|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Tecnologías de Desarrollo de Software**  **C:\Users\tonto_000\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\B4DA7523.tmp**  **Práctica final:** AppMusic | | |

Grupo 3.1

**Profesor/a:** Jesús Joaquín García Molina

**Realizado por:**   
Alberto Cuadrado López (alberto.cuadradol@um.es) y Sergio Escudero Manzano ([sergio.escuderom@um.es](mailto:sergio.escuderom@um.es)).

**Convocatoria:** Junio 2021

Índice

[1. Diagrama de clases del dominio 3](#_Toc73469609)

[1.1. Elementos destacables del dominio 3](#_Toc73469610)

[2. Diagrama de colaboración (añadir playlist) 4](#_Toc73469611)

[3. Arquitectura de la aplicación 5](#_Toc73469612)

[4. Patrones de diseño utilizados 6](#_Toc73469613)

[5. Componentes utilizados 7](#_Toc73469614)

[5.1. Cargador de canciones 7](#_Toc73469615)

[5.2. Pulsador 7](#_Toc73469616)

[5.3. JCalendar 7](#_Toc73469617)

[5.4. Tema oscuro 7](#_Toc73469618)

[5.5. Generación de PDFs 7](#_Toc73469619)

[6. Tests unitarios 8](#_Toc73469620)

[7. Manual de usuario 9](#_Toc73469621)

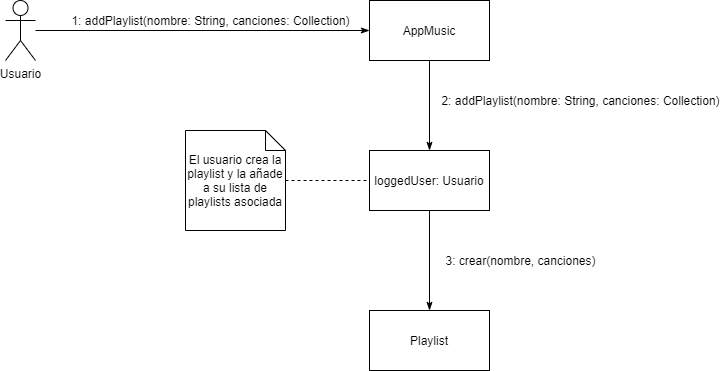
[8. Observaciones 15](#_Toc73469622)

# Diagrama de clases del dominio

## Elementos destacables del dominio

* loggedUser: esta variable estará en el controlador y guardará el usuario que está usando la aplicación en ese momento, puesto que solo va a haber hasta un usuario usando la aplicación al mismo tiempo. Esta variable se establece al ejecutar login(username, password) o logout() de modo que preservamos la seguridad de cara a los usuarios.
* Reproductor: el reproductor solo va a almacenar una canción y es la que esté reproduciendo en ese momento. Esto es, cuando se ejecute reproducirCancion().
* Catálogos: los catálogos hacen uso del patrón singleton al igual que el controlador, de modo que son entidades independientes.

# Diagrama de colaboración (añadir playlist)



En este caso no hace falta acceder al catálogo de usuarios por lo mencionado anteriormente, y es que el controlador contiene una variable con el usuario que haya iniciado sesión, de modo que no hay que acceder siempre a dicho catálogo.

# Arquitectura de la aplicación

La aplicación sigue el patrón Modelo-Vista-Controlador, de modo que identificamos 3 partes diferentes:

* Vista: GUI programada haciendo uso de Swing.
* Controlador: clase AppMusic encargada de interactuar con el modelo.
* Modelo: conjunto de clases que representan la información del sistema.

Por otro lado, el modelo es persistente gracias al uso del patrón DAO, de modo que las clases Usuario, Canción y Playlist no se pierden al reiniciar la aplicación. Esto es gracias a la base de datos H2 proporcionada y su respectivo controlador.

