# **Formulario: Vista de Desarrollo del Proyecto de Software**

**Proyecto:** Reserva BHAS  
**Grupo:** 06  
**Fecha:** 04-10-2024

**1. Estructura del sistema**

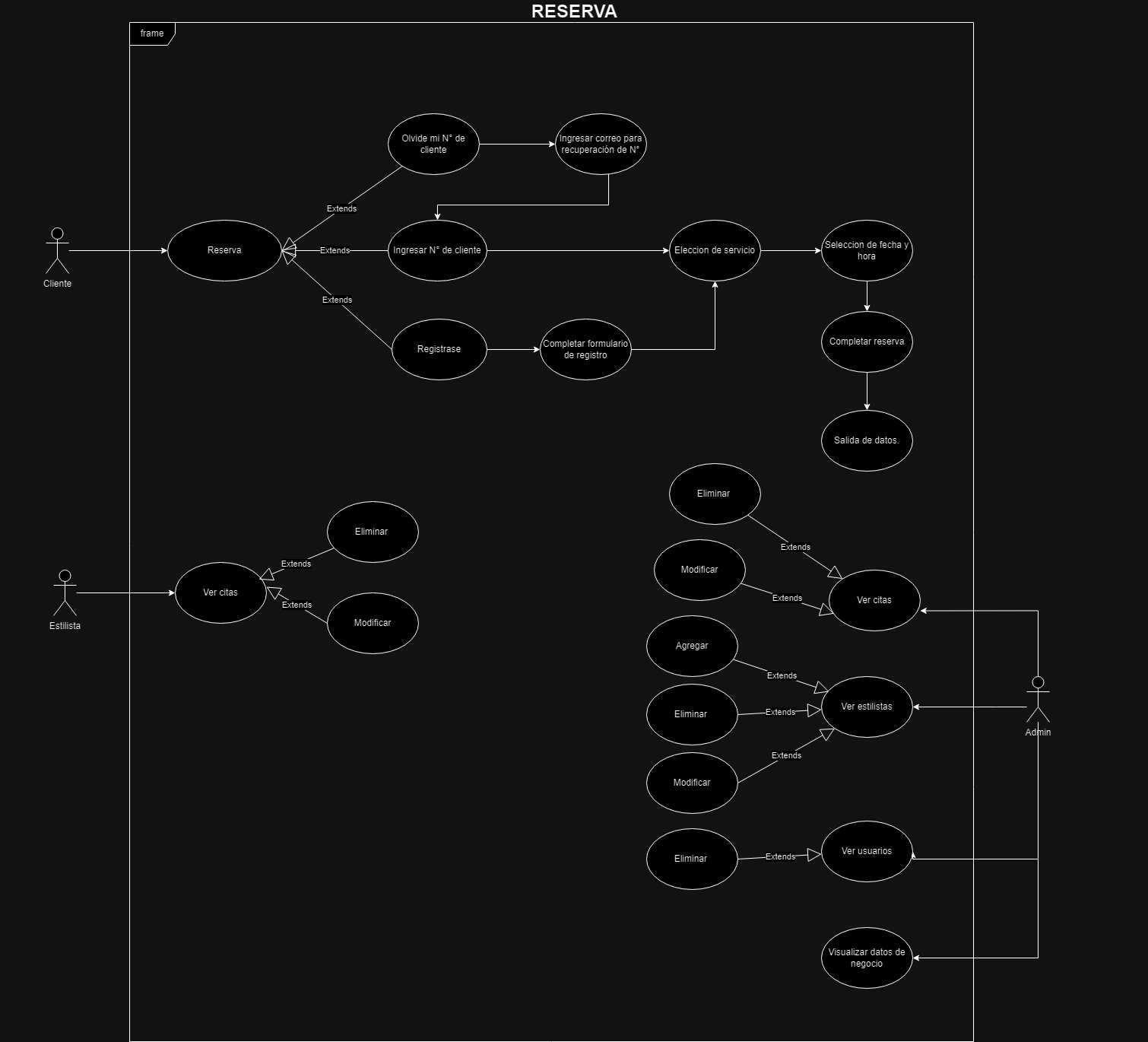
Describe la estructura del sistema de tu proyecto, mencionando los módulos principales y cómo interactúan entre sí. Puede usar diagramas de arquitectura para complementar tu descripción.

