

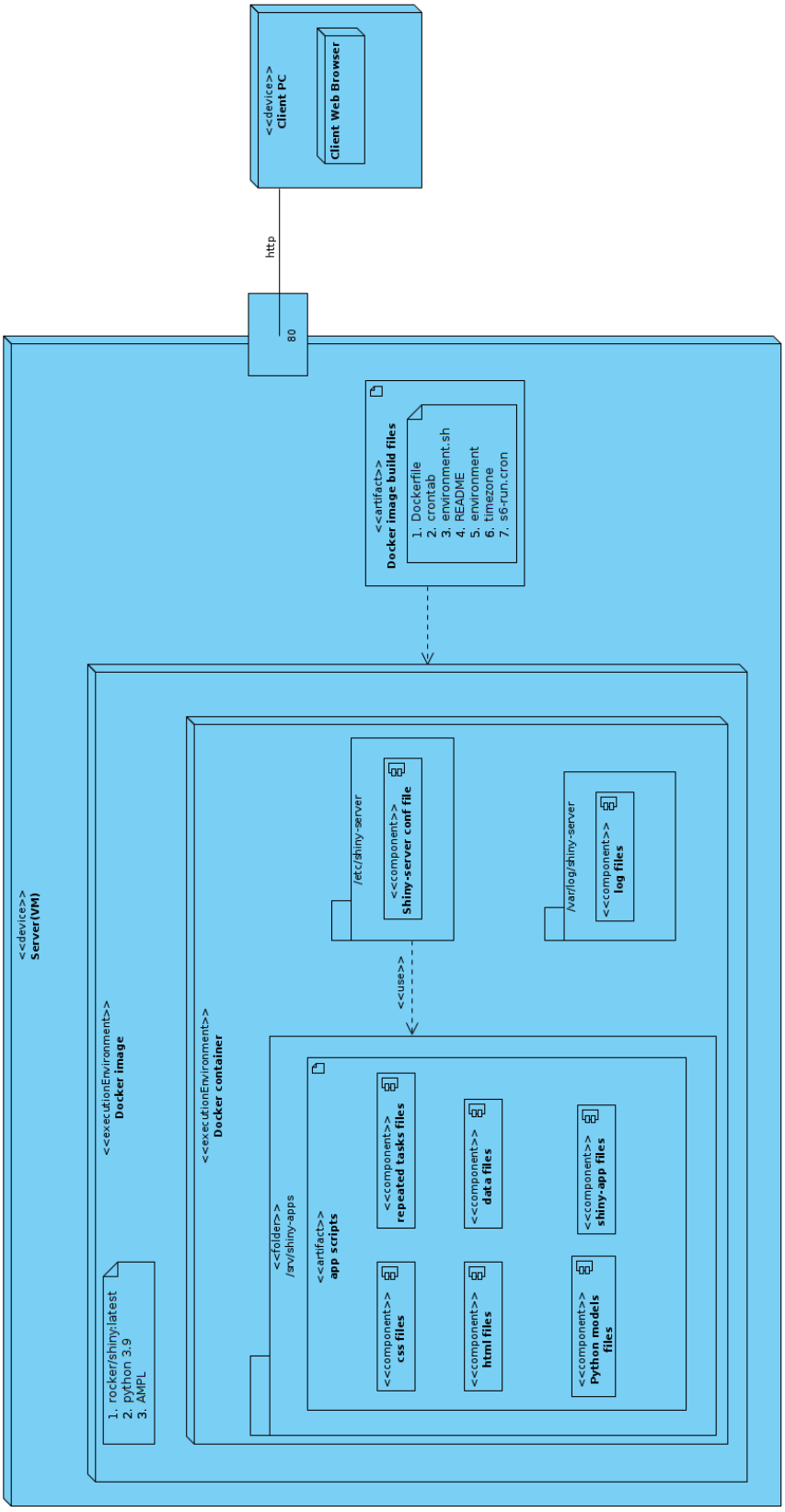


Figura 6.1: Diagrama de despliegue. Realizado con 7.3.1.

6.2 Diseño de casos de uso

Los casos de uso presentados en 5.4, se representan en diseño mediante **diagramas de secuencia** UML, donde se muestran las líneas de vida de los elementos y procesos, así como el intercambio de mensajes entre los mismos. En esta sección se comentan los más importantes, quedando los diagramas de secuencia incluidos en B.2.

Caso de uso Configurar gráfico analítico

- **Precondición:** El sistema muestra el menú lateral, desplegado o no, donde se permite seleccionar la opción de "análisis de valores".
- **Postcondición:** Se muestra la visualización gráfica configurada en el elemento shiny central de la página.
- **Camino principal:**
  1. El usuario selecciona "análisis de valores".
  2. El sistema carga la ventana de "análisis de valores", muestra las opciones de configuración y la visualización gráfica por defecto.
  3. El usuario selecciona el gráfico de series temporales.
  4. Ir al caso de uso "Visualizar gráfico analítico".
  5. El usuario selecciona un activo.
  6. Ir al caso de uso "Visualizar gráfico analítico".
  7. El usuario selecciona un intervalo temporal.
  8. Ir al caso de uso "Visualizar gráfico analítico".
  9. El usuario selecciona la periodicidad de los datos que desea visualizar.
  10. Ir al caso de uso "Visualizar gráfico analítico".
- **Caminos alternativos:**
  - 3a. El usuario selecciona el gráfico de "velas japonesas".
    - El sistema muestra el gráfico de velas japonesas del activo seleccionado.
    - Ir al paso 5.
  - 3b. El usuario selecciona el gráfico de "barras".
    - El sistema muestra el gráfico de barras del activo seleccionado.
    - Ir al paso 5.
  - 3c. El usuario selecciona el gráfico de "serie de rendimientos".
    - El sistema muestra el gráfico de la serie de rendimientos del activo seleccionado.
    - Ir al paso 5.
  - 3d. El usuario selecciona el gráfico de "serie de disminuciones".
    - El sistema muestra el gráfico de la serie de disminuciones del activo seleccionado.
  - 3e. El usuario selecciona el gráfico de "boxplot de rendimientos".
    - El sistema muestra el gráfico del boxplot de rendimientos del activo seleccionado.
    - Ir al paso 5.
  - 5a. El usuario selecciona otro activo.
    - Ir al paso 4.
  - 7a. El usuario selecciona un indicador técnico.
    - Ir al paso 4.

Caso de uso Configurar modelo de optimización

- **Pre-condición:** El sistema muestra el menú lateral, desplegado o no, donde se permite seleccionar la opción de “optimización de carteras”.
- **Post-condición:** Se muestran de manera gráfica los resultados de modelo de optimización de carteras construido.
- **Camino principal:**
  1. El usuario selecciona “optimización de carteras”.
  2. El sistema carga la ventana de “optimización de carteras”, con la configuración por defecto.
  3. El usuario selecciona el porcentaje de datos que desea utilizar para entrenamiento del modelo de optimización.
  4. El sistema calcula el modelo de optimización con la configuración establecida.
  5. El sistema guarda los datos de evaluación del modelo.
  6. Ir al caso de uso “Visualizar resultados de modelo de optimización”.
  7. El usuario selecciona el modelo biobjetivo de Markowitz e introduce un tipo de interés de referencia para el cálculo del ratio sharpe.
  8. Ir al paso 4.
  9. El usuario selecciona el modelo maximización de la rente e introduce un riesgo máximo.
  10. Ir al paso 4.
  11. El usuario selecciona el modelo de minimización del riesgo e introduce una rentabilidad mínima.
  12. Ir al paso 4.
  13. El usuario selecciona un conjunto de activos.
  14. Ir al paso 4.
  15. El usuario establece el intervalo temporal que desea para sus datos.
  16. Ir al paso 4.
  17. El usuario establece el tipo de rentabilidad que desea utilizar.
  18. Ir al paso 4.
  19. El usuario selecciona el capital que desea invertir.
  20. Ir al paso 4.
  21. El usuario selecciona el porcentaje mínimo que se ha de invertir en cada activo de la cartera.
  22. Ir al paso 4.
  23. El usuario selecciona el porcentaje máximo que se ha de invertir en cada activo de la cartera.
  24. Ir al paso 4.
  25. El usuario selecciona el mínimo de activos entre los cuales desea diversificar la cartera.
  26. Ir al paso 4.
- **Caminos alternativos:**
  - 7a. El usuario selecciona el modelo biobjetivo de Markowitz, introduce un tipo de interés de referencia para el cálculo del ratio sharpe y un valor para el parámetro mu.
    - Ir al caso de uso “Visualizar variaciones en función del parámetro mu”.

Caso de uso Configurar modelo de predicción

- **Precondición:** El sistema muestra el menú lateral, desplegado o no, donde se permite seleccionar la opción de "modelos de predicción".
- **Postcondición:** Se muestran de manera gráfica los resultados de modelo de predicción de valores construido.
- **Camino principal:**
  1. El usuario selecciona "modelos de predicción".
  2. El sistema carga la ventana de "modelos de predicción", con la configuración por defecto.
  3. El usuario selecciona el activo que desea utilizar para construir el modelo de predicción.
  4. El usuario selecciona el intervalo temporal que desea para sus datos.
  5. El usuario selecciona la periodicidad de los datos que desea utilizar.
  6. El usuario selecciona el porcentaje de datos que desea utilizar para entrenamiento.
  7. El usuario selecciona el conjunto de métricas que desea utilizar para evaluar los modelos que construya.
  8. El usuario selecciona el modelo ARIMA.
  9. El sistema muestra las opciones de configuración del modelo ARIMA.
  10. El usuario configura el modelo ARIMA y selecciona "Ejecutar modelo".
  11. El sistema genera el modelo con la configuración establecida.
  12. Ir al caso de uso "Visualizar modelo de predicción".
- **Caminos alternativos:**
  - 8a. El usuario selecciona el modelo Random Forest.
    - El sistema muestra las opciones de configuración del modelo Random Forest.
    - El usuario configura el modelo Random Forest y selecciona "Ejecutar modelo".
    - Ir al paso 11.
  - 8b. El usuario selecciona el modelo XGBoost.
    - El sistema muestra las opciones de configuración del modelo XGBoost.
    - El usuario configura el modelo XGBoost y selecciona "Ejecutar modelo".
    - Ir al paso 11.
  - 8a. El usuario selecciona el modelo SVR.
    - El sistema muestra las opciones de configuración del modelo SVR.
    - El usuario configura el modelo SVR y selecciona "Ejecutar modelo".
    - Ir al paso 11.
  - 8a. El usuario selecciona el modelo MLP.
    - El sistema muestra las opciones de configuración del modelo MLP.
    - El usuario configura el modelo MLP y selecciona "Ejecutar modelo".
    - Ir al paso 11.
    - El sistema muestra las opciones de configuración del modelo Simple RNN.
    - El usuario configura el modelo Simple RNN y selecciona "Ejecutar modelo".

- Ir al paso 11.

- El sistema muestra las opciones de configuración del modelo LSTM.
- El usuario configura el modelo LSTM y selecciona “Ejecutar modelo”.
- Ir al paso 11.

- El sistema muestra las opciones de configuración del modelo GRU.
- El usuario configura el modelo GRU y selecciona “Ejecutar modelo”.
- Ir al paso 11.

- El sistema muestra las opciones de configuración del modelo CNN.
- El usuario configura el modelo CNN y selecciona “Ejecutar modelo”.
- Ir al paso 11.

6.3 Diseño interfaz

En esta sección se presentan los elementos de diseño de la interfaz del sistema. Como se comentó en 2.4, el diseño de la interfaz ha estado guiado por *mockups*, construidos con la herramienta 7.2.2. Se presentan a su vez tanto los componentes web como los distintos elementos de las aplicaciones *shiny* 7.2.5.

6.3.1 Mockups

Se incluyen en esta sección únicamente los *mockups* finales de la aplicación. Los realizados en fases anteriores no se incluyen.

En primer lugar, se muestra la pantalla inicial de la aplicación 6.2, sirviendo de base a su vez para ilustrar el menú lateral y el despliegue del menú de mercados 6.3.

Todas las secciones tienen la misma estructura. Un ejemplo de una de ellas, que a su vez se utilizará para mostrar la interacción de estas secciones con el despliegue del menú lateral, se muestra en las Figuras 6.4 y 6.5.

Finalmente se muestra el mockup de la sección de ayuda 6.6.

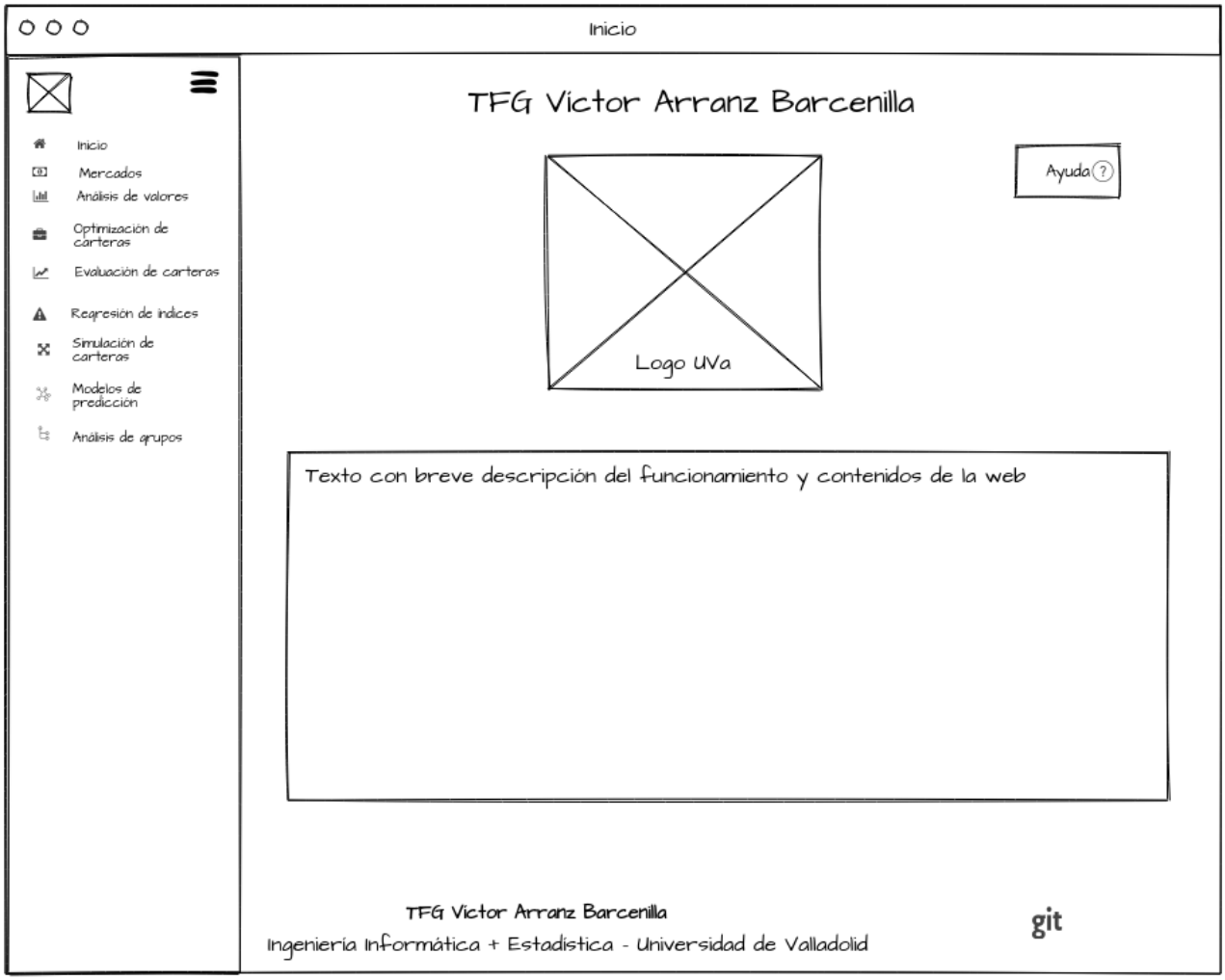


Figura 6.2: Mockup de la pantalla inicial.

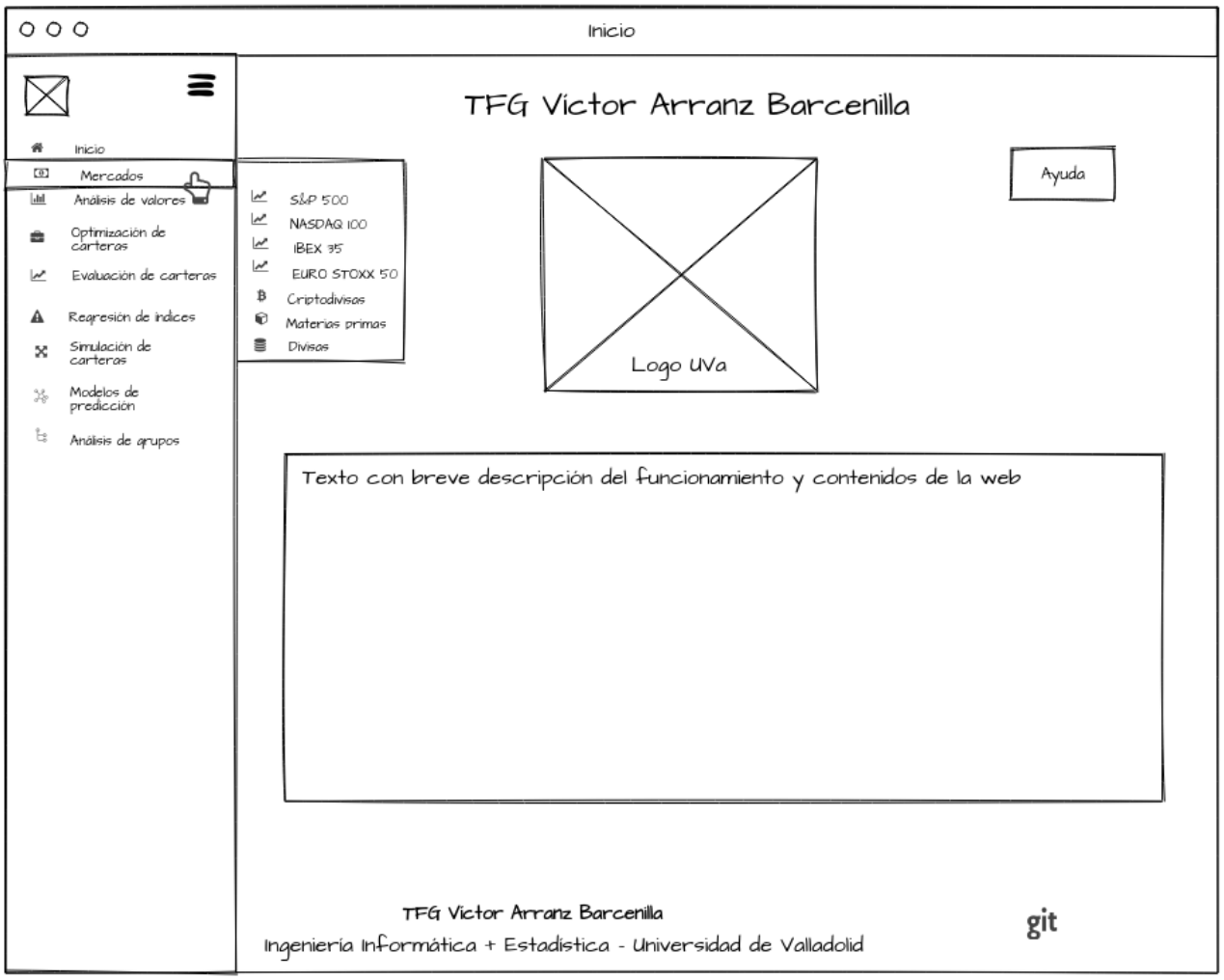


Figura 6.3: Mockup de la pantalla inicial con el menú de mercados desplegado.

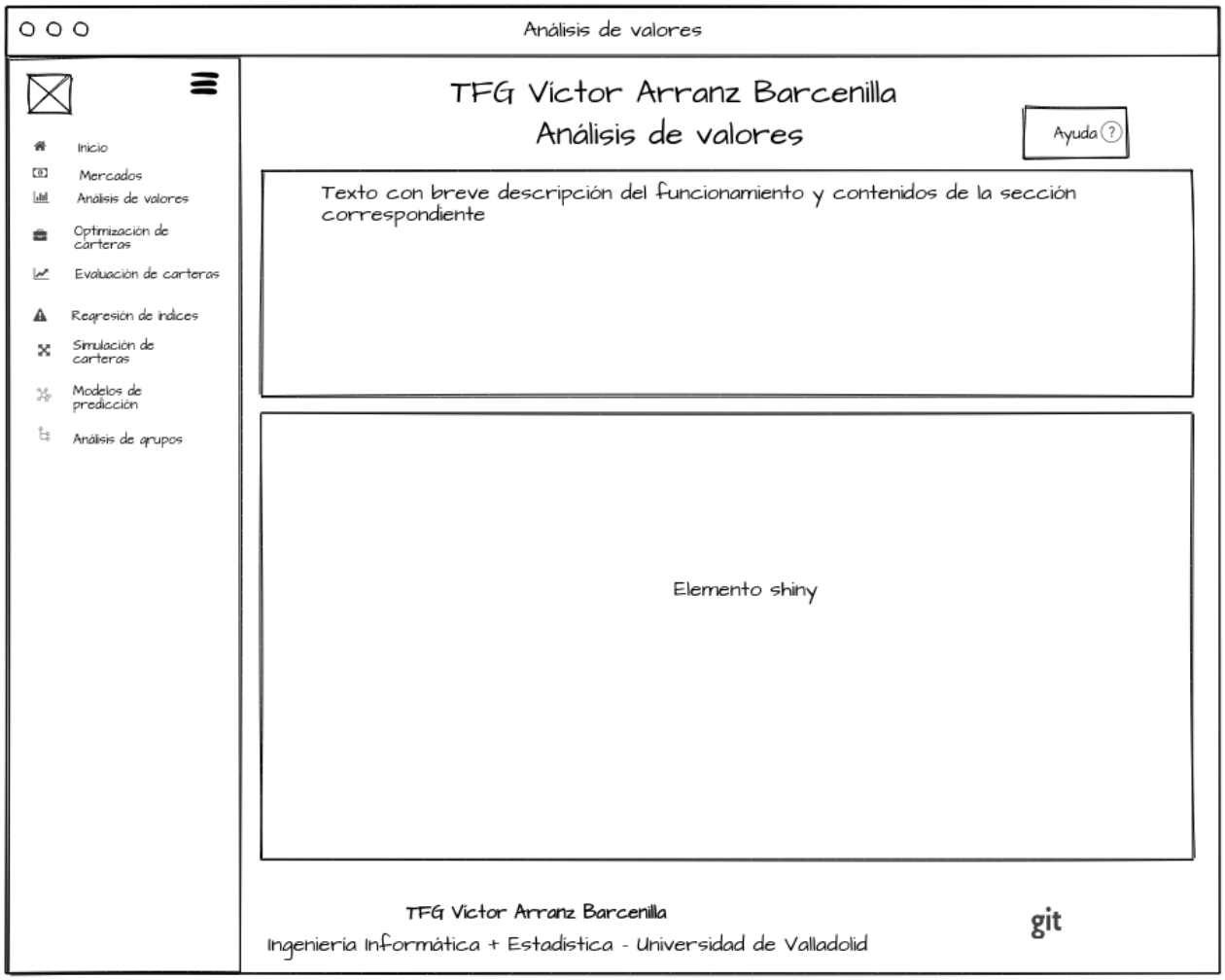


Figura 6.4: Mockup de la pantalla de análisis de valores con el menú lateral desplegado.

