



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Grado en Matemáticas e Informática

Trabajo Fin de Grado

**Análisis de Tendencias Musicales
Globales: Extracción y Visualización de
Datos de Spotify con AWS y Power BI**

Autor: Jesús Fernández López

Tutor(a): Alejandro Rodríguez González

Madrid, enero 2024

Agradecimientos

A mis padres por todo el apoyo brindado,
mi hermano y su familia, mis 3 sobrinos
y mi compañera durante estos años.

Gracias por todo

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Matemáticas e Informática

Título: Análisis de Tendencias Musicales Globales: Extracción y
 Visualización de Datos de Spotify con AWS y Power BI.

Enero 2024

Autor: Jesús Fernández López

Tutor:

Alejandro Rodríguez González

Lenguajes y Sistemas Informáticos e Ingeniería de Software Informáticos
e Ingeniería del Software

ETSI Informáticos

Universidad Politécnica de Madrid

Resumen

El trabajo consiste en el desarrollo de un sistema automatizado el cual se encarga de recopilar, almacenar, transformar y mostrar datos musicales sobre las diez canciones más escuchadas de Spotify de diez países elegidos (España, Italia, Alemania, Francia, Reino Unido, Estados Unidos, Canadá, Argentina, Brasil y México). El objetivo principal es presentar estos datos de manera accesible y significativa al usuario.

Primeramente, para seleccionar los países a estudiar se han tenido en cuenta diversos criterios como la zona geográfica, población, cultura y actividad musical en Spotify. Así, los datos utilizados proporcionarán informes más representativos y completos al usuario.

Los datos recopilados se obtienen de la plataforma Spotify® utilizando su API web oficial (Application Programming Interface). Estos datos se almacenan en la nube de Amazon Web Services® (AWS), concretamente en una base de datos Amazon DynamoDB. Para realizar este proceso de extracción, almacenamiento y transformación de datos se utiliza el servicio AWS Lambda, que permite ejecutar código en la nube sin necesidad de un servidor, y Amazon S3 (Simple Storage Services) que proporciona servicio de almacenamiento en la nube.

La presentación de los datos se realiza a través Power BI, una herramienta de Microsoft® que permite representar gráficamente la información almacenada, lo que facilita la interpretación de los datos al usuario.

En resumen, el proyecto busca crear un sistema automatizado que integre tecnologías actuales como los servicios en la nube de AWS, la API web de Spotify y Power BI con el fin de permitir a los usuarios visualizar los datos de manera intuitiva y utilizarlos según sus necesidades.

Abstract

The job involves developing an automated system that is responsible for collecting, storing, transforming, and displaying musical data about the top ten most played songs on Spotify in ten selected countries (Spain, Italy, Germany, France, the United Kingdom, the United States, Canada, Argentina, Brazil, and Mexico). The main goal is to present this data in an accessible and meaningful way to the user.

Firstly, in order to select the countries to study, various criteria such as geographic location, population, culture, and musical activity on Spotify have been taken into account. This data provides the user with more representative and comprehensive reports.

The collected data is extracted from the Spotify® platform using its official web API. This data is stored in the Amazon Web Services® (AWS) cloud, specifically in an Amazon DynamoDB database. To perform this process of data extraction, storage, and transformation, AWS Lambda service is used, allowing code execution in the cloud without the need for a server, and Amazon S3 (Simple Storage Services) provide cloud storage service.

The data presentation is carried out through Power BI, a Microsoft® tool that allows for graphical representation of the stored information, making it easier for the user to interpret the data.

In summary, the project aims to create an automated system that integrates current technologies such as AWS cloud services, Spotify web API, and Power BI in order to enable users to intuitively visualize the data and use it according to their needs.

Tabla de contenidos

1	Introducción	1
2	Motivación y background.....	3
3	Desarrollo	4
3.1	Diseño de la arquitectura	4
3.2	Etapas de desarrollo.....	7
3.2.1	Introducción a Amazon Web Services y configuración de Spotify for Developers.....	7
3.2.1.1	Amazon Web Services	7
3.2.1.2	Spotify for Developers	11
3.2.2	Extracción de datos	14
3.2.2.2	Spotify Web API.....	14
3.2.2.3	Función Lambda “spotify_extract”, Spotipy y Boto3.....	15
3.2.3	Almacenamiento y tratamiento de datos	19
3.2.3.2	Amazon DynamoDB.....	19
3.2.3.3	Amazon S3 (Simple Storage Service).....	22
3.2.3.4	Funciones Lambda “extract_DB_to_S3” y “export_jsonS3_to_URL”	23
3.2.4	Automatización y Monitorización	23
3.2.4.2	Amazon EventBridge	23
3.2.4.3	Amazon CloudWatch	27
3.2.5	Visualización de datos y diseño.....	28
4	Manual de uso	34
5	Resultados y conclusiones	39
6	Análisis de Impacto	41
7	Bibliografía	42
8	Anexos	44

1 Introducción

El proyecto trata de la creación de un sistema automatizado de recopilación, tratamiento y visualización de datos sobre las diez canciones más populares en Spotify® de diez países (España, Italia, Alemania, Francia, Reino Unido, Estados Unidos, Canadá, Argentina, Brasil y México), con el fin de que los datos estén disponibles en un formato accesible e intuitivo, lo que permitirá a los usuarios utilizarlos de manera efectiva según sus necesidades y preferencias.

Para la implementación del proyecto, se han utilizado diversos servicios de computación en la nube, en particular los ofrecidos por Amazon Web Services® (AWS). Estos servicios posibilitan la automatización de las tareas necesarias para llevar a cabo el proyecto como son la extracción, almacenamiento, transformación y visualización de los datos.

Los datos se extraen mediante la API oficial de Spotify®. Para utilizar esta herramienta se creará una cuenta de Spotify y una App asociada a esta cuenta, para así obtener las credenciales necesarias y poder realizar las llamadas necesarias a la API. De cada canción, se extraerán cinco parámetros musicales calculados por Spotify: tempo, danceability (bailabilidad), acousticness (acústica), energy (energía) y valence (valencia). Además, se registrará un id asociado a cada elemento de la tabla, el nombre de la canción, el país del ranking al que corresponde cada canción y el nombre del artista.

Para llevar a cabo el almacenamiento de los datos de Spotify se utilizará Amazon DynamoDB, un servicio de base de datos NoSQL que se adaptará perfectamente a la estructura de los datos recopilados. Todos estos datos se extraerán utilizando una función Lambda (servicio AWS Lambda) que se implementará en lenguaje Python, utilizando librerías como Boto3 para la conexión con los servicios de Amazon Web Services®, y Spotipy¹ para la conexión con la API de Spotify®. Esta función permite la automatización del proceso, garantizando que la base de datos se mantenga siempre actualizada, ya que se ejecutará de manera programada mediante la creación de un disparador Cloudwatch Event, (las ejecuciones se automatizarán para que se realicen todos los lunes entre las 00:00 y las 00:15). Esta arquitectura asegura que el usuario siempre disponga de información actualizada.

La presentación y visualización de los datos se llevará a cabo a través de la herramienta Power BI de Microsoft®. Power BI permite crear informes y paneles interactivos que facilitan la interpretación de los datos. Funciona conectándose a diversos orígenes para obtener datos. Para facilitar la automatización de este proceso, se realizarán distintos pasos:

En primer lugar, una Función Lambda exportará los datos de la tabla DynamoDB (Amazon Web Services®) hacia un S3 Bucket (Simple Storage Service, Amazon Web Services®) en formato JSON. Otra Función Lambda se encarga de obtener este JSON y generar una URL con los datos del archivo en el mismo formato. Finalmente, desde Power BI (Microsoft®) se importarán los datos utilizando como fuente de datos la URL asociada a la función Lambda.

Además, con esta herramienta se transformarán los datos exportados de DynamoDB (Amazon Web Services®) para facilitar su manipulación en la propia herramienta.

¹ Biblioteca de Python que proporciona una interfaz para interactuar con la API de Spotify.

En resumen, Power BI (Microsoft®) simplificará la visualización y presentación de los datos, brindando a los usuarios una forma intuitiva de explorar, comprender y comparar toda la información musical generada por el sistema mediante el diseño de un informe interactivo para el usuario.

2 Motivación y background

La música, a lo largo de la historia, ha sido un lenguaje universal que trasciende las barreras culturales y geográficas. En la actualidad, la revolución tecnológica ha dado lugar a una era dorada de la música, marcada por la accesibilidad a una gran diversidad musical a través de plataformas de streaming como Spotify®, Amazon Music®, SoundCloud®...

La motivación detrás de este proyecto radica en la importancia de comprender las tendencias musicales de la época. La música, aparte de ser un medio de entretenimiento, también es un indicador cultural y social que refleja estados de ánimo, tendencias y cambios en la sociedad. Por tanto, analizar y comprender las preferencias musicales actuales es crucial tanto para los profesionales de la música como para la industria musical, que busca adaptarse a un mercado en constante cambio debido a la fugacidad de las tendencias de hoy en día.

Este proyecto abarca tres campos principales que son las nuevas tecnologías, la música y los datos, los cuales representan grandes pilares en mi vida. En primer lugar, he nacido y crecido escuchando todo tipo de música en gran parte gracias a mi familia y su gran pasión por ella. Además, he pasado diez años de mi vida estudiando la enseñanza elemental y profesional de Oboe en el Conservatorio Profesional Tomas de Torrejón y Velasco de Albacete, donde me formé como músico y como persona junto a grandes amigos músicos.

Tras esta etapa, decidí mudarme a Madrid a estudiar esta carrera y lo que más me llamó la atención desde el principio fueron todas las nuevas tecnologías que me enseñaron y que fui conociendo durante los cuatro años, hasta que en el segundo año descubrí que mi área de mayor interés era todo lo relacionado con los datos y sus diversas aplicaciones.

Por todo esto, elegí este proyecto como Trabajo Final de Grado para completar mi etapa como estudiante.

3 Desarrollo

En este capítulo se detallarán todas las tareas realizadas y decisiones tomadas en cada etapa del desarrollo del proyecto. Además, se indicarán las tecnologías involucradas en cada etapa del desarrollo y se explicará la decisión de su uso.

3.1 Diseño de la arquitectura

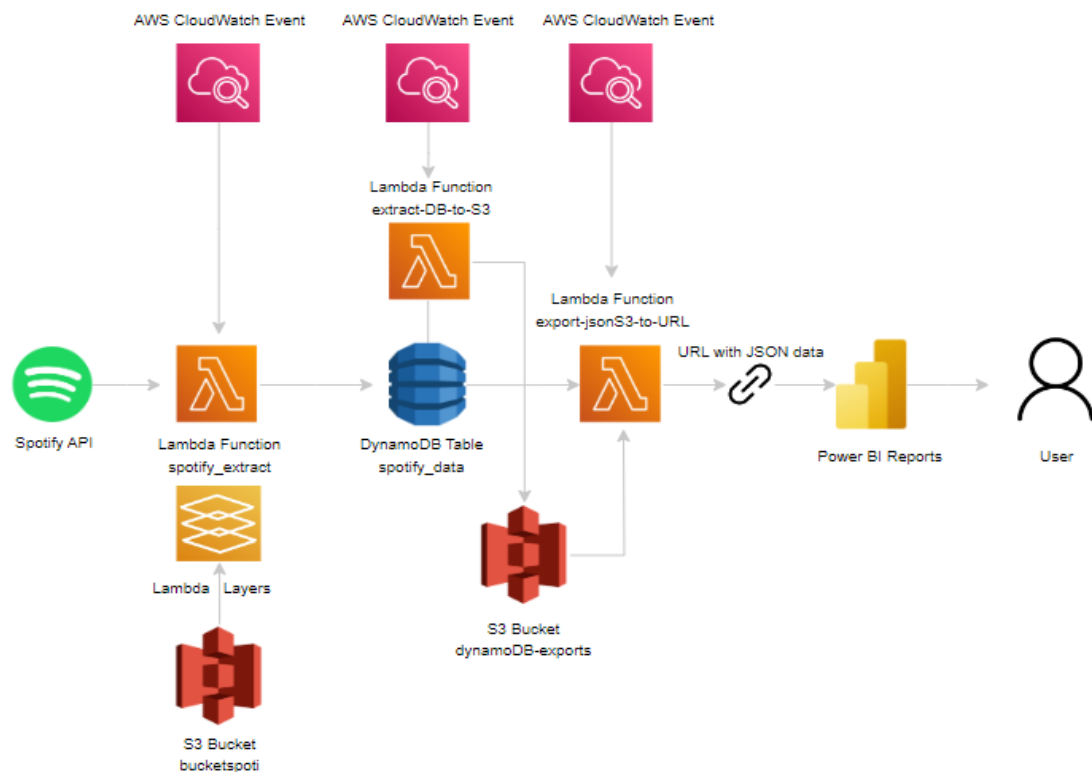


Ilustración 1: Arquitectura del proyecto

En esta ilustración se puede observar un esquema diagrama de la arquitectura general del proyecto. Las principales fases de esta arquitectura son las siguientes:

1. Extracción de datos

Para la extracción de datos se utilizarán la API de Spotify como fuente de datos y la Función Lambda “spotify_extract” para realizar las llamadas necesarias a esta API para su posterior extracción y almacenamiento.

- **Spotify API:** Este servicio ofrecido por Spotify permite acceder a los datos y las funcionalidades del servicio de música en streaming desde

otras aplicaciones [1] (en este caso, sólo se hará uso del acceso a datos).

Para hacer uso de ella se creará una cuenta de Spotify en <https://www.spotify.com/es/signup>, y seguidamente se accederá a la cuenta de desarrollo asociada en <https://developer.spotify.com/>. Dentro de la cuenta de Spotify for Developers se creará una “app” que nos proporcionará el Client ID y el Client Secret necesarios para solicitar un token de acceso. Estos tres valores deben incluirse en las llamadas realizadas a la API para así poder obtener los datos necesarios. Las llamadas a la API se realizarán de manera simplificada utilizando la librería de Python Spotipy la cual facilita toda la interacción entre el sistema y Spotify.

- **Función Lambda “spotify_extract”:** AWS Lambda es un servicio informático sin servidor y basado en eventos que le permite ejecutar código para prácticamente cualquier tipo de aplicación o servicio backend sin necesidad de aprovisionar o administrar servidores [2]. Esta función se encargará de diversas tareas. Por una parte, realizará las llamadas a la API necesarias utilizando los métodos proporcionados por la librería Spotipy. Después, se creará una lista con todos los datos obtenidos en las llamadas realizadas en la que cada elemento de la lista está formado por un diccionario que contiene los siguientes pares clave-valor: id, canción, artista, playlist, tempo, danceability, acousticness, energy y valence. Seguidamente, utilizando la librería Boto3, se añadirán todos los elementos de la lista a una base de datos Amazon DynamoDB. Esta función se ejecutará en Python 3.8 y tendrá una capa Lambda (archivo ZIP que contiene bibliotecas, un tiempo de ejecución personalizado u otras dependencias [3]) para garantizar el funcionamiento correcto de la función.

2. Almacenamiento

Para almacenar todos los datos obtenidos y archivos generados se utilizarán los servicios DynamoDB y S3 de AWS. Además, se utilizará la función explicada en el punto anterior para el guardado de datos en DynamoDB.

