



Documentación Técnica

MuuuSIC

Juan Pablo Salas González
Asael Victoriano Martínez
Paola Monserrat Witrigo Castro

Índice

Documento de Visión / Requerimientos.....	3
Objetivo del sistema.....	3
Usuarios.....	3
Requerimientos funcionales.....	3
Cronograma final de actividades realizadas en los sprints (planeado vs real).....	19
Alcance final del producto.....	20
Costo total del desarrollo.....	21
Arquitectura.....	22
Diagrama general.....	22
Tecnologías utilizadas.....	23
Diseño de la base de datos (ERD o modelo lógico).....	24
Estructura de módulos.....	25
UI/UX.....	28
Wireframes.....	28
Flujo de pantallas.....	31
Estilo visual y decisiones de diseño.....	32
Bitácora de Cambios.....	33
Versiones.....	33
Funciones agregadas, modificadas o descartadas.....	34
Registro de decisiones técnicas.....	34

Documento de Visión / Requerimientos

Objetivo del sistema

Desarrollar una aplicación web llamada “MuuuSIC” que proporcione un espacio para los amantes de la música donde puedan reseñar, calificar y compartir sus opiniones sobre canciones y álbumes, llenando el vacío actual de plataformas de este tipo (similar a un “Letterboxd para música” pero con el reproductor de música incluido).

Usuarios

- **Usuario:** Un usuario es alguien que utiliza el sistema para reseñar música, interactuar con reseñas, explorar la música que se encuentra en el sitio y gestionar sus playlists.
- **Artista:** Un artista es un rol que utiliza el sistema para gestionar su música y los permisos de esta.

Requerimientos funcionales

1. **Caso de uso:** Explorar Música
 - 1.1. **Actores:** Usuario (Principal)
 - 1.2. **Resumen:** Permite a un usuario buscar, navegar, compartir, reproducir y descargar música, visualizando contenido personalizado basado en los géneros preferidos del usuario y las sugerencias que proporciona el sistema.
 - 1.3. **Precondiciones:**
 - El usuario debe tener una sesión activa en el sistema.
 - La base de datos debe contener registros de canciones y álbumes disponibles.
 - El navegador del usuario debe soportar reproducción de audio HTML5.
 - El usuario debe haber completado la selección inicial de 3 géneros musicales.

1.4. **Flujo Principal (Happy Path):**

1. El Usuario accede al módulo "Explorar".
2. El sistema muestra una lista de reproducción con las 10 canciones "Top" basadas en los géneros seleccionados por el usuario.
3. El sistema muestra adicionalmente todas las playlists y álbumes compartidos por otros usuarios.
4. El Usuario selecciona una canción, álbum o playlist para ver su contenido.
5. El Usuario presiona el botón de "Reproducir".
6. El sistema carga el audio en la barra de reproducción global (base_player) y comienza la emisión del sonido.
7. El Usuario visualiza los controles de pausa, siguiente, atrás y volumen en la barra inferior.

1.5. **Flujos Alternativos:**

1.5.A. Búsqueda de Música: En el paso 2, el usuario utiliza la barra de búsqueda para encontrar una canción o artista específico. El sistema actualiza la vista con los resultados coincidentes.

1.5.B. Descarga de Música (Offline): En el paso 4, el usuario presiona el botón de descarga. El sistema guarda la canción automáticamente en la sección "Música sin conexión" del perfil del usuario.

1.5.C. Encolar Canción: El usuario presiona el botón para encolar una canción. El sistema añade la pista al final de la lista de reproducción actual sin interrumpir la canción en curso.

1.6. **Postcondiciones:**

- La canción seleccionada se está reproduciendo en el reproductor global, persistiendo aunque el usuario cambie de pantalla.
- Si se realizó una descarga, el ítem está accesible en la vista `musica_offline.html`.
- La cola de canciones se actualiza con la nueva selección.

1.7. Reglas de Negocio:

RN1: La lista de recomendación inicial ("Top 10") debe filtrarse estrictamente por los 3 géneros elegidos por el usuario durante el registro.

RN2: La funcionalidad de descarga depende de los permisos otorgados por el Artista propietario del álbum; si el artista deshabilitó las descargas, el botón no estará activo.

RN3: La reproducción debe ser funcional mediante la API de Jamendo.

RN4: El reproductor debe mantenerse visible y funcional en todas las vistas de la plataforma.

1.8. Requisitos Especiales:

- Conexión a Internet estable para el consumo de la API de Jamendo (streaming).
- Interfaz responsiva que permita expandir la barra de reproducción a pantalla completa para visualizar reseñas.

1.9. Frecuencia de Uso:

Alta (Cada vez que el usuario ingresa a la plataforma para escuchar música).

2. Caso de uso: Reseñar Música

2.1. **Actores:** Usuario (Principal).

2.2. **Resumen:** Permite al usuario expresar su opinión y sentimientos sobre una canción o álbum específico mediante una calificación numérica (estrellas) y un comentario de texto opcional. Las reseñas quedan asociadas tanto al contenido musical como al perfil del usuario.

2.3. **Precondiciones:**

- El usuario debe tener una sesión activa en el sistema.
- El usuario debe haber accedido a la vista de reseñas en canción
- El ítem a reseñar canción debe existir en la base de datos.

2.4. **Flujo Principal (Happy Path):**

1. El Usuario visualiza la sección de reseñas dentro del detalle de una canción o en el reproductor.
2. El Usuario selecciona una calificación mediante un sistema de estrellas (escala del 1 al 5).
3. El Usuario escribe un comentario de texto explicando su opinión (Opcional).
4. El Usuario presiona el botón "Publicar Reseña".
5. El Sistema valida que la calificación haya sido asignada.
6. El Sistema guarda la reseña en la base de datos vinculándola al usuario y a la canción.
7. El Sistema actualiza la lista de reseñas visibles para mostrar la nueva entrada inmediatamente.

2.5. **Flujos Alternativos:**

2.5.A. Reseña sin Comentario: En el paso 3, el usuario omite escribir texto. El sistema permite continuar en el paso 4 ya que el comentario no es obligatorio.

2.6. **Postcondiciones:**

- La reseña es visible públicamente en el apartado de la canción correspondiente.
- La reseña se agrega a la lista de "Reseñas hechas" en el perfil personal del usuario.

2.7. **Reglas de Negocio:**

RN1: La calificación es obligatoria y debe ser un valor entero entre 1 y 5 estrellas.

RN2: Solo se permite reseñar canciones, no aplica para *playlists* de otros usuarios.

RN3: Los comentarios de texto no son obligatorios para enviar una calificación.

2.8. Requisitos Especiales:

La interfaz de estrellas debe ser interactiva y visualmente clara.

2.9. Frecuencia de Uso:

Media (Depende de la voluntad del usuario de compartir su opinión tras escuchar música).

