

DOCUMENTACIÓN - RECOMMENDING MUSE

1. Introducción

El Recommending Muse es una aplicación web desarrollada como proyecto final para la asignatura de Web Semántica en el Politécnico Colombiano Jaime Isaza Cadavid. El sistema utiliza tecnologías semánticas (RDF y OWL) para representar conocimiento musical y proporciona recomendaciones personalizadas basadas en los gustos del usuario.

Desarrolladores:

- Jhon Kevin Murillo Martínez

Asignatura: Web Semántica y Agentes Inteligentes

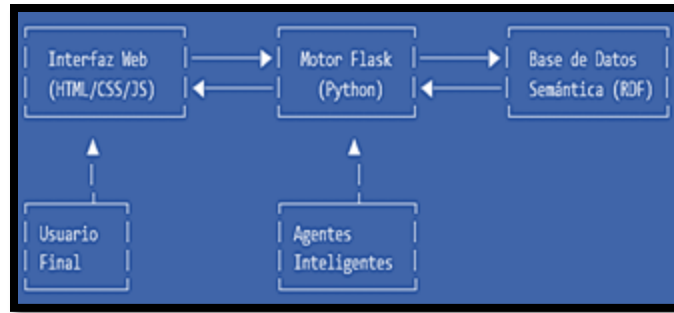
Institución: Politécnico Colombiano Jaime Isaza Cadavid

Año: 2025

2. Objetivos del Proyecto

- Implementar una ontología musical para representar relaciones entre usuarios, canciones y géneros
- Desarrollar un sistema de recomendación inteligente basado en preferencias
- Crear una interfaz intuitiva para la interacción usuario-sistema
- Resolver problemas comunes en sistemas de recomendación como:
 - Dominancia de géneros populares
 - Burbujas de filtro
 - Experiencia para nuevos usuarios

3. Arquitectura del Sistema



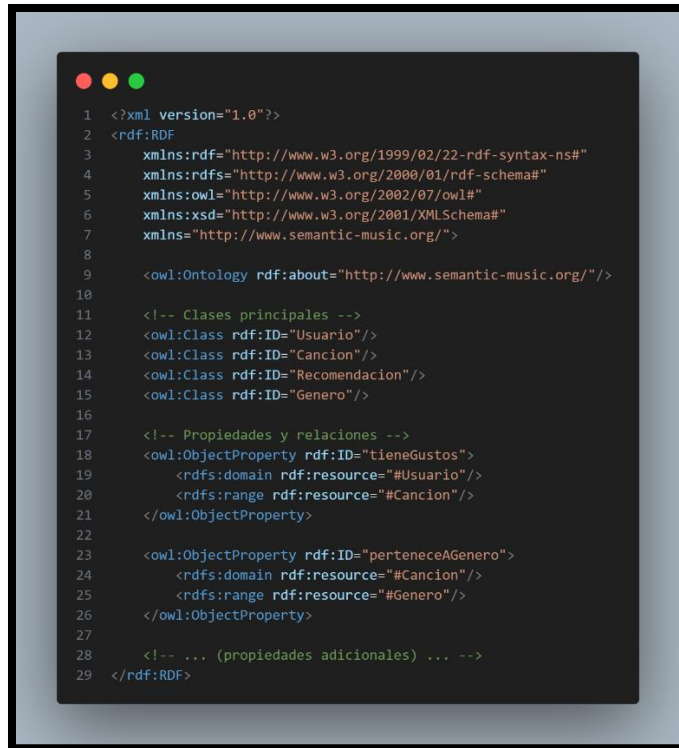
[Diagrama de Arquitectura]

4. Tecnologías Utilizadas

- Lenguajes: Python, HTML, CSS
- Frameworks: Flask (backend)
- Procesamiento Semántico: RDFLib (manejo de RDF/OWL)
- Almacenamiento: Archivos RDF/OWL
- IDE: Visual Studio Code

5. Ontología Musical

5.1. Estructura de la Ontología (musica.owl)



[Diagrama de Ontología]

5.2. Géneros Musicales Definidos

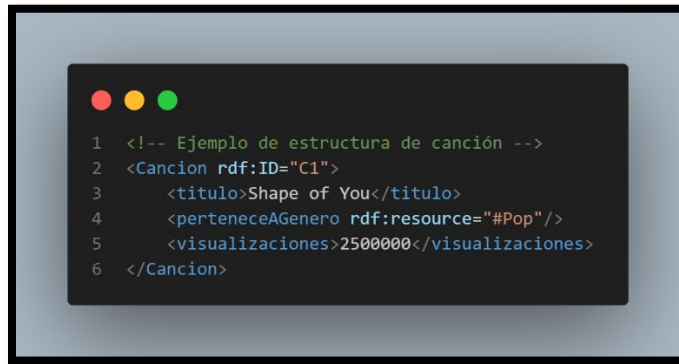
La ontología define 12 géneros musicales:

