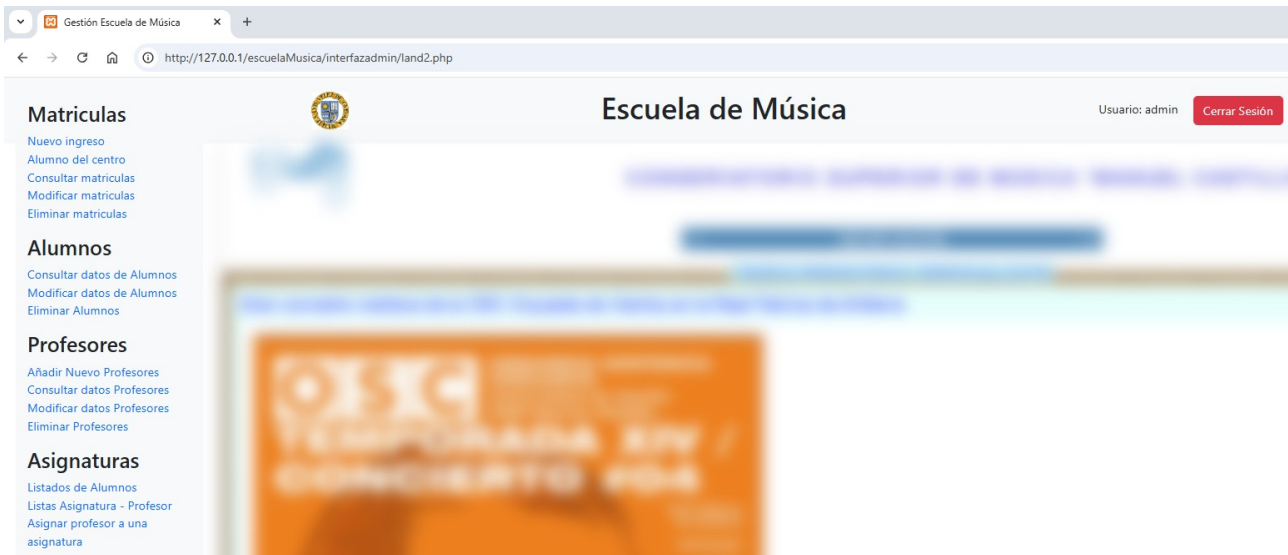


# Proyecto: Gestor de Escuela de Música (PHP)



Guillermo De La Torre Aguilera  
Horas de Libre Configuración: PHP  
2º ASIR- IES LUIS VÉLEZ DE GUEVARA  
Curso 2024-2025

# ÍNDICE

1. Introducción.....	3
1.1 Tecnologías Utilizadas.....	3
2. Estructura del Proyecto.....	4
2.1 Diagrama de Arquitectura.....	4
3. Usuarios y Roles.....	5
3.1 Administrativos.....	5
3.2 Profesores.....	5
4. Base de Datos.....	6
5. Estructura de archivos.....	7
6. Flujo de Autenticación, Sesiones y Cookies.....	8
7. Registro de usuarios de Profesores.....	8
8. Implementación.....	9

# 1. Introducción

La gestión de una escuela de música requiere un control organizado y eficiente de diversos elementos, como los alumnos matriculados, los profesores encargados de impartir las asignaturas, las especialidades musicales y la evaluación del rendimiento académico de los estudiantes.

Tradicionalmente, este tipo de administración se ha llevado a cabo mediante registros en papel o sistemas poco estructurados, lo que puede derivar en errores, pérdida de información y dificultad para acceder a los datos en tiempo real.

Este proyecto tiene como objetivo desarrollar una interfaz gráfica accesible desde un navegador web que facilite la gestión de los datos de una escuela de música. Se pretende centralizar la información en una única plataforma, optimizando la administración de los estudiantes, docentes y asignaturas.

El sistema está diseñado siguiendo la estructura de los estudios de un conservatorio elemental de música, permitiendo registrar matrículas, asignar asignaturas a los profesores y gestionar el progreso académico de los alumnos a través de la calificación de sus desempeños en distintos trimestres.

