



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**título del TFG  
Documentación Técnica**



Presentado por nombre alumno  
en Universidad de Burgos — 5 de junio de 2024  
Tutor: nombre tutor



---

# Índice general

---

<b>Índice general</b>	<b>i</b>
<b>Índice de figuras</b>	<b>iii</b>
<b>Índice de tablas</b>	<b>iv</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	11
<b>Apéndice B Especificación de Requisitos</b>	<b>17</b>
B.1. Introducción . . . . .	17
B.2. Objetivos generales . . . . .	18
B.3. Catálogo de requisitos . . . . .	18
B.4. Especificación de requisitos . . . . .	21
<b>Apéndice C Especificación de diseño</b>	<b>37</b>
C.1. Introducción . . . . .	37
C.2. Diseño de datos . . . . .	37
C.3. Diseño procedimental . . . . .	37
C.4. Diseño arquitectónico . . . . .	37
<b>Apéndice D Documentación técnica de programación</b>	<b>39</b>
D.1. Introducción . . . . .	39
D.2. Estructura de directorios . . . . .	39
D.3. Manual del programador . . . . .	39

D.4. Compilación, instalación y ejecución del proyecto . . . . .	39
D.5. Pruebas del sistema . . . . .	39
<b>Apéndice E Documentación de usuario</b>	<b>41</b>
E.1. Introducción . . . . .	41
E.2. Requisitos de usuarios . . . . .	41
E.3. Instalación . . . . .	41
E.4. Manual del usuario . . . . .	41
<b>Apéndice F Anexo de sostenibilización curricular</b>	<b>43</b>
F.1. Introducción . . . . .	43
<b>Bibliografía</b>	<b>45</b>

---

# Índice de figuras

---

B.1. Diagrama de casos de uso. Fuente: elaboración propia . . . . .	22
---	----

---

# Índice de tablas

---

A.1. Costes en RRHH . . . . .	12
A.2. Costes de <i>HW</i> y <i>SW</i> . . . . .	12
A.3. Costes infraestructura . . . . .	13
A.4. Costes totales . . . . .	13
A.5. Licencias de terceros . . . . .	14
B.1. CU-1 Consultar portada. . . . .	23
B.2. CU-2 Mostrar carrusel noticias generales. . . . .	24
B.3. CU-3 Mostrar mejores y peores valores. . . . .	25
B.4. CU-4 Mostrar gráfica. . . . .	26
B.5. CU-5 Gestionar usuario. . . . .	27
B.6. CU-6 Registro. . . . .	28
B.7. CU-6 Login. . . . .	29
B.8. CU-8 Logout. . . . .	30
B.9. CU-9 <i>DashBoard</i> . . . . .	31
B.10.CU-10 Crear nuevo valor en cartera. . . . .	32
B.11.CU-11 Mostrar <i>donut</i> , sectores y divisas. . . . .	33
B.12.CU-12 Ver gráfica de Markowitz, ratio de Sharpe y pesos. . . . .	34
B.13.CU-13 Ver posiciones abiertas y evolución de cartera. . . . .	35

## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

En un proyecto con un equipo completo de personas trabajando, podríamos definir la planificación de un proyecto software como una etapa que implica detallar los objetivos y el alcance del proyecto, identificar los requisitos y recursos necesarios, establecer un cronograma con hitos clave, asignar tareas a los miembros del equipo y prever los posibles riesgos.

Dentro del contexto de este trabajo académico, se va a detallar en qué ha consistido la planificación temporal y se tratará de hacer un estudio de viabilidad realista.

### A.2. Planificación temporal

En la memoria se ha indicado que se ha utilizado una metodología ágil de gestión de proyectos, con clara fundamentación en Scrum[5]. Algunos de los aspectos más relevantes que han cubierto esta filosofía han sido:

- Desarrollo incremental a través de iteraciones llamadas *sprints*.
- Utilización de repositorio *Git* para solicitud de mejoras y para realizar un seguimiento de la evolución, con acceso para los tutores desde las fases iniciales.
- Control temporal a través de los *sprints*, valorando al inicio de cada uno cuál sería la duración adecuada basándonos en las tareas que se iban a realizar y el producto que se esperaba al final de esa iteración.
- Seguimiento de tareas en un panel *Kanban*.

Además de los *sprints* se diseñaron hitos relevantes denominados *Milestones* en *GitHub* que sirven como referencia de la producción realizada.

## Sprints

A continuación se muestra un resumen de los *sprints* que han tenido lugar en las diferentes fases de este trabajo:

### Sprint 0. Presentación inicial a tutores del TFG

- **Duración:** 19/10/2023 a 23/10/2023
- **Objetivo:** Presentar una idea de TFG a los posibles tutores.
- **Contexto:** Inicio del último curso del Grado de Ingeniería Informática. Se trata de contactar con los posibles tutores que controlen el desarrollo de un TFG. Este TFG estará basado en la idea de desarrollar una herramienta, escrita en Python, para realizar análisis financiero.
- **Tareas:**
  - Contactar con los posibles tutores por email.
  - Contactar por Teams con los tutores que muestren interés en el TFG e intercambiar primeras impresiones.
  - Determinar las herramientas necesarias para desarrollar un TFG.
  - Preguntar por consejos y opiniones que puedan ser relevantes.
  - Concertar cita para una futura reunión cuando se haya avanzado con los primeros pasos del TFG.

### Sprint 1. Empezar TFG con selección de herramientas adecuadas

- **Duración:** 24/10/2023 a 08/11/2023
- **Objetivo:** Empezar el desarrollo del TFG y su correspondiente documentación. Hacerlo utilizando las herramientas adecuadas sugeridas por los tutores en el trascurso del sprint 0.
- **Contexto:** Tras haber mantenido una primera reunión con los tutores, se detectan ciertas necesidades en cuanto a la utilización de herramientas adecuadas para el desarrollo de un TFG.
- **Tareas:**



- Lectura completa de documentación de TFGs disponible en la plataforma UBUVirtual.
- Rellenar el formulario de *Oferta de TFG Grado de Informática Online*. Buscar referencias bibliográficas teóricas más adecuadas.
- Utilización de *Zube*[3] para gestionar un proyecto con metodología ágil. Para ello, hay que buscar la documentación adecuada que permita entender su utilización.
- Crear un repositorio en *GitHub* que permita llevar un seguimiento de las acciones realizadas.
- En el repositorio, añadir a los tutores e integrar con *Zube*.
- Determinar la mejor herramienta para la documentación del TFG. Elegir entre  $\text{\LaTeX}$  *Word*. Justificar la elección y documentar.
- Buscar un gestor de referencias bibliográficas: *Zotero vs Mendeley*.
- Concertar cita para una futura reunión cuando se haya avanzado con los primeros pasos del TFG.

