



MUSIC
MUSE

INTRODUCCIÓN

MusicMuse es una aplicación web innovadora que permite a los artistas emergentes gestionar y distribuir su música de manera independiente, eliminando la necesidad de intermediarios.



MOTIVACIÓN

Fomentar la interacción directa entre músicos y oyentes, facilitando la difusión y descubrimiento de nuevos talentos en la industria musical.





OBJETIVOS PROPUESTOS

Aplicación web enfocada a la difusión de música.

Visualmente llamativa.

Sencillez en añadir música.

Personal: app web llamativa, funcional y dinámica

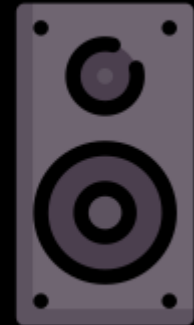
Programación: asentar las bases prendidas.

OBJETIVOS ESPECIFICOS

Implementación MVC

Registrar usuarios y permisos

Trabajar con datos de la BBDD





METODOLOGIA USADA

Metodología ágil

En el desarrollo de MusicMuse, se combinaron técnicas tradicionales y digitales de gestión de proyectos para un control efectivo y estructurado. Se emplearon tarjetas para una visualización física del progreso y la organización de tareas, complementadas con el uso de Notion para la coordinación digital, documentación y seguimiento de objetivos. El enfoque iterativo permitió adaptaciones continuas y revisiones periódicas, asegurando la alineación del proyecto con los objetivos establecidos y la capacidad de responder a cambios imprevistos.

TECNOLOGIAS Y HERRAMIENTAS



FRONTED



BACKEND



IDE



ANALISIS DEL PROYECTO

-Requisitos funcionales:

Funciones y características de la app:

- *Visualizar artistas de la semana

- *Registrarse

- *Login de usuario

-Requisitos no funcionales:

El sistema debe ser seguro y proteger los datos de los usuarios.

Tiempo de aprendizaje inferior a una hora.

App rapida y responder en menos de 2 segundos.

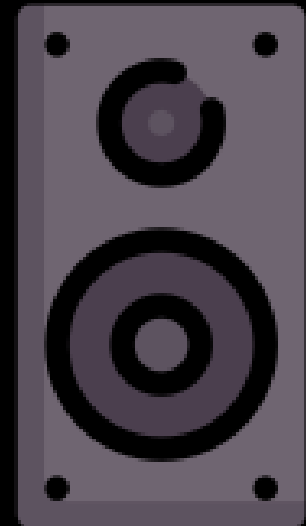
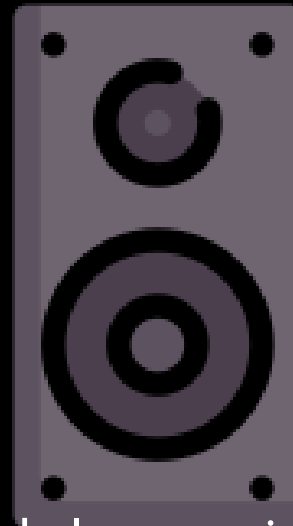


