UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL ROSARIO

Cátedra: Soporte a la gestión de datos con programación visual.

4º Año Ingeniería en Sistemas de Información.

TPI – Red social de cine.

Año 2021

Alumnos:

- Bermejo Zambrini, Gonzalo Martín.
- Miguel Cabrera.

Docentes:

- Mario Castagnino.
- Juan Ignacio Torres.

Índice

Descripción del proyecto

El proyecto consta de una aplicación web orientada a una red social de cine, en la cual, el usuario podrá crear listas personalizadas y agregar a estas películas y series según quiera, además, podrá realizar comentarios para compartir su opinión acerca de dichos entretenimientos. El usuario también podrá visualizar el detalle de las películas, series y personas, que consiste en portada, descripción, créditos, trailers y demás.

Se visualizarán distintas listas como populares, próximos estrenos, ranking, etc.

La aplicación será desarrollada en Python y los datos de las películas/series/personas serán obtenidos por medio de la API de TMDB a través de una clave que nos proporcionó dicha web.

Requerimientos

Funcionales

RF	Descripción
01	Registrar nuevo usuario.
02	Autenticar usuario.
03	Cambiar contraseña de usuario.
04	Mostrar listado de películas/series/personas populares.
05	Mostrar listado de películas/series más valoradas.
06	Mostrar listado de próximos estrenos de películas.
07	Mostrar detalle de películas/series/personas.
08	Mostrar tendencias de películas.
09	Mostrar películas filtradas por género.
10	Registrar nuevo comentario para una película/serie.
11	Crear listas personalizadas.
12	Agregar películas/series a lista personalizada.
13	Visualizar recomendaciones según película/serie.

