

# Descripción de la Arquitectura de Software

# Table of Contents

1. Contexto y Alcance del Sistema.....	7
1.1. Visión del Sistema.....	7
1.2. Alcance .....	7
1.3. Audiencia .....	9
1.4. Glosario .....	9
2. Definición de requerimientos.....	13
2.1. Planteamiento del Problema .....	13
2.2. Alcance y propósito del diseño .....	13
2.3. Diagrama de casos de uso.....	16
2.4. Descripciones de casos de uso.....	20
3. Impulsores Arquitectónicos (ASRs) y Priorización .....	56
3.1. Utility Tree .....	56
3.2. Escenarios de atributos de calidad.....	58
3.3. Restricciones y preocupaciones .....	78
4. Método de diseño .....	80
4.1. Método de diseño .....	80
5. Arquitectura .....	84
5.1. Vista lógica .....	84
5.2. Modelo de conceptos de negocio.....	87
5.3. Vista de Proceso .....	88
5.4. Vista de Implementación .....	93
6. Registro de decisiones arquitectónicas.....	126
6.1. Record architecture decisions.....	126
7. Aspectos Transversales (Cross-Cutting Concerns) .....	127
7.1. Seguridad y Auditoría (Trazabilidad) .....	127
7.2. Gestión de Transacciones (Atomicidad) .....	127
7.3. Manejo de Excepciones y Observabilidad .....	128
8. Trazabilidad .....	129
9. Apéndice.....	132
9.1. Vista de Contexto .....	132

**UNIVERSIDAD VERACRUZANA - REGIÓN XALAPA FACULTAD DE ESTADÍSTICA E INFORMÁTICA**

*Licenciatura en Ingeniería de Software*

*Experiencia Educativa Diseño de Software (Periodo Agosto 25 - Enero 26)*

*Académico: Jorge Octavio Ocharan Hernández*

**Sistema Hotelero**

**PRESENTAN:**

Barradas Sánchez Genaro Alejandro | | [zs23014083@estudiantes.uv.mx](mailto:zs23014083@estudiantes.uv.mx)

Bello Peralta Lizeth Guadalupe | | [zs23014074@estudiantes.uv.mx](mailto:zs23014074@estudiantes.uv.mx)

Morales García Omar | | [zs23014039@estudiantes.uv.mx](mailto:zs23014039@estudiantes.uv.mx)

**REPOSITORIO: [GitHub](#)**

## Control de versiones

Versión	Fecha	Autor(es)	Resumen de cambios significativos
0.0.1	7 de septiembre de 2025	Genaro Alejandro Barradas Sánchez	Creación de documento inicial con apartados
0.0.1.0	7 de septiembre de 2025	Genaro Alejandro Barradas Sánchez  Lizeth Guadalupe Bello Peralta	Se añadieron los borradores de los casos de uso
0.0.1.1	8 de septiembre de 2025	Genaro Alejandro Barradas Sánchez  Lizeth Guadalupe Bello Peralta	Se realizó una segunda iteración de los casos de uso
0.0.1.2	9 de septiembre de 2025	Omar Morales García	Se añadió el primer borrador del diagrama de clases
0.0.1.3	11 de septiembre de 2025	Genaro Alejandro Barradas Sánchez  Lizeth Guadalupe Bello Peralta	Se realizaron los cambios en base a las observaciones del profesor al listado de casos de uso
0.0.1.4	14 de septiembre de 2025	Lizeth Guadalupe Bello Peralta	Se añadieron las restricciones del proyecto, actores dentro del sistema, alcance del proyecto, visión del sistema y planteamiento del problema
0.0.1.5	17 de septiembre de 2025	Genaro Alejandro Barradas Sánchez  Omar Morales García  Lizeth Guadalupe Bello Peralta	Se identificaron y plasmaron los QAs del sistema con sus respectivos escenarios y trazabilidad

<b>Versión</b>	<b>Fecha</b>	<b>Autor(es)</b>	<b>Resumen de cambios significativos</b>
0.0.2.0	14 de septiembre de 2025	Genaro Alejandro Barradas Sánchez  Lizeth Guadalupe Bello Peralta  Omar Morales García	Se realizó e integró el diagrama Utility tree
0.0.2.1	21 de septiembre de 2025	Omar Morales García	Se creó y anexó el diagrama de contexto de nivel 0 del sistema
0.0.2.3	21 de septiembre de 2025	Omar Morales García	Se añadió el segundo borrador del diagrama de clases del sistema
0.0.3.1	26 de septiembre de 2025	Lizeth Guadalupe Bello Peralta	Se añadió el diccionario de datos correspondiente al sistema
0.0.3.2	27 de septiembre de 2025	Genaro Alejandro Barradas Sánchez	Se creó el glosario y añadieron conceptos al mismo
0.0.3.3	30 de septiembre de 2025	Lizeth Guadalupe Bello Peralta	Se añadió el Planteamiento del Problema
0.0.3.4	30 de septiembre de 2025	Genaro Alejandro Barradas Sánchez	Se adjuntó la tabla de versiones al documento
0.0.4	30 de septiembre de 2025	Omar Morales García	Se adjuntan imágenes de diagramas de casos de uso, por actores y paquetes
0.0.4.1	30 de septiembre de 2025	Lizeth Guadalupe Bello Peralta	Se añaden dos descripciones de casos de uso del actor Administrador
0.0.4.2	01 de octubre de 2025	Omar Morales García	Se añaden tres descripciones de casos de uso del actor Huésped
0.0.4.3	02 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se añaden tres descripciones de casos de uso del actor Recepción
0.0.4.4	02 de octubre de 2025	Omar Morales García	Se añaden tres descripciones de casos de uso del actor Huésped

Versión	Fecha	Autor(es)	Resumen de cambios significativos
0.0.4.5	03 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se añaden cuatro descripciones de casos de uso del actor Recepción
0.0.5	03 de octubre de 2025	Omar Morales García	Se añadió la visión y alcance del sistema
0.0.6	04 de octubre de 2025	Omar Morales García	Se añadió el diagrama DFD de nivel 1.
0.0.7	04 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se añadió la audiencia del documento
0.1.0	04 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se añadió el alcance y el propósito del documento
0.1.1	06 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se añadió estructura principal del proyecto para el hito 2
0.1.1.1	09 de octubre de 2025	Omar Morales García	Se añadieron preocupaciones y restricciones
0.1.1.2	09 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Arreglo de errores de texto y espaciado
0.1.2	09 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se agregó la vista lógica, específicamente el diagrama de clases y su debida justificación
0.1.3	10 de octubre de 2025	Omar Morales García	Se añadieron diagramas de componentes y de objetos
0.1.4	10 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se agregó el método de diseño del proyecto
0.1.5	15 de octubre de 2025	Lizeth Guadalupe Bello Peralta	Se añadió el diagrama de conceptos del negocio
0.1.6	18 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se corrigen errores menores en nombrado de imágenes y texto
0.1.7	23 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se añadió apartado para ADR
0.1.8	30 de octubre de 2025	Omar Morales García	Se agregaron diagramas de componentes
0.1.8.1	30 de octubre de 2025	Omar Morales García	Correcciones a diagramas de componentes

Versión	Fecha	Autor(es)	Resumen de cambios significativos
0.1.9	30 de octubre de 2025	Omar Morales García	Se agregaron diagramas de comunicación
0.1.9.1	30 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se agregaron diagramas de comunicación
0.1.9.2	30 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se arreglaron diagramas
0.2.0	31 de octubre de 2025	Omar Morales García	Se realizaron cambios en el nombrado de diagramas
0.2.0.1	31 de octubre de 2025	Omar Morales García	Se agregaron diagramas de comunicación
0.2.0.2	31 de octubre de 2025	Omar Morales García	Se realizaron cambios dentro de diagramas de comunicación
0.2.0.3	31 de octubre de 2025	Omar Morales García	Se realizaron cambios dentro de diagramas de componentes
0.2.0.4	31 de octubre de 2025	Omar Morales García	Se agregaron diagramas de comunicación
0.2.0.5	31 de octubre de 2025	Genaro Alejandro Barradas Sánchez	Se agregaron diagramas de comunicación
0.2.0.6	31 de octubre de 2025	Omar Morales García	Se agregaron diagramas de comunicación
0.2.0.7	1 de noviembre de 2025	Genaro Alejandro Barradas Sánchez	Se agregaron diagramas de comunicación
0.2.0.8	1 de noviembre de 2025	Lizeth Guadalupe Bello Peralta	Se agregaron diagramas de comunicación
0.2.0.10	2 de noviembre de 2025	Lizeth Guadalupe Bello Peralta	Se agregaron diagramas de comunicación
0.2.0.11	2 de noviembre de 2025	Lizeth Guadalupe Bello Peralta	Se agregaron diagramas de comunicación
0.2.0.12	2 de noviembre de 2025	Genaro Alejandro Barradas Sánchez	Vinculación de diagramas de comunicación al texto
0.2.0.13	2 de noviembre de 2025	Omar Morales García	Vinculación de diagramas de comunicación al texto
0.2.1	4 de noviembre de 2025	Omar Morales García	Se agregaron diagramas de interfaz
0.2.1.1	5 de noviembre de 2025	Omar Morales García	Se agregaron diagramas de interfaz

<b>Versión</b>	<b>Fecha</b>	<b>Autor(es)</b>	<b>Resumen de cambios significativos</b>
0.2.1.2	6 de noviembre de 2025	Omar Morales García	Se agregaron diagramas de interfaz
0.2.1.3	6 de noviembre de 2025	Omar Morales García	Cambio dentro del nombrado de diagramas
0.2.4.0	20 de noviembre de 2025	Omar Morales García	Se añadió un diagrama de despliegue
0.2.5.0	20 de noviembre de 2025	Omar Morales García	Se añadió la descripción y justificación del diagrama de despliegue
0.3.0.0	30 de noviembre de 2025	Genaro Alejandro Barradas Sánchez	Se añadió diagramas de secuencia
0.3.1.0	30 de noviembre de 2025	Omar Morales García	Se añadió la descripción del diagrama de secuencia

# Chapter 1. Contexto y Alcance del Sistema

## 1.1. Visión del Sistema

La visión del Sistema de Reservaciones es establecer la plataforma como líder tecnológica, convirtiendo las reservaciones y sistema en una ventaja competitiva a nivel global.

Actualmente, la dependencia en soluciones locales ineficientes y dispersas genera inconsistencias críticas, errores de inventario y un alto riesgo de incidentes (como dobles reservas y dobles cobros). El sistema de reservaciones busca erradicar estos problemas mediante una arquitectura escalable y robusta diseñada para soportar a más de 5,000 hoteles y un millón de habitaciones.

Nuestra prioridad es garantizar la confianza y consistencia en el momento de realizar reservas: el sistema debe operar con una tasa de incidentes de dobles cobros o dobles reservas de cero, incluso bajo escenarios de alta concurrencia. Esto se logrará a través de transacciones atómicas y un riguroso enfoque en la seguridad y trazabilidad.

Siendo así que, permitirá la implementación de precios dinámicos y políticas avanzadas de overbooking gestionadas centralmente para optimizar los ingresos. Finalmente, la eficiencia operacional es clave; el sistema ofrecerá una interfaz intuitiva y altamente disponible que permitirá al 90% del personal novato de recepción completar tareas críticas (check-in/out) de manera autónoma en minutos, garantizando una operación fluida y un servicio al cliente superior.

El Sistema de Reservaciones es el pilar para un crecimiento rentable, la estandarización de procesos y la excelencia operativa en toda la cadena hotelera.

## 1.2. Alcance

El Alcance del Sistema de Reservaciones define las fronteras del diseño arquitectónico, especificando qué se incluye dentro de la plataforma y qué se excluye, delegándose a sistemas externos o a la infraestructura. El sistema está diseñado para reemplazar las soluciones locales y gestionar el ciclo de vida completo de la reserva y la habitación a través de una plataforma unificada.

### 1.2.1. Funcionalidades Incluidas (In-Scope)

El Sistema de Reservaciones abarcará los siguientes módulos funcionales principales:

1. Módulo de reservaciones y venta (Huésped) Este módulo gestiona la interacción directa con el cliente y el proceso de venta:
  - Búsqueda y consulta de hoteles: Permitir al huésped buscar hoteles por región/fecha y consultar la disponibilidad de tipos de habitación en tiempo real.
  - Gestión de la Reserva: Incluye la creación de la reserva con control de atomicidad, la modificación de fechas/habitaciones y la cancelación de la reserva, aplicando las políticas de cobro correspondientes.
  - Consulta de Historial: Permitir al huésped revisar sus reservas pasadas y activas.

