

Diseño y arquitectura de software

**Proyecto Final**

18 de junio del 2022

**Alejandro Cedillo Gamez**

**A00824742**

**Sergio Guasso Idalias**

**A00826042**

* Express in 3 to 5 simple sentences the main functional requirements of the system.
  + What are the main actors of the system?

La API con IMDB, los endpoints y los datos del usuario.

* + What can we do with the system?

Algoritmo de recomendaciones, no solamente de películas, ya que al tener los datos del usuario esto es replicable a otro tipo de bases de datos, por ejemplo música o libros.

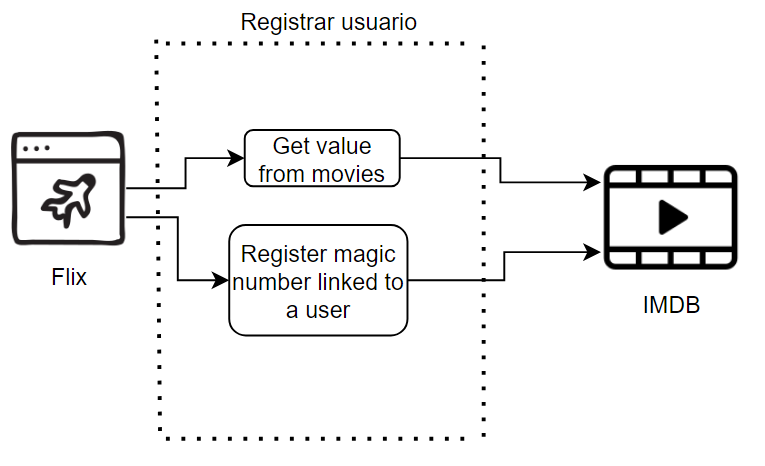
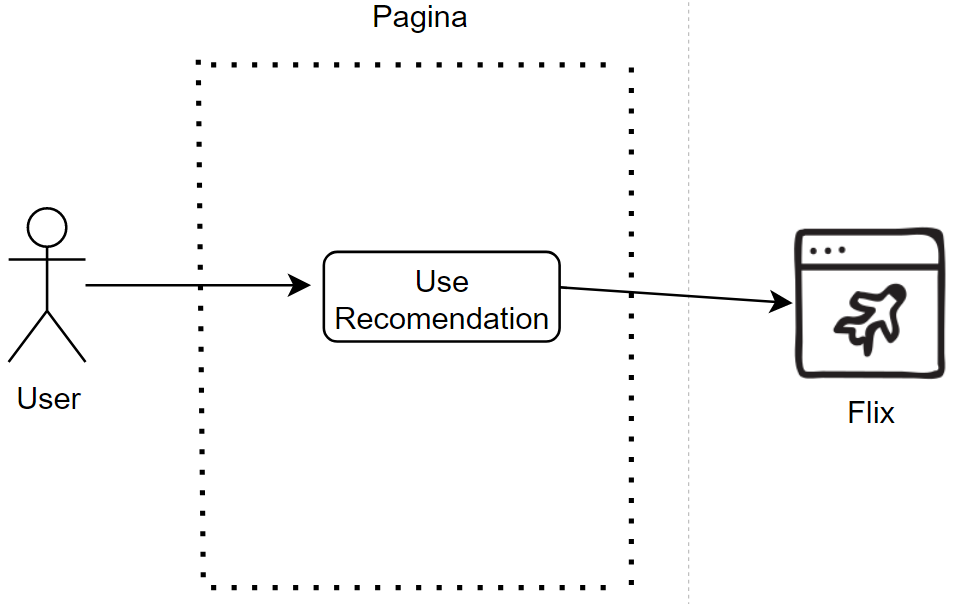
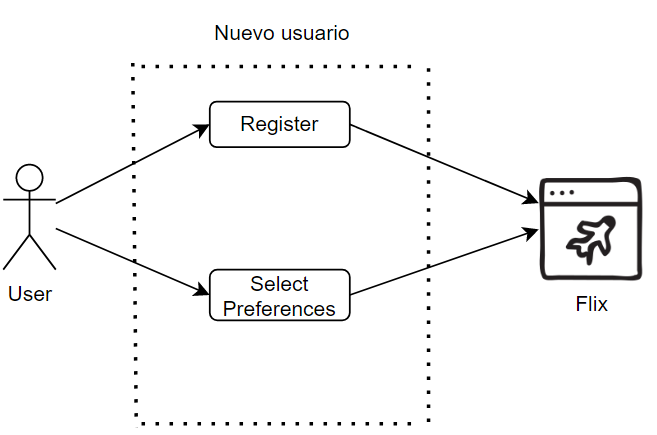
* + What would you ask the stakeholders to clarify your questions about functional requirements? (Write at least 3 questions)

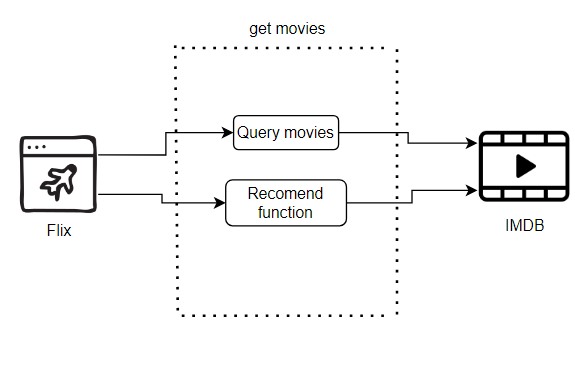
1. ¿El sistema debe registrar si el usuario ya vio alguna película?
2. ¿El usuario puede cambiar sus preferencias? ¿Puede elegir una categoría que no desea le sea recomendado?
3. ¿Puede el usuario seleccionar preferencias, ejemplo, películas favoritas, directores o actores?