3. Caso de uso: Interactuar con Reseñas

3.1. Actores: Usuario (Principal).

3.2. Resumen: Permite a los usuarios interactuar socialmente dentro de la plataforma expresando su acuerdo o desacuerdo con las opiniones de otros. Los usuarios pueden reaccionar mediante "likes" (me gusta) o "dislikes" (no me gusta) a las reseñas públicas encontradas en canciones.

3.3. Precondiciones:

- El usuario debe tener una sesión activa en el sistema.
- Debe existir al menos una reseña publicada por otro usuario en la canción o álbum que se está visualizando.

3.4. Flujo Principal (Happy Path):

1. El Usuario accede a la vista de reseñas.
2. El Sistema muestra el panel de reseñas con los comentarios de la comunidad.
3. El Usuario lee una reseña de otro usuario.
4. El Usuario presiona el botón de "Like" (o "Dislike") en la reseña seleccionada.
5. El Sistema registra la reacción.
6. El Sistema actualiza visualmente el contador de reacciones de esa reseña para reflejar el cambio.

3.5. Flujos Alternativos:

3.5.A. Quitar Reacción: El usuario presiona nuevamente el botón de reacción que ya había seleccionado previamente. El sistema elimina la reacción y decrementa el contador.

3.5.B. Cambiar Reacción: El usuario que ya había dado "Like" presiona "Dislike". El sistema elimina el "Like" y suma el "Dislike" (o viceversa).

3.5.C. Visualización Propia: El usuario visualiza sus propias reseñas. El sistema permite dar "Like" a sí mismo o "Dislike".

3.6. Postcondiciones:

- El contador de reacciones de la reseña queda actualizado en la base de datos.
- La interacción es visible para otros usuarios que consulten la misma reseña.

3.7. Reglas de Negocio:

RN1: Los usuarios podrán dar tanto *likes* como *dislikes* a las reseñas que quieran.

RN2: La interacción se limita a las reseñas; no se "reaccionan" a las canciones directamente (eso se maneja con "Favoritos").

3.8. Requisitos Especiales:

Respuesta visual inmediata al hacer clic en los iconos de reacción para una buena experiencia de usuario.

3.9. Frecuencia de Uso:

Alta (Actividad común al leer opiniones de la comunidad).

4. Caso de uso: Gestionar Playlists

4.1. Actores: Usuario (Principal).

4.2. **Resumen:** Permite al usuario organizar su colección musical personal mediante la creación de listas de reproducción personalizadas. El usuario puede crear nuevas listas, configurar su privacidad, agregar o eliminar canciones y gestionar su lista automática de "Favoritos".

4.3. **Precondiciones:**

- El usuario debe tener una sesión activa en el sistema.
- Deben existir canciones registradas en la base de datos para poder agregarlas.

4.4. **Flujo Principal (Happy Path):**

1. El Usuario navega por la aplicación (Explorar) y selecciona la opción "Agregar a playlist (+)" en una canción específica.
2. El Usuario ingresa un título obligatorio para la lista o añade a una playlist creada.
3. El Sistema crea la playlist y confirma la acción.
4. El Sistema vincula la canción a la lista seleccionada.

4.5. **Flujos Alternativos:**

4.5.A. Gestión de Favoritos: El usuario marca el icono de "Corazón" o "Favorito" en cualquier canción. El sistema agrega automáticamente esta canción a la lista predeterminada "Mis Favoritos".

4.5.B. Eliminar canciones favoritas: El usuario entra a uno de sus favoritos, selecciona una canción y presiona nuevamente el icono "Corazón". El sistema quita la canción de la lista sin borrarla de la base de datos global.

4.5.C. Edición de Detalles: Después del paso 5 el usuario modifica el título, la imagen o la visibilidad (de pública a privada o viceversa) de una playlist existente.

4.6. **Postcondiciones:**

- La nueva playlist aparece listada en el perfil del usuario.

- Si la playlist es pública, es visible para otros usuarios en la sección de playlist recomendadas para otros usuarios.
- Las canciones agregadas persisten en la lista para futuras sesiones.

4.7. **Reglas de Negocio:**

RN1: Se pueden agregar tantas canciones como el usuario desee, pero **NO** se permite agregar entidades de tipo "Álbum" completas como un solo ítem dentro de una playlist.

RN2: Las playlists deben tener un título obligatorio.

RN3: Una vez definidos los géneros del usuario al inicio, estos influyen en las recomendaciones, pero el usuario es libre de agregar música de cualquier género a sus playlists.

4.8. **Requisitos Especiales:**

Capacidad de subir y procesar imágenes para las portadas de las playlists.

Actualización inmediata de la vista "Mis Playlists" tras la creación o edición.

4.9. **Frecuencia de Uso:**

Media (Depende de qué tan activo sea el usuario organizando su música).

5. **Caso de uso:** Gestionar Música

5.1. **Actores:** Artista (Principal).

5.2. **Resumen:** Permite a los usuarios con rol de "Artista" subir y administrar su catálogo musical en la plataforma. Esto incluye la creación de álbumes y la carga de las canciones asociadas a estos, asegurando que el contenido esté disponible para los demás usuarios.

5.3. **Precondiciones:**

- El usuario debe tener una sesión activa.
- El usuario debe tener el rol de Artista asignado (debe haber solicitado y obtenido este perfil).
- Los archivos de audio deben estar en un formato compatible.

5.4. **Flujo Principal (Happy Path):**

1. El Artista accede a su perfil y selecciona la opción de "Subir Música" o "Gestión de Artista".
2. El Sistema solicita los datos del nuevo álbum (título, portada, género). Y habilita/deshabilita las descargas.
3. El Artista ingresa la información del álbum.
4. El Artista carga los archivos de audio correspondientes a las canciones del álbum.
5. El Artista confirma la subida.
6. El Sistema valida que las canciones estén asociadas correctamente al álbum.
7. El Sistema guarda el álbum y las canciones en la base de datos.
8. El Sistema confirma que el contenido ha sido publicado exitosamente.

5.5. **Flujos Alternativos:**

1.5.A. Cancelación: El artista cancela el proceso de subida antes de confirmar. No se guardan cambios.

5.6. **Postcondiciones:**

- El nuevo álbum y sus canciones son visibles en la plataforma para la exploración y reproducción por parte de los usuarios "Usuario".
- El álbum aparece en el perfil del Artista.

5.7. **Reglas de Negocio:**

RN1: Los usuarios con perfil "Usuario" estándar **NO** pueden subir música; esta función es exclusiva para Artista.

RN2: No se permite subir canciones sueltas ("singles"); todas las canciones deben pertenecer obligatoriamente a un álbum.

RN3: Los artistas pueden subir una cantidad ilimitada de álbumes.

5.8. **Requisitos Especiales:**

Capacidad de almacenamiento para archivos de audio e imágenes de portada.

5.9. **Frecuencia de Uso:**

Baja-Media (Ocurre cada vez que el artista lanza nuevo material).

HU-01 – Registro de usuario

Como usuario, **quiero** crear una cuenta **para** poder acceder a la plataforma.

