



Ley Nº 3.688/08

Misión: Es una institución educativa de gestión privada, con capital humano altamente calificado y comprometida en ofrecer una educación integral de calidad, en todos los niveles educativos, inspirada en la cultura propia y universal, basada en los valores humanos, la investigación científica, el servicio a la comunidad, el desarrollo artístico y cultural, para la formación de ciudadanos socialmente responsables.

Facultad de Ciencias y Tecnologías (FACyT)

TRABAJO FINAL

Título del trabajo:

Sistema de Gestión para “The Grand Hall”

Asignatura:

INGENIERÍA DE SOFTWARE II

Nombre del estudiante: Ricardo José Cano Centurión

Carrera: Ingeniería Informática

Semestre: 9no

Nombre del docente: Benito Almada

29 de abril de 2025, Fernando de la Mora

INDICE

1. Información General	4
2. Descripción General del Proyecto.....	5
Objetivo del Software.....	5
Resumen del Sistema	5
Problema que Resuelve.....	5
Audiencia Objetivo	6
Alcance del Proyecto	6
3. Requerimientos.....	8
3.1 Requerimientos Funcionales (RF--).....	8
3.2 Requerimientos No Funcionales (RNF--).....	9
4. Arquitectura del Sistema.....	10
Tipo de arquitectura.....	10
Lenguajes y Frameworks Utilizados	10
DER.....	10
Diagrama de Arquitectura General:	11
5. Diseño del Sistema.....	12
5.1 Diagrama de Casos de Uso	12
5.2 Diagrama de Clases / Estructura de Datos.....	14
5.3 Diagrama de Secuencia / Flujo de Procesos	16
5.4 Mockups / Wireframes / Diseño de UI	29
6. Desarrollo.....	32
6.1 Estructura de Carpetas del Proyecto	32
6.2 Descripción de Módulos o Componentes.....	33
7. Instalación y Configuración.....	35
7.1 Requisitos Previos	35
7.2 Instalación Paso a Paso	36
7.3 Variables de Entorno / Configuración	37
8. Manual de Usuario	38
Login y Registro	38
Gestión de Usuarios (Solo Administrador).....	39

Acceso a Funcionalidades según Rol.....	40
Carga de Datos / Archivos.....	41
Exportación de Resultados	41
9. Pruebas	43
9.1 Estrategia de Pruebas.....	43
9.2 Casos de Prueba Ejemplo	44
10. Seguridad.....	47
Políticas de acceso por rol.....	47
Encriptación de contraseñas.....	47
Validación de formularios.....	48
Prevención de ataques comunes	48
Mejoras futuras	49
11. Mantenimiento y Actualizaciones	50
Cómo actualizar el sistema	50
Logs de cambios / Changelog	51
Consideraciones para backups y recuperación	52
12. Anexos.....	53
Glosario de términos	53
Créditos	54

1. Información General

- Nombre del Proyecto: Sistema de Gestión para “The Grand Hall”
- Version: 1.0
- Fecha de Creación: 02/04/2025
- Última Actualización: 27/04/2025
- Autor: Ricardo Cano
- Responsable del Proyecto: Ricardo Cano
- Contacto: 0984422760

2. Descripción General del Proyecto

Objetivo del Software

El objetivo principal del sistema es gestionar de manera eficiente las reservas de un salón de eventos, permitiendo a los administradores y empleados realizar tareas como la gestión de clientes, reservas, servicios y facturas, asegurando un control centralizado y organizado.

Resumen del Sistema

El sistema es una aplicación web diseñada para centralizar y automatizar las operaciones relacionadas con la gestión de un salón de eventos. Los administradores y empleados tienen acceso completo a las funcionalidades de gestión de clientes, reservas, eventos, servicios y facturación, mientras que los administradores se encargan exclusivamente de la gestión de usuarios y de reportes. El sistema incluye autenticación segura, roles diferenciados y una interfaz intuitiva para realizar las operaciones necesarias.

Problema que Resuelve

El sistema aborda los siguientes problemas comunes en la gestión de salones de eventos:

1. **Desorganización en la gestión de reservas:** Evita conflictos de horarios y garantiza la disponibilidad de salones.
2. **Falta de centralización:** Centraliza la información de clientes, reservas, servicios y facturas en una base de datos relacional.
3. **Errores manuales:** Reduce errores humanos al automatizar tareas administrativas.
4. **Falta de seguridad:** Protege la información sensible mediante autenticación segura y cifrado de contraseñas.
5. **Falta de control:** Proporciona un control claro sobre las operaciones del salón, con roles diferenciados para empleados y administradores.

Audiencia Objetivo

- ✓ **Administradores del salón de eventos:** Encargados con permiso exclusivo de gestionar los usuarios del sistema y reportes de usuarios, clientes y reservas, así como también cuentan con acceso para gestionar clientes, reservas, eventos, servicios y facturación.
- ✓ **Empleados del salón de eventos:** Encargados de gestionar clientes, reservas, eventos, servicios y facturación.
- ✓ **Clientes del salón de eventos:** Personas que desean realizar reservas para eventos.

Alcance del Proyecto

El sistema de gestión de reservas para un salón de eventos tiene como objetivo principal centralizar y automatizar las operaciones relacionadas con la administración de clientes, reservas, eventos, servicios y facturación. A continuación, se detallan las áreas clave del alcance del proyecto:

1. Gestión de Usuarios (Administrador)

- El administrador tiene acceso exclusivo a la tabla de usuarios.
- Funcionalidades disponibles para el administrador:
 - Registro de nuevos usuarios con roles específicos (Administrador, Empleado).
 - Edición de usuarios existentes (cambio de nombre o rol).
 - Eliminación de usuarios.
 - Cifrado de contraseñas para garantizar la seguridad de las credenciales.
- Control de acceso basado en roles:
 - **Administrador:** Tiene acceso completo al sistema, y también cuenta con acceso a la gestión de usuarios y reportes.
 - **Empleado:** Tiene acceso completo al sistema, exceptuando los módulos exclusivos de los administradores.

2. Gestión de Clientes (Administrador y Empleado)

- Registro de clientes con información básica como nombre, teléfono, correo electrónico y CI/RUC.
- Validación de datos para garantizar la integridad (por ejemplo, formato de correo electrónico y longitud del teléfono).
- Edición y eliminación de clientes existentes.
- Relación de clientes con reservas realizadas.

3. Gestión de Salones (Administrador y Empleado)

- Registro de salones disponibles para eventos, incluyendo su capacidad y precio.
- Edición y eliminación de salones.
- Relación de salones con reservas realizadas.

4. Gestión de Reservas (Administrador y Empleado)

- Creación de reservas para clientes, asociando un salón y una fecha específica.
- Edición y cancelación de reservas existentes.
- Relación de reservas con eventos y servicios contratados.

5. Gestión de Servicios (Administrador y Empleado)

- Registro de servicios adicionales ofrecidos por el salón.
- Asignación de servicios a reservas específicas mediante una tabla intermedia
- Edición y eliminación de servicios disponibles.

6. Gestión de Facturación (Administrador y Empleado)

- Generación de facturas para reservas realizadas, incluyendo el total a pagar y el timbrado.
- Relación de facturas con reservas para un control financiero claro.
- Validación de datos para garantizar la integridad de las facturas.

7. Seguridad

- Uso de contraseñas cifradas con bcrypt para proteger las credenciales de los usuarios.
- Validación de datos en todas las entradas para evitar errores y garantizar la integridad de la información.
- Control de acceso basado en roles para limitar las funcionalidades según el tipo de usuario.

8. Reportes (Administrador)

- Generación de reportes sobre reservas realizadas, clientes y usuarios.

3. Requerimientos

3.1 Requerimientos Funcionales (RF--)

1. **RF01:** El sistema debe permitir a los usuarios iniciar sesión con credenciales válidas.
2. **RF02:** El administrador puede agregar, editar y eliminar usuarios.
3. **RF03:** El sistema debe permitir al administrador y al empleado gestionar clientes (agregar, editar y eliminar).
4. **RF04:** El sistema debe permitir al administrador y al empleado gestionar reservas (crear, editar y cancelar).
5. **RF05:** El sistema debe permitir al administrador y al empleado gestionar eventos asociados a reservas.
6. **RF06:** El sistema debe permitir al administrador y al empleado gestionar servicios adicionales (agregar, editar y eliminar).
7. **RF07:** El sistema debe permitir al administrador y al empleado asignar servicios a reservas específicas.
8. **RF08:** El sistema debe generar facturas para reservas realizadas.
9. **RF09:** El sistema debe mostrar listas de usuarios, clientes, reservas, eventos, servicios y facturas con opciones de gestión.
10. **RF10:** El sistema debe validar los datos ingresados antes de almacenarlos en la base de datos.
11. **RF11:** El sistema debe permitir al administrador generar reportes para el control de clientes, usuarios y reservas.

3.2 Requerimientos No Funcionales (RNF--)

1. **RNF01:** El sistema debe responder a las solicitudes en menos de 2 segundos.
2. **RNF02:** Las contraseñas de los usuarios deben ser cifradas utilizando bcrypt.
3. **RNF03:** El sistema debe ser accesible desde navegadores modernos como Google Chrome, Firefox y Microsoft Edge.
4. **RNF04:** La base de datos debe garantizar la integridad de los datos mediante claves primarias y restricciones.
5. **RNF05:** El sistema debe ser fácil de usar, con una interfaz intuitiva y bien organizada.
6. **RNF06:** El sistema debe ser escalable para permitir la adición de nuevas funcionalidades en el futuro.
7. **RNF07:** Los datos sensibles, como contraseñas y facturación, deben estar protegidos contra accesos no autorizados.

4. Arquitectura del Sistema

Tipo de arquitectura

El sistema utiliza una arquitectura Cliente-Servidor con un enfoque MVC (Modelo-Vista-Controlador):

- ✓ **Modelo:** Representado por las tablas de la base de datos SQLite, que almacenan la información de usuarios, clientes, reservas, eventos, servicios y facturas.
- ✓ **Vista:** Plantillas HTML renderizadas con Flask para mostrar la interfaz de usuario.
- ✓ **Controlador:** Lógica de negocio implementada en Python con Flask, que conecta la vista con el modelo y gestiona las solicitudes del usuario.

