

1. Introducción:

El propósito fundamental de este proyecto es desarrollar un sistema web dedicado al registro de reservaciones y control de hospedaje en un entorno hotelero. Este sistema tiene como finalidad brindar a los usuarios la capacidad de efectuar reservas, administrar servicios y mantener un eficaz seguimiento de clientes, habitaciones y empleados en el hotel. Todo esto será llevado a cabo mediante la implementación de la metodología Scrum.

2. Justificación:

El proyecto responde a la creciente demanda de soluciones tecnológicas en el sector hotelero. La necesidad de optimizar los procesos asociados a las reservaciones y el control de hospedaje, junto con la oportunidad de desarrollar habilidades profesionales, fundamenta la relevancia de este proyecto. Aspira a ofrecer una solución integral que mejore la eficiencia y la experiencia del cliente en el ámbito hotelero.

3. Objetivos

3.1. Objetivo General:

Concebir e implementar un sistema web avanzado para la gestión integral de reservaciones y el control de hospedaje en un entorno hotelero. Este sistema, basado en tecnologías web como PHP, HTML, JavaScript, Bootstrap, y respaldado por una base de datos relacional en MySQL, tiene como objetivo primordial optimizar los procesos vinculados al registro, seguimiento y administración de reservas, garantizando una operatividad eficiente y una experiencia fluida para los usuarios.

3.2. Objetivos Específicos:

Diseñar la Arquitectura del Sistema:

Definir la estructura del sistema, identificando módulos y componentes necesarios para su correcto funcionamiento en el contexto hotelero.

Implementar la Base de Datos:

Crear la base de datos relacional, con las tablas necesarias para almacenar información de reservas, clientes, habitaciones y servicios, utilizando MySQL.

Desarrollar la Interfaz de Usuario (Front-end):

Implementar una interfaz de usuario interactiva y amigable, utilizando JavaScript, HTML y Bootstrap para mejorar la experiencia del usuario durante el proceso de reserva y consulta de información.

Desarrollar el Back-end:

Implementar la lógica de negocio del sistema utilizando PHP, asegurando la integración con la base de datos y proporcionando funcionalidades robustas para la gestión de reservas y control de hospedaje.

4. Metodología de Desarrollo - Scrum

El desarrollo del proyecto, centrado en un Sistema Web para el registro de reservaciones y control de hospedaje de un hotel, se llevará a cabo empleando la metodología Scrum.

4.1 Metodología Scrum:

Scrum, un marco de trabajo ágil, será implementado para orquestar el desarrollo de este proyecto. Este enfoque, comúnmente utilizado en el desarrollo de software y adecuado para equipos de diversas dimensiones, se organiza alrededor de ciclos de desarrollo iterativos e incrementales conocidos como "sprints".

5. Alcance y Desarrollo con Metodología Scrum

El alcance y desarrollo del Sistema Web para el registro de reservaciones y control de hospedaje de un hotel se guiará por la metodología Scrum, asegurando una gestión eficiente del proyecto. El enfoque Scrum se dividirá en distintas fases para garantizar un desarrollo iterativo y progresivo.

5.1 Definición del Producto:

En la fase inicial, el equipo definirá detalladamente los requisitos del sistema, identificando funcionalidades clave como el registro de reservaciones, la asignación de habitaciones, y la gestión de servicios asociados.

5.2 Backlog del Producto:

Se creará un backlog del producto que incluirá todas las funcionalidades priorizadas para el sistema. Cada elemento del backlog será evaluado y estimado en términos de complejidad y esfuerzo.

5.3 Planificación de Sprints:

Los sprints, periodos de desarrollo iterativos, serán planificados, teniendo en cuenta la capacidad del equipo y la prioridad de los elementos del backlog. Cada sprint se centrará en implementar un conjunto específico de funcionalidades.

5.4 Desarrollo:

Durante cada sprint, el equipo llevará a cabo el desarrollo de las funcionalidades asignadas, siguiendo las mejores prácticas de codificación y asegurando la integración con la base de datos. El trabajo será revisado y ajustado al final de cada sprint.

5.5 Revisiones y Retrospectivas:

Al concluir cada sprint, se llevarán a cabo revisiones para demostrar las funcionalidades implementadas y retrospectivas para identificar áreas de mejora en el proceso de desarrollo y rendimiento del equipo.

5.6 Entrega Incremental:

El producto será entregado de manera incremental al final de cada sprint, permitiendo una rápida retroalimentación y ajustes según las necesidades emergentes del cliente o del equipo.

5.7 Evaluación y Ajuste Continuo:

A lo largo del desarrollo, se realizarán evaluaciones continuas del progreso del proyecto y ajustes en la planificación según sea necesario, asegurando una adaptación efectiva a los cambios y una entrega exitosa del Sistema Web para el registro de reservaciones y control de hospedaje del hotel.

6. Backlog del Producto:

Registro y Autenticación:

El sistema debe permitir a los clientes registrarse y autenticarse de forma segura.

Los datos requeridos incluyen nombre, correo electrónico y contraseña.

Gestión de Productos:

Se deben registrar y listar productos disponibles en el sistema, incluyendo detalles como nombre, descripción, precio y categoría.

Carrito de Compras:

Los clientes pueden agregar productos al carrito de compras.

El carrito debe almacenarse en la tabla cart con información sobre el usuario, el producto, cantidad, etc.

Categorías de Productos:

Definir y mostrar categorías de productos en la interfaz.

