

Proyecto Integrador

1 - Diseñar el **Buscador, Tarjeta de Película y Tarjeta de Detalle**, que permite la búsqueda de películas, la presentación y el detalle de la misma, con puro css o alguna librería a elección.

2 - Trabajar con una estructura de la home, con al menos dos filas de tarjetas de películas. Se sugiere Implementar infinite scroll. Investigar qué librería usar.

3 - Crear el microservicio **order-service**.

4 - Crear el controlador **OrdenResource**. Crear la base de datos y la tabla **ORDENES** con los siguientes campos:

```
id: Long
fecha_creacion: LocalDate
socio_id: FK a la tabla SOCIOS
estado_orden_id: FK a la tabla ESTADOS_ORDENES
monto_total: double ( > 0)
cupon_id: fk a la tabla CUPONES
```

5- Generar las entidades necesarias y realizar los mapeos con JPA. Las entidades y/o DTO deberán estar en sus paquetes correspondientes.

6 - Realizar un segundo microservicio de socios, **socios-service**, para poder consultar la información desde **order-service**.

Crear la tabla de **SOCIOS** con los campos mínimos: (Agregar más datos si considera necesario)

```
id: Long
nombre: String
apellido: String
fecha_alta: LocalDate
activo: boolean
```

7 - Configurar *Spring Security* para **OrdenResource**. Excluir otros controllers de la seguridad. Crear el controller **AuthController**, quien recibirá user y password y retornará un jwt. Crear las clases necesarias para poder generar y parsear jwt.

8 - Implementar autorización a nivel de métodos de **OrdenResource** (*hasRole()*)

Crear las tablas y relaciones entre USUARIOS y ROLES. Configurar los roles de usuario en la base de datos.

9- Realizar pruebas con POSTMAN de todos los verbos HTTP expuestos en **order-service**. Generar collection.

NOTAS:

- a) Entregar el resultado final, con su correspondiente repositorio GIT.
- b) Se deberán entregar los correspondientes dml para la puesta inicial antes de realizar las pruebas. Los ddl no son necesarios puesto que el esquema es creado por jpa.
- c) Grabar un video para poder demostrar el trabajo realizado o entregar el resultado final desplegado en github pages u otro hosting a elección.