



Proyecto:

“Sistema de Gestión de Gimnasios”

Carrera: Tecnicatura Universitaria en Programación

Cátedra: Práctica Profesional

Tutores:

- Benaventana, Nelson
- Magariños, Luis

Integrantes:

- Andrich, Julián
- Briceño, Andrés
- Dirie, Daiana

AÑO 2021

Índice	1
1. Introducción	3
1.1. Presentación del Proyecto	3
2. Análisis del Sistema	3
2.1. Objetivos	3
2.2. Límites	3
2.3. Descripción Conceptual del Dominio	3
2.4. Modelo de Clases Conceptuales	4
2.5. Requerimientos	5
2.6. Diagrama de Gantt	6
2.7. Diagrama de Casos de Usos	7
2.8. Especificaciones de Casos de Usos	8
3. Diseño del Sistema	15
3.1. Diagrama de Clases	15
3.2. Procesos	16
3.2.1. Proceso Login de Usuario	16
3.2.1.1. Diagrama FlowChart	16
3.2.1.2. Rutinas	17
3.2.1.3. Pantallas	30
3.2.2. Proceso Restablecer Contraseña	31
3.2.2.1. Diagrama FlowChart	31
3.2.2.2. Rutinas	32
3.2.2.3. Pantallas	32
3.2.3. Proceso Información Personal	33
3.2.3.1. Diagrama FlowChart	33
3.2.3.2. Rutinas	34
3.2.3.3. Pantallas	34
3.2.4. Proceso Gestión Alumno	36
3.2.4.1. Diagrama FlowChart	36
3.2.4.2. Rutinas	37
3.2.4.3. Pantallas	38
3.2.4.4. Salida Impresa	41
3.2.5. Proceso Gestión Dashboard	42
3.2.5.1. Diagrama FlowChart	42
3.2.5.2. Rutinas	43
3.2.5.3. Pantallas	44
3.2.6. Proceso Gestión Usuario	47
3.2.6.1. Diagrama FlowChart	47
3.2.6.2. Rutinas	48
3.2.6.3. Pantallas	48

3.2.7.	Proceso Gestión Rutina	51
3.2.7.1.	Diagrama FlowChart	51
3.2.7.2.	Rutinas	52
3.2.7.3.	Pantallas	53
3.2.7.4.	Salida Impresa	55
3.2.8.	Proceso Gestión Asistencia	56
3.2.8.1.	Diagrama FlowChart	56
3.2.8.2.	Rutinas	57
3.2.8.3.	Pantallas	58
3.2.9.	Proceso Gestión Clase	61
3.2.9.1.	Diagrama FlowChart	61
3.2.9.2.	Rutinas	62
3.2.9.3.	Pantallas	63
3.2.10.	Proceso Gestión Cuota	66
3.2.10.1.	Diagrama FlowChart	66
3.2.10.2.	Rutinas	67
3.2.10.3.	Pantallas	68
3.2.11.	Proceso Gestión Tarifa	71
3.2.11.1.	Diagrama FlowChart	71
3.2.11.2.	Rutinas	72
3.2.11.3.	Pantallas	73
3.2.12.	Proceso Gestión Ejercicio	75
3.2.12.1.	Diagrama FlowChart	75
3.2.12.2.	Rutinas	76
3.2.12.3.	Pantallas	77
4.	Base de Datos	79
4.1.	Esquema de Base de Datos	79
4.2.	Diccionario de Datos	80

➤ Introducción:

Presentación del Proyecto

Proyecto: Sistema de Gestión de Gimnasios

El siguiente proyecto fue diseñado en base al gimnasio "Atenas GYM" de la ciudad de Concordia, Entre Ríos.

Dicho gimnasio brinda el servicio de entrenamiento y cuenta con una capacidad máxima de 10 alumnos por turno, supervisados por 2 profesores.

La actividad deportiva de la clase y el área de recepción, en la cual todas las acciones de gestión son realizadas de forma manual, son responsabilidad de los profesores presentes en ese momento.



Análisis del Sistema:

Objetivos

Los objetivos de este sistema se basan en automatizar la gestión y administración del gimnasio.

Límites del sistema

El sistema comienza desde que un cliente abona la primera cuota hasta la suspensión de los servicios por parte del mismo.

Descripción Conceptual del Dominio

Para comenzar con el servicio de Atenas GYM, el cliente puede solicitar información de las clases y aranceles en la recepción del establecimiento o telefónicamente.

El profesor de turno le dará a elegir las opciones de días y horarios en los que el cliente podría concurrir a entrenar y le informará que puede tomar hasta 2 clases de prueba sin costo.

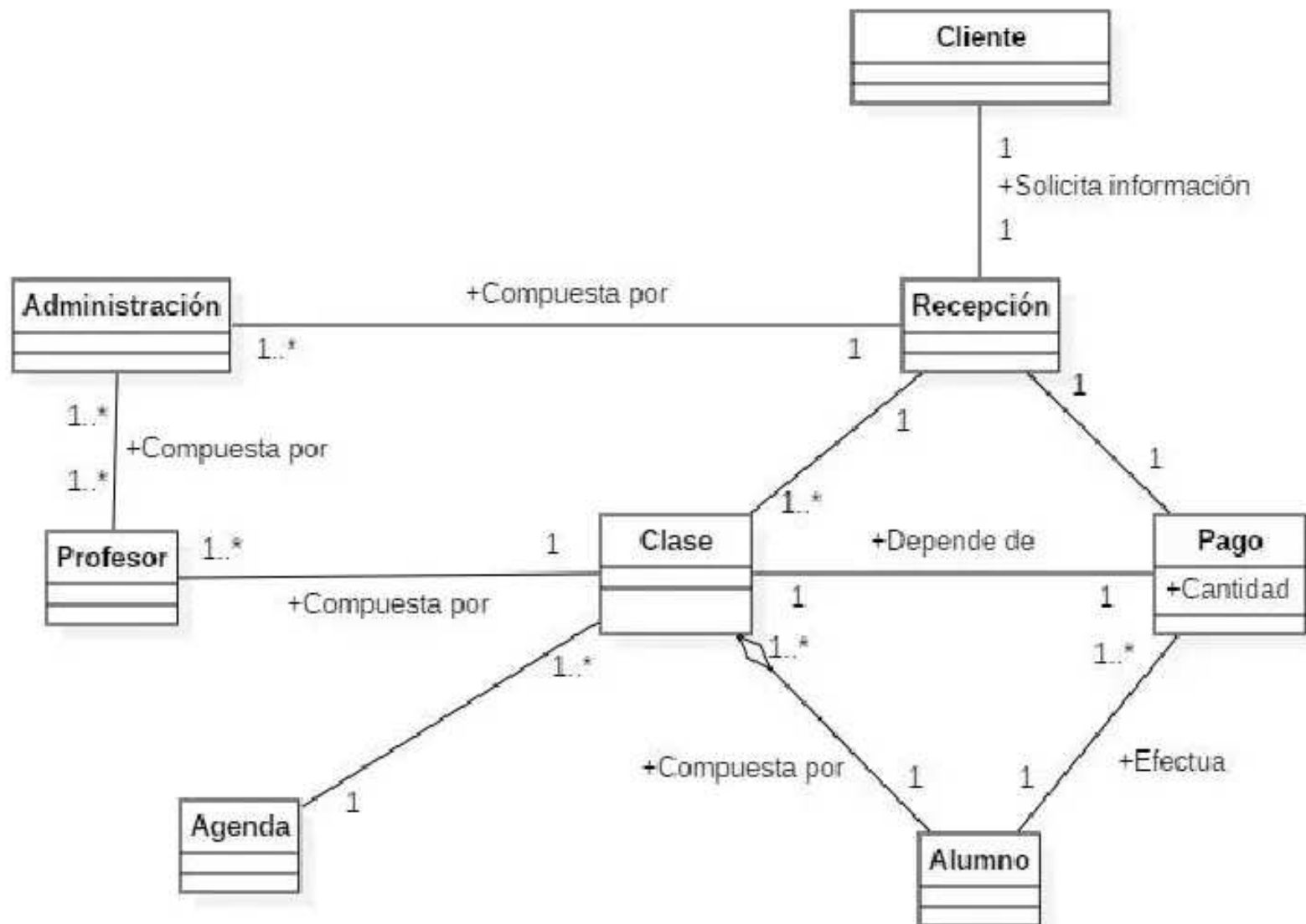
También le informará que al momento de inscribirse deberá presentar una ficha médica y abonar la cuota correspondiente al mes en curso.

El profesor de turno, registrará el pago en un cuaderno donde se asientan los mismos. También incluirá los datos del alumno en una planilla destinada para tal fin.

Pasadas las primeras 2 semanas de entrenamiento, el profesor diseñará una ficha con la rutina para dicho alumno.

Al finalizar cada mes de entrenamiento el alumno deberá abonar la cuota, recibiendo un comprobante de la misma.

Modelo de Clases Conceptuales



Requerimientos

Requerimientos Funcionales:

- El sistema funcionará en plataformas web.
- El sistema debe contar con acceso de usuarios válidos.
- Los alumnos y profesores podrán consultar y modificar su información personal.
- Los alumnos podrán ver las rutinas personalizadas con posibilidad de imprimirlas.
- Los profesores podrán dar de alta, modificar y dar de baja las rutinas personalizadas que se le otorgan al alumno.
- Los profesores podrán gestionar las tarifas.
- El profesor administrador podrá dar de alta, modificar y dar de baja una clase.
- Los profesores podrán obtener un listado de los alumnos inscriptos agrupados por clase.
- Los alumnos podrán consultar las clases a las que está inscripto.
- Los alumnos y profesores podrán visualizar su perfil completo.
- Los profesores podrán registrar nuevos ejercicios.
- Los alumnos podrán hacer consultas sobre el total de las cuotas abonadas y los profesores podrán registrar y consultar las cuotas abonadas por los alumnos.
- Los profesores podrán registrar y consultar la cantidad de veces que el alumno realizó un entrenamiento en la semana.
- Los profesores podrán dar de alta, baja y modificar cuentas de alumnos y el administrador realizar las mismas acciones con sus empleados.

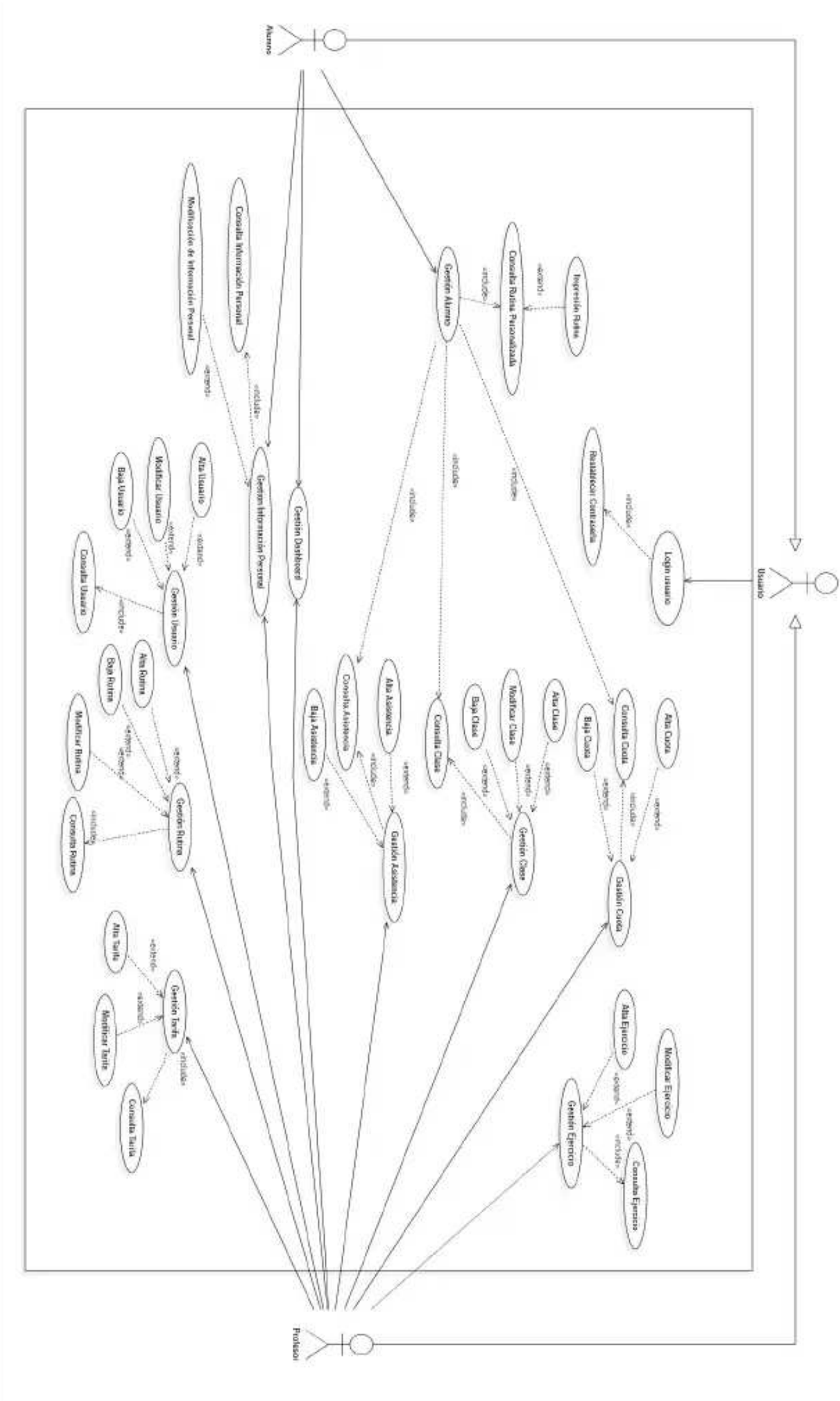
Requerimientos No Funcionales:

- Seguridad.
- Accesibilidad.
- Usabilidad.
- Estabilidad.
- Portabilidad.
- Escalabilidad.
- Mantenibilidad.

Diagrama de Gantt

	Agosto			Septiembre				Octubre				Noviembre				Diciembre				Enero				Febrero				Marzo				Abril				Mayo				Junio				Julio				
Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Actividades																																																
Presentación del proyecto																																																
Relevamiento																																																
Objetivos																																																
Limites																																																
Dominio del sistema																																																
Diagrama de clases conceptual																																																
Requerimientos																																																
Diagrama de Gantt																																																
Diagrama de casos de uso																																																
Especificaciones de casos de uso																																																
Diagrama de clases																																																
Procesos																																																
Plan de implementación																																																
Desarrollo																																																
Implementación																																																

Diagrama de Casos De Usos



Especificaciones de casos de uso:

Caso de uso: Login de Usuario

- Actor: usuario.
- Descripción: el caso de uso permite al ingreso al sistema.
- Flujo básico:
 - El usuario ingresa el nombre de usuario elegido junto a su contraseña.
 - El sistema valida los datos mediante una base de datos.
 - El sistema autoriza el ingreso a la aplicación.
- Flujos Alternos:
 - En el caso que la pagina no permita el ingreso:
 - Se notificará mediante un mensaje que el nombre de usuario y/o contraseña son incorrectos. Luego se le permitirá volver a intentar el ingreso de datos en el formulario.
- Puntos de inclusión:
 - Restablecer contraseña: el usuario puede restablecer su contraseña en el caso que se la olvide.
- Puntos de extensión: no tiene.
- Notas: no tiene.

Caso de uso: Gestión Alumno

- Actor: alumno
- Descripción: el caso de uso permite al alumno gestionar su cuenta.
- Flujo básico:
 - El sistema mostrará en pantalla el dashboard correspondiente donde el alumno podrá realizar las operaciones que solicite.
- Flujos Alternos:
 - En caso que no haya una sesión activa:
 - Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión:
 - Consulta de Cuotas:
 - El alumno podrá consultar la cantidad de cuotas que abonó y cuánto falta para que caduque su última cuota.

- Consulta Rutina Personalizada:
 - El alumno podrá consultar su rutina personalizada con posibilidad de imprimirla.
- Consulta de Clases:
 - El alumno podrá consultar las clases a las que está inscripto.
- Consulta de Asistencias:
 - El alumno podrá consultar sus asistencias.
- Puntos de extensión: no tiene
- Notas: no tiene.

Caso de uso: Gestión Información Personal

- Actor: alumno y profesor.
- Descripción: permite al usuario consultar y modificar su información personal.
- Flujo básico:
 - El alumno o profesor seleccionará el botón de información personal.
 - La base de datos devuelve dicha información y el sistema la carga en la pantalla.
 - El alumno o profesor puede realizar modificaciones en el caso de necesitarlo.
- Flujos Alternos:
 - En caso que no haya una sesión activa:
 - Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión:
 - Consulta Información personal:
 - El alumno o profesor puede consultar su información personal.
- Puntos de extensión:
 - Modificación de información personal:
 - El alumno o profesor puede modificar su información personal.
- Notas: no tiene.

Caso de uso: Gestión Cuota

- Actor: profesor.
- Descripción: el profesor podrá registrar y consultar las cuotas abonadas por los alumnos.
- Flujo básico:
 - El profesor selecciona el botón dependiendo si desea registrar o consultar la cuota.
 - El sistema cargará los datos de la base de datos y los mostrará en pantalla.

- El profesor hará los cambios pertinentes dependiendo la opción que haya elegido.
- Flujos Alternos:
 - En caso que no haya una sesión activa:
 - Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión:
 - Consulta:
 - El profesor podrá consultar las cuotas abonadas de cada alumno con el objetivo de verificar que cumpla con el plazo de abonado.
- Puntos de extensión:
 - Alta:
 - El profesor tendrá la posibilidad de registrar la cuota, una vez que el alumno haya abonado en el gimnasio.
 - Baja:
 - El profesor podrá dar de baja la cuota de un alumno cuando lo desee.
- Notas: no tiene.

Caso de uso: Gestión Asistencia

- Actor: profesor.
- Descripción: el caso de uso permite al profesor registrar, consultar y dar de baja las asistencias de los alumnos.
- Flujo básico:
 - El profesor ingresa a la sección de asistencias.
 - Según la acción que desea realizar, clickea en el botón correspondiente.
- Flujos Alternos:
 - En caso que no haya una sesión activa:
 - Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión:
 - Consulta:
 - El profesor podrá consultar la asistencia del alumno con el fin de verificar que el mismo cumpla con el plan abonado.
- Puntos de extensión:
 - Alta:
 - El profesor podrá registrar las asistencias de los alumnos.
 - Baja:
 - El profesor podrá dar de baja la asistencia de un alumno cuando lo desee.

- Notas: no tiene.

Caso de uso: Gestión Clase

- Actor: profesor.
- Descripción: el caso de uso permite al profesor registrar, consultar y dar de baja las clases del gimnasio.
- Flujo básico:
 - El profesor ingresa a la sección de clases.
 - Dependiendo de la acción que desee realizar, selecciona el botón correspondiente.
- Flujos Alternos:
 - En caso que no haya una sesión activa:
 - Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión:
 - Consulta:
 - El profesor podrá consultar las clases que hay registradas junto a los alumnos inscriptos a las mismas.
- Puntos de extensión:
 - Alta:
 - El profesor podrá dar de alta una clase.
 - Baja:
 - El profesor podrá dar de baja una clase.
 - Modificar:
 - El profesor podrá modificar la información de una clase.
- Notas: no tiene.

Caso de uso: Gestión Rutina

- Actor: profesor.
- Descripción: el caso de uso permite al profesor dar de alta, modificar y dar de baja una rutina.
- Flujo básico:
 - El profesor ingresa a la sección de rutinas.
 - Dependiendo de la acción que desee realizar, selecciona el botón correspondiente.
- Flujos Alternos:
 - En caso que no haya una sesión activa:

- Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión:
 - Consulta:
 - El profesor podrá consultar la rutina emitida según el alumno.
- Puntos de extensión:
 - Alta:
 - El profesor podrá dar de alta las rutinas de los alumnos.
 - Baja:
 - El profesor podrá dar de baja las rutinas de los alumnos.
 - Modificar:
 - El profesor podrá modificar las rutinas de los alumnos.
- Notas: no tiene.

Caso de uso: Gestión Usuario

- Actor: profesor.
- Descripción: el caso de uso permite al profesor dar de alta, modificar y dar de baja un usuario.
- Flujo básico:
 - El profesor ingresa a la sección de usuarios.
 - Dependiendo de la acción que desee realizar, selecciona el botón correspondiente.
- Flujos Alternos:
 - En caso que no haya una sesión activa:
 - Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión:
 - Consulta:
 - El profesor podrá consultar los usuarios inscriptos en el sistema.
- Puntos de extensión:
 - Alta:
 - El profesor podrá dar de alta un usuario.
 - Baja:
 - El profesor podrá dar de baja un usuario.
 - Modificar:

- El profesor podrá modificar un usuario.

➤ Notas: el único rol capaz de realizar los puntos de extensión será el rol administrador.

Caso de uso: Gestión Dashboard

- Actor: alumno y profesor.
- Descripción: el caso de uso permite al alumno o profesor visualizar el dashboard respectivo según el rol que cumplan en el sistema.
- Flujo básico:
 - Según el rol que cumpla, el alumno o profesor verá la pantalla principal del sistema.
 - Según la acción que desee realizar, el alumno o profesor, tendrá en un sidebar los botones que lo llevaran a dichas acciones.
- Flujos Alternos:
 - En caso que no haya una sesión activa:
 - Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión: no tiene.
- Puntos de extensión: no tiene.
- Notas: no tiene.

Caso de uso: Gestión Tarifa

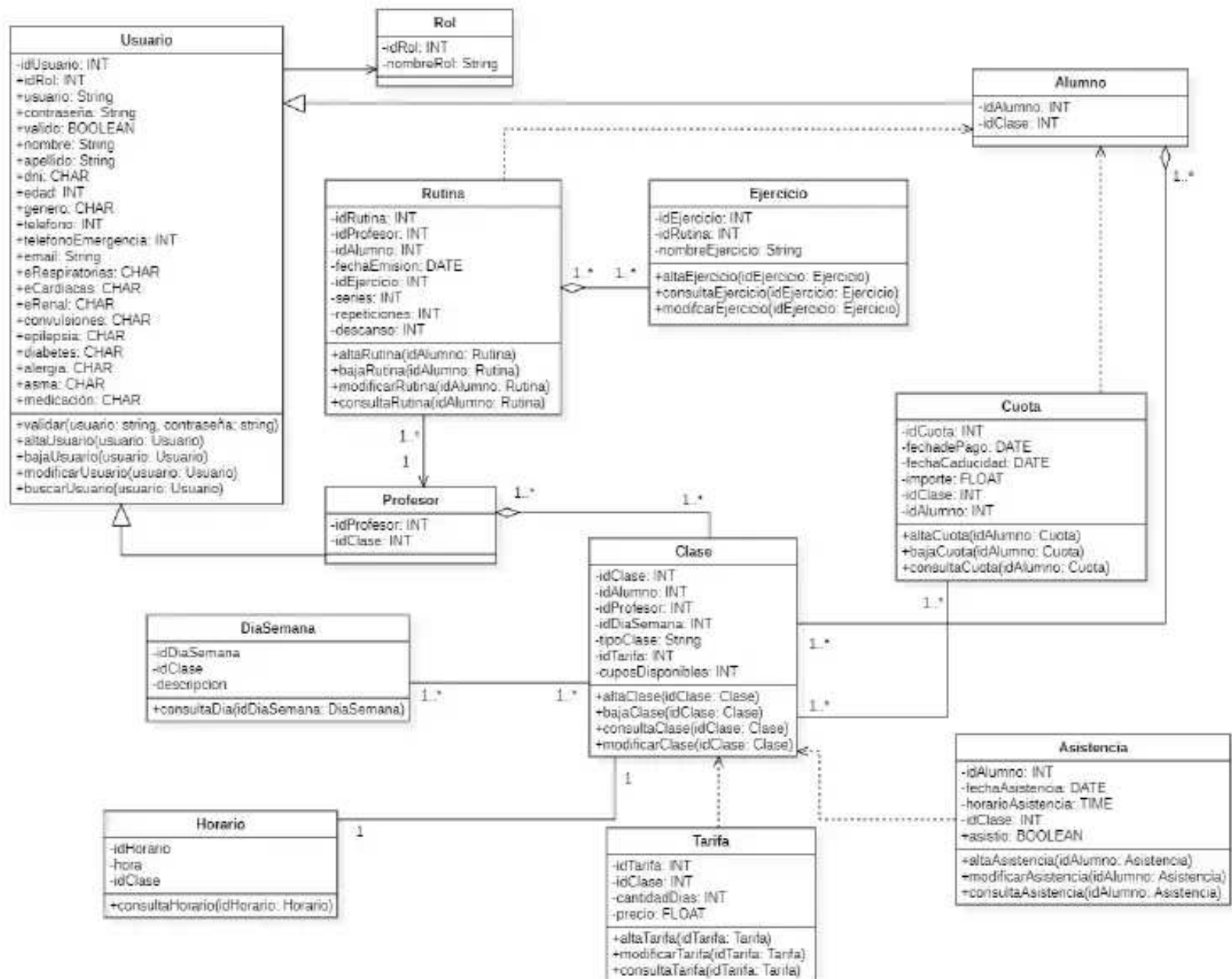
- Actor: profesor.
- Descripción: el caso de uso permite al profesor dar de alta y modificar las tarifas de las clases según la cantidad de días.
- Flujo básico:
 - El profesor ingresa a la sección de tarifas.
 - Dependiendo de la acción que desee realizar, selecciona el botón correspondiente.
- Flujos Alternos:
 - En caso que no haya una sesión activa:
 - Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión:
 - Consulta:
 - El profesor podrá consultar las tarifas registradas en la base de datos.
- Puntos de extensión:

- Alta:
 - El profesor podrá dar de alta una tarifa.
- Modificar:
 - El profesor podrá modificar una tarifa.
- Notas: no tiene.

