

SPOTIFY TO VYNIL

HECHO POR:

- ELÍAS GONZÁLEZ VALDEPEÑAS (elias.gonzalez@alu.uclm.es)
- RODRIGO DE LA HOZ LÓPEZ (rodrigo.de@alu.uclm.es)
- ALBERTO PALENCIA QUIÑÓNES (alberto.palencia1@alu.uclm.es)

Desarrollador Backend -> Alberto Palencia

Desarrollador Frontend -> Elías González

Desarrollador Testing -> Rodrigo de la Hoz

Enlace GitHub

<https://github.com/AlbertoPalenciaQuinones/ISI-LAB.git>

Índice

1.	Objetivo del Sprint 2	3
1.1.	Microservicios	3
2.	Arquitectura Global	4
2.1.	Descripción	4
2.2.	Diagrama de Arquitectura	4
2.3.	Diagrama de Secuencia (Ejemplo de Búsqueda de Álbumes)	5
3.	Integración API	6
3.1.	Cómo acceder a la API de Last.fm	6
3.2.	Cómo acceder a la API de Discogs	6
4.	Base de Datos	8
4.1.	Diseño de la Base de Datos	8
4.2.	Tablas Principales	8
4.3.	Modelo Entidad-Relación (ERD)	9
4.4.	Pruebas Ejecutables en MySQL	9
5.	Balsamiq Mockups	12

1. Objetivo del Sprint 2

El objetivo principal de este sprint es desarrollar un servicio web que conecte Last.fm y Discogs para ofrecer información sobre versiones físicas de álbumes y facilitar su compra, replicando la funcionalidad del Sprint 1 pero con las nuevas APIs. Last.fm reemplaza a Spotify, ya que proporciona información detallada sobre los álbumes, incluyendo imágenes y datos de oyentes, mientras que Discogs permite verificar la disponibilidad de álbumes en formatos físicos como vinilos y CDs.

1.1. Microservicios

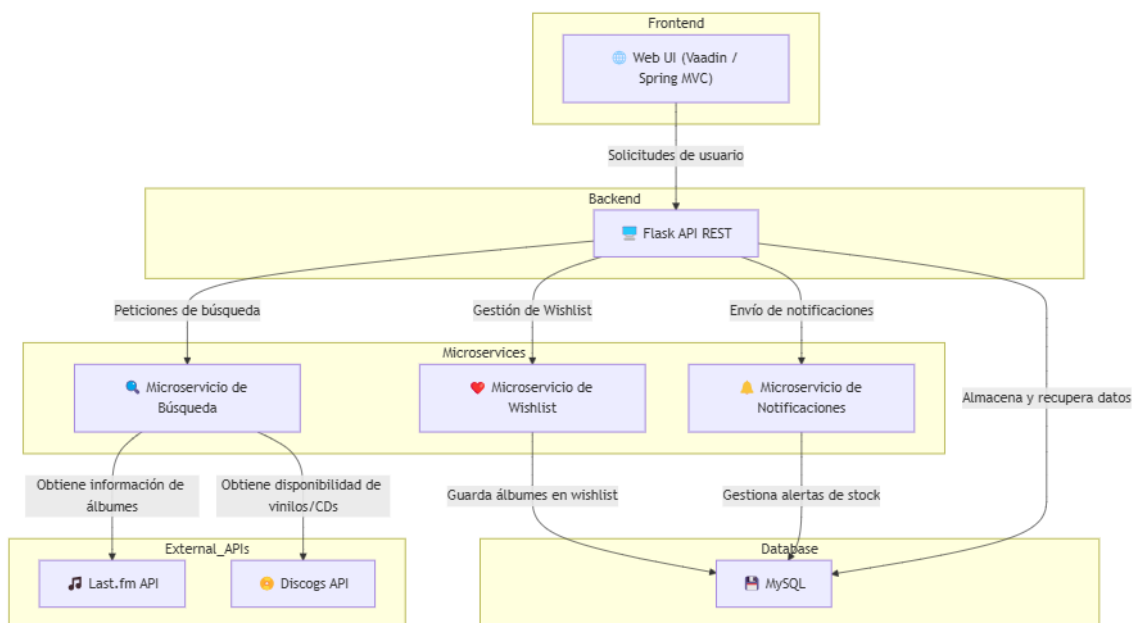
1. Microservicio de Búsqueda
 - Endpoint: /search
 - Método: GET
 - Función: Busca álbumes en Last.fm y devuelve los resultados.
2. Microservicio de Wishlist
 - Endpoint: /wishlist
 - Método: POST, GET, DELETE
 - Función: Agrega/elimina álbumes de la wishlist del usuario.
3. Microservicio de Notificaciones
 - Endpoint: /notifications
 - Método: POST, GET
 - Función: Maneja alertas de disponibilidad de álbumes.

