

DuocUC



DOCUMENTO DAS ARQUITECTURA DE SOFTWARE PROYECTO

Documento de Arquitectura de Software IEEE-1477-2000 Control del Documento

Nombre del Proyecto: Rhea Reserve
Arquitectura Proyecto Semestral
Integrantes: Franco Fuentes / Miguel Garate

Contenido

No se encontraron entradas de tabla de contenido.

1. Introducción

El presente informe tiene como objetivo describir en detalle la arquitectura de un sistema denominado **Rhea** diseñado para automatizar la administración de reservas y procesos operativos comunes dentro de pequeñas y medianas empresas del sector de servicios, tales como barberías y salones de belleza.

El sistema propuesto busca mejorar la eficiencia operativa, permitiendo a los negocios gestionar de manera óptima sus reservas, horarios de empleados, servicios y clientes, con el fin de optimizar los recursos y elevar la satisfacción del cliente.

El sistema desarrollado será implementado en una plataforma web, permitiendo acceso desde cualquier dispositivo con conexión a internet. El sistema también proporcionará una interfaz intuitiva y accesible para todos los usuarios (clientes, empleados y dueños), asegurando que las gestiones se realicen de manera eficiente y organizada, basada en los estándares de usabilidad y seguridad.

2. Propósito

El propósito de este documento es servir como guía para el diseño e implementación de la arquitectura del sistema Rhea. En él se detallan las decisiones técnicas, los componentes principales y los puntos de integración que permitirán implementar una solución robusta y escalable, capaz de gestionar múltiples negocios bajo un esquema de administración centralizado.

Este informe es esencial para los desarrolladores y stakeholders del proyecto, ya que proporciona una visión detallada de la estructura y comportamiento del sistema, así como las relaciones entre las diferentes funcionalidades y componentes. Asimismo, establece las bases para futuras expansiones y mejoras del sistema, considerando factores de disponibilidad, seguridad y usabilidad.

3. Alcance

El sistema Rhea gestionará las siguientes áreas:

- Gestión de reservas, permitiendo a los clientes seleccionar servicios, profesionales y horarios.
- Administración de negocios, donde los dueños pueden configurar detalles de su negocio, servicios, horarios y empleados.

- Roles y permisos, controlando el acceso a funcionalidades específicas según el rol de usuario.
- Pagos y notificaciones, facilitando opciones de pago en línea y recordatorios automáticos a clientes.
- Generación de reportes y estadísticas, brindando a los dueños datos sobre reservas, ingresos y rendimiento.

4. Equipo de Trabajo-Actores del Desarrollo

El desarrollo del sistema requiere la participación de un equipo multidisciplinario con roles claramente definidos para asegurar que el proyecto se lleve a cabo de manera eficiente. A continuación, se presenta una tabla con los principales roles y sus responsabilidades:

Rol	Responsabilidades
Cliente-Stakeholder	Representa las necesidades y expectativas de los negocios (principalmente barberías).
Product Owner	Define las características y prioridades del sistema.
Desarrolladores Backend	Implementan la lógica de negocio, gestionan el servidor y la base de datos.
Desarrolladores Frontend	Diseñan la interfaz de usuario y se encargan de la experiencia del cliente.
DBA-Especialista en Base de Datos	Configura y optimiza la base de datos para un rendimiento eficiente.
Tester/QA	Realiza pruebas para asegurar que el sistema funcione sin errores.
Analista de Requisitos	Identifica y documenta las necesidades funcionales y no funcionales.
Administrador Plataformas	Administra la infraestructura y servicios necesarios para el despliegue.

5. Recomendaciones de conformidad con esta práctica.

Para asegurar que la arquitectura del sistema cumpla con los requisitos y estándares establecidos, se recomiendan las siguientes prácticas:

- **Revisiones de Arquitectura:** Implementar revisiones de arquitectura cada sprint para ajustar los requisitos a cambios o mejoras.

- **Patrón de Diseño Desacoplado:** Utilizar patrones de diseño como **MVC** (Model-View-Controller) para asegurar modularidad y escalabilidad.
- **Metodologías Ágiles:** Adoptar **Scrum** para la entrega continua de mejoras y funcionalidades, lo que permitirá responder a cambios en las necesidades de negocio.

6. Referencias y estándares aplicables a este documento:

El documento sigue los siguientes estándares de calidad para asegurar la robustez y escalabilidad del sistema:

- **IEEE 830-1998 ST:** Estandarización de requisitos de software.
- **ISO 9126-2001:** Métricas de calidad del software.
- **Modelo 4+1 de Kruchten:** Modelo estructural basado en múltiples vistas de escenarios.
- **ATAM (Architecture Tradeoff Analysis Method):** Metodología para evaluar y documentar decisiones arquitectónicas

7. Definiciones, acrónimos y abreviaciones.

Término	Definición
DAS	Documento de Arquitectura de Software
MVC	Modelo-Vista-Controlador, patrón de diseño utilizado para separar la lógica de negocio de la interfaz de usuario
UML	Unified Modeling Language, lenguaje gráfico para representar la arquitectura del sistema
CRUD	Operaciones básicas de Create, Read, Update, Delete para la gestión de datos.

8. DESCRIPCIÓN DE ARQUITECTURA:

La arquitectura del sistema se presenta mediante el uso de vistas basadas en el Modelo 4+1 de Kruchten, con el fin de proporcionar una visión completa y

estructurada del sistema desde diferentes perspectivas. La arquitectura del sistema se presenta mediante el uso de vistas basadas en el Modelo 4+1 de Kruchten, con el fin de proporcionar una visión completa y estructurada del sistema desde diferentes perspectivas.

8.1 VISTAS:

Cada vista se representará en un diagrama UML para describir la estructura y comportamiento del sistema.

8.2 TIPOS DE VISTAS:

1. **Vista de Escenarios (Diagrama de Caso de Uso):** Describe las funcionalidades del sistema y cómo interactúan los diferentes actores (clientes, empleados y dueños) con él.
2. **Vista Lógica (Diagrama de Clases):** Detalla la estructura de datos y las relaciones entre las principales entidades del sistema (como reservas, servicios y empleados).
3. **Vista de Procesos (Diagrama de Actividad):** Muestra el flujo de actividades dentro del sistema, como el proceso de reservar un servicio.
4. **Vista de Despliegue (Diagrama de Componentes):** Representa la distribución de componentes en la infraestructura física, como el servidor, la base de datos y el cliente.
5. **Vista Física (Diagrama Topológico):** Representa la infraestructura de red y la comunicación entre los diferentes servicios y usuarios del sistema.

8.3 Framework Conceptual

8.3.1 Descripción de la arquitectura en contexto

El sistema **Rhea** será diseñado y desarrollado utilizando un enfoque ágil basado en **Scrum**, lo que permitirá la entrega incremental de funcionalidades, retroalimentación constante y la capacidad de adaptarse a cambios en los requisitos durante el ciclo de vida del proyecto. Este enfoque garantiza que las necesidades emergentes puedan ser abordadas y maximiza el valor del producto para los usuarios.

El diseño se basará en la separación de vistas, ofreciendo diferentes perspectivas de la arquitectura para una comprensión integral:

- **Vista de Escenarios:** Diagrama de Casos de Uso, que describe las interacciones de los usuarios (clientes, empleados, dueños) con el sistema y sus principales flujos de acción.
- **Vista Lógica:** Incluye el Modelo de Clases para entidades y relaciones y el Modelo

de Datos para la estructura de la base de datos.

- **Vista de Procesos:** Diagrama de Actividad para mostrar flujos internos y módulos interactivos.
- **Vista de Despliegue:** Diagrama de Componentes para representar la distribución de la lógica de negocio, interfaz de usuario y acceso a datos.
- **Vista Física:** Diagrama Topológico para mostrar la distribución del sistema en el entorno físico, como servidores, redes y componentes de infraestructura.

Estas vistas están interconectadas para asegurar la coherencia y alineación con los requisitos de negocio, brindando una perspectiva holística de la arquitectura del sistema.

El estilo arquitectónico se centrará en una combinación de patrones orientados a servicios y arquitectura por capas, donde cada módulo del sistema corresponde a casos de uso específicos que resuelven procesos definidos en la operación.

Procesos Clave:

1. **Gestión de Reservas:** Administración de reservas y horarios.
2. **Gestión de Servicios y Empleados:** Configuración de servicios y personal.
3. **Monitoreo y Seguridad del Sistema:** Gestión de acceso y control de actividades.

8.3.1 Actores Usuarios y sus roles:

Este documento representa la identificación de Actores/Usuarios Stakeholders y sus roles a partir de la interpretación de los casos de uso del Negocio asociados.

ACTOR	ROL / CARGO	FUNCION
Cliente	Usuario Final	Realiza reservas, selecciona servicios y profesionales, recibe notificaciones y confirma citas.
Empleado	Profesional del Negocio	Consulta la agenda de reservas del día, actualiza disponibilidad y modifica citas en casos específicos.
Dueño	Administrador del Negocio	Configura y gestiona el negocio, crea y edita servicios, define horarios, y gestiona empleados y reportes.
Product Owner	Representante del Cliente	Define los requisitos y prioridades del sistema, asegura el alineamiento con las necesidades del negocio.
Desarrollador	Backend y Frontend	Desarrolla y mantiene las funcionalidades y la interfaz del sistema, asegurando escalabilidad y usabilidad.
Tester QA	Control de Calidad	Realiza pruebas funcionales y de integración para asegurar que el sistema

		cumple con los requisitos establecidos.
Administrador de Plataformas	Infraestructura	Gestiona los servidores y entornos de despliegue para asegurar disponibilidad y rendimiento óptimo.

