



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

DEPARTAMENTO DE INGENIERÍA EN SOFTWARE

LICENCIATURA EN INGENIERÍA EN SOFTWARE

INGENIERÍA WEB

PROYECTO SEMESTRAL: FASE 4

DIEGO CORRALES 8-1001-1890

JUAN COPRI 8-976-1088

DANIEL PEREZ 20-14-7935

JOSUÉ PINO 8-1012-688

PROFESORA: DRA. ELBA VALDERRAMA

GRUPO 1SF134

II SEMESTRE

2025

Índice

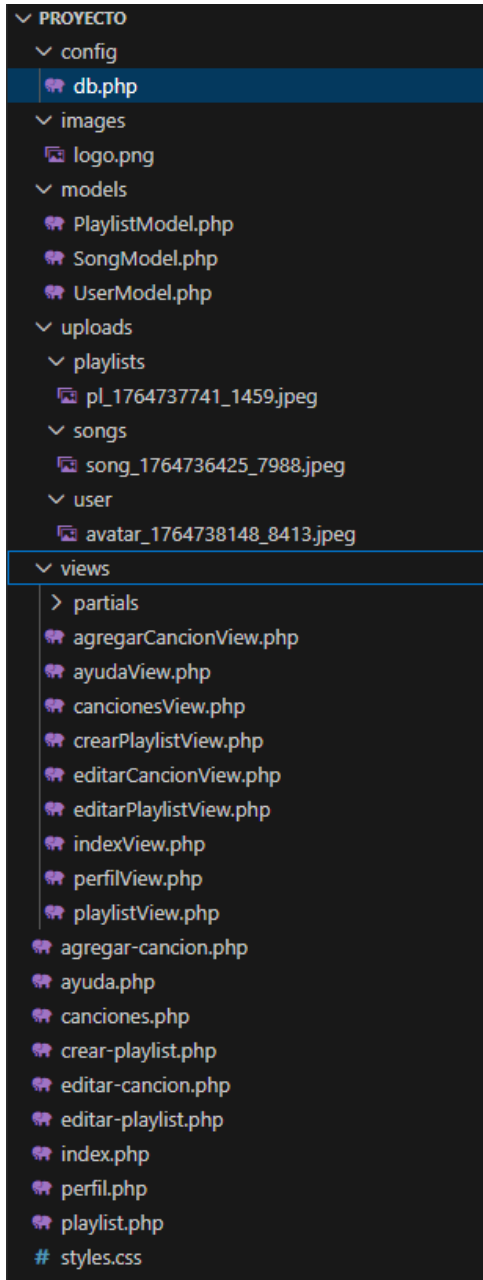
Fase 4: BackEnd	3
Enlace De Repositorio.....	3
Estructura Del Backend Y El Html.....	3
La Estructura Jerárquica/Diagrama	4
Enlace Al Sitio Web	5
Código	5
Tabla de Aportes	62

Fase 4: BackEnd

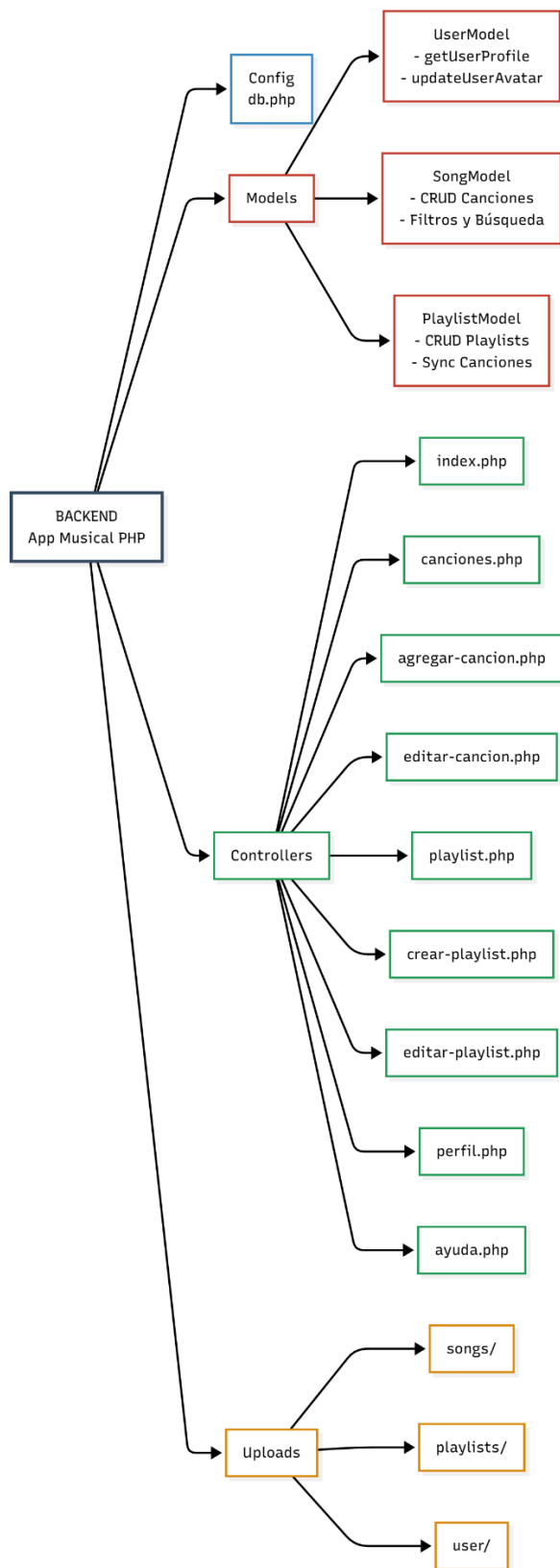
Enlace De Repositorio

Link: <https://github.com/crdz05/IngWeb---ProyectoFinal-Pino-Corrales-Perez-Copri>

Estructura Del BackEnd Y El Html



La Estructura Jerárquica/Diagrama



Enlace Al Sitio Web

Plataforma Usada: <https://dash.infinityfree.com>

Link de pag web: <http://fplay.rf.gd>

Código

Index.php

```
<?php
// index.php → Página de inicio (dashboard)

require __DIR__ . '/config/db.php';
require __DIR__ . '/models/SongModel.php';
require __DIR__ . '/models/PlaylistModel.php';

$pageTitle = 'Inicio - App Musical';
$activePage = 'inicio';

// 1. Canciones recientes (para la sección de abajo y para el card de
"Canciones")
$recentSongs = getRecentSongs($pdo, 3); // máximo 3

// Primera canción (para la imagen grande de "Canciones")
$firstSong = $recentSongs[0] ?? null;

// 2. Playlists (para el card de "Playlist")
$playlists = getPlaylists($pdo);
$firstPlaylist = $playlists[0] ?? null;

// Cargar vistas
require __DIR__ . '/views/partials/header.php';
require __DIR__ . '/views/indexView.php';
require __DIR__ . '/views/partials/footer.php';
```

Canciones.php

```
<?php
// canciones.php → Controlador de la página "Ver Canciones"

require __DIR__ . '/config/db.php';
require __DIR__ . '/models/SongModel.php';

$pageTitle = 'Ver Canciones - App Musical';
$activePage = 'canciones';

// --- Eliminar canción (si viene ?eliminar=ID en la URL) ---
if (isset($_GET['eliminar']) && ctype_digit($_GET['eliminar'])) {
    $id = (int) $_GET['eliminar'];

    if ($id > 0) {
        deleteSong($pdo, $id);
        header('Location: canciones.php?msg=eliminada');
        exit;
    }
}

// --- Filtros recibidos por GET ---
$search = trim($_GET['buscar'] ?? '');
$genre = $_GET['genero'] ?? 'Todos';
$artist = $_GET['artista'] ?? 'Todos';
$year = $_GET['ano'] ?? 'Todos';

// Array de filtros para el modelo
$filters = [
    'search' => $search,
    'genre' => $genre,
    'artist' => $artist,
    'year' => $year,
];

// Obtenemos las canciones desde el modelo
$canciones = getSongs($pdo, $filters);

// Listas para los combos de Artista y Año
$artistsList = getDistinctArtists($pdo);
$yearsList = getDistinctYears($pdo);

// Mensaje opcional (ej: después de eliminar o agregar/editar)
```

```

$message = '';
if (!empty($_GET['msg'])) {
    switch ($_GET['msg']) {
        case 'eliminada':
            $message = 'Canción eliminada correctamente.';
            break;
        case 'agregada':
            $message = 'Canción agregada correctamente.';
            break;
        case 'editada':
            $message = 'Canción actualizada correctamente.';
            break;
    }
}

// Cargamos vistas
require __DIR__ . '/views/partials/header.php';
require __DIR__ . '/views/cancionesView.php';
require __DIR__ . '/views/partials/footer.php';

```

agregar-cancion.php

```

<?php
// agregar-cancion.php → Controlador para agregar nuevas canciones

require __DIR__ . '/config/db.php';
require __DIR__ . '/models/SongModel.php';

$pageTitle = 'Agregar Canción - App Musical';
$activePage = 'canciones';

$errors = [];
$oldData = [
    'titulo' => '',
    'artista' => '',
    'genero' => '',
    'ano' => '',
    'album' => '',
    'duracion' => '',
];

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // 1. Recibir datos del formulario

