



Universidad Tecnológica de México

División de Ingenierías

Materia: Gestión de base de datos

Prof.: Aguilar Cruz Miguel

Actividad:

Proyecto final gestión de base de datos

Sistema de gestión para una escuela de música

Alumno: Valencia Muñoz Fabian Alejandro

Fecha: 08/12/2025

Contenido

| | |
|--|----|
| 1.- Objetivo del proyecto..... | 4 |
| 2.- Características del Sistema | 4 |
| 3.- Alcance..... | 5 |
| 4.1. Modelo lógico (tablas y relaciones) | 6 |
| 4.1.1. Tabla Alumnos | 6 |
| 4.1.2. Tabla Cursos | 6 |
| 4.1.3. Tabla Instructores..... | 7 |
| 4.1.4. Tabla Usuarios | 7 |
| 4.1.5. Tabla Cupos..... | 8 |
| 4.1.6. Tabla Inscripciones..... | 8 |
| 4.1.7. Tablas de Auditoría | 9 |
| 4.1.7.1. Tabla Auditoria_Inscripciones | 9 |
| 4.1.7.2. Tabla Auditoria_Cursos | 9 |
| 4.1.7.3. Tabla Auditoria_Instructores | 10 |
| 4.1.7.4. Tabla Auditoria (General) | 10 |
| 4.1.7.4. Tabla Auditoria_Inscripciones_Actualizacion..... | 10 |
| 4.1.7.5. Tabla Auditoria_Inscripciones_Eliminacion | 11 |
| 4.1.6.2. Auditoría de Cursos | 12 |
| 4.1.6.3. Relación con los triggers | 14 |
| 5.- Procedimientos almacenados | 15 |
| 5.1. sp_RegistrarInscripcion | 15 |
| 5.2. sp_RegistrarPago | 15 |
| 5.3. sp_HistorialInscripcionesUsuario | 16 |
| 5.4. sp_ReportesInscripciones | 16 |
| 6. Triggers | 17 |
| 6.1. Trigger de validación de disponibilidad de curso..... | 17 |
| 6.2. Triggers de auditoría | 17 |
| 6.2.1. Triggers de auditoría para Inscripciones | 17 |

| | |
|--|----|
| 6.2.2. Triggers de auditoría para Cursos | 18 |
| 7. Manejo de errores (TRY/CATCH) | 19 |
| 8. Seguridad y roles..... | 19 |

1.- Objetivo del proyecto

Desarrollar un sistema de base de datos integral que permita gestionar usuarios, alumnos, cursos, inscripciones, pagos en una escuela de música. El sistema busca optimizar el proceso de inscripción a los cursos musicales desde el registro del usuario hasta la confirmación del pago, garantizando integridad, trazabilidad y eficiencia en las operaciones.

El sistema tiene la función de automatizar el proceso de inscripción y administración de los cursos de la escuela de música desde el registro del alumno hasta su inscripción y pago

2.- Características del Sistema

a) Mantener un control preciso de:

- Usuarios y alumnos registrados en la plataforma.
- Cursos disponibles, activos e inactivos.
- Instructores y sus especialidades.
- Inscripciones de alumnos a los cursos.
- Pagos realizados por los alumnos.
- Cupos disponibles por cada curso.
- Actividad de los usuarios (auditoría).

b) Implementar buenas prácticas de:

- Integridad referencial mediante llaves primarias y foráneas.
- Transacciones para garantizar la consistencia de los registros.
- Manejo de errores mediante bloques TRY/CATCH.
- Auditoría de cambios a través de tablas de seguimiento.
- Índices y campos identidad para asegurar la trazabilidad de la información.

3.- Alcance

El sistema cubre los siguientes procesos:

- Registro y actualización de usuarios y alumnos.
- Registro y administración de cursos musicales.
- Registro de instructores y sus especialidades.
- Registro de inscripciones de alumnos a los cursos.
- Cancelación de inscripciones con actualización del cupo disponible.
- Registro de pagos realizados por los alumnos.

- Consulta de:
 - o Historial de inscripciones por alumno.
 - o Inscripciones por periodo.
 - o Cursos disponibles y cupos restantes.
 - o Pagos realizados por alumno.

- Generación de reportes de:
 - o Cursos más solicitados.
 - o Instructores con mayor número de alumnos.
 - o Total de inscripciones por periodo.
 - o Clasificación de alumnos por nivel de inversión.

4.1. Modelo lógico (tablas y relaciones)

A continuación, se describen las tablas principales del sistema de la escuela de música y sus relaciones.

4.1.1. Tabla Alumnos

Campos principales:

- alumno_id (PK, INT, generado por IDENTITY)
- nombre_completo (VARCHAR(100), NULL)
- email (VARCHAR(100), NULL)
- telefono (VARCHAR(100), NULL)
- direccion (VARCHAR(100), NULL)
- tipo_documento (VARCHAR(20), NULL)
- numero_documento (VARCHAR(20), NULL)
- fecha_registro (DATE, NULL, valor por defecto = GETDATE())

Restricciones y reglas:

- Clave primaria en alumno_id.
- Índice en email para optimizar búsquedas.