8.3.2 Actividades de arquitectura en el ciclo de vida

El desarrollo de la arquitectura del sistema **Rhea** se aborda mediante un ciclo de vida iterativo incremental basado en **Scrum**. Esta metodología permite entregas parciales y frecuentes de software funcional, facilitando la validación temprana de los requisitos por parte de los stakeholders y el product owner, adaptándose a cambios en el entorno de negocio.

Cada sprint tiene una duración de 2 a 3 semanas, centrado en la construcción y mejora de un módulo específico del sistema. Al finalizar cada sprint, se realiza una revisión detallada de los resultados para asegurar la alineación con las expectativas del cliente y priorizar las tareas para el siguiente sprint.

8.3.3 Descripciones prácticas de arquitectura.

Durante el proceso de desarrollo, la arquitectura del sistema **Rhea** fue escalada desde un escenario principal de caso de uso hasta un diseño modular a través de los siguientes pasos:

1. **Análisis de Requerimientos:**
 - a. Se analizaron los requisitos iniciales del sistema para identificar los casos de uso principales, tales como "**Gestionar Reservas**", "**Configurar Negocio**" y "**Administrar Pagos**".
 - b. Este análisis permitió definir las funcionalidades clave para cada tipo de usuario (clientes, empleados y dueños) y sentó las bases para una arquitectura que soporte la operación diaria del negocio y la interacción con los clientes.
2. **Definición de la Vista Lógica:**
 - a. A partir del escenario principal, se identificaron las **entidades y relaciones clave** en un **diagrama de clases UML**, representando el núcleo del sistema.
 - b. Las entidades principales incluyen **Reservas**, **Servicios**, **Clientes**, **Empleados** y **Pagos**, estableciendo una estructura de datos coherente para la interacción entre estos elementos.
3. **Modelo de Datos:**
 - a. Se desarrolló un **diagrama de base de datos** para definir la estructura de la información de manera precisa, alineada con la vista lógica.
 - b. Este modelo asegura que las entidades fundamentales estén conectadas de manera óptima, con relaciones entre negocios, servicios, empleados y

clientes, permitiendo consultas y operaciones eficientes.

4. **Vista de Procesos:**

- a. Se modelaron los flujos de las actividades y transacciones principales del sistema mediante **diagramas de actividad UML**.
- b. Estos diagramas reflejan procesos como la reserva de citas, la modificación de disponibilidad y la gestión de pagos, ofreciendo una representación clara de las interacciones entre componentes y actores.

5. **Despliegue:**

- a. Se definió la infraestructura física y lógica para alojar el sistema en la nube, asegurando disponibilidad y escalabilidad.
- b. La configuración de despliegue incluye **servidores de aplicaciones, bases de datos distribuidas**, y elementos de balanceo de carga para gestionar el tráfico de usuarios.
- c. El sistema fue diseñado para operar en una arquitectura de capas, separando la lógica de negocio, el acceso a datos y la interfaz de usuario.

6. **Definición de Planes de Pruebas y Técnicas de Testing:**

- a. Se establecieron planes de pruebas para validar el correcto funcionamiento del sistema. Estas pruebas incluyen **pruebas unitarias, pruebas de integración y pruebas de usabilidad**.
- b. Además, se desarrollaron **pruebas de carga** para evaluar la capacidad del sistema de manejar múltiples reservas y usuarios simultáneos, y **pruebas de seguridad** para garantizar la protección de los datos de los clientes y transacciones de pago.

8.4 Documentación de la arquitectura

- **ISO/IEC 25010:** Se utilizó este estándar para evaluar la calidad del sistema en términos de **funcionalidad, seguridad, usabilidad y eficiencia**, asegurando que **Rhea** cumpla con altos niveles de desempeño y protección de datos en el sector de servicios.
- **IEEE 1471-2000:** Proporcionó la guía para la **representación de vistas arquitectónicas** y la **identificación de stakeholders** clave, facilitando la alineación de la arquitectura con los objetivos de negocio y necesidades de usuarios, dueños y empleados.
- **Modelo 4+1 de Kruchten:** Estructura la arquitectura del sistema en cinco vistas principales (Escenarios, Lógica, Proceso, Despliegue y Física), permitiendo visualizar **Rhea** desde múltiples perspectivas y garantizando una implementación modular y escalable.
- **Planes de Prueba:**
 - **Pruebas Unitarias:** Verificación de las funciones individuales, como la creación de reservas, el procesamiento de pagos y la configuración de servicios, asegurando que cada componente funcione correctamente de forma independiente.

- **Pruebas de Integración:** Evaluación de la interacción entre módulos, como la sincronización entre la reserva de citas y la disponibilidad de empleados, para confirmar que los componentes funcionan correctamente en conjunto.
- **Pruebas Funcionales:** Pruebas de las funcionalidades clave desde el punto de vista del usuario, como la reserva y modificación de citas, para asegurar que el sistema cumple con los requisitos establecidos.
- **Pruebas de Usabilidad:** Evaluación de la interfaz para asegurar que los usuarios puedan interactuar con el sistema de manera intuitiva y eficiente.
- **Pruebas de Seguridad:** Verificación de la protección de datos sensibles y control de accesos, asegurando que el sistema previene accesos no autorizados y protege la información de clientes y negocios.
- **Pruebas de Carga:** Pruebas de rendimiento bajo múltiples usuarios y reservas simultáneas para garantizar que el sistema responde de manera eficiente bajo demanda elevada.

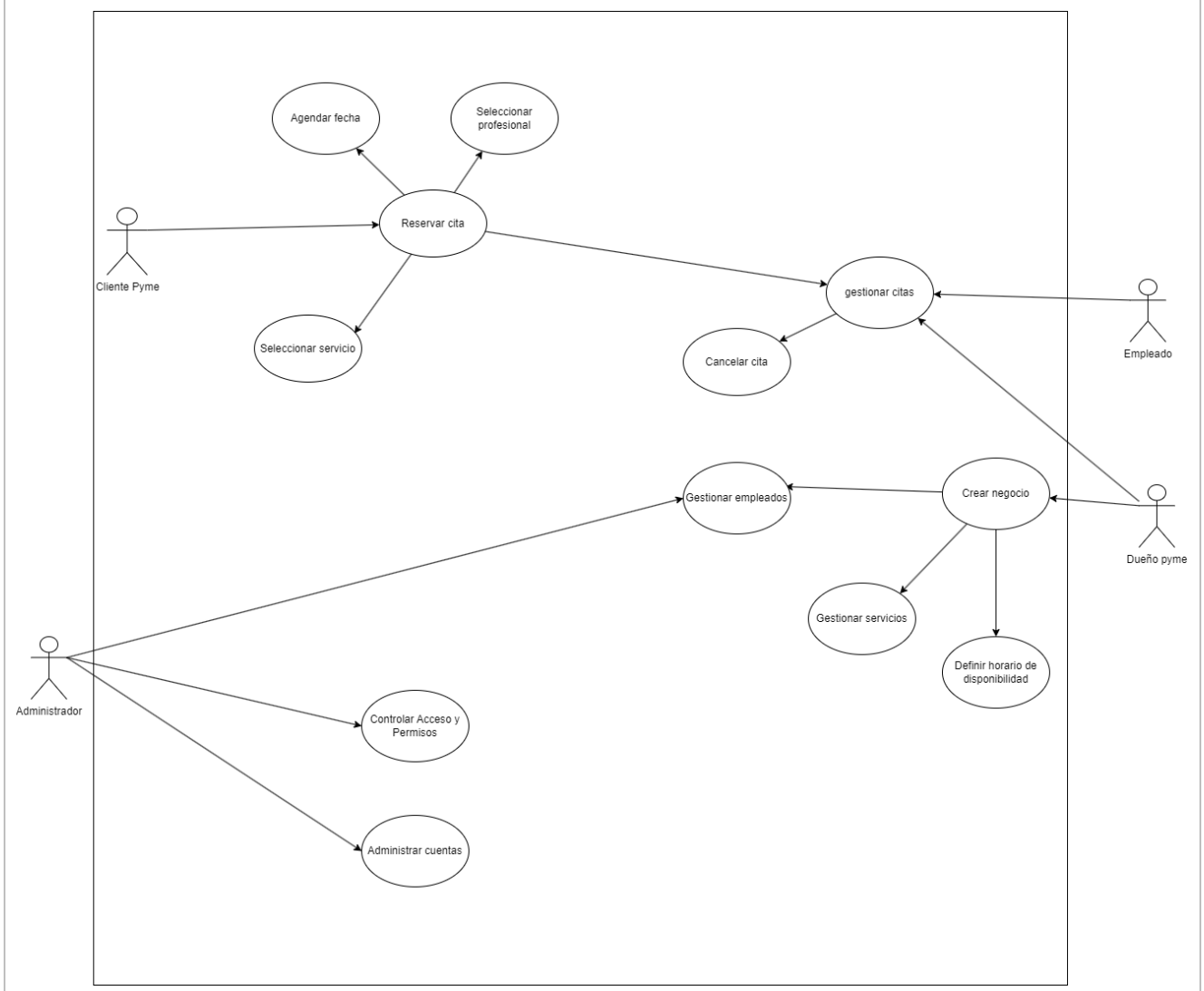
8.5 Selección de puntos de vista de la arquitectura

VISTAS	UML	Cantidad
Escenario Principal	Diagrama de Caso de uso	1
Vista Lógica	Diagrama de Clases MER	1
Vista de Proceso	Diagrama de Actividad:	3
Vista Física	Diagrama Topológico de Servicios	1
Vista de Despliegue	Diagrama de componentes Diagrama de Capas (MVC)	1

8.6 Vistas de la arquitectura

UML-VISTA DE ESCENARIO

Diagrama de Caso de uso extendido /Subproyecto [Rhea Reserve]



Gestión de Negocios: Permite al dueño del negocio crear, modificar y configurar los detalles del negocio, como nombre, dirección y descripción. También incluye la gestión de empleados, horarios, y servicios específicos ofrecidos (corte de cabello, tratamientos, etc.).