Para garantizar un uso adecuado y seguro, el sistema proporciona acceso diferenciado a dos tipos de usuarios: administrativos y profesores. Cada uno de ellos contará con permisos específicos dentro de la plataforma, asegurando que cada perfil tenga acceso únicamente a las funciones que le corresponden.

Además, este software busca mejorar la accesibilidad y la eficiencia de los procesos internos, permitiendo a los administrativos gestionar las inscripciones, asignaturas y calificaciones de forma rápida y sencilla. Los profesores, por su parte, podrán consultar los listados de alumnos y registrar sus notas de manera intuitiva, sin depender de registros manuales o documentos físicos.

Este sistema no solo moderniza la gestión académica de la escuela de música, sino que también sienta las bases para futuras mejoras, como la generación de reportes automatizados, notificaciones por correo electrónico y la implementación de herramientas de comunicación entre profesores y alumnos.

## 1.1 Tecnologías Utilizadas

- **Lenguaje Backend:** PHP
- **Base de Datos:** MySQL
- **Frontend:** HTML, CSS, JavaScript
- **Frameworks y Librerías:** Bootstrap , jQuery (para estilos)
- **Control de Versiones:** Git

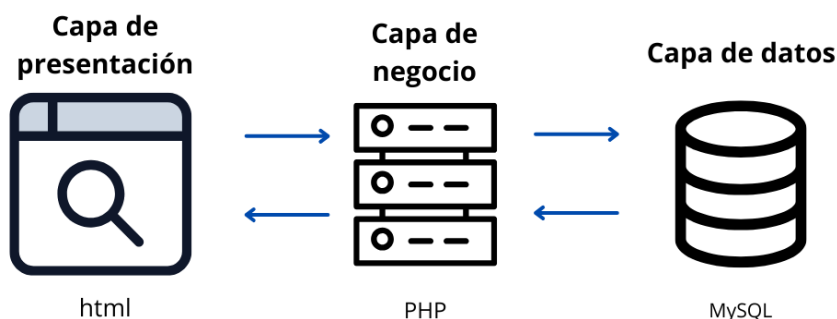
## 2. Estructura del Proyecto

El sistema está compuesto por los siguientes módulos:

- **Módulo de Autenticación:** Permite el acceso seguro al sistema mediante login y sesiones.
- **Módulo de Gestión Administrativa:** Permite la creación, modificación y eliminación de matrículas, alumnos, profesores y asignaturas.
- **Módulo de Gestión de Profesores:** Permite a los profesores ver listados de sus asignaturas e ingresar calificaciones de los alumnos.

### 2.1 Diagrama de Arquitectura

El sistema sigue un **Modelo Cliente-Servidor** con una **arquitectura en tres capas (3-tier architecture)**, organizada de la siguiente manera:



#### 1. Capa de Presentación (Frontend):

- Implementada en **HTML, CSS, JavaScript y Bootstrap**.
- Se encarga de la interfaz gráfica con la que interactúan los usuarios (administrativos y profesores).

#### 2. Capa de Lógica de Negocio (Backend):

- Desarrollada en **PHP**, donde se gestionan las reglas de negocio (la vista de los distintos usuarios y la interacción con la base de datos ).
- Controla la autenticación de usuarios, validaciones y la interacción con la base de datos.
- Contiene los scripts para gestionar la información de alumnos, profesores, matrículas y notas.

#### 3. Capa de Datos (Base de Datos):

- Basada en **MySQL** como sistema de gestión de bases de datos.
- Contiene las tablas estructuradas con relaciones entre alumnos, profesores, asignaturas, matrículas y notas.

## 3. Usuarios y Roles

### 3.1 Administrativos

- Gestionar alumnos (crear, modificar, eliminar).
- Gestionar profesores (crear, modificar, eliminar).
- Gestionar asignaturas.
- Gestionar matrículas.
- Consultar información de alumnos y profesores.

