



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

A quien me parece más adecuado agradecer ...

La frasecilla que me parezca, o la segunda dedicatoria ...

Agradecimientos

Han sido muchas personas las que han ayudado a que este proyecto haya salido adelante En primer lugar agradecérselo a mis tutores ya que con sus constantes revisiones y correcciones tanto la aplicación como la memoria ha ido por el camino adecuado.

A este grupo, a aquel otro ...

Resumen

Aquí deberá ir el resumen del trabajo, cuyo esquema (una frase para cada parte) podría ser:
MOTIVACION: OBJETIVO: TAREAS: RESULTADOS: CONCLUSION:

Abstract

The same 'Resumen' but in English, please ...

Índice general

Índice de cuadros	III
Índice de figuras	V
TODO List	VII
1. Introducción	1
1.1. Introducción	1
1.2. Motivación	1
1.3. Objetivos	1
2. Planificación	3
2.1. Metodología	3
2.2. Fases y costes	4
2.3. Actividades	4
2.4. Recursos	5
2.5. Planificación inicial	5
2.6. Presupuesto inicial	6
2.7. Desviaciones con respecto a la planificación inicial	7
2.8. Coste final	7
2.9. Análisis de riesgos	7
3. Marco Conceptual	11
4. Soluciones Existentes	13
5. Análisis	15
6. Diseño	17
6.1. Diseño	17
7. Implementación	19
7.1. Herramientas de Desarrollo	19
7.2. Despliegue del servidor	19
7.3. Implementación	19
8. Pruebas	21
9. Conclusiones	23
9.1. Aportaciones	23
9.2. Trabajo futuro	23

Appendices	25
Apéndice A. Manual de Instalación	27
Apéndice B. Manual de Usuario	29
Apéndice C. Manual del Desarrollador	31
Bibliografía	35

Índice de cuadros

2.1. Sprints de desarrollo del proyecto previstos.	4
2.2. Tabla probabilidades Barry Boehm.	8
2.3. Tabla impactos Barry Boehm.	8
2.4. Tabla de riesgos del proyecto.	8
2.5. Tabla de planes de prevencion y contingencia de riesgos.	9

Índice de figuras

Todo list

Al principio, está todo por hacer.	1
Revisar que las referencias WEB contengan fecha de último acceso	33
Centralizar items de bibliografía en el .bib	33

Introducción

Al principio, está todo por hacer.

Sobre cómo realizar un TFG, la memoria y todas esas cosas, se puede consultar el libro de García[1].
Suerte!

Por otra parte, las órdenes Linux aparecen siempre así: `$ ls -al`

1.1 Introducción

Nuestra visión del problema, la necesidad, el por qué es interesante hacer esto ...

1.2 Motivación

Qué nos lleva a plantearnos hacer este proyecto ...

1.3 Objetivos

Los objetivos son logros, no lo olvides.

Nuestro objetivo principal

Si hubiese, los secundarios

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 de un proyecto de una manera productiva y eficaz [referencia].

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[referencia], 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 sprint, el producto ha de ser funcional para el cliente. 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 Fases y costes

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

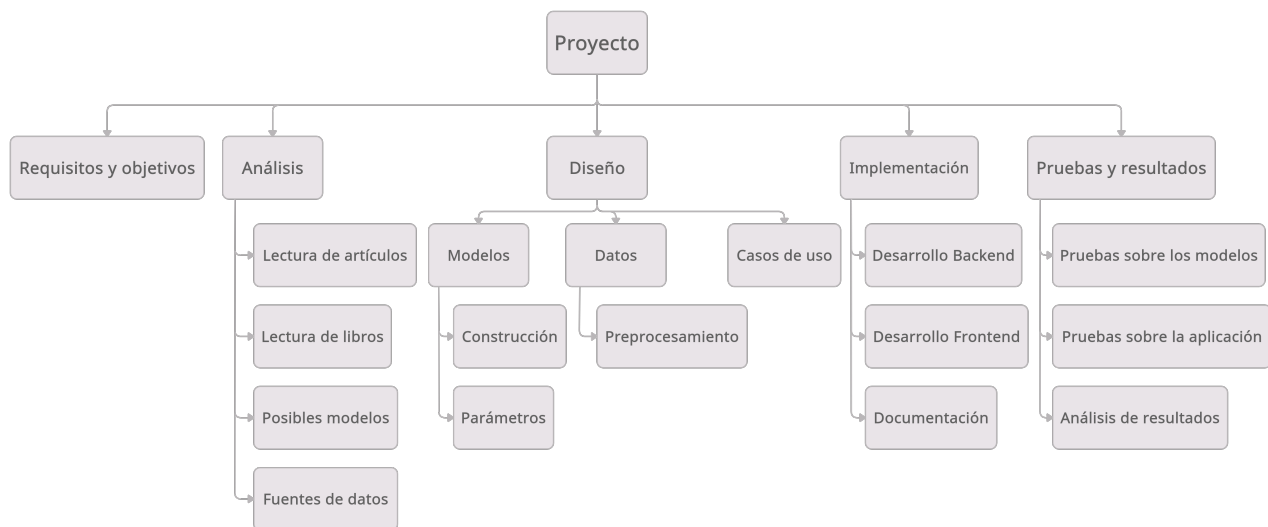
Sprint	Nombre de actividad	Semanas
Sprint 1	Análisis y aprendizaje de las herramientas a utilizar para el despliegue del servidor web	1 - 2
Sprint 1	Despliegue del servidor web	1 - 2
Sprint n	Escritura de la memoria del TFG	16 - 17

Cuadro 2.1: Sprints de desarrollo del proyecto previstos.