Lenguajes y Frameworks Utilizados

- ✓ Backend: Python con Flask.
- ✓ Frontend: HTML, CSS y JavaScript.
- ✓ Base de Datos: SQLite.

DER

Sistema De Gestion de Reservas para Salon de Eventos

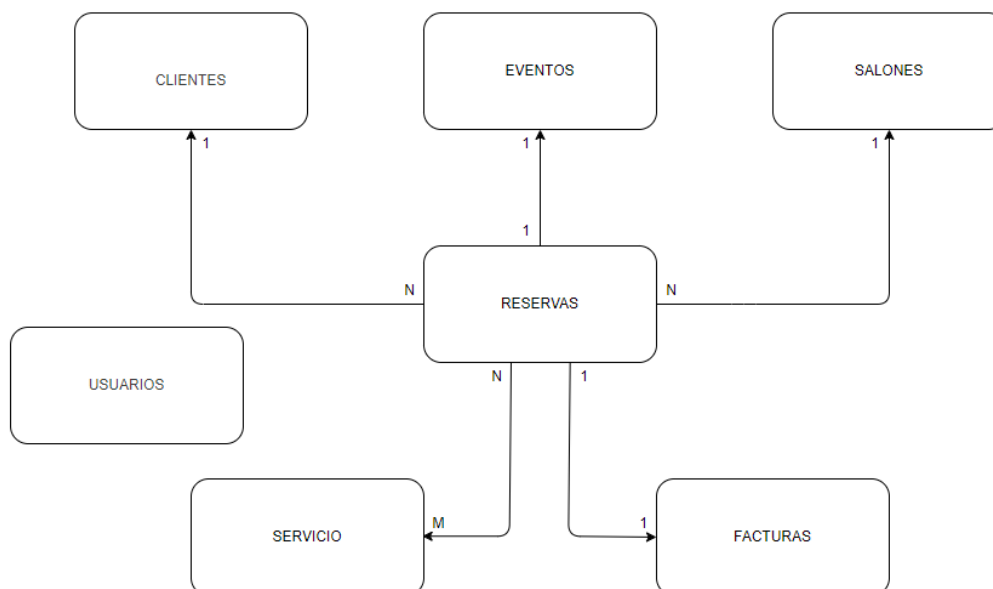
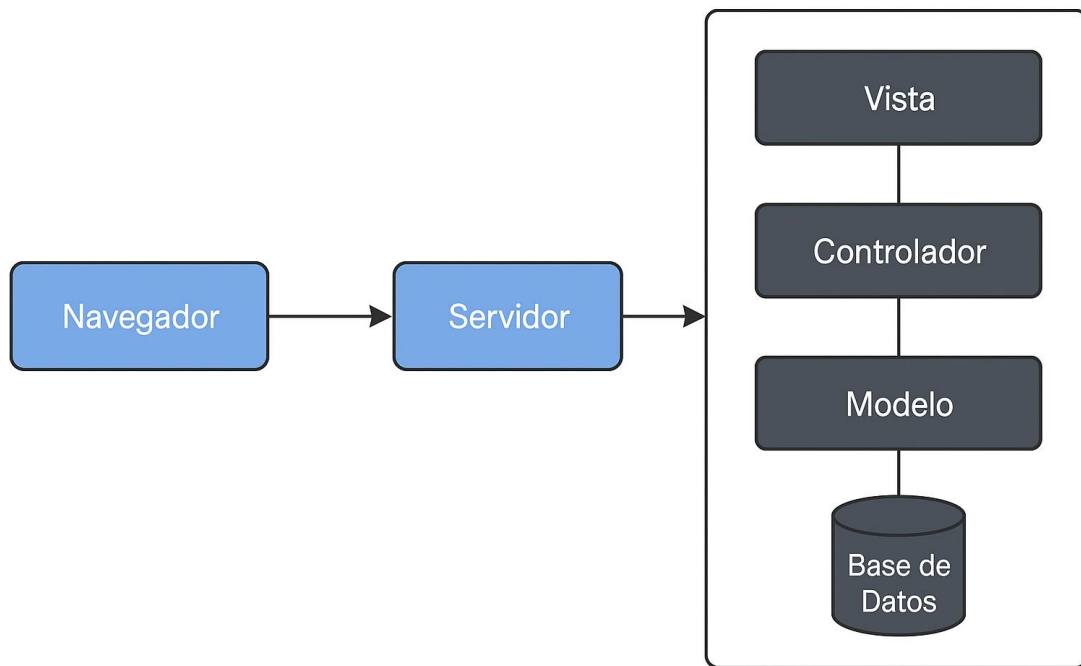


Diagrama de Arquitectura General:



5. Diseño del Sistema

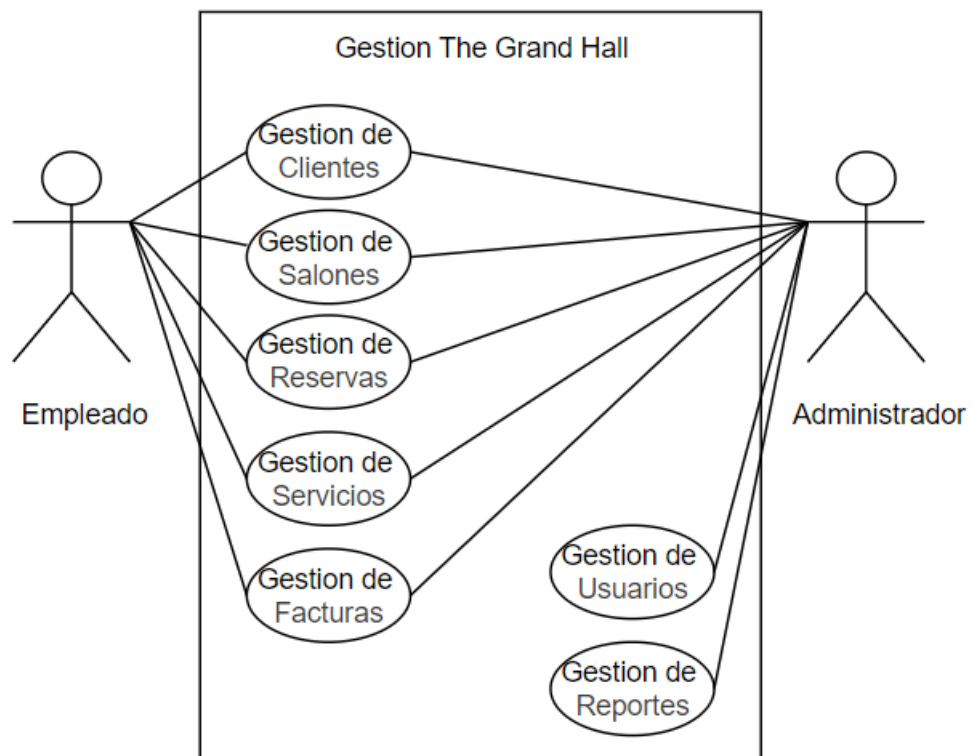
5.1 Diagrama de Casos de Uso

Actores:

- Administrador (rol 1): Tiene acceso completo al sistema, incluyendo la gestión de usuarios, reportes y todas las demás funcionalidades.
- Empleado (rol 2): Tiene acceso limitado a la gestión de clientes, salones, reservas, servicios y facturas.

Casos de uso principales:

- Gestionar usuarios (solo administrador).
- Gestionar reportes (solo administrador).
- Gestionar clientes.
- Gestionar salones.
- Gestionar reservas.
- Gestionar servicios.
- Gestionar facturas.
- Generar reportes en PDF (clientes, reservas, usuarios).
- Generar facturas en PDF.
- Buscar clientes por CI/RUC.
- Buscar reservas por cliente.
- Iniciar / cerrar sesión.



5.2 Diagrama de Clases / Estructura de Datos

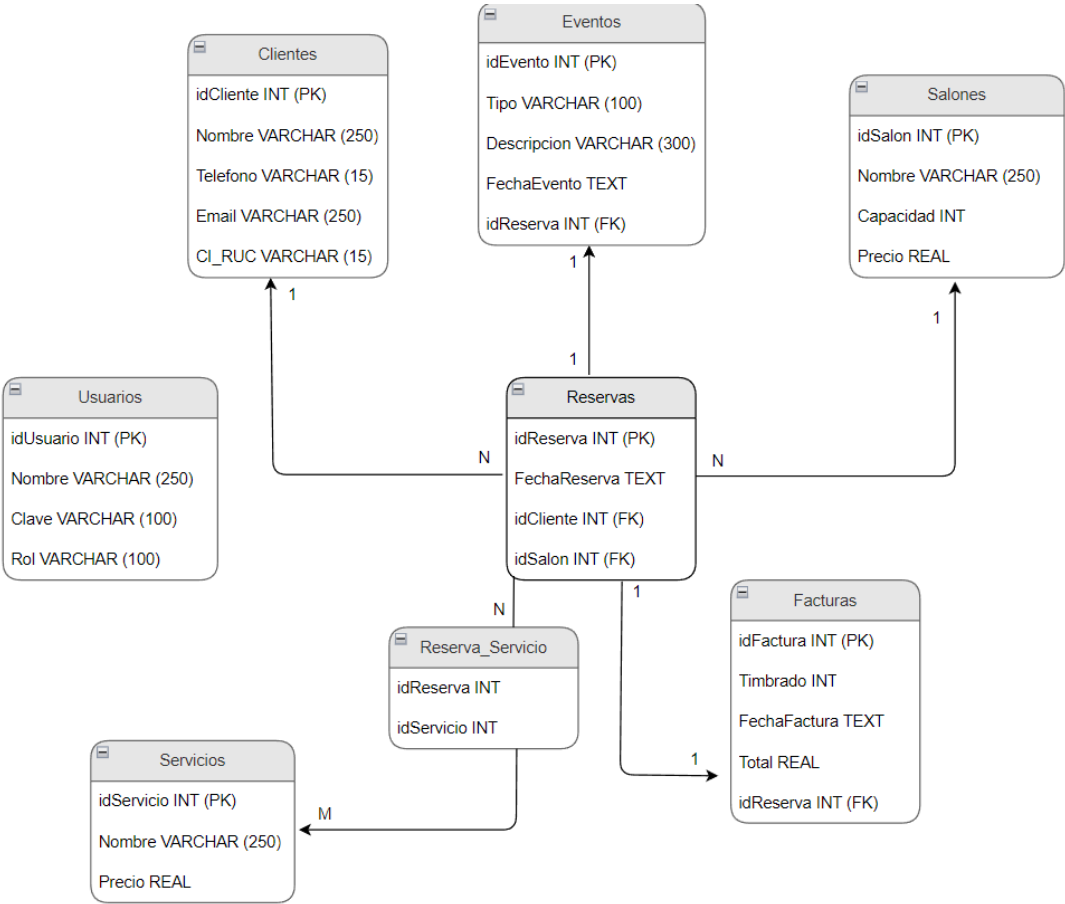
Entidades:

- Usuario: idUsuario, Nombre, Clave, Rol.
- Cliente: idCliente, Nombre, Telefono, Email, CI_RUC.
- Salón: idSalon, Nombre, Capacidad, Precio.
- Reserva: idReserva, FechaReserva, idCliente, idSalon.
- Servicio: idServicio, Nombre, Precio.
- ReservaServicio: idReservaServicio, idReserva, idServicio.
- Factura: idFactura, idReserva, FechaFactura, Total.