Relaciones:

- Relación 1 a N con Inscripciones (un alumno puede tener varias inscripciones).
- Relación 1 a 1 opcional con Usuarios (un alumno puede tener un usuario de acceso).

4.1.2. Tabla Cursos

Campos principales:

- curso_id (PK, INT, generado por IDENTITY iniciando en 100)
- nombre_curso (VARCHAR(100), NULL)
- nivel (VARCHAR(30), NULL)
- duracion_semanas (INT, NULL)
- costo (DECIMAL(10,2), NULL)
- estado_curso (VARCHAR(20), valor por defecto = 'Activo')

- fecha_inicio (DATE, NULL)
- fecha_fin (DATE, NULL)

Restricciones e índices:

- Clave primaria en curso_id.
- Índice en nombre_curso.

Relaciones:

- Relación 1 a N con Inscripciones (un curso puede tener muchos alumnos inscritos).
- Relación 1 a 1 con Cupos (cada curso tiene un registro de cupos).

4.1.3. Tabla Instructores

Campos principales:

- instructor_id (PK, INT, generado por IDENTITY iniciando en 200)
- nombre_completo (VARCHAR(100), NULL)
- especialidad (VARCHAR(50), NULL)
- usuario (VARCHAR(50), NULL)
- contraseña (VARCHAR(100), NULL)
- fecha_ingreso (DATE, NULL)
- estado (VARCHAR(20), valor por defecto = 'Activo')

Relaciones:

- Relación 1 a N con Inscripciones (un instructor puede impartir varios cursos a diferentes alumnos).

4.1.4. Tabla Usuarios

Campos principales:

- usuario_id (PK, INT, generado por IDENTITY)
- username (VARCHAR(50), NOT NULL, único)
- password_hash (VARCHAR(255), NOT NULL)
- rol (VARCHAR(20), NOT NULL)
- alumno_id (FK → Alumnos.alumno_id, NULL)

Restricciones:

- Clave primaria en usuario_id.
- Restricción UNIQUE en username.
- Llave foránea en alumno_id.

Relaciones:

- Relación N a 1 con Alumnos (varios usuarios pueden estar relacionados a un alumno o ser administradores sin alumno).

4.1.5. Tabla Cupos

Campos principales:

- cupo_id (PK, INT, generado por IDENTITY)
- curso_id (FK → Cursos.curso_id, NOT NULL, único)
- cupos_maximos (INT, NOT NULL)
- cupos_disponibles (INT, NOT NULL)

Restricciones:

- Clave primaria en cupo_id.
- Llave foránea en curso_id.
- Restricción UNIQUE en curso_id.

Relaciones:

- Relación 1 a 1 con Cursos (cada curso tiene un único registro de cupos).

4.1.6. Tabla Inscripciones

Campos principales:

- inscripcion_id (PK, INT, generado por IDENTITY)
- alumno_id (FK → Alumnos.alumno_id)
- curso_id (FK → Cursos.curso_id)
- instructor_id (FK → Instructores.instructor_id)
- fecha_inscripcion (DATE, NULL)
- metodo_pago (VARCHAR(50), NULL)
- total_pago (DECIMAL(10,2), NULL)

- estado_inscripcion (VARCHAR(50), valor por defecto = 'Activa')
- comprobante_pago (VARCHAR(255), NULL)

Índices:

- Índices implícitos por llaves foráneas para optimizar consultas.

Relaciones:

- Relación N a 1 con Alumnos.
- Relación N a 1 con Cursos.
- Relación N a 1 con Instructores.

4.1.7. Tablas de Auditoría

4.1.7.1. Tabla Auditoria_Inscripciones

Registra los cambios realizados en la tabla Inscripciones.

Campos principales:

- auditoria_ins_id (INT, PK, IDENTITY)
- inscripcion_id (INT)
- accion (VARCHAR(20))
- estado_anterior (VARCHAR(50))
- estado_nuevo (VARCHAR(50))
- usuario (VARCHAR(50))
- fecha_evento (DATETIME, por defecto GETDATE())

4.1.7.2. Tabla Auditoria_Cursos

Registra los cambios de costo en los cursos.

Campos principales:

- aud_curso_id (INT, PK, IDENTITY)
- curso_id (INT)
- accion (VARCHAR(20))
- costo_anterior (DECIMAL(10,2))
- costo_nuevo (DECIMAL(10,2))

- usuario (VARCHAR(50))
- fecha_evento (DATETIME, por defecto GETDATE())

4.1.7.3. Tabla Auditoria_Instructores

Registra cambios en la especialidad de los instructores.

Campos principales:

- aud_inst_id (INT, PK, IDENTITY)
- instructor_id (INT)
- accion (VARCHAR(20))
- especialidad_anterior (VARCHAR(50))
- especialidad_nueva (VARCHAR(50))
- usuario (VARCHAR(50))
- fecha_evento (DATETIME, por defecto GETDATE())

4.1.7.4. Tabla Auditoria (General)

Registra errores y eventos generales del sistema.

