

**Documento de prescripciones técnicas y
funcionales del proyecto de los módulos :
Desarrollo en la parte Cliente y Desarrollo
en la parte Servidor**



Índice de contenidos

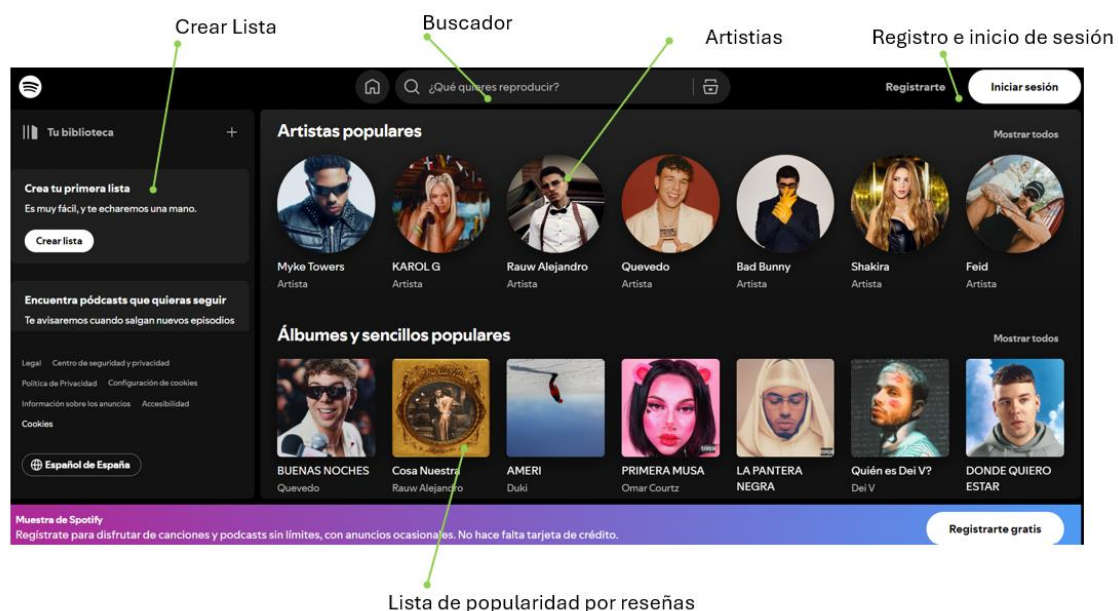
1.	Objeto del Proyecto (MySpotify)	3
2.	Funcionalidades principales del sistema	3
3.	Requisitos Técnicos	5
3.1.	Arquitectura de la aplicación web.....	5
3.2.	Base de datos	5
3.3.	Funcionalidades de geolocalización	5
3.4.	Requisitos de Rendimiento	5
3.5.	Seguridad	6

1. Objeto del Proyecto (MySpotify)

El presente documento tiene como objeto **definir los requisitos técnicos y funcionales** para el desarrollo de una aplicación web similar a Spotify, que permita **la reproducción de música** y la **interacción social entre usuarios**, añadiendo funcionalidades **como reseñas de grupos musicales** y la **localización geográfica de usuarios** que compartan intereses musicales.

La idea principal del proyecto es hacer una versión de Spotify personalizada con los gustos musicales de los alumnos y en este contexto desarrollar lo aprendido en los módulos de segundo, principalmente desarrollo en servidor y cliente.

2. Funcionalidades principales del sistema



- **Reproducción de música en streaming:**
 - Los usuarios **podrán escuchar canciones en tiempo real** desde una biblioteca de música cargada en el servidor o directamente.
 - La aplicación deberá **ofrecer funciones como** "Play", "Pause", "Siguiente", "Anterior"
 - La aplicación deberá ofrecer la **función de "Búsqueda de canciones/artistas"**
- **Gestión de usuarios:**
 - **Autoregistro** de usuarios mediante correo electrónico y contraseña.
 - **Autenticación** basada en credenciales simples
 - **Perfil de usuario** con información básica (nombre, foto de perfil, gustos musicales ,etc).

