Programación Multimedia y Dispositivos Móviles

Proyecto - Spotify

Autor: Antonio Calabuig Puigvert y Sebastián Villa Ponce
Centro: IES Salvador Gadea
Departamento: Informática
Ciclo: Desarrollo de Aplicaciones Multiplataforma
Fecha: 2026-02-15

Proyecto - Spotify

> Implementación MyMusicApp con datos desde API - HTTP + Authentication

1. Introducción

Vas programar una aplicación de móvil completamente desde cero en la que se van a evaluar todo el contenido del curso:

- Creación de componentes
- Creación de hooks
- Uso de useEffect y useState
- Creación de estilos (NativeWind)
- Navegavción con Stack, Tabs y Drawer
- CRUD sobre información desde fuentes externas (API) mediante Axios y TanStackQuery
- Almacenamiento local plano (AsnycStorage) y encriptado (SecureStore)
- Autenticación. Simulación mendiante estados Zustand sobre dos endpoints personalizados (/login, /register)

Para el desarrollo de la aplicación se pueden reutilizar diversos cores implementados a lo largo del curso en las aplicaciones prácticas que se han ido realizando.

Enlaces para la consulta de documentación oficial:

- NativeWind como librería de estilos.
- Stack para enrutado.
- Drawer configurado bajo las necesidades de la aplicación.
- Tabs necesarios bajo las necesidades de la aplicación.
- Axios.
- TanStack para optimización de llamadas a API.
- AsyncStorage para almacenar información clave-valor en texto plano.
- SecureStore para almacenar de forma local información encriptada.
- Zustand para gestión del estado de los permisos.
- Authentication - Core similar al de ProductsApp.
- API Spotify

2. Enunciado

Diseño y estilos

El diseño y estilos se definirá según la necesidad de la aplicación. Se puede definir la propia **paleta de colores** a partir de NativeWind así como la **fuente** con mejor se adapte para trabajar en la aplicación. Se debe mantener un estilo de pantallas homogéneo.

Implementación de pantallas

1. Login
2. Registro
3. Tabs
    1. Home
    2. Busqueda
    3. Biblioteca
    4. Añadir (+) (modal) *¡BOOM!*
4. Drawer
    1. Perfil
    2. Configuración
    3. Suscripciones (si premium)

Pantallas internas

1. Playlists
2. Playlist
3. Podcasts
4. Podcast
5. Capitulo
6. Artistas
7. Artista
8. Albumes
9. Álbum

Descripción del contenido de las pantallas

1. **Login**
   Login mediante usuario y contraseña contra el endpoint de /login

2. **Registro**
   Registro mediante datos mínimos para crear un usuario contra el endpoint de /register

3. **Tabs**

    1. **Home**
       Tres Flatlists:

        - **FL1**: Horizontal con listado de playlists seguidas por el usuario
        - **FL2**: Horizontal con listado de álbumes seguidos por el usuario
        - **FL3**: Vertical con el listado de canciones que le gustan al usuario

    2. **Búsqueda** Pantalla con un cuadro de búsqueda en la parte superior que a partir del tercer carácter lanza llamada a la API para buscar información entre **playlists, artistas, álbumes, canciones y podcasts** para traer el listado de búsqueda filtrado.

       Para que la pantalla de inicio no aparezca vacía, tendremos 4 botones que enlazarán con las pantallas de **Playlists**, **Podcasts**, **Artistas** y **Álbumes**. En este caso todos, no los que sigue el usuario.

       Al hacer click sobre un elemento representados en la lista de búsqueda:

        - Si es una **playlist** nos lleva al listado de canciones de la propia playlist.
        - Si es un **artista** nos lleva al listado de álbumes del propio artista.
        - Si es un **álbum** nos lleva al listado de canciones del propio álbum.
        - Si es un **podcast** nos lleva al listado de capítulos del propio podcast.
        - Si es una **canción** mostrará un *(+)* a la derecha para añadirla a una playlist.

    3. **Biblioteca** Contine un Flatlist horizontal estático con botones que intercambian la información de lo que muestra la Flatlist vertical del contenido. *Botones*:

        - Listas que sigue el usuario.
        - Podcasts que sigue el usuario.
        - Álbumes que sigue el usuario.
        - Artistas que sigue el usuario. *Por defecto muestra*:
        - En primer lugar una lista estática de las canciones que le gustan al usuario.
        - Listado de playlists que sigue el usuario.

       Al hacer click sobre una playlist nos lleva al listado de sus canciones.

4. **Añadir** (+) (modal) *¡BOOM!* Al pulsar sobre esta opción de las Tabs, debe abrir un modal que nos permita crear una playlist.

4. **Drawer**

    1. **Perfil** Permite modificar los datos personales del usuario. Al guardar los cambios deben refrescarse sobre la pantalla.

    2. **Configuración** Permite cambiar los parámetros de configuración del usuario. Al guardar los cambios deben refrescarse sobre la pantalla.

    3. **Suscripciones** (*si premium*) En caso del que el usuario sea premium, enlazara con una pantalla de listado de sus suscripciones. *(No es necesario mostrar información sobre el método de pago)*.

Pantallas internas

1. **Playlists**:
   Listado de todas las playlists (*Es con la que enlaza el botón de la pantalla de búsqueda*).

2. **Playlist**:
   Listado de canciones de una playlist. *<Permitir agregar canción a otra playlist>*

3. **Podcasts**:
   Listado de todos los podcasts (*Es con la que enlaza el botón de la pantalla de búsqueda*).

4. **Podcast**:
   Listado de capitulos de un podcast > **Capítulo** (*Detalle de capítulo*)

5. **Artistas**:
   Listado de artistas (*Es con la que enlaza el botón de la pantalla de búsqueda*)

6. **Artista**:
   Listado de ábumes de un artista

7. **Albumes**:
   (*Es con la que enlaza el botón de la pantalla de búsqueda*)
   Listado de álbumes > **Álbum** (Detalle de álbum con su listado de canciones)

Fuente de datos

La fuente de datos será vuestra propia API desarrollada en el módulo de Acceso a Datos está disponible en API Spotify.
La información de la que dispone es méramente una muestra singular y para nada cercana a la realidad.

Puede que en algún momento se precise la modificación de la misma para poder enviar la información de forma estructurada según lo requiera cada una de las pantallas.

Los endpoints se encuentran disponibles en cada una de las diferentes coleccioens de Postman que se han desarrollado en la implementación de la API de cada alumno.

Recomendación

Intentad reutilizar componentes de renderizado de listas en la medida que sea posible. Recordad que en ocasiones hemos hecho Mappers de datos para adecuar una respuesta de datos a otra más simple.

> Por ejemplo:
>
> - Si todos los listados en su mayoría contendrán información como título, descripción, artista, etc..., se puede crear un mapper genérico con campos nullables y booleanos para saber tipo de elemento y una vez dibujado sobre el componente mostrar opciones o no.