**Descripción de la estructura:**

El sistema de reserva BHAS está basado en una arquitectura de cliente-servidor. La aplicación web se compone de un front-end que interactúa con los usuarios y un back-end que gestiona la lógica de negocio y los datos. La estructura general del sistema se divide en los siguientes módulos:

* **Cliente (Front-End):** Desarrollado con HTML, CSS, y JavaScript. Se encarga de la interfaz gráfica donde los clientes pueden seleccionar estilistas, elegir servicios, y agendar citas.
* **Servidor (Back-End):** Implementado con Python y Flask. Gestiona la lógica de negocio, las consultas a la base de datos y la validación de datos del usuario. Controla las rutas y responde a las solicitudes del cliente.
* **Base de datos:** Utiliza una base de datos MySQL para almacenar información de usuarios, estilistas, servicios, y citas.
* **Integraciones:** Envío de correos electrónicos mediante un servicio de correo (por ejemplo, SMTP) para confirmaciones y recordatorios.

**Diagrama de la estructura del sistema:**   
*(Adjunta un diagrama UML o un diagrama de bloques que muestre los módulos principales y sus interacciones.)*

**

**2. Módulos principales**

Enumera y describe los módulos principales del sistema, explicando sus responsabilidades y cómo interactúan con otros módulos.

**Módulo 1 (Ingreso):  
Responsabilidad:** Gestionar el inicio de sesión y el registro de usuarios.  
**Interacción con otros módulos:** Se comunica con el módulo de base de datos para verificar credenciales y registrar nuevos usuarios.

**Módulo 2 (Reserva de citas):  
Responsabilidad:** Permitir la selección de estilistas, servicios y horas disponibles.  
**Interacción con otros módulos:** Utiliza la base de datos para obtener la disponibilidad y confirmar citas.

**Módulo 3 (Notificaciones):  
Responsabilidad:** Enviar correos de confirmación y recordatorios a los usuarios.  
**Interacción con otros módulos:** Recibe datos del módulo de reservas para programar notificaciones.

*(Agrega más módulos si es necesario)*

**3. Dependencias y Librerías**

Especifica las dependencias y librerías/frameworks que estás utilizando en tu proyecto, indicando para qué funcionalidad o módulo se usan.

**Dependencias y Librerías:**

* **Flask:** Para crear el servidor web.
* **MySQL:** Para la gestión de la base de datos.
* **Email SMTP (por ejemplo, Flask-Mail):** Para el envío de correos electrónicos de confirmación y recordatorios.

**4. Organización del Código**

Explica cómo está organizado el código en términos de carpetas y paquetes dentro del proyecto. Incluye ejemplos o un esquema de la estructura de directorios.

**Organización del código:**

* **/static:** Archivos CSS, JavaScript e imágenes.
* **/templates:** Plantillas HTML generadas dinámicamente.
* **/app.py:** Archivo principal de la aplicación que define las rutas y la lógica del servidor.

*(Opcional: Adjunta una imagen o diagrama de la estructura de directorios del proyecto.)*

**5. Reglas de Integración**

Especifica las reglas que tu equipo sigue para la integración del código entre los módulos. Esto puede incluir restricciones sobre el acceso a datos, convenciones de desarrollo, etc.

**Reglas de integración:**

Se trabajarán las pruebas del proyecto en copias apartes de la original, **cerciorándose** que funciones correctamente para que se pueda subir a GITHUB.  
Cada nueva funcionalidad debe pasar por pruebas unitarias antes de ser integradas al repositorio principal.

**6. Prácticas de desarrollo**

Detalla las prácticas de desarrollo que estás utilizando en el proyecto, como las pruebas unitarias, integración continua, revisión de código, etc.

**Prácticas de desarrollo:**

**Pruebas Unitarias:** Cada módulo se somete a pruebas específicas para garantizar la funcionalidad.

**Revisión de Código:** El equipo revisa el código de cada miembro antes de hacer un merge en la rama principal.

**Control de versiones:** Uso de Git para mantener el control de versiones y la colaboración en equipo.

**7. Herramientas de desarrollo**

Mencione las herramientas que está utilizando para el desarrollo del proyecto (por ejemplo, IDE, sistemas de control de versiones, etc.).

**Herramientas:**

**Visual Studio Code (VS Code):** Editor de código principal.

**GitHub:** Para el control de versiones y la colaboración.

**MySQL:** Base de datos utilizada para almacenar los datos del sistema.

**8. Diagrama de Arquitectura del Sistema**

Adjunta un diagrama de arquitectura general del sistema que incluye los módulos, las interacciones entre ellos y las tecnologías involucradas.

**Diagrama de arquitectura general:**   
*(Adjunta el diagrama en esta sección)*