Caso de uso: Gestión Ejercicio

- Actor: profesor.
- Descripción: el caso de uso permite al profesor dar de alta y modificar los ejercicios que se pueden incluir en las rutinas.
- Flujo básico:
 - El profesor ingresa a la sección de ejercicio.
 - Dependiendo de la acción que desee realizar, selecciona el botón correspondiente.
- Flujos Alternos:
 - En caso que no haya una sesión activa:
 - Si no hay una sesión activa al momento de solicitar este caso de uso se mostrará en pantalla un mensaje avisando al usuario que debe loguearse para poder ingresar a dicha sección.
- Puntos de inclusión:
 - Consulta:
 - El profesor podrá consultar los ejercicios registradas en la base de datos
- Puntos de extensión:
 - Alta:
 - El profesor podrá dar de alta un nuevo ejercicio.
 - Modificar:
 - El profesor podrá modificar un ejercicio.
- Notas: no tiene.

Diagrama de Clases:

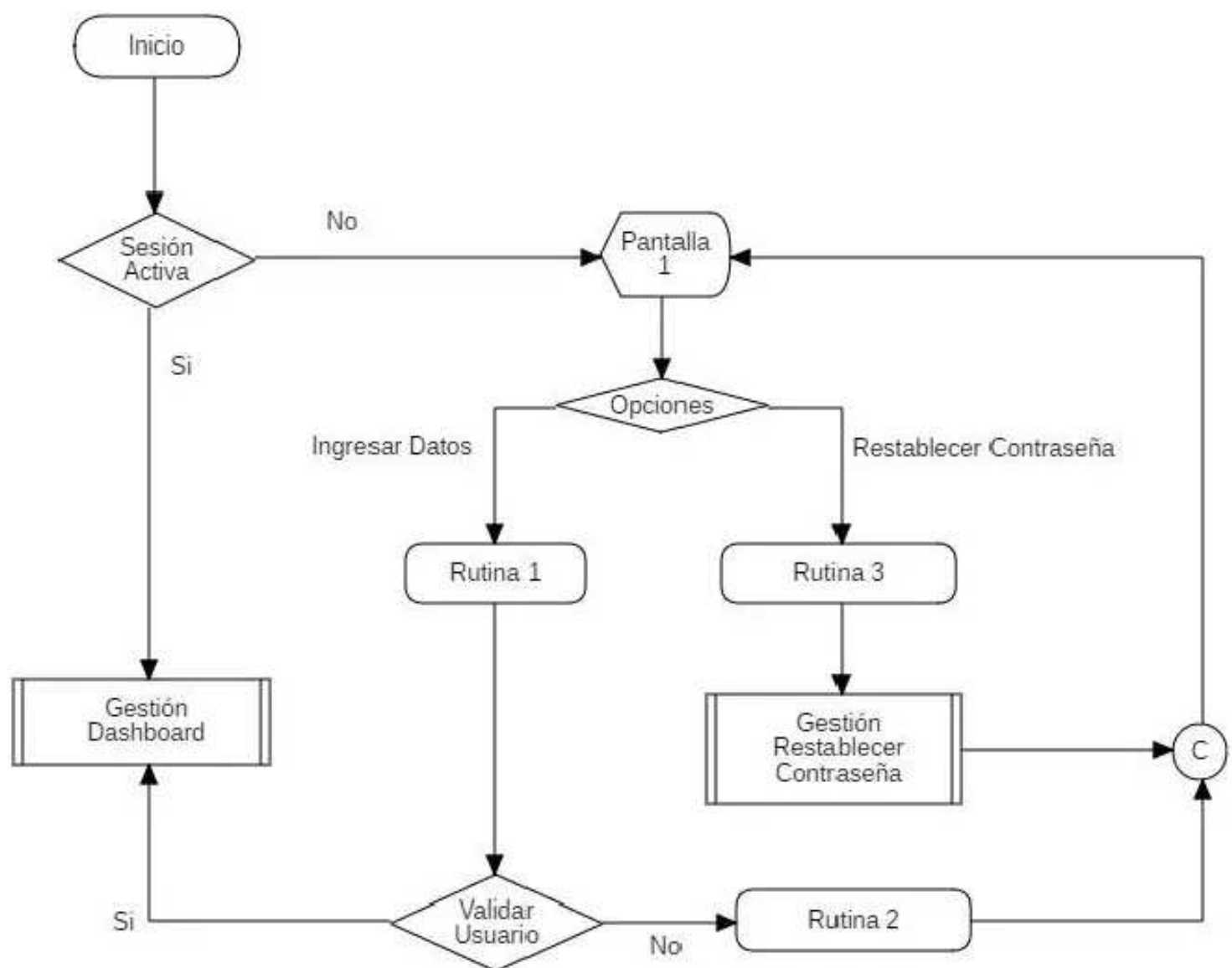


Procesos

Proceso Login de Usuario

→ Diagrama FlowChart:

Proceso: Login Usuario



→ Rutinas:

```
public class Rol {  
  
    private int idRol;  
    private string nombreRol;  
  
    public Rol (idRol, nombreRol) {  
        this.idRol = idRol;  
        this.nombreRol = nombreRol;  
    }  
  
    public int Rol::getidRol () {  
        retornar this.idRol;  
    }  
  
    public string Rol::getNombreRol () {  
        retornar this.nombreRol;  
    }  
}  
  
public class Usuario {  
  
    private int idUsuario;  
    public int Rol_idRol;  
    public string usuario;  
    public string contraseña;  
    public logico valido;  
    public string nombre;  
    public string apellido;  
    public char dni;  
    public int edad;  
    public char genero;  
    public int telefono;  
    public int telefonoEmergencia;  
    public string email;  
    public char eRespiratorias;  
    public char eCardiacas;  
    public char eRenal;  
  
    public char convulsiones;  
    public char epilepsia;
```

```
public char diabetes;  
public char alergia;  
public char asma;  
public char medicacion;
```

```
public Usuario (int idUsuario,  
                int Rol_idRol,  
                string usuario,  
                string contraseña,  
                logico valido,  
                string nombre,  
                string apellido,  
                char dni,  
                int edad,  
                char genero,  
                int telefono,  
                int telefonoEmergencia,  
  
                string email,  
                char eRespiratorias,  
                char eCardiacas,  
                char eRenal,  
                char convulsiones,  
                char epilepsia,  
                char diabetes,  
                char alergia,  
                char asma,  
                char medicacion) {
```

```
    this.idUsuario = idUsuario;  
    this.Rol_idRol = Rol_idRol;  
    this.usuario = usuario;  
    this.contraseña = contraseña;  
    this.valido = valido;  
    this.nombre = nombre;  
    this.apellido = apellido;  
    this.dni = dni;  
    this.edad = edad;  
    this.genero = genero;  
    this.telefono = telefono;  
    this.telefonoEmergencia = telefonoEmergencia;  
    this.email = email;  
    this.eRespiratorias = eRespiratorias;  
  
    this.eCardiacas = eCardiacas;  
    this.eRenal = eRenal;
```

```

this.convulsiones = convulsiones;
this.epilepsia = epilepsia;
this.diabetes = diabetes;
this.alergia = alergia;
this.asma = asma;
this.medicacion = medicacion;
}

```

```

public buscarUsuario (string usuario) {
    array L_usuario;

    L_usuario = SELECT * FROM usuario, alumno, profesor
                INNER JOIN alumno ON usuario.usuario = alumno.usuario
                INNER JOIN profesor ON usuario.usuario = profesor.usuario
                WHERE usuario IN (SELECT * FROM usuario
                                WHERE usuario.usuario = usuario)

                AND (usuario.usuario = alumno.usuario
                    OR usuario.usuario = profesor.usuario);
    Si L_usuario(0).usuario != "" entonces
        retornar L_usuario;
    FinSi
}

```

```

public validar (string usuario, string contraseña) {
    this.valido = falso;

    buscarUsuario(usuario);

    Si L_usuario(0).usuario != ""
        Si L_usuario(0).contraseña = contraseña entonces
            this.valido = verdadero;
            this.idUsuario = L_usuario(0).idUsuario;
            this.idRol = L_usuario(0).idRol;
        Sino
            mostrar mensaje("Usuario o Contraseña inválidos");
        Fin Si
    Sino
        mostrar mensaje("Debe introducir un usuario y contraseña");
    Fin Si

    retornar this.valido;
}

```

```

public altaUsuario (string usuario) {
    buscarUsuario(usuario);
    Si L_usuario(0).usuario = "" entonces
        CREATE TABLE IF NOT EXISTS Usuario (
            idUsuario INT AUTO_INCREMENT,
            usuario VARCHAR,
            contraseña VARCHAR,
            Rol_idRol INT,
            PRIMARY KEY (idUsuario)
        );

    Segun Caso Rol_idRol
        Caso 0 // alumno
            CREATE TABLE IF NOT EXISTS Alumno (
                idAlumno INT AUTO_INCREMENT,

                usuario_idUsuario INT,
                nombre VARCHAR,
                apellido VARCHAR,
                dni CHAR,
                edad INT,
                genero CHAR,
                telefono INT,
                email VARCHAR,
                telefono_emergencia INT,
                eRespiratorias CHAR,
                eCardiacas CHAR,
                eRenal CHAR,
                convulsiones CHAR,
                epilepsia CHAR,
                diabetes CHAR,
                alergia CHAR,
                asma CHAR,
                medicacion CHAR,
                PRIMARY KEY (idAlumno),
                FOREIGN KEY (usuario_idUsuario)
                REFERENCES usuario (idUsuario));

        INSERT INTO Usuario (usuario, contraseña, Rol_idRol)
        VALUES(this.usuario, this.contraseña, this.Rol_idRol);

        INSERT INTO Alumno (usuario_idUsuario, nombre, apellido, dni, edad,

```

```

genero, telefono, email, telefono_emergencia, eRespiratorias, eCardiacas,
eRenal, convulsiones, epilepsia, diabetes, alergia, asma, medicacion)
VALUES(this.usuario_idUsuario, this.nombre, this.apellido, this.dni,
this.edad, this.genero, this.telefono, this.email, this.telefono_emergencia,
this.eRespiratorias, this.eCardiacas, this.eRenal, this.convulsiones,
this.epilepsia, this.diabetes, this.alergia, this.asma, this.medicacion);

```

```

Caso 1 //profesor
CREATE TABLE IF NOT EXISTS Profesor(
    idProfesor INT AUTO_INCREMENT,
    usuario_idUsuario INT,
    nombre VARCHAR,
    apellido VARCHAR,
    dni CHAR,
    edad INT,
    genero CHAR,
    telefono INT,
    email VARCHAR,
    telefono_emergencia INT,
    eRespiratorias CHAR,
    eCardiacas CHAR,
    eRenal CHAR,
    convulsiones CHAR,
    epilepsia CHAR,
    diabetes CHAR,
    alergia CHAR,
    asma CHAR,
    medicacion CHAR,
    PRIMARY KEY (idProfesor),
    FOREIGN KEY (usuario_idUsuario)
    REFERENCES usuario (idUsuario));

```

```

INSERT INTO Usuario (usuario, contraseña, Rol_idRol)
VALUES(this.usuario, this.contraseña, this.Rol_idRol);

```

```

INSERT INTO Profesor (usuario_idUsuario, nombre, apellido, dni, edad,
genero, telefono, email, telefono_emergencia, eRespiratorias, eCardiacas,
eRenal, convulsiones, epilepsia, diabetes, alergia, asma, medicacion)
VALUES(this.usuario_idUsuario, this.nombre, this.apellido, this.dni,
this.edad, this.genero, this.telefono, this.email, this.telefono_emergencia,
this.eRespiratorias, this.eCardiacas, this.eRenal, this.convulsiones,
this.epilepsia, this.diabetes, this.alergia, this.asma, this.medicacion);

```

Caso 2

//administrador

```
CREATE TABLE IF NOT EXISTS Profesor(  
    idProfesor INT AUTO_INCREMENT,  
    usuario_idUsuario INT,  
    nombre VARCHAR,  
    apellido VARCHAR,  
    dni CHAR,  
    edad INT,  
    genero CHAR,  
    telefono INT,  
    email VARCHAR,  
    telefono_emergencia INT,  
    eRespiratorias CHAR,  
    eCardiacas CHAR,  
  
    eRenal CHAR,  
    convulsiones CHAR,  
    epilepsia CHAR,  
    diabetes CHAR,  
    alergia CHAR,  
    asma CHAR,  
    medicacion CHAR,  
    PRIMARY KEY (idProfesor),  
    FOREIGN KEY (usuario_idUsuario)  
    REFERENCES usuario (idUsuario));
```

```
INSERT INTO Usuario (usuario, contraseña, Rol_idRol)  
VALUES(this.usuario, this.contraseña, this.Rol_idRol);
```

```
INSERT INTO Profesor (usuario_idUsuario, nombre, apellido, dni, edad,  
genero, telefono, email, telefono_emergencia, eRespiratorias, eCardiacas,  
eRenal, convulsiones, epilepsia, diabetes, alergia, asma, medicacion)  
VALUES(this.usuario_idUsuario, this.nombre, this.apellido, this.dni,  
this.edad, this.genero, this.telefono, this.email, this.telefono_emergencia,  
this.eRespiratorias, this.eCardiacas, this.eRenal, this.convulsiones,  
this.epilepsia, this.diabetes, this.alergia, this.asma, this.medificacion);
```

```
}
```

```
}
```

```
}
```

```
public bajaUsuario (string usuario) {  
    buscarUsuario(usuario);
```

```

Si L_usuario(0).usuario != "" entonces
    DELETE FROM usuario      //delete on cascade
    WHERE L_usuario(0).idUserio = idUsuario
}

public modificarUsuario (string usuario) {
    buscarUsuario(usuario);
    Si L_usuario(0).usuario != "" entonces
        Segun Caso Rol_idRol
            Caso 0 //alumno
                UPDATE usuario SET L_usuario(0).usuario = usuario;
                UPDATE usuario SET L_usuario(0).contraseña = contraseña;
                UPDATE usuario SET L_usuario(0).usuario = usuario;
                UPDATE alumno SET L_usuario(0).nombre = nombre;
                UPDATE alumno SET L_usuario(0).apellido = apellido;
                UPDATE alumno SET L_usuario(0).dni = dni;
                UPDATE alumno SET L_usuario(0).edad = edad;

                UPDATE alumno SET L_usuario(0).genero = genero;
                UPDATE alumno SET L_usuario(0).telefono = telefono;
                UPDATE alumno SET L_usuario(0).email = email;
                UPDATE alumno SET L_usuario(0).telefono_emergencia =
telefono_emergencia;
                UPDATE alumno SET L_usuario(0).eRespiratorias = eRespiratorias;
                UPDATE alumno SET L_usuario(0).eCardiacas = eCardiacas;
                UPDATE alumno SET L_usuario(0).eRenal = eRenal;
                UPDATE alumno SET L_usuario(0).convulsiones = convulsiones;
                UPDATE alumno SET L_usuario(0).epilepsia = epilepsia;
                UPDATE alumno SET L_usuario(0).diabetes = diabetes;
                UPDATE alumno SET L_usuario(0).alergia = alergia;
                UPDATE alumno SET L_usuario(0).asma = asma;
                UPDATE alumno SET L_usuario(0).medicacion = medicacion;
            Caso 1 //profesor
                UPDATE usuario SET L_usuario(0).usuario = usuario;
                UPDATE usuario SET L_usuario(0).contraseña = contraseña;
                UPDATE usuario SET L_usuario(0).usuario = usuario;
                UPDATE profesor SET L_usuario(0).nombre = nombre;
                UPDATE profesor SET L_usuario(0).apellido = apellido;
                UPDATE profesor SET L_usuario(0).dni = dni;
                UPDATE profesor SET L_usuario(0).edad = edad;
                UPDATE profesor SET L_usuario(0).genero = genero;
                UPDATE profesor SET L_usuario(0).telefono = telefono;
                UPDATE profesor SET L_usuario(0).email = email;

                UPDATE profesor SET L_usuario(0).telefono_emergencia =
telefono_emergencia;

```



```

UPDATE profesor SET L_usuario(0).eRespiratorias = eRespiratorias;
UPDATE profesor SET L_usuario(0).eCardiacas = eCardiacas;
UPDATE profesor SET L_usuario(0).eRenal = eRenal;
UPDATE profesor SET L_usuario(0).convulsiones = convulsiones;
UPDATE profesor SET L_usuario(0).epilepsia = epilepsia;
UPDATE profesor SET L_usuario(0).diabetes = diabetes;
UPDATE profesor SET L_usuario(0).alergia = alergia;
UPDATE profesor SET L_usuario(0).asma = asma;
UPDATE profesor SET L_usuario(0).medicacion = medicacion;

```

Caso 2 //administrador

```

UPDATE usuario SET L_usuario(0).usuario = usuario;
UPDATE usuario SET L_usuario(0).contraseña = contraseña;
UPDATE usuario SET L_usuario(0).usuario = usuario;
UPDATE profesor SET L_usuario(0).nombre = nombre;
UPDATE profesor SET L_usuario(0).apellido = apellido;
UPDATE profesor SET L_usuario(0).dni = dni;
UPDATE profesor SET L_usuario(0).edad = edad;

UPDATE profesor SET L_usuario(0).genero = genero;
UPDATE profesor SET L_usuario(0).telefono = telefono;
UPDATE profesor SET L_usuario(0).email = email;
UPDATE profesor SET L_usuario(0).telefono_emergencia =
telefono_emergencia;
UPDATE profesor SET L_usuario(0).eRespiratorias = eRespiratorias;
UPDATE profesor SET L_usuario(0).eCardiacas = eCardiacas;
UPDATE profesor SET L_usuario(0).eRenal = eRenal;
UPDATE profesor SET L_usuario(0).convulsiones = convulsiones;
UPDATE profesor SET L_usuario(0).epilepsia = epilepsia;
UPDATE profesor SET L_usuario(0).diabetes = diabetes;
UPDATE profesor SET L_usuario(0).alergia = alergia;
UPDATE profesor SET L_usuario(0).asma = asma;
UPDATE profesor SET L_usuario(0).medicacion = medicacion;
    }
}
}

```

```

public int Usuario::getidUsuario () {
    retornar this.idUsuario;
}

```

```

public string Usuario::getUsuario () {
    retornar this.usuario;
}

```

```

public void Usuario::setUsuario (string usuario) {
    this.usuario = usuario;
}

public string Usuario::getContraseña () {
    retornar this.contraseña;
}

public void Usuario::setContraseña (string contraseña) {
    this.contraseña = contraseña;
}

public logico Usuario::getValido () {
    retornar this.valido;
}

public void Usuario::setValido (logico valido) {
    this.valido = valido;
}

public string Usuario::getNombre () {
    retornar this.nombre;
}

public void Usuario::setNombre (string nombre) {
    this.nombre = nombre;
}

public string Usuario::getApellido () {
    retornar this.apellido;
}

public void Usuario::setApellido (string apellido) {
    this.apellido = apellido;
}

```

```
public char Usuario::getDni () {  
    retornar this.dni;  
}
```

```
public void Usuario::setDni (char dni) {  
    this.dni = dni;  
}
```

```
public int Usuario::getEdad () {  
    retornar this.edad;  
}
```

```
public void Usuario::setEdad (int edad) {  
    this.edad = edad;  
}
```

```
public char Usuario::getGenero () {  
    retornar this.genero;  
}
```

```
public void Usuario::setGenero (char genero) {  
    this.genero = genero;  
}
```

```
public int Usuario::getTelefono () {  
    retornar this.telefono;  
}
```

```
public void Usuario::setTelefono (int telefono) {  
    this.telefono = telefono;  
}
```

```
public int Usuario::getTelefonoEmergencia () {  
    retornar this.telefonoEmergencia;  
}
```

```
public void Usuario::setTelefonoEmergencia (int telefonoEmergencia) {  
    this.telefonoEmergencia = telefonoEmergencia;  
}
```

```
public string Usuario::getEmail () {  
    retornar this.email;  
}
```

```
public void Usuario::setEmail (string email) {  
    this.email = email;  
}
```

```
public char Usuario::getERespiratorias () {  
    retornar this.eRespiratorias;  
}
```

```
public void Usuario::setERespiratorias (char eRespiratorias) {  
    this.eRespiratorias = eRespiratorias;  
}
```

```
public char Usuario::getECardiacas () {  
    retornar this.eCardiacas;  
}
```

```
public void Usuario::setECardiacas (char eCardiacas) {  
    this.eCardiacas = eCardiacas;  
}
```

```
public char Usuario::getERenal () {  
    retornar this.eRenal;  
}
```

```
public void Usuario::setERenal (char eRenal) {  
    this.eRenal = eRenal;  
}
```

```
public char Usuario::getConvulsiones () {  
    retornar this.convulsiones;  
}
```

```
public void Usuario::setConvulsiones (char convulsiones) {  
    this.convulsiones = convulsiones;  
}
```

```
public char Usuario::getEpilepsia () {  
    retornar this.epilepsia;  
}
```

```
public void Usuario::setEpilepsia (char epilepsia) {  
    this.epilepsia = epilepsia;  
}
```

```
public char Usuario::getDiabetes () {  
    retornar this.diabetes;  
}
```

```
public void Usuario::setDiabetes (char diabetes) {  
    this.diabetes = diabetes;  
}
```

```
public char Usuario::getAlergia () {  
    retornar this.alergia;  
}
```

```
public void Usuario::setAlergia (char alergia) {  
    this.alergia = alergia;  
}
```

```
public char Usuario::getAsma () {  
    retornar this.asma;  
}
```

```

public void Usuario::setAsma (char asma) {
    this.asma = asma;
}

public char Usuario::getMedicacion () {
    retornar this.medicacion;
}

public void Usuario::setMedicacion (char medicacion) {
    this.medicacion = medicacion;
}

}

public class Alumno extends Usuario {

    private int idAlumno;
    private int idClase;

    public Alumno (int idAlumno, int Clase_idClase) {
        this.idAlumno = idAlumno;
        this.idClase = idClase;
    }

    public int Alumno::getidAlumno () {
        retornar this.idAlumno;
    }

}

public class Profesor extends Usuario {

    private int idProfesor;
    private int idClase;

    public Profesor (int idProfesor, int idClase) {
        this.idProfesor = idProfesor;
        this.idClase = idClase;
    }

    public int Profesor::getidProfesor () {
        retornar this.idProfesor ;
    }

}