Relaciones:

- Un usuario tiene un rol.
- Un cliente puede tener varias reservas.
- Un salón puede estar asociado a varias reservas.
- Una reserva puede incluir varios servicios (relación muchos a muchos).
- Una factura está asociada a una reserva.

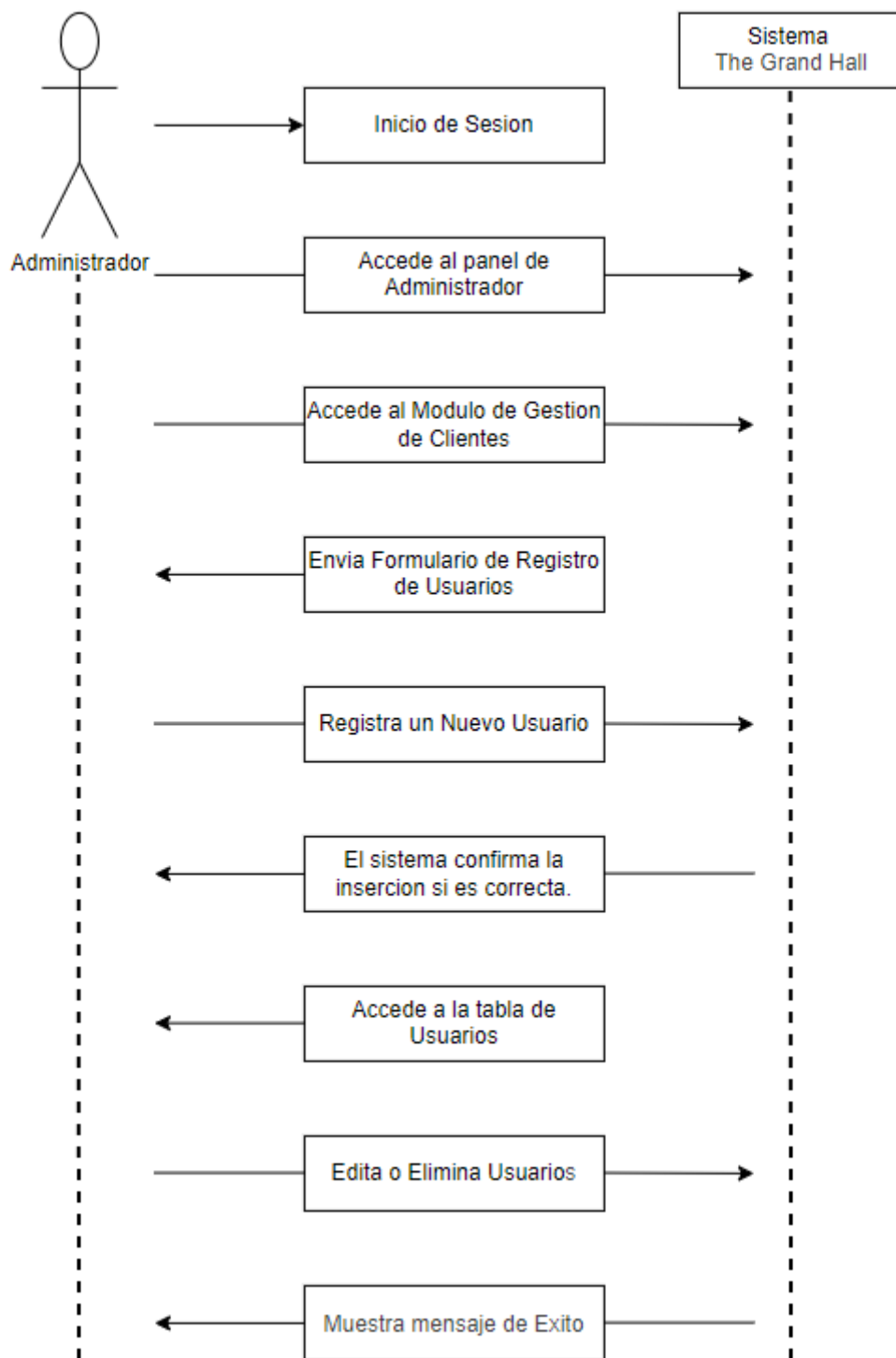
Diagrama de Clases



5.3 Diagrama de Secuencia / Flujo de Procesos

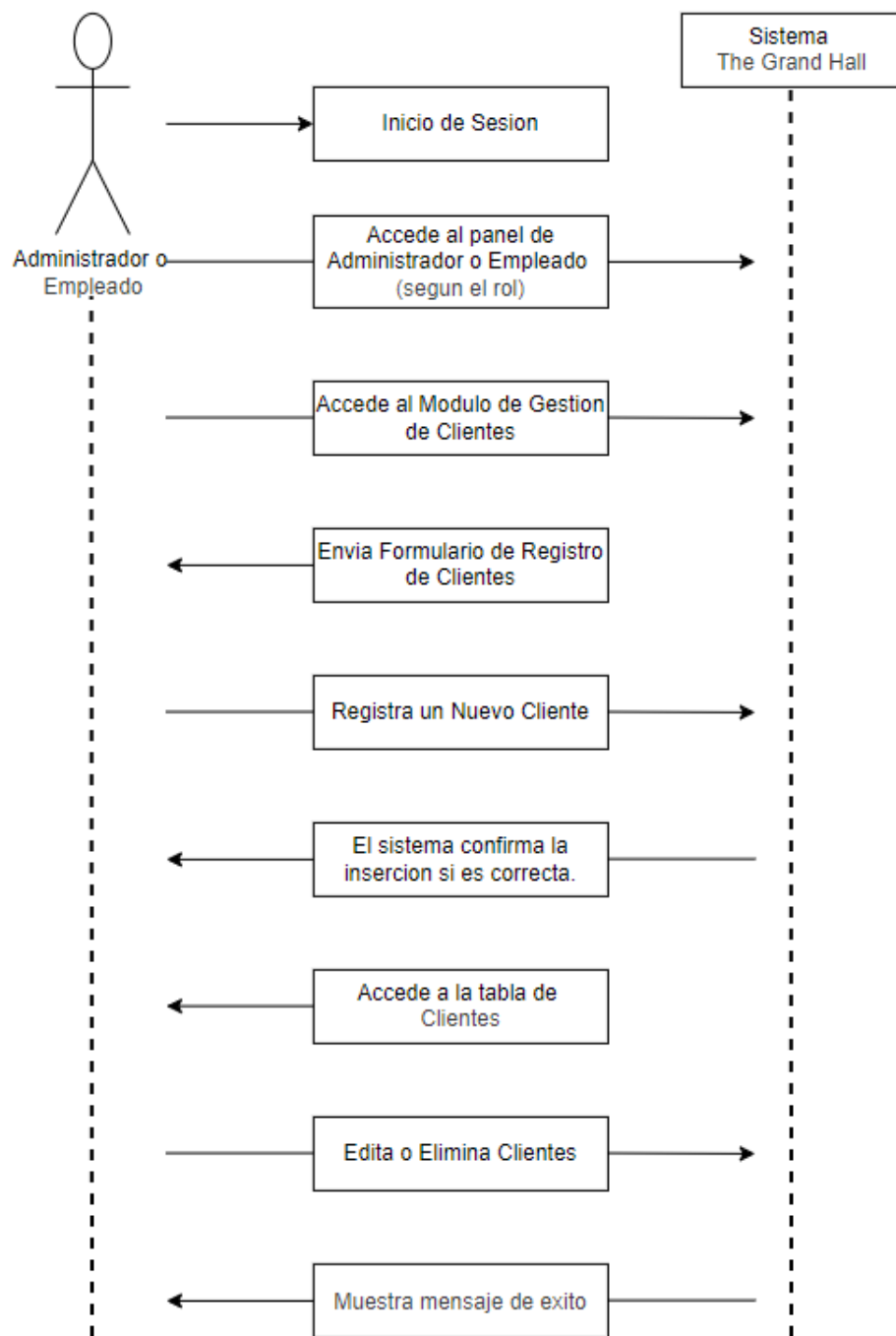
Módulo de Usuarios (solo Administrador)

1. El administrador accede al módulo de gestión de usuarios desde el panel de administración.
2. El sistema muestra una tabla con el listado de usuarios existentes.
3. El administrador puede realizar las siguientes acciones:
 - **Agregar usuario:**
 1. Hace clic en el botón "Agregar Usuario".
 2. Completa el formulario con los datos del nuevo usuario (nombre, contraseña, rol).
 3. El sistema valida los datos y guarda el usuario en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Editar usuario:**
 1. Selecciona un usuario de la tabla y hace clic en "Editar".
 2. Modifica los datos necesarios.
 3. El sistema valida los cambios y actualiza el usuario en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Eliminar usuario:**
 1. Selecciona un usuario de la tabla y hace clic en "Eliminar".
 2. El sistema solicita confirmación.
 3. Si se confirma, elimina el usuario de la base de datos.
 4. Muestra un mensaje de éxito.



