

INFORME ROOM RESERVE



TRABAJO PRESENTADO A:  
ING. SANTIAGO HYUN DORADO

EN LA MATERIA, LABORATORIO INGENIERÍA DE SOFTWARE II

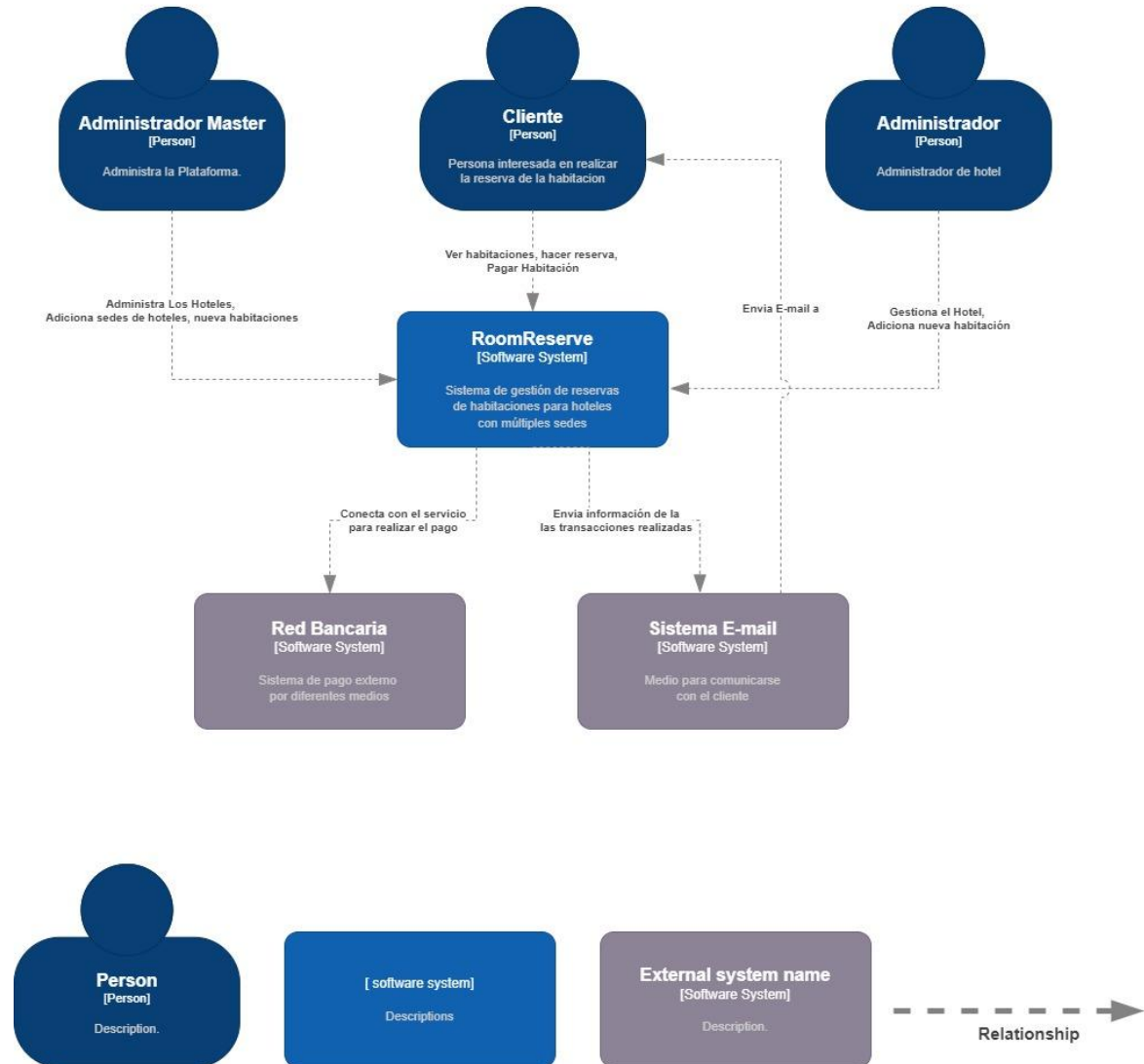
PRESENTADO POR:  
YAQUELIN ALEJANDRA GOMEZ  
JOSE RICARDO GALINDEZ LOPEZ  
KEVIN DARYANY MORALES CRUZ

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
PROGRAMA DE INGENIERÍA DE SISTEMAS  
POPAYÁN  
FEBRERO de 2022

<b>MODELO DE CONTEXTO(C4).</b>	<b>3</b>
<b>REPRESENTACIÓN DE LA ARQUITECTURA.</b>	<b>4</b>
<b>VISTA DE REQUERIMIENTOS:</b>	<b>4</b>
MVP.	4
<b>ESCENARIOS DE CALIDAD.</b>	<b>5</b>
<b>VISTA DE CONTENEDORES (C4)</b>	<b>6</b>
<b>VISTA DE COMPONENTES(C4)</b>	<b>7</b>
<b>VISTA DE MÓDULOS (UML).</b>	<b>8</b>
VISTA DE CLIENTE.	8
VISTA SERVIDOR.	9
VISTA DE COMPONENTES Y CONECTORES(UML).	9
VISTA DE IMPLEMENTACIÓN(MAVEN,UML).	10
VISTA DE INSTALACIÓN(Allocation, UML).	10
VISTA DE DATOS (MODELO ENTIDAD-RELACIÓN).	11
<b>RESUMEN DE LAS DECISIONES DE ARQUITECTURA Y SU JUSTIFICACIÓN.</b>	<b>11</b>
DISEÑO DETALLADO DE LOS MÓDULOS IMPLEMENTADOS	
Diagrama de Clases Servidor	12
Diagrama de Clases Cliente	13
Diagrama de clases Commons	14
<b>DIAGRAMA DE SECUENCIA (UML)</b>	<b>15</b>
Reservar	15
Login	16
Agregar Habitacion	16
<b>PROTOTIPOS DE INTERFAZ DE USUARIO.</b>	<b>17</b>
<b>CÓDIGO FUENTE.</b>	<b>18</b>

## MODELO DE CONTEXTO(C4).

### RoomReserve - C4 - Nivel 1



## **REPRESENTACIÓN DE LA ARQUITECTURA.**

La arquitectura está representada por diferentes vistas utilizando Modelo C4 Y UML, con el fin de visualizar y entender mejor la arquitectura del sistema. La arquitectura utilizada fue la de cliente-servidor por medio del estilo arquitectónico de capas este estilo nos da un desacoplamiento de las partes que componen el sistema, gracias a esto se facilitará agregar o modificar algún componente debido al modularidad del proyecto evitando los errores de funcionalidad. También se realizó la separación de roles (administrador, hotel, cliente).