Gestión de Usuarios y Roles: Facilita la creación y administración de los diferentes roles de usuarios dentro del sistema: clientes, empleados y dueños. Cada rol tiene permisos específicos según las funcionalidades a las que puede acceder.

Notificación de Reservas: El sistema envía automáticamente notificaciones por correo electrónico o WhatsApp a los clientes sobre sus reservas confirmadas, recordatorios de citas y notificaciones de cambios en su estado de reserva, asegurando que estén al tanto de sus próximas citas.

Visualización de Historial de Reservas: Los clientes pueden acceder a su cuenta para ver el historial de reservas, permitiéndoles revisar citas pasadas y futuras, y realizar modificaciones o cancelaciones antes de la fecha límite.

Gestión de Pagos: El sistema permite a los clientes realizar pagos en línea o seleccionar la opción de pago en tienda. Para los pagos en línea, se utilizan métodos de pago seguros como Transbank para procesar las transacciones de manera eficiente.

Aplicación de Políticas de Cancelación: Si un cliente cancela una reserva después de la fecha límite o incumple con las políticas de cancelación del negocio, el sistema puede aplicar cargos automáticos o limitar futuras reservas en función de la política del negocio.

Reserva de Servicios y Profesionales:

Los clientes pueden consultar la disponibilidad de servicios y profesionales en tiempo real, seleccionando el servicio y profesional deseado, así como el horario conveniente. Esto asegura una mejor experiencia para el cliente y optimiza la ocupación de los empleados.

Generación de Informes y Reportes: El sistema permite al dueño generar informes sobre las reservas realizadas, los ingresos generados y la eficiencia de los empleados. Los reportes pueden ser diarios, semanales o mensuales, proporcionando una visión general de la operación del negocio.

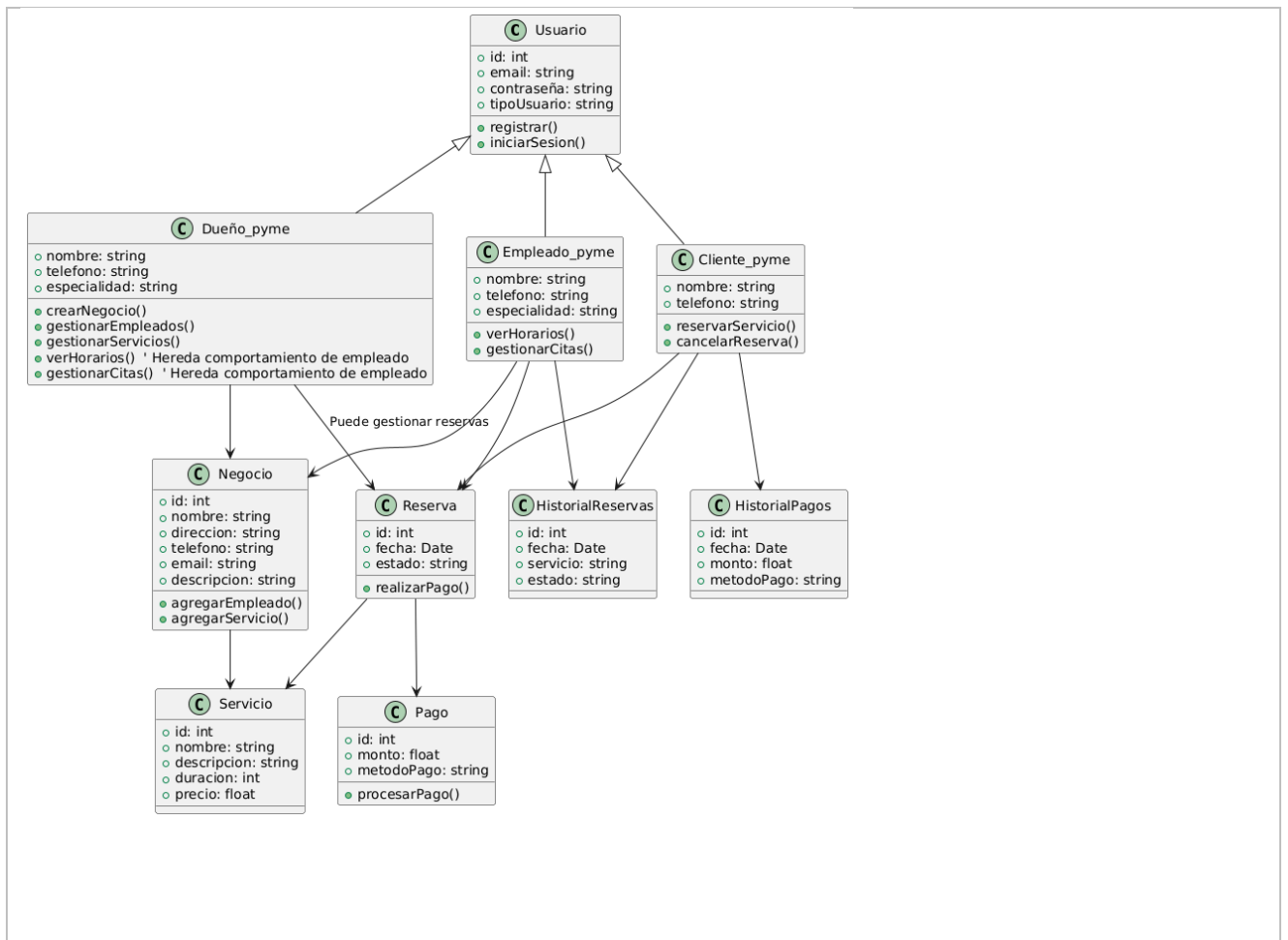
Alertas Automáticas: El sistema envía notificaciones automáticas sobre eventos importantes, como recordatorios de reserva, confirmaciones de pago, o alertas de citas que han sido canceladas por el cliente.

Gestión de Permisos y Acceso: El dueño del negocio puede controlar el acceso de los empleados y asignar permisos específicos según su rol, asegurando que solo accedan a las funcionalidades necesarias para su trabajo (ej. solo el dueño puede ver reportes financieros).

Registro de Clientes Frecuentes y Fidelización: El sistema mantiene un registro de los clientes frecuentes, permitiendo al negocio identificar y ofrecer beneficios de fidelización a los clientes recurrentes, mejorando la relación cliente-negocio.

UML-VISTA LÓGICA-DIAGRAMA DE CLASES

[Diagrama de Clases]



El Diagrama de Clases para el sistema de gestión de reservas **Rhea** muestra las clases involucradas, sus atributos, métodos y las relaciones entre ellas. Este diagrama modela la estructura y el comportamiento del sistema de una manera orientada a objetos, facilitando la organización y gestión de servicios, usuarios, reservas y pagos para negocios en el sector de servicios.

1. Usuario

a. Atributos:

- i. id, email, contraseña, tipoUsuario

b. Métodos:

- i. registrar(), iniciarSesion()

2. Dueño_pyme

a. Atributos:

- i. nombre, telefono, especialidad

b. Métodos:

- i. crearNegocio(), gestionarEmpleados(), gestionarServicios(), verHorarios(), gestionarCitas()

3. Empleado_pyme

a. Atributos:

- i. nombre, telefono, especialidad

b. Métodos:

- i. verHorarios(), gestionarCitas()

4. Cliente_pyme

- a. **Atributos:**
 - i. nombre, telefono
 - b. **Métodos:**
 - i. reservarServicio(), cancelarReserva()
- 5. Negocio**
- a. **Atributos:**
 - i. id, nombre, dirección, telefono, email, descripción
 - b. **Métodos:**
 - i. agregarEmpleado(), agregarServicio()
- 6. Servicio**
- a. **Atributos:**
 - i. id, nombre, descripcion, duracion, precio
 - b. **Métodos:**
 - i. Sin métodos específicos definidos en este nivel, ya que los métodos de gestión están en **Negocio**.
- 7. Reserva**
- a. **Atributos:**
 - i. id, fecha, estado
 - b. **Métodos:**
 - i. realizarPago()
- 8. Pago**
- a. **Atributos:**
 - i. id, monto, metodoPago
 - b. **Métodos:**
 - i. procesarPago()
- 9. HistorialReservas**
- a. **Atributos:**
 - i. id, fecha, servicio, estado
 - b. **Métodos:**
 - i. Sin métodos específicos en este nivel.
- 10. HistorialPagos**
- a. **Atributos:**
 - i. id, fecha, monto, metodoPago
 - b. **Métodos:**
 - i. Sin métodos específicos en este nivel.