```

RUTINA 1:

validar(usuario,contraseña);

RUTINA 2:

Muestra mensaje ("El usuario y/o contraseña son inválidos. Vuelva a intentarlo")

txtUsuario = "";

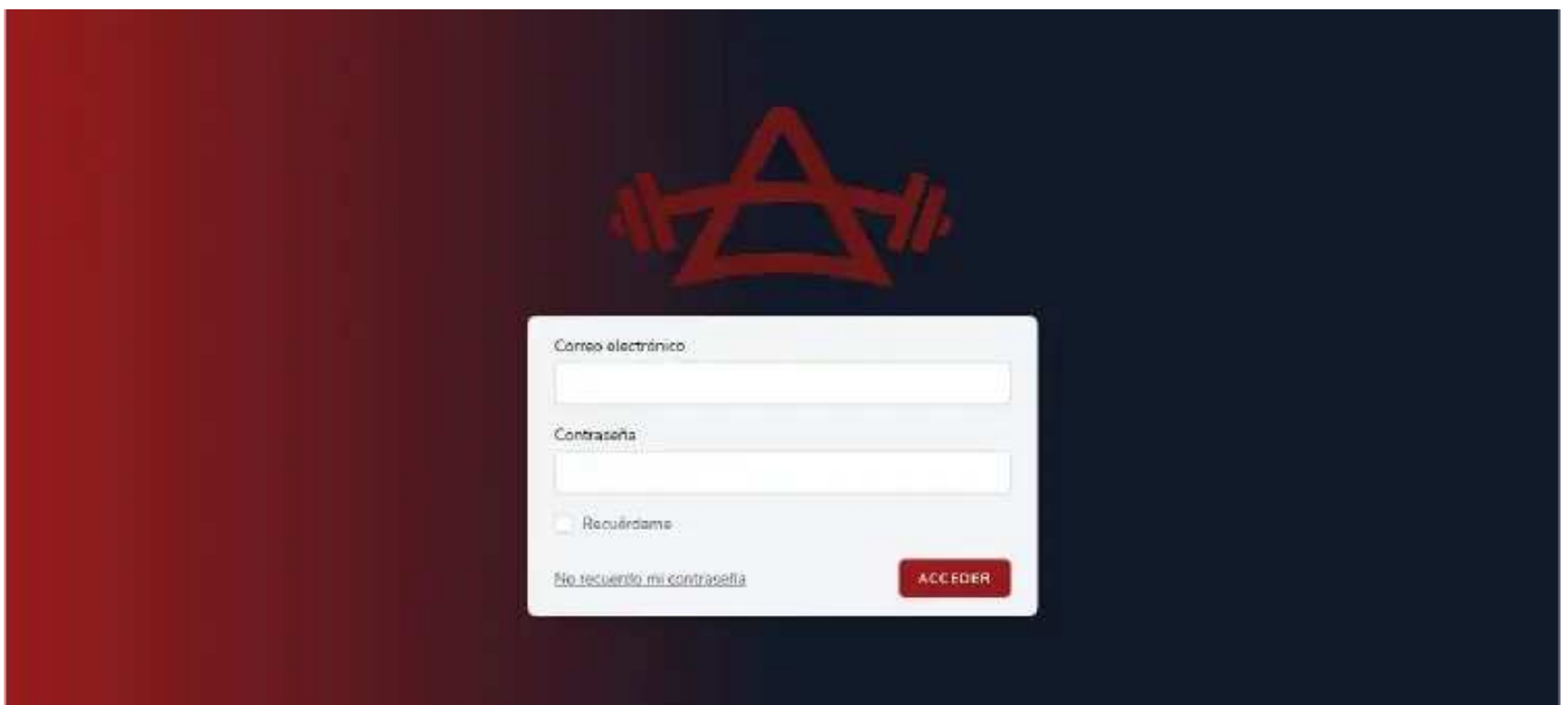
txtPass = "";

RUTINA 3:

Llama al subproceso Gestión Restablecer contraseña.

→ Pantallas:

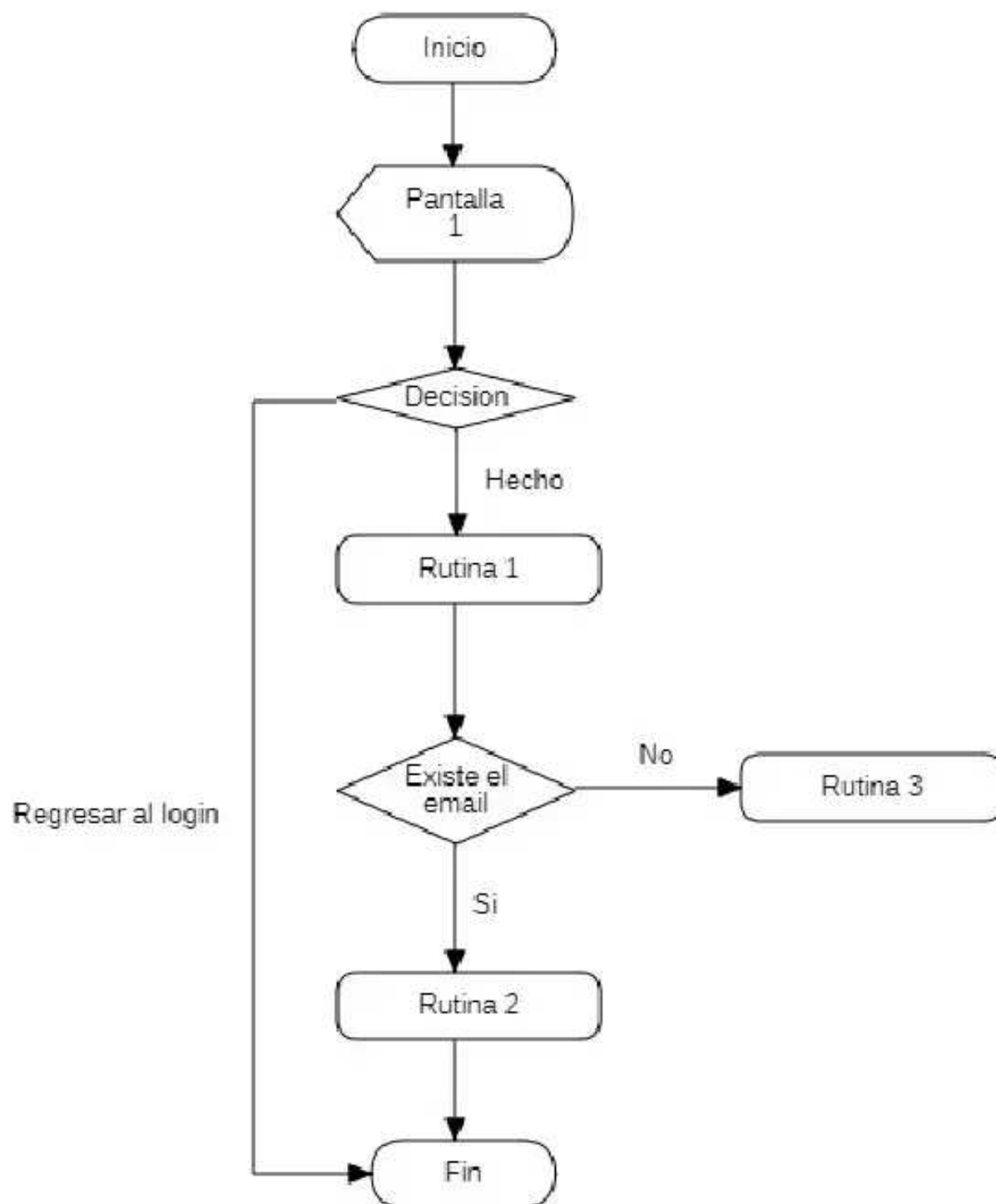
Pantalla 1 (Pantalla de logueo):



Proceso Restablecer Contraseña

→ Diagrama FlowChart:

Proceso: Restablecer Contraseña



→ Rutinas:

RUTINA 1:

Verifica que el email ingresado exista en el sistema.

RUTINA 2:

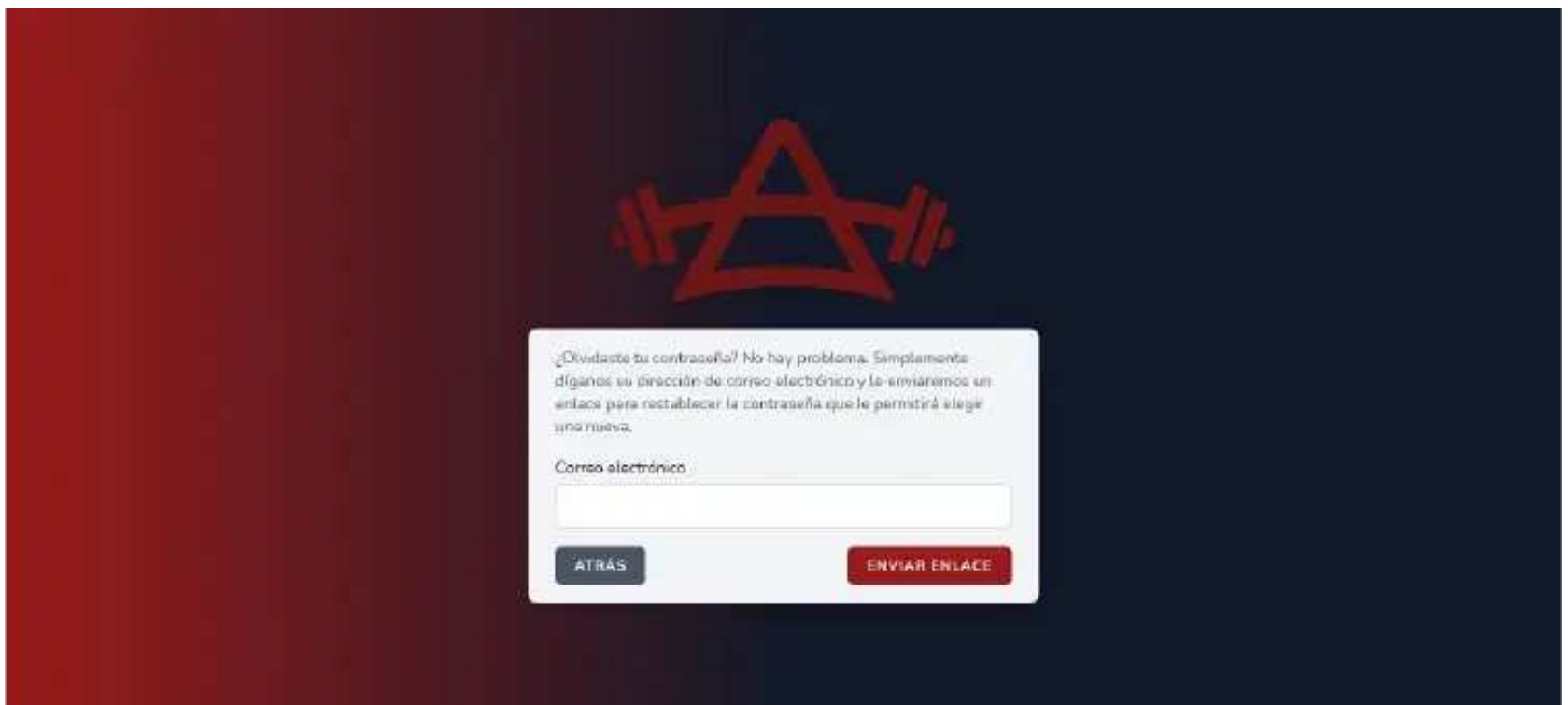
Se envía un email con una contraseña provisoria.

RUTINA 3:

Muestra un mensaje de error “El email ingresado no existe” y limpia la caja de texto.

→ Pantallas:

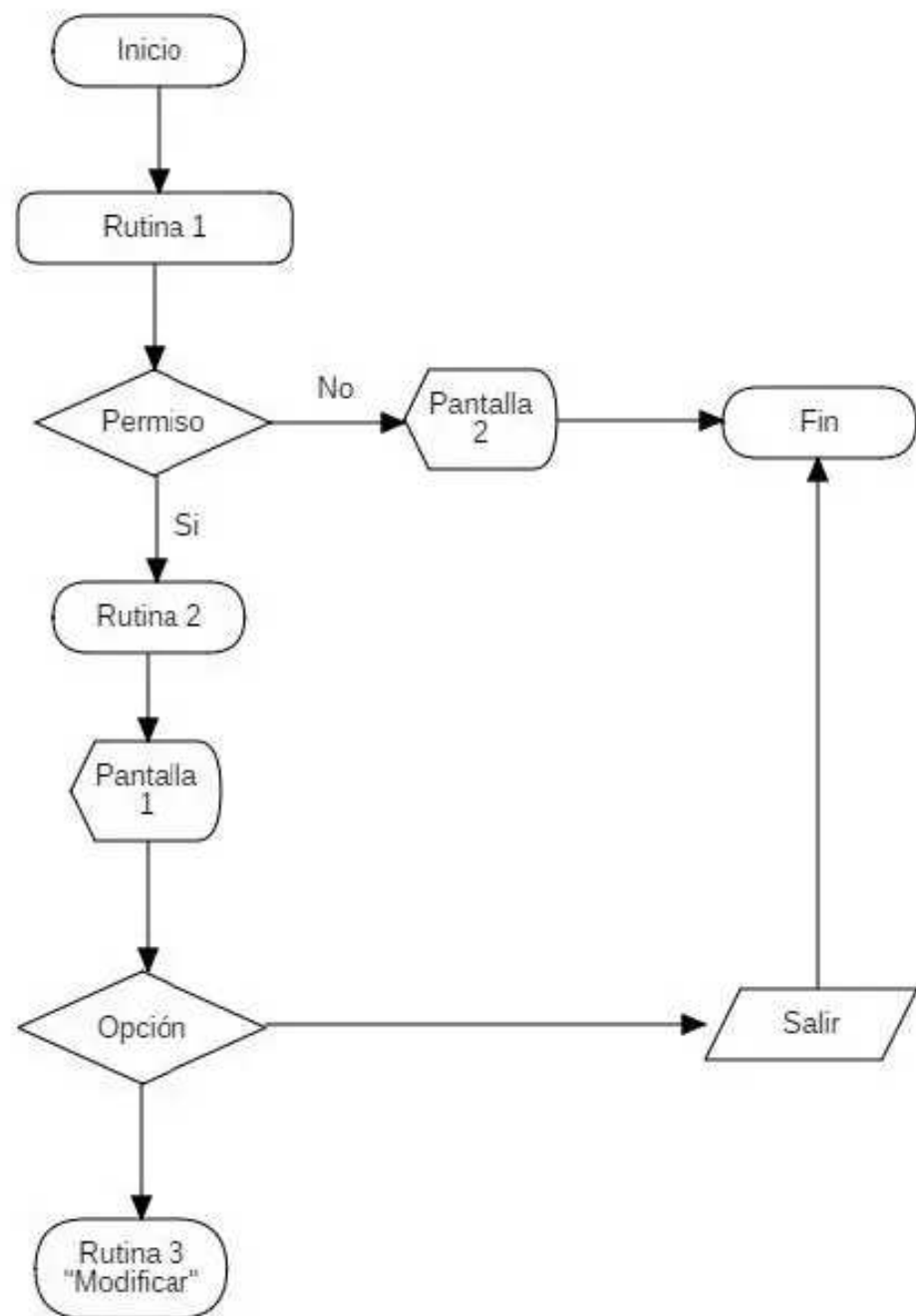
Pantalla 1 (Restablecer contraseña):



Proceso Información Personal

→ Diagrama FlowChart:

Proceso: Gestión Información Personal



→ Rutinas:

RUTINA 1:

Verifica que el usuario tenga permiso para ingresar a dicho proceso.

RUTINA 2:

Consulta los datos del usuario y los imprime en pantalla.

RUTINA 3:

Modifica la información personal del usuario y actualiza la pantalla.

→ Pantallas:

Pantalla 1 (Información Personal):

The screenshot displays the 'Perfil' (Profile) page within the Atenas GYM application. The page features a dark blue header with navigation links: Asistencias, Clases, Rutinas, Ejercicios, Usuarios, Cuotas, Tarifas, and Horarios. The main content area is light gray and contains a white form for editing the user's profile. The form includes the following fields:

- Correo electrónico: admin@admin.com
- Teléfono: 412341234
- Teléfono de emergencia: 123456
- Nueva contraseña: (empty)
- Confirmar nueva contraseña: (empty)

Below the form is a green button labeled 'EDITAR PERFIL'. The footer of the page shows 'Atenas GYM © 2021'.

Pantalla 2 (Error de validación):

401

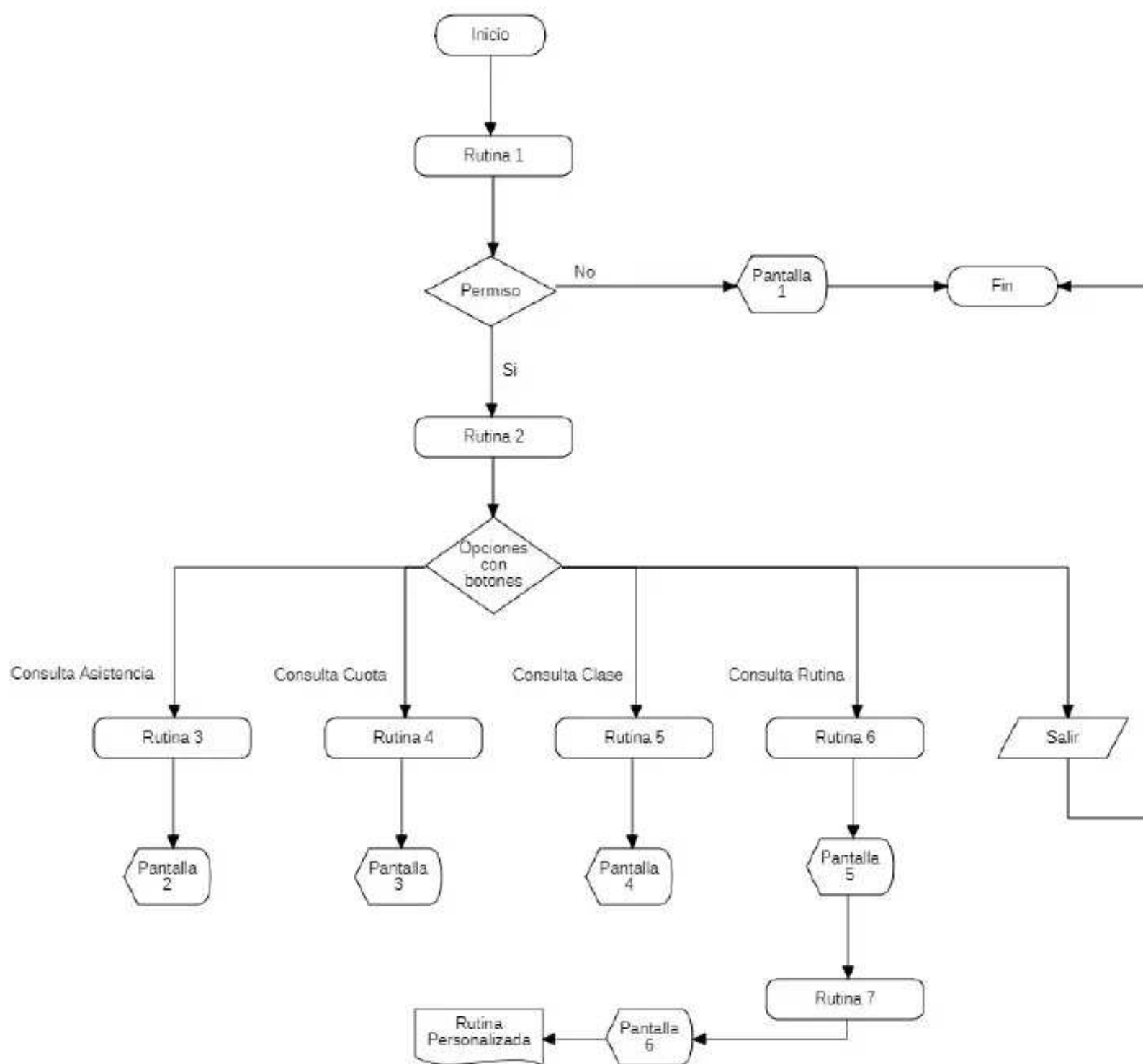
Usted no esta autorizado para
ingresar a esta ruta

IR A DASHBOARD

Proceso Gestión Alumno

→ Diagrama FlowChart:

Proceso: Gestión Alumno



→ Rutinas:

RUTINA 1:

Verifica que el usuario tenga permiso para ingresar a dicho proceso

RUTINA 2:

Despliega los botones con las acciones que el alumno puede seleccionar.

RUTINA 3:

Consulta en la base de datos las asistencias del alumno y las imprime en una pantalla.

RUTINA 4:

Consulta en la base de datos las cuotas abonadas por el alumno y las imprime en una pantalla.

RUTINA 5:

Consulta en la base de datos a qué clases está inscripto el alumno y las imprime en una pantalla.

RUTINA 6:

Muestra un buscador con filtros para seleccionar la clase correspondiente.

RUTINA 7:

Consulta la rutina personalizada del alumno en la base de datos y la muestra en pantalla. Con posibilidad de imprimirla.