- Pop
- Rock
- Electronica
- Jazz
- Hip_Hop
- Metal
- Reggaeton
- Salsa
- Country

- Blues
- K_Pop
- Indie

6. Base de Datos Semántica (canciones.rdf)

La base de datos contiene 120 canciones (10 por género) con sus metadatos:



[Estructura de datos RDF]

7. Componentes del Sistema

7.1. app.py (Backend Principal)

El corazón del sistema implementa:

- Carga y procesamiento de datos RDF
- Gestión de preferencias de usuarios
- Algoritmo de recomendaciones
- Rutas y controladores Flask

```
1 # Configuración de namespaces
2 MUSIC_NS = Namespace("http://www.semantic-music.org/")
3
4 # Almacenamiento en memoria de gustos de usuarios
5 user_preferences = {}
6
7 def load_songs():
8     """Carga canciones desde archivo RDF"""
9     try:
10         g = Graph()
11         g.parse("canciones.rdf")
12         # ... (procesamiento) ...
13     except Exception as e:
14         print(f"Error: {str(e)}")
```

[Parte del Corazón del Sistema]

7.2. Algoritmo de Recomendaciones

El sistema implementa un algoritmo híbrido que combina:

- Filtrado por contenido (basado en géneros)
- Sistema de descubrimiento (nuevos géneros)
- Popularidad global

7.2.1. Prevención de Dominancia de Géneros

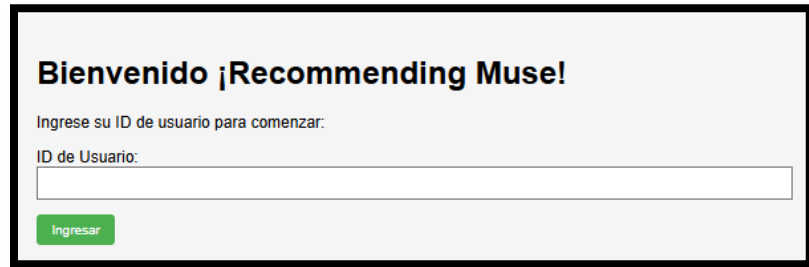
Para evitar que géneros populares dominen las recomendaciones:

```
1 # Lógica para limitar representación por género
2 genre_limits = {genre: 0 for genre in liked_genres}
3
4 for song in candidate_songs:
5     if genre_limits[song['genre']] < 2: # Máximo 2 canciones por género
6         main_recommendations.append(song)
7         genre_limits[song['genre']] += 1
8         if len(main_recommendations) >= 5:
9             break
```

[Diagrama de algoritmo de recomendaciones]

7.3. Interfaz de Usuario

7.3.1. index.html (Página de Inicio)



Bienvenido ¡Recommending Muse!

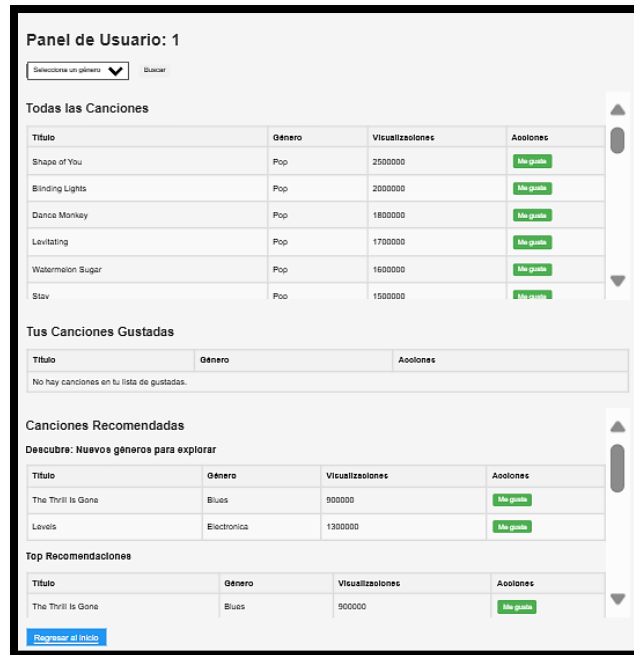
Ingrese su ID de usuario para comenzar.