- Integración de pagos seguros: Captura segura y tokenización de los datos de pago, garantizando la idempotencia para evitar dobles cobros.
2. Módulo de recepción y operaciones (Recepcionista) Este módulo está enfocado en la eficiencia operativa y el manejo diario de huéspedes:
- Check-in / Check-out: Funciones optimizadas para el personal de recepción, con actualización inmediata del estado de la habitación. El diseño debe priorizar la usabilidad para alcanzar el objetivo de que un novato complete la tarea después de 10 minutos de instrucción.
  - Gestión de cuentas y cobros: Consolidación de todos los cargos de la estancia (tarifa base, impuestos, consumos adicionales) y procesamiento del cobro final con alta seguridad.
  - Servicios adicionales: Registro de cargos a la cuenta por consumos en el bar, lavandería, gimnasio u otros servicios.
  - Gestión de habitaciones: Funcionalidades para cambiar a un huésped de habitación de forma atómica y para marcar habitaciones como "Fuera de Servicio".
3. Módulo de administración y configuración (Administrador) Este módulo se centra en las herramientas gerenciales para optimizar ingresos y capacidad:
- Gestión de hoteles y tipos de habitación: Alta y modificación de hoteles, así como la definición de la categorización de habitaciones, amenidades y capacidades.
  - Configuración avanzada de precios: Definición de tarifas base, precios dinámicos, promociones, descuentos y políticas de temporadas altas.
  - Configuración de políticas de overbooking: Ajuste de los umbrales de sobreventa de inventario, lo que es crítico para la maximización de la capacidad de venta.
  - Monitoreo de inventario: Consulta en tiempo real de la ocupación y disponibilidad en toda la cadena.
4. Módulo de seguridad y auditoría (Auditor) Este módulo es esencial para los requisitos de seguridad y confiabilidad del sistema:
- Generación de reportes de trazabilidad: Consultas avanzadas sobre actividades sospechosas (cancelaciones rápidas, cambios de precio o descuentos) que requieren un log de auditoría inmutable.
  - Reconciliación Financiera: Reportes para conciliar ingresos por servicios adicionales y validar que todos los cobros en línea se hayan procesado y registrado correctamente.
  - Validación de Políticas: Herramientas para asegurar la uniformidad de precios y políticas entre diferentes hoteles o regiones.
5. Módulo de acceso al sistema (Usuario en general) Este módulo se encargará de la gestión de usuarios y seguridad:
- Conceder acceso al sistema: Se encargará de dar acceso a determinadas funcionalidades del sistema a usuarios válidos.
  - Seguridad: No permitirá el acceso a usuarios que no se encuentren registrados dentro del sistema.
  - Generación de reportes: Obtendrá datos sobre los inicios de sesión realizados dentro del sistema, así como las acciones que han realizado.

## 1.2.2. Funcionalidades Excluidas (Out-of-Scope)

Las siguientes áreas funcionales no forman parte del alcance directo del Sistema de Reservas y se consideran responsabilidades de sistemas o servicios externos:

- Procesamiento de Pagos: El Sistema de Reservas no almacenará ni procesará directamente información de tarjetas de crédito. Esta responsabilidad se delega por completo a una pasarela de pagos (Payment Gateway).
- Gestión de Sistemas de Fidelización: El manejo de programas de puntos de lealtad, niveles de membresía o recompensas para huéspedes. El SRC solo notificará a un sistema externo el check-out para que este acredeite los puntos.
- Servicios de Limpieza y Mantenimiento: La gestión de la logística interna de las camareras o técnicos. El SRC solo actualizará el estado de la habitación a "Limpieza Necesaria" (L/N), pero no gestionará la asignación de tareas.
- Marketing y CRM: Campañas de correo electrónico, gestión de relaciones con el cliente más allá del historial transaccional de reservas.

## 1.3. Audiencia

Este documento está dirigido a todas las partes interesadas involucradas en el diseño, desarrollo, implementación y mantenimiento del Sistema Hotelero. En particular, está destinado a:

- Equipo de desarrollo: Ingenieros de software, programadores y diseñadores responsables de implementar la arquitectura y los componentes descritos.
- Analistas: Encargados de validar que los requisitos funcionales y no funcionales se reflejen correctamente en el diseño propuesto.
- Equipo de pruebas: Para comprender la estructura del sistema, los módulos clave y las interacciones entre componentes al momento de diseñar y ejecutar pruebas.
- Arquitectos de software: Para verificar la coherencia del diseño con las decisiones arquitectónicas tomadas y asegurar la calidad del modelo final.
- Académico: Responsable de evaluar la correcta aplicación de principios de diseño, patrones y tácticas en el proyecto.
- Usuarios internos de recepción, administración y gestión hotelera: Aunque no se espera que lean el documento completo, podrían consultarlos parcialmente para comprender cómo el sistema satisface sus necesidades funcionales.

## 1.4. Glosario

Concepto	Descripción / significado	Estándar
	A	

Concepto	Descripción / significado	Estándar
<b>Acceso al sistema</b>	Proceso mediante el cual un usuario inicia sesión en el sistema utilizando sus credenciales (usuario y contraseña) para realizar tareas según su rol. Incluye controles sobre gestión de accesos (autenticación, privilegios mínimos, gestión de identidades, control de contraseñas, sesiones).	ISO/IEC 27001 Seguridad de la Información
<b>Atomicidad</b>	Propiedad de una transacción (especialmente la Reserva) que garantiza que esta se ejecuta completamente o no se ejecuta en absoluto. Es esencial para el requisito de "Confianza y consistencia en la reserva".	Principio ACID / Arquitectura Distribuida
<b>ASR (Architecturally Significant Requirement)</b>	Requisito que tiene un impacto profundo en la estructura del código y es de alto valor para el negocio. Generalmente capturados como Escenarios de Calidad (QAS).	Marco ADD
B		
<b>Baja Latencia</b>	Requisito de rendimiento que exige que las operaciones críticas (consulta de Disponibilidad, confirmación de Pago) se completen en un tiempo muy corto, medido en milisegundos (ej. < 500 ms en QAS-03).	Atributo de Calidad (Rendimiento)
C		
<b>Concurrencia</b>	Capacidad del sistema para manejar múltiples solicitudes de reserva o consulta simultáneamente (por varios Huéspedes y Recepcionistas) sin comprometer la integridad de los datos de inventario.	Atributo de Calidad (Rendimiento)
<b>Confianza y Consistencia</b>	Objetivo de Negocio primordial que se traduce en el requisito de tasa de incidentes de dobles cobros o dobles reservas de cero.	Objetivo de Negocio
D		
<b>Desacoplamiento</b>	Principio de diseño que busca reducir la dependencia entre los módulos del sistema (Cliente-Servidor) y entre el cliente y los sistemas del servidor, facilitando la Mantenibilidad.	Principio de Diseño
<b>Disponibilidad</b>	Atributo de Calidad que mide la proporción de tiempo que el sistema o una función crítica (ej. consulta de disponibilidad) está operativa y accesible.	ISO/IEC 25010
E		

Concepto	Descripción / significado	Estándar
<b>Escalabilidad</b>	Capacidad del SRC para manejar un aumento significativo en la carga de trabajo (más de 5,000 hoteles y 1,000,000 de habitaciones) sin degradar el Rendimiento o la Latencia.	Atributo de Calidad
I		
<b>Idempotencia</b>	Táctica de seguridad y diseño que garantiza que una operación de cobro, si se repite accidentalmente, solo se procesará una vez. Es clave para evitar dobles cobros.	Principio de Diseño / Seguridad
<b>Inventario</b>	La colección de recursos vendibles del hotel, principalmente las Habitaciones, junto con sus estados (Disponible, Bloqueada, Reservada).	Entidad de Negocio
L		
<b>Log Inmutable</b>	Almacén de datos de solo escritura que registra todos los eventos críticos de seguridad y transacción para garantizar la trazabilidad y el no repudio.	Táctica de Seguridad
M		
<b>Mantenibilidad</b>	Atributo de Calidad que describe la facilidad con la que el sistema puede ser modificado, corregido y actualizado, un impulsor clave para la adopción de un sistema Cliente-Servidor.	ISO/IEC 25010
O		
<b>Observabilidad</b>	La capacidad del sistema para entender su estado interno a partir de datos externos, esencial para monitorear el Rendimiento y diagnosticar fallas en un sistema Cliente-Servidor.	Atributo de Calidad
<b>Overbooking</b>	Política estratégica de gestión de inventario que permite la sobreventa controlada de Habitaciones, basándose en la probabilidad histórica de cancelaciones para maximizar los ingresos.	Regla de Negocio
P		
<b>Pasarela de Pagos</b>	Sistema externo con el que el SRC se comunica para procesar y autorizar transacciones financieras (Cobro y reembolso). Se encuentra out-of-scope.	Entidad Externa

<b>Concepto</b>	<b>Descripción / significado</b>	<b>Estándar</b>
<b>PCI-DSS</b>	Estándar de seguridad obligatorio para cualquier entidad que procese, almacene o transmita datos de tarjetas de crédito. Cumplimiento delegado a la Pasarela de Pagos.	Estándar de Seguridad
	Q	
<b>QAS (Escenario de Calidad)</b>	Descripción formal de una prueba de diseño que evalúa un Atributo de Calidad específico bajo ciertas condiciones (ej. QAS-03 evalúa la Disponibilidad bajo Falla de DB).	Herramienta de Arquitectura
	R	
<b>Rendimiento</b>	Atributo de Calidad que mide la velocidad y eficiencia del sistema bajo carga, incluyendo métricas como latencia y capacidad de respuesta.	ISO/IEC 25010
<b>Reserva</b>	La entidad central de negocio que representa el acuerdo contractual entre el Hotel y el Huésped, su ciclo de vida es el corazón del SRC.	Entidad de Negocio
	S	
<b>SR (Sistema de Reservaciones)</b>	Objeto principal de diseño de este proyecto. Plataforma unificada y robusta que reemplaza las soluciones locales de los hoteles.	Contexto del Sistema
	T	
<b>Trazabilidad</b>	Requisito de Seguridad y Auditoría que exige que cada evento crítico tenga un rastro documentado, inmutable y verificable.	Atributo de Calidad (Seguridad)
<b>Táctica</b>	Una decisión de diseño específica y concreta utilizada para controlar o asegurar un atributo de calidad, sirviendo como bloque constructivo para un Patrón Arquitectónico (ej. "Usar caché" es una Táctica para Rendimiento).	Concepto de Arquitectura
	U	
<b>Usabilidad</b>	Atributo de Calidad clave para el módulo de recepción que mide la facilidad de uso del sistema, con el objetivo de reducir el tiempo de entrenamiento de los recepcionistas novatos.	ISO/IEC 25010

# **Chapter 2. Definición de requerimientos**

## **2.1. Planteamiento del Problema**

La industria hotelera es una de las más competidas a nivel mundial, lo que exige mantenerse a la vanguardia y dar servicios superiores para sobresalir. La cadena hotelera, posicionada como un líder global en la hospitalidad, opera a una escala sin precedentes que necesita máxima eficiencia operativa. Por lo tanto, para mantener su liderazgo estratégico, es imperativo gestionar un vasto inventario de 1,000,000 de habitaciones de forma unificada. Esto requiere uniformidad en el proceso para gestionar las más de 5000 propiedades a nivel mundial. A pesar de esta magnitud, la realidad operativa actual revela una vulnerabilidad crítica: cada hotel utiliza una solución local de reservas, lo que genera inconsistencias entre hoteles, errores y dificultades en el proceso de centralización.

Este problema de fragmentación se traduce en que se crean múltiples fuentes de información, lo que genera diversos reportes o ingresos que pueden diferir entre el hotel y el corporativo. Esto hace necesaria una conciliación manual que consume tiempo y retrasa la toma de decisiones estratégicas. Además, los empleados de central o gerentes de hotel también pierden tiempo al tener que ingresar datos, exportar o manipular información de diferentes sistemas, elevando la fricción operativa de la cadena.

Sumado a esta ineficiencia, el riesgo tecnológico es significativo. La arquitectura basada en soluciones locales individuales está desactualizada. Mantener estos sistemas requiere soporte especializado y caro, lo cual ya no es escalable ni sostenible. En consecuencia, la infraestructura actual, al tener 5000 puntos de entrada en lugar de uno centralizado, multiplica las vulnerabilidades, es decir, el riesgo de ciberataques o fallas en el cumplimiento de normativas de privacidad de datos, comprometiendo la fiabilidad de la operación.