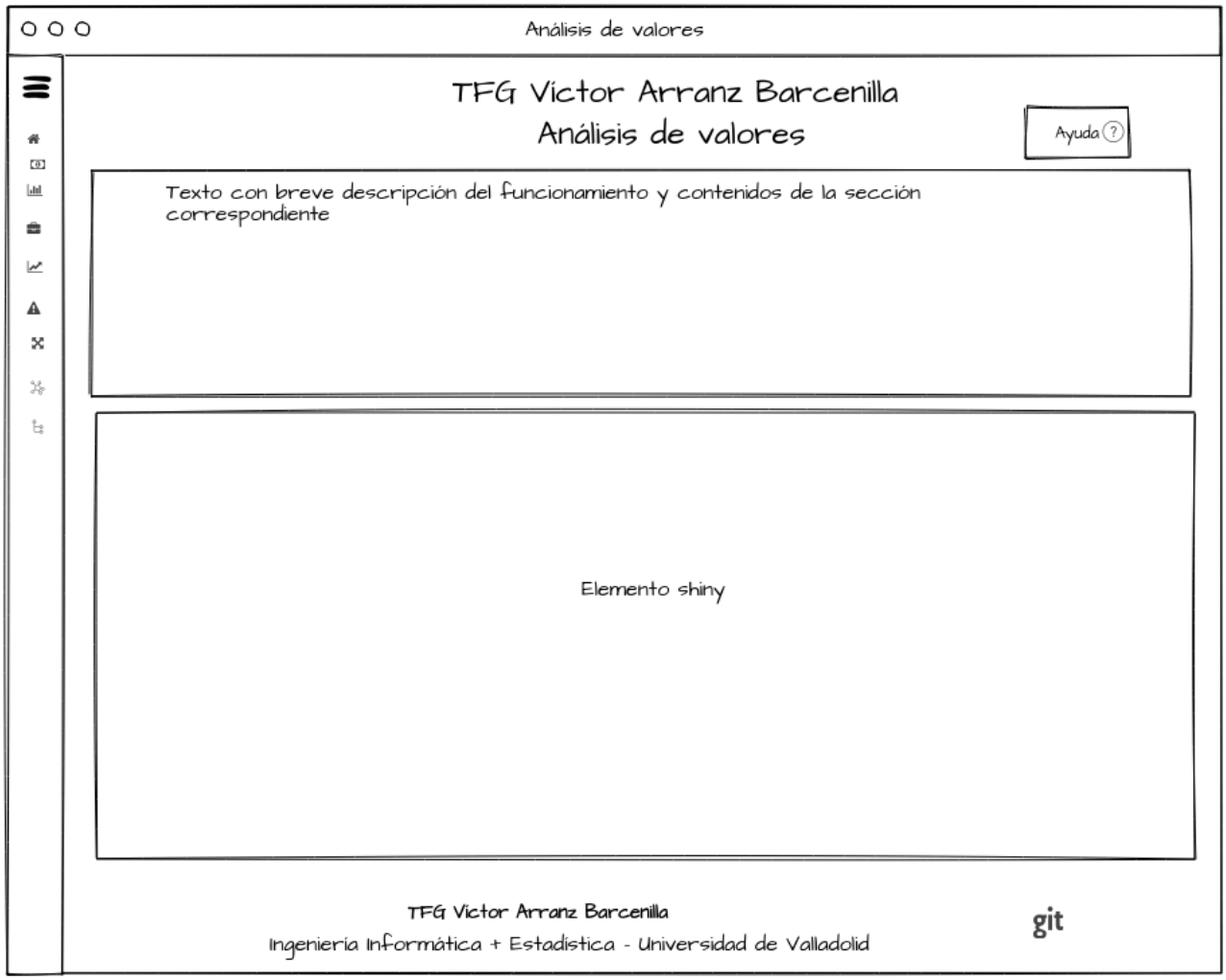


Figura 6.5: Mockup de la pantalla de análisis de valores con el menú lateral oculto.

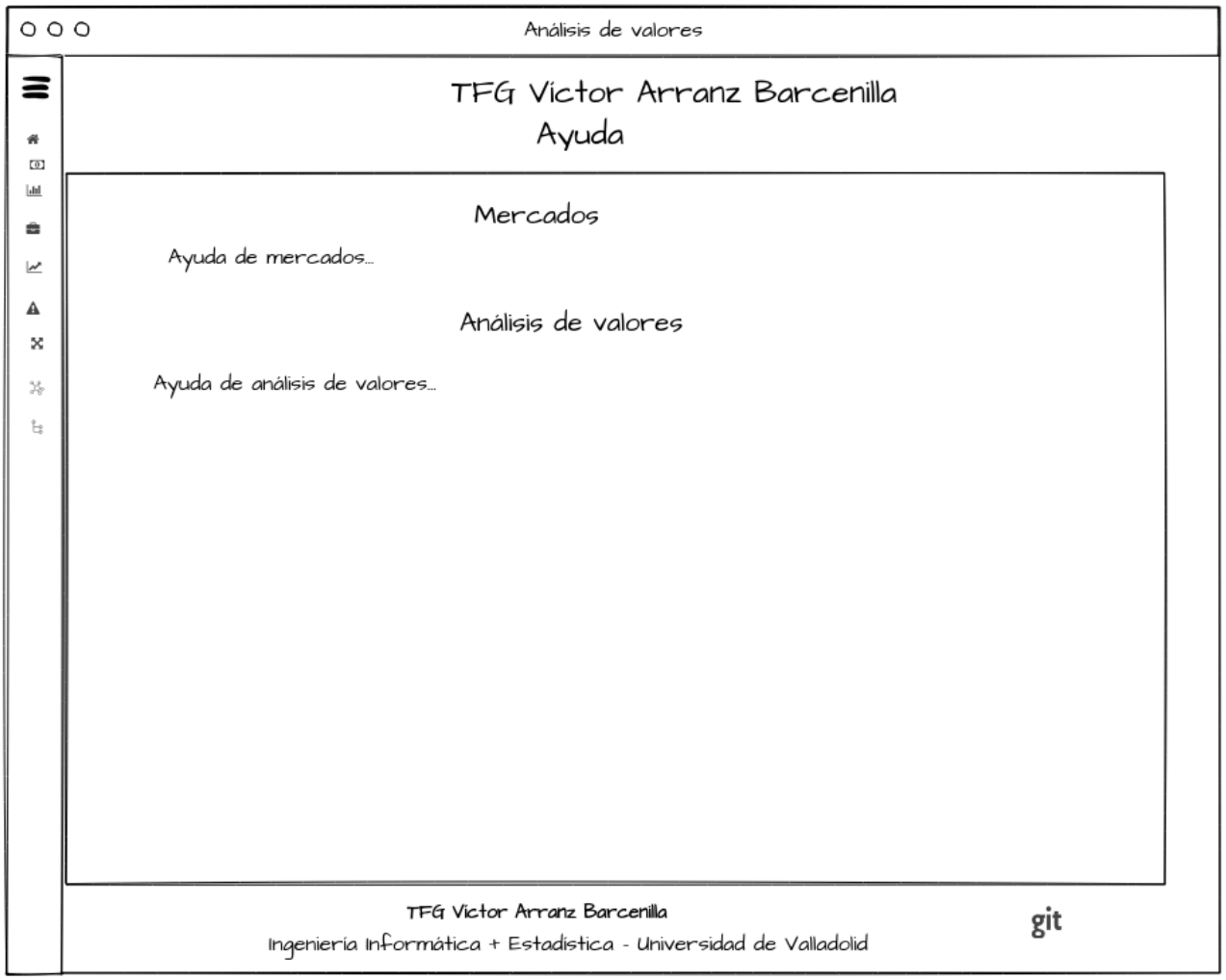


Figura 6.6: Mockup de la pantalla de ayuda.

6.3.2 Elementos interfaz

Landing page

La *landing page* o página de aterrizaje es la página que se muestra a los usuarios cuando acceden a una aplicación o sitio web. Suele haber una *landing page* general que se muestra inicialmente sin necesidad de registro y varias *landing pages* diferentes que se muestran para cada tipo diferente de usuario en las páginas web que requieren de registro. Estas páginas suelen tener pocos elementos y *links* y tienen el objetivo de captar la atención del usuario y hacer que permanezca en la aplicación o que se registre.

En el presente proyecto, la página de aterrizaje contiene un contenedor central que informa de las funcionalidades proporcionadas por el sitio web y de la autoría del proyecto, además de contar con el menú lateral de navegación que estará presente en todas las ventanas. Del mismo modo se muestra el pie de página común a toda la web, donde en la parte derecha de la pantalla se encuentra el enlace al repositorio de *gitlab* (ver 7.4.3) que permite acceder al proyecto en una nueva pestaña y en la parte central se muestra la autoría del proyecto. En la parte superior derecha aparece el botón de ayuda que estará presente en todas las pantallas de la aplicación.

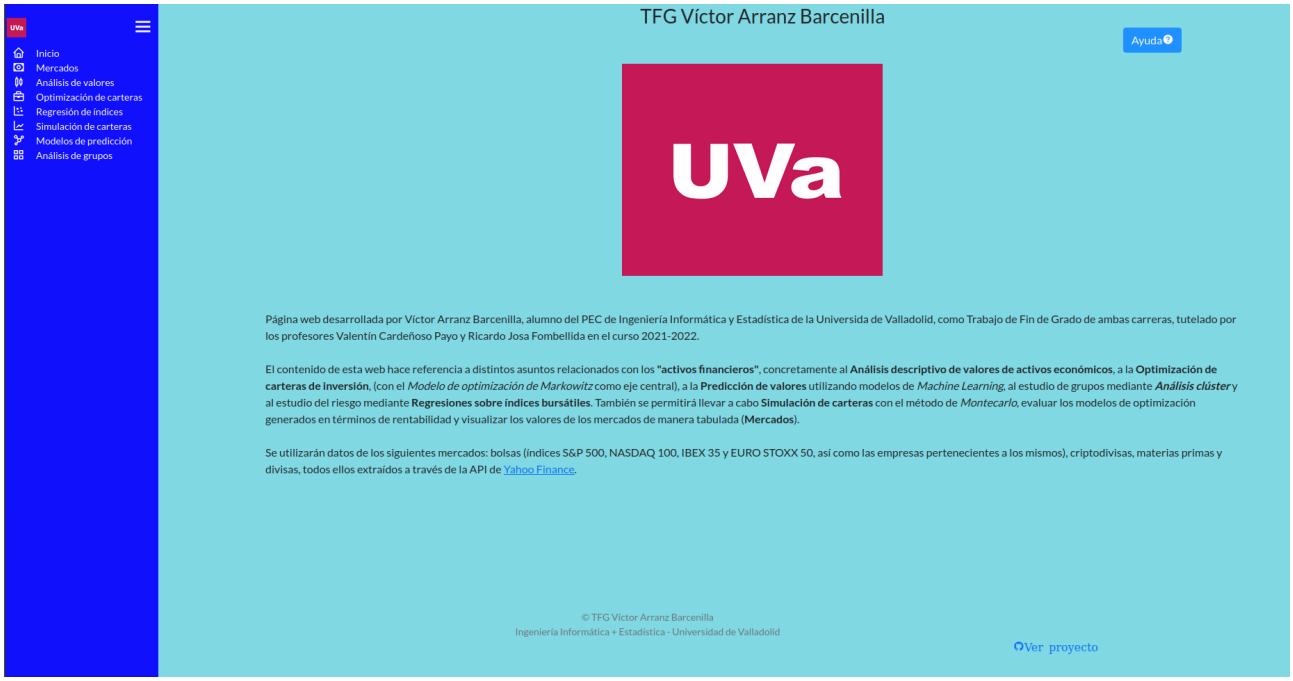


Figura 6.7: Página de aterrizaje.

Menú de navegación

El menú de navegación de un sitio web organiza y estructura los contenidos de la misma, dirigiendo a su vez el flujo del usuario por la aplicación.

El menú de navegación implementado en la web de este proyecto, puede desplegarse (ver Figura 6.7) o contraerse (ver Figura 6.8) pulsando sobre el botón de tres barras situado en la parte superior del mismo. Cuando está contraído, queda visible el logotipo de cada sección y el nombre únicamente se muestra al pasar el ratón por el logotipo correspondiente. Las opciones que muestra son siempre las mismas a excepción de cuando el usuario se haya ubicado en "optimización de carteras", ya que, después de llevar a cabo las acciones de esta, se le mostrará también la opción de "evaluar cartera". La opción de mercados tiene un submenú que se muestra cuando el usuario pasa el ratón por esa zona (ver Figura 6.9).

Figura 6.8: Menú de navegación plegado.

Figura 6.9: Menú de navegación con el submenú de mercados desplegado.

Capítulo 7

Implementación

7.1 Despliegue del servidor

El objetivo de este trabajo desde un primer momento fue la creación de una aplicación web (como se detalló en el apartado 1.3). Como se pretendió desde un primer momento poder acceder a ella desde cualquier dispositivo y lugar a través de internet, se decidió que lo mejor era desplegar el servicio web en una máquina virtual de la escuela.

La máquina solicitada fue una Ubuntu 20.04.3 LTS de 2 núcleos con memoria RAM de 8G y memoria de disco de 16G.

7.1.1 Docker

El sistema elegido para realizar el montaje del servidor web en la máquina virtual fue Docker [97]. Esta elección se debió a la facilidad que provee este sistema para crear, probar e implementar aplicaciones de un modo rápido. Docker permite llevar a cabo la virtualización de un sistema operativo, pudiéndose desplegar y manipular múltiples sistemas en una misma máquina. [98] Docker proporciona un nivel extra de abstracción y utiliza características de aislamiento de Linux como los grupos y los espacios de nombres para permitir que los distintos contenedores se ejecuten dentro de una única instancia de Linux. Estos contenedores son unidades que incluyen todo lo necesario para que un determinado software se ejecute (librerías, herramientas de sistema, código, etc) y constituyen el punto clave de Docker. Su propósito es ejecutar varios procesos de manera separada, de forma que todos mantengan las características de seguridad que tendrían ejecutándose como sistemas individuales y permitiendo así un mayor aprovechamiento de la infraestructura

Los sistemas Docker se componen de dos elementos, las imágenes y los contenedores:

- **Imágenes:** una imagen es la definición de un sistema operativo. Solamente ha de instalarse una vez y compartirla permite replicar el sistema Docker construido.
- **Contenedores:** son instancias de la imagen y han de ser cargados cada vez que requiramos una instancia. Se pueden ejecutar al mismo tiempo varios contenedores de una misma imagen y contienen todo lo necesario para que las aplicaciones puedan ejecutarse.

Ventajas de los contenedores Docker [99]:

- **Modularidad:** la separación en contenedores permite modificar una parte de una aplicación sin que el resto se vea alterado, pudiendo así actualizarla, repararla o ampliarla.

- **Control de versiones de imágenes y restauración:** Docker está basado en imágenes, lo que permite compartir un sistema o aplicación con todos sus elementos en cualquier entorno. A su vez, una imagen Docker está formada por varias capas de manera que al ejecutar un comando, la imagen se modifica con la creación de una nueva capa. Esto permite controlar los cambios realizados mediante el registro de los mismos en las imágenes y poder volver a versiones anteriores de un sistema en caso de requerirse.
- **Rapidez:** Docker permite establecer un servicio en pocos segundos. Además, cada proceso se encuentra en un contenedor distinto, lo que permite compartirlos con aplicaciones o sistemas nuevos. A su vez, como los cambios pueden realizarse a nivel de capa o contenedor, no es necesario volver a reiniciar o cargar el sistema operativo.

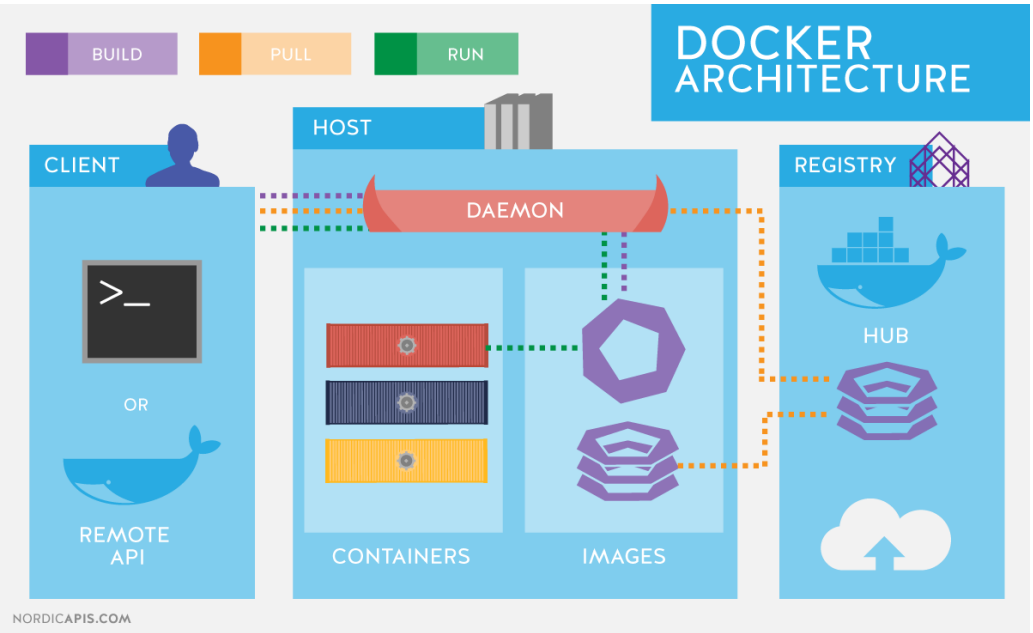


Figura 7.1: Arquitectura Docker. Tomado de [100].

Dockerfile

Toda aplicación Docker tiene un archivo de configuración denominado **Dockerfile** que contiene las instrucciones para construir una imagen. Este archivo puede constar de varias instrucciones, entre las que cabe destacar las siguientes[referencia]:

- **FROM:** describe qué imagen se está construyendo.
- **RUN:** comandos que imitan la línea de comandos.
- **COPY:** ficheros con el código que se desea ejecutar. Han de estar en el mismo directorio que el dockerfile.
- **CMD:** comando a ejecutar cada vez que es lanza el Docker.
- **WORKDIR:** establece el directorio de trabajo.

Otra instrucción interesante es `ARG WHEN`, que permite especificar la fecha que queremos que se use en los paquetes y librerías de la aplicación, independientemente de cuándo se utilice esta.

En el apéndice A se incluye la guía que describe los pasos seguidos en este trabajo con la utilización de docker para el despliegue del sistema.

7.2 Herramientas de Desarrollo

En esta sección se realizará una breve descripción de las distintas herramientas empleadas en el desarrollo de este proyecto, comentándose desde aquellas utilizadas para realizar la memoria, hasta las principales librerías utilizadas en los distintos *scripts* generados como entregables del proyecto.

7.2.1 Extracción de datos

La extracción de los datos utilizados en la web se realiza en tiempo real a través de la API de *Yahoo Finance* del modo comentado en 4. En dicha sección se comenta también la extracción de los nombres y símbolos de los valores mediante ***scraping***.

Parsehub

ParseHub [101] es una herramienta de *scraping* de escritorio muy poderosa y versátil. Tiene dos versiones: gratuita y de pago, siendo la gratuita la utilizada en el presente proyecto. Posee una interfaz muy intuitiva y fácil de utilizar, que se muestra en un tutorial que permanece disponible en la aplicación en todo momento. ParseHub permite descargar los datos que necesitemos de una web los en dos principales formatos que se utilizan en la actualidad: `CSV` y `JSON`. La extracción de datos es muy simple debido a que no hay que programar nada, lo que se desea descargar se selecciona haciendo *clicks* con el ratón del modo que explica el tutorial antes mencionado. Posee una API desde la que se puede acceder utilizando diversos lenguajes de programación y la versión de pago permite funcionalidades interesantes como el poder programar extracciones periódicas de datos de una web.

7.2.2 Diseño de la Interfaz

Mockflow

[102]*Mockflow* es una herramienta destina realización *online* de *mockups*, práctica muy habitual en el desarrollo de software que tiene la finalidad de mostrar visualmente cómo se desea que sea el resultado de una interfaz. Del mismo modo que los tradicionales bocetos realizados a papel, permite modificar los elementos incluidos hasta llegar a representar la apariencia final de un proyecto. *Mockflow* permite crear *wireframes* en la nube y es excelente para integrar rápidamente las herramientas por ejemplo con *Slack* y *Trello*. Tiene la posibilidad de utilizar parte de su funcionalidad de manera gratuita, lo cual ha sido suficiente para este trabajo. En este modo, la web limita el número de proyectos que se pueden crear y el número de páginas que pueden componerlos.

7.2.3 Frontend

HTML

[103] El Lenguaje de Marcado de Hipertexto es el lenguaje que se utiliza para desplegar una página web y sus contenidos. No es un lenguaje de programación como tal, sino que es un lenguaje de marcado que define la estructura de una página web. Un elemento HTML se distingue de otro texto en un documento mediante “etiquetas”, que consisten en el nombre del elemento rodeado por los caracteres “<” y “>”. El nombre de un

elemento dentro de una etiqueta no distingue entre mayúsculas y minúsculas y puede ser de distinto tipo: contenedores, imágenes, hipervínculos, etc. Los enlaces entre páginas web son un elemento fundamental de estos sistemas y el término "hipertexto" de este lenguaje hace referencia a ellos.

Bootstrap

[104] Es un *framework* de código abierto utilizado en el diseño de aplicaciones web. Contiene plantillas para diversos elementos como botones, contenedores o menús de navegación. Se encarga únicamente de diseño *frontend*, es decir, únicamente de la interfaz de usuario, en la capa de presentación. Utiliza elementos CSS, HTML y Javascript sencillos y es un proyecto de *GitHub* que utilizan empresas como la *NASA*.