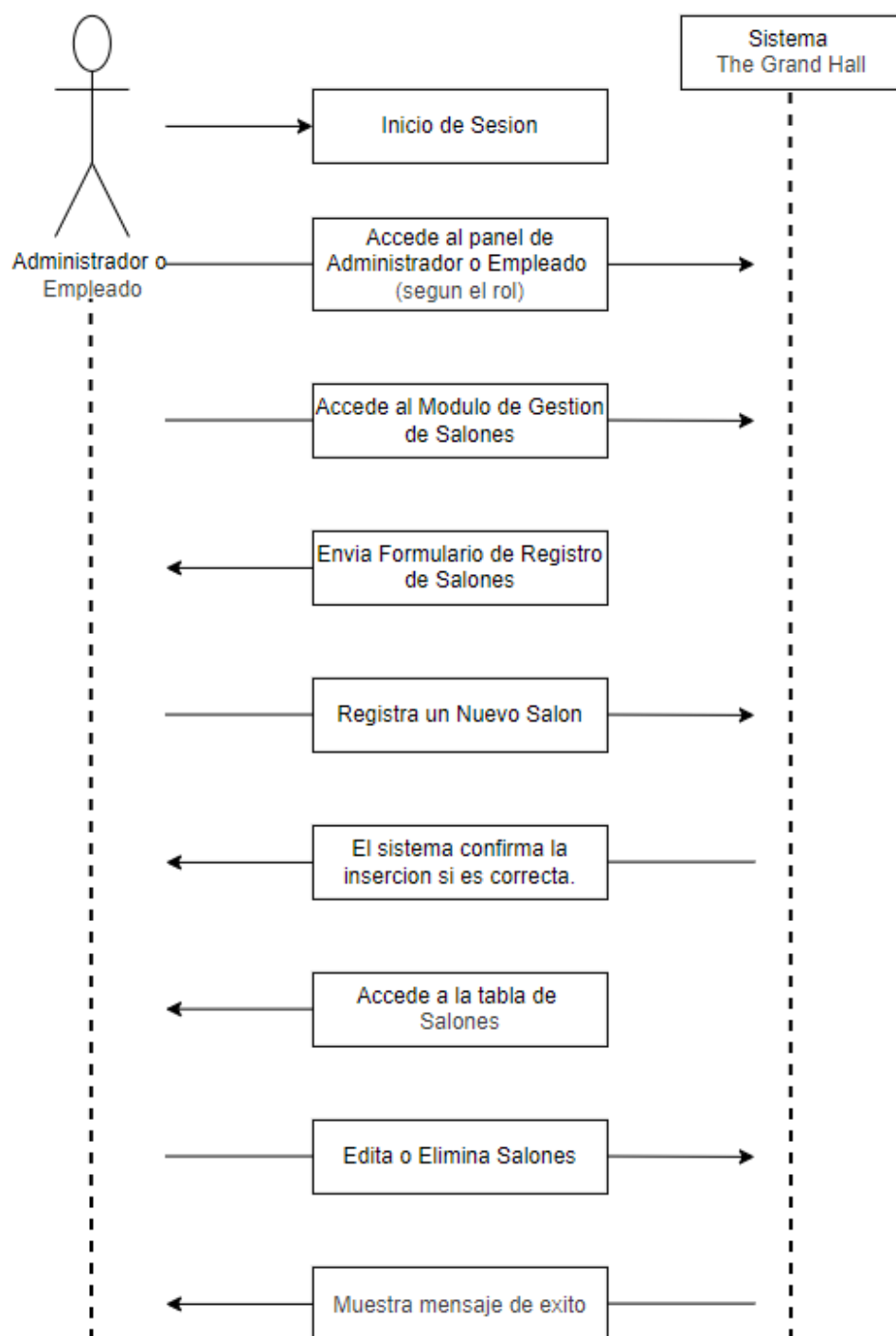
Módulo de Clientes

1. El usuario (Administrador o Empleado) accede al módulo de gestión de clientes.
2. El usuario puede realizar las siguientes acciones:
 - **Agregar cliente:**
 1. Hace clic en el botón "Agregar Cliente".
 2. Completa el formulario con los datos del cliente (nombre, teléfono, email, CI/RUC).
 3. El sistema valida los datos y guarda el cliente en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Editar cliente:**
 1. Selecciona un cliente de la tabla y hace clic en "Editar".
 2. Modifica los datos necesarios.
 3. El sistema valida los cambios y actualiza el cliente en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Eliminar cliente:**
 1. Selecciona un cliente de la tabla y hace clic en "Eliminar".
 2. El sistema solicita confirmación.
 3. Si se confirma, elimina el cliente de la base de datos.
 4. Muestra un mensaje de éxito.



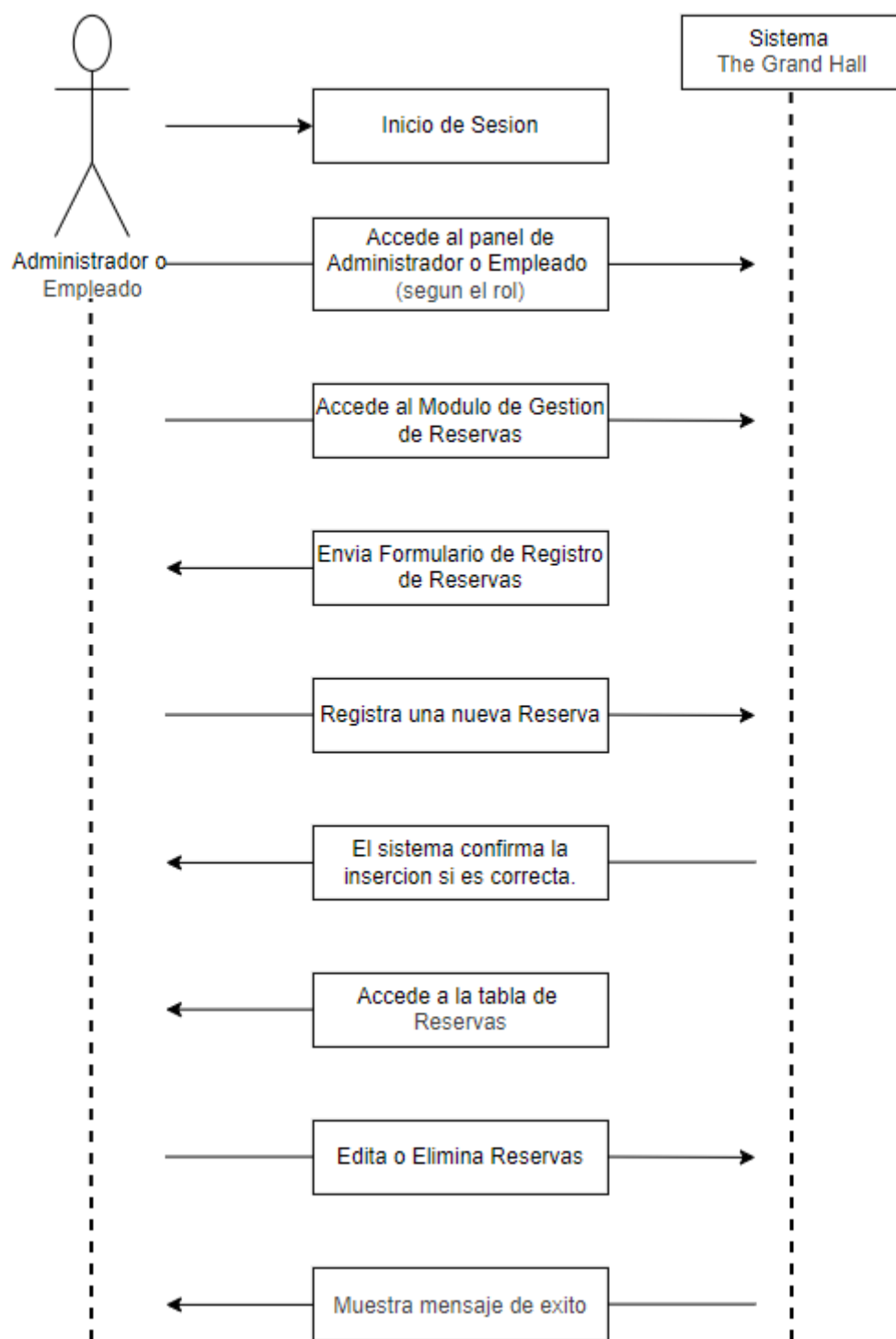
Módulo de Salones

1. El usuario accede al módulo de gestión de salones.
2. El sistema muestra una tabla con el listado de salones existentes.
3. El usuario puede realizar las siguientes acciones:
 - **Agregar salón:**
 1. Hace clic en el botón "Agregar Salón".
 2. Completa el formulario con los datos del salón (nombre, capacidad, precio).
 3. El sistema valida los datos y guarda el salón en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Editar salón:**
 1. Selecciona un salón de la tabla y hace clic en "Editar".
 2. Modifica los datos necesarios.
 3. El sistema valida los cambios y actualiza el salón en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Eliminar salón:**
 1. Selecciona un salón de la tabla y hace clic en "Eliminar".
 2. El sistema solicita confirmación.
 3. Si se confirma, elimina el salón de la base de datos.
 4. Muestra un mensaje de éxito.



