



**Tecnológico
de Monterrey**

Diseño y arquitectura de software (Gpo 2)

Proyecto Final (primer avance)

Profesor: Mario Alberto González López

José Elías Garza Vázquez A00824494

Actores, objetivo y funcionalidad

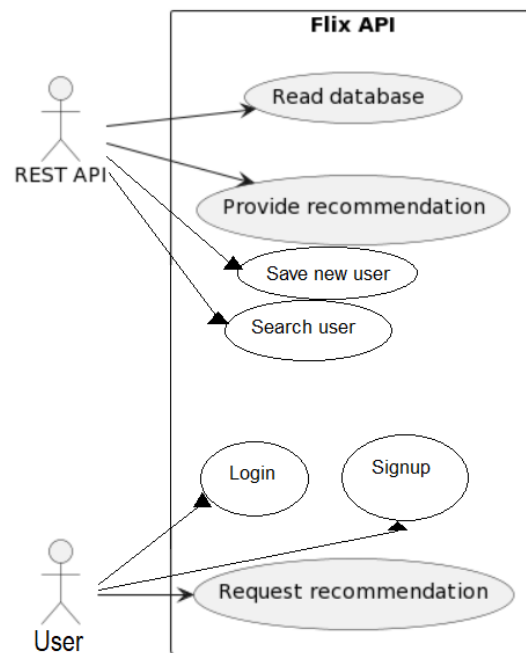
Considerando que el sistema actual tiene como objetivo generar endpoints para que otros servicios puedan consumir la información contenida en una base de datos, es válido considerar este sistema como una API. Teniendo en cuenta la información anterior, podemos considerar que tenemos 3 actores: el Servidor, el Servicio, y la API misma.

Con este sistema podemos proveer de recomendaciones de películas a servicios que lo requieran, en orden descendente o ascendente de rating dependiendo de sus necesidades. Las recomendaciones ofrecidas se pueden personalizar por medio de un algoritmo que genera identificadores para los usuarios.

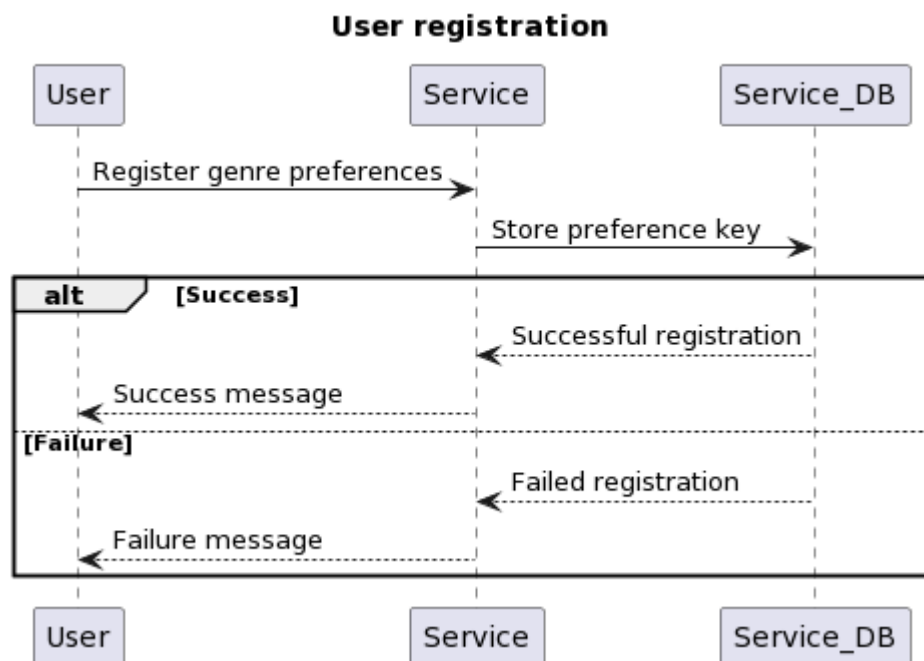
Algunas preguntas que se podrían hacer a los stakeholders para clarificar sus necesidades son las siguientes:

- [non-functional] ¿Cuántas requests esperan hacer por servicio a la API por hora?
- [non-functional] ¿Con cuántos servicios pretenden consumir la API concurrentemente?
- [functional] ¿El registro de llaves de preferencia también debe ser realizado por el API?
- [functional] ¿Cuál es el formato y estructura que mejor sirve de las respuestas a llamadas de la API?
- [functional] ¿Es necesaria alguna interfaz web gráfica?