Campos principales:

- auditoria_id (INT, PK, IDENTITY)
- tabla_afectada (VARCHAR(50))
- accion (VARCHAR(50))
- usuario (VARCHAR(50))
- descripcion (VARCHAR(MAX))
- fecha_error (DATETIME, por defecto GETDATE())

4.1.7.4. Tabla Auditoria_Inscripciones_Actualizacion

Registra las modificaciones realizadas sobre una inscripción existente.

Campos principales:

- auditoria_ins_id (INT, PK, IDENTITY)
- inscripcion_id (INT)
- alumno_id_anterior (INT)
- curso_id_anterior (INT)

- instructor_id_anterior (INT)
- fecha_inscripcion_anterior (DATE)
- total_pago_anterior (DECIMAL(10,2))
- metodo_pago_anterior (VARCHAR(50))
- estado_inscripcion_anterior (VARCHAR(50))
- alumno_id_nuevo (INT)
- curso_id_nuevo (INT)
- instructor_id_nuevo (INT)
- fecha_inscripcion_nueva (DATE)
- total_pago_nuevo (DECIMAL(10,2))
- metodo_pago_nuevo (VARCHAR(50))
- estado_inscripcion_nuevo (VARCHAR(50))
- fecha_evento (DATETIME, por defecto GETDATE())

Descripción:

Cada fila guarda en un mismo registro los valores antes y después de una actualización (UPDATE) realizada sobre una inscripción.

4.1.7.5. Tabla Auditoria_Inscripciones_Eliminacion

Registra las inscripciones eliminadas de forma lógica o física.

Campos principales:

- auditoria_id (INT, PK, IDENTITY)
- inscripcion_id (INT)
- alumno_id (INT)
- curso_id (INT)
- instructor_id (INT)
- fecha_inscripcion (DATE)
- total_pago (DECIMAL(10,2))
- metodo_pago (VARCHAR(50))
- estado_inscripcion (VARCHAR(50))
- fecha_eliminacion (DATETIME, por defecto GETDATE())

Descripción:

Cada registro contiene el último estado conocido de la inscripción antes de ser eliminada del sistema.

4.1.6.2. Auditoría de Cursos

a) Tabla AuditoriaCursosInsercion

Registra la inserción de nuevos cursos en el sistema.

Campos principales:

- aud_curso_id (INT, PK, IDENTITY)
- curso_id (INT)
- nombre_curso (VARCHAR(100))
- nivel (VARCHAR(30))
- duracion_semanas (INT)
- costo (DECIMAL(10,2))
- estado_curso (VARCHAR(20))
- fecha_inicio (DATE)
- fecha_fin (DATE)
- fecha_insercion (DATETIME, por defecto GETDATE())

Descripción:

Cada registro representa el estado completo del curso recién creado en el sistema.

b) Tabla AuditoriaCursosActualizacion

Registra los cambios realizados en los datos del curso (por ejemplo, cambios de costo, nivel o estado).

Campos principales:

- aud_curso_id (INT, PK, IDENTITY)
- curso_id (INT)

Valores anteriores:

- nombre_curso_anterior (VARCHAR(100))

- nivel_anterior (VARCHAR(30))
- duracion_semanas_anterior (INT)
- costo_anterior (DECIMAL(10,2))
- estado_curso_anterior (VARCHAR(20))
- fecha_inicio_anterior (DATE)
- fecha_fin_anterior (DATE)

Valores nuevos:

- nombre_curso_nuevo (VARCHAR(100))
- nivel_nuevo (VARCHAR(30))
- duracion_semanas_nuevo (INT)
- costo_nuevo (DECIMAL(10,2))
- estado_curso_nuevo (VARCHAR(20))
- fecha_inicio_nuevo (DATE)
- fecha_fin_nuevo (DATE)
- fecha_evento (DATETIME, por defecto GETDATE())

Descripción:

Cada fila guarda en un mismo registro los valores antes y después de una actualización (UPDATE) realizada sobre un curso.

c) Tabla AuditoriaCursosEliminacion

Registra los cursos que se eliminan del catálogo del sistema.

Campos principales:

- aud_curso_id (INT, PK, IDENTITY)
- curso_id (INT)
- nombre_curso (VARCHAR(100))
- nivel (VARCHAR(30))
- duracion_semanas (INT)
- costo (DECIMAL(10,2))
- estado_curso (VARCHAR(20))
- fecha_inicio (DATE)

- fecha_fin (DATE)
- fecha_eliminacion (DATETIME, por defecto GETDATE())

Descripción:

Cada registro contiene el último estado conocido del curso antes de ser eliminado del sistema.

