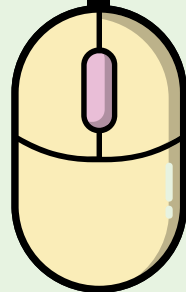


# Music Hub API

Diseño e implementación



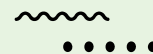
Integrantes: Xavier Mora, Guillermo W, Marta Canino, Dulibeth Medina





»»»»

# AGENDA

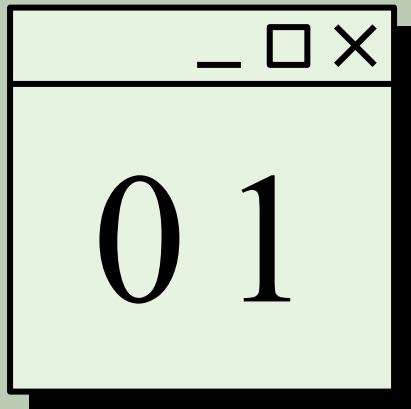


0 1 Introducción

0 2 Base de datos

0 3 API REST

0 4 Implementación  
cliente





# Introducción



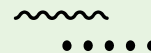
## Objetivo

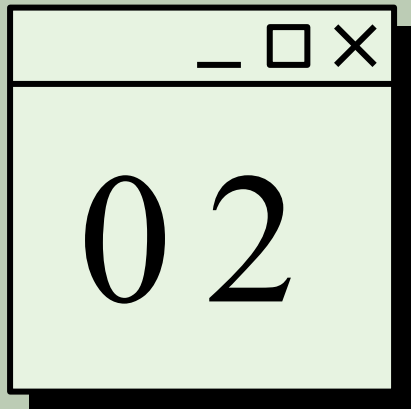
- ✓ Desarrollar una API, llamada **MusicHub**, para la gestión y consulta de datos musicales utilizando MongoDB.



## Descripción

- ✓ **MusicHub** proporciona una plataforma centralizada para administrar y consultar información detallada sobre música.
- ✓ La API ofrece funcionalidades avanzadas como **paginación y filtrado** para manejar eficientemente **colecciones grandes**, asegurando un acceso rápido y ordenado a los datos.





# Base de datos



## mongoDB®

- ✓ Para este proyecto, se utiliza **MongoDB** como sistema de gestión de bases de datos NoSQL.
- ✓ Se utilizó un **dataset** que contenía alrededor de 10,000 registros sobre música. Este conjunto se transformó en un formato JSON y se utilizó para cargar inicialmente la base de datos MongoDB.



```
{
  "track_uri": "spotify:track:1XAZlnVtthcDZt2NI1Dtxo",
  "track_name": "Justified & Ancient - Stand by the Jams",
  "artist_uris": [
    "spotify:artist:6dYrdR1NZSKaVxYg5IrvCH"
  ],
  "artist_names": [
    "The KLF"
  ],
  "album_uri": "spotify:album:4MC0ZjNtVP1nDD5lsLxFjc",
  "album_name": "Songs Collection",
  "album_artist_uris": [
    "spotify:artist:6dYrdR1NZSKaVxYg5IrvCH"
  ],
  "album_artist_names": [
    "The KLF"
  ],
  "album_release_date": "1992-08-03",
  "album_image_url": "https://i.scdn.co/image/ab67616d0000b27355346bc1f268730f607f9544"
},
```

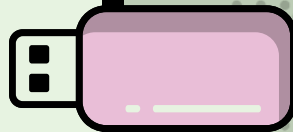
03

.....

