



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 — 1 de julio  
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 . . . . .	2
A.3. Estudio de viabilidad . . . . .	4
<b>Apéndice B Especificación de Requisitos</b>	<b>5</b>
B.1. Introducción . . . . .	5
B.2. Objetivos generales . . . . .	5
B.3. Catalogo de requisitos . . . . .	5
B.4. Especificación de requisitos . . . . .	7
<b>Apéndice C Especificación de diseño</b>	<b>21</b>
C.1. Introducción . . . . .	21
C.2. Diseño de datos . . . . .	21
C.3. Diseño procedimental . . . . .	22
C.4. Diseño arquitectónico . . . . .	22
<b>Apéndice D Documentación técnica de programación</b>	<b>31</b>
D.1. Introducción . . . . .	31
D.2. Estructura de directorios . . . . .	31
D.3. Manual del programador . . . . .	31

D.4. Compilación, instalación y ejecución del proyecto . . . . .	31
D.5. Pruebas del sistema . . . . .	31
<b>Apéndice E Documentación de usuario</b>	<b>33</b>
E.1. Introducción . . . . .	33
E.2. Requisitos de usuarios . . . . .	33
E.3. Instalación . . . . .	33
E.4. Manual del usuario . . . . .	33
<b>Bibliografía</b>	<b>35</b>

---

## Índice de figuras

---

B.1. Diagrama en caso de obtener un nuevo modelo . . . . .	7
B.2. Diagrama en caso de obtener los resultados y las métricas . . . .	8
B.3. Diagrama en caso de querer añadir valoraciones . . . . .	8
B.4. Diagrama en caso de mostrar la ayuda . . . . .	9
C.1. Esquema del patrón MVC [2] . . . . .	23
C.2. Diagrama UML del proyecto . . . . .	24
C.3. Diagrama UML del paquete <i>vista</i> . . . . .	25
C.4. Diagrama UML del paquete <i>controlador</i> . . . . .	26
C.5. Diagrama UML del paquete <i>modelo</i> . . . . .	27

---

# Índice de tablas

---

B.1. CU-01 Gestión de los datos intermedios . . . . .	10
B.2. CU-02 Guardar matrices . . . . .	10
B.3. CU-3 Cargar matrices . . . . .	11
B.4. CU-4 Gestión de los modelos . . . . .	11
B.5. CU-5 Mostrar modelos . . . . .	12
B.6. CU-6 Seleccionar modelos . . . . .	12
B.7. CU-7 Guardar modelos . . . . .	13
B.8. CU-8 Cargar modelos . . . . .	13
B.9. CU-9 Gestión de los resultados . . . . .	14
B.10.CU-10 Guardar resultados . . . . .	14
B.11.CU-11 Mostrar resultados . . . . .	15
B.12.CU-12 Mostrar predicciones . . . . .	15
B.13.CU-13 Gestión de los usuarios . . . . .	16
B.14.CU-14 Añadir valoraciones . . . . .	17
B.15.CU-15 Gestión de los conjuntos de datos . . . . .	17
B.16.CU-16 Mostrar los conjuntos de datos . . . . .	18
B.17.CU-17 Seleccionar el conjunto de datos . . . . .	18
B.18.CU-18 Añadir conjunto de datos . . . . .	19
B.19.CU-19 Ayuda de la aplicación . . . . .	19
B.20.CU-20 Mostrar información sobre los modelos . . . . .	20
B.21.CU-21 Mostrar información sobre los conjuntos de datos . . . . .	20

## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

La planificación es un punto fundamental en cualquier proyecto software. En ella, se tiene que estimar qué cosas hay que hacer y cuánto tiempo y dinero va a llevar terminirlas. De esta manera, se podrá comprobar si se llega a tiempo a los distintos plazos y así evitar retrasos, lo que supondría pérdidas de dinero y tiempo.

La planificación se puede dividir en dos partes clave:

- **Planificación temporal:** consiste en dividir el proyecto en etapas (*sprints*) con una duración de x días. En estas etapas hay que dejar claro qué cosas se quieren conseguir y cuánto tiempo o esfuerzo se va a dedicar a cada una de ellas. Los *sprints* no tienen por qué durar siempre lo mismo, se pueden ajustar a las necesidades de cada momento.
- **Estudio de viabilidad:** con este estudio se podrá ver si el proyecto puede seguir adelante (es viable) o si no, con lo que habría que tomar medidas para cambiar la situación. Se divide en:
  - **Viabilidad económica:** según la cual se deberá calcular por cuánto tiene que venderse el servicio para compensar el dinero, el tiempo y el esfuerzo puestos en el proyecto. Se incluye salario del personal y la compra de software de terceros necesario para la realización del proyecto.
  - **Viabilidad legal:** se estudia el licenciamiento del software que se va a utilizar para proceder dentro de la legalidad.

## A.2. Planificación temporal

Para el desarrollo del proyecto se decidió seguir la metodología ágil *Scrum*. Al ser un proyecto académico con una sola persona trabajando en él, no se siguen a raja tabla todos los rasgos característicos de *Scrum*. Las características que se han seguido son:

- Desarrollo incremental en cada *sprint*.
- Cada *sprint* tiene una duración aproximada de dos semanas.
- Se realizan reuniones al final de cada *sprint* con el objetivo de repasar lo hecho y pensar en los objetivos del siguiente *sprint*.

A continuación se desarrolla el contenido de cada *sprint*.

### Sprint 1 (- 13/12/2018)

Este *sprint* se corresponde con el milestone **Búsqueda de información inicial**. En él se estudia el capítulo dedicado a los sistemas de recomendación del libro *Mining of Massive Datasets* [5].

También se realiza el curso de *fast.ai* [1]. Además, se escoge la librería de *LightFM* y se obtiene un modelo inicial.