DIAGRAMA CASOS DE USO

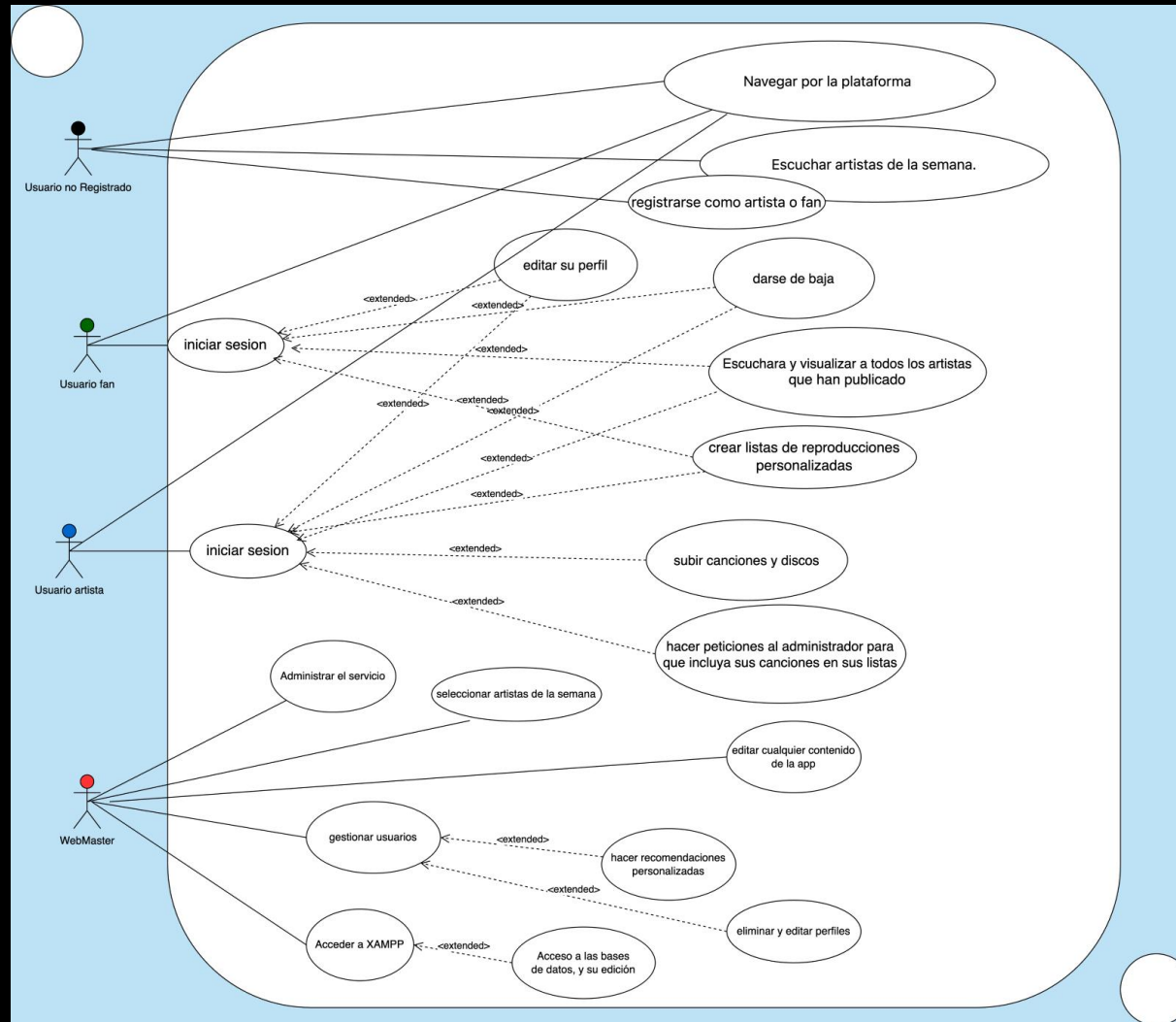
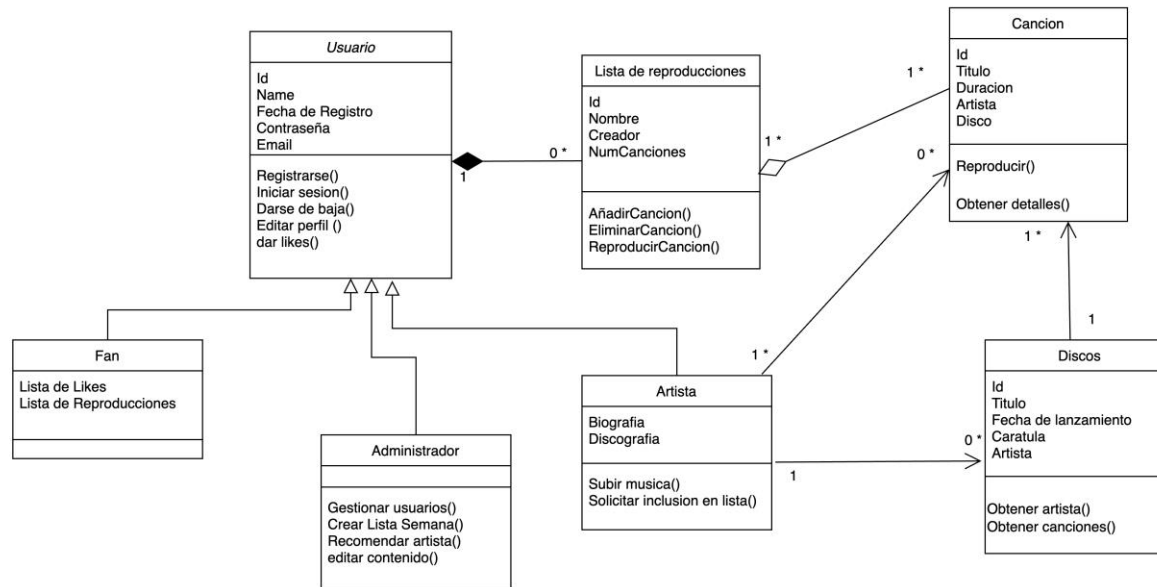