### 3.2 Profesores

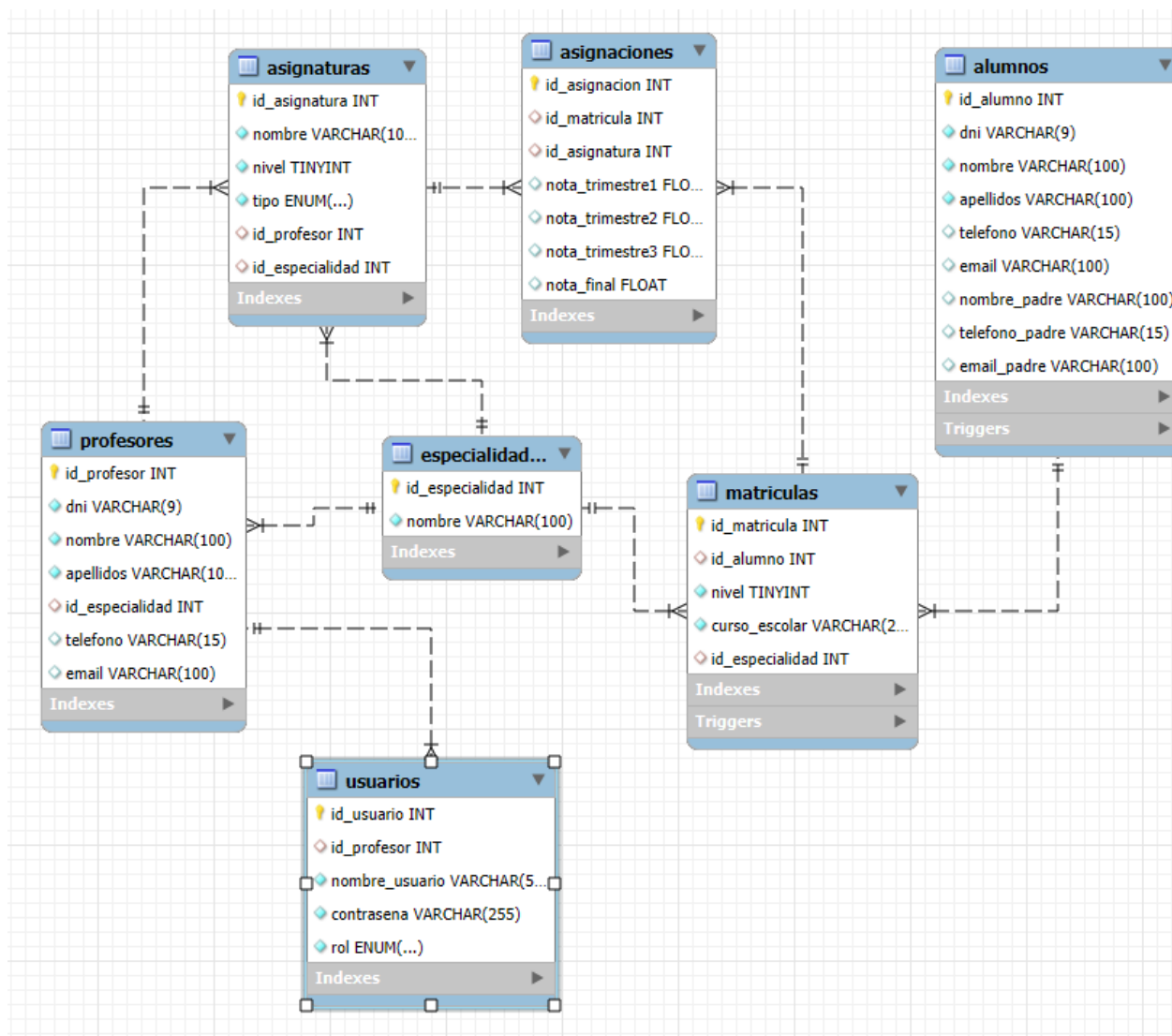
- Visualizar las asignaturas asignadas.
- Consultar lista de alumnos.
- Ingresar notas de cada trimestre.