### Sprint 2 (13/12/2018 - 18/01/2019)

Este *sprint* se corresponde con el milestone **Explorar modelo con LightFM**. En él se intenta comprender el funcionamiento de *LightFM* y se utiliza el conjunto de datos de *Movielens* con el modelo.

### Sprint 3 (18/01/2019 - 16/02/2019)

Este *sprint* se corresponde con el milestone **Recomendación híbrida y por contenido con LightFM**. En él se obtienen las primeras versiones del modelo híbrido y por contenido de *LightFM*.

También se obtiene un modelo inicial con *PyTorch* y se empieza a pensar en las métricas que se usarán para evaluar los modelos.

### Sprint 4 (16/02/2019 - 03/03/2019)

Este *sprint* se corresponde con el milestone **Trabajar en la documentación**. En él se empieza a ampliar los apartados de la documentación relativos a los conjuntos de datos utilizados.



**Sprint 5 (03/03/2019 - 13/03/2019)**

Este *sprint* se corresponde con el milestone **División de los datos en LightFM**. En él se dividen los conjuntos de datos en *train* y *test* y se obtienen de nuevo los modelos.

**Sprint 6 (13/03/2019 - 28/03/2019)**

Este *sprint* se corresponde con el milestone **Medidas de calidad LightFM**. En él se obtienen las métricas finales para los modelos de *LightFM*.

Además, se guardan las matrices y los modelos en archivos *pickle* como medida de persistencia.

**Sprint 7 (28/03/2019 - 05/04/2019)**

Este *sprint* se corresponde con el milestone **Últimos pasos con LightFM**. En él se intenta acabar todo lo relacionado con los modelos clásicos (tanto código como documentación).

**Sprint 8 (05/04/2019 - 24/04/2019)**

Este *sprint* se corresponde con el milestone **Ampliar documentación**. En él se amplía la documentación relativa a *LightFM* y se empiezan a obtener los primeros diagramas UML con la estructura del proyecto hasta el momento.

**Sprint 9 (24/04/2019 - 08/05/2019)**

Este *sprint* se corresponde con el milestone **Cambios en el diseño, Flask y DL**. En él se empieza a obtener la interfaz web con *Flask* y se escoge *Spotlight* como la librería de aprendizaje profundo.

También se modifica la documentación para ir acorde a los cambios del código.

**Sprint 10 (08/05/2019 - 22/05/2019)**

Este *sprint* se corresponde con el milestone **Primeros pasos con Spotlight**. En él se obtiene un primer modelo de *Spotlight* y se unifican los archivos de Entrada y Persistencia para que solo haya uno de cada.

Se continúa trabajando en la documentación.

**Sprint 11 (22/05/2019 - 05/06/2019)**

Este *sprint* se corresponde con el milestone **Completar modelos de Spotlight**. En él se obtienen el resto de modelos de *Spotlight* y se generalizan los métodos de lectura de datos y obtención de matrices.

Se continúa trabajando en la documentación.

**Sprint 12 (05/06/2019 - 12/06/2019)**

Este *sprint* se corresponde con el milestone **Diseño GUI**. En él se piensa qué forma va a tener la interfaz web obtenida a través de *Flask*.

**Sprint 13 (12/06/2019 - 20/06/2019)**

Este *sprint* se corresponde con el milestone **Obtención GUI**. En él se termina de crear la interfaz web y se empiezan a emplear los métodos obtenidos anteriormente para que la aplicación sea funcional.

**Sprint 14 (20/06/2019 - 03/07/2019)**

Este *sprint* se corresponde con el milestone **Último milestone**. En él se termina todo lo que queda por hacer en el proyecto. La interfaz es totalmente funcional y la documentación está terminada.

**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 los datos intermedios:** el programa tiene que ser capaz de gestionar los datos intermedios:

- **RF-1.1 Guardar matrices:** el programa tiene que ser capaz de guardar las matrices generadas por los modelos para ahorrar tiempo en siguientes ejecuciones.
- **RF-1.2 Cargar matrices:** el programa tiene que ser capaz de cargar las matrices que el usuario quiera.
- **RF-2 Gestión de los modelos:** el programa tiene que ser capaz de gestionar los modelos:
  - **RF-2.1 Mostrar modelos:** el programa tiene que ser capaz de mostrar todos los tipos de modelos disponibles.
  - **RF-2.2 Seleccionar modelos:** el programa tiene que ser capaz de dejar al usuario seleccionar el tipo de modelo que quiera.
  - **RF-2.3 Guardar modelos:** el programa tiene que ser capaz de guardar los modelos obtenidos para ahorrar tiempo en futuras ejecuciones.
  - **RF-2.4 Cargar modelos:** el programa tiene que ser capaz de cargar los modelos previamente guardados.
- **RF-3 Gestión de los resultados:** el programa tiene que ser capaz de gestionar los resultados:
  - **RF-3.1 Guardar resultados:** el programa tiene que ser capaz de guardar los resultados generados por los modelos.
  - **RF-3.2 Mostrar resultados:** el programa tiene que ser capaz de mostrar los resultados obtenidos por los distintos modelos.
  - **RF-3.3 Mostrar predicciones:** el programa tiene que ser capaz de mostrar las predicciones obtenidas por los distintos modelos.
- **RF-4 Gestión de los usuarios:** el programa tiene que ser capaz de introducir las valoraciones de nuevos usuarios:
  - **RF-4.1 Añadir valoraciones:** el programa tiene que ser capaz de simular la entrada de nuevos usuarios permitiendo el añadido de nuevas valoraciones.
- **RF-5 Gestión de los conjuntos de datos:** el programa tiene que:
  - **RF-5.1 Mostrar los conjuntos de datos:** el programa tiene que ser capaz de mostrar los distintos conjuntos de datos de prueba.
  - **RF-5.2 Seleccionar el conjunto de datos:** el programa tiene que ser capaz de mostrar dejar que el usuario seleccione uno de los conjuntos de datos de prueba.
  - **RF-5.3 Añadir conjunto de datos:** el programa tiene que ser capaz de dejar que el usuario añada un conjunto de datos propio.