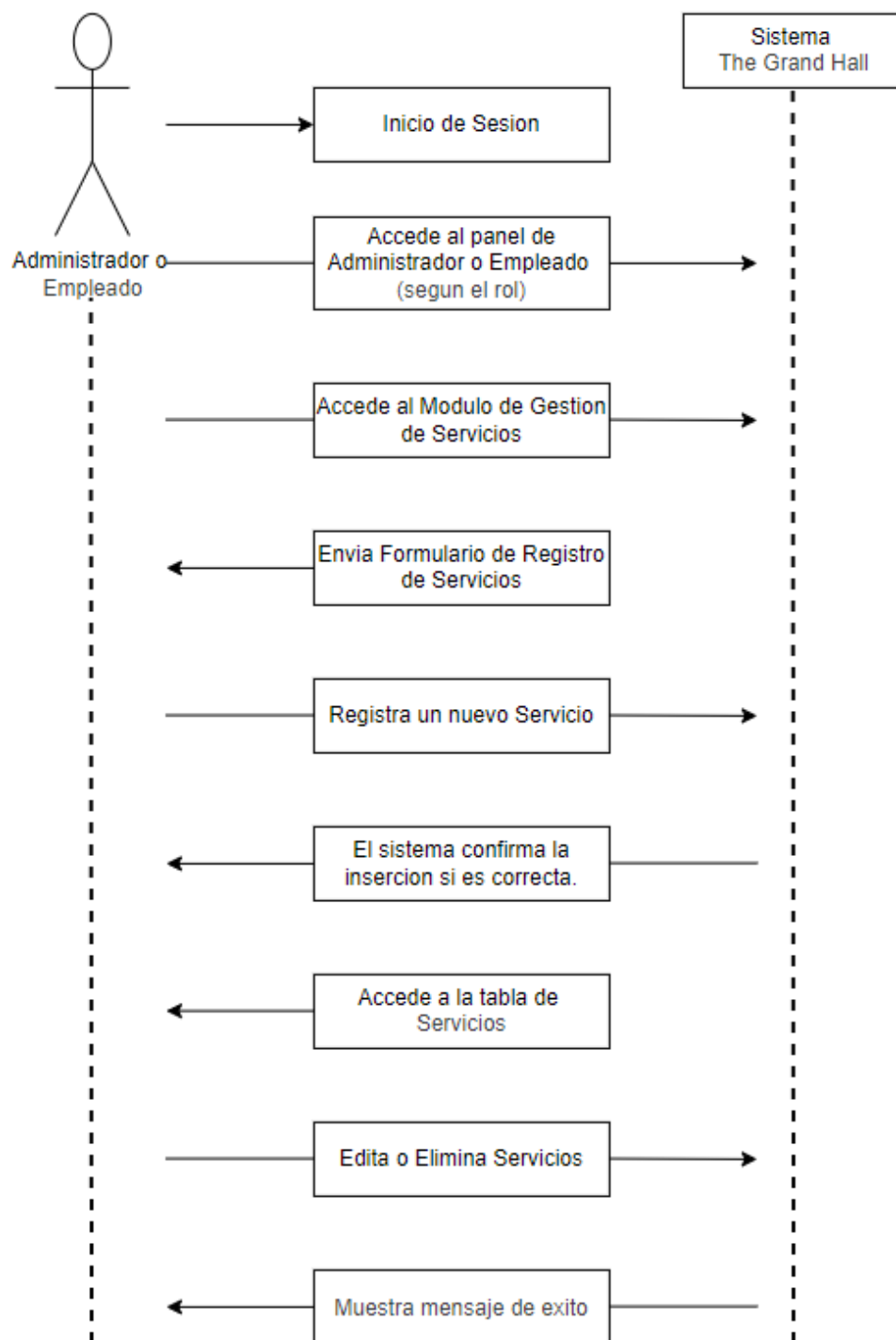
Módulo de Reservas

1. El usuario accede al módulo de gestión de reservas.
2. El sistema muestra una tabla con el listado de reservas existentes.
3. El usuario puede realizar las siguientes acciones:
 - **Agregar reserva:**
 1. Hace clic en el botón "Agregar Reserva".
 2. Completa el formulario con los datos de la reserva (cliente, salón, servicios, fecha).
 3. El sistema valida los datos y guarda la reserva en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Editar reserva:**
 1. Selecciona una reserva de la tabla y hace clic en "Editar".
 2. Modifica los datos necesarios.
 3. El sistema valida los cambios y actualiza la reserva en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Eliminar reserva:**
 1. Selecciona una reserva de la tabla y hace clic en "Eliminar".
 2. El sistema solicita confirmación.
 3. Si se confirma, elimina la reserva de la base de datos.
 4. Muestra un mensaje de éxito.



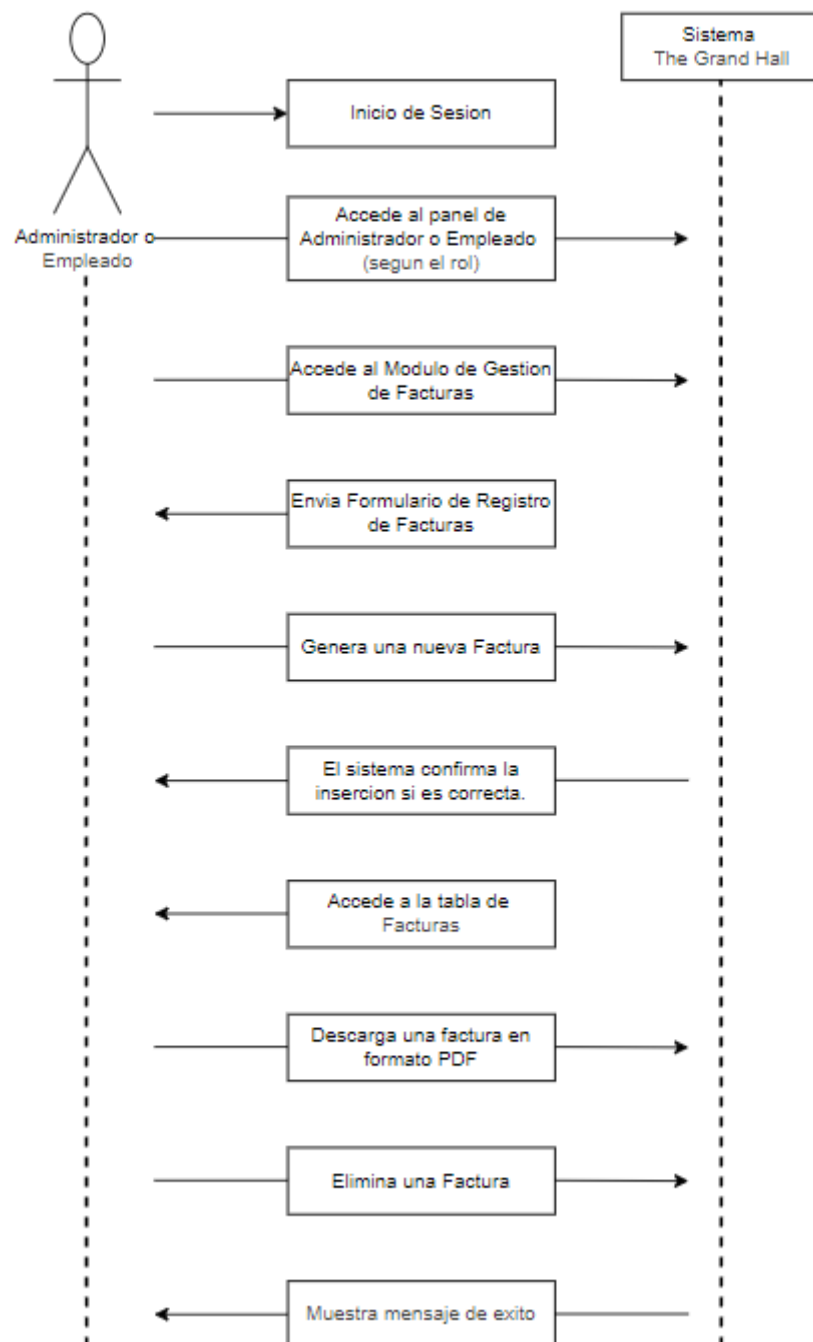
Módulo de Servicios

1. El usuario accede al módulo de gestión de servicios.
2. El sistema muestra una tabla con el listado de servicios existentes.
3. El usuario puede realizar las siguientes acciones:
 - **Agregar servicio:**
 1. Hace clic en el botón "Agregar Servicio".
 2. Completa el formulario con los datos del servicio (nombre, precio).
 3. El sistema valida los datos y guarda el servicio en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Editar servicio:**
 1. Selecciona un servicio de la tabla y hace clic en "Editar".
 2. Modifica los datos necesarios.
 3. El sistema valida los cambios y actualiza el servicio en la base de datos.
 4. Muestra un mensaje de éxito.
 - **Eliminar servicio:**
 1. Selecciona un servicio de la tabla y hace clic en "Eliminar".
 2. El sistema solicita confirmación.
 3. Si se confirma, elimina el servicio de la base de datos.
 4. Muestra un mensaje de éxito.