Por consiguiente, la justificación de este proyecto radica en la necesidad de transformar la gestión de reservaciones de un costo operativo a una ventaja competitiva. La implementación de una plataforma centralizada y robusta es la única vía para optimizar las operaciones diarias a esta escala. Un sistema unificado permitirá llevar a cabo procesos más eficientes, asegurando un incremento estructural de los ingresos y mejorando la satisfacción del cliente. Al estandarizar la arquitectura tecnológica, el nuevo sistema permitirá una centralización de datos, eliminando la conciliación manual y, por consiguiente, recortando tiempos y mitigando riesgos de seguridad.

Además, esta transformación desbloqueará la capacidad de implementar precios dinámicos y políticas de ocupación avanzadas a nivel de cadena, garantizando la máxima monetización del inventario y la escalabilidad futura.

## **2.2. Alcance y propósito del diseño**

El presente apartado tiene como finalidad formalizar las fronteras del diseño arquitectónico del Sistema Hotelero, estableciendo los límites conceptuales, técnicos y organizacionales dentro de los cuales se desarrollará la solución de software. Su propósito es definir con precisión qué aspectos del sistema serán abordados por el diseño, quiénes participan en el proceso arquitectónico y cuáles son sus preocupaciones principales, las cuales funcionarán como base para la definición de los

impulsores arquitectónicos (Architectural Significant Requirements, ASRs) que guiarán las decisiones técnicas posteriores.

El diseño arquitectónico traduce los requerimientos funcionales y no funcionales en una estructura coherente, organizada y justificable, donde cada decisión responde a una necesidad o atributo de calidad identificado. Por ello, este apartado delimita lo que el sistema debe hacer (qué casos de uso y funcionalidades cubre) frente a cómo será diseñado para hacerlo posible (qué estilos, tácticas, componentes y patrones implementará).

### **2.2.1. Propósito del diseño arquitectónico**

El propósito del diseño arquitectónico es proporcionar una representación estructurada del sistema que permita entender su organización interna, interacción de componentes, flujo de información y comportamiento en ejecución, asegurando que cumpla con los objetivos del negocio y los atributos de calidad priorizados: usabilidad, disponibilidad, seguridad y rendimiento.

Este diseño actúa como un lenguaje común entre los distintos interesados del proyecto (analistas, desarrolladores, evaluadores y usuarios), estableciendo una visión compartida de cómo la solución técnica materializa los requisitos del sistema.

En particular, este documento busca:

- Establecer las fronteras técnicas del diseño, indicando qué capas, módulos, clases e interfaces son parte de esta versión.
- Identificar los stakeholders que influyen en las decisiones arquitectónicas.
- Documentar sus preocupaciones (concerns), que servirán como base para definir los impulsores arquitectónicos.
- Garantizar la trazabilidad entre las necesidades de los interesados y las decisiones técnicas que las atienden.
- Proporcionar un marco de referencia para el método de diseño ADD (Attribute Driven Design).

### **2.2.2. Fronteras del diseño**

Dentro de las fronteras del diseño se incluyen:

- Las capas internas del sistema: presentación, lógica de negocio y persistencia de datos.
- Los casos de uso de huésped, recepción, administración y auditoría, que representan la funcionalidad principal del sistema.
- Los modelos de datos relacionados con habitaciones, reservas, clientes, servicios y transacciones.
- Las decisiones arquitectónicas fundamentales (estilos, patrones, tácticas y restricciones).

Fuera de las fronteras del diseño se consideran:

- Los sistemas externos no implementados directamente (pasarelas reales de pago, contabilidad o facturación).

- Los mecanismos avanzados de auditoría o analítica.

### 2.2.3. Stakeholders (interesados)

El diseño arquitectónico del Sistema Hotelero debe responder a las necesidades, expectativas y responsabilidades de los distintos interesados (stakeholders) involucrados en su desarrollo, operación y evaluación. Cada uno de ellos influye directa o indirectamente en las decisiones arquitectónicas al aportar requisitos funcionales o atributos de calidad que condicionan el diseño.

Stakeholder	Rol / Interés	Responsabilidad dentro del diseño
Huésped	Usuario final del servicio hotelero. Realiza reservas, consultas y cancelaciones a través del recepcionista o de interfaces de atención.	Indirectamente define las necesidades de usabilidad, confiabilidad y seguridad de los datos personales y pagos. Sus expectativas de rapidez y claridad influyen en el diseño de la interfaz y los tiempos de respuesta.
Recepcionista	Usuario operativo del sistema. Gestiona el check-in, check-out, reservas y cobros.	Su trabajo diario orienta los requerimientos de usabilidad, eficiencia operativa y disponibilidad continua. Las decisiones de interfaz, validación y retroalimentación se fundamentan en sus flujos de trabajo.
Administrador del hotel	Supervisa la operación general, controla la disponibilidad de habitaciones, tarifas y reportes financieros.	Su enfoque está en la confiabilidad, precisión e integridad de la información, así como en la seguridad de las transacciones y la trazabilidad de los procesos de cobro y auditoría.
Auditor	Encargado de verificar la veracidad de la información contable y operativa registrada en el sistema.	Requiere trazabilidad, seguridad y consistencia de datos. Sus expectativas impulsan tácticas de auditoría, registro de logs y control de acceso a la información.
Equipo de desarrollo	Conforma a los estudiantes encargados de construir el sistema conforme al diseño arquitectónico.	Implementa los componentes, interfaces y módulos definidos en el diseño, asegurando que cumplan con los atributos de calidad establecidos (usabilidad, seguridad, rendimiento, disponibilidad).
Analista de sistemas	Define los casos de uso y valida que el diseño cubra los requerimientos funcionales.	Garantiza la coherencia entre los requerimientos capturados y la estructura del diseño, facilitando la trazabilidad entre necesidades y soluciones técnicas.

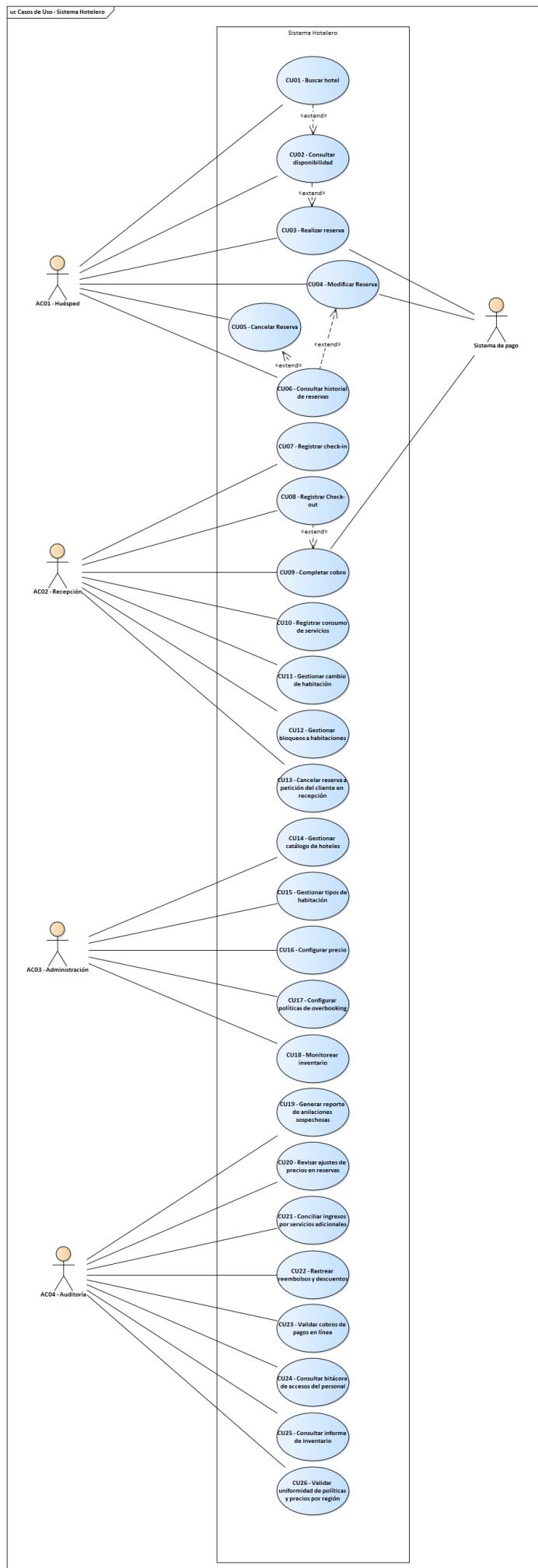
Stakeholder	Rol / Interés	Responsabilidad dentro del diseño
Académico asesor	Evaluador académico y supervisor técnico del proyecto.	Verifica que la arquitectura cumpla criterios de corrección técnica, consistencia interna y justificación de decisiones conforme al método ADD.

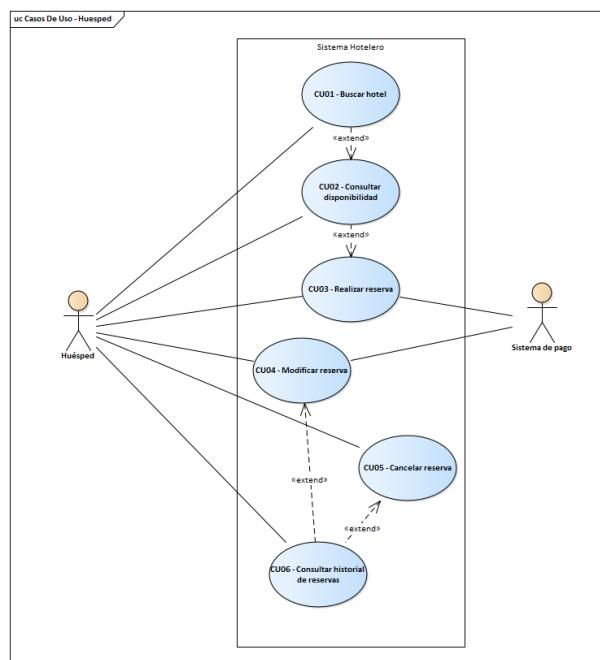
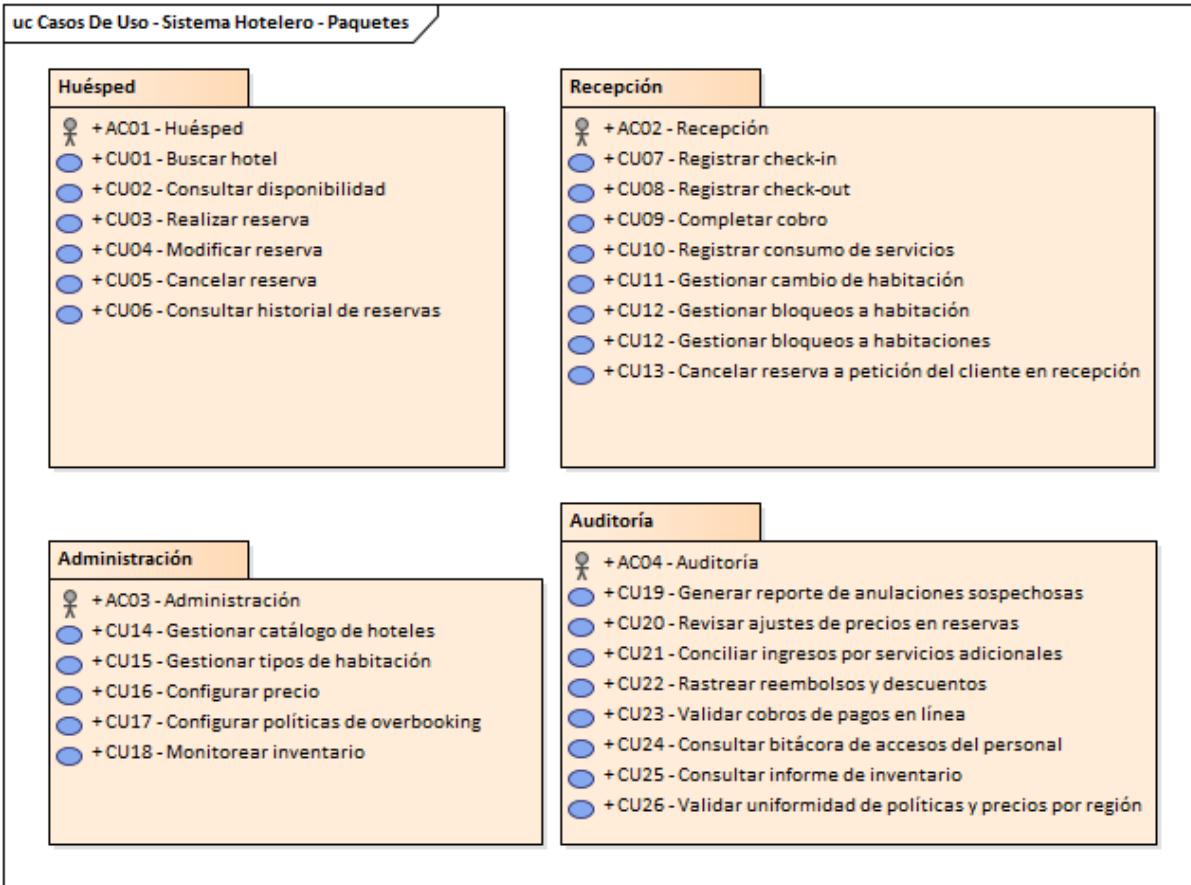
## 2.2.4. Función del diseño como base para impulsores arquitectónicos

