Sistema de Clasificación Automatizada de Música Libre – Proyecto DAMIAN

* 1. DEFINICIÓN DEL PROBLEMA Y REQUERIMIENTOS

**FASE INICIO**

* 1.1. Modelado del Proyecto
* 1.1.1 Presentación breve del sistema
* 1.1.2 Propósito del proyecto
* 1.2. Delimitación del sistema
* 1.2.1 Público beneficiado con el sistema
* 1.2.2 Descripción de los procesos de negocio actuales
* 1.2.3 Diagrama de Actividades (a completar)
* 1.3. Requerimientos del Sistema de Información
* 1.3.1 Objetivos del nuevo sistema
* 1.3.2 Requerimientos funcionales y no funcionales
* 1.3.3 Arquitectura general del sistema (Diagrama de bloques)
* 2. DOCUMENTACIÓN FUNCIONAL
* 2.1 Documento Visión del sistema
* 2.2 Modelo de Casos de Uso (general)
* 2.2.1 Diagrama de Casos de Uso (a completar)
* 2.2.2 Narración de los Casos de Uso
* 2.3 Glosario del sistema
* 2.4 Especificaciones complementarias
* 3. DISEÑO ARQUITECTÓNICO Y LÓGICO
* 3.1 Modelo de dominio del problema
* 3.2 Diagrama de Clases (conceptual y de diseño)
* 3.3 Diagrama de Paquetes / Capas
* 3.4 Diagrama de Entidad-Relación
* 4. DISEÑO DE INTERFAZ Y COMPORTAMIENTO
* 4.1 Prototipos de interfaz de usuario (básico)
* 4.2 Diagrama de Estados (si aplica)
* 4.3 Diagrama de Secuencia (descarga y clasificación)
* 5. IMPLEMENTACIÓN Y PRUEBAS
* 5.1 Lenguaje y librerías utilizadas
* 5.2 Scripts y ejecución (resumen del código)
* 5.3 Casos de prueba y resultados esperados
* 6. ANEXOS
* 6.1 Capturas de ejecución
* 6.2 Logs / directorios generados
* 6.3 Enlace a repositorio (si aplica)

# Proyecto: Descargador de Música Libre y Clasificador por Voz

## 🌐 Resumen General

Este sistema automatizado permite buscar, descargar y clasificar canciones libres de derechos desde sitios como Free Music Archive (FMA). Su objetivo principal es ayudar a creadores de contenido a construir una librería musical sin copyright, separando canciones con voz de instrumentales.

## ⚙️ Tecnologías utilizadas

\* \*\*Lenguaje\*\*: Python 3.11+

\* \*\*Librerías\*\*:

\* `requests`: Descarga de datos y MP3 desde la web

\* `beautifulsoup4`: Extracción de enlaces HTML

\* `pydub`: Análisis básico de audio (detección de voz)

\* \*\*Dependencia externa\*\*:

\* `ffmpeg`: Requerido por `pydub` para manejar archivos MP3

## 📚 Estructura de Carpetas Generadas

```

/musica\_descargada/

├── musica\_con\_voz/

├── musica\_sin\_voz/

```

## 📆 Flujo del sistema (UML simplificado)

```mermaid

flowchart TD

A[Inicio] --> B[Descargar HTML de FMA]

B --> C[Extraer enlaces de canciones]

C --> D[Descargar archivo MP3]

D --> E[Detectar si tiene voz con PyDub]

E --> F{Tiene voz?}

F -- Sí --> G[Guardar en /musica\_con\_voz]

F -- No --> H[Guardar en /musica\_sin\_voz]

G & H --> I[Repetir hasta N canciones]

I --> Z[Fin]

```

## 👁️ Módulos del sistema

### 1. descarga\\_pagina(url)

Descarga el HTML de una página web (con timeout de 10 segundos).

### 2. detectar\\_voz(ruta\\_audio)

Usa `pydub` para analizar un fragmento del audio. Si el nivel dBFS está por encima de -35dB, se considera que hay voz.

### 3. descargar\\_y\\_guardar(url, nombre, carpeta)

Descarga un archivo desde un enlace directo y lo guarda con nombre seguro.

### 4. extraer\\_canciones()

Función principal: recorre las canciones, descarga cada una, detecta la voz y la clasifica.

## 🌐 URL actual utilizada

[https://freemusicarchive.org/genre/Pop/](https://freemusicarchive.org/genre/Pop/)

## 📝 Futuras mejoras

\* Soporte para múltiples géneros (entrada por consola o menú).

\* Integración con otros sitios (NCS, FreeToUseMusic).

\* Detección de voz con machine learning (como Whisper).

\* Exportación de metadatos a CSV o base de datos.

\* Interfaz gráfica (Tkinter o Web).

## ✅ Requisitos de instalación

```bash

pip install requests beautifulsoup4 pydub

# Y asegurarse de tener ffmpeg instalado y en PATH

```

---

📅 \*\*Versión:\*\* 1.0

📄 \*\*Autor:\*\* Damian Juarez

🎓 \*\*Licencia:\*\* Uso educativo y libre distribución