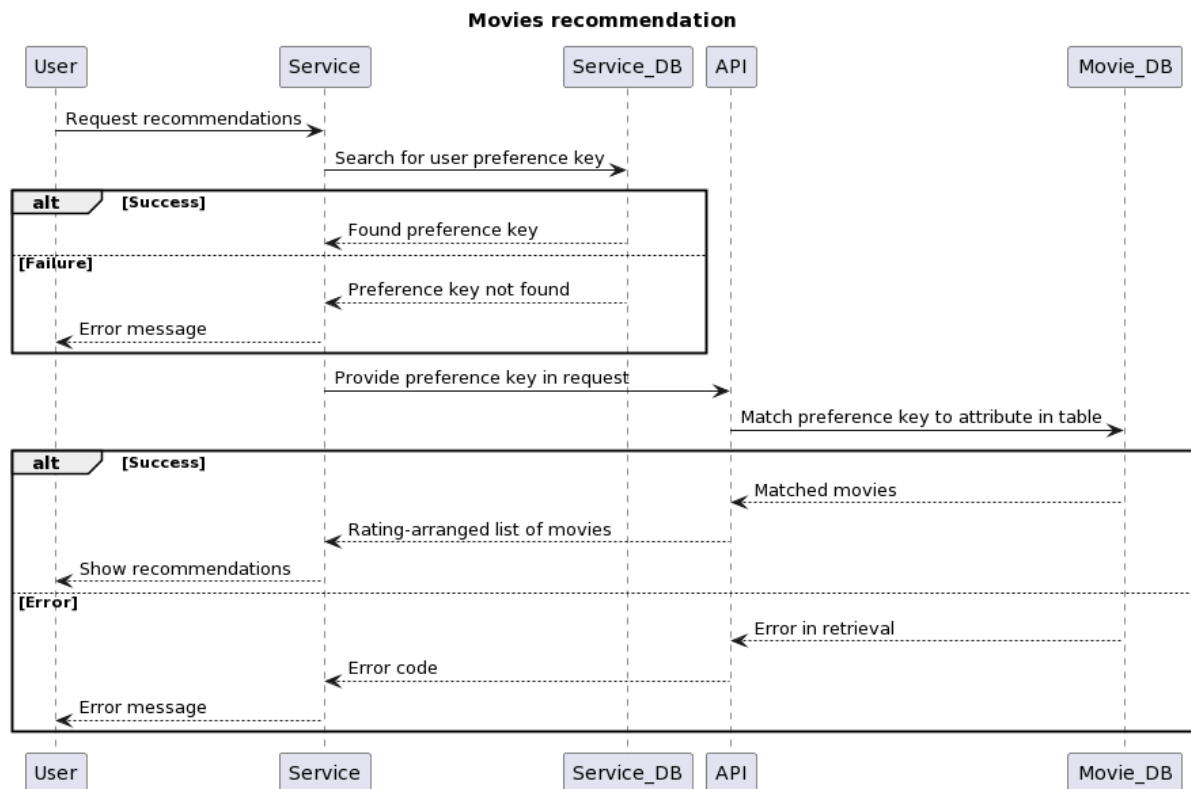
Use cases diagram



Sequence diagrams



Sequence diagrams



Requerimientos funcionales

- La API deberá hacer recomendaciones de películas
- Tener opción de en los requests hechos incluir la propiedad "rating", que si es falsa, debe mostrar las recomendaciones en orden descendente de rating, o en ascendente de lo contrario.
- Incluir funcionalidad de login
- Incluir funcionalidad de sign up
- Almacenar en la base de datos de usuarios la llave de preferencia del usuario en cuestión.

Requerimientos no funcionales

- Capacidad para una cantidad (a determinar) de requests por tiempo determinado.
- Capacidad para sostener una cantidad (a determinar) de llamadas concurrentes.
- Altamente modular.
- Acorde a los principios de SOLID.
- Uso de decoradores para futuro mantenimiento.

- Manejar casos de error a través de respuestas.
- Calcular la llave de preferencias del usuario con el algoritmo propuesto.