



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Sistema de Recomendación  
basado en Aprendizaje  
Profundo  
Documentación Técnica**



Presentado por Raúl Negro Carpintero  
en Universidad de Burgos — 24 de abril  
de 2019

Tutor: Bruno Baruque Zanón



---

# 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 . . . . .	1
<b>Apéndice B Especificación de Requisitos</b>	<b>3</b>
B.1. Introducción . . . . .	3
B.2. Objetivos generales . . . . .	3
B.3. Catalogo de requisitos . . . . .	3
B.4. Especificación de requisitos . . . . .	4
<b>Apéndice C Especificación de diseño</b>	<b>5</b>
C.1. Introducción . . . . .	5
C.2. Diseño de datos . . . . .	5
C.3. Diseño procedimental . . . . .	6
C.4. Diseño arquitectónico . . . . .	6
<b>Apéndice D Documentación técnica de programación</b>	<b>9</b>
D.1. Introducción . . . . .	9
D.2. Estructura de directorios . . . . .	9
D.3. Manual del programador . . . . .	9

D.4. Compilación, instalación y ejecución del proyecto . . . . .	9
D.5. Pruebas del sistema . . . . .	9
<b>Apéndice E Documentación de usuario</b>	<b>11</b>
E.1. Introducción . . . . .	11
E.2. Requisitos de usuarios . . . . .	11
E.3. Instalación . . . . .	11
E.4. Manual del usuario . . . . .	11
<b>Bibliografía</b>	<b>13</b>

---

# Índice de figuras

---

C.1. Esquema del patrón MVC . . . . .	7
C.2. Diagrama UML de la parte de LightFM . . . . .	8

---

## Índice de tablas

---

## *Apéndice A*

---

# **Plan de Proyecto Software**

---

**A.1. Introducción**

**A.2. Planificación temporal**

**A.3. Estudio de viabilidad**

Viabilidad económica

Viabilidad legal





## *Apéndice B*

---

# Especificación de Requisitos

---

### B.1. Introducción

Este anexo recoge los objetivos generales y la especificación de requisitos del proyecto.

### B.2. Objetivos generales

El proyecto persigue los siguientes objetivos generales:

- Comprender los sistemas de recomendación tanto clásicos como basados en aprendizaje profundo.
- Recoger y evaluar los resultados obtenidos por los dos modelos sobre diferentes conjuntos de datos.
- Comparar los resultados.

### B.3. Catalogo de requisitos

Los requisitos derivados de los objetivos del proyecto son los siguientes:

#### Requisitos funcionales

- **RF-1 Gestión de usuarios:** el programa tiene que ser capaz de gestionar los nuevos usuarios.

- **RF-1.1 Añadir usuarios:** el programa tiene que ser capaz de añadir nuevos usuarios.
- **RF-1.2 Modificar usuarios:** el programa tiene que ser capaz de modificar las características de los usuarios y/o sus valoraciones.
- **RF-1.3 Eliminar usuarios:** el programa tiene que ser capaz de eliminar usuarios.
- **RF-1.4 Listar usuarios:** el programa tiene que ser capaz de listar usuarios.
- **RF-2 Gestión de los datos:** el programa tiene que ser capaz de gestionar los datos.
  - **RF-2.1 Listar datos:** el programa tiene que ser capaz de listar el contenido de los conjuntos de datos.
  - **RF-2.2 Seleccionar datos:** el programa tiene que ser capaz de seleccionar el conjunto de dato que el usuario quiera.
- **RF-3 Gestión de los resultados:** el programa tiene que ser capaz de gestionar los resultados.
- **RF-4 Gestión de los modelos:** el programa tiene que ser capaz de gestionar los modelos.
- **RF-5 Ayuda de la aplicación:** el usuario debe poder obtener ayuda sobre las funcionalidades del programa.

## Requisitos no funcionales

- **RNF-1 Usabilidad:** la interfaz gráfica tiene que ser intuitiva y fácil de usar.
- **RNF-2 Soporte:** el programa tiene que dar soporte a versiones iguales o mayores a Python 3.
- **RNF-4 Localización:** el programa tiene que estar preparado para soportar varios idiomas.

## B.4. Especificación de requisitos

## Apéndice C

---

# Especificación de diseño

---

### C.1. Introducción

En este apéndice se explica cómo están conformados los datos que utilizan las librerías usadas en el proyecto, así como la forma en la que está estructurado el mismo.

### C.2. Diseño de datos

Todos los datos que he utilizado a lo largo del proyecto están en formato .csv. Lo más normal es que para cada conjunto de datos tenga los siguientes archivos:

- *ratings.csv*
- *users.csv*
- *items.csv*

La estructura de estos archivos suele ser: *idUser*, *idItem*, *rating*, *timestamp* para *ratings.csv*, *idUser*, *name*, *feature1*, ..., *featureN* para *users.csv* y *idItem*, *name*, *feature1*, ..., *featureN* para *items.csv*.

Para poder trabajar con los datos primero los paso a *DataFrames* de *pandas* [?].

## Datos con LightFM

Una vez obtenidos los *DataFrames* para cada *.csv*, necesito convertirlos a *Dataset* de *LightFM* [?] para poder trabajar con ellos.

Esta clase se encarga de convertir los datos almacenados en los *DataFrames* en *matrices COO* y *matrices CSR*.