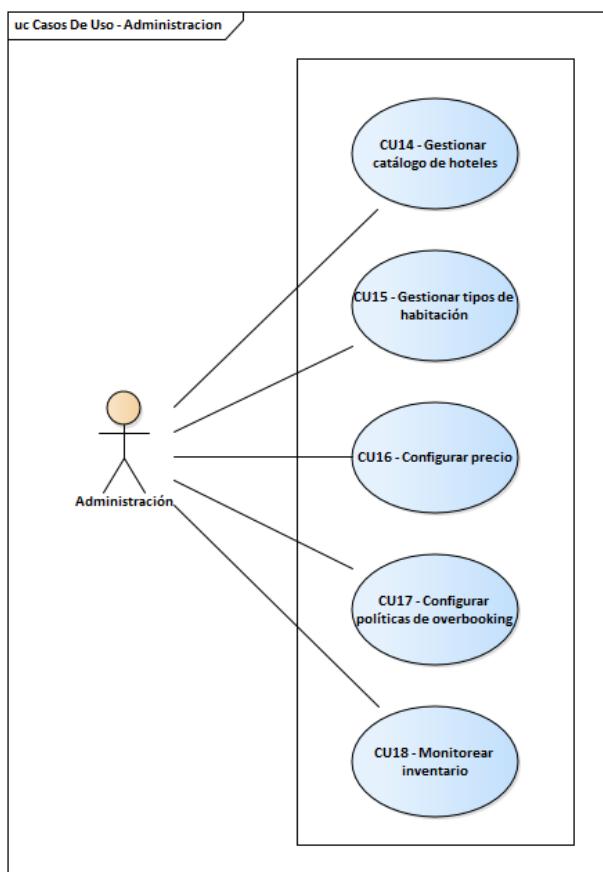
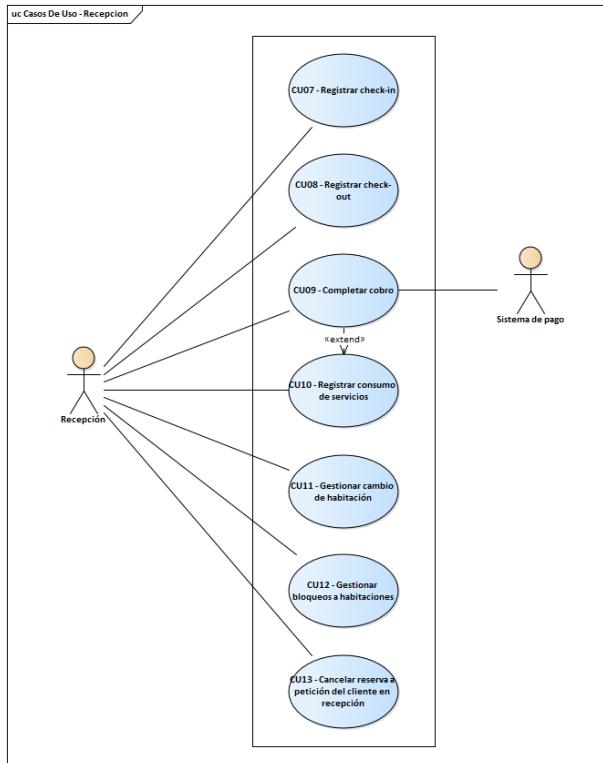
Cada uno de los concerns anteriores servirá como entrada directa para definir los impulsores arquitectónicos (ASRs) que guiarán las decisiones de diseño. Por ejemplo:

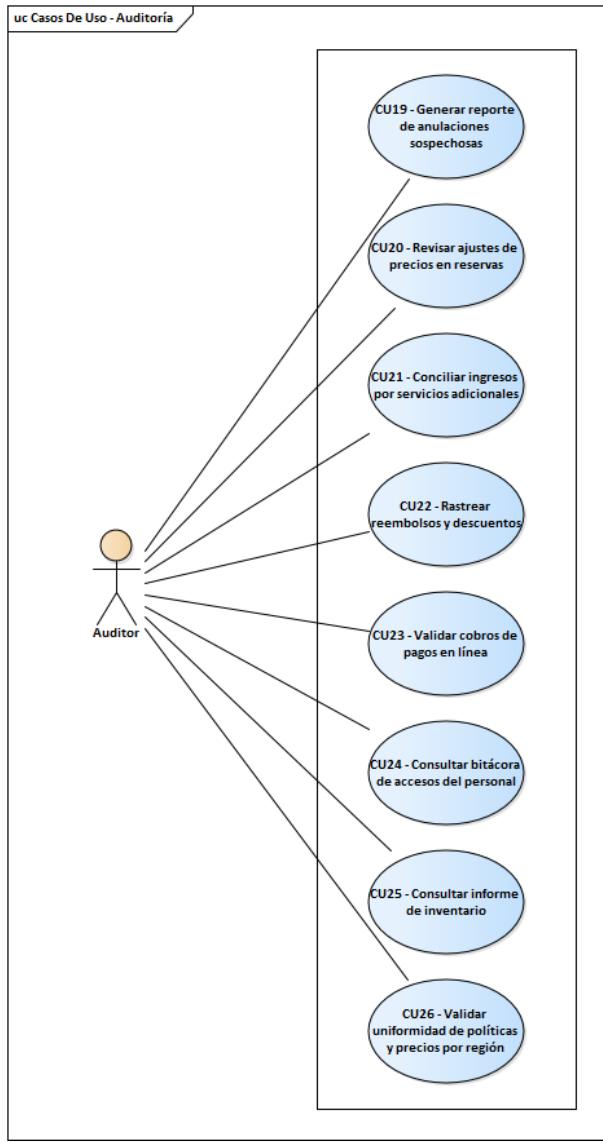
- Los concerns de usabilidad impulsarán decisiones sobre diseño de interfaz, validación y retroalimentación visual.
- Los concerns de disponibilidad determinarán tácticas de replicación, rollback y manejo de errores.
- Los concerns de seguridad orientarán la aplicación de mecanismos de autenticación y auditoría.
- Los concerns de rendimiento guiarán la selección de patrones de concurrencia, caching y optimización de consultas.

## 2.3. Diagrama de casos de uso









## 2.4. Descripciones de casos de uso

### 2.4.1. AC01 - Huésped

**CU01 - Buscar hotel**

<b>Nombre</b>	<b>CU01 - Buscar hotel</b>
Actor(es)	Huésped (AC01)
Objetivo	El HUÉSPED busca hoteles disponibles que cumplan con sus criterios de búsqueda (destino, fechas).
Precondiciones	PRE-1. Existen HOTELES registrados y accesibles dentro del sistema.

Nombre	<b>CU01 - Buscar hotel</b>
Flujo normal	<p>1. El sistema muestra una lista de hoteles disponibles con barra de búsqueda.</p> <p>2. El HUÉSPED selecciona un hotel y da clic en “Aceptar”.</p> <p>3. El sistema muestra información del hotel con filtros.</p> <p>4. El HUÉSPED visualiza la información y da clic en “Cancelar”.</p> <p>5. El sistema cierra la ventana.</p> <p>6. Termina el caso de uso.</p>
Flujo alterno	<p><b>FA 2.1:</b> El HUÉSPED da clic en “Salir”. El sistema confirma y finaliza.</p> <p><b>FA 4.1:</b> El HUÉSPED consulta disponibilidad → continúa en CU02.</p>
Excepción	<b>EX1:</b> No hay hoteles registrados. El sistema muestra error y finaliza.
Postcondiciones	POST-1. El HUÉSPED visualiza la lista de hoteles disponibles.

## **CU02 - Consultar disponibilidad**

Nombre	<b>CU02 - Consultar disponibilidad</b>
Actor(es)	Huésped (AC01)
Objetivo	El HUÉSPED consulta la disponibilidad de una habitación de un hotel.
Precondiciones	PRE-1. Existen hoteles con habitaciones registradas y tarifas actualizadas.
Flujo normal	<p>1. El sistema muestra información del hotel con filtros.</p> <p>2. El HUÉSPED selecciona una habitación.</p> <p>3. El sistema muestra información de la habitación.</p> <p>4. El HUÉSPED da clic en “Reservar”.</p> <p>5. El sistema muestra disponibilidad de la habitación.</p> <p>6. El HUÉSPED da clic en “Cancelar”.</p> <p>7. Termina el caso de uso.</p>
Flujo alterno	<p><b>FA 2.1:</b> El HUÉSPED da clic en “Salir”. El sistema confirma y finaliza.</p> <p><b>FA 6.1:</b> El HUÉSPED decide reservar → continúa en CU03.</p>
Excepción	<b>EX1:</b> No hay habitaciones registradas. El sistema muestra error y finaliza.
Postcondiciones	POST-1. El HUÉSPED consulta la disponibilidad de una habitación.

### **CU03 - Realizar reserva**

<b>Nombre</b>	<b>CU03 - Realizar reserva</b>
Actor(es)	Huésped (AC01)
Objetivo	El HUÉSPED realiza la reserva de una habitación.
Precondiciones	PRE-1. La habitación seleccionada está disponible y con tarifas actualizadas.
Flujo normal	<p>1. El sistema valida disponibilidad y muestra la información de la habitación.</p> <p>2. El HUÉSPED da clic en “Reservar”.</p> <p>3. El sistema muestra ventana con fechas y botón “Reservar”.</p> <p>4. El HUÉSPED selecciona fechas y confirma.</p> <p>5. El sistema valida y muestra información de pago.</p> <p>6. El HUÉSPED ingresa datos personales y confirma.</p> <p>7. El sistema procesa pago y muestra mensaje de confirmación.</p> <p>8. El HUÉSPED da clic en “Aceptar”.</p> <p>9. Termina el caso de uso.</p>
Flujo alterno	<p><b>FA 2.1:</b> El HUÉSPED cancela el proceso.</p> <p><b>FA 6.1:</b> El HUÉSPED paga en check-out.</p> <p><b>FA 7.1:</b> Pago rechazado.</p>
Excepción	<p><b>EX1:</b> Error de conexión con BD.</p> <p><b>EX2:</b> Error de conexión con pasarela de pago.</p>
Postcondiciones	POST-1. El HUÉSPED realiza la reserva y el sistema la registra.

### **CU04 - Modificar reserva**

<b>Nombre</b>	<b>CU04 - Modificar reserva</b>
Actor(es)	Huésped (AC01)
Objetivo	El HUÉSPED modifica una reserva activa.
Precondiciones	PRE-1. El sistema tiene reservas registradas.

Nombre	<b>CU04 - Modificar reserva</b>
Flujo normal	<p>1. El sistema recupera reservas activas.</p> <p>2. El HUÉSPED selecciona una reserva.</p> <p>3. El sistema muestra detalles con opción de modificar.</p> <p>4. El HUÉSPED selecciona “Modificar reserva”.</p> <p>5. El sistema muestra campos editables.</p> <p>6. El HUÉSPED edita y confirma.</p> <p>7. El sistema valida cambios y procesa pago si aplica.</p> <p>8. El HUÉSPED confirma.</p> <p>9. El sistema registra cambios y finaliza.</p>
Flujo alterno	<p><b>FA 2.1:</b> Cancelación del proceso.</p> <p><b>FA 4.1:</b> El flujo continúa en CU05 - Cancelar reserva.</p> <p><b>FA 9.1:</b> Pago rechazado.</p>
Excepción	<p><b>EX1:</b> Error de conexión con BD.</p> <p><b>EX2:</b> Error con el banco.</p>
Postcondiciones	POST-1. La reserva queda modificada en el sistema.