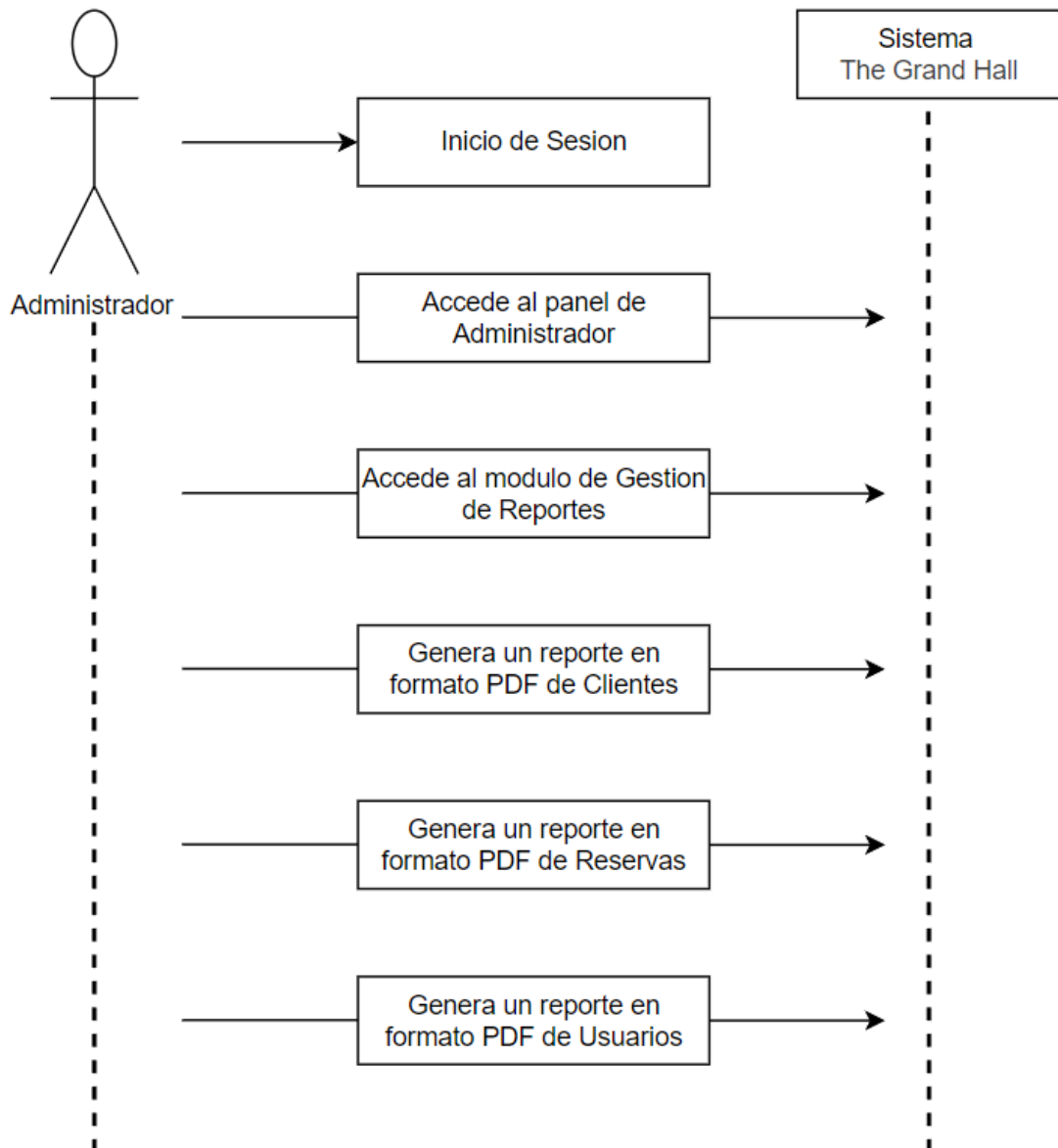
Módulo de Facturas

1. El usuario accede al módulo de gestión de facturas.
2. El sistema muestra una tabla con el listado de facturas existentes.
3. El usuario puede realizar las siguientes acciones:
 - **Generar factura:**
 1. Hace clic en el botón "Generar factura" asociado a una reserva.
 2. El sistema genera la factura en PDF y la guarda en la base de datos.
 3. Muestra un mensaje de éxito.
 - **Eliminar factura:**
 1. Selecciona una factura de la tabla y hace clic en "Eliminar".
 2. El sistema solicita confirmación.
 3. Si se confirma, elimina la factura de la base de datos.
 4. Muestra un mensaje de éxito.



Módulo de Reportes

1. El administrador accede al módulo de generación de reportes.
2. Selecciona el tipo de reporte que desea generar (clientes, reservas, usuarios).
3. El sistema genera el reporte en PDF.



5.4 Mockups / Wireframes / Diseño de UI

- **Pantalla de Login**

Características:

- Campo para ingresar el nombre de usuario.
- Campo para ingresar la contraseña.
- Selección de rol (Administrador o Empleado).
- Botón para iniciar sesión.
- Validación de credenciales y redirección al panel correspondiente según el rol



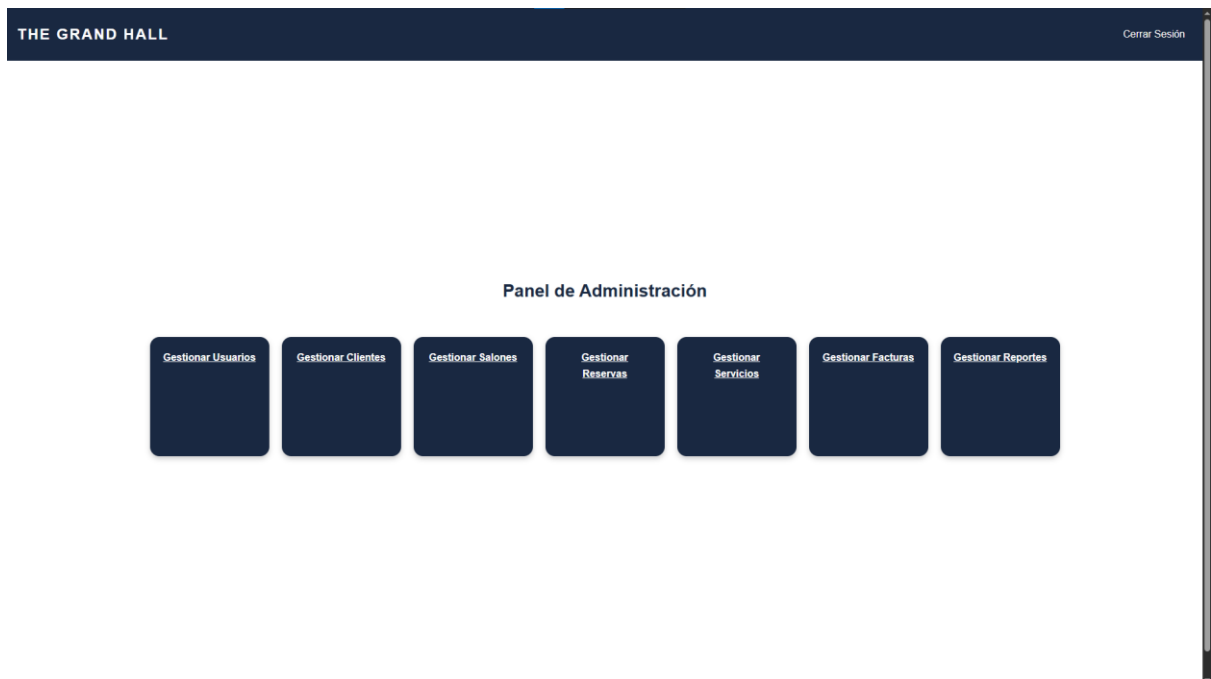
The mockup shows a login interface within a rounded rectangle. On the left is a square image with a dark blue background, featuring a golden archway with columns and the text 'THE GRAND HALL' at the bottom. To the right of the image, the title 'Inicio de Sesión' is centered. Below the title are three input fields: 'Usuario:' (a text box), 'Contraseña:' (a text box), and 'Ingresar como:' (a dropdown menu with the placeholder text 'Seleccione un rol'). At the bottom right is a dark blue button with the white text 'Iniciar Sesión'.

Pantalla de Login con campos de usuario, contraseña, selección de rol y botón de inicio de sesión.

- **Panel de Administrador**

Características:

- Menú superior con el nombre del sistema y botón "Cerrar Sesión".
- Botones para acceder a los módulos:
 - Gestionar Usuarios
 - Gestionar Clientes
 - Gestionar Salones
 - Gestionar Reservas
 - Gestionar Servicios
 - Gestionar Facturas
 - Gestionar Reportes

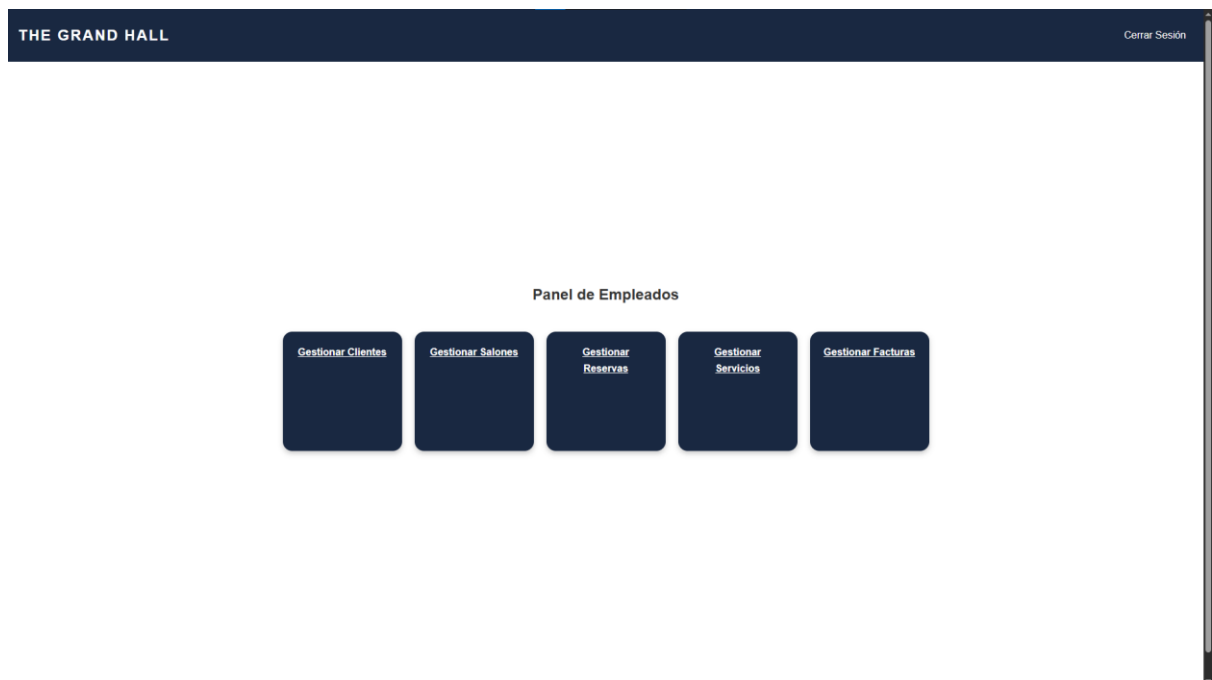


Panel principal del Administrador con acceso a todos los módulos del sistema.