4.1.6.3. Relación con los triggers

Cada una de estas tablas de auditoría se llena mediante triggers definidos en la base de datos, de la siguiente manera:

- Triggers sobre la tabla Cursos:
 - trg_Cursos_Insercion → inserta registros en Auditoria_Cursos cuando se crea un curso.
 - trg_Cursos_Actualizacion → guarda los valores anteriores y nuevos en Auditoria_Cursos cuando se actualiza un curso.
 - trg_Cursos_Eliminacion → registra la eliminación de un curso en Auditoria_Cursos.
- Triggers sobre la tabla Inscripciones:
 - trg_Inscripciones → registra cambios de estado en la tabla Auditoria_Inscripciones.
- Triggers sobre la tabla Instructores:
 - trg_Auditoria_Instructores → registra modificaciones en la especialidad de los instructores.

Los triggers utilizan las tablas lógicas inserted y deleted para obtener los valores anteriores y nuevos, almacenándolos en las tablas de auditoría correspondientes, garantizando la trazabilidad completa de los movimientos del sistema.

5.- Procedimientos almacenados

A continuación, se documentan los procedimientos principales del sistema.

5.1. sp_RegistrarInscripcion

Función:

- Registra una nueva inscripción de un alumno a un curso, generando su respectivo registro en el sistema.

Lógica general:

1. Validar que el usuario exista y esté activo.
2. Validar que el curso exista y esté activo.
3. Verificar que el curso tenga cupo disponible.
4. Iniciar transacción (BEGIN TRAN).
5. Insertar el registro en la tabla Inscripciones.
6. Actualizar el estado de la inscripción.
7. Registrar la acción en la tabla Auditoria_Inscripciones.
8. Hacer COMMIT en caso de éxito o ROLLBACK en caso de error (usando TRY/CATCH).

5.2. sp_RegistrarPago

Función:

- Registrar el pago de una inscripción realizada por un usuario.

Lógica general:

1. Validar que la inscripción exista.
2. Validar que la inscripción no esté cancelada.
3. Iniciar transacción.

4. Insertar el registro en la tabla Pagos.
5. Actualizar el estado de la inscripción a Pagado.
6. Registrar la acción en la tabla Auditoria_Inscripciones.
7. Hacer COMMIT o ROLLBACK según el resultado.

5.3. sp_HistorialInscripcionesUsuario

Función:

- Devolver todas las inscripciones realizadas por un usuario, incluyendo el detalle del curso.

Salidas:

- Información de inscripción (fecha, estado).
- Información del curso (nombre, nivel, duración, costo).

Uso:

- Consultas del usuario.
- Seguimiento académico.
- Reportes administrativos.

5.4. sp_ReportesInscripciones

Función:

- Generar reportes de inscripciones según distintos criterios.

Ejemplos de reportes:

- Inscripciones por periodo (rango de fechas).
- Cursos con mayor número de inscritos.
- Pagos realizados por periodo.
- Total de inscripciones por curso.
- Ranking de cursos más solicitados.

6. Triggers

6.1. Trigger de validación de disponibilidad de curso

Tabla objetivo:

- Inscripciones

Función:

- Validar que el curso tenga cupo disponible antes de confirmarse el registro en la tabla Inscripciones.
- En caso de no cumplir la condición, impedir la operación.

6.2. Triggers de auditoría

6.2.1. Triggers de auditoría para Inscripciones

• **Trigger:** trg_Inscripciones_Insercion

• **Tabla objetivo:** Inscripciones

• **Momento de ejecución:** AFTER INSERT

Función:

1. Cada vez que se inserta una nueva inscripción en la tabla Inscripciones, el trigger toma los valores de la tabla lógica inserted.
2. Inserta un registro en la tabla **Auditoria_Inscripciones** con los campos: IncripcionId, accion, estado_nuevo, usuario y la fecha de auditoría (fecha_evento).

• **Trigger:** trg_Inscripciones_Actualizacion

• **Tabla objetivo:** Inscripciones

• **Momento de ejecución:** AFTER UPDATE

Función:

1. Usa las tablas lógicas deleted (valores antes de la actualización) e inserted (valores después de la actualización).

2. Inserta un registro en **Auditoria_Inscripciones** con:

- IncripcionId
- EstadoAnterior
- EstadoNuevo
- Usuario
- Fecha del evento

- **Trigger:** trg_Inscripciones_Eliminacion
- **Tabla objetivo:** Inscripciones
- **Momento de ejecución:** AFTER DELETE

Función:

1. Toma los datos de la tabla lógica deleted, que contiene las inscripciones eliminadas.
2. Inserta un registro en **Auditoria_Inscripciones** con:
InscripcionId, accion, estado_anterior, usuario y fecha_evento.
3. Deja registro del último estado de la inscripción antes de ser eliminada.

6.2.2. Triggers de auditoría para Cursos

- **Trigger:** trg_Cursos_Insercion
- **Tabla objetivo:** Cursos
- **Momento de ejecución:** AFTER INSERT

Función:

1. Cada vez que se inserta un curso en la tabla Cursos, el trigger toma los valores de la tabla lógica inserted.
2. Inserta un registro en **Auditoria_Cursos** con:
curso_id, accion, costo_nuevo, usuario y fecha_evento.