### **CU05 - Cancelar reserva**

Nombre	<b>CU05 - Cancelar reserva</b>
Actor(es)	Huésped (AC01)
Objetivo	El HUÉSPED cancela una reserva activa.
Precondiciones	PRE-1. El sistema tiene reservas registradas.

Nombre	<b>CU05 - Cancelar reserva</b>
Flujo normal	<p>1. El sistema recupera reservas activas.</p> <p>2. El HUÉSPED selecciona una reserva.</p> <p>3. El sistema muestra detalles con opción de cancelar.</p> <p>4. El HUÉSPED selecciona “Cancelar reserva”.</p> <p>5. El sistema solicita confirmación.</p> <p>6. El HUÉSPED confirma.</p> <p>7. El sistema cancela la reserva y muestra mensaje.</p> <p>8. El HUÉSPED confirma.</p> <p>9. Termina el caso de uso.</p>
Flujo alterno	<p><b>FA 2.1:</b> Cancelación del proceso.</p> <p><b>FA 6.1:</b> El HUÉSPED no confirma cancelación.</p>
Excepción	<p><b>EX1:</b> Error de conexión con BD.</p> <p><b>EX2:</b> Error en cancelación de la reserva.</p>
Postcondiciones	POST-1. La reserva queda cancelada en el sistema.

#### **CU06 - Consultar historial de reserva**

Nombre	<b>CU06 - Consultar historial de reservas</b>
Actor(es)	Huésped (AC01)
Objetivo	El HUÉSPED revisa todas las reservas pasadas, activas y canceladas asociadas a su cuenta.
Precondiciones	PRE-1. El sistema cuenta con reservas registradas por el HUÉSPED.

Nombre	<b>CU06 - Consultar historial de reservas</b>
Flujo normal	<p>1. El sistema recupera de la base de datos las reservas activas, concluidas y canceladas del HUÉSPED, mostrando los resultados con opciones de filtro (Activas, Pasadas, Canceladas) y el botón “Cancelar”.</p> <p>2. El HUÉSPED visualiza la lista de reservas y selecciona una para ver el detalle.</p> <p>3. El sistema muestra la información completa de la reserva seleccionada con los botones “Modificar reserva” (si está activa), “Cancelar reserva” (si está activa) y “Cancelar”.</p> <p>4. El HUÉSPED selecciona “Cancelar”.</p> <p>5. Termina el caso de uso.</p>
Flujo alterno	<p><b>FA 2.1:</b> El HUÉSPED da clic en “Salir”. El sistema muestra confirmación y finaliza.</p> <p><b>FA 4.1:</b> El flujo continúa en CU04 - Modificar reserva.</p> <p><b>FA 4.2:</b> El flujo continúa en CU05 - Cancelar reserva.</p>
Excepción	<b>EX1:</b> El sistema no puede conectarse con la base de datos. Muestra mensaje de error y finaliza.
Postcondiciones	POST-1. El HUÉSPED visualiza el historial de sus reservaciones.

## 2.4.2. AC02 - Recepción

### CU07 - Registrar check-in

Nombre	<b>CU07 - Registrar check-in</b>
Actor(es)	Recepción (AC02)
Objetivo	El recepcionista registra la entrada del huésped en el sistema.
Precondiciones	PRE-1. Existe una reserva activa a nombre del huésped.  PRE-2. El huésped se presenta en la recepción

Nombre	CU07 - Registrar check-in
Flujo normal	<p>1. El sistema muestra una ventana donde solicita el nombre del huésped o persona que hizo la reserva y el botón “Buscar”.</p> <p>2. El recepcionista ingresa los datos de búsqueda de la reserva y da clic en la opción “Buscar”.</p> <p>3. El sistema busca la reserva en la base de datos y despliega los detalles de la reserva (nombre del huésped, fecha de inicio y fin, habitación asignada) junto a los botones “Confirmar check-in” y “Cancelar”. (ver FA 3.1) (ver EX1)</p> <p>4. El recepcionista valida los datos y da clic en el botón “Confirmar check-in”. (ver FA 4.1)</p> <p>5. El sistema muestra una ventana con campos a llenar para realizar la pre-autorización de la tarifa estándar (dependiendo de la habitación seleccionada) y depósito de seguridad con cargo en la tarjeta de crédito del huésped con el botón “Realizar pre-autorización”. (ver FA 5.1)</p> <p>6. El recepcionista solicita la tarjeta de crédito del huésped y llena los campos solicitados y da clic en el botón “Realizar pre-autorización”.</p> <p>7. El sistema cambia el estado de la habitación a “Ocupada” y muestra la ventana con un código autogenerado de 6 dígitos para acceder a la habitación junto con el botón “Finalizar”. (ver EX1)</p> <p>8. El recepcionista entrega la llave al huésped y confirma en el sistema con botón “Finalizar”.</p> <p>9. El sistema muestra el mensaje “Check-in registrado exitosamente”.</p> <p>10. Termina el caso de uso.</p>

<b>Nombre</b>	<b>CU07 - Registrar check-in</b>
Flujo alterno	<p><b>FA 3.1</b> No existe la reserva en la base de datos</p> <ol style="list-style-type: none"> <li>1. El sistema muestra la ventana emergente con el mensaje “Reserva no encontrada, vuelve a intentarlo” y el botón “Aceptar”.</li> <li>2. El recepcionista da clic en el botón “Aceptar”.</li> <li>3. Regresa al paso 1 del flujo normal.</li> </ol> <p><b>FA 4.1</b> El recepcionista cancela el proceso de la búsqueda de reserva</p> <ol style="list-style-type: none"> <li>1. El recepcionista da clic en el botón “Cancelar”.</li> <li>2. El sistema muestra la ventana de confirmación con el mensaje “¿Está seguro que quiere salir?” con los botones “Aceptar” y “Cancelar”.</li> <li>3. Si el recepcionista da clic en el botón “Aceptar”, termina el caso de uso.</li> <li>4. Si el recepcionista da clic en el botón “Cancelar”, regresa al paso 3 del flujo normal.</li> </ol> <p><b>FA 5.1</b> El huésped realizó una reserva prepagada</p> <ol style="list-style-type: none"> <li>1. Continúa al paso 7 del flujo normal.</li> </ol>
Excepción	<p><b>EX1.</b> El sistema no puede conectarse con la base de datos.</p> <ol style="list-style-type: none"> <li>1. El sistema muestra la ventana emergente de error y el mensaje “Error en la conexión con la base de datos, inténtalo más tarde” junto al botón “Aceptar”.</li> <li>2. El recepcionista da clic en el botón “Aceptar” para cerrar la ventana.</li> <li>3. Termina el caso de uso.</li> </ol>
Postcondiciones	<p>POST-1. El huésped queda registrado como en estancia</p> <p>POST-2. La habitación cambia su estado a “Ocupada”.</p>

#### **CU08 - Registrar check-out**

<b>Nombre</b>	<b>CU08 - Registrar check-out</b>
Actor(es)	Recepción (AC02)
Objetivo	El recepcionista registra la salida del huésped en el sistema y bloquea la habitación.
Precondiciones	PRE-1. El huésped debe tener una estancia activa.

Nombre	CU08 - Registrar check-out
Flujo normal	<p>1. El sistema muestra una ventana donde solicita el número de reserva y el botón “Buscar”.</p> <p>2. El recepcionista ingresa los datos de búsqueda de la reserva y da clic en la opción “Buscar”.</p> <p>3. El sistema busca la reserva en la base de datos y despliega los detalles de la reserva (nombre del huésped, fecha de inicio y fin, habitación asignada y los cargos pendientes (si hay)) junto a los botones “Confirmar check-out” y “Cancelar”. (ver EX1)</p> <p>4. El recepcionista valida los datos y da clic en el botón “Confirmar check-out”. (ver FA 4.1)</p> <p>5. El sistema calcula el total a pagar y muestra los botones “Cobrar ahora”, “Continuar” y “Regresar”. (ver FA 5.1)</p> <p>6. El recepcionista da clic en el botón “Continuar” (ver FA 6.1) (ver FA 6.2)</p> <p>7. El sistema cambia el estado de la habitación a “Disponible”, genera un comprobante impreso junto con el botón “Finalizar”. (ver EX1)</p> <p>8. El recepcionista entrega el recibo al huésped y confirma en el sistema con botón “Finalizar”.</p> <p>9. El sistema muestra el mensaje “Check-out realizado exitosamente”.</p> <p>10. Termina el caso de uso.</p>

<b>Nombre</b>	<b>CU08 - Registrar check-out</b>
Flujo alterno	<p><b>FA 4.1</b> El recepcionista cancela el proceso de la búsqueda de reserva</p> <ol style="list-style-type: none"> <li>1. El recepcionista da clic en el botón “Cancelar”.</li> <li>2. El sistema muestra la ventana de confirmación con el mensaje “¿Está seguro que quiere salir?” con los botones “Aceptar” y “Cancelar”.</li> <li>3. Si el recepcionista da clic en el botón “Aceptar”, termina el caso de uso.</li> <li>4. Si el recepcionista da clic en el botón “Cancelar”, regresa al paso 3 del flujo normal.</li> </ol> <p><b>FA 5.1</b> La reserva fue de prepago</p> <ol style="list-style-type: none"> <li>1. Continúa al paso 7 del flujo normal.</li> </ol> <p><b>FA 6.1</b> El recepcionista procede a completar el cobro.</p> <ol style="list-style-type: none"> <li>1. El recepcionista da clic en el botón “Cobrar ahora”.</li> <li>2. El flujo normal continúa en el CU09 - Completar cobro</li> <li>3. Termina el caso de uso</li> </ol> <p><b>FA 6.2</b> El recepcionista procede a completar el cobro.</p> <ol style="list-style-type: none"> <li>1. El recepcionista da clic en el botón “Regresar”.</li> <li>2. Regresa al paso 5 del flujo normal.</li> </ol>
Excepción	<p><b>EX1.</b> El sistema no puede conectarse con la base de datos.</p> <ol style="list-style-type: none"> <li>1. El sistema muestra la ventana emergente de error y el mensaje “Error en la conexión con la base de datos, inténtalo más tarde” junto al botón “Aceptar”.</li> <li>2. El recepcionista da clic en el botón “Aceptar” para cerrar la ventana.</li> <li>3. Termina el caso de uso.</li> </ol>
Postcondiciones	<p>POST-1. La habitación se marca como “Disponible”.</p> <p>POST-2. El huésped queda registrado como salida finalizada.</p>

### **CU09 - Completar cobro**

<b>Nombre</b>	<b>CU09 - Completar cobro</b>
Actor(es)	Recepción (AC02)
Objetivo	El recepcionista realiza el cobro al huésped una vez terminada su estancia en el hotel
Precondiciones	PRE-1. Deben existir cargos pendientes por pagar
Flujo normal	<p>1. El sistema muestra la ventana con el monto total y con el botón “Procesar pago”.</p> <p>2. El recepcionista da clic en “Procesar pago”.</p> <p>3. El sistema envía solicitud a pasarela de pagos, y muestra en pantalla el mensaje “La transacción se realizó correctamente” junto con el botón “Aceptar”. (ver EX1)</p> <p>4. El recepcionista observa el resultado en pantalla y confirma con el botón “Aceptar”.</p> <p>5. El sistema registra la transacción en la base de datos, muestra el mensaje “Pago completado exitosamente” junto al botón “Aceptar” y genera comprobante impreso. (ver EX2)</p> <p>6. El recepcionista da clic en “Aceptar” y entrega comprobante impreso al huésped.</p> <p>7. Termina el caso de uso.</p>
Flujo alterno	N/A
Excepción	<p><b>EX1.</b> El sistema no puede conectarse con la pasarela de pagos</p> <p>1. El sistema muestra la ventana emergente de error y el mensaje “Ha ocurrido un error al realizar el pago, inténtalo más tarde” junto al botón “Aceptar”.</p> <p>2. El recepcionista da clic en el botón “Aceptar” para cerrar la ventana.</p> <p>3. Termina el caso de uso.</p> <p><b>EX2.</b> El sistema no puede conectarse con la base de datos.</p> <p>1. El sistema muestra la ventana emergente de error y el mensaje “Error en la conexión con la base de datos, inténtalo más tarde” junto al botón “Aceptar”.</p> <p>2. El recepcionista da clic en el botón “Aceptar” para cerrar la ventana.</p> <p>3. Termina el caso de uso.</p>

<b>Nombre</b>	<b>CU09 - Completar cobro</b>
Postcondiciones	<p>POST-1. El pago queda registrado en el sistema.</p> <p>POST-2. Se genera comprobante impreso.</p>