- **Tabla Amazon DynamoDB:** Amazon DynamoDB es una base de datos NoSQL en la que no hay que predefinir ningún tipo de dato o atributos, sin servidor y completamente administrada. Además, su capacidad de escalabilidad automática y su flexibilidad en el modelo de datos se adaptan perfectamente a las necesidades de almacenamiento, y las herramientas de monitoreo y métricas facilitan la supervisión constante del rendimiento de la tabla, permitiendo ajustar la capacidad de almacenamiento según sea necesario [4]. En este contexto se configurará una tabla de DynamoDB (spotify_data) para almacenar los datos obtenidos en la fase anterior de manera eficiente.
- **Amazon S3 Buckets:** Amazon S3 (Simple Storage Service) es un servicio de almacenamiento en la nube altamente escalable y versátil ofrecido por. S3 se utilizará con un doble propósito: en primer lugar, se creará un “bucket” para almacenar los datos exportados desde la tabla

DynamoDB en formato JSON (spotibucket), lo que proporciona un medio eficiente y seguro para conservar los datos recopilados. En segundo lugar, otro “bucket” se creará para alojar las bibliotecas y recursos necesarios para la Función Lambda encargada de la extracción de datos desde la API de Spotify. Esto garantiza un acceso rápido y confiable a las dependencias requeridas por la función Lambda, lo que resulta fundamental para su correcto funcionamiento. En resumen, Amazon S3 es una pieza fundamental de la infraestructura de este proyecto, contribuyendo tanto al almacenamiento de datos como a la disponibilidad de recursos esenciales para la ejecución de código [5].

3. Tratamiento de datos

- **Función Lambda “extract-DB-to-S3”:** Esta función lambda es la encargada de exportar los datos almacenados en la tabla de DynamoDB “spotify_data”, convertirlos a formato JSON y almacenarlos en el S3 bucket “dynamodb-export”. Para ello creará un archivo JSON el cual contiene los datos en el formato final y se identificará con la fecha exacta en la que se generó para así tener un registro de todos los archivos generados y las fechas de cada uno. Finalmente utilizando la librería Boto3 se guardará el archivo en el bucket para realizar la siguiente fase.
- **Función Lambda “export-jsonS3-to-URL”:** Esta función lambda se encargará de obtener el archivo JSON más reciente del bucket “dynamodb-export” para así proporcionar los datos más actualizados a Power BI. Para ello, se configurará una URL asociada a la función la cual contendrá los datos actualizados en formato JSON. Esta URL actuará como origen de datos para Power BI y desde ella se importarán los datos para poder generar los informes pertinentes.
- **Power BI:** Tras importar los datos desde la URL asociada a la función “export-jsonS3-to-URL” se realizará una última transformación de los datos con Power BI utilizando el editor de Power Query en Power BI, para finalmente poder generar los informes necesarios y trabajar con los datos en un formato óptimo.

4. Visualización

Para la visualización de los datos se hará uso de la herramienta Power BI de Microsoft ®, la cual permite conectarse con los datos, crear informes básicos y atractivos y compartirlos con los usuarios [6]. Estos informes serán diseñados para proporcionar una visión general e intuitiva de los datos con algunas acciones interactivas como el filtrado de datos por país, canción, artista...

5. Automatización y monitorización

Una de las ideas fundamentales de la implementación del proyecto es la automatización de todas las tareas posibles. Para ello se utilizará el servicio Amazon EventBridge de AWS que permitirá la monitorización y automatización de tareas. En concreto se implementarán tres reglas programadas que se asociarán cada una a cada función lambda. Estas reglas se agregarán como desencadenadores (combinación de una distribución de

CloudFront, un comportamiento de la caché y un evento que provoca la ejecución de una función) a las funciones y se encargarán de que las funciones se ejecuten todos los lunes desde las 00:00 a las 00:15. Por otro lado, se podrán monitorizar todas las ejecuciones de las funciones mediante CloudWatch para así poder revisar los registros de las ejecuciones pasadas y asegurarse del correcto funcionamiento del sistema, o en caso contrario, poder visualizar los errores experimentados y corregirlos [7].

3.2 Etapas de desarrollo

En este apartado se definen las distintas etapas en las que consiste el desarrollo del proyecto en orden cronológico y se detallan las tareas llevadas a cabo en cada etapa.

3.2.1 Introducción a Amazon Web Services y configuración de Spotify for Developers

En esta sección se realiza una introducción a Amazon Web Services (AWS), se definen los servicios de la nube de AWS utilizados en el proyecto y se documenta el proceso realizado para la creación de la aplicación de Spotify.

3.2.1.1 Amazon Web Services

Este proyecto se realiza con una cuenta de AWS Academy que ofrece cursos y recursos de aprendizaje que permiten a los estudiantes desarrollar diferentes habilidades relacionadas con la nube de AWS [8]. De la misma manera se podría realizar con una cuenta personal de Amazon Web Services y su capa gratuita, pero existe la posibilidad de que la capa gratuita no cubra los gastos del proyecto por lo que podría experimentar cobros mensuales en su método de pago por parte de AWS.

Con la cuenta de AWS Academy se cubren todas las necesidades del proyecto sin ningún cobro ya que el fin del proyecto es académico y por ello se hace uso de este recurso de AWS.

Para utilizar los servicios de AWS mediante AWS Academy se utilizará un laboratorio. Este laboratorio proporciona un entorno de pruebas para la exploración ad hoc de los servicios de AWS.

Se trata de un entorno de larga duración. La sesión termina cuando el temporizador llegue a 0:00, pero se conservan los datos y recursos que se hayan creado en la cuenta de AWS. Si más adelante se inicia una sesión nueva (por ejemplo, el día siguiente), el trabajo sigue en el entorno del laboratorio.

En la siguiente ilustración se muestra la pantalla una vez se ha iniciado sesión en AWS Academy y se ha cargado el laboratorio.

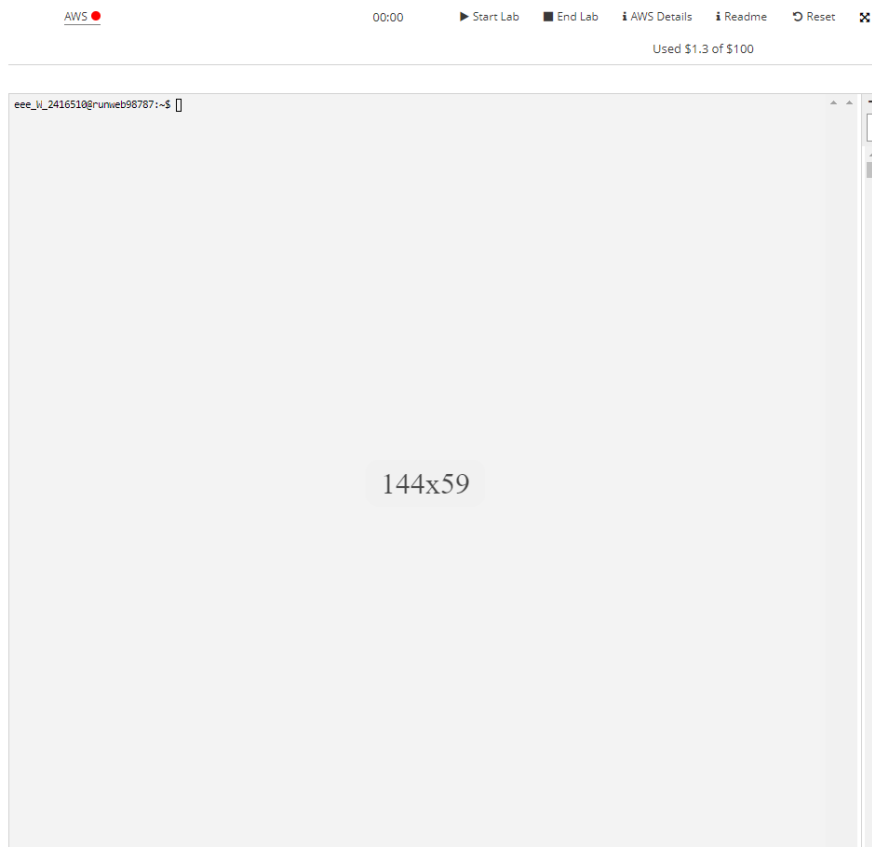


Ilustración 2: AWS Academy Learner Lab apagado

Para iniciar el laboratorio se hace click en “Start Lab”. Así, el temporizador cambia a 04:00 lo que significa que se disponen de cuatro horas de uso del laboratorio. Para aumentar el tiempo de uso basta con volver a hacer click en “Start Lab” y el tiempo restante volverá a ser de cuatro horas.

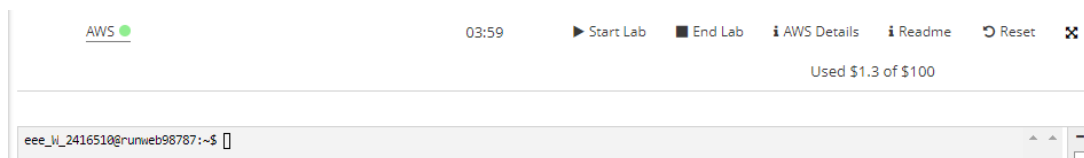


Ilustración 3: AWS Academy Learner Lab iniciado

Una vez se ha iniciado el laboratorio, en el apartado “AWS Details” se encuentran las credenciales de acceso a la nube. Estas credenciales se utilizarán para autenticar y autorizar el acceso a los servicios de AWS desde scripts o herramientas que interactúan con la nube de AWS. Además, son esenciales para la seguridad de los recursos y datos en AWS.

Un factor a tener en cuenta es que las credenciales se modifican cada vez que se inicia el laboratorio. Así, cada vez que se realiza esta acción se debe actualizar la configuración de credenciales en las herramientas necesarias.

AWS

03:52

Start Lab

End Lab

AWS Details

Readme

Reset

Used \$1.3 of \$100

eee_w_2416510@runweb98787:~\$

Cloud Access

AWS CLI:

Copy and paste the following into ~/.aws/credentials

```
[default]
aws_access_key_id=ASIAVL5EQ5XQ8MBHMF3
aws_secret_access_key=PI59HwOFwUqFh6HtFQ16qVukqTTa+mqdEbTQTQE
aws_session_token=FwoGZXIvYXdzEEoaDE9yYv6XXw2XhAGgRCK5AbHrX9/ZIxtkk
7KkHmoovzVbhiACTkKfsmPTbs/+V8uVEx4+k/k95Z11rHsM0GvxPyndr4E5X2sdofa8F
zbXvAD4fSp1gSZ+o3URmzsVq6KPyYR85BxEFP/PDFZzXH8tRoNu5AtGX0R2Bo0SHFER
1nY4Z4Fkz8cr1dKoJXASrHzgzxHOGNw8CtVU1V7wHBfJgxnMwAQEsTY7fgIUqJMsst1
Olm79/7gEaM+kpEMUC/QwiREgAePBaZS1tKM2C5akGMi2TcQE3Hzx5qVoTSPJXXj0B5
Ai0VYI17J0ZE26BI7UV65gd0V4wYQe4pdNtZc=
```

Cloud Labs

Remaining session time: 03:56:25(237 minutes)

Session started at: 2023-10-25T09:24:44-0700

Session to end at: 2023-10-25T13:27:02-0700

Accumulated lab time: 2 days 08:45:00 (3405 minutes)

No running instance

SSH key

Show

Download PEM

Download PPK

AWS SSO

Download URL

AWSAccountid

369175358945

Region

us-east-1

Ilustración 4: Credenciales de acceso a AWS

Este proyecto se basa en la implementación de un programa informático en la nube de AWS, plataforma la cual cumple con todos los requisitos necesarios para que el programa funcione correctamente y todos los servicios utilizados se alojen en la nube. Esta plataforma de servicios en la nube creada por Amazon ofrece una gran variedad de servicios, desde almacenamiento de bases de datos hasta la creación de aplicaciones de machine learning y recursos para profesionales de DevOps. Entre todos los servicios ofrecidos por AWS, para la implementación del proyecto se han hecho uso de los siguientes:

- **Bases de datos:** AWS ofrece una variedad de servicios de bases de datos para satisfacer las necesidades de diferentes aplicaciones y cargas de trabajo. Algunas de ellas son Amazon RDS, Amazon Neptune, Amazon Aurora y Amazon DynamoDB. Para este proyecto se ha elegido el servicio Amazon DynamoDB ya que ofrece escalabilidad automática, flexibilidad en cuanto al modelo de datos, seguridad y control de acceso y facilidad para integrar otros servicios de AWS [3].
- **Almacenamiento:** AWS ofrece diversos servicios de almacenamiento en la nube como Amazon S3, Amazon EFS, Amazon FSx, Amazon Storage Gateway... En este contexto se ha decidido utilizar Amazon S3 (Simple Storage Service) ya que es un servicio que permite almacenar una gran cantidad de datos, con una alta durabilidad y disponibilidad, y con una

interfaz y una consola que facilita la interacción con otros servicios como Funciones Lambda o tablas DynamoDB [5].

- **Computación sin servidor:** Para la computación en la nube AWS ofrece diversas opciones. Entre ellas se encuentran AWS Lambda, Amazon EC2, AWS Batch y Amazon App Runner. Para la ejecución del código implementado en el proyecto se ha utilizado el servicio AWS Lambda, el cual permite ejecutar código sin aprovisionar ni administrar servidores. Además, este servicio se ocupa de todo lo necesario para que el código se ejecute y permite configurar el código para que se ejecute automáticamente [9].
- **Automatización:** AWS ofrece servicios de automatización que permiten administrar recursos de manera eficiente en la nube y simplificar tareas. Algunos de los principales servicios de automatización son AWS CloudFormation, AWS Step Functions y Amazon EventBridge. Este último servicio es el elegido para la automatización de las tareas del proyecto ya que ofrece facilidades para integrarlo con funciones como definir fácilmente reglas que describan qué eventos deben desencadenar las funciones Lambda, y poder mantener un registro de todas las reglas en un solo lugar. Esto simplifica la gestión de eventos y garantiza la coherencia en la automatización. [10]
- **Monitorización:** AWS ofrece una gama de servicios de monitoreo y administración que permiten supervisar, administrar y optimizar los recursos en la nube de manera efectiva. Algunos de estos servicios son AWS CloudTrail, Amazon X-Ray, Amazon Managed Grafana y Amazon CloudWatch. Debido a su sencilla integración con Amazon Eventbridge y con todos los servicios utilizados se ha elegido Amazon CloudWatch. Este servicio recopila y visualiza los registros, métricas y datos de evento en tiempo real para simplificar el mantenimiento de aplicaciones [11] [12].

En conclusión, Amazon Web Services (AWS) es una plataforma de servicios en la nube ofrecida por Amazon que ha transformado la forma en que las empresas y los individuos gestionan sus recursos informáticos. AWS proporciona una amplia gama de servicios, incluyendo capacidad de cómputo, almacenamiento, bases de datos, análisis, inteligencia artificial, redes, Internet de las cosas (IoT) y mucho más. Esta plataforma escalable y flexible ha revolucionado la forma en que las organizaciones implementan, escalan y gestionan sus aplicaciones y servicios en línea. AWS ofrece varias ventajas, como la capacidad de reducir costos, mejorar la agilidad, aumentar la seguridad y acceder a una amplia gama de servicios, lo que permite a las empresas centrarse en su innovación en lugar de preocuparse por la infraestructura. Esta infraestructura en la nube ha habilitado una amplia gama de aplicaciones y soluciones tecnológicas que se benefician de la escalabilidad y confiabilidad de AWS, lo que la convierte en una de las principales plataformas de servicios en la nube [13].