Utilizar la tabla categorias para gestionar las categorías.

Habitaciones y Reservas:

Mantener un registro de habitaciones disponibles en la tabla habitaciones.

Los clientes pueden realizar reservas (reservar), especificando fechas de entrada y salida, y asociándolas con una habitación disponible.

Clientes:

Registrar información detallada del cliente en la tabla clientes.

Asociar reservas y productos a clientes mediante claves foráneas.

Facturación y Pagos:

Implementar la tabla orders para registrar pedidos de productos y habitaciones.

Incluir información sobre método de pago, estado del pago y detalles de facturación.

Administración de Pisos y Habitaciones:

Definir pisos en la tabla pisos y categorías de habitaciones en la tabla hcate.

Asociar habitaciones con pisos y categorías.

Historial de Reservas:

Mantener un historial de reservas en la tabla rs_history para referencia futura.

Interfaz Administrativa:

Crear una interfaz administrativa para la gestión de usuarios, productos, habitaciones y pedidos.

7. Sprints

Sprint 1: Registro y Autenticación (Duración: 2 semanas)

Tarea 1: Configuración de la Interfaz de Registro

Desarrollar la interfaz de registro de usuarios.

Validar y almacenar la información en la tabla usuarios.

Tarea 2: Implementación de la Autenticación

Desarrollar el sistema de autenticación para permitir que los usuarios accedan al sistema de reservas.

Sprint 2: Gestión de Productos (Duración: 3 semanas)

Tarea 1: Creación de la Interfaz de Administración de Productos

Desarrollar una interfaz para que el personal del hotel pueda agregar, editar y eliminar productos.

Tarea 2: Implementación del CRUD de Productos

Habilitar la funcionalidad completa de Crear, Leer, Actualizar y Eliminar productos en la base de datos (productos).

Sprint 3: Carrito de Compras y Categorías de Productos (Duración: 2 semanas)

Tarea 1: Desarrollo del Carrito de Compras

Permitir a los usuarios agregar productos al carrito y visualizar su contenido.

Almacenar la información del carrito en la tabla cart.

Tarea 2: Gestión de Categorías de Productos

Desarrollar la interfaz y la lógica para asignar productos a categorías.

Utilizar la tabla categorias para organizar productos.

Sprint 4: Habitaciones y Reservas (Duración: 4 semanas)

Tarea 1: Administración de Habitaciones

Desarrollar una interfaz para gestionar habitaciones y pisos (habitaciones y pisos).

Asociar habitaciones con categorías y pisos.

Tarea 2: Funcionalidad de Reservas

Implementar la capacidad de realizar reservas (reservar), especificando fechas de entrada y salida, y asociándolas con una habitación disponible.

Sprint 5: Clientes y Facturación (Duración: 3 semanas)

Tarea 1: Gestión de Clientes

Desarrollar una interfaz para registrar y gestionar información detallada del cliente (clientes).

Asociar reservas y productos a clientes mediante claves foráneas.

Tarea 2: Facturación y Pagos

Implementar la tabla orders para registrar pedidos de productos y habitaciones.

Incluir información sobre método de pago, estado del pago y detalles de facturación.

Sprint 6: Interfaz Administrativa y Mejoras (Duración: 2 semanas)

Tarea 1: Creación de la Interfaz Administrativa

Desarrollar una interfaz administrativa para la gestión de usuarios, productos, habitaciones y pedidos.

