

Back End**1.1 Agregar Playlist**

- Descripción: Agregar una playlist a la base de datos
- End point: localhost:8080/playlists/agregarPlaylist
- Método: POST
- Body:

```
{
  "id": 3,
  "nombre": "Gym",
  "creador": "Carlos"
}
```

- Validaciones:

Campo	Validación
id	debe ser un número válido
nombre	debe ser un string entre 5 y 30 caracteres
creador	debe ser un string entre 5 y 30 caracteres

- Errores:

Code	Message	HTTP
InvalidTokenException	Id no es válido	401
InvalidBodyException	El nombre debe tener entre 5 y 50 caracteres	422
InvalidBodyException	El creador debe tener entre 5 y 50 caracteres	422
Forbidden		403

- Respuesta:
HTTP status 200 y mensaje: registro exitoso

1.2 Agregar Canción

- Descripción: Agregar una canción a la playlist
- End point: localhost:8080/playlists/agregarCancion
- Método: POST
- Body

```
{
  "id": 6,
  "nombre": "Monster",
  "artista": "Skillet",
  "año": 2021
}
```

- Validaciones:

Campo	Validación
id	debe ser un número válido
nombre	debe ser un string entre 5 y 50 caracteres
artista	debe ser un string entre 5 y 50 caracteres
año	debe ser un número entre 1990 y 2021

- Errores:

Code	Message	HTTP
InvalidTokenException	Id no es válido	401
InvalidBodyException	El nombre debe tener entre 5 y 50 caracteres	422
InvalidBodyException	El artista debe tener entre 5 y 50 caracteres	422
InvalidBodyException	El año debe ser un número entre 1990 y 2021	422
Forbidden		403

- Respuesta:
HTTP status 200 y mensaje: registro exitoso

1.3 Agregar Libro

- Descripción: Agregar un libro a la playlist
- End point: localhost:8080/playlists/agregarLibro
- Método: POST
- Body:

```
{
  "id":5,
  "nombre":"The Lord of the Ring",
  "autor":"John R. R. Martin",
  "genero":"Fantasia",
  "año":1954
}
```

- Validaciones:

Campo	Validación
id	debe ser un número válido
nombre	debe ser un string entre 5 y 50 caracteres
autor	debe ser un string entre 10 y 50 caracteres
género	debe ser un string entre 5 y 20 caracteres
año	debe ser un número menor a 2021

- Errores:

Code	Message	HTTP
InvalidTokenException	Id no es válido	401
InvalidBodyException	El nombre debe tener entre 5 y 50 caracteres	422
InvalidBodyException	El autor debe tener entre 10 y 50 caracteres	422
InvalidBodyException	El género debe tener entre 5 y 20 caracteres	422
InvalidBodyException	El año debe ser un número menor a 2021	422
Forbidden		403

- Respuesta:
HTTP status 200 y mensaje: registro exitoso

1.4 Agregar PlaylistLibroCancion

- Descripción: Agregar un playlist_libro_canción, (tabla intermedia para evitar relacion muchos a muchos)
- End point: localhost:8080/playlists/agregarPlaylistLibroCancion
- Método: POST
- Body:

```
{
  "idPLC":4,
  "fecha":"30/10/2021",
  "libroId":4,
  "playlistId":1
}
```

- Validaciones:

Campo	Validación
idPLC	debe ser un número válido
fecha	debe ser un string de formato dd/mm/aaaa
libroId y/o cancionId	debe ser un id válido de libro y/o canción
playlistId	debe ser un id válido de playlist

- Errores:

Code	Message	HTTP
InvalidTokenException	Id no es válido	401
InvalidBodyException	La fecha no tiene formato dd/mm/aaaa	422
InvalidBodyException	El id foráneo no existe o es erroneo	422
Forbidden		403

- Respuesta:
HTTP status 200 y mensaje: registro exitoso

2.1 Obtener Playlist

- Descripción: Poder sacar o mostrar la playlist que se desea
- End point: localhost:8080/playlists/obtenerPlaylists
- Método: GET
- Validaciones:

Campo	Validación
id	debe ser un número válido

- Errores:

Code	Message	HTTP
InvalidTokenException	Id no es válido	401
UserNotFoundException	Playlist no encontrada	500

- Respuesta:

```
{
  "id": 1,
  "nombre": "My Music",
  "creador": "Carlos"
}
```

2.2 Obtener Canción

- Descripción: Poder sacar o mostrar la canción que se desea
- End point: localhost:8080/playlists/obtenerCancion
- Método: GET
- Validaciones:

Campo	Validación
id	debe ser un número válido

- Errores:

Code	Message	HTTP
InvalidTokenException	Id no es válido	401
UserNotFoundException	Canción no encontrada	500

- Respuesta:

```
{
  "id": 1,
  "nombre": "Loose yourself",
  "artista": "Eminem",
  "año": 2004
}
```