En cuanto a la estructura de la interfaz, estamos hablando de que tenemos una clase que contiene todos los demás JFrames de la aplicación, en nuestro caso es MainWindow. Esta contiene un border layout el cual se distribuye de la siguiente manera:

* Arriba: Información y botones (mejora de cuenta, carga de canciones y cierre de sesión).
* Abajo: Controles e información de reproducción.
* Izquierda: Panel de navegación y lista de playlists.
* Centro: Paneles de búsqueda, modificación de playlist, etc.

Cabe destacar que el centro contiene varios paneles, pero solo puede tener uno visible al mismo tiempo, de modo que, cambiar entre las diferentes partes de la aplicación es cuestión de ocultar todos los paneles y hacer visible el que hayamos seleccionado.

# Patrones de diseño utilizados

Algunos de los patrones de diseño que hemos empleado son los siguientes:

* Observer (Reproductor): hemos visto necesario aplicar este patrón en este lugar ya que cuando el reproductor termina de reproducir una canción, en la interfaz necesitamos saber cuando acaba para actualizar la información que se le muestra al usuario.
* Observer (Swing): para conocer cuando el usuario interactúa con la aplicación, se hace uso de este patrón, para introducir código a esas acciones por medio de Listeners, los cuales se ejecutan cuando se realiza una determinada acción, como pulsar un botón.
* Controlador (AppMusic): para la gestión de información del sistema, utilizamos el patrón controlador, que va a ser el encargado de interactuar con las clases del dominio, centrando las acciones en una única clase.
* Experto y creador: dentro del modelo se respetan los patrones experto y creador, los cuales distribuyen las responsabilidades en las clases correspondientes del dominio.
* Factoría abstracta: para la implementación de un servicio de persistencia independiente de la aplicación, hacemos uso de este patrón, de modo que separamos la declaración de la persistencia de la implementación, permitiendo al modelo optar por varias opciones cambiando un parámetro a la hora de obtener la factoría.

# Componentes utilizados

## Cargador de canciones

Este componente se encarga de cargar las canciones de un archivo XML extrayendo sus datos y URL. De por sí, el componente nos permitía cargar las canciones, pero para adaptarlo al sistema y evitar el acoplamiento, se hace uso de una interfaz IBuscadorCanciones y de CancionesListener, de modo que podemos establecer el archivo de canciones y añadir un listener para notificar del evento a nuestra aplicación.

## Pulsador

Este pulsador se ha utilizado en la interfaz para abrir la selección de archivo XML para la carga de canciones. Es simplemente un botón que se enciende o se apaga cada vez que se pulsa.

## JCalendar

JCalendar nos permite añadir el selector de fecha que encontramos en la ventana de registro, de modo que es más accesible y sencillo para el usuario, puesto que no tiene que seguir un patrón concreto al introducir la fecha, sino que el propio componente lo gestiona.

## Tema oscuro

Para el tema oscuro de swing que hemos utilizado, hemos empleado una librería de código abierto disponible en este [enlace](https://github.com/weisJ/darklaf). Su uso es muy sencillo y se ejecuta al inicio de la aplicación, de modo que cambia todo el estilo de Swing automáticamente. Toda la información está en su repositorio de GitHub.

## Generación de PDFs

La librería ITextPDF nos permite generar PDFs de manera sencilla y rápida. En nuestro caso hemos hecho que los archivos generados sean muy simples para no extender demasiado el código, puesto que es una funcionalidad secundaria.

# Tests unitarios

Para la realización de los tests, hemos optado por la clase Usuario, la cual es la más compleja del modelo y ofrece múltiples posibilidades a la hora de realizar pruebas.

Los 3 primeros tests están relacionados con los descuentos, y es que, estos se dan en función de la fecha de nacimiento del usuario. Cuando el usuario es menor de 25 años, se le da un descuento de joven, y cuando es mayor de 65, un descuento de jubilado. Es por eso por lo que en estos tests comprobamos si usuarios con diferentes edades reciben los descuentos o no, para comprobar esta funcionalidad.

Por otro lado, comprobamos si un usuario es capaz de crear una playlist y guardarla, añadiéndola y retirándola del usuario. Si la playlist no es null, es que se ha registrado correctamente.