## Sprint 2. Mostrar primeras tablas y gráficos en servidor web

- **Duración:** 08/11/2023 a 29/11/2023
- **Objetivo:** Comenzar a desarrollar una aplicación web, mostrando información de precios cotizados y de gráficos. Almacenar información en una base de datos SQLite3, no en DataFrames ni en archivos .csv.
- **Contexto:** Una de las primeras recomendaciones de los tutores fue no utilizar archivos .csv para almacenar la información, por lo tanto, se va a utilizar un sistema de almacenamiento en base de datos. La idea general es que dichos datos se descargan con una API, se almacenan y, posteriormente, se procesan para mostrar la información relevante al usuario en archivos HTML.
- **Tareas:**
  - Preparar entorno local para utilizar una base de datos SQLite3. Instalar una GUI para la BD (no relevante para el usuario final).
  - Preparar clases (Descargador, Vista, Ticker y/o similares) que permitan automatizar la tarea de almacenar información en una base de datos.
  - Mostrar un primer producto - no necesariamente visualmente atractivo - en un servidor web con tablas extraídas de la BD.
  - Documentar, en el capítulo 4 de la memoria, las herramientas utilizadas.

- Concertar cita para una futura reunión con los tutores.

### **Sprint 3. Mejorar lógica de web y proteger enlaces con registro. Mostrar gráficos y dar primeros estilos**

- **Duración:** 29/11/2023 a 13/12/2023
- **Objetivo:** Mejorar la lógica de acceso a la web. Proteger los enlaces no públicos mediante registro de usuarios. Mostrar gráficos y no sólo tablas en un estilo visualmente atractivo.
- **Contexto:** Una vez llevado el proyecto a *producción* (subido a un servidor web) hay que mejorar y asegurar los enlaces para que no sean accesibles en un mal uso de los mismos. Además, es necesario hacer un registro de usuarios que permita mostrar información relevante dependiendo del rol (invitado / registrado).
- **Tareas:**
  - Crear lógica de control de usuarios registrados.
  - Asignar permiso de acceso a enlaces dependiendo del tipo de usuario.
  - Crear portada y dar primeros estilos visuales.
  - Preparar un índice o zona de *breadcrumbs* de la página para mejorar la navegación.
  - Diseñar una vista que permita mostrar gráficos según el valor cotizado escogido.

### **Sprint 4. Manejar imágenes estáticas. Añadir base de datos de DJ30 y buscar método para auto-actualizar las BDs**

- **Duración:** 14/12/2023 a 28/12/2023
- **Objetivo:** Mejorar estética y mostrar imágenes estáticas. Crear y añadir una nueva base de datos. Hacer las bases de datos auto-actualizables.
- **Contexto:** El proyecto en local permite mostrar imágenes estáticas en la navbar y en portada, pero en *producción* (subido a un servidor web) no. Hay que mejorar este aspecto para que resulte visualmente atractivo. Además, hay que crear y añadir una nueva base de datos para el índice DJ30 y hay que buscar un método para auto-actualizar las BD en producción: con *cron*[4] o con *tasks*.

**■ Tareas:**

- Mostrar imágenes estáticas en producción (en servidor web).
- Crear una nueva BD y hacerla accesible.
- Investigar e implantar un método de auto-actualización de BDs en producción.

**Sprint 5. Añadir un carrusel de noticias y cambiar el acceso a las BDs. Incorporar tests y mejorar los logs****■ Duración:** 04/01/2024 a 18/01/2024

- **Objetivo:** Añadir noticias relacionadas con los mercados financieros, para hacer un interfaz más útil y agradable para el usuario. Cambiar la lógica de acceso a las bases de datos, para evitar las sentencias en SQL y aprovechar las capacidades del patrón MVT[2] de Django. Aplicar primeros tests para mejorar el comportamiento y aumentar la seguridad.

- **Contexto:** Ahora mismo la portada principal de la página es una imagen estática que permite hacer *login*, pero que no aporta la utilidad que se espera de este tipo de aplicaciones. Por ello, parece razonable añadir una zona de noticias relacionadas con los mercados financieros, que hagan la página más dinámica y agradable de usar. Además, aunque tengo diseñados unos pequeños tests, se tiene que mejorar toda la estructura de *testing* y ampliar el *logger*.

**■ Tareas:**

- Añadir una nueva aplicación al proyecto, que permita controlar lo que ocurre en la portada:
  - Mostrar noticias en portada.
  - Mostrar información sobre los mejores y peores stocks en portada.
  - Mejorar el logging para lo que ocurra en esta nueva aplicación.
- Mejorar la lógica de acceso a las BDs. Aplicar MVC y evitar consultas directas de SQL en la medida de lo posible.
- Ampliar los tests y mejorar la estructura de testing.

**Sprint 6. Añadir un *dashboard* para almacenar y controlar una cartera por parte del usuario. Operaciones CRUD en BDs****■ Duración:** 22/01/2024 a 05/02/2024

- **Objetivo:** Añadir un área personal que permita realizar el control de una cartera de acciones, con posiciones abiertas y posiciones objetivo. Poder crear informes con la composición de la cartera y mostrar porcentajes de inversión de forma agradable al usuario.
- **Contexto:** Aunque un usuario puede consultar los datos de un valor concreto y de su índice de referencia, ahora mismo no puede tener una lista de seguimiento que le permita controlar sus inversiones. Se pretende integrar una nueva aplicación que permita ofrecer un *dashboard* a los usuarios, para que puedan hacer un seguimiento de sus inversiones.
- **Tareas:**
  - Añadir una nueva aplicación al proyecto, que permita controlar un dashboard.
    - Permitir al usuario almacenar información sobre sus inversiones actuales y futuras.
    - Mostrar acciones compradas, i.e., posiciones abiertas.
    - Mostrar una lista de seguimiento.
    - Mejorar el *logging* para lo que ocurra en esta nueva aplicación.
    - Preparar los tests más relevantes para el control de esta nueva aplicación.
    - Integrar esta aplicación con las actuales.
  - Aprovechar la lógica mejorada de acceso a las BDs para facilitar todas las operaciones CRUD.
  - Ampliar los tests.