- **Trigger:** trg_Cursos_Actualizacion
- **Tabla objetivo:** Cursos
- **Momento de ejecución:** AFTER UPDATE

Función:

1. Usa las tablas lógicas deleted (valores anteriores) e inserted (valores nuevos).
 2. Inserta un registro en **Auditoria_Cursos** con:
 - **curso_id**.
 - **Valores anteriores:** costo_anterior, especialidad_anterior (si aplica), estado_anterior.
 - **Valores nuevos:** costo_nuevo, especialidad_nueva (si aplica), estado_nuevo.
 - **fecha_evento** con la fecha y hora del cambio.
- Es útil para rastrear, por ejemplo, **cambios en el costo del curso o en su estado**.

7. Manejo de errores (TRY/CATCH)

En los procedimientos críticos del sistema (**sp_RegistrarInscripcion**, **sp_RegistrarPago**).

- Se implementan bloques **TRY/CATCH** para:
 - o Controlar errores de **integridad referencial**.
 - o Controlar errores de **negocio** (curso sin cupo, inscripción inexistente, inscripción ya pagada, etc.).

Dentro del bloque CATCH:

- Registrar el error en la tabla **Auditoria** (tabla_afectada, accion, usuario, descripcion, fecha_error).
- Ejecutar **ROLLBACK TRAN** para garantizar la atomicidad de la transacción.

8. Seguridad y roles

1. Rol Administrador

- Permisos completos sobre:
 - o Tablas de catálogo (**Usuarios, Cursos, Instructores**).

- o Tablas de transacciones (**Inscripciones, Pagos**).
- o Tablas de auditoría.

- Puede:
 - o Registrar, modificar y eliminar cursos.
 - o Registrar instructores.
 - o Gestionar inscripciones de alumnos.
 - o Consultar pagos e historial del sistema.

2. Rol Usuario

- Permisos sobre:
 - o Consulta de cursos disponibles.
 - o Registro de inscripciones.
 - o Registro de pagos.
 - o Consulta de su historial de inscripciones.
- No tiene permisos para:
 - o Modificar directamente las auditorías.
 - o Administrar cursos o instructores.
 - o Acceder a información de otros usuarios.

Login.



Escuela de Música

Bienvenido al Portal Académico

Ingresar

¿Eres alumno nuevo?

[Crear Cuenta](#)

