

# Euskoflix - Sprint 1

Ingeniería del Software

Cañadillas Patricio, Daniel

González Llaguno, Urtzi

Lahuerta Vázquez, Xabier

Pérez García, Iván

22 de marzo de 2019

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Planificación</b>	<b>2</b>
2.1. Historias de usuario . . . . .	2
2.2. Sprints . . . . .	2
2.3. Herramientas . . . . .	3
<b>3. Diseño</b>	<b>5</b>
3.1. Diagrama de clases . . . . .	5
3.2. Diagrama relacional . . . . .	8
<b>4. Desarrollo</b>	<b>9</b>
4.1. Sprint 1 . . . . .	9

# 1. Introducción

El proyecto Euskoflix forma parte de la asignatura Ingeniería del Software. El objetivo consiste en desarrollar un sistema de recomendación entorno a películas.

El proyecto pretende ser una aplicación de los conocimientos teóricos aprendidos en clase, tales como: la aplicación de patrones de diseño de software; los principios SOLID; la metodología de trabajo ágil Scrum y la importancia de una documentación adecuada con diagramas que expliquen el funcionamiento del sistema.

El desarrollo de la aplicación sigue el marco de desarrollo ágil **Scrum**. Primero se establecen las **historias de usuario** en base a los requisitos impuestos por el cliente (en este caso el cliente es la persona que imparte la asignatura). A continuación, se realiza una planificación temporal que se representa en **sprints**, los cuales consisten en entregar un prototipo del producto para mostrar al cliente el estado presente del desarrollo.

## 2. Planificación

Hemos optado por seguir la planificación propuesta en clase, donde se han establecido unas fechas para los sprints y nosotros tenemos que crear las historias de usuario.

### 2.1. Historias de usuario

Las historias de usuario son una representación sencilla de los requisitos impuestos por el cliente. Hemos definido 3 historias de usuario para el primer sprint.

1. **Iniciar el sistema:** Se crea el esqueleto vacío de las estructuras de datos.
2. **Carga de datos:** Se cargan los datos de usuarios y películas en el sistema.
3. **Obtener datos sobre una película:** Se conecta al servicio TMDb para obtener información adicional sobre una película concreta.

### 2.2. Sprints

Se han establecido 3 sprints. Comenzando el Sprint 1 el 27 de febrero y terminando el Sprint 3, es decir, el proyecto, el 8 de mayo.

- **Sprint 1 - Cargar datos:** Desde el 27 febrero hasta el 20 de marzo.
- **Sprint 2 - Valoración del producto:** Desde el 20 de marzo hasta el 10 de abril.
- **Sprint 3 - Elaboración de la recomendación:** Desde el 10 de abril hasta el 8 de mayo.

## 2.3. Herramientas

Se han utilizado las siguientes herramientas para apoyar el desarrollo del proyecto:

- **Git (concretamente GitLab)**: Para el control de versiones del código y para gestionar las tareas Scrum.
- **L<sup>A</sup>T<sub>E</sub>X**: Para la composición de documentos.
- **Maven**: Gestor de dependencias y librerías externas del código.
- **SQLite**: Sistema de gestión de bases de datos.
- **Visual Paradigm**: Creación de diagramas de clases, relacionales y de secuencia.