### Sprint 7. Refactoring y testing. Poder eliminar stocks en seguimiento desde el DashBoard.

- **Duración:** 07/02/2024 a 14/02/2024
- **Objetivo:** Ahora mismo hay bastantes casos sin cubrir con los tests, especialmente en las *views* de las *apps* incorporadas al proyecto. Por ello, se hace necesario, antes de continuar, realizar un testeo exhaustivo para asegurar una buena base en desarrollos posteriores.

Además, voy a incorporar una nueva funcionalidad dentro del DashBoard, que permitirá al usuario eliminar valores en seguimiento (de manera similar a como se eliminan actualmente los valores en cartera).

- **Contexto:** Tras intentar seguir un enfoque TDD en la creación del DashBoard en el sprint 6, se detecta la necesidad de resolver algunos fallos y cubrir el código, al 100 %, con tests. Además, hay que mejorar el estilo y la calidad del código, así como medir dicha calidad con alguna herramienta que ofrezca información final que justifique las mejoras.
- **Tareas:**
  - Cubrir todas las normas de estilo de *PEP-8* enhancement
  - Solucionar problema formularios *DashBoard* con precios de más de 2 decimales.
  - Incluir funcionalidad de eliminación de stocks en seguimiento.
  - Completar tests unitarios de `News.views`
  - Completar tests unitarios `DashBoard.views`
  - Completar tests unitarios de `Analysis.views`
  - Medir con *flake8* y *pylint* la calidad del código.

## Sprint 8. Correlaciones entre valores con NetworkX

- **Duración:** 17/02/2024 a 02/03/2024
- **Objetivo:** Permitir al usuario comparar la evolución de un valor con respecto al resto, en un tiempo determinado. La idea es mostrar al usuario un grafo creado con *NetworkX*[\[1\]](#) que permita entender las correlaciones (positiva y negativa) que haya entre diferentes valores. De esta manera se podrá hacer un análisis de diversificación de cartera, ya que tener valores muy correlacionados puede ser indicativo de no tener la cartera bien balanceada.
- **Contexto:** El usuario puede acceder a la información de un valor pero no puede compararlo con otros. Además, aunque tenga valores comprados o en seguimiento, realmente, ahora mismo, no se le está ofreciendo capacidad analítica, así que hay que proporcionársela.
- **Tareas:**
  - Añadir dos índices adicionales, con sus valores, para ampliar la información disponible y ofrecer un mayor abanico comparativo.
  - Agregar una funcionalidad que permita ver la correlación de un valor con todos los demás disponibles.
    - Como la correlación puede ser positiva o negativa, hacer diferenciación entre ambas.

- Además, puede que interese obtener información sobre las mayores correlaciones (positivas o directas; y negativas o inversas)
- Una vez obtenidas las correlaciones, sobre los precios de cierre, ver la información en grafos de *NetworkX*.
- Adicionalmente, intentar permitir al usuario ver los gráficos de la mayor correlación de forma conjunta.
- Ampliar los tests a la nueva funcionalidad.
- Integrar los grafos en la plataforma web de forma temporal para cada usuario (sin almacenar en el servidor).

### **Sprint 9. Aplicación de modelos. *Forecasting* para series temporales**

- **Duración:** 13/03/2024 a 27/03/2024
- **Objetivo:** Aportar una visión diferenciadora al análisis de valores cotizados, permitiendo que un usuario pueda aplicar un modelo que le permita tener una idea intuitiva de la futura evolución de la cotización. En cualquier caso, se informará al usuario de que los datos aportados por el modelo no pueden ser considerados como una fuente única para tomar decisiones de inversión.
- **Contexto:** En las webs que ofrecen información de productos cotizados no se ofrece un apartado de experimentación con modelos de *forecasting*. Puede ser un factor diferenciador para aquellos usuarios que entiendan los riesgos y las ventajas de aplicar estos modelos.
- **Tareas:**
  - Añadir el análisis a través de un modelo ARIMA.
  - Añadir el análisis a través de una red LSTM.
  - Permitir al usuario ver alguna información gráfica que facilite la comprensión de los resultados.
  - Ampliar los tests de las nuevas funcionalidades.
  - Integrar las gráficas o diagramas en la plataforma web de forma temporal para cada usuario (sin almacenar en el servidor).

### **Sprint 10. Documentación y limpieza de código y avanzar considerablemente en la memoria.**

- **Duración:** 10/04/2024 a 24/04/2024

- **Objetivo:** Documentación de código y de la memoria.
- **Contexto:** Actualmente el proyecto está documentado con notas internas del autor y debe empezarse a documentar utilizando las herramientas y guías oficiales.
- **Tareas:**
  - Configurar *MikTex*.
  - Descargar y preparar plantilla oficial de Trabajos de Fin de Grado.
  - Añadir resumen / abstract.
  - Documentar introducción.
  - Describir objetivos del proyecto.
  - Documentar conceptos teóricos.
  - Explicar las técnicas y herramientas utilizadas.
  - Indicar aspectos relevantes del desarrollo del proyecto que se hayan dado hasta ahora.
  - Detallar trabajos relacionados.
  - Añadir referencias bibliográficas en todos los apartados de la memoria.
  - Medir la calidad del código e introducir las mejoras necesarias.

### **Sprint 11. Gráfica de Markowitz con frontera eficiente y ratio de Sharpe. Mejoras visuales.**

- **Duración:** 09/05/2024 a 19/05/2024
- **Objetivo:** Favorecer el estudio de una cartera de valores a través de la distribución de pesos de los mismos. Mostrar una gráfica de Markowitz con la cartera ideal encontrada por simulación de Montecarlo y, también, por optimización de funciones. Realizar pequeñas mejoras estéticas y funcionales.
- **Contexto:** Es necesario mejorar el *DashBoard* de los usuarios para que reciban una información más completa y, de entre las opciones barajadas, una de las más interesantes es la de mostrar cuál sería la mejor distribución de los valores que ya tenga en cartera el usuario, basándome en los retornos y pesos en la propia cartera.
- **Tareas:**
  - Añadir nuevos métodos que permitan hacer una simulación de Montecarlo con diferentes pesos de valores en una cartera.

- Añadir nuevos métodos al *DashBoard* para calcular la frontera eficiente.
- Desarrollar las funciones de optimización que permitan calcular la mejor distribución de pesos.
- Preparar los métodos necesarios para mostrar la gráfica de Markowitz junto con la frontera eficiente y las mejores carteras encontradas.
- Implementar los tests necesarios para las nuevas funcionalidades.
- Integrar en las plantillas existentes del *DashBoard* para mostrar.
- Documentar de forma interna en el apartado de conceptos teóricos (apartado 3 de la memoria del TFG).
- Mejoras estéticas en botones y página principal.
- Añadir referencias bibliográficas en todos los apartados de la memoria aunque sea a nivel local.
- Medir la calidad de nuevo código y testear de forma automática con *GitHub actions*.
- Desplegar nueva documentación de código de forma automática con *Sphinx*.