Contacto y Ubicación



Granjas de México 123, Narvarte Poniente, Benito Juárez,
03310

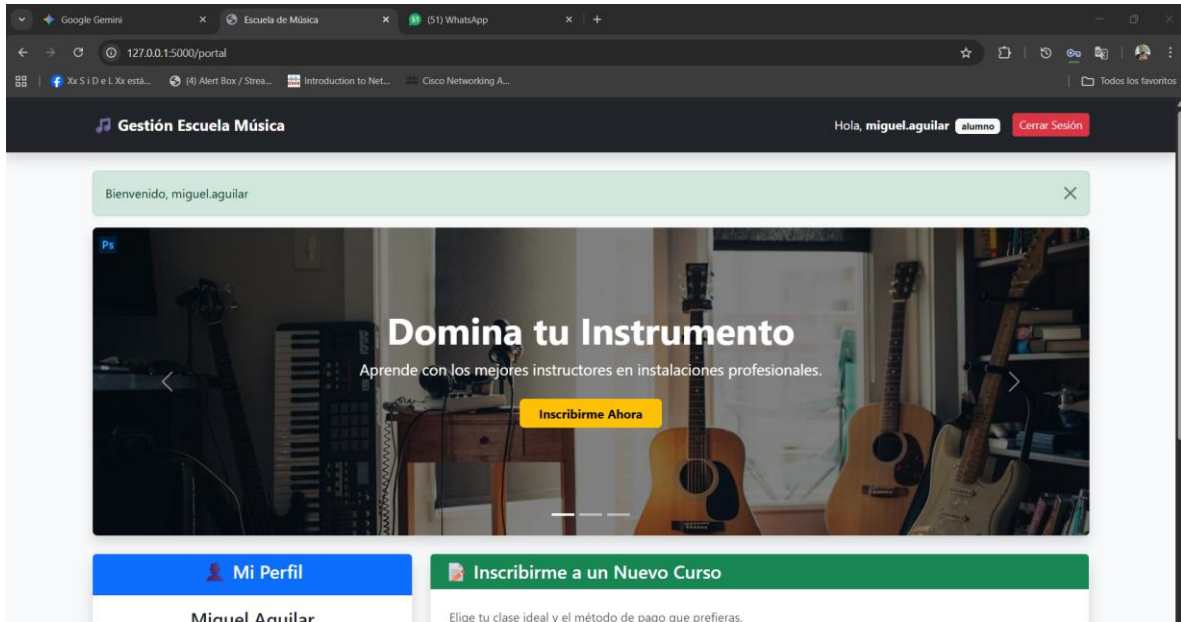


fabiansidel3@musica.com

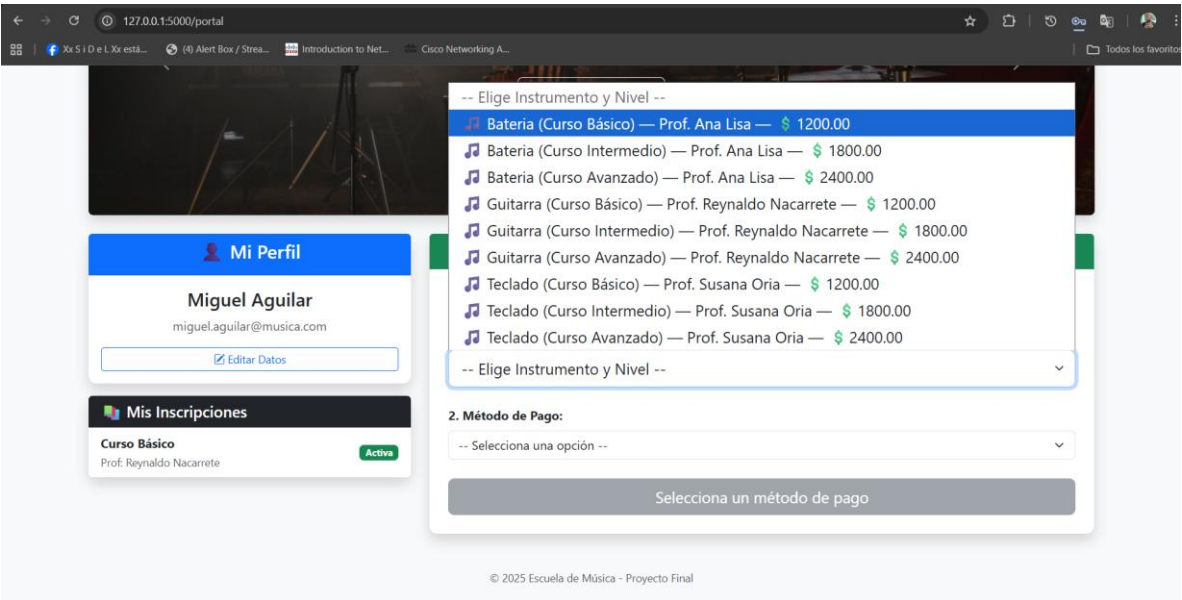


55 1009 0908

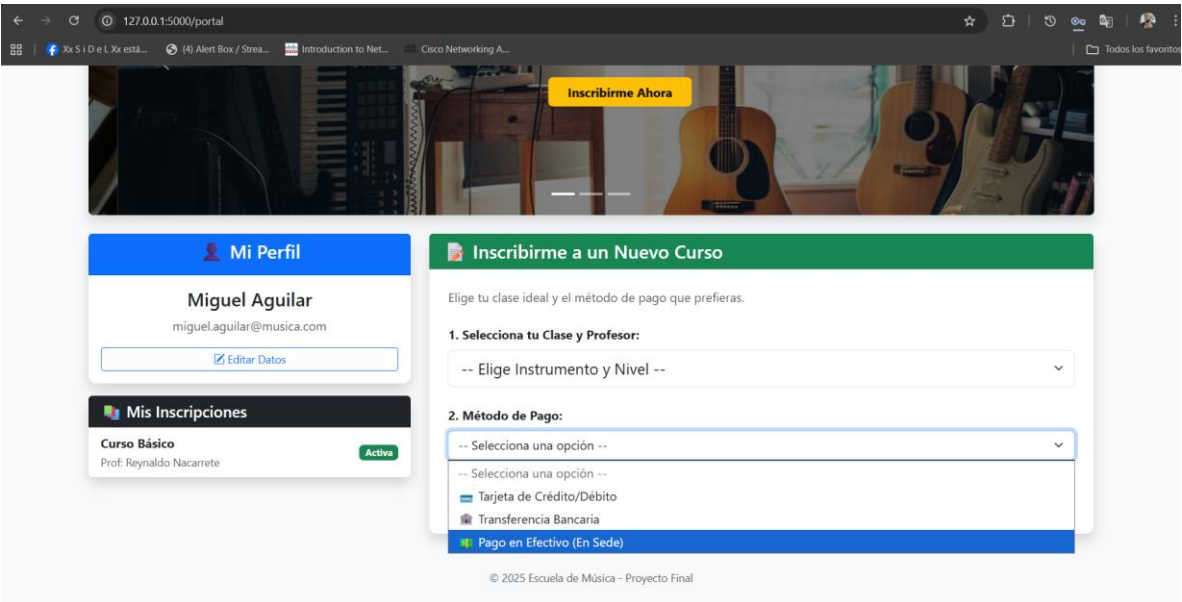
Interfaz usuario



Cursos por inscribir



Métodos de pago



Comprobante de pago

Mi Perfil

Miguel Aguilar

miguel.aguilar@musica.com

Editar Datos

Mis Inscripciones

Curso Básico

Prof: Reynaldo Nacarrete

Activo

Curso Intermedio

Prof: Reynaldo Nacarrete

Pendiente

Inscribirme a un Nuevo Curso

Elige tu clase ideal y el método de pago que prefieras.