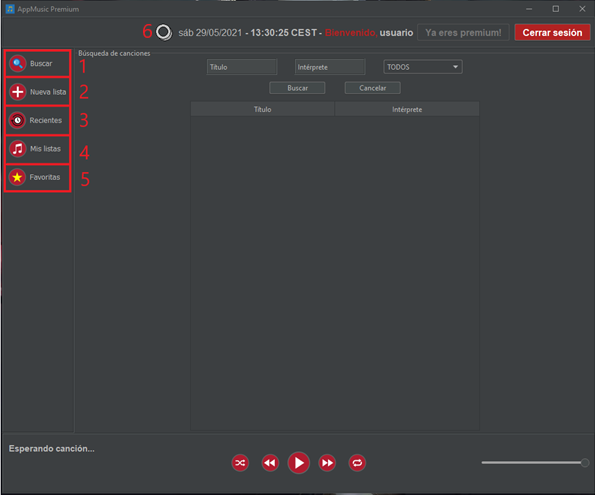
Para terminar, comprobamos que las playlists registradas se pueden eliminar. Antes hemos comprobado que las playlists se registran, por lo que en esta prueba tenemos la certeza de que el usuario la contiene. Después de registrarla, la eliminamos e intentamos recuperarla. El resultado debe ser null, puesto que la hemos eliminado.

# Manual de usuario

Pasamos ahora a ver cada una de las funcionalidades de AppMusic que no son tan triviales como “Iniciar sesión”, “Registrarse”, “Ser premium” o “Cerrar sesión” y que podemos observar en la *Figura X1*:

1. **Buscar canciones:** en esta ventana podremos buscar las canciones que queramos dependiendo del título, intérprete y estilo *(ver Figura X2)*.
2. **Nueva lista:** en esta ventana podremos crear listas de reproducciones, modificarlas y añadir las canciones que queramos a cada una de ellas *(ver Figura X3)*.
3. **Recientes:** en esta ventana podremos ver las canciones más recientemente reproducidas *(ver Figura X4)*.
4. **Mis listas:** en esta ventana podremos ver cada una de las listas de reproducciones que hemos creado anteriormente y si somos usuario premium generar un archivo pdf con la lista y sus canciones *(ver Figura X5)*.
5. **Favoritas:** esta ventana es solo para usuarios premium y aquí podremos ver la lista de canciones más escuchadas con la canción y el número de veces reproducida por el usuario *(ver Figura X6)*.
6. **Cargar canciones:** a partir de un archivo XML podremos cargar canciones que podremos escuchar posteriormente *(ver Figura X7)*.

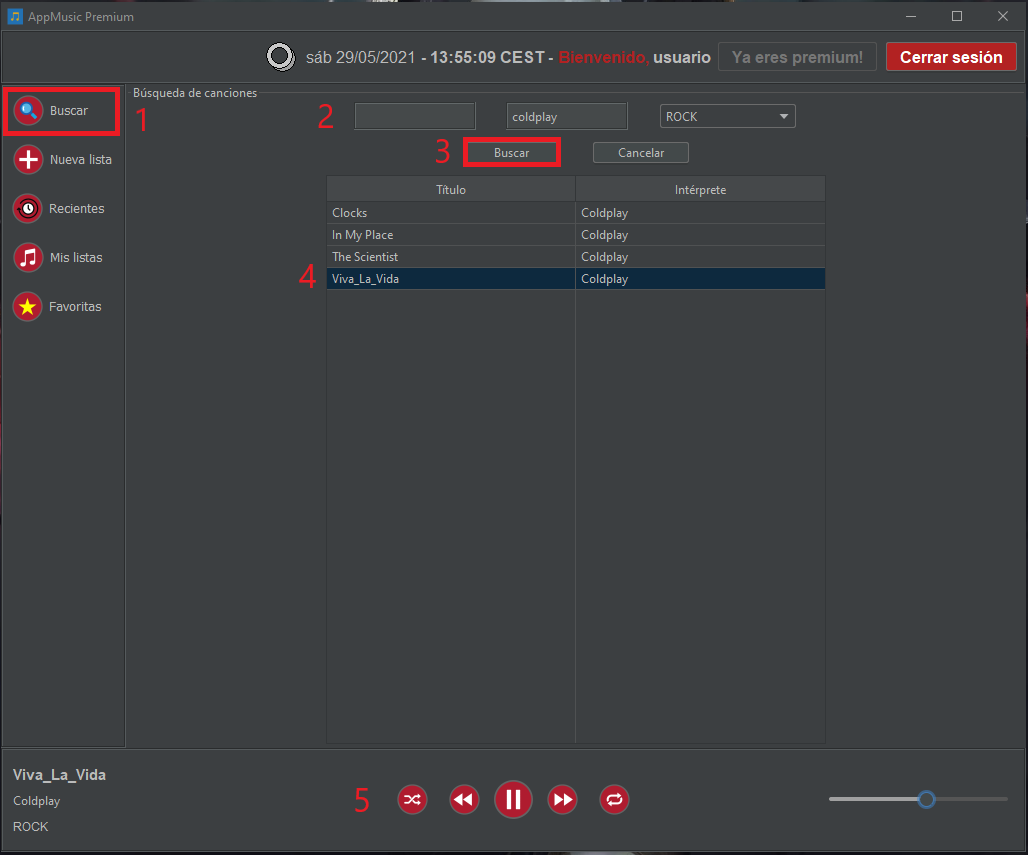
Aparte cabe destacar que en cada una de las ventanas podremos utilizar todas las funciones del reproductor de música.