```

```

$soldData['titulo'] = trim($_POST['titulo'] ?? '');
$soldData['artista'] = trim($_POST['artista'] ?? '');
$soldData['genero'] = trim($_POST['genero'] ?? '');
$soldData['ano'] = trim($_POST['ano'] ?? '');
$soldData['album'] = trim($_POST['album'] ?? '');
$soldData['duracion'] = trim($_POST['duracion'] ?? '');

// 2. Validaciones básicas
if ($soldData['titulo'] === '') {
    $errors[] = 'El título de la canción es obligatorio.';
}

if ($soldData['artista'] === '') {
    $errors[] = 'El artista es obligatorio.';
}

if ($soldData['ano'] !== '' && !ctype_digit($soldData['ano'])) {
    $errors[] = 'El año debe ser un número entero.';
}

// 3. Manejo de la imagen (opcional)
$imagePath = null;

if (!empty($_FILES['imagen']['name']) && $_FILES['imagen']['error'] !==
UPLOAD_ERR_NO_FILE) {
    if ($_FILES['imagen']['error'] === UPLOAD_ERR_OK) {
        $uploadDir = __DIR__ . '/uploads/songs/';
        if (!is_dir($uploadDir)) {
            mkdir($uploadDir, 0777, true);
        }

        $ext = strtolower(pathinfo($_FILES['imagen']['name'],
PATHINFO_EXTENSION));
        $allowed = ['jpg', 'jpeg', 'png', 'gif', 'webp'];

        if (!in_array($ext, $allowed)) {
            $errors[] = 'Formato de imagen no permitido. Usa JPG, PNG,
GIF o WEBP.';
        } else {
            $basename = 'song_' . time() . '_' . mt_rand(1000, 9999) .
'.' . $ext;
            $destPath = $uploadDir . $basename;
            $webPath = 'uploads/songs/' . $basename; // ruta que se
guarda en la BD

```



```

        if (move_uploaded_file($_FILES['imagen']['tmp_name'],
$destPath)) {
            $imagePath = $webPath;
        } else {
            $errors[] = 'No se pudo subir la imagen de la canción.';
        }
    }
} else {
    $errors[] = 'Ocurrió un error al subir la imagen de la
canción.';
}
}

// 4. Si no hay errores, guardar en la BD
if (empty($errors)) {
    $data = [
        'title'      => $oldData['titulo'],
        'artist'     => $oldData['artista'],
        'genre'      => $oldData['genero'] !== '' ? $oldData['genero'] :
null,
        'year'       => $oldData['ano'] !== '' ? (int)$oldData['ano'] :
null,
        'album'      => $oldData['album'] !== '' ? $oldData['album'] :
null,
        'duration'   => $oldData['duracion'] !== '' ?
$oldData['duracion'] : null,
        'image_path' => $imagePath,
    ];

    if (createSong($pdo, $data)) {
        header('Location: canciones.php?msg=agregada');
        exit;
    } else {
        $errors[] = 'Ocurrió un error al guardar la canción en la base
de datos.';
    }
}
}

// Cargar vistas
require __DIR__ . '/views/partials/header.php';
require __DIR__ . '/views/agregarCancionView.php';
require __DIR__ . '/views/partials/footer.php';

```

editar-cancion.php

```
<?php
// editar-cancion.php → Controlador para editar una canción

require __DIR__ . '/config/db.php';
require __DIR__ . '/models/SongModel.php';

$pageTitle = 'Editar Canción - App Musical';
$activePage = 'canciones';

$errors = [];
$formData = [
    'titulo' => '',
    'artista' => '',
    'genero' => '',
    'ano' => '',
    'album' => '',
    'duracion' => '',
];

// 1. Validar ID
if (empty($_GET['id']) || !ctype_digit($_GET['id'])) {
    header('Location: canciones.php');
    exit;
}

$id = (int) $_GET['id'];

// 2. Obtener canción de la BD
$song = getSongById($pdo, $id);

if (!$song) {
    die("La canción con ID $id no existe.");
}

// imagen actual (puede ser null)
$currentImagePath = $song['image_path'] ?? null;

// 3. Rellenar formulario inicial con datos de la canción
$formData['titulo'] = $song['title'] ?? '';
```

```

$formData['artista'] = $song['artist'] ?? '';
$formData['genero'] = $song['genre'] ?? '';
$formData['ano'] = $song['year'] ?? '';
$formData['album'] = $song['album'] ?? '';
$formData['duracion'] = $song['duration'] ?? '';

// 4. Procesar POST (si el usuario envía el formulario)
if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    // Sobrescribir con lo que viene del form
    $formData['titulo'] = trim($_POST['titulo'] ?? '');
    $formData['artista'] = trim($_POST['artista'] ?? '');
    $formData['genero'] = trim($_POST['genero'] ?? '');
    $formData['ano'] = trim($_POST['ano'] ?? '');
    $formData['album'] = trim($_POST['album'] ?? '');
    $formData['duracion'] = trim($_POST['duracion'] ?? '');

    // Validaciones básicas
    if ($formData['titulo'] === '') {
        $errors[] = 'El título de la canción es obligatorio.';
    }

    if ($formData['artista'] === '') {
        $errors[] = 'El artista es obligatorio.';
    }

    if ($formData['ano'] !== '' && !ctype_digit($formData['ano'])) {
        $errors[] = 'El año debe ser un número entero.';
    }

    // Manejo de la imagen (opcional)
    // Por defecto mantenemos la imagen actual
    $imagePath = $currentImagePath;

    if (!empty($_FILES['imagen']['name']) && $_FILES['imagen']['error'] !==
    UPLOAD_ERR_NO_FILE) {
        if ($_FILES['imagen']['error'] === UPLOAD_ERR_OK) {
            $uploadDir = __DIR__ . '/uploads/songs/';
            if (!is_dir($uploadDir)) {
                mkdir($uploadDir, 0777, true);
            }
        }
    }
}

```

```

        $ext = strtolower(pathinfo($_FILES['imagen']['name'],
PATHINFO_EXTENSION));
        $allowed = ['jpg', 'jpeg', 'png', 'gif', 'webp'];

        if (!in_array($ext, $allowed)) {
            $errors[] = 'Formato de imagen no permitido. Usa JPG, PNG,
GIF o WEBP.';
        } else {
            $basename = 'song_' . time() . '_' . mt_rand(1000, 9999) .
            '.' . $ext;

            $destPath = $uploadDir . $basename;
            $webPath = 'uploads/songs/' . $basename;

            if (move_uploaded_file($_FILES['imagen']['tmp_name'],
$destPath)) {
                $imagePath = $webPath;
            } else {
                $errors[] = 'No se pudo subir la nueva imagen de la
canción.';
            }
        }
    } else {
        $errors[] = 'Ocurrió un error al subir la imagen de la
canción.';
    }
}

// Si no hay errores, actualizar en la BD
if (empty($errors)) {
    $data = [
        'title'      => $formData['titulo'],
        'artist'     => $formData['artista'],
        'genre'      => $formData['genero'] !== '' ? $formData['genero']
: null,
        'year'       => $formData['ano'] !== '' ? (int)$formData['ano']
: null,
        'album'      => $formData['album'] !== '' ? $formData['album'] :
null,
        'duration'   => $formData['duracion'] !== '' ?
$formData['duracion'] : null,
        'image_path' => $imagePath,
    ];

    if (updateSong($pdo, $id, $data)) {

```

```

        header('Location: canciones.php?msg=editada');
        exit;
    } else {
        $errors[] = 'Ocurrió un error al actualizar la canción en la
base de datos.';
    }
}
}
}

// 5. Cargar vistas
require __DIR__ . '/views/partials/header.php';
require __DIR__ . '/views/editarCancionView.php';
require __DIR__ . '/views/partials/footer.php';

```

Playlist.php

```

<?php
// playlist.php → Controlador para "Mis Playlists"

require __DIR__ . '/config/db.php';
require __DIR__ . '/models/PlaylistModel.php';

$pageTitle = 'Mis Playlists - App Musical';
$activePage = 'playlist';

$message = '';

// Eliminar playlist si viene ?eliminar=ID
if (isset($_GET['eliminar']) && ctype_digit($_GET['eliminar'])) {
    $id = (int) $_GET['eliminar'];

    if ($id > 0) {
        if (deletePlaylist($pdo, $id)) {
            header('Location: playlist.php?msg=eliminada');
            exit;
        } else {
            $message = 'No se pudo eliminar la playlist. Intente
nuevamente.';
        }
    }
}

// Mensaje opcional por acciones anteriores

```

```

if (isset($_GET['msg'])) {
    if ($_GET['msg'] === 'creada') {
        $message = 'Playlist creada correctamente.';
    } elseif ($_GET['msg'] === 'editada') {
        $message = 'Playlist actualizada correctamente.';
    } elseif ($_GET['msg'] === 'eliminada') {
        $message = 'Playlist eliminada correctamente.';
    }
}

// Obtener playlists desde el modelo
$playlists = getPlaylists($pdo);

// Cargar vistas
require __DIR__ . '/views/partials/header.php';
require __DIR__ . '/views/playlistView.php';
require __DIR__ . '/views/partials/footer.php';

```

crear-playlist.php

```

<?php
// crear-playlist.php → Controlador para crear una nueva playlist

require __DIR__ . '/config/db.php';
require __DIR__ . '/models/PlaylistModel.php';

$pageTitle = 'Crear Playlist - App Musical';
$activePage = 'playlist';

$errors = [];
$formData = [
    'nombre' => '',
    'descripcion' => '',
];

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // 1. Datos del formulario
    $formData['nombre'] = trim($_POST['nombre'] ?? '');
    $formData['descripcion'] = trim($_POST['descripcion'] ?? '');

    // 2. Validaciones básicas
    if ($formData['nombre'] === '') {
        $errors[] = 'El nombre de la playlist es obligatorio.';
    }
}