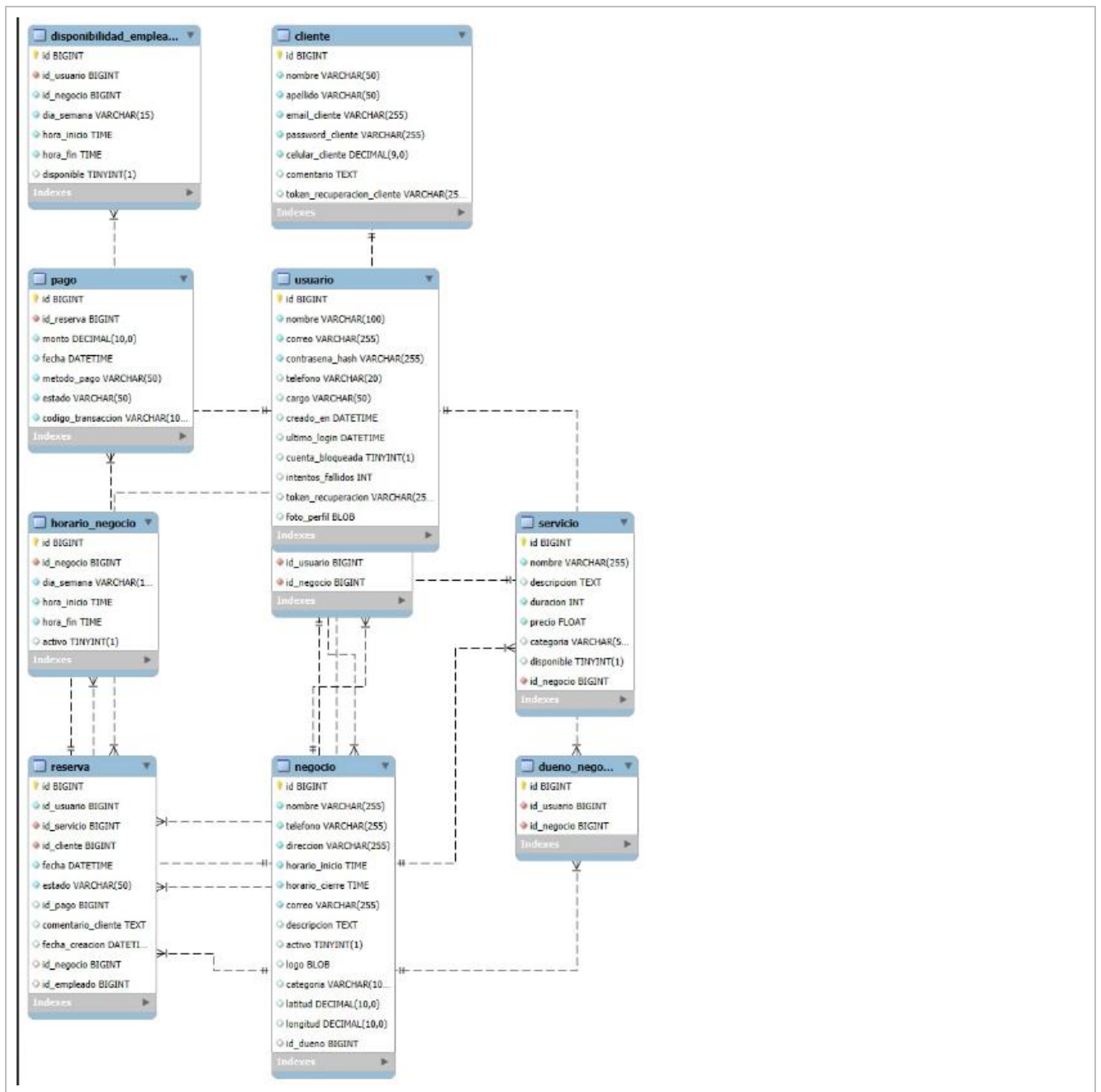
Relaciones:

- **Usuario** tiene una relación de herencia con **Dueño_pyme**, **Empleado_pyme** y **Cliente_pyme** (el usuario puede tener diferentes roles dentro del sistema).
- **Dueño_pyme** tiene una relación de asociación con **Negocio** (un dueño administra un negocio y gestiona empleados y servicios).
- **Negocio** tiene una relación de composición con **Servicio** y **Reserva** (un negocio posee varios servicios y realiza reservas).
- **Cliente_pyme** está asociado con **Reserva** (un cliente puede realizar múltiples reservas de servicios).
- **Reserva** tiene una relación de asociación con **Pago** y **HistorialReservas** (cada reserva está vinculada a un historial y un pago).

- **HistorialReservas** y **HistorialPagos** están asociados con el **Usuario** (cada usuario puede consultar su historial de reservas y pagos).

UML-VISTA LÓGICA-DIAGRAMA DE BASE DE DATOS

[Diagrama de BD - MER]



El Modelo Entidad-Relación (MER) para el sistema de gestión de reservas **Rhea** representa las entidades involucradas, sus atributos y las relaciones entre ellas. Este modelo permite visualizar la organización de la información relacionada con la administración de reservas, pagos, usuarios y servicios dentro de un negocio en el sector de servicios.

2. Elementos Principales

- Entidades:
 - Usuario

- **Atributos:** ID_Usuario (PK), Nombre, Correo, Contraseña, Cargo, Último Login, Cuenta Bloqueada, Intentos Fallidos, Foto_Perfil
- **Relación:** Está relacionado con las entidades **Cliente**, **Empleado_Pyme** y **Dueño_Negocio**.

≠ **Cliente**

- **Atributos:** ID_Cliente (PK), Nombre, Apellido, Email, Password, Celular, Comentario, Token_Recuperacion
- **Relación:** Está relacionado con Reserva y puede realizar múltiples reservas en el sistema.

○ **Dueño_Negocio**

- **Atributos:** ID_Dueño (PK), ID_Usuario (FK), ID_Negocio (FK)
- **Relación:** Está relacionado con **Negocio** y gestiona los servicios, empleados y reservas del negocio.

○ **Negocio**

- **Atributos:** ID_Negocio (PK), Nombre, Dirección, Teléfono, Email, Descripción, Horario de Atención, Latitud, Longitud
- **Relación:** Tiene una relación de composición con **Servicio** y **Empleado_Pyme**. También está relacionado con **Horario_Negocio** para definir su disponibilidad.

○ **Servicio**

- **Atributos:** ID_Servicio (PK), Nombre, Descripción, Duración, Precio, Categoría
- **Relación:** Está vinculado a **Negocio** y puede ser seleccionado en una **Reserva**.

○ **Empleado_Pyme**

- **Atributos:** ID_Empleado (PK), ID_Usuario (FK), ID_Negocio (FK)
- **Relación:** Relacionado con **Disponibilidad_Empleado** para definir su horario y con **Reserva** para gestionar las citas.

○ **Horario_Negocio**

- **Atributos:** ID_Horario (PK), ID_Negocio (FK), Día de la Semana, Hora Inicio, Hora Fin, Activo
- **Relación:** Define la disponibilidad del negocio para cada día de la semana.

○ **Disponibilidad_Empleado**

- **Atributos:** ID_Disponibilidad (PK), ID_Usuario (FK), Día de la Semana, Hora Inicio, Hora Fin, Disponible
- **Relación:** Define la disponibilidad de los empleados para cada día de la semana.

○ **Reserva**

- **Atributos:** ID_Reserva (PK), ID_Usuario (FK), ID_Cliente (FK), ID_Servicio (FK), Fecha, Estado, ID_Pago, Comentario Cliente, Fecha Creación
- **Relación:** Vinculado con **Cliente**, **Empleado_Pyme** y **Pago** para gestionar las citas y el pago asociado.