Criterios de aceptación

- CA1: El usuario debe ingresar correo, contraseña y nombre.
- CA2: Debe enviarse un correo de verificación.
- CA3: No se permite registrar correos duplicados.
- CA4: Si el usuario no tiene cuenta no puede acceder.

Prioridad: Alta

Story Points: 8

HU-02 – Selección de Género

Como usuario, **quiero** elegir mis géneros musicales **para** encontrar la música que me gusta.

Criterios de aceptación

- CA1: Se mostrará entre una selección de géneros.
- CA2: El usuario debe elegir 3 géneros.
- CA3: Se deberá de enviar la selección de géneros para poder ingresar a la pantalla principal.
- CA4: Una vez enviado los géneros, estos no van a poder ser cambiados.

Prioridad: Media

Story Points: 3

HU-03– Explorar música

Como usuario, **quiero** explorar qué playlist o álbumes hay en la plataforma **para** encontrar música nueva

Criterios de aceptación

- CA1: El usuario podrá explorar una playlist de 10 canciones basadas en los géneros que escogió.
- CA2: Se verán TODAS las playlist y álbumes compartidos por otros usuarios INDEPENDIENTE del género que haya escogido el usuario.
- CA3: El usuario al ingresar a las playlist y álbumes se podrá ver el contenido.
- CA4: El usuario podrá reproducir el contenido de la playlist o de los álbumes.
- CA5: Se podrá buscar la música que el usuario quiera mediante una barra de búsqueda.
- CA6: Se podrá descargar una canción, álbum o playlist, y esta se guarda automáticamente en el perfil del usuario en música sin conexión.

Prioridad: Alta

Story Points: 21

HU-04 – Reseñar

Como usuario, **quiero** poder reseñar las canciones **para** poder expresar mi opinión y mis sentimientos acerca de la música.

Criterios de aceptación

- CA1: Solo se puede reseñar en canciones.
- CA1: Agregar un panel donde se vean las reviews.
- CA2: Se podrá dar un rating con base en estrellas (1 - 5) a canciones.
- CA3: Los comentarios NO son obligatorios al calificar una canción.
- CA4: Los usuarios podrán dar likes o dislikes a las reseñas que quieran.

Prioridad: Media

Story Points: 8

HU-05– Perfil de usuario

Como usuario, **quiero** ingresar a mi perfil **para** poder personalizarlo.

Criterios de aceptación

- CA1: Se verán las reseñas hechas.
- CA2: Se podrán ver las canciones en favorito.
- CA3: Se podrá editar el nombre.
- CA4: Se podrá cambiar la foto de perfil.
- CA5: Se verá la lista de reproducciones que el usuarios ha creado.
- CA6: Se podrá agregar las redes sociales que quiera.
- CA7: Se puede añadir descripción del perfil, editar nombre/apellido/nombre de usuario.

Prioridad: Media

Story Points: 13

HU-06 – Crear playlist

Como usuario, **quiero** crear playlists **para** poder escuchar la música que me gusta

Criterios de aceptación

- CA1: Se deberá de poner un título a la playlist.
- CA2: Se podrá agregar tantas canciones como quiera el usuario pero NO álbumes.
- CA3: Se podrá poner una imagen a la playlist al crearla.
- CA4: Se podrá elegir la visibilidad de la playlist para los otros usuarios (público - privada).

Prioridad: Media

Story Points: 5

HU-07 – Barra de reproducción

Como usuario, **quiero** poder reproducir la música **para** poder escucharla.

Criterios de aceptación

- CA1: Tendrá un botón de play y pausa.
- CA2: Habrá un botón de siguiente y atrás.
- CA3: Habrá un botón que permite poner en aleatorio cualquier canción del álbum o playlist que esté escuchando el usuario.
- CA4: Si se pone en aleatorio y solo hay una canción en la lista, se pausa después de la canción escuchada.
- CA5: Se podrá modificar el volumen de la canción.

- CA6: Se verá un apartado donde este el nombre de la canción, su título y poder marcar esa canción en favoritos.
- CA7: Se podrá reseñar en la barra de reproducción la canción que se está escuchando.

Prioridad: Alta

Story Points: 13

HU-08– Compartir

Como usuario, **quiero** compartir mi música **para** poder mostrarlo a otras personas.

Criterios de aceptación

- CA1: Se tendrá que generar un código QR cuando el usuario quiera compartir el álbum o playlist.
- CA2: El código QR enviará al usuario a la canción compartida.
- CA3: También se puede compartir la canción por medio de un link.

Prioridad: Baja

Story Points: 3

HU-09 – Encolar Música

Como usuario, **quiero** poder encolar la música **para** poder escucharla en el orden que quiera.

Criterios de aceptación

- CA1: Se podrá mandar las canciones que quiera el usuario a su lista de reproducción actual mediante un botón encontrado en la canción.

- CA2: El orden en el que el usuario añada las canciones en su lista de reproducción es el orden en el que se escuchan las canciones (COLA).
- CA3: El usuario podrá eliminar sus canciones encoladas.

Prioridad: Media

Story Points: 8

HU-10– Subir musica

Como artista, **quiero** mis canciones o álbumes **para** compartir mi arte.

- CA1: Los usuarios NO podrán subir música, solo si el usuario es artista.
- CA2: Los artistas podrán subir tantos álbumes como quieran.
- CA3: No se podrá subir canciones si no están en un álbum.
- CA4: El artista puede habilitar o deshabilitar las descargas de las canciones.

Prioridad: Media

Story Points: 5

HU-11 – Perfil Artista

Como artista, **quiero** este perfil **para** compartir mi arte.

- CA1: Los artistas TIENEN que tener una cuenta de usuario para poder existir.
- CA2: Si eres artista SOLO podrás subir álbumes.
- CA3: En su perfil tendrán un identificador que diga si un usuario es artista o no.

Prioridad: Media

Story Points: 5

Requerimientos no funcionales

1. Interfaz y Usabilidad (UI/UX)

- Diseño Moderno: El sistema debe contar con una interfaz moderna y un diseño lo más limpio posible.
- Consistencia Visual: Se utilizarán *Django Templates* para mantener una consistencia visual en todas las pantallas, separando la lógica del *frontend* en módulos estáticos.
- Persistencia del Reproductor: El reproductor de música debe ser global, manteniéndose visible y funcional sin interrupciones mientras el usuario navega entre diferentes vistas.

2. Rendimiento y Almacenamiento

- Optimización de Almacenamiento: El sistema no almacenará archivos de audio pesados en el servidor local; utilizará la API de Jamendo para el *streaming* y catálogo, reduciendo la complejidad de almacenamiento.
- Velocidad de Renderizado: Se prioriza el renderizado del lado del servidor (*Server-Side Rendering*) con Django para asegurar velocidad en el desarrollo y la carga de páginas.

