Práctica 1 – Videoclub online

Enunciado

Implementa una aplicación web cuya función es proporcionar contenidos multimedia a sus usuarios (**videoclub online**).

En la web habrá dos tipos de roles para usuarios:

- USER: Para usuarios consumidores de contenidos multimedia
- ADMIN: Para usuarios administradores

Los casos de uso a implementar están representados en el siguiente diagrama UML:

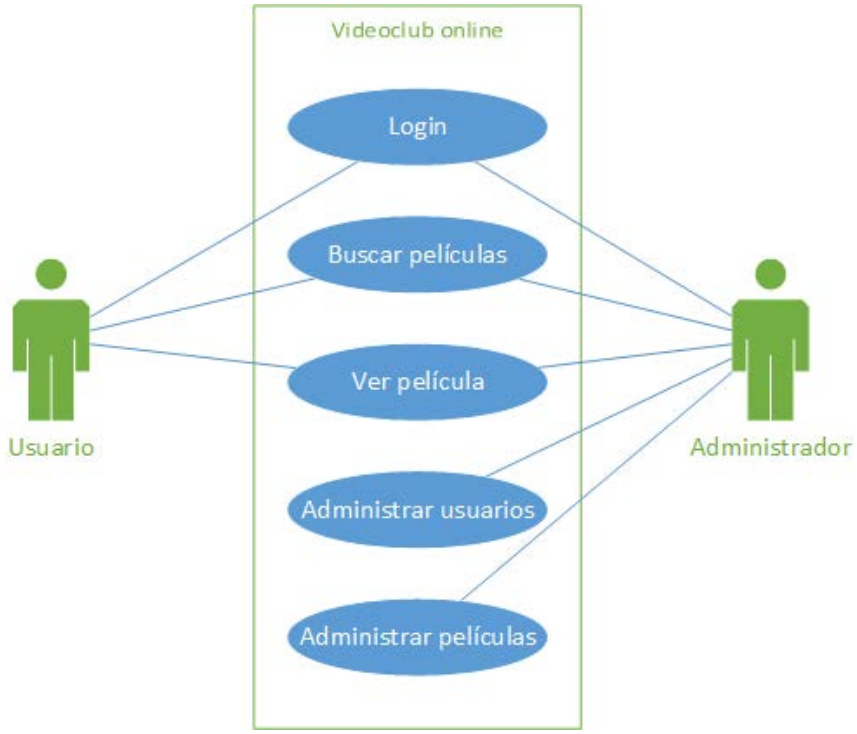

Figura 1 - Diagrama de casos de uso de la aplicación

Login. Todos los usuarios tendrán que estar autenticados en el sistema. Para ello se usará un formulario que pedirá las credenciales de acceso (nombre de usuario y contraseña). Estas credenciales serán verificadas por el servidor. Todos los usuarios del sistema serán almacenados de forma persistente en una base de datos en el lado servidor.

Buscar películas. Una vez autenticado, todos los usuarios podrán buscar películas. Las películas serán modeladas como entidades persistentes en la base de datos del sistema. La función de búsqueda se realizará al menos usando el campo de **nombre de película**.

Cada película deberá almacenar al menos la siguiente información:

- Nombre de la película
- URL del contenido de la película (vídeo)

- Descripción
- Año
- Director
- Reparto de actores
- URL de la portada
- Valoración

Ver película. Tras la búsqueda de películas se podrán visualizar las mismas. Para ello se usará el cambio "Contenido de la película" según se ha definido en la entidad película. No es necesario que la película enlace con su contenido real. Como alternativa, podemos usar contenidos de YouTube similares, como el tráiler de la película en cuestión. Por ejemplo, si hemos almacenando la película "Interstellar" en nuestro sistema, como contenido podemos almacenar la URL usada para embeber el vídeo correspondiente de YouTube:

https://www.youtube.com/watch?v=UoSSbmD9vqc

Figura 2 - Ejemplo de contenido de película

Este código URL será almacenado en la base de datos y posteriormente enviado al cliente, que directamente lo añadirá a la página HTML para que el navegador muestre el vídeo. Por ejemplo:

```
<iframe width="560" height="315"
        src="https://www.youtube-nocookie.com/embed/UoSSbmD9vqc"
        frameborder="0" allowfullscreen></iframe>
```

Administrar usuarios. Este caso de uso sólo podrá ser llevado a cabo por usuarios de tipo ADMIN. Ten en cuenta que los usuarios normales no tienen la capacidad de darse de alta en el sistema. Esta función (por simplicidad) es realizada por el usuario administrador.

El administrador (tipo ADMIN) deberá poder realizar operaciones CRUD para los usuarios (tipo USER) del sistema. Por lo tanto el administrador deberá poder dar de alta, de baja, modificar y eliminar usuarios. La información asociada a cada usuario se almacenará en la base de datos del sistema. Esta información deberá ser al menos:

- Nombre de usuario
- Contraseña (no se debe almacenar en claro en la base de datos)
- Correo electrónico

El usuario administrador debe existir en la base de datos. Puedes usar un medio de inserción de datos en base a un componente Spring en el arranque de la aplicación, tal y como hemos visto en clase.

Administrar películas. Este caso de uso sólo podrá ser llevado a cabo por usuarios de tipo ADMIN. Los administradores deberán poder realizar operaciones CRUD en las películas.

Para dar de alta una película, los admiradores rellenarán obligatoriamente los campos "**Nombre de la película**" y "**Contenido de la película**". La idea es que cuando se vaya a leer una película, si el resto de campos están vacíos se usará un servicio REST externo para obtener el resto de información, esto es:

- Descripción
- Año
- Director
- Género
- Reparto de actores
- URL de la portada
- Valoración

Hay varios servicios REST públicos para consultar información de películas. Por su simplicidad, te recomiendo que uses **OMDb** ("*The Open Movie Database*"): http://www.omdbapi.com/. La idea es crear un cliente REST para este servicio que desde el lado servidor de nuestra aplicación web consulta al servicio REST para obtener la información anterior (descripción, año, director, reparto, portada, valoración). Un ejemplo de petición a este servicio sería la siguiente URL: http://www.omdbapi.com/?t=Interstellar

Implementación

Los requisitos de implementación de esta práctica son los siguientes:

- Se usará Java en el lado servidor, concretamente Spring Boot y el resto tecnologías que hemos visto en clase
- Se usará Maven para el empaquetado de la aplicación
- En la parte cliente se deberá usar HTML y Bootstrap. La aplicación debe ser adaptable (*responsive*) y se debe visualizar correctamente en navegadores con una resolución de pantalla reducida. Si fuese necesario usar JavaScript se usará jQuery
- La presentación se hará en base a Spring MVC con Thymeleaf como tecnología de plantillas
- La aplicación deberá disponer obligatoriamente de una base de datos. Por motivos de simplicidad y portabilidad, se pide usar una base de datos en memoria tipo H2. Puedes hacer un componente Spring que realice una carga inicial de datos de prueba. El acceso a la base de datos deberá ser mediante Spring Data JPA
- El cliente REST debe estar implementado en Retrofit
- Para facilitar la implementación, en el Blackboard se puede encontrar la clase `Film.java`, que puede ser usada en dos situaciones:
    - Para almacenar el estado de las películas en la base de datos (clase *entidad* JPA)
    - Para recibir la respuesta del servicio REST en el cliente Retrofit