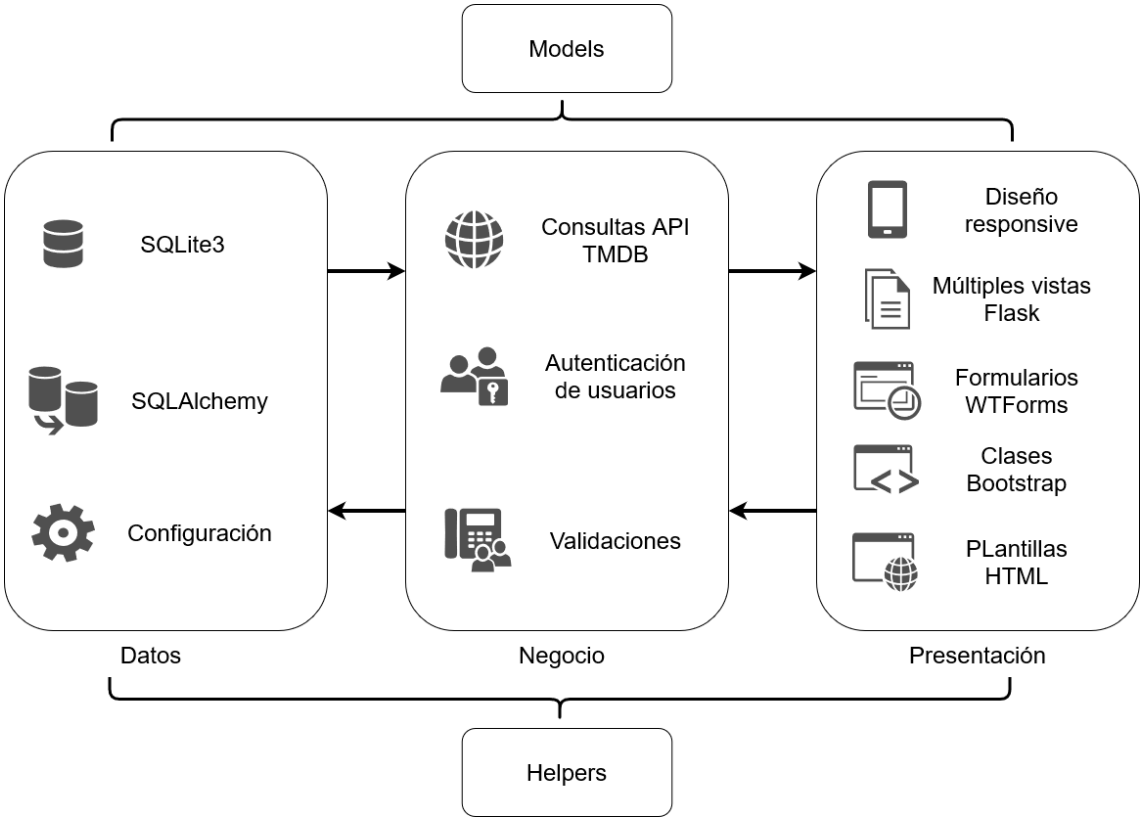
No Funcionales

RNF	Descripción
01	El sistema debe funcionar correctamente en múltiples navegadores
02	Todas las contraseñas deben guardarse con encriptado criptográfico
03	Todas los Tokens / API Keys o similares no deben exponerse de manera pública
04	El sistema debe diseñarse con la arquitectura en 3 capas
05	El sistema debe utilizar control de versiones mediante GIT.
06	El sistema debe estar programado en Python 3.8 o superior.
07	El sistema debe ser multiusuario.
08	El sistema debe utilizar una base de datos SQL o NoSQL

