



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Aplicación del Aprendizaje
Semisupervisado en el
descubrimiento de ataques a
Sistemas de Recomendación
Documentación Técnica**



Presentado por Patricia Hernando Fernández
en Universidad de Burgos — 21 de octubre
de 2022

Tutor: Álvaro Arnaiz González

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	5
Apéndice B Especificación de Requisitos	7
B.1. Introducción	7
B.2. Objetivos generales	7
B.3. Catalogo de requisitos	7
B.4. Especificación de requisitos	7
Apéndice C Especificación de diseño	9
C.1. Introducción	9
C.2. Diseño de datos	9
C.3. Diseño procedimental	9
C.4. Diseño arquitectónico	9
Apéndice D Documentación técnica de programación	11
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
Apéndice E Documentación de usuario	13
E.1. Introducción	13
E.2. Requisitos de usuarios	13
E.3. Instalación	13
E.4. Manual del usuario	13
Bibliografía	15

Índice de figuras

A.1. <i>Burndown Report Sprint 01</i>	2
A.2. <i>Burndown Report Sprint 02</i>	3

Índice de tablas

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

Apéndice A

Plan de Proyecto Software

A.1. Introducción

A.2. Planificación temporal

Planificación por *sprints*

Sprint 1:

- *Planning meeting*

Durante la reunión se marcaron los siguientes objetivos:

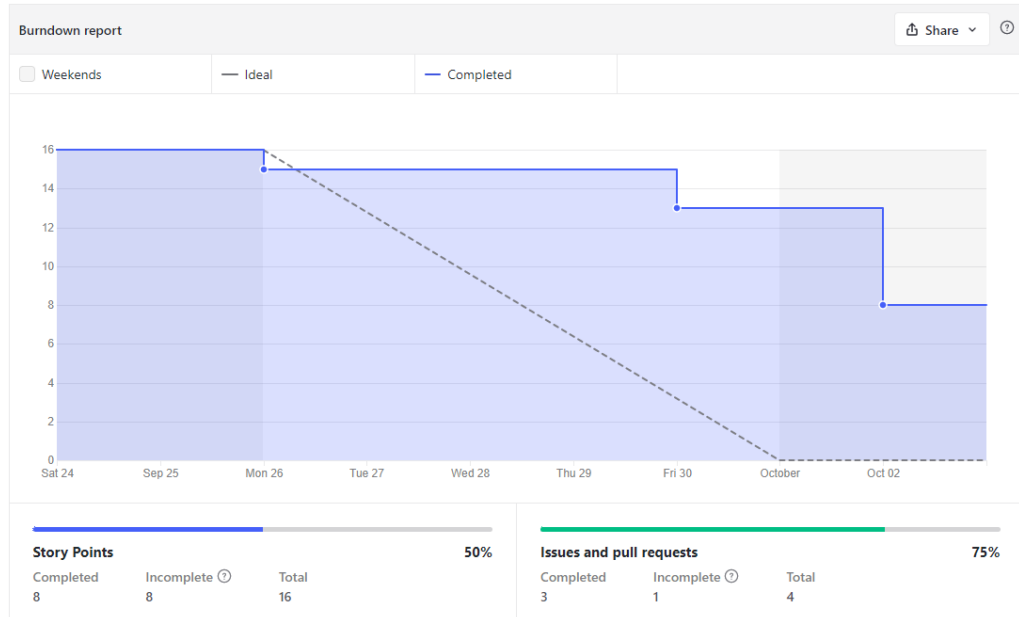
1. Configuración básica: incluyendo la creación del repositorio, la correcta instalación de ZenHub, la creación de entornos virtuales (miniconda, SKLearn, etc.) y la familiarización con conceptos *scrum*: *milestones*, *sprints*, *epics*, etc.
2. Memoria: comienzo de la redacción incluyendo las secciones de introducción, conceptos teóricos (aprendizaje automático) y trabajo relacionado.
3. Investigación: búsqueda del código SSADR-CoF y de las bases de datos utilizadas en el paper.
4. Lectura de papers: Engelen y Hoos [1], García, Triguero y Herrera [4], y Zhou y Duan [5].

- **Marcas temporales**

El *sprint* se desarrolló entre el 24 de septiembre de 2022 y el 2 de octubre del 2022.

■ *Burndown Report*

Figura A.1: *Burndown Report Sprint 01*



Como se puede comprobar, no todos los objetivos marcados fueron cumplidos: la estimación del tiempo fue demasiado optimista, además de no contar con el tiempo requerido en solucionar problemas técnicos (L^AT_EX). Se dejó para próximos sprints la lectura del último paper.

- ***Sprint review meeting*** Durante la reunión se fijaron ciertas correcciones en la memoria (mejorar referencias bibliográficas y la sección de «Trabajos relacionados»), además de la necesidad de introducir una sección teórica de ataques a los sistemas de recomendación.

Sprint 2:

■ *Planning meeting*

Objetivos del siguiente Sprint:

1. Configuración: debido a la gran cantidad de tiempo invertida en solucionar errores de compilación en L^AT_EX, se decidió migrar el proyecto a una nueva instalación basada en Debian.

