Proyecto VERANUM vista UML

****

Integrantes: Matias Mena, Leandro Ruiz y Cesar Veliz

Asignatura: Ingeniería en software

Caso de estudio: 1 Veranum

Docente: Jose Luis Silva

Semestre y año: Primer Semestre segundo año

ÍNDICE

[**Introducción 3**](#_m9bpwyz9ya46)

[**2. Contexto del Proyecto 3**](#_hf0kq05apn7c)

[**3. Técnicas de Modelamiento UML Aplicadas 4**](#_11ldoho80dem)

[3.1 Diagrama de Casos de Uso 4](#_eozh7ehhot50)

[3.2 Diagrama de Actividad 5](#_guctb27zpsib)

[3.3 Diagrama de Estados 6](#_ohff7eean6t6)

[3.4 Diagrama de Componentes 8](#_686eeu9p3j1)

[3.5 Diagrama de secuencia 9](#_3jnyqkz3bwqn)

[3.6 Diagrama de despliegue 10](#_xsqlqjvmsd1f)

[**4. Modelo 4+1 11**](#_3r8ojz5phms7)

[4.1. Vista Lógica 11](#_aha0knxgn29z)

[**4.2. Vista de Desarrollo 12**](#_9qb00vruu7tr)

[**4.3. Vista de Procesos 13**](#_c9ytkbtnwja1)

[**4.4 Vista Física 14**](#_bcb4n2edqxcg)

[4.5. +1. Vista de Casos de Uso 15](#_ovn4hs8af8n)

[**5. Estándares de Calidad Aplicados 16**](#_cx1xztwzycjk)

[Reservas en Línea 16](#_egb6t92i56lx)

[Gestión de Usuarios 16](#_gwyorddboaye)

# 

# Introducción

Este informe tiene como propósito mostrar el trabajo realizado en el diseño del sistema para la empresa hotelera VERANUM. Para esto se usaron diagramas UML que nos ayudaron a representar cómo funcionará el sistema, qué partes lo componen y cómo se relacionan entre sí.

También se aplicó el modelo arquitectónico 4+1, que nos permitió ver el sistema desde distintas miradas, como la parte lógica, física y de casos de uso. A lo largo del desarrollo, se consideraron principios básicos de calidad en el diseño, con el objetivo de que el sistema sea claro, fácil de mantener y adaptable en el futuro.

# 2. Contexto del Proyecto

La empresa hotelera VERANUM cuenta con dos hoteles en Santiago y la Quinta Región. Esta empresa ofrece servicios de alojamiento, alimentación, lavandería y arriendo de lugares para realizar eventos. El problema que ha surgido es que la empresa ha perdido competitividad debido a su limitada presencia digital, una gestión poco eficiente y la falta de servicios modernos que demanda el cliente actual. Esto ha provocado una baja en las ventas.

La solución propuesta es desarrollar un sistema informático que permita gestionar reservas en línea, publicar promociones, controlar el stock del restaurante y administrar servicios adicionales.

Las funcionalidades claves que presenta el sistema es la reservas online con validación de disponibilidad, la gestión de clientes, mostrar las habitaciones y eventos de manera actualizada, administrar las promociones, control de stock del restaurante y que genere reportes de uso y/o ventas.