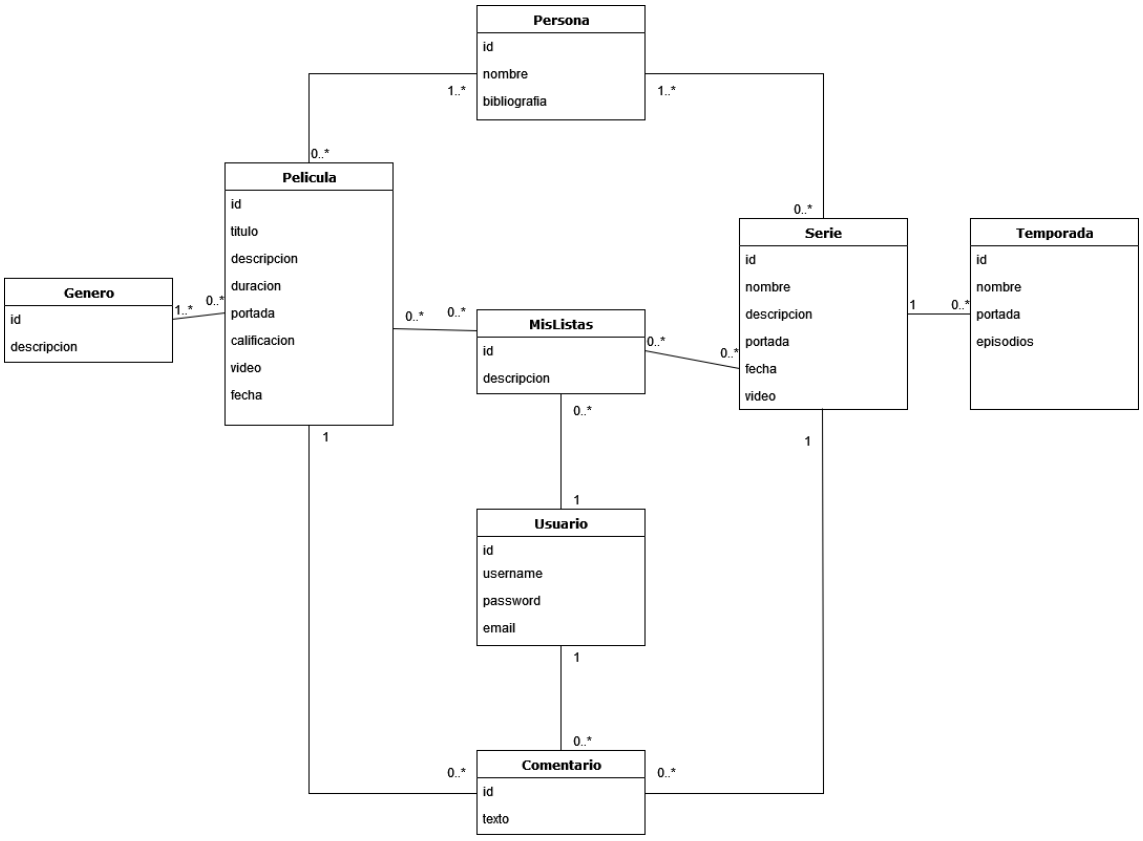
Reglas de Negocio

RN	Descripción
01	La longitud de las contraseñas debe ser entre 8 y 16 caracteres y solo contienen letras y números.
02	El nombre de usuario debe ser único.
03	El email debe ser único.
04	El nombre de usuario debe contener entre 6 y 30 caracteres y no puede contener símbolos.
05	Los usuarios no autenticados no pueden realizar comentarios.
06	Los usuarios no autenticados no pueden crear listas personalizadas.
07	Los usuarios no autenticados no pueden agregar películas/series a las listas.
08	La información de las películas/series/personas se obtienen del api de TMDB.

Bosquejo de Arquitectura



Modelo de Dominio



Casos de uso

Se describen los casos de uso más importantes.

Código y nombre del caso de uso: CUR01 **-** Registrar usuario					
Nivel	**Estructura**	**Alcance**	**Caja**	**Instanciación**	**Interacción**
Resumen	Sin-estructurar	Sistema	Negra	Real	Semántico
Meta del caso de uso: Registrarse como usuario					
Actores					
Primario: Usuario			**Otros:**		
Precondiciones					
De negocio: **De Sistema:**					
Disparador: Usuario selecciona la opción de registrarse					
Flujo de sucesos					
Camino Básico	1. Sistema muestra formulario de registro. 2. Usuario completa el formulario y lo envía. Sistema registra datos. 3. Sistema valida que el usuario no exista. 4. Sistema verifica que los datos del formulario sean válidos. 5. Sistema crea el nuevo usuario e informa la situación.				

Camino Alternativo	**3.a. <durante> Usuario ya existe:** 1. Sistema informa la situación. Vuelve al paso 2. **4.a. <durante> Caracteres inválidos en algún campo del formulario:** 1. Sistema informa la situación. Vuelve al paso 2. **4.b. <durante> Longitud inválida de nombre de usuario:** 1. Sistema informa la situación. Vuelve al paso 2. **4.c. <durante> Longitud inválida de contraseña:** 1. Sistema informa la situación. Vuelve al paso 2. **4.d. <durante> Email no existe:** 1. Sistema informa la situación. Vuelve al paso 2.
Postcondiciones	
De negocio • **Éxito:** Usuario se registra en la web y recibe mensaje de confirmación. • **Fracaso:** Usuario no se registra en la web. • **Éxito alternativo:** - **De sistema** • **Éxito:** Nuevo usuario registrado. • **Fracaso:** No se registra nuevo usuario. • **Éxito alternativo:** -	

Código y nombre del caso de uso: CUR02 **–** Iniciar sesión					
Nivel	**Estructura**	**Alcance**	**Caja**	**Instanciación**	**Interacción**
Resumen	Sin-estructurar	Sistema	Negra	Real	Semántico
Meta del caso de uso: Iniciar sesión de usuario					
Actores					
Primario: Usuario		**Otros:**			
Precondiciones					
De negocio: **De Sistema:**					
Disparador: Usuario selecciona la opción de iniciar sesión.					
Flujo de sucesos					
Camino Básico	1. Sistema muestra formulario de inicio de sesión. 2. Usuario completa el formulario y lo envía. Sistema registra datos. 3. Sistema valida que el usuario exista. 4. Sistema autentica al usuario y permite el acceso.				
Camino Alternativo	**3.a. <durante> Usuario no existe:** 1. Sistema informa la situación. Vuelve al paso 2. 4.a. <durante> Contraseña no coincide: 1. Sistema informa la situación. Vuelve al paso 2.				