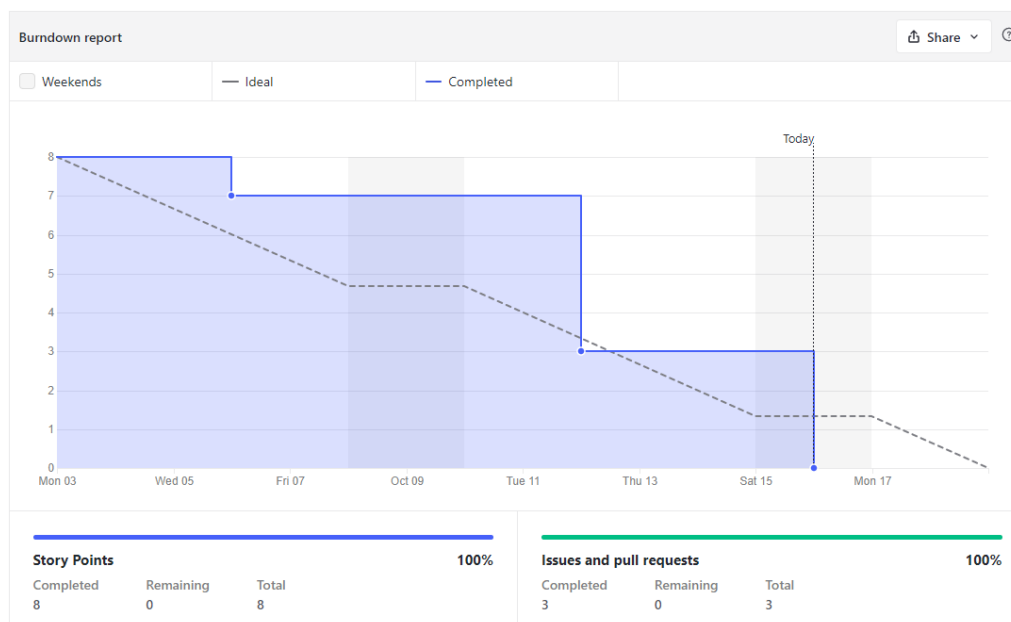
2. Correcciones: aspectos estilísticos y completar información.
3. Lectura: Mingdan y Qingshan [3] con el objetivo de introducir una sección teórica de ataques.
4. Memoria: redacción completa de los modelos de ataque en los aspectos teóricos.

■ Marcas temporales

El *sprint* se desarrolló entre el 03 de octubre de 2022 y el 18 de octubre del 2022.

■ *Burndown Report*

Figura A.2: *Burndown Report Sprint 02*



En este *sprint* sí se cumplió con los objetivos marcados. Sin embargo, la estimación de tiempo tampoco fue la adecuada, requiriendo más de lo previsto.

- ***Sprint review meeting*** Durante la reunión se resolvieron dudas acerca de bibliografía, referencias y trabajo previo. Además, se acordó empezar a programar, definiendo así los *issues* desarrollados en el siguiente *sprint*.

Sprint 3:■ ***Planning meeting***

Durante esta reunión, se decidió empezar a programar el *co-forest*. Para ello, se definieron los siguientes pasos:

1. Librerías: se acordó aprender a utilizar las librerías más comunes en el *data science*. Entre ellas: Matplotlib, Numpy, Pandas, etc. Además, se requirió la correcta configuración del entorno virtual, haciendo que el tiempo dedicado al *issue* fuese mayor de lo estimado (problemas en el *PATH* y con las dependencias).
2. *SKLearn*: aprovechando la correcta documentación de la librería, se decidió repasar los conceptos teóricos básicos, además del manejo de la «interfaz» (métodos comunes). Entre ellos:
 - *Decision trees*
 - *Self training*
 - *Random Forest*
3. Lectura: se concertó la relectura del artículo de Zhou [5] con la intención de comprender el algoritmo y del *paper* «original» del *co-forest* [2].
4. Documentación: se pactó la corrección de los errores previamente señalados y la inclusión del *sprint* en los anexos.

- **Marcas temporales** El *sprint* se desarrolló entre el 19 de octubre de 2022 y el 02 de noviembre del 2022.

■ ***Burndown Report***■ ***Sprint review meeting******Sprint N:***■ ***Planning meeting***■ **Marcas temporales**■ ***Burndown Report***■ ***Sprint review meeting***

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. Catalogo de requisitos

B.4. Especificación de requisitos

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

Bibliografía

- [1] Jesper Engelen and Holger Hoos. A survey on semi-supervised learning. *Machine Learning*, 109, 02 2020.
- [2] 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.
- [3] Si Mingdan and Qingshan Li. Shilling attacks against collaborative recommender systems: a review. *Artificial Intelligence Review*, 53, 01 2018.
- [4] Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*, 42, 02 2015.
- [5] Quanqiang Zhou and Liangliang Duan. Semi-supervised recommendation attack detection based on co-forest. *Comput. Secur.*, 109(C), oct 2021.