Javascript

[105] Es un lenguaje de programación interpretado, que se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. También se puede utilizar en el lado del servidor. También tiene utilidad en aplicaciones externas a la web.

jQuery

[106] Es una librería de Javascript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción. Esta librería permite a los desarrolladores crear abstracciones para interacción y animación de bajo nivel, efectos avanzados y *widgets* temáticos de alto nivel. A su vez, facilita la creación de páginas web dinámicas y aplicaciones web.

7.2.4 CSS

[107] El lenguaje de "Hojas de Estilo en Cascada" es un lenguaje utilizado para definir la presentación y aspecto de un documento escrito con un lenguaje de marcado como HTML. Es el lenguaje más habitualmente utilizado para crear el aspecto visual de las páginas web y de las interfaces escritas en cualquier tipo de documento XML.

7.2.5 Backend

R

R es un lenguaje de programación especialmente enfocado al análisis estadístico, el análisis de datos y la ciencia de datos, con múltiples funcionalidades que permiten realizar todo tipo de test estadísticos y crear multitud de modelos, integrando una enorme variedad de algoritmos útiles en diversas áreas de la estadística. Es un lenguaje de software libre que surgió como una reinterpretación del lenguaje S. Su gran variedad de librerías destinadas al cálculo y la visualización hacen que sea el lenguaje más popular en el ámbito científico, con gran popularidad en campos como el *machine learning* , el *big data*, la bioinformática, la biomedicina o la minería de datos entre otros.

R puede ser utilizado desde línea de comandos, aunque lo más habitual es utilizarlo desde algún editor gráfico como **RStudio**, propio de R, o Visual Studio Code.

En muchos aspectos R es comparado con Python, y en este trabajo se utilizará en las partes de extracción, procesado y visualización de la información (que también podrían haberse llevado a cabo con Python), así como en la parte de optimización de carteras junto a AMPL (ver 7.2.5).

Tidyverse

[108] Librería de R que a su vez consiste en un conjunto de librerías especialmente dedicadas a *Data Sciente*. Todos estos paquetes que contiene comparten una misma filosofía de diseño, así como gramática y estructuras de datos. Ejemplos de los mismos son **ggplot2**, dedicado a las representaciones gráficas o **tidyr**, dedicado al procesamiento y manipulación de datos.

Tidyquant

[91] Librería de R que a su vez consiste en un conjunto de librerías especialmente dedicadas al análisis económico. Trabaja con estructuras y diseños que tienen la ventaja de permitir utilizar las funciones de esta librería junto a las de tidyverse, siendo esta su principal ventaja. Algunos de los paquetes que contiene son **quantmod**, que permite descargar datos de activos económicos como los de este trabajo a través de las APIs de distintas páginas web, o **Performance Analyitics**, que contiene funciones y herramientas especialmente dedicadas al análisis financiero, como son las funciones utilizadas en este proyecto para calcular los distintos indicadores técnicos o para obtener los rendimientos de la serie de un valor.

Shiny

[109] Paquete de R dedicado a la construcción de páginas web interactivas directamente desde R. Con ella se puede tanto diseñar aplicaciones independientes, como incrustar aplicaciones en documentos R Markdown, o, como en este trabajo, incrustar elementos *iframe* en documentos HTML de páginas web convencionales. Shiny genera el código necesario para construir la web, pero a su vez permite la inclusión de elementos HTML, CSS y Javascript directamente dentro de sus aplicaciones. Una aplicación shiny se compone de los siguientes elementos:

- **ui.R:** controla la distribución (*layout*) y la apariencia de la aplicación.
- **server.R:** controla la lógica y los datos de la aplicación.
- **app.R:** crea la aplicación a partir de un elemento ui y de un elemento server.

Si bien tanto el elemento ui como el server pueden estar contenidos en un único fichero de R, resulta más recomendable tener dos archivos separados que luego se unan en el script app. En el caso de este trabajo, donde los elementos se insertan dentro de un documento HTML, es suficiente con tener los scripts ui.R y server.R en una carpeta e incluir adecuadamente el elemento *iframe*, es decir, no es necesario construir un script app.R como tal, aunque si se hiciera funcionaría igualmente.

La arquitectura empleada por las aplicaciones shiny corresponde a un modelo Cliente-Servidor, donde se utiliza **programación reactiva**. Esto se traduce en que cuando el usuario (cliente) modifica alguno de los elementos de la aplicación shiny, las salidas que dependen de dicho cambio son reconstruidas por el servidor y reenviadas al cliente. Esto se muestra en la Figura 7.2.

Un elemento fundamental de las aplicaciones construidas con esta librería son los **widgets** de control, que son los elementos con los que puede interactuar el usuario de la aplicación, dándole la capacidad de enviar mensajes a la aplicación shiny. Algunos están ya predefinidos en el paquete, ver la Figura 7.3.

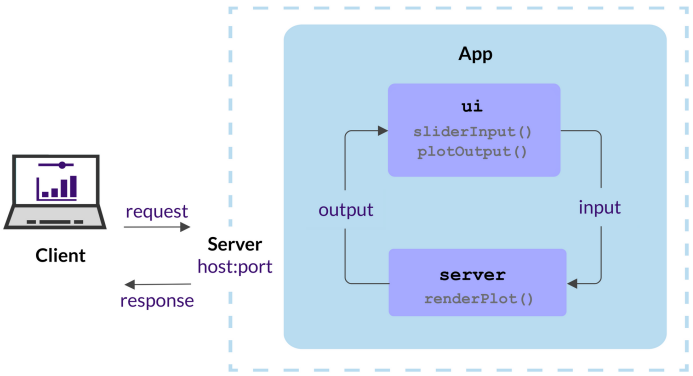


Figura 7.2: Shiny widgets. Tomado de [110].

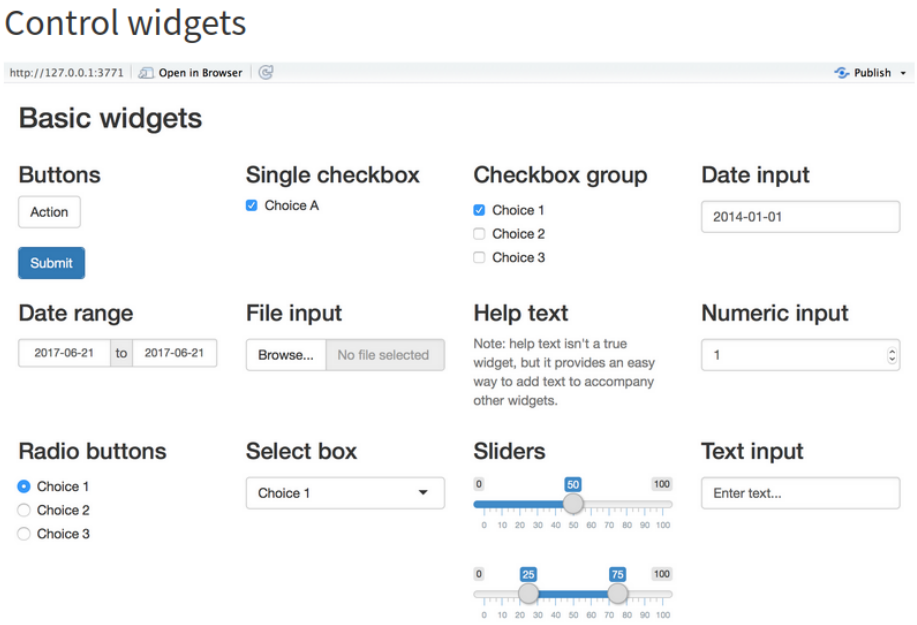


Figura 7.3: Shiny widgets. Tomado de [109].

Plotly

[111] Librería de R (también existente en otros lenguajes como Python) que proporciona funcionalidades dedicadas a la construcción de visualizaciones interactivas. Se basa en la construcción gráfica de ggplot2 y tiene funciones expresamente dedicadas al análisis gráfico de datos financieros como los de este proyecto.

Reticulate

[112] Librería de R que proporciona herramientas para la interoperabilidad entre R y Python en diferentes contextos: incluyendo código Python en un archivo R markdown, incluyendo llamadas a funciones Python desde R o incluyendo directamente scripts. A su vez también permite la traducción entre objetos de un lenguaje y otro como es el caso de los *dataframes* a través de la librería pandas de Python 7.2.5. Finalmente, permite el uso de entornos como *conda*.

rAMPL

[113] Librería de R que proporciona una interfaz de acceso al intérprete de AMPL (7.2.6), también existente para otros lenguajes. Permite generar y resolver modelos de programación matemática directamente con AMPL, utilizando su sintaxis propia, de modo que la librería únicamente ejerce de intermediaria entre este y el lenguaje desde el que se utiliza, lo que da un resultado rápido y eficiente en el uso de recursos. Se proporcionan funciones para asignar directamente los datos a los parámetros y conjuntos de AMPL, que pueden utilizarse en lugar de los procedimientos normales de lectura de datos. Del mismo modo pueden extraerse desde R los objetos generados en la resolución del modelo.. La arquitectura de la API de AMPL se muestra en la Figura 7.4.

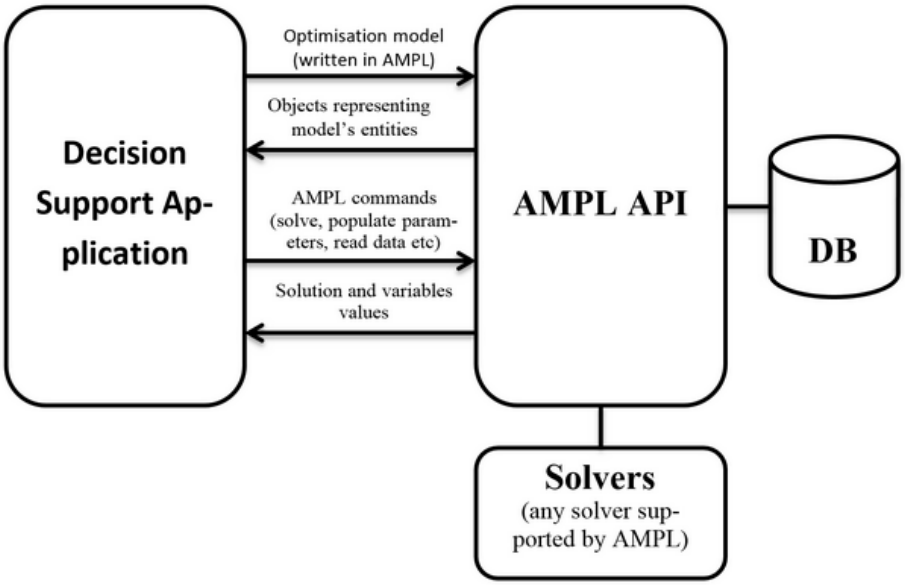


Figura 7.4: Arquitectura de la API de AMPL. Tomado de [113].

Python

[114] Python es un lenguaje de programación de propósito general muy popular hoy en día debido a sus altas capacidades y prestaciones, a su flexibilidad y a su sintaxis sencilla, muy próxima al lenguaje humano. Es un lenguaje interpretado, lo que quiere decir que no requiere del uso de compilador, y multiparadigma, ya que soporta tanto orientación a objetos, como programación imperativa y programación funcional. Es de código abierto y muy atractivo en el campo del Desarrollo Rápido de Aplicaciones (RAD) porque ofrece tipificación dinámica y opciones de encuadernación dinámicas. Se emplea en el desarrollo de aplicaciones de todo tipo, principalmente en los siguientes campos:

- *Data analytics* y *big data*.
- *Data mining*.
- *Data science*.
- Inteligencia artificial.
- *Blockchain*.
- *Machine learning*.

- Desarrollo web.
- Programación y desarrollo 3D.

En este trabajo se empleará en la construcción de los modelos de predicción de valores, ya que, aunque este tipo de construcciones de aprendizaje automático también se pueden llevar a cabo con R, Python lo hace de manera más rápida y eficiente.

Numpy

[115] Librería de Python más importante para la computación con fines científicos. Proporciona múltiples herramientas para el tratamiento matemático de la información como funciones matemáticas o funciones para manipular todo tipo de matrices, todo ello de manera sencilla. Sus principales ventajas están referidas al cálculo numérico y al manejo de grandes cantidades de datos, para lo cual define un tipo especial de objetos denominados ***arrays***.

Pandas

[116] Librería de Python especializada en el análisis y manejo de estructuras de datos. Presenta importantes funcionalidades como la lectura y tratamiento sencillo de diversos tipos de ficheros: `CSV`, `EXCEL` o `SQL`; la definición de estructuras de datos basadas en los *arrays* de la librería numpy, acceso a datos sencillo e intuitivo y funciones para ordenar, tratar o manipular conjuntos de datos.

Tensorflow

[117] Tensorflow es una plataforma de código abierto que facilita la creación de modelos de aprendizaje automático de redes neuronales de manera sencilla en múltiples entornos como aplicaciones de escritorio, de móvil o entornos de nube. Desarrollado por *Google* originalmente para uso interno pero actualmente de uso libre y extendido a todos los ámbitos. Se basa en unidades de procesamiento denominadas **tensores**.

Keras

[118] Biblioteca de Python para la creación de redes neuronales que permite utilizar todas las capacidades de Tensorflow. Es de uso sencillo y fue concebido para su uso como interfaz. Ofrece un conjunto de abstracciones intuitivas y de alto nivel haciendo sencillo el desarrollo de modelos de aprendizaje profundo independientemente del *backend* computacional utilizado. El paquete también existe para R. Los modelos de redes neuronales utilizados en este proyecto en la predicción de valores se construyen utilizando la interfaz de Tensorflow que proporciona Keras.

Scikit-learn

[119] Librería de Python para aprendizaje automático. Proporciona herramientas simples para llevar a cabo análisis de carácter predictivo y para implementar modelos de distintos tipos, en problemas de *clustering*, clasificación o regresión, así como herramientas destinadas al procesamiento de datos o la comparación, validación y selección de modelos. Varios de los modelos de predicción construidos en este trabajo se llevan a cabo con esta librería.

xgboost

[120] Librería de Python que implementa de manera eficiente el algoritmo del Descenso del Gradiente Estocástico XGBoost (ver 3.8.9).

statsmodels

[121] Librería de Python que proporciona diversas funcionalidades del mundo de la Estadística, como funciones para realizas distintos test o contrastes de hipótesis, realizar estimaciones, explorar datos o manejar modelos estadísticos.

7.2.6 AMPL

[122] Intérprete que soporta la creación completa de modelos de optimización, incluyendo las fases de formulación, evaluación y mantenimiento de los mismos. Utiliza una sintaxis propia y sencilla, que permite construir de manera sencilla modelos que luego se integren en sistemas mayores a través de su API, que permite utilizarlo desde lenguajes como C#, Matlab, Python o R (ver 7.2.5). AMPL integra su lenguaje de modelado con un lenguaje de comandos para el análisis y la depuración y un lenguaje de scripts para manipular los datos e implementar las estrategias de optimización. AMPL soporta diversos *solvers* (programas que resuelven problemas de optimización) y permite configurar las opciones de cada uno de ellos, enviarles un problema y extraer los resultados. El *solver* utilizado en este trabajo es **CPLEX** (ver 7.6.4).

7.3 Análisis y diseño

7.3.1 Visual Paradigm

[123] Es una herramienta UML que proporciona múltiples funcionalidades que van desde el diseño de diagramas hasta la generación automática de código o la ingeniería inversa a partir de scripts programados. Puede ser utilizado tanto en entornos Windows, como Unix como Cloud y fue lanzado en 2002 por *Visual Paradigm International Ltd.*.

7.4 Otras herramientas

7.4.1 Overleaf

Overleaf es una herramienta colaborativa *online* que permite realizar de manera completa el proceso de redacción y edición de documentos utilizando ***LaTeX***, de manera especialmente orientada a la creación de documentos científicos. Permite el uso colaborativo en tiempo real y produce la salida totalmente compilada de manera automática en segundo plano a medida que escribe. Esta herramienta tiene ventajas como: no necesidad de instalaciones ni actualizaciones al ser *online*, funciona en todo tipo de entornos, visualización inmediata del resultado al tener el compilador integrado, posee plantillas para realizar distintos tipos de documentos –como TFGs, TFMs o tesis doctorales– y posibilidad de restaurar versiones anteriores de los documentos.

7.4.2 Gantt Project

Gantt Project es una herramienta de escritorio de uso gratuito dedicada a la planificación de proyectos. Es una aplicación sencilla de utilizar que permite llevar a cabo tareas como desglosar un trabajo, construir diagramas de Gantt (ver sección 2.4), asignar recursos o calcular el coste de un proyecto. Además permite exportar el resultado como PDF y extraer imágenes como PNG, así como importar proyectos creados con otras herramientas como *Microsoft Project*.

7.4.3 Gitlab

[124] Sitio web que proporciona un sevicio de alojamiento de recursos mediante un repositorio git, junto a las debidas herramientas de gestión de este tipo de lugares.

7.5 Componentes de GUI

Todas las secciones de la aplicación web cuentan con una serie de elementos comunes:

- **Menú de navegación:** ver 6.3.2.
- **Pie de página:** ver 6.3.2.
- **Botón de ayuda:** acceso a la sección de ayuda de la web.
- **Descripción de la sección:** en la parte superior de la ventana se muestra un cuadro de texto que explica el contenido de la sección que se está visualizando, así como los aspectos básicos para entender su funcionamiento.
- **Elemento shiny:** en la parte central de la ventana se encuentra una aplicación shiny (o varias en el caso de la sección de mercados) donde se desarrolla la funcionalidad propia de la ventana correspondiente. Se incluye en el documento HTML mediante una etiqueta *iframe*, utilizadas para incrustar una página HTML dentro de la página actual

7.5.1 Componentes shiny

En 7.2.5 se comentó qué es un *widget* shiny y se mostraron cuáles son los principales (Figura 7.3). Los que se han utilizado para implementar las distintas interacciones en la web de este proyecto son:

- ***Select box:*** permite elegir un elemento o varios dentro de una lista.

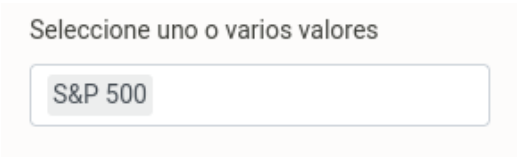


Figura 7.5: Ejemplo de *select box*.

- ***Date input:*** permite seleccionar un intervalo de fechas.

Figura 7.6: Ejemplo de *date input*.

- ***Slider input:*** permite seleccionar un valor numérico dentro de una escala.

Figura 7.7: Ejemplo de *slider input*.

- ***Numeric input:*** permite introducir un valor numérico.

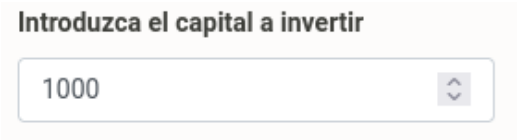


Figura 7.8: Ejemplo de *numeric input*.

- ***Action button:*** permite desencadenar una acción cuando se pulsa.

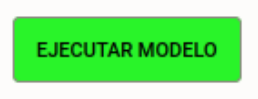


Figura 7.9: Ejemplo de *action button*.

- ***Radio buttons:*** permite seleccionar una entre varias opciones posibles.

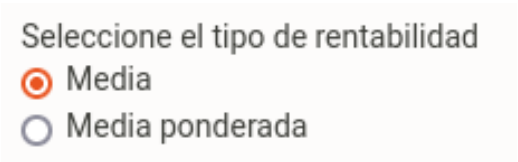


Figura 7.10: Ejemplo de *radio buttons*.

7.6 Secciones web

7.6.1 Inicio

Página inicial de la web, hace las veces de *landing page*, ver 6.3.2.

7.6.2 Mercados

Esta sección muestra las cotizaciones de los mercados considerados en este trabajo en forma de tablas. La información que se visualiza es tanto el valor de la cotización de cada acción como el porcentaje de subida o de bajada con respecto al día anterior. En primer lugar, se incluyen los mercados bursátiles, donde se muestran tanto los índices S&P 500, NASDAQ 100, IBEX 35 y EURO STOXX 50 como las empresas que los componen. En segundo lugar, se muestran las principales criptomonedas de la actualidad. En tercer lugar, se muestran algunas

de as materias primas más importantes de dicho mercado y, finalmente, se muestran algunas de las divisas de los países más importantes del mundo.

7.6.3 Análisis de valores

Esta sección se centra en el análisis descriptivo de valores mostrando distintos gráficos de interés dentro del contexto de la inversión. El gráfico de líneas de la serie temporal de un valor muestra la cotización de un valor y a su vez permite la superposición de la serie de distintos valores en un periodo dado, así como la superposición de distintos indicadores técnicos que ayudan a entender e interpretar dicho gráfico. Los gráficos de velas y de barras muestran información sobre las fluctuaciones en los precios de cierre ajustados de un valor, donde el color verde identifica una subida en el valor del activo y el rojo una bajada. El gráfico de velas incluye también los precios máximo y mínimo y a su vez se acompaña de un gráfico de barras que representa el volumen. El *boxplot* de rendimientos permite representar la distribución de la serie de rendimientos mostrada en el gráfico anterior. Además de estos gráficos, se permite visualizar la serie de rendimientos que muestra la evolución del rendimiento otorgado por un valor en un periodo de tiempo dado. La serie de disminuciones permite representar las variaciones en el valor de un activo al mostrar la rentabilidad total acumulada frente al máximo de la rentabilidad máxima acumulada.

7.6.4 Optimización de carteras

Sección que aborda el problema de la optimización de carteras explicado de manera teórica en 3.7.

7.6.5 Evaluación de carteras

En esta sección se llevará a cabo la evaluación de la cartera obtenida con el modelo de optimización de carteras creado en la sección de optimización de carteras. Dicho proceso se realizará sobre los datos reservados en esa sección para llevar a cabo la evaluación y se realizará en términos de rentabilidad. En el caso de que el modelo de optimización entrenado sea biobjetivo (maximización de la rentabilidad y minimización del riesgo de manera simultánea) se mostrará la evolución de los rendimientos de las 4 principales carteras eficientes consideradas en este trabajo: mínimo riesgo, máximo rendimiento, máximo ratio sharpe y equiponderada. Por el contrario, si el modelo es un modelo básico (solamente tiene como objetivo uno de los dos antes mencionados), únicamente se representará la evolución de la cartera óptima obtenida. En caso de que el modelo fuera biobjetivo pero que el parámetro μ; se fije, la representación será análoga a la realizada con los modelos básicos.

7.6.6 Regresión de índices

Sección que aborda el estudio de las regresiones entre índices y carteras y la descomposición del riesgo explicada en 3.7.4.

7.6.7 Simulación de carteras

Sección que trata la simulación de carteras con el método de Montecarlo comentada en 3.7.5.

7.6.8 Modelos de predicción

Sección que aborda la elaboración de modelos de inteligencia artificial para la predicción de valores de activos comentado en 3.8.2 con los modelos ya explicados.

7.6.9 Análisis de grupos

Sección que trata la elaboración de grupos de activos con un comportamiento similar en términos de rentabilidad, explicado en 3.9.1.