- **RF-6 Ayuda de la aplicación:** el programa tiene que ofrecer información al usuario:
  - **RF-6.1 Mostrar información sobre modelos:** el programa tiene que ser capaz de mostrar información sobre los distintos modelos que se pueden escoger.
  - **RF-6.2 Mostrar información sobre conjuntos de datos:** el programa tiene que ser capaz de mostrar información sobre los distintos conjuntos de datos de prueba.

## 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-3 Localización:** el programa tiene que estar preparado para soportar varios idiomas.

## B.4. Especificación de requisitos

En esta sección se mostrará el diagrama de casos de uso y se desarrollará cada uno de ellos.

### Diagrama de casos de uso

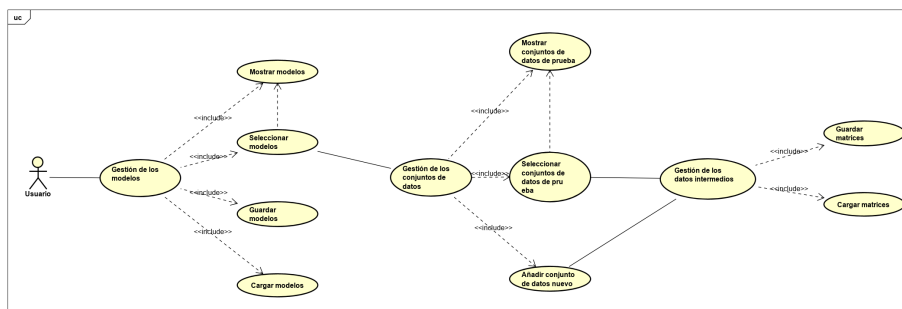


Figura B.1: Diagrama en caso de obtener un nuevo modelo

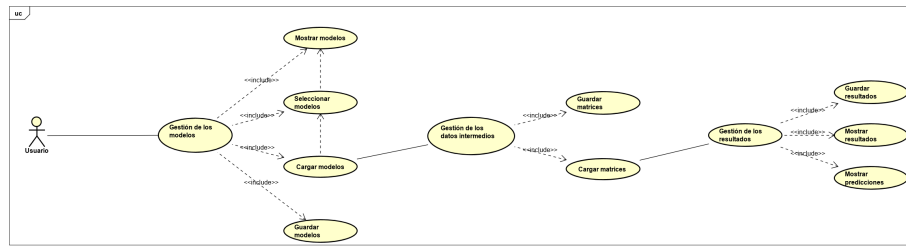


Figura B.2: Diagrama en caso de obtener los resultados y las métricas

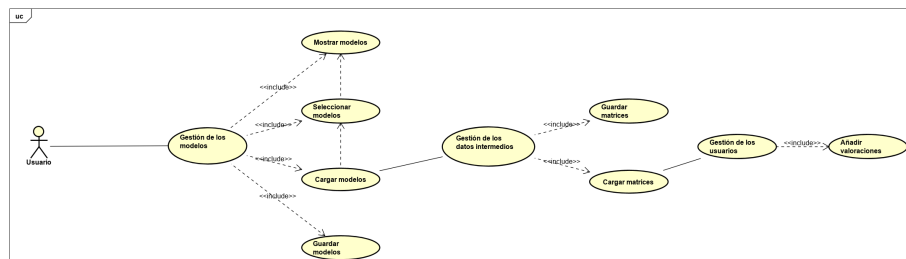


Figura B.3: Diagrama en caso de querer añadir valoraciones

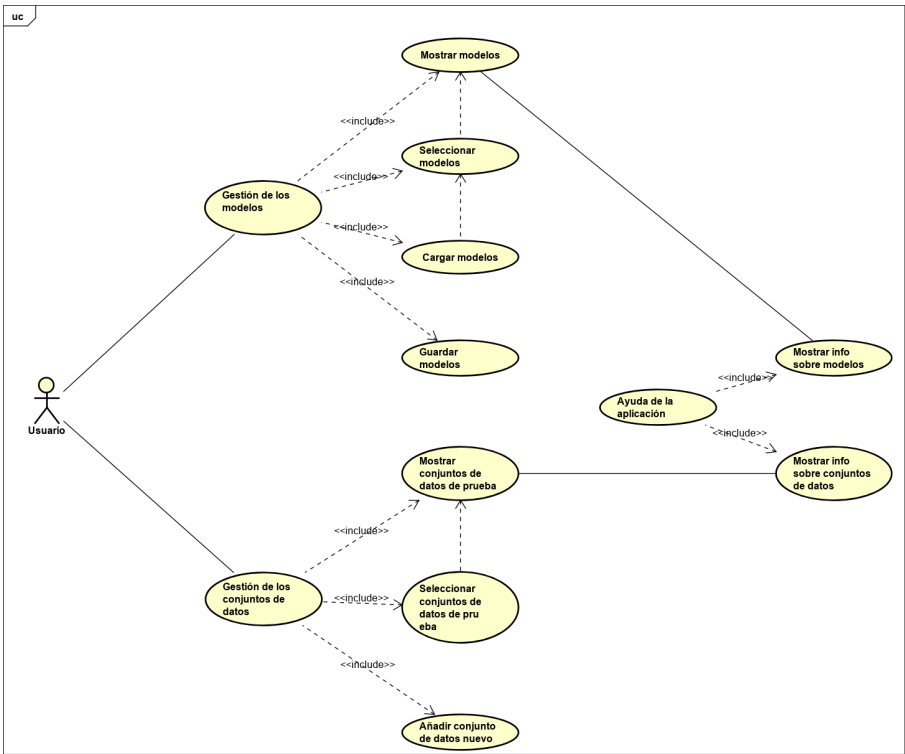


Figura B.4: Diagrama en caso de mostrar la ayuda

Actores

Con la aplicación solo interactuará un actor, el usuario que esté probando la aplicación en un momento determinado.