→ Pantallas:

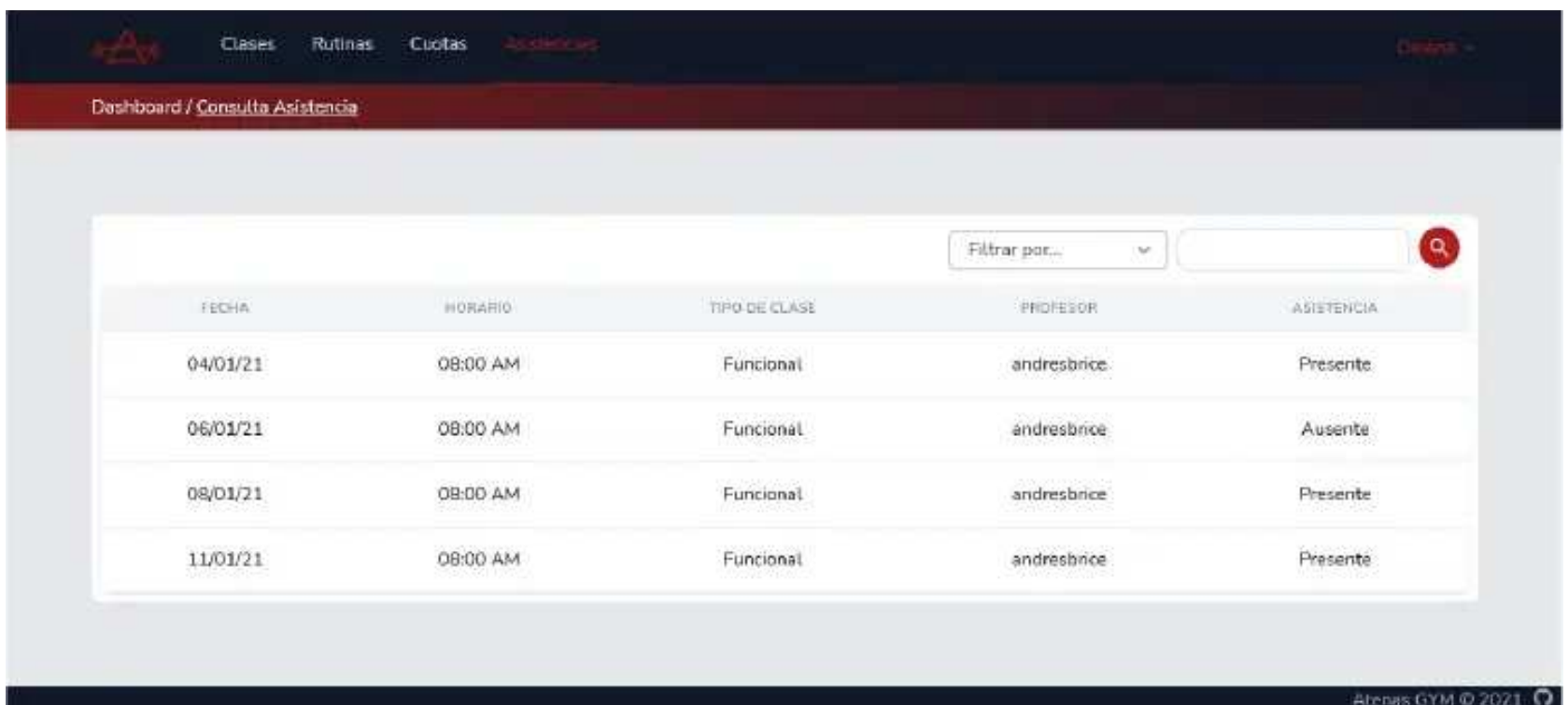
Pantalla 1(Error de Validación):

401

Usted no esta autorizado para
ingresar a esta ruta

[IR A DASHBOARD](#)

Pantalla 2 (Consulta Asistencia):



FECHA	HORARIO	TIPO DE CLASE	PROFESOR	ASISTENCIA
04/01/21	08:00 AM	Funcional	andresbrice	Presente
06/01/21	08:00 AM	Funcional	andresbrice	Ausente
08/01/21	08:00 AM	Funcional	andresbrice	Presente
11/01/21	08:00 AM	Funcional	andresbrice	Presente

Pantalla 3 (Consulta Cuota):

Clares Rutinas Cuotas Asistencias

Usuarios

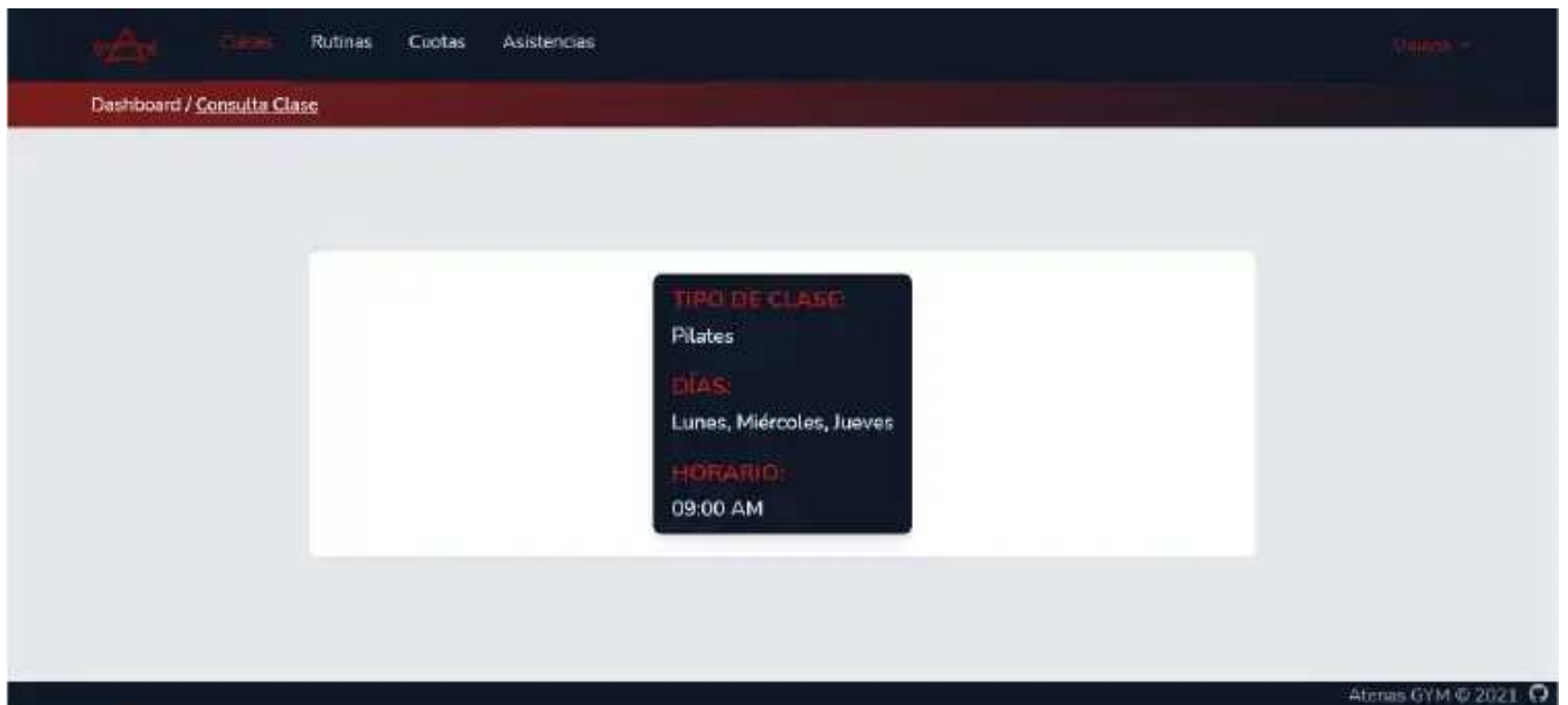
Dashboard / Consulta Cuota

Filtrar por...

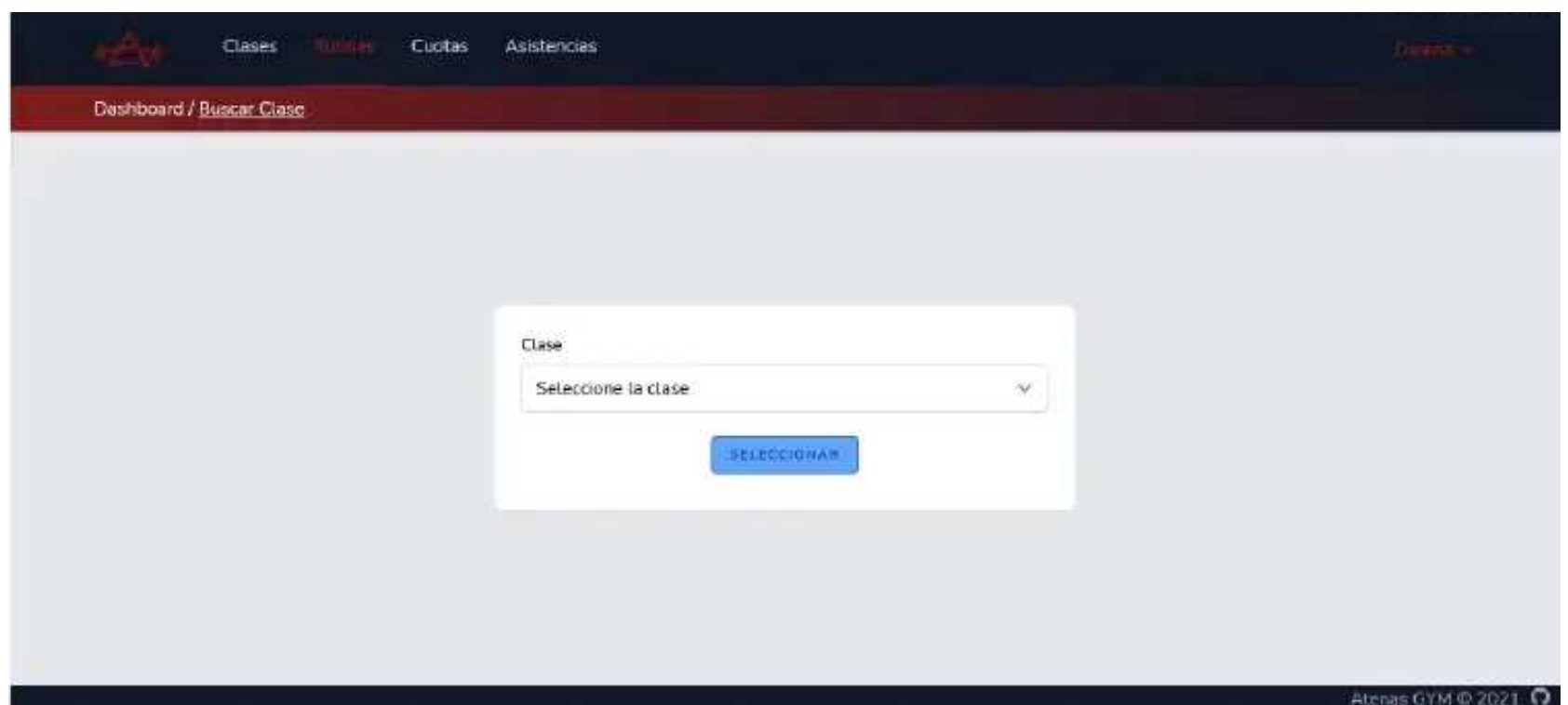
FECHA	FECHA DE CADUCIDAD	DIAS	IMPORTE
04/01/21	04/02/21	3	\$1200
04/02/21	04/03/21	3	\$1200
04/03/21	04/04/21	3	\$1200
04/04/21	04/05/21	3	\$1200

Atenas GYM © 2021

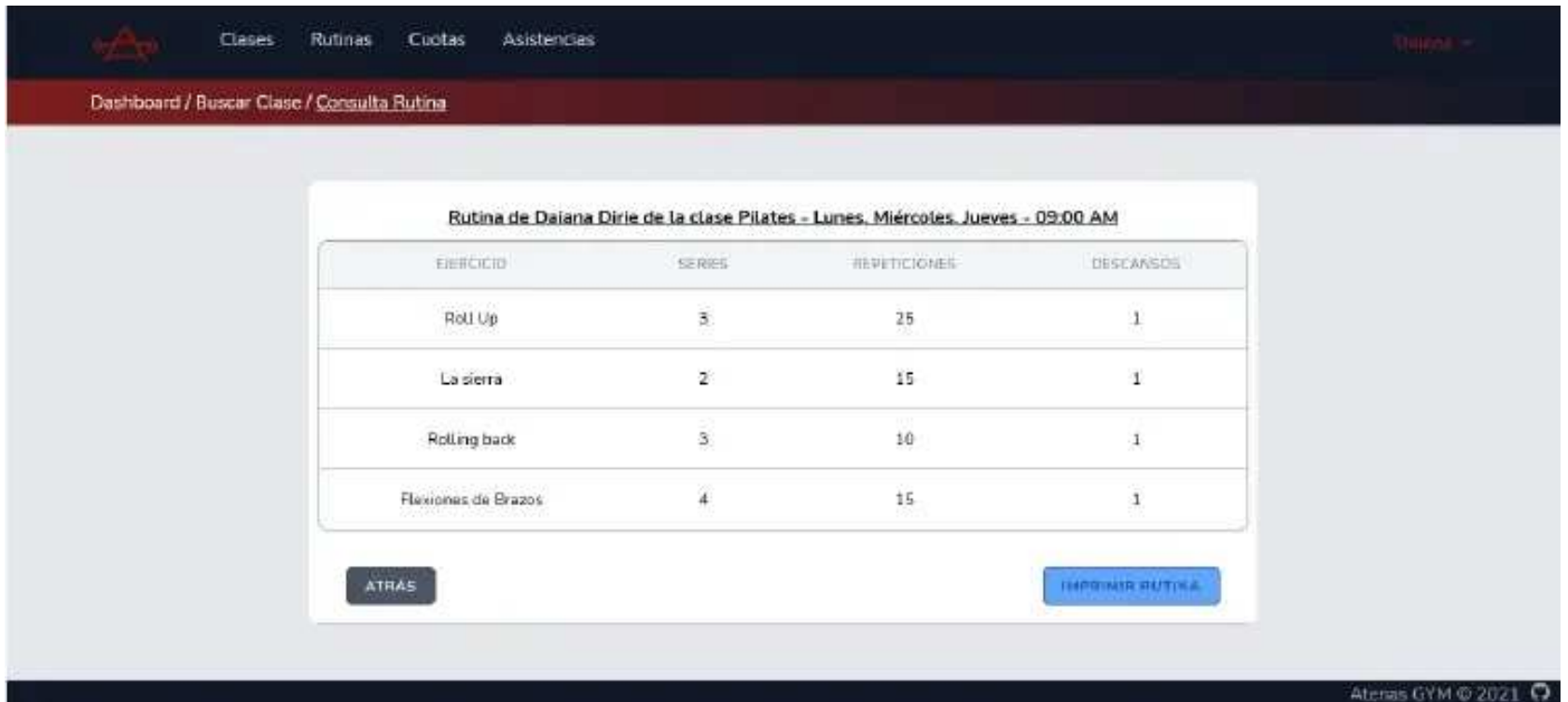
Pantalla 4 (Consulta Clase):



Pantalla 5 (Consulta Rutina – Seleccionar Clase):



Pantalla 6 (Consulta Rutina):



Dashboard / Buscar Clase / Consulta Rutina

Rutina de Daiana Dirie de la clase Pilates - Lunes, Miércoles, Jueves - 09:00 AM

EJERCICIO	SERIES	REPETICIONES	DESCANSOS
Roll Up	3	25	1
La sierra	2	15	1
Rolling back	3	10	1
Flexiones de Brazos	4	15	1

ATRÁS IMPRIMIR RUTINA

Atenas GYM © 2021

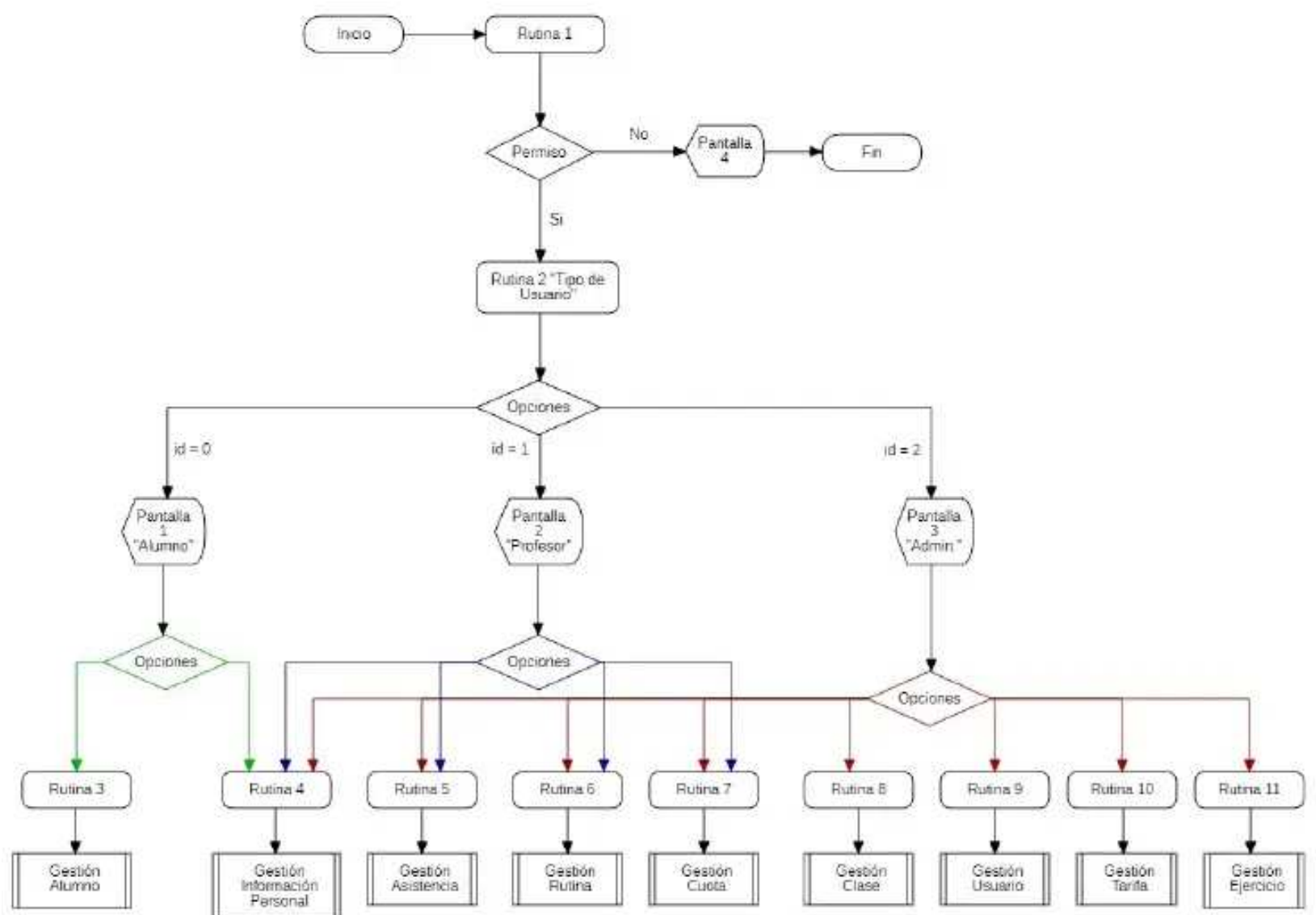
Salida Impresa:

Rutina de Daiana Dirie de la clase Pilates - Lunes, Miércoles, Jueves - 09:00 AM			
EJERCICIO	SERIES	REPETICIONES	DESCANSOS
Roll Up	3	25	1
La sierra	2	15	1
Rolling back	3	10	1
Flexiones de Brazos	4	15	1

Proceso Gestión Dashboard

→ Diagrama FlowChart:

Proceso: Gestión Dashboard



→ Rutinas:

RUTINA 1:

Verifica el permiso del usuario.

RUTINA 2:

Según el tipo de usuario determina qué rol cumple en el sistema otorgando los privilegios necesarios.

RUTINA 3:

Llama al subproceso Gestión Alumno.

RUTINA 4:

Llama al subproceso Gestión Informacion Personal.

RUTINA 5:

Llama al subproceso Gestión Asistencia.

RUTINA 6:

Llama al subproceso Gestión Rutina.

RUTINA 7:

Llama al subproceso Gestión Cuota.

RUTINA 8:

Llama al subproceso Gestión Clase.

RUTINA 9:

Llama al subproceso Gestión Usuario.

RUTINA 10:

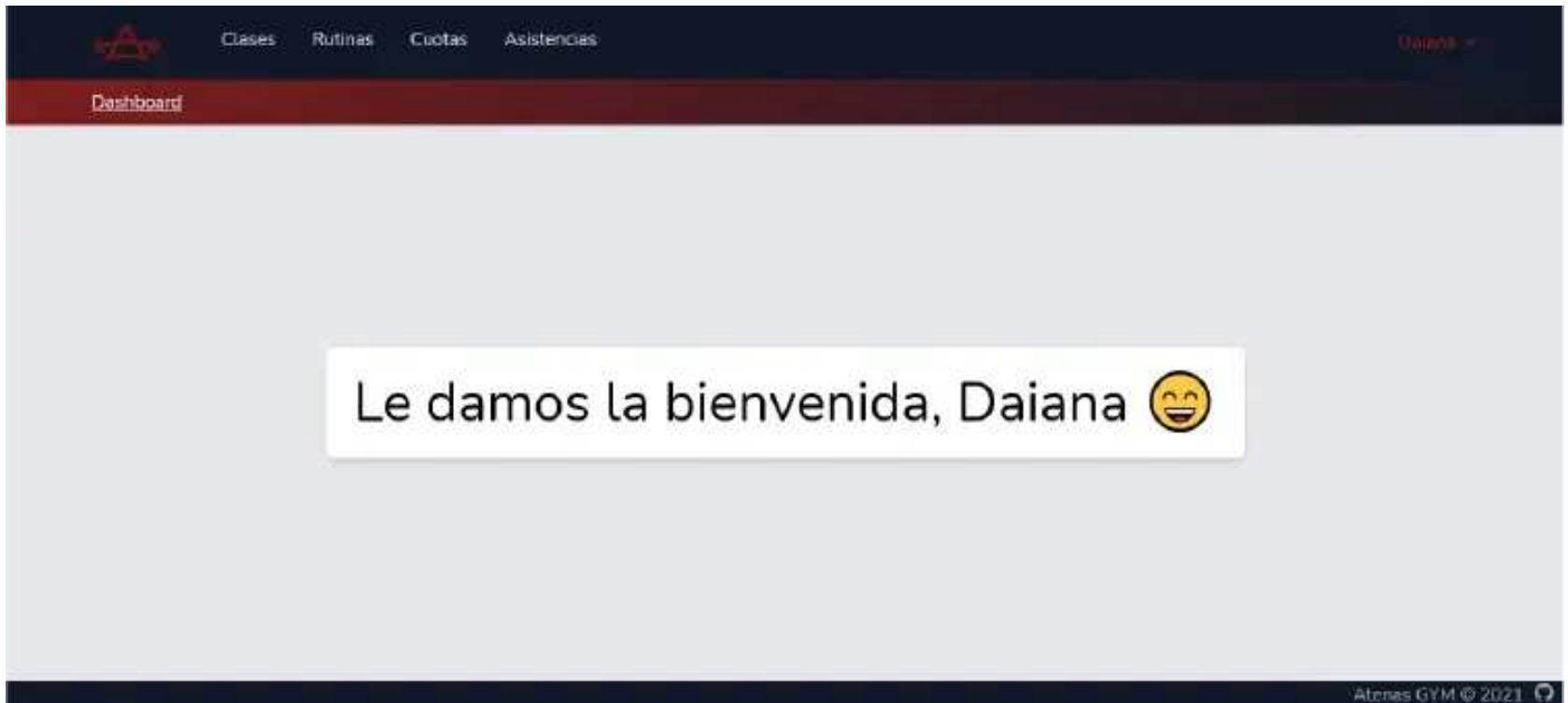
Llama al subproceso Gestión Tarifa.