```

```

    }

    // 3. Manejo de imagen (opcional)
    $imagePath = null;

    if (!empty($_FILES['imagen']['name']) && $_FILES['imagen']['error'] !==
    UPLOAD_ERR_NO_FILE) {
        if ($_FILES['imagen']['error'] === UPLOAD_ERR_OK) {
            $uploadDir = __DIR__ . '/uploads/playlists/';
            if (!is_dir($uploadDir)) {
                mkdir($uploadDir, 0777, true);
            }

            $ext = strtolower(pathinfo($_FILES['imagen']['name'],
    PATHINFO_EXTENSION));
            $allowed = ['jpg', 'jpeg', 'png', 'gif', 'webp'];

            if (!in_array($ext, $allowed)) {
                $errors[] = 'Formato de imagen no permitido para la
    playlist. Usa JPG, PNG, GIF o WEBP.';
            } else {
                $basename = 'pl_' . time() . '_' . mt_rand(1000, 9999) . '.'
    . $ext;

                $destPath = $uploadDir . $basename;
                $webPath = 'uploads/playlists/' . $basename; // ruta que se
    guarda en la BD

                if (move_uploaded_file($_FILES['imagen']['tmp_name'],
    $destPath)) {
                    $imagePath = $webPath;
                } else {
                    $errors[] = 'No se pudo subir la imagen de la
    playlist.';
                }
            }
        } else {
            $errors[] = 'Ocurrió un error al subir la imagen de la
    playlist.';
        }
    }

    // 4. Guardar si no hay errores
    if (empty($errors)) {
        $data = [

```

```

        'user_id'      => null, // en este proyecto no manejamos
usuarios reales
        'name'         => $formData['nombre'],
        'description' => $formData['descripcion'] !== '' ?
$formData['descripcion'] : null,
        'image_path' => $imagePath,
    ];

    if (createPlaylist($pdo, $data)) {
        header('Location: playlist.php?msg=creada');
        exit;
    } else {
        $errors[] = 'Ocurrió un error al crear la playlist en la base de
datos.';
    }
}
}

// Cargar vistas
require __DIR__ . '/views/partials/header.php';
require __DIR__ . '/views/crearPlaylistView.php';
require __DIR__ . '/views/partials/footer.php';

```

editar-playlist.php

```

<?php
// editar-playlist.php → Controlador para editar una playlist

require __DIR__ . '/config/db.php';
require __DIR__ . '/models/PlaylistModel.php';

$pageTitle = 'Editar Playlist - App Musical';
$activePage = 'playlist';

$errors = [];
$formData = [
    'nombre'      => '',
    'descripcion' => '',
];

// 1. Validar ID

```



```

if (empty($_GET['id']) || !ctype_digit($_GET['id'])) {
    header('Location: playlist.php');
    exit;
}

$id = (int) $_GET['id'];

// 2. Obtener playlist desde la BD
$playlist = getPlaylistById($pdo, $id);

if (!$playlist) {
    die("La playlist con ID $id no existe.");
}

// Imagen actual (puede ser null)
$currentImagePath = $playlist['image_path'] ?? null;

// 3. Obtener canciones asociadas actualmente a la playlist
$playlistSongIds = getPlaylistSongIds($pdo, $id);

// 4. Obtener todas las canciones disponibles para mostrarlas como opciones
$allSongs = [];
$stmt = $pdo->query("SELECT id, title, artist FROM songs ORDER BY title
ASC");
$allSongs = $stmt->fetchAll();

// 5. Rellenar formulario inicial con los datos de la playlist
$formData['nombre'] = $playlist['name'] ?? '';
$formData['descripcion'] = $playlist['description'] ?? '';

// 6. Procesar POST (si el usuario envía el formulario)
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Datos del formulario
    $formData['nombre'] = trim($_POST['nombre'] ?? '');
    $formData['descripcion'] = trim($_POST['descripcion'] ?? '');

    // Canciones seleccionadas (checkboxes)
    $selectedSongs = $_POST['songs'] ?? [];
    if (!is_array($selectedSongs)) {
        $selectedSongs = [];
    }

    // Validaciones básicas

```

```

    if ($formData['nombre'] === '') {
        $errors[] = 'El nombre de la playlist es obligatorio.';
    }

    // Manejo de imagen (opcional)
    // Por defecto se mantiene la imagen actual
    $imagePath = $currentImagePath;

    if (!empty($_FILES['imagen']['name']) && $_FILES['imagen']['error'] !==
    UPLOAD_ERR_NO_FILE) {
        if ($_FILES['imagen']['error'] === UPLOAD_ERR_OK) {
            $uploadDir = __DIR__ . '/uploads/playlists/';
            if (!is_dir($uploadDir)) {
                mkdir($uploadDir, 0777, true);
            }

            $ext = strtolower(pathinfo($_FILES['imagen']['name'],
    PATHINFO_EXTENSION));
            $allowed = ['jpg', 'jpeg', 'png', 'gif', 'webp'];

            if (!in_array($ext, $allowed)) {
                $errors[] = 'Formato de imagen no permitido para la
    playlist. Usa JPG, PNG, GIF o WEBP.';
            } else {
                $basename = 'pl_' . time() . '_' . mt_rand(1000, 9999) . '.'
    . $ext;

                $destPath = $uploadDir . $basename;
                $webPath = 'uploads/playlists/' . $basename;

                if (move_uploaded_file($_FILES['imagen']['tmp_name'],
    $destPath)) {
                    $imagePath = $webPath;
                } else {
                    $errors[] = 'No se pudo subir la nueva imagen de la
    playlist.';
                }
            }
        } else {
            $errors[] = 'Ocurrió un error al subir la imagen de la
    playlist.';
        }
    }

    // Si no hay errores, actualizar en la BD

```

```

        if (empty($errors)) {
            $data = [
                'name'          => $formData['nombre'],
                'description' => $formData['descripcion'] !== '' ?
$formData['descripcion'] : null,
                'image_path'   => $imagePath,
            ];

            if (updatePlaylist($pdo, $id, $data)) {
                // Actualizar canciones asociadas
                syncPlaylistSongs($pdo, $id, $selectedSongs);

                header('Location: playlist.php?msg=editada');
                exit;
            } else {
                $errors[] = 'Ocurrió un error al actualizar la playlist en la
base de datos.';
            }
        }

        // Si hubo errores, mantenemos la selección de canciones del POST
        $playlistSongIds = array_map('intval', $selectedSongs);
    }

    // 7. Cargar vistas
    require __DIR__ . '/views/partials/header.php';
    require __DIR__ . '/views/editarPlaylistView.php';
    require __DIR__ . '/views/partials/footer.php';

```

Ayuda.php

```

<?php
// ayuda.php → Página de ayuda

require __DIR__ . '/config/db.php'; // <- IMPORTANTE para que el header
tenga $pdo

$pageTitle = 'Ayuda - App Musical';

```

```
$activePage = 'ayuda';

// Cargar layout MVC
require __DIR__ . '/views/partials/header.php';
require __DIR__ . '/views/ayudaView.php'; // tu contenido de ayuda
require __DIR__ . '/views/partials/footer.php';
```

Perfil.php

```
<?php
// perfil.php → Perfil de usuario dummy con avatar

require __DIR__ . '/config/db.php';
require __DIR__ . '/models/UserModel.php';

$pageTitle = 'Mi Perfil - App Musical';
$activePage = 'perfil';

$errors = [];
$message = '';

// 1. Obtener perfil (id=1)
$userProfile = getUserProfile($pdo);
$currentAvatar = $userProfile['avatar_path'] ?? null;

// 2. Procesar subida de avatar si viene POST
if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    if (!empty($_FILES['avatar']['name']) && $_FILES['avatar']['error'] !==
    UPLOAD_ERR_NO_FILE) {
        if ($_FILES['avatar']['error'] === UPLOAD_ERR_OK) {
            $uploadDir = __DIR__ . '/uploads/user/';
            if (!is_dir($uploadDir)) {
                mkdir($uploadDir, 0777, true);
            }

            $ext = strtolower(pathinfo($_FILES['avatar']['name'],
            PATHINFO_EXTENSION));
            $allowed = ['jpg', 'jpeg', 'png', 'gif', 'webp'];

            if (!in_array($ext, $allowed)) {
```

```

        $errors[] = 'Formato de imagen no permitido. Usa JPG, PNG,
GIF o WEBP.';
    } else {
        $basename = 'avatar_' . time() . '_' . mt_rand(1000, 9999) .
        '.' . $ext;

        $destPath = $uploadDir . $basename;
        $webPath = 'uploads/user/' . $basename;

        if (move_uploaded_file($_FILES['avatar']['tmp_name'],
$destPath)) {
            if (updateUserAvatar($pdo, $webPath)) {
                $currentAvatar = $webPath;
                $message = 'Avatar actualizado correctamente.';
            } else {
                $errors[] = 'No se pudo guardar el avatar en la base
de datos.';
            }
        } else {
            $errors[] = 'No se pudo subir el archivo de avatar.';
        }
    }
} else {
    $errors[] = 'Ocurrió un error al subir el avatar.';
}
} else {
    $errors[] = 'No seleccionaste ninguna imagen para el avatar.';
}
}
}

```

```

// 3. Cargar vistas
require __DIR__ . '/views/partials/header.php';
require __DIR__ . '/views/perfilView.php';
require __DIR__ . '/views/partials/footer.php';

```

SongModel.php

```

<?php
// models/SongModel.php

/**
 * Devuelve las canciones más recientes para el inicio.
 */