2. Arquitectura Global

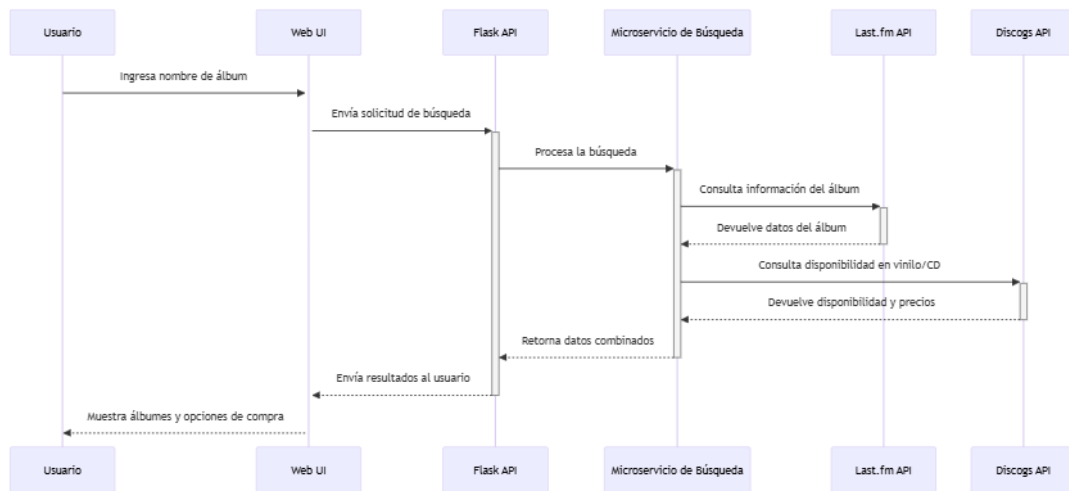
2.1. Descripción

La arquitectura del sistema *Music Finder* se basa en una estructura de microservicios que permite la integración fluida entre Last.fm, Discogs y la plataforma web.

2.2. Diagrama de Arquitectura



2.3. Diagrama de Secuencia (Ejemplo de Búsqueda de Álbumes)



3. Integración API

3.1. Cómo acceder a la API de Last.fm

Para acceder a la API de Last.fm, es necesario crear una cuenta en su plataforma de desarrolladores.

1. Ir al sitio web: **<https://www.last.fm/api/account/create>**
2. **Registrarse** en Last.fm.
3. Crear una nueva aplicación:
 - Introducir un **nombre para la aplicación** (por ejemplo: MusicFinder).
 - Agregar una **descripción breve** sobre su uso.
 - En **Application Homepage**, poner un enlace de referencia (puede ser `http://localhost` si es para pruebas).
 - En **Callback URL**, dejarlo vacío si no se necesita autenticación avanzada.

El siguiente paso será obtener la API Key. Last.fm proporcionará: API Key → Clave pública para acceder a la API. Shared Secret → Clave privada para autenticaciones avanzadas.

Guarda la API Key, ya que se usará en todas las peticiones a la API de Last.fm.

Por último, hay que hacer una prueba de conexión, para verificar que la API funciona correctamente, puedes hacer una petición en el navegador o en Postman:

Curl -X GET

```
"http://ws.audioscrobbler.com/2.0/?method=album.getinfo&api_key=TU_API_KEY&artist=The+Beatles&album=Abbey+Road&format=json"
```

3.2. Cómo acceder a la API de Discogs

Para utilizar la API de Discogs, primero hay que obtener un token de autenticación.

1. Ir al sitio web: **<https://www.discogs.com/settings/developers>**
2. **Registrarse** en Discogs.
3. Registrar una nueva aplicación:
 - Ingresar un nombre para la aplicación (ejemplo: MusicFinder).
 - Añadir una descripción breve sobre su uso.
 - Indicar una URL de referencia (puede ser `http://localhost` para pruebas).