RUTINA 11:

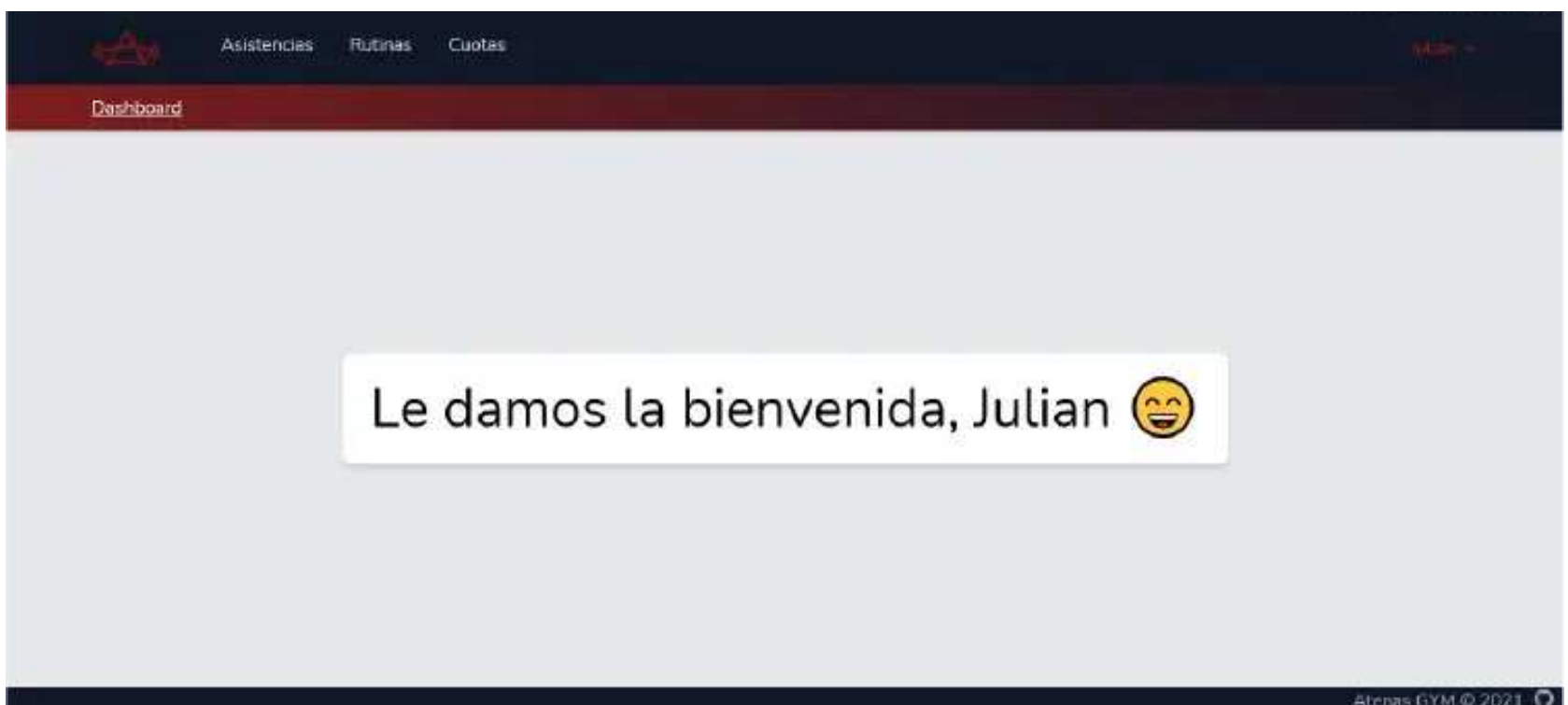
Llama al subproceso Gestión Ejercicio.

→ Pantallas:

Pantalla 1 (Pantalla Alumno):



Pantalla 2 (Pantalla Profesor):



Pantalla 3 (Pantalla Administrador):



Pantalla 4 (Error de validación):

401

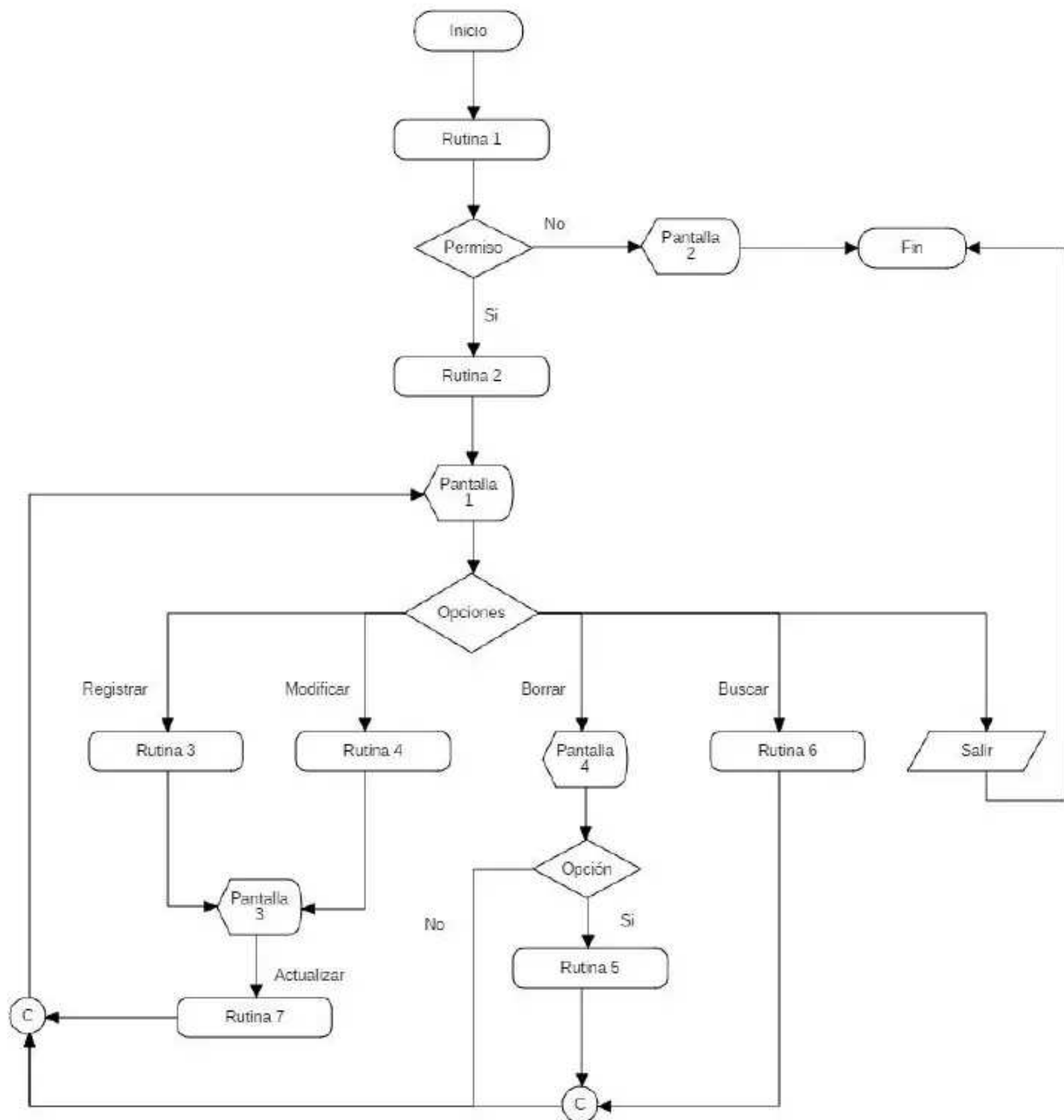
Usted no esta autorizado para
ingresar a esta ruta

IR A DASHBOARD

Proceso Gestión Usuario

→ Diagrama FlowChart:

Proceso: Gestión Usuario



→ Rutinas:

RUTINA 1:

Verifica que el usuario tenga permiso para ingresar a dicho proceso.

RUTINA 2:

Muestra una grilla con los datos de la tabla usuario con sus respectivos botones de acciones.

RUTINA 3:

Muestra un formulario vacío para el ingreso de datos del registro de usuario.

RUTINA 4:

Muestra los campos del formulario con los datos del registro seleccionado para actualizarlos.

RUTINA 5:

Borra y actualiza el registro seleccionado en la tabla y la base de datos

RUTINA 6:

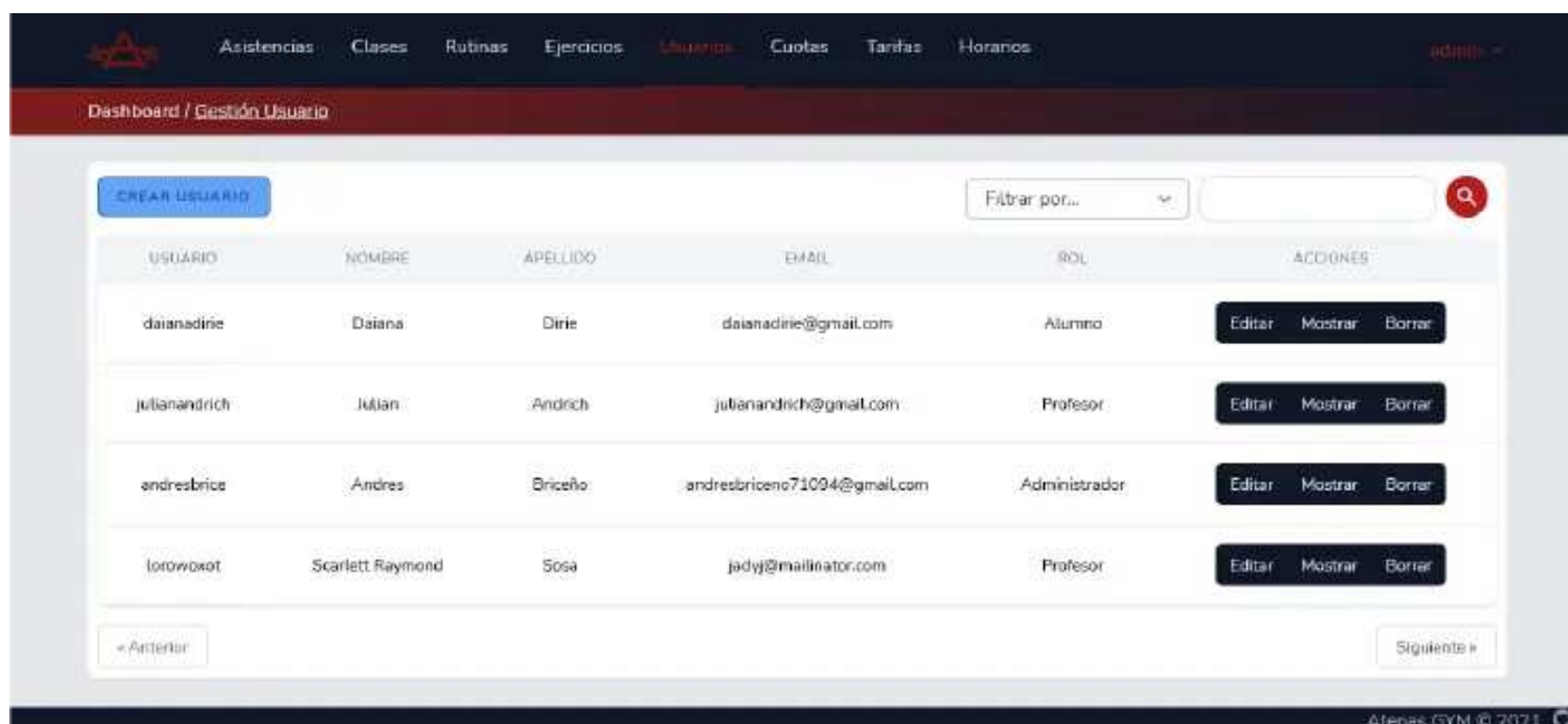
Busca y selecciona un alumno por su nombre.

RUTINA 7:

Graba o actualiza los registros del usuario en la base de datos.

→ Pantallas:

Pantalla 1 (Gestión Usuario):



Pantalla 2 (Error de validación):

401

Usted no esta autorizado para
ingresar a esta ruta

[IR A DASHBOARD](#)

Pantalla 3 (Registro de Usuario):

Atenas

AsistenciasClasesRutinasEjerciciosUsuariosCuotasTarifasHorarios

Atenas GYM

Dashboard / Gestión Usuario / Crear Usuario

Nombre de Usuario

Correo electrónico

DNI

Nombre

Apellido

Género

- Seleccionar Género -

Edad

Teléfono

Teléfono de Emergencia

Rol

- Seleccionar Rol -

Historial Médico

☐ E. Respiratorias

☐ E. Cardíacas

☐ E. Renales

☐ Convulsiones

☐ Epilepsia

☐ Diabetes

☐ Asma

☐ Alergias

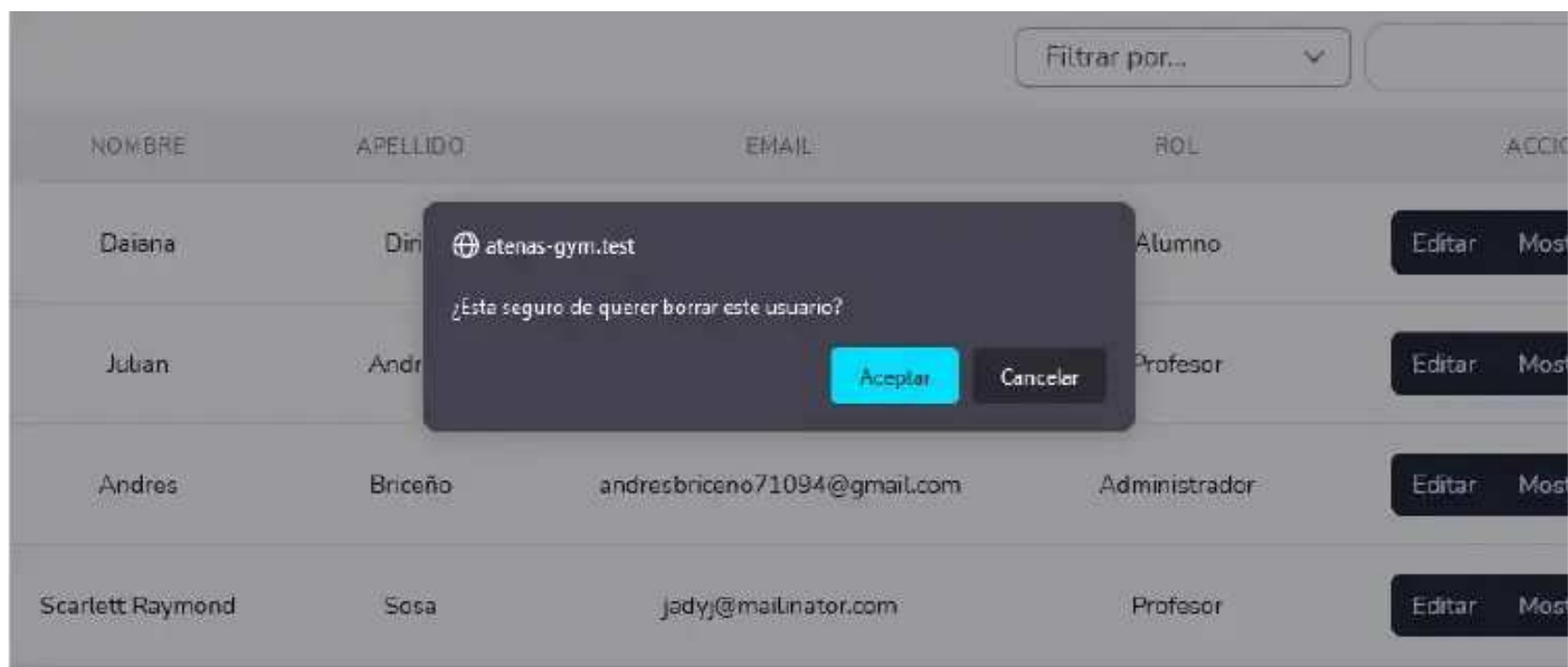
☐ Medicación

ATRÁS

CREAR USUARIO

Atenas GYM © 2021

Pantalla 4 (Advertencia):

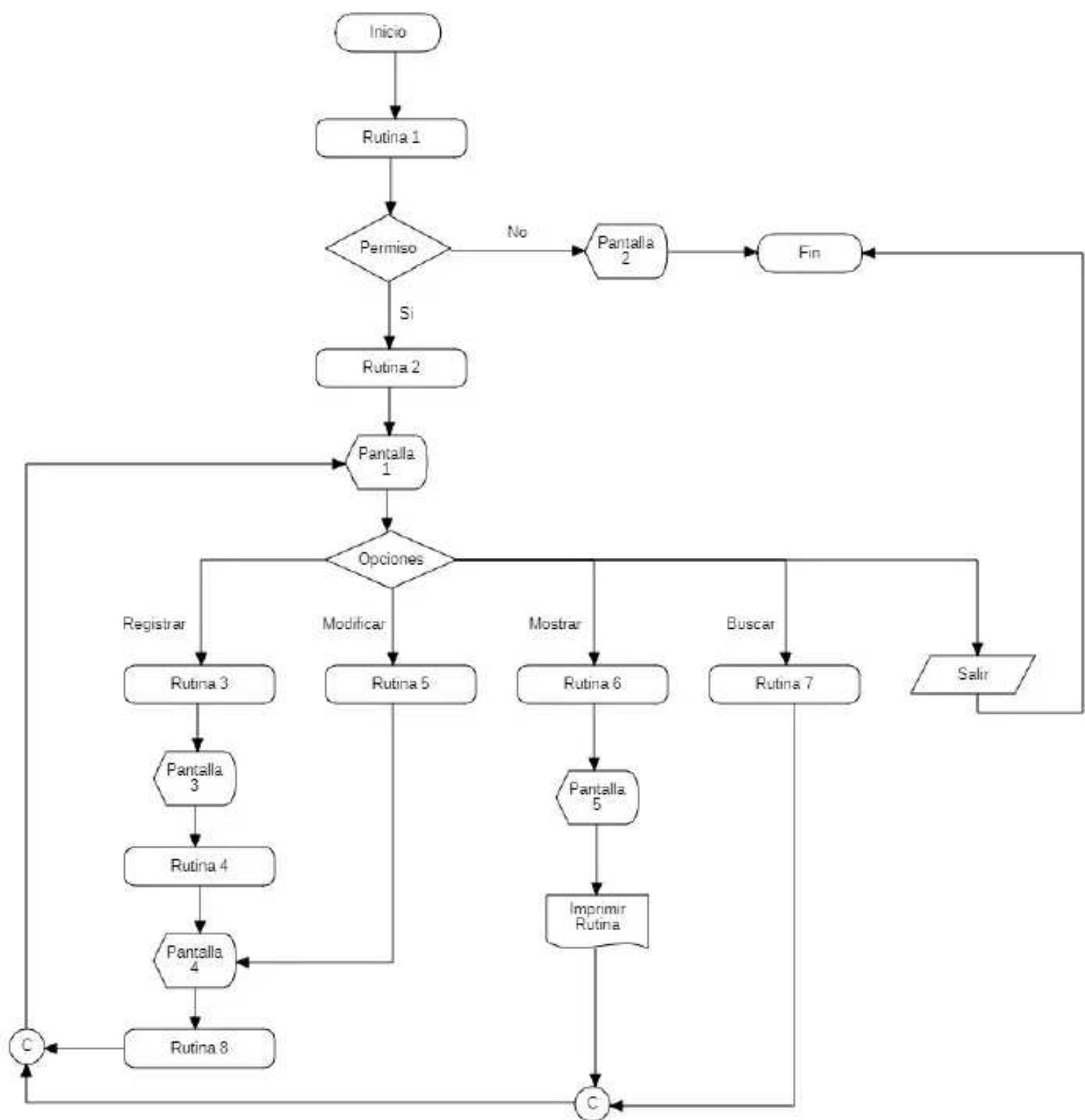


Proceso Gestión Rutina



Diagrama FlowChart:

Proceso: Gestión Rutina



→ Rutinas:

RUTINA 1:

Verifica que el usuario tenga permiso para ingresar a dicho proceso.

RUTINA 2:

Trae los registros de los alumnos de la base de datos, muestra los botones y filtros.

RUTINA 3:

Se despliega una ventana modal, donde se podrá buscar un alumno filtrado por nombre, apellido o usuario, seleccionar el profesor, la fecha de emisión y seleccionar los ejercicios de la rutina.

RUTINA 4:

Muestra los campos del formulario con los datos del registro seleccionado para actualizarlos.

RUTINA 5:

Despliega una pantalla modal con la última rutina emitida según el alumno, con la posibilidad de imprimirla.

RUTINA 6:

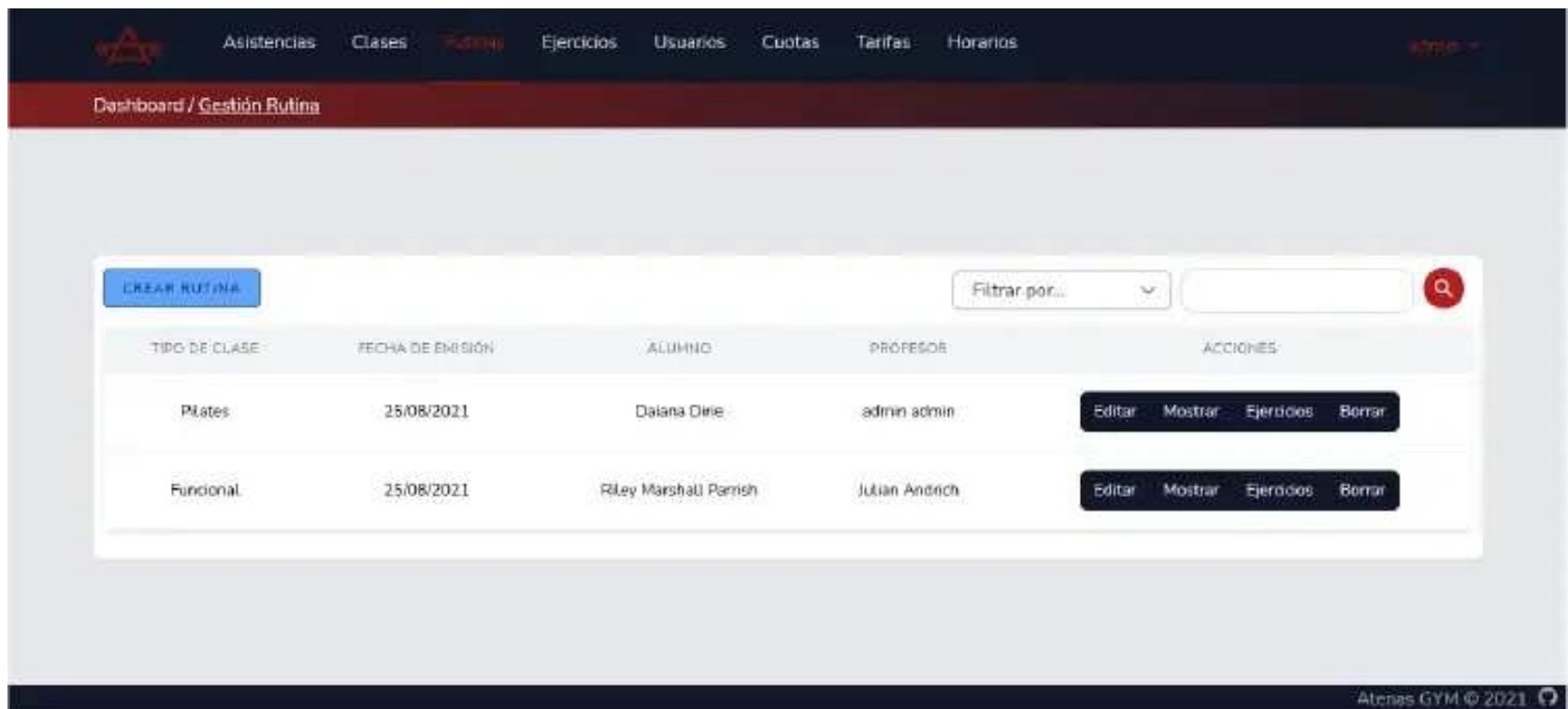
Busca y selecciona un alumno por su nombre.

RUTINA 7:

Graba o actualiza los registros del usuario en la base de datos.