3. Restricciones Técnicas y Compatibilidad

- Navegador: El navegador del cliente debe soportar obligatoriamente la reproducción de audio bajo el estándar HTML5.
- Conectividad: Se requiere una conexión a Internet estable para el consumo de la API de Jamendo y la reproducción de música en *streaming*.
- Base de Datos: El sistema debe operar utilizando MySQL como gestor de base de datos.
- Stack Tecnológico: El desarrollo está basado en el uso de Django (Backend) y JS/HTML/CSS (Frontend).

4. Seguridad y Roles

- Unicidad de Cuentas: El sistema debe impedir el registro de correos electrónicos duplicados.
- Gestión de Permisos: El sistema debe diferenciar estrictamente entre los roles de "Usuario" y "Artista".
- Verificación: El sistema debe requerir una sesión activa para acceder a las funcionalidades principales (exploración, reproducción, reseñas).

Cronograma final de actividades realizadas en los sprints (planeado vs real)

Planeado	Logrado
1. Terminar las primeras 2 historias del backlog.	1. Empezar con la primera parte de la 3ra historia.
2. Empezar con la primera parte de la 3ra historia.	2. Crear, insertar y unir todo con la base de datos.
3. Crear, insertar y unir todo con la base de datos.	3. Crear imágenes para los Géneros.
4. Crear imágenes para los Géneros.	4. Dejar funcional playlist, favoritos y álbumes.
5. Unir las 2 historias entre sí.	5. Poder descargar canciones y álbumes en música sin conexión.
6. Terminar las historias 3 y 4 del backlog.	6. Poder personalizar el perfil del usuario.
7. Empezar con algunas partes de la 5ta historia.	7. Barra de reproducción funcional.
8. Dejar funcional la personalización del perfil del usuario.	8. Crear playlist funcional.
9. Dejar funcional playlist, favoritos y álbumes.	9. Perfil de usuario a medias: (Nos faltó el poder ver las reseñas hechas por el usuario).
10. Poder descargar canciones y álbumes en música sin conexión.	10. Reseñas Funcionales.
11. Poder personalizar el perfil del usuario.	11. Compartir música.
12. Barra de reproducción funcional.	12. Poder encolar muisca.
	13. Crear un usuario artista.
	14. Poder subir álbumes SOLO si eres

13. Crear playlist funcional. 14. Poder compartir música. 15. Perfil de usuario funcional. 16. Reseñas funcionales. 17. Compartir música. 18. Poder encolar música. 19. Crear un usuario artista. 20. Poder subir álbumes SOLO si eres artista.	artista.
--	----------

Alcance final del producto

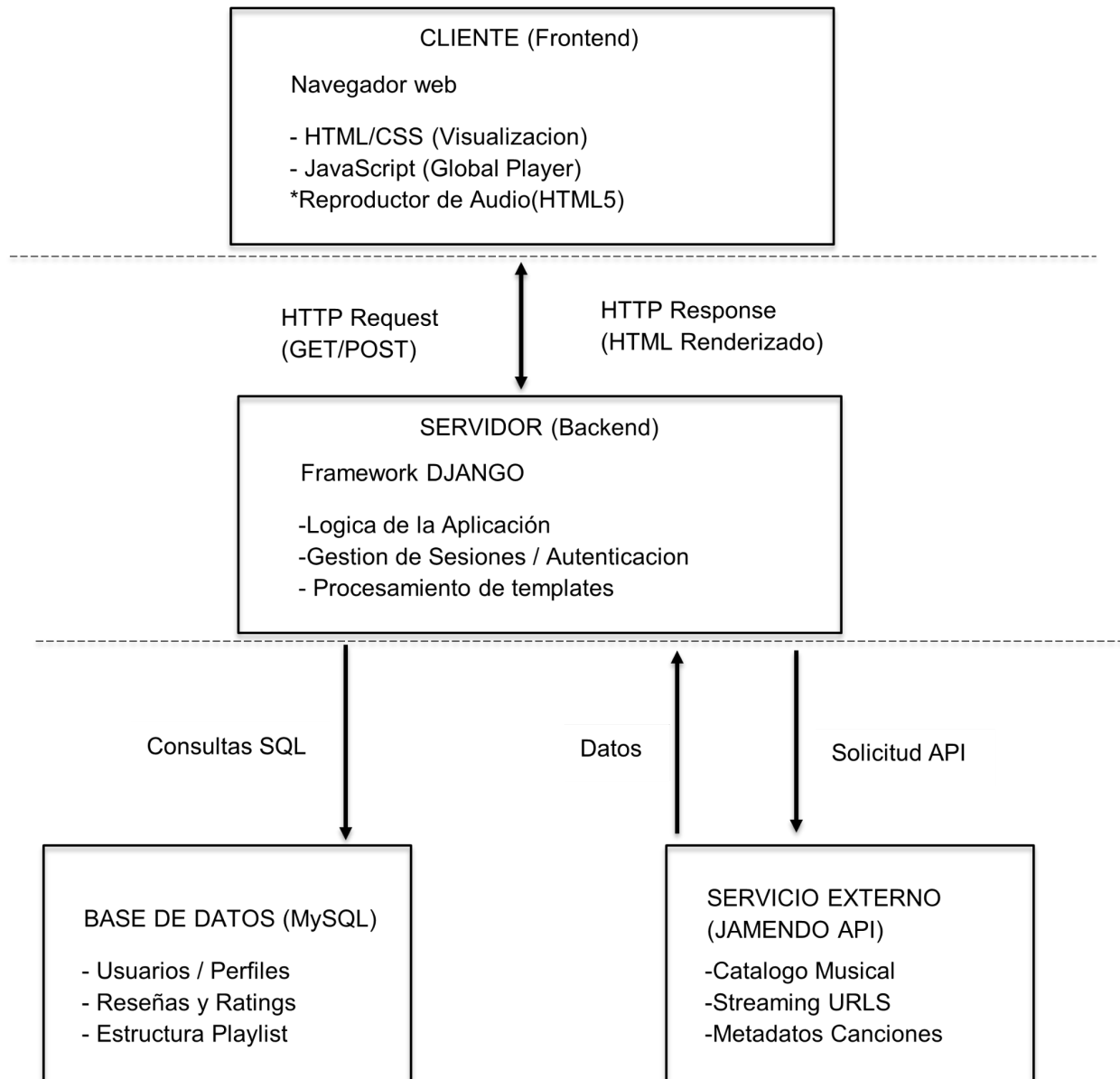
El alcance final del producto consiste en una aplicación web desarrollada en Django que consume la API de Jamendo para visualizar y reproducir música mediante un reproductor global. La plataforma integra una gestión completa de cuentas (registro e inicio de sesión) y permite personalizar la experiencia a través de la selección de géneros, administración de favoritos, creación de playlists y una vista de perfil dedicada. Además, el sistema facilita la interacción social permitiendo compartir contenido mediante enlaces o códigos QR, e incluye un sistema de reseñas para opinar sobre las canciones, así como una funcionalidad de "música sin conexión" implementada mediante un guardado lógico en la base de datos.