○ **Pago**

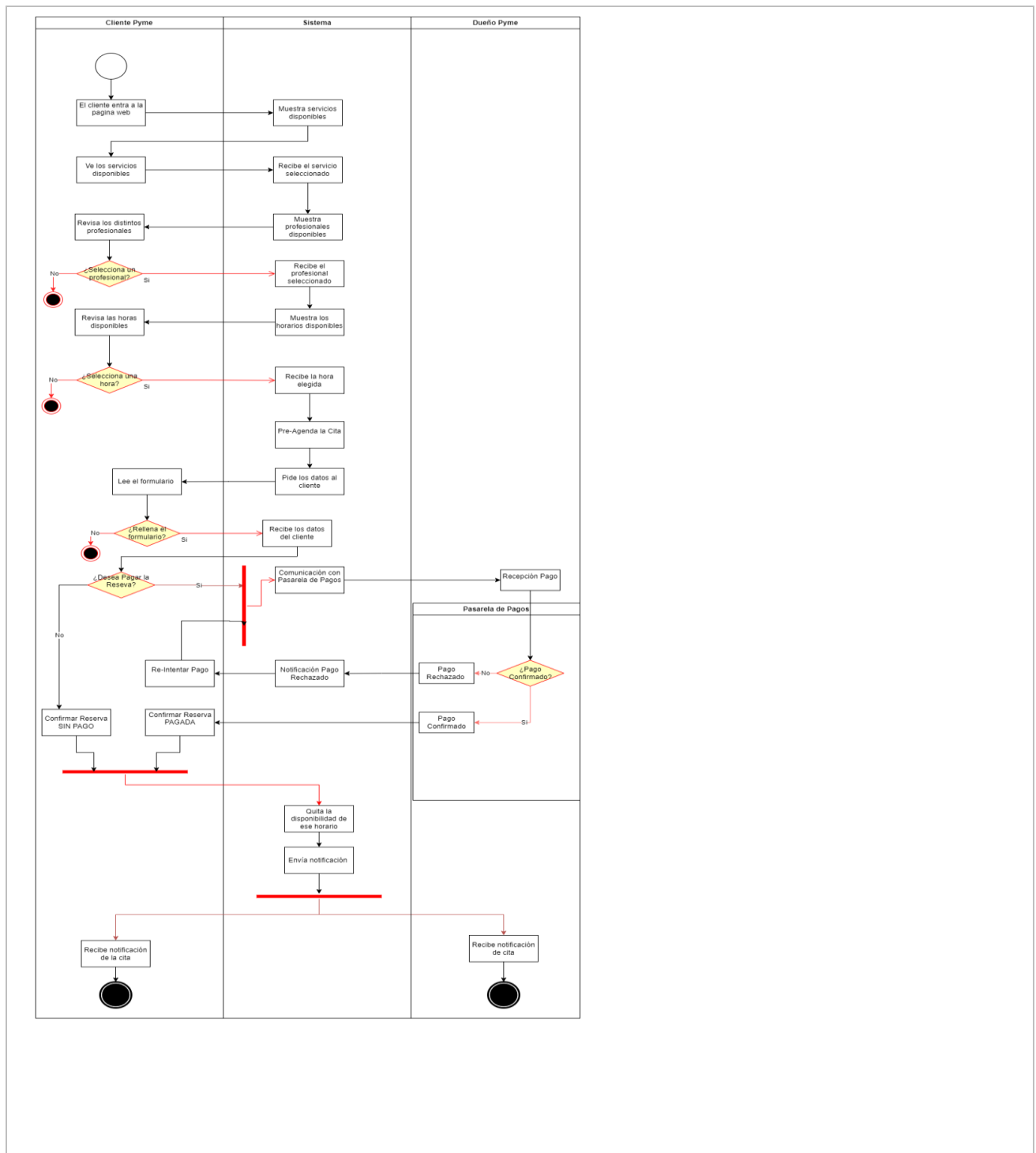
- **Atributos:** ID_Pago (PK), ID_Reserva (FK), Monto, Fecha, Método de Pago, Estado, Código Transacción
- **Relación:** Relacionado con **Reserva** para gestionar el pago de una cita específica.

- **Relaciones:**

- **Usuario** está relacionado con las entidades **Cliente**, **Empleado_Pyme** y **Dueño_Negocio** según su rol en el sistema.
- **Negocio** tiene una relación de composición con **Servicio** y **Empleado_Pyme** (un negocio posee varios servicios y empleados).
- **Cliente** puede realizar múltiples **Reservas**, que están vinculadas a un **Servicio** específico y gestionadas por un **Empleado_Pyme**.
- **Reserva** tiene una relación de asociación con **Pago**, permitiendo el procesamiento de pagos para cada cita reservada.
- **Horario_Negocio** y **Disponibilidad_Empleado** definen los horarios y la disponibilidad de los empleados para gestionar citas.

UML-VISTA DE PROCESO-DIAGRAMAS DE ACTIVIDAD

[Diagramas de Actividad]



El diagrama de actividad representa el flujo de actividades que se llevan a cabo en el proceso de **reserva de cita y pago** en el sistema **Rhea**. Este diagrama es útil para visualizar los pasos secuenciales y las decisiones que el cliente debe tomar al reservar un servicio y optar por pagar en línea o en el local.

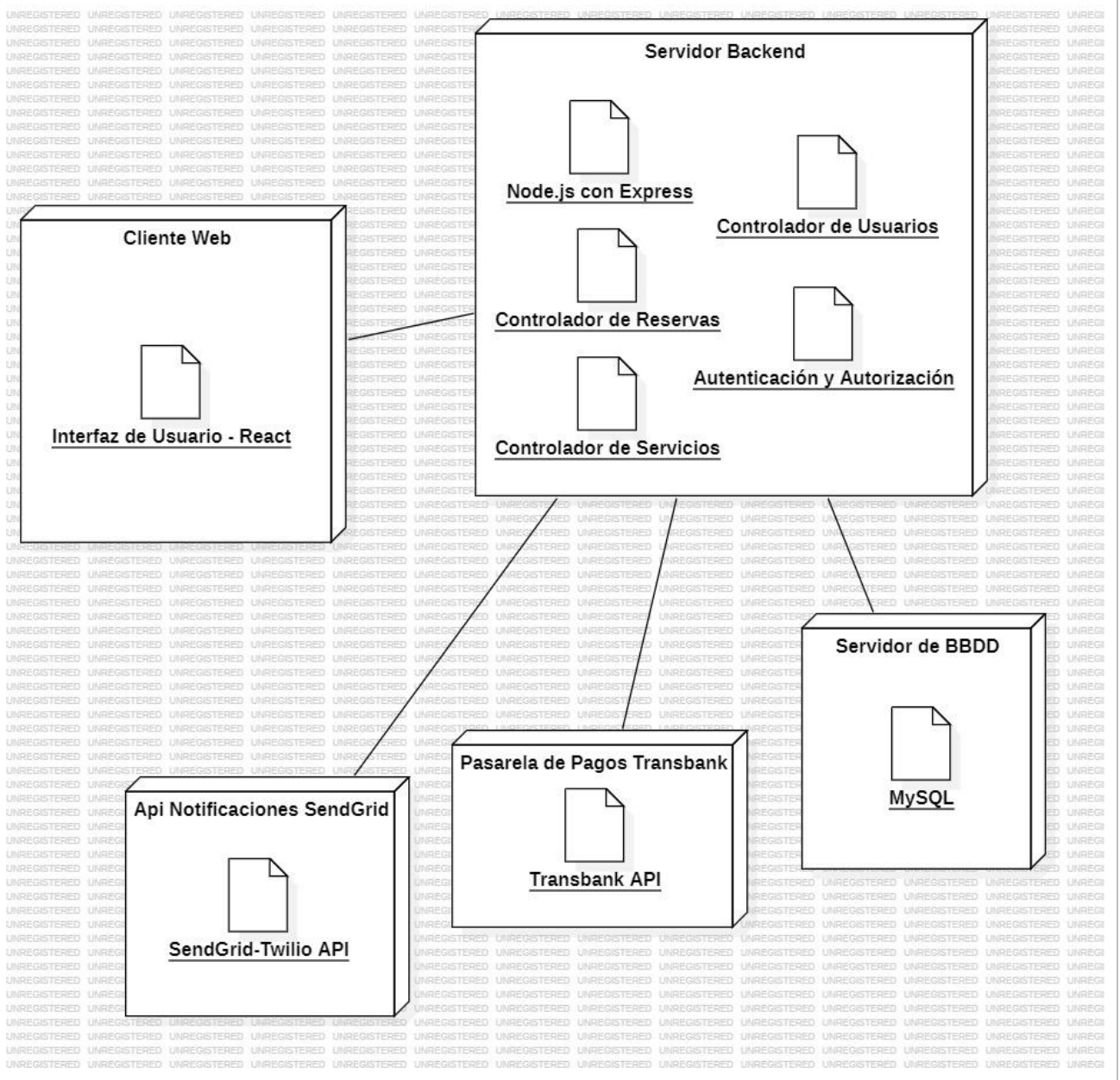
Componentes:

1. **Inicio:** El cliente accede a la página web de Rhea y comienza el proceso de reserva.

2. **Selección del Servicio:** El sistema muestra los servicios disponibles, y el cliente selecciona el servicio deseado.
3. **Selección del Profesional:** El sistema muestra los profesionales disponibles para el servicio seleccionado, y el cliente elige uno.
4. **Selección de la Hora:** El sistema muestra los horarios disponibles, y el cliente selecciona una hora para la cita.
5. **Formulario de Reserva:** El cliente revisa y completa el formulario de reserva con sus datos necesarios.
6. **Opciones de Pago:** El cliente elige si desea realizar el pago en línea o pagar en el local.
 - **Si elige pagar en línea:**
 - El sistema se comunica con la pasarela de pagos para procesar el pago.
 - **Verificación del Pago:**
 - Si el pago es exitoso, se confirma la reserva y se envía una notificación.
 - Si el pago falla, el sistema notifica al cliente y le da la opción de reintentar.
 - **Si elige pagar en el local:**
 - La reserva se confirma sin pago, y se envía una notificación al cliente y al dueño del negocio.
7. **Actualización del Estado de Reserva:** Una vez que el pago es exitoso o la reserva se confirma sin pago, el sistema actualiza la disponibilidad del horario elegido.
8. **Notificación de Confirmación:** El sistema envía una notificación al cliente y al profesional seleccionado, confirmando la reserva y los detalles de la cita.
9. **Fin:** El proceso finaliza con la reserva confirmada y, si corresponde, el pago realizado y registrado en el sistema.

UML-VISTA DE DESPLIEGUE

Sistema de Gestión de Condominios



Cliente Web: Interfaz de Usuario (Frontend): Permite a los clientes, dueños y empleados acceder al sistema Rhea a través de un navegador. La interfaz, desarrollada en React.js o Vue.js, facilita la selección de servicios, la reserva de citas y la administración de configuraciones de negocio.