#### **CU10 - Registrar consumo de servicios**

<b>Nombre</b>	<b>CU10 - Registrar consumo de servicios</b>
Actor(es)	Recepción (AC02)
Objetivo	El recepcionista registra consumos adicionales del huésped durante su instancia (desayuno, consumo de productos, lavandería, etc.)
Precondiciones	PRE-1. El huésped debe tener una estancia activa.
Flujo normal	<p>1. El sistema muestra la ventana “Registro de consumo”, solicitando los datos del huésped y número de habitación junto al botón “Buscar”.</p> <p>2. El recepcionista ingresa los datos solicitados y da clic en “Buscar”.</p> <p>3. El sistema consulta la base de datos y muestra la cuenta activa de dicha habitación, el campo para introducir el nuevo servicio y los botones “Aregar servicio” y “Regresar”. (ver EX1)</p> <p>4. El recepcionista ingresa el tipo de servicio consumido, cantidad y costo y da clic en “Aregar servicio”. (ver FA 4.1)</p> <p>5. El sistema actualiza la cuenta de la habitación y muestra el mensaje “El servicio se agregó correctamente” con el botón “Aceptar”. (ver EX1)</p> <p>6. El recepcionista da clic en “Aceptar”.</p> <p>7. Termina el caso de uso</p>
Flujo alterno	<p><b>FA 4.1</b> El recepcionista decide regresar</p> <p>1. El recepcionista da clic en el botón “Regresar”.</p> <p>2. El sistema muestra la ventana de confirmación con el mensaje “¿Está seguro que quiere salir?” con los botones “Aceptar” y “Cancelar”.</p> <p>3. Si el recepcionista da clic en el botón “Aceptar”, termina el caso de uso.</p> <p>4. Si el recepcionista da clic en el botón “Cancelar”, regresa al paso 3 del flujo normal.</p>

<b>Nombre</b>	<b>CU10 - Registrar consumo de servicios</b>
Excepción	<p><b>EX1.</b> El sistema no puede conectarse con la base de datos.</p> <p>1. El sistema muestra la ventana emergente de error y el mensaje “Error en la conexión con la base de datos, inténtalo más tarde” junto al botón “Aceptar”.</p> <p>2. El recepcionista da clic en el botón “Aceptar” para cerrar la ventana.</p> <p>3. Termina el caso de uso.</p>
Postcondiciones	POST-1. El consumo queda registrado en la cuenta del huésped.

### **CU11 - Gestionar cambio de habitación**

<b>Nombre</b>	<b>CU11 - Gestionar cambio de habitación</b>
Actor(es)	Recepción (AC02)
Objetivo	El recepcionista cambia al huésped de habitación de acuerdo a la disponibilidad y respetando tarifas
Precondiciones	<p>PRE-1. El huésped tiene una estancia activa.</p> <p>PRE-2. Debe haber habitaciones disponibles.</p>

Nombre	CU11 - Gestionar cambio de habitación
Flujo normal	<p>1. El sistema muestra la ventana “Cambio de habitación” solicitando el nombre del huésped y el botón “Buscar”.</p> <p>2. El recepcionista ingresa los datos y presiona “Buscar”.</p> <p>3. El sistema muestra una lista de habitaciones con disponibilidad con el botón “Asignar”. (ver EX2)</p> <p>4. El recepcionista selecciona una habitación y da clic en el botón “Asignar”.</p> <p>5. El sistema muestra la habitación actual a la izquierda y la habitación nueva seleccionada a la izquierda y muestra el cargo a cobrar por el cambio y el botón “Pagar en efectivo” “Pagar con tarjeta de crédito” “Regresar”.</p> <p>6. El recepcionista da clic en “Pagar con tarjeta de crédito”. (ver FA 6.1) (ver FA 6.2)</p> <p>7. El sistema muestra una ventana con campos a llenar para realizar el cargo en la tarjeta de crédito del huésped y los botones “Realizar cobro” y “Cancelar”.</p> <p>8. El recepcionista solicita la tarjeta de crédito del huésped y llena los campos solicitados y da clic en el botón “Realizar cobro”. (ver FA 8.1)</p> <p>9. El sistema envía solicitud a pasarela de pagos, y muestra en pantalla el mensaje “La transacción se realizó correctamente” junto con el botón “Aceptar”. (ver EX1)</p> <p>10. El recepcionista observa el resultado en pantalla y confirma con el botón “Aceptar”.</p> <p>11. El sistema registra la transacción en la base de datos, muestra el mensaje “Pago completado exitosamente” junto al botón “Aceptar”, también cambia el estado de las habitaciones, la anterior a “Disponible” y la actualizada a “Ocupada” y genera comprobante impreso. (ver EX2)</p> <p>12. El recepcionista da clic en “Aceptar” y entrega comprobante impreso al huésped.</p> <p>13. Termina el caso de uso</p>

Nombre	CU11 - Gestionar cambio de habitación
Flujo alterno	<p><b>FA 6.1</b> El huésped decide pagar en efectivo</p> <ol style="list-style-type: none"> <li>1. El recepcionista da clic en el botón “Pagar en efectivo”.</li> <li>2. El sistema muestra en la ventana la cantidad a cobrar y el campo para introducir la cantidad pagada desbloquea la caja registradora.</li> <li>3. El recepcionista introduce la cantidad a cobrar en la caja y escribe en el campo la cantidad introducida.</li> <li>4. El sistema realiza la resta y muestra en pantalla el cambio que se debe regresar.</li> <li>5. El recepcionista toma el dinero de cambio y cierra la caja.</li> <li>6. El sistema genera el comprobante impreso, y registra la transacción en la base de datos, y muestra el mensaje “Pago completado exitosamente” junto al botón “Aceptar”. (ver EX2)</li> <li>7. El recepcionista da clic en el botón “Aceptar” y entrega el comprobante impreso al huésped.</li> <li>8. Termina el caso de uso.</li> </ol> <p><b>FA 6.2</b> El recepcionista decide regresar</p> <ol style="list-style-type: none"> <li>1. El recepcionista da clic en el botón “Regresar”.</li> <li>2. El sistema muestra la ventana de confirmación con el mensaje “¿Está seguro que quiere salir?” con los botones “Aceptar” y “Cancelar”.</li> <li>3. Si el recepcionista da clic en el botón “Aceptar”, termina el caso de uso.</li> <li>4. Si el recepcionista da clic en el botón “Cancelar”, regresa al paso 5 del flujo normal.</li> </ol> <p><b>FA 8.1</b> El recepcionista decide cancelar</p> <ol style="list-style-type: none"> <li>1. El recepcionista da clic en el botón “Cancelar”.</li> <li>2. El sistema muestra la ventana de confirmación con el mensaje “¿Está seguro que quiere salir?” con los botones “Aceptar” y “Cancelar”.</li> <li>3. Si el recepcionista da clic en el botón “Aceptar”, termina el caso de uso.</li> <li>4. Si el recepcionista da clic en el botón “Cancelar”, regresa al paso 7 del flujo normal.</li> </ol>

<b>Nombre</b>	<b>CU11 - Gestionar cambio de habitación</b>
Excepción	<p><b>EX1.</b> El sistema no puede conectarse con la pasarela de pagos</p> <p>1. El sistema muestra la ventana emergente de error y el mensaje “Ha ocurrido un error al realizar el pago, inténtalo más tarde” junto al botón “Aceptar”.</p> <p>2. El recepcionista da clic en el botón “Aceptar” para cerrar la ventana.</p> <p>3. Termina el caso de uso.</p> <p><b>EX2.</b> El sistema no puede conectarse con la base de datos.</p> <p>1. El sistema muestra la ventana emergente de error y el mensaje “Error en la conexión con la base de datos, inténtalo más tarde” junto al botón “Aceptar”.</p> <p>2. El recepcionista da clic en el botón “Aceptar” para cerrar la ventana.</p> <p>3. Termina el caso de uso.</p>
Postcondiciones	POST-1. La reserva se actualiza con la nueva habitación.

### **CU12 - Gestionar bloqueos de habitaciones**

<b>Nombre</b>	<b>CU12 - Gestionar bloqueos de habitaciones</b>
Actor(es)	Recepción (AC02)
Objetivo	El recepcionista marca una habitación como fuera de servicio por determinado motivo (mantenimiento o limpieza) impidiendo su reserva
Precondiciones	PRE-1. La habitación debe existir en el sistema.

Nombre	<b>CU12 - Gestionar bloqueos de habitaciones</b>
Flujo normal	<p>1. El sistema muestra la ventana “Bloqueo de habitación” solicitando el número de habitación, junto al botón “Continuar”.</p> <p>2. El recepcionista introduce el número de habitación a bloquear.</p> <p>3. El sistema verifica que la habitación no esté en estado “Ocupada” y muestra en la ventana un cuadro combinado para seleccionar el motivo junto a los botones “Bloquear” y “Cancelar”. (ver EX1)</p> <p>4. El recepcionista selecciona del cuadro combinado una opción y da clic en “Bloquear”. (ver FA 4.1)</p> <p>5. El sistema cambia el estado de la habitación a “Fuera de servicio” y muestra el mensaje “La habitación se ha bloqueado correctamente” con el botón “Aceptar”. (ver EX1)</p> <p>6. El recepcionista da clic en el botón “Aceptar”.</p> <p>7. Termina el caso de uso</p>
Flujo alterno	<p><b>FA 8.1</b> El recepcionista decide cancelar el proceso</p> <p>1. El recepcionista da clic en el botón “Cancelar”.</p> <p>2. El sistema muestra la ventana de confirmación con el mensaje “¿Está seguro que quiere salir?” con los botones “Aceptar” y “Cancelar”.</p> <p>3. Si el recepcionista da clic en el botón “Aceptar”, termina el caso de uso.</p> <p>4. Si el recepcionista da clic en el botón “Cancelar”, regresa al paso 7 del flujo normal.</p>
Excepción	<p><b>EX1.</b> El sistema no puede conectarse con la base de datos.</p> <p>1. El sistema muestra la ventana emergente de error y el mensaje “Error en la conexión con la base de datos, inténtalo más tarde” junto al botón “Aceptar”.</p> <p>2. El recepcionista da clic en el botón “Aceptar” para cerrar la ventana.</p> <p>3. Termina el caso de uso.</p>
Postcondiciones	POST-1. La habitación queda en estado “Fuera de servicio”.

### **CU13 - Cancelar reserva a petición del cliente en recepción**

<b>Nombre</b>	<b>CU13 - Cancelar reserva a petición del cliente en recepción</b>
Actor(es)	Recepción (AC02)
Objetivo	El recepcionista cancela la reserva directamente en el hotel si el cliente lo solicita
Precondiciones	PRE-1. El huésped debe tener una reserva activa.
Flujo normal	<p>1. El sistema muestra la ventana “Cancelar reserva” con campos a llenar de los datos del huésped junto al botón “Buscar”.</p> <p>2. El recepcionista ingresa los datos solicitados y da clic en “Buscar”.</p> <p>3. El sistema consulta la base de datos y muestra los detalles de la reserva, y las políticas de cancelación junto al botón “Finalizar” y “Regresar”. (ver EX1)</p> <p>4. El recepcionista le explica las políticas al huésped, y da clic “Finalizar”. (ver FA 4.1)</p> <p>5. El sistema actualiza el estado de la reserva a “Cancelada” y muestra el mensaje “Reserva cancelada exitosamente” con el botón “Aceptar”. (ver EX1)</p> <p>6. El recepcionista da clic en “Aceptar”.</p> <p>7. Termina el caso de uso.</p>
Flujo alterno	<p><b>FA 4.1</b> El recepcionista decide regresar</p> <p>1. El recepcionista da clic en el botón “Regresar”.</p> <p>2. Regresa al paso 1 del flujo normal.</p>
Excepción	<p><b>EX1.</b> El sistema no puede conectarse con la base de datos.</p> <p>1. El sistema muestra la ventana emergente de error y el mensaje “Error en la conexión con la base de datos, inténtalo más tarde” junto al botón “Aceptar”.</p> <p>2. El recepcionista da clic en el botón “Aceptar” para cerrar la ventana.</p> <p>3. Termina el caso de uso.</p>
Postcondiciones	<p>POST-1. La reserva queda cancelada</p> <p>POST-2. La habitación queda disponible.</p>