3.2.1.2 Spotify for Developers

En primer lugar, se necesita crear una cuenta de Spotify para poder acceder a sus servicios ya que es la fuente de donde se extraen los datos para el proyecto.

Para ello se introduce un correo y contraseña en la web oficial de Spotify, o se realiza el registro con una cuenta de Facebook o Google:

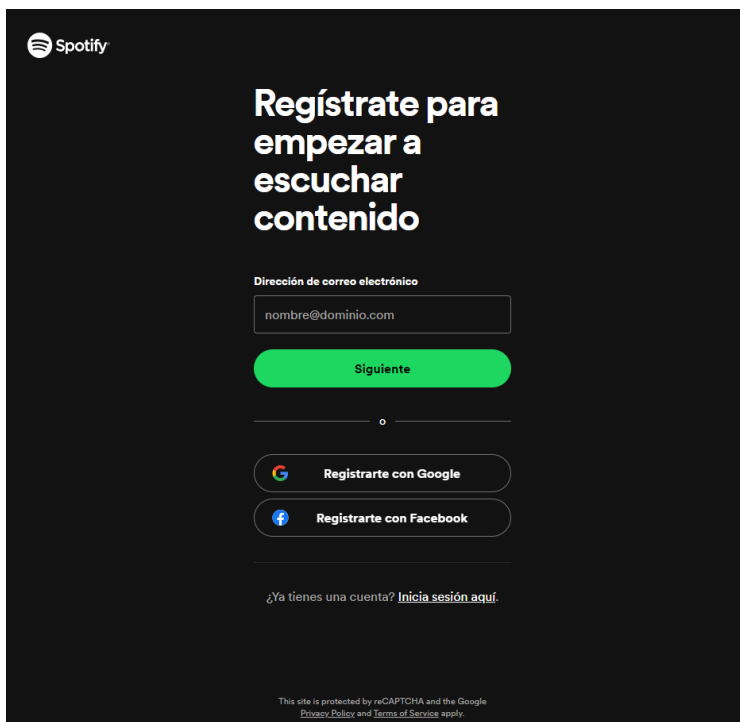


Ilustración 5: Pantalla login de Spotify

Una vez se haya creado correctamente la cuenta de Spotify, el siguiente paso de la configuración de Spotify es iniciar sesión en la web de Spotify for Developers con las mismas credenciales de la cuenta anterior de Spotify ya que la cuenta de Developers está asociada a la cuenta de Spotify.

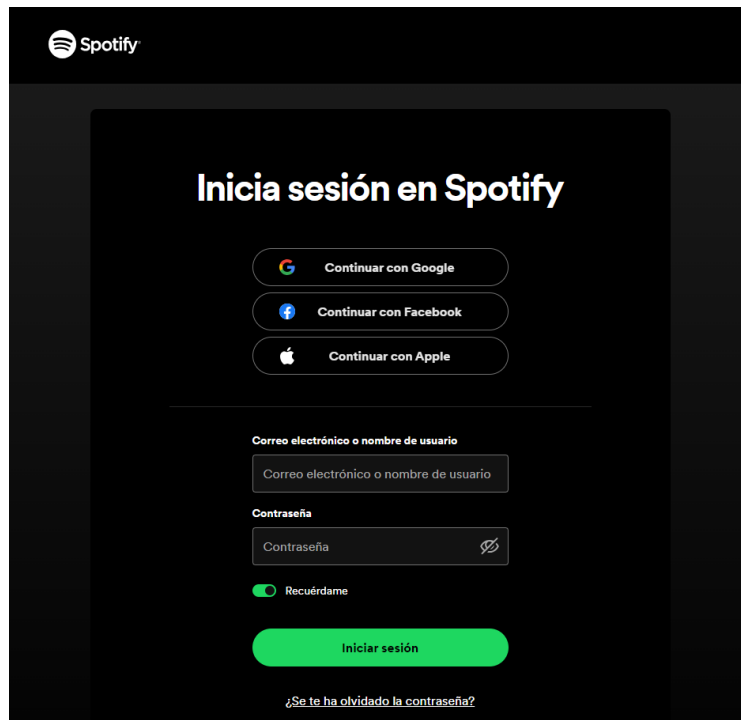


Ilustración 6: Pantalla login de Spotify for Developers

Tras acceder a la cuenta, en el Dashboard del perfil se creará una App de Spotify for Developers haciendo click en el botón “Create app”. Para este proceso hay que rellenar los campos descritos en este paso, pero en este proyecto solo es necesario especificar el nombre de la App, la descripción y la URI, ya que realmente no se está creando una aplicación web o móvil, si no un sistema automatizado y desplegado en la nube en el que no se necesita un sitio web asociado a la aplicación.

Spotify for Developers Documentation Community Jesus Fernandez Lopez

Dashboard > Create app

Create app

App name

TFG

App description

Esta aplicación se utilizará para implementar el uso de la API de Spotify en un trabajo de carácter académico. Se extraerán datos de Spotify mediante su API con el fin de realizar un análisis de tendencias musicales en distintos países

Website

Redirect URI

http://localhost:8080

A URI where users can be redirected after authentication success or failure

Which API/SDKs are you planning to use?

☒ Web API [Read more about Web API](#)

☐ Web Playback SDK [Read more about Web Playback SDK](#)

☐ Android [Read more about Android](#)

☐ Ads API [Read more about Ads API](#)

☐ iOS [Read more about iOS](#)

☐ I understand and agree with Spotify's [Developer Terms of Service](#) and [Design Guidelines](#)

Save Cancel

Ilustración 7: Pantalla de creación de una App de Spotify for Developers

El fin de crear esta App es obtener las credenciales necesarias para poder utilizar la API de Spotify. Los campos que se necesitan son Client ID y Client Secret, los cuales se encuentran en la pantalla de configuración de la aplicación creada en el anterior paso. Estos valores se utilizarán más adelante en las llamadas a la API.

Spotify for Developers Documentation Community Jesus Fernandez Lopez

Dashboard > TFG > Settings > Basic Information

T Basic Information

Basic Information User Management Extension Requests

Client ID

d62b07fe27474a81bde1e05064ab6f2e

App Status

Development mode

Client secret

[Masked]

[Hide client secret](#) [Rotate client secret](#)

Ilustración 8: Credenciales de la App de Spotify for Developers

Además de proporcionar estas credenciales, la creación de la App también permite visualizar algunos parámetros asociados a la App como los usuarios activos que la están utilizando, su localización y las peticiones realizadas a la API agrupadas por endpoints que se están realizando.

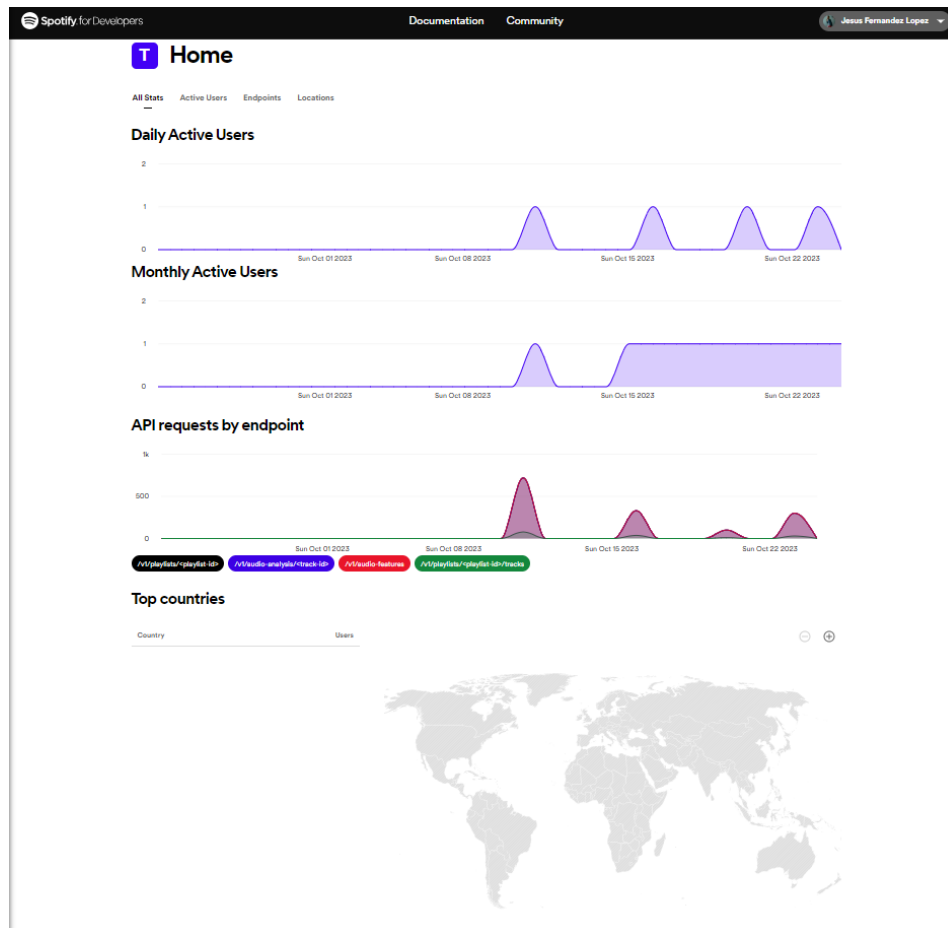


Ilustración 9: Pantalla de monitorización de la actividad de la App de Spotify for Developers

3.2.2 Extracción de datos

En esta segunda fase se explica detalladamente el proceso seguido para la recolección de los datos utilizados en el proyecto.

3.2.2.2 Spotify Web API

La API Web de Spotify es el origen de donde se obtienen los datos para el proyecto y actúa como fuente primaria de información.

Se trata de una interfaz de programación de aplicaciones que permite a los desarrolladores acceder a los datos y funciones del servicio de música en streaming de Spotify. Funciona siguiendo estos pasos básicos:

1. Autenticación: Para utilizar la API de Spotify, debe autenticarse utilizando credenciales válidas de Spotify. Esto se hace mediante el flujo de autorización OAuth, que permite a la aplicación obtener un token de acceso válido.
2. Solicitudes HTTP: Una vez que se ha autenticado la aplicación, puede realizar solicitudes HTTP a las URL de la API de Spotify para acceder a los datos. Las solicitudes pueden ser de varios tipos, como GET para

obtener información o POST para modificar datos, según la acción que se quiera llevar a cabo. En este caso sólo se hará uso de las solicitudes GET.

3. Endpoints y Recursos: La API de Spotify ofrece una serie de endpoints y recursos que representan diferentes tipos de datos, como listas de reproducción, canciones, álbumes, artistas, etc. Cada endpoint corresponde a un tipo de acción o consulta específica.
4. Respuestas JSON: Cuando se realiza una solicitud a la API, esta responde con datos en formato JSON. Los datos devueltos pueden incluir detalles sobre canciones, álbumes, listas de reproducción, información de artistas y más.
5. Uso de Tokens de Acceso: Para garantizar la seguridad y la autorización, se utiliza un token de acceso en las solicitudes. Este token se incluye en el encabezado de la solicitud HTTP para demostrar que la aplicación tiene permiso para acceder a los datos.
6. Límites de Uso: La API de Spotify suele imponer límites de uso en cuanto a la cantidad de solicitudes que se pueden realizar en un período de tiempo determinado. Estos límites varían según el tipo de cuenta de desarrollador y el nivel de acceso.

La API de Spotify es una forma estructurada de interactuar con los datos y funciones de Spotify. Se puede utilizar para crear aplicaciones, servicios y herramientas que aprovechan la base de datos de música y metadatos de Spotify para una variedad de propósitos, como la creación de aplicaciones de recomendación, análisis de tendencias musicales, creación de listas de reproducción personalizadas y mucho más [1].

3.2.2.3 Función Lambda “spotify_extract”, Spotipy y Boto3

Para llevar a cabo la tarea de extracción de datos se implementa una Función AWS Lambda llamada `spotify_extract` haciendo uso del servicio AWS Lambda. Este servicio permite ejecutar el código de la función en la nube sin necesidad de un servidor y facilita la integración con otros servicios de AWS [2].

Esta función se implementa en Python 3.8, pero existen otras opciones compatibles con AWS Lambda como Java, Node.js, Go, Ruby, .NET...

Lambda > Funciones > Crear una función

Crear una función Información

Las aplicaciones de Repositorio de aplicaciones sin servidor de AWS se han trasladado a [Crear una aplicación](#).

☒ Crear desde cero
Empiece con un sencillo ejemplo "Hello World".

☐ Utilizar un proyecto
Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.

☐ Imagen del contenedor
Seleccione una imagen de contenedor para implementar para la función.

Información básica

Nombre de la función
Escriba un nombre para describir el propósito de la función.

spotify_extractt

Utilice exclusivamente letras, números, guiones o guiones bajos. No incluya espacios.

Tiempo de ejecución Información
Elija el lenguaje que desea utilizar para escribir la función. Tenga en cuenta que el editor de código de la consola solo admite Node.js, Python y Ruby.

Python 3.8

Arquitectura Información
Elija la arquitectura del conjunto de instrucciones que desea para el código de la función.

☒ x86_64
☐ arm64

Permisos Información
De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar este rol predeterminado más adelante al agregar los disparadores.

► Cambiar el rol de ejecución predeterminado

Ilustración 10: Creación de una función Lambda

Además, esta función se apoya en dos librerías fundamentales:

- **Spotipy:** Biblioteca de Python que se utiliza para interactuar con la API de Spotify de manera más sencilla y conveniente. Ofrece una serie de funciones y métodos que facilitan la autenticación, la realización de solicitudes a la API de Spotify y el procesamiento de las respuestas. Algunas de las ventajas de su uso son:
 1. Autenticación simplificada: Spotipy simplifica el proceso de autenticación OAuth 2.0 con Spotify. Maneja la obtención y renovación de tokens de acceso, lo que facilita la autenticación de una aplicación con la API de Spotify.
 2. Acceso a la API de Spotify: Spotipy proporciona una interfaz en Python para acceder a la API de Spotify y realizar solicitudes HTTP para obtener datos, como información sobre canciones, álbumes, listas de reproducción...
 3. Abstracción de la API: Spotipy abstrae gran parte de la complejidad de la API de Spotify, lo que permite a los desarrolladores interactuar con ella de manera más intuitiva y orientada a objetos. Proporciona métodos para buscar canciones, artistas, álbumes...

4. Acceso a metadatos de audio: Spotipy permite acceder a metadatos detallados sobre pistas de audio.
5. Compatibilidad con Python: Al estar escrita en Python, Spotipy se integra fácilmente en proyectos y aplicaciones que utilizan este lenguaje de programación.