```

```

function getRecentSongs(PDO $pdo, int $limit = 3): array
{
    $sql = "SELECT id, title, artist, genre, year, image_path
            FROM songs
            ORDER BY created_at DESC
            LIMIT :limit";

    $stmt = $pdo->prepare($sql);
    $stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
    $stmt->execute();

    return $stmt->fetchAll();
}

/**
 * Lista canciones con filtros opcionales (buscar, género, artista, año).
 *
 * $filters = [
 *     'search' => 'texto a buscar en título o artista',
 *     'genre'   => 'Rock' | 'Pop' | 'Todos' | '',
 *     'artist'  => 'nombre artista' | 'Todos' | '',
 *     'year'    => '2024' | '2023' | 'Todos' | ''
 * ]
 */
function getSongs(PDO $pdo, array $filters = []): array
{
    $where = [];
    $params = [];

    $search = trim($filters['search'] ?? '');
    $genre   = trim($filters['genre']   ?? '');
    $artist  = trim($filters['artist']  ?? '');
    $year    = trim($filters['year']    ?? '');

    if ($search !== '') {
        $where[] = "(title LIKE :search OR artist LIKE :search)";
        $params[':search'] = '%' . $search . '%';
    }

    if ($genre !== '' && $genre !== 'Todos') {
        $where[] = "genre = :genre";
        $params[':genre'] = $genre;
    }
}

```

```

        if ($artist !== '' && $artist !== 'Todos') {
            $where[] = "artist = :artist";
            $params[':artist'] = $artist;
        }

        if ($year !== '' && $year !== 'Todos') {
            $where[] = "year = :year";
            $params[':year'] = (int)$year;
        }

        $sql = "SELECT id, title, artist, genre, year, image_path
                FROM songs";

        if ($where) {
            $sql .= " WHERE " . implode(' AND ', $where);
        }

        $sql .= " ORDER BY id DESC";

        $stmt = $pdo->prepare($sql);
        $stmt->execute($params);

        return $stmt->fetchAll();
    }

    /**
     * Elimina una canción por ID.
     */
    function deleteSong(PDO $pdo, int $id): bool
    {
        $stmt = $pdo->prepare("DELETE FROM songs WHERE id = :id");
        return $stmt->execute([':id' => $id]);
    }

    /**
     * Crea una nueva canción en la base de datos.
     */
    function createSong(PDO $pdo, array $data): bool
    {
        $sql = "INSERT INTO songs (title, artist, genre, year, album, duration,
                                image_path)
                VALUES (:title, :artist, :genre, :year, :album, :duration,
                        :image_path)";
    }

```

```

$stmt = $pdo->prepare($sql);

return $stmt->execute([
    ':title'      => $data['title'],
    ':artist'     => $data['artist'],
    ':genre'      => $data['genre'] ?? null,
    ':year'       => $data['year'] ?? null,
    ':album'      => $data['album'] ?? null,
    ':duration'   => $data['duration'] ?? null,
    ':image_path' => $data['image_path'] ?? null,
]);
}

/**
 * Obtiene una canción por ID.
 */
function getSongById(PDO $pdo, int $id): ?array
{
    $stmt = $pdo->prepare("SELECT * FROM songs WHERE id = :id");
    $stmt->execute([':id' => $id]);
    $song = $stmt->fetch();

    return $song ?: null;
}

/**
 * Actualiza una canción existente.
 */
function updateSong(PDO $pdo, int $id, array $data): bool
{
    $sql = "UPDATE songs
        SET title      = :title,
            artist     = :artist,
            genre      = :genre,
            year       = :year,
            album      = :album,
            duration   = :duration,
            image_path = :image_path
        WHERE id = :id";

    $stmt = $pdo->prepare($sql);

    return $stmt->execute([

```



```

        ':title'      => $data['title'],
        ':artist'     => $data['artist'],
        ':genre'      => $data['genre'] ?? null,
        ':year'       => $data['year'] ?? null,
        ':album'      => $data['album'] ?? null,
        ':duration'   => $data['duration'] ?? null,
        ':image_path' => $data['image_path'] ?? null,
        ':id'         => $id,
    ];
}

/**
 * Devuelve lista de artistas distintos (para el combo de filtros).
 */
function getDistinctArtists(PDO $pdo): array
{
    $stmt = $pdo->query("SELECT DISTINCT artist FROM songs WHERE artist IS NOT NULL AND artist <> '' ORDER BY artist ASC");
    return $stmt->fetchAll(PDO::FETCH_COLUMN);
}

/**
 * Devuelve lista de años distintos (para el combo de filtros).
 */
function getDistinctYears(PDO $pdo): array
{
    $stmt = $pdo->query("SELECT DISTINCT year FROM songs WHERE year IS NOT NULL ORDER BY year DESC");
    return $stmt->fetchAll(PDO::FETCH_COLUMN);
}

```

PlaylistModel.php

```

<?php
// models/PlaylistModel.php

/**
 * Obtiene todas las playlists con el conteo de canciones.
 */
function getPlaylists(PDO $pdo): array

```

```

{
    $sql = "
        SELECT
            p.id,
            p.name,
            p.description,
            p.image_path,
            p.created_at,
            COUNT(ps.song_id) AS song_count
        FROM playlists p
        LEFT JOIN playlist_song ps ON p.id = ps.playlist_id
        GROUP BY p.id, p.name, p.description, p.image_path, p.created_at
        ORDER BY p.id DESC
    ";

    $stmt = $pdo->query($sql);
    return $stmt->fetchAll();
}

/**
 * Elimina una playlist por ID.
 * (Las relaciones en playlist_song se borran por la FK ON DELETE CASCADE)
 */
function deletePlaylist(PDO $pdo, int $id): bool
{
    $stmt = $pdo->prepare("DELETE FROM playlists WHERE id = :id");
    return $stmt->execute([':id' => $id]);
}

/**
 * Crea una nueva playlist.
 *
 * $data = [
 *     'user_id'      => int|null,
 *     'name'         => string,
 *     'description'  => string|null,
 *     'image_path'   => string|null
 * ]
 */
function createPlaylist(PDO $pdo, array $data): bool
{
    $sql = "INSERT INTO playlists (user_id, name, description, image_path)
        VALUES (:user_id, :name, :description, :image_path)";

```

```

$stmt = $pdo->prepare($sql);

return $stmt->execute([
    ':user_id'    => $data['user_id']    ?? null,
    ':name'       => $data['name'],
    ':description' => $data['description'] ?? null,
    ':image_path' => $data['image_path'] ?? null,
]);
}

/**
 * Obtiene una playlist por ID.
 */
function getPlaylistById(PDO $pdo, int $id): ?array
{
    $stmt = $pdo->prepare("SELECT * FROM playlists WHERE id = :id");
    $stmt->execute([':id' => $id]);
    $row = $stmt->fetch();

    return $row ?: null;
}

/**
 * Actualiza una playlist existente.
 *
 * $data = [
 *     'name'          => string,
 *     'description'   => string|null,
 *     'image_path'    => string|null
 * ]
 */
function updatePlaylist(PDO $pdo, int $id, array $data): bool
{
    $sql = "UPDATE playlists
        SET name          = :name,
            description    = :description,
            image_path     = :image_path
        WHERE id = :id";

    $stmt = $pdo->prepare($sql);

    return $stmt->execute([
        ':name'          => $data['name'],
        ':description'   => $data['description'] ?? null,
    ]);
}

```

```

        ':image_path' => $data['image_path'] ?? null,
        ':id'         => $id,
    ]);
}

/**
 * Devuelve los IDs de canciones asociadas a una playlist (para checkboxes).
 */
function getPlaylistSongIds(PDO $pdo, int $playlistId): array
{
    $stmt = $pdo->prepare("SELECT song_id FROM playlist_song WHERE
playlist_id = :id");
    $stmt->execute([':id' => $playlistId]);

    return $stmt->fetchAll(PDO::FETCH_COLUMN);
}

/**
 * Sincroniza las canciones de una playlist:
 * borra las actuales y agrega las nuevas (si se usan checkboxes).
 *
 * @param int    $playlistId
 * @param int[]  $songIds
 */
function syncPlaylistSongs(PDO $pdo, int $playlistId, array $songIds): bool
{
    // Borrar relaciones actuales
    $stmt = $pdo->prepare("DELETE FROM playlist_song WHERE playlist_id =
:id");
    $stmt->execute([':id' => $playlistId]);

    // Insertar nuevas
    if (empty($songIds)) {
        return true;
    }

    $stmt = $pdo->prepare("INSERT INTO playlist_song (playlist_id, song_id)
VALUES (:playlist_id, :song_id)");

    foreach ($songIds as $songId) {
        if (!ctype_digit((string)$songId)) {
            continue;
        }
        $stmt->execute([

```

```

        ':playlist_id' => $playlistId,
        ':song_id'      => (int)$songId,
    ]);
}

return true;
}

```

UserModel.php

```

<?php
// models/UserModel.php

/**
 * Devuelve el perfil del usuario dummy (id 1).
 * Si no existe, intenta crearlo vacío.
 */
function getUserProfile(PDO $pdo): array
{
    $stmt = $pdo->prepare("SELECT * FROM user_profile WHERE id = 1");
    $stmt->execute();
    $row = $stmt->fetch();

    if (!$row) {
        $pdo->exec("INSERT INTO user_profile (id, avatar_path) VALUES (1,
NULL)");
        $stmt = $pdo->prepare("SELECT * FROM user_profile WHERE id = 1");
        $stmt->execute();
        $row = $stmt->fetch();
    }

    return $row ?: ['id' => 1, 'avatar_path' => null];
}

/**
 * Actualiza solo el avatar del usuario dummy.
 */
function updateUserAvatar(PDO $pdo, string $avatarPath): bool
{

```

```

    $stmt = $pdo->prepare("UPDATE user_profile SET avatar_path = :avatar
WHERE id = 1");
    return $stmt->execute([':avatar' => $avatarPath]);
}