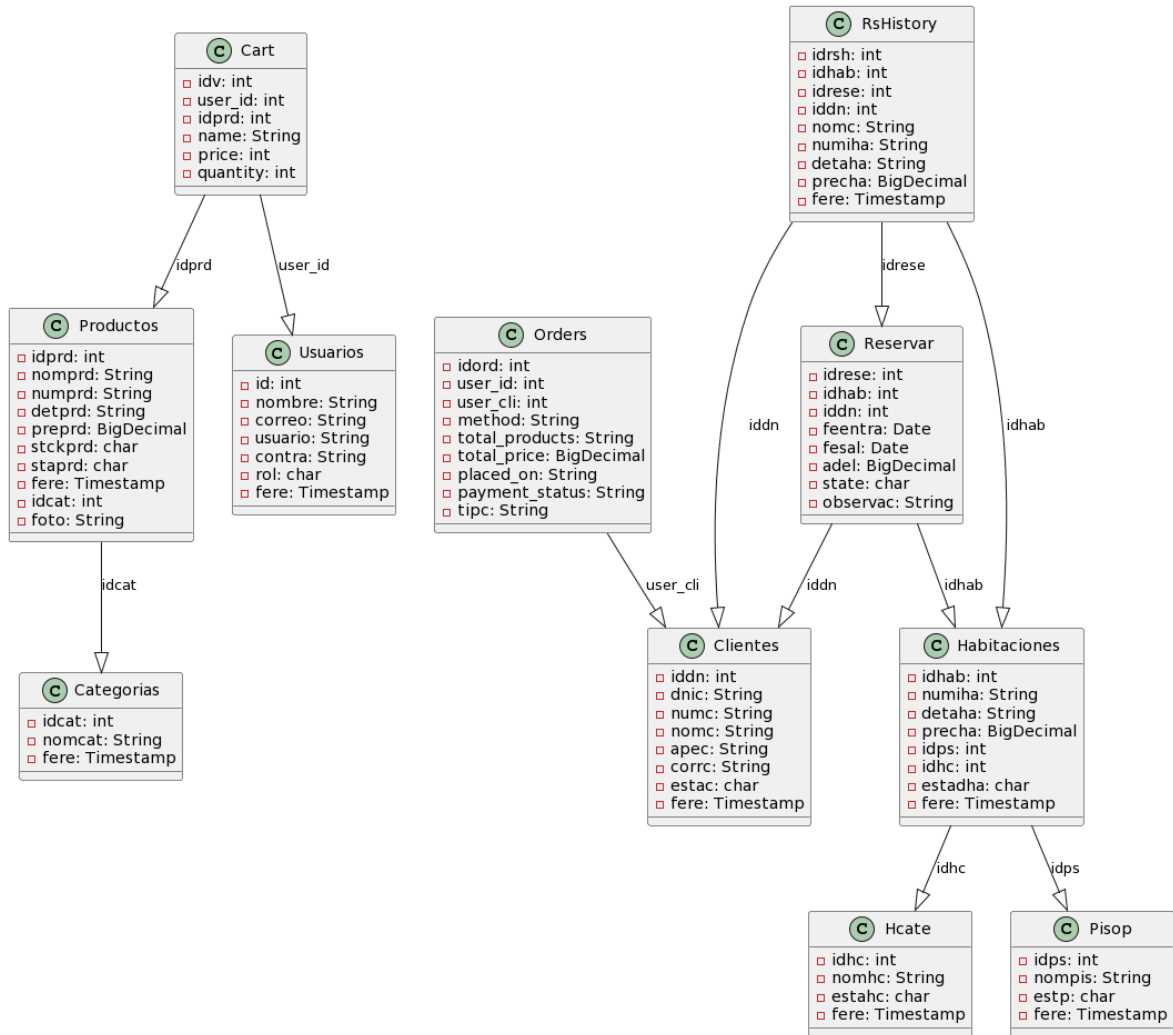
Tarea 2: Implementación de Mejoras

Revisar la funcionalidad existente y realizar mejoras según la retroalimentación del usuario.

Asegurarse de que la aplicación sea intuitiva y fácil de usar.