Postcondiciones	
De negocio • **Éxito:** Usuario inicia sesión en la web. • **Fracaso:** Usuario inexistente. • **Éxito alternativo:** - **De sistema** • **Éxito:** Sistema da acceso al usuario. • **Fracaso:** Sistema no encuentra que el usuario esté registrado. • **Éxito alternativo:** -	

Código y nombre del caso de uso: CUR03 **–** Buscar película/serie/persona					
Nivel	**Estructura**	**Alcance**	**Caja**	**Instanciación**	**Interacción**
Resumen	Sin-estructurar	Sistema	Negra	Real	Semántico
Meta del caso de uso: Mostrar información de película					
Actores					
Primario: Usuario		**Otros:**			
Precondiciones					
De negocio: Usuario visualiza la página principal. **De Sistema:** Sistema muestra página principal.					
Disparador: Usuario desea buscar una película.					
Flujo de sucesos					
Camino Básico	1. Sistema muestra formulario de búsqueda. 2. Usuario completa el formulario y lo envía. Sistema registra datos. 3. Sistema consulta a la API el listado de películas/series/personas. 4. Sistema muestra listado de películas, series y personas. 5. Usuario selecciona la opción deseada. Sistema muestra información de la película/serie/persona seleccionada.				
Camino Alternativo	**4.a. <anterior> No hay películas coincidentes con la búsqueda:** 1. Sistema informa la situación. 2. Termina el caso de uso.				
Postcondiciones					
De negocio • **Éxito:** Usuario visualiza información de la película/serie/persona. • **Fracaso:** No hay películas coincidentes. • **Éxito alternativo:** - **De sistema** • **Éxito:** Sistema muestra información de la película/serie/persona. • **Fracaso:** Sistema no encuentra resultados coincidentes. • **Éxito alternativo:** -					

Stack Tecnológico

Definir que tecnologías se van a utilizar en cada capa y una breve descripción sobre por qué se escogió esa tecnología.

Capa de Datos

- SQLite3
- SQLAlchemy

Se utilizó una base de datos sqlite3 y como ORM se utilizó sqlalchemy para manejar la base de datos debido a las facilidades que nos brinda esta herramienta y otro motivo fue para aprender nuevas tecnologías.

Capa de Negocio

Para obtener la información de las películas, consumimos la API de TMDB, la cual es una API pública. Para realizar las consultas se necesita una "api key", la cual solicitamos y utilizamos.

Capa de Presentación

- Flask.
- HTML5.
- BOOTSTRAP4.
- CSS3.

Optamos por la utilización de Flask para la capa de presentación debido a nuestra falta de experiencia con python necesitábamos un framework con una curva de aprendizaje baja. También, gracias a que Flask utiliza Jinja2 para el manejo de datos en las plantillas HTML, nos facilita mostrar la información en la web. Al ser inexpertos en el frontend, creemos

que lo mejor es utilizar clases bootstrap para la estética de la web y refinar cosas puntuales con nuestro escaso conocimiento de css.

Dependencias

Se requiere instalar estas librerías de Python para que el sistema funcione.

```
certifi==2021.10.8
charset-normalizer==2.0.7
click==8.0.3
colorama==0.4.4
Flask==2.0.2
Flask-Login==0.5.0
Flask-WTF==0.15.1
greenlet==1.1.2
gunicorn==20.1.0
idna==3.3
itsdangerous==2.0.1
Jinja2==3.0.2
MarkupSafe==2.0.1
requests==2.26.0
SQLAlchemy==1.4.25
urllib3==1.26.7
Werkzeug==2.0.2
WTForms==2.3.3
```

Repositorio

Para el control de versiones, utilizamos Git y GitHub. A continuación, se deja el link del repositorio.

Link: https://github.com/gmbz/frro-soporte-TPI-09