- **Panel de Empleado**

Características:

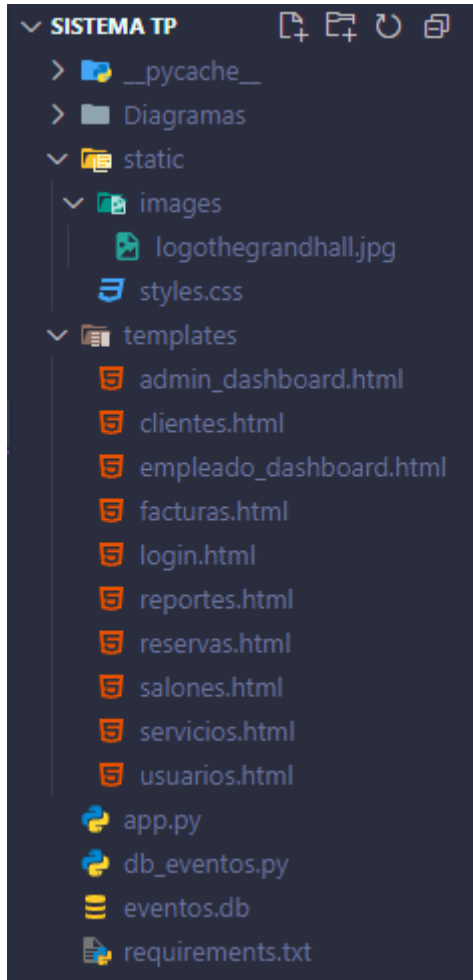
- Menú superior con el nombre del sistema y botón "Cerrar Sesión".
- Botones para acceder a los módulos:
 - Gestionar Clientes
 - Gestionar Salones
 - Gestionar Reservas
 - Gestionar Servicios
 - Gestionar Facturas



Panel del Empleado con módulos habilitados según su rol.

6. Desarrollo

6.1 Estructura de Carpetas del Proyecto



6.2 Descripción de Módulos o Componentes

Módulo de Autenticación:

- **Responsabilidad:** Gestiona el inicio y cierre de sesión, validación de credenciales y control de acceso por rol.
- **Funciones clave:**
 - Login y logout de usuarios.
 - Validación de sesión.
 - Redirección según nivel de acceso.
 - Seguridad: Hash de contraseñas y protección de rutas.

Módulo de Administración de Usuarios:

- **Responsabilidad:** Permite al administrador gestionar los usuarios del sistema.
- **Funciones clave:**
 - CRUD de usuarios (crear, editar, eliminar, listar).
 - Asignación y edición de roles.
 - Acceso restringido al administrador.

Módulo de Clientes:

- **Responsabilidad:** Registro y gestión de clientes.
- **Funciones clave:**
 - CRUD de clientes (crear, editar, eliminar, listar).

Módulo de Salones:

- **Responsabilidad:** Registro y gestión de salones.
- **Funciones clave:**
 - CRUD de salones (crear, editar, eliminar, listar).

Módulo de Reservas:

- **Responsabilidad:** Registro y gestión de reservas.
- **Funciones clave:**
 - CRUD de reservas (crear, editar, eliminar, listar).
 - Asociación de servicios a reservas.
 - Búsqueda de reservas por cliente.

Módulo de Servicios:

- **Responsabilidad:** Registro y gestión de servicios.
- **Funciones clave:**
 - CRUD de servicios (crear, editar, eliminar, listar).

Módulo de Facturas:

- **Responsabilidad:** Generación y gestión de facturas.
- **Funciones clave:**
 - Generación de facturas en PDF.
 - Eliminación de facturas.

Módulo de Reportes:

- **Responsabilidad:** Generación de reportes en PDF.
- **Funciones clave:**
 - Reporte de clientes.
 - Reporte de reservas.
 - Reporte de usuarios.
 - Acceso restringido al administrador.

7. Instalación y Configuración

7.1 Requisitos Previos

Antes de instalar y configurar el sistema, debes de cumplir con los siguientes requisitos:

1. Sistema Operativo:

- Windows 10/11, macOS, o Linux.

2. Software necesario:

- Python 3.10 o superior: Asegúrate de que Python esté instalado y configurado en tu sistema.
- SQLite: Base de datos integrada en Python.
- pip: Administrador de paquetes de Python (incluido con Python).

3. Librerías de Python requeridas:

- bcrypt: Para el cifrado de contraseñas.
- flask: Framework web para la aplicación.
- reportlab: Para la generación de reportes en PDF.
- blinker: Para señales internas de Flask.
- chardet: Para la detección de codificación de texto.

4. Herramientas adicionales:

- Editor de texto o IDE (recomendado: Visual Studio Code).
- Navegador web (Google Chrome, Firefox, etc.).

7.2 Instalación Paso a Paso

1. Clonar o descargar el proyecto:

- Descarga el proyecto desde el repositorio o copia los archivos en tu máquina local.
- Link Repositorio: <https://github.com/RicardoCano06/Sistema-TP.git>

2. Instalar las dependencias:

- Ejecuta el siguiente comando para instalar las librerías necesarias desde el archivo requirements.txt:

✓ `pip install -r requirements.txt`

3. Configurar la base de datos:

- Asegúrate de que el archivo eventos.db esté en la raíz del proyecto.
- Si necesitas crear las tablas desde cero, ejecuta el script db_eventos.py:

✓ `python db_eventos.py`

- Esto creará las tablas necesarias en la base de datos SQLite.

4. Ejecutar la aplicación:

- Inicia el servidor Flask ejecutando el archivo app.py:

▪ `python app.py`

- El servidor estará disponible en `http://127.0.0.1:5000`.

5. Acceder al sistema:

- Abre un navegador web y accede a `http://127.0.0.1:5000`.
- Usa las credenciales predeterminadas para iniciar sesión:
 - Administrador: Usuario: admin, Contraseña: admin.
 - Empleado: Usuario: user, Contraseña: user.

7.3 Variables de Entorno / Configuración

1. Clave secreta de Flask:

- La clave secreta para las sesiones está definida en el archivo app.py:
 - `app.secret_key = 'THG_tpingsft'`
- Puedes cambiar esta clave por una más segura si es necesario.

2. Configuración de la base de datos:

- La base de datos SQLite está configurada en el archivo app.py:
 - `conn = sqlite3.connect('eventos.db')`
- Si deseas usar otra base de datos (por ejemplo, MySQL o PostgreSQL), deberás modificar esta configuración y las consultas SQL.

3. Archivos estáticos y plantillas:

- Los archivos estáticos (CSS, imágenes) están en la carpeta static.
- Las plantillas HTML están en la carpeta templates.

4. Configuración del servidor:

- El servidor Flask está configurado para ejecutarse en modo de depuración:
 - `app.run(debug=True)`
- Para producción, desactiva el modo de depuración y usa un servidor como Gunicorn o uWSGI.

8. Manual de Usuario

Login y Registro

1. Acceso al sistema:

- Abre tu navegador web y accede a la URL del sistema: `http://127.0.0.1:5000`.
- Ingresa tu nombre de usuario, contraseña y selecciona tu rol (Administrador o Empleado).
- Haz clic en el botón "Iniciar Sesión".
- Si las credenciales son correctas:
 - Los administradores serán redirigidos al Panel de Administración.
 - Los empleados serán redirigidos al Panel de Empleado.
- Si las credenciales son incorrectas, aparecerá un mensaje de error.

2. Cerrar sesión:

- Haz clic en el botón "Cerrar Sesión" en la parte superior derecha de cualquier pantalla para salir del sistema.

Gestión de Usuarios (Solo Administrador)

1. Acceso al módulo:

- Desde el Panel de Administración, haz clic en el botón "Gestionar Usuarios".

2. Crear un usuario:

- Haz clic en el botón "Agregar Usuario".
- Completa el formulario con los datos del usuario:
 - Nombre.
 - Contraseña.
 - Rol (Administrador o Empleado).
- Haz clic en "Guardar" para registrar el usuario.

3. Editar un usuario:

- En la tabla de usuarios, selecciona el usuario que deseas modificar y haz clic en "Editar".
- Modifica los datos necesarios y haz clic en "Guardar".

4. Eliminar un usuario:

- En la tabla de usuarios, selecciona el usuario que deseas eliminar y haz clic en "Eliminar".
- Confirma la acción en el mensaje emergente.

5. Cambiar contraseña:

- Selecciona un usuario y haz clic en "Cambiar Contraseña".
- Ingresa la nueva contraseña y confírmala.
- Haz clic en "Guardar".

Acceso a Funcionalidades según Rol

1. Administrador:

- Tiene acceso completo a todos los módulos:
 - Gestión de Usuarios.
 - Gestión de Clientes.
 - Gestión de Salones.
 - Gestión de Reservas.
 - Gestión de Servicios.
 - Gestión de Facturas.
 - Generación de Reportes.

2. Empleado:

- Tiene acceso limitado a los siguientes módulos:
 - Gestión de Clientes.
 - Gestión de Salones.
 - Gestión de Reservas.
 - Gestión de Servicios.
 - Gestión de Facturas.

Carga de Datos / Archivos

1. Agregar datos en los módulos:

- En cada módulo (Clientes, Salones, Reservas, Servicios), haz clic en el botón "Agregar".
- Completa el formulario con los datos requeridos.
- Haz clic en "Guardar" para registrar los datos.

2. Carga de datos en Reservas:

- Al crear una reserva, selecciona:
 - Cliente. (Busqueda por C.I de Cliente)
 - Salón.
 - Fecha de la reserva.
 - Servicios asociados (opcional).
- Haz clic en "Guardar" para registrar la reserva.

Exportación de Resultados

1. Generar Facturas en PDF:

- Accede al módulo Facturas.
- Selecciona una reserva y haz clic en "Generar Factura".
- La factura se generará en formato PDF y estará disponible para descargar.

2. Generar Reportes en PDF:

- Accede al módulo Reportes (solo Administrador).
- Selecciona el tipo de reporte que deseas generar:
 - Reporte de Clientes.
 - Reporte de Reservas.
 - Reporte de Usuarios.
- El sistema generará el reporte en formato PDF y se podrá descargar.

9. Pruebas

9.1 Estrategia de Pruebas

Actualmente, las pruebas del sistema se realizan de manera manual, verificando que cada funcionalidad cumpla con los requisitos establecidos. Las pruebas se dividen en las siguientes categorías:

Pruebas Funcionales

- Objetivo: Validar que las funcionalidades principales del sistema cumplan con los requisitos del usuario.
- Ejemplo de pruebas funcionales:
 - Probar el flujo de inicio de sesión con diferentes roles (Administrador y Empleado).
 - Validar la creación, edición y eliminación de clientes, salones, reservas y servicios.
 - Verificar que los reportes en PDF se generen correctamente.