Costo total del desarrollo

Considerando una tarifa individual de 30 USD por hora y un equipo de tres desarrolladores, el costo operativo conjunto se establece en 90 USD por hora. El ciclo de desarrollo abarcó un total de 64 horas efectivas, estructuradas en cuatro sprints de cuatro días con jornadas diarias de cuatro horas. Esto genera un costo base de 5,760 USD. Finalmente, se aplicó un margen de contingencia del 20% (factor de 1.20) para cubrir imprevistos, resultando en un presupuesto final de 6,912 USD.

Arquitectura

Diagrama general



1. Capa de Presentación (Frontend):

- El usuario interactúa a través de un navegador web.
- Las interfaces son renderizadas desde el servidor utilizando Django Templates (HTML/CSS), lo que garantiza una carga inicial rápida

- Se utiliza JavaScript para la interactividad del "Reproductor Global" (global_player.js), permitiendo que la música continúe sonando mientras el usuario navega entre diferentes vistas y también para algunas animaciones integradas.

2. Capa de Lógica (Backend - Django):

- Actúa como el controlador central. Recibe las solicitudes HTTP del cliente y gestiona la lógica de negocio.
- Se comunica con la API de Jamendo para buscar metadatos de canciones y obtener enlaces de streaming, evitando el almacenamiento de archivos de audio pesados en el propio servidor.

3. Capa de Datos (MySQL):

- Almacena la información relacional crítica del negocio: credenciales de usuarios, perfiles de artistas, reseñas, calificaciones y la estructura de las playlists (referencias a canciones, no los archivos de audio).

4. Servicios Externos (Jamendo API):

- Provee el catálogo musical y el servicio de streaming de audio, reduciendo la carga de infraestructura y almacenamiento del proyecto.

Tecnologías utilizadas

Backend: Django

Frontend: JS/HTML/CSS

BD: MySQL

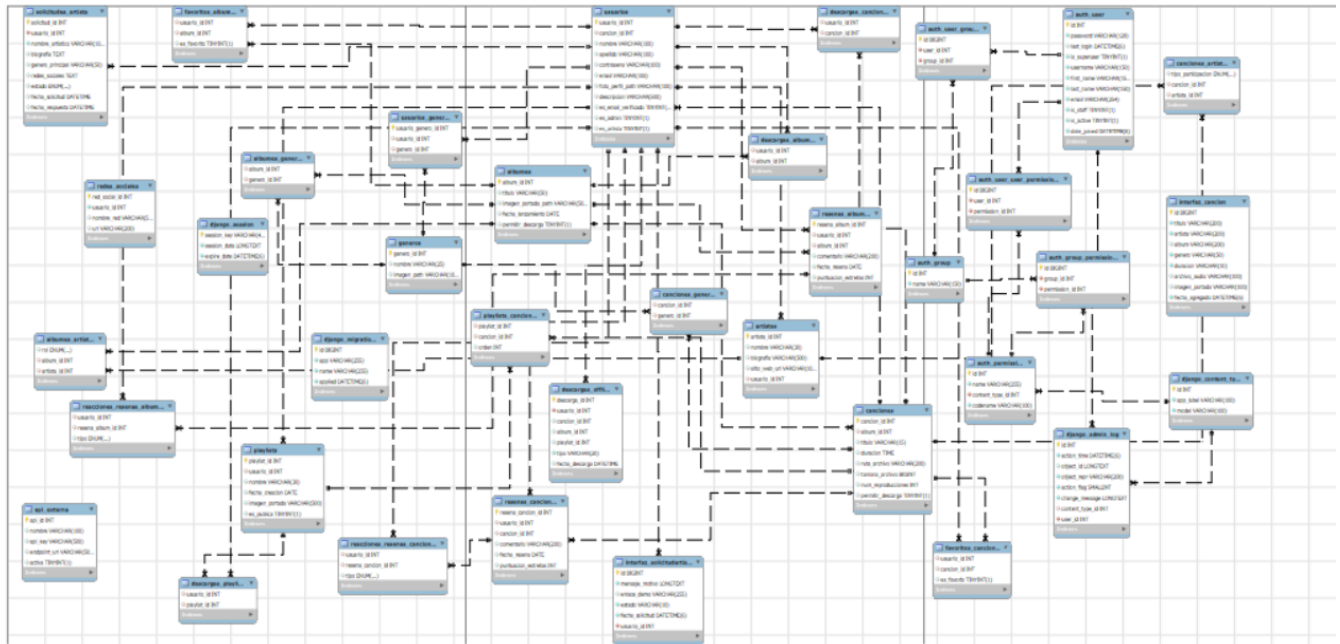
API: Jamendo

Autenticación: EmailJS Api

Control de Versiones: Git - Github

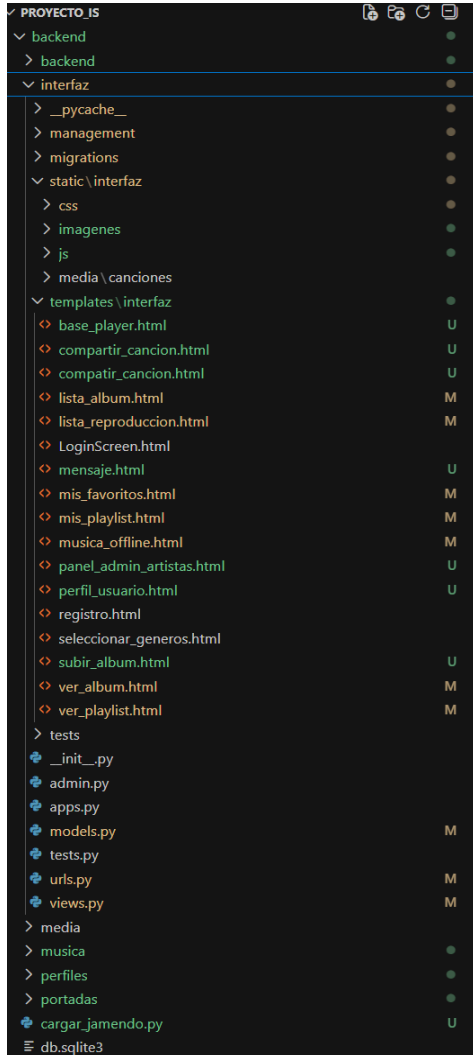
Entornos de desarrollo ocupados: Visual Studio

Diseño de la base de datos (ERD o modelo lógico)



Estructura de módulos

Se definió una estructura personalizada para mantener el orden de los archivos estáticos y plantillas, separando los recursos de la interfaz de la configuración base.



1. Módulo de Vistas HTML (templates/interfaz)

Aquí se almacena el **diseño** y la estructura visual de las páginas, pero sin lógica de programación (solo marcadores de posición que Django rellena).

- **templates/interfaz/**: Contiene todos los archivos .html que Django utiliza para renderizar la página.

- **base_player.html**: Es el módulo más importante, ya que centraliza la barra de reproducción para que se muestre en todas las páginas, asegurando que los controles sean consistentes.