Casos de uso

A continuación, se desarrollará cada caso de uso:

CU-01	Gestión de los datos intermedios
Versión	1.0

continúa en la página siguiente

continúa desde la página anterior

CU-01		Gestión de los datos intermedios
<b>Autor</b>		Raúl Negro Carpintero
<b>Requisitos asociados</b>		RF-1 & RF-1.1 & RF-1.2
<b>Descripción</b>		Gestión de los datos intermedios.
<b>Precondiciones</b>		Los datos intermedios tienen que existir.
<b>Acciones</b>		El usuario escoge los datos que quiere cargar o guardar.
<b>Postcondiciones</b>		Los datos intermedios se cargan o se guardan.
<b>Excepciones</b>		Los datos intermedios no existen.

Tabla B.1: CU-01 Gestión de los datos intermedios

CU-02		Guardar matrices
<b>Versión</b>		1.0
<b>Autor</b>		Raúl Negro Carpintero
<b>Requisitos asociados</b>		RF-1 & RF-1.1
<b>Descripción</b>		Guardar las matrices de datos.
<b>Precondiciones</b>		Tienen que existir las matrices.
<b>Acciones</b>		El usuario guarda las matrices generadas por el sistema.
<b>Postcondiciones</b>		Las matrices se guardan en archivos <i>.pickle</i>
<b>Excepciones</b>		Los matrices no existen.

Tabla B.2: CU-02 Guardar matrices

CU-03		Cargar matrices
<b>Versión</b>		1.0
<b>Autor</b>		Raúl Negro Carpintero
<b>Requisitos asociados</b>		RF-1 & RF-1.2

continúa en la página siguiente



continúa desde la página anterior

CU-03	Cargar matrices
<b>Descripción</b>	Cargar las matrices de datos.
<b>Precondiciones</b>	Tienen que existir los archivos <i>.pickle</i> con las matrices.
<b>Acciones</b>	El usuario carga las matrices.
<b>Postcondiciones</b>	Las matrices se cargan y se vinculan al sistema.
<b>Excepciones</b>	No hay archivos <i>.pickle</i> con las matrices.

Tabla B.3: CU-3 Cargar matrices

CU-04	Gestión de los modelos
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-2 & RF-2.1 & RF-2.2 & RF-2.3 & RF-2.4
<b>Descripción</b>	Gestión de los modelos.
<b>Precondiciones</b>	Se tienen que haber generado los modelos.
<b>Acciones</b>	El usuario escoge el tipo de modelo que quiere obtener, guardar o cargar.
<b>Postcondiciones</b>	El modelo escogido se crea, guarda o carga.
<b>Excepciones</b>	No hay modelos generados.

Tabla B.4: CU-4 Gestión de los modelos

CU-05	Mostrar modelos
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-2 & RF-2.1
<b>Descripción</b>	Se listan todos los tipos de modelos que se pueden obtener.

continúa en la página siguiente

continúa desde la página anterior

CU-05	Mostrar modelos
<b>Precondiciones</b>	-
<b>Acciones</b>	El usuario piensa que tipo de modelo utilizar.
<b>Postcondiciones</b>	-
<b>Excepciones</b>	No hay tipos de modelos que mostrar.

Tabla B.5: CU-5 Mostrar modelos

CU-06	Seleccionar modelos
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-2 & RF-2.2
<b>Descripción</b>	Se selecciona un modelo.
<b>Precondiciones</b>	Se han mostrado todos los tipos de modelos.
<b>Acciones</b>	El usuario escoge el tipo de modelo que quiere crear, guardar o cargar.
<b>Postcondiciones</b>	El tipo de modelo que se quiere utilizar se queda seleccionado para trabajar con él.
<b>Excepciones</b>	No existe el modelo seleccionado.

Tabla B.6: CU-6 Seleccionar modelos

CU-07	Guardar modelos
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-2 & RF-2.3
<b>Descripción</b>	Se guarda un modelo.
<b>Precondiciones</b>	El modelo que se quiere guardar debe existir.

continúa en la página siguiente

continúa desde la página anterior

CU-07	Guardar modelos
<b>Acciones</b>	Se guarda el modelo en un archivo <i>.pickle</i> .
<b>Postcondiciones</b>	Se obtiene un archivo <i>.pickle</i> con el modelo.
<b>Excepciones</b>	No hay ningún modelo que guardar.

Tabla B.7: CU-7 Guardar modelos

CU-08	Cargar modelos
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-2 & RF-2.4
<b>Descripción</b>	Se carga un modelo.
<b>Precondiciones</b>	El archivo <i>.pickle</i> con el modelo que se quiere cargar debe existir.
<b>Acciones</b>	Se carga el modelo desde un archivo <i>.pickle</i> .
<b>Postcondiciones</b>	Se obtiene el modelo.
<b>Excepciones</b>	No hay ningún archivo <i>.pickle</i> con el modelo deseado.

Tabla B.8: CU-8 Cargar modelos

CU-09	Gestión de los resultados
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-3 & RF-3.1 & RF-3.2 & RF-3.3
<b>Descripción</b>	Gestión de los resultados.
<b>Precondiciones</b>	El modelo tiene que estar creado. Los resultados tienen que haber sido calculados.

continúa en la página siguiente

continúa desde la página anterior

CU-09	Gestión de los resultados
<b>Acciones</b>	Tienen que existir los resultados. Las predicciones tiene que haber sido calculadas. Guardar los resultados. Mostrar los resultados.
<b>Postcondiciones</b>	Mostrar las predicciones. Se cargan los resultados. Se muestran los resultados.
<b>Excepciones</b>	Se muestran las predicciones. Los resultados no se han calculado. Las predicciones no se han calculado. No hay ningún modelo.

