



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**título del TFG  
Documentación Técnica**



Presentado por Mario Sanz Pérez  
en Universidad de Burgos — 12 de marzo  
de 2024

Tutor: Álvaro Arnaiz González



---

# 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 . . . . .	5
<b>Apéndice B Especificación de Requisitos</b>	<b>7</b>
B.1. Introducción . . . . .	7
B.2. Objetivos generales . . . . .	7
B.3. Catálogo de requisitos . . . . .	7
B.4. Especificación de requisitos . . . . .	7
<b>Apéndice C Especificación de diseño</b>	<b>9</b>
C.1. Introducción . . . . .	9
C.2. Diseño de datos . . . . .	9
C.3. Diseño procedimental . . . . .	9
C.4. Diseño arquitectónico . . . . .	9
<b>Apéndice D Documentación técnica de programación</b>	<b>11</b>
D.1. Introducción . . . . .	11
D.2. Estructura de directorios . . . . .	11
D.3. Manual del programador . . . . .	11

D.4. Compilación, instalación y ejecución del proyecto . . . . .	11
D.5. Pruebas del sistema . . . . .	11
<b>Apéndice E Documentación de usuario</b>	<b>13</b>
E.1. Introducción . . . . .	13
E.2. Requisitos de usuarios . . . . .	13
E.3. Instalación . . . . .	13
E.4. Manual del usuario . . . . .	13
<b>Apéndice F Anexo de sostenibilización curricular</b>	<b>15</b>
F.1. Introducción . . . . .	15
<b>Bibliografía</b>	<b>17</b>

---

## Índice de figuras

---

---

# Índice de tablas

---

B.1. CU-1 Nombre del caso de uso. . . . .	8
---	---

## *Apéndice A*

---

# **Plan de Proyecto Software**

---

### **A.1. Introducción**

Este apéndice tiene como propósito proporcionar una visión detallada del proceso de planificación y estudio de viabilidad que ha sido fundamental en el desarrollo del proyecto de software presentado. A través de una metodología estructurada, se han abordado las diferentes tareas a lo largo del tiempo, asegurando un seguimiento exhaustivo y adaptativo de cada fase del proyecto. Además, este documento explora en profundidad el estudio de viabilidad realizado, abarcando aspectos tanto legales como económicos. El análisis legal es crucial para asegurar que el proyecto cumpla con todas las normativas y legislaciones aplicables, minimizando así posibles riesgos legales. Por otro lado, el estudio económico proporciona una evaluación detallada de la viabilidad financiera del proyecto, incluyendo estimaciones de costos, análisis de retorno de inversión y proyecciones de rentabilidad a largo plazo.

### **A.2. Planificación temporal**

Como se explica en la sección 4 de la memoria, la metodología a seguir es Scrum, salvando las distancias con el número de personas que suele haber en un contexto habitual, ya que únicamente habrá un desarrollador. El proyecto se elige antes de empezar el curso, únicamente para poder informarse y leer artículos relacionados con el tema del aprendizaje semi supervisado. Al inicio con sprint más largos y a medida que se avanza en el desarrollo los sprints se reducen a una o dos semanas de duración, realizando una reunión al inicio de cada uno.

## Sprint 1

Este sprint corresponde a las fechas entre el 7 de noviembre y el 18 de diciembre. Se comienza con una reunión en la que se establecen las siguientes tareas:

- Crear un repositorio en GitHub [3] donde poder subir los cambios del proyecto, más concretamente, la plantilla de documentación inicial para ir familiarizándose con L<sup>A</sup>T<sub>E</sub>X.
- También se manda terminar de leer el artículo [4] y se asigna una nueva lectura acerca de ensembles [2]. Esto es necesario ya que de entre los algoritmos a implementar, alguno de ellos será un ensemble. Concepto que se desconocía antes de iniciar la lectura.
- Encontrar un programa adecuado para el seguimiento del proyecto que soporte Scrum.

El repositorio se crea siguiendo la plantilla de documentación ya creada en 2016 y publicada en Github. Para poder empezar a familiarizarse con L<sup>A</sup>T<sub>E</sub>X es necesario instalar los programas necesarios en el equipo local. Se prueban TeXstudio y TeXworks como editores, y por gusto y comodidad se elige TeXstudio como editor de archivos. También se instala MiKTeX, que es una distribución de TeX/LaTeX para sistemas operativos Windows. Se continúa con la lectura establecida, adquiriendo conceptos de ensembles, estos son, ¿Qué es el boosting?, ¿Qué es el bagging?, ¿Qué posibilidades hay de combinar varios modelos? entre otros. En cuanto al programa utilizado para el seguimiento de las tareas y sprints, se prueban varios como Zenhub (descartado por ser de pago), Jira y Taiga. El primero era la mejor opción dado su relación con Github, pero al ser de pago se descarta. Entre Jira y Taiga se elige la segunda por su accesibilidad, los numerosos problemas de Jira hacen que la plataforma de Taiga sea la elegida. Esta herramienta se explica en el apartado cuatro de la memoria, pero su sencillez y el hecho de poder comunicarse con GitHub hacen fácil su uso. Aun así, queda abierto a cambiar debido a que no es la herramienta que ofrece más posibilidades.

## Sprint 2

Sprint correspondiente a las fechas entre el 18 de noviembre y el 15 de enero. Se inicia con una reunión previa para establecer las tareas:



- Finalizar la lectura de [2].
- Comenzar la documentación con conceptos teóricos acerca del aprendizaje automático vistos hasta el momento.
- Aprender a usar el entorno de flask en python.

