

```
1 import spotipy → para conectar con la api de spotify
2 from spotipy.oauth2 import SpotifyClientCredentials
```

para autenticacion con credenciales ←

Credenciales sacadas del dashboard en la pagina de la api

```
6 # credenciales de la API
7 app_id = '0b8cd645c7284c1aa9519857fbccd9ca'
8 secret_id = 'a8c395d102c74bf9b7915f18ed2dfc8d'
```

se crea el manejador de la autenticacion con las credenciales

```
10 # inicializar la API con autenticación
11 auth_manager = SpotifyClientCredentials(client_id=app_id, client_secret=secret_id)
12 sp = spotipy.Spotify(auth_manager=auth_manager)
```

en la variable auth_manager guardamos la instancia de la autenticacion con las credenciales

→ se crea el objeto sp con el que haremos las llamadas a la api

```
12 # retorna las 10 canciones mas populares del artista
13 def get_top_tracks(artist_name):
```

funcion que retornara el top 10, se pasa el nombre del artista como parametro

```
14     if not artist_name:
15         return None, "Ingresa el nombre de un artista"
```

si no se le pasa el nombre del artista retornara ese mensaje

dentro de un bloque try se ejecuta la logica de la peticion a la api, esto para poder manejar excepciones en caso de que ocurra una

```
17 ~ try:
18     result = sp.search(q=artist_name, type='artist')
19 ~     if not result['artists']['items']:
20         return None, "Artista no encontrado"
```

se hace la busqueda del artista y se guarda en la variable result, con toda la info del artista

si no se encuentra, devuelve ese mensaje

de la lista de resultados toma al primer artista que coincide con la busqueda

ahora la variable artist contiene algo asi

```
22 # selecciona el primer resultado de la búsqueda de artista
23 artist = result['artists']['items'][0]
24 artist_id = artist['id'] obtenemos el id del artista
25 artist_name = artist['name'] # obtener el nombre exacto del artista
```

el nombre exacto para que no importe si lo escribimos mal en la busqueda

```
{
  "id": "1dfeR4HwDbWqFHLkxsgId",
  "name": "Queen",
  "popularity": 85,
  "genres": ["rock"],
  "followers": { "total": 20000000 }
},
{
  "id": "3EhbVgyfGd7HkpsagwL9GS",
  "name": "Queen + Adam Lambert",
  "popularity": 60
}
```

obtenemos el top del artista utilizando el id que antes conseguimos de la peticion

```
27 # obtener el top del artista
28 top_tracks = sp.artist_top_tracks(artist_id)
29
30 # lista con las 10 mejores canciones
31 songs = [track['name'] for track in top_tracks['tracks'][:10]]
32 return artist_name, songs
33
34 except Exception as e: en caso de alguna excepcion la
35     return None, str(e) retorna
```

en songs guarda las 10 primeras canciones en orden de popularidad

retorna el nombre del artista y el top 10

¿Como haría si quiero traer más información del artista aparte de su top 10 (seguidores, año o más características)?

```
23 artist = result['artists']['items'][0]
```

Dentro de artist, el JSON devuelto por Spotify tiene otros datos útiles:

- artist['followers']['total'] → Cantidad de seguidores.
- artist['genres'] → Géneros musicales asociados.
- artist['popularity'] → Popularidad (0-100).
- artist['images'] → Lista con imágenes del artista.

Así se podría conseguir esta otra información:

```
artist_id = artist['id']  
artist_name = artist['name']  
followers = artist['followers']['total']  
genres = artist['genres']  
popularity = artist['popularity']
```

¿Cómo conseguimos las credenciales?

En la página del API de Spotify, después de registrar una cuenta, dentro del dashboard en la opción “Crear una aplicación”, ponemos el nombre de la app y una descripción, después de creada se asigna el client id y el secret id.

¿Paradigma de programación?

Estructurada ya que sigue un flujo de control claro y secuencial.