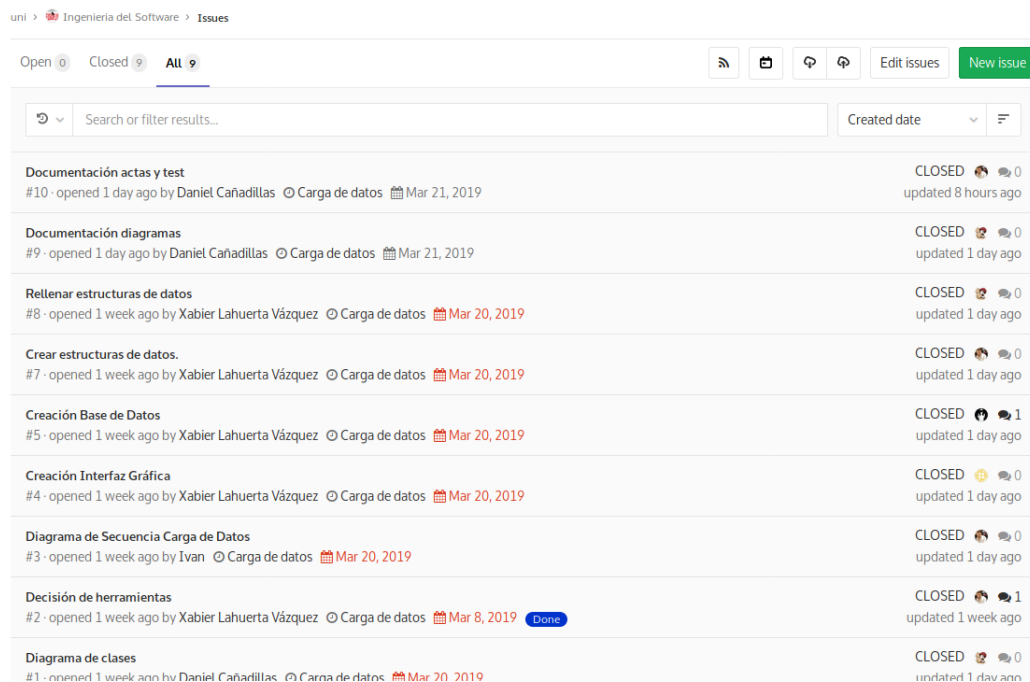


Figura 1: Gestión de Scrum con GitLab.

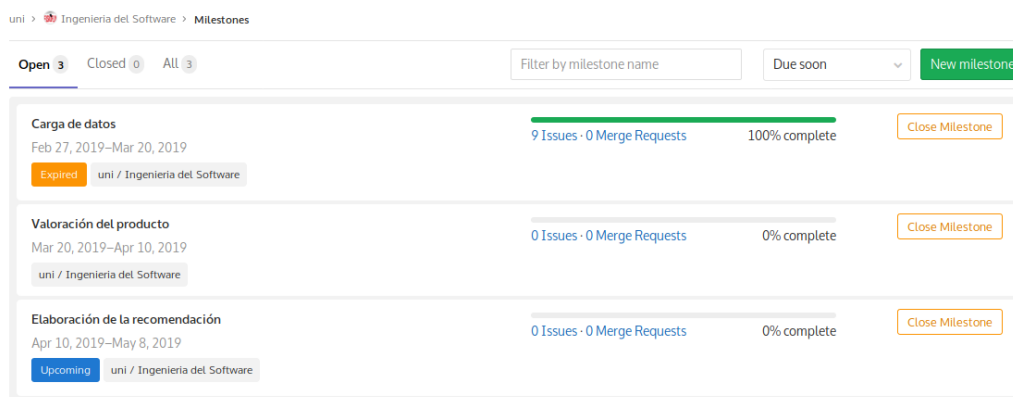


Figura 2: Gestión de sprints con GitLab.

## 3. Diseño

### 3.1. Diagrama de clases

A continuación, se muestra la estructura actual del diagrama de clases diseñado:

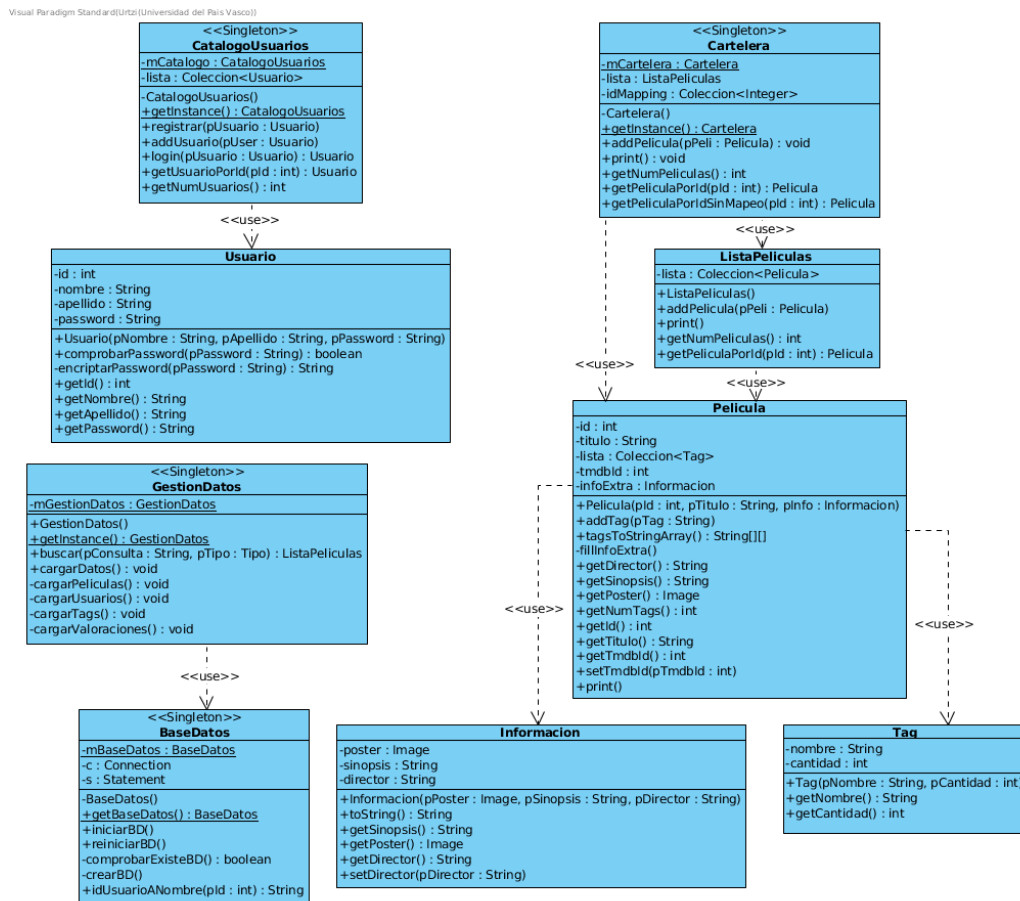


Figura 3: Diagrama de clases tras Sprint 1 (parte 1).

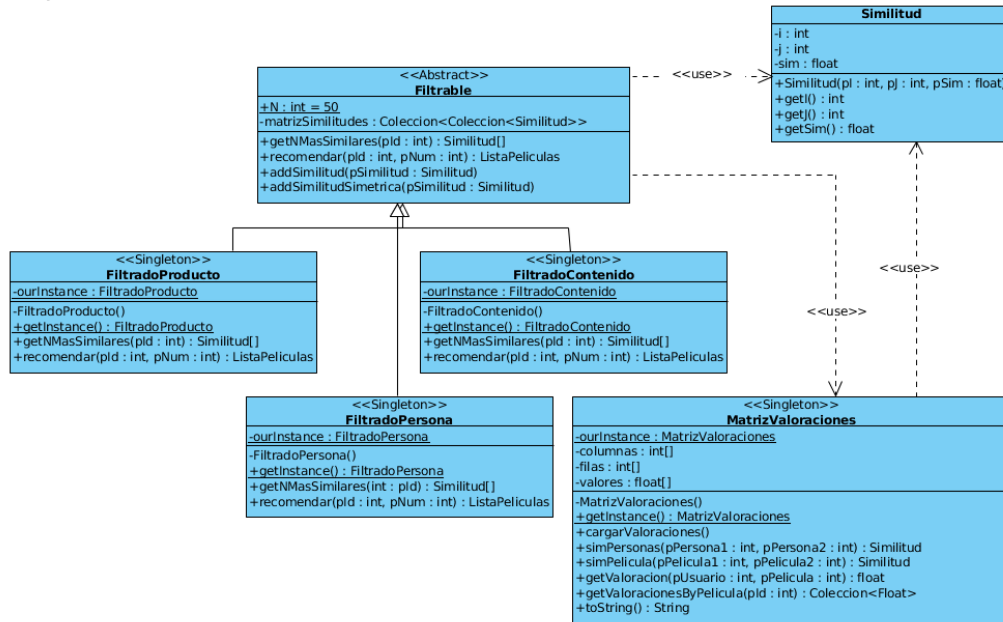


Figura 4: Diagrama de clases tras Sprint 1 (parte 2).

- **CatalogoUsuarios** : Singleton usado para almacenar y tratar la información de los usuarios.
  - `registrar(pUsuario)` : método para los nuevos usuarios que se registren.
  - `login(pUsuario)` : método para los usuarios registrados inicien sesión.
- **Usuario** : Almacena la información del usuario.
  - `comprobarPassword(pPassword)` : método para comprobar que el usuario ha escrito bien su contraseña.
  - `encriptarPassword(pPassword)` : método para encriptar nuestra contraseña y no sea fácil descubrirla.
- **Cartelera** : Singleton usado para almacenar y tratar la información de las películas.