→ Pantallas:

Pantalla 1 (Gestión Rutina):



Pantalla 2 (Error de validación):



Pantalla 3 (Registrar Rutina):

Dashboard / Gestión Rutina / [Crear Rutina](#)

Profesor: admin admin

Alumno: Daiana Dirie - DNI: 98764562 x

Clase: Pilates - 09:00:00 - Lunes, Miércoles, Jueves x

[ATRÁS](#) [CREAR RUTINA](#)

Atenas GYM © 2021

Pantalla 4 (Última Rutina Emitida):

Dashboard / Gestión Rutina / [Mostrar Rutina](#)

Rutina de Daiana Dirie de la clase Pilates - Lunes, Miércoles, Jueves - 09:00 AM

EJERCICIO	SERIES	REPETICIONES	DESCANSOS
Roll Up	3	25	1
La sierra	2	15	1
Rolling back	3	10	1
Flexiones de Brazos	4	15	1

[ATRÁS](#) [IMPRIMIR RUTINA](#)

Atenas GYM © 2021

Salida Impresa

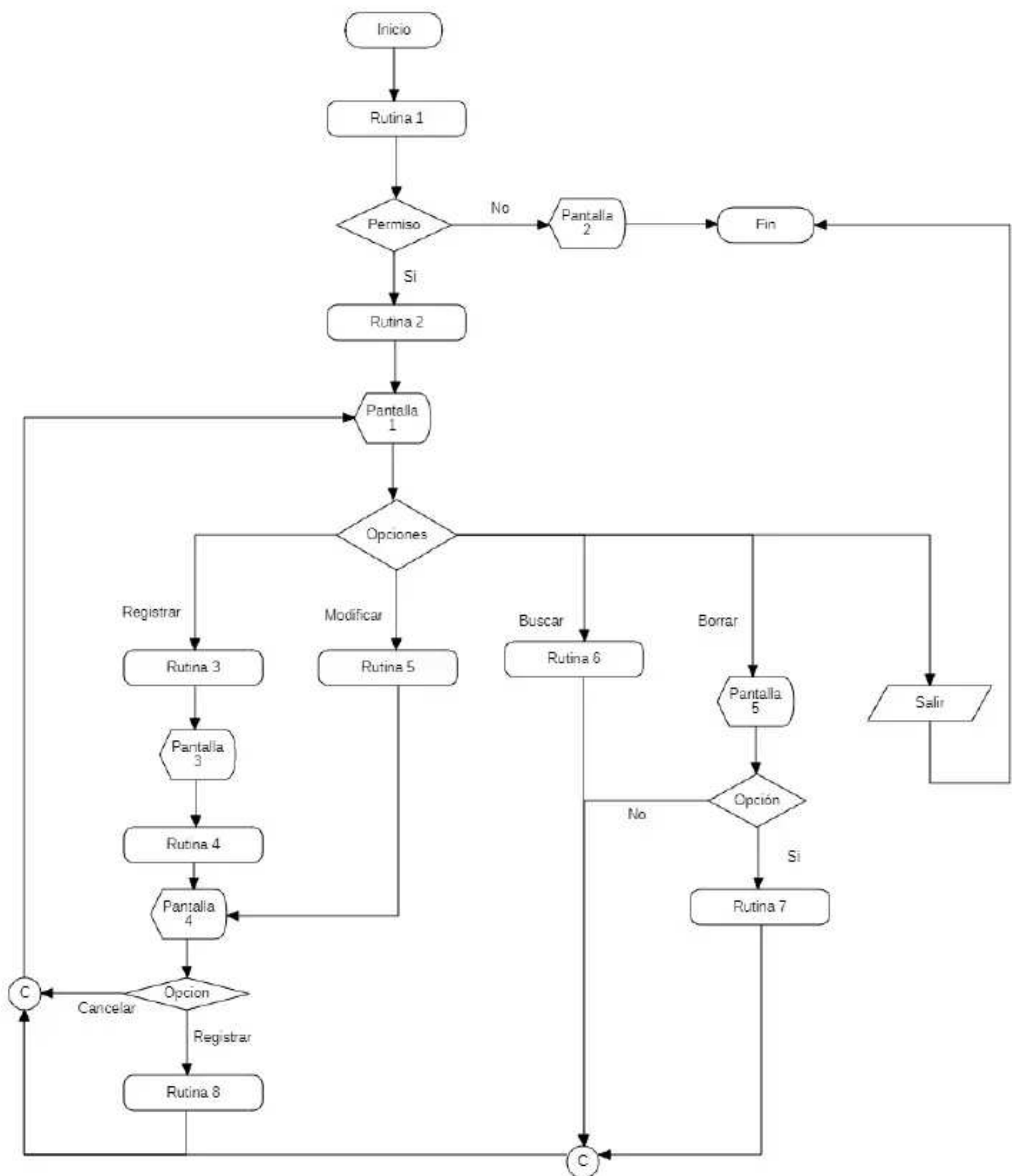
Rutina de Daiana Dirie de la clase Pilates - Lunes, Miércoles, Jueves - 09:00 AM

EJERCICIO	SERIES	REPETICIONES	DESCANSOS
Roll Up	3	25	1
La sierra	2	15	1
Rolling back	3	10	1
Flexiones de Brazos	4	15	1

Proceso Gestión Asistencia

→ Diagrama FlowChart:

Proceso: Gestión Asistencia



→ Rutinas:

RUTINA 1:

Verifica que el usuario tenga permiso para ingresar a dicho proceso.

RUTINA 2:

Muestra en pantalla una grilla con los datos traídos de la base de datos de la tabla Clase. Muestra botones.

RUTINA 3:

Muestra un buscador con filtros para seleccionar la clase correspondiente.

RUTINA 4:

Muestra en pantalla el listado de alumnos filtrados por clase para registrar su asistencia.

RUTINA 5:

Trae los datos de la asistencia de la clase seleccionada y muestra el listado de los alumnos para modificarlos.

RUTINA 6:

Busca y selecciona según el filtro que el usuario haya seleccionado.

RUTINA 7:

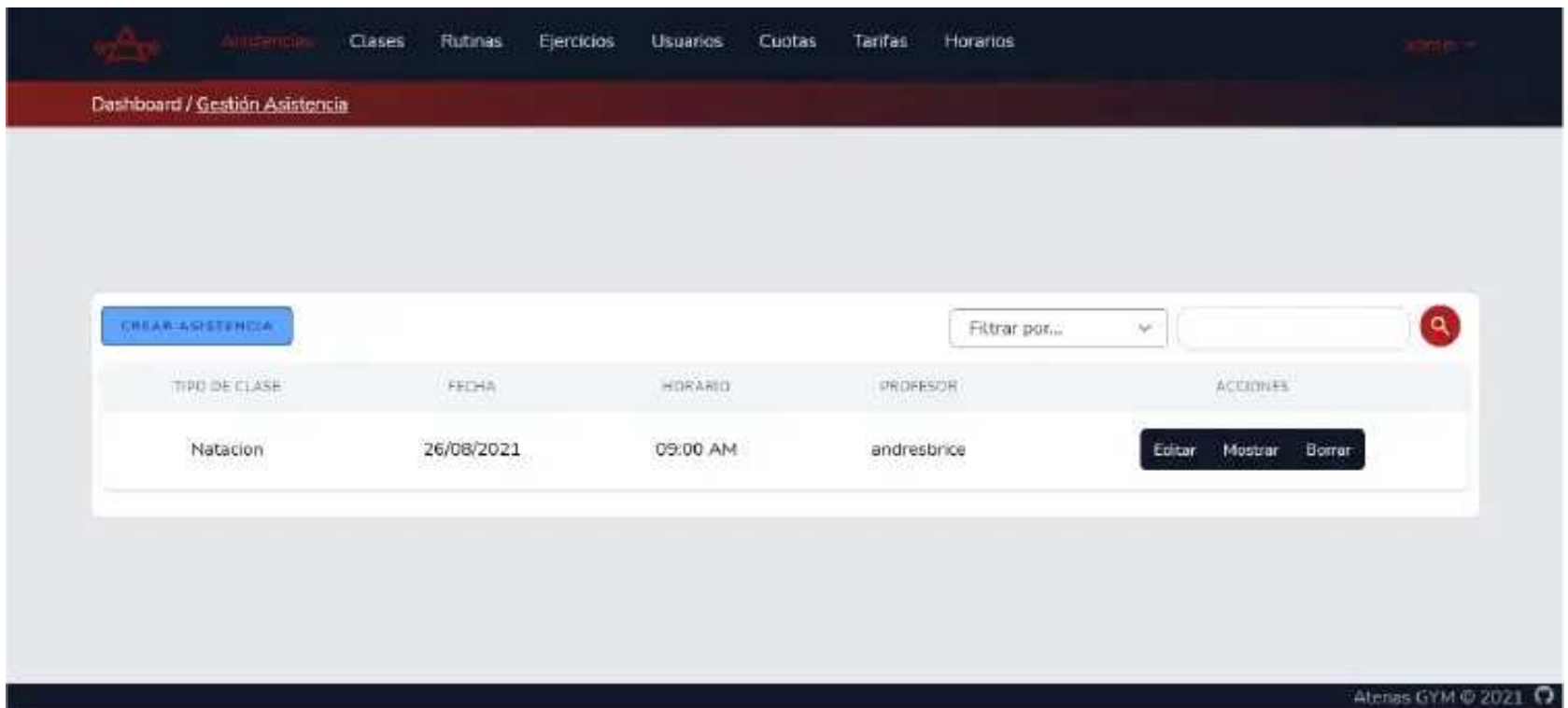
Borra y actualiza el registro seleccionado en la tabla y la base de datos.

RUTINA 8:

Graba o actualiza los registros de la clase en la base de datos.

→ Pantallas:

Pantalla 1 (Gestión Asistencia):



Pantalla 2 (Error de validación):



Pantalla 3 (Buscador):

The screenshot shows the 'Buscar Clase' (Search Class) interface. At the top, there is a navigation bar with the following menu items: **Asistencia**, **Clases**, **Rutinas**, **Ejercicios**, **Usuarios**, **Cuentas**, **Tarifas**, and **Horarios**. Below the navigation bar, a breadcrumb trail reads: **Dashboard / Gestión Asistencia / Buscar Clase**. The main content area features a search form with three input fields: **Fecha** (with a placeholder 'dd/mm/aaaa'), **Horario** (with a placeholder 'Seleccione el horario'), and **Clase** (with a placeholder 'Seleccione la clase'). Below these fields are two buttons: **ATRÁS** (Back) and **BUSCAR CLASE** (Search Class). The footer of the page displays 'Atenas GYM © 2021'.

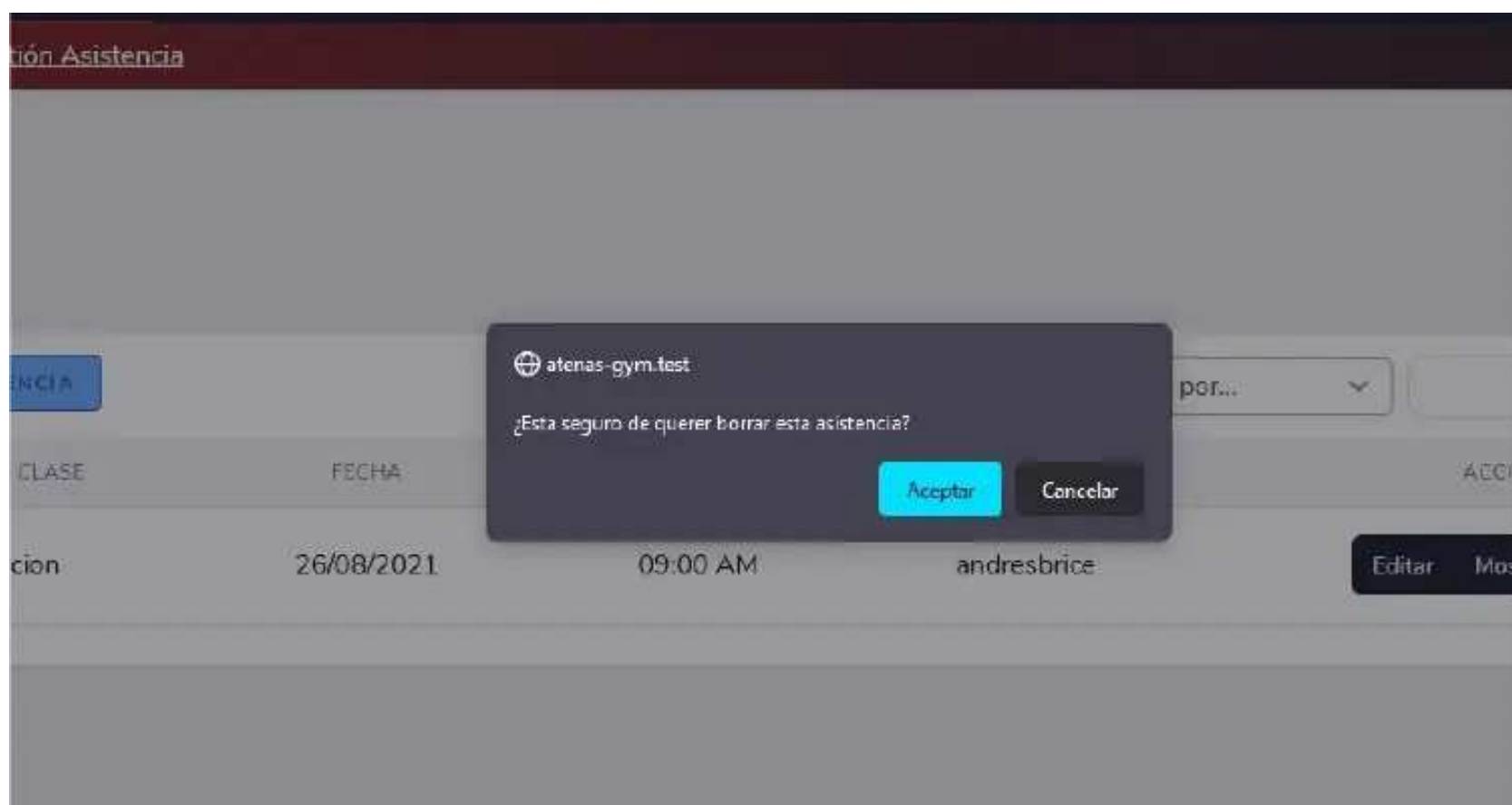
Pantalla 4 (Registrar Asistencia):

The screenshot shows the 'Registrar Asistencia' (Register Attendance) interface. At the top, the navigation bar is identical to the previous screen. The breadcrumb trail is: **Dashboard / Gestión Asistencia / Buscar Clase / Crear Asistencia**. The main content area displays a form for registering attendance. It includes a list of example data:

- Fecha: Ejemplo
- Horario: Ejemplo
- Tipo de Clase: Ejemplo
- Profesor: Ejemplo

Below this list is a table with 8 rows, each representing a student. The table has two columns: the student's name and a checkbox for marking attendance. The students listed are: **Alumno 1**, **Alumno 2**, **Alumno 3**, **Alumno 4**, **Alumno 5**, **Alumno 6**, **Alumno 7**, and **Alumno 8**. At the bottom of the form are two buttons: **ATRÁS** (Back) and **CREAR ASISTENCIA** (Create Attendance). The footer of the page displays 'Atenas GYM © 2021'.

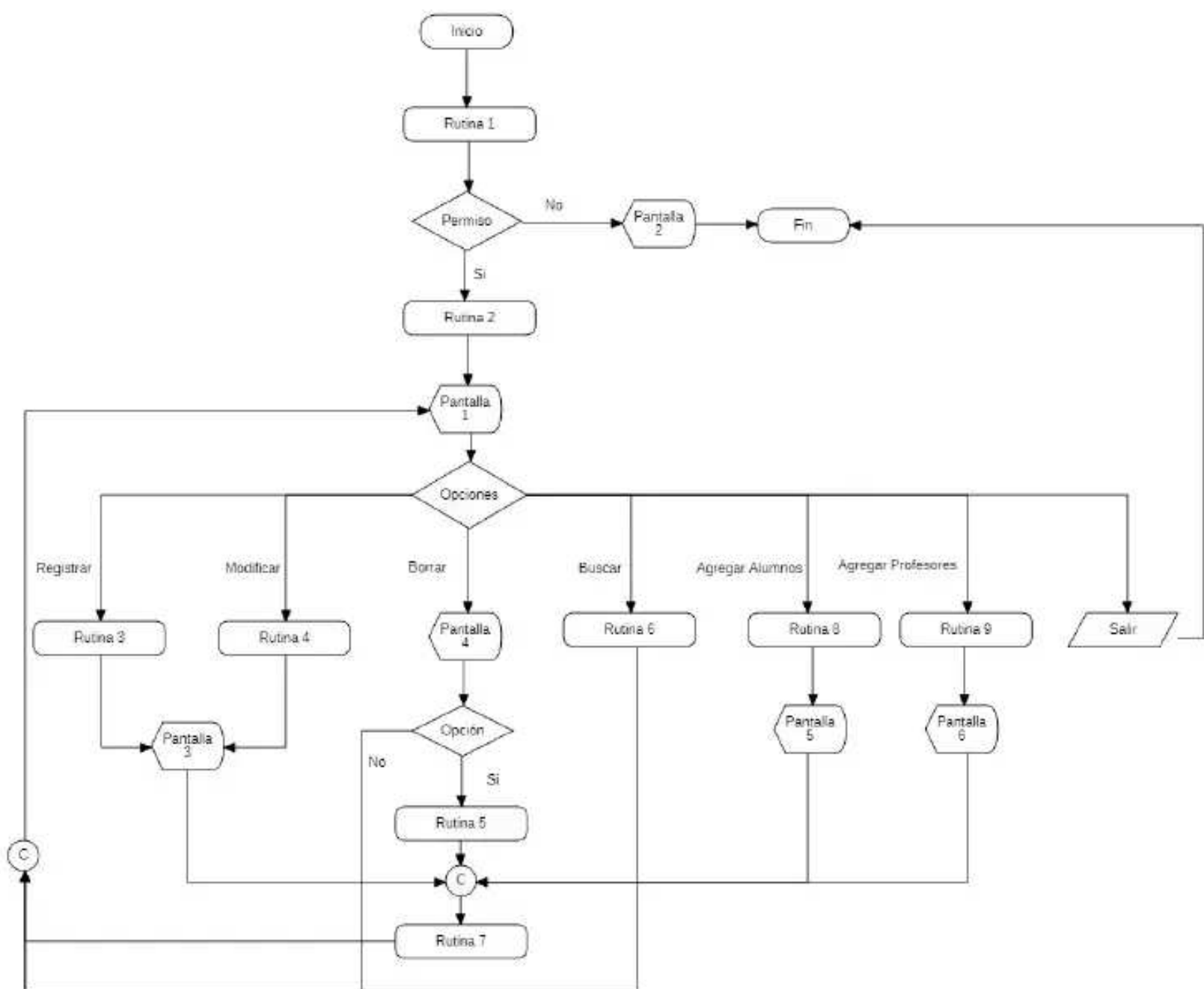
Pantalla 5(Advertencia):



Proceso Gestión Clase

→ Diagrama FlowChart

Proceso: Gestión Clase



→ Rutinas:

RUTINA 1:

Verifica que el usuario tenga permiso para ingresar a dicho proceso.

RUTINA 2:

Trae los datos de la tabla Clase de la base de datos y los muestra en pantalla en una grilla.

RUTINA 3:

Despliega una ventana modal con un formulario donde el usuario podrá, en una caja de texto, escribir el tipo de la nueva clase, el horario, seleccionar el profesor y los días de la semana que se dictara la misma.

RUTINA 4:

Despliega una ventana modal con un formulario con los datos del registro seleccionado para actualizarlos.

RUTINA 5:

Borra el registro seleccionado en la tabla y la base de datos.

RUTINA 6:

Busca y selecciona según el filtro que el usuario haya seleccionado.

RUTINA 7:

Graba o actualiza los registros de la clase en la base de datos.

RUTINA 8:

Muestra un buscador con filtros para seleccionar los alumnos correspondientes.

RUTINA 9:

Muestra un buscador con filtros para seleccionar los profesores correspondientes.

→ Pantallas:

Pantalla 1 (Gestión Clase):

TIPO DE CLASE	CUPOS DISPONIBLES	HORARIO	DIAS DE ENTRENAMIENTO	TARIFA	ACCIONES
Pilates	10	08:00 AM	Lunes, Miércoles, Jueves	\$1200	Editar, Mostrar, Alumnos, Profesores, Borrar
Pilates	8	09:00 AM	Lunes, Miércoles, Jueves	\$1200	Editar, Mostrar, Alumnos, Profesores, Borrar
Natacion	9	09:00 AM	Lunes, Martes	\$900	Editar, Mostrar, Alumnos, Profesores, Borrar
Futbol	9	09:00 AM	Lunes, Martes	\$900	Editar, Mostrar, Alumnos, Profesores, Borrar

Pantalla 2 (Error de Validación):



Pantalla 3 (Registrar Clase):

The screenshot shows the 'Registrar Clase' (Register Class) form in the Atenas GYM system. The form is centered on a light gray background. It contains the following fields and controls:

- Tipo de Clase:** A text input field.
- Horario:** A dropdown menu with the placeholder text '- Seleccionar Horario -'.
- Días de la semana:** A group of five radio buttons labeled 'Lunes', 'Martes', 'Miércoles', 'Jueves', and 'Viernes'.
- Buttons:** A dark gray 'ATRÁS' (Back) button and a blue 'CREAR CLASE' (Create Class) button.

The top navigation bar includes links for 'Asistencias', 'Clases', 'Rutinas', 'Ejercicios', 'Usuarios', 'Cuotas', 'Tarifas', and 'Horarios'. The breadcrumb trail at the top reads 'Dashboard / Gestión Clase / Crear Clase'. The footer indicates 'Atenas GYM © 2021'.

Pantalla 4 (Advertencia):

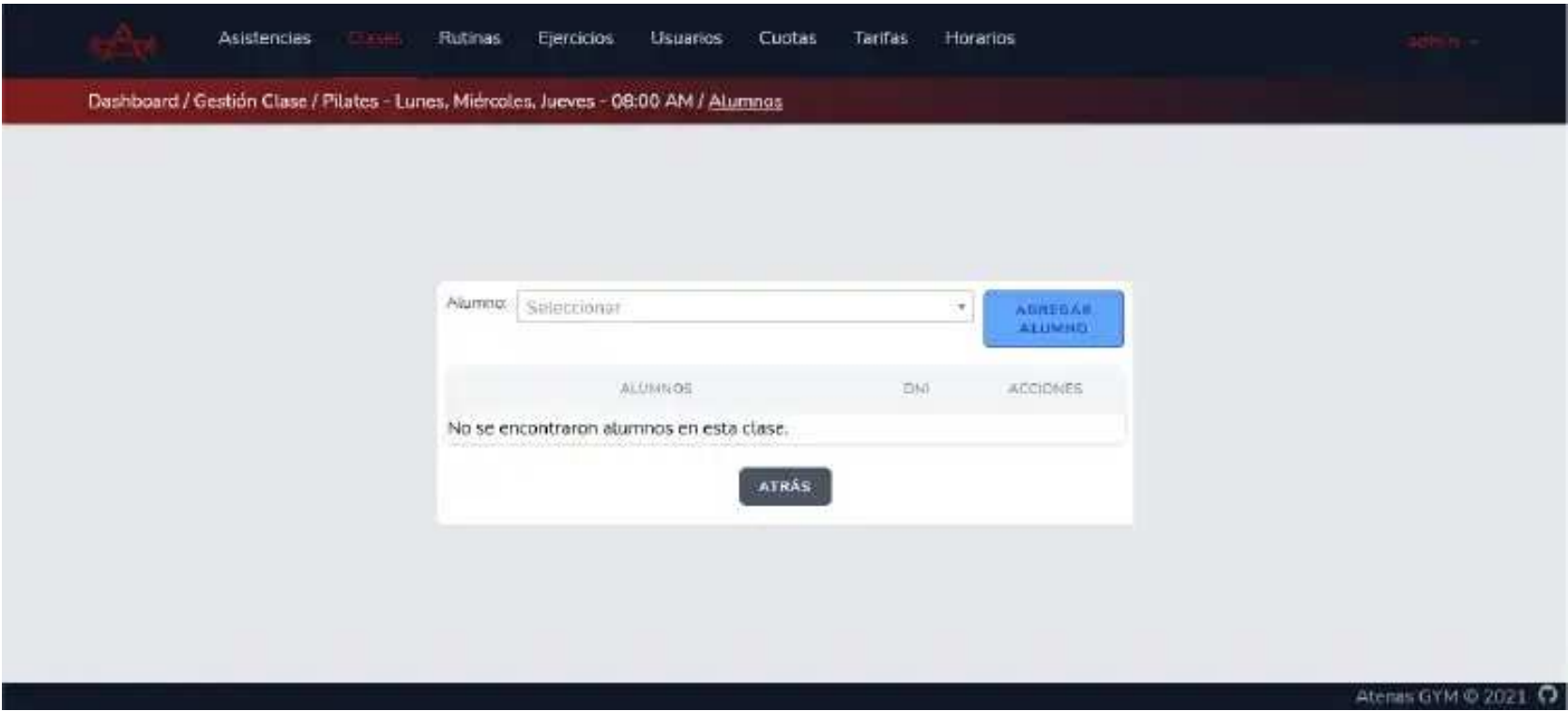
The screenshot shows a warning dialog box overlaid on a table of classes. The dialog box is dark gray with a white border and contains the following text:

- Logo:** A globe icon followed by the text 'atenas-gym.test'.
- Message:** '¿Esta seguro de querer borrar esta clase?' (Are you sure you want to delete this class?).
- Buttons:** A red 'Aceptar' (Accept) button and a gray 'Cancelar' (Cancel) button.