### Sprint 12. Mejorar gráficas ARIMA. Incluir estrategias basadas en *machine learning*

- **Duración:** 22/05/2024 a 26/05/2024
- **Objetivo:** Completar documentación teórica de ARIMA en memoria, añadir apartado de *trading* algorítmico con su documentación y contemplar redes LSTM como posible mejora del proyecto.
- **Contexto:** Ahora mismo se puede acceder a un apartado de *forecasting* con redes LSTM pero no me convencen los resultados obtenidos, así que se dejará como una de las posibles mejoras del proyecto. Sin embargo, voy a añadir un apartado nuevo al *Lab* sobre *trading* algorítmico.

Además, tengo que terminar la documentación de ARIMA en la memoria y añadir la nueva documentación de *trading* algorítmico.

- **Tareas:**
  - Terminar documentación ARIMA.
  - Mejorar gráficas y plantillas HTML para ARIMA.
  - Incluir mejoras de estrategia basada en ML, en lugar de redes LSTM, y documentar.
  - Pasar redes LSTM a apartado de posibles mejoras del proyecto.



- Realizar nuevos tests para estrategias basadas en ML.
- Crear un *release* con el cambio, porque es relevante.

### **Sprint 13. Finalizar memoria. Empezar anexos. Correcciones en plantillas y en *cron* de servidor web**

- **Duración:** 31/05/2024 a 04/06/2024
- **Objetivo:** Finalizar memoria y empezar anexos. Realizar mejoras estéticas. Corregir tarea *cron* de actualización de BDs en servidor web. Realizar la *release* que está pendiente desde el *sprint* anterior, no realizado por falta de comprobaciones.
- **Contexto:** En este momento es posible trabajar con todas las herramientas, pero quedan algunos detalles que mejorar antes de realizar un *release*. Además, es necesario avanzar con la documentación de la memoria (que ya tiene cubierta toda la parte teórica).
- **Tareas:**
  - Finalizar sección 5 de memoria.
  - Finalizar sección 6 de memoria.
  - Finalizar sección 7 de memoria.
  - Empezar anexos.
  - Mejorar gráficas y plantillas HTML para estrategias basadas en ML.
  - Ampliar tests para cubrir todos los métodos.
  - Mejorar la calidad de código. Comprobar con *pylint*.
  - Mejoras gráficas en grafos de correlaciones de *NetworkX* y adaptación de plantillas en local.
  - Corregir problemas de tarea *cron* en servidor web.
  - Crear un *release* con los cambios, siempre y cuando haya terminado todas las correcciones.

## **A.3. Estudio de viabilidad**

### **Viabilidad económica**

Se va a realizar una estimación de costes lo más aproximada a la realidad, como si el proyecto se hubiese realizado en un entorno empresarial.

### Estudio de costes

De manera general se pueden considerar costes de personal, de *hardware* y *software* y, en caso de necesidad, costes de infraestructura. La mayor partida económica se tiene que destinar a los recursos humanos:

Concepto	Coste anual	Prorratio (6 meses)
<i>Salario bruto</i>	25.000,00€	12.500,00€
<i>Retención IRPF</i>	4.250,00€	2.125,00€
<i>Seguridad Social</i>	1.587,50€	793,75€
<i>Salario neto</i>	19.162,50€	9.581,25€

Tabla A.1: Costes en RRHH

Se aplica un 17 % de retención sobre la nómina, considerando ausencia de deducciones tributarias.

A continuación se muestran los costes de *hardware* y *software*. Para el *hardware* se estima una amortización de 5 años y una utilización de 6 meses. El *software*, por su parte, tendrá una amortización estimada de 2 años.

Concepto	Coste	Coste amortizado
<i>Ordenador portátil</i>	1.600,00€	160,00€
<i>Licencia MS Windows 10 pro</i>	279,00€	69,75€

Tabla A.2: Costes de HW y SW

Por último, se realiza un análisis de costes de infraestructura (solo asimilable en caso de necesidad) de nuevo, el coste amortizado se calcula a 6 meses:

Concepto	Coste anual	Coste amortizado
<i>Consumo eléctrico</i>	350,00€	175,00€
<i>Espacio coworking</i>	4.000,00€	2.000,0€
<i>Material de oficina</i>	10,00€	5,00€
<i>Alojamiento web</i>	72,00€	36,00€

Tabla A.3: Costes infraestructura

Los costes totales estimados son:

Concepto	Coste
<i>Costes en RRHH</i>	12.500,00€
<i>Costes de HW y SW</i>	229.75€
<i>Costes infraestructura</i>	2.216,00€
<b>TOTAL</b>	14.945,75€

Tabla A.4: Costes totales

## Análisis de beneficios

Este trabajo no se plantea inicialmente como un mecanismo para generar beneficios económicos, sino como una herramienta gratuita para ayudar a los inversores a mejorar la distribución de sus carteras y para que los analistas financieros experimenten con diferentes técnicas.

En caso de monetizar el proyecto se podría optar por incluir publicidad en la web para generar ingresos que, al menos, cubran los gastos anuales.

## Viabilidad legal

### Descargo de responsabilidades

El *software* que se proporciona en este trabajo es solo para fines informativos y no debe considerarse como asesoramiento financiero. La información proporcionada no está garantizada como precisa o completa y puede cambiar sin previo aviso.

El usuario es el único responsable de sus decisiones de inversión y de las consecuencias de las mismas. No se debe confiar en este *software* para tomar

decisiones de inversión sin realizar una investigación profunda y consultar con un asesor financiero calificado.

### Licencias *software*

En este trabajo se utilizan múltiples librerías de terceros. Además, aunque los datos son almacenados en bases de datos propias, hay que considerar que se hace uso de información externa a través de la API *yFinance*.