1. Selecciona tu Clase y Profesor:

Guitarra (Curso Intermedio) — Prof. Reynaldo Nacarrete — \$ 1800.00

2. Método de Pago:

Transferencia Bancaria

Subir Comprobante

Cuenta: 1234-5678-90 | Banco: BBVA

Sube tu Comprobante (Foto o PDF):

Seleccionar archivo

Sin archivos seleccionados

Confirmar Pago e Inscribirse

© 2025 Escuela de Música - Proyecto Final

Interfaz de Administrador

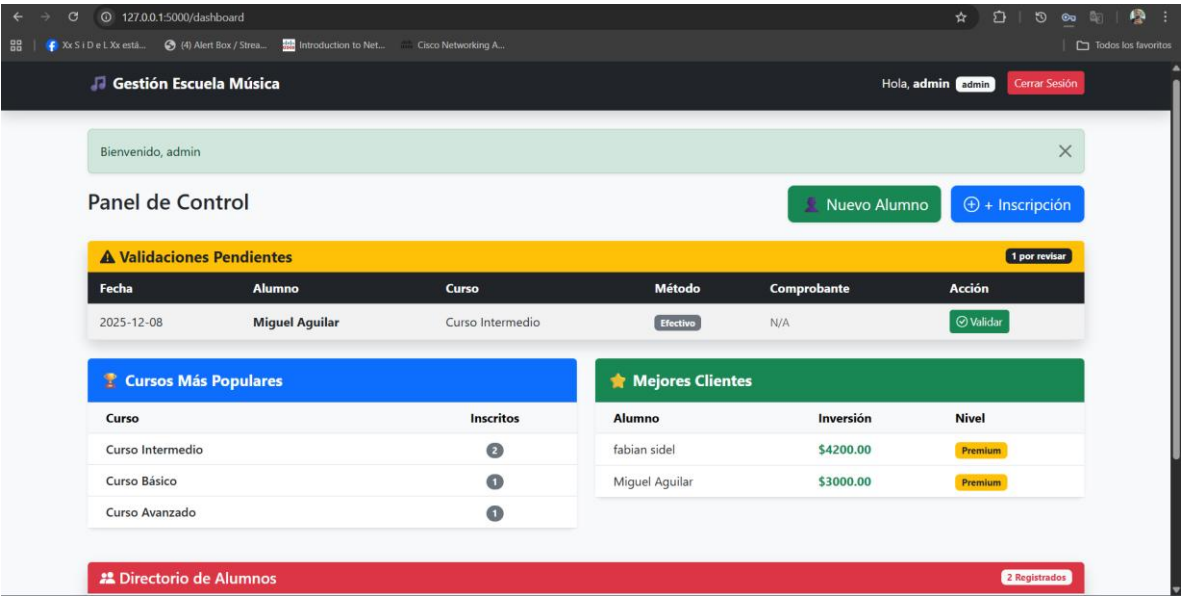
Validaciones pendientes

Cursos más populares

Mejores clientes

Añadir nuevo alumno

Inscripciones de alumno