Nombre	Apellidos	Nivel	Curso Escolar	Especialidad	Profesor	Nota Trimestre 1	Nota Trimestre 2	Nota Trimestre 3
Juan	Pérez	1	2023-2024	Piano	Carlos	5	5	5
Ana	Gómez	1	2023-2024	Piano	Carlos			
Valentina	Fernández	2	2023-2024	Piano	Carlos			
Nicolás	Jiménez	2	2023-2024	Piano	Carlos			
Fernando	González	3	2023-2024	Piano	Carlos			
Lucía	Hernández	3	2023-2024	Piano	Carlos			
Valeria	Fernández	4	2023-2024	Piano	Carlos			
Nicolás	Jiménez	4	2023-2024	Piano	Carlos			

## 4. Base de Datos

La base de datos se compone de las siguientes tablas principales:

- **Especialidades**(id\_especialidad, nombre)
- **Alumnos**(id\_alumno, dni, nombre, apellidos, telefono, email, nombre\_padre, telefono\_padre, email\_padre)
- **Profesores**(id\_profesor, dni, nombre, apellidos, id\_especialidad, telefono, email)
- **Asignaturas**(id\_asignatura, nombre, nivel, tipo, id\_profesor, id\_especialidad)
- **Matriculas**(id\_matricula, id\_alumno, nivel, curso\_escolar, id\_especialidad)
- **Asignaciones**(id\_asignacion, id\_matricula, id\_asignatura, nota\_trimestre1, nota\_trimestre2, nota\_trimestre3, nota\_final)
- **Usuarios**(id\_usuario, id\_profesor, nombre\_usuario, contrasena, rol)



## 5. Estructura de archivos

El repositorio consta de las siguientes carpetas:

**BBDD:** script de bases de datos y datos de prueba.

**Includes:** archivo de conexión y cerrar sesión.

**login:** archivo de formulario de login y archivo para el procesamiento del login (comprobación de usuario y redireccionamiento a la interfaz correspondiente).

**css y js:** :archivos de estilos de Bootstrap

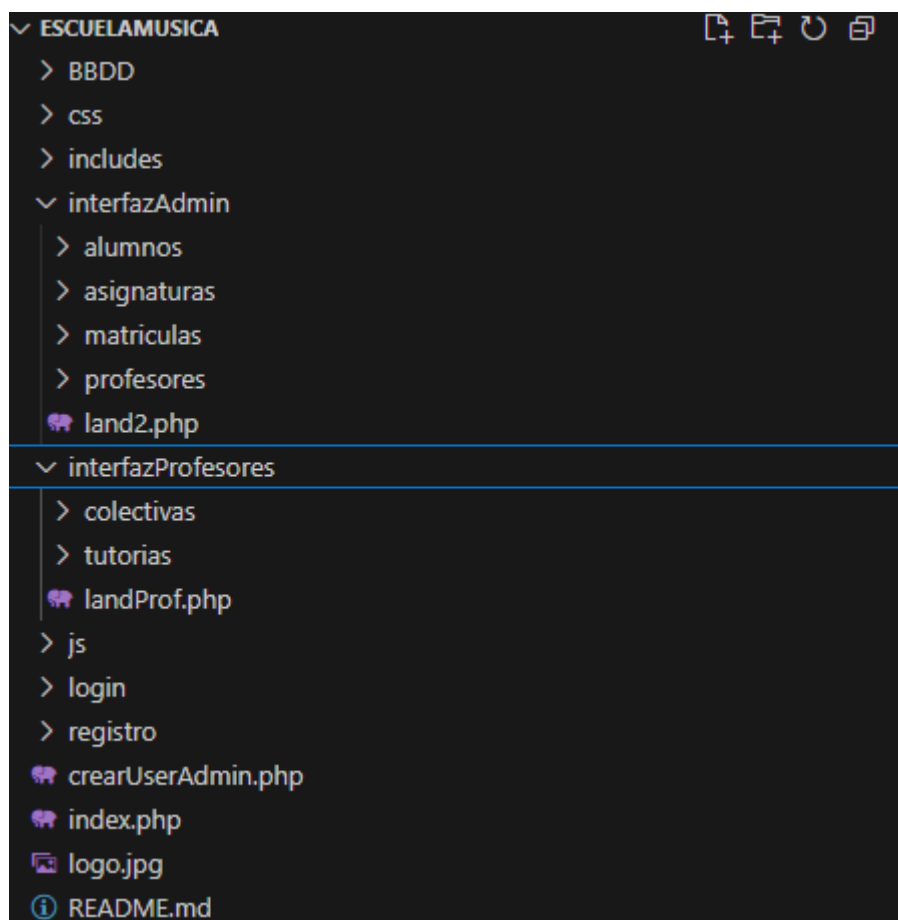
**interfazAdmin:** los archivos de la interfaz de administrativos (una carpeta por cada enlace del navegador lateral)