- **Creación y gestión de listas de reproducción:**
 - Los usuarios podrán crear, modificar y eliminar **listas de reproducción** personalizadas.
 - Opción de **compartir listas de reproducción con otros usuarios**.
- **Reseñas de grupos musicales:**
 - Los usuarios podrán **escribir reseñas de artistas y grupos musicales**.
 - Los usuarios podrán valorar artistas, canciones
 - Las reseñas deberán poder ser valoradas (con "Me gusta") y comentadas por otros usuarios.
- **Localización geográfica de usuarios:**
 - Los usuarios **podrán permitir que la aplicación acceda a su ubicación para asociarse con grupos de música**.
 - **Posibilidad de visualizar un mapa con las ubicaciones aproximadas de otros usuarios a los que les gusta un determinado grupo musical**.
 - **Función para "buscar cerca de mí"** para encontrar personas con gustos musicales similares.
- Función de **recomendación** .
 - En función de las preferencias los usuarios podrán recibir recomendaciones de
 - Personas con gustos similares
 - Grupos y artistas que me puedan interesar
 - Listas que me puedan interesar
- **Funcionalidades sociales**
 - Los usuarios podrán **seguir a otros usuarios y tener una lista de amigos**.
 - Se podrá **ver la actividad reciente de los amigos** (como las reseñas publicadas o listas de reproducción compartidas).
- **Mensajes y notificaciones (ampliable):**
 - Sistema de mensajería entre usuarios.
 - Notificaciones de eventos importantes, como comentarios en reseñas o cuando un amigo sigue una nueva banda.
- **Roles de acceso a la aplicación:**
 - La aplicación deberá dar soporte a los siguientes roles: usuario registrado, usuario administrador, usuario explotación

3. Requisitos Técnicos

3.1. Arquitectura de la aplicación web

- **Front-end:**
 - Se debe desarrollar una interfaz web responsive utilizando tecnologías como HTML5, CSS3 y JavaScript (usando un framework como React, Vue.js o Angular).
- **Back-end:**
 - El servidor debe ser desarrollado en **PHP** utilizando Symfony para facilitar el desarrollo.
 - Se implementará una API REST para gestionar las operaciones de creación, lectura, actualización y eliminación (CRUD) de las distintas entidades del sistema (usuarios, canciones, listas de reproducción, reseñas, etc.).
 - Se utilizará API's para acceder a servicios externos. Como la listas de canciones más escuchado

3.2. Base de datos

- **MongoDB y MySQL** como sistema de base de datos para almacenar usuarios, canciones, listas de reproducción, reseñas y otros datos de la aplicación. **(NOTA se puede usar MySQL como transaccional y MongoDB como base de datos de metadatos por ejemplo para el perfil del usuario, donde figuren gustos y preferencias)**

3.3. Funcionalidades de geolocalización

- Uso de la API de **Google Maps** o **Leaflet** para la visualización de mapas y la localización geográfica de los usuarios.
- Integración con **HTML5 Geolocation API** para obtener la ubicación del usuario desde el navegador.

3.4. Requisitos de rendimiento

- La aplicación debe soportar la reproducción en streaming sin interrupciones **.(NOTA a aquí se podría explorar la posibilidad de meter una memoria de cache)**
- La carga de los mapas y la información geográfica debe ser eficiente y no afectar la experiencia de usuario.

3.5. Seguridad

- Se deben implementar prácticas de seguridad robustas y reconocidas, como la protección contra ataques **XSS** y **CSRF**. Fuente de documentación OWASP
- El sistema de autenticación **debe usar contraseñas cifradas** (a valorar mecanismos de recuperación segura de cuentas).
- En las interacciones con terceros (como Google Maps u otros), se deben utilizar APIs de manera segura y con claves apropiadas.

4. Escalabilidad

- La arquitectura debe estar preparada para **escalar horizontalmente** en caso de incremento de usuarios, con un sistema de balanceo de carga en el servidor. **(NOTA este requisitos no se desarrolla en los módulos propuestos, pero podría abordarse como parte del módulo de despliegue)**

5. Requisitos de Pruebas

- Se debe realizar **testing unitario y funcional** para asegurar el correcto funcionamiento de todas las partes de la aplicación. **(NOTA este requisitos no se desarrolla en los módulos propuestos, pero podría abordarse como parte del módulo de despliegue)**
- Se realizarán **pruebas de carga** para asegurar el rendimiento del servidor en situaciones de alta demanda. **(NOTA aquí se puede usar Jmeter)**