Las licencias asociadas a las librerías y APIs utilizadas, se detallan en la siguiente tabla:

Librería / API	Licencia
<i>Python</i>	OSI-Open Source
<i>Django</i>	BSD
<i>yFinance</i>	Apache 2.0
<i>Pandas</i>	BSD
<i>Numpy</i>	BSD 3-Clause
<i>Scikit-Learn</i>	BSD 3-Clause
<i>Scipy</i>	BSD
<i>NetworkX</i>	BSD 3-Clause
<i>Matplotlib</i>	Licencia libre propia
<i>Plotly</i>	MIT
<i>Statsmodels</i>	BSD 3-Clause
<i>Feedparser</i>	BSD 3-Clause
<i>NewsAPI</i>	MIT
<i>pmdarima</i>	MIT
<i>Keras</i>	Apache 2.0

Tabla A.5: Licencias de terceros

Todas las librerías y el *framework Django* utilizan licencias de código abierto permisivas - la más restrictiva sería Apache 2.0 - lo que significa que el código de *FAT: Financial Analysis Tool* se puede usar, modificar y distribuir libremente, incluyendo para fines comerciales, siempre que se cumplan los requisitos de atribución y exención de responsabilidad establecidos en cada licencia.

Por tanto, se toma la decisión de publicar este proyecto bajo licencia CC BY-NC-SA 4.0[? ].

### **Consideraciones adicionales**

Dada la condición académica de este proyecto no hay limitación en el uso de los datos financieros almacenados y no se recoge información personal; pero dependiendo del uso que se dé a este *software* es posible que se deba cumplir con regulaciones específicas, como la Ley de Protección de Datos de la Unión Europea (GDPR) o la Ley de Protección de la Privacidad de la Información Financiera (FINRA).



## *Apéndice B*

---

# Especificación de Requisitos

---

## B.1. Introducción

En este anexo se recogen los requisitos y casos de uso que se han tenido en cuenta para el desarrollo de este trabajo. Se han seguido las recomendaciones del estándar IEEE 830-1998[?] como guía de buenas prácticas.

En este proyecto se ha tratado de crear una especificación de requisitos que cumpla con las siguientes condiciones:

- Completa y consistente
  - Incluir todos los requisitos y referencias necesarias.
  - Ser coherente con los propios requisitos y otros documentos de especificación.
- Claridad y accesibilidad
  - Redacción clara para evitar malas interpretaciones.
  - Uso de términos y definiciones precisos.
- Verificabilidad y trazabilidad
  - Debe de existir un método finito y sin costo para probar los requisitos.
  - Uso de términos y definiciones precisos.
- Modificabilidad y priorización
  - Fácilmente modificable.
  - Jerarquía de priorización según relevancia para el negocio (o fin).

- Correctitud y accesibilidad
  - El software debe cumplir con los requisitos de la especificación.
  - Accesibilidad y facilidad de comprensión para los usuarios y desarrolladores.

## B.2. Objetivos generales

Este trabajo persigue los siguientes objetivos generales:

- Proporcionar una herramienta que otorgue capacidad crítica en inversiones personales.
- El sistema tiene que poder controlar la evolución de valores en cartera y en seguimiento.
- Tiene que mostrarse información relevante en cuanto a distribución de carteras y posibles pesos para los valores escogidos.
- Permitir trabajar, de manera experimental, con herramientas de *trading* algorítmico.
- Facilitar el acceso a la información con una página web pública y gratuita.

## B.3. Catálogo de requisitos

### Requisitos funcionales

- **RF-1 Mostrar portada con información general de diferentes índices bursátiles:** la web tiene que tener información agregada de todos los índices disponibles.
  - **RF-1.1 Mostrar carrusel de noticias generales:** en la portada se tiene que disponer de noticias relevantes dentro del mundo bursátil.
  - **RF-1.2 Mostrar mejores y peores valores de cada índice:** se tiene que poder consultar, de forma ágil, cuáles han sido los mejores y peores valores de cada índice.
    - **RF-1.2.1 Mostrar gráfica:** tiene que estar disponible una gráfica evolutiva junto a cada valor de los mostrados.



- **RF-2 Control de usuarios:** los usuarios tendrán acceso a información adicional si están dados de alta.
  - **RF-2.1 Permitir registro:** el usuario se podrá registrar.
  - **RF-2.2 Permitir hacer *login*:** el usuario podrá acceder con su cuenta en la web.
  - **RF-2.3 Permitir *logout*:** el usuario podrá cerrar su sesión.
- **RF-3 Gestionar *DashBoard*:** tiene que haber una zona de usuario, denominada *DashBoard*, con información relevante sobre valores seleccionados por el usuario.
  - **RF-3.1 Crear nuevo valor en cartera:** el usuario podrá añadir un nuevo valor, con fecha y precio de compra, a su cartera.
    - **RF-3.1.1 Mostrar 'donut', sectores y divisas:** si un usuario tiene valores en cartera, éste contará con diagramas de información relevante.
    - **RF-3.1.2 Ver gráfica de Markowitz, ratio de Sharpe y pesos:** si un usuario tiene valores en cartera podrá ver la mejor distribución de pesos para la misma.
    - **RF-3.1.3 Ver posiciones abiertas y evolución de cartera:** si hay valores en cartera se podrá comprobar la evolución de la misma.
  - **RF-3.2 Eliminar valor de cartera:** el usuario podrá eliminar un valor de los que tuviera almacenados.
  - **RF-3.3 Crear un nuevo valor en seguimiento:** el usuario podrá hacer seguimiento de valores de su interés.
  - **RF-3.4 Eliminar valor de seguimiento:** el usuario podrá eliminar un valor de los que tuviera en seguimiento.
- **RF-4 Consultar índice:** se debe presentar, de forma ordenada, información relevante de cada índice por separado.
  - **RF-4.1 Mostrar tabla de valores:** al consultar un índice se mostrará una tabla con todos los componentes de ese índice, con información relevante sobre la última sesión disponible.
  - **RF-4.2 Consultar un valor del índice:** se podrá consultar información de un único valor.
    - **RF-4.2.1 Dar acceso a gráfica interactiva:** se ofrecerá una gráfica interactiva, con medias móviles y volumen.