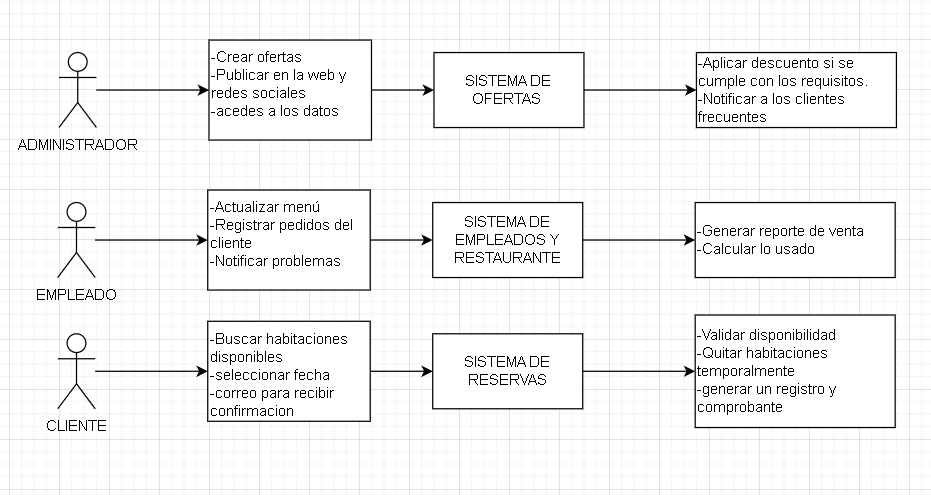
# 3. Modelamiento UML

En el desarrollo del sistema para “VERANUM” fuimos trabajando con distintos diagramas UML que nos ayudaron a organizar mejor las ideas, entender cómo debía funcionar el sistema y definir bien cada parte. Estos diagramas nos sirvieron tanto para pensar la lógica del sistema como para que después los programadores puedan construirlo de forma clara.

## 

## 3.1 Diagrama de Casos de Uso

Este diagrama muestra las acciones principales que pueden realizar los distintos tipos de usuarios del sistema VERANUM, como clientes o administradores. Es útil para visualizar de forma general qué funciones ofrece el sistema y quién las utiliza.



## 

## 3.2 Diagrama de Actividad

El diagrama de actividad representa el paso a paso de cómo se llevan a cabo ciertos procesos dentro del sistema. Este diagrama nos ayuda a entender el flujo de trabajo y mejorar la eficiencia del sistema.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

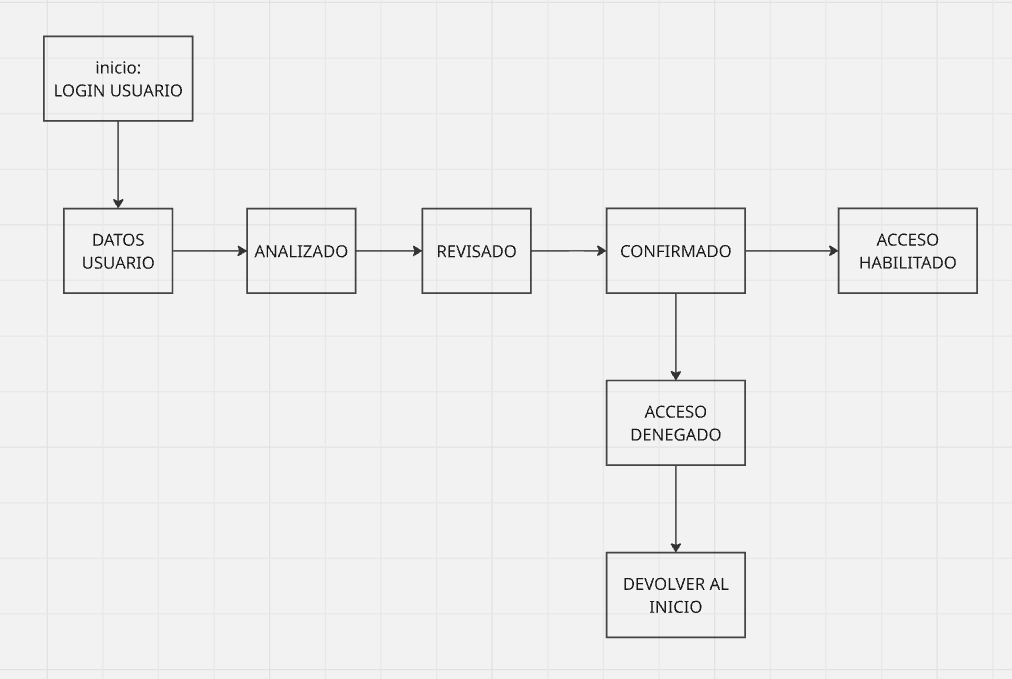
## 

## 

## 

## 3.3 Diagrama de Estados

El diagrama de estado nos ayuda a entender de forma general cómo cambian ciertos elementos del sistema según las acciones del usuario o del sistema. Es útil para visualizar el ciclo de vida del sistema.



