****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**“*SISTEMA DE GESTIÓN HOTELERA”***

Curso: *Programación III*

Docente: Ing. Elard Ricardo Rodríguez Marca

Integrantes:

***Nina Vargas, Luigui Augusto 2019065166***

***Chambe Torres, Edgard Reynaldo 2019064917***

***Condori Varga, Tomas Yoel 2018000487***

***Rojas Bedregal, Brian Eric***

**Tacna – Perú**

***2023***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | CHAMBE-NINA-CONDORI-ROJAS | Ing. Elard Rodríguez |  | 18/06/2023 | Versión 1.0 |
| 2.0 | CHAMBE-NINA-CONDORI-ROJAS | Ing. Elard Rodríguez |  | 24/06/2023 | Actualización Diagrama secuencia, actividades y objetos  Gestionar Habitación |
| 3.0 | CHAMBE-NINA-CONDORI-ROJAS | Ing. Elard Rodríguez |  | 30/06/2023 | Correcion Diagrama Entidad Relacion  Diagrama de Componentes |

***Sistema de Gestión Hotelera***

**Documento de Arquitectura de Software**

Versión *{1.0}*

ÍNDICE GENERAL

**Tabla de contenido**

[**1.**](#_heading=h.gjdgxs) **INTRODUCCIÓN**

[Propósito](#_heading=h.30j0zll)

[Alcance](#_heading=h.1fob9te)

[Definición, siglas y abreviaturas](#_heading=h.3znysh7)

Organización del documento

2. Objetivos y restricciones arquitectónicas

2.1 Priorización de requerimientos

2.2 Restricciones

3. Representación de la arquitectura del sistema

3.2 [Vista de Caso de uso](#_heading=h.4i7ojhp)

[2.1.2. Diagramas de casos de uso](#_heading=h.2xcytpi)

2.1.2 Descripción de casos de uso

3.3 [Vista Lógica](#_heading=h.1ci93xb)

2,1,3 Diagrama de Subsistemas (paquetes)

2.1. 3 Diagrama de secuencia (vista de diseño)

2.1.4 Diagrama de colaboracion (vista de diseño)

2.1.5 Diagrama de actividad (vista de diseño)

[2.1.4. Diagrama de Objetos](#_heading=h.3whwml4)

[2.1.5. Diagrama de Clases](#_heading=h.2bn6wsx)

2.1.6 Diagrama de Base de Datos (Relacional o no Relacional)

3.4 [Vista de Implementación](#_heading=h.qsh70q)

2.1. 8 Diagrama de arquitectura

[2.1.9. Diagrama de componentes](#_heading=h.3as4poj)

3.5 Vista de procesos

2.1. 9 Diagrama de procesos del Sistema (Técnica)

3.6 [Vista de Implantación](#_heading=h.1pxezwc)

[2.1.8. Diagrama de Despliegue](#_heading=h.49x2ik5)

4[**.** Atributos de Calidad del Software](#_heading=h.2p2csry)

[Escenario de Funcionalidad](#_heading=h.147n2zr)

[Escenario de Usabilidad](#_heading=h.3o7alnk)

[Escenario de Confiabilidad](#_heading=h.23ckvvd)

[Escenario de Rendimiento](#_heading=h.ihv636)

[Escenario de Mantenibilidad](#_heading=h.32hioqz)

# **INTRODUCCIÓN**

## **Propósito**

Proporcionar al usuario (Cliente, Administrador), un sistema centralizado para gestionar y llevar a cabo el proceso de reserva hotelera para gestionar la información del inventario de las habitaciones y su disponibilidad, tarifas, pagos, reporte del pago realizado por el cliente.

El sistema se desarrollará con el objetivo de brindar una experiencia de usuario intuitiva y eficiente, con funcionalidades que satisfagan las necesidades y promuevan la interacción y la satisfacción del cliente. Además, se buscará garantizar la seguridad y privacidad de los datos de los usuarios, así como asegurar un rendimiento óptimo y una escalabilidad adecuada para futuros crecimientos.

El propósito final del sistema es mejorar la experiencia del usuario, facilitar la gestión interna de los servicios del hotel, así como promover la fidelidad de los clientes y generar un entorno favorable para el crecimiento y el éxito del negocio.

## **Alcance**

El alcance de nuestro sistema se centra en la región Tacna, en el hotel “Virgen de las Peñas”. Como se verá posteriormente en este documento, se mostrará la representación arquitectónica de los diferentes niveles de abstracción.

## **Definición, siglas y abreviaturas**

**Patrón de arquitectura MVC:** Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

**Patrón de arquitectura tres capas:** El patrón de arquitectura de 3 capas divide una aplicación en capas: presentación, lógica de negocio y acceso a datos. La capa de presentación maneja la interfaz de usuario, la capa de lógica de negocio controla la lógica de la aplicación y la capa de acceso a datos interactúa con la base de datos. Esto permite una mejor organización, reutilización de código y escalabilidad.