7.6.10 Ayuda

Sección de ayuda de la web. Consiste en una nueva ventana compuesta únicamente de un cuadro de texto que informa del contenido de cada una de las secciones así como de las distintas opciones de configuración de cada una de ellas, con el fin de servir de consulta si le usuario lo requiere.

7.6.11 Algoritmos

La implementación de las distintas secciones tiene una estructura similar siempre, con una lectura de datos, un preprocesamiento y tratamiento de los mismos consistente en eliminar valores ausentes, ajustar dimensiones en caso de que se trabaje con varias series temporales y estas difieran y unificar el formato (trabajar con *dataframes*, ya que el tipo de objeto que provee 7.2.5 es diferente).

En cuanto a los algoritmos que llevan a cabo las tareas concretas de cada sección, cabe mencionar que la de *clustering* utiliza el algoritmo de K medias comentado en 3.9.1, que la de regresiones de índices utiliza una regresión lineal simple, que la de modelos de predicción utiliza los procedimientos comentados en 3.8.2 y que la de optimización resuelve los modelos utilizando el *solver CPLEX* [125] (a través de AMPL 7.2.6), que utiliza el ***algoritmo SIMPLEX*** (escrito en lenguaje C, de ahí el nombre), base de la programación lineal [126].

A modo de ejemplo se muestran los diagramas de flujo de las secciones de predicción de modelos (Figura 7.11) y de optimización de carteras (Figura 7.12), donde se describen de manera simple y breve los pasos a seguir para obtener las salidas deseadas.

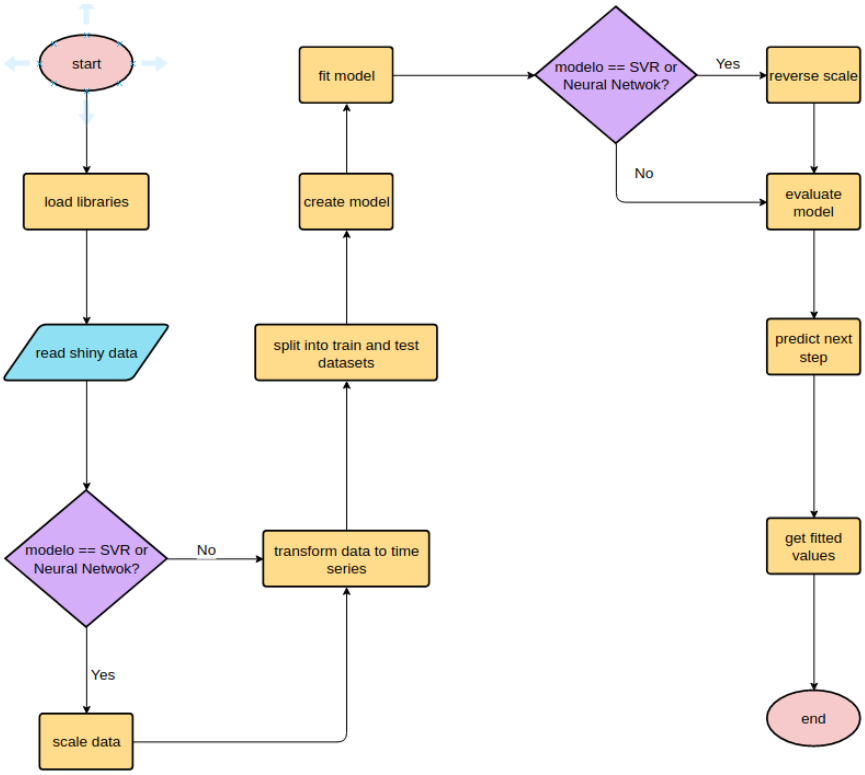


Figura 7.11: Diagrama de flujo de la sección de modelos de predicción. Realizado con 7.3.1.

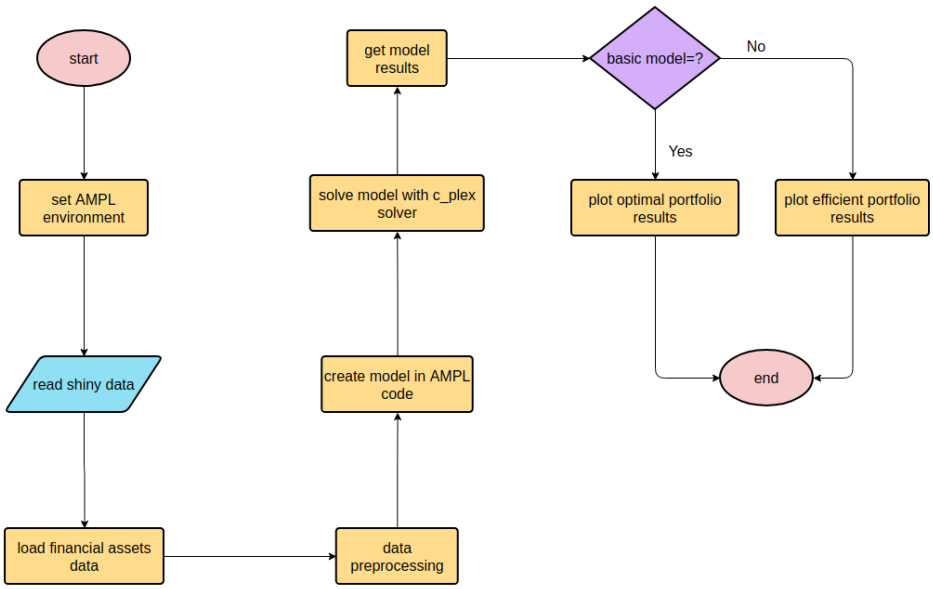


Figura 7.12: Diagrama de flujo de la sección de optimización de carteras. Realizado con 7.3.1.

Capítulo 8

Ejemplos de Uso

En este capítulo se ilustra el uso de la aplicación, mostrando ejemplos concretos del uso de cada sección e interpretando los resultados obtenidos como muestra de la utilidad de la misma.

8.1 Mercados

En la Figura 8.1 se muestran las cuatro aplicaciones shiny que componen el cuerpo de esta sección, constituyendo un *dashboard* con cuatro tablas, una para cada tipo de mercado de los considerados en este trabajo. En primer lugar, se ven los mercados bursátiles, con los índices S&P 500, NASDAQ 100, IBEX 35 y EURO STOXX 50 en primer lugar, seguidos de las empresas de los mismos, comenzando por las del primero de los mencionados. En segundo lugar se muestra la tabla de las criptodivisas, en tercero la de las materias primas y, en cuarto, la de las divisas. Las tablas mostradas permiten la interacción del usuario, que puede buscar un activo en su mercado correspondiente con el cuadro de búsqueda, así como moverse entre las distintas páginas de la tabla para ver los valores que no se muestran y variar el número de los que aparecen de manera simultánea en cada tabla. La columna “Cambio” muestra el porcentaje del cambio en el valor de un valor con respecto al día anterior. El color nos da un indicativo rápido y visual de que en el día anterior, la mayoría de los activos mostrados inicialmente sufrieron una devaluación.

8.2 Análisis de valores

En la Figura 8.2 se muestra un ejemplo de uso de la sección de análisis, donde se representa la serie temporal del precio del índice S&P 500 en los últimos tres meses con datos diarios. Junto al gráfico de la serie se añaden algunos indicadores técnicos a modo de ejemplo, en distintos ejes en función de la escala en la que se mueva cada uno (hay un total de 3 ejes adicionales al eje que representa la serie original). En este caso los mostrados son el MACD, el RSI y el CCI, como indica la leyenda. Ver 3.2.2 para saber más acerca de la información que aportan.

En 8.3 vemos otro de los gráficos de esta sección, el de velas, japonesas. Se muestra además como, al pasar el ratón por alguna de las velas (interacción con el gráfico) se muestra una etiqueta con la información de la vela. Los colores indican una subida (verde) o bajada (rojo) en el valor de la acción. Del mismo modo se añade el volumen, como es típico en este tipo de gráficos, mediante un diagrama de barras convencional donde los colores representan análogamente un aumento o disminución en dicho indicador. Sin incluir el índice se muestra en 8.4 el gráfico de barras, de interpretación análoga al de velas, aunque algo menos visual.

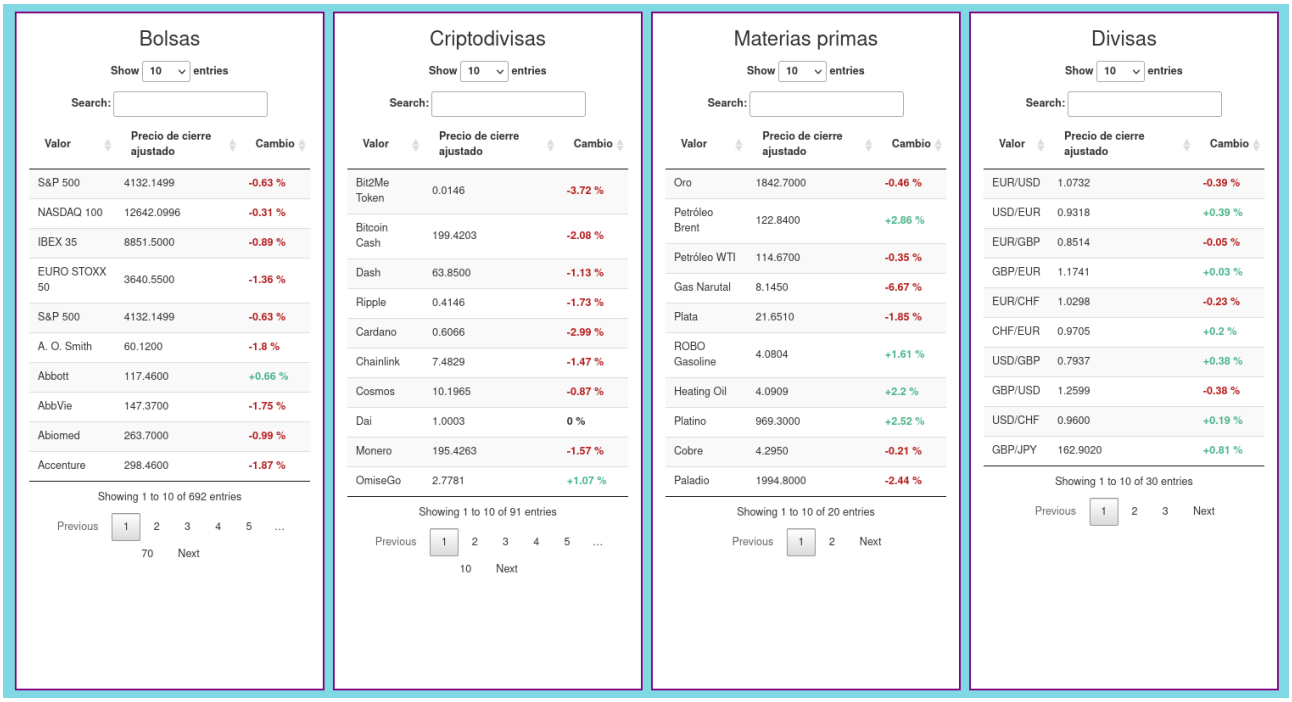


Figura 8.1: Elementos shiny sección de mercados.

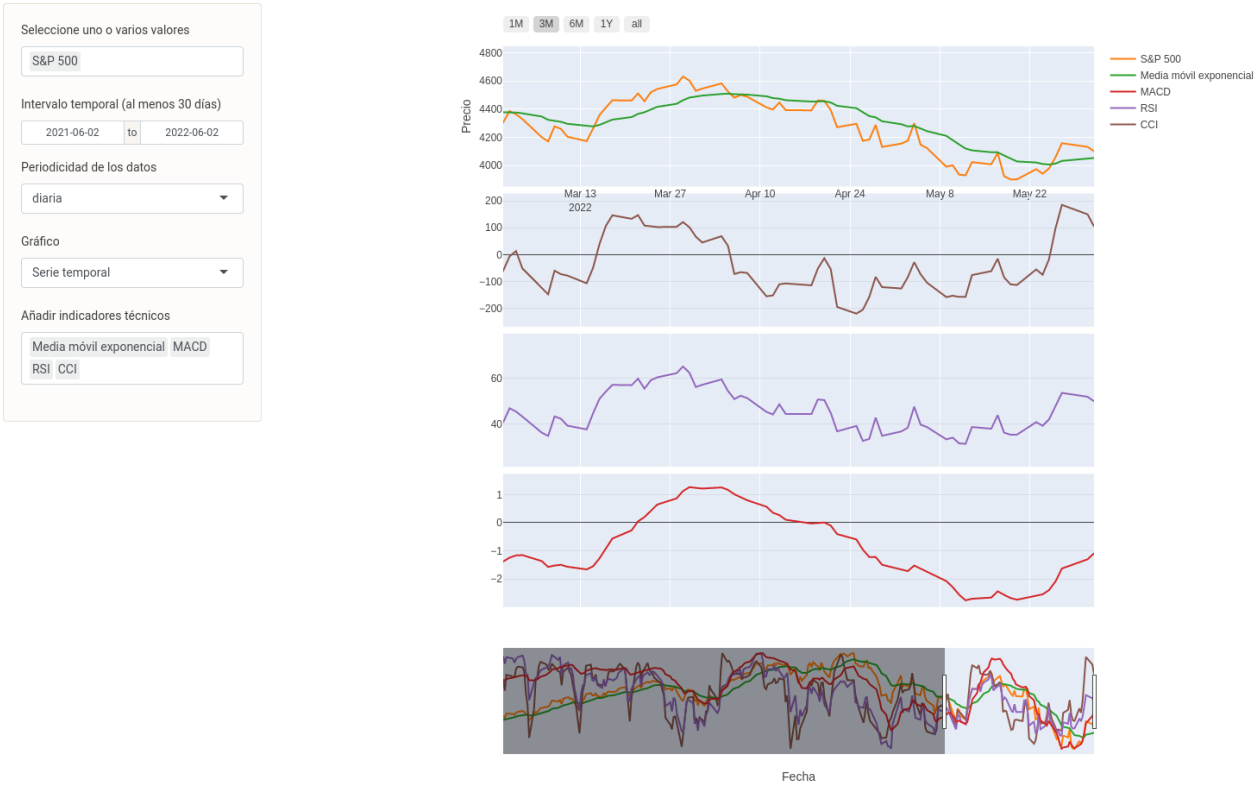


Figura 8.2: Gráfico de la serie temporal del índice S&P 500 de los últimos 3 meses.

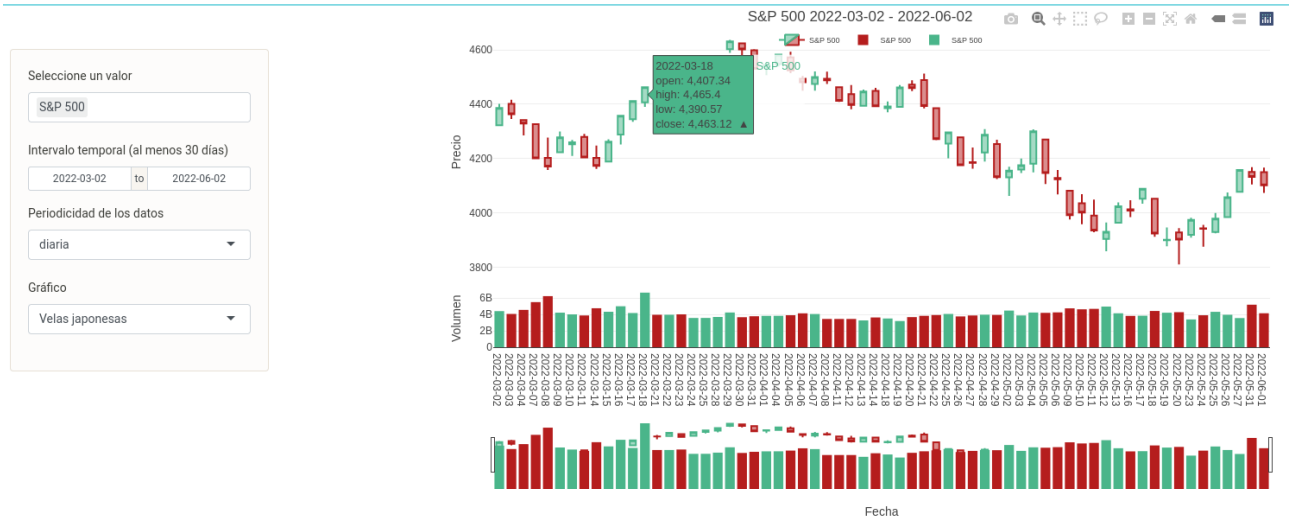


Figura 8.3: Gráfico de velas japonesas del índice S&P 500 de los últimos 3 meses.

Las Figuras 8.5 y 8.6 muestran información de los rendimientos del índice durante los últimos tres meses, con un comportamiento bastante estable en torno al 0, aunque con un aumento de la variabilidad a partir de mediados de abril, probablemente fruto de la inestabilidad mundial de los últimos meses, fruto del conflicto entre Ucrania y Rusia, acompañado del aumento de la inflación en Estados Unidos, mercado donde opera el S&P 500.

Finalmente, en 8.7 se representa la serie de disminuciones del índice en los últimos 3 meses. En este gráfico se representa la rentabilidad acumulada en esa franja temporal y se toma como límite superior el máximo punto de la rentabilidad acumulada del valor en dicho periodo. Cada vez que la rentabilidad cae por debajo del 0 se considera una reducción y la representación se hace en términos de porcentaje. Cabe destacar que el pico inferior se alcanza en mayo, con una pérdida de valor de en torno al 15 % con respecto a su máximo de los últimos tres meses.

Figura 8.4: Gráfico de barras del índice S&P 500 de los últimos 3 meses.

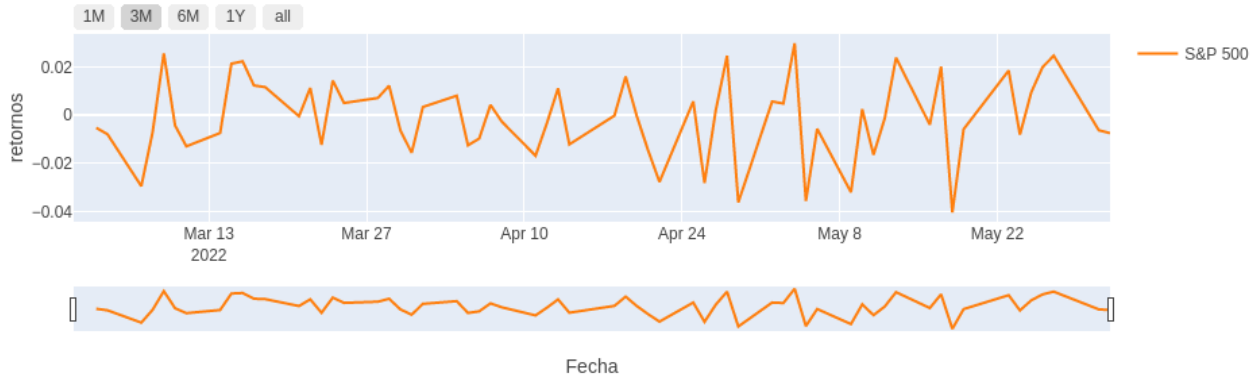


Figura 8.5: Gráfico de la serie de rendimientos del índice S&P 500 de los últimos 3 meses.

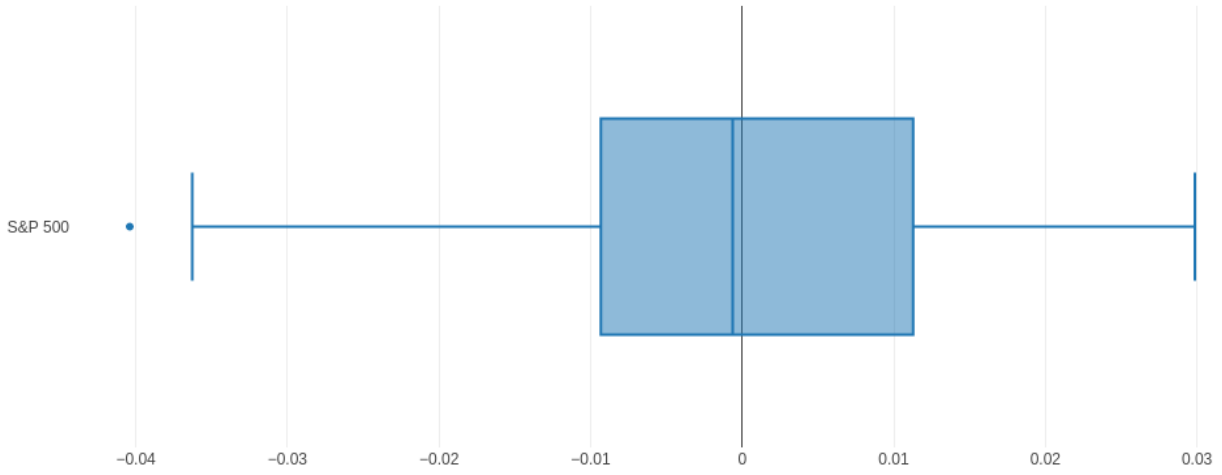


Figura 8.6: Gráfico boxplot de rendimientos del índice S&P 500 de los últimos 3 meses.

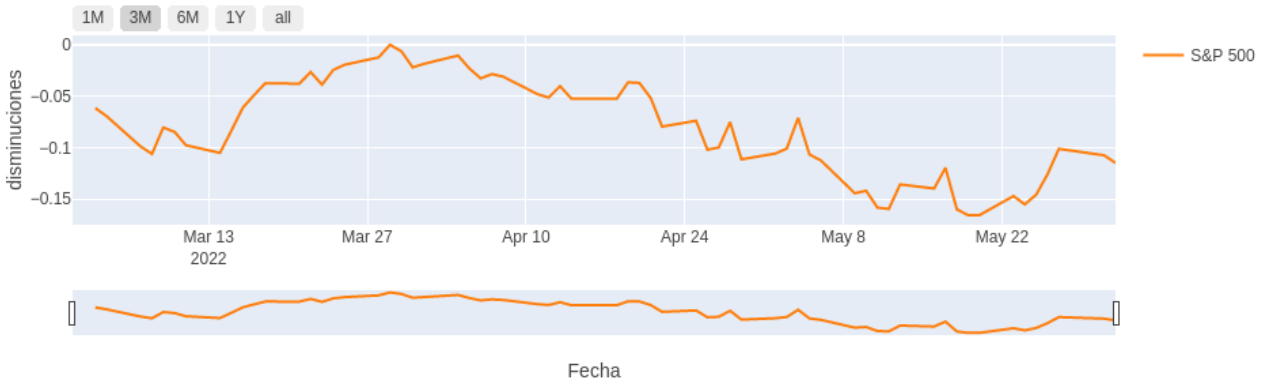


Figura 8.7: Gráfico de la serie de disminuciones del índice S&P 500 de los últimos 3 meses.