ID de Usuario:

[Ingresar](#)

[Captura de pantalla de index.html]

7.3.2. dashboard.html (Panel Principal)



Panel de Usuario: 1

Selecciona un género

Todas las Canciones

Título	Género	Visualizaciones	Acciones
Shape of You	Pop	2500000	Me gusta
Blinding Lights	Pop	2000000	Me gusta
Dance Monkey	Pop	1800000	Me gusta
Levitating	Pop	1700000	Me gusta
Watermelon Sugar	Pop	1600000	Me gusta
Stay	Pop	1500000	Me gusta

Tus Canciones Gustadas

Título	Género	Acciones
No hay canciones en tu lista de gustadas.		

Canciones Recomendadas

Descubre: Nuevos géneros para explorar

Título	Género	Visualizaciones	Acciones
The Thrill Is Gone	Blues	900000	Me gusta
Levels	Electronica	1300000	Me gusta

Top Recomendaciones

Título	Género	Visualizaciones	Acciones
The Thrill Is Gone	Blues	900000	Me gusta

[Regresar al inicio](#)

[Captura de pantalla de dashboard.html]

Contiene tres secciones principales:

- Todas las canciones: Lista completa filtrable por género

- Tus canciones gustadas: Música que el usuario ha marcado como "Me gusta"
- Canciones recomendadas: Sugerencias personalizadas

8. Flujo de Trabajo del Sistema

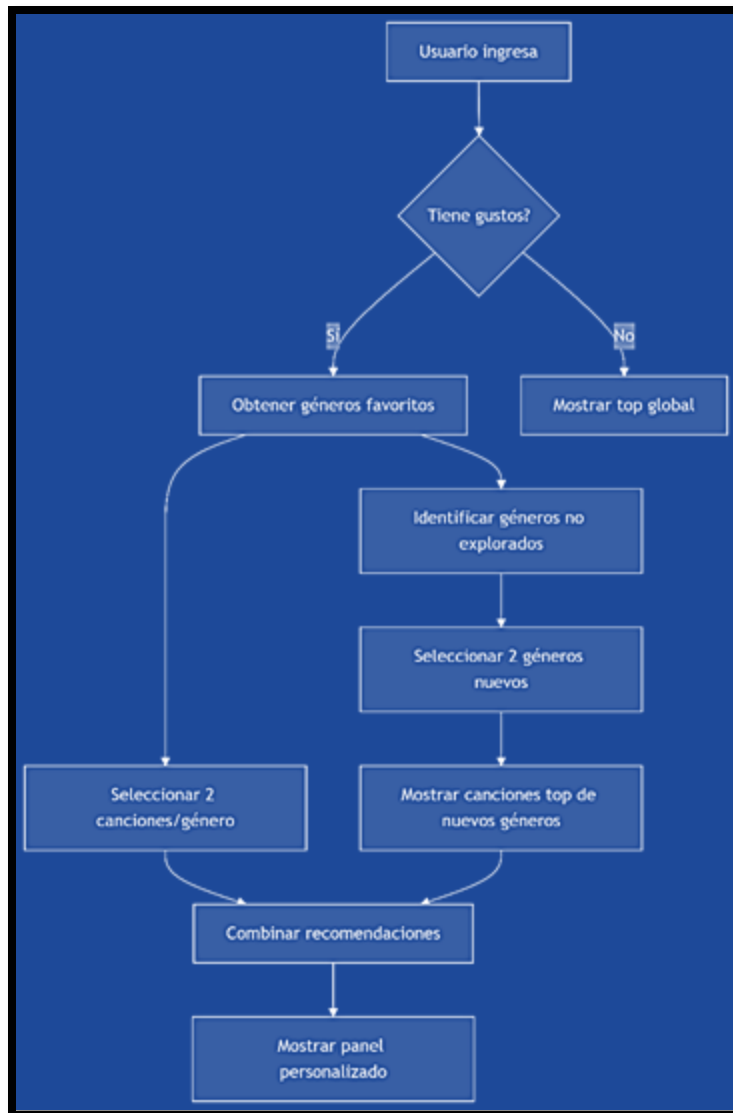
8.1. Para Nuevos Usuarios

1. Ingresa ID de usuario
2. Recibe recomendaciones basadas en popularidad global
3. Explora géneros diversos en "Descubre nuevos géneros"

8.2. Para Usuarios Registrados

1. Sistema aprende de sus "Me gusta"
2. Genera recomendaciones basadas en sus géneros favoritos
3. Sugiere géneros relacionados para descubrimiento

8.3. Proceso de Recomendación



[Diagrama de Flujo de Recomendación]

9. Problemas Resueltos y Soluciones Implementadas

9.1. Dominancia de Géneros Populares

Problema: Los géneros con más visualizaciones dominaban las recomendaciones, ocultando géneros menos populares pero relevantes para el usuario.

Solución: Implementación de límite por género (máximo 2 canciones por género en recomendaciones principales).