Una vez creada la aplicación, se generará un: Token de Acceso Personal → Clave única para autenticarse en la API. Este token no expira y se usa en todas las peticiones a Discogs.

Para probar la API de Discogs, se puede hacer una petición con curl o en Postman:

```
curl -X GET
```

```
https://api.discogs.com/database/search?q=Abbey+Road&type=release&token=TU_DI  
SCOGS_ACCESS_TOKEN
```

4. Base de Datos

4.1. Diseño de la Base de Datos

La base de datos se implementará en MySQL y almacenará información clave sobre los usuarios, álbumes, wishlist y alertas, con datos obtenidos de Last.fm API y Discogs API.

4.2. Tablas Principales

Usuarios:

- id_usuario (UUID, PK)
- nombre (VARCHAR)
- email (VARCHAR, único)
- contraseña (HASHED)
- fecha_creacion (TIMESTAMP DEFAULT CURRENT_TIMESTAMP)

Álbumes:

- id_album (VARCHAR, PK) → ID de Discogs
- nombre (VARCHAR) → Nombre del álbum
- artista (VARCHAR) → Nombre del artista
- year (INT) → Año de lanzamiento
- formato (VARCHAR) → Formato (Vinilo, CD, Digital)
- url (VARCHAR) → Enlace al álbum en Discogs
- sello_discografico (VARCHAR) → Nombre del sello discográfico
- rating (FLOAT) → Calificación de la comunidad
- lastfm_listeners (INT) → Número de oyentes en Last.fm
- lastfm_plays (INT) → Número de reproducciones en Last.fm
- lastfm_url (VARCHAR) → Enlace al álbum en Last.fm
- lastfm_image (VARCHAR) → Imagen del álbum en Last.fm
- lastfm_tags (VARCHAR) → Etiquetas del álbum en Last.fm (géneros)
- discogs_availability (BOOLEAN) → Indica si el álbum está disponible en stock

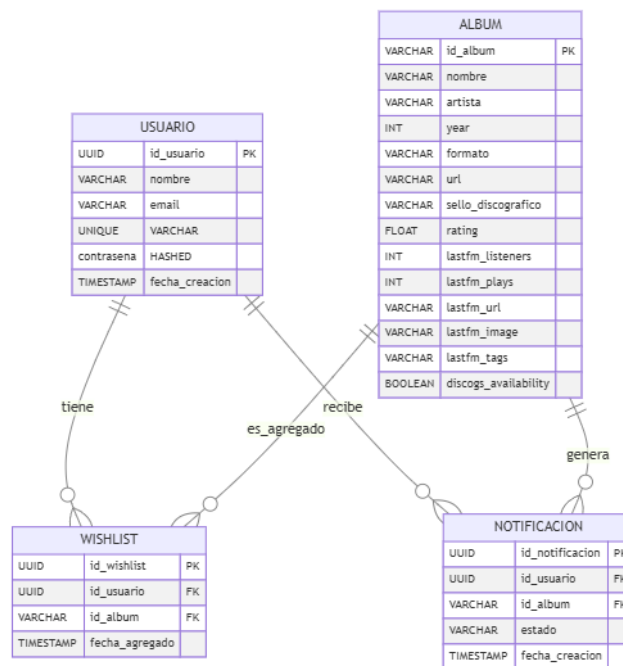
Wishlist:

- id_wishlist (UUID, PK)
- id_usuario (UUID, FK -> Usuarios)
- id_album (VARCHAR, FK -> Álbumes)
- fecha_agregado (TIMESTAMP DEFAULT CURRENT_TIMESTAMP)

Notificaciones:

- id_notificacion (UUID, PK)
- id_usuario (UUID, FK -> Usuarios)
- id_album (VARCHAR, FK -> Álbumes)
- estado (VARCHAR CHECK (estado IN ('pendiente', 'enviada')))
- fecha_creacion (TIMESTAMP DEFAULT CURRENT_TIMESTAMP)