2. Módulo de Recursos Estáticos (static/interfaz)

Este módulo almacena todos los archivos que el navegador del usuario necesita descargar una sola vez (CSS, JavaScript, imágenes).

- **static/interfaz/js/global_player.js**: Este archivo contiene todo el código **JavaScript** para que la barra de reproducción funcione, asegurando que los controles de audio (play, pause, volumen) trabajen continuamente aunque el usuario cambie de página.

- **static/interfaz/css/**: Contiene los archivos de estilo que le dan la apariencia moderna y el estilo visual al proyecto.
- **static/interfaz/img/**: Contiene las imágenes, como el logo o las portadas por defecto.

3. Directorio Base y Aplicación Principal (interfaz/)

A. models.py (La capa de Datos o "Modelos")

- **Propósito:** Definir la estructura de la información que se guardará en la base de datos **MySQL**.
- **Contiene:** Las clases de Python que representan las tablas en la base de datos. Por ejemplo, la clase Usuario, la clase Reseña, la clase Playlist y la clase Artista.

B. views.py (La capa de Lógica o "Vistas")

- **Propósito:** Contiene las funciones de Python que manejan la lógica de negocio y las operaciones que debe realizar la aplicación. Es el **controlador** de la aplicación.
- **Contiene:** Las funciones para cada URL que el usuario visita. Por ejemplo, una función para el *login*, una función para *crear una reseña* y una función para *explorar música*.

C. urls.py (El "Enrutador")

- **Propósito:** Actúa como el mapa de direcciones de la aplicación.
- **Contiene:** Un listado de todas las direcciones web (URLs) que existen en el sistema y a qué función de views.py debe llamar cada una.

D. settings.py (La Configuración Global backend/)

- **Propósito:** Almacenar todas las configuraciones esenciales para que el proyecto funcione.
- **Contiene:** Los parámetros de conexión a la base de datos (MySQL), las claves secretas de la API de Jamendo.

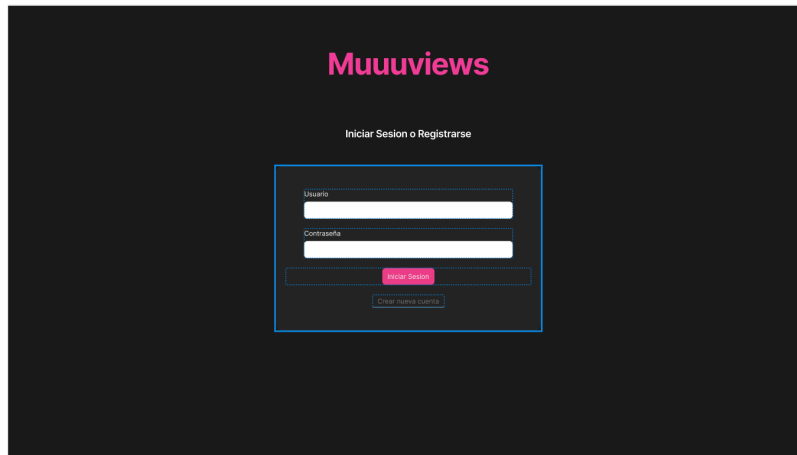
E. cargar_jamendo.py (Integración de Servicios Externos)

- **Propósito:** Aislar toda la lógica de comunicación con recursos externos.

- **Contiene:** Las funciones específicas encargadas de construir las peticiones HTTP y procesar las respuestas JSON de la **Jamendo API**.