## 

## 

## 

## 

## 

## 

## 

## 

## 

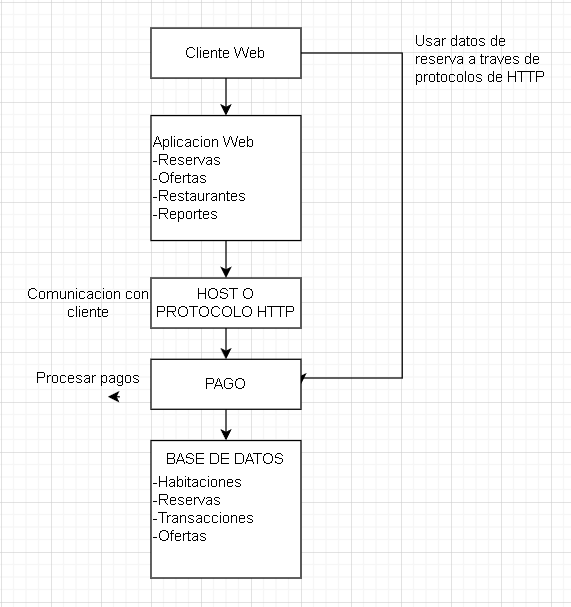
## 

## 

## 

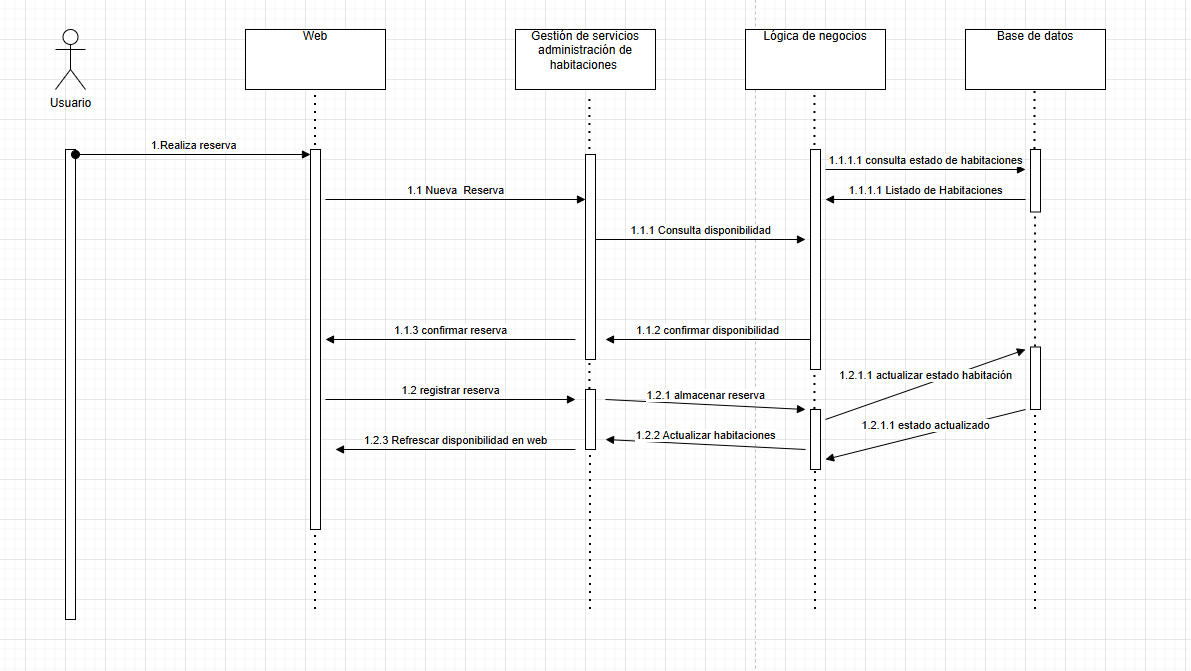
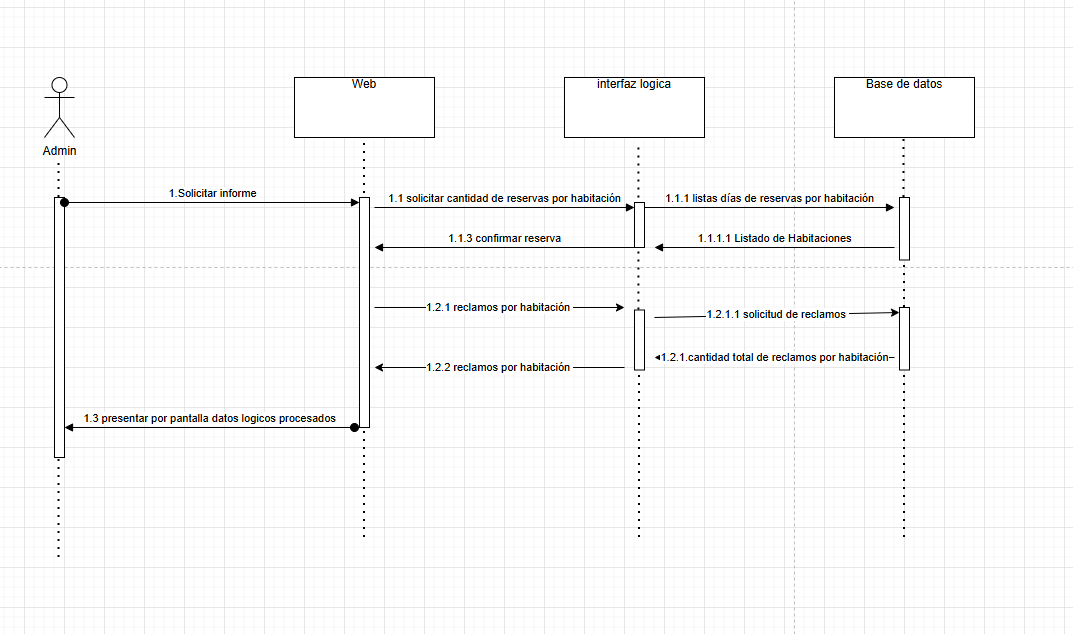
## 3.4 Diagrama de Componentes