**Servidor Web:** Es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente.

**Vista:** representa el modelo en un formulario adecuado para la interacción, generalmente un elemento de interfaz de usuario. MVC se ve a menudo en aplicaciones web, donde la vista es la página HTML y el código que recopila datos dinámicos para la página.

**Controlador:** Procesa y responde a eventos, generalmente acciones del usuario, e invoca cambios en el modelo y quizás en la vista. La parte lógica de toda la aplicación es como la parte intermedia que interactúa tanto entre el modelo y la vista.

**UML:** Lenguaje unificado de Modelado, se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software.

**BASE DE DATOS:** es el tipo de almacenamiento que usaremos para guardar los datos de los usuarios registrados que utilicen nuestro sistema.

# **REPRESENTACIÓN DE ARQUITECTURA**

## Priorización de Requerimientos

|  |  |  |  |
| --- | --- | --- | --- |
| Origen  ID | Nombre | Descripción | Prioridad |
| RF - 01 | Autenticar Usuario | El usuario podrá iniciar sesión para acceder al sistema. | Alta |
| RF - 02 | Registrar Cliente | El sistema debe permitir al cliente crear una cuenta. | Alta |
| RF - 03 | Realizar reserva | El cliente podrá realizar directamente su reserva una vez haga su elección. | Media |
| RF - 04 | Actualizar Reserva | El administrador podrá ver la actualización del estado de la reserva | Alta |
| RF – 05 | Visualizar Reporte Reserva | El sistema mostrará al cliente un reporte de su reserva. | Media |
| RF - 06 | Realizar Pago | El sistema redirigirá al stripe de pago donde la plataforma validará el pago correspondiente. | Baja |
| RF - 07 | Gestionar habitaciones disponibles | El administrador podrá gestionar las habitaciones disponibles | Alta |
| RF – 08 | Visualizar reserva | El usuario podrá visualizar las reservas realizadas | Alta |
| RF – 09 | Visualizar habitaciones disponibles | El usuario podrá visualizar las habitaciones disponibles en el hotel | Alta |

## Requerimientos No Funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| Origen  ID | Descripción de Requerimiento | Descripción | Prioridad |
| RNF - 01 | Usabilidad | El sistema contará con una experiencia de uso intuitiva y amigable al usuario en base al diseño | Alta |
| RNF - 02 | Disponibilidad | El sistema debe estar disponible en todo momento para que los usuarios puedan realizar reservas en cualquier momento del día. | Media |
| RNF - 03 | Seguridad | El sistema debe asegurar que los datos estén protegidos del acceso no autorizado. | Alta |
| RNF - 04 | Integración | El sistema debe ser capaz de integrarse con otros sistemas relevantes, como sistemas de pago y motores de búsqueda de hoteles. | Bajo |
| RNF - 05 | Rendimiento | El sistema debe responder rápidamente a las solicitudes de reserva para brindar una experiencia fluida a los usuarios. | Media |

## 

## **Restricciones**

|  |  |
| --- | --- |
| Número | Descripción |
| 1 | No pasarse del tiempo de entrega del proyecto. |
| 2 | Uso del lenguaje JavaScript como base de desarrollo para el nuevo sistema web. |
| 3 | No excederse del costo o presupuesto del proyecto. |

## Vista de Caso de Uso (Modelo de Caso de uso)

## Vista Lógica (Diagrama de Paquetes)

Diagrama

Descripción generada automáticamente

# **Análisis de Requerimientos del Sistema a implementar**

* 1. Diagrama de Casos de Uso (Narrativa)

|  |  |  |
| --- | --- | --- |
| **AUTENTICAR USUARIO** | | |
| Actores | USUARIO | |
| Descripción | El usuario validará su identidad para ingresar al sistema. La autenticación del usuario se hará mediante un usuario y una contraseña.En el caso del usuario cliente si no se encuentra registrado en el sistema,podrá registrarse para poder utilizar el sistema. | |
| **Flujo normal de eventos** | | |
| **Acción del actor** | | **Respuesta del sistema** |
| 1. El usuario inicia sesión en el sistema. | | 2. El sistema mostrará la interfaz principal. |
| 3. El usuario podrá iniciar su proceso de reservación. | | 4. El Sistema mostrará la interfaz para proceder con la reserva. |

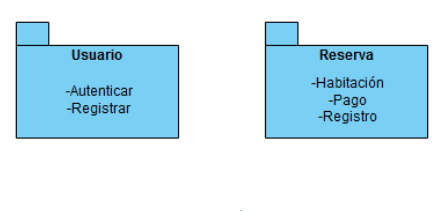
|  |  |  |
| --- | --- | --- |
| **REGISTRAR CLIENTE** | | |
| **Actores** | Cliente | |
| **Descripción** | El cliente ingresará a la interfaz de Registrar Cliente en caso de que sea un cliente nuevo, procederá a ingresar sus datos y el sistema comprobará y guardará su información para luego finalizar registrando al cliente con éxito para que pueda utilizar el sistema. | |
| **Flujo normal de eventos** | | |
| **Acción del actor** | | **Respuesta del sistema** |
| 1. El nuevo usuario ingresara a la interfaz Registrar cliente. | | 2. El sistema mostrará la interfaz de registro Cliente. |
| 3. El nuevo usuario procederá a ingresar su información requerida para registrarse. | | 4. El Sistema validará y guardará la información y finalizará con la creación del nuevo usuario. |