```

Db.php

```

<?php
// config/db.php

// Detectamos si estamos en InfinityFree (servidor) o en XAMPP (localhost)
$isProduction = (isset($_SERVER['HTTP_HOST']) && $_SERVER['HTTP_HOST'] !==
'localhost');

// Configuración dependiendo del entorno
if ($isProduction) {
    // === DATOS DE INFINITYFREE ===
    $host = 'sql303.infinityfree.com';
    $db = 'if0_40584663_app_musical';
    $user = 'if0_40584663';
    $pass = 'PROYECTO12345';
} else {
    // === DATOS DE XAMPP (LOCALHOST) ===
    $host = 'localhost';
    $db = 'app_musical';
    $user = 'root';
    $pass = '';
}

// Construcción del DSN
$dsn = "mysql:host=$host;dbname=$db;charset=utf8mb4";

// Opciones de PDO
$options = [
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
];

// Intento de conexión
try {
    $pdo = new PDO($dsn, $user, $pass, $options);
}

```

```

} catch (PDOException $e) {
    die('Error de conexión a la base de datos: ' . $e->getMessage());
}

```

Vistas

EditarPlaylistView.php

```

<?php
// views/editarPlaylistView.php
?>

<!-- Breadcrumb -->
<nav class="breadcrumb">
    <a href="index.php">Inicio</a>
    <span>/</span>
    <a href="playlist.php">Playlist</a>
    <span>/</span>
    <span class="current">Editar playlist</span>
</nav>

<h1 class="page-title">Editar Playlist</h1>

<div class="form-container">

    <?php if (!empty($errors)): ?>
        <ul style="color: red; margin-bottom: 15px;">
            <?php foreach ($errors as $e): ?>
                <li><?= htmlspecialchars($e) ?></li>
            <?php endforeach; ?>
        </ul>
    <?php endif; ?>

    <form class="playlist-form" method="post" action=""
enctype="multipart/form-data">
        <!-- Nombre -->
        <div class="form-group">
            <label for="nombre" class="form-label">Nombre de la
playlist</label>
            <input

```

```

        type="text"
        id="nombre"
        name="nombre"
        class="form-input"
        value="<?= htmlspecialchars($formData['nombre']) ?>"
    >
</div>

<!-- Descripción -->
<div class="form-group">
    <label for="descripcion" class="form-label">Descripción
(opcional)</label>
    <textarea
        id="descripcion"
        name="descripcion"
        class="form-textarea"
        rows="3"
    ><?= htmlspecialchars($formData['descripcion']) ?></textarea>
</div>

<!-- Imagen -->
<div class="form-group">
    <label for="imagen" class="form-label">Imagen de la playlist
(opcional)</label>
    <input
        type="file"
        id="imagen"
        name="imagen"
        class="form-input"
        accept="image/*"
    >
    <p style="font-size: 0.85rem; color: #666; margin-top: 4px;">
        Si no seleccionas una nueva imagen, se mantendrá la actual
(o la tarjeta gris por defecto).
    </p>

    <?php if (!empty($currentImagePath)): ?>
        <div style="margin-top: 10px;">
            <span style="font-size: 0.85rem; color: #666;">Imagen
actual:</span>
            <div style="margin-top: 5px; width: 120px; height:
120px; border-radius: 8px; overflow: hidden; background: #eee;">
                <img src="<?= htmlspecialchars($currentImagePath)
?>" alt="Imagen actual"

```



```

        style="width: 100%; height: 100%; object-fit:
cover;">
    </div>
</div>
<?php endif; ?>
</div>

<!-- Canciones de la playlist -->
<div class="form-group">
    <label class="form-label">Canciones en la playlist</label>
    <p style="font-size: 0.85rem; color: #666; margin-bottom: 8px;">
        Marca las canciones que quieres que formen parte de esta
playlist.
    </p>

    <?php if (empty($allSongs)): ?>
        <p style="font-size: 0.9rem; color: #999;">
            No hay canciones registradas en el sistema. Primero
agrega canciones en la sección
            <strong>Canciones</strong>.
        </p>
    <?php else: ?>
        <div class="songs-checkbox-list">
            <?php foreach ($allSongs as $song): ?>
                <?php
                    $checked = in_array((int)$song['id'],
$playlistSongIds, true) ? 'checked' : '';
                    $label    = $song['title'];
                    if (!empty($song['artist'])) {
                        $label .= ' - ' . $song['artist'];
                    }
                >
                <label class="checkbox-item">
                    <input
                        type="checkbox"
                        name="songs[]"
                        value="<?= (int)$song['id'] ?>"
                        <?= $checked ?>
                    >
                    <span><?= htmlspecialchars($label) ?></span>
                </label>
            <?php endforeach; ?>
        </div>
    <?php endif; ?>

```

```

        </div>

        <!-- Botones -->
        <div class="form-actions">
            <button
                type="button"
                class="btn-cancel"
                onclick="window.location.href='playlist.php'"
            >
                Cancelar
            </button>

            <button type="submit" class="btn-submit">
                Guardar cambios
            </button>
        </div>
    </form>
</div>

```

PerfilView.php

```

<?php
// views/perfilView.php
?>

<!-- Breadcrumb -->
<nav class="breadcrumb">
    <a href="index.php">Inicio</a>
    <span>/</span>
    <span class="current">Perfil</span>
</nav>

<h1 class="page-title">Mi Perfil</h1>

<div class="profile-layout">
    <!-- Columna izquierda: avatar y resumen -->
    <aside class="profile-sidebar">
        <div class="profile-card">
            <!-- Avatar grande -->
            <div
                class="profile-avatar-large"
                <?php if (!empty($currentAvatar)): ?>

```

```

        style="background-image: url('<?=
htmlspecialchars($currentAvatar) ?>');
        background-size: cover;
        background-position: center;"
    <?php endif; ?>
</div>

<!-- Form para cambiar foto SOLO avatar -->
<form method="post" action="" enctype="multipart/form-data"
style="margin-top: 15px;">
    <input
        type="file"
        id="avatar"
        name="avatar"
        accept="image/*"
        style="display: none;"
    >
    <label for="avatar" class="btn-secondary" style="display:
inline-block; text-align: center; cursor: pointer;">
        Cambiar foto
    </label>

    <button type="submit" class="btn-submit" style="margin-top:
10px; width: 100%;">
        Guardar avatar
    </button>
</form>

<?php if (!empty($message)): ?>
    <p style="color: green; margin-top: 10px; font-size:
0.9rem;">
        <?= htmlspecialchars($message) ?>
    </p>
<?php endif; ?>

<?php if (!empty($errors)): ?>
    <ul style="color: red; margin-top: 10px; font-size:
0.9rem;">
        <?php foreach ($errors as $e): ?>
            <li><?= htmlspecialchars($e) ?></li>
        <?php endforeach; ?>
    </ul>
<?php endif; ?>

```

```

        <!-- Info rápida dummy -->
        <div class="profile-summary">
            <h3 class="section-subtitle">Resumen</h3>
            <p><strong>Usuario:</strong> Usuario</p>
            <p><strong>Plan:</strong> Premium</p>
            <p><strong>Miembro desde:</strong> 2024</p>
        </div>
    </div>
</aside>

<!-- Columna derecha: formulario dummy (solo visual) -->
<section class="profile-main">
    <div class="form-container">
        <h2 class="section-title">Información de la cuenta</h2>

        <!-- ESTE FORM ES SOLO VISUAL, NO SE GUARDA NADA -->
        <form class="profile-form">
            <!-- Nombre de usuario -->
            <div class="form-group">
                <label for="username" class="form-label">Nombre de
usuario</label>
                <input type="text" id="username" name="username"
class="form-input" value="Usuario">
            </div>

            <!-- Correo electrónico -->
            <div class="form-group">
                <label for="email" class="form-label">Correo
electrónico</label>
                <input type="email" id="email" name="email" class="form-
input" value="usuario@ejemplo.com">
            </div>

            <!-- Nombre completo -->
            <div class="form-group">
                <label for="fullname" class="form-label">Nombre
completo</label>
                <input type="text" id="fullname" name="fullname"
class="form-input" value="Nombre Apellido">
            </div>

            <!-- Fecha de nacimiento -->
            <div class="form-group">

```

```

        <label for="birthdate" class="form-label">Fecha de
nacimiento</label>
        <input type="date" id="birthdate" name="birthdate"
class="form-input" value="2000-01-01">
    </div>

    <!-- País -->
    <div class="form-group">
        <label for="country" class="form-label">País</label>
        <select id="country" name="country" class="form-select
form-select-filled">
            <option value="panama" selected>Panamá</option>
            <option value="mexico">México</option>
            <option value="colombia">Colombia</option>
            <option value="argentina">Argentina</option>
            <option value="españa">España</option>
        </select>
    </div>

    <!-- Separator -->
    <hr class="form-separator">

    <!-- Cambiar contraseña (solo visual) -->
    <h3 class="section-subtitle">Cambiar contraseña</h3>

    <div class="form-group">
        <label for="current-password" class="form-
label">Contraseña actual</label>
        <input type="password" id="current-password"
name="current-password" class="form-input" placeholder=".....">
    </div>

    <div class="form-group">
        <label for="new-password" class="form-label">Nueva
contraseña</label>
        <input type="password" id="new-password" name="new-
password" class="form-input" placeholder=".....">
    </div>

    <div class="form-group">
        <label for="confirm-password" class="form-
label">Confirmar nueva contraseña</label>
        <input type="password" id="confirm-password"
name="confirm-password" class="form-input" placeholder=".....">