## **VISTA DE REQUERIMIENTOS:**

### **MVP.**

El proyecto busca desarrollar una aplicación que permita a los usuarios hacer una reservación en un hotel deseado a una de sus habitaciones disponibles. La aplicación permitirá llevar el control de los Diferentes hoteles, de las habitaciones, el pago y la gestión de estas mismas y así poder prestar un mejor servicio para el usuario final y el administrador llevar mejor un informe semanal y mensual de sus servicios.

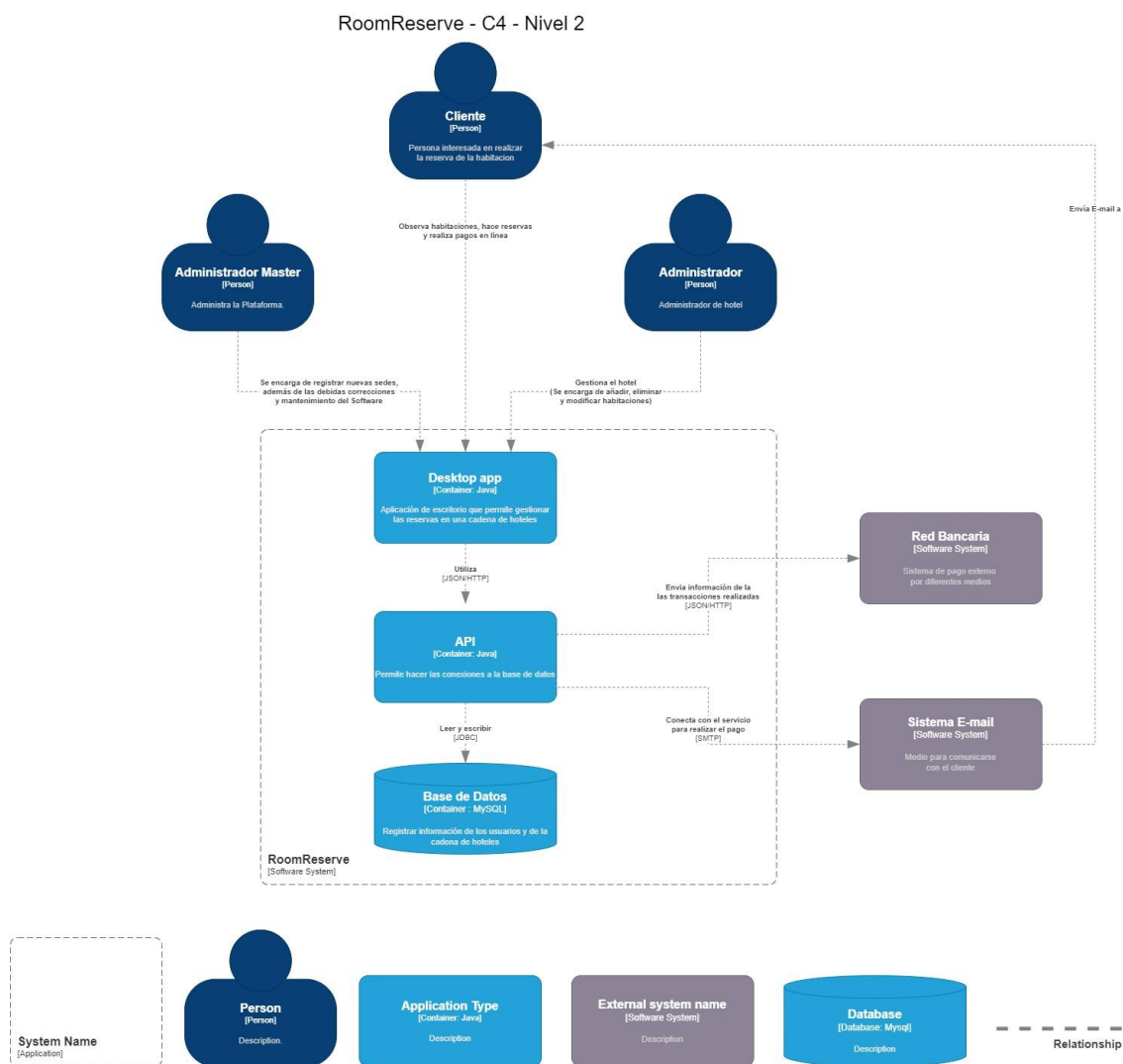
El sistema inicialmente brindará apoyo en los siguientes procesos:

- Visualizar habitación: El usuario podrá consultar las características, disponibilidad y reservar la habitación
- Reservar habitación: Un usuario después de haber seleccionado la habitación acorde a sus necesidades, puede generar una reserva de la misma estableciendo el medio de pago siempre y cuando la habitación se encuentre disponible en las fechas seleccionadas.
- Adicionar habitación: Un administrador podrá realizar la inserción de una habitación a su Hotel y modificar los días de disponibilidad