DIAGRAMA DE CLASES

*La relacion entre el usuario y las listas de reproducción es de composición, ya que si el usuario deja de existir su lista de reproducciones desaparece
*La relacion entre la lista de reproducciones y las canciones es de agregación ya que estas existen de forma independiente, no están ligadas a las listas

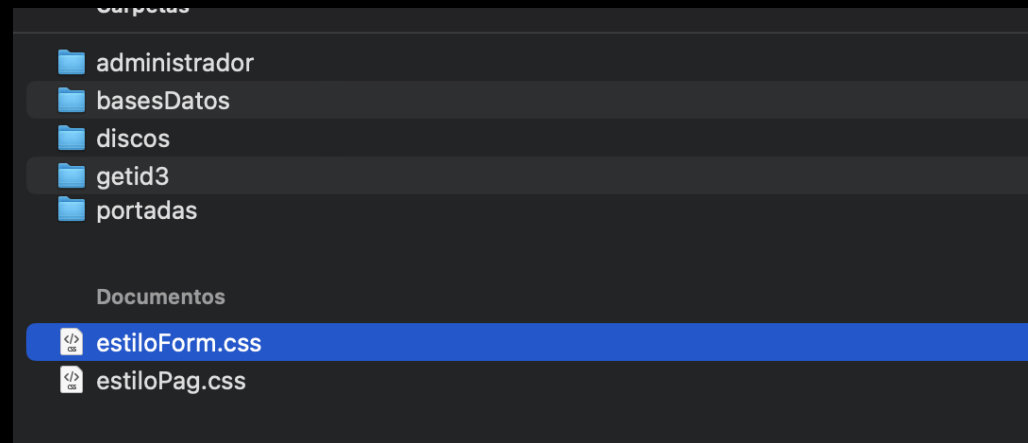


DISEÑO DEL PROYECTO

- **MusicMuse adopta un diseño visual minimalista similar a plataformas como Spotify, utilizando una paleta de colores con tonos oscuros y grises, destacando el rojo para energizar la interfaz y destacar elementos clave.**
- **Elección del Esquema de Colores:** Los tonos oscuros en la interfaz mejoran la legibilidad y el confort visual en sesiones prolongadas, mientras que el rojo se utiliza para dirigir intuitivamente a los usuarios hacia funciones importantes.
- **Inspiración y Objetivos del Diseño:** El diseño minimalista de MusicMuse busca minimizar distracciones, facilitar la navegación y establecer una identidad visual única mediante el uso del color rojo, mejorando la experiencia del usuario y la conexión emocional con la plataforma.
- **Implementación Técnica:** El diseño se llevó a cabo usando CSS3 y Bootstrap, asegurando que la interfaz sea responsiva y funcione eficientemente en diversos dispositivos y tamaños de pantalla.

ORGANIZACION ARCHIVOS Y CARPETAS

Carpetas principales



BASE DE DATOS MySQL

Para implementar la base de datos de mi proyecto MusicMuse, diseñé y desarrollé un esquema de base de datos utilizando MySQL

Tablas principales:

```
CREATE TABLE Usuarios (  
  UsuarioID INT AUTO_INCREMENT PRIMARY KEY,  
  NombreUsuario VARCHAR(255) NOT NULL,  
  Correo VARCHAR(255) NOT NULL,  
  ContraseñaHash VARCHAR (255) NOT NULL,  
  Rol VARCHAR (50),  
  FechaCreacionPerfil DATE,  
  TokenRecuperacionContraseña VARCHAR (255)  
);
```

```
CREATE TABLE Artistas (  
  ArtistaID INT AUTO_INCREMENT PRIMARY KEY,  
  UsuarioID INT,  
  Nombre VARCHAR(255) NOT NULL,  
  Descripcion TEXT,  
  Pais VARCHAR(100),  
  FOREIGN KEY (UsuarioID) REFERENCES Usuarios(UsuarioID)  
);
```

CONEXION A BASE DE DATOS

```
$host = 'localhost';  
$usuario = 'root';  
$nombre_base_datos = 'BaseMusicas';  
  
// Crear una única conexión  
$conn = new mysqli($host, $usuario, "", $nombre_base_datos);  
  
if ($conn->connect_error) {  
    die("Conexión fallida: " . $conn->connect_error);  
}  
  
mysqli_set_charset($conn, "utf8");
```



Consultas a la base de datos

Para obtener el rol del usuario y otras consultas, se utilizan declaraciones preparadas para evitar inyecciones SQL:

```
$sql = "SELECT Rol
        FROM Usuarios
        WHERE NombreUsuario = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("s", $nombreUsu);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    // Obtener el rol del usuario
    $row = $result->fetch_assoc();
    if ($row['Rol'] == "artista") {
        $_SESSION['rol'] = 0;
    } else {
        $_SESSION['rol'] = 1;
    }
}
```