El diagrama de componentes nos muestra cómo está dividido el sistema en módulos o bloques funcionales, como gestión de usuarios, reservas o promociones. Es clave para que los programadores sepan cómo organizar el código y qué hace cada parte del sistema.Los cuales pasan por el host y los datos o información dependiendo de los



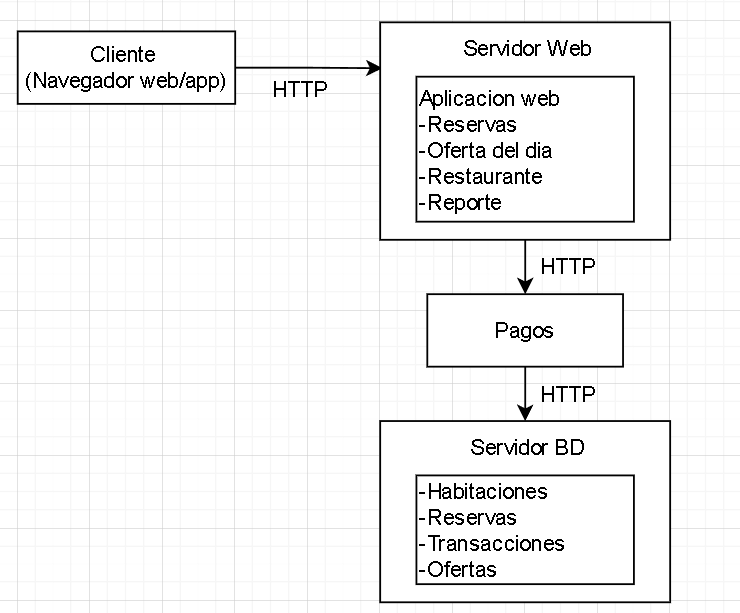
## 3.5 Diagrama de secuencia

El diagrama de secuencia nos permite ver y entregar un orden en que se comunican los distintos módulos del sistema para cumplir una función. Ayuda a entender la lógica interna del sistema.