8.3 Optimización de carteras

Puesto que son numerosas las opciones de configuración de esta sección y múltiples las salidas que a su vez pueden obtenerse, se comentarán aquí únicamente algunas de ellas. El ejemplo elegido para este análisis parte de un conjunto de valores procedentes de diversos mercados, con datos entre el 2 de junio de 2021 y el 2 de junio de 2022:

- Índice NASDAQ 100.
- Empresas Apple, Boeing, BBVA, Repsol, Coca-Cola y Starbucks.
- Criptomoneda Bitcoin.
- Materia prima Oro.
- Divisa USD/EUR.

Figura 8.8: Composición de las carteras notables.

La media elegida es la ponderada (corrección de estimadores) con un factor de corrección de $0{,}9$, el modelo seleccionado es el biobjetivo de Markowitz, el tipo de interés de referencia para el cálculo del ratio Sharpe es 0, el capital invertido es 1000 y las restricciones impuestas indican que no se puede invertir ni menos de un $2\,\%$ ni más de un $37\,\%$ en un activo y que el número máximo de empresas en el que se puede invertir es 6 (restricción de cardinalidad). Nótese que, para cumplir de manera simultánea la restricción de cota inferior y la de cardinalidad, esta segunda ha de convertirse necesariamente en una restricción de igualdad estricta, ya que de otra forma no podrían cumplirse ambos propósitos. El porcentaje de datos utilizados para entrenamiento es el 80 %.

En la Figura 8.8 se muestra tanto la navegación entre las distintas salidas del modelo, como la primera de ellas. La tabla muestra como la cartera de máximo ratio Sharpe parece lograr un buen equilibrio entre riesgo y rentabilidad, frente a las otras carteras notables que buscan los extremos de estos valores. En el gráfico de puntos se muestra esto de manera más visual. Cabe destacar que la cartera equiponderada, que en 3.7.2 se comentó como interesante, no se muestra en este caso ya que la restricción de cardinalidad hace que no tenga sentido un reparto equitativo del capital entre todos los activos introducidos en el modelo.

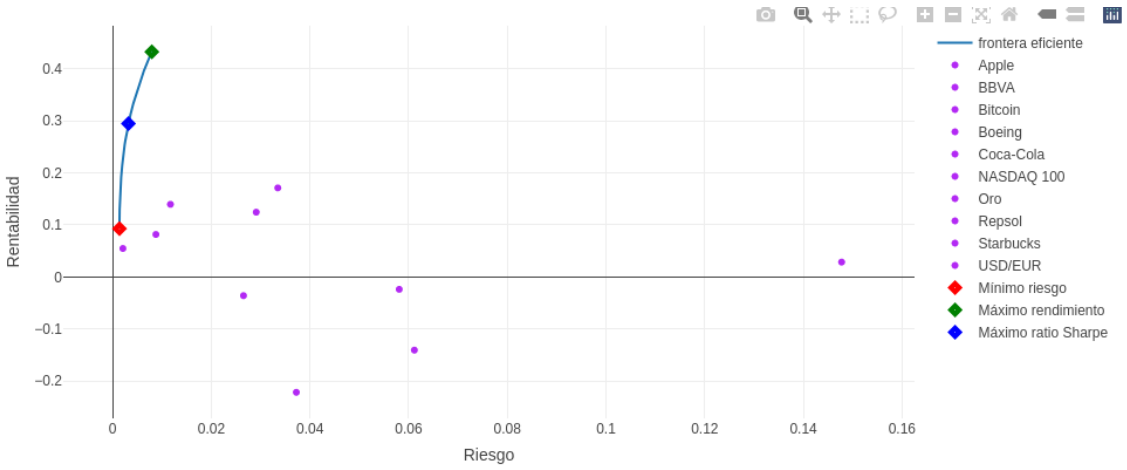


Figura 8.9: Frontera eficiente.

La Figura 8.9 muestra el gráfico de la frontera eficiente obtenida con el modelo construido. Los puntos morados representan las carteras que se obtendrían en caso de invertir el 100 % del capital en cada uno de los activos de manera individual. Se aprecia cómo la curva de la frontera eficiente es muy redondeada, lo que se debe a que Bitcoin es un activo de mucho riesgo (es el punto morado situado más a la derecha de todos), característica inherente al mercado de las criptomonedas del que forma parte. Si eliminásemos este activo la frontera tendería mucho más rápidamente a aplanarse. Las carteras eficientes muestran la misma disposición que en el gráfico de puntos de la Figura 8.8, y vemos cómo, además de Bitcoin, hay otros cuatro activos que no resultan rentables si solo se invierte en ellas. Son BBVA, Boeing, Starbucks y el índice NASDAQ 100. La composición de las cartera que constituyen la frontera eficiente se muestra en la Figura 8.10, obtenida en la tercera pestaña de resultados “Gráficos carteras notables”. En esta misma pestaña se pueden visualizar las composiciones de las carteras notables, tanto en forma de gráfico de sectores como 8.11 como de barras 8.12. En el primer caso vemos que el oro es el activo con mayor peso en la cartera de máximo ratio sharpe, llegando al máximo permitido por la restricción, mientras que Coca-Cola y NASDAQ 100 únicamente cubren el porcentaje mínimo impuesto por la restricción. En el segundo ejemplo mostramos la cartera de máximo rendimiento, donde tanto Repsol como BBVA llegan al máximo permitido del $37\,\%$ y en este caso son nuevamente Coca-Cola y NASDAQ 100 las que se quedan en la cota superior del $2\,\%$, en este caso junto a Boeing.

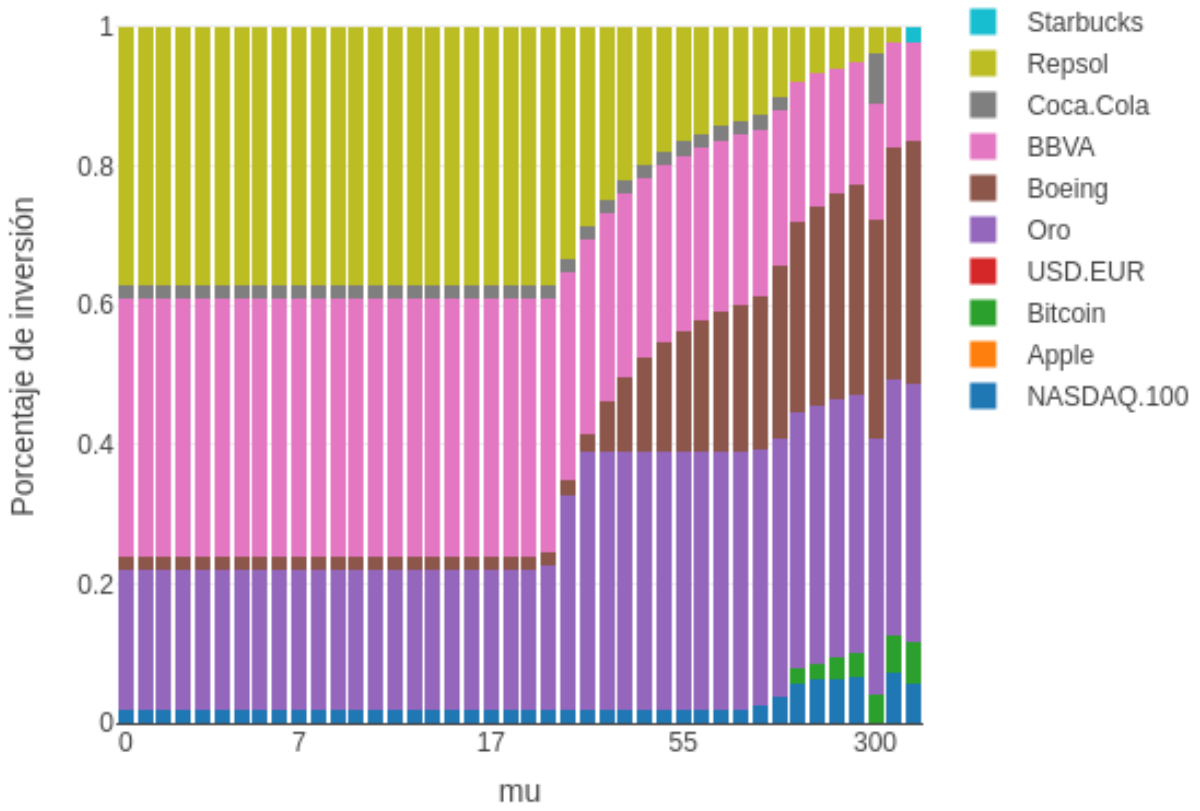


Figura 8.10: Composición cartera eficiente.

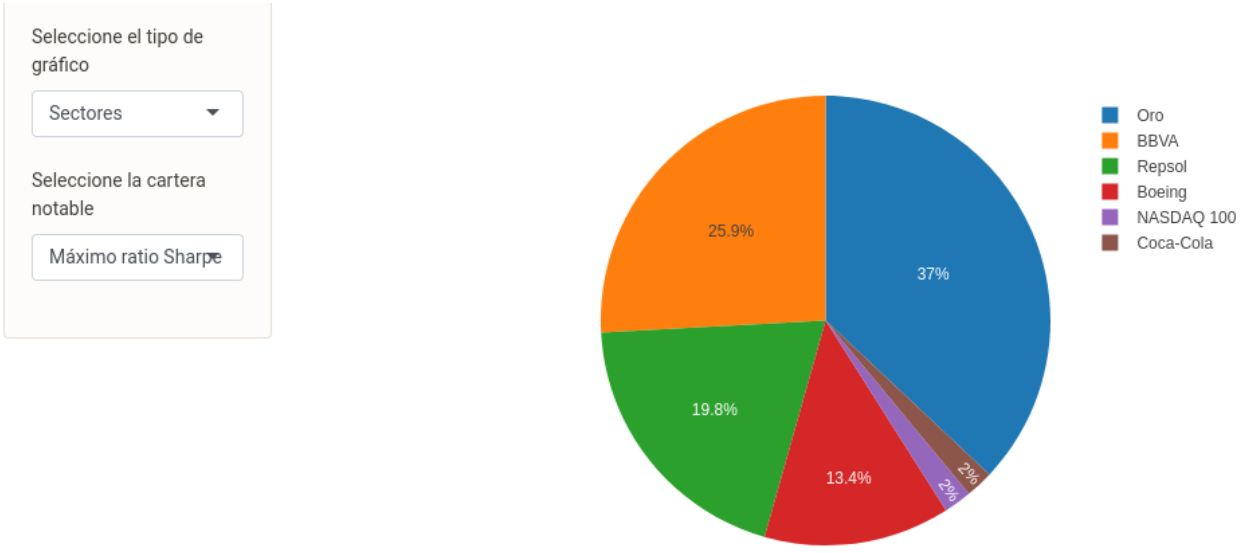


Figura 8.11: Composición de carteras notables en diagrama de sectores.

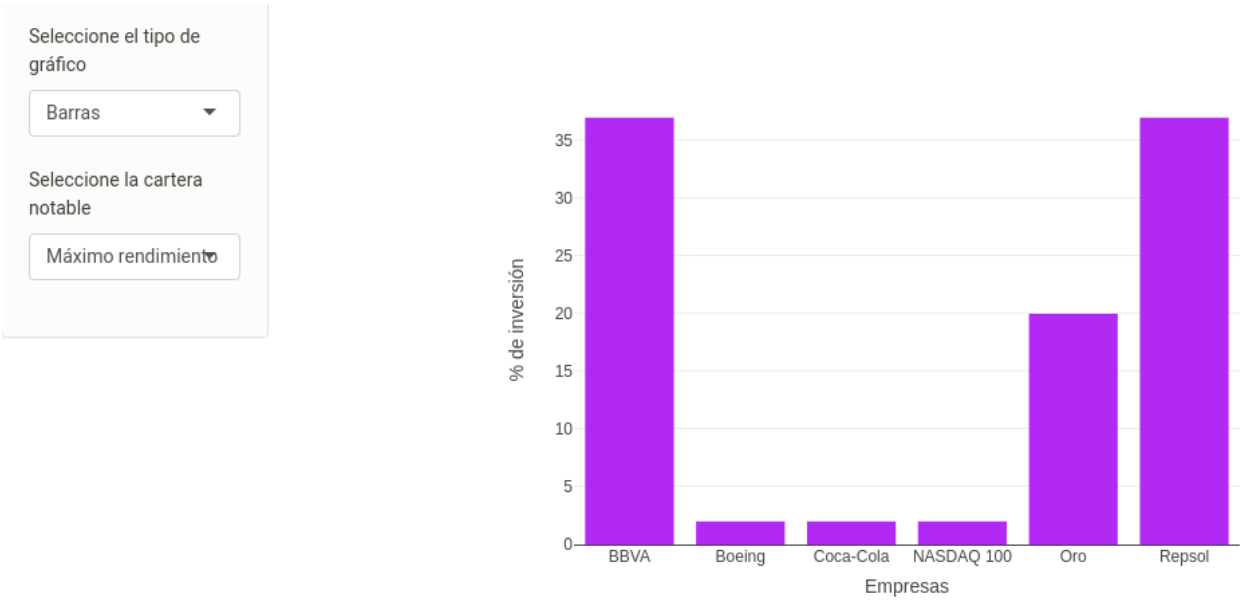


Figura 8.12: Composición de carteras notables en diagrama de barras.

Otra interesante salida que podemos obtener en esta sección es la que se ve en la Figura 8.13, donde se muestra el porcentaje de inversión en cada empresa que contempla cada una de las carteras notables.

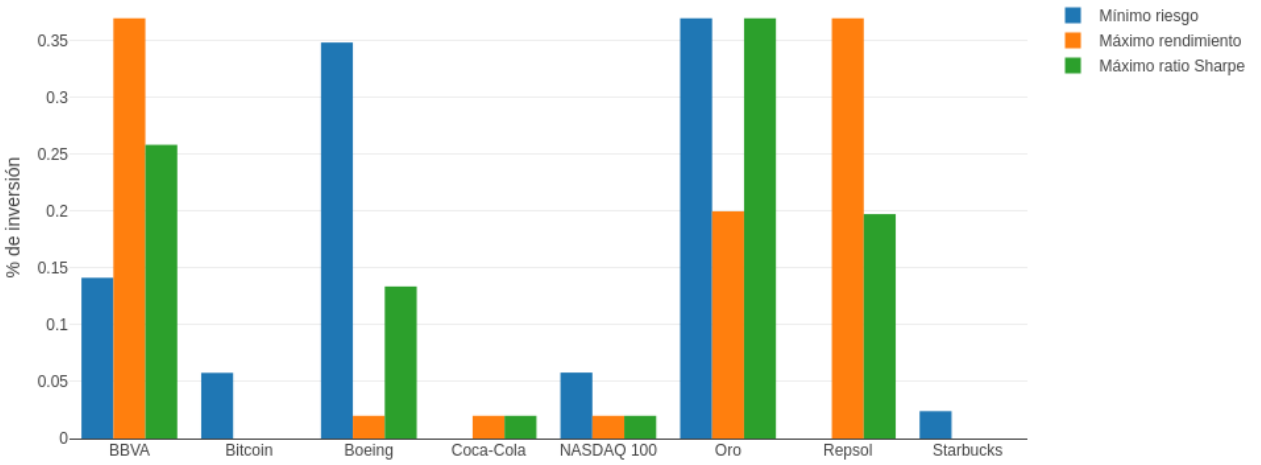
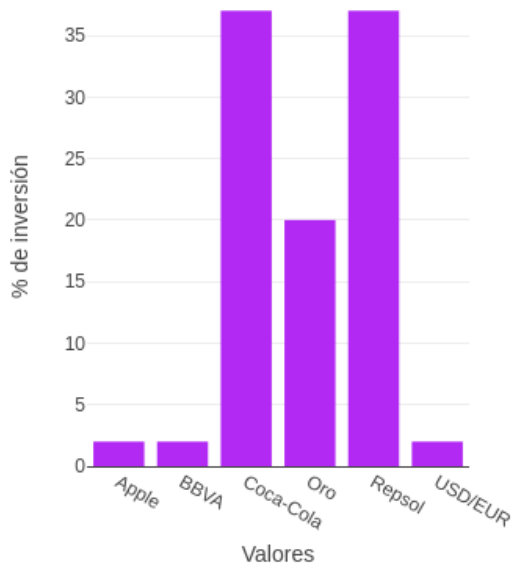


Figura 8.13: Comparación de carteras notables.

Finalmente, en 8.14 vemos cómo se puede obtener la composición de una cartera óptima variando el valor del parámetro μ de manera manual. Esta representación la podemos obtener en forma de diagrama de sectores, de barras como el mostrado o en forma de *radar chart*, además de la tabla que se muestra siempre. Comprobamos para el ejemplo mostrado que, con un $\mu = 2$ Coca-cola y Repsol cubren el máximo de la inversión permitida, mientras que Apple, BBVA y la divisa USD/EUR se quedan en el mínimo del $2\,\%$.



Activo	Porcentaje	Inversión
Apple	0.02	20.00
BBVA	0.02	20.00
USD/EUR	0.02	20.00
Oro	0.20	200.00
Coca-Cola	0.37	370.00
Repsol	0.37	370.00

Figura 8.14: Gráfico en función de mu.

8.4 Evaluación de carteras

Únicamente tras haber realizado la optimización de una cartera podremos llevar a cabo su evaluación. En este caso, esta se realiza con el $20\,\%$ de los datos comprendidos entre el 2 de junio de 2021 y el 2 de junio de 2022, para la cartera óptima obtenida con el $\mu = 2$ fijado en el final del apartado anterior. Si se hubiera hecho sin fijar dicho parámetro e igualmente con el modelo biobjetivo, en lugar de evaluar la rentabilidad de una cartera óptima (no las hay), se evaluarían las carteras notables construidas.

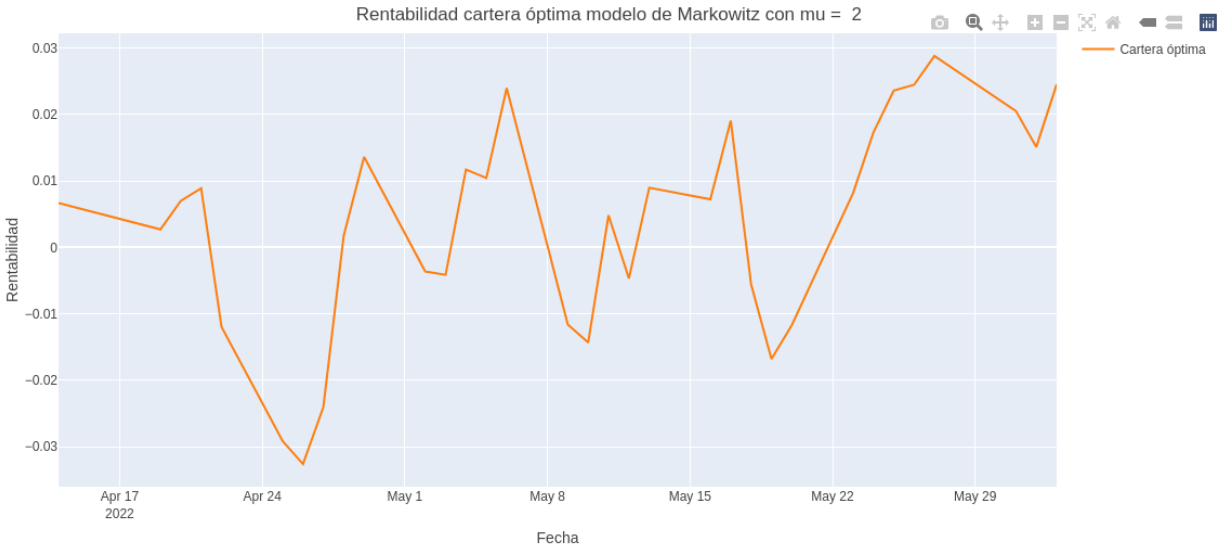
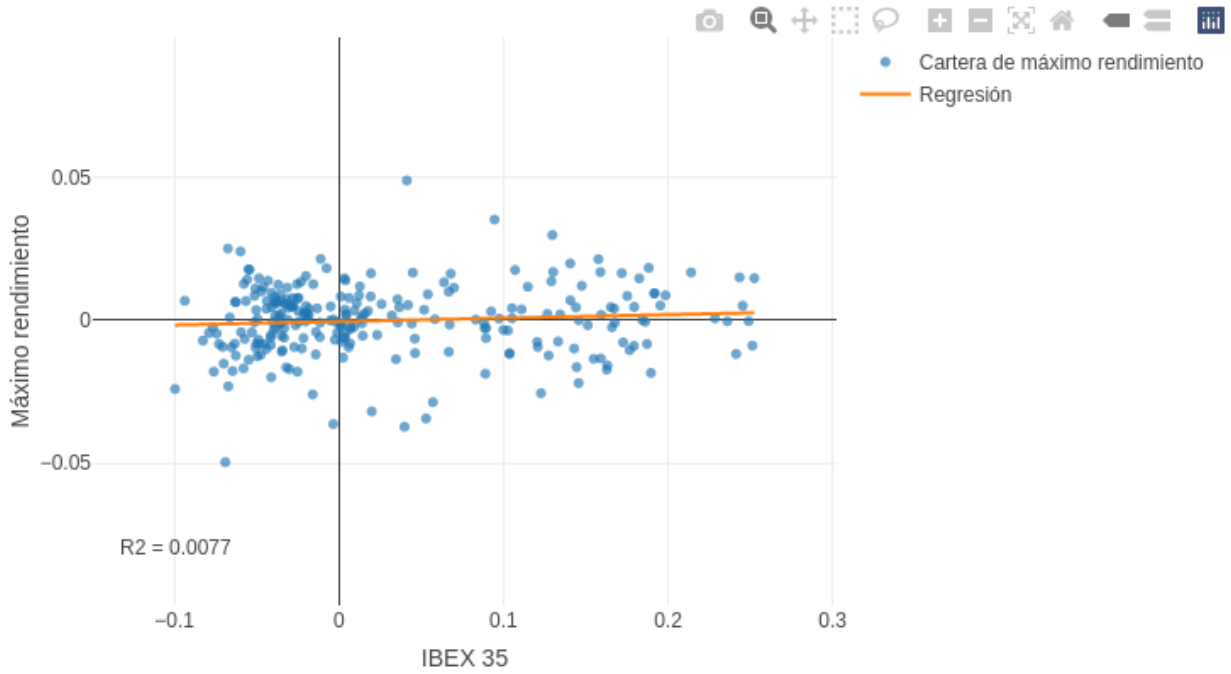


Figura 8.15: Evaluación de cartera óptima.

