1 Introducción

Se desea desarrollar una *gramola* virtual. Una gramola, rockola o *jukebox* es una máquina que antiguamente estaba presente en muchos bares y que permitía al cliente poner una canción previo pago de cierta cantidad (Figura 1). Por ejemplo: un cliente llegaba al bar y se dirigía a la gramola; elegía una canción de las ofrecidas por el aparato, echaba una moneda de 25 pesetas y la canción empezaba a sonar.

Figura 1. Una gramola*, tomada de https://www.innovaspain.com/gramola-digital-santiago-de-chile/*

En nuestro caso, desarrollaremos una *gramola* virtual que ofreceremos como servicio a establecimientos (bares, por ejemplo) y que usará la API de algún proveedor de música en streaming. Describimos este enunciado con Spotify.

2 Escenario de uso

Un escenario de uso habitual será el siguiente:

1) El personal del bar abre el establecimiento, selecciona una *playlist* y esta empieza a sonar (la de la Figura 2, por ejemplo).

Figura 2. Una Playlist de Spotify

2) En otro dispositivo (un ordenador o una tablet, que será el que utilicen los clientes para elegir las canciones que desean reproducir), el personal del bar se conecta a nuestro sistema con un nombre y contraseña, con lo que aparece la interfaz de nuestra aplicación (la *gramola)*.
3) Llega un cliente al bar. A través de la aplicación *gramola* que vamos a desarrollar, busca una canción mediante la API de Spotify, paga cierto importe mediante un servicio de pagos y la canción se "cuela" en la *playlist*, de modo que empezará a sonar cuando termine la que se está reproduciendo actualmente.
   Por ejemplo, en la Figura 2 está sonando "Una noche sin ti". Supongamos que el cliente busca la canción "Whole lotta love", de Led Zeppelin y paga el importe. Cuando termine "Una noche sin ti", comenzará a sonar el tema seleccionado y, cuando este finalice, seguirá sonando la siguiente canción de la lista de reproducción ("Mucho mejor", como se ve en la Figura 2).
4) Es posible que varios clientes seleccionen varias canciones. Por ejemplo, después de haber seleccionado "Whole lotta love", y estando aún sonando "Una noche sin ti", otro cliente elige "Creep", de Radiohead.
   En este caso, terminará "Una noche sin ti", seguirá "Whole lotta love", continuará "Creep" y, luego, "Mucho mejor".

Para que desde el bar se pueda acceder a la aplicación *gramola*, el bar tiene que estar registrado en nuestro backend.

3 Requisitos

Los profesores, como jefes del proyecto, imponemos una serie de requisitos funcionales y no funcionales. A continuación describimos algunos.

3.1 Arquitectura

La Figura 3 muestra una vista arquitectónica a muy alto nivel del sistema completo, y que debe aplicarse obligatoriamente. Como se ve:

1) Hay un front-end con dos grupos de funcionalidades:
    a. *Gestión de cuentas de usuario*. Los usuarios, en este caso, son los bares, que pueden contratar nuestro servicio de la gramola. Se permitirá crear una cuenta, loguearse, recuperar la contraseña, etcétera.
    b. *Funcionalidades de gramola*: son las que utilizan los clientes del bar para buscar, añadir, pagar y reproducir canciones. Como se ve en la figura, este grupo de funcionalidades utiliza nuestro backend, el servicio de música y el servicio de pagos.