## C.3. Diseño procedimental

## C.4. Diseño arquitectónico

Para la realización de este proyecto se ha seguido el patrón arquitectónico MVC (*Modelo Vista Controlador*). El objetivo de este patrón es dividir el código en función de su propósito. Sus partes son:

- *Modelo*: el acceso a los datos. Se corresponde con las clases de Entrada y Salida, que leen los datos para dárselo al sistema de recomendación y guardan los resultados.
- *Vista*: la visualización de los datos. Se corresponde con las clases de Interfaz, que muestran la información solicitada.
- *Controlador*: la manipulación de los datos. Se corresponde con las clases de Sistema, que crean los sistemas de recomendación gracias a los datos proporcionados por las clases de Entrada.

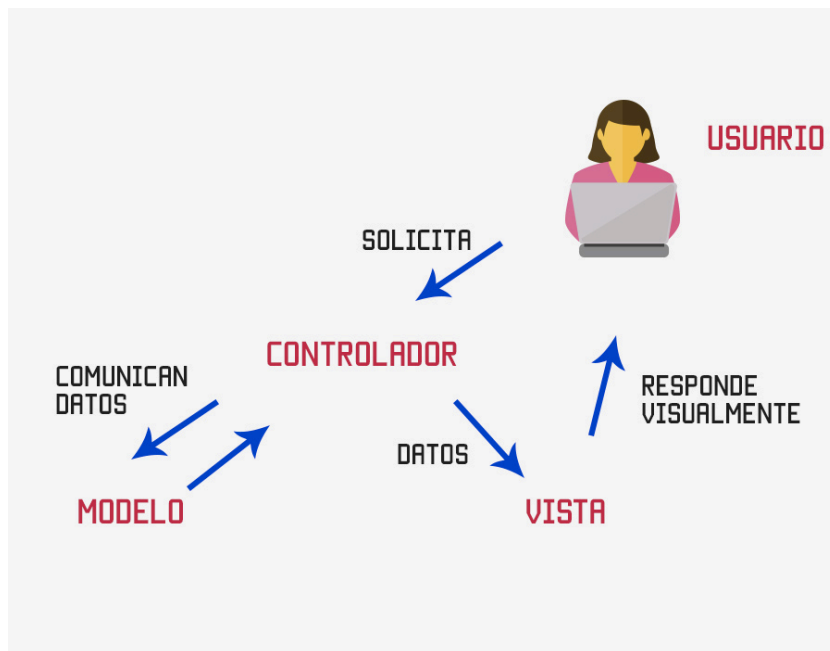


Figura C.1: Esquema del patrón MVC

## Diseño con LightFM

Para la parte del modelo clásico, la estructura del proyecto es la siguiente:

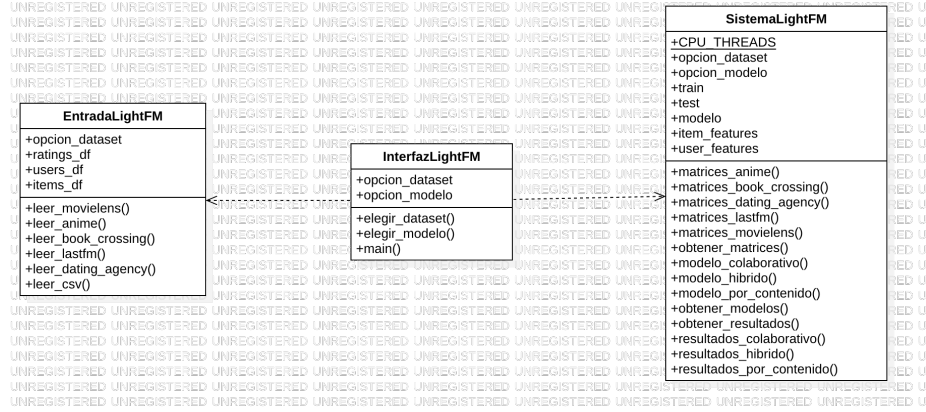


Figura C.2: Diagrama UML de la parte de LightFM

En la clase `EntradaLightFM` tenemos:

- `leer_csv`: este método llama a los distintos métodos de lectura para recoger los datos de los `.csv` en función del conjunto de datos escogido mediante la clase `InterfazLightFM`.
- `leer_x`: estos métodos recogen los datos de los `.csv` para cada conjunto de datos.

En la clase `InterfazLightFM` tenemos:

- `elegir_dataset`: este método muestra un menú mediante el cual elegimos un conjunto de datos que utilizar.
- `elegir_modelo`: este método muestra un menú mediante el cual elegimos un modelo concreto a crear.
- `main`: método principal del programa. Es el encargado de llamar a todos los métodos necesarios de la clase `SistemaLightFM`.

En la clase `SistemaLightFM` tenemos:

- **obtener\_matrices**: este método llama a los distintos métodos de creación de las matrices necesarias para poder obtener los modelos en función del conjunto de datos escogido mediante la clase **InterfazLightFM**.
- **matrices\_x**: estos métodos crean las matrices necesarias para cada conjunto de datos.
- **obtener\_modelos**: este método llama a los distintos métodos de creación de modelos en función del modelo escogido mediante la clase **InterfazLightFM**.
- **modelo\_x**: estos métodos crean los distintos modelos de recomendación.
- **obtener\_resultados**: este método llama a los métodos de obtención de resultados en función del modelo escogido mediante la clase **InterfazLightFM**.
- **resultados\_x**: estos métodos obtienen los resultados para cada modelo de recomendación.





## *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**

---