UI/UX

Wireframes



Muuuvviews

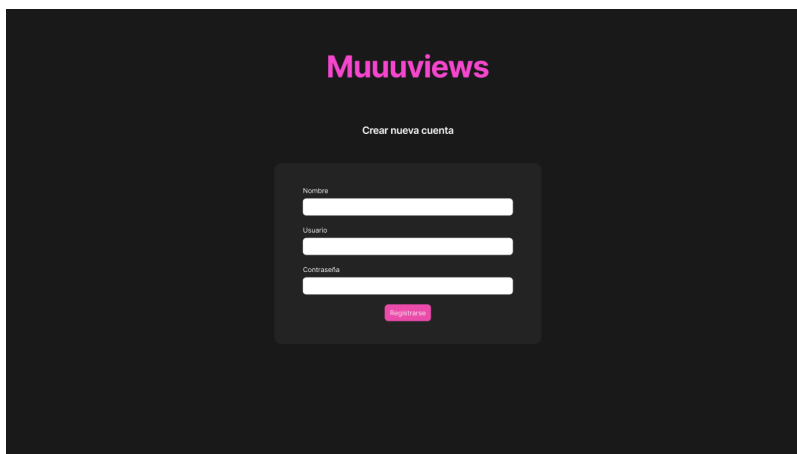
Iniciar Sesión o Registrarse

Usuario

Contraseña

Iniciar Sesión

Crear nueva cuenta



Muuuvviews

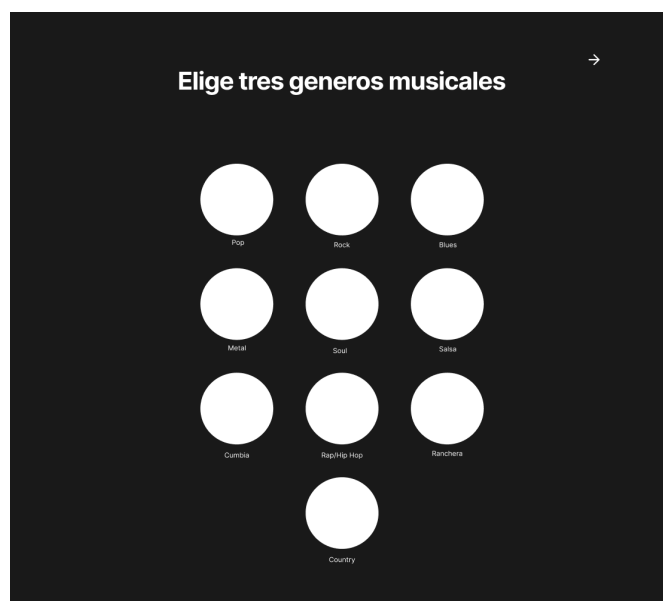
Crear nueva cuenta

Nombre

Usuario

Contraseña

Registrar



Elige tres generos musicales →

Pop

Rock

Blues

Metal

Soul

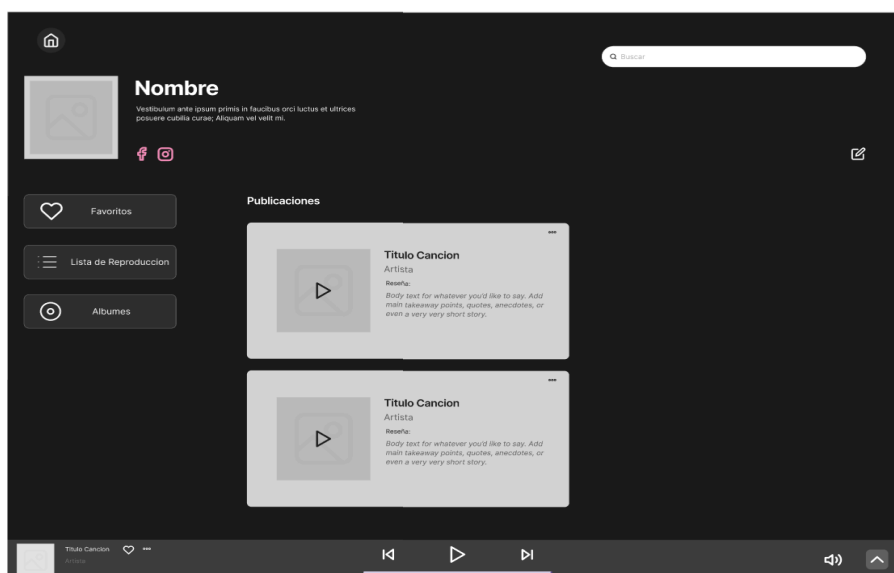
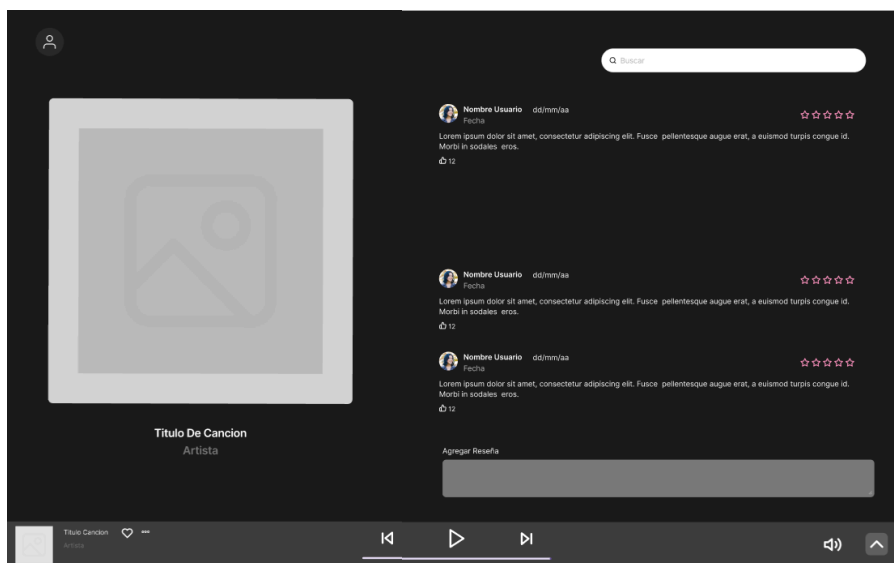
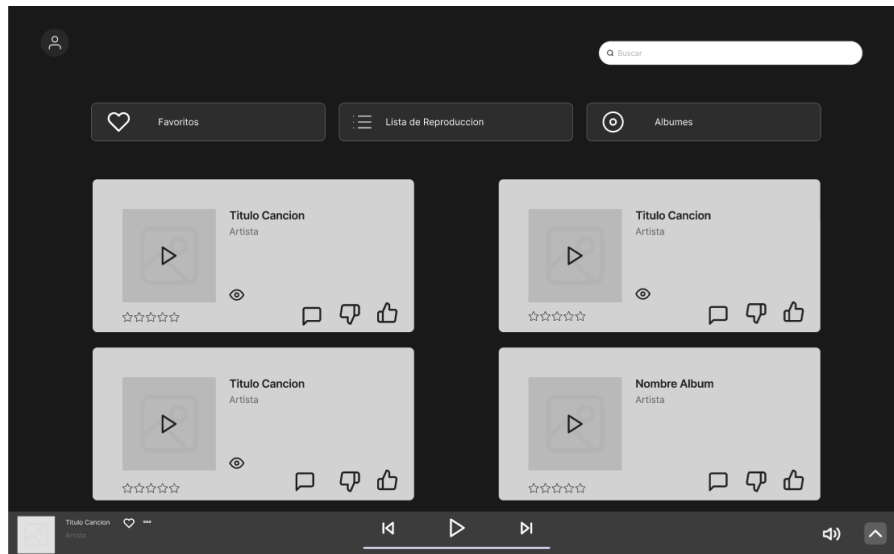
Salsa

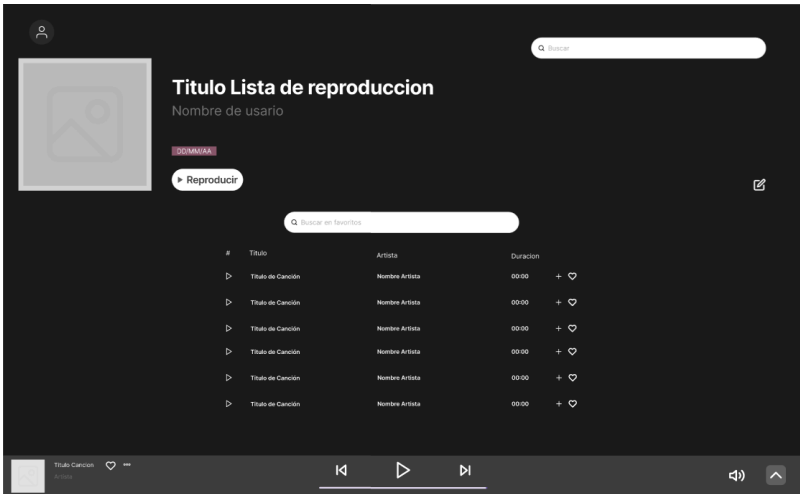
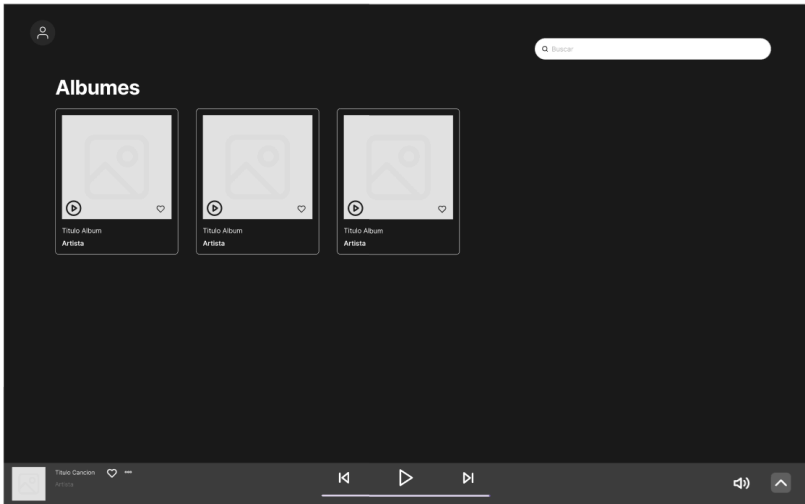
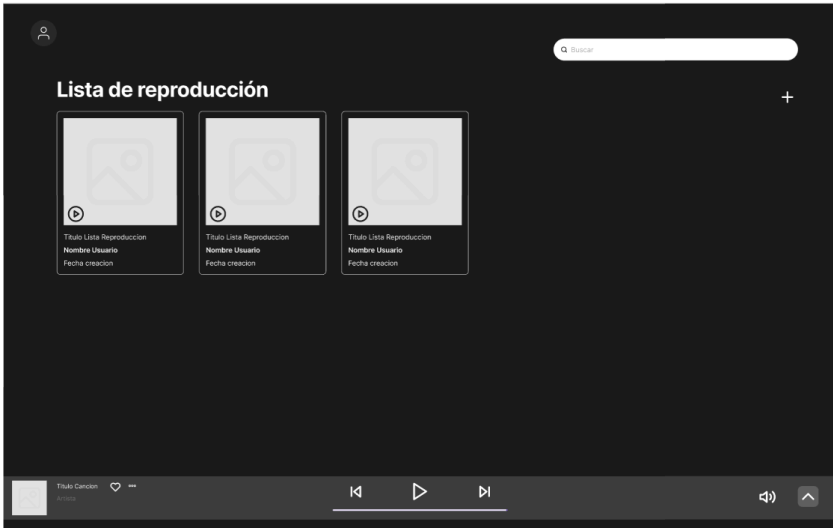
Cumbia