*Figura X1*

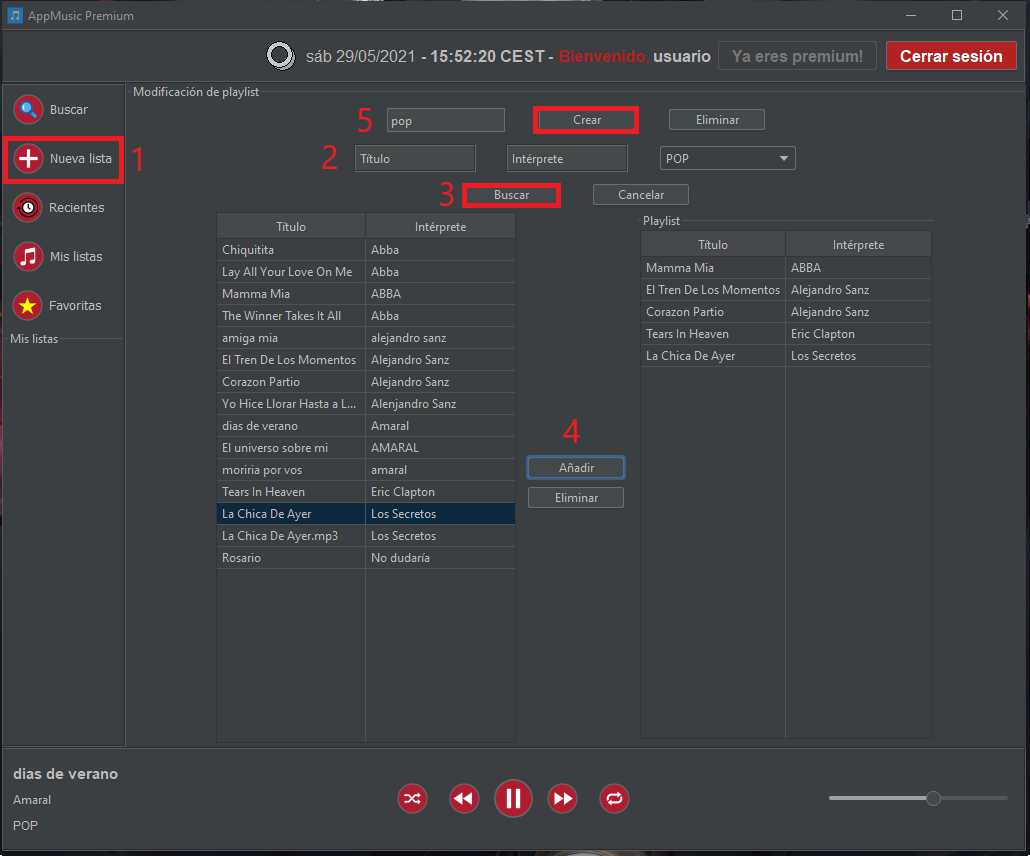
Ahora vamos a ir viendo cada funcionalidad con sus respectivos pasos:

**1.- Buscar canciones:** para buscar canciones primero pinchamos en el menú en *“Buscar”* (1)y ahora podremos buscar canciones utilizando como filtros de búsqueda el título, intérprete y estilo musical, introduciendo los datos en los campos de texto de *“Título”* e *“Intérprete”* y seleccionando un estilo disponible (2) y pinchamos en el botón de *“Buscar”* (3). Entonces el usuario podrá seleccionar y reproducir cualquiera de la lista de canciones retornada (4) además de contar de las funciones del reproductor descritas anteriormente (pausar, avanzar, retroceder, repetir, subir y bajar volumen) (5), todos estos pasos se pueden ver con mejor detalle en la *Figura X2*.



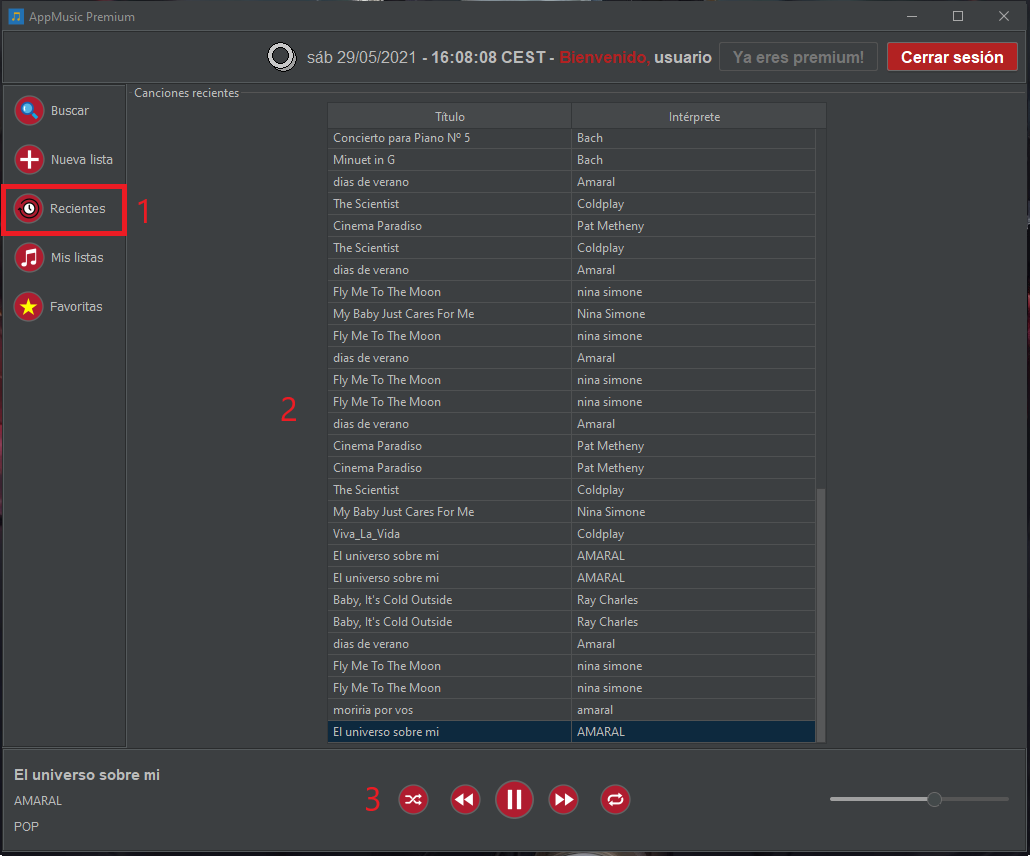
*Figura X2*

**2.- Nueva lista:** para crear una nueva lista de reproducción primero pinchamos en el menú en *“Nueva lista”* (1)y ahora podremos buscar canciones utilizando como filtros de búsqueda el título, intérprete y estilo musical como en la ventana de *“Buscar canciones”* (2) y pinchamos en el botón de *“Buscar”* (3). Entonces podremos seleccionar y añadir o eliminar cualquiera de las canciones de esa lista de reproducción (4), finalmente pinchamos en el botón de *“Crear”* (5), todos estos pasos se pueden ver con mejor detalle en la *Figura X3*.