**interfazProfesores:** archivos de la interfaz de profesores.

**registro:** contiene los archivos para registrar un usuario profesor.

Archivo **index.php** redirige al login.

Archivo **crearUserAdmin.php**: crea el usuario admin para la interfaz de administrativo.



## 6. Flujo de Autenticación, Sesiones y Cookies

### Login y Autenticación

1. El usuario accede a la página de login.
2. Introduce sus credenciales (nombre de usuario y contraseña).
3. El sistema verifica las credenciales en la base de datos .
4. Si son correctas, se inicia la sesión y se redirige a la interfaz correspondiente (administrativa o de profesor).
5. Si las credenciales son incorrectas, se muestra un mensaje de error y vuelve al login.

### Sesiones

Se utilizan sesiones para guardar información del login de cada usuario profesor, y mostrar información personalizada (solo se muestran listados de sus asignaturas).

### Cookies

El uso de las cookies solo se ha implementado para guardar la información de la fecha y hora del último inicio de sesión.

En siguientes versiones se podrían utilizar para otras funcionalidades como crear un menú secundario en la derecha para acciones más realizadas en la interfaz de usuario administrativo.

## 7. Registro de usuarios de Profesores

Para el registro de los profesores, se ha realizado un formulario de registro para ingresar su **DNI**, nombre de usuario, y contraseña. Por lo tanto, para que un profesor pueda registrarse, debe existir como profesor en la base de datos. El usuario administrativo debe ingresar los datos del profesor previamente.

### Registro de Profesor

DNI del Profesor

Nombre de Usuario

Contraseña

Registrar

Escuela de Música

Profesor: María López  
Último inicio de sesión: Primera vez o sin registros recientes.

Cerrar Sesión

Asignaturas

- Tutorías
- Instrumento Guitarra
- Colectivas
- Música colectiva de Guitarra

Asignatura: Instrumento Guitarra

Nombre	Apellidos	Nivel	Curso Escolar	Especialidad	Profesor	Nota Trimestre 1	Nota Trimestre 2	Nota Trimestre 3
Luis	Martínez	1	2023-2024	Guitarra	María			
María	López	1	2023-2024	Guitarra	María			
Gabriel	Morales	2	2023-2024	Guitarra	María			
Isabella	Cruz	2	2023-2024	Guitarra	María			
Javier	Torres	3	2023-2024	Guitarra	María			
Clara	Ramírez	3	2023-2024	Guitarra	María			
Gabriel	Morales	4	2023-2024	Guitarra	María			
Isabella	Cruz	4	2023-2024	Guitarra	María			