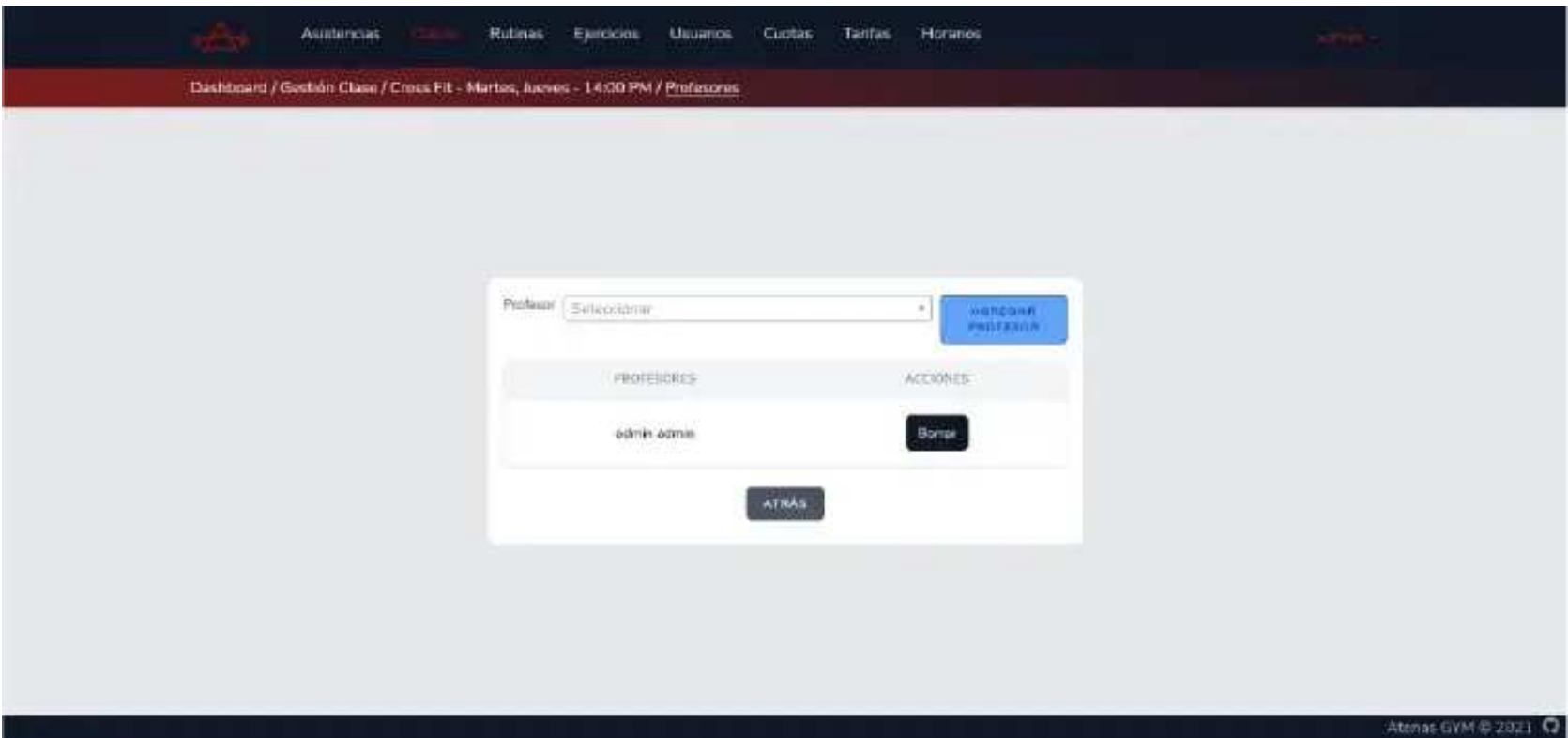
The background table displays class information with the following columns: 'DISPONIBLES', 'HORARIO', 'DIAS DE ENTRENAMIENTO', 'TARIFA', and 'ACCIONES'. The visible rows are:

DISPONIBLES	HORARIO	DIAS DE ENTRENAMIENTO	TARIFA	ACCIONES
10	08:00 A			Editar, Mostrar, Alumnos, P
8	09:00 A			Editar, Mostrar, Alumnos, P
9	09:00 AM	Lunes, Martes	\$900	Editar, Mostrar, Alumnos, P
9	09:00 AM	Lunes, Martes	\$900	Editar, Mostrar, Alumnos, P

Pantalla 5 (Agregar Alumnos):



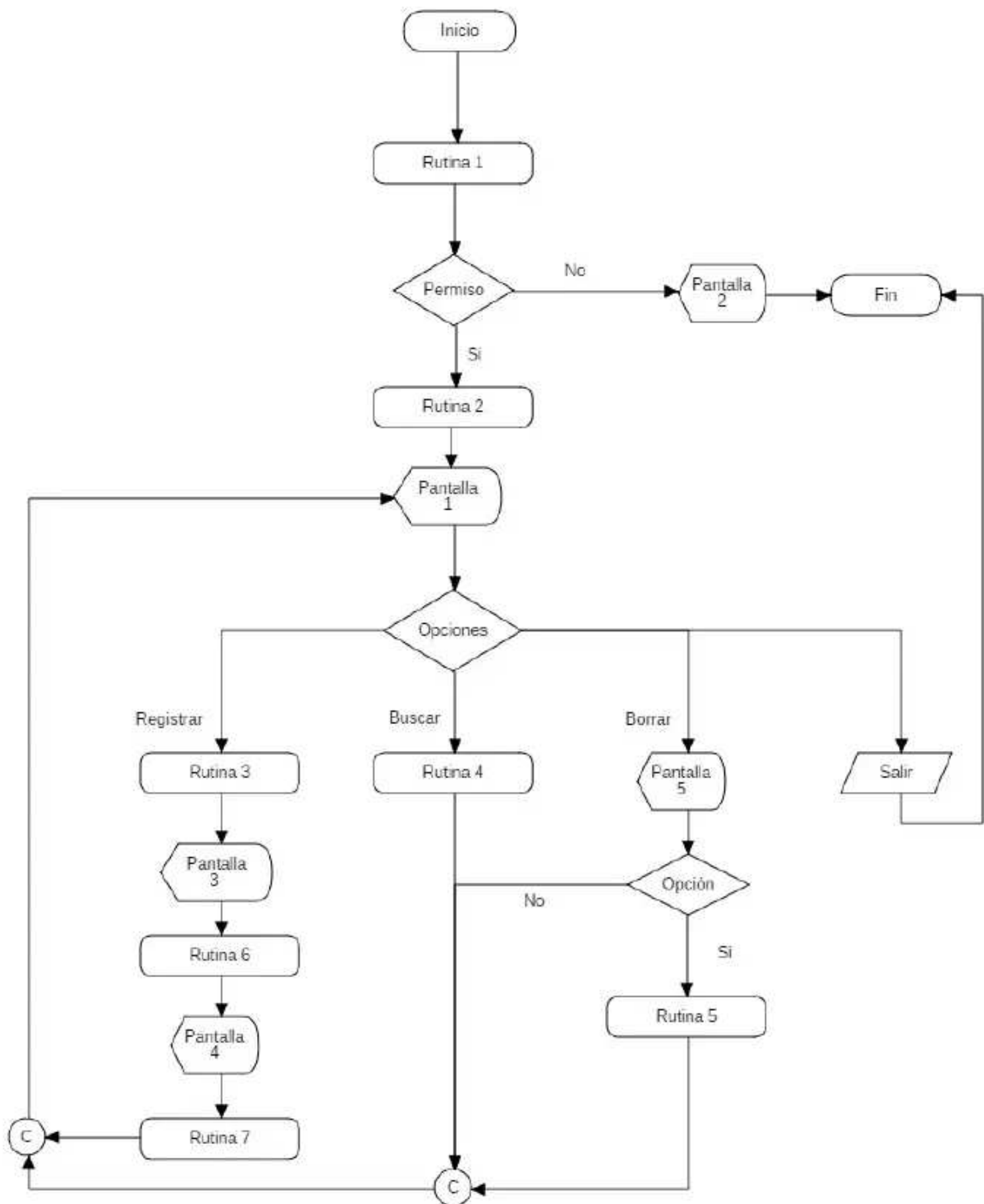
Pantalla 6 (Agregar Profesores):



Proceso Gestión Cuota

→ Diagrama FlowChart:

Proceso: Gestión Cuota



→ Rutinas:

RUTINA 1:

Verifica que el usuario tenga permiso para ingresar a dicho proceso.

RUTINA 2:

Muestra en pantalla una grilla con los datos traídos de la base de datos con las cuotas que ya fueron registradas. También muestra botones y filtro por nombre de usuario, nombre o apellido.

RUTINA 3:

Se muestra en pantalla una grilla con los datos de los alumnos inscriptos, junto a botón para registrar la cuota y filtro por nombre de usuario, nombre o apellido.

RUTINA 4:

Busca un alumno por su usuario, nombre o apellido.

.

RUTINA 5:

Borra y actualiza el registro seleccionado en la tabla y la base de datos.

RUTINA 6:

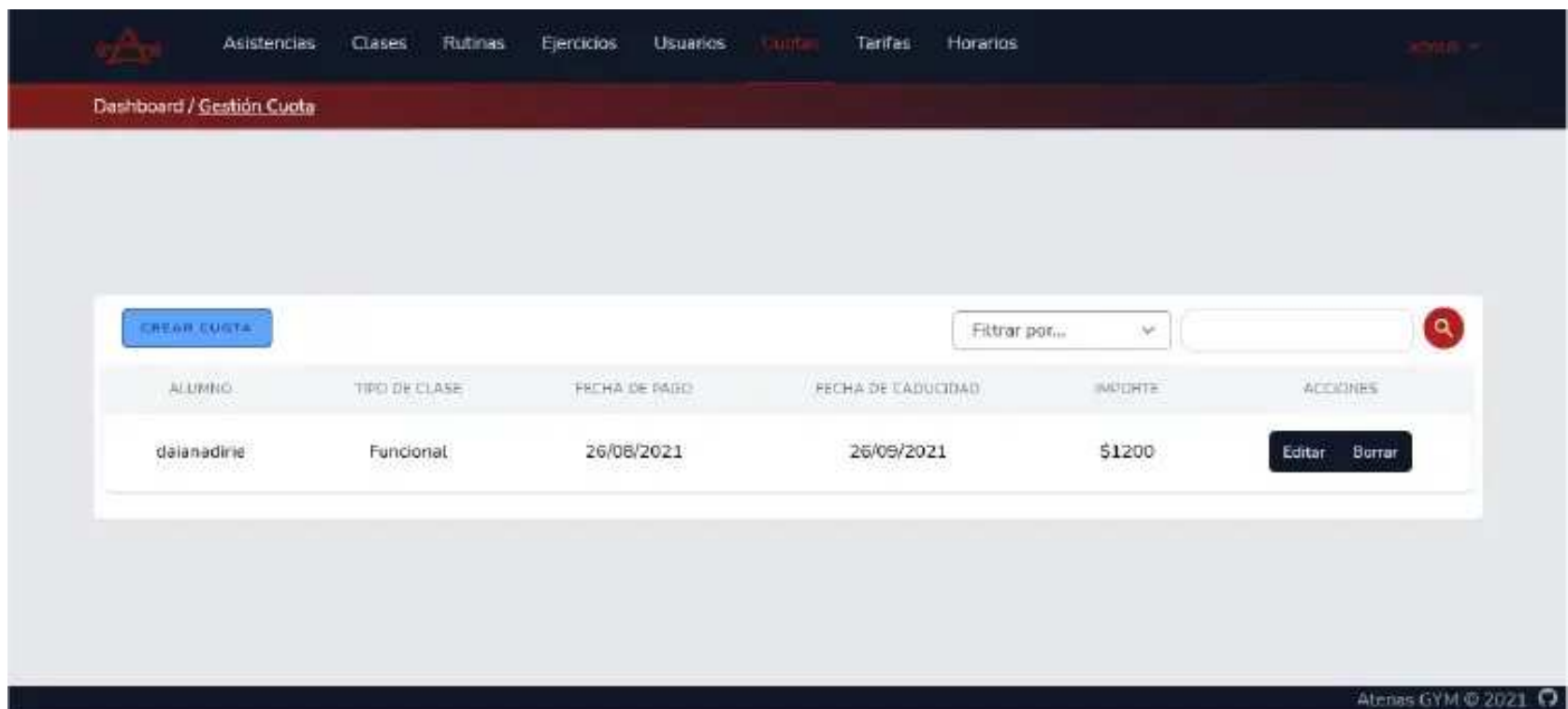
Abre una pantalla modal con los datos extraídos de la base de datos según el alumno que abone la cuota

RUTINA 7:

Graba los registros de la cuota en la base de datos.

→ Pantallas:

Pantalla 1 (Gestión Cuota):



Pantalla 2 (Error de Validación):



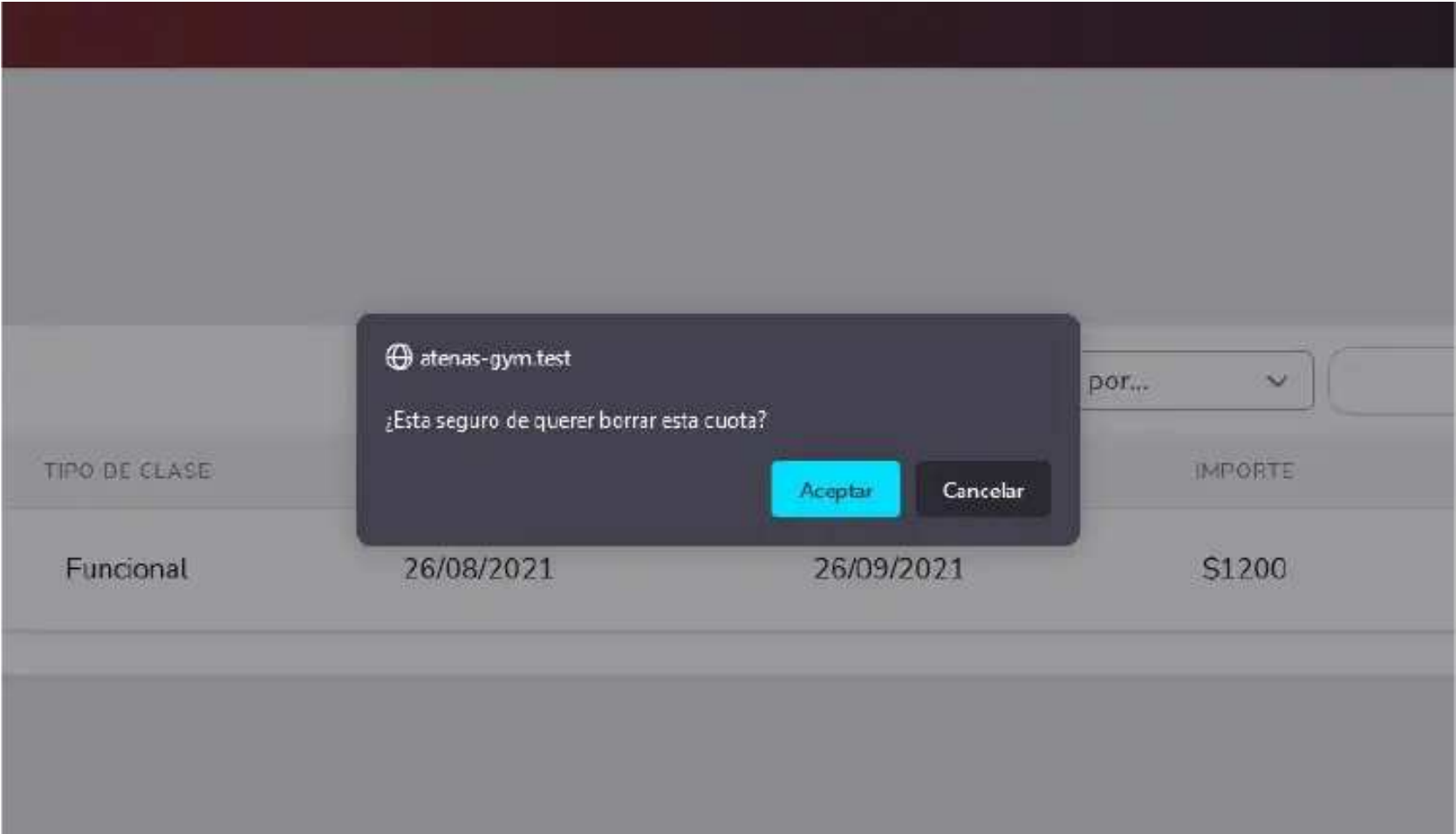
Pantalla 3 (Registro de Cuota):

The screenshot shows a web application interface for 'Atenas GYM'. The top navigation bar includes links for Asistencias, Clases, Rutinas, Ejercicios, Usuarios, Cuotas, Tarifas, and Horarios. The breadcrumb trail indicates the current path: Dashboard / Gestión Cuota / Seleccionar Alumno. The main content area features a form with two dropdown menus: 'Alumno' with the placeholder 'Seleccione un alumno' and 'Tipo de Clase' with the placeholder 'Seleccione la clase'. Below these are two buttons: 'ATRÁS' and 'SELECCIONAR ALUMNO'. The footer displays 'Atenas GYM © 2021'.

Pantalla 4 (Registrar Cuota):

The screenshot shows the 'Registrar Cuota' screen in the 'Atenas GYM' application. The breadcrumb trail is: Dashboard / Gestión Cuota / Seleccionar Alumno / Crear Cuota. The form contains four input fields: 'Alumno', 'Tipo de Clase', 'Importe', and 'Fecha de Pago' (with a date format hint 'dd / mm / aaaa'). At the bottom of the form are two buttons: 'ATRÁS' and 'CREAR CUOTA'. The footer shows 'Atenas GYM © 2021'.

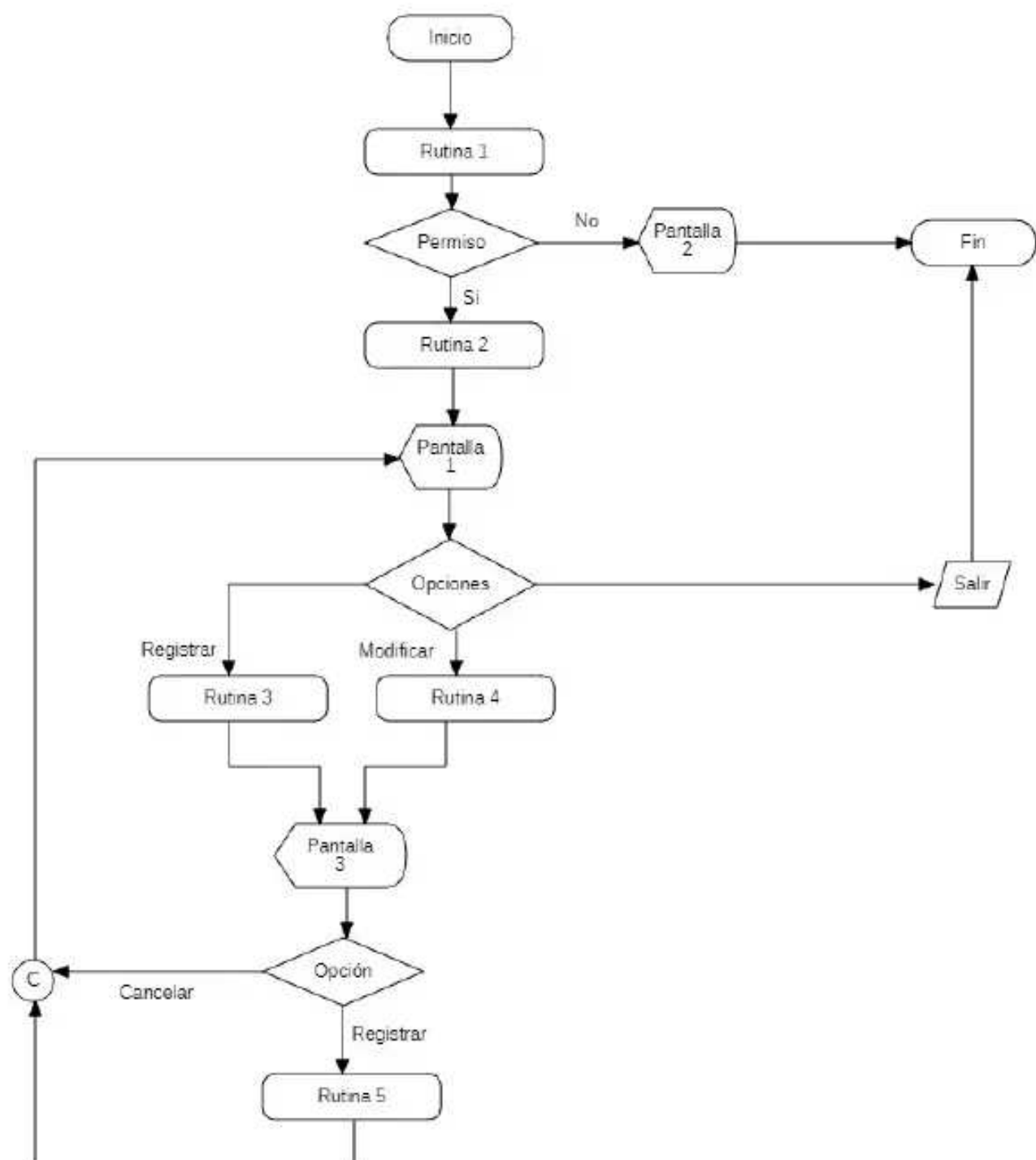
Pantalla 5 (Advertencia):



Proceso Gestión Tarifa

→ Diagrama Flowchart:

Proceso: Gestion Tarifa



→ Rutina:

RUTINA 1

Verifica que el usuario tenga permiso para ingresar a dicho proceso.