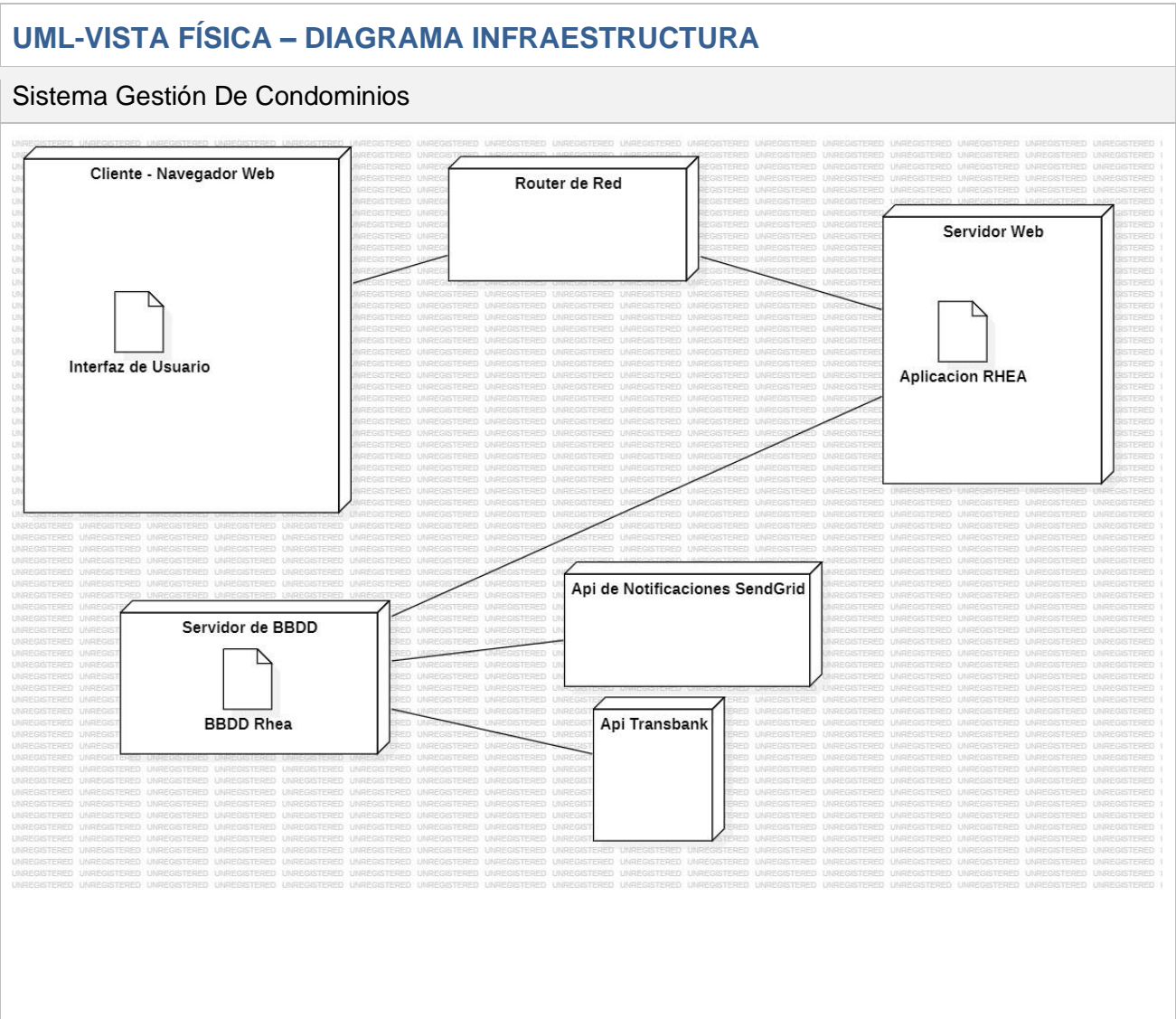
Servidor de Aplicaciones Aplicación Web Rhea: Aloja la lógica de negocio y la API de la aplicación, gestionando solicitudes para reservas, servicios, y horarios. Este backend en Node.js con Express conecta los diferentes componentes del sistema y procesa las interacciones de los usuarios.

Servidor de Base de Datos Base de Datos Rhea: Almacena información esencial, como datos de clientes, reservas, servicios y pagos, permitiendo la consulta y actualización en

tiempo real para gestionar el negocio de manera eficiente.

Servidor de Notificaciones: Sistema de Notificación: Envía recordatorios, confirmaciones de citas y alertas de pagos a través de servicios externos como SendGrid o Twilio, asegurando una comunicación oportuna con los usuarios.

Servidor de Transbank: Pasarela de Pago: Procesa los pagos en línea de las reservas. El servidor se comunica con Transbank para validar las transacciones, asegurando que los clientes puedan pagar sus citas de manera segura y que el sistema registre el estado de los pagos.



1. **Centro de Datos Principal:**
 - a. Es el núcleo de la infraestructura donde se alojan los componentes críticos de **Rhea**. Incluye servidores dedicados para:
 - i. **Servidor Web / Backend:** Aloja la aplicación principal de **Rhea** (Node.js con lógica de negocio).
 - ii. **Servidor de Base de Datos:** Almacena datos clave como información

- de usuarios, reservas, servicios y pagos.
- iii. **Servidor de Notificaciones:** Conectado a servicios externos como SendGrid o Twilio para enviar confirmaciones y recordatorios a los clientes.
- b. El centro de datos está protegido mediante un **firewall**, asegurando la seguridad y confidencialidad de los datos en el sistema.
- 2. **Red de Usuarios y Dispositivos de Acceso:**
 - a. Representa los dispositivos de acceso de los clientes y usuarios del sistema **Rhea** (por ejemplo, computadoras o teléfonos móviles de clientes, dueños y empleados).
 - b. Los usuarios acceden a **Rhea** a través de la **Interfaz de Usuario en el Cliente Web**, y sus solicitudes son dirigidas hacia el **Centro de Datos** mediante un **Router de Red**, que canaliza el tráfico entre los clientes y el backend del sistema.
- 3. **Conexiones Externas:**
 - a. La conexión a Internet, proporcionada por un **Proveedor de Servicios de Internet (ISP)**, permite el acceso del **Centro de Datos** y los usuarios a servicios externos esenciales para **Rhea**.
 - b. Estos servicios externos incluyen la **Pasarela de Pago** (Transbank) para procesar transacciones de reservas en línea, y las **APIs de Notificación** para enviar alertas y confirmaciones a los clientes.
- 4. **Dispositivos de los Usuarios (Clientes, Dueños y Empleados):**
 - a. Aunque no se muestra en detalle en el diagrama, los usuarios de **Rhea** pueden acceder a la aplicación desde cualquier dispositivo personal con conexión a Internet. Estos dispositivos se conectan al **Servidor Web** en el Centro de Datos y permiten realizar reservas, gestionar servicios, y consultar su historial de citas.

Usabilidad y Calidad de software- Front-end -Vistas Principales

Vistas principales del sistema – Vista Panel de Reservas del Profesional / Dueño
Vista de Cliente Final Reservar Cita.

Rhea Reserve

Crea Tu Negocio

Panel de Reservas

Servicios

Profesionales

Notificaciones

Calendario

Configuración

Soporte

Filtrar por

Ordenar por

8:30

9:00

9:30

10:00

10:30

11:00

11:30

12:00

12:30

13:00

13:30

14:00

14:30

15:00

15:30

16:00

16:30

17:00

17:30

Corte de Pelo

60 minutos

Lucas Acevedo

Confirmado

Perfilado de Cejas

60 minutos

Miguel Osorio

Confirmado

Perfilado de Cejas

60 minutos

Christian Labraro

Pendiente

Perfilado de Cejas

Cancelado

Corte de Pelo + Barba

90 minutos

Franco Ignacio

Confirmado

Rhea Reserve

Black Buffalo Men's Salon

Selecciona el Servicio a Realizar.

Corte de Pelo

Corte de Pelo Clasico + Lavado

60 minutos

\$15.000

Perfilado de Cejas

Perfilado de cejas

30 minutos

\$3.000

Corte de Pelo + Barba

Corte de Pelo Clasico + Lavado +
Perfilado de Barba

90 minutos

\$18.000

Corte de Pelo + Barba

Corte de Pelo Clasico + Lavado +
Perfilado de Barba

\$18.000

Siguiente

Direccion: Av Alberto Llona 4311

Correo: blabuff@barber.cl

Horario de Atencion

Lunes de 08:00 a 19:00 hrs.

Martes de 08:00 a 19:00 hrs.

Miercoles de 08:00 a 19:00 hrs.

Jueves de 08:00 a 19:00 hrs.

Viernes de 08:00 a 19:00 hrs.

Sabado - Cerrado

Domingo - Cerrado

La interfaz del **Panel de Reservas** en **Rhea** ha sido diseñada para proporcionar una experiencia de usuario intuitiva, moderna y eficiente, enfocada en la administración y visualización de reservas diarias para negocios que operan bajo un sistema de citas, como barberías y salones de belleza. La disposición clara y el uso de colores permiten a los dueños y empleados ver rápidamente el estado y los detalles de cada reserva.