|  |  |  |
| --- | --- | --- |
| GESTIONAR HABITACIÓN | | |
| Actores | ADMINISTRADOR | |
| Descripción | El administrador podrá gestionar las habitaciones disponibles y habilitarlas cuando ya se encuentren desocupadas y listas. | |
| Flujo normal de eventos | | |
| **Acción del actor** | | Respuesta del sistema |
| 1.El usuario Inicia sesión en el sistema | | 2. El sistema mostrará la interfaz principal. |
| 3.Selecciona la opción Gestionar Habitación. | | 4. El sistema mostrará la interfaz para gestionarlo. |
| 5. Ingresa la información al sistema. | | 6. El Sistema actualizará la disponibilidad de la habitación habilitada. |

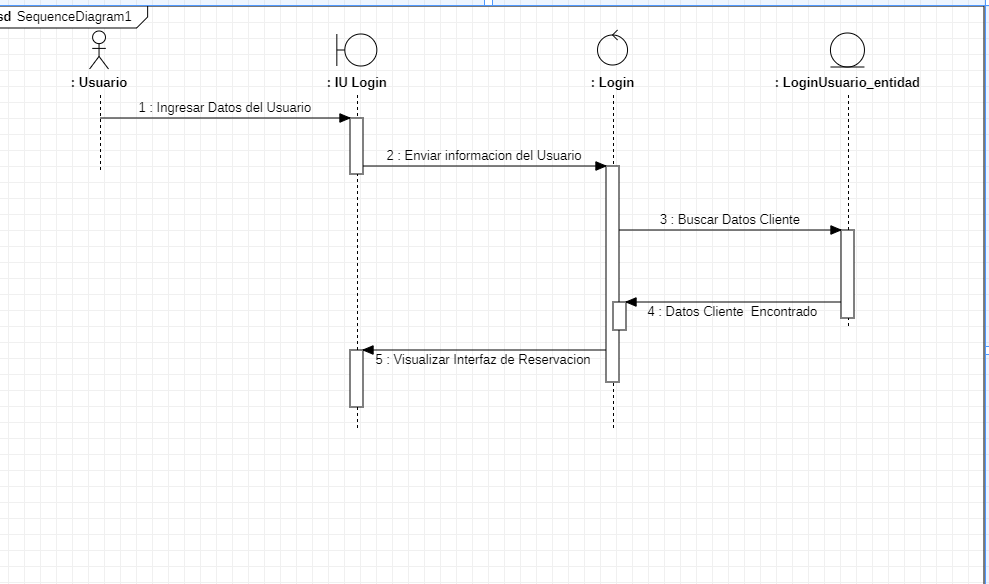
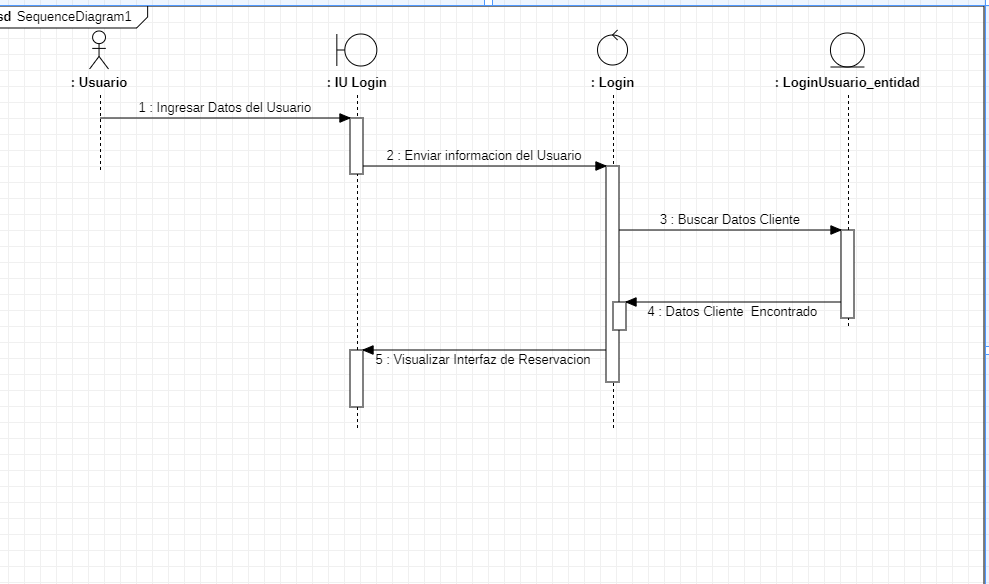
|  |  |  |
| --- | --- | --- |
| REALIZAR RESERVA | | |
| Actores | USUARIO | |
| Descripción | El usuario podrá realizar la reserva ingresando los datos solicitados en el formulario de reserva, una vez realizada la reserva podrá visualizar un reporte de pago de la reserva realizada, luego realizará el pago total y concluirá con el registro de la reserva. Así mismo, el usuario podrá visualizar y ver el estado de la reserva. | |
| Flujo normal de eventos | | |
| **Acción del actor** | | Respuesta del sistema |
| 1. El usuario Inicia sesión en el sistema | | 2. El sistema mostrará la interfaz principal |
| 3. Selecciona el tipo de habitación | | 4. El sistema mostrará el formulario de reserva. |
| 5. Ingresa los datos requeridos en el formulario | | 6. El Sistema muestra el reporte de pago por la reserva realizada. |