8.5 Regresión de índices

En esta sección tendremos que utilizar una configuración semejante a la planteada en la sección de optimización, ya que para estudiar la regresión de una cartera frente a un índice hay que construir previamente la cartera y eso implica utilizar modelos de optimización también aquí. La diferencia a la hora de ejecutar esta sección radica en dos aspectos: en primer lugar, las empresas han de pertenecer todas a un mismo índice, por lo que seleccionar uno será lo primero que hagamos para luego elegir las empresas con las que se planteará la construcción de la cartera. En segundo lugar, aquí la ejecución no es reactiva, es decir, no se actualiza el modelo generado ni la visualización mostrada cada vez que el usuario modifica alguno de los elementos de configuración, sino que aquí la ejecución se desencadena pulsando un botón. Este sistema se utilizará en el resto de secciones y se ha tomado esta decisión ya que parece más coherente y, desde luego, da menos problemas, que la forma de trabajar en la parte de optimización. En esa se trabaja diferente porque se planteó la posibilidad de que quizás fuera más interesante poder modificar con más soltura el modelo generado y ver de manera más rápida los cambios en la salida.

En este caso, las empresas elegidas han sido pertenecientes al índice español IBEX 35: Acciona, Grup ACS, Banco Sabadell, Telefónica, Iberdrola, Inditex, Mapfre, Endesa y Caixabank. Los datos en este caso fueron del 3 de junio de 2021 al 3 de junio de 2022 y la configuración volvió a ser semejante a la anterior, con un modeo biobjetivo con media ponderada (factor de descuento de $0{,}9$), tipo de interés de referencia igual a 0, cota inferior del $2\,\%$, superior del $37\,\%$ y máximo de 7 empresas. Se eligió realizar la regresión frente a la cartera de máximo rendimiento.



Riesgo.sistematico	Riesgo.especifico	Riesgo.total
0.0000000214983148223596	0.000146814884598637	0.000146836382913459

Figura 8.16: Regresión de la cartera con máximo ratio Sharpe frente al IBEX 35.

En 8.16 vemos la regresión de los rendimientos de la cartera obtenida frente a los rendimientos del índice. El sentido positivo de la pendiente indica que los rendimientos de ambos se espera que se muevan en la misma dirección. El valor del R^2 es bajo, como muestra la clara dispersión en los puntos de los rendimientos de la cartera.

En la tabla se muestra la descomposición del riesgo explicada en 3.7.4, donde vemos que la mayor parte del riesgo proviene del riesgo sistemático, es decir, de los activos que componen propiamente la cartera.

8.6 Simulación de carteras

Seleccionamos los mismos diez activos que en la sección de optimización, con un intervalo temporal comprendido entre el 3 de junio del 2021 y el 3 de junio de 2022. El tipo de interés de referencia para el ratio Sharpe lo mantenemos en 0. Este estadístico se utilizará para dar el color a las distintas carteras que se obtendrán en la simulación. Realizando 10000 simulaciones obtenemos el resultado mostrado en la Figura 8.17.

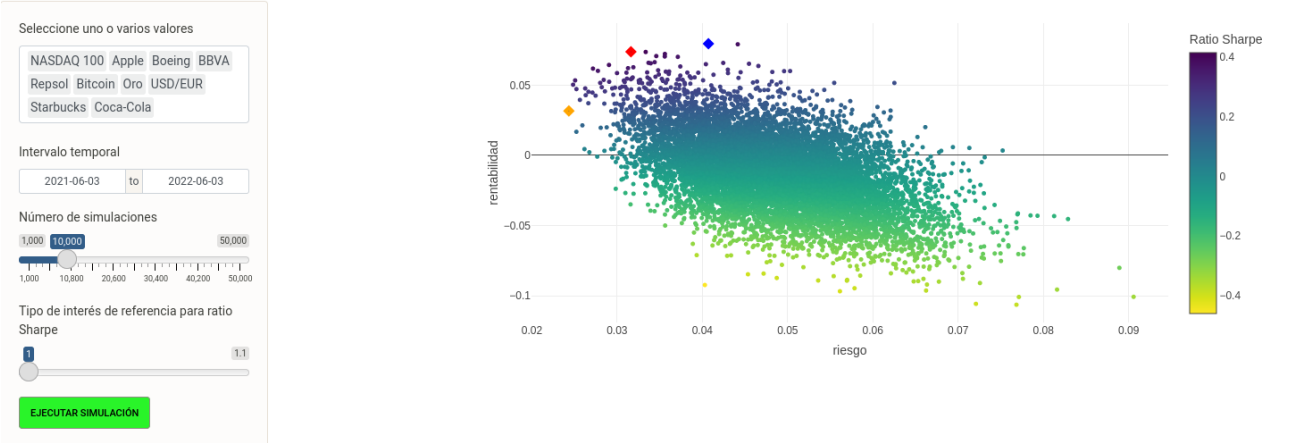


Figura 8.17: Simulación de carteras.

Vemos que las tres carteras notables (se omite la equiponderada) forman parte de lo que sería la envolvente convexa de la nube de puntos, como se explicó en 3.7.5.

8.7 Modelos de predicción

En esta sección utilizamos únicamente un valor, que en este caso es el índice bursátil S&P 500, con datos diarios del último año. A fin de poder establecer una comparación utilizamos 3 modelos: Random Forest, LSTM y ARIMA, con las características que indica la primera salida que obtenemos en la Figura 8.18. A modo de ejemplo mostramos también la salida de la pestaña “Panel del modelo” para el primero de los construidos (Figura 8.19) donde vemos que tanto la curva original como la curva ajustada por el modelo son muy próximas. A su vez se muestra que la predicción para el 3 de junio de 2022 (como los datos utilizados son diarios, la predicción obtenida es para el día siguiente) es de $4097{,}3734$. Las métricas de la parte inferior muestran la evaluación del ajuste del modelo.

La comparación de modelos de la tercera pestaña de resultados la vemos en la Figura 8.20. La explicación de las métricas está comentada en 3.8.3, pero a modo de resumen decir que para las tres métricas seleccionadas (RMSE, MSE y MAE) interesa que el valor sea lo más bajo posible, por lo que podemos concluir que para el ejemplo mostrado, el modelo ARIMA es el que da mejores resultados y el Random Forest el que peores, quedando el LSTM en medio. Comentar que esto no indica que un modelo sea mejor ni peor que otro, únicamente muestra cómo ha sido el rendimiento de cada uno de ellos en un ejemplo concreto. Para poder establecer una comparativa real entre modelos habría que diseñar un experimento más complejo que involucraría distintos conjuntos de datos y un mayor número de pruebas. Además, los parámetros seleccionados para construir los modelos no han sido elegidos mediante ninguna metodología de evaluación, ya que el objetivo del ejemplo es ilustrar el funcionamiento de la sección de modelos de predicción. Sin embargo, esta es una funcionalidad que permite la web construida, ya que podemos introducir variantes del mismo modelo y comparar los resultados en la pestaña de “Comparación de modelos”, con el fin de obtener la configuración que proporcione mejores resultados, insisto, para unos datos concretos.

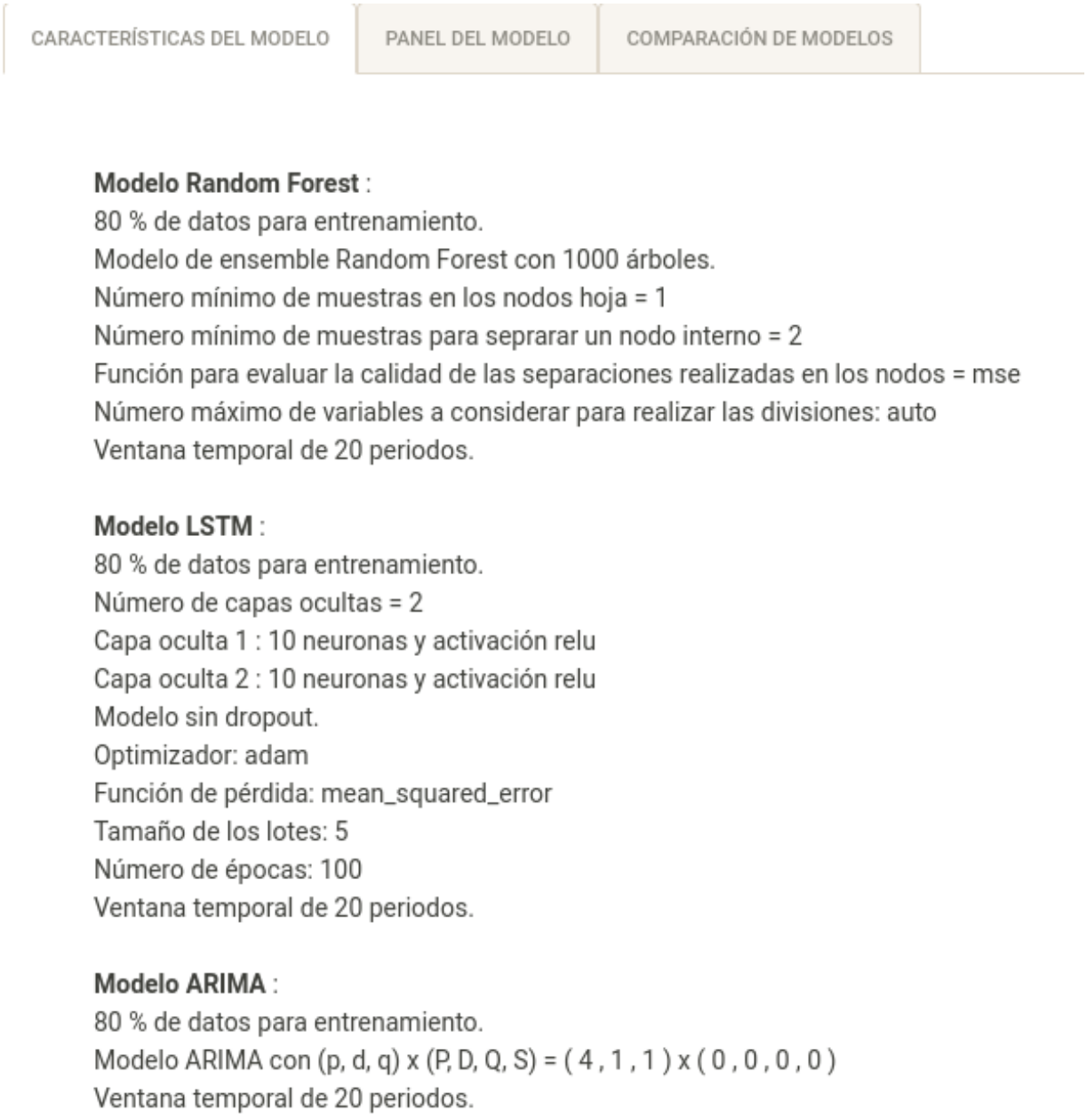


Figura 8.18: Descripción de los modelos ajustados.

Métrica	Valor
RMSE	160.49
MSE	25758.33
MAE	123.56

Figura 8.19: Ajuste del modelo Random Forest.

Métrica	Random.Forest	LSTM	ARIMA
RMSE	160.49	128.57	72.27
MSE	25758.33	16530.63	5223.59
MAE	123.56	102.83	56.58

Figura 8.20: Comparación de modelos.

8.8 Análisis de grupos

En esta sección introducimos un número grande de activos, pertenecientes a diversos mercados, con el objetivo de ver si los activos de los distintos mercados tienen un comportamiento similar en términos de rentabilidad como cabría esperar. Los datos corresponden al último año.

- **Índice bursátil S&P 500:** el propio índice, Activision Blizzard, American Express, Amazon, American Water, Apple, Disney y eBay.
- **Índice bursátil IBEX 35:** Banco Sabadell, BBVA y Banco Santander.
- **Criptodivisas:** Bitcoin, Ethereum, Cardano, Polkadot y Ripple.
- **Divisas:** USD/JPY, EUR/JPY, GBP/USD.
- **Materias primas:** oro, algodón y café.

Se buscaron 5 grupos pues y se mostraron los *centroides* en la visualización, mostrada en la Figura 8.21.

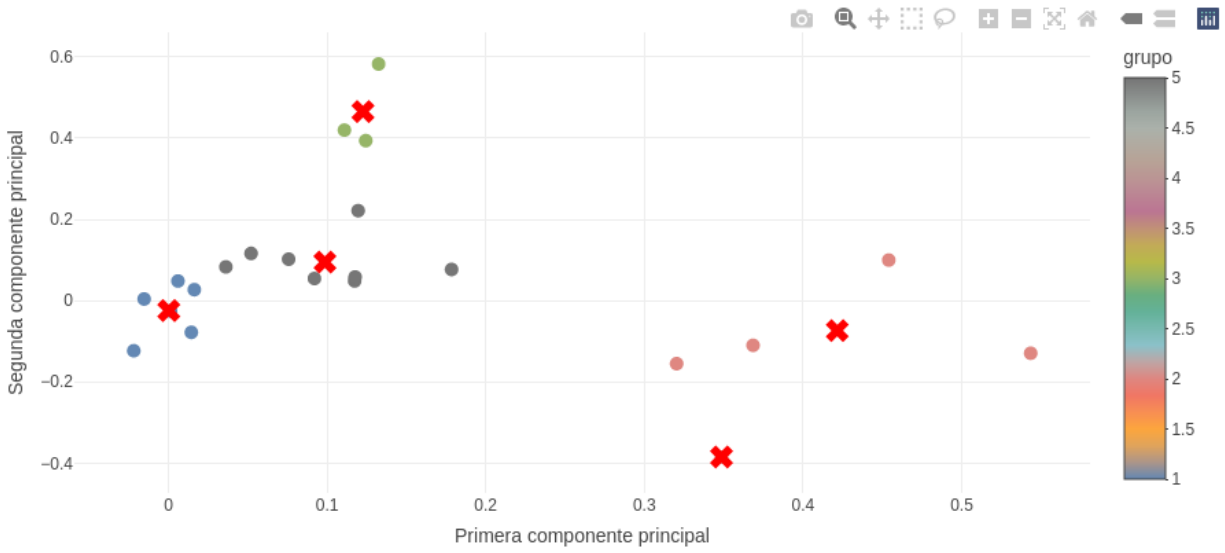


Figura 8.21: Resultado análisis *cluster* .

Vemos cómo se forman los 5 grupos deseados, aunque en vista de la distribución de los puntos, quizás los dos *clusters* de la derecha podrían haber formado uno únicamente. Esto a su vez se ve reforzado con el hecho de que todos los activos de esos dos *clusters* son las criptodivisas introducidas, siendo Ripple la que forma el *cluster* de abajo, el cual queda compuesto únicamente por dicho activo. En cuanto al resto de grupos formados, quizás pueda resultar más conveniente distinguir únicamente dos en lugar de tres, ya que los puntos no parecen mostrar una diferenciación clara en grupos. El conjunto superior (color verde) lo componen las tres empresas del IBEX 35. Esto era esperable, ya que además las tres pertenecen al sector de la banca, por lo que sus comportamientos en términos de rentabilidad se suponen semejantes. Los otros dos *clusters* formados son algo más heterogéneos. El de la izquierda de todo (color azul) está compuesto por las tres divisas introducidas, pero además cuenta con la empresa American Water y con el oro. Esto último puede tener cierta lógica ya que el oro en muchos casos es utilizado como activo refugio y muchas divisas están respaldadas en él (ver 3.5.1). El *cluster* restante (color negro) contiene el resto de empresas del S&P 500 y las materias primas restantes.

8.9 Ayuda

A modo de ejemplo se muestra una parte de la sección de ayuda en la Figura 8.22, correspondiente a las secciones de análisis de valores y de optimización de carteras. Vemos cómo se comenta brevemente el objetivo y funcionalidad de cada una, así como se enumeran algunas de las posibles condiciones de uso o configuración que puedan confundir inicialmente al usuario cuando se enfrenta a ellas.

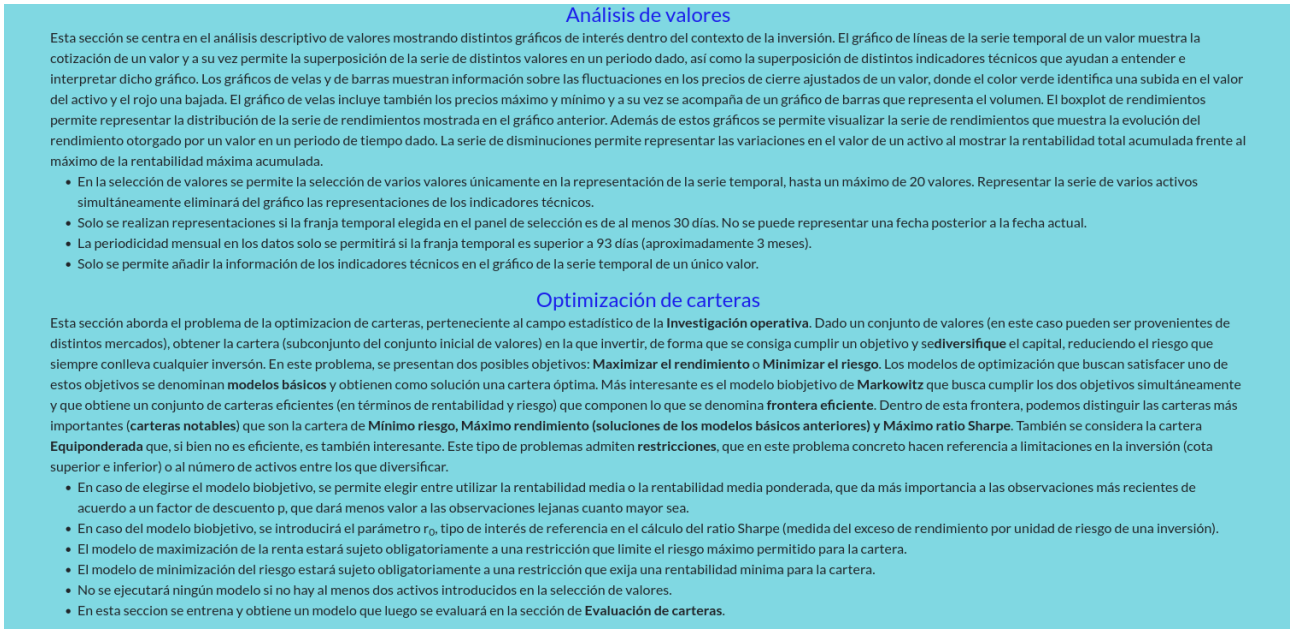


Figura 8.22: Extracto de la sección de ayuda.

8.10 Pruebas

Las pruebas realizadas para el sistema construido se han ido llevando a cabo durante todo el proceso de desarrollo. La metodología empleada ha consistido en ir realizando **pruebas de integración** de cada uno de los elementos que se iban añadiendo. Previamente, para comprobar que cada nuevo componente funcionaba del modo adecuado se realizaban **pruebas unitarias** a nivel local, donde se descargaba un ejemplo de datos como los que utilizaría la web y se comprobaba que el gráfico, tabla, modelo o cualquiera que fuera el tipo de componente que se estaba desarrollando, funcionaba correctamente. De este modo, el paso siguiente consistía en añadir la funcionalidad a la web y verificar su funcionamiento una vez integrada en el sistema. A modo documental, sin pretender extender en exceso esta parte, se muestra un ejemplo de cómo podrían formalizarse un par de pruebas de integración en este sistema:

ID	**Objetivo**	**Descripción**	**Resultado esperado**
1	Comprobar gráfico de frontera eficiente	Ejecutar modelo de Markowitz y seleccionar la segunda pestaña de resultados	Gráfico de la frontera eficiente correctamente realizado
2	Comprobar modelo Random Forest	Configurar y ejecutar modelo Random Forest	Todas las salidas del modelo muestran un resultado correcto

Cuadro 8.1: Ejemplo de pruebas de integración del sistema.

Parte IV

Discusión y Conclusiones

Capítulo 9

Discusión y Ampliaciones

Se comentan las futuras líneas de trabajo relacionadas con este proyecto, así como posibles mejoras del mismo.

9.1 Mejoras de la interfaz gráfica

En esta primera versión la interfaz gráfica resulta aceptable, pero hay diversos elementos que se podrían mejorar en una segunda versión:

1. El fondo de las distintas secciones tiene un color azul plano, podría incluirse una imagen diseñada que hiciera referencia a la temática del lugar y que diera una sensación más moderna.

2. El cuadro de texto con la descripción de cada sección podría ser un elemento que pudiera expandirse y ocultarse, de forma que no reste protagonismo al elemento shiny central.

3. La sección de ayuda podría implementarse de un modo más intuitivo, ya que lo que se muestra es idéntico en todas las secciones y esto puede complicar encontrar lo deseado.

4. El hecho de que sea posible (y el cómo hacerlo) seleccionar varios modelos de predicción de valores quizás no sea del todo intuitivo y podría mejorarse.

5. Sería interesante añadir la posibilidad de comparar carteras óptimas obtenidas con distintas configuraciones del sistema, del mismo modo que los distintos modelos se comparan.

6. Mejorar la extracción de datos para evitar los casos en los que pueda dar problemas. Esto se puede realizar tratando de investigar y gestionar manualmente el tema de las *cookies* en la conexión o tratando de utilizar otras fuentes de datos (o combinando distintas fuentes). Otras fuentes de datos financieros gratuitas son: Quandl, FRED, Tiingo, Bloomberg o Alpha Ventage.

7. Quizás separar la sección de evaluación de la de optimización no sea del todo útil y fuera mejor incluir la primera dentro de la segunda.

8. La forma de mostrar las características de los modelos de predicción construidos es muy básica y podría mejorarse. Quizás, esa pestaña podría suprimirse y la información de la construcción de los modelos se podría mostrar en la sección de comparación al hacer “hover” sobre cada uno de ellos.

9. Incluir en las distintas secciones de la web un botón que permita elegir al usuario en qué divisa desea visualizar la información (euros o dólares), ya que, pese a que lo habitual es tratar cada activo en la divisa de su país, puede resultar interesante a la hora de realizar comparaciones.

10. Mejorar la descarga de datos, ya que la descarga de datos desde la API de *Yahoo Finance* no acaba de ser del todo eficiente y falla en ocasiones, sobre todo cuando se realizan varias peticiones en un espacio breve de tiempo.

11. Podría ser más interesante que la sección de optimización tuviera un botón que indicara el inicio de los cálculos de modo análogo a las otras secciones, en lugar de mantener la reactividad en todo momento mientras se configura el modelo.

9.2 Mejoras del código

La calidad del código es algo siempre a cuidar en un proyecto *software*. Un aspecto clave que se podría mejorar en este caso es el de los elementos shiny, ya que R no es un lenguaje pensado para la computación de ese tipo de sistemas. En [127] se muestran algunas posibles ideas relativas a este punto.

9.3 Comparativa Python - R

Debido a que ambos son los lenguajes más populares en ciencia de datos, podría ser interesante llevar a cabo las distintas funcionalidades de la aplicación en uno u otro lenguaje en función de la elección del usuario, así como en ambos de manera simultánea. Esto debería ir acompañado de alguna herramienta que permita comparar los resultados en términos de recursos, eficiencia y eficacia.