2.3 Obtener Libro

- Descripción: Poder sacar o mostrar el libro que se desea
- End point: localhost:8080/playlists/obtenerLibro
- Método: GET
- Validaciones:

Campo	Validación
id	debe ser un número válido

- Errores:

Code	Message	HTTP
InvalidTokenException	Id no es válido	401
UserNotFoundException	Libro no encontrado	500

- Respuesta:

```
{
  "id": 3,
  "nombre": "The Lord of the Ring",
  "autor": "John R. R. Tolkien",
  "genero": "Fantasía",
  "año": 1954
}
```

2.4 Obtener *PlaylistLibroCancion*

- Descripción: Poder sacar o mostrar el *playlistLibroCancion* que se desea
- End point: localhost:8080/playlists/obtenerPlaylistLibroCancion
- Método: GET
- Validaciones:

Campo	Validación
id	debe ser un número válido

- Errores:

Code	Message	HTTP
InvalidTokenException	Id no es válido	401
UserNotFoundException	PlaylistLibroCancion no encontrada	500

- Respuesta:

```
{
  "idPLC": 3,
  "fecha": "28/10/2021",
  "cancionId": null,
  "playlistId": 2,
  "libroId": 3
}
```


3.1 Borrar Playlist

- Descripción: Borrar una playlist dentro de la base de datos
- End point: localhost:8080/playlists/borrarPlaylist
- Método: DELETE
- Validaciones:

Campo	Validación
id	debe ser un número válido

- Errores:

Code	Message	HTTP
UserNotFoundException	Playlist no encontrada	500

- Respuesta: HTTP status 200 y mensaje: playlist borrada

3.2 Borrar Canción

- Descripción: Borrar una canción dentro de la base de datos
- End point: /borrarCancion
- Método: DELETE
- Validaciones:

Campo	Validación
id	debe ser un número válido

- Errores:

Code	Message	HTTP
UserNotFoundException	Usuario no encontrado	500

- Respuesta: HTTP status 200 y mensaje: canción borrada

3.3 Borrar Libro

- Descripción: Borrar un libro dentro de la base de datos
- End point: localhost:8080/playlists/borrarLibro
- Método: DELETE
- Validaciones:

Campo	Validación
id	debe ser un número válido

- Errores:

Code	Message	HTTP
UserNotFoundException	Libro no encontrado	500

- Respuesta: HTTP status 200 y mensaje: libro borrado

3.4 Borrar PlaylistLibroCanción

- Descripción: Borrar una PlaylistLibroCancion dentro de la base de datos
- End point: localhost:8080/playlists/borrarPlaylistLibroCancion
- Método: DELETE
- Validaciones:

Campo	Validación
id	debe ser un número válido

- Errores:

Code	Message	HTTP
UserNotFoundException	PlaylistLibroCancion no encontrada	500

- Respuesta: HTTP status 200 y mensaje: playlistLibroCancion borrada

4.1 Actualizar Playlist

- Descripción: Actualizar el nombre de una playlist seleccionada
- End point: localhost:8080/playlists/actualizarPlaylist
- Método: PUT
- Body:

```
{
  ... "id":3,
  ... "nombre":"C&L"
}
```

- Validaciones:

Campo	Validación
id	debe ser un número válido
nombre	debe ser un string entre 2 y 25 caracteres

- Errores:

Code	Message	HTTP
AliasExistsException		500

- Respuesta: HTTP status 200 y mensaje: playlist actualizado

4.2 Actualizar Canción

- Descripción: Actualizar al artista de una canción seleccionada
- End point: /actualizarCancion
- Método: PUT
- Body:

```
...  "id":3,  
...  "artista":"Gera-MX"
```

- Validaciones:

Campo	Validación
id	debe ser un número válido
artista	debe ser un string entre 5 y 50 caracteres

- Errores:

Code	Message	HTTP
AliasExistsException		500

- Respuesta: HTTP status 200 y mensaje: canción actualizada.

4.3 Actualizar Libro

- Descripción: Actualizar el año de un libro seleccionado
- End point: /actualizarLibro
- Método: PUT
- Body:

```
{
  "id":1,
  "año":1997
}
```

- Validaciones:

Campo	Validación
id	debe ser un número válido
año	debe ser un número menor a 2021

- Errores:

Code	Message	HTTP
AliasExistsException		500

- Respuesta: HTTP status 200

4.3 Actualizar PlaylistLibroCancion

- Descripción: Actualizar la fecha de una PlaylistLibroCancion seleccionada
- End point: /actualizarLibro
- Método: PUT
- Body:

```
{
  ... "id": 3,
  ... "fecha": "02/11/2021"
}
```

- Validaciones:

Campo	Validación
id	debe ser un número válido
fecha	debe tener formato dd/mm/aaaa

- Errores:

Code	Message	HTTP
AliasExistsException		500

- Respuesta: HTTP status 200