En resumen, Spotipy simplifica la interacción con la API de Spotify al proporcionar una interfaz de Python fácil de usar. Esto ahorra tiempo y esfuerzo para crear aplicaciones que requieren acceso a la extensa biblioteca musical y los datos relacionados de Spotify [14].

- **Boto3:** Biblioteca de Python que se utiliza para interactuar con los servicios de Amazon Web Services (AWS). Proporciona una interfaz de programación de aplicaciones (API) que permite a los desarrolladores controlar y automatizar diversos servicios y recursos de AWS utilizando código Python. Algunas de las ventajas de su uso son:

1. Acceso a servicios de AWS: Boto3 facilita la interacción con una amplia gama de servicios de AWS.
2. Automatización de tareas: Permite a los desarrolladores automatizar tareas relacionadas con AWS.
3. Gestión de recursos: Con Boto3, puedes crear, modificar y eliminar recursos de AWS directamente desde tu código Python. Esto es útil para implementar infraestructura como código (IaC) y administrar recursos de manera programática.
4. Seguridad y autenticación: Proporciona métodos para autenticar tu aplicación y gestionar las credenciales de AWS de manera segura. Puedes configurar el acceso y los permisos adecuados para interactuar con los recursos de AWS.
5. Integración con marcos y aplicaciones: Boto3 se integra con varios marcos y bibliotecas de Python, lo que facilita su uso en una amplia gama de aplicaciones y proyectos.

En resumen, Boto3 es una herramienta fundamental para trabajar con AWS y automatizar tareas, gestionar recursos y crear aplicaciones que interactúen con los servicios en la nube de Amazon. Facilita la creación de aplicaciones escalables y flexibles en la infraestructura de AWS [15].

Mediante el uso de la librería Spotipy la función realiza llamadas a la API de Spotify para extraer los siguientes datos de cada canción registrada:

- **Canción:** Nombre de la canción o pista de música.
- **Artista:** El artista es el músico o grupo que interpreta o crea la canción. Puede ser un cantante, una banda, un compositor o cualquier entidad responsable de la producción musical.
- **Playlist:** Playlist a la que pertenece la canción.
- **Tempo:** El tempo general estimado de una pista en pulsaciones por minuto (BPM). En terminología musical, el tempo es la velocidad o el ritmo de una pieza determinada y se deriva directamente de la duración promedio del tiempo [16].

- **Danceability:** La bailabilidad describe qué tan adecuada es una pista para bailar en función de una combinación de elementos musicales que incluyen el tempo, la estabilidad del ritmo, la fuerza del ritmo y la regularidad general. Un valor de 0,0 es el menosailable y 1,0 es el másailable [16].
- **Acousticness:** Una medida de confianza de 0,0 a 1,0 sobre si la pista es acústica. 1,0 representa una alta confianza en que la pista es acústica [16].
- **Energy:** La energía es una medida de 0,0 a 1,0 y representa una medida perceptiva de intensidad y actividad. Normalmente, las pistas enérgicas son rápidas y ruidosas. Por ejemplo, el death metal tiene mucha energía, mientras que un preludio de Bach obtiene una puntuación baja en la escala. Las características que contribuyen a este atributo incluyen rango dinámico, volumen percibido, timbre, velocidad de inicio y entropía general [16].
- **Valence:** Una medida de 0,0 a 1,0 que describe la positividad musical que transmite una pista. Las pistas con valencia alta suenan más positivas (p. ej., felices, alegres, eufóricas), mientras que las pistas con valencia baja suenan más negativas (p. ej., tristes, deprimidas, enfadadas) [16].

Todos estos datos son proporcionados y calculados internamente por Spotify. Para acceder a ellos se han utilizado los siguientes métodos y clases de Spotify:

- **SpotifyClientCredentials:** Clase que se utiliza para configurar las credenciales del cliente de Spotify mediante la creación de un administrador de flujo de credenciales de cliente. Se proporcionan los valores de las credenciales `client_id` y `client_secret` para autenticar la aplicación con la API de Spotify [14]. Estas credenciales se configuran como variables de entorno en la función lambda.

</

Ilustración 11: Configuración de variables de entorno de la función lambda `spotify_extract`

- **Spotify:** Se crea una instancia de la clase Spotify para interactuar con la API de Spotify. Esta instancia se configura con las credenciales del cliente de Spotify que se han establecido previamente. La instancia de Spotify se utiliza para realizar llamadas a la API y recuperar información sobre canciones y listas de reproducción [14].
- **Playlist_tracks:** Este método se utiliza para obtener una lista de las canciones de una lista de reproducción (playlist) de Spotify. Se especifica el identificador de la lista de reproducción y se limita la cantidad de canciones recuperadas a 10 en este caso para obtener las diez canciones más escuchadas de cada país [14].
- **Audio_analysis:** Este método se utiliza para obtener el análisis de audio de una canción específica. Se proporciona el identificador de la canción y devuelve información detallada sobre el tempo, la duración y otros aspectos del audio de la canción [14].
- **Audio_features:** Este método se utiliza para obtener características de audio de una o más canciones. Se proporciona una lista de identificadores de canciones, y devuelve información sobre características como la bailabilidad, la acústica, la energía y la valencia de las canciones [14].

Estas funciones, métodos y clases de Spotipy permiten interactuar con la API de Spotify para obtener los datos pertinentes y posteriormente almacenarlos.

3.2.3 Almacenamiento y tratamiento de datos

En esta fase de desarrollo se explica detalladamente el proceso de almacenamiento y tratamiento de los datos, y los servicios en la nube utilizados.

3.2.3.2 Amazon DynamoDB

Después de haber obtenido los datos de la API de Spotify, el siguiente paso es transformarlos y almacenarlos en la nube. Para ello, se decide utilizar Amazon DynamoDB.

Amazon DynamoDB es un servicio de base de datos NoSQL administrado que ofrece un rendimiento rápido y predecible, así como una perfecta escalabilidad. DynamoDB le permite delegar las cargas administrativas que supone tener que utilizar y escalar bases de datos distribuidas, para no tener que preocuparse del aprovisionamiento, la instalación ni la configuración del hardware, ni tampoco de las tareas de replicación, aplicación de parches de software o escalado de clústeres. DynamoDB también ofrece el cifrado en reposo, que elimina la carga y la complejidad operativa que conlleva la protección de información confidencial [17].

Con DynamoDB, puede crear tablas de base de datos capaces de almacenar y recuperar cualquier cantidad de datos, así como de atender cualquier nivel de tráfico de solicitudes. Puede escalar la capacidad de rendimiento de las tablas

para aumentarla o reducirla sin tiempos de inactividad ni reducción del desempeño [17].

Además, DynamoDB distribuye automáticamente los datos y el tráfico de las tablas entre un número suficiente de servidores para satisfacer sus requisitos de almacenamiento y rendimiento, al mismo tiempo que mantiene un desempeño uniforme y rápido. Todos los datos se almacenan en discos de estado sólido (SSD) y se replican automáticamente en varias zonas de disponibilidad de una región de AWS, con objeto de ofrecer prestaciones integradas de alta disponibilidad y durabilidad de los datos [17].

Para almacenar los datos en DynamoDB, lo primero es crear una tabla DynamoDB. En la configuración de esta tabla definiremos el nombre, la clave de partición, la clase de la tabla y la capacidad

Crear tabla

Detalles de la tabla [Información](#)

DynamoDB es una base de datos sin esquemas que solo requiere un nombre de tabla y una clave principal al crear la tabla.

Nombre de la tabla
Se utilizará para identificar su tabla.

Entre 3 y 255 caracteres. Solo se pueden usar letras, números, guiones bajos (_), guiones (-) y puntos (.).

Clave de partición
La clave de partición forma parte de la clave principal de la tabla. Se trata de un valor hash que se utiliza para recuperar elementos de la tabla, así como para asignar datos entre hosts por cuestiones de escalabilidad y disponibilidad.

De 1 a 255 caracteres, distingue entre mayúsculas y minúsculas.

Clave de ordenación - opcional
Puede utilizar una clave de ordenación como segunda parte de la clave principal de una tabla. La clave de ordenación le permite ordenar o buscar entre todos los elementos que comparten la misma clave de partición.

De 1 a 255 caracteres, distingue entre mayúsculas y minúsculas.

Configuración de la tabla

☐ Configuración predeterminada
La forma más rápida de crear la tabla. Puede modificar estos ajustes ahora o después de que se haya creado la tabla.

☒ Personalizar configuración
Utilice estas características avanzadas para que DynamoDB funcione mejor de acuerdo a sus necesidades.

Clase de tabla

Seleccione la clase de tabla para optimizar el costo de la tabla en función de los requisitos de la carga de trabajo y los patrones de acceso a los datos.

Elegir la clase de tabla

☒ Estándar de DynamoDB
La clase de tabla de uso general predeterminada. Se recomienda para la gran mayoría de las tablas que almacenan datos a los que se accede con frecuencia, donde el rendimiento (lecturas y escrituras) es el principal costo de la tabla.

☐ DynamoDB Standard-IA
Se recomienda para las tablas que almacenan datos a los que se accede con poca frecuencia, donde el almacenamiento es el principal costo de la tabla.

Ilustración 12: Configuración de la tabla DynamoDB “spotify_data”

Como podemos observar en la imagen, el nombre de la tabla elegido es “spotify_data”, la clave de partición definida es el campo “id” (cada entrada (canción) de la tabla tiene un id único proporcionado por la librería uuid de Python). Esta clave forma parte de la clave principal de la tabla. Se trata de un valor hash que se utiliza para recuperar elementos de la tabla, así como para asignar datos entre hosts por cuestiones de escalabilidad y disponibilidad.

La configuración se realiza de manera personalizada, con una clase estándar de DynamoDB y la capacidad de lectura/escritura bajo demanda. Se implanta este tipo de configuración para adaptarse a las necesidades del proyecto, garantizar un rendimiento predecible y simplificar la administración sin preocuparse por configuraciones manuales de capacidad. Además, la capacidad de lectura/escritura bajo demanda implica que DynamoDB ajustará automáticamente la capacidad de lectura y escritura de la tabla en función de la demanda de tráfico. No es necesario configurar una capacidad fija, ya que DynamoDB escalará automáticamente para acomodar los picos de carga.

Una vez creada la tabla “spotify_data”, la función lambda “spotify_extract” se encarga de agregar elementos a la tabla. Para ello se apoya en la librería Boto3 utilizando las siguientes funciones:

1. Resource('dynamodb'): Esta función se utiliza para crear un recurso DynamoDB que permite interactuar con las tablas de la base de datos.
2. Table(table_name): Se utiliza para crear una instancia de la tabla DynamoDB que se desea manipular (en este caso a la tabla spotify_data).
3. Put_item(Item): Se utiliza para agregar elementos (filas) a la tabla DynamoDB.

Así queda la tabla tras ejecutar la Función Lambda “spotify_extract”. Con esto se consigue tener los datos disponibles en la nube para poder acceder a ellos de manera eficiente y segura.

spotify_data Vista previa automática Ver los detalles de la tabla

▼ Escanear o consultar elementos

☒ Examen ☐ Consulta

Seleccione una tabla o un índice: Tabla - spotify_data

Seleccione la proyección de atributos: Todos los atributos

► Filtros

Ejecutar Restablecer

Elementos devueltos (100)

🔄 Acciones ▼ Crear elemento

<input type="checkbox"/>	id (Codemo)	Acousticness	Artista	cancion	Danceability	Energy	Playlist	Tempo	Valence
<input type="checkbox"/>	c06343ce-61b5-4891...	0.0908	BENNETT	Vois sur ton...	0.634	0.824	Top 50 - Ge...	137.959	0.371
<input type="checkbox"/>	46d101f9-56ba-4a3f...	0.12	AYLVA	Hälslich	0.914	0.489	Top 50 - Ge...	108.072	0.559
<input type="checkbox"/>	611128dd-f91d-4842...	0.345	Taylor Swift	"Slut" (Tayl...	0.629	0.413	Top 50 - USA	77.983	0.306
<input type="checkbox"/>	42473f26-5a6b-435a...	0.677	Soolking	Casanova	0.801	0.832	Top 50 - Fr...	132.029	0.758
<input type="checkbox"/>	90947158-8a02-4fb9...	0.701	Kenya Grace	Strangers	0.628	0.523	Top 50 - Un...	169.982	0.416
<input type="checkbox"/>	211fa724-c84e-456f...	0.191	Mora	MODELITO	0.711	0.752	Top 50 - Sp...	145.047	0.528
<input type="checkbox"/>	a0b7915c-5cd7-4dff...	0.0887	Bad Bunny	PERRO NEG...	0.911	0.778	Top 50 - Me...	96.057	0.345
<input type="checkbox"/>	a4f9eb16-3b5b-4180...	0.409	VillaBanks	Papaya	0.674	0.827	Top 50 - Italy	139.983	0.419
<input type="checkbox"/>	6dae0baa-ef57-4fbc...	0.256	Tate McRae	greedy	0.75	0.733	Top 50 - Ca...	111.004	0.844

Ilustración 13: Tabla “Spotify_data” tras ejecutar la función “Spotify_extract”

3.2.3.3 Amazon S3 (Simple Storage Service)

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en la industria. Los clientes de todos los tamaños e industrias pueden usar Amazon S3 para almacenar y proteger cualquier cantidad de datos para una variedad de casos de uso, como lagos de datos, sitios web, aplicaciones móviles, respaldo y restauración, archivado, aplicaciones empresariales, dispositivos IoT y big data. analítica. Amazon S3 proporciona funciones de administración para que pueda optimizar, organizar y configurar el acceso a los datos para cumplir con los requisitos necesarios [18].

Para el desarrollo del proyecto se crean dos Buckets que desempeñan funciones diferentes:

- **Bucket bucketspoti:** Este primer bucket contiene un archivo .zip llamado “spotify_layer.zip” en el que se encuentran todas las librerías necesarias para que “Spotify_extract” pueda ejecutarse sin errores de dependencias.