Tabla B.9: CU-9 Gestión de los resultados

CU-10	Guardar resultados
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-3 & RF-3.1
<b>Descripción</b>	Guardar resultados.
<b>Precondiciones</b>	Los resultados tienen que haber sido calculados. El modelo tiene que estar creado.
<b>Acciones</b>	Guardar los resultados.
<b>Postcondiciones</b>	Se guardan los resultados.
<b>Excepciones</b>	Los resultados no se han calculado. No hay ningún modelo.

Tabla B.10: CU-10 Guardar resultados

CU-11	Mostrar resultados
-------	--------------------

continúa en la página siguiente

continúa desde la página anterior

CU-11	Mostrar resultados
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-3 & RF-3.2
<b>Descripción</b>	Mostrar los resultados.
<b>Precondiciones</b>	Los resultados tienen que haber sido calculados. El modelo tiene que estar creado.
<b>Acciones</b>	Mostrar los resultados.
<b>Postcondiciones</b>	Se muestran los resultados.
<b>Excepciones</b>	Los resultados no se han calculado. No hay ningún modelo.

Tabla B.11: CU-11 Mostrar resultados

CU-12	Mostrar predicciones
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-3 & RF-3.3
<b>Descripción</b>	Mostrar las predicciones.
<b>Precondiciones</b>	El modelo tiene que estar creado. Las predicciones tiene que haber sido calculadas.
<b>Acciones</b>	Mostrar las predicciones.
<b>Postcondiciones</b>	Se muestran las predicciones.
<b>Excepciones</b>	Las predicciones no se han calculado. No hay ningún modelo.

Tabla B.12: CU-12 Mostrar predicciones

CU-13	Gestión de los usuarios
-------	-------------------------

continúa en la página siguiente

continúa desde la página anterior

CU-13		Gestión de los usuarios
<b>Versión</b>		1.0
<b>Autor</b>		Raúl Negro Carpintero
<b>Requisitos asociados</b>	asocia-	RF-4 & RF-4.1
<b>Descripción</b>		Gestión de los usuarios.
<b>Precondiciones</b>		Deben existir las matrices de datos. Debe existir el usuario con el que se quieren añadir valoraciones.
<b>Acciones</b>		Añadir valoraciones.
<b>Postcondiciones</b>		Se añaden valoraciones y/o usuarios a un conjunto de datos.
<b>Excepciones</b>		No existen las matrices de datos. No existe el usuario al que se le quiere añadir valoraciones.

Tabla B.13: CU-13 Gestión de los usuarios

CU-14		Añadir valoraciones
<b>Versión</b>		1.0
<b>Autor</b>		Raúl Negro Carpintero
<b>Requisitos asociados</b>	asocia-	RF-4 & RF-4.1
<b>Descripción</b>		Añadir valoraciones de un usuario existente o nuevo.
<b>Precondiciones</b>		Deben existir las matrices de datos. Debe existir el usuario con el que se quieren añadir valoraciones.
<b>Acciones</b>		Añadir valoraciones sobre un usuario nuevo o uno ya existente.
<b>Postcondiciones</b>		Se añaden valoraciones y/o usuarios a un conjunto de datos.

continúa en la página siguiente

continúa desde la página anterior

CU-14	Añadir valoraciones
<b>Excepciones</b>	No existen las matrices de datos. No existe el usuario al que se le quiere añadir valoraciones.

Tabla B.14: CU-14 Añadir valoraciones

CU-15	Gestión de los conjuntos de datos
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-5 & RF-5.1 & RF-5.2 & RF-5.3
<b>Descripción</b>	Gestión de los conjuntos de datos.
<b>Precondiciones</b>	Tienen que existir los conjuntos de datos de prueba. Tiene que existir un conjunto de datos para usar.
<b>Acciones</b>	Listar los conjuntos de datos de prueba. Seleccionar un conjunto de datos de prueba. Añadir conjunto de datos nuevo.
<b>Postcondiciones</b>	Se carga el conjunto de datos de prueba seleccionado. Se añade el conjunto de datos nuevo.
<b>Excepciones</b>	No existen conjuntos de datos de prueba. No existe ningún conjunto de datos que añadir.

Tabla B.15: CU-15 Gestión de los conjuntos de datos

CU-16	Mostrar los conjuntos de datos
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-5 & RF-5.1

continúa en la página siguiente

continúa desde la página anterior

CU-16	Mostrar los conjuntos de datos
<b>Descripción</b>	Listar los conjuntos de datos de prueba.
<b>Precondiciones</b>	Tienen que existir los conjuntos de datos de prueba.
<b>Acciones</b>	Listar los conjuntos de datos de prueba.
<b>Postcondiciones</b>	Se selecciona un conjunto de datos de prueba.
<b>Excepciones</b>	No existen conjuntos de datos de prueba.

Tabla B.16: CU-16 Mostrar los conjuntos de datos

CU-17	Seleccionar el conjunto de datos
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-5 & RF-5.2
<b>Descripción</b>	Seleccionar un conjunto de datos de prueba.
<b>Precondiciones</b>	Tienen que existir los conjuntos de datos de prueba.
<b>Acciones</b>	Seleccionar un conjunto de datos de prueba.
<b>Postcondiciones</b>	Se carga el conjunto de datos de prueba seleccionado.
<b>Excepciones</b>	No existen conjuntos de datos de prueba. No hay ningún conjunto de datos de prueba seleccionado.

Tabla B.17: CU-17 Seleccionar el conjunto de datos

CU-18	Añadir conjunto de datos
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-5 & RF-5.3
<b>Descripción</b>	Añadir nuevo conjunto de datos.
<b>Precondiciones</b>	-

continúa en la página siguiente



continúa desde la página anterior

CU-18	Añadir conjunto de datos
<b>Acciones</b>	Añadir conjunto de datos nuevo.
<b>Postcondiciones</b>	Se añade el nuevo conjunto de datos.
<b>Excepciones</b>	No existe ningún conjunto de datos que añadir.