>>>>

~~~~~

API REST



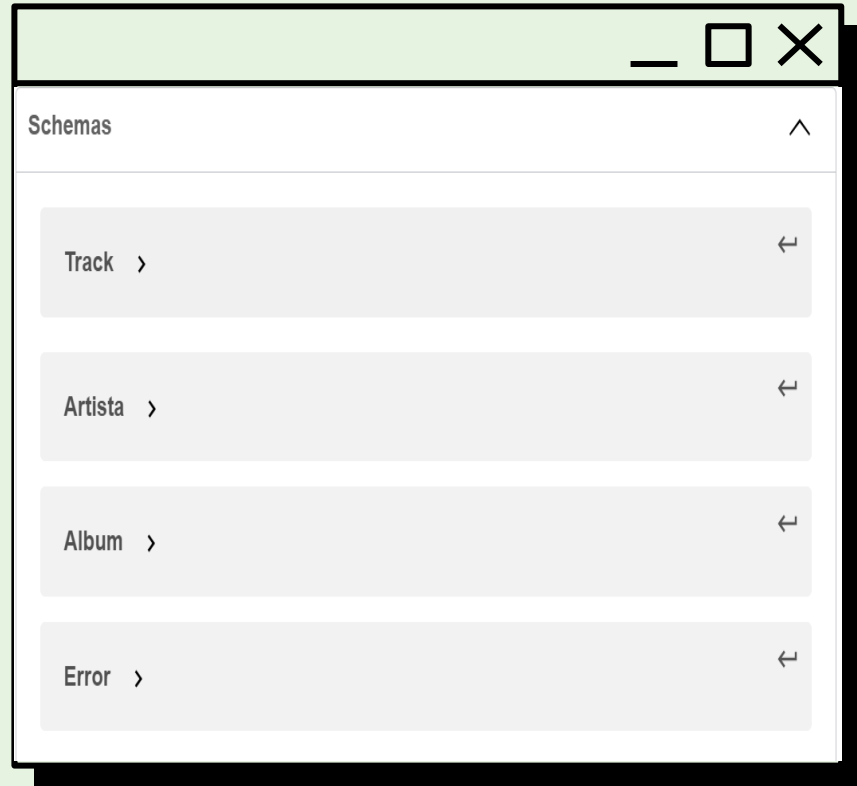
# API Music Hub

La API incluye los siguientes RECURSOS principales:

- ✓ **/albums:** Permite acceder a la información de los álbumes disponibles.
- ✓ **/artists:** Proporciona detalles sobre los artistas de música.
- ✓ **/tracks:** Ofrece información sobre las canciones disponibles en la plataforma.

.....

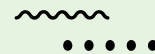
>>>>>







# API Music Hub (II)

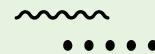


Las operaciones **CRUD** disponibles: Crear, leer, actualizar y eliminar álbumes, artistas y canciones.

|        |                     |                                          |   |
|--------|---------------------|------------------------------------------|---|
| GET    | /tracks             | Obtener una lista de tracks.             | ▼ |
| POST   | /tracks             | Crear un nuevo track.                    | ▼ |
| GET    | /tracks/{track_uri} | Obtener detalles de un track específico. | ▼ |
| PUT    | /tracks/{track_uri} | Actualizar un track específico.          | ▼ |
| DELETE | /tracks/{track_uri} | Eliminar un track específico.            | ▼ |



# API Music Hub (III)

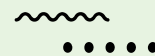


Las operaciones **CRUD** disponibles: Crear, leer, actualizar y eliminar álbumes, artistas y canciones.

|        |                       |                                            |   |
|--------|-----------------------|--------------------------------------------|---|
| GET    | /artists              | Obtener una lista de artists.              | ▼ |
| POST   | /artists              | Crear un nuevo artista.                    | ▼ |
| GET    | /artists/{artist_uri} | Obtener detalles de un artista específico. | ▼ |
| PUT    | /artists/{artist_uri} | Actualizar un artista específico.          | ▼ |
| DELETE | /artists/{artist_uri} | Eliminar un artista específico.            | ▼ |



# API Music Hub (IV)



Las operaciones **CRUD** disponibles: Crear, leer, actualizar y eliminar álbumes, artistas y canciones.

|        |                     |                                          |   |
|--------|---------------------|------------------------------------------|---|
| GET    | /albums             | Obtener una lista de álbumes.            | ▼ |
| POST   | /albums             | Crear un nuevo álbum.                    | ▼ |
| GET    | /albums/{album_uri} | Obtener detalles de un álbum específico. | ▼ |
| PUT    | /albums/{album_uri} | Actualizar un álbum específico.          | ▼ |
| DELETE | /albums/{album_uri} | Eliminar un álbum específico.            | ▼ |

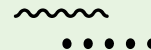
# APIs Externas



- Nuestra API también consume información de tres diferentes API externas:

- OpenAI
- Spotify API
- MusicBrainz

La información que se consume está completamente integrada con el resto de la API, además, nuestra API es resiliente a caídas de las APIs externas



>>>>

# APIs Externas

~~~~~  
.....