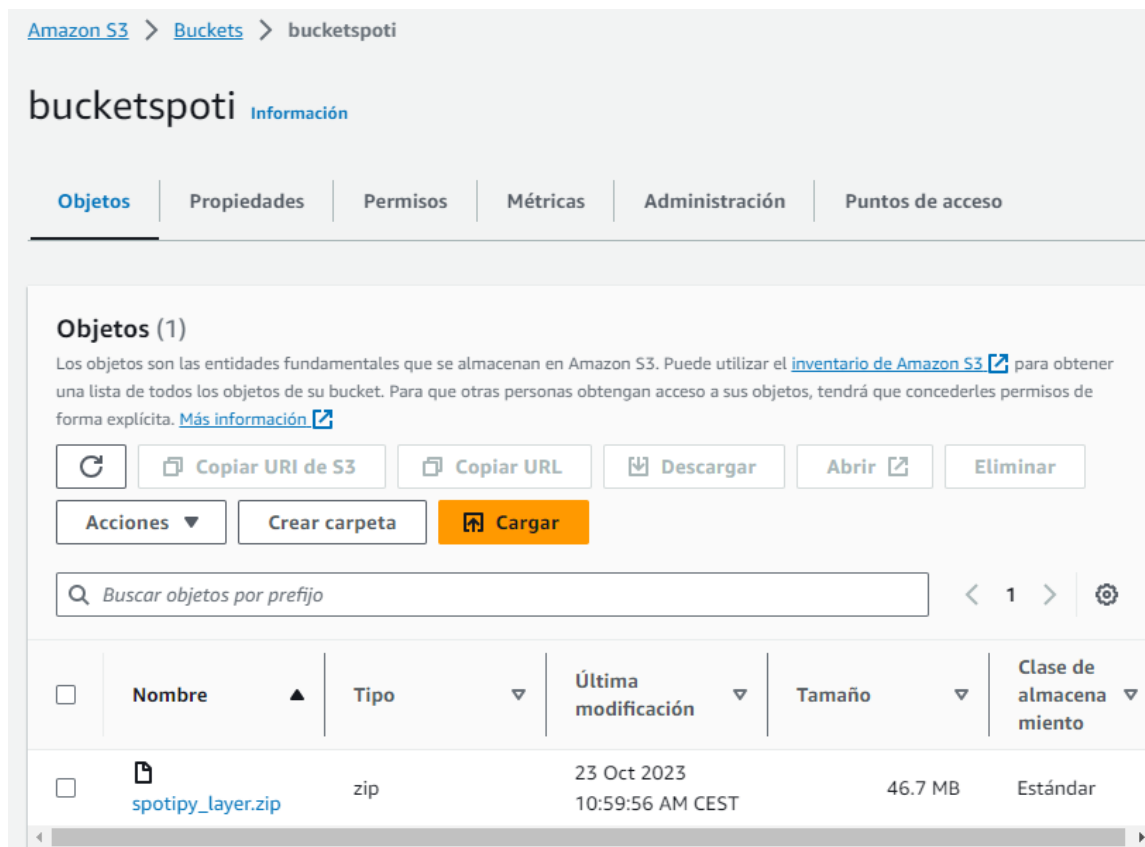


Ilustración 14: S3 Bucket “bucketspoti”

- **Bucket dynamodb-exportt:** Este bucket se encarga de almacenar todos los archivos JSON generados por la Función Lambda extract_DB_to_S3. Estos archivos se generan todos los lunes y contienen los datos de la tabla Spotify_data en formato JSON. Además, en el nombre del archivo se especifica la fecha de creación para tener un historial de los datos almacenados.

3.2.3.4 Funciones Lambda “extract_DB_to_S3” y “export_jsonS3_to_URL”

Estas funciones Lambda tienen como principal tarea la exportación y transformación de datos almacenados en la tabla DynamoDB. Su objetivo es mantener estos datos disponibles y actualizados para su posterior visualización y análisis. Esta regularidad en la exportación, que se realiza semanalmente, garantiza que los usuarios finales siempre dispongan de información actualizada.

A continuación, se detalla el funcionamiento de las funciones:

- **Función extract_DB_to_S3:** Esta Función Lambda es la encargada de obtener los datos de la tabla Spotify_data y los convierte a formato JSON. Seguidamente sube el JSON exportado al bucket S3 dynamodb-exportt. El archivo JSON almacenado en el bucket contiene en su nombre la fecha exacta en la que se generó para así poder llevar un seguimiento de los datos utilizados cada semana.
- **Función export_jsonS3_to_URL:** Esta función se encarga de recuperar el contenido archivo JSON más recientemente almacenado en el bucket dynamodb-exportt y lo devuelve como respuesta en una URL asociada a la función Lambda. Más adelante, esta URL que contiene los datos en formato JSON actuará como origen de datos para la herramienta de visualización, lo que facilitará la conexión entre esta herramienta y nuestro sistema desplegado en la nube.

3.2.4 Automatización y Monitorización

Para automatizar y monitorear las tareas descritas anteriormente se utilizan los servicios de AWS EventBridge y CloudWatch respectivamente.

3.2.4.2 Amazon EventBridge

EventBridge es un servicio sin servidor que utiliza eventos para conectar los componentes de la aplicación, lo que facilita la creación de aplicaciones escalables basadas en eventos [19]. En el proyecto se ha utilizado este servicio para configurar reglas las cuales se encargan de ejecutar las funciones AWS Lambda implementadas según una programación específica.

Existen dos tipos de reglas programadas:

- Reglas que se ejecutan a un ritmo regular (por ejemplo, cada 30 minutos)
- Reglas que se ejecutan en momentos específicos (por ejemplo, a las 8:00 a.m. PST el primer lunes de cada mes)

El tipo de regla más adecuado para el proyecto es el segundo, ya que se necesita especificar la fecha y la hora. Para ello se define una expresión cron para cada regla, la cual determina la fecha y la hora en la que la regla debe ejecutarse [20]. Una expresión cron es una combinación de 6 valores que indican a EventBridge cuándo ejecutar la regla. Por ejemplo, cron(0 12 * * ? *) ejecuta la regla todos los días a las 12:00 h UTC+0 [20].

Las reglas creadas para el proyecto son las siguientes:

- **Spotify_monday00:** Esta regla se encarga de ejecutar la función Spotify_extract todos los lunes a las 00:00 CET. A continuación, se detalla la configuración de la regla:

En esta primera pantalla de configuración se detallan el nombre de la regla, la descripción y el tipo de regla (programada en este caso).

The screenshot shows the 'Definición de detalles de reglas' (Rule Details Definition) step in the Amazon EventBridge console. The breadcrumb trail at the top is 'Amazon EventBridge > Reglas > spotify_monday00 > Editar regla'. Below this, it says 'Paso 1 de 5' and 'Definición de detalles de reglas' with a link to 'Información'. The main section is titled 'Detalles de la regla' and contains the following fields:

- Nombre:** A text box containing 'spotify_monday00'. Below it, a note states: '64 caracteres como máximo; se admiten números, letras en mayúscula o en minúscula y los caracteres -, _ y .'. There is also a link to 'Información'.
- Descripción - opcional:** A text box with the placeholder 'Ingrese la descripción'.
- Bus de eventos:** A dropdown menu with 'default' selected. Below it, a note says: 'Seleccione el bus de eventos al que se aplica esta regla, ya sea el predeterminado o uno de socios.' There is also a link to 'Información'.
- Habilitar la regla en el bus de eventos seleccionado:** A toggle switch that is currently turned on.
- Tipo de regla:** Two radio button options:
 - ☐ **Regla con un patrón de eventos**
Regla que se ejecuta cuando un evento coincide con el patrón de eventos definido. EventBridge envía el evento al destino especificado.
 - ☒ **Programar**
Una regla que se ejecuta según una programación

At the bottom right, there are two buttons: 'Cancelar' and 'Siguiente'.

Ilustración 15: Detalles de la regla Spotify_Monday00

El segundo paso de la configuración es la definición del patrón de programación. En este caso se utiliza una programación precisa con una expresión Cron, concretamente cron(00 00 ? * Mon *)

Definir programación [Información](#)

Patrón de programación

Patrón de programación

Elija el tipo de programación que mejor se adapte a sus necesidades.

☒ Una programación detallada que se ejecuta a una hora específica, como las 8.00 h PST del primer lunes de cada mes.

☐ Un horario que se ejecuta con una frecuencia periódica; p. ej., cada 10 minutos.

Expresión cron [Información](#)

Definición de la expresión CRON para la programación

 **CRON** (

Minutos Horas Día del mes Mes Día de la semana Año

Fechas de los siguientes 10 desencadenadores

UTC ▼

Mon, 08 Jan 2024 00:00:00 UTC
Mon, 15 Jan 2024 00:00:00 UTC
Mon, 22 Jan 2024 00:00:00 UTC
Mon, 29 Jan 2024 00:00:00 UTC
Mon, 05 Feb 2024 00:00:00 UTC
Mon, 12 Feb 2024 00:00:00 UTC
Mon, 19 Feb 2024 00:00:00 UTC
Mon, 26 Feb 2024 00:00:00 UTC
Mon, 04 Mar 2024 00:00:00 UTC
Mon, 11 Mar 2024 00:00:00 UTC

Cancelar

Anterior

Siguiente

Ilustración 16: Definición de patrón de programación de la regla Spotify_Monday00

En el tercer paso se selecciona el destino de la regla. El destino es donde se envía el evento cuando coincide con el patrón de eventos de la regla (en este caso se selecciona la Función Lambda `spotify_extract`).

Amazon EventBridge > Reglas > spotify_monday00 > Editar regla

Paso 3 de 5

Seleccionar destinos

Permisos

Nota: Al usar la consola de EventBridge, EventBridge configurará automáticamente los permisos correspondientes para los destinos seleccionados. Si utiliza CloudFormation, el SDK o la CLI de AWS, debe configurar los permisos adecuados.

Destino 1

Tipos de destino
Seleccione un bus de eventos de EventBridge, un destino de la API de EventBridge (socio de SaaS) u otro servicio de AWS como destino.

☐ Bus de eventos de EventBridge
☐ Destino de la API de EventBridge
☒ Servicio de AWS

Seleccione un destino [Información](#)
Seleccione los destinos que desee invocar cuando un evento coincida con el patrón de eventos o cuando se active la programación (hay un límite de 5 destinos por regla).

Función

► Configurar la versión o el alias

► Configuración adicional

Ilustración 17: Selección de destino de la regla Spotify_Monday00

Así, la regla `spotify_monday00` queda configurada y lista para realizar su función.

- **Spotify_monday08:** Esta regla se encarga de ejecutar la función `extract_DB_to_S3` todos los lunes a las 00:08 CET. La expresión cron es `cron(08 00 ? * Mon *)` y el destino la Función Lambda `extract_DB_to_S3`.
- **Spotify_monday15:** Esta regla se encarga de ejecutar la función `export_jsonS3_to_URL` todos los lunes a las 00:15. La expresión cron es `cron(15 00 ? * Mon *)` y el destino la Función Lambda `export_jsonS3_to_URL`.

La configuración de las reglas `spotify_monday08` y `spotify_monday15` es similar a la configuración mostrada en las ilustraciones de la primera regla por lo que no se documenta con ilustraciones. Los únicos cambios para realizar son el nombre y descripción, la expresión cron y el destino de la regla (detallados anteriormente).

3.2.4.3 Amazon CloudWatch

Este servicio ofrecido por AWS monitorea los recursos y las aplicaciones de Amazon Web Services (AWS) que ejecuta en AWS en tiempo real. Se puede utilizar CloudWatch para recopilar y hacer un seguimiento de métricas, que son las variables que puede medir en los recursos y aplicaciones.

La página de inicio de CloudWatch muestra automáticamente las métricas sobre todos los servicios de AWS utilizados. También puede crear adicionalmente paneles personalizados para mostrar métricas sobre sus aplicaciones personalizadas, y mostrar colecciones personalizadas de métricas.

También permite crear alarmas que vigilen métricas y envíen notificaciones o realicen cambios automáticamente en los recursos que está monitoreando cuando se infringe un umbral [21].

Este servicio permite monitorear las ejecuciones de las Funciones Lambda y comprobar que no ha habido ningún error. Así se asegura que el sistema está funcionando correctamente y los datos se actualizan semanalmente, además de realizar un seguimiento de los recursos utilizados en la nube.

En la siguiente ilustración se presentan los grupos de registros de las ejecuciones de la Función Lambda `spotify_extract` que permiten el monitoreo de dichas ejecuciones.

The screenshot displays the AWS CloudWatch console interface for a log group named `/aws/lambda/spotify_extract`. The top navigation bar includes buttons for 'Acciones', 'Ver en Logs Insights', 'Empezar a seguir', and 'Buscar el grupo de registros'. Below this, the 'Detalles del grupo de registros' section provides key information:

- ARN:** `arn:aws:logs:us-east-1:369175358945:log-group:/aws/lambda/spotify_extract*`
- Bytes almacenados:** 97.23 KB
- Reglas de Contributor Insights:** -
- Horas de creación:** Hace 9 días
- Filtros de métricas:** 0
- ID de clave de KMS:** -
- Retención:** No vence nunca
- Filtros de suscripción:** 0
- Protección de datos:** -
- Recuento de datos confidenciales:** -

Below the details, a tabbed interface shows 'Flujos de registros' (selected), 'Etiquetas', 'Filtros de métricas', 'Filtros de suscripción', 'Contributor Insights', and 'Protección de datos'. The 'Flujos de registros' section shows 8 log streams. A search bar and filters (Coincidencia exacta, Mostrar caducado) are present. The log streams are listed in a table with columns for 'Secuencia de registro' and 'Hora del último evento'.

Secuencia de registro	Hora del último evento
2023/10/31/[\$LATEST]625a92cb44be4de6b96124...	2023-10-31 17:27:31 (UTC+01:00)
2023/10/31/[\$LATEST]83b27d0df0034008ae9b8b...	2023-10-31 10:38:45 (UTC+01:00)
2023/10/30/[\$LATEST]326f4452a6414e10942a8d...	2023-10-31 00:02:32 (UTC+01:00)
2023/10/30/[\$LATEST]12bf221193ed40b0bb118b...	2023-10-30 11:16:41 (UTC+01:00)
2023/10/23/[\$LATEST]43ccea4e73dd46c39cbcd0f...	2023-10-23 16:33:08 (UTC+02:00)
2023/10/23/[\$LATEST]9c4adb7086f5434583d74d...	2023-10-23 11:15:22 (UTC+02:00)
2023/10/23/[\$LATEST]da4a740751d347f7abe4f9...	2023-10-23 11:13:19 (UTC+02:00)
2023/10/23/[\$LATEST]03fc0e151c504c91934f0f2...	2023-10-23 11:08:16 (UTC+02:00)

Ilustración 18: Grupos de registros de la función `Spotify_extract`

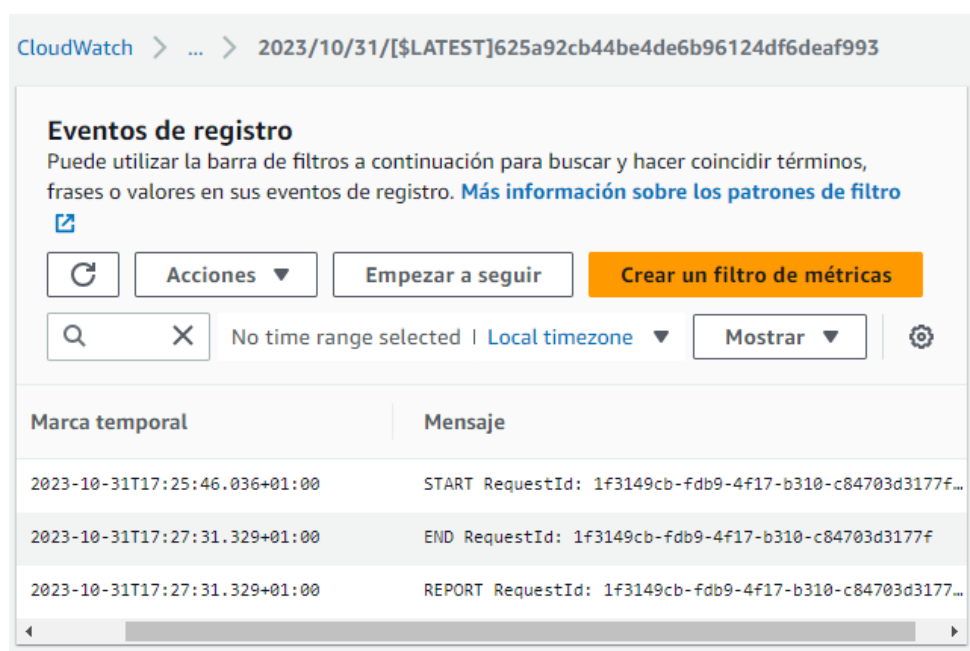


Ilustración 19: Registros de una ejecución de la función Spotify_extract

3.2.5 Visualización de datos y diseño

Una vez el sistema está desplegado en la nube y funcionando correctamente, se necesita una última pieza que permita visualizar los datos de manera sencilla al usuario. Para ello se elige Power BI, un servicio de análisis de datos de Microsoft® que permite conectarse a los datos, transformarlos y visualizarlos [22].