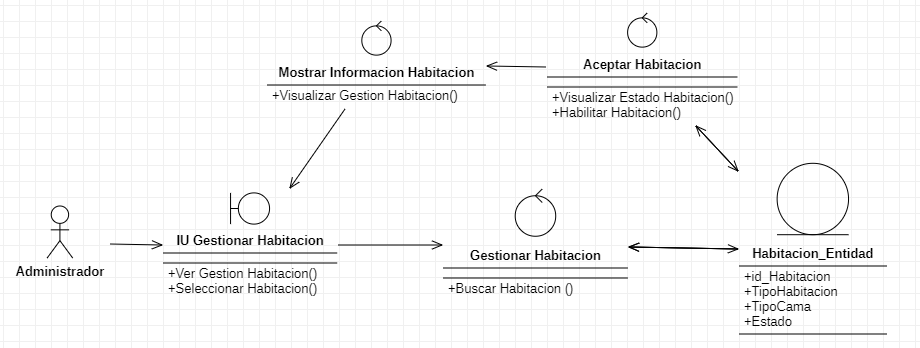
* 1. **Vista Lógica**

## **3.2.1. Diagrama de Subsistemas (paquetes)**

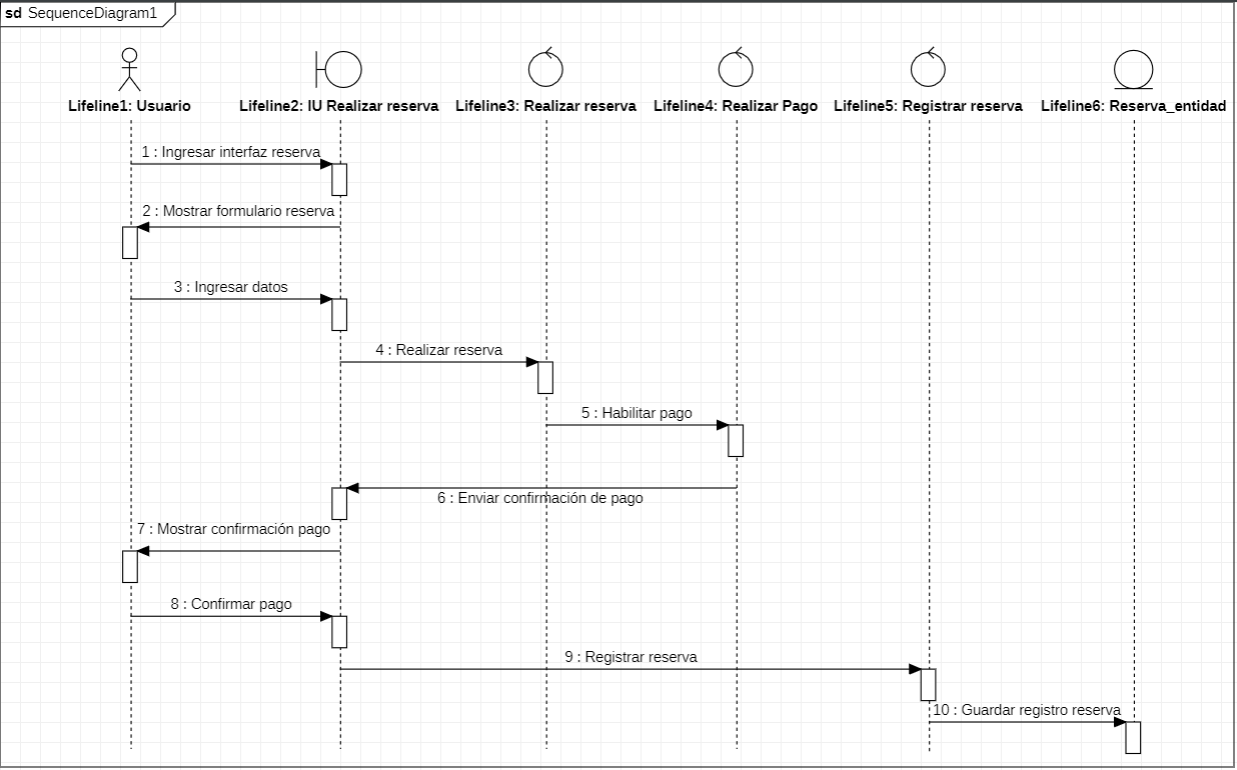


## **3.2.2. Diagrama de Secuencia (vista de diseño)**

* **Autenticar Usuario**
* **Registrar usuario**