Actualizar Notas



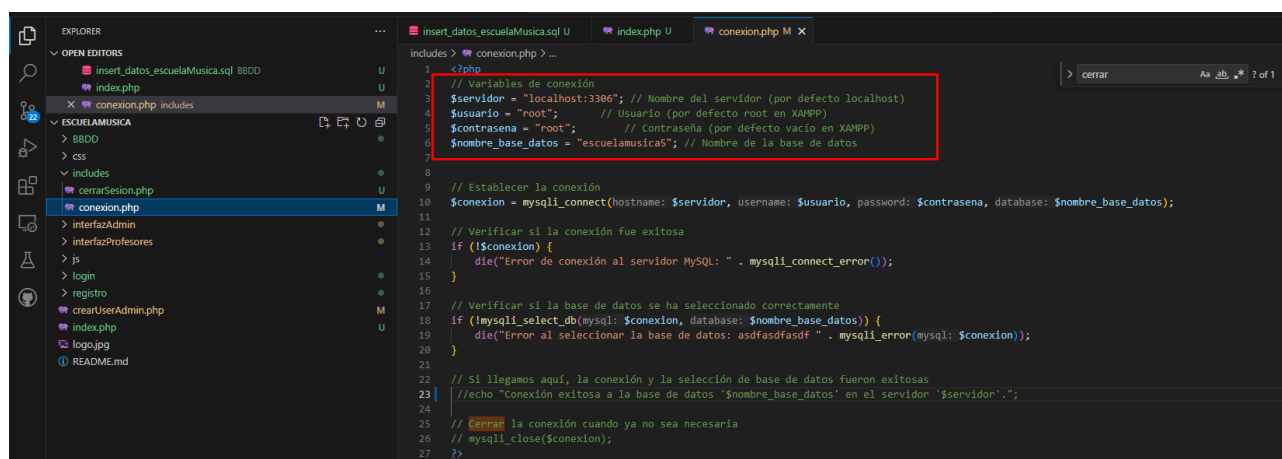
## 8. Implementación

1º Copiar el repositorio entero menos la carpeta BBDD, en la carpeta *C:/xampp/htdocs/escuelaMusica* para acceder posteriormente por el navegador poniendo en la URL *localhost/escuelaMusica* o *127.0.0.1/escuelaMusica*

2º La carpeta BBDD contiene el script para la creación de la base de datos (*escuelaMusica5.sql*) y unos datos de prueba (*insert\_datos\_escuelaMusica.sql*).

-Ejecutar el script de la carpeta BBDD para la creación de la base de datos y el script de datos de prueba en el servidor de bases de datos a utilizar (Workbench o PHPmyAdmin)

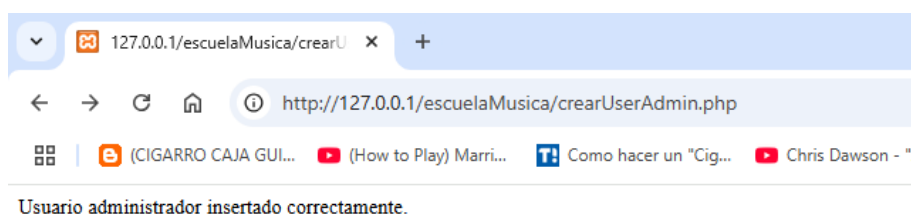
- Configurar el archivo de conexión con los datos correspondientes (usuario, puerto, nombre de base de datos...) para la conexión de bases de datos (*includes/conexion.php*)



```

1 <?php
2 // Variables de conexión
3 $servidor = "localhost:3306"; // Nombre del servidor (por defecto localhost)
4 $usuario = "root"; // Usuario (por defecto root en XAMPP)
5 $contrasena = "root"; // Contraseña (por defecto vacío en XAMPP)
6 $nombre_base_datos = "escuelamusica5"; // Nombre de la base de datos
7
8
9 // Establecer la conexión
10 $conexion = mysqli_connect(hostname: $servidor, username: $usuario, password: $contrasena, database: $nombre_base_datos);
11
12 // Verificar si la conexión fue exitosa
13 if (!$conexion) {
14     die("Error de conexión al servidor MySQL: " . mysqli_connect_error());
15 }
16
17 // Verificar si la base de datos se ha seleccionado correctamente
18 if (!mysqli_select_db(mysqli: $conexion, database: $nombre_base_datos)) {
19     die("Error al seleccionar la base de datos: asdfasdf " . mysqli_error(mysqli: $conexion));
20 }
21
22 // Si llegamos aquí, la conexión y la selección de base de datos fueron exitosas
23 //echo "Conexión exitosa a la base de datos '$nombre_base_datos' en el servidor '$servidor'.";
24
25 // Cerrar la conexión cuando ya no sea necesaria
26 mysqli_close($conexion);
27
  
```

3º Acceder al archivo *127.0.0.1/escuelaMusica/crearUsuarioAdmin.php* para crear un usuario administrativo.