Con Power BI se consigue:

- **Conexión con los datos:** El primer paso es conectarse a los datos. Power BI permite conectarse a datos de muchos orígenes diferentes como archivos Excel, bases de datos SQL, Oracle, scripts... En este caso se utiliza la opción de Web, ya que permite conectar Power BI con una dirección URL. Esta opción simplifica la conexión con los datos ya que la única configuración a realizar es proporcionar la dirección URL de la Función Lambda `export_jsonS3_to_URL` la cual contiene los datos en formato JSON (como se ha explicado anteriormente).

En las siguientes ilustraciones se documenta la configuración para la conexión con los datos.

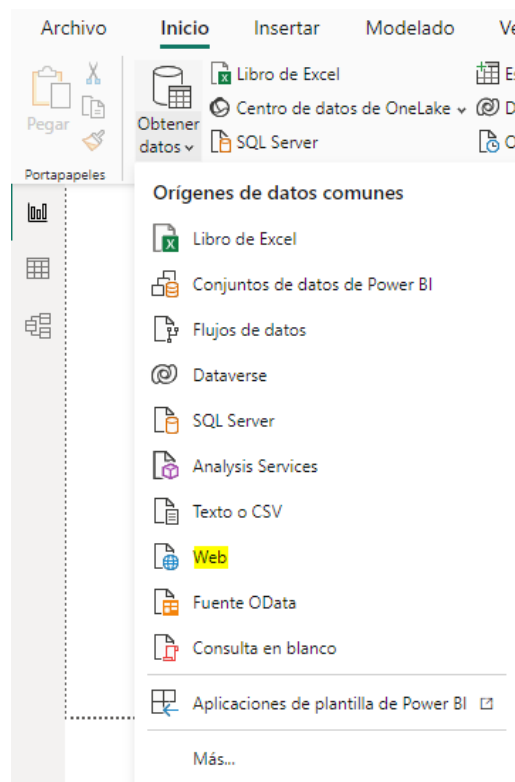


Ilustración 20: Selección de origen de datos

De web

☒ Básico ☐ Uso avanzado

Dirección URL

<https://...a.lambda-url.us-east-1.on.aws/>

Aceptar

Cancelar

Ilustración 21: Configuración de conexión con los datos en Power BI

- **Transformación de datos:** El siguiente paso tras realizar la conexión es transformar y limpiar los datos para crear un modelo que facilite la visualización y creación del informe. Para ello se utiliza Power Query, de Power BI. Power Query es un motor de transformación y preparación de datos. Incluye una interfaz gráfica para obtener datos de orígenes y un editor de Power Query para aplicar transformaciones [23]. A continuación, se muestran los pasos aplicados para la creación del modelo final:

	id	canción	Artista	Playlist	1.2 Danceability	1.2 Acousticness	1.2 Valence	1.2 Energy	1.2 Tempo
1	06821-3759-4247-8043-e9150576296	CA	Anuel AA	Spain	0.823	0.17	0.647	0.866	99.1
2	126a7-90c0-40fc-950d-a5d2b3bea0f0	My Love Mine All Mine	Mitski	United Kingdom	0.504	0.868	0.121	0.308	111.1
3	3893-be0d-441c-b0fe-a215b0ac2d75	Last Night	Morgan Wallen	Canada	0.492	0.467	0.478	0.675	203.1
4	ee266-809c-42cf-a8d8-d5e17886728	Cadillac (feat. Arlie Sive)	Boro	Italy	0.805	0.13	0.21	0.879	94.1
5	a14c-30f4-4168-ac36-ce7c376783ab	vampire	Olivia Rodrigo	United Kingdom	0.511	0.177	0.35	0.532	136.1
6	604e9-9887-4093-a88c-c1c04486831	I remember Everything (feat. Kacey Musgraves)	Zach Bryan	USA	0.429	0.354	0.153	0.493	77.1
7	986d7-4d5c-4279-902a-c86a199a0e0a	HIBIKI	Bad Bunny	Spain	0.801	0.604	0.529	0.645	119.1
8	10128-604e-66a2-9e1e-x1833c79d518	Petit génie	Jungtji	France	0.8	0.126	0.966	0.581	126.1
9	03021-fa38-4759-8b78-02899f9218b	9 9 9	SIRA	Germany	0.767	0.364	0.368	0.539	128.1
10	a031a-c55e-45ff-a370-35190ca3907a	DENTRO DA ILUX	Luan Pereira	Brazil	0.73	0.143	0.362	0.527	129.1
11	258a4-1baf-4408-9c0c-9670e7e829fc	Que Orda	Calle 24	Mexico	0.808	0.183	0.837	0.844	124.1
12	3ce2f-c42-48f7-9184-8a63938b011	First Person Shooter (feat. J. Cole)	Drake	Canada	0.47	0.0277	0.248	0.64	263.1
13	9f1af-d71b-484a-a40a-1a93-20492320	Baddest (feat. RIAH, Frowdian, Triggs & Talure)	Chase & Status	United Kingdom	0.62	0.228	0.539	0.893	87.1
14	1b0d6-e058-40f0-902a-50e26030760	Mão das Portas-Mãe - Ao Vivo	Gustavo Lima	Brazil	0.857	0.636	0.753	0.778	156.1
15	9e049-34a8-471a-8a14-53031a16130a	Water	Tyla	United Kingdom	0.673	0.0856	0.519	0.722	117.1
16	894e9-f1c1-4d19-86a3-d069413cc0f	I remember Everything (feat. Kacey Musgraves)	Zach Bryan	Canada	0.429	0.354	0.155	0.493	77.1
17	3ac75-48ba-448a-980e-2b12518a1007	Insta Love	Marracash	Italy	0.69	0.0721	0.905	0.795	101.1
18	8480c-5d86-4e8b-855e-1658e537738	Lollipop	Darell	Spain	0.822	0.179	0.694	0.711	102.1
19	82f1c-04df-4a0a-9f56-f0988fca399a	COME VUOI	Geolier	Italy	0.756	0.13	0.896	0.631	130.1
20	d5ae5-5192-4079-a89a-b43b0c3baf07	El Mamor	Kiko Rivera	Spain	0.884	0.0406	0.729	0.718	139.1
21	7a7a0-c077-4e13-a09a-11990463434a	Si No Hubiera	Rigo Montoya	Italy	0.537	0.337	0.524	0.411	98.1
22	8a81f-14180-419c-a039-0570a2567881	greedy	Tate McRae	United Kingdom	0.75	0.256	0.824	0.728	111.1
23	84186-943a-422b-8a0c-a6a236f735a	Morinda	Dave	France	0.91	0.232	0.587	0.574	115.1
24	b8342-fc9e-45ab-817f-5be18b9960a8	PERRO NEGRO	Bad Bunny	Mexico	0.911	0.0887	0.945	0.778	96.1
25	26dae-28f0-4eb4-8d5e-09159a70a58a	Cruel Summer	Taylor Swift	United Kingdom	0.552	0.117	0.564	0.702	169.1
26	2cef5-a89b-40cb-8230-491d6b30948f	IGGAF (feat. Yeat)	Drake	Canada	0.683	0.0464	0.188	0.67	136.1
27	945e-9440-7227-a97f-4a6d78ef579a	EL AMOR DE SU VIDA	Grupo Frontera	Mexico	0.616	0.06	0.746	0.834	151.1
28	3e47e-840c-488b-810c-c499f9f98b0a	Vive sur ton chemin - Techno Mix	808Mafia	Germany	0.634	0.908	0.371	0.824	177.1
29	8088-879f-471a-8402-x376535950a	Stronger	Kemec France	France	0.628	0.701	0.435	0.529	160.1
30	13a0f-1153-46da-b09b-a030f73e08f3	BON TON (feat. Lazza, Blanco, Sfera Ebbasta & Michelangelo)	Drillonaire	Italy	0.676	0.0813	0.656	0.81	130.1
31	7661b-9d33-4a1e-8556-6530e95f9d3	Sprinter	Dave	United Kingdom	0.916	0.186	0.661	0.583	139.1
32									

Ilustración 24: Visualización del modelo final desde Power Query

Para finalizar la transformación se aplican los cambios realizados y se guarda el modelo, con el botón “Cerrar y aplicar” de la esquina superior izquierda de la ilustración 25:

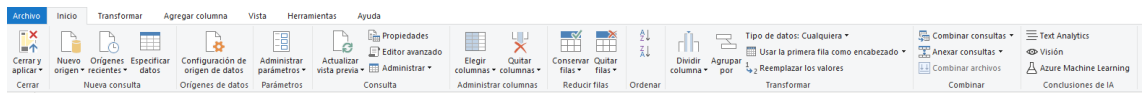


Ilustración 25: Cinta de opciones de Editor de Power Query

- **Diseño del informe final:** Una vez se ha creado el modelo de datos deseado, el siguiente paso es crear y diseñar el informe final el cual se proporcionará al usuario. A continuación, se detallan los objetos visuales que componen el informe final.

Este informe está formado por distintos objetos visuales:

- **Segmentación de datos:** Las segmentaciones de datos son un tipo de filtrado. Se muestran en la página del informe y limitan la parte del conjunto de datos que se muestra en las demás visualizaciones del informe. En la esquina superior izquierda del informe se pueden observar estos dos elementos con título “País” y “Artista” con los que el usuario puede filtrar los datos seleccionando los valores que desee de estos dos campos. En la siguiente ilustración se muestra un ejemplo de cómo utilizar este filtrado.

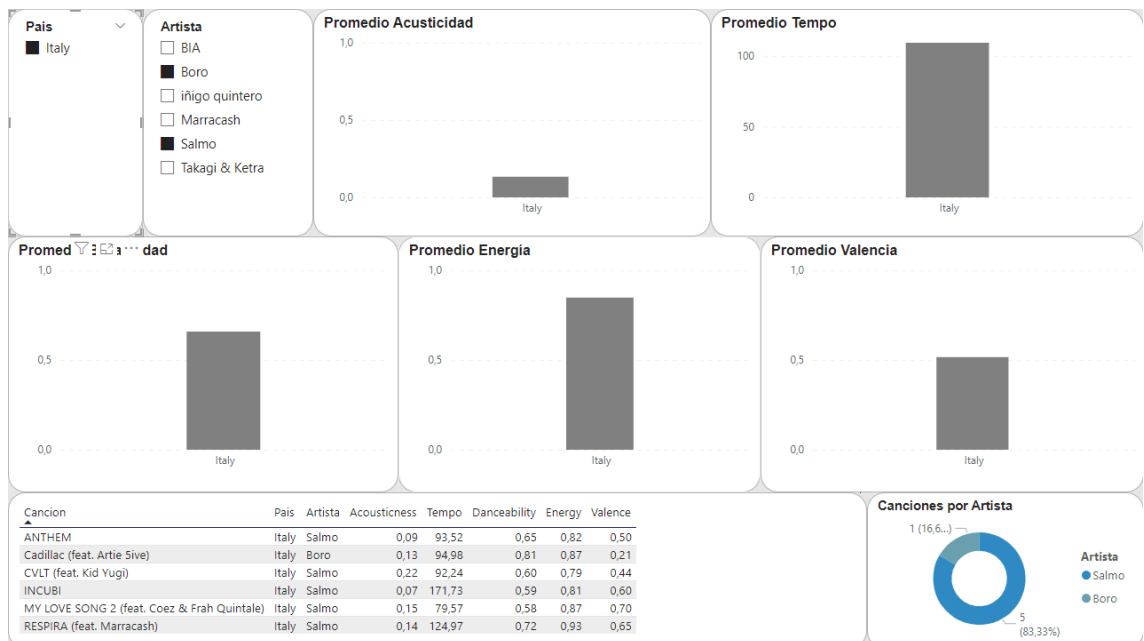


Ilustración 26: Ejemplo de filtrado por País y Artista.

- Gráficos de columnas agrupadas: Este objeto visual se utiliza para proporcionar de manera clara la información sobre los cinco parámetros musicales (acusticidad, tiempo, bailabilidad, energía y valencia), y poder comparar los valores fácilmente dependiendo del país. En el informe se dispone de cinco gráficas de este tipo, una para cada parámetro musical. En ellas se muestra sobre el eje X la lista de países y sobre el eje Y el valor promedio del parámetro musical asociado a la gráfica.

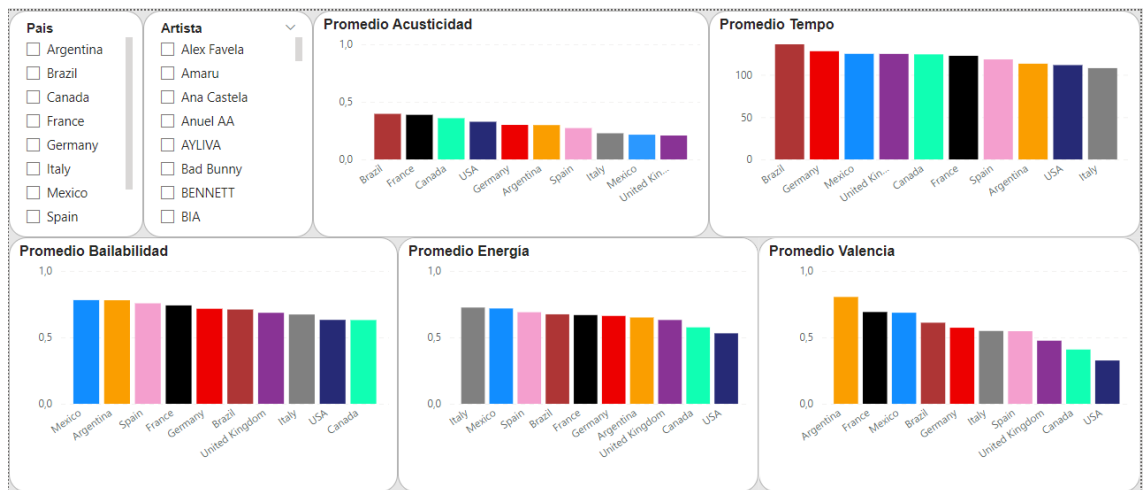


Ilustración 27: Gráficos de columnas agrupadas del informe final.

- Tabla: En la parte inferior del informe se proporciona una tabla con todos los datos del modelo para permitir al usuario acceder a la información detallada de manera sencilla. En la parte derecha se encuentra una barra de desplazamiento vertical para navegar por la tabla y poder observar todos los datos.