**ESCENARIOS DE CALIDAD.**

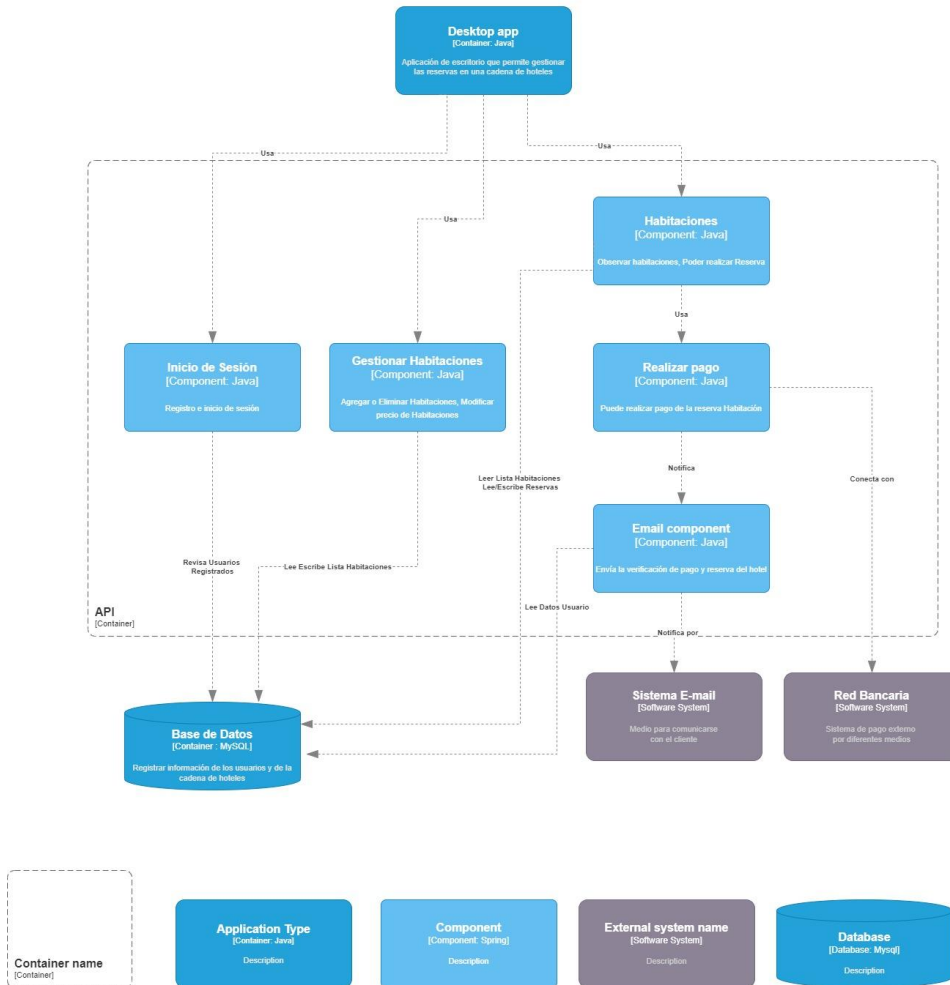
Scenario Brainstorming	
Quality Attribute	Scenario
Modificabilidad	La aplicación debe permitir agregar nuevas sedes, sin afectar a los que ya se encuentran registrados.
	El sistema debe estar disponible ante cambios futuros o correcciones de errores para su correcta evolución.
Escalabilidad	El sistema debe estar diseñado abierto al cambio, es decir el programa debe ser capaz de evolucionar.
Usabilidad	El cliente debe poder navegar por la aplicación fácilmente (Debe ser intuitiva y fácil de manejar).
	Los usuarios deben de ser capaces de entender de forma rápida el escenario del sistema como por ejemplo la interfaz del cliente el cual podrá visualizar las habitaciones disponibles en un tiempo promedio de 3 segundos.
Seguridad	Los datos suministrados por los diferentes usuarios (cliente, administrador) deben permanecer privados.
	Se debe garantizar un ocultamiento de la información (tarjetas de crédito, débito y demás formas de pago en línea, datos personales del cliente, como su teléfono, su email, entre otras) hacia terceros.

## VISTA DE CONTENEDORES (C4)



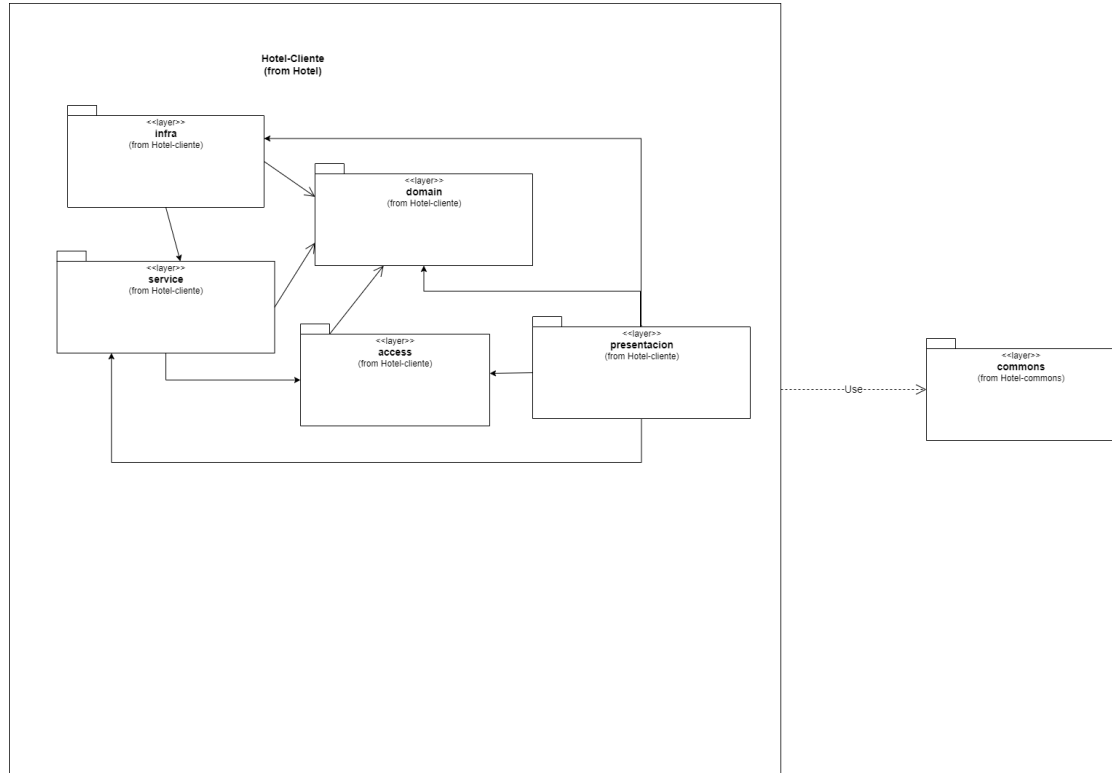
## VISTA DE COMPONENTES(C4)