* Create a Use Cases Diagram of the intended functionality.

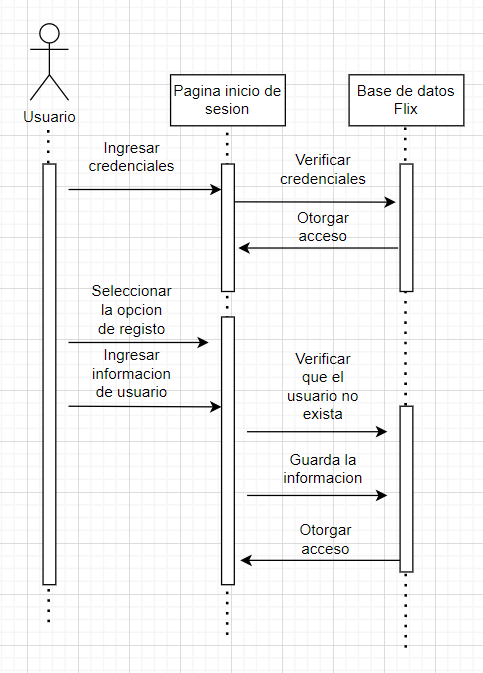
Link a draw.io de los diagramas:

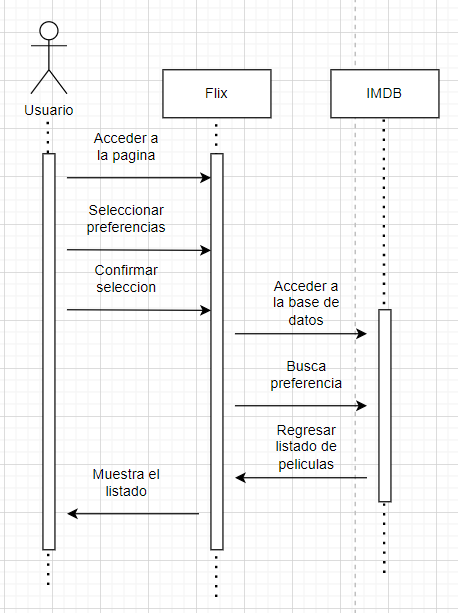
<https://drive.google.com/file/d/1_7Zw9ssdiPbzeU0S8TX4lb8xvnSHL6Hm/view?usp=sharing>





* Create a Sequence Diagram about a user registration and another for retrieval of movie recommendations for a user.



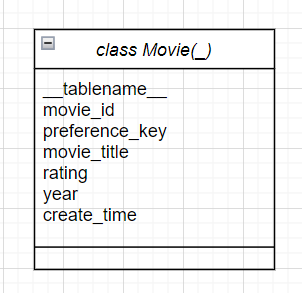


* Consider that our system needs to be highly available and in the future we need to include new movies and different parameters to update our algorithm
  + What non functional requirements should we take into consideration?

1. La organización de los elementos funcionales en la página,
2. Los programas y el lenguaje que se utilizara para la creación de conexiones con la base de datos IMDB
3. La paleta de colores.
4. Cantidad de películas a recomendar.
   * Functional requirements:
5. Capacidad de registrar credenciales de usuario en la base de datos
6. Casillas seleccionables para la selección de preferencias
7. Poder seleccionar el nombre de una película para poder obtener más información

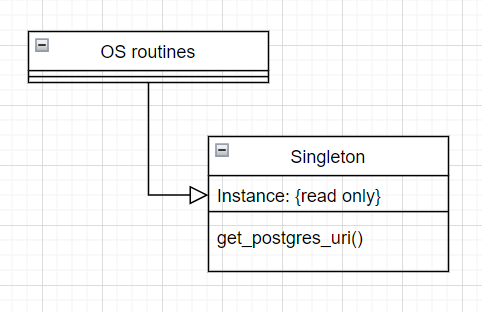
**Design patterns**

* Builder



La clase de Movie nos permite crear una tabla con atributos ya establecidos, lo que nos permite añadir de manera fácil nuevos valores en ella y poder referenciarlos. Para poder llamar a un atributo de la clase utilizamos su nombre en otras definiciones *models.Movie(...)*

* Singleton



El elemento singleton nos permite referenciar, por medio de la función *get\_postgres\_uri()* los distintos métodos que contiene el módulo OS con respecto a sus rutinas. Esta función nos permite declarar otros elementos a través de ella, ya que nos ofrece la misma información con menos pasos. Un ejemplo sería la declaración de *engine* y la de *session*.