Tabla B.18: CU-18 Añadir conjunto de datos

CU-19	Ayuda de la aplicación
<b>Versión</b>	1.0
<b>Autor</b>	Raúl Negro Carpintero
<b>Requisitos asociados</b>	RF-6 & RF-6.1 & RF-6.2
<b>Descripción</b>	Ayuda de la aplicación.
<b>Precondiciones</b>	Tiene que haber información sobre los modelos. Tiene que haber información sobre los conjuntos de datos de prueba.
<b>Acciones</b>	Mostrar información sobre los modelos. Mostrar información sobre los conjuntos de datos de prueba.
<b>Postcondiciones</b>	Se muestra la información sobre los modelos. Se muestra la información sobre los conjuntos de datos de prueba.
<b>Excepciones</b>	No existe información sobre los modelos. No existe información sobre los conjuntos de datos de prueba.

Tabla B.19: CU-19 Ayuda de la aplicación

CU-20	Mostrar información sobre los modelos
<b>Versión</b>	1.0

continúa en la página siguiente

continúa desde la página anterior

CU-20		Mostrar información sobre los modelos
<b>Autor</b>		Raúl Negro Carpintero
<b>Requisitos asociados</b>		RF-6 & RF-6.1
<b>Descripción</b>		Mostrar información sobre los modelos.
<b>Precondiciones</b>		Tiene que haber información sobre los modelos.
<b>Acciones</b>		Mostrar información sobre los modelos.
<b>Postcondiciones</b>		Se muestra la información sobre los modelos.
<b>Excepciones</b>		No existe información sobre los modelos.

Tabla B.20: CU-20 Mostrar información sobre los modelos

CU-21		Mostrar información sobre los conjuntos de datos
<b>Versión</b>		1.0
<b>Autor</b>		Raúl Negro Carpintero
<b>Requisitos asociados</b>		RF-6 & RF-6.2
<b>Descripción</b>		Mostrar información sobre los conjuntos de datos de prueba.
<b>Precondiciones</b>		Tiene que haber información sobre los conjuntos de datos de prueba.
<b>Acciones</b>		Mostrar información sobre los conjuntos de datos de prueba.
<b>Postcondiciones</b>		Se muestra la información sobre los conjuntos de datos de prueba.
<b>Excepciones</b>		No existe información sobre los conjuntos de datos de prueba.

Tabla B.21: CU-21 Mostrar información sobre los conjuntos de datos

## 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 se han utilizado a lo largo del proyecto están en formato .csv. Lo más normal es que para cada conjunto de datos se tengan 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 se pasan a *DataFrames* de *pandas* [6].

## Datos con LightFM

Una vez obtenidos los *DataFrames* para cada *.csv*, es necesario convertirlos a *Dataset* de *LightFM* [3] para poder trabajar con ellos.

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

## Datos con Spotlight

A diferencia de *LightFM*, en *Spotlight* no se trabaja con matrices dispersas, si no con arrays de *NumPy* (aunque se ofrece la posibilidad de transformar los arrays en *matrices COO* y *matrices CSR*).

*Spotlight* utiliza su propia clase, *Interactions* [4], para convertir los *DataFrames* y así poder utilizar los datos.

Otro aspecto a tener en cuenta en *Spotlight* son las secuencias, utilizadas en el modelo de secuencia implícito [?]. Las recomendaciones pueden verse como secuencias; dados los ítems con los que ha interactuado un usuario, ¿cuáles serán los próximos ítems con los que interactuará? De esta manera, una vez obtenidas las interacciones "normales", hay que pasarlas a interacciones de secuencia con el método `to_sequence()` [?].

## Persistencia

Para la realización del proyecto se necesita guardar de alguna manera tanto los datos intermedios (matrices de interacción) como los propios modelos y los resultados obtenidos al evaluarlos.

Esto se consigue gracias a *pickle* [?].

## 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 los clases de Sistema, que crean los sistemas de recomendación gracias a los datos proporcionados por las clases de Entrada.

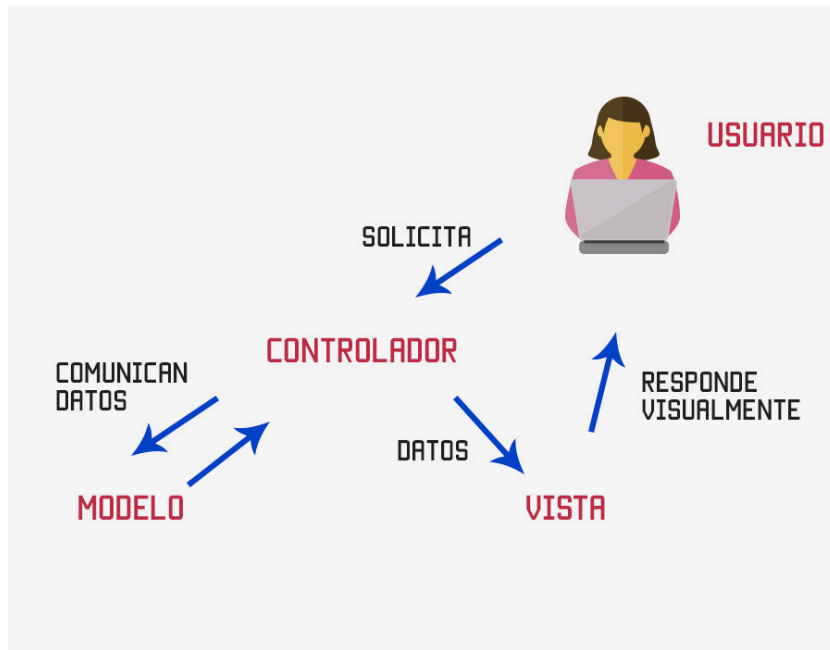


Figura C.1: Esquema del patrón MVC [2]

La estructura del proyecto siguiendo este patrón quedaría de la siguiente forma:

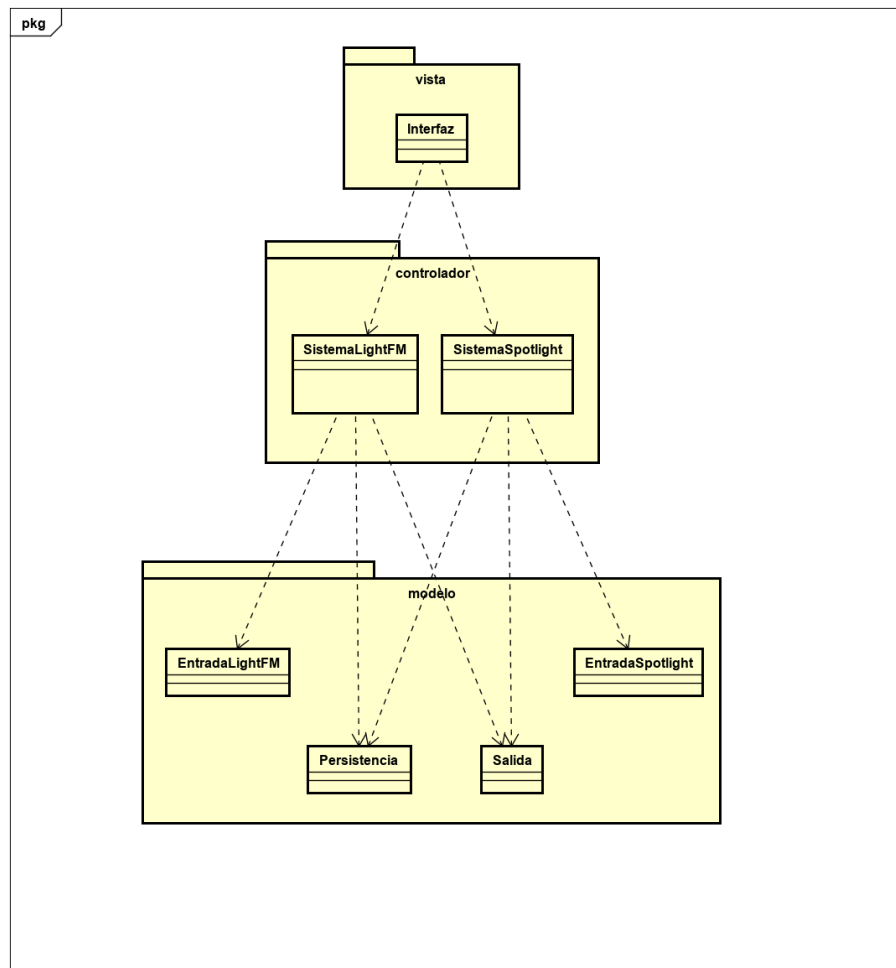
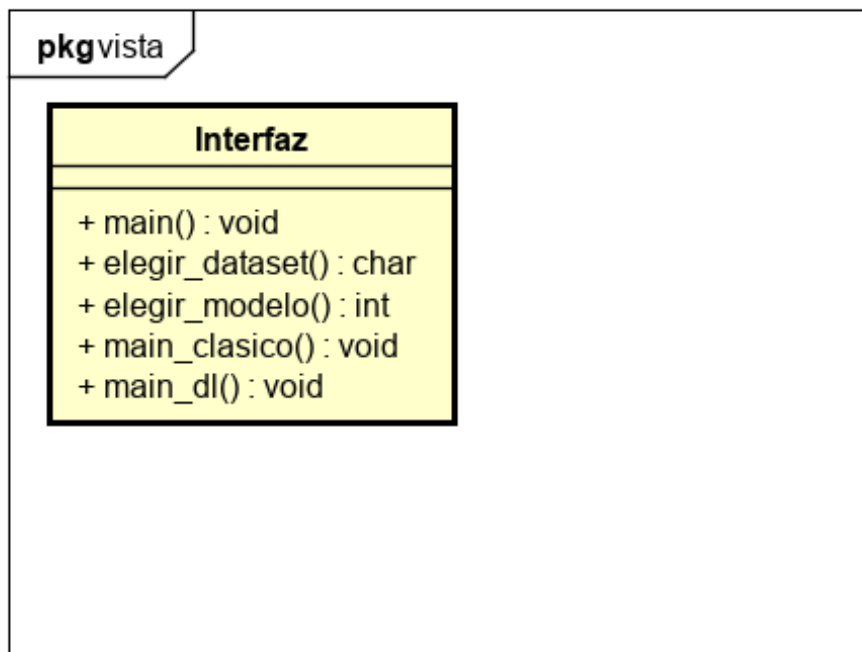
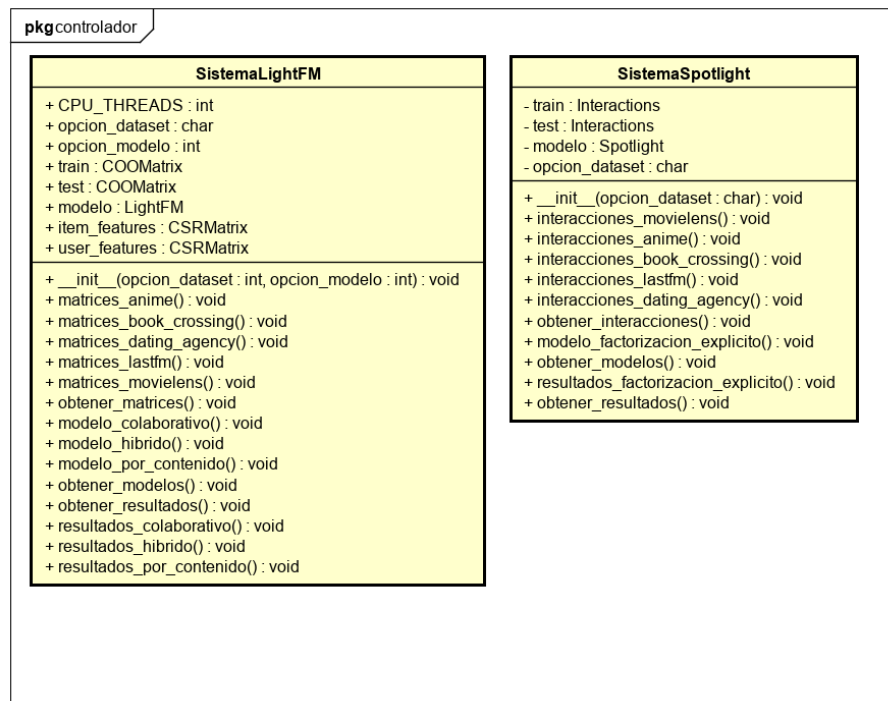