Rap/Hip Hop

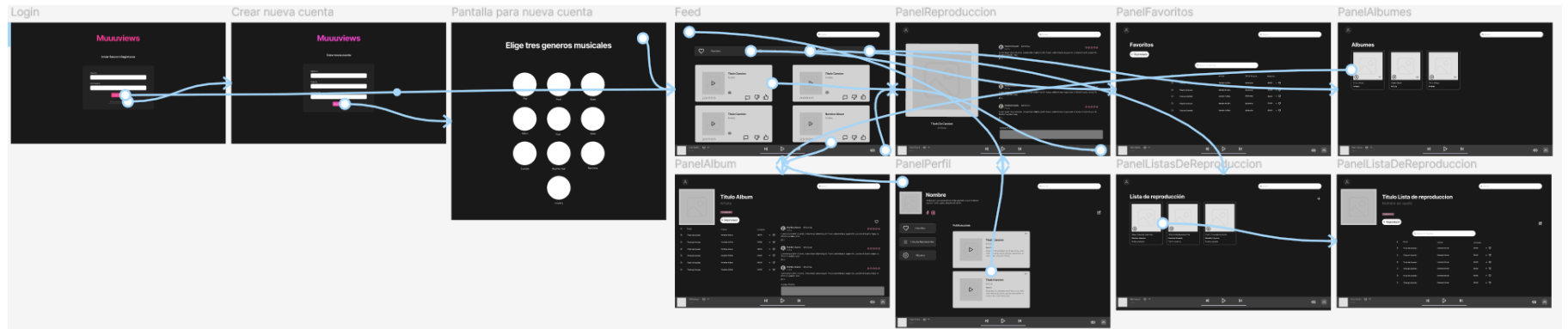
Ranchera

Country





Flujo de pantallas



Estilo visual y decisiones de diseño

Patrón: Servidor renderizado con Django Templates para consistencia y simplicidad (HTML/CSS) y JS solo para interacción (reproductor).

Reproductor: Centralizado en `base_player.html` para mantener controls visibles en todas las páginas.

Separación UI: Templates por página y static por tipo (js/imagenes) para mantener orden del frontend.

Bitácora de Cambios

Versiones

v0.1 (Sprint 1) – Base + Onboarding

Historias completadas: HU-01 Registro de usuario, HU-02 Selección de Género

Estado y Logros: Integraciones iniciales con la Base de Datos y flujo de usuario establecido → conexión entre las pantallas de registro y selección de géneros.

v0.2 (Sprint 2) – Exploración + Biblioteca

Historias completadas: HU-03 Explorar música

Estado y Logros: Funcionalidad completa de exploración musical. Se implementó la Biblioteca (playlists, favoritos, álbumes), el modo "Música sin conexión" (guardado lógico) y la personalización básica del perfil.

v0.3 (Sprint 3) – Player + Playlists + Reseñas + Perfil parcial + Compartir Música

Historias completadas: HU-05 (parcial), HU-06, HU-07, HU-04, HU-08

Estado y Logros: El Reproductor Global está completamente funcional. Se habilitó la creación de Playlists, la funcionalidad de Reseñas y la opción de Compartir música mediante códigos QR o URLs. El Perfil de Usuario está parcialmente implementado (pendiente la visualización de reseñas del propio usuario).

v1.0 (Sprint 4) – Encolar Música + Perfil Artista + Subir Musica + Playlist

Historias completadas: HU-05, HU-09, HU-10, HU-11

Estado y Logros: Funcionalidad de Encolar canciones lista, con visualización de la lista de espera. El Perfil de Usuario está 100% completo, incluyendo la visualización de las reseñas. Se implementó la funcionalidad de Perfil de Artista para la subida de álbumes (previa solicitud de acceso), permitiendo a los artistas cargar su contenido.

v1.1 (Sprint 4) - Todo lo demás + Diseño limpio trending + Depuración de código + Branding

Historias completadas: Todas

Estado y Logros: El equipo tomó la decisión unánime de modificar el nombre del producto, pasando de Muuuvviews a Muusic, para alinearlos mejor con la identidad de marca deseada. Además de esta actualización de branding, se realizaron importantes mejoras técnicas y visuales: se ejecutó una modificación completa del diseño y la interfaz de usuario (UX/UI) para garantizar que el producto estuviera a la vanguardia de las tendencias actuales. Finalmente, se llevó a cabo una depuración exhaustiva del código para asegurar una base de software más limpia, legible y mantenible a futuro.

Funciones agregadas, modificadas o descartadas

HU-04 – Reseñar

Se modificó el tiempo en el que se hacía debido a que necesitaba de otras historias para poder completarse como: Perfil de usuario y barra de reproducción .

HU-05 – Perfil de usuario

Se agregó el poder ver las reseñas hechas.

HU-07 – Barra de reproducción

Se agregó el botón de ver las canciones encoladas.

Registro de decisiones técnicas

Se eligió Jamendo API para catálogo/streaming en lugar de almacenar audio propio (reduce complejidad de storage y hay mucha variedad y es gratuito).

Se implementó la app con Django + Templates (HTML/CSS) + JavaScript para velocidad de desarrollo y renderizado server-side.

Se usó MySQL como base de datos del proyecto.

Se centralizó la reproducción en un reproductor global (base_player.html + global_player.js) para mantener controles consistentes entre pantallas.

Se definió el módulo `static/interfaz/` para separar recursos (JS/imagenes) de los `templates/interfaz/` y mantener orden del frontend.

Se priorizó entregar primero el flujo base del producto (registro → géneros → exploración → player) antes de completar funciones sociales avanzadas (compartir, cola, artista).