RUTINA 2

Muestra en pantalla una grilla con las tarifas registradas hasta el momento con su respectivo botón para modificarlas y un botón para registrar nuevas tarifas. Cuando la grilla posea cinco registros se deshabilita el botón Registrar Tarifa.

RUTINA 3

Se muestra en pantalla una ventana modal con un formulario para indicar la cantidad de días y el precio de la tarifa.

RUTINA 4

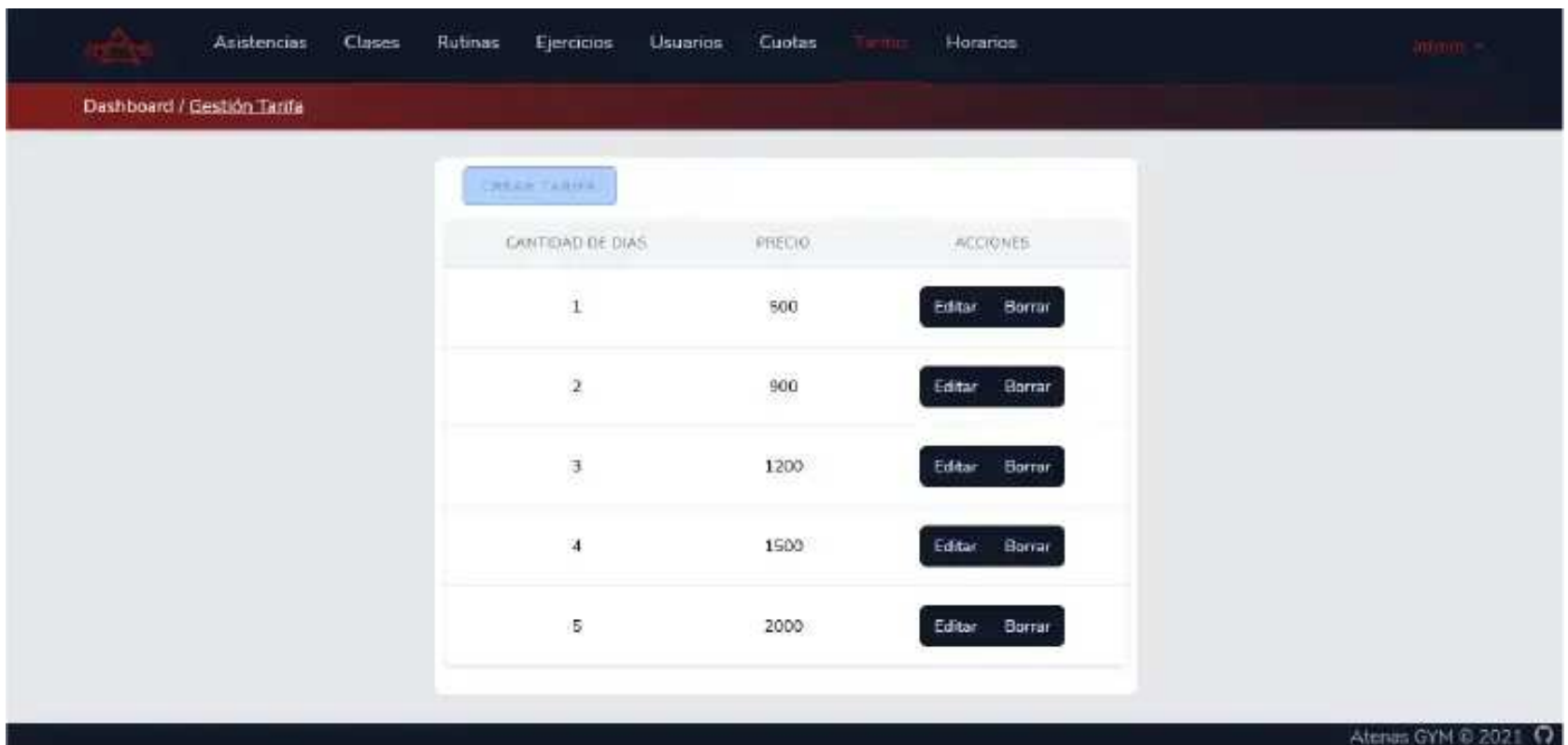
Despliega una ventana modal con un formulario con los datos del registro seleccionado para actualizarlos.

RUTINA 5

Graba los registros de la cuota en la base de datos.

→ Pantallas:

Pantalla 1 (Gestión Tarifa):



CANTIDAD DE DIAS	PRECIO	ACCIONES
1	800	Editar Borrar
2	900	Editar Borrar
3	1200	Editar Borrar
4	1500	Editar Borrar
5	2000	Editar Borrar

Pantalla 2 (Error de validación):



Pantalla 3 (Modificar Tarifa):

Atenas GYM

Asistencias Clases Rutinas Ejercicios Usuarios Cuotas **Tarifa** Horarios [Admin](#)

Dashboard / Gestión Tarifa / [Editar Tarifa](#)

Cantidad de Dias

1

Precio

500

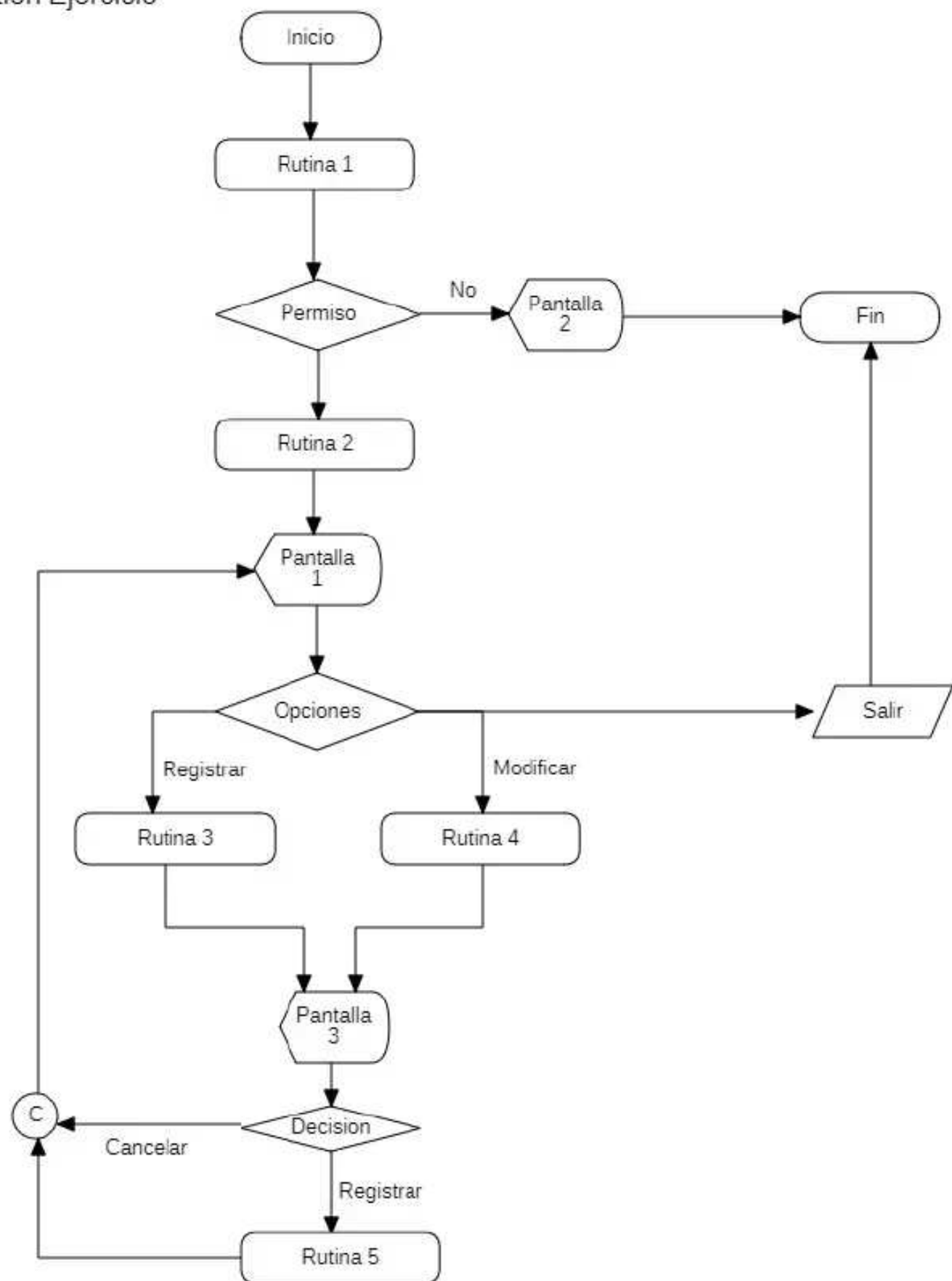
ATRÁS EDITAR TARIFA

Atenas GYM © 2021

Proceso Gestión Ejercicio

→ Diagrama Flowchart:

Proceso Gestion Ejercicio



→ Rutinas

RUTINA 1

Verifica que el usuario tenga permiso para ingresar a dicho proceso.

RUTINA 2

Muestra en pantalla una grilla con los ejercicios registrados hasta el momento con un filtro por nombre o descripción con su respectivo botón para modificarlos y un botón para registrar nuevos ejercicios.

RUTINA 3

Se muestra en pantalla una ventana modal con un formulario para registrar nuevos ejercicios.

RUTINA 4

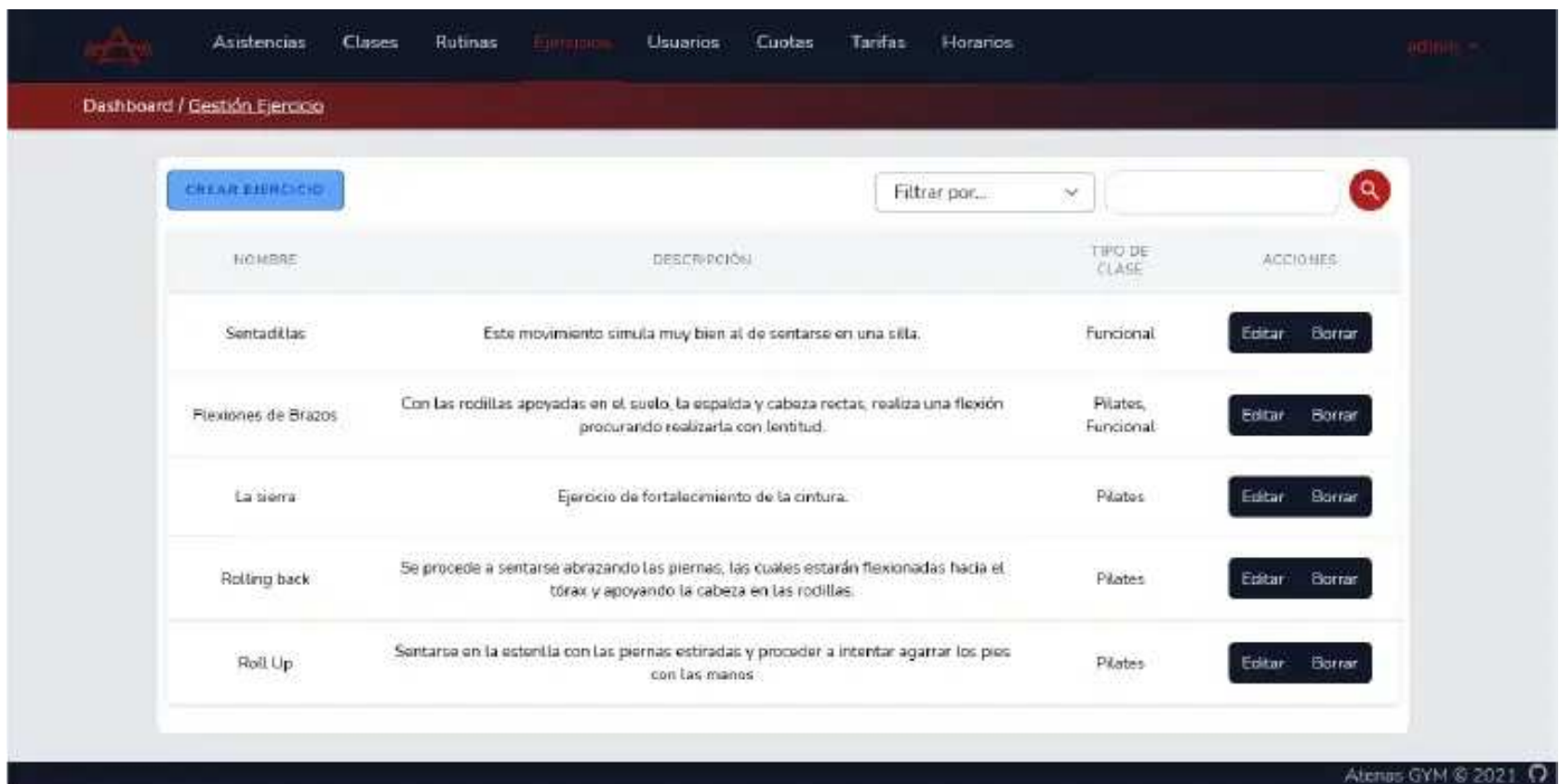
Despliega una ventana modal con un formulario con los datos del registro seleccionado para actualizarlos.

RUTINA 5

Graba los registros de la cuota en la base de datos.

→ Pantallas

Pantalla 1 (Gestión Ejercicio):



NOMBRE	DESCRIPCIÓN	TIPO DE CLASE	ACCIONES
Sentadillas	Este movimiento simula muy bien al de sentarse en una silla.	Funcional	Editar Borrar
Flexiones de Brazos	Con las rodillas apoyadas en el suelo, la espalda y cabeza rectas, realiza una flexión procurando realizarla con lentitud.	Pilates, Funcional	Editar Borrar
La sierra	Ejercicio de fortalecimiento de la cintura.	Pilates	Editar Borrar
Rolling back	Se procede a sentarse abrazando las piernas, las cuales estarán flexionadas hacia el tórax y apoyando la cabeza en las rodillas.	Pilates	Editar Borrar
Roll Up	Sentarse en la estera con las piernas estiradas y proceder a intentar agarrar los pies con las manos.	Pilates	Editar Borrar

Pantalla 2 (Error de validación):

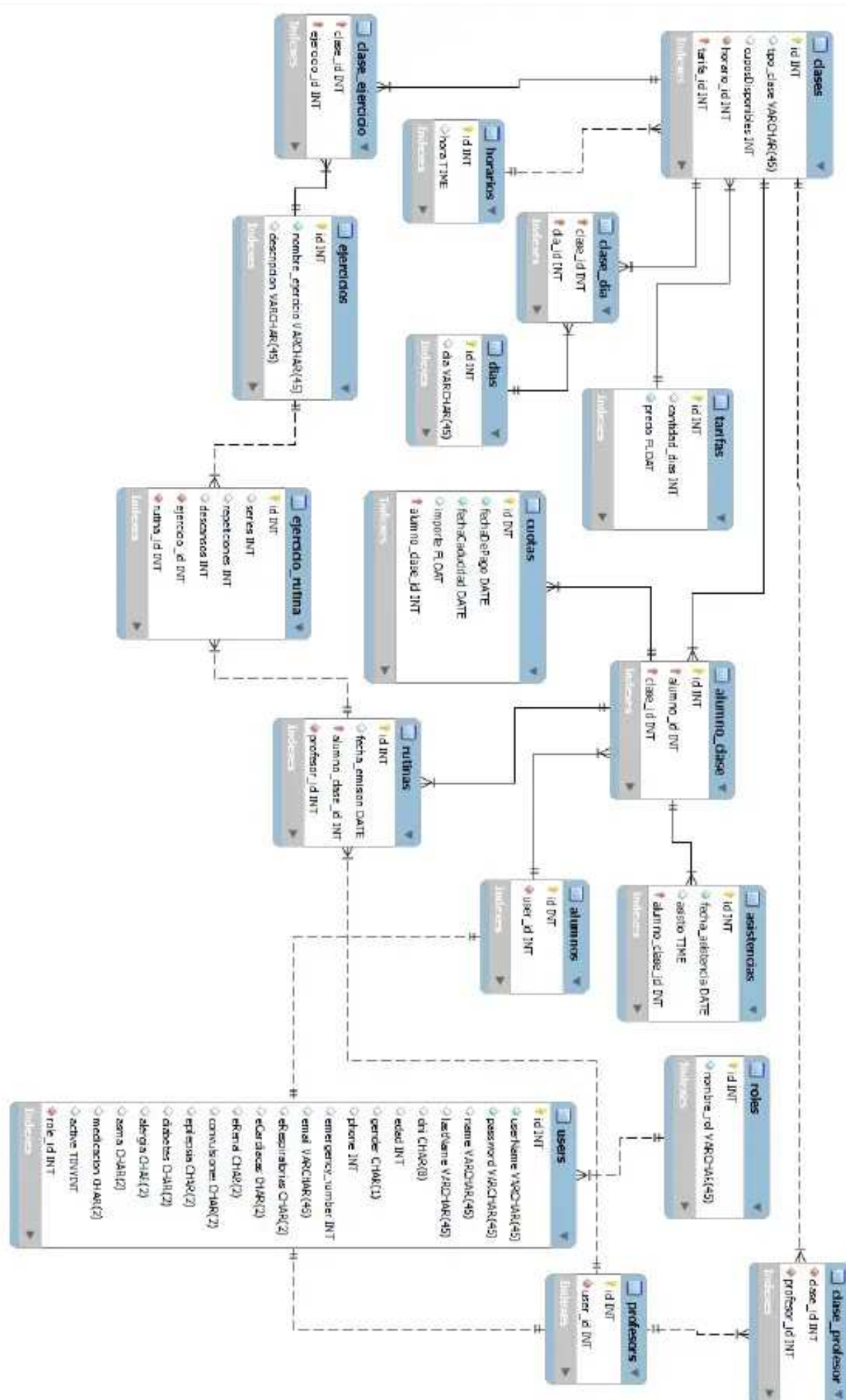
401

Usted no está autorizado para
ingresar a esta ruta

[IR A DASHBOARD](#)

Pantalla 3 (Registrar Ejercicio):

The screenshot displays the 'Registrar Ejercicio' (Register Exercise) interface. At the top, a dark navigation bar contains the Aterias GYM logo and menu items: Asistencias, Clases, Rutinas, Ejercicios (highlighted in red), Usuarios, Cuotas, Tarifas, and Horarios. A user profile 'Admin' is visible on the right. Below the navigation bar, a breadcrumb trail reads 'Dashboard / Gestión Ejercicio / Crear Ejercicio'. The main content area features a central white form with the following fields: 'Tipo de Clase' with a dropdown menu showing 'Seleccionar', 'Ejercicio' with a text input field, and 'Descripción' with a larger text area. At the bottom of the form are two buttons: 'ATRÁS' (Back) and 'CREAR EJERCICIO' (Create Exercise). The footer of the page indicates 'Aterias GYM © 2021' with a small circular icon.



❖ Diccionario de Datos:

Tabla: Usuario

Campo	Tipo	Tamaño	Descripción
id	INT	45	Clave primaria de usuario
userName	Varchar	45	Nombre de usuario
password	Varchar	45	Contraseña del usuario
name	Varchar	45	Nombre del usuario
lastName	INT		Apellido del usuario
dni	CHAR	2	DNI del usuario
edad	INT		Edad del usuario
gender	CHAR	2	Género del usuario
phone	INT		Teléfono del usuario
emergency_number	INT		Teléfono de emergencia del usuario
email	Varchar	45	Email del usuario
eRespiratorias	CHAR	2	Ficha médica
eCardiacas	CHAR	2	Ficha médica
eRenal	CHAR	2	Ficha médica
convulsiones	CHAR	2	Ficha médica
epilepsia	CHAR	2	Ficha médica
diabetes	CHAR	2	Ficha médica
alergia	CHAR	2	Ficha médica
asma	CHAR	2	Ficha médica
medicacion	CHAR	2	Ficha médica
active	TINYINT		Estado del usuario
role_id	INT		Clave foránea de la tabla rol

Tabla: Rol

Campo	Tipo	Tamaño	Descripción
id	INT		Clave primaria de rol
nombre_rol	Varchar	45	Nombre del rol

Tabla: Alumno

Campo	Tipo	Tamaño	Descripción
id	INT		Clave primaria de alumno
user_id	INT		Clave foránea de tabla usuario

Tabla: Profesor

Campo	Tipo	Tamaño	Descripción
id	INT		Clave primaria de profesor
user_id	INT		Clave foránea de tabla usuario

Tabla: Cuota

Campo	Tipo	Tamaño	Descripción
id	INT		Clave primaria
fecha_de_pago	Date		Fecha de pago
fecha_caducidad	Date		Fecha de vencimiento
importe	Float		Precio de la cuota
alumno_clase_id	INT		Clave foránea de la tabla alumno_clase

Tabla: Clase

Campo	Tipo	Tamaño	Descripción
id	INT		Clave primaria
tipo_clase	Varchar	45	Tipo de clase
cupos_disponibles	INT		Cantidad de cupos disponibles de la clase
horario_id	INT		Clave foránea de la tabla horario
tarifa_id	INT		Clave foránea de la tabla tarifa

Tabla: Horario

Campo	Tipo	Tamaño	Descripción
id	INT		Clave Primaria
hora	Time		Hora de la clase

Tabla: Tarifa

Campo	Tipo	Tamaño	Descripción
id	INT		Clave Primaria
cantidad_dias	INT		Cantidad de Días
precio	Float		Precio de Tarifa

Tabla: Asistencia

Campo	Tipo	Tamaño	Descripción
id	INT		Clave Primaria
fecha_asistencia	DATE		Fecha de la asistencia
alumno_clase_id	INT		Clave Foránea de la tabla Alumno_Clase
asistio	Bool		Estado de la asistencia

Tabla: Rutina

Campo	Tipo	Tamaño	Descripción
id	INT		Clave primaria
fecha_emision	DATE		Fecha de emisión
alumno_clase_id	INT		Clave foránea de la tabla alumno_clase
profesor_id	INT		Clave foránea de la tabla profesor

Tabla: Ejercicio

Campo	Tipo	Tamaño	Descripción
id	INT		Clave primaria
nombre_ejercicio	Varchar	45	Nombre del ejercicio
descripcion	Varchar	45	Descripción del ejercicio

Tabla: Días

Campo	Tipo	Tamaño	Descripción
id	INT		Clave Primaria
dia	Varchar	45	Días de la semana

Tabla: clase_dia

Campo	Tipo	Tamaño	Descripción
clase_id	INT		Clave Foránea de la tabla Clase
dia_id	INT		Clave Foránea de la tabla días

Tabla: Alumno_clase

Campo	Tipo	Tamaño	Descripción
clase_id	INT		Clave foránea de la tabla Clase
alumno_id	INT		Clave foránea de la tabla Alumno

Tabla: Clase_Profesor

Campo	Tipo	Tamaño	Descripción
clase_id	INT		Clave foránea de la tabla Clase
profesor_id	INT		Clave foránea de la tabla Profesor

Tabla: Ejercicio_Rutina

Campo	Tipo	Tamaño	Descripción
id	INT		Clave primaria
rutina_id	INT		Clave Foránea de la tabla Rutina
ejercicio_id	INT		Clave Foránea de la tabla Ejercicio
Series	INT		Series que el alumno debe repetir el ejercicio
Repeticiones	INT		Cantidad de repeticiones que el alumno tiene que efectuar cada serie
Descanso	INT		Descanso que debe cumplir entre cada serie.

Tabla: Clase_Ejercicio

Campo	Tipo	Tamaño	Descripción
clase_id	INT		Clave foránea de la tabla clases
ejercicio_id	INT		Clave foránea de la tabla ejercicios