## 3.6 Diagrama de despliegue

El diagrama de despliegue nos permite representar en qué partes físicas se instala el sistema, como el servidor o el navegador del cliente. Es útil para planificar la infraestructura técnica y saber cómo se conectan los distintos dispositivos.



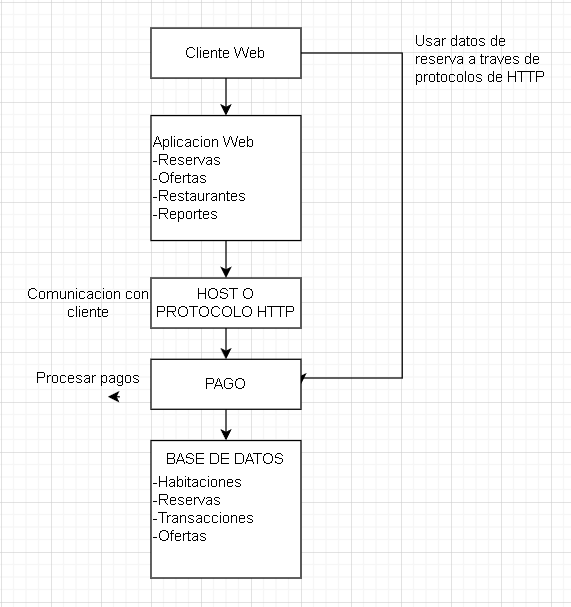
4. Modelo 4+1

El modelo 4+1 se llama así porque está compuesto por cuatro vistas las cuales son lógica, desarrollo, procesos y física más una vista adicional: la de casos de uso.

Este modelo nos sirve para diseñar sistemas más claros, estructurados y fáciles de mantener, ya que permite analizar el software desde todos los ángulos importantes.

## 4.1. Vista Lógica

En este caso, la vista lógica se representa a través del diagrama de componentes, ya que nos permite ver cómo está organizado el sistema por dentro. En la imagen se muestra claramente cómo cada módulo cumple una función específica, como gestionar reservas, promociones o usuarios. Esta vista ayuda a entender la estructura lógica del sistema y cómo se conectan sus diferentes partes para que todo funcione correctamente.



# 

## 4.2. Vista de Desarrollo

En este caso usamos el mismo diagrama que en este caso es el diagrama de componentes para explicar la vista de desarrollo, pero nos estamos enfocando en cómo se va a construir el sistema. Acá se ve cómo se divide el trabajo para los programadores, ya que cada módulo puede desarrollarse por separado. Esto facilita la organización del equipo y hace que mantener el sistema sea más fácil en el futuro.