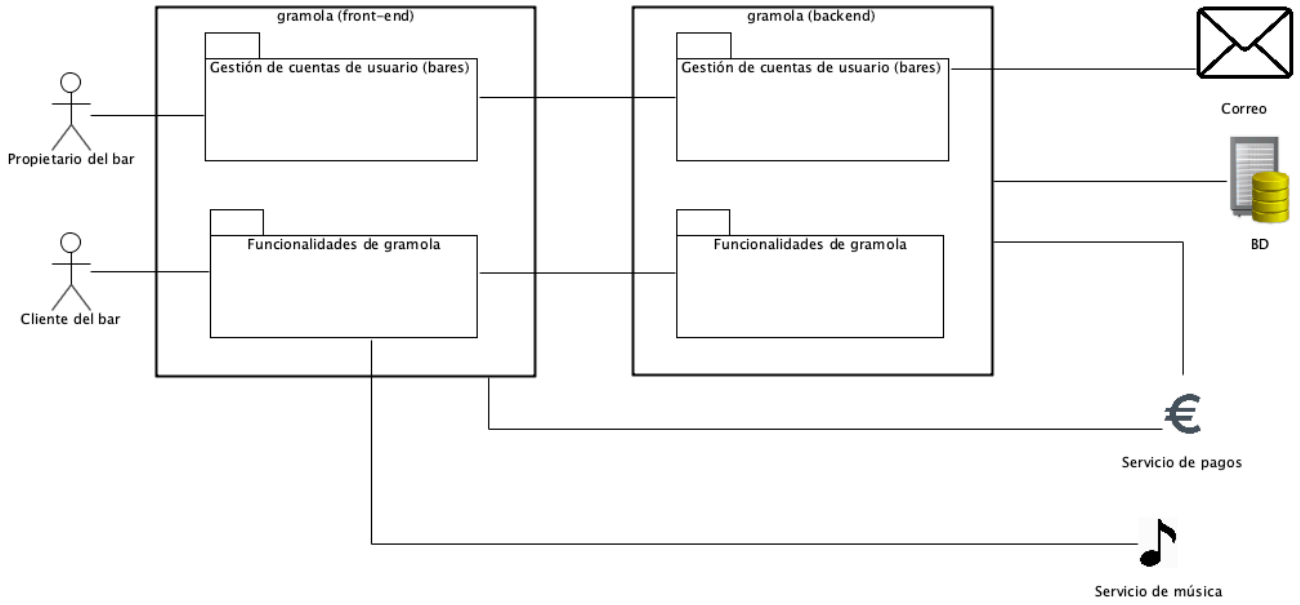

Figura 3. Vista arquitectónica del sistema

2) Hay un backend que ofrece las "contrapartes" de las funcionalidades del front:
    a. En *Gestión de cuentas de usuario*, se implementan las funcionalidades necesarias para dar servicio al registro, logueo, etcétera, que se ejecutan desde el front. Nótese que este grupo utiliza un servicio de correo electrónico, que se usa para confirmar la creación de una cuenta y para recuperar la contraseña.
    b. En *Funcionalidades de gramola,* el backend guarda en la base de datos las canciones que los usuarios van solicitando. Además, se conecta con el proveedor de pagos para gestionar el pago del importe que cuesta poner una canción.

3.2 Casos de uso del lado cliente (front-end)

El actor "Propietario del bar" podrá ejecutar al menos las siguientes funcionalidades:

1) Crear una cuenta para el bar. Deberá proporcionar el nombre del bar, un correo electrónico, una contraseña y su confirmación. Cuando el backend recibe estos datos,

le enviará un correo electrónico de confirmación que incluirá en su cuerpo un mensaje de bienvenida y un link con un token. Cuando haga clic en este link, se le reconducirá a otra pantalla para que pague el servicio. Se ofrecerán suscripciones mensuales y anuales. Los precios no se "hardcodearán" ni se guardarán en archivos de recursos, sino que estarán guardados en la base de datos.
2) Loguearse con el correo electrónico y la contraseña.
3) Recuperar la contraseña. Se escribirá el correo electrónico en un formulario y el backend enviará un correo a esa dirección con un token para que el usuario pueda cambiarla.
4) Cerrar sesión.

El actor "Cliente del bar" podrá:

1) Buscar una canción.
2) Insertar una canción en la cola (recuerde que la canción "se cuela" según se ha explicado en la Sección 2, "Escenario de uso"). Esto requiere el pago de cierta cantidad, que estará en la base de datos.
3) Cuando la canción se inserta, pasa a sonar a continuación.

3.3 Casos de uso del servidor (backend)

El backend dará servicio a todos los casos de uso del lado cliente del actor "Propietario del bar". Además:

1) Almacenará en la base de datos las canciones que los clientes vayan eligiendo.
2) Apoyará en la realización de los pagos cuando los clientes elijan canciones.

3.4 Lenguajes y entornos de programación

El backend se desarrollará en Java y se usará el framework Spring. Para el lado cliente se utilizará Angular.

La base de datos se implementará con MySQL.

3.5 Arquitectura del front y del back

El front constará de vistas, modelos y servicios.

El back estará estructurado en controladores, servicios, repositorios y el resto de clases que se consideren necesarias.

3.6 Uso de IA

Por supuesto, pueden utilizarse servicios de Inteligencia Artificial para el desarrollo del trabajo, lo cual no exime de saber qué se está haciendo, ya que en la defensa de la práctica, que se hace presencialmente, se puede preguntar sobre qué hace cierto fragmento de código, que se reproduzcan "a mano" trazas de ejecución, etcétera.

4 Criterios de valoración

La práctica debe hacerse individualmente. Obviamente, se podrá pedir ayuda a otros compañeros, preguntar cómo han hecho algo, resolver dudas con ellos, etcétera, pero cada uno es responsable de su práctica y debe conocer todos sus detalles.

La práctica se valora sobre 10 puntos conforme a estos criterios generales:

Criterio	**Puntos**	**Notas**
Gestión de cuentas de usuario (front y back)	2	Si esto no funciona perfectamente, la práctica está suspensa.
Buscar una canción e insertarla en la cola	2	Si esto no funciona perfectamente, la práctica está suspensa.
Reproducir canción	2	Si se usa Spotify no se tiene cuenta premium, la canción no podrá reproducirse. En este caso, puede optarse por usar otro sistema de streaming o por simular de alguna manera que la canción se está reproduciendo.
Casos de prueba funcionales	2	Se determinarán más adelante los escenarios de prueba
Interfaz responsive	1	
Buen diseño arquitectónico	1	

5 Entrega y defensa

La práctica se subirá al campus virtual antes del sábado 17 de enero. La defensa se hará presencialmente con posterioridad al examen (el día 19 de enero).