## 2.4.3. AC03 - Administrador

### CU14 - Gestionar hoteles de la cadena

<b>Nombre</b>	<b>Gestionar hoteles de la cadena</b>
Actor(es)	Administrador
Objetivo	El administrador mantiene la información de los hoteles de la cadena actualizada, permitiendo la adición de nuevas propiedades y la modificación de las existentes
Precondiciones	PRE-01 Se tienen los datos mínimos para el alta o la modificación
Flujo normal	<p>1. El administrador selecciona la opción “Gestionar catálogo”</p> <p>2. El sistema presenta la interfaz principal de gestión (incluye la lista de hoteles y la opción de “Registrar Nuevo Hotel”)</p> <p>3. El administrador selecciona la opción “Registrar nuevo hotel” (ver FA 3.1)</p> <p>4. El sistema presenta un formulario para ingresar la información del hotel (nombre, dirección, categoría, contacto, cantidad de habitaciones)</p> <p>5. El administrador ingresa los datos obligatorios y confirma el registro</p> <p>6. El sistema valida la información dada, crea el registro con los datos proporcionados, notifica el éxito de la operación y guarda los movimientos del administrador en una bitácora</p> <p>7. Termina caso de uso</p>
Flujo alterno	<p><b>FA 3.1 Modificación de un Hotel Existente</b></p> <p>1. El administrador busca y selecciona un hotel, da clic en “Modificar información”</p> <p>2. El sistema recupera y muestra los datos actuales en el formulario de edición</p> <p>3. El administrador modifica los campos requeridos y confirma la modificación</p> <p>4. El sistema valida los cambios, actualiza el registro, notifica al administrador y guarda los movimientos del administrador en una bitácora</p> <p>5. Termina caso de uso</p>

<b>Nombre</b>	<b>Gestionar hoteles de la cadena</b>
Excepción	<p><b>EX1.</b> Fallo al registrar movimiento en la bitácora</p> <ol style="list-style-type: none"> <li>1. El sistema revierte la transacción para mantener la integridad entre los datos y la bitácora.</li> <li>2. Además, notifica al administrador del fallo y vuelve al menú principal</li> <li>3. Termina caso de uso</li> </ol>
Postcondición	POST-01 Se crea un nuevo registro de hotel en el catálogo POST-02 El registro del hotel seleccionado se actualiza con los nuevos datos

### **CU15 - Gestionar tipos de habitación**

<b>Nombre</b>	<b>Configurar tipos de habitación</b>
Actor(es)	Administrador
Objetivo	El administrador define, actualiza y gestiona las categorías de habitaciones, sus amenidades y sus capacidades
Precondiciones	PRE-01 El hotel a quien se le asociará las habitaciones ya está registrado en sistema
Flujo normal	<ol style="list-style-type: none"> <li>1. El administrador selecciona la opción “Configurar habitación”</li> <li>2. El sistema presenta la interfaz para seleccionar el hotel al cuál se le van a configurar sus habitaciones.</li> <li>3. El administrador selecciona el hotel y la opción “Registrar nuevo tipo” (ver FA 3.1) (ver FA 3.2)</li> <li>4. El sistema presenta un formulario para ingresar la categorización, las capacidades y la lista de amenidades de la habitación</li> <li>5. El administrador ingresa los datos, selecciona las amenidades y confirma la configuración</li> <li>6. El sistema valida la información, guarda el registro, notifica al administrador y guarda sus movimientos en la bitácora</li> <li>7. Termina caso de uso</li> </ol>

Nombre	Configurar tipos de habitación
Flujo alterno	<p><b>FA 3.1</b> Modificación de un Hotel Existente</p> <ol style="list-style-type: none"> <li>1. El administrador selecciona un hotel, da clic en “Modificar Configuración”</li> <li>2. El sistema recupera y muestra los datos actuales de los tipos de habitación</li> <li>3. El administrador actualiza los campos requeridos</li> <li>4. El sistema valida los cambios, actualiza el registro, notifica al administrador sobre el éxito de la operación y guarda sus movimientos en la bitácora</li> <li>5. Termina caso de uso</li> </ol> <p><b>FA 3.2</b> Dar de baja un tipo de habitación</p> <ol style="list-style-type: none"> <li>1. El administrador selecciona un hotel, da clic en “Dar de baja el tipo”</li> <li>2. El sistema verifica si existen habitaciones activas o reservas futuras asociadas con ese tipo de habitación</li> <li>3. Después, pide la confirmación de baja</li> </ol>
Excepción	<p><b>EX1.</b> Fallo al registrar movimiento en la bitácora</p> <ol style="list-style-type: none"> <li>1. El sistema revierte la transacción para mantener la integridad entre los datos y la bitácora.</li> <li>2. Además, notifica al administrador del fallo y vuelve al menú principal</li> <li>3. El administrador confirma la baja</li> <li>4. El sistema cambia el estado de la habitación, notifica al administrador el éxito de la operación y guarda en la bitácora los movimientos de la operación</li> <li>5. Termina caso de uso</li> </ol>
Postcondición	<p>POST-01 Se crea un nuevo registro de habitación en un hotel del catálogo</p> <p>POST-02 La habitación seleccionada de un hotel seleccionado se actualiza con los nuevos datos</p> <p>POST-03 La habitación seleccionada de un hotel seleccionado es dada de baja</p>

## CU16 - Configurar Precios

Nombre	Configurar Precios
Actor(es)	Administrador (AC03)
Objetivo	El sistema debe permitir al Administrador definir reglas complejas que determinen cuánto cuesta una habitación en cualquier momento dado.
Precondiciones	PRE-01 Se tienen los datos mínimos para el alta o la modificación
Disparador	El Administrador selecciona la opción “Gestión de Precios”
Flujo normal	<p>1. El sistema muestra una lista de las reglas y promociones existentes</p> <p>2. El Administrador selecciona “Crear Nueva Regla/Promoción” (ver FA 2.1) (ver FA 2.2)</p> <p>3. El sistema pide el alcance de la regla (si aplica a un hotel, a un tipo de habitación en específico)</p> <p>4. El Administrador define el Periodo de Aplicación y el mecanismo de precio (tarifa fija, modificador de porcentaje o descuento promocional)</p> <p>5. El sistema valida que la nueva regla no supere ni contradiga una regla existente con mayor prioridad y guarda la nueva regla en el catálogo</p> <p>6. Termina el caso de uso</p>

Nombre	Configurar Precios
Flujo alterno	<p><b>FA 2.1</b> Modificación de una Regla de Precio Existente</p> <ol style="list-style-type: none"> <li>1. El administrador da clic en “Modificar Regla”</li> <li>2. El sistema recupera las reglas activas y las presenta junto con herramientas de filtro (por hotel, tipo de hotel, rango de fechas)</li> <li>3. El administrador busca y selecciona la regla que desea modificar</li> <li>4. El sistema recupera y muestra los parámetros actuales de la regla en el formulario de edición</li> <li>5. El administrador realiza los cambios necesarios y confirma la modificación</li> <li>6. El sistema valida la nueva información y actualiza la regla en el catálogo (ver EX-1)</li> <li>7. Termina caso de uso</li> </ol> <p><b>FA 2.2</b> Copiar Regla de Precio o Promoción a otro Periodo/Hotel</p> <ol style="list-style-type: none"> <li>1. El administrador da clic en Duplicar regla</li> <li>2. El sistema genera una copia de la regla y pide al Administrador que defina los nuevos parámetros (hotel, rango de fechas de aplicación)</li> <li>3. El Administrador ingresa los nuevos parámetros y confirma la creación de la copia</li> <li>4. El sistema valida la información, crea y guarda la nueva regla en el catálogo</li> <li>5. Termina caso de uso</li> </ol>
Excepción	<p><b>EX1.</b> Conflicto de Reglas</p> <ol style="list-style-type: none"> <li>1. El sistema detecta que la nueva promoción tiene un periodo que se superpone con otra promoción existente.</li> <li>2. El sistema fuerza a ajustar los parámetros</li> <li>3. Termina caso de uso</li> </ol>

<b>Nombre</b>	<b>Configurar Precios</b>
Postcondición	<p>POST-01 Se registra una nueva regla de pago</p> <p>POST-02 Una regla de pago es modificada</p> <p>POST-03 Se copia una regla de pago en otro hotel</p>

### **CU17 - Configurar Políticas de Overbooking**

<b>Nombre</b>	<b>Configurar Políticas de Overbooking</b>
Actor(es)	Administrador (AC03)
Objetivo	El sistema permite que el administrador ajuste el umbral de riesgo aceptable para cada tipo de habitación
Precondiciones	PRE-01 Se tienen los datos mínimos para el alta o la modificación
Disparador	El Administrador selecciona la opción “Gestión de Overbooking”
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra una lista de las políticas existentes</li> <li>2. El Administrador selecciona la opción “Definir Nueva Política” (ver FA 2.1) (ver FA 2.2)</li> <li>3. El sistema presenta la lista de hoteles y tipos de habitación</li> <li>4. El Administrador selecciona el hotel, el tipo de habitación; define el Periodo de Aplicación de la política y el Porcentaje Máximo de Overbooking</li> <li>5. El sistema valida que el porcentaje sea menor al 10% y que las fechas sean válidas, guarda la política y ajusta la disponibilidad</li> <li>6. Termina caso de uso</li> </ol>

Nombre	Configurar Políticas de Overbooking
Flujo alterno	<p><b>FA 2.1</b> Modificación de una Política de Overbooking Existente</p> <ol style="list-style-type: none"> <li>1. El administrador da clic en “Modificar Política Existente”</li> <li>2. El sistema recupera el listado de las políticas activas y las presenta junto con herramientas de filtro (por hotel, tipo de hotel, rango de fechas)</li> <li>3. El administrador busca y selecciona la política que desea modificar</li> <li>4. El sistema recupera y muestra los parámetros actuales de la política en el formulario de edición</li> <li>5. El administrador realiza los cambios necesarios y confirma la modificación</li> <li>6. El sistema valida la nueva información y actualiza la política en el catálogo</li> <li>7. Termina caso de uso</li> </ol> <p><b>FA 2.2</b> Dar de baja una política de Overbooking</p> <ol style="list-style-type: none"> <li>1. El administrador selecciona la opción “Eliminar Política”</li> <li>2. El sistema presenta el listado de políticas activas y pide la selección</li> <li>3. El Administrador selecciona la política que desea eliminar</li> <li>4. El sistema pide al Administrador la confirmación final de la eliminación</li> <li>5. El Administrador confirma la eliminación</li> <li>6. El sistema elimina la política del catálogo, volviendo la disponibilidad de esas habitaciones a su capacidad física real (ver EX-1)</li> <li>7. Termina caso de uso</li> </ol>

Nombre	Configurar Políticas de Overbooking
Excepción	<p><b>EX.1 Fallo en la Actualización de Disponibilidad</b></p> <p>1. El sistema falla al enviar la actualización de la disponibilidad efectiva al módulo de reservas</p> <p>2. Se realiza una reversión de la transacción y se notifica al Administrador “Error crítico al actualizar el sistema de disponibilidad. La operación ha sido revertida y no está activa”</p> <p>3. Termina caso de uso</p>
Postcondición	<p>POST-01 Se registra una nueva política de overbooking</p> <p>POST-02 Una política de overbooking es modificada</p> <p>POST-03 Se elimina una política de overbooking correctamente</p>

### CU18 - Monitorear inventario

Nombre	<b>CU18 - Monitorear inventario</b>
Actor(es)	Administrador (AC03)
Objetivo	El administrador obtiene una vista panorámica de cómo está el negocio para tomar decisiones estratégicas
Precondiciones	PRE-1 Existen hoteles registrados y accesibles dentro del sistema
Postcondiciones	POST-1 El Administrador tiene visibilidad del estado actual del inventario según los criterios especificados
Flujo normal	<p>1. El sistema solicita criterios de consulta como fecha, hotel, región y tipo de habitación</p> <p>2. El administrador especifica los criterios de consulta (ver FA 2.1)</p> <p>3. El sistema recupera la información del inventario que coincide con los criterios seleccionados, calcula métricas (% de ocupación, habitaciones disponibles y tendencias) y presenta el consolidado de disponibilidad y ocupación (ver FA 3.1) (ver EX-1) (ver EX-2)</p> <p>4. Termina caso de uso</p>

<b>Nombre</b>	<b>CU18 - Monitorear inventario</b>
Flujo alterno	<p><b>FA 2.1</b> El administrador cancela la consulta</p> <ol style="list-style-type: none"> <li>1. El Administrador decide no continuar con la consulta</li> <li>2. Termina caso de uso</li> </ol> <p><b>FA 3.1</b> El administrador modifica los criterios</p> <ol style="list-style-type: none"> <li>1. El administrador decide cambiar los filtros de búsqueda</li> <li>2. Regresa al paso 1</li> </ol>
Excepciones	<p><b>EX-1</b> Algún hotel no reporta información actualizada</p> <ol style="list-style-type: none"> <li>1. El sistema presenta los datos disponibles e indica qué hoteles no tienen información actualizada</li> <li>2. Termina caso de uso</li> </ol> <p><b>EX-2</b> No hay datos históricos suficientes para el periodo solicitado</p> <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje para hacerle saber al Administrador que no se encontraron resultados con los criterios seleccionados</li> <li>2. Termina caso de uso</li> </ol>