* **Gestionar Habitación**

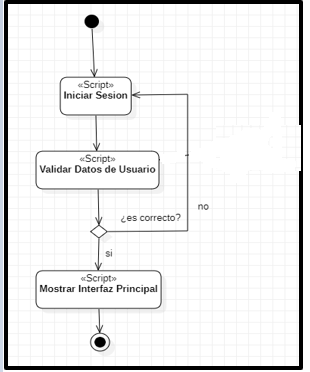


* **Realizar Reserva**

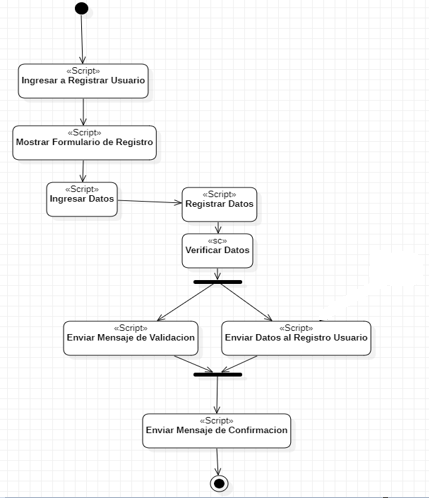


## **3.2.4. Diagrama de Actividades**

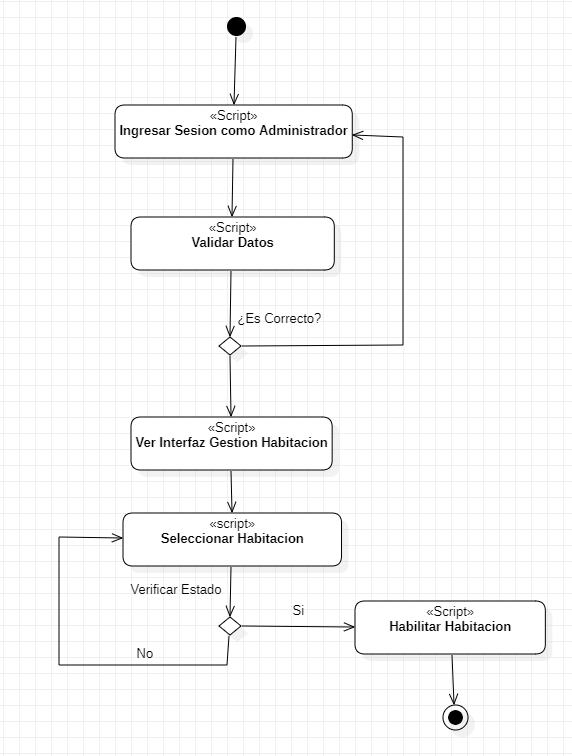
* Autenticar Usuario