1. **Estilo Moderno y Profesional:**

- a. La interfaz utiliza una paleta de colores vibrantes pero sobria, destacando visualmente cada tipo de estado de las reservas (confirmado, pendiente, cancelado). Este diseño moderno aporta una apariencia profesional que resalta la identidad visual de la plataforma **Rhea Reserve**.

2. **Interactividad y Usabilidad:**

- a. La navegación es intuitiva gracias a la disposición en columnas horarias, permitiendo al usuario desplazarse rápidamente por las reservas del día. Además, filtros y opciones de ordenación en la parte superior facilitan la búsqueda y organización de las reservas por servicio o profesional.

3. **Iconografía Clara y Menú Lateral:**

- a. El menú lateral utiliza iconos intuitivos para representar cada sección importante del sistema, como **Servicios**, **Profesionales**, **Notificaciones**, **Calendario** y **Soporte**. Esto permite a los usuarios acceder fácilmente a distintas áreas del sistema sin perder de vista el Panel de Reservas.

4. **Visualización Detallada de las Reservas:**

- a. Cada reserva muestra información esencial, como el tipo de servicio, la duración, el profesional asignado y el estado de la reserva, con un código de color claro:
 - i. **Verde:** Confirmado
 - ii. **Amarillo:** Pendiente
 - iii. **Rojo:** Cancelado

Vista Reserva Seleccionar Servicios.

1. **Información del Negocio:**

- a. En la parte superior de la pantalla, se muestra el nombre y logo del negocio, ofreciendo una identidad visual que genera confianza.
- b. A la derecha, se proporciona información esencial como la **dirección**, **correo electrónico de contacto** y **horario de atención**, permitiendo al cliente tener claridad sobre la disponibilidad del negocio.

2. **Selección de Servicios:**

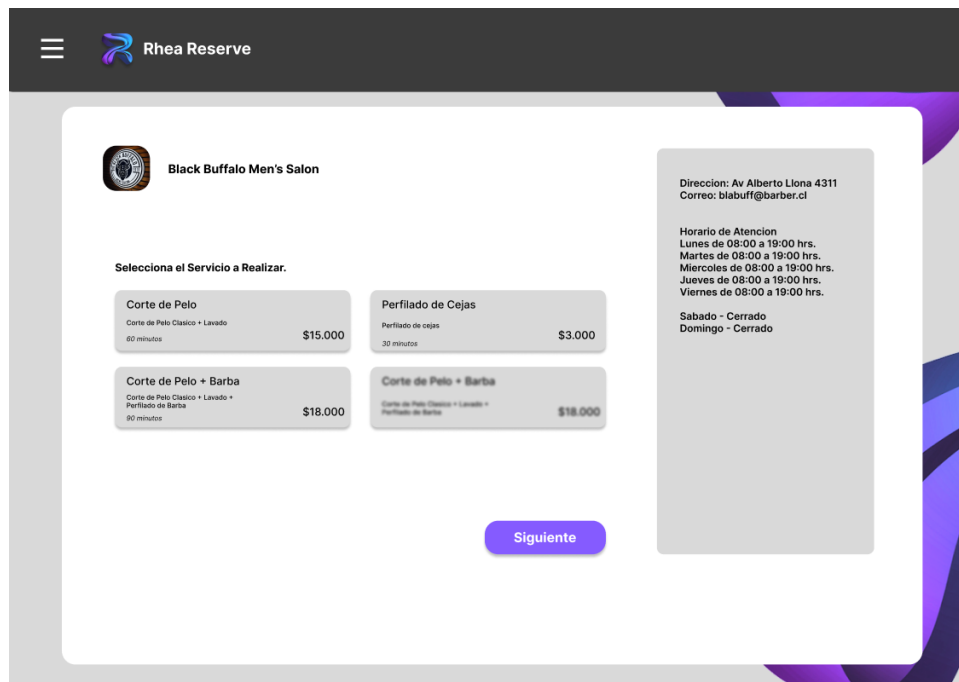
- a. Los servicios se presentan en tarjetas individuales que destacan el **nombre del servicio**, una breve descripción, la duración y el precio, lo que facilita al cliente comparar opciones y elegir el servicio que necesita.
- b. Las tarjetas están diseñadas con un estilo minimalista, usando colores neutros y fuentes claras para asegurar que la información sea legible y visualmente atractiva.

3. **Experiencia de Usuario Intuitiva:**

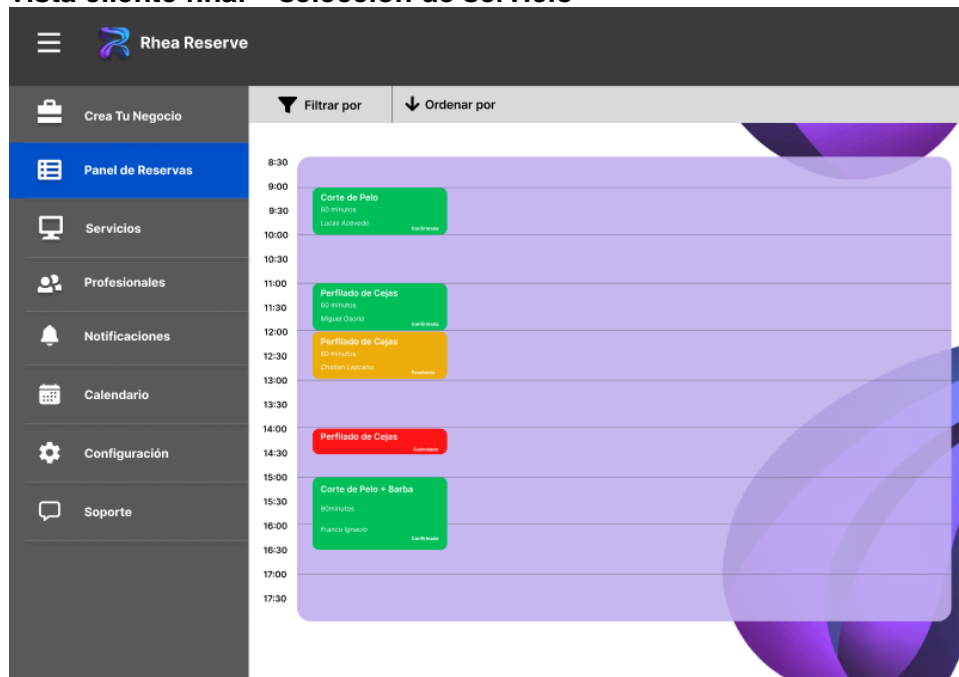
- a. La opción “**Siguiente**” permite avanzar al siguiente paso del proceso de reserva, ofreciendo una navegación guiada que simplifica el flujo de la reserva.
- b. El diseño claro y organizado ayuda a reducir la complejidad del proceso, permitiendo que los clientes reserven sus servicios de manera eficiente y sin confusión.

Usabilidad y Calidad de software – Vistas específicas de usuario

Vistas principales del sistema – Vista Principal de Usuario a escritorio de trabajo principal de su componente de sistema



vista cliente final – seleccion de servicio



Vista de empleado – panel de reservas

¡Bienvenido (Nombre Completo)!
Comencemos con el proceso de crear tu negocio!

Información de la Cuenta

Cuenta Electrónica
barberspace@gmail.com

Nombre del Negocio
Black Buffalos

¿Cuántos Profesionales Atienden tu Negocio?
2 - 4

Tipo de Negocio
Barbería

Dirección del Negocio

Horario de Apertura

		desde:	hasta:	
Lunes	desde:	08:00	hasta: 19:00	<input type="checkbox"/> Cerrado
Martes	desde:	08:00	hasta: 19:00	<input type="checkbox"/> Cerrado
Miércoles	desde:	08:00	hasta: 19:00	<input type="checkbox"/> Cerrado
Jueves	desde:	08:00	hasta: 19:00	<input checked="" type="checkbox"/> Cerrado

Vista dueño - creacion del negocio

Vistas principales del sistema: Cada tipo de usuario tiene un "escritorio" o vista principal personalizada que le permite acceder a las funcionalidades de **Rhea** que son relevantes para su rol en el sistema.

- **Dueño (Administrador):**
 - **Acceso Total:** El dueño tiene control completo sobre el negocio, permitiéndole gestionar reservas, empleados, horarios y servicios. También puede configurar su perfil de negocio y ver reportes de ingresos.
 - **Vista Principal:** En su escritorio, el dueño puede visualizar un **panel de reservas**, opciones de configuración del negocio, un acceso rápido a reportes de ingresos, y una sección para gestionar empleados y servicios.
- **Cliente (Usuario que Reserva):**
 - **Reservas y Perfil:** Los clientes tienen una vista centrada en la selección y reserva de servicios. También pueden consultar su historial de reservas y actualizar su perfil.

- **Vista Principal:** La interfaz muestra la disponibilidad de servicios, horarios, y opciones de reserva rápida. También incluye una sección de perfil donde el cliente puede modificar su información personal.
- **Profesional (Empleado del Negocio):**
 - **Gestión de Reservas:** Los profesionales pueden ver su horario de trabajo, las reservas del día, y administrar su disponibilidad.
 - **Vista Principal:** En su escritorio, el profesional visualiza un **panel de citas** que detalla las reservas confirmadas, pendientes o canceladas, y puede ajustar su disponibilidad en el calendario.