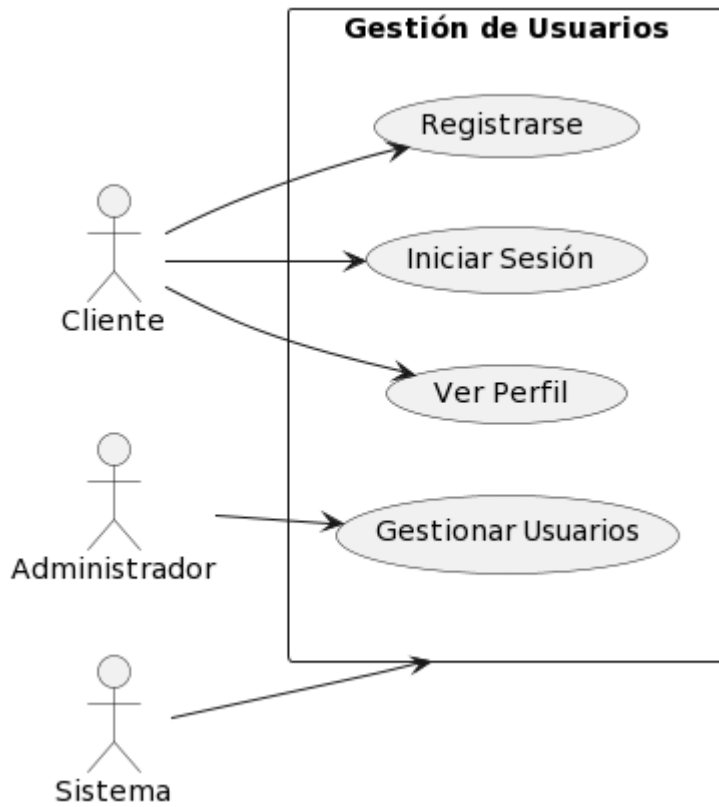
9. Diagramas

9.1. Diagrama de clases

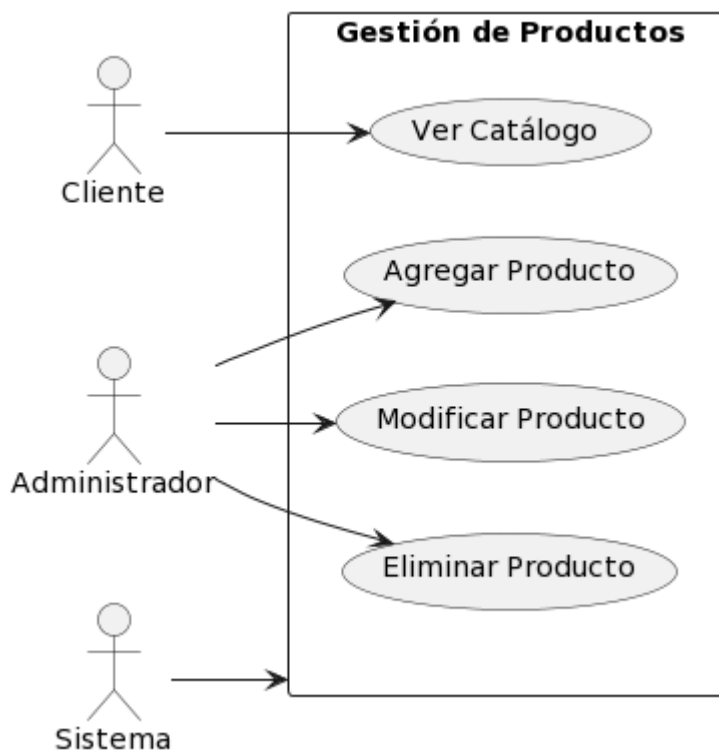


9.2. Diagrama de casos de uso

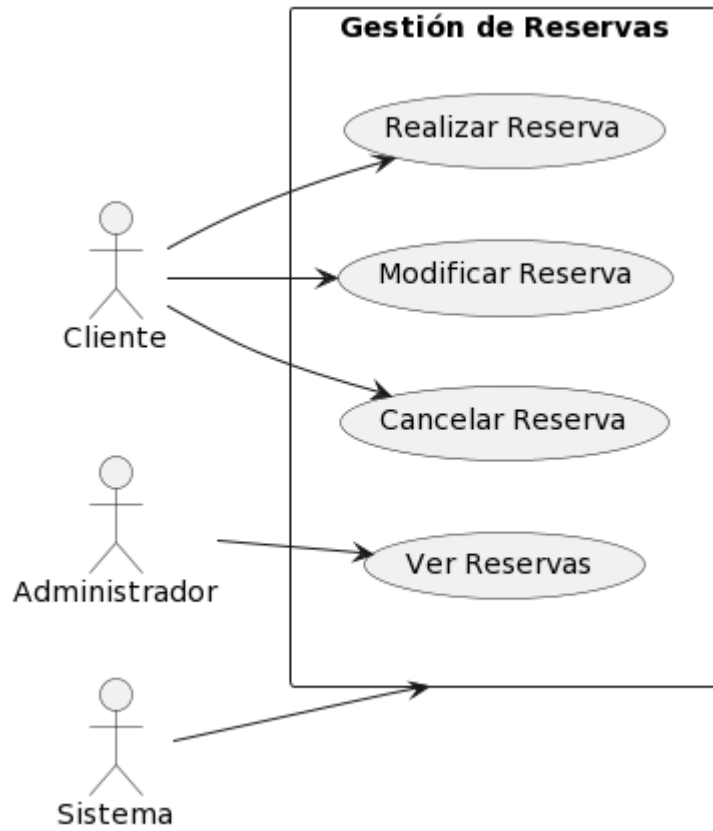
9.2.1 Diagrama de casos de uso (Gestión de usuarios)



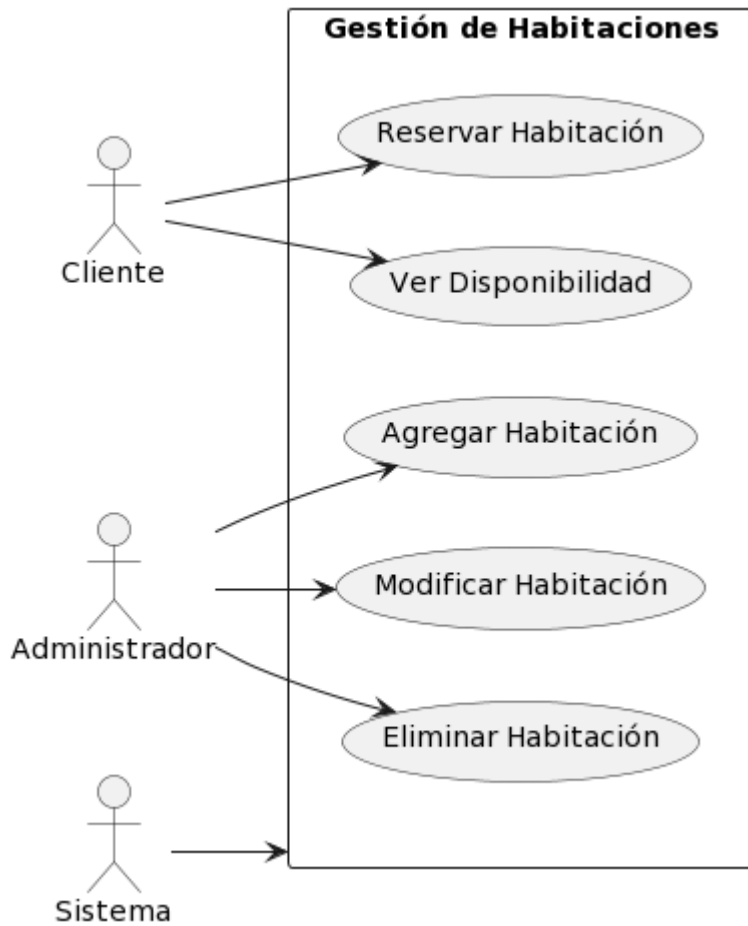
9.2.2. Diagrama de casos de uso (Gestión de productos)