```

```

        </div>

        <!-- Botones dummy -->
        <div class="form-actions">
            <button type="button" class="btn-cancel">
                Cancelar
            </button>

            <button type="button" class="btn-submit">
                Guardar cambios
            </button>
        </div>
    </form>
</div>
</section>
</div>

```

AyudaView.php

```

<?php
// views/ayudaView.php
?>

<!-- Breadcrumb -->
<nav class="breadcrumb">
    <a href="index.php">Inicio</a>
    <span></span>
    <span class="current">Ayuda</span>
</nav>

<!-- Page Title -->
<h1 class="page-title">Centro de Ayuda</h1>

<div class="form-container-wide">

    <!-- Guía rápida -->
    <section class="help-section">
        <h2 class="section-subtitle">Guía rápida</h2>
        <p>
            Esta aplicación te permite gestionar tu biblioteca musical,
            administrar playlists,

```

```
        editar canciones y visualizar la información de tu perfil. Aquí
encontrarás respuestas
        a las dudas más comunes sobre cómo usar cada sección.
    </p>
</section>

<hr class="form-separator">

<!-- Cómo usar la aplicación -->
<section class="help-section">
    <h2 class="section-subtitle">¿Cómo usar la aplicación?</h2>

    <h3 class="faq-title">Inicio</h3>
    <p class="faq-text">
        En la página de <strong>Inicio</strong> verás un resumen general
de la aplicación y,
        opcionalmente, una lista de canciones recientes o accesos
rápidos a las funciones
        principales como <strong>Canciones</strong> y
<strong>Playlist</strong>.
    </p>

    <h3 class="faq-title">Canciones</h3>
    <p class="faq-text">
        En la sección <strong>Canciones</strong> puedes ver el listado
de todas las canciones
        registradas en el sistema. Desde aquí puedes:
    </p>
    <ul class="contact-list">
        <li>Buscar canciones por título o artista.</li>
        <li>Filtrar por género musical.</li>
        <li>Agregar nuevas canciones.</li>
        <li>Editar o eliminar canciones existentes.</li>
    </ul>

    <h3 class="faq-title">Playlist</h3>
    <p class="faq-text">
        La sección de <strong>Playlist</strong> te permite crear listas
de reproducción personalizadas.
        Podrás ver tus playlists, crear nuevas y editar las existentes
para elegir qué canciones
        incluye cada una.
    </p>
```

```

    <h3 class="faq-title">Perfil</h3>
    <p class="faq-text">
        En <strong>Perfil</strong> se muestra información básica del
        usuario que está usando la
        aplicación. En este proyecto es solo una vista ilustrativa para
        dar la idea de un usuario
        logeado, por lo que los datos mostrados son de ejemplo.
    </p>
</section>

<hr class="form-separator">

<!-- Preguntas frecuentes -->
<section class="help-section">
    <h2 class="section-subtitle">Preguntas frecuentes</h2>

    <div class="faq-item">
        <h3 class="faq-title">¿Cómo agrego una canción nueva?</h3>
        <p class="faq-text">
            Ve a la sección <strong>Canciones</strong> y haz clic en
            <strong>Agregar canción</strong>. Completa el formulario con
            el título,
            artista, género, año y demás datos que desees registrar, y
            luego guarda
            la información.
        </p>
    </div>

    <div class="faq-item">
        <h3 class="faq-title">¿Cómo creo una playlist?</h3>
        <p class="faq-text">
            En la página <strong>Playlist</strong>, selecciona el botón
            <strong>Nueva Playlist</strong> (o <strong>Crear
            Playlist</strong>).
            Asigna un nombre y una descripción, guarda la playlist y
            luego podrás
            agregarle canciones desde la opción de
            <strong>Editar</strong>.
        </p>
    </div>

    <div class="faq-item">
        <h3 class="faq-title">¿Puedo editar mis playlists?</h3>
        <p class="faq-text">

```


Sí. En la sección **Playlist**, ubica la playlist que deseas modificar y haz clic en **Editar**. Desde ahí puedes cambiar su nombre, descripción y seleccionar las canciones que formarán parte de ella.

</p>
</div>

<div class="faq-item">
 <h3 class="faq-title">¿Los datos de perfil son reales?</h3>
 <p class="faq-text">

No. En este proyecto, la sección **Perfil** es únicamente decorativa. Los datos que se muestran son estáticos y sirven para simular cómo se vería un usuario logeado en una versión futura con sistema de inicio de sesión.

</p>
</div>
</section>

<hr class="form-separator">

<!-- Contacto -->
 <section class="help-section">
 <h2 class="section-subtitle">Contacto</h2>
 <p class="faq-text">

Para soporte técnico o dudas relacionadas con este proyecto académico, puedes comunicarte con el equipo responsable del curso de **Ingeniería Web**.

</p>

 <ul class="contact-list">
 Correo: Grupo9@Ingweb.com
 Grupo: Proyecto Final - Grupo 9
 Año: 2025

</section>

</div>

CrearPlaylistView.php

```
<?php
// views/crearPlaylistView.php
?>

<!-- Breadcrumb -->
<nav class="breadcrumb">
    <a href="index.php">Inicio</a>
    <span>/</span>
    <a href="playlist.php">Playlist</a>
    <span>/</span>
    <span class="current">Crear playlist</span>
</nav>

<h1 class="page-title">Crear Playlist</h1>

<div class="form-container">

    <?php if (!empty($errors)): ?>
        <ul style="color: red; margin-bottom: 15px;">
            <?php foreach ($errors as $e): ?>
                <li><?= htmlspecialchars($e) ?></li>
            <?php endforeach; ?>
        </ul>
    <?php endif; ?>

    <form class="playlist-form" method="post" action=""
enctype="multipart/form-data">
        <!-- Nombre -->
        <div class="form-group">
            <label for="nombre" class="form-label">Nombre de la
playlist</label>
            <input
                type="text"
                id="nombre"
                name="nombre"
                class="form-input"
                placeholder="Ejemplo: Rock para estudiar"
                value="<?= htmlspecialchars($formData['nombre']) ?>"
            >
        </div>

        <!-- Descripción -->
```

```

        <div class="form-group">
            <label for="descripcion" class="form-label">Descripción
(opcional)</label>
            <textarea
                id="descripcion"
                name="descripcion"
                class="form-textarea"
                placeholder="Breve descripción de la playlist"
                rows="3"
            ><?= htmlspecialchars($formData['descripcion']) ?></textarea>
        </div>

<!-- Imagen -->
<div class="form-group">
    <label for="imagen" class="form-label">Imagen de la playlist
(opcional)</label>
    <input
        type="file"
        id="imagen"
        name="imagen"
        class="form-input"
        accept="image/*"
    >
    <p style="font-size: 0.85rem; color: #666; margin-top: 4px;">
        Si no seleccionas una imagen, se mostrará la tarjeta gris
por defecto.
    </p>
</div>

<!-- Botones -->
<div class="form-actions">
    <button
        type="button"
        class="btn-cancel"
        onclick="window.location.href='playlist.php'"
    >
        Cancelar
    </button>

    <button type="submit" class="btn-submit">
        Guardar playlist
    </button>
</div>
</form>

```

```
</div>
```

PlaylistView.php

```
<?php
// views/playlistView.php
?>

<!-- Breadcrumb -->
<nav class="breadcrumb">
    <a href="index.php">Inicio</a>
    <span>/</span>
    <span class="current">Playlist</span>
</nav>

<!-- Page Header -->
<div class="page-header">
    <h1 class="page-title">Playlists</h1>
    <button class="btn-primary" onclick="window.location.href='crear-
playlist.php'">
        
        Crear playlist
    </button>
</div>

<?php if (!empty($message)): ?>
    <p style="color: green; margin-bottom: 15px;">
        <?= htmlspecialchars($message) ?>
    </p>
<?php endif; ?>

<!-- Grid de playlists -->
<div class="playlists-grid">
    <?php if (empty($playlists)): ?>
        <p>No hay playlists registradas.</p>
    <?php else: ?>
        <?php foreach ($playlists as $p): ?>
            <div class="playlist-card">
                <?php if (!empty($p['image_path'])): ?>
                    <div class="playlist-card-image">
```

```

        style="background-image: url('<?=
htmlspecialchars($p['image_path']) ?>');
        background-size: cover;
        background-position: center;">
    </div>
    <?php else: ?>
        <!-- Cuadro gris por defecto -->
        <div class="playlist-card-image"></div>
    <?php endif; ?>

    <h3 class="playlist-card-title">
        <?= htmlspecialchars($p['name']) ?>
    </h3>

    <?php if (!empty($p['description'])): ?>
        <p class="playlist-card-description">
            <?= htmlspecialchars($p['description']) ?>
        </p>
    <?php endif; ?>

    <p class="playlist-card-count">
        <?= (int)$p['song_count'] ?> canciones
    </p>

    <!-- Botón ROJO grande arriba -->
    <div class="playlist-card-actions-top">
        <button
            class="btn-danger btn-full"
            onclick="window.location.href='editar-
playlist.php?id=<?= $p['id'] ?>';"
        >
            Agregar canciones
        </button>
    </div>

    <!-- Botones inferiores: Eliminar / Editar -->
    <div class="playlist-card-actions">
        <button
            class="btn-outline"
            onclick="if(confirm('¿Eliminar esta playlist?')){
window.location.href='playlist.php?eliminar=<?= $p['id'] ?>'; }"
        >
            Eliminar
        </button>
    </div>

```

```

            <button
                class="btn-dark"
                onclick="window.location.href='editar-
playlist.php?id=<?= $p['id'] ?>';"
            >
                Editar
            </button>
        </div>
    </div>
    <?php endforeach; ?>
    <?php endif; ?>
</div>