9.4 Evaluación del rendimiento de la web

Debido a que, como se ha comentado anteriormente, los elementos shiny no trabajan del modo deseado en términos de velocidad, sería interesante incluir formas de medir dicho rendimiento y de ver en qué se gasta el tiempo (generación de modelos, ejecución, generación de la visualización, ...). Del mismo modo, es de interés comparar los modelos de predicción en términos de eficiencia, no solo en cuanto a la evaluación de sus predicciones. Más sobre esto se comenta en [128].

9.5 Registro de usuarios

Permitir el registro de usuarios en la web sería una funcionalidad más que interesante, permitiendo a estos guardar sus análisis, comparar los análisis realizados, gestionar carteras, evaluar el rendimiento de carteras guardadas, evaluar el rendimiento de modelos construidos, etc.

9.6 Modelo de *Black-Litterman*

En [129] se comenta el modelo de Black-Litterman como alternativa de mejora al modelo de Markowitz trabajado en este proyecto. Podría ser interesante tanto construir una aplicación centrada en él, como más aún incluirlo en esta y poder compararlo con el modelo de Markowitz.

9.7 Nuevos modelos de IA

Pese a que la web ya contiene una buena colección de modelos de predicción diferentes, pueden incluirse aún muchos más. Algoritmos de *ensembles* como del *Adaboost*, algoritmos clásicos como *Naive Bayes* o *K Nearest Neighbors*, o distintos tipos de redes neuronales como las *Restricted Boltzmann Machine*, las *Generative Adversarial Networks*, las *Deep Belief Networks* o las *Deep Q-Network* pueden ser incluidos y sería de gran interés, del mismo modo que modelos que creen sistemas de redes neuronales más complejos mediante combinación, nuevos mecanismos (como el *attention mechanism* de las redes recurrentes) o *ensembles*. Sin embargo, modelos que introduzcan el análisis de sentimientos (*sentiment analysis*) resultan de especial interés, al tratar de incluir aspectos del análisis fundamental en los modelos. Estos modelos, generalmente se basan en el procesamiento de textos de noticias o, más frecuentemente, de comentarios de usuarios en redes sociales (especialmente *Twitter*). Algunos ejemplos se comentan en [130], [131] y [132].

Otro método que podría incluirse y que destaca por su interés en la predicción de series temporales es el algoritmo *Prophet*, desarrollado por *Facebook* y detallado en [133].

9.8 Aumento de la personalización

Aumento de las características y parámetros que pueden configurarse en los modelos de predicción de valores, así como aumentar la personalización permitida en las distintas visualizaciones realizadas. También el tema de la selección de activos en las distintas secciones de la web se podría personalizar de mayor manera, por ejemplo, agrupando activos por sectores, como es habitual en índices como el IBEX 35 o el S&P 500. Se podrían añadir otros mercados bursátiles, como los mercados orientales, más desconocidos en nuestro mundo occidental pero enorme interés en la economía mundial

9.9 Profundizar en el tema de *clustering*

El tema del análisis *cluster* se toca con muy poca profundidad en este trabajo y es un tema a explorar más en detalle en el futuro con el fin de ver qué puede aportar este tipo de estudios al análisis de valores de activos financieros.

9.10 Creación de un *bot* asistente

El estudio de los distintos modelos de inteligencia artificial aplicados a la predicción de valores puede enfocarse con otras perspectiva diferente si se plantea la construcción de un bot asistente que se encargue de recomendar inversiones, ventas o estrategias de actuación en base a los resultados que obtenga con los modelos que ejecute. Un enfoque incluso más ambicioso sería el de construir un *bot* que directamente haga las operaciones por el usuario en base a los resultados que obtenga.

Capítulo 10

Conclusiones

La principal conclusión de este proyecto es la consecución de los objetivos planteados inicialmente en 1.3, ya que se ha conseguido desarrollar una aplicación web que añada las mejoras o ampliaciones de los trabajos realizados anteriormente sobre la misma temática y a que se han conseguido implementar todas las funcionalidades que inicialmente se establecieron como requisitos del proyecto.

En cuanto a las tecnologías utilizadas, cabe mencionar el aprendizaje significativo experimentado por el desarrollador en cuanto a las tecnologías básicas del desarrollo web, así como acerca de los distintos problemas abordados en las secciones de la aplicación. También cabe mencionar el aumento en conocimientos sobre librerías de Python, el aprendizaje de AMPL como *solver* y de librerías de R enfocadas al análisis financiero, así como al aumento de conocimientos generales sobre este mundo.

Relacionado con lo dicho anteriormente, un punto clave de este proyecto era el aprendizaje y exploración de la librería shiny de R, el cual ha resultado exitoso lográndose cumplir todos los objetivos propuestos con un resultado más que aceptable. Sin embargo, incrustar elementos shiny en una página web parece no haber resultado del todo eficiente, ya que los tiempos de carga de las distintas secciones de la web no son inmediatos como cabría desear, sino que llevan algunos segundos. Del mismo modo, la ejecución de algunos de los métodos desarrollados puede ser un poco lenta, lo que lleva a cuestionar si realmente shiny es un método ideal para implementar sistemas de este estilo, donde se busque un tiempo de respuesta rápido y un control visual adecuado al mundo web y de los dispositivos móviles. Quizás sea interesante la utilización de las aplicaciones shiny en otro tipo de contextos, donde estos aspectos no jueguen un papel tan clave. Un último aspecto a cuestionar en cuanto a las tecnologías utilizadas en el proyecto es el hecho de que el acceso a los datos a través de la API de *Yahoo Finance* no es eficiente y, si se realizan varias peticiones muy seguidas, da problemas (en principio, relacionado con el uso de las *cookies* en la conexión). Esto hace que no sea eficiente utilizar este sistema de obtención de datos en una aplicación que busca respuestas rápidas.

Sin embargo, la página web desarrollada no deja de ser una primera versión completa y funcional marcada por el *deadline* del Trabajo de Fin de Grado, y podrían aplicarse numerosos cambios de cara a mejorar tanto sus prestaciones actuales como a incluir nuevos elementos.

Bibliografía

[1] corine solutions. *UX tools in Artificial Neural Network Design*. https://corinne.solutions/2018/10/03/ux-tools-artificial-neural-network-design/. [Online; accessed 26-May-2022].

[2] Harry Markowitz. *Portfolio Selection.* Vol. 7, No. 1, pp. 77-91. The Journal of Finance, 1952.

[3] César Hernández. «Predicción y Clasificación de series temporales bursátiles mediante Redes Neuronales Recurrentes». En: *Universidad de Valladolid* (2020).

[4] César Hernández. «Optimización de carteras de inversión en la industria tecnológica». En: *Universidad de Valladolid* (2020).

[5] Sonia Aparicio. «Aplicación Shiny para la optimización de carteras de inversión». En: *Universidad de Valladolid* (2020).

[6] Víctor Arranz. «Optimización interactiva de carteras de inversión». En: *Universidad de Valladolid* (2022).

[7] Banco Santander. *Metodologías de desarrollo de software: ¿qué son?* https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html. [Online; accessed 1-Mar-2022].

[8] OpenWebinars. *Qué es un Sprint de Scrum.* https://openwebinars.net/blog/que-es-un-sprint-scrum/. [Online; accessed 1-Mar-2022].

[9] Jobted. *Sueldo del Ingeniero Informático en España*. https://www.jobted.es/salario/ingeniero-inform%C3%A1tico. [Online; accessed 1-Mar-2022].

[10] Nius diario. *La luz en modo montaña rusa y lo que nos espera este 2022*. https://www.niusdiario.es/economia/consumo/precio-luz-gas-2022-previsiones_18_3258948467.html. [Online; accessed 1-Mar-2022].

[11] epdata. *Precio de la factura de la luz, datos y estadísticas*. https://www.epdata.es/datos/precio-factura-luz-datos-estadisticas/594?accion=2. [Online; accessed 02-Jun-2022].

[12] Bog Hughes y Mike Cotterel. *Software Project Management.* Fifth Edition. Mc Graw-Hill Education, 2009. ISBN: 978-0-07-7122799.

[13] EAE Business School. *Guía PMBOK: definición, estructura y tips de estudio*. https://retos-operaciones-logistica.eae.es/que-es-la-guia-pmbok-y-como-influye-en-la-administracion-de-proyectos/. [Online; accessed 1-Mar-2022].

[14] J. Bogle. *Finanzas personales.* First Edition. New York: Editorial Willey, 2007.

[15] Marcial Córdoba Padilla. *Mercado de valores.* Primera Edición. Instituto Mexicano de Contadores Públicos., 2020. ISBN: 9786075630243.

[16] BBVA. *Activos financieros, ¿qué son?* https://www.bbva.es/finanzas-vistazo/ef/fondos-inversion/activos-financieros.html. [Online; accessed 2-Mar-2022].

[17] Alejandro Scherk. *Manual de Análisis Fundamental*. Quinta Edición. INVERSOR EDICIONES, S.L., 2010. ISBN: 978-84-15304-01-2.

[18] Isaac Kofi Nti, Adebayo Felix Adekoya y Benjamin Asubam Weyori. «A systematic review of fundamental and technical analysis of stock market predictions». En: *Artificial Intelligence Review* 53 (4 abr. de 2020), págs. 3007-3057. ISSN: 15737462. DOI: 10.1007/s10462-019-09754-z.

[19] Dev Shah, Haruna Isah y Farhana Zulkernine. «Stock market analysis: A review and taxonomy of prediction techniques». En: *International Journal of Financial Studies* 7 (2 2019). ISSN: 22277072. DOI: 10.3390/ijfs7020026.

[20] Rankia. *Medias móvil simple, exponencial y ponderada: formulas y ejemplos* . https://www.rankia.cl/blog/analisis-ipsa/2039072-medias-movil-simple-exponencial-ponderada-formulas-ejemplos. [Online; accessed 2-Mar-2022].

[21] Admiral markets. *¿Cómo operar con las Bandas de Bollinger?* https://admiralmarkets.com/es/education/articles/forex-strategy/bandas-de-bollinger. [Online; accessed 2-Mar-2022].

[22] Enbolsa. *Análisis e Indicadores Técnicos. ¿Qué son y cómo utilizarlos?* https://www.enbolsa.net/analisis-tecnico/. [Online; accessed 2-Mar-2022].

[23] Avatrade. *ROC trading strategies*. https://www.avatrade.es/educacion/professional-trading-strategies/macd-trading-strategies. [Online; accessed 2-Mar-2022].

[24] Avatrade. *Accumulation/Distribution Indicator (A/D)* . https://www.investopedia.com/terms/a/accumulationdistribution.asp. [Online; accessed 2-Mar-2022].

[25] StockCharts. *%B indicator*. https://school.stockcharts.com/doku.php?id=technical_indicators:bollinger_band_perce. [Online; accessed 2-Mar-2022].

[26] Investopedia. *On-Balance Volume (OBV) Definition*. https://www.investopedia.com/terms/o/onbalancevolume.asp. [Online; accessed 2-Mar-2022].

[27] Investopedia. *Average True Range (ATR)*. https://www.investopedia.com/terms/a/atr.asp. [Online; accessed 2-Mar-2022].

[28] Economipedia. *Indicador ADX (Average Directional Index)*. https://economipedia.com/definiciones/indicador-adx-average-directional-index.html. [Online; accessed 2-Mar-2022].

[29] John J. Murphy. *Análisis técnico de los mercados financieros*. Primera Edición. Gestión 2000, S.A., 2000. ISBN: 84-8088-442-8.

[30] Wikipedia. *Teoría de Dow*. https://es.wikipedia.org/wiki/Teor%C3%ADa_de_Dow. [Online; accessed 06-Jun-2022].

[31] Ámbito financiero. *Tipos de gráficos más usados en el análisis técnico bursátil*. https://ambito-financiero.com/tipos-de-graficos-mas-usados-analisis-tecnico-bursatil/. [Online; accessed 2-Mar-2022].

[32] Expansión. *Averiguar las tendencias del mercado*. https://www.expansion.com/mercados/curso-invertir-bolsa/averiguar-tendencias-mercado.html. [Online; accessed 2-Mar-2022].

[33] Wikipedia. *Bolsa de valores*. https://es.wikipedia.org/wiki/Bolsa_de_valores. [Online; accessed 4-Mar-2022].

[34] Economipedia. *Bolsa de valores*. https://economipedia.com/definiciones/bolsa-de-valores.html. [Online; accessed 4-Mar-2022].

[35] Wikipedia. *S&P 500*. https://es.wikipedia.org/wiki/S%26P_500. [Online; accessed 4-Mar-2022].

[36] gualestrit. *Los 11 sectores del S&P 500 y sus ETFs*. https://www.gualestrit.com/los-11-sectores-del-sp-500-y-sus-etfs/. [Online; accessed 4-Mar-2022].

[37] Wikipedia. *Nasdaq*. https://es.wikipedia.org/wiki/NASDAQ#Nasdaq-100. [Online; accessed 4-Mar-2022].

[38] Wikipedia. *Ibex 35*. https://es.wikipedia.org/wiki/IBEX_35. [Online; accessed 4-Mar-2022].

[39] BME renta variable. *Listado de empresas y sectores*. https://www.bmerv.es/esp/aspx/Empresas/EmpresasPorSectores.aspx?sector=05. [Online; accessed 4-Mar-2022].

[40] Wikipedia. *Stoxx Europe 600*. https://es.wikipedia.org/wiki/EURO_STOXX_50. [Online; accessed 13-Mar-2022].

[41] Bit2me Academy. *¿Qué es una criptomoneda?* https://academy.bit2me.com/que-es-una-criptomoneda/. [Online; accessed 4-Mar-2022].

[42] Coinbase. *What can you do with market cap?* https://www.coinbase.com/es/learn/crypto-basics/what-is-market-cap. [Online; accessed 14-Mar-2022].

[43] Xataka. *Qué es blockchain: la explicación definitiva para la tecnología más de moda*. https://www.xataka.com/especiales/que-es-blockchain-la-explicacion-definitiva-para-la-tecnologia-mas-de-moda. [Online; accessed 4-Mar-2022].

[44] Bit2me Academy. *¿Qué es Bitcoin (BTC)?* https://academy.bit2me.com/que-es-bitcoin-btc-criptomoneda/. [Online; accessed 5-Mar-2022].

[45] Wikipedia. *Bitcoin*. https://es.wikipedia.org/wiki/Bitcoin#Acontecimientos_notables. [Online; accessed 5-Mar-2022].

[46] Wikipedia. *Ethereum*. https://es.wikipedia.org/wiki/Ethereum. [Online; accessed 5-Mar-2022].

[47] vozpópuli. *El mercado de materias primas*. https://www.estrategiasdeinversion.com/herramientas/diccionario/mercados/mercado-de-materias-primas-t-1510. [Online; accessed 5-Feb-2022].

[48] Wikipedia. *Mercado de materias primas*. https://es.wikipedia.org/wiki/Mercado_de_materias_primas. [Online; accessed 5-Mar-2022].

[49] Wikipedia. *El oro como activo financiero*. https://www.finanzasclaras.es/el-oro-como-activo-financiero/. [Online; accessed 5-Mar-2022].

[50] Valora analytic. *¿Por qué el oro funciona como activo refugio?* https://www.valoraanalitik.com/2021/03/15/por-que-el-oro-funciona-como-activo-refugio/. [Online; accessed 5-Mar-2022].

[51] Wikipedia. *Mercado de divisas*. https://es.wikipedia.org/wiki/Mercado_de_divisas. [Online; accessed 7-Mar-2022].

[52] BBVA. *El mercado de divisas: ¿Qué es y cómo funciona?* https://www.bbva.com/es/mercado-divisas-que-es-como-funciona/. [Online; accessed 7-Mar-2022].

[53] Robert J.Vanderbei. *Linear Programming Foundations and Extensions*. Second Edition. Kluwer A.P, 1997.

[54] Selfbank'. *¿Qué es el ratio de Sharpe?* https://www.selfbank.es/centro-de-ayuda/fondos-de-inversion/que-es-el-ratio-de-sharpe. [Online; accessed 19-May-2022].

[55] Rankia'. *Portafolios de inversión: ¿Qué es la frontera eficiente?* https://www.rankia.mx/blog/como-comenzar-invertir-bolsa/3307417-portafolios-inversion-que-frontera-eficiente. [Online; accessed 19-May-2022].

[56] Afi Inversiones Globales'. *Beta: medida del riesgo sistemático*. https://www.afi-inversiones.es/b61.html. [Online; accessed 20-May-2022].

[57] Wikipedia. *Método de Montecarlo*. https://es.wikipedia.org/wiki/M%C3%A9todo_de_Montecarlo. [Online; accessed 20-May-2022].

[58] Bernard Brenyah. *Efficient Frontier & Portfolio Optimization with Python [Part 2/2]*. https://medium.com/python-data/efficient-frontier-portfolio-optimization-with-python-part-2-2-2fe23413ad94. [Online; accessed 20-May-2022].

[59] IArtificial.net. *¿Clasificación o Regresión?* https://www.iartificial.net/clasificacion-o-regresion/. [Online; accessed 11-Mar-2022].

[60] Rankia. *Análisis bursátil; ¿Qué es un precio de cierre ajustado?* https://www.rankia.mx/blog/como-comenzar-invertir-bolsa/4320061-analisis-bursatil-que-precio-cierre-ajustado. [Online; accessed 9-Mar-2022].

[61] Jason Brownlee. *How to Convert a Time Series to a Supervised Learning Problem in Python*. https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/. [Online; accessed 03-May-2022].

[62] Santiago de la Fuente Fernández. «Facultad Ciencias Económicas y Empresariales Departamento de Economía Aplicada». En: ().

[63] I. Santamaría S. Van Vaerenbergh. «Métodos kernel para clasificación». En: (2018).

[64] Tom Sharp. *An Introduction to Support Vector Regression (SVR)*. https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2. [Online; accessed 03-May-2022].

[65] M Nabipour y col. «Deep Learning for Stock Market Prediction». En: *Entropy (Basel, Switzerland)* 22 (8 2020), pág. 840. ISSN: 1099-4300. DOI: 10.3390/e22080840.

[66] Juan Bosco Mendoza Vega. *Tutorial: XGBoost en Python*. https://medium.com/@jboscomendoza/tutorial-xgboost-en-python-53e48fc58f73. [Online; accessed 21-May-2022].

[67] Khan Academy. *Descenso del gradiente*. https://es.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/optimizing-multivariable-functions/a/what-is-gradient-descent#:~:text=El%20descenso%20de%20gradiente%20es%20un%20algoritmo%20que%20estima%20num%C3%A9ricamente,0%20como%20hemos%20visto%20antes.. [Online; accessed 21-May-2022].

[68] Gabriel Tseng. *Gradient Boosting and XGBoost*. https://medium.com/@gabrieltseng/gradient-boosting-and-xgboost-c306c1bcfaf5. [Online; accessed 21-May-2022].

[69] Teodoro Calonge Cano. *Apuntes de la asignatura Técnicas de Aprendizaje Automático*. 2021.

[70] Diego Calvo. *Función de activación - Redes neuronales*. https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/. [Online; accessed 24-May-2022].

[71] David E. Rumelhart y James L. McClelland. «Learning Internal Representations by Error Propagation». En: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. 1987, págs. 318-362.

[72] The Science o Machine learning. *Backpropagation*. https://www.ml-science.com/backpropagation. [Online; accessed 24-May-2022].

[73] Diederik P Kingma y Jimmy Ba. «Adam: A Method for Stochastic Optimization». eng. En: (2014).

[74] Jason Brownlee. *Loss and Loss Functions for Training Deep Learning Neural Networks*. https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/. [Online; accessed 25-May-2022].

[75] Vicente Rodríguez. *Dropout y Batch normalization*. https://vincentblog.xyz/posts/dropout-y-batch-normalization. [Online; accessed 25-May-2022].

[76] David E. Rumelhart, Geoffrey E. Hinton y Ronald J. Williams. «Learning representations by back-propagating errors». En: *Nature* 323 (1986), págs. 533-536.

[77] Jordi Torres.AI. *Redes neuronales recurrentes*. https://torres.ai/redes-neuronales-recurrentes/. [Online; accessed 25-May-2022].

[78] Data Science. *Una introducción a las redes neuronales recurrentes*. https://datascience.eu/es/aprendizaje-automatico/una-introduccion-a-las-redes-neuronales-recurrentes/. [Online; accessed 25-May-2022].

[79] Qing Li y col. «A Multimodal Event-Driven LSTM Model for Stock Prediction Using Online News». En: *IEEE transactions on knowledge and data engineering* 33 (10 2021), págs. 3323-3337. ISSN: 1041-4347. DOI: 10.1109/TKDE.2020.2968894.

[80] Alejandro Casallas García. *Llenado de series de datos de 2014 a 2019 de PM2.5 por medio de una red neuronal y una regresión lineal*. https://www.researchgate.net/profile/Alejandro-Casallas-Garcia/publication/343450066_Llenado_de_series_de_datos_de_2014_a_2019_de_PM25_por_medio_de_una_red_neuronal_y_una_regresion_lineal/links/5f2ad695299bf13404a5ac30/Llenado-de-series-de-datos-de-2014-a-2019-de-PM25-por-medio-de-una-red-neuronal-y-una-regresion-lineal.pdf?_sg%5B0%5D=xXhGFKT7x6oOfqOlSIovsqban6dCa8tCSU-CtTfe6taLwRLi5UD-zsMwERSniddTzEg7e-kM_9gDpr96ryqlwA.PZl-guXLeXgCyXqV7qTXYf0zwlmu7Uu7EvzCWoRIFuZhQcMpdlhX6y7Hqbl3x5ayH_mARDq4ZgvZXcGIQ-Q6FA&_sg%5B1%5D=2FBubXwXROKmNITEAXs3MOkKReuI-3fM8iBVhufOamQrypXkEtLNsX8sZoNKS85Ch0Sm9OlHpLZCXAjs9Y2kpYmbk55DZ6nT6H2cF8s6yjJT.PZl-guXLeXgCyXqV7qTXYf0zwlmu7Uu7EvzCWoRIFuZhQcMpdlhX6y7Hqbl3x5ayH_mARDq4ZgvZXcGIQ-Q6FA&_iepl=. [Online; accessed 25-May-2022].

[81] Vijaysinh Lendave. *LSTM Vs GRU in Recurrent Neural Network: A Comparative Study*. https://analyticsindiamag.com/lstm-vs-gru-in-recurrent-neural-network-a-comparative-study/. [Online; accessed 25-May-2022].

[82] Teodoro Calonge Cano. *Apuntes de la asignatura Minería de datos*. 2022.

[83] Rikiya Yamashita y col. «Convolutional neural networks: an overview and application in radiology». En: *Insights into imaging* 9 (4 2018), págs. 611-629. ISSN: 1869-4101. DOI: 10.1007/s13244-018-0639-9.