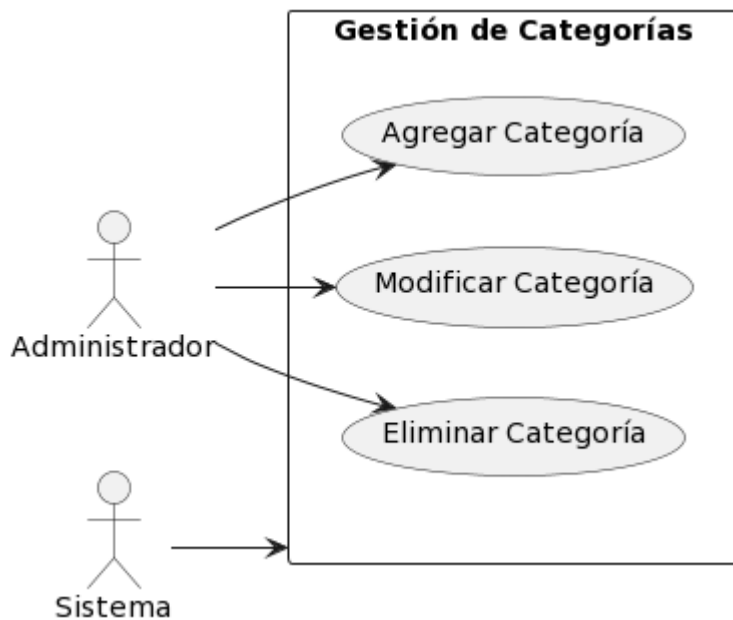
9.2.3. Diagrama de casos de uso (Gestión de Reservas)



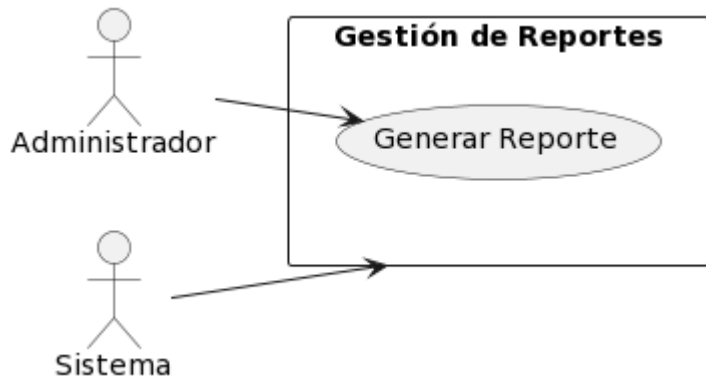
9.2.4. Diagrama de casos de uso (Gestión de Habitaciones)



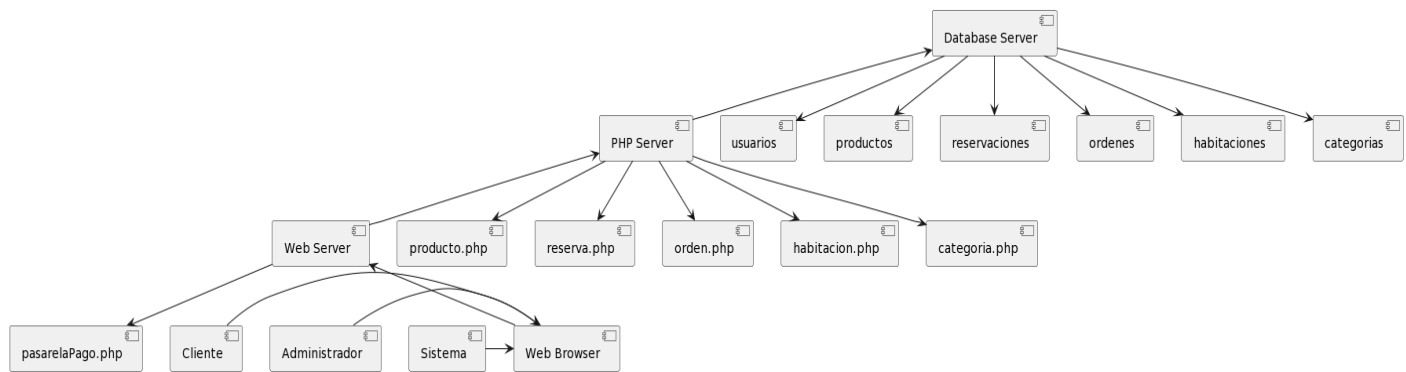
9.2.5 Diagrama de casos de uso (Gestión de categorías)