```

EditarCancionView.php

```

<?php
// views/editarCancionView.php
?>

<!-- Breadcrumb -->
<nav class="breadcrumb">
    <a href="index.php">Inicio</a>
    <span></span>
    <a href="canciones.php">Canciones</a>
    <span></span>
    <span class="current">Editar canción</span>
</nav>

<h1 class="page-title">Editar Canción</h1>

<div class="form-container">
    <?php if (!empty($errors)): ?>
        <ul style="color: red; margin-bottom: 15px;">
            <?php foreach ($errors as $e): ?>
                <li><?= htmlspecialchars($e) ?></li>
            <?php endforeach; ?>
        </ul>
    <?php endif; ?>

    <form class="song-form" method="post" action="" enctype="multipart/form-
data">

```

```

<!-- Título -->
<div class="form-group">
    <label for="titulo" class="form-label">Título</label>
    <input
        type="text"
        id="titulo"
        name="titulo"
        class="form-input"
        value="<?= htmlspecialchars($formData['titulo']) ?>"
    >
</div>

<!-- Artista -->
<div class="form-group">
    <label for="artista" class="form-label">Artista</label>
    <input
        type="text"
        id="artista"
        name="artista"
        class="form-input"
        value="<?= htmlspecialchars($formData['artista']) ?>"
    >
</div>

<!-- Género -->
<div class="form-group">
    <label for="genero" class="form-label">Género</label>
    <select
        id="genero"
        name="genero"
        class="form-select form-select-filled"
    >
        <?php
            $generos = ['' => 'Selecciona un género', 'Rock' => 'Rock',
'Pop' => 'Pop',
                        'Jazz' => 'Jazz', 'Reggaeton' => 'Reggaeton',
'Electrónica' => 'Electrónica'];
            foreach ($generos as $value => $label):
                $selected = ($formData['genero'] === $value) ?
'selected' : '';
                ?>
                <option value="<?= htmlspecialchars($value) ?>" <?=
$selected ?>>
                    <?= htmlspecialchars($label) ?>

```

```

        </option>
        <?php endforeach; ?>
    </select>
</div>

<!-- Año -->
<div class="form-group">
    <label for="ano" class="form-label">Año</label>
    <input
        type="text"
        id="ano"
        name="ano"
        class="form-input"
        value="<?php htmlspecialchars($formData['ano']) ?>"
    >
</div>

<!-- Álbum -->
<div class="form-group">
    <label for="album" class="form-label">Álbum</label>
    <input
        type="text"
        id="album"
        name="album"
        class="form-input"
        value="<?php htmlspecialchars($formData['album']) ?>"
    >
</div>

<!-- Duración -->
<div class="form-group">
    <label for="duracion" class="form-label">Duración</label>
    <input
        type="text"
        id="duracion"
        name="duracion"
        class="form-input"
        value="<?php htmlspecialchars($formData['duracion']) ?>"
    >
</div>

<!-- Imagen -->
<div class="form-group">

```



```

        <label for="imagen" class="form-label">Imagen de la canción
(opcional)</label>
        <input
            type="file"
            id="imagen"
            name="imagen"
            class="form-input"
            accept="image/*"
        >
        <p style="font-size: 0.85rem; color: #666; margin-top: 4px;">
            Si no seleccionas una nueva imagen, se mantendrá la actual
(o la tarjeta gris por defecto).
        </p>

        <?php if (!empty($currentImagePath)): ?>
            <div style="margin-top: 10px;">
                <span style="font-size: 0.85rem; color: #666;">Imagen
actual:</span>
                <div style="margin-top: 5px; width: 120px; height:
120px; border-radius: 8px; overflow: hidden; background: #eee;">
                    
                </div>
            </div>
        <?php endif; ?>
    </div>

    <!-- Botones -->
    <div class="form-actions">
        <button
            type="button"
            class="btn-cancel"
            onclick="window.location.href='canciones.php'"
        >
            Cancelar
        </button>

        <button type="submit" class="btn-submit">
            Guardar cambios
        </button>
    </div>
</form>

```

```
</div>
```

AgregarCancionView.php

```
<?php
// views/agregarCancionView.php
?>

<!-- Breadcrumb -->
<nav class="breadcrumb">
    <a href="index.php">Inicio</a>
    <span>/</span>
    <a href="canciones.php">Canciones</a>
    <span>/</span>
    <span class="current">Agregar canción</span>
</nav>

<h1 class="page-title">Agregar Canción</h1>

<div class="form-container">
    <?php if (!empty($errors)): ?>
        <ul style="color: red; margin-bottom: 15px;">
            <?php foreach ($errors as $e): ?>
                <li><?= htmlspecialchars($e) ?></li>
            <?php endforeach; ?>
        </ul>
    <?php endif; ?>

    <form class="song-form" method="post" action="" enctype="multipart/form-
data">
        <!-- Título -->
        <div class="form-group">
            <label for="titulo" class="form-label">Título</label>
            <input
                type="text"
                id="titulo"
                name="titulo"
                class="form-input"
                placeholder="Título de la canción"
                value="<?= htmlspecialchars($oldData['titulo']) ?>"
            >
```

```

</div>

<!-- Artista -->
<div class="form-group">
    <label for="artista" class="form-label">Artista</label>
    <input
        type="text"
        id="artista"
        name="artista"
        class="form-input"
        placeholder="Nombre del artista o banda"
        value="<?= htmlspecialchars($oldData['artista']) ?>"
    >
</div>

<!-- Género -->
<div class="form-group">
    <label for="genero" class="form-label">Género</label>
    <select
        id="genero"
        name="genero"
        class="form-select form-select-filled"
    >
        <?php
            $generos = ['' => 'Selecciona un género', 'Rock' => 'Rock',
                'Pop' => 'Pop',
                'Jazz' => 'Jazz', 'Reggaeton' => 'Reggaeton',
                'Electrónica' => 'Electrónica'];
            foreach ($generos as $value => $label):
                $selected = ($oldData['genero'] === $value) ? 'selected'
: '';
                ?>
                <option value="<?= htmlspecialchars($value) ?>" <?=
$selected ?>>
                    <?= htmlspecialchars($label) ?>
                </option>
            <?php endforeach; ?>
        </select>
    </div>

<!-- Año -->
<div class="form-group">
    <label for="ano" class="form-label">Año</label>
    <input

```

```

        type="text"
        id="ano"
        name="ano"
        class="form-input"
        placeholder="Año de lanzamiento"
        value="<?= htmlspecialchars($oldData['ano']) ?>"
    >
</div>

<!-- Álbum -->
<div class="form-group">
    <label for="album" class="form-label">Álbum</label>
    <input
        type="text"
        id="album"
        name="album"
        class="form-input"
        placeholder="Nombre del álbum (opcional)"
        value="<?= htmlspecialchars($oldData['album']) ?>"
    >
</div>

<!-- Duración -->
<div class="form-group">
    <label for="duracion" class="form-label">Duración</label>
    <input
        type="text"
        id="duracion"
        name="duracion"
        class="form-input"
        placeholder="Ejemplo: 03:45"
        value="<?= htmlspecialchars($oldData['duracion']) ?>"
    >
</div>

<!-- Imagen -->
<div class="form-group">
    <label for="imagen" class="form-label">Imagen de la canción
(opcional)</label>
    <input
        type="file"
        id="imagen"
        name="imagen"
        class="form-input"

```

```

        accept="image/*"
    >
    <p style="font-size: 0.85rem; color: #666; margin-top: 4px;">
        Si no seleccionas una imagen, se mostrará la tarjeta gris
por defecto.
    </p>
</div>

<!-- Botones -->
<div class="form-actions">
    <button
        type="button"
        class="btn-cancel"
        onclick="window.location.href='canciones.php'"
    >
        Cancelar
    </button>

    <button type="submit" class="btn-submit">
        Guardar canción
    </button>
</div>
</form>
</div>

```

CancionesView.php

```

<?php
// views/cancionesView.php
?>

<!-- Breadcrumb -->
<nav class="breadcrumb">
    <a href="index.php">Inicio</a>
    <span>/</span>
    <a href="canciones.php">Canciones</a>
    <span>/</span>
    <span class="current">Ver canciones</span>
</nav>

<!-- Page Header -->

```

```

<div class="page-header">
  <h1 class="page-title">Ver Canciones</h1>
  <button class="btn-primary" onclick="window.location.href='agregar-
cancion.php'">
    
    Agregar canción
  </button>
</div>

<?php if (!empty($message)): ?>
  <p style="color: green; margin-bottom: 15px;">
    <?= htmlspecialchars($message) ?>
  </p>
<?php endif; ?>

<!-- FORM DE FILTROS -->
<form method="get" action="canciones.php">

  <!-- Barra de búsqueda LARGA + botón afuera a la derecha -->
  <div class="search-filters">
    <!-- Marco de la barra -->
    <div class="search-bar">
      <input
        type="text"
        name="buscar"
        placeholder="Buscar por título o artista"
        class="search-input"
        value="<?= htmlspecialchars($search) ?>"
      >
      <button class="search-icon" type="submit">
        
      </button>
    </div>

    <!-- Botón separado, con espacio respecto a la barra -->
    <button class="btn-secondary" type="submit" style="margin-left:
10px;">
      Aplicar filtros
    </button>
  </div>

  <!-- Filtros ABAJO (Género, Artista, Año) -->
  <div class="filters-row">
    <!-- Género -->