Ingresar datos:

```
<?php
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $host = 'localhost'; // Cambiar por tu host
    $usuario = 'root'; // Cambiar por tu usuario de la base de datos
    $nombre_base_datos = 'BaseMusicas'; // Cambiar por el nombre de tu base de datos

    // Crear conexión
    $con = mysqli_connect($host, $usuario, '', $nombre_base_datos) or die("Problemas con la conexión a la base de datos");

    // Establecer la codificación de caracteres a UTF-8
    mysqli_set_charset($con, "utf8");

    // Obtener los valores del formulario
    $nombreArtista = $_POST['artistName'];
    $biografia = $_POST['Bio'];
    $pais = $_POST['pais'];
    $contraseña = $_POST['password'];
    $email = $_POST['email'];
    $nombreUsuario = $_POST['nombre'];
    $rol = "artista";
    $fechaAlta = date('Y-m-d'); // Obtiene la fecha actual en formato 'Año-Mes-Día'

    // Codificar la contraseña para guardarla de forma segura
    $hash = password_hash($contraseña, PASSWORD_DEFAULT);

    $sqlUsuario = "INSERT INTO Usuarios (NombreUsuario, Correo, ContraseñaHash, Rol, FechaCreacionPerfil) VALUES (?, ?, ?, ?, ?)";
    $stmtUsuario = $con->prepare($sqlUsuario);
    $stmtUsuario->bind_param("sssss", $nombreUsuario, $email, $hash, $rol, $fechaAlta);

    // Ejecutar la consulta para insertar el usuario
```

Funciones para reproductor

```
//SECCION PARA MANEJAR REPRODUCUTOR
var audio= document.querySelector('audio');
var cancion= document.getElementById('cancionplay');
var prev= document.getElementById('prev');
var play= document.getElementById('play');
var next=document.getElementById('next');
var current_time=document.getElementById('current_time');
var current_audio=document.getElementById('current_audio');
var progres=document.getElementById('progres');
var progresContainer=document.querySelector('.progresContainer');
var canciones=['Cebollita Macabra - Introducción al Mundo de Mike',
'Delber Grady - Stay', 'Toni Manzano - Perro Azul', 'Muqui - Tarde de Abril'];
var audioIndex=1;

loadAudio(canciones[audioIndex])

function loadAudio(cancionPasada){
  cancion.textContent=cancionPasada
  audio.src=`administrador/canciones/${cancionPasada}.wav`
  audio.addEventListener("loadedmetadata", ()=>{
    timesong(audio.duration, current_audio)
  })
}
```

```
function prevSong(){
  audioIndex--
  if(audioIndex < 0){
    audioIndex =canciones.length -1;
  }
  loadAudio( audioIndex);
  playSong();
}

function nextSong(){
  audioIndex ++
  if(audioIndex>canciones.length -1){
    audioIndex =0;
  }
  loadAudio( audioIndex);
  playSong();
}
```

Despliegue y pruebas

Certeza de que todos los requisitos funcionales pedidos por el cliente han sido cumplidos.

- .Logging de usuario
- .Unlogging de usuario
- .Registro de usuario
- .Perfil de usuario – Actualizar datos
- .Perfil de usuario – Baja usuario
- .Mostrar artista
- .Subir musica a la plataforma
- .Añadir fotografía de disco
- .Editar canciones
- .Eliminar canciones
- .Subir disco
- .Mostrar disco subido
- .Quitar disco de la discografía
- .Filtrar artistas en la plataforma
- .Reproducir canciones

CONCLUSIONES

Aplicacion web conseguida.

DIFICULTADES

Requisitos y diagramas

Intergrar el reproductor en la web

Gestion de canciones

MEJORAS FUTURAS

- **Intercomunicación entre Usuarios:** Se introducirán funciones de red social completas para permitir interacciones más ricas entre usuarios y artistas, como comentar, compartir y reaccionar a publicaciones y música.
- **Creación de Listas Personalizadas:** Se ampliará la funcionalidad para que los usuarios puedan crear y gestionar listas de reproducción personalizadas, con opciones de privacidad ajustables.
- **Mercadotecnia y Venta de Música:** Se desarrollará un espacio para que los artistas vendan su música y mercancía directamente a los fans, proporcionando una nueva fuente de ingresos y enriqueciendo el valor de la plataforma.
- **Mejoras Tecnológicas y de Seguridad:** Se continuará mejorando la infraestructura técnica para soportar un mayor volumen de datos y asegurar la propiedad intelectual y privacidad de los usuarios.

DEMOSTRACIÓN



MUSIC
NUSE