8.7 Consistencia en la cantidad de vistas de la arquitectura.

Se asegura que cada vista arquitectónica de **Rhea Reserve** esté alineada con los requisitos funcionales y no funcionales establecidos para el sistema. Esto implica que todas las vistas (escenario, lógica, procesos, despliegue e infraestructura) cumplen una función específica en la representación del sistema, evitando redundancias y proporcionando una visión coherente y completa de la arquitectura.

Cada vista ha sido diseñada para abordar aspectos únicos y complementarios, permitiendo que el sistema mantenga consistencia en su diseño, organización y funcionalidad. Esto asegura que el desarrollo y la implementación de **Rhea Reserve** se realicen de manera estructurada y conforme a los objetivos definidos en los requisitos del sistema.

8.7.1 DESCRIPCION DE MODULOS

Nombre del módulo	Descripción	Componentes incluidos
Gestión de Reservas	Permite a los clientes realizar, modificar y cancelar reservas, y a los dueños gestionar el calendario de citas.	Módulo de reservas, Módulo de notificaciones, Módulo de calendario
Gestión de Servicios	Administra los servicios disponibles en el negocio, permitiendo agregar, modificar o eliminar servicios.	Módulo de creación de servicios, Módulo de actualización, Módulo de eliminación
Gestión de Usuarios y Roles	Controla la creación de cuentas, administración de perfiles y roles para dueños, clientes y profesionales.	Módulo de registro, Módulo de autenticación, Módulo de perfiles
Gestión de Pagos	Facilita la integración con la pasarela de pagos para	Módulo de integración con Transbank, Módulo de gestión de pagos

	procesar y registrar pagos de reservas.	
Reportes e Informes	Genera reportes de reservas, ingresos y estadísticas sobre el uso de los servicios.	Módulo de reportes, Módulo de generación de estadísticas

8.7.2 DESCRIPCIÓN DE COMPONENTES

Nombre del Componente	Descripción	Componentes Relaciones
Motor de Reservas	Gestiona la creación, modificación y cancelación de reservas en el sistema.	Módulo de Reservas, Módulo de Calendario
Motor de Notificaciones	Envía notificaciones a los clientes y dueños a través de correo o mensajes de texto.	Módulo de Reservas, Módulo de Usuarios
Gestor de Usuarios y Roles	Maneja la creación y administración de perfiles y roles, controlando el acceso a funcionalidades según el tipo de usuario.	Módulo de Autenticación, Módulo de Perfiles
Procesador de Pagos	Facilita el manejo de pagos en línea a través de la integración con la pasarela de pago (Transbank).	Módulo de Pagos, Módulo de Notificaciones
Generador de Reportes	Crea reportes y estadísticas sobre reservas, ingresos y uso de servicios.	Módulo de Reportes, Módulo de Reservas

8.7.3 DESCRIPCIÓN DE CONECTORES

Nombre del Conector	Tipo	Propósito
Rest API	Comunicación	Transferencia de datos entre módulos, permitiendo consultas y actualizaciones.
WebSocket (Socket.io)	Comunicación IRT	Permite actualizaciones en tiempo real en la interfaz de usuario, como la notificación

		de cambios en el estado de una reserva
--	--	--

8.7.3.1 Arquitectura lógica

La arquitectura lógica de **Rhea Reserve** está organizada en capas para optimizar la modularidad, escalabilidad y mantenimiento del sistema. A continuación, se describen las principales capas de la arquitectura:

1. **Capa de Presentación (Frontend):**
 - a. **Responsabilidad:** Gestiona la interacción con el usuario final. A través de una interfaz intuitiva, permite a los clientes, dueños y empleados acceder a las funcionalidades correspondientes, como la reserva de citas, la configuración de servicios y la gestión de perfiles.
 - b. **Tecnologías:** Construida en **React** para ofrecer una experiencia de usuario fluida y reactiva, compatible con dispositivos móviles y escritorio.
2. **Capa de Servicios y API (Backend):**
 - a. **Responsabilidad:** Procesa las peticiones de los usuarios y conecta la capa de presentación con la lógica de negocio. Los microservicios de esta capa manejan funcionalidades específicas como reservas, pagos y notificaciones.
 - b. **Tecnologías:** Implementada en **Node.js** con **Express** para construir APIs RESTful, ofreciendo comunicación entre los componentes y con otros servicios externos (ej., Transbank para pagos).
3. **Capa de Lógica de Negocio:**
 - a. **Responsabilidad:** Contiene las reglas de negocio y los procesos principales de **Rhea Reserve**, como la validación de disponibilidad de servicios, cálculo de precios y gestión de notificaciones.
 - b. **Modularidad:** Dividida en módulos que representan diferentes aspectos del sistema (ej., módulo de reservas, módulo de pagos, módulo de usuarios), permitiendo una fácil escalabilidad y mantenimiento.
4. **Capa de Persistencia (Base de Datos):**
 - a. **Responsabilidad:** Gestiona el almacenamiento de datos, incluyendo información de usuarios, reservas, pagos y servicios. Se asegura la integridad y seguridad de los datos mediante transacciones y controles de acceso.
 - b. **Tecnología:** Utiliza **MySQL** para la persistencia de datos, con optimización en índices y consultas para mejorar el rendimiento en tiempo de respuesta.
5. **Capa de Integración con Servicios Externos:**
 - a. **Responsabilidad:** Interactúa con servicios externos esenciales para la operación de **Rhea Reserve**, tales como **SendGrid** para notificaciones y **Transbank** para pagos.
 - b. **Propósito:** Facilita la comunicación con servicios de terceros mediante APIs, asegurando una integración segura y eficiente con funcionalidades adicionales.

8.7.3.2 Performances

El sistema **Rhea Reserve** está diseñado para manejar una alta concurrencia, soportando hasta 1000 usuarios simultáneamente sin comprometer el rendimiento. Con el objetivo de lograr tiempos de respuesta inferiores a 2 segundos por transacción, se implementarán las siguientes estrategias:

- **Escalabilidad Horizontal:** Utilizando **contenedores Docker** para desplegar microservicios en una infraestructura de nube como **AWS**, permitiendo aumentar la capacidad del sistema conforme a la demanda de usuarios. Esto asegura que se puedan añadir recursos de manera ágil y eficiente.
- **Balanceo de Carga:** Un **balanceador de carga** distribuirá las solicitudes de los usuarios entre las distintas instancias de los microservicios, garantizando una distribución equitativa del tráfico y evitando que un único servicio se sature.
- **Optimización de Consultas:** Se aplicarán técnicas de optimización en las consultas a la base de datos (MySQL), como el uso de índices y consultas asíncronas, para reducir los tiempos de respuesta y mejorar la eficiencia en el acceso a los datos.

8.7.3.3 Criterios de Calidad

Para asegurar que **Rhea Reserve** cumpla con los niveles de calidad esperados, el sistema debe adherirse a los siguientes criterios:

- **Disponibilidad 24/7:** La arquitectura estará diseñada para garantizar un tiempo de actividad continuo. Para lograr esto, se implementarán mecanismos de redundancia y planes de recuperación ante desastres en la infraestructura de AWS, asegurando que el sistema esté disponible en todo momento.
- **Respuesta Rápida:** Todas las transacciones y operaciones de usuario deben completarse en menos de 2 segundos. Esto se logrará mediante la implementación de técnicas de caché en áreas de alta frecuencia de consulta y un diseño eficiente de la API, optimizando el procesamiento de datos para proporcionar una experiencia rápida y fluida a los usuarios.

8.7.3.4 Detalles de la implementación

La implementación del sistema **Rhea Reserve** se basará en una arquitectura de microservicios, utilizando las siguientes tecnologías:

- **Backend:** Node.js con Express para la creación de APIs RESTful que soporten la comunicación entre los componentes y con servicios externos como la pasarela de pagos.
- **Frontend:** React junto con Tailwind CSS para desarrollar una interfaz de usuario moderna, intuitiva y responsiva, adaptada a dispositivos móviles y de escritorio.
- **Base de Datos:** MySQL para la gestión eficiente de datos estructurados, optimizada con índices y consultas específicas para mejorar la velocidad de respuesta.
- **Infraestructura:** Azure, utilizando contenedores Docker para una implementación flexible y escalable de los microservicios. Se utilizarán servicios como **Azure App Service** y **Azure Kubernetes Service (AKS)** para orquestar y administrar los

contenedores, lo cual permite una fácil adaptación a picos de demanda y alta disponibilidad.

8.7.3.5 Lenguajes y plataformas

Las tecnologías y versiones específicas que se utilizarán en el desarrollo del sistema incluyen:

- **Node.js:** v20.x
- **Express:** v4.x
- **React:** v18.x
- **Tailwind CSS:** v3.x
- **MySQL:** v8.0
- **Docker:** v20.x
- **Azure:** Servicios de **Azure App Service**, **Azure Kubernetes Service (AKS)** y **Azure SQL Database** para la gestión de datos y escalabilidad en la nube.
- **Python:** v3.9 (para scripts adicionales y automatización)
- **Visual Studio Code:** v1.70 (IDE para el desarrollo del sistema)

9.1 Acceso a fuentes de información del proyecto

Para acceder a la carpeta de evidencias acumuladas y otros documentos relevantes del proyecto, puede utilizar el siguiente enlace:

[Carpeta de Evidencia del Proyecto - Drive](#)
[Vínculo a Repositorio GitHub](#)

Esta carpeta incluye documentación técnica, diagramas, pruebas de rendimiento y cualquier otra evidencia relacionada con el desarrollo y la implementación del sistema