



Feedback del Módulo 1 Bootcamp Data Analytics

¡Hola Equipo 2! 🎉

Antes que nada, me gustaría felicitaros por el esfuerzo y la dedicación que habéis puesto a lo largo de este proyecto. A continuación, podéis ver el feedback de vuestro código, así como algunos comentarios más generales.

FEEDBACK POR ÁREAS

1. Extracción de Datos

A destacar:

- ✅ Se realiza una extracción muy completa desde dos fuentes distintas (Spotify y LastFM), lo que permite enriquecer los datos con popularidad y género musical.
- ✅ Se utilizan funciones personalizadas como `extraccion_por_paises` y `filtro_de_datos`, que facilitan la reutilización del código.
- ✅ Se controla la ejecución del scraping de géneros mediante `try/except`, lo cual es un buen ejemplo de manejo básico de errores.

Áreas de mejora:

- ◆ **Modularización:** Las funciones podrían dividirse aún más para separar responsabilidades (por ejemplo, una función para procesar artistas, otra para construir el DataFrame, etc.). Esto permitiría escalar el proyecto de forma más controlada.
 - ◆ **Manejo de errores:** Aunque se usa `try/except` para la API de LastFM, no se contempla el manejo de errores en la extracción desde Spotify. Incluir `try-except` también en esas llamadas mejoraría la robustez del script.
 - ◆ **Seguridad:** Las credenciales de ambas APIs (`CLIENT_ID`, `API_KEY`) están “visibles” en el código. Como buena práctica, deberían almacenarse en variables de entorno o en un archivo `.env` (aunque entendemos que esto aún no se ha visto en clase).
-

2. Limpieza y Transformación de Datos

A destacar:

- ✅ Se crea una función específica (`eliminar_nan`) para adaptar los datos a los requisitos de MySQL, convirtiendo `NaN` en `None`, lo cual es clave para evitar errores de inserción.
- ✅ Se realiza un ordenamiento por popularidad y se filtran las 50 canciones más populares por zona y periodo, lo cual muestra un buen criterio de curación de datos.
- ✅ Se controla la duplicidad de canciones y artistas mediante `set()` y comprobaciones antes de insertar.



Feedback del Módulo 1 Bootcamp Data Analytics

Áreas de mejora:

- ◆ **Validación:** Aunque se eliminan duplicados en canciones/artistas, no hay validaciones para datos incorrectos o inconsistentes antes de la inserción. Incorporar validaciones por tipo de dato o longitud de campos ayudaría a prevenir errores.
 - ◆ **Automatización:** Se podría automatizar aún más la carga de CSVs y generación de `DataFrames` para evitar definir cada uno manualmente, por ejemplo, con bucles.
 - ◆ **Transformaciones adicionales:** Algunos nombres de artista incluyen varios nombres separados por comas (por ejemplo, "Artista A, Artista B"). Normalizar esto podría mejorar la integridad en las relaciones con la tabla `artistas`.
-

3. Diseño de la Base de Datos

A destacar:

- ✓ Se estructura la base de datos en tablas bien diferenciadas (`artistas`, `años`, `zonas`, `países`, `canciones`, y sus correspondientes tablas intermedias), lo que demuestra una muy buena comprensión del modelo relacional.
- ✓ Se definen de forma correcta las claves primarias y foráneas, incluyendo opciones de `ON DELETE CASCADE` y `ON UPDATE CASCADE` que garantizan una buena integridad referencial.
- ✓ Se han utilizado claves compuestas (`PRIMARY KEY (id_pais, id_cancion)` y similares) en las tablas intermedias, lo cual es una práctica recomendada en relaciones muchos a muchos.

Áreas de mejora:

- ◆ **Tipado de campos:** Aunque `VARCHAR(250)` es suficientemente amplio para la mayoría de nombres de canciones, podría ser interesante revisar si algunos campos pueden tener un tamaño más optimizado o si algunos podrían beneficiarse de validaciones adicionales (por ejemplo, `CHECK` para popularidad entre 0 y 100).
 - ◆ **Estandarización de nombres:** Algunas tablas tienen nombres con la estructura "entidad_en_entidad" (`pais_en_cancion`), mientras que otras siguen otro criterio como `lustro_de_cancion`. Homogeneizar estos nombres podría mejorar la legibilidad del modelo.
-



Feedback del Módulo 1

Bootcamp Data Analytics

4. Queries

A destacar:

- ✅ Las consultas están bien formuladas y permiten hacer comparaciones tanto a nivel de país como a nivel de zona, lo que da pie a un análisis detallado de popularidad musical por regiones.
- ✅ Se utilizan correctamente subconsultas, cláusulas `IN`, `DISTINCT`, `ORDER BY` y `LIMIT` para segmentar y priorizar resultados.
- ✅ Las queries están bien orientadas a responder preguntas concretas y relevantes dentro del contexto del proyecto.

Áreas de mejora:

- ◆ **Eficiencia:** En lugar de repetir múltiples consultas similares por país, sería útil parametrizar o agrupar estas consultas utilizando estructuras más dinámicas (por ejemplo, usando `GROUP BY` para rankings por país o zona).
 - ◆ **Claridad:** Algunas queries podrían beneficiarse de alias (`AS`) para mejorar la lectura de los resultados y de comentarios inline que expliquen brevemente la lógica detrás.
-

5. Documentación, Comentarios y Conclusiones

A destacar:

- ✅ A lo largo del código hay comentarios explicativos que facilitan la comprensión de las funciones y del flujo general del proyecto.
- ✅ Se incluyen mensajes `print()` y descripciones para orientar sobre lo que hace cada bloque, lo cual ayuda especialmente al momento de revisar o reutilizar el código.
- ✅ Se indica claramente qué valores corresponden a qué zonas y rangos de años, lo que facilita el uso de funciones y la conexión con los datos.

Áreas de mejora:

- ◆ **Comentarios en SQL:** Las consultas SQL podrían tener más explicaciones inline sobre qué se busca con cada una, especialmente en la parte de comparación entre países.
 - ◆ **Conclusiones:** No se incluye una sección final donde se analicen los resultados obtenidos a través de las queries. Añadir una síntesis o visualización de los datos permitiría obtener insights más accionables.
 - ◆ **Estructura narrativa:** Añadir una breve introducción y conclusiones por zonas o décadas musicales ayudaría a contextualizar el análisis y darle un cierre más sólido al proyecto.
-



Feedback del Módulo 1 Bootcamp Data Analytics

COMENTARIOS GENERALES

Habéis diseñado un proyecto ambicioso y bien ejecutado, con múltiples capas de complejidad: extracción multifuente, procesamiento, diseño relacional, inserción eficiente y análisis SQL. Con un poco más de automatización, estructura narrativa y documentación, este trabajo podría escalar perfectamente a un proyecto profesional.

 **¡Gran trabajo, equipo!**  