- ListaPelículas : Clase encargada de tratar la información de las películas.
- Película : Almacena la información de la película.
  - fillInfoExtra() : método para conseguir la información extra de la película.
- Información : Almacena la información de la película obtenida de la API.
- Tag : Almacena el nombre de la etiqueta y la cantidad de veces que se ha mencionado dicha etiqueta.
- GestiónDatos : Fachada encarga de gestionar la obtención y manejo de datos desde la base de datos o la API.
  - cargarDatos() : método para cargar los datos de nuestra base datos.
  - cargarPelículas() : método para cargar las películas de nuestra base datos.
  - cargarUsuarios() : método para cargar los usuarios de nuestra base datos.
  - getValoraciones() : método par obtener todas las valoraciones.
  - getValoracionesUsuarios() : método para obtener las valoraciones de los usuarios.
  - getValoracionesPelículas() : método para obtener las valoraciones de las películas.
  - getInfoExtra() : método para los datos extra de la película a través de la API.
  - getTags(pPelícula) : método para obtener las apariciones de cada tag en la película.
- BaseDatos : Clase encargada de la creación y de las consultas a la bases de datos.
  - crearBD() : método para crear la base de datos.

- `anadirDatos()` : método para rellenar la base de datos creada con los datos de los CSV.
- `MatrizValoraciones` : Singleton encargado de gestionar la matriz de valoraciones
  - `getValoracionesByPelicula(pID)` : método para obtener las valoraciones de una película.
- `Similitud` : Clase que almacena la similitud y el índice que los componentes que son similares
- `Filtrable` : Clase abstracta que contiene el array de similitudes.
- `FiltradoProducto` : Clase encargada de filtrar en base del producto.
- `FiltradoContenido` : Clase encargada de filtrar en base del contenido.
- `FiltradoPersona` : Clase encargada de filtrar en base de los usuarios.

## 3.2. Diagrama relacional

El programa utiliza SQLite como sistema de gestión de base de datos. En la base de datos se almacena información sobre: usuarios, películas y géneros. También se almacena la información surgida de las relaciones entre las entidades, tales como: valoraciones, etiquetas y géneros de las películas. A continuación se muestra el diagrama relacional de la base de datos:

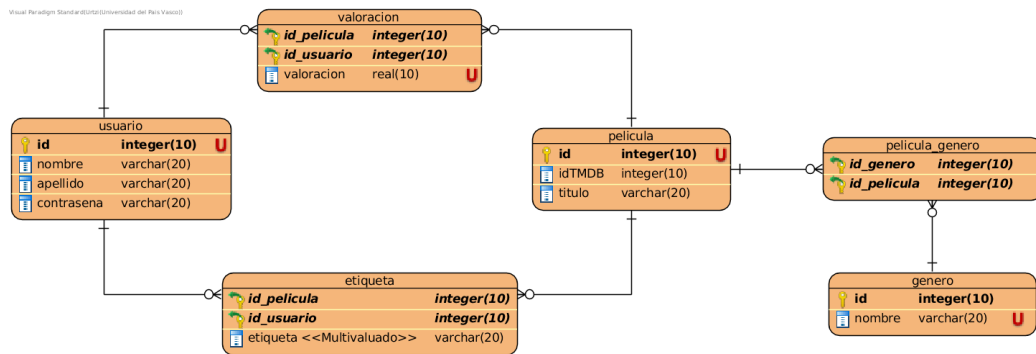


Figura 5: Diagrama relacional de la base de datos.

## 4. Desarrollo

### 4.1. Sprint 1

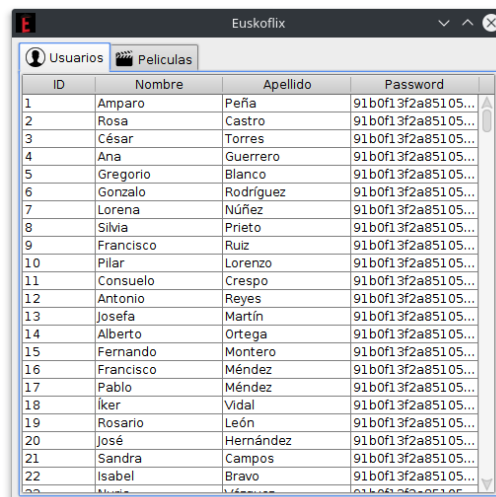
En este Sprint se ha realizado la carga de los datos necesarios para que el sistema pueda hacer recomendaciones.