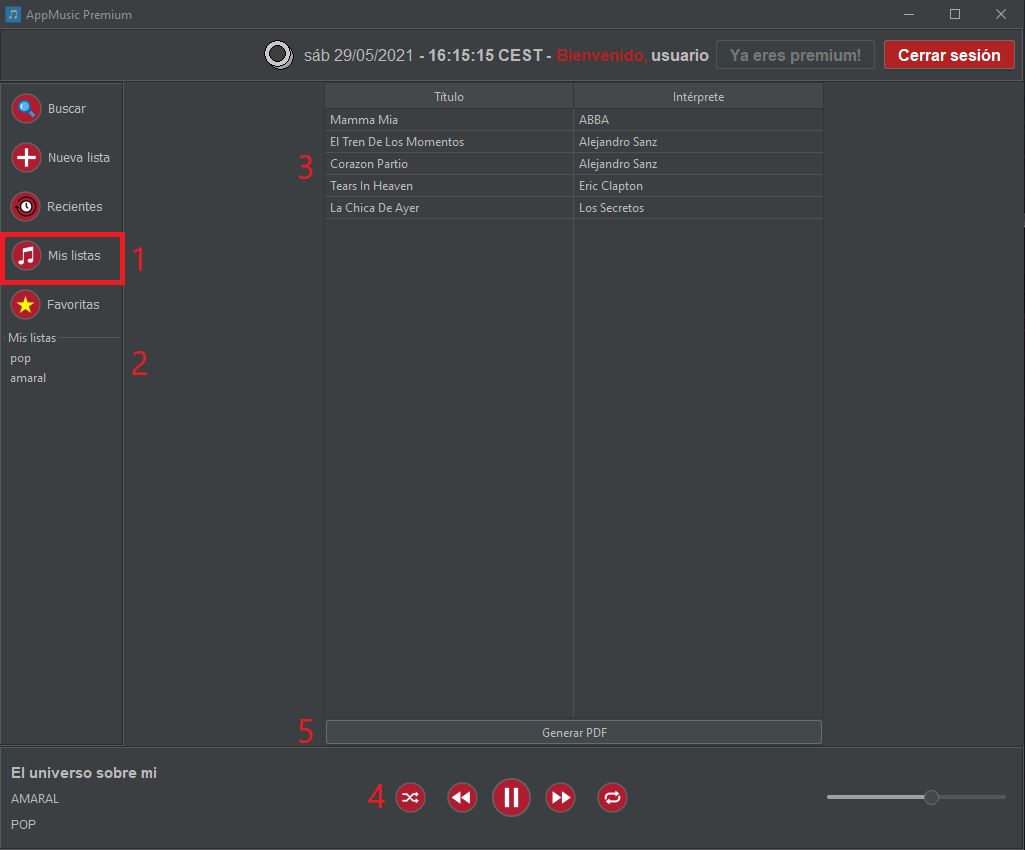
*Figura X3*

**3.- Recientes:** para ver las canciones que hemos reproducido más recientemente simplemente pinchamos en el menú en *“Recientes”* (1)y ahora podremos seleccionar y reproducir cualquiera de la lista de canciones retornada (2) además de contar de las funciones del reproductor descritas anteriormente (pausar, avanzar, retroceder, repetir, subir y bajar volumen) (3), todos estos pasos se pueden ver con mejor detalle en la *Figura X4*.