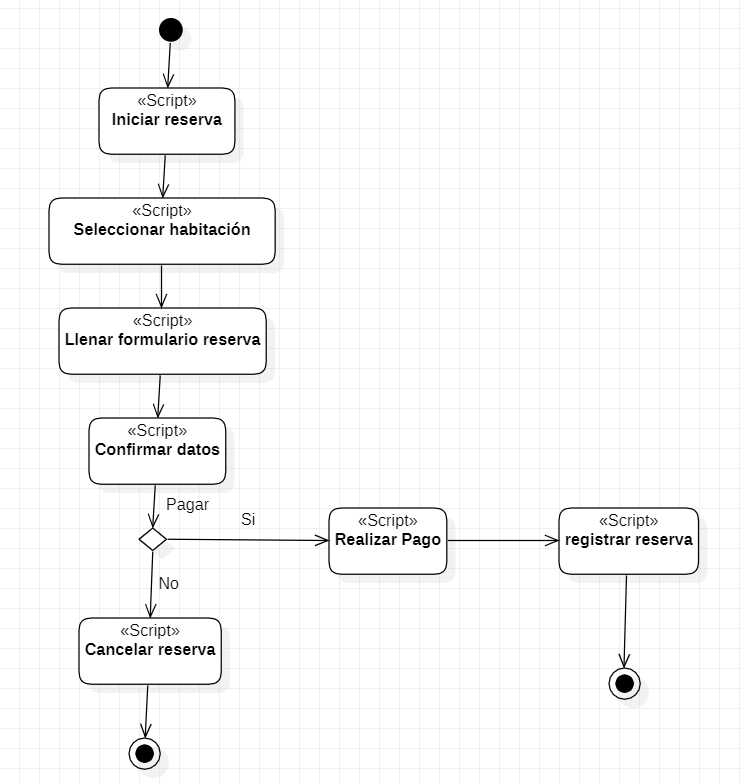
* Registrar Usuario



* **Gestionar Habitación**

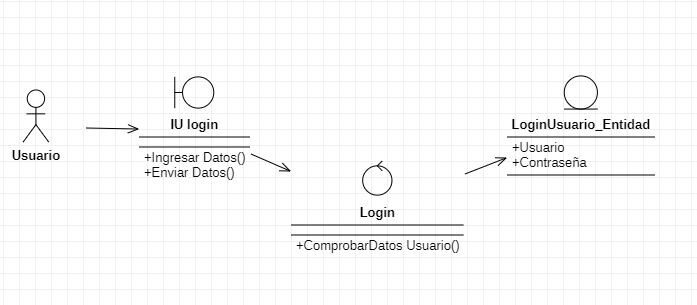


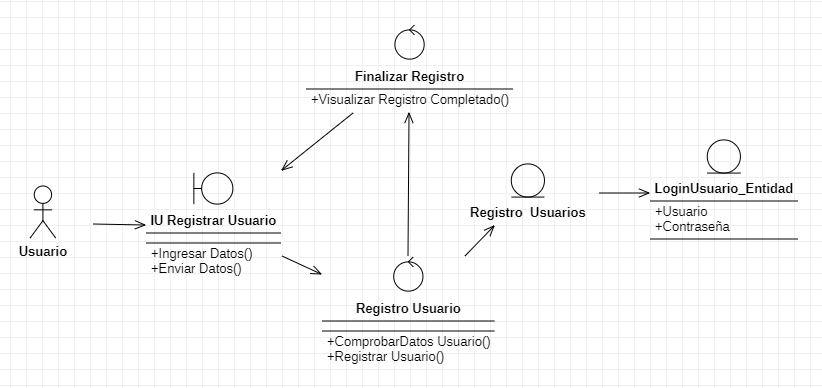
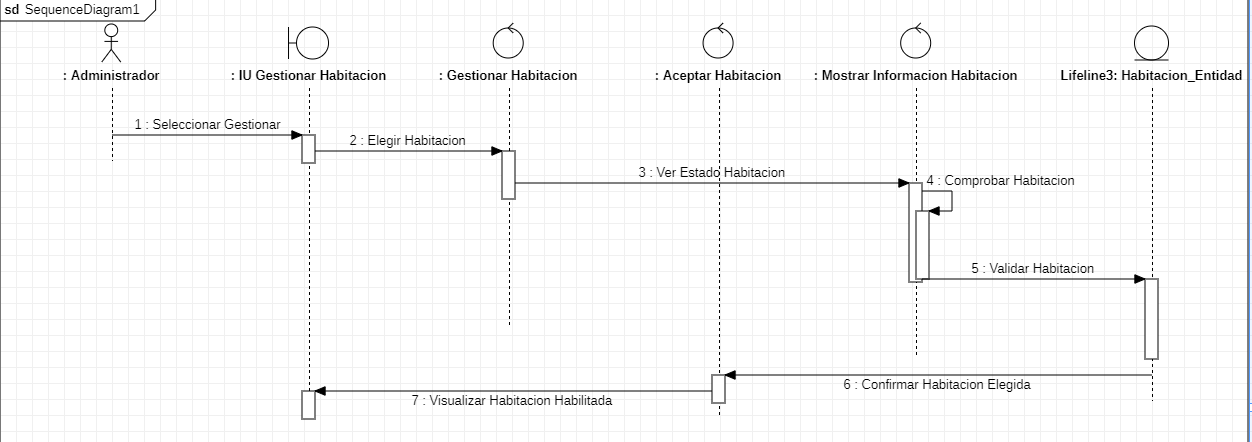
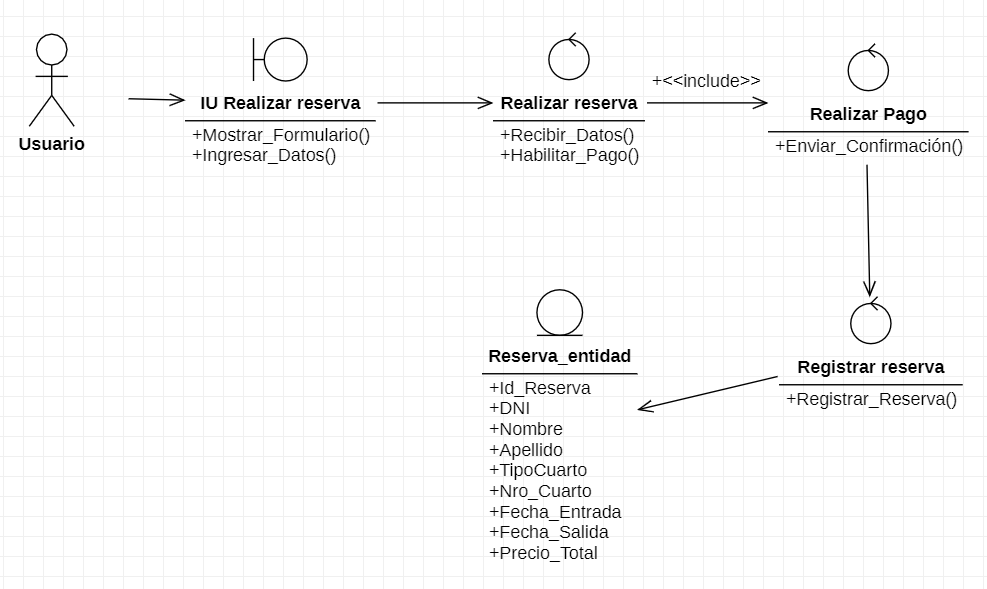
* Realizar Reserva