# 

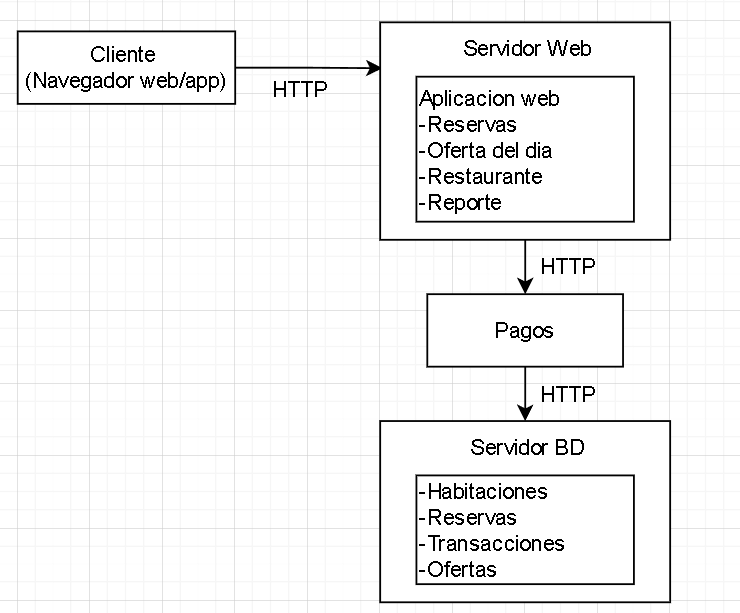
## 4.3. Vista de Procesos

En este proyecto, la vista de procesos se representa con el diagrama de actividad, ya que muestra paso a paso cómo fluye la información dentro del sistema cuando se realiza una acción, como hacer una reserva o aplicar una promoción. Esto permite entender el comportamiento interno del sistema cuando está en funcionamiento.

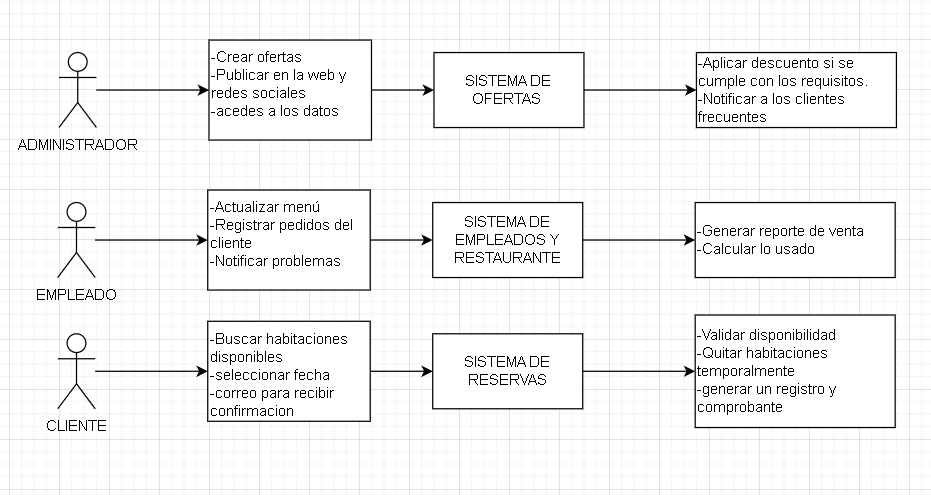
# 

## 4.4 Vista Física

En la vista física se representa con el diagrama de despliegue, ya que muestra en qué lugares vive el sistema, por ejemplo, se ejecuta en el navegador del cliente, el servidor web y dónde está la base de datos. Esto ayuda a visualizar la infraestructura técnica y cómo se conectan todos los elementos.



## 4.5. +1. Vista de Casos de Uso

En este proyecto, la vista de casos de uso se representa con el diagrama de casos de uso, ya que muestra las funciones principales del sistema y cómo interactúan los diferentes usuarios con él. Es una vista clave para entender qué necesita el sistema cubrir desde el punto de vista del cliente o del personal del hotel.

# 5. **Estándares de Calidad Aplicados**

Como parte del análisis de calidad y usabilidad del sistema propuesto para la empresa hotelera VERANUM, se aplicó el método heurístico de Nielsen. Este método se basa en 10 principios de usabilidad reconocidos, que permiten identificar mejoras desde el punto de vista del usuario.

A continuación, se evalúa cada ítem del product backlog según estas heurísticas:

### **Reservas en Línea**

Descripción: Consultar disponibilidad, reservar habitaciones y recibir confirmación.  
 Heurísticas Aplicadas:

* Visibilidad del estado del sistema: Confirmaciones visibles y actualizaciones en tiempo real.
* Prevención de errores: Restricción de fechas inválidas o incompletas.
* Diseño estético y minimalista: Formularios claros, solo los campos necesarios.

### **Gestión de Usuarios**

Descripción: Crear cuentas, iniciar sesión, actualizar información.  
 Heurísticas Aplicadas:

* Reconocimiento antes que recuerdo: Recordar usuario o permitir autocompletar.
* Control y libertad del usuario: Función de “recuperar contraseña”.
* Consistencia y estándares: Diseño común en formularios de acceso.