- **RF-4.2.2 Mostrar distribución de retornos:** habrá una gráfica con la distribución que hayan seguido los retornos en los últimos meses.
  - **RF-4.2.3 Mostrar datos del último mes:** disponer datos en formato tabular para ver precios de apertura y cierre de los últimos 30 días.
  - **RF-4.2.4 Ver evolución del sector:** facilitar una comparativa con el sector de referencia del valor.
  - **RF-4.2.5 Comparar con otros valores:** existirá la posibilidad de comparar gráficas de precios de cierre con otros valores.
  - **RF-4.2.6 Mostrar grafos de correlación:** mostrar grafos de alta correlación positiva y negativa con otros valores.
- **RF-4.3 Mostrar noticias relacionadas:** facilitar enlaces de fuentes RSS que estén relacionadas con el índice.
- **RF-4.4 Ver gráfica de evolución:** mostrar gráfica de evolución del índice en su conjunto.
- **RF-5 Gestionar *Lab*:** dar acceso a un laboratorio virtual.
  - **RF-5.1 Trabajar con ARIMA:** permitir realizar estimaciones y búsqueda de parámetros con ARIMA.
    - **RF-5.1.1 Buscar parámetros (p,d,q) con funciones ACF y PACF:** se mostrarán gráficas de funciones para interpretación del usuario.
    - **RF-5.1.2 Introducir (p,d,q) de forma manual:** el usuario podrá introducir los parámetros deseados para aplicar un modelo ARIMA.
    - **RF-5.1.3 Buscar (p,d,q) de forma automática:** se proporcionará una funcionalidad de búsqueda automática de los parámetros más adecuados a un valor.
    - **RF-5.1.4 Hacer búsqueda por rejilla para (p,d,q):** se podrá realizar una búsqueda de parámetros (p,d,q) de entre una serie de posibles valores preestablecidos.
  - **RF-5.2 Trabajar con *trading* algorítmico:** facilitar el acceso a herramientas de *trading* algorítmico.
    - **RF-5.2.1 Usar algoritmo de cruce de medias:** se mostrará el resultado de buscar las mejores medias móviles para un valor, en un período de tiempo concreto.

- **RF-5.2.2 Utilizar estrategia basada en *machine learning*:** permitir interactuar con modelos de regresión y clasificación para estimar la tendencia de la próxima sesión de un índice.

## Requisitos no funcionales

En este apartado se tratará de dar detalle de aquellas características que no son funcionales pero que aportan un valor añadido al proyecto y a la interacción con la herramienta desarrollada:

- **RNF-1 Escalabilidad:** la web tiene que permitir y favorecer, en la medida de lo posible, la incorporación de nuevas funcionalidades y de bases de datos adicionales.
- **RNF-2 Privacidad:** los datos de usuario, como nombres o contraseñas, se debe gestionar de forma segura.
- **RNF-3 Disponibilidad:** la web debe ser compatible con los navegadores más modernos y tendrá alta disponibilidad a través de un servicio de alojamiento fiable.
- **RNF-4 Usabilidad:** la interfaz de la web será *user friendly*, resultar intuitiva y facilitará comentarios de ayuda adicional en los apartados más especializados.
- **RNF-5 Mantenibilidad:** se tiene que favorecer un mantenimiento posterior a la puesta en producción. Además, se facilitará información a los desarrolladores para que puedan realizar mejoras incrementales posteriores.

## B.4. Especificación de requisitos

### Casos de uso

A continuación se muestran todos los casos de uso contemplados. Muchos de estos casos de uso se plantearon en las fases iniciales del proyecto y se fueron mejorando en las sucesivas iteraciones de cada *sprint*. Se trató de realizar mejoras incrementales para satisfacer las expectativas de un inversor o potencial cliente:

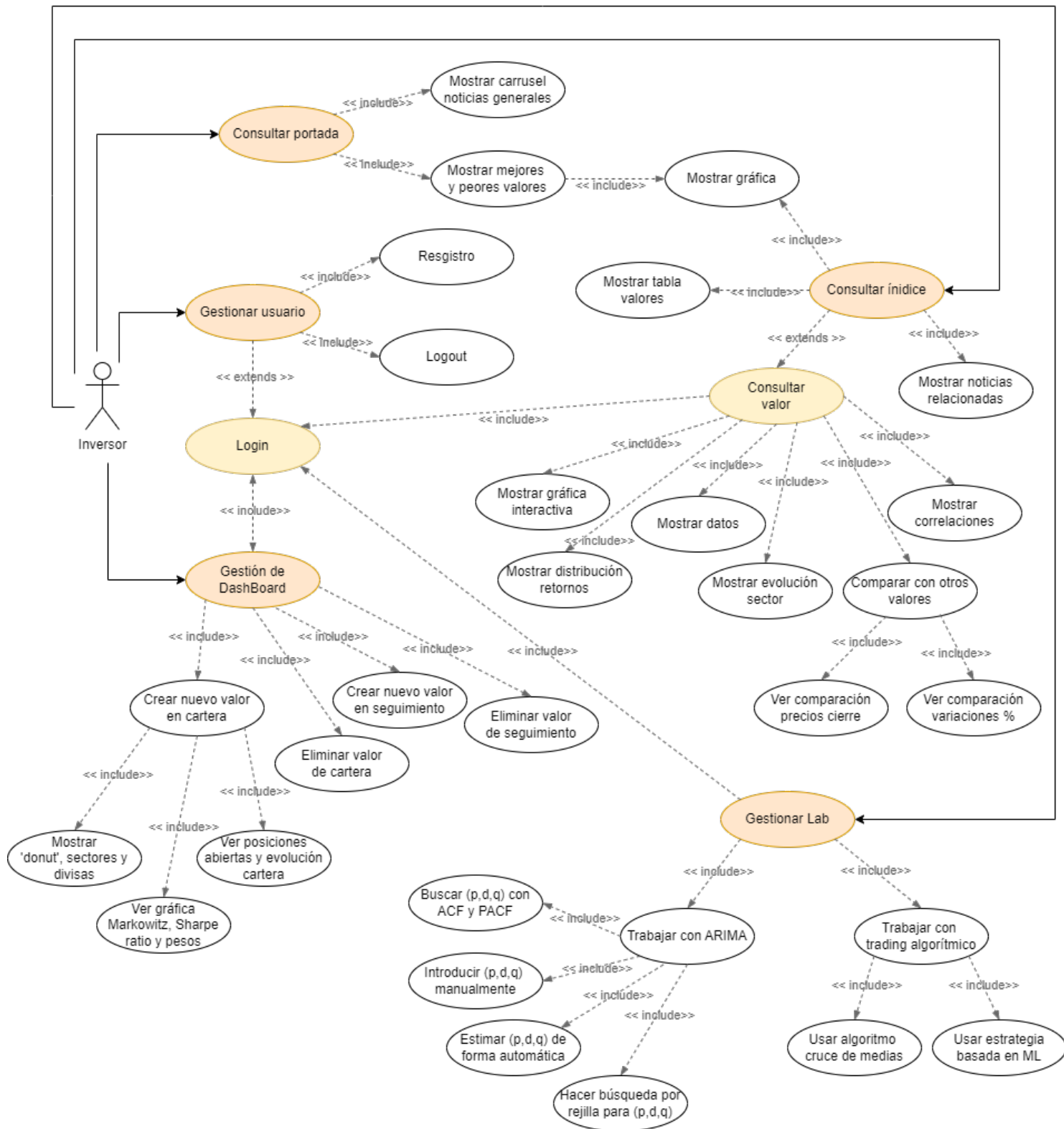


Figura B.1: Diagrama de casos de uso. Fuente: elaboración propia

CU-1	Consultar portada
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-1, RF-1.1, RF-1.2 y RF-1.2.1
<b>Descripción</b>	Permite al usuario ver la página principal
<b>Precondición</b>	La base de datos se encuentra disponible
<b>Acciones</b>	<ol style="list-style-type: none"><li>1. El usuario accede a la página web.</li><li>2. Se muestran noticias del mundo bursátil.</li><li>3. Se listan los índices con los mejores y peores valores</li></ol>
<b>Postcondición</b>	El número de valores mostrados es múltiplo del número de índices disponibles.
<b>Excepciones</b>	<ul style="list-style-type: none"><li>■ Error al cargar valores (mensaje de error).</li><li>■ Error al cargar una noticia (pasar a siguiente).</li></ul>
<b>Importancia</b>	Alta

Tabla B.1: CU-1 Consultar portada.

CU-2	Mostrar carrusel noticias generales
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-1, RF-1.1
<b>Descripción</b>	Mostrar carrusel de noticias bursátiles
<b>Precondición</b>	La API <i>NewsAPI</i> está activa
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario ve texto preliminar de una noticia y su imagen asociada.</li> <li>2. Se <i>navega</i> entre noticias con selectores laterales o se dejan pasar automáticamente.</li> <li>3. Se muestra un botón para <i>leer más</i> en la fuente original.</li> </ol>
<b>Postcondición</b>	La imagen y el enlace externo se corresponden con el texto de la noticia.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Error al cargar una noticia (pasar a siguiente).</li> </ul>
<b>Importancia</b>	Media

Tabla B.2: CU-2 Mostrar carrusel noticias generales.

CU-3	Mostrar mejores y peores valores
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-1, RF-1.2 y RF-1.2.1
<b>Descripción</b>	Mostrar 3 mejores y 3 peores valores de cada índice en la última sesión bursátil.
<b>Precondición</b>	Las bases de datos están disponibles.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario ve valores de cierre de última sesión de mejores valores de un índice.</li> <li>2. El usuario ve valores de cierre de última sesión de peores valores de un índice.</li> <li>3. Se muestra gráfica asociada a cada valor.</li> </ol>
<b>Postcondición</b>	Las gráficas se corresponden con los valores mejores y peores y se muestran en orden de mejor a peor
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Error al cargar un valor (mensaje de error).</li> </ul>
<b>Importancia</b>	Media

Tabla B.3: CU-3 Mostrar mejores y peores valores.

CU-4	Mostrar gráfica
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-1, RF-1.2 y RF-1.2.1
<b>Descripción</b>	Mostrar gráficas de valores o índices.
<b>Precondición</b>	Las bases de datos están disponibles.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario ve gráficas de última sesión de mejores valores de un índice.</li> <li>2. El usuario ve gráficas de última sesión de peores valores de un índice.</li> <li>3. Se muestran valores de cierre asociados a cada valor.</li> <li>4. El usuario consulta gráfica de índice.</li> </ol>
<b>Postcondición</b>	Las gráficas se corresponden con los valores mejores y peores y se muestran en orden de mejor a peor. O la gráfica del índice consultado está disponible.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Error al cargar un valor o índice (mensaje de error).</li> </ul>
<b>Importancia</b>	Media

Tabla B.4: CU-4 Mostrar gráfica.



CU-5	Gestionar usuario
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-2, RF-2.1, RF-2.2 y RF-2.3
<b>Descripción</b>	Controlar sesión de usuario.
<b>Precondición</b>	La base de datos de usuarios está disponible.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario se registra.</li> <li>2. El usuario hace <i>login</i>.</li> <li>3. Diferentes acciones dependiendo de la labor a realizar.</li> <li>4. El usuario hace <i>logout</i> para cerrar sesión.</li> </ol>
<b>Postcondición</b>	Tras <i>logout</i> la cookie de sesión se inhabilita.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Error al registrar por contraseña o nombre (mensaje informativo).</li> <li>■ Error en <i>login</i> por contraseña o nombre (mensaje informativo).</li> </ul>
<b>Importancia</b>	Alta

Tabla B.5: CU-5 Gestionar usuario.

CU-6	Registro
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-2 y RF-2.1
<b>Descripción</b>	Permitir registro de usuarios.
<b>Precondición</b>	El usuario no está registrado.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario se registra.</li> <li>2. Mostrar un mensaje de bienvenida.</li> <li>3. Dejar usuario ya activo con <i>login</i>.</li> <li>4. Redirigir a página principal.</li> </ol>
<b>Postcondición</b>	Nuevo usuario en base de datos.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Error al registrar por contraseña o nombre (mensaje informativo).</li> <li>■ Error por usuario duplicado (mensaje informativo).</li> </ul>
<b>Importancia</b>	Alta

Tabla B.6: CU-6 Registro.

CU-7	Login
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-2 y RF-2.2
<b>Descripción</b>	Permitir <i>login</i> de usuarios.
<b>Precondición</b>	El usuario no está logueado.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario hace <i>login</i>.</li> <li>2. Mostrar <i>DashBoard</i>.</li> <li>3. Permitir acceso a funciones adicionales.</li> </ol>
<b>Postcondición</b>	Usuario logueado y con acceso a funciones adicionales.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Error al loguear (mensaje informativo y repetición).</li> <li>■ Error forzado (mensaje informativo).</li> </ul>
<b>Importancia</b>	Alta

Tabla B.7: CU-6 Login.

CU-8	Logout
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-2 y RF-2.3
<b>Descripción</b>	Permitir <i>logout</i> de usuarios.
<b>Precondición</b>	El usuario está logueado.
<b>Acciones</b>	<ol style="list-style-type: none"><li>1. El usuario hace <i>logout</i>.</li><li>2. Mostrar página principal.</li><li>3. No permitir acceso a funciones adicionales.</li></ol>
<b>Postcondición</b>	Usuario no logueado y sin acceso a funciones adicionales.
<b>Excepciones</b>	
<b>Importancia</b>	Baja

Tabla B.8: CU-8 Logout.