2.3 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.



2.4 Recursos

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

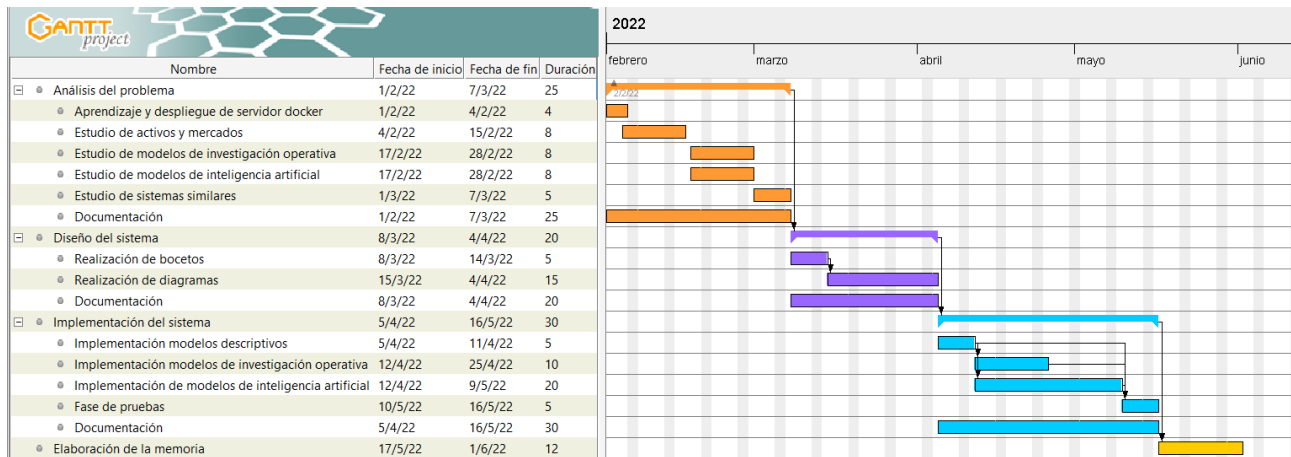
- **Humanos:** el alumno encargado de desarrollar el proyecto en su totalidad.
- **Tecnológicos:** el equipo de trabajo que utilizará el desarrollador, la máquina virtual que alojará la aplicación (proporcionada por la Escuela de Ingeniería Informática de la Universidad de Valladolid), conexión a internet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

2.5 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, papers 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 produciendo 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.



2.6 Presupuesto inicial

Teniendo en cuenta los recursos indicados en 2.4 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 [<https://www.jobted.es/salario/ingeniero-inform%C3%A1tico>], 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, etc).

- 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 [referencia]. 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 4542 mAh que, teniendo en cuenta que el voltaje en España es de 230V da un valor de 1044.66 Wh 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 de la máquina virtual al desconocerse este.
- 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€

2.7 Desviaciones con respecto a la planificación inicial

2.8 Coste final

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 [referencia], consistente en analizar cualitativamente tanto las probabilidades de ocurrencia como los impactos de los riesgos.

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.[referencia]
- **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.

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 2.2: Tabla probabilidades Barry Boehm.

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 2.3: Tabla impactos Barry Boehm.

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 2.4: 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 pandémica 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.	Reelaboración 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.	Reelaboración 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 2.5: Tabla de planes de prevención y contingencia de riesgos.

Capítulo 3

Marco Conceptual

Soluciones Existentes

Capítulo 5

Análisis

Diseño

6.1 Diseño

Implementación

7.1 Herramientas de Desarrollo

7.2 Despliegue del servidor

7.3 Implementación

Pruebas

Conclusiones

9.1 Aportaciones

9.2 Trabajo futuro

Appendices

Apéndice A

Manual de Instalación

Apéndice B

Manual de Usuario

Apéndice C

Manual del Desarrollador

Revisar que las referencias WEB contengan fecha de último acceso

Centralizar items de bibliografía en el .bib

Bibliografía

- [1] J.M.G. García y col. *Cómo escribir un trabajo de fin de grado: algunas experiencias y consejos prácticos*. Colección Síntesis. Editorial Síntesis, 2014. ISBN: 9788490770481. URL: <https://books.google.es/books?id=xpcWogEACAAJ>.