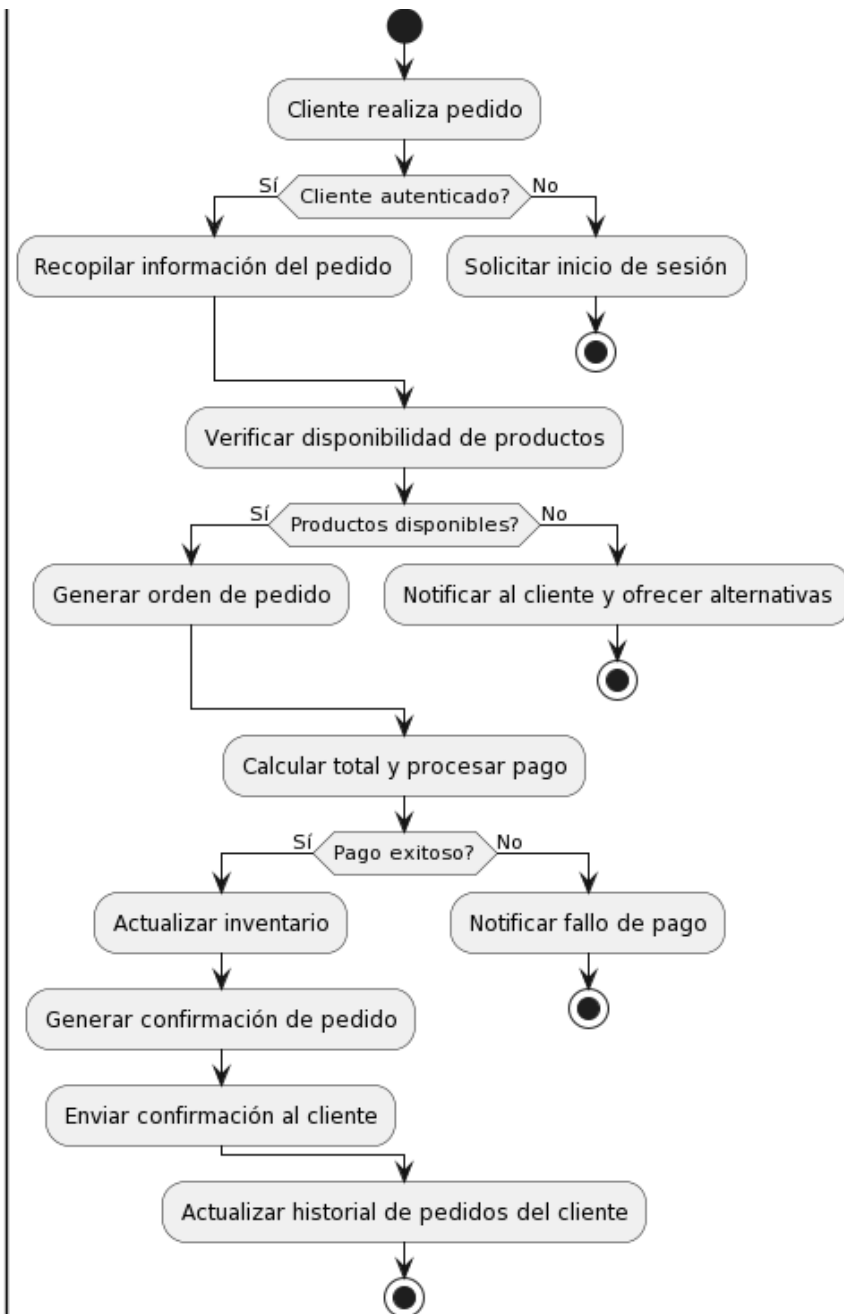
9.2.6. Diagrama de casos de uso (Gestión de Reportes)