4.3. Modelo Entidad-Relación (ERD)



4.4. Pruebas Ejecutables en MySQL

Prueba de Inserción

```

1 • INSERT INTO usuarios (id_usuario, nombre, email, contraseña)
2   VALUES ('123e4567-e89b-12d3-a456-426614174000', 'Juan Pérez', 'juan@example.com', 'hashedpassword');
3
4 INSERT INTO albums (id_album, nombre, artista, year, formato, url)
5   VALUES ('14186441', 'Abbey Road', 'The Beatles', 1969, 'Vinyl', 'https://discogs.com/14186441');

```

```

1 • SELECT * FROM musicfinder.usuarios;

```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:					
	id_usuario	nombre	email	contraseña	fecha_creacion
▶	123e4567-e89b-12d3-a456-426614174000	Juan Pérez	juan@example.com	hashedpassword	2025-03-10 16:50:42
*	NULL	NULL	NULL	NULL	NULL

Prueba de Modificación

```

1 • UPDATE usuarios
2   SET nombre = 'Juan P. Gómez'
3   WHERE id_usuario = '123e4567-e89b-12d3-a456-426614174000';

```

```

1 • SELECT * FROM musicfinder.usuarios;

```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:					
	id_usuario	nombre	email	contraseña	fecha_creacion
▶	123e4567-e89b-12d3-a456-426614174000	Juan P. Gómez	juan@example.com	hashedpassword	2025-03-10 16:50:42
*	NULL	NULL	NULL	NULL	NULL




Prueba de Eliminación

```

1 • DELETE FROM wishlist WHERE id_usuario = '123e4567-e89b-12d3-a456-426614174000';

```

1 • `SELECT * FROM musicfinder.wishlist;`

Result Grid   Filter Rows: Edit: 

	id_wishlist	id_usuario	id_album	fecha_agregado
*	NULL	NULL	NULL	NULL

5. Balsamiq Mockups

Inicio:



Pantalla inicial al iniciar sesión:



Búsqueda por artistas:

Detalles álbum:



Nombre del EP

Artista

Fecha de lanzamiento

Género

Sello Discográfico

DESCRIPCIÓN

Consultar Disponibilidad

A1 - Canción 1

A2 - Canción 2

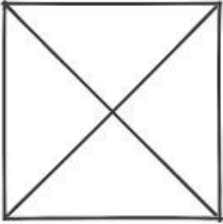
B1 - Canción 3

B2 - Canción 4

☐ Marcar como deseado

Disponibilidad

discos:



Nombre del EP

Artista

Fecha de lanzamiento

Género

Sello Discográfico

DESCRIPCIÓN

Consultar Disponibilidad

A1 - Canción 1

A2 - Canción 2

B1 - Canción 3

B2 - Canción 4

☐ Marcar como deseado

Búsqueda por canciones:

Búsqueda por Artistas	Búsqueda por Canciones	Wishlist	Notificaciones
<input type="text" value="Q Buscar por cancion..."/>		<button>Buscar</button>	
Canción 1		<button>Álbum</button>	<div>^</div> <div></div> <div>v</div>
Canción 2		<button>Álbum</button>	
Canción 3		<button>Álbum</button>	

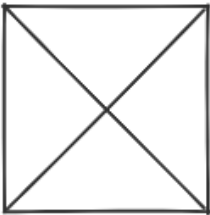
Wishlist:

Búsqueda por Artistas	Búsqueda por Canciones	Wishlist	Notificaciones
Álbum 1		<button>Ver Detalles</button>	♥
Álbum 2		<button>Ver Detalles</button>	♥
Álbum 3		<button>Ver Detalles</button>	♥
Álbum 4		<button>Ver Detalles</button>	♥

Notificaciones:

Búsqueda por Artistas	Búsqueda por Canciones	Wishlist	Notificaciones
Descripción	Fecha	Información	Visto
Nuevo álbum de Daft Punk	2024-03-10	"Random Access Memories"	<input checked="" type="checkbox"/>
Nueva recomendación	2024-03-07	"OK Computer"	<input type="checkbox"/>
Alerta de disponibilidad de disco	2024-03-06	Disponibilidad en Discogs	<input checked="" type="checkbox"/>

Usuario:



Usuario: **nickname**

Nombre: **Nombre**

Apellidos: **Apellidos**

Correo: **Correo**

Teléfono: **Teléfono**

Géneros favoritos:

Artistas favoritos:

Ir a wishlist