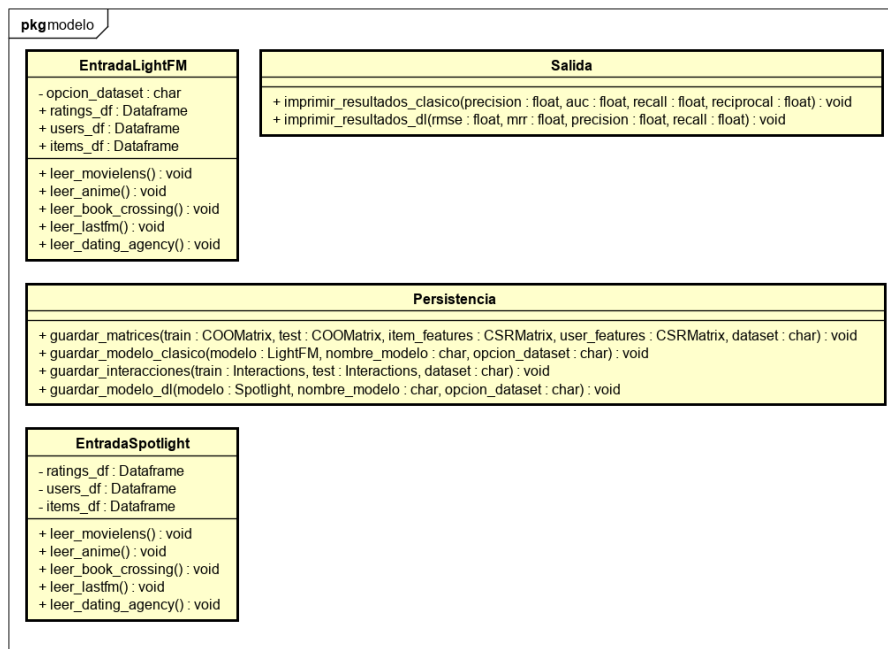
Figura C.2: Diagrama UML del proyecto

Por separado, los paquetes contienen:

Figura C.3: Diagrama UML del paquete *vista*

Figura C.4: Diagrama UML del paquete *controlador*



Figura C.5: Diagrama UML del paquete *modelo*

## Diseño con LightFM

Para la parte del modelo clásico, se tienen los siguientes archivos: En la clase `EntradaLightFM` se tiene:

- `leer_x`: estos métodos recogen los datos de los *.csv* para cada conjunto de datos. Estos métodos son utilizados por los métodos de obtención de matrices de la clase `SistemaLightFM`.

En el archivo `Salida` se tiene:

- `imprimir_resultados_clasico`: este método imprime las métricas del modelo clásico escogido.

En el archivo `Persistencia` se tiene:

- `guardar_matrices`: este método guarda en un archivo *pickle* las matrices que *LightFM* necesita para obtener los modelos.

- `guardar_modelo`: este método guarda el modelo obtenido por *LightFM* en un archivo *pickle*.

En el archivo `Interfaz` se tiene:

- `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_clasico`: programa principal si el modelo escogido es *LightFM*.

En la clase `SistemaLightFM` tenemos:

- `matrices_x`: estos métodos crean las matrices necesarias para cada conjunto de datos.
- `modelo_x`: estos métodos crean los distintos modelos de recomendación.
- `resultados_x`: estos métodos obtienen los resultados para cada modelo de recomendación.

## Diseño con Spotlight

Para la parte del modelo basado en aprendizaje profundo, se tienen los siguientes archivos: En la clase `EntradaSpotlight` se tiene:

- `leer_x`: estos métodos recogen los datos de los *.csv* para cada conjunto de datos. Estos métodos son utilizados por los métodos de obtención de interacciones de la clase `SistemaSpotlight`.

En el archivo `Salida` se tiene:

- `imprimir_resultados_dl`: este método imprime las métricas del modelo basado en aprendizaje profundo escogido.

En el archivo `Persistencia` se tiene:

- `guardar_interacciones`: este método guarda en un archivo *pickle* las interacciones que *Spotlight* necesita para obtener los modelos.
- `guardar_modelo_dl`: este método guarda el modelo obtenido por *Spotlight*.

En el archivo `Interfaz` se tiene:

- `main_dl`: programa principal si el modelo escogido es *Spotlight*.

En la clase `SistemaSpotlight` tenemos:

- `interacciones_x`: estos métodos crean las interacciones necesarias para cada conjunto de datos.
- `modelo_x`: estos métodos crean los distintos modelos de recomendación.
- `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

---

- [1] fast ai. *Practical Deep Learning for Coders, v3*.
- [2] Uriel Hernandez. Mvc (model, view, controller) explicado.
- [3] Maciej Kula. *Dataset construction*.
- [4] Maciej Kula. *Interactions*.
- [5] Maciej Kula. *Interactions - to<sub>s</sub>equen<sub>c</sub>e()*.
- [6] Maciej Kula. *Sequential models*.
- [7] Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman. *Mining of Massive Datasets*, chapter 9, page 513. Universidad de Standford, 2010.
- [8] Pandas. *DataFrame*.
- [9] Python Software Foundation. *pickle - Python object serialization*.