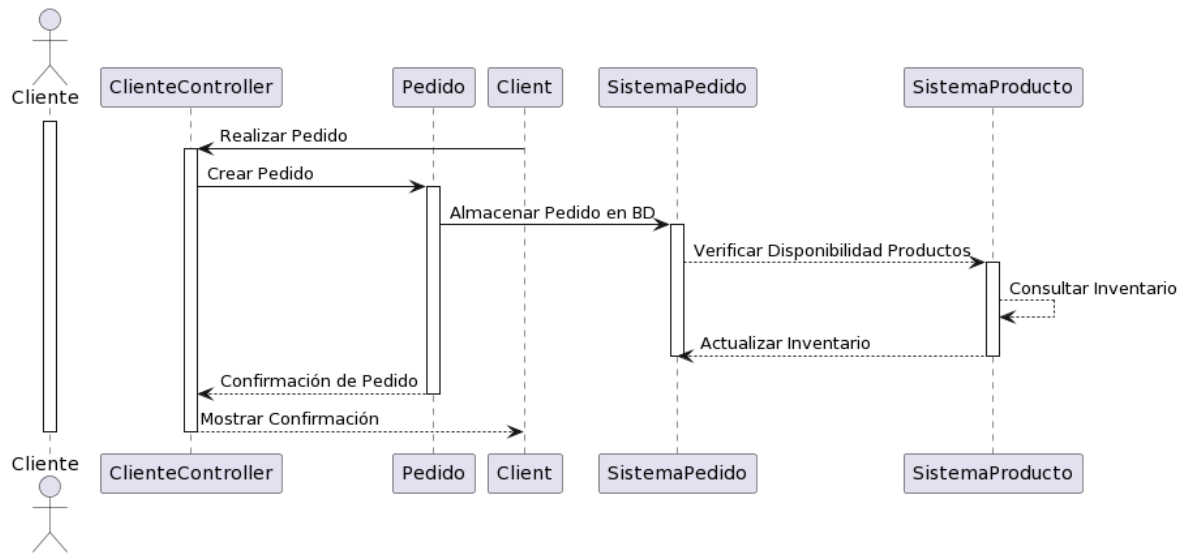
9.3. Diagrama de componentes



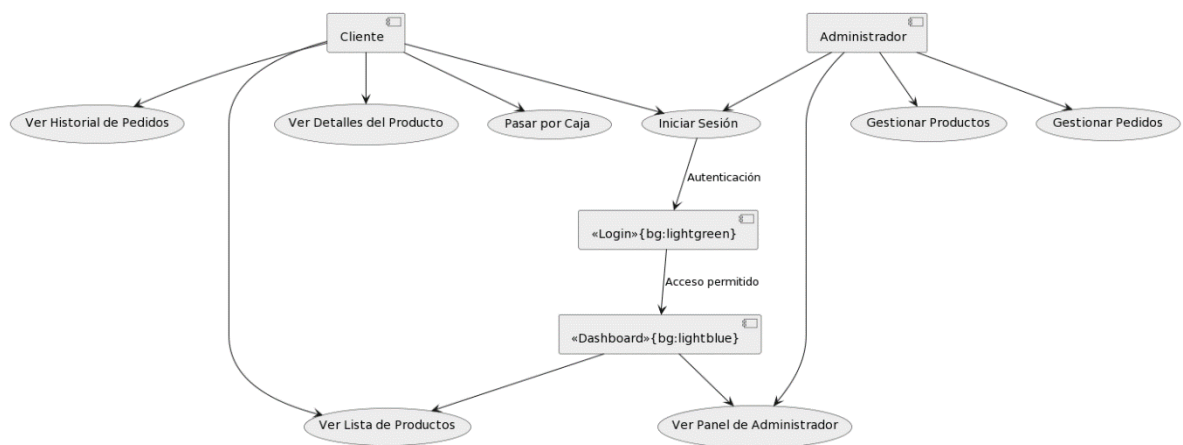
9.4. Diagrama de actividades



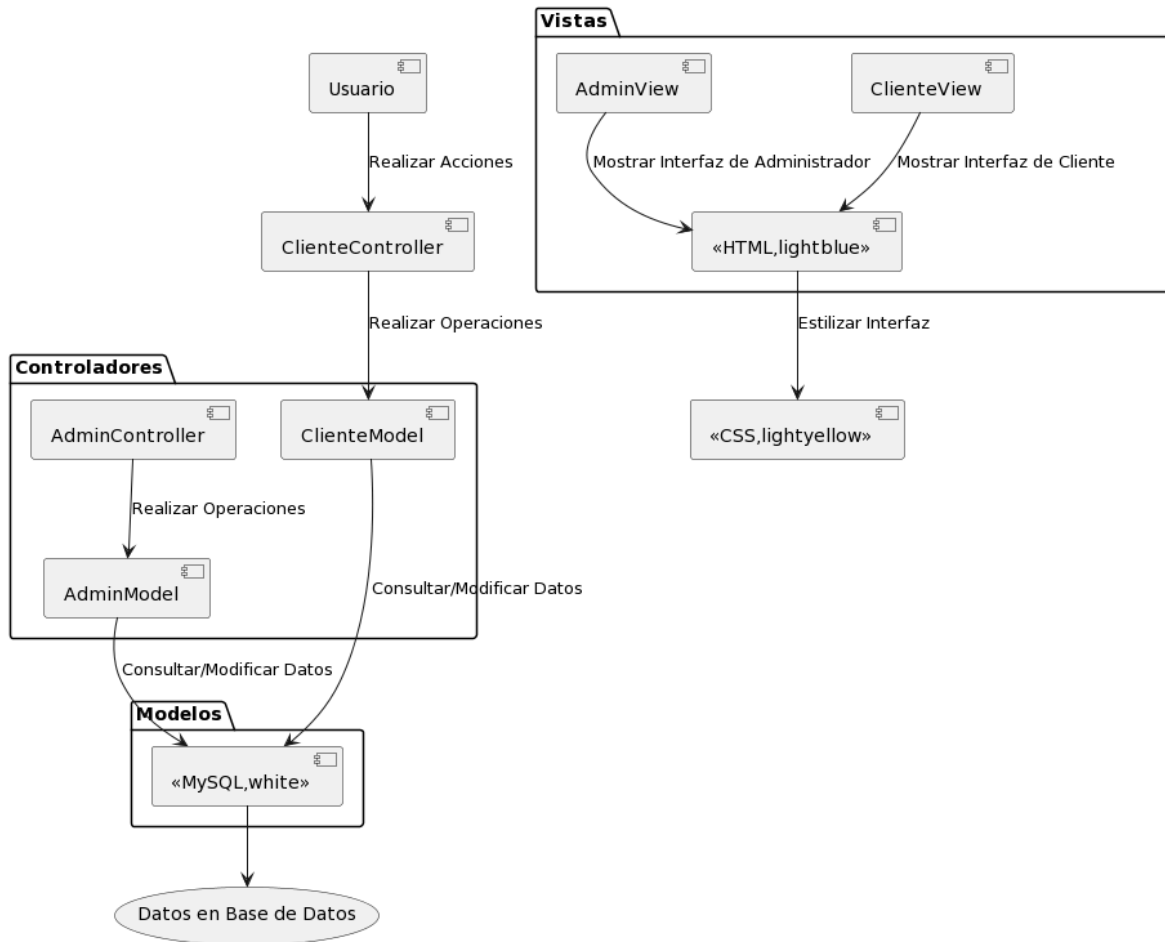
9.5. Diagrama de secuencia



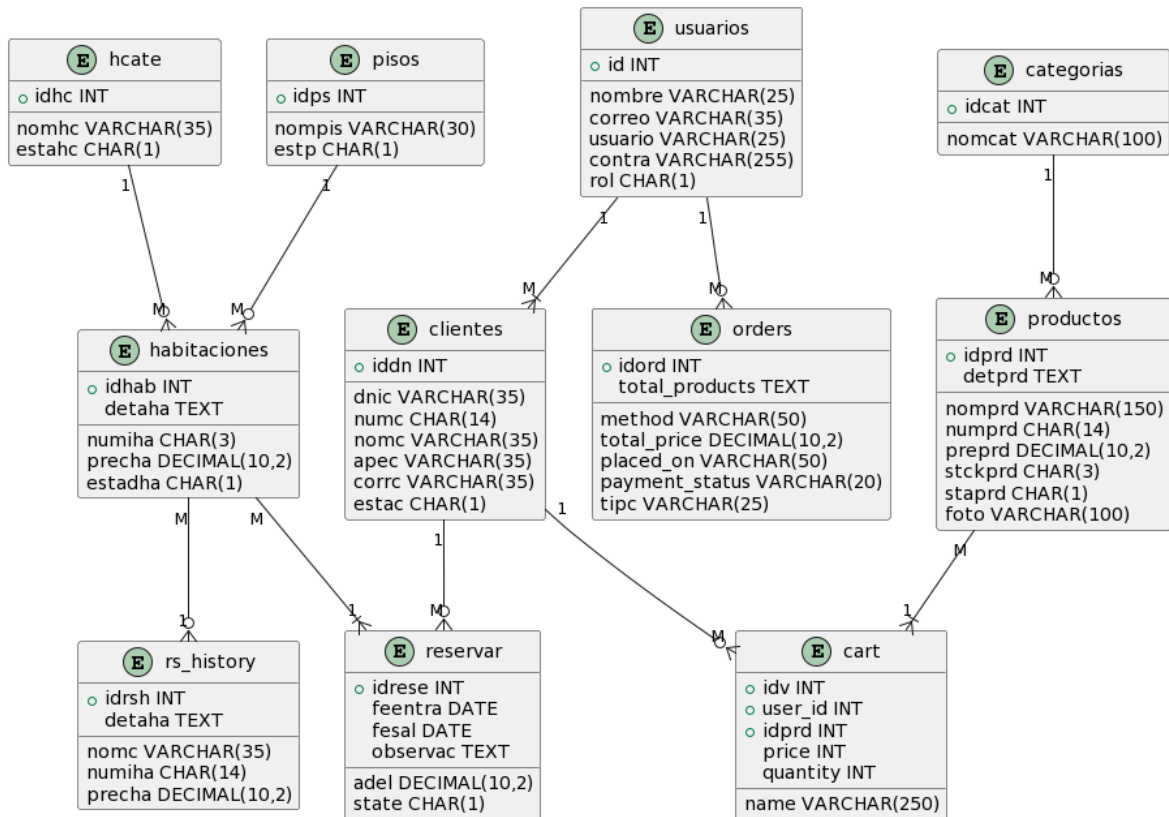
9.6. Diagrama de navegacional



9.7. Diagrama de implementación MVC



9.8. Diagrama de entidad relación



10. Diccionario de datos

cart:

idv: int(11) (clave primaria)

user_id: int(11) (clave foránea referenciando a usuarios.id)

idprd: int(11) (clave foránea referenciando a productos.idprd)

name: varchar(250)

price: int(11)

quantity: int(11)

categorias:

idcat: int(11) (clave primaria)

nomcat: varchar(100)

fer: timestamp

clientes:

iddn: int(11) (clave primaria)

dnic: varchar(35)

numc: char(14)

nomc: varchar(35)

apec: varchar(35)

corrc: varchar(35)

estac: char(1)

fere: timestamp

habitaciones:

idhab: int(11) (clave primaria)

numiha: char(3)

detaha: text

precha: decimal(10,2)

idps: int(11) (clave foránea referenciando a pisos.idps)

idhc: int(11) (clave foránea referenciando a hcate.idhc)

estadha: char(1)

fere: timestamp

hcate:

idhc: int(11) (clave primaria)

nomhc: varchar(35)

estahc: char(1)

fere: timestamp

orders:

idord: int(11) (clave primaria)

user_id: int(11) (clave foránea referenciando a usuarios.id)
user_cli: int(11) (clave foránea referenciando a clientes.iddn)
method: varchar(50)
total_products: text
total_price: decimal(10,2)
placed_on: varchar(50)
payment_status: varchar(20)