RoomReserve - C4 - Nivel 3



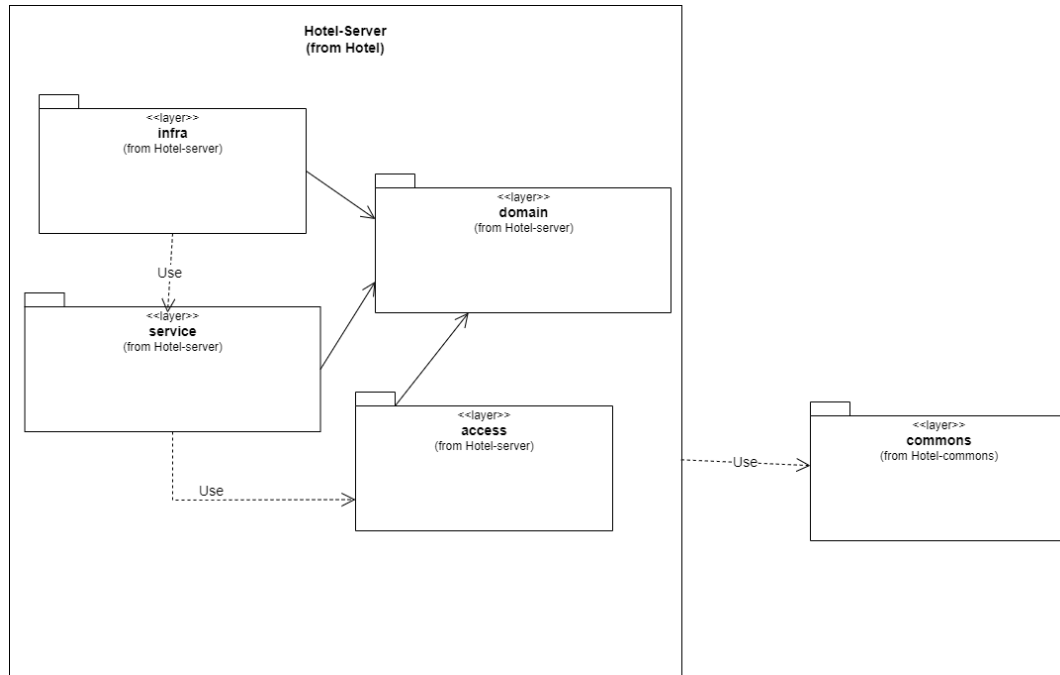
## VISTA DE MÓDULOS (UML).

### VISTA DE CLIENTE.

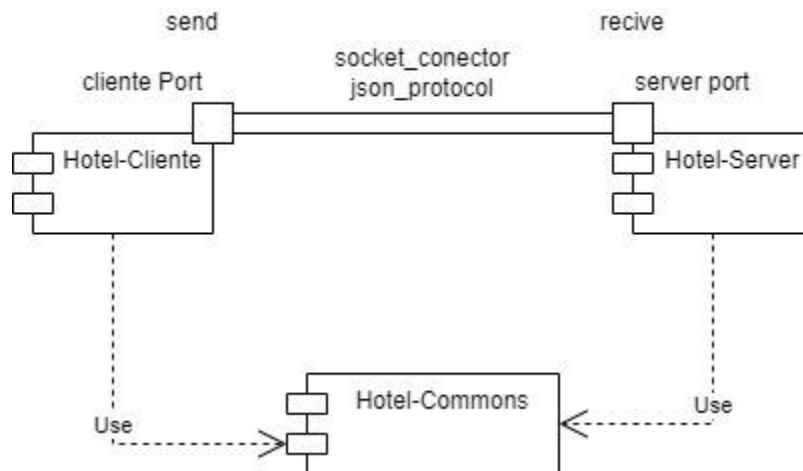




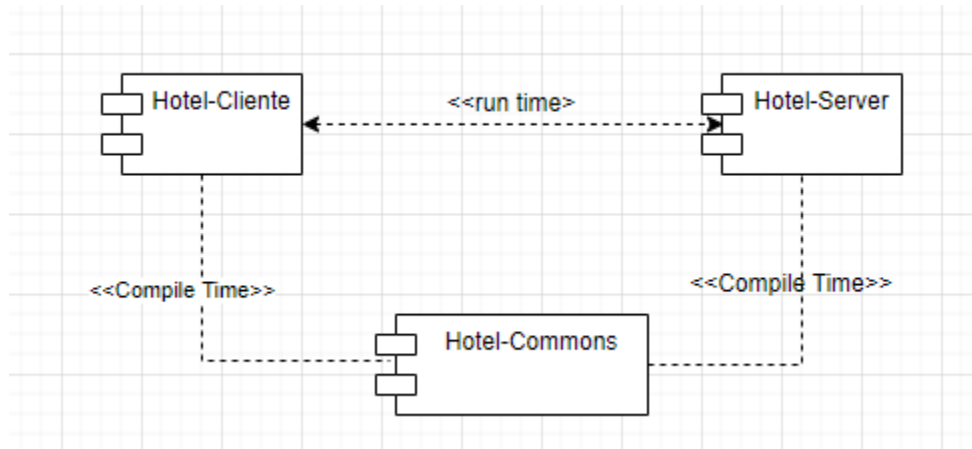
## VISTA SERVIDOR.



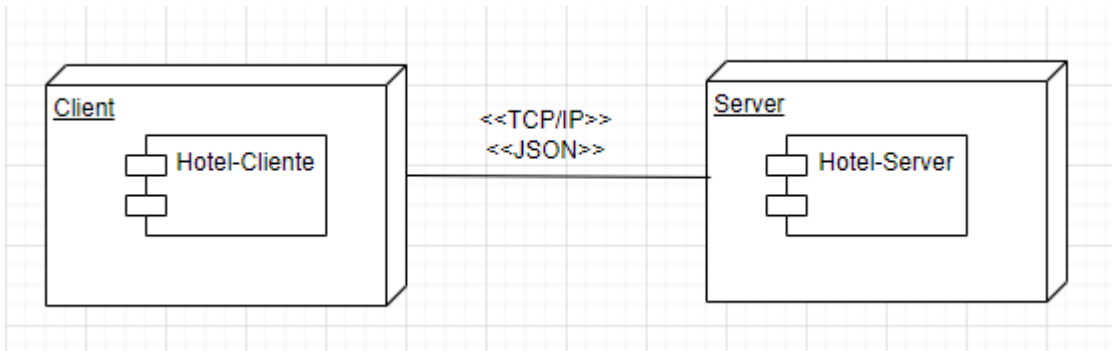
## VISTA DE COMPONENTES Y CONECTORES(UML).