Cancion	Pais	Artista	Acousticness	Tempo	Danceability	Energy	Valence
"Slut!" (Taylor's Version) (From The Vault)	USA	Taylor Swift	0,35	77,98	0,63	0,41	0,31
9 bis 9	Germany	SIRA	0,36	128,07	0,77	0,54	0,37
ANTHEM	Italy	Salmo	0,09	93,52	0,65	0,82	0,50
Baile do Bruxo	Brazil	Tropa do Bruxo	0,89	162,52	0,73	0,23	0,52
Barulho Do Foguete - Ao Vivo	Brazil	Zé Neto & Cristiano	0,17	79,99	0,65	0,87	0,62
Blonde Chaya - Sped up	Germany	Amaru	0,30	170,93	0,72	0,80	0,83

Ilustración 28: Tabla del informe final.

- Gráfico de anillos: Se utiliza este objeto visual para mostrar las canciones agrupadas por artista con el fin de que el usuario pueda observar fácilmente cuántas canciones de cada artista se tienen.

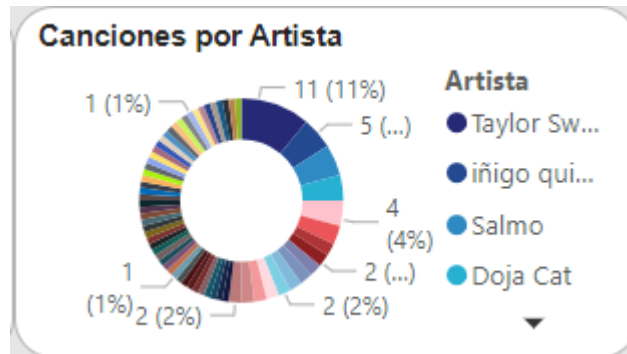


Ilustración 29: Gráfico de anillos de canciones agrupadas por artistas.

A continuación, se muestra la vista completa del informe final tal y como se proporciona al usuario.

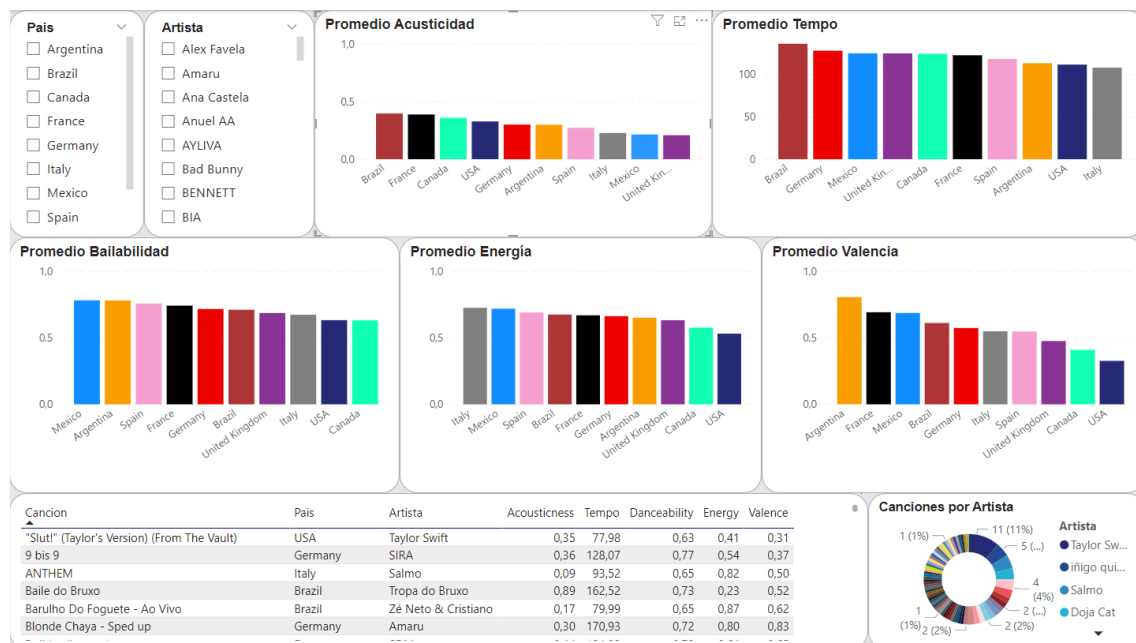


Ilustración 30: Informe final

4 Manual de uso

En este capítulo se detalla un manual de uso del informe final de Power BI. El informe de Power BI actúa como una interfaz para el usuario mediante la cual se presentan todos los datos recopilados por el sistema. Las acciones más relevantes que se pueden llevar a cabo para su buen uso son las siguientes:

- 1. Actualización de datos:** El sistema actualiza semanalmente los datos alojados en la nube de Amazon Web Services de manera automática, pero en el informe se deben actualizar manualmente. Para ello, cada vez que se accede al informe es recomendable actualizar los datos. La primera manera es utilizar la acción “Actualizar” que encontramos en la pestaña de inicio, en la sección de consultas.



Ilustración 31: Cinta de opciones de Inicio de Power BI

La otra manera, desde el panel de “Datos” en la parte derecha de la pantalla, en el modelo importado con nombre “spotify_data” existe la opción de actualizar los datos, siendo el resultado el mismo que el de la opción anterior.

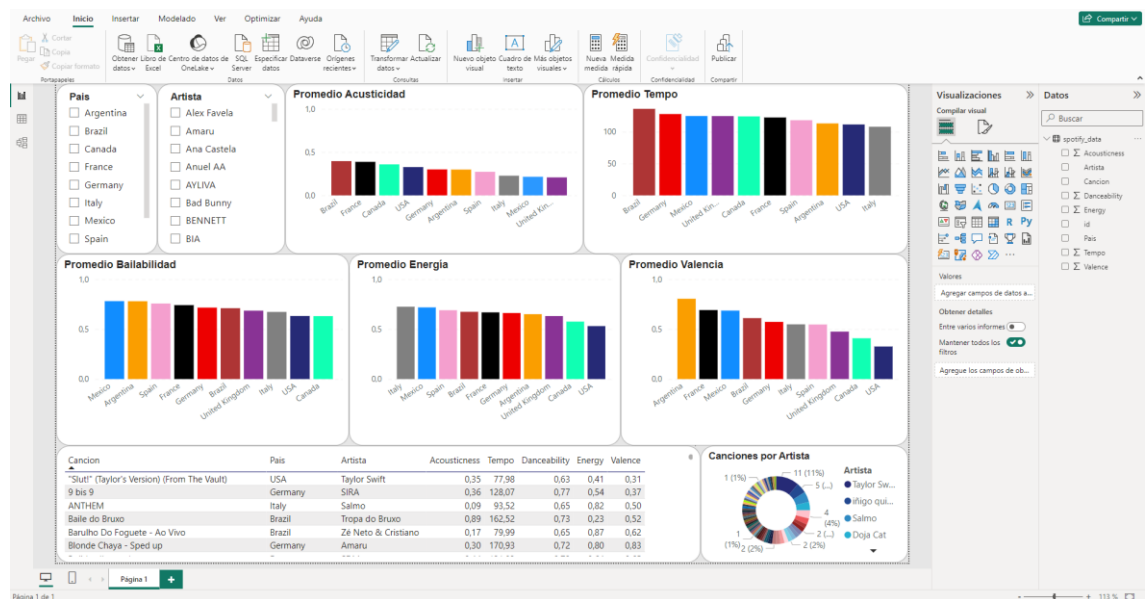


Ilustración 32: Paneles de Visualizaciones y Datos

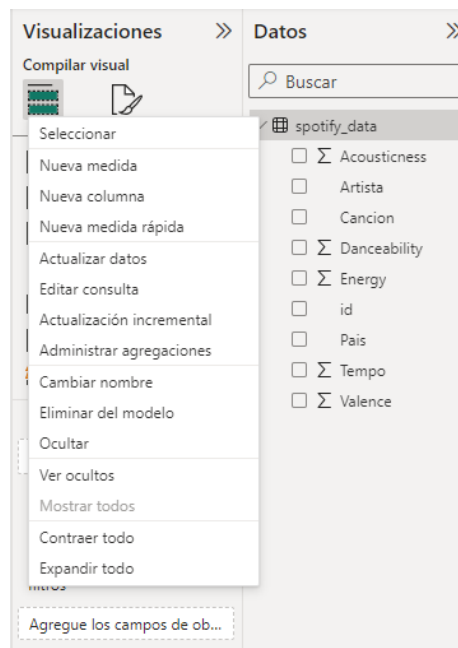


Ilustración 33: Opciones del modelo

2. Filtrado de datos: Los informes de Power BI no son informes estáticos, si no que permiten la interacción del usuario con el informe mediante una interfaz sencilla de utilizar. Así, el usuario puede modificar la vista del informe a su parecer sin que el informe original se vea modificado. Algunas de estas acciones son:

- Filtrado por País: El usuario puede filtrar los datos por el atributo país, mostrando así los datos asociados al país o países seleccionados.



Ilustración 34: Filtrado por País

- Filtrado por artista: El usuario puede filtrar los datos por el atributo artista, mostrando así solo los datos asociados a las canciones del artista o artistas seleccionados.

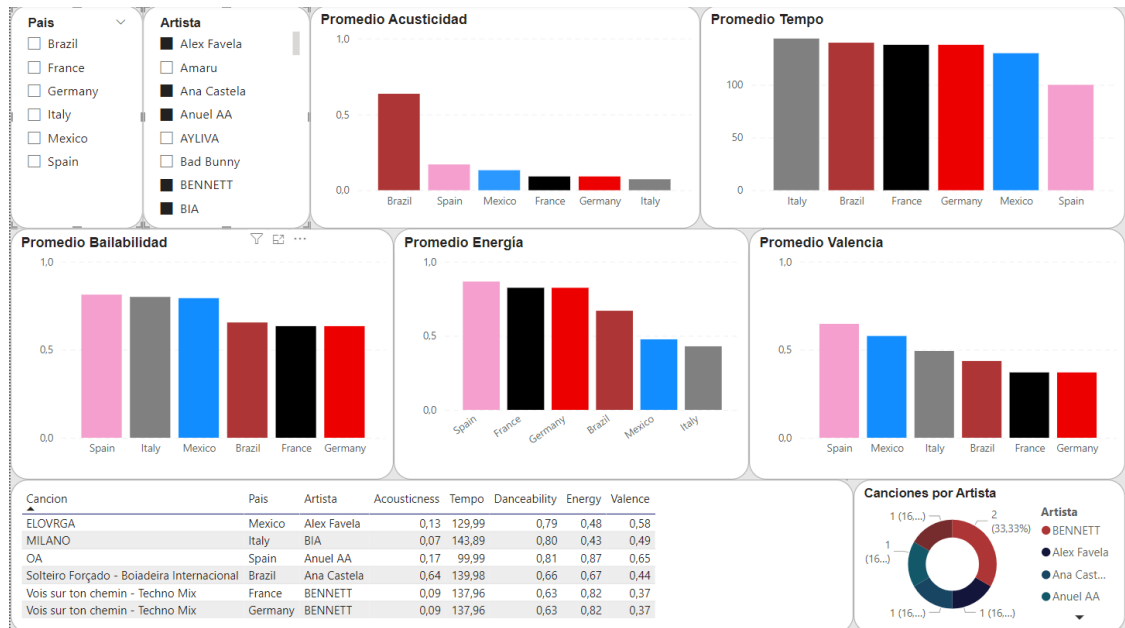


Ilustración 35: Filtrado por Artista

- Filtrado por canción: El usuario puede filtrar los datos por el atributo canción, mostrando los datos asociados a la canción o canciones seleccionadas. Este filtro se realiza en la tabla de la parte inferior izquierda del informe.

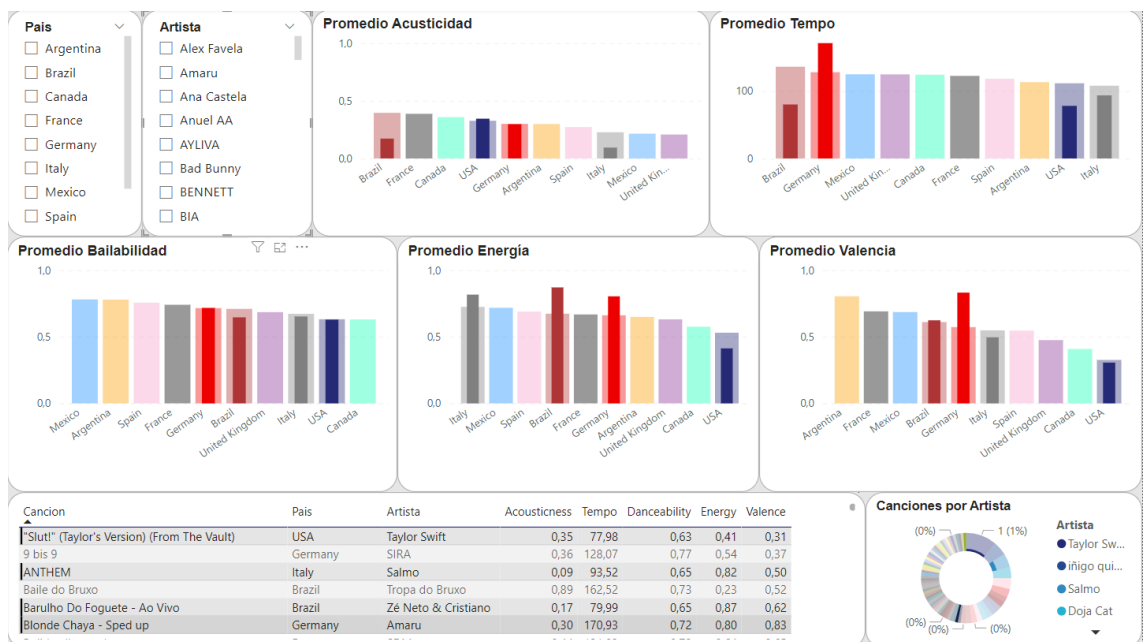


Ilustración 36: Filtrado por Canción

Para realizar una selección múltiple de filtrado basta con mantener la tecla “control” del teclado del ordenador al seleccionar las opciones. Todos los filtros detallados anteriormente se pueden combinar y así filtrar por distintos atributos a la vez. Para volver al estado original del informe basta con clicar en cualquier zona del informe o utilizar la opción de “borrado de selecciones” en los filtros de País y Artista, la cual aparece en la esquina superior derecha de los elementos.

Además, todos los elementos del informe son interactivos y el usuario puede utilizarlos de la manera que más le beneficie.

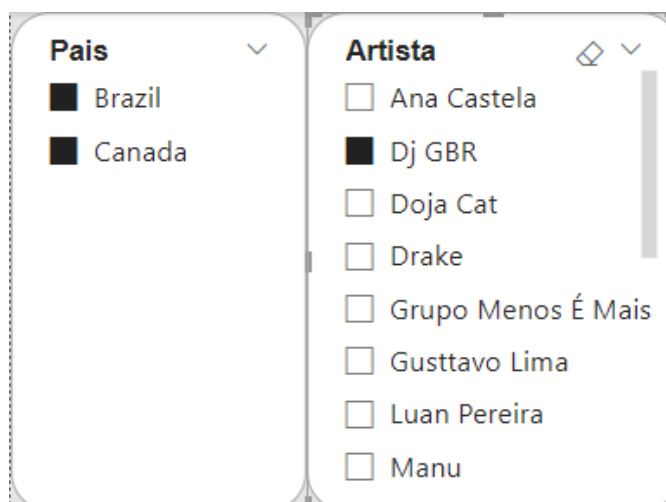


Ilustración 37: Borrado de filtro

3. Personalización del informe: El informe diseñado en el proyecto proporciona una visualización general y sencilla de los datos. De todas maneras, el usuario puede personalizar y modificar el informe a su disposición si fuera necesario. Para ello puede añadir páginas al informe y añadir en ellas cualquier objeto visual que necesite.