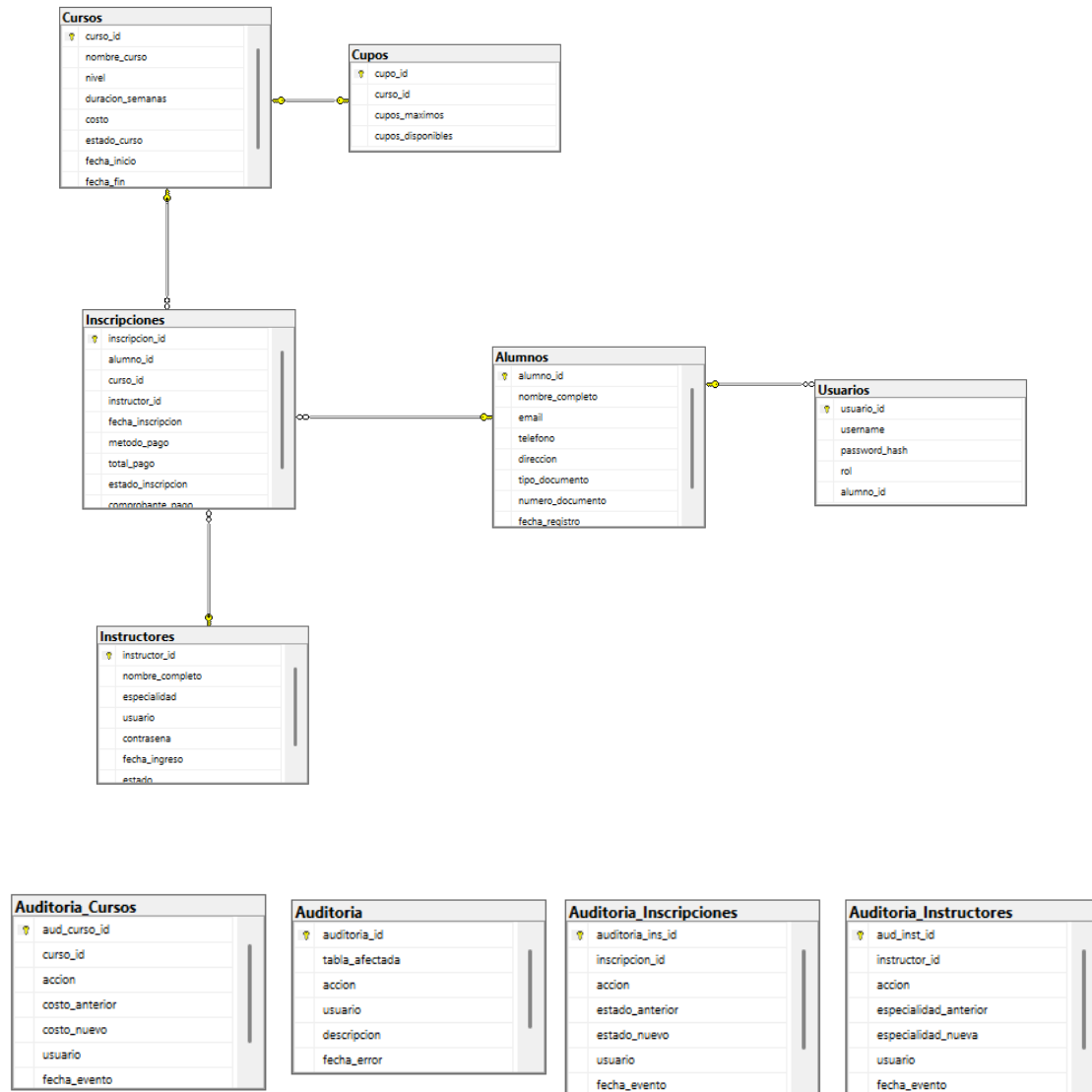
Directorio Alumnos para gestión

| Directorio de Alumnos | | | | | 2 Registrados |
|-----------------------|----------------|---------------------------|--------------|----------|---------------|
| ID | Nombre | Email | Teléfono | Acciones | |
| 6 | fabian sidel | fabian.sidel@musica.com | 101200120 | Eliminar | |
| 7 | Miguel Aguilar | miguel.aguilar@musica.com | 551001293913 | Eliminar | |

© 2025 Escuela de Música - Proyecto Final

Diagramas de flujo

De Clases



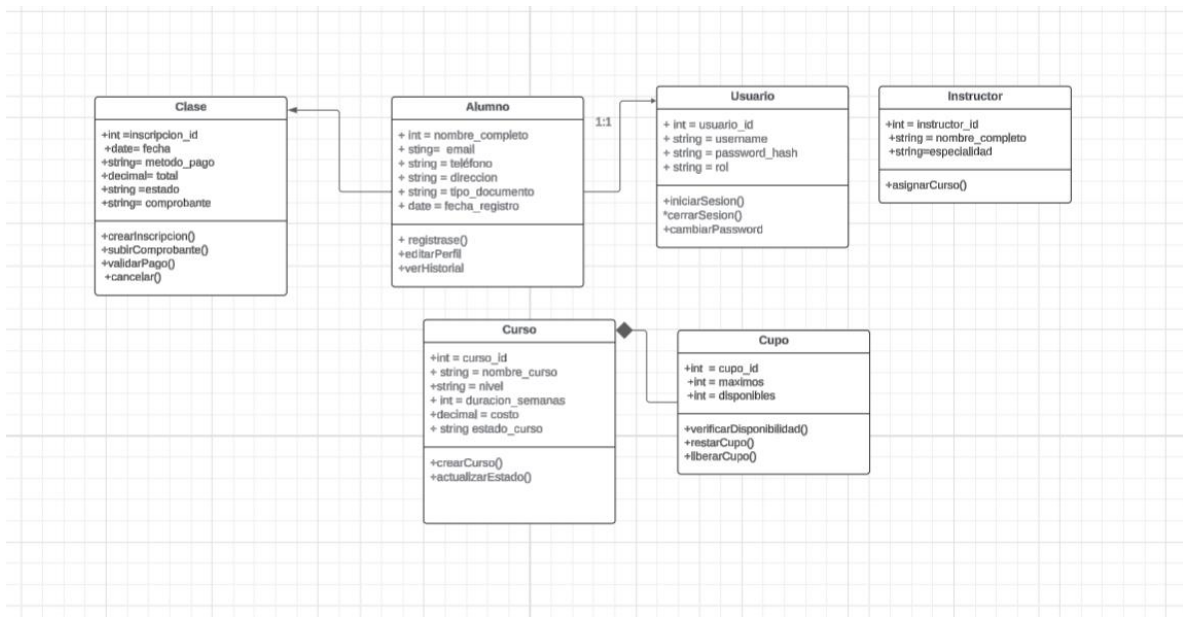


Diagrama de Casos de uso.