Pruebas de Integración

- Objetivo: Verificar que los módulos del sistema interactúen correctamente entre sí.
- Ejemplo de pruebas de integración:
 - Crear un cliente y asociarlo a una reserva.
 - Generar una factura para una reserva con servicios asociados.
 - Verificar que los reportes incluyan los datos correctos.

Pruebas de Seguridad

- Objetivo: Identificar y mitigar vulnerabilidades en el sistema.
- Ejemplo de pruebas de seguridad:
 - Validar que las contraseñas estén cifradas en la base de datos.
 - Probar que las rutas protegidas no sean accesibles sin autenticación.
 - Verificar que los datos sensibles no se expongan en las respuestas del servidor.

9.2 Casos de Prueba Ejemplo

Caso de Prueba 1: Inicio de Sesión

- Descripción: Validar que el sistema permita iniciar sesión con credenciales válidas.
- Precondiciones: El usuario debe estar registrado en la base de datos.
- Pasos:
 1. Acceder a la pantalla de inicio de sesión.
 2. Ingresar un nombre de usuario y contraseña válidos.
 3. Seleccionar el rol correspondiente.
 4. Hacer clic en "Iniciar Sesión".
- Resultado esperado: El usuario es redirigido al panel correspondiente según su rol.

Caso de Prueba 2: Crear Cliente

- Descripción: Validar que el sistema permita crear un cliente con datos válidos.
- Precondiciones: El usuario debe estar autenticado como Administrador o Empleado.
- Pasos:
 1. Acceder al módulo de gestión de clientes.
 2. Hacer clic en "Agregar Cliente".
 3. Completar el formulario con datos válidos (nombre, teléfono, email, CI/RUC).
 4. Hacer clic en "Guardar".
- Resultado esperado: El cliente se agrega correctamente y aparece en la tabla de clientes.

Caso de Prueba 3: Generar Factura

- Descripción: Validar que el sistema genere una factura en PDF para una reserva.
- Precondiciones: Debe existir al menos una reserva con servicios asociados.
- Pasos:
 1. Acceder al módulo de facturas.
 2. Seleccionar una reserva y hacer clic en "Generar Factura".
 3. Descargar el archivo PDF generado.
- Resultado esperado: El archivo PDF contiene los datos correctos de la reserva, cliente, salón y servicios.

Caso de Prueba 4: Acceso Restringido

- Descripción: Validar que un usuario no autenticado no pueda acceder a rutas protegidas.
- Precondiciones: El usuario no debe haber iniciado sesión.
- Pasos:
 1. Intentar acceder a una ruta protegida, como /admin_dashboard.
- Resultado esperado: El sistema redirige al usuario a la pantalla de inicio de sesión.

Caso de Prueba 5: Generar Reporte de Clientes

- Descripción: Validar que el sistema genere un reporte en PDF con los datos de los clientes.
- Precondiciones: Deben existir clientes registrados en la base de datos.
- Pasos:
 1. Acceder al módulo de reportes.
 2. Seleccionar "Reporte de Clientes".
 3. Descargar el archivo PDF generado.
- Resultado esperado: El archivo PDF contiene una tabla con los datos de los clientes registrados.

10. Seguridad

Políticas de acceso por rol

El sistema implementa un control de acceso basado en roles para garantizar que los usuarios solo puedan acceder a las funcionalidades permitidas según su rol:

- **Roles definidos:**
 - **Administrador:** Tiene acceso completo a todos los módulos, incluyendo la gestión de usuarios, reportes y facturas.
 - **Empleado:** Tiene acceso limitado a los módulos de clientes, salones, reservas, servicios y facturas.
- **Restricción de acceso:**
 - Cada ruta verifica si el usuario está autenticado y si su rol le permite acceder a la funcionalidad solicitada.
 - Los usuarios no autenticados son redirigidos a la pantalla de inicio de sesión.

Encriptación de contraseñas

El sistema utiliza **bcrypt** para proteger las contraseñas de los usuarios:

- **Cifrado:** Las contraseñas se cifran antes de almacenarse en la base de datos, utilizando un "salt" único para cada contraseña.
- **Verificación:** Durante el inicio de sesión, las contraseñas ingresadas se comparan con los hashes almacenados en la base de datos.
- **Beneficio:** Esto protege contra ataques de fuerza bruta y diccionario.

Validación de formularios

El sistema valida los datos ingresados por los usuarios para garantizar la integridad de la información:

- **Validaciones en el servidor:**
 - Verificación de campos obligatorios (por ejemplo, nombre, teléfono, email).
 - Validación de formatos (por ejemplo, correos electrónicos válidos, números en campos de teléfono).
 - Prevención de duplicados (por ejemplo, CI/RUC o correos electrónicos ya registrados).
- **Mensajes de error:** Los errores se comunican al usuario mediante alertas o mensajes en la interfaz.

Prevención de ataques comunes

1. Prevención de SQL Injection:

- Todas las consultas a la base de datos utilizan parámetros preparados, evitando que los datos ingresados por el usuario se interpreten como código SQL.

2. Gestión de sesiones:

- El sistema utiliza una clave secreta para proteger las sesiones.
- Las sesiones se eliminan al cerrar sesión, evitando accesos no autorizados.

Mejoras futuras

Aunque el sistema implementa medidas básicas de seguridad, se pueden considerar las siguientes mejoras para fortalecer la protección:

1. Protección contra Cross-Site Scripting (XSS):

- Implementar sanitización de datos ingresados por los usuarios antes de mostrarlos en las páginas para evitar la ejecución de scripts maliciosos.

2. Protección contra Cross-Site Request Forgery (CSRF):

- Implementar tokens CSRF en las solicitudes POST para evitar que un atacante pueda realizar acciones en nombre de un usuario autenticado.

3. Pruebas automatizadas:

- Implementar pruebas unitarias y de integración utilizando herramientas como unittest o pytest para garantizar que las funcionalidades del sistema se comporten correctamente.

4. Cifrado de datos sensibles:

- Además de las contraseñas, considerar el cifrado de otros datos sensibles almacenados en la base de datos, como correos electrónicos o números de teléfono.

5. Registro de auditoría:

- Implementar un sistema de registro de auditoría para rastrear las acciones realizadas por los usuarios, como inicios de sesión, modificaciones de datos y eliminaciones.

11. Mantenimiento y Actualizaciones

Cómo actualizar el sistema

1. Actualizar el código fuente:

- Si el sistema está alojado en un servidor local:
 - Descarga la nueva versión del código desde el repositorio o fuente de distribución.
 - Reemplaza los archivos existentes en la carpeta del proyecto con los nuevos archivos.
- Si el sistema está en un servidor remoto:
 - Usa herramientas como scp o git pull para actualizar los archivos en el servidor.

2. Actualizar las dependencias:

- Si se han agregado nuevas dependencias en el archivo requirements.txt, instálalas ejecutando:
 - ✓ `pip install -r requirements.txt`

3. Actualizar la base de datos:

- Si hay cambios en la estructura de la base de datos (nuevas tablas, columnas, etc.), ejecuta el script de migración correspondiente o actualiza manualmente la base de datos.
- Por ejemplo, puedes usar un script SQL para agregar nuevas columnas o tablas:
 - ✓ `ALTER TABLE Clientes ADD COLUMN Direccion TEXT;`

4. Reiniciar el servidor:

- Después de realizar los cambios, reinicia el servidor Flask para que los cambios surtan efecto:
 - ✓ `python app.py`

Logs de cambios / Changelog

Mantener un registro de los cambios realizados en el sistema para facilitar el seguimiento de actualizaciones. Por ejemplo utilizando este formato:

Changelog:

- **Versión 1.1.0 (27/04/2025):**
 - Agregado: Generación de reportes en PDF para usuarios y reservas.
 - Mejorado: Validación de formularios en el módulo de clientes.
 - Corregido: Error al eliminar usuarios que no existían en la base de datos.
- **Versión 1.0.0 (20/04/2025):**
 - Lanzamiento inicial del sistema.
 - Funcionalidades principales:
 - Gestión de usuarios, clientes, salones, reservas, servicios y facturas.
 - Inicio de sesión con control de acceso por rol.
 - Generación de facturas en PDF.

Consideraciones para backups y recuperación

1. Backups de la base de datos:

- Realiza copias de seguridad periódicas del archivo de base de datos eventos.db para evitar la pérdida de datos.
- Puedes automatizar el proceso de backup con un script:

✓ `cp eventos.db backups/eventos_$(date +%Y%m%d).db`

2. Backups del código fuente:

- Mantén una copia del código fuente en un repositorio de control de versiones como GitHub o GitLab.
- Asegúrate de que los cambios realizados en el sistema estén correctamente versionados.

3. Recuperación en caso de fallos:

- Si el sistema falla, sigue estos pasos:
 - Restaura el archivo de base de datos desde el backup más reciente.
 - Verifica que el código fuente esté actualizado y sin errores.
 - Reinicia el servidor Flask.

4. Pruebas después de la recuperación:

- Verifica que las funcionalidades principales (inicio de sesión, gestión de datos, generación de reportes) funcionen correctamente después de la recuperación.

12. Anexos

Glosario de términos

- **Flask:** Framework de Python para el desarrollo de aplicaciones web.
- **SQLite:** Motor de base de datos ligero utilizado para almacenar usuarios, clientes, reservas, facturas, etc.
- **bcrypt:** Biblioteca para cifrar contraseñas de forma segura.
- **Session (sesión):** Mecanismo que almacena información temporal del usuario durante su navegación.
- **PDF (Portable Document Format):** Formato de archivo utilizado para reportes y facturas generadas dinámicamente.
- **Reservas:** Registros de eventos contratados por clientes.
- **Facturas:** Documentos generados para confirmar la transacción de una reserva.
- **Salones:** Espacios físicos disponibles para eventos.
- **Servicios:** Extras contratados en una reserva, como catering, decoración, sonido, etc.

Créditos

- **Desarrollador principal:** Ricardo Cano
- **Cliente:** The Grand Hall (Salón de Eventos)
- **Frameworks y librerías utilizadas:**
 - Flask
 - SQLite3
 - ReportLab (para generación de PDFs)
 - bcrypt (para seguridad de contraseñas)
- **Diseño de base de datos:** Ricardo Cano
- **Fecha de desarrollo:** Abril 2025