9.2. Burbuja de Filtro

Problema: Los usuarios quedaban atrapados en recomendaciones de los mismos géneros.

Solución: Sistema de descubrimiento que introduce géneros no explorados.

9.3. Experiencia para Nuevos Usuarios

Problema: Sin historial de gustos, no se podían generar recomendaciones personalizadas.

Solución: Sistema de recomendación basado en popularidad global y diversificación de géneros.

9.4. Manejo de Datos Inconsistentes

Problema: Errores en archivos RDF detenían la aplicación.

Solución: Implementación de manejo robusto de excepciones:

A screenshot of a code editor window with a dark background and light-colored text. The code is written in Python and shows a function named 'load_songs()' with a try-except block. The try block contains a comment '# Intento de carga'. The first except block catches 'ParserError' and 'BadSyntax' as 'e', prints an error message with a red 'X' icon, and returns an empty list. The second except block catches 'OSError' as 'e', prints another error message with a red 'X' icon, and returns an empty list. The code ends with a comment '# ... (más excepciones) ...'.

```
1 def load_songs():
2     try:
3         # Intento de carga
4     except (ParserError, BadSyntax) as e:
5         print(f"❌ Error de sintaxis: {str(e)}")
6         return []
7     except OSError as e:
8         print(f"❌ Error de archivo: {str(e)}")
9         return []
10    # ... (más excepciones) ...
```

[Parte de la Lógica de Excepciones]

10. Manual de Instalación y Uso

10.1. Requisitos del Sistema

- Python 3.8+
- pip (gestor de paquetes)
- Navegador web moderno

10.2. Instalación

- **Clonar repositorio:** git clone https://github.com/UJohnnyU/Recommending_Muse.git
- **Instalar Dependencias:** pip install Flask, rdflib

10.3. Ejecución

- python app.py
- **Acceder en navegador:** <http://localhost:5000>

11. Pruebas y Validación

Se realizaron pruebas con diversos perfiles de usuario:

1. **Usuario nuevo:** Verificar diversidad de recomendaciones
2. **Usuario con gustos definidos:** Validar personalización
3. **Usuario explorador:** Comprobar sistema de descubrimiento

Métricas de evaluación:

- Precisión de recomendaciones
- Diversidad de géneros sugeridos
- Tiempo de respuesta
- Satisfacción del usuario (pruebas cualitativas)

12. Conclusiones

- Los sistemas basados en conocimiento semántico permiten recomendaciones más explicables
- La ontología musical proporciona una base sólida para relaciones complejas

- El equilibrio entre personalización y descubrimiento es clave para la experiencia de usuario
- La prevención de dominancia de géneros mejora significativamente la diversidad

Anexos:

Bienvenido ¡Recommending Muse!

Ingrese su ID de usuario para comenzar:

ID de Usuario:

Ingresar

Panel de Usuario: 1

Todas las Canciones

Título	Género	Visualizaciones	Acciones
God's Plan	Hip_Hop	1600000	<input type="button" value="Me gusta"/>
Humble	Hip_Hop	1400000	<input type="button" value="Me gusta"/>
Lose Yourself	Hip_Hop	1200000	<input type="button" value="Me gusta"/>
In Da Club	Hip_Hop	1000000	<input type="button" value="Me gusta"/>
Hotline Bling	Hip_Hop	900000	<input type="button" value="Me gusta"/>
Money Trees	Hip_Hop	800000	<input type="button" value="Me gusta"/>

Tus Canciones Gustadas

Título	Género	Acciones
Dance Monkey	Pop	<input type="button" value="No me gusta"/>
Sicko Mode	Hip_Hop	<input type="button" value="No me gusta"/>

Canciones Recomendadas

Para ti: Basado en tus gustos

Título	Género	Visualizaciones	Acciones
Shape of You	Pop	2500000	Me gusta
Blinding Lights	Pop	2000000	Me gusta
God's Plan	Hip_Hop	1600000	Me gusta
Humble	Hip_Hop	1400000	Me gusta

Descubre: Nuevos géneros para explorar

Título	Género	Visualizaciones	Acciones
Old Town Road	Country	1300000	Me gusta
Tennessee Whiskey	Country	1100000	Me gusta
Dakiti	Reggaeton	1500000	Me gusta
Con Calma	Reggaeton	1300000	Me gusta

Top Recomendaciones

Título	Género	Visualizaciones	Acciones
Shape of You	Pop	2500000	Me gusta
Blinding Lights	Pop	2000000	Me gusta
God's Plan	Hip_Hop	1600000	Me gusta
Humble	Hip_Hop	1400000	Me gusta
Old Town Road	Country	1300000	Me gusta
Tennessee Whiskey	Country	1100000	Me gusta

[Regresar al inicio](#)