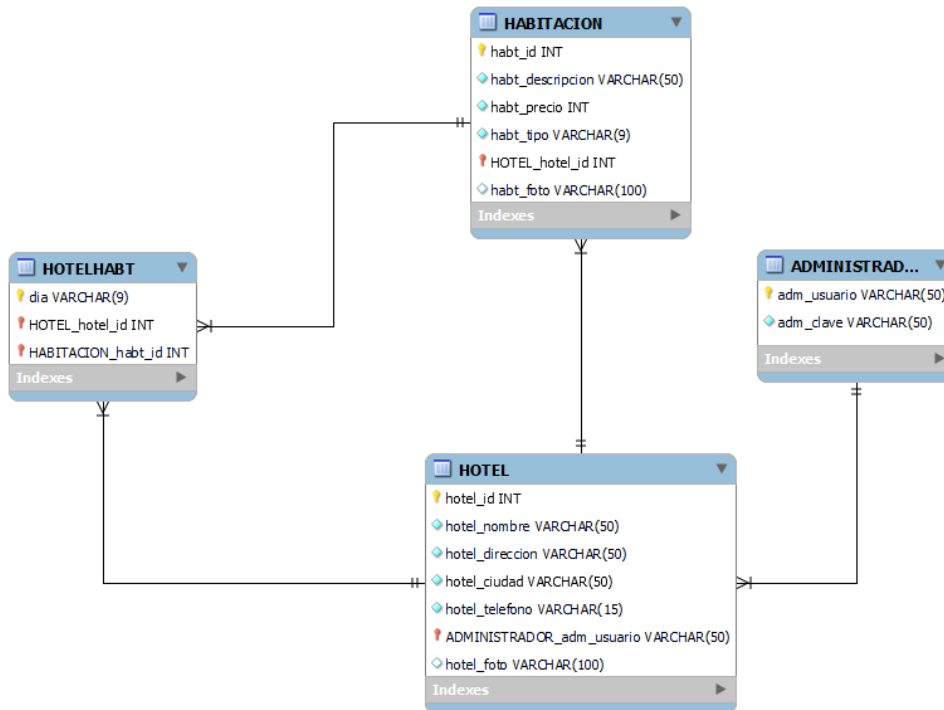
### VISTA DE IMPLEMENTACIÓN(MAVEN,UML).