```

```

<div class="filter-group">
    <label>Género</label>
    <select class="filter-select" name="genero">
        <?php
            $generos = ['Todos', 'Rock', 'Pop', 'Jazz', 'Reggaeton',
'Electrónica'];
            foreach ($generos as $g):
                $selected = ($genre === $g || ($genre === '' && $g ===
'Todos')) ? 'selected' : '';
                ?>
                    <option value="<?= htmlspecialchars($g) ?>" <?=
$selected ?>>
                        <?= htmlspecialchars($g) ?>
                    </option>
            <?php endforeach; ?>
    </select>
</div>

<!-- Artista -->
<div class="filter-group">
    <label>Artista</label>
    <select class="filter-select" name="artista">
        <option value="Todos">Todos</option>
        <?php foreach ($artistsList as $art): ?>
            <?php $selected = ($artist === $art) ? 'selected' : '';
            ?>
                <option value="<?= htmlspecialchars($art) ?>" <?=
$selected ?>>
                    <?= htmlspecialchars($art) ?>
                </option>
        <?php endforeach; ?>
    </select>
</div>

<!-- Año -->
<div class="filter-group">
    <label>Año</label>
    <select class="filter-select" name="ano">
        <option value="Todos">Todos</option>
        <?php foreach ($yearsList as $y): ?>
            <?php $selected = ((string)$year === (string)$y) ?
'selected' : ''; ?>
                <option value="<?= htmlspecialchars($y) ?>" <?=
$selected ?>>

```

```

        <?= htmlspecialchars($y) ?>
    </option>
    <?php endforeach; ?>
</select>
</div>
</div>
</div>
</form>

<!-- Songs Grid -->
<div class="songs-grid">
    <?php if (empty($canciones)): ?>
        <p>No hay canciones que coincidan con los filtros.</p>
    <?php else: ?>
        <?php foreach ($canciones as $c): ?>
            <div class="song-card">
                <?php if (!empty($c['image_path'])): ?>
                    <div class="song-card-image"
                        style="background-image: url('<?=
htmlspecialchars($c['image_path']) ?>');
                        background-size: cover;
                        background-position: center;">

                </div>
                <?php else: ?>
                    <!-- Cuadro gris por defecto -->
                    <div class="song-card-image"></div>
                <?php endif; ?>

                <h3 class="song-card-title">
                    <?= htmlspecialchars($c['title']) ?>
                </h3>
                <p class="song-card-artist">
                    <?= htmlspecialchars($c['artist']) ?>
                </p>
                <p class="song-card-info">
                    <?= htmlspecialchars($c['genre'] ?? '') ?>
                    <?php if (!empty($c['year'])): ?>
                        - <?= htmlspecialchars($c['year']) ?>
                    <?php endif; ?>
                </p>
                <div class="song-card-actions">
                    <button
                        class="btn-outline"
                        onclick="if(confirm('¿Eliminar esta canción?')){
window.location.href='canciones.php?eliminar=<?= $c['id'] ?>'; }"

```



```

        >
        Eliminar
    </button>

    <button
        class="btn-dark"
        onclick="window.location.href='editar-
cancion.php?id=<?= $c['id'] ?>';"
    >
        Editar
    </button>
</div>
</div>
<?php endforeach; ?>
<?php endif; ?>
</div>

```

IndexView.php

```

<?php
// views/indexView.php
?>

<h1 class="greeting">Buenas tardes</h1>

<!-- Tarjetas grandes -->
<div class="cards-container">
    <!-- Card Canciones -->
    <div class="card" onclick="window.location.href='canciones.php'">
        <?php if (!empty($firstSong) && !empty($firstSong['image_path'])):
?>
            <div class="card-image"
                style="background-image: url('<?=
htmlspecialchars($firstSong['image_path']) ?>');
                background-size: cover;
                background-position: center;">
            </div>
        <?php else: ?>
            <div class="card-image"></div>
        <?php endif; ?>
    </div>

```

```

        <h3 class="card-title">Canciones</h3>
        <p class="card-description">Gestiona tu biblioteca musical</p>
    </div>

    <!-- Card Playlist -->
    <div class="card" onclick="window.location.href='playlist.php'">
        <?php if (!empty($firstPlaylist) &&
!empty($firstPlaylist['image_path'])): ?>
            <div class="card-image"
                style="background-image: url('<?=
htmlspecialchars($firstPlaylist['image_path']) ?>');
                background-size: cover;
                background-position: center;">

            </div>
            <?php else: ?>
                <div class="card-image"></div>
            <?php endif; ?>

        <h3 class="card-title">Playlist</h3>
        <p class="card-description">Organiza tus favoritas</p>
    </div>

    <!-- Card Perfil -->
    <div class="card" onclick="window.location.href='perfil.php'">
        <?php if (!empty($headerAvatar)): ?>
            <div class="card-image"
                style="background-image: url('<?=
htmlspecialchars($headerAvatar) ?>');
                background-size: cover;
                background-position: center;">

            </div>
            <?php else: ?>
                <div class="card-image"></div>
            <?php endif; ?>

        <h3 class="card-title">Perfil</h3>
        <p class="card-description">Configuración y datos</p>
    </div>
</div>

<!-- Canciones recientes -->
<section class="recent-section">
    <h2 class="section-title">Canciones Recientes</h2>

```

```

<?php if (empty($recentSongs)): ?>
    <p>No hay canciones registradas todavía.</p>
<?php else: ?>
    <div class="song-list">
        <?php foreach ($recentSongs as $song): ?>
            <div class="song-item">
                <div class="song-title">
                    <?= htmlspecialchars($song['title']) ?>
                </div>
                <div class="song-info">
                    <?= htmlspecialchars($song['artist'] ?? 'Artista
desconocido') ?>

                    <?php if (!empty($song['genre'])): ?>
                        - <?= htmlspecialchars($song['genre']) ?>
                    <?php endif; ?>
                    <?php if (!empty($song['year'])): ?>
                        - <?= htmlspecialchars($song['year']) ?>
                    <?php endif; ?>
                </div>
            </div>
        <?php endforeach; ?>
    </div>
<?php endif; ?>
</section>

```

Header.php

```

<?php
// views/partials/header.php

// Obtener avatar para mostrarlo en la esquina superior derecha
$headerAvatar = null;

if (isset($pdo)) {
    require_once __DIR__ . '/../../models/UserModel.php';

    try {
        $profileHeader = getUserProfile($pdo);
        $headerAvatar = $profileHeader['avatar_path'] ?? null;
    } catch (Throwable $e) {

```

```

        $headerAvatar = null; // por si hay algún problema, no romper el
header
    }
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title><?= htmlspecialchars($pageTitle ?? 'App Musical') ?></title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
<header>
    <!-- LOGO ESTÁTICO: ahora con imagen -->
    <div class="logo">
        
    </div>

    <!-- Navegación principal -->
    <nav>
        <a href="index.php"      class="<?= ($activePage ?? '') === 'inicio'
? 'active' : '' ?>">Inicio</a>
        <a href="canciones.php" class="<?= ($activePage ?? '') ===
'canciones' ? 'active' : '' ?>">Canciones</a>
        <a href="playlist.php"  class="<?= ($activePage ?? '') ===
'playlist' ? 'active' : '' ?>">Playlist</a>
        <a href="perfil.php"    class="<?= ($activePage ?? '') === 'perfil'
? 'active' : '' ?>">Perfil</a>
        <a href="ayuda.php"     class="<?= ($activePage ?? '') === 'ayuda'
? 'active' : '' ?>">Ayuda</a>
    </nav>

    <!-- USER DROPDOWN MENU -->
    <div class="user-menu-container">
        <div class="user-profile" onclick="toggleUserMenu()">
            <!-- Avatar en la esquina superior derecha -->
            <div
                class="user-avatar"
                <?php if (!empty($headerAvatar)): ?>
                    style="background-image: url('<?=
htmlspecialchars($headerAvatar) ?>');
                    background-size: cover;

```

```

                background-position: center;"
            <?php endif; ?>
        ></div>
        <span class="user-name">Usuario</span>
    </div>

    <div id="userDropdown" class="user-dropdown">
        <a href="perfil.php">Editar Perfil</a>
    </div>
</div>
</header>

<main class="main-content">

```

Footer.php

```

</main>

<footer>
    © 2025 Ingeniería Web - Proyecto Final
</footer>

<script>
    function toggleUserMenu() {
        const menu = document.getElementById("userDropdown");
        menu.style.display = (menu.style.display === "flex") ? "none" :
"flex";
    }

    document.addEventListener("click", function(event) {
        const userMenu = document.querySelector(".user-menu-container");
        const dropdown = document.getElementById("userDropdown");

        if (!userMenu.contains(event.target)) {
            dropdown.style.display = "none";
        }
    });
</script>
</body>
</html>

```

Tabla de Aportes

Fase 4: BackEnd

Parte/Actividad	Fecha de Realización	Miembro del Equipo Responsable
Configuración del archivo db.php y prueba de conexión PDO a MySQL.	01/12/2025	Diego Corrales
Implementación del archivo SongModel.php con CRUD de canciones.	01/12/2025	Daniel Pérez
Implementación del archivo PlaylistModel.php con CRUD y conteo.	02/12/2025	Juan Copri
Implementación del archivo UserModel.php para gestión de usuarios.	02/12/2025	Josué Pino
Integración de header.php y footer.php en todas las vistas.	03/12/2025	Diego Corrales
Lógica dinámica en index.php e indexView.php.	03/12/2025	Daniel Pérez
Listado dinámico de canciones (canciones.php y cancionesView.php).	04/12/2025	Juan Copri
Creación de playlists con validaciones.	04/12/2025	Josué Pino
Funcionalidad para agregar canciones a playlists.	05/12/2025	Diego Corrales
Edición de canciones con precarga desde BD.	05/12/2025	Daniel Pérez
Edición de playlists con actualización de datos.	06/12/2025	Juan Copri
Página de perfil con estadísticas y actualización.	06/12/2025	Josué Pino