CU-9	Gestionar <i>DashBoard</i>
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-3, RF-3.1, RF-3.1.1, RF-3.1.2, RF-3.1.3, RF-3.2, RF-3.3 y RF-3.4
<b>Descripción</b>	Mostrar información agregada de valores en cartera y en seguimiento de un usuario.
<b>Precondición</b>	El usuario está logueado y las bases de datos están disponibles.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario hace <i>login</i>.</li> <li>2. Mostrar página de <i>DashBoard</i> con información de usuario.</li> </ol>
<b>Postcondición</b>	El usuario puede consultar precios, añadir o quitar valores en cartera y añadir o quitar valores en seguimiento.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ No hay valores previos en cartera (se muestran tablas vacías).</li> <li>■ No hay valores previos en seguimiento (se muestran tablas vacías).</li> </ul>
<b>Importancia</b>	Alta

Tabla B.9: CU-9 *DashBoard*.

CU-10	Crear nuevo valor en cartera
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-3, RF-3.1, RF-3.1.1, RF-3.1.2 y RF-3.1.3
<b>Descripción</b>	Añadir un valor a la cartera del usuario.
<b>Precondición</b>	El usuario está logueado y las bases de datos están disponibles.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede al <i>DashBoard</i>.</li> <li>2. Pinchar en botón <i>Nueva posición de cartera</i>.</li> </ol>
<b>Postcondición</b>	El usuario ve un nuevo valor asociado a su cuenta.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Valor no existe (mensaje informativo y reintento).</li> <li>■ Fecha no válida (mensaje informativo y reintento).</li> <li>■ Precio valor no adecuado para la fecha (mensaje informativo y reintento).</li> <li>■ Valores formulario fuera de rango (mensaje informativo y reintento).</li> </ul>
<b>Importancia</b>	Alta

Tabla B.10: CU-10 Crear nuevo valor en cartera.

CU-11	Mostrar <i>donut</i> , sectores y divisas
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-3, RF-3.1 y RF-3.1.1
<b>Descripción</b>	Mostrar información de composición de cartera en gráficas y diagramas que resulten agradables para el usuario.
<b>Precondición</b>	El usuario está logueado, las bases de datos están disponibles y el usuario tiene valores guardados en cartera.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede al <i>DashBoard</i>.</li> <li>2. Mostrar <i>donut</i> con valores en cartera.</li> <li>3. Mostrar diagrama de sectores en los que se está invertido.</li> <li>4. Mostrar inversión por tipo de divisa.</li> <li>5. Ver cambios de divisa de última sesión disponible.</li> </ol>
<b>Postcondición</b>	El usuario ve información relevante sobre el estado de su cartera.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ No hay valores previos en cartera (se muestran tablas vacías).</li> </ul>
<b>Importancia</b>	Media

Tabla B.11: CU-11 Mostrar *donut*, sectores y divisas.

<b>CU-12</b>	<b>Ver gráfica de Markowitz, ratio de Sharpe y pesos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-3, RF-3.1 y RF-3.1.2
<b>Descripción</b>	Mostrar información de distribución de pesos de los valores en cartera.
<b>Precondición</b>	El usuario está logueado, las bases de datos están disponibles y el usuario tiene valores guardados en cartera.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede al <i>DashBoard</i>.</li> <li>2. Mostrar gráfica de Markowitz.</li> <li>3. Ver simulación de Montecarlo de 10.000 carteras.</li> <li>4. Indicar frontera eficiente.</li> <li>5. Mostrar ratio de Sharpe.</li> <li>6. Mostrar información de distribución de pesos en tablas.</li> </ol>
<b>Postcondición</b>	El usuario ve información relevante sobre posibles distribuciones mejores para su cartera.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ No hay valores previos en cartera (se muestran tablas vacías).</li> </ul>
<b>Importancia</b>	Alta

Tabla B.12: CU-12 Ver gráfica de Markowitz, ratio de Sharpe y pesos.



CU-13	Ver posiciones abiertas y evolución de cartera
<b>Versión</b>	1.0
<b>Autor</b>	Rodrigo Merino Tovar
<b>Requisitos asociados</b>	RF-3, RF-3.1 y RF-3.1.3
<b>Descripción</b>	Mostrar información de evolución de inversiones realizadas.
<b>Precondición</b>	El usuario está logueado, las bases de datos están disponibles y el usuario tiene valores guardados en cartera.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede al <i>DashBoard</i>.</li> <li>2. Mostrar evolución individual de valores en cartera.</li> <li>3. Hacer cálculos de cambio de divisa a euros.</li> <li>4. Mostrar evolución de cartera en euros.</li> </ol>
<b>Postcondición</b>	El usuario ve la evolución de su cartera.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ No hay valores previos en cartera (se muestran tablas vacías).</li> </ul>
<b>Importancia</b>	Alta

Tabla B.13: CU-13 Ver posiciones abiertas y evolución de cartera.



## *Apéndice C*

---

# **Especificación de diseño**

---

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico



## *Apéndice D*

---

# **Documentación técnica de programación**

---

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema



## *Apéndice E*

---

# **Documentación de usuario**

---

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario





## *Apéndice F*

---

# **Anexo de sostenibilización curricular**

---

### **F.1. Introducción**

Este anexo incluirá una reflexión personal del alumnado sobre los aspectos de la sostenibilidad que se abordan en el trabajo. Se pueden incluir tantas subsecciones como sean necesarias con la intención de explicar las competencias de sostenibilidad adquiridas durante el alumnado y aplicadas al Trabajo de Fin de Grado.

Más información en el documento de la CRUE [https://www.crue.org/wp-content/uploads/2020/02/Directrices\\_Sostenibilidad\\_Crue2012.pdf](https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sostenibilidad_Crue2012.pdf).

Este anexo tendrá una extensión comprendida entre 600 y 800 palabras.



---

## Bibliografía

---

- [1] NetworkX developers. *NetworkX*. <https://networkx.org/>, 2024. Online; Accedido el 22-Abr-2024.
- [2] Django. *Django documentation*. <https://docs.djangoproject.com/en/5.0/faq/general/>, 2023. Online; Accedido el 20-Abr-2024.
- [3] Pivit Inc. *Zube*. <https://zube.io/>, 2024. Online; Accedido el 22-Abr-2024.
- [4] Wikipedia. *cron (Unix)*. [https://es.wikipedia.org/wiki/Cron\\_\(Unix\)](https://es.wikipedia.org/wiki/Cron_(Unix)), 2024. Online; Accedido el 04-Jun-2024.
- [5] Wikipedia. *Scrum (desarrollo de software)*. [https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)), 2024. Online; Accedido el 20-Abr-2024.



Esta obra está bajo una licencia Creative Commons  
Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional  
([CC BY-NC-SA 4.0 DEED](https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es)).