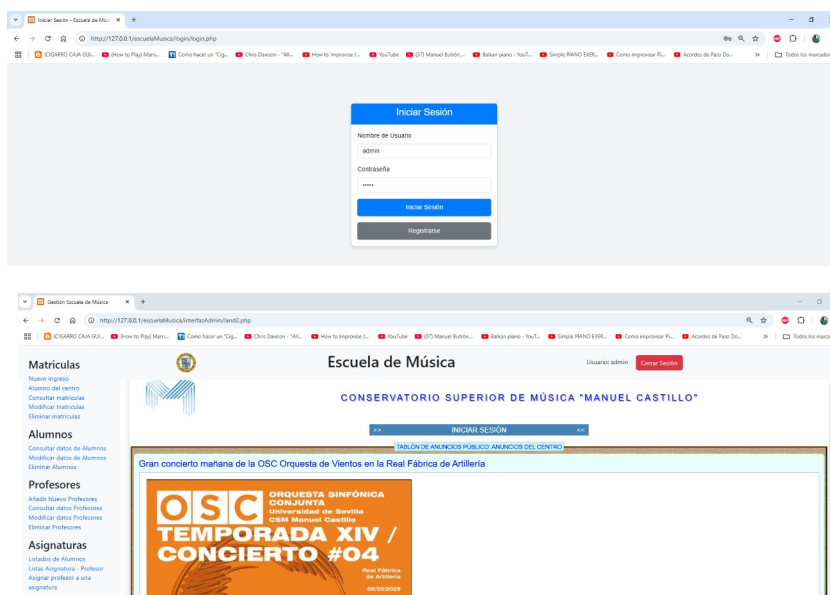
**Nota:** Se puede cambiar el usuario administrativo por defecto o crear uno nuevo diferente con ese mismo archivo (*escuelaMusica/crearUsuarioAdmin.php*), configurando las variables de *\$nombre\_usuario* y *\$contrasena\_admin*

```

1 <?php
2 // ESTE ARCHIVO SÓLO CREA UN USUARIO ADMINISTRATIVO PARA LA INTERFAZ D
3 // Configuración de la conexión a la base de datos
4 include("includes/conexion.php");
5
6 // Verificar si la conexión fue exitosa
7 if ($conexion) {
8     die("Error de conexión: " . mysqli_connect_error());
9 }
10
11 // Datos del usuario administrador
12 $nombre_usuario = "admin"; // Nombre de usuario
13 $contrasena_admin = "admin"; // Contraseña en texto claro
14 $rol = "admin"; // Rol del usuario: admin
15
16 // Crear el hash de la contraseña
17 $hashed_contrasena = password_hash($contrasena_admin, algo: PASSWORD_DEFAULT);
18
19 // Consulta para insertar el usuario administrador
20 $query = "INSERT INTO usuarios (nombre_usuario, contrasena, rol) VALUES ('$nombre_usuario', '$hashed_contrasena', '$rol')";
21
22 // Ejecutar la consulta de inserción
23 if (mysqli_query($conexion, $query)) {
24     echo "Usuario administrador insertado correctamente.";
25 } else {
26     echo "Error al insertar el usuario: " . mysqli_error($conexion);
27 }
28
29 // Cerrar la conexión a la base de datos
30 mysqli_close($conexion);
31
32

```

4º Acceder desde la URL con el usuario administrativo (por defecto *admin:admin*)



Desde la interfaz de administrativo ya podemos realizar todas las acciones de matriculas. Alumnos, asignaturas... así como añadir un nuevo profesor para que realice el registro de usuario.

Una vez se registre un profesor con usuario y contraseña, podrá ingresar a su interfaz del profesor.