[84] Data Scientest. *Convolutional Neural Network*. https://datascientest.com/es/convolutional-neural-network-es#maxpooling. [Online; accessed 25-May-2022].

[85] coderz.py. *CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more*. https://coderzpy.com/cnn-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more/. [Online; accessed 25-May-2022].

[86] Rodrigo Araujo. «A Semi-Supervised Approach for Kernel-Based Temporal Clustering». En: 2015.

[87] gamco. *Clustering para analizar el dato*. https://gamco.es/clustering-para-analizar-el-dato/. [Online; accessed 27-May-2022].

[88] Luis Ángel García y Miguel Alejandro Fernández. *Apuntes de la asignatura del grado en Estadística Análisis de Datos*. 2020.

[89] Linh Ngo. *Principal component analysis explained simply*. https://blog.bioturing.com/2018/06/14/principal-component-analysis-explained-simply/. [Online; accessed 27-May-2022].

[90] Yahoo Finance. https://es.finance.yahoo.com/. [Online; accessed 14-Mar-2022].

[91] business science. *tidyquant*. https://business-science.github.io/tidyquant/. [Online; accessed 31-May-2022].

[92] Xataka. *API: qué es y para qué sirve*. https://www.xataka.com/basics/api-que-sirve. [Online; accessed 14-Mar-2022].

[93] Wikpedia. *cron*. https://en.wikipedia.org/wiki/Cron. [Online; accessed 02-Jun-2022].

[94] Javier Gatón. «ADevTest: Herramienta para la evaluación de conocimientos de programación». En: *Universidad de Valladolid* (2021).

[95] *Escribiendo historias de usuario*. https://scrum.mx/informate/historias-de-usuario. [Online; accessed 13-May-2022].

[96] creately. *La Guía Fácil de los Diagramas de Despliegue UML*. https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/. [Online; accessed 26-May-2022].

[97] https://www.docker.com/. [Online; accessed 24-May-2022].

[98] Wikipedia. *Docker (software)*. https://es.wikipedia.org/wiki/Docker_(software). [Online; accessed 24-May-2022].

[99] Docker. *Dockerfile reference*. https://docs.docker.com/engine/reference/builder/. [Online; accessed 24-May-2022].

[100] Cloud native wiki. *Docker Architecture*. https://www.aquasec.com/cloud-native-academy/docker-container/docker-architecture/. [Online; accessed 24-May-2022].

[101] https://www.parsehub.com/. [Online; accessed 06-Mar-2022].

[102] https://www.mockflow.com/. [Online; accessed 20-May-2022].

[103] mozilla.org. *HTML: Lenguaje de etiquetas de hipertexto*. https://developer.mozilla.org/es/docs/Web/HTML. [Online; accessed 31-May-2022].

[104] Wikipedia. *Bootstrap (framework)*. https://es.wikipedia.org/wiki/Bootstrap_(framework). [Online; accessed 31-May-2022].

[105] Wikpedia. *Javascript*. https://es.wikipedia.org/wiki/JavaScript. [Online; accessed 31-May-2022].

[106] Wikpedia. *Javascript*. https://es.wikipedia.org/wiki/JavaScript. [Online; accessed 31-May-2022].

[107] Wikpedia. *jQuery*. https://es.wikipedia.org/wiki/JQuery. [Online; accessed 31-May-2022].

[108] tidyverse.org. *Tidyverse*. https://www.tidyverse.org/. [Online; accessed 31-May-2022].

[109] Rstudio. *Shiny*. https://shiny.rstudio.com/. [Online; accessed 31-May-2022].

[110] Peter Solymos. *The Anatomy of a Shiny Application*. https://www.r-bloggers.com/2021/04/the-anatomy-of-a-shiny-application/. [Online; accessed 31-May-2022].

[111] plotly. *tPlotly R Open Source Graphing Library*. https://plotly.com/r/. [Online; accessed 31-May-2022].

[112] Rstudio. *R interface to Python*. https://rstudio.github.io/reticulate/. [Online; accessed 31-May-2022].

[113] AMPL. *AMPL R API*. https://rampl.readthedocs.io/en/latest/. [Online; accessed 31-May-2022].

[114] Becas Santander. *Python: qué es y por qué deberías aprender a utilizarlo*. https://www.becas-santander.com/es/blog/python-que-es.html. [Online; accessed 31-May-2022].

[115] Numpy.org. *Numpy*. https://numpy.org/. [Online; accessed 31-May-2022].

[116] aprendeconalf.es. *La librería Pandas*. https://aprendeconalf.es/docencia/python/manual/pandas/. [Online; accessed 31-May-2022].

[117] Javier Buhigas. *Todo lo que necesitas saber sobre TensorFlow, la plataforma para Inteligencia Artificial de Google*. https://puentesdigitales.com/2018/02/14/todo-lo-que-necesitas-saber-sobre-tensorflow-la-plataforma-para-inteligencia-artificial-de-google/. [Online; accessed 31-May-2022].

[118] Wikipedia. *Keras*. https://es.wikipedia.org/wiki/Keras. [Online; accessed 31-May-2022].

[119] Scikit-learn. *scikit-learn*. https://scikit-learn.org/stable/. [Online; accessed 31-May-2022].

[120] dmlc XGBoost. *XGBoost Documentation*. https://xgboost.readthedocs.io/en/stable/. [Online; accessed 31-May-2022].

[121] statsmodels.org. *statsmodels*. https://www.statsmodels.org/stable/index.html. [Online; accessed 31-May-2022].

[122] AMPL. *AMPL*. https://ampl.com/. [Online; accessed 31-May-2022].

[123] Wikpedia. *Visual Paradigm*. https://en.wikipedia.org/wiki/Visual_Paradigm. [Online; accessed 31-May-2022].

[124] gitlab. *What is gitlab?* https://about.gitlab.com/what-is-gitlab/. [Online; accessed 31-May-2022].

[125] IBM. *IBM ILOG CPLEX Optimizer*. https://www.ibm.com/analytics/cplex-optimizer. [Online; accessed 01-Jun-2022].

[126] Wikipedia. *Algoitmo símplex*. https://es.wikipedia.org/wiki/Algoritmo_s%C3%ADmplex. [Online; accessed 01-Jun-2022].

[127] Krystian Igras. *4 Tips to Make Your Shiny Dashboard Faster*. https://www.rstudio.com/blog/4-tips-to-make-your-shiny-dashboard-faster/. [Online; accessed 01-Jun-2022].

[128] Garrett Grolemund. *Faster Shiny Apps with profiling tools*. https://github.com/rstudio/ShinyDeveloperConference/blob/master/Profiling/shiny-profiling.pdf. [Online; accessed 01-Jun-2022].

[129] Luis C. Franco-Arbeláez, Claudia T. Avendaño-Rúa y Haroldo Barbutín-Díaz. «Modelo de markowitz y modelo de Black-Litterman en la optimización de portafolios de inversión». En: *Tecno - Lógicas (Instituto Tecnológico Metropolitano)* 26 (26 2011), págs. 71-88. ISSN: 0123-7799. DOI: 10.22430/22565337.40.

[130] Mingqin Chen y col. «A Quantitative Investment Model Based on Random Forest and Sentiment Analysis». En: *Journal of physics. Conference series* 1575 (1 2020), pág. 12083. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1575/1/012083.

[131] Weiling Chen y col. «Leveraging social media news to predict stock index movement using RNN-boost». En: *Data and Knowledge Engineering* 118 (nov. de 2018), págs. 14-24. ISSN: 0169023X. DOI: 10.1016/j.datak.2018.08.003.

[132] Jinghua Tan y col. «A tensor-based eLSTM model to predict stock price using financial news». En: vol. 2019-January. IEEE Computer Society, 2019, págs. 1658-1667. ISBN: 9780998133126. DOI: 10.24251/hicss.2019.201.

[133] Sean J. Taylor y Benjamin Letham. «Forecasting at Scale». En: *The American statistician* 72 (1 2018), págs. 37-45. ISSN: 0003-1305. DOI: 10.1080/00031305.2017.1380080.

[134] *Visual Studio Code*. https://code.visualstudio.com/. [Online; accessed 31-May-2022].

Apéndice A

Manual de Usuario

En primer lugar se presentará la dirección de la web construida y posteriormente se explicarán el modo de llevar a cabo las principales acciones disponibles para el usuario.

A.1 Web

La dirección de la web es `http://virtual.lab.inf.uva.es:20222/` y el código se encuentra alojado en mi gitlab en `https://gitlab.inf.uva.es/victarr/tfg_informatica` en la carpeta code.

A.2 Acciones del usuario

Las acciones a llevar a cabo por el usuario en este sitio web son muy sencillas, ya que no requiere de ningún tipo de registro para su utilización.

- El modo de moverse entre las distintas secciones de la web es mediante el menú lateral comentado en 6.3.2. Haciendo *click* sobre la sección de mercados se accede a un *dashboard* que contiene tablas con los valores de los distintos mercados contemplados en la aplicación. Por otro lado, pasando el ratón dicho elemento en el menú se desplegará el submenú de la misma que permite acceder a la sección particular de cada mercado, pinchando en cada una de ellas.

- Para acceder a la sección de ayuda basta con pulsar el botón de ayuda ubicado en la parte superior derecha de todas las ventanas. El menú de navegación sigue disponible en esta parte de la web, por lo que la navegación permanece inalterada.

- La sección de evaluación de carteras solo estará disponible tras acceder inmediatamente antes a la sección de optimización. Si pese a acceder a ella, no se genera ninguna cartera, se le notifica al usuario cuando procede a evaluar.

- En todo momento el enlace al repositorio gitlab donde se alojan todos los archivos del proyecto se encuentra disponible en la parte derecha del pie de página.

- La interacción y configuración de las diferentes shiny apps no se detalla en mayor medida, ya que los *shiny widgets* son suficientemente intuitivos y autodescriptivos. Mencionar que las secciones de simulación

de carteras, regresiones de índices, análisis de grupos y modelos de predicción requieren de pulsar el botón inferior “Ejecutar modelo” para que lleven a cabo sus acciones tras modificar la configuración. La última de ellas, permite incluir tantos modelos como se desee, pulsando este botón tras establecer las características de cada uno de ellos.

Apéndice B

Manual del Desarrollador

En este capítulo se expondrá un breve comentario sobre cómo se ha trabajado a nivel de desarrollo con distintos aspectos del presente proyecto, así una explicación sobre cómo modificar el sistema desarrollado para ampliar las funcionalidades, corregir errores o incluir cualquier tipo de elemento o mejora.

En primer lugar, comentar que la aplicación corre dentro de un contenedor docker sobre *shiny server*, que proporciona todos los elementos necesarios para correr una aplicación web interactiva que contenga *shiny apps*, como es el caso de la aplicación que se ha construido.

Este sistema, crea una carpeta que tiene un *node* (que es un entorno en tiempo de ejecución de Javascript), que dentro del contenedor se encuentra en `/opt/shiny-server/ext`). Dicho node se lanza con una *app llamada shiny-server* que está en `/opt/shiny-server/lib` donde hay un fichero *main.js*. Por tanto, hay un único servidor que es *shiny server* que es el que interpreta los archivos .R que son locales y es capaz de lanzar el intérprete que necesita para ejecutarlos.

B.1 Máquina virtual

De cara a trabajar del modo más organizado y eficiente posible, se trató de organizar la máquina virtual de un modo inteligente. Para ello, en el directorio `$HOME` del usuario del alumno (victor), se crearon los siguientes directorios:

- **/bin:** en este directorio se incluyen los archivos ejecutables creados para facilitar y agilizar diferentes tareas.
  - *dcclean:* elimina todos los contenedores de una instancia de docker.
  - *dmybuild:* realiza el `build` de una imagen docker especificada.
  - *shiny-start:* lanza el servidor web y redirecciona las carpetas de la aplicación a sus correspondientes direcciones del contenedor docker.
- **/my-dockers:** directorio donde se guardan los directorios de las distintas imágenes docker construidas a lo largo del proyecto:
  - */r-hola:* imagen muy básica construida sobre *r-base* que únicamente ejecuta un “hola mundo!”. El objetivo de la misma fue probar el funcionamiento de docker y empezar a familiarizarse con las imágenes docker y con los ficheros *Dockerfile*.

- */r-python-prueba:* imagen utilizada para probar la utilización de python desde R a través de la librería *reticulate* (ver 7.2.5), con un script de prueba que a su vez utilizaba algunas de las librerías python que luego se necesitarían para la sección de modelos de predicción.
- */r-ampl-prueba:* imagen utilizada para probar la utilización del solver AMPL desde R a través de la librería *rAMPL* (ver 7.2.5), con un script de prueba que a su vez implementaba un modelo de Markowitz (ver 3.7.2) de optimización de carteras que luego se necesitaría para la sección de optimización de carteras (ver 3.7).
- */shiny-prueba:* imagen utilizada para probar la librería shiny (ver 7.2.5) de R ejecutando para ello los *scripts* de ejemplo ubicados en el directorio /my-shiny y referenciados mediante enlaces simbólicos (vienen siendo algo semejante a los accesos directos de *Windows*). Se corrige el *Dockerfile* original del directorio my-shiny debido a que directamente descargado de internet tenía algún problema con las librerías que requería ser subsanado para su utilización.
- */shiny-ampldnn:* imagen utilizada para probar de manera simultánea tanto R con aplicaciones *shiny* como AMPL como python. Se utilizó a su vez como imagen base sobre la que ir probando la web desarrollada este trabajo que incluye HTML, CSS y Javascript además de los lenguajes mencionados.

Todos los directorios incluidos en /my-dockers se construyeron con una misma estructura de directorios, que es la siguiente:

- **/build:** contiene lo necesario para crear una imagen docker:
  - **/apps:** aplicación que se va a desplegar en el servidor del contenedor. Contiene los ficheros que necesita esta para su **creación**.
  - **/etc:** ficheros de configuración.
  - ***/scripts*:** scripts *Shell* necesarios para crear la imagen y acrhivo .R con la instalación de todos los paquetes necesarios (así como sus dependencias) para que el sistema funcione.
  - **Dockerfile:** archivo con la configuración de la imagen *docker* a crear. Ver 7.1.1.
  - **README:** breve documentación del resto de elementos del directorio en formato de texto. A su vez aquí se encuentran otros ficheros necesarios para configurar adecuadamente el entorno.
- **etc:** contiene el fichero de configuración shiny (en las imágenes donde se usa shiny, quedando vacío el directorio en las demás).
- **logs:** contiene los ficheros de salida de consola generados en la ejecución de la imagen.
- **apps:** contiene los ficheros de la aplicación necesarios en tiempo de **ejecución**. A su vez, estos se encuentran divididos en subdirectorios, de la misma forma que se encuentran gitlab en la carpeta code:
  - **RData :**archivos de datos que utiliza la aplicación. Contiene los ficheros `CSV` utilizados en la sección de mercados (ver 4) y el fichero `RData` mencionado en la misma sección.
  - **html_files :**archivos HTML de la aplicación web.
  - **images :**imágenes utilizadas en la apliación web.
  - **python_models :**modelos de predicción de valores de la aplicación, implementados en el lenguaje Python.
  - **repeatedTasks :**fichero de descarga automática de datos de manera programada con *cron* para utilización en la sección de mercados, comentado en 4.
  - **shinyApps :**contiene las carpetas de cada aplicación shiny construida, donde cada una contiene un archivo *ui.R* y un archivo *server.R* (ver 7.2.5).

  Otros archivos que se encuentran en esta carpeta son:

  - **index.html:** página *index* de la página web, es decir, la página a la que se accede a través de internet. Su contenido es el de la *landing page*, ver 6.3.2.
  - **estilos.css:** hoja de estilos de la página web.
  - **main.js:** archivo Javascript que controla el funcionamiento dinámico de la web.

  Modificando las carpetas y ficheros mencionados, se pueden modificar todos los elementos que componen la web, lo que permite solucionar errores y cambiar cualquier tipo de aspecto. Si se desean añadir nuevas funcionalidades, son estos los ficheros y directorios que habría que modificar o donde habría que añadir el nuevo contenido.

- **/my-shiny:** imagen shiny funcional y operativa construida inicialmente a modo de ejemplo y prueba del funcionamiento del shiny-server. Contiene la misma estructura que los directorios de /my-dockers.

- **/my-node:** aplicación web sencilla creada en la fase inicial para probar que el puerto de escucha utilizado por el servidor (el 80) escuchaba correctamente.

Mencionar finalmente que la imagen docker que utiliza el sistema construido finalmente, así como los archivos mencionados para crearla desde 0 (Dockerfile, archivos de entorno, archivos de instalación, etc) se encuentran en mi gitlab en la dirección `https://gitlab.inf.uva.es/victarr/tfg_informatica` en la carpeta docker files.

B.1.1 Conexión al servidor a través de Visual Studio Code

Visual Studio Code [134] es uno de los IDE (*Integration Development Environment*) más populares a día de hoy. Desarrollado por Microsoft para Windows, Linux y macOS, de uso gratuito y de código abierto, debe su fama a la gran cantidad de opciones que proporciona para el desarrollador,

entre las que cabe destacar: soporte para depuración, integración de control de versiones con git, gran cantidad de lenguajes con herramientas de ayuda de sintaxis, finalización inteligente de código y refactorización, así como múltiples opciones de personalización del entorno.

Figura B.1: Logotipo de Visual Studio Code. Tomado de [134].

Visual Studio Code permite a su vez trabajar de manera remota con un servidor, teniendo en todo momento el código alojado en este. Esta característica la aprovechamos en el presente trabajo utilizando una conexión **ssh** con la máquina virtual utilizada en el proyecto. Hay dos posibilidades, a través del usuario y contraseña del servidor, o utilizando claves públicas y privadas.

- **Con usuario y contraseña.**

  1. Instalación de Visual Studio Code en la máquina desde la que se va a trabajar.
  2. Instalación de la extensión *Remote-ssh* desde el menú de extensiones de Visual Studio Code.
  3. Presionar la tecla *F1* y escribir *remote* en el buscador que se nos abre.
  4. Hacer *click* en la opción *Remote-SSH: Add New Host...*.
  5. Introducimos la dirección ssh del servidor al que nos queremos conectar, en este caso, la dirección ssh de la máquina virtual: *ssh victor@virtual.lab.inf.uva.es*.
  6. Escogemos la dirección en donde se guardarán las credenciales SSH para la conexión remota con nuestro servidor (normalmente, la primera opción que nos aparece).

7. En caso de desearlo podemos editar el fichero *config* seleccionando *Open config* en el *pop up* que se nos muestra en la parte inferior derecha de la pantalla. Aquí podemos cambiar el nombre del *host* para facilicar la identificación del mismo.
8. Seleccionamos la opción *connect* de dicho *pop up* e introducimos la contraseña de nuestro usuario en el servidor y la conexión quedará establecida. Los archivos del servidor son accesibles a través del explorador de archivos.
9. Para realizar conexiones a dicho servidor en nuevas sesiones tendremos que dar nuevamente *F1* y seleccionar la opción *Remote-SSH: Connect Current Window to Host...* y seleccionar la conexión que queremos establecer.
10. Para desconectarse del servidor pinchamos en la parte inferior izquierda de la pantalla en el recuadro donde pone *SSH: nombre_conexion* y seleccionamos *Cerrar conexión remota* en las opciones que se nos muestran.

- **Con claves públicas y privadas.**

1. Seguimos los dos primeros pasos del procedimiento anterior. Tras ello, procedemos a la generación de claves en nuestra máquina local. En un terminal, escribimos el comando `$ sshkeygen` para la generación de las claves. Esto genera dos archivos (par de claves RSA): *id_rsa* e *id_rsa.pub* donde el segundo de ellos es la clave pública que debemos copiar a nuestro servidor.
2. Para copiar la clave pública utilizamos el siguiente comando en el terminal:
   `$ sshcopyid i $HOME/.ssh/id_rsa.pub usuario@servidor`.
3. La clave pública se habrá copiado en el servidor en el directorio oculto *$HOME/.ssh* en un archivo llamado *authorized_keys*.
4. El modo de proceder desde Visual Studio Code una vez establecida la conexión con el servidor es análogo al descrito en el procedimiento anterior, con la salvedad de que ahora editamos el fichero *config* añadiendo un campo *IdentifyFile* seguido de la ruta al directorio donde esté nuestra clave privada (*$HOME/.ssh/clave_privada*).
5. Si al establecer la conexión salta algún cartel indicando que no se ha podido establecer la conexión, tendremos que dar permisos de lectura al fichero de la clave privada, con el comando `$ sudo chmod 400 clave_privada`.

B.2 Diagramas de secuencia

Debido a las dificultades para conseguir mostrar adecuadamente los diagramas de secuencia de los casos de uso en la memoria, se incluyen los enlaces a gitlab donde se encuentra cada uno de ellos. Todos han sido realizados con Visual Paradigm 7.3.1.

Nota: el caso de uso "Configurar modelo de predicción" se divide en dos diagramas, uno con los modelos de redes neuronales y otro con los demás, con el objetivo de facilitar la visualización del mismo. Ambos diagramas componen un único caso de uso.

- **Configurar gráfico analítico:** `https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Configurar_gr%C3%A1fico_anal%C3%ADtico.pdf`

- **Configurar modelo de optimización:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Configurar_modelo_de_optimizaci%C3%B3n.pdf
- **Configurar modelo de predicción (modelos sin redes neuronales):** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Configurar_modelo_de_predicci%C3%B3n_1.pdf
- **Configurar modelo de predicción (modelos de redes neuronales):** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Configurar_modelo_de_predicci%C3%B3n_2.pdf
- **Consultar información de un mercado:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Consultar_informaci%C3%B3n_de_un_mercado.pdf
- **Ver la relación entre una cartera y un índice bursátil:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Ver_la_relación_entre_una_cartera_y_un_índice_bursátil.pdf
- **Simular carteras :** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Simular_carteras.pdf
- **Ver grupos de activos en términos de rentabilidad:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Ver_grupos_de_activos_en_t%C3%A9rminos_de_rentabilidad.pdf
- **Visualizar resultados de modelo de optimización:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Ver_la_relación_entre_una_cartera_y_un_índice_bursátil.pdf
- **Evaluar cartera:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Evaluar_cartera.pdf
- **Comparar modelos de predicción:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Comparar_modelos_de_predicci%C3%B3n.pdf
- **Visualizar gráfico analítico:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Ver_la_relación_entre_una_cartera_y_un_índice_bursátil.pdf
- **Visualizar modelo de predicción:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Ver_la_relación_entre_una_cartera_y_un_índice_bursátil.pdf
- **Visualizar indicadores técnicos:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Ver_la_relación_entre_una_cartera_y_un_índice_bursátil.pdf
- **Visualizar variaciones en función del parámetro mu:** https://gitlab.inf.uva.es/victarr/tfg_informatica/-/blob/main/diagrams/Visualizar_variaciones_en_funci%C3%B3n_del_par%C3%A1metro_mu.pdf