Se han implementado las estructuras adecuadas para la recoger y utilizar la información sobre las películas y usuarios. También se ha desarrollado una interfaz gráfica que permite visualizar el contenido de dichas estructuras para verificar si la carga ha sido correcta.

Adicionalmente, se ha implementado la API de TMDb (The Movie Database) para obtener el póster y la sinopsis de la película.



Figura 6: Ventana de carga del sistema.

The image is a screenshot of a web application window titled 'Euskoflix'. It has two tabs: 'Usuarios' (selected) and 'Películas'. Below the tabs is a table with four columns: 'ID', 'Nombre', 'Apellido', and 'Password'. The table contains 22 rows of user data.

ID	Nombre	Apellido	Password
1	Amparo	Pefia	91b0f13f2a85105...
2	Rosa	Castro	91b0f13f2a85105...
3	César	Torres	91b0f13f2a85105...
4	Ana	Guerrero	91b0f13f2a85105...
5	Gregorio	Blanco	91b0f13f2a85105...
6	Gonzalo	Rodríguez	91b0f13f2a85105...
7	Lorena	Núñez	91b0f13f2a85105...
8	Silvia	Prieto	91b0f13f2a85105...
9	Francisco	Ruiz	91b0f13f2a85105...
10	Pilar	Lorenzo	91b0f13f2a85105...
11	Consuelo	Crespo	91b0f13f2a85105...
12	Antonio	Reyes	91b0f13f2a85105...
13	Josefa	Martín	91b0f13f2a85105...
14	Alberto	Ortega	91b0f13f2a85105...
15	Fernando	Montero	91b0f13f2a85105...
16	Francisco	Méndez	91b0f13f2a85105...
17	Pablo	Méndez	91b0f13f2a85105...
18	Iker	Vidal	91b0f13f2a85105...
19	Rosario	León	91b0f13f2a85105...
20	José	Hernández	91b0f13f2a85105...
21	Sandra	Campos	91b0f13f2a85105...
22	Isabel	Bravo	91b0f13f2a85105...

Figura 7: Ventana que contiene la lista de los usuarios del sistema.

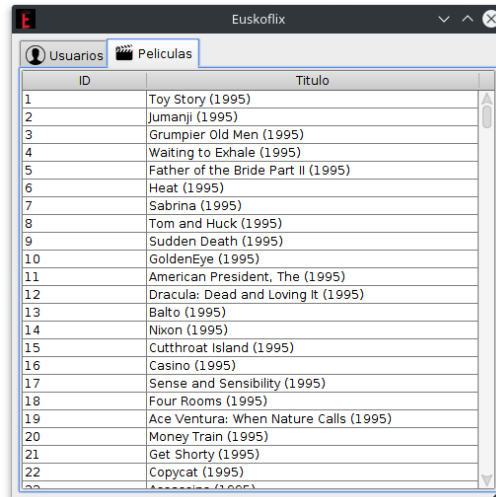


Figura 8: Ventana que contiene la lista de las películas del sistema.

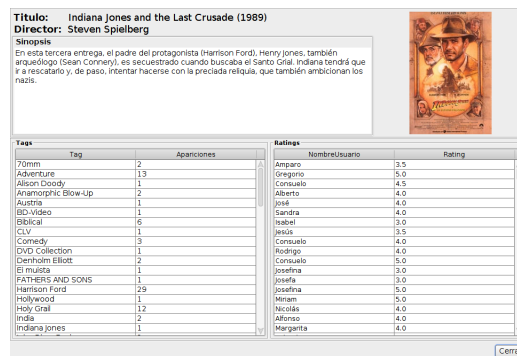


Figura 9: Ventana mostrando información sobre una película.

**Conclusiones del primer sprint:** Por el momento no se han presentado dificultades. El diseño inicial se ha hecho pensando de cara a futuras expansiones y la implementación ha ido acorde al diseño.