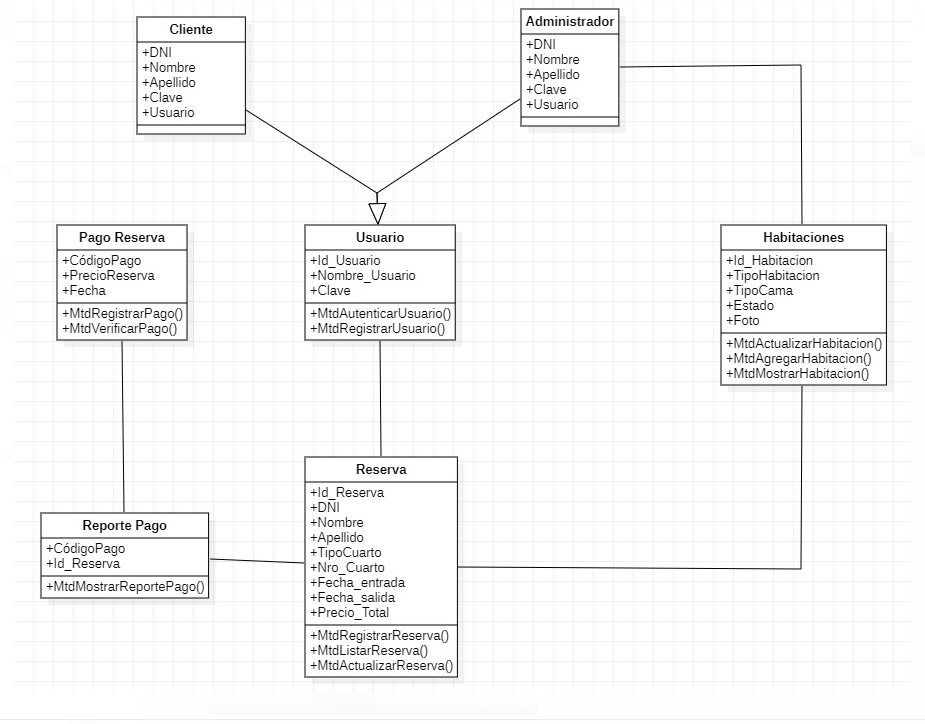
## **3.2.5. Diagrama de Objetos**

* Autenticar Usuario



* Registrar Usuario
*  **Gestionar Habitación**
* Realizar Reserva

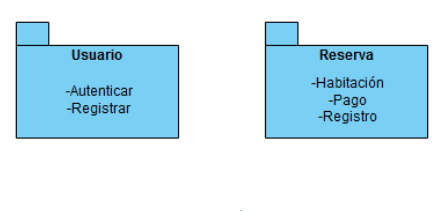
## **3.2.6. Diagrama de Clases**



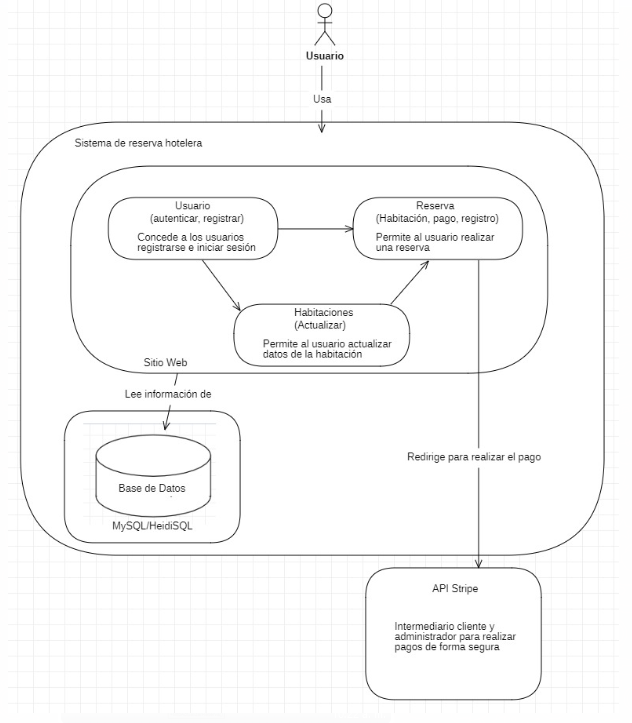
## **3.2.7. Diagrama de Base de datos (Relacional)**