La lectura y documentacion se realiza durante el periodo de vacaciones, mientras que el aprendizaje de flask, se lleva a cabo con la ayuda de la asignatura cursada de Diseño y Mantenimiento del Software, en la que se realiza una web que se implementa con este *framework*.

Se encuentra una aplicación llamada Zappier, la cual permite construir triggers entre aplicaciones. En este caso sirve para que cada vez que se cree una *issue* en github, se cree como historia de usuario en el product backlog de Taiga, donde despues se gestionará independientemente. Esta aplicación está de nuevo explicada en el apartado 4 de la memoria.

## Sprint 3

Sprint correspondiente a las fechas entre el 15 de enero y el 1 de febrero. Se tiene una reunión previa para asignar las tareas de:

- Lectura del algoritmo Co-Forest [1] y su pseudocódigo.
- Búsqueda de trabajos relacionados para coger ideas propias para el proyecto.
- Continuar documentando los conceptos teóricos (ensembles, Co-Forest).

En este periodo se completa la lectura del artículo del algoritmo Co-Forest y del apartado del trabajo de Patricia y sus estudios relacionados con este algoritmo. El principal estudio que realiza consiste en resolver un error del pseudocódigo, donde un valor podía coger el valor 0 cuando se utiliza como divisor. Mediante tres propuestas, se inicializa este valor con diferentes operaciones y se muestran varias gráficas para poder evaluar la mejor opción.

Se realiza una búsqueda amplia de aplicaciones web para visualización de algoritmos, apuntando y explicando las más interesantes en el apartado 6 de la memoria. Se realiza la implementación de la técnica de Scrum en el apartado 4 de la memoria. Se deja para sprint posteriores la documentación del CoForest, ya que puede que los conceptos adquiridos no sean los correctos hasta que no se haga su implementación y se vean los resultados.

## **Sprint 4 (RESUMIDO, HAY QUE COMPETAR)**

Sprint correspondiente a las fechas entre el 1 de febrero y el 14 de febrero.. Primera implementacion del algoritmo Co-forest. Importante utilizar el estudio realizado por Patricia el año pasado para ayudar en la implementacion(COMPLETAR CON SU ESTUDIO REAL)(ESTA RESUMIDO, EXPLICAR EXTENSO) Primeras veces los resultados no son muy buenos. Ademas de la busqueda de paginas anteriores, se encuentra una muy buena opcion, la cual es <https://ml-visualizer.herokuapp.com/>. Se actualiza documentacion incluyendo este apartado.

## **Sprint 5 (RESUMIDO, HAY QUE COMPETAR)**

Otras dos semanas aproximadamente(algo mas). Se meustra al tutor la pagina encontrada, se acuerda que puede ser una buena idea para la web y comenzar a mirar opciones de relacionar algoritmo con visualizacion en pantalla, ayudado del trabajo de david del año anterior, la cual realiza una llamada donde calcula todo y devuelve un json que luego interpreta. Se resuelven dudas del algoritmo Coforest y se realiza un estudio para compararlo con el de patricia, comprobando asi su eficiencia.

## **Sprint 6 (RESUMIDO, HAY QUE COMPETAR)**

Se inicia con una reunión el 7 de marzo, surge un cambio de planes, dejando las siguientes tareas por hacer:

- Continuar documentacion (coforest, trabajos relacionados)
- El trabajo sera una version 2.0 del trabajo de David, para no tener que realizar una web que ya esta hecha. Complicaciones fork con github
- Esto implica tener que familiarizarse con el trabajo de David, su estructura, web, etcetera.
- corregir estilo de programacion. Estandarizar mediante la guia de estilo de python.
- Revisar coforest debido a que la comparacion con el de patricia no es del todo buena. randomstate.

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

**Viabilidad económica**

**Viabilidad legal**



## *Apéndice B*

---

# **Especificación de Requisitos**

---

### **B.1. Introducción**

Una muestra de cómo podría ser una tabla de casos de uso:

### **B.2. Objetivos generales**

### **B.3. Catálogo de requisitos**

### **B.4. Especificación de requisitos**

CU-1	Ejemplo de caso de uso
<b>Versión</b>	1.0
<b>Autor</b>	Alumno
<b>Requisitos asociados</b>	RF-xx, RF-xx
<b>Descripción</b>	La descripción del CU
<b>Precondición</b>	Precondiciones (podría haber más de una)
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Pasos del CU</li> <li>2. Pasos del CU (añadir tantos como sean necesarios)</li> </ol>
<b>Postcondición</b>	Postcondiciones (podría haber más de una)
<b>Excepciones</b>	Excepciones
<b>Importancia</b>	Alta o Media o Baja...

Tabla B.1: CU-1 Nombre del caso de uso.

## *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] Ming Li and Zhi-Hua Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE transactions on systems, man, and cybernetics - Part A: Systems and Humans*, 37(6):1088–1098, 2007.
- [2] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45, 2006.
- [3] Mario Sanz Pérez. Tfg-semi-supervised-learning. <https://github.com/msp1015/TFG-Semi-Supervised-Learning>, 2023. Github.
- [4] Jesper E. van Engelen and Holger H. Hoos. A survey on semi supervised learning. *Machine Learning*, 109:373–400, 2020.