OpenAi  
API

Es una herramienta que permite a los desarrolladores integrar y utilizar los modelos de lenguaje avanzados.

```
$ curl -s -u :$OPENAI_API_KEY -H 'Content-Type: application/json' https://api.openai.com/v1/completions -d '{
  > "model": "davinci",
  > "temperature": 0,
  > "stop": "\n",
  > "prompt": "Q: What is human life expectancy in the United States?
\nA: Human life expectancy in the United States is 78 years.\n\nQ: What is the meaning of life?\n"
  > }' | jq
{
  "id": "cmpl-RRcXqSMfadADL1tst9gxPTJ9",
  "object": "text_completion",
  "created": 1591865181,
  "model": "davinci:2020-05-03",
  "choices": [
    {
      "text": "A: The meaning of life is 42.",
      "index": 0,
      "logprobs": null,
      "finish_reason": "stop"
    }
  ]
}
```

>>>>

# APIs Externas

~~~~~  
.....



## Spotify API

Permite obtener datos de Spotify de música, en formato **json**

```
1 {  
2   ...  
3   is_playable: true  
4   linked_from: {  
5     external_urls: {  
6       spotify: "https://open.spotify.com/track/6kLCHFM39w  
7     },  
8     href: "https://api.spotify.com/v1/tracks/6kLCHFM39wkF  
9     id: "6kLCHFM39wkFjOuyPGLGeQ",  
10    type: "track",  
11    uri: "spotify:track:6kLCHFM39wkFjOuyPGLGeQ"  
12  },  
13  ...  
14  uri: "spotify:track:6ozxp1TAjWO0B1UxN8ia0A"  
15 }
```

>>>>

# APIs Externas

~~~~~  
.....



MusicBrainz

Enciclopedia de música abierta,  
proporciona metadatos musicales en  
formato **XML**

```
<?xml version="1.0" encoding="UTF-8"?>

<metadata xmlns="http://musicbrainz.org/ns/mm
  <artist id="c0b2500e-0cef-4130-869d-732b2
    <name>Tori Amos</name>
    <sort-name>Amos, Tori</sort-name>
    <life-span begin="1963-08-22"/>
    <release-list>
      <!-- releases omitted -->
    </release-list>
  </artist>
</metadata>
```

03

.....

>>>>>

~~~~~

# Implementación cliente







>>>>

## Interfaz de cliente

También se ha implementado, con **node.js**, una **interfaz** sencilla a través de la cual los clientes pueden:

- ☐ Interactuar con nuestra API
- ☐ Realizar operaciones sobre nuestra base de datos
- ☐ Solicitar y obtener información de las APIs externas



Meet us



# Interfaz Cliente



Music Hub

localhost:3000/api/v1/



MusicHub

Inicio

Álbumes

Canciones

Chat

Spotify

MusicBrainz

## Bienvenido a MusicHub

Tu plataforma definitiva para explorar, descubrir y disfrutar de la música. Accede a álbumes, canciones, y más. Interactúa con nuestros servicios y experimenta la mejor experiencia musical.

Explorar Álbumes

Explorar Canciones

# Interfaz Cliente (II)



Las operaciones CRUD también se pueden llevar a cabo en la interfaz a través de formularios:

## (GET) Barra de Búsqueda

Buscar álbum por URI:

Buscar

## (DELETE) Eliminar Álbum

ID del Álbum a Eliminar:

Eliminar Álbum

## (PUT)

URI actual del Álbum:

Nombre del Álbum:

Nueva URI del Álbum:

Nombres de los Artistas:

Fecha de Lanzamiento:

URL de la Imagen del Álbum:

put

## (POST) Añadir Álbum

Nombre del Álbum:

URI del Álbum:

Nombres de los Artistas:

Fecha de Lanzamiento:

URL de la Imagen del Álbum:

Añadir Álbum

# Interfaz Cliente (III)



Se tienen interfaces para las tres APIs externas

## Chat

Type your message...

Send

## Buscador de canciones de Spotify

Introduce el nombre de la canción:

Buscar

## Obtén información de MusicBrainz de un artista

Introduce el nombre del artista:

Buscar