### VISTA DE INSTALACIÓN(Allocation, UML).



## VISTA DE DATOS (MODELO ENTIDAD-RELACIÓN).

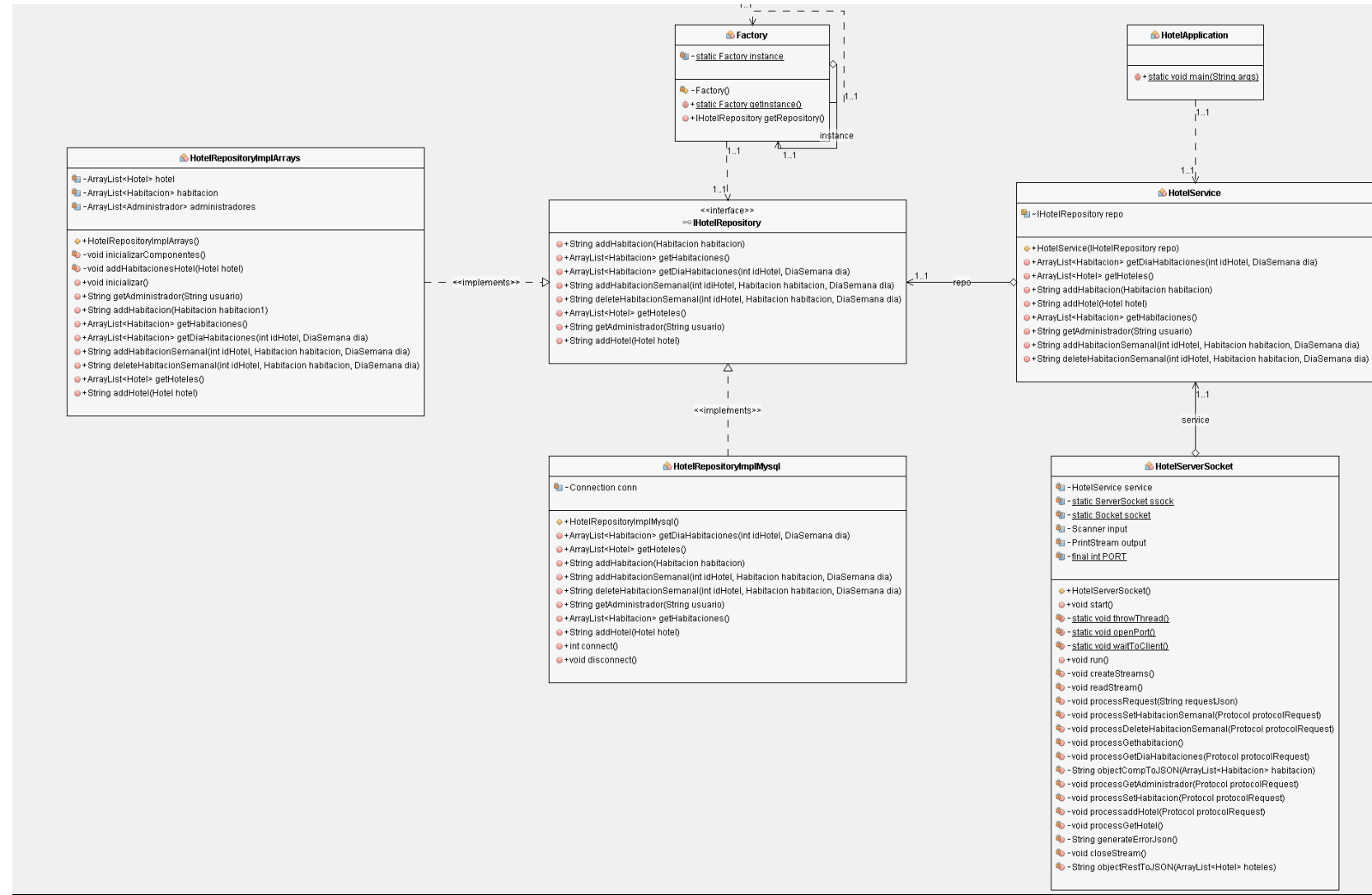


## RESUMEN DE LAS DECISIONES DE ARQUITECTURA Y SU JUSTIFICACIÓN.

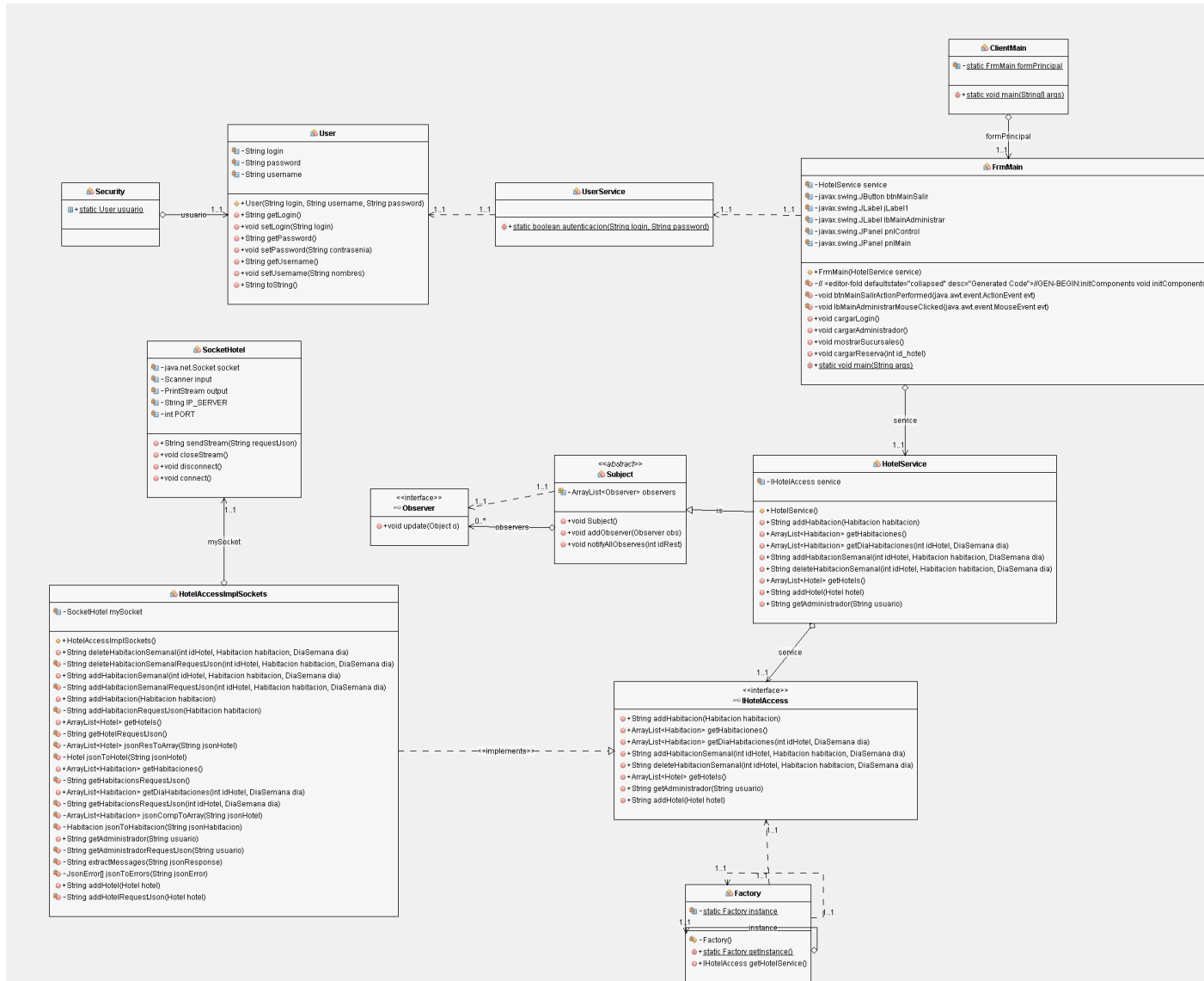
Para esta aplicación se ha optado por una aplicación de escritorio con posible migración (en un reléase superior) a aplicaciones móviles o web – Tomando como opción idónea la solución "Cliente - Servidor" para la primera iteración, esto debido a que solución "Cliente - Servidor" posee grandes ventajas que ayudarán al desarrollo de esta aplicación, como lo es el facilitar la integración con otros sistemas (característica necesaria para los pagos en línea, el envío de mensajes vía email y la utilización del GPS), además de garantizar una modularidad y escalabilidad del sistema (buscando ampliar las funcionalidades y aportar valor al cliente en cada iteración). Por su parte, como decisión de lenguaje de programación para trabajar, el equipo se inclina hacia el uso de "Java" debido a que, además de ser uno de los lenguajes más populares hoy en día, de acuerdo con el TIOBE index, el cual se puede encontrar en el siguiente enlace: <https://www.tiobe.com/tiobe-index/>, es un lenguaje de fácil comprensión, seguro y permite ser compilado en cualquier plataforma (gracias a la máquina virtual de Java "JVM"), otorgando al equipo y al producto final, el poder acoger una mayor cantidad de clientes (Clientes usuarios de Windows, clientes usuarios de Linux, clientes usuarios de MacOS, entre otros. Además la Tecnología multi-thread & sockets tcp: Esto nos permite recibir múltiples peticiones sin que el servidor colapse. Api swing java: nos permite crear interfaces gráficas las cuales son visualizables para el cliente, además favorece la usabilidad, y Mysql: Nos permite el registro de datos requeridos por la aplicación, incluyendo las búsquedas