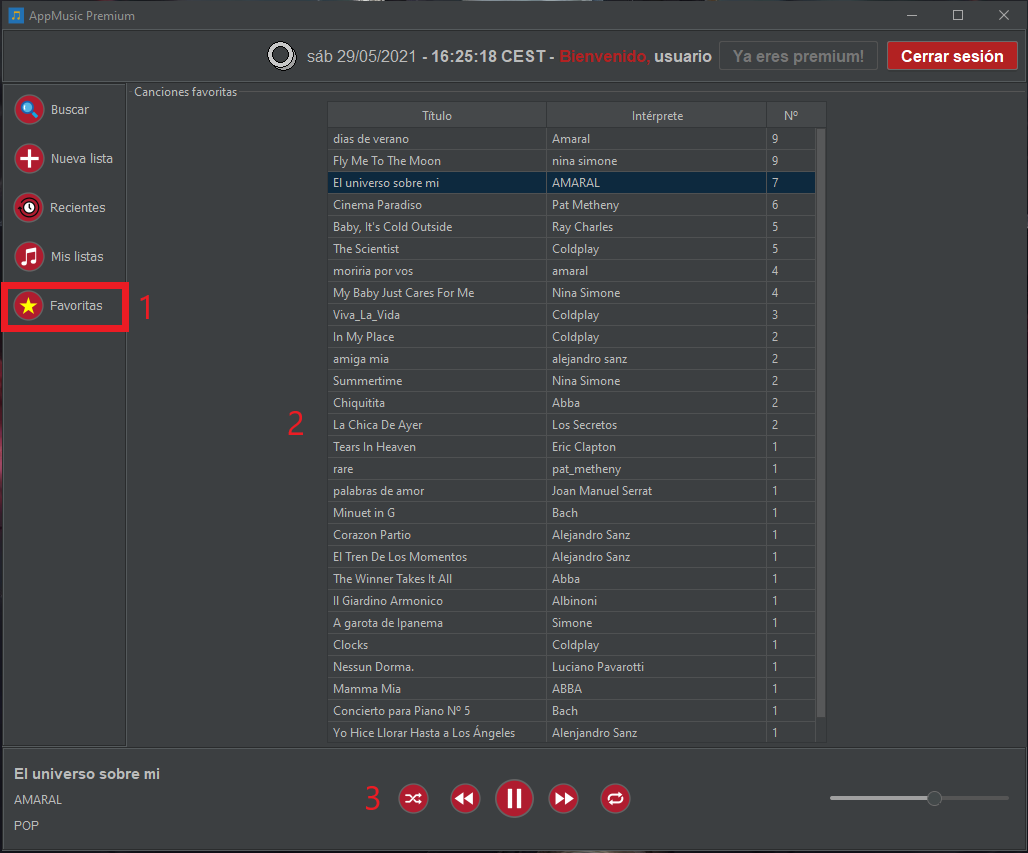
*Figura X4*

**4.- Mis listas:** para ver las canciones de las lista de reproducción que hemos creado anteriormente primero pichamos en *“Mis listas”* (1) y podremos ver y seleccionar nuestras listas de reproducción (2)*”,* ahora podremos seleccionar y reproducir cualquiera de la lista de canciones retornada (3) además de contar de las funciones del reproductor descritas anteriormente (pausar, avanzar, retroceder, repetir, subir y bajar volumen) (4) y por último si somos usuario premium podremos darle al botón de *“Generar PDF”* para generar un archivo pdf con el nombre de la lista y las canciones(5), todos estos pasos se pueden ver con mejor detalle en la *Figura X5*.