4. Parámetros musicales: En el informe se muestran datos sobre cinco parámetros musicales los cuales son calculados por Spotify. Aquí se definen cada uno de ellos detallando los valores que pueden tomar:

- Tempo: El tempo general estimado de una pista en pulsaciones por minuto (BPM). En terminología musical, el tempo es la velocidad o el ritmo de una pieza determinada y se deriva directamente de la duración promedio del tiempo [16].
- Danceability (bailabilidad): La bailabilidad describe qué tan adecuada es una pista para bailar en función de una combinación de elementos musicales que incluyen el tempo, la estabilidad del ritmo, la fuerza del ritmo y la regularidad general. Un valor de 0,0 es el menosailable y 1,0 es el másailable [16].
- Acousticness (acusticidad): Una medida de confianza de 0,0 a 1,0 sobre si la pista es acústica. 1,0 representa una alta confianza en que la pista es acústica [16].
- Energy (energía): La energía es una medida de 0,0 a 1,0 y representa una medida perceptiva de intensidad y actividad. Normalmente, las pistas enérgicas son rápidas y ruidosas. Por ejemplo, el death metal tiene mucha energía, mientras que un preludio de Bach obtiene una puntuación baja en la escala. Las características que contribuyen a este atributo incluyen rango dinámico, volumen percibido, timbre, velocidad de inicio y entropía general [16].
- Valence (valencia): Una medida de 0,0 a 1,0 que describe la positividad musical que transmite una pista. Las pistas con valencia alta suenan

más positivas (p. ej., felices, alegres, eufóricas), mientras que las pistas con valencia baja suenan más negativas (p. ej., tristes, deprimidas, enfadadas) [16].

5 Resultados y conclusiones

La realización de este proyecto lleva a la conclusión de que los servicios de computación en la nube (en este caso, los de Amazon Web Services) permiten crear una gran variedad de programas y soluciones tecnológicas sin la necesidad de utilizar los medios tradicionales. Además, AWS proporciona grandes ventajas y características como:

1. **Facilidad de uso:** AWS está diseñado para permitir que los proveedores de aplicaciones, los proveedores de software independientes y los distribuidores puedan hospedar de una forma rápida y segura su aplicación, tanto si es una aplicación existente como si es una nueva aplicación basada en SaaS. Puede utilizar la consola de administración de AWS o las API de servicio web bien documentadas para obtener acceso a la plataforma de hospedaje de aplicaciones de AWS [25].
2. **Flexible:** AWS le permite seleccionar el sistema operativo, el lenguaje de programación, la plataforma de aplicaciones web, la base de datos, así como el resto de servicios que necesite. Con AWS, tendrá acceso a un entorno virtual que le permite cargar el software y los servicios que necesita su aplicación. Esto facilita el proceso de migración de las aplicaciones existentes y mantiene las opciones para crear nuevas soluciones [25].
3. **Rentable:** Únicamente tendrá que afrontar el costo de la potencia de cómputo, el almacenamiento y demás tipos de recursos que utilice, sin contratos a largo plazo ni compromisos iniciales. Para obtener más información sobre la comparación de los costos de otras alternativas de hospedaje con AWS, consulte el Centro de ahorro de AWS [25].
4. **Confianza:** Con AWS, tendrá a su disposición una infraestructura informática global escalable, segura y de confianza, la columna vertebral virtual de Amazon.com, una tienda online valorada en varios miles de millones de dólares que lleva en el candelero más de una década [25].
5. **Escalabilidad y alto desempeño:** Con las herramientas de AWS, Auto Scaling y Elastic Load Balancing, su aplicación podrá ampliarse o reducirse según la demanda. Gracias al respaldo de la sólida infraestructura de Amazon, tendrá acceso a los recursos informáticos y de almacenamiento siempre que los necesite [25].
6. **Seguro:** Aplica un enfoque integral para proteger y reforzar nuestra infraestructura, incluidas medidas físicas, operativas y de software. Para obtener más información, consulte el Centro de seguridad de AWS [25].
7. **Ahorro de costos:** La nube le permite reemplazar los gastos fijos (como los centros de datos y servidores físicos) por gastos variables y pagar solo por los recursos de TI a medida que los utiliza. Además, debido a las economías de escala, los gastos variables son mucho menores del monto que pagaría por ocuparse por cuenta propia de estos recursos [25].

En cuanto a aprendizaje académico, se han adquirido conocimientos sobre los servicios utilizados de AWS como AWS Lambda, Amazon DynamoDB, Amazon CloudWatch y Amazon S3, el diseño e implementación de soluciones basadas en la nube, desarrollo de código en Python, desarrollo de ETLs, herramientas Business Intelligence (Power BI) y funcionamiento de APIs (Spotify Web API).

Por otra parte, se ha conseguido implementar una solución tecnológica basada en la nube con la intención de facilitar y dar soporte a la toma de decisiones en la industria musical.

A nivel personal, la realización del proyecto ha ayudado a comprender el funcionamiento de la programación en la nube y cómo configurar, implementar y utilizar esta nueva tecnología para la resolución del problema planteado. Además, las tecnologías utilizadas son muy comunes en el ámbito laboral y no tanto en el ámbito universitario, lo cual es un punto a favor.

6 Análisis de Impacto

En este capítulo se lleva a cabo un análisis de impacto de los resultados obtenidos durante el desarrollo del Trabajo de Fin de Grado sobre los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030.

- **ODS 7 (Energía Asequible y No Contaminante):** La utilización de los servicios en la nube de AWS proporciona una infraestructura compartida, eliminando así la necesidad de crear una infraestructura propia y ahorrando en el consumo de energía. Por otro lado, AWS permite ajustar los recursos lo cual contribuye a la eficiencia energética.

- **ODS 9 (Industria, Innovación e Infraestructura):** AWS ofrece herramientas avanzadas y servicios gestionados que facilitan el desarrollo de aplicaciones. Con ello, se promueve la innovación en el desarrollo de soluciones tecnológicas eficientes y sostenibles.

- **ODS 11 (Ciudades y Comunidades Sostenibles):** La utilización de servicios en la nube proporciona una base tecnológica escalable y flexible. Con los servicios proporcionados por AWS se podrían llevar a cabo diversas mejoras y soluciones para la gestión de ciudades.

- **ODS 12 (Producción y Consumo Responsables):** Con la utilización de servicios en la nube se consigue una gestión de recursos mas eficiente, ya que, al pagar solo por los recursos utilizados, se promueve la optimización de recursos a utilizar consiguiendo así un consumo responsable por parte de los usuarios.

- **ODS 17 (Alianzas para lograr los Objetivos):** Con AWS se facilita la colaboración debido a sus plataformas compartidas. La adopción de servicios en la nube promueve la colaboración y conectividad, lo que es fundamental para abordar desafíos globales de manera conjunta.

7 Bibliografía

- [1] Spotify, «Spotify,» [En línea]. Available: <https://developer.spotify.com/documentation/web-api>.
- [2] I. o. s. f. Amazon Web Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://aws.amazon.com/es/lambda/>.
- [3] I. o. s. f. Amazon Web Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://aws.amazon.com/es/pm/dynamodb/>.
- [4] I. o. s. f. Amazon Web Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html>.
- [5] «Amazon Web Services,» [En línea]. Available: <https://aws.amazon.com/es/s3/>.
- [6] Microsoft, «Microsoft,» 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/power-bi/transform-model/desktop-query-overview#power-query-editor>.
- [7] I. o. s. a. Amazon Web Services, «Amazon Web Services,» 2023. [En línea]. Available: https://docs.aws.amazon.com/es_es/AmazonCloudFront/latest/DeveloperGuide/lambda-edge-add-triggers.html.
- [8] I. o. s. f. Amazon Web Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://aws.amazon.com/es/training/awsacademy/>.
- [9] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: https://docs.aws.amazon.com/es_es/whitepapers/latest/aws-overview/compute-services.html#aws-lambda.
- [10] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://aws.amazon.com/es/eventbridge/>.
- [11] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://aws.amazon.com/es/cloudops/monitoring-and-observability/>.
- [12] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://aws.amazon.com/es/cloudwatch/>.
- [13] I. o. s. f. Amazon Web Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://aws.amazon.com/es/what-is-aws/>.
- [14] P. Lamere, «Spotipy,» 2014. [En línea]. Available: <https://spotipy.readthedocs.io/en/2.22.1/#>.
- [15] I. Amazon Web Services, «Boto3 Amazon Web Services,» 2023. [En línea]. Available: <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>.
- [16] Spotify, «Spotify for Developers,» 2023. [En línea]. Available: <https://developer.spotify.com/documentation/web-api/reference/get-audio-features>.
- [17] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developerguide/Introduction.html.
- [18] I. o. s. a. Amazon Web Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>.

- [19] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: https://docs.aws.amazon.com/es_es/eventbridge/latest/userguide/eb-what-is.html.
- [20] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-create-rule-schedule.html>.
- [21] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: https://docs.aws.amazon.com/es_es/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html.
- [22] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/power-bi/fundamentals/desktop-what-is-desktop>.
- [23] A. W. Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/power-query/power-query-what-is-power-query>.
- [24] Wikipedia, «Wikipedia,» 2023. [En línea]. Available: https://es.wikipedia.org/wiki/Power_BI.
- [25] I. o. s. f. Amazon Web Services, «Amazon Web Services,» 2023. [En línea]. Available: <https://aws.amazon.com/es/application-hosting/benefits/>.

8 Anexos

1. Código de la función Lambda `spotify_extract`:

```
1 import spotipy
2 from spotipy.oauth2 import SpotifyClientCredentials
3 import boto3
4 import uuid
5 import os
6 from decimal import Decimal
7
8 def lambda_handler(event, context):
9
10     def get_playlist_name(playlist_id):
11         playlist_info = sp.playlist(playlist_id)
12         return playlist_info['name']
13
14     # Spotify Credentials
15     client_id = os.environ['spotify_client_id']
16     client_secret = os.environ['spotify_client_secret']
17     # Spotify API Client
18     client_credentials_manager = SpotifyClientCredentials(
19         client_id=client_id,
20         client_secret=client_secret,
21     )
22     sp = spotipy.Spotify(client_credentials_manager=client_credentials_manager)
23
24     # DynamoDB connection
25     dynamodb = boto3.resource('dynamodb')
26     table_name = 'spotify_data'
27     table = dynamodb.Table(table_name)
28
29     scan = table.scan(
30         ProjectionExpression='#k',
31         ExpressionAttributeNames={
32             '#k': 'id'
33         }
34     )
```

```

35 # Clean DB
36 with table.batch_writer() as batch:
37     for each in scan['Items']:
38         batch.delete_item(Key=each)
39
40 playlist_ids = [
41     '3719dQZEVXBjFN1Vw8d9', # España
42     '3719dQZEVXBjQnJ7RRhdSX', # Italia
43     '3719dQZEVXBjZcmkrIHGU', # Alemania
44     '3719dQZEVXBjPWWFssbupI', # Francia
45     '3719dQZEVXBjLnolsZ8PSNw', # Reino Unido
46     '3719dQZEVXBjLRQDuF5je8p', # Estados Unidos
47     '3719dQZEVXBjMMYzro89mvp', # Argentina
48     '3719dQZEVXBjMXbn3EUUhlg', # Brasil
49     '3719dQZEVXBj03qyFxbk0E1', # México
50     '3719dQZEVXBjKj23U1GF4IR' # Canadá
51 ]
52 # Data list
53 canciones_data = []
54
55 for playlist_id in playlist_ids:
56     # Top 10 tracks
57     results = sp.playlist_tracks(playlist_id, limit=10)
58
59     for track in results['items']:
60         track_name = track['track']['name']
61         artist_name = track['track']['artists'][0]['name']
62         track_id = track['track']['id']
63         playlist = get_playlist_name(playlist_id)
64         # Audio analysis
65         analysis = sp.audio_analysis(track_id)
66         features = sp.audio_features([track_id])[0]
67         unique_id = str(uuid.uuid4())
68         tempo = Decimal(str(analysis['track']['tempo']))
69         dance = Decimal(str(features['danceability']))
70         acoustic = Decimal(str(features['acousticness']))
71         energy = Decimal(str(features['energy']))
72         valence = Decimal(str(features['valence']))
73
74         cancion_data = {
75             'id': unique_id,
76             'cancion': track_name,
77             'Artista': artist_name,
78             'Playlist': playlist,
79             'Tempo': tempo,
80             'Danceability': dance,
81             'Acousticness': acoustic,
82             'Energy': energy,
83             'Valence': valence
84         }
85         canciones_data.append(cancion_data)
86
87     # Insert data DynamoDB
88     for cancion_data in canciones_data:
89         table.put_item(Item=cancion_data)
90
91     return {
92         'statusCode': 200,
93         'body': 'Canciones agregadas a DynamoDB exitosamente'
94     }
95

```

2. Código de la Función Lambda extract-DB-to-S3:

```
1 import json
2 import boto3
3 import datetime
4
5 def lambda_handler(event, context):
6     s3 = boto3.client('s3')
7     bucket_name = 'dynamodb-exportt'
8
9     objects = s3.list_objects_v2(Bucket=bucket_name)
10
11     # Ordenarlos por fecha (mas recientes a menos)
12     sorted_objects = sorted(objects['Contents'], key=lambda x: x['LastModified'], reverse=True)
13
14     if sorted_objects:
15         # Nombre del ultimo archivo
16         file_name = sorted_objects[0]['Key']
17
18         # Obtener los datos
19         obj = s3.get_object(Bucket=bucket_name, Key=file_name)
20         data = json.loads(obj['Body'].read())
21
22         return {
23             'statusCode': 200,
24             'headers': {
25                 'Content-Type': 'application/json'
26             },
27             'body': json.dumps(data),
28             'file_name': file_name
29         }
30     else:
31         return {
32             'statusCode': 404,
33             'body': 'No se ha encontrado ningún archivo JSON'
34         }
```

3. Código de la Función Lambda export-jsonS3-to-URL:

```
1 import json
2 import boto3
3 import datetime
4
5 def lambda_handler(event, context):
6
7     dynamodb_table_name = "spotify_data"
8     s3_bucket_name = "dynamodb-exportt"
9     export_file_name = f"export-{datetime.datetime.now().isoformat()}.json"
10
11     # Client config
12     dynamodb = boto3.client('dynamodb')
13     s3 = boto3.client('s3')
14
15     response = dynamodb.scan(Table_name=dynamodb_table_name)
16     data = json.dumps(response['Items'], default=str)
17     s3.put_object(Bucket=s3_bucket_name, Key=export_file_name, Body=data)
18
19     return {
20         'statusCode': 200,
21         'body': json.dumps('Exportacion exitosa')
22     }
23
```