## DISEÑO DETALLADO DE LOS MÓDULOS IMPLEMENTADOS

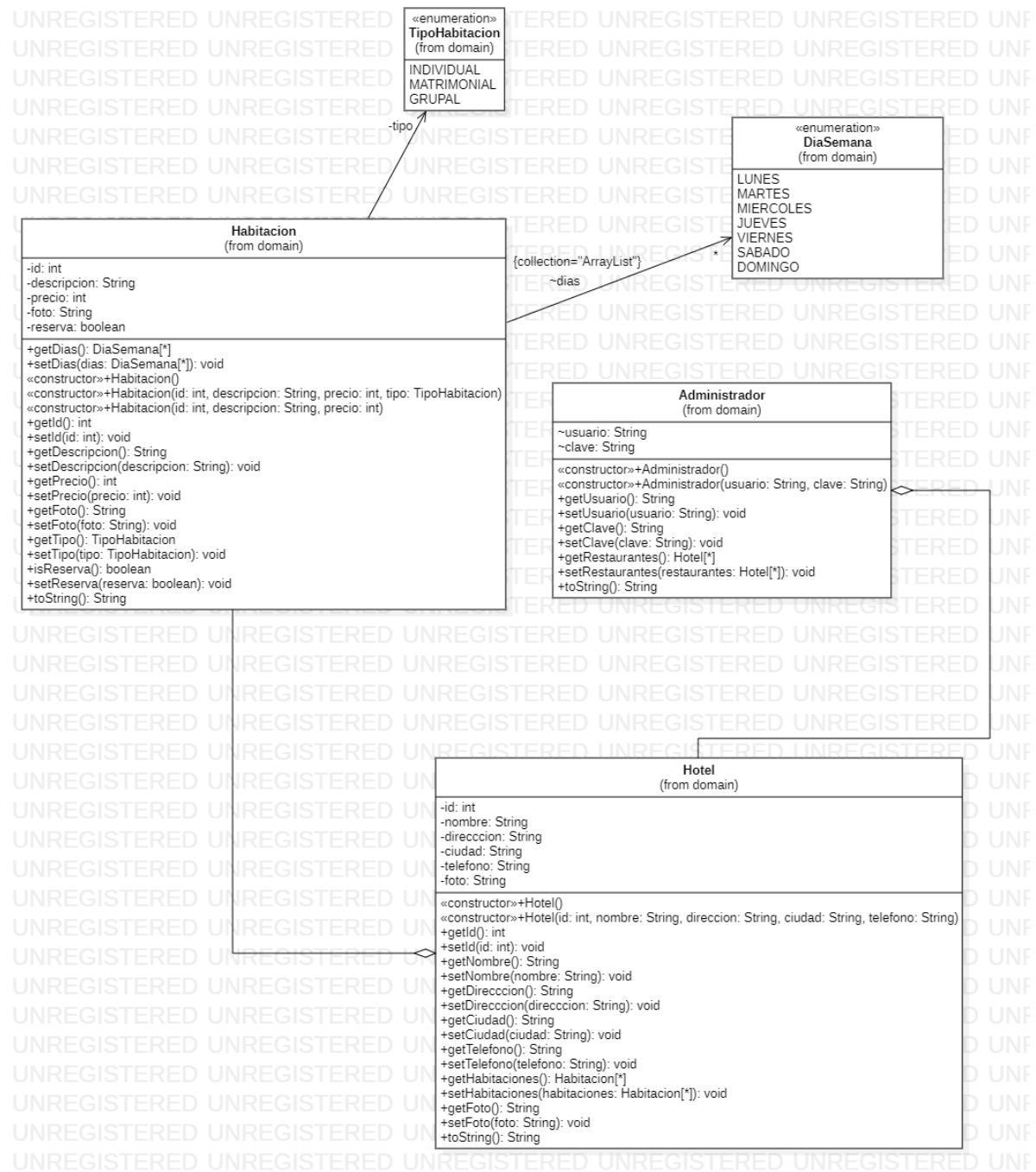
### Diagrama de Clases Servidor



## Diagrama de Clases Cliente

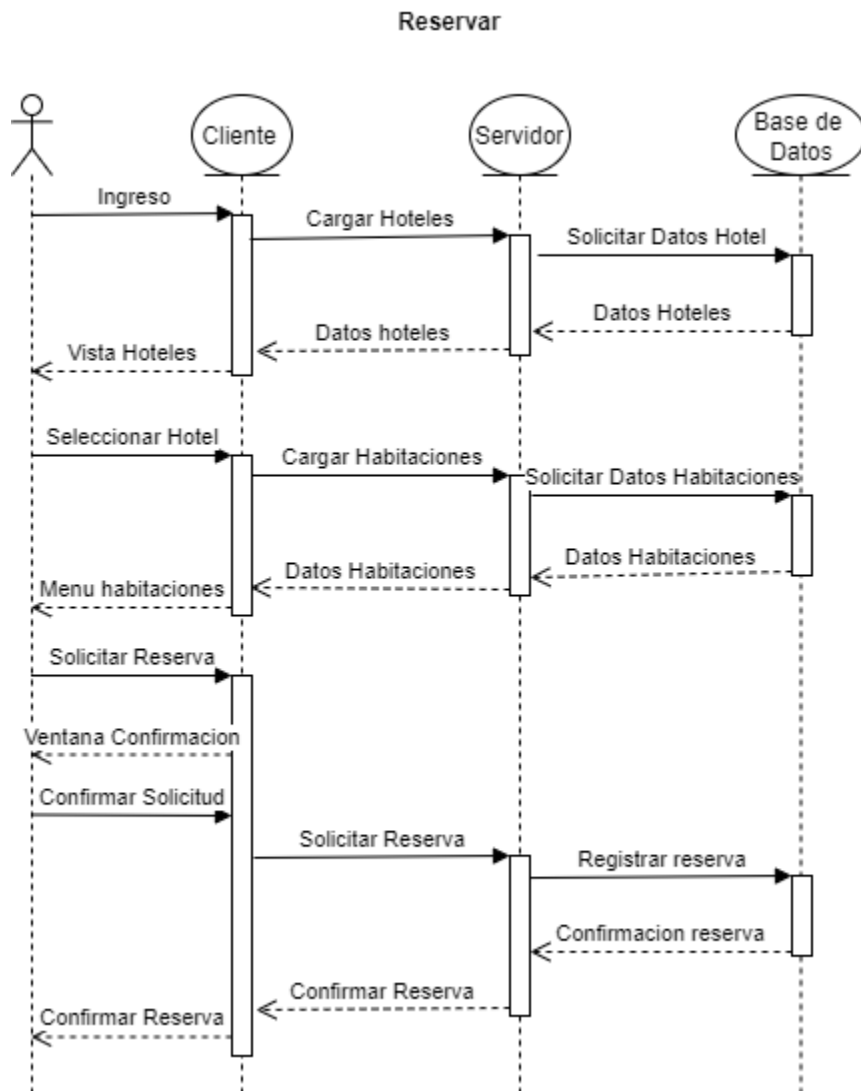


## Diagrama de clases Commons

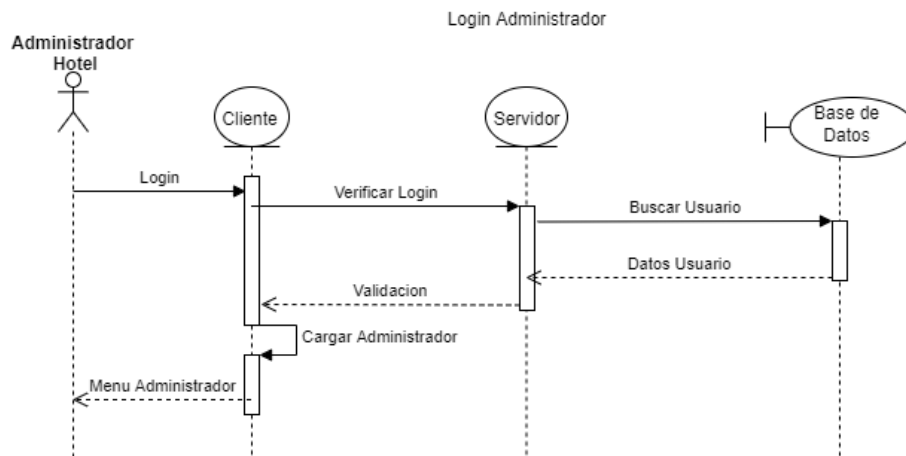


## DIAGRAMA DE SECUENCIA (UML)

### Reservar

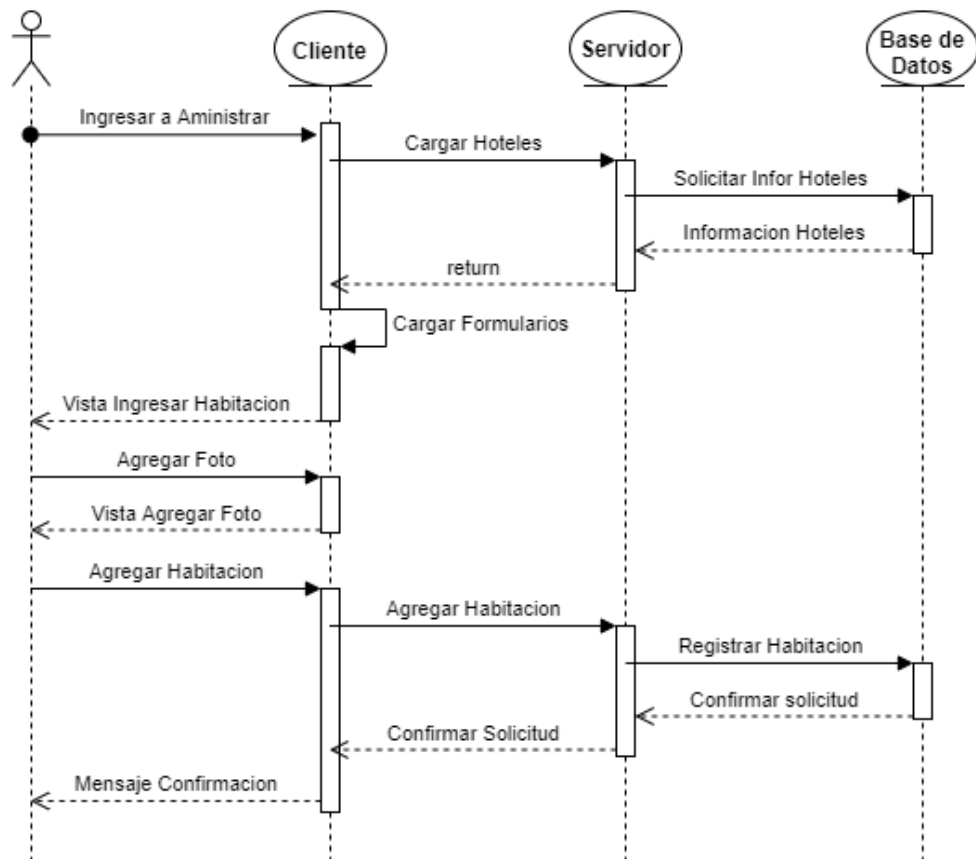