**3.3. Vista de Implementación (vista de desarrollo)**

**3.3.1. Diagrama de arquitectura software (paquetes)**

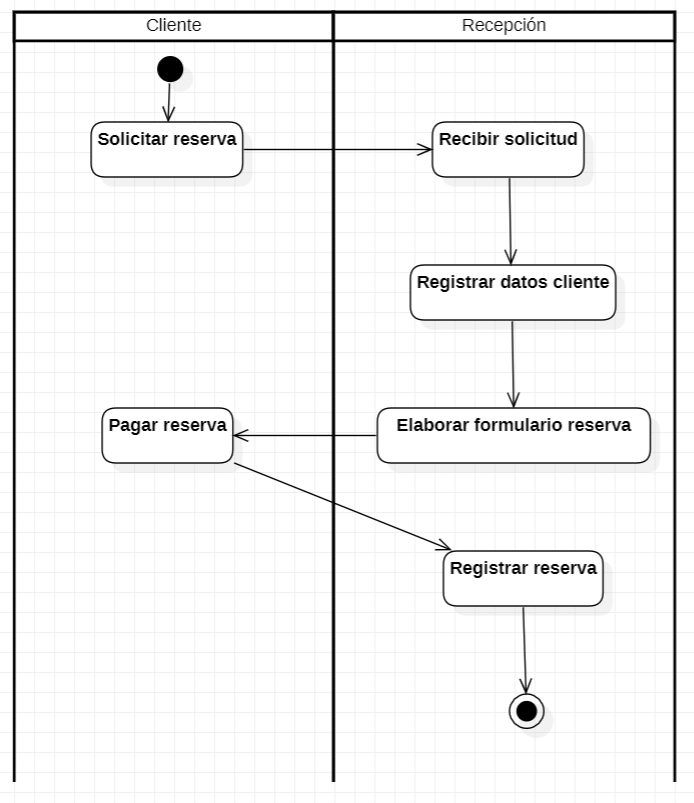


**3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)**



**3.4 Vista de procesos**

**3.4. 1 Diagrama de procesos actual**

****

**3.4. 2 Diagrama de procesos propuesto**

**Diagrama

Descripción generada automáticamente**

**Diagrama

Descripción generada automáticamente**

**3.5** [**Vista de Implantación**](#_heading=h.1pxezwc)

**3**[**.5.1. Diagrama de Despliegue**](#_heading=h.49x2ik5)

**Diagrama, Esquemático

Descripción generada automáticamente**

4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

* Escenario de Disponibilidad

**Necesidad:** El sistema debe estar disponible en todo momento para que los usuarios puedan realizar reservas en cualquier momento del día.

**Solución:** Mantener la página web activa durante el tiempo las 24 horas, siendo posible su uso en ese intervalo de tiempo, en caso de no poder acceder también se puede llamar a soporte para acceder.

Justificación: Al ser una página web, debe estar disponible en todo momento.

* Escenario de Usabilidad

**Necesidad:** Se precisa que los usuarios puedan entender rápida e intuitivamente el uso de la página web.

**Solución:** Implementación de una interfaz sencilla y atractiva al ojo para que los usuarios puedan realizar su navegación desde el primer minuto.

Justificación: Al hacer que una interfaz sea fácil de manipular, provoca que los usuarios se sientan relajados de usarla y no alejarlos del negocio.

**Necesidad:** Un usuario sin experiencia intenta buscar y reservar una habitación en menos de 5 minutos.

**Solución:** Proporcionar una función de reserva rápida que permita al usuario realizar la reserva en solo unos pocos clics,mostrar claramente el precio total y cualquier información relevante, como políticas de cancelación, para ayudar al usuario a tomar una decisión informada de manera rápida.

**Justificación:** Al mostrar información clara y concisa sobre las habitaciones disponibles, el precio total y las políticas relevantes, el usuario podrá tomar decisiones informadas en poco tiempo.

* Escenario de rendimiento

**Necesidad:** El sistema debe responder rápidamente a las solicitudes de reserva para brindar una experiencia fluida a los usuarios.

**Conclusión:** Correcta y buena implementación de los requerimientos establecidos sin ningún fallo, generando tranquilidad en las operaciones de los usuarios.

Justificación: Al implementarse de buena manera las funciones del sistema web, se evitan posibles errores que retrasen y ponga lento los procesos realizados dentro del sistema.

* Escenario de Seguridad

Si un usuario intenta ingresar con una contraseña errónea se notificará el mal ingreso de esta y enviar un email al usuario al tercer intento.

Fuente: Usuario

Estímulo: Contraseña incorrecta

Respuesta: Bloquear la cuenta y notificar que el número de intentos permitido del login, ha excedido el límite, enviar email de recuperación.

Medición de la respuesta: 5 - 10 segundos

* Escenario de Integración

Necesidad: Se requiere que el sistema web sea capaz integrarse con otros sistemas relevantes, como sistemas de pago y motores de búsqueda de hoteles.

Solución: Utilizar APIs proporcionadas por los sistemas de pagos y motores de búsqueda de hoteles para permitir la comunicación y transferencia de datos entre ellos y el sistema web.