Spotifire

GRAN ESCALA – PRIMAVERA 2025: PROYECTO FINAL

SPOTIFIRE

Integrantes:

- Rodrigo Alan García Pérez
- Emanuel Enrique Ortiz Bassoco
- Ulises Reyes García

ARQUITECTURA DEL PRODUCTO DE DATOS SPOTIFIRE

Spotifire es una plataforma de analytics musical que recopila, procesa y analiza datos de escucha de Spotify para generar insights personalizados sobre los hábitos musicales de los usuarios. La arquitectura está diseñada como un sistema distribuido en el servicio de nube de AWS con recolección automática de datos, procesamiento ETL y análisis en tiempo real.

Componentes Principales

1. Capa de Presentación (Frontend)

Aplicación Web Flask

- **Ubicación**: EC2 instance
- **Puerto**: 8080 (HTTPS con certificado SSL)
- **Tecnologías**: Flask, Jinja2, Bootstrap, Chart.js
- **Funcionalidades**:
  - Dashboard interactivo con visualizaciones
  - Autenticación OAuth con Spotify
  - Gestión de sesiones de usuario
  - API endpoints para datos en tiempo real

2. Capa de Aplicación (Backend)

Servidor Web Flask

- **Estructura Modular**: Blueprints para organización
- **Blueprints**:
  - auth_bp: Manejo de autenticación y OAuth
  - dashboard_bp: Lógica del dashboard y APIs

Servicios Backend:

- **Spotify Service**: Integración con Spotify Web API
- **Athena Service**: Consultas analíticas a AWS Athena
- **Token Management**: Gestión y renovación de tokens OAuth

3. Capa de Recolección de Datos

Spotifire

Recolectores Automáticos

```
scripts/
├── spotify_periodic_collector.py    # Datos de reproducción reciente
├── update_history.py               # Datos históricos (likes, follows, tops)
└── spotify_s3_uploader.py         # Sincronización con S3
```

Servicios Systemd (Programación Automática):

- **spotify-collector.service**: Recolección cada 4 horas
- **spotify-s3-uploader.service**: Subida a S3 una hora después de recolección
- **spotify-historic-collector.service:** Recoleccion de datos historicos semanalmente

Datos Recolectados:

- **Recently Played**: Canciones reproducidas recientemente (CSV)
- **Liked Tracks**: Canciones marcadas como "Me Gusta" (JSON)
- **Top Tracks**: Canciones más escuchadas (JSON)
- **Followed Artists**: Artistas seguidos (JSON)

4. Capa de Almacenamiento

Amazon S3 (Data Lake)

```
s3://itam-analytics-ragp/
├── spotifire/
│   ├── raw/                    # Datos crudos por usuario
│   │   └── {user_id}/
│   │       ├── recently_played_*.csv
│   │       ├── likes_list.json
│   │       ├── top_tracks.json
│   │       └── followed_artists.json
│   └── processed/              # Datos procesados (Parquet)
│       ├── individual/         # Datos de reproducción
│       ├── likes/              # Datos de likes
│       ├── top_tracks/         # Top tracks
│       └── followed_artists/   # Artistas seguidos
└── athena-results/             # Resultados temporales de Athena
```

Local Storage (EC2)

Spotifire

```
/home/ec2-user/spotifire/
├── data/
│   ├── users_data/          # Tokens de usuarios (JSON)
│   ├── collected_data/      # Datos recolectados localmente
│   └── sessions/            # Sesiones web
└── logs/                    # Logs de aplicación
```

5. Capa de Procesamiento ETL

AWS Glue Jobs

- **spotify_etl_job.py**: Procesa datos de reproducción (CSV → Parquet)
- **etl_data_historica.py**: Procesa datos históricos (JSON → Parquet)

Transformaciones ETL:

1. **Limpieza de Datos**: Manejo de nulos, duplicados
2. **Enriquecimiento**: Campos derivados (hora, día de semana, temporada)
3. **Normalización**: Conversión de tipos de datos
4. **Particionamiento**: Organización por usuario
5. **Compresión**: Formato Parquet con compresión Snappy

AWS Glue Data Catalog

- **Database**: spotify_analytics
- **Tables**:
  - user_tracks: Datos de reproducción unificados
  - likes: Canciones con "Me Gusta"
  - top_tracks: Canciones más escuchadas
  - followed_artists: Artistas seguidos

6. Capa de Analytics

AWS Athena

- **Consultas SQL**: Análisis de patrones de escucha
- **Insights Generados**:
  - Top artistas por usuario
  - Patrones de escucha por hora del día
  - Comparación semana vs fin de semana
  - Distribución de popularidad (emergentes vs establecidos)
  - Perfiles musicales personalizados

Spotifire

Diagrama de la arquitectura