## Login



## Agregar Habitacion

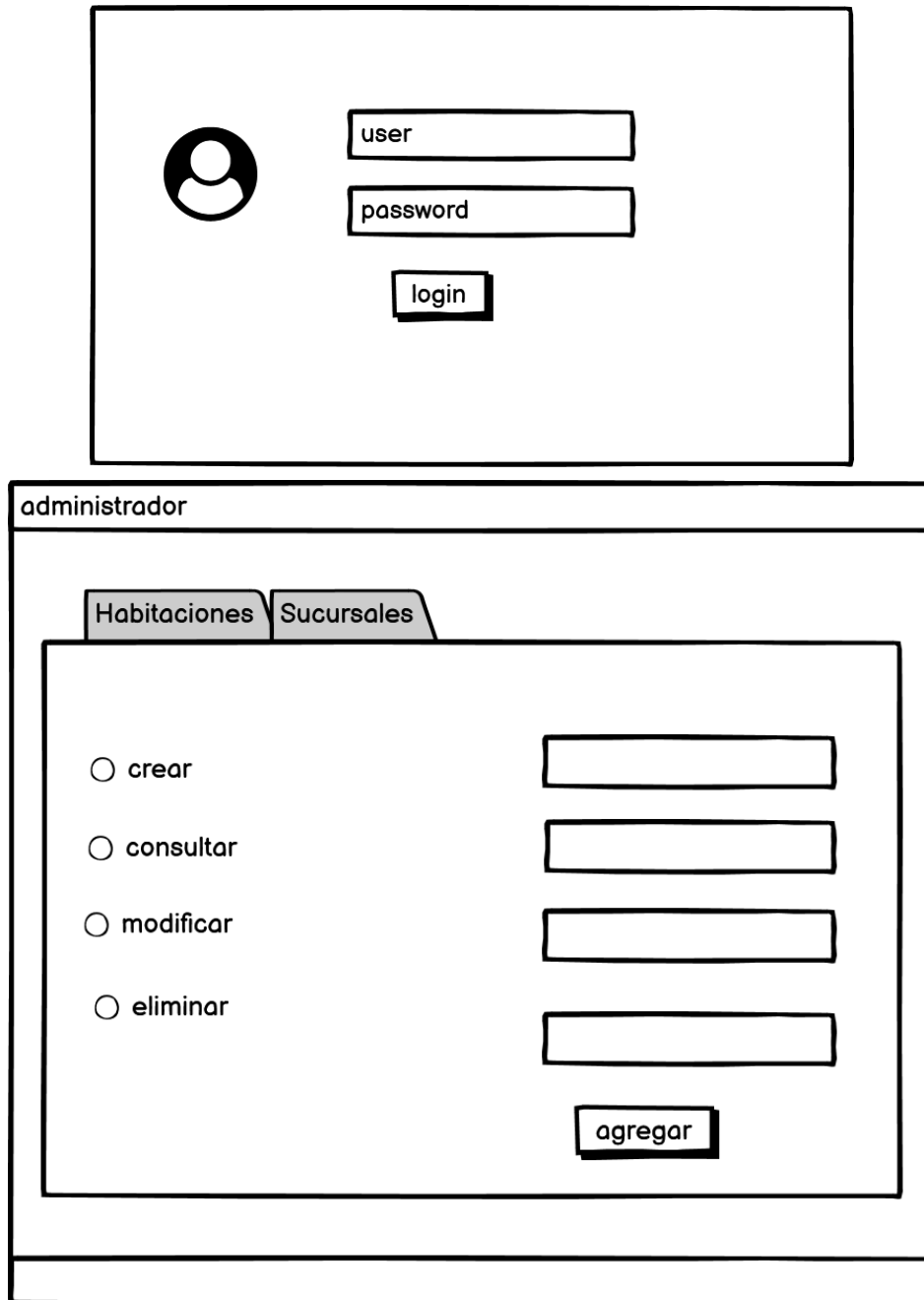
### Agregar Habitacion





## PROTOTIPOS DE INTERFAZ DE USUARIO.

[illegible]



**DIAGRAMA C4.**

<https://viewer.diagrams.net/?tags=%7B%7D&highlight=0000ff&edit=blank&layers=1&nav=1#G1KK5FHar5ojQgotwsEEo8B06qfaQbXyV>

**CÓDIGO FUENTE.**

<https://github.com/kevindm9/RoomReserve>