tipc: varchar(25)

pisos:

idps: int(11) (clave primaria)
nompis: varchar(30)
estp: char(1)
fere: timestamp

productos:

idprd: int(11) (clave primaria)
nomprd: varchar(150)
numprd: char(14)
detprd: text
preprd: decimal(10,2)
stckprd: char(3)
staprd: char(1)
fere: timestamp
idcat: int(11) (clave foránea referenciando a categorias.idcat)
foto: varchar(100)

reservar:

idrese: int(11) (clave primaria)

idhab: int(11) (clave foránea referenciando a habitaciones.idhab)

iddn: int(11) (clave foránea referenciando a clientes.iddn)

feentra: date

fesal: date

adel: decimal(10,2)

state: char(1)

observac: text

rs_history:

idrsh: int(11) (clave primaria)

idhab: int(11) (clave foránea referenciando a habitaciones.idhab)

idrese: int(11) (clave foránea referenciando a reservar.idrese)

iddn: int(11) (clave foránea referenciando a clientes.iddn)

nomc: varchar(35)

numiha: char(14)

detaha: text

precha: decimal(10,2)

fere: timestamp

usuarios:

id: int(11) (clave primaria)

nombre: varchar(25)

correo: varchar(35)

usuario: varchar(25)

contra: varchar(255)

rol: char(1)

fere: timestamp

Notas:

Las claves primarias están indicadas como clave primaria.

Las claves foráneas están indicadas como clave foránea referenciando a tabla.columna.

Se han incluido los tipos de datos y las restricciones de las columnas en la medida de lo posible.

La relación entre las tablas se establece a través de las claves foráneas.

Los índices y restricciones también se han incluido en el diccionario.