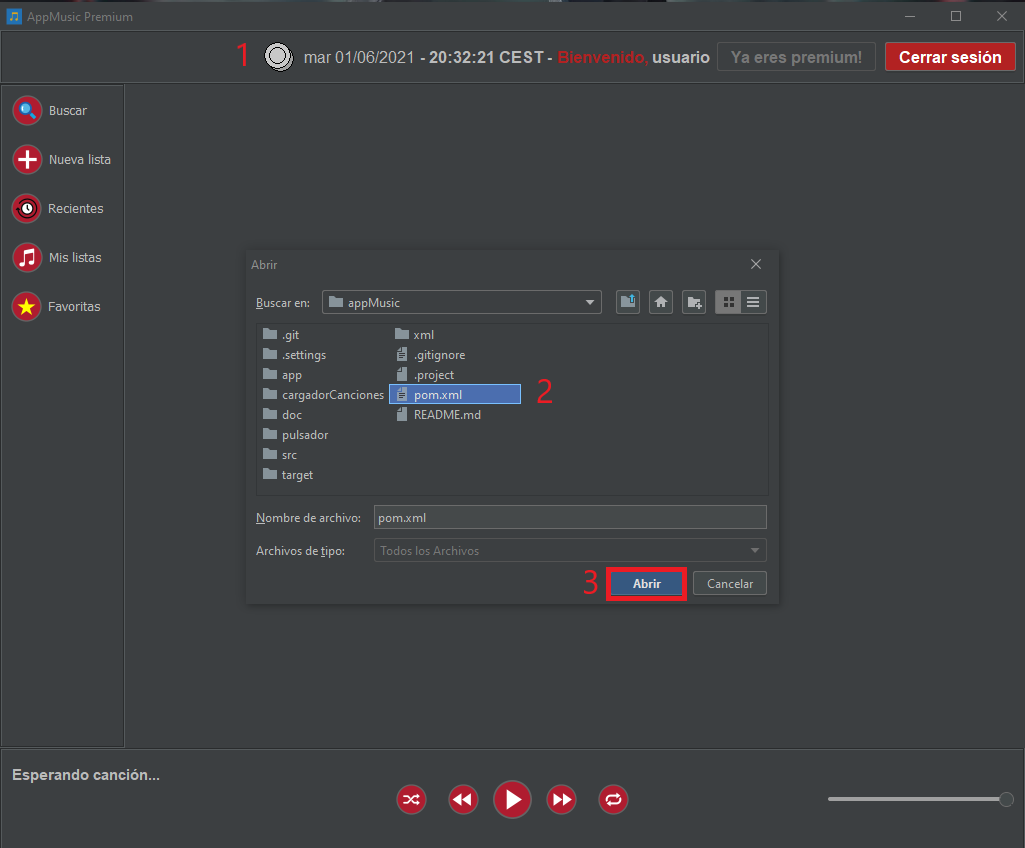
*Figura X5*

**5.- Favoritas:** para ver las canciones que más hemos reproducido en la aplicación pinchamos en el menú en *“Favoritas”* (1)y ahora podremos seleccionar y reproducir cualquiera de la lista de canciones retornada (2) además de contar de las funciones del reproductor descritas anteriormente (pausar, avanzar, retroceder, repetir, subir y bajar volumen) (3), todos estos pasos se pueden ver con mejor detalle en la *Figura X6*.



*Figura X6*

**6.- Cargar canciones:** para cargar canciones a partir de un fichero XML solo habrá que pulsar el botón de la parte superior de la interfaz (1) y se nos abrirá una ventana donde tendremos que buscar y seleccionar nuestro archivo XML (2) y finalmente pinchamos en el botón de “Abrir” (3), esperamos un tiempo que depende de las canciones y el tamaño de estas, cuando se hayan cargado todas ya las podremos buscar y reproducir cuando queramos, todos estos pasos se pueden ver con mejor detalle en la *Figura X7*.



*Figura X7*

# Observaciones

En primer lugar, queremos comentar el hecho de que la práctica está muy completa y sin duda ha sido la más entretenida y formativa del curso. Pensamos que es un buen punto de partida para introducirse en el desarrollo de software y en el mundo de los patrones de diseño.

Por otro lado, ver por primera vez que lo que escribimos toma forma y podemos verlo y escucharlo, es muy motivador, y seguramente para muchos otros sea igual. Las prácticas de esta asignatura son diferentes, y suponen una gran motivación para aquellos que cursen la asignatura y estén en la misma situación que nosotros, es decir, introduciéndonos en la Ingeniería del Software.

Esperamos que esta asignatura perdure y siga creciendo y mejorando, ya que creemos que es la más formativa de los 3 cursos que llevamos.