#### 2.4.4. AC04 - Auditoría

##### CU19 - Generar Reporte de Anulaciones Sospechosas

<b>Nombre</b>	<b>CU19 - Generar Reporte de Anulaciones Sospechosas</b>
Actor(es)	Auditor (AC04)
Objetivo	El auditor necesita detectar patrones sospechosos en las reservaciones
Precondiciones	PRE-1. Existen reservas registradas y accesibles dentro del sistema.
Postcondiciones	POST-1 El Auditor tiene evidencia documentada para tomar acción de los movimientos sospechosos

Nombre	<b>CU19 - Generar Reporte de Anulaciones Sospechosas</b>
Flujo normal	<p>1. El sistema solicita criterios de consulta como fechas, hotel, tipo de anomalía</p> <p>2. El auditor especifica los criterios (ver FA 2.1) (ver FA 3.1)</p> <p>3. El sistema analiza registros históricos identificando patrones anómalos de cancelación y genera un reporte con los hallazgos</p> <p>4. Termina el caso de uso</p>
Flujo alterno	<p><b>FA 2.1</b> El Auditor cancela la consulta 1. El Auditor decide no continuar con la consulta</p> <p>2. Termina caso de uso</p> <p><b>FA 3.1</b> El Auditor modifica los criterios</p> <p>1. El Auditor decide cambiar los filtros de búsqueda</p> <p>2. Regresa al paso 1</p>
Excepciones	<p><b>EX1.</b> Algún hotel no reporta información actualizada</p> <p>1. El sistema presenta los datos disponibles e indica qué hoteles no tienen información actualizada además de la información faltante</p> <p>2. Regresa al flujo normal</p> <p><b>EX2.</b> No hay datos históricos suficientes para el análisis</p> <p>1. El sistema muestra un mensaje para hacerle saber al Auditor que no se encontraron resultados con los criterios de consulta seleccionados</p> <p>2. Regresa al flujo normal</p>

#### **CU20 - Revisar ajustes de precios en reservas**

Nombre	<b>CU20 - Revisar ajustes de precios en reservas</b>
Actor(es)	Auditor (AC04)
Objetivo	El auditor necesita detectar si alguien modificó el precio de una reserva después de crearla
Precondiciones	PRE-1. Existen reservas registradas y accesibles dentro del sistema.
Postcondiciones	POST-1 El Auditor tiene evidencia documentada para tomar acción de los cambios sospechosos

Nombre	<b>CU20 - Revisar ajustes de precios en reservas</b>
Flujo normal	<p>1. El sistema solicita criterios de consulta como fechas, hotel, rango de monto de ajuste, tipo de ajuste y usuario que realizó el cambio</p> <p>2. El auditor especifica los criterios (ver FA 2.1)</p> <p>3. El sistema analiza registros históricos identificando patrones anómalos de cambios de precios o descuentos injustificados y presenta el reporte con el Auditor (ver FA 3.1) (ver EX-1) (ver EX-2) . Termina el caso de uso</p>
Flujo alterno	<p><b>FA 2.1</b> El Auditor cancela la consulta</p> <p>1. El Auditor decide no continuar con la consulta</p> <p>2. Termina caso de uso</p> <p><b>FA 3.1</b> El Auditor modifica los criterios</p> <p>1. El Auditor decide cambiar los filtros de búsqueda</p> <p>2. Regresa al paso 1</p>
Excepciones	<p><b>EX-1</b> Algún hotel no reporta información actualizada</p> <p>1. El sistema presenta los datos disponibles e indica qué hoteles no tienen información actualizada además de la información faltante</p> <p>2. Termina caso de uso</p> <p><b>EX-2</b> No hay datos históricos suficientes para el análisis</p> <p>1. El sistema muestra un mensaje para hacerle saber al Auditor que no se encontraron resultados con los criterios de consulta seleccionados</p> <p>2. Regresa al paso 1 del flujo normal</p>

### **CU21 - Conciliar ingresos por servicios adicionales**

Nombre	<b>CU21 - Conciliar ingresos por servicios adicionales</b>
Actor(es)	Auditor (AC04)
Objetivo	El auditor necesita detectar alguna discrepancia entre los servicios registrados y los servicios reportados
Precondiciones	PRE-1. Existen servicios registrados y accesibles dentro del sistema.
Postcondiciones	POST-1 El Auditor tiene visibilidad de inconsistencias entre servicios consumidos y cobrados

<b>Nombre</b>	<b>CU21 - Conciliar ingresos por servicios adicionales</b>
Flujo normal	<p>1. El sistema solicita criterios de consulta como fechas, hotel y tipo de servicio</p> <p>2. El auditor especifica los criterios (ver FA 2.1)</p> <p>3. El sistema compara registros de consumo contra registro de cobro e identifica discrepancias. Después presenta el reporte de conciliación al auditor (ver FA 3.1, ver FA 3.2, ver EX-1, ver EX-2)</p> <p>4. Termina el caso de uso</p>
Flujo alterno	<p><b>FA 2.1</b> El Auditor cancela la consulta</p> <p>1. El Auditor decide no continuar con la consulta</p> <p>2. Termina caso de uso</p> <p><b>FA 3.1</b> El Auditor modifica los criterios</p> <p>1. El Auditor decide cambiar los filtros de búsqueda</p> <p>2. Regresa al paso 1</p> <p><b>FA 3.2</b> No se detectaron inconsistencias</p> <p>1. El sistema informa que todos los servicios están correctamente conciliados</p> <p>2. Termina caso de uso</p>
Excepciones	<p><b>EX-1</b> Algún hotel no reporta información actualizada</p> <p>1. El sistema presenta los datos disponibles e indica qué hoteles no tienen información actualizada además de la información faltante</p> <p>2. Termina caso de uso</p> <p><b>EX-2</b> No hay registros para el periodo consultado</p> <p>1. El sistema informa que no existen registros de servicios para los criterios especificados</p> <p>2. Regresa al paso 1</p>
Reglas de negocio	RN-1 Se considera una inconsistencia cuando: . Un servicio fue consumido pero no cobrado . Un servicio fue cobrado pero no hay registro de consumo . El monto cobrado no coincide con la tarifa del servicio . Hay servicios consumidos después de registrado el check-out

## CU22 - Rastrear reembolsos y descuentos

Nombre	<b>CU22 - Rastrear reembolsos y descuentos</b>
Actor(es)	Auditor (AC04)
Objetivo	El auditor necesita verificar que cada reembolso y descuento tenga una justificación válida
Precondiciones	PRE-1. Existen reembolsos y/o descuentos registrados en el sistema
Postcondiciones	POST-1 El Auditor tiene evidencia documentada de reembolsos y descuentos para validar su justificación
Flujo normal	<p>1. El sistema solicita criterios de consulta como fechas, hotel, tipo: reembolso/descuento, rango de monto, usuario autorizador</p> <p>2. El auditor especifica los criterios (ver FA 2.1)</p> <p>3. El sistema recupera el historial de reembolsos/descuentos con su trazabilidad (quién lo hizo, cuándo lo hizo, motivo y autorización) e identifica casos sin justificación o fuera de política (ver FA 3.1, ver EX-1, ver EX-2)</p> <p>4. Termina el caso de uso</p>
Flujo alterno	<p><b>FA 2.1</b> El Auditor cancela la consulta</p> <p>1. El Auditor decide no continuar con la consulta</p> <p>2. Termina caso de uso</p> <p><b>FA 3.1</b> El Auditor modifica los criterios</p> <p>1. El Auditor decide cambiar los filtros de búsqueda</p> <p>2. Regresa al paso 1</p>
Excepciones	<p><b>EX-1</b> Algún hotel no reporta información actualizada</p> <p>1. El sistema presenta los datos disponibles e indica qué hoteles no tienen información actualizada además de la información faltante</p> <p>2. Termina caso de uso</p> <p><b>EX-2</b> No hay registros para el periodo consultado</p> <p>1. El sistema informa que no existen registros de servicios para los criterios especificados</p> <p>2. Regresa al paso 1</p>

<b>Nombre</b>	<b>CU22 - Rastrear reembolsos y descuentos</b>
Reglas de negocio	RN-1 Se considera sospechoso un reembolso/descuento que: . No tenga justificación documentada . No tiene autorización o fue autorizado por usuario sin permisos . No corresponde con las políticas vigentes . No tiene documentación de respaldo para montos altos

### **CU23 - Validar cobros de pagos en línea**

<b>Nombre</b>	<b>CU23 - Validar cobros de pagos en línea</b>
Actor(es)	Auditor (AC04)
Objetivo	El auditor necesita conciliar que coincidan los registros del sistema, la pasarela de pago y el banco
Precondiciones	PRE-1. Existen pagos en línea registrados en el sistema
Postcondiciones	POST-1 El Auditor tiene visibilidad de discrepancias entre los pagos registrados procesados y recibidos
Flujo normal	<p>1. El sistema solicita criterios de análisis como fechas, hotel, estado de pago, rango de monto</p> <p>2. El auditor especifica los criterios (ver FA 2.1)</p> <p>3. El sistema compara registros internos con registros de la pasarela de pago para después identificar inconsistencias y presentar el reporte de conciliación (ver FA 3.1, ver EX-1, ver EX-2)</p> <p>4. Termina el caso de uso</p>
Flujo alterno	<p><b>FA 2.1</b> El Auditor cancela la consulta</p> <p>1. El Auditor decide no continuar con la consulta</p> <p>2. Termina caso de uso</p> <p><b>FA 3.1</b> El Auditor modifica los criterios</p> <p>1. El Auditor decide cambiar los filtros de búsqueda</p> <p>2. Regresa al paso 1</p>

<b>Nombre</b>	<b>CU23 - Validar cobros de pagos en línea</b>
Excepciones	<p><b>EX-1</b> Información incompleta de la pasarela de pago</p> <ol style="list-style-type: none"> <li>1. El sistema presenta el reporte con los datos disponibles e indica que no pudo obtener información actualizada de la pasarela de pago</li> <li>2. Termina caso de uso</li> </ol> <p><b>EX-2</b> No hay registros para el periodo consultado</p> <ol style="list-style-type: none"> <li>1. El sistema informa que no existen registros de pagos en línea para los criterios especificados</li> <li>2. Regresa al paso 1</li> </ol>
Reglas de negocio	<p>RN-1 Se considera una inconsistencia cuando: . El sistema registra un pago como exitoso pero la pasarela lo tiene como rechazado o no existe . Existe un pago en la pasarela sin registro en el sistema . El monto registrado no coincide con el monto procesado . Hay pagos duplicados</p> <p>RN-2 Para cada pago debe existir un ID de transacción único proporcionado por la pasarela de pago</p>

#### **CU24 - Consultar bitácora de accesos del personal**

<b>Nombre</b>	<b>CU24 - Consultar bitácora de accesos del personal</b>
Actor(es)	Auditor (AC04)
Objetivo	El auditor necesita consultar el historial de accesos del personal al sistema para identificar actividades sospechosas
Precondiciones	PRE-1. El sistema registra automáticamente todos los accesos del personal
Postcondiciones	POST-1 El Auditor tiene visibilidad de los movimientos del personal en el sistema
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema solicita criterios de consulta como fecha, empleado, hotel y módulo/tipo de acción</li> <li>2. El auditor especifica los criterios de búsqueda (ver FA 2.1)</li> <li>3. El sistema recupera los registros de acceso que coinciden con los criterios y los presenta al Auditor (ver FA 3.1, ver EX-1)</li> <li>4. Si el auditor identifica actividad sospechosa, marca los registros para investigación adicional</li> <li>5. Termina el caso de uso</li> </ol>

<b>Nombre</b>	<b>CU24 - Consultar bitácora de accesos del personal</b>
Flujo alterno	<p><b>FA 2.1</b> El Auditor cancela la consulta</p> <ol style="list-style-type: none"> <li>1. El Auditor decide no continuar con la consulta</li> <li>2. Termina caso de uso</li> </ol> <p><b>FA 3.1</b> El Auditor modifica los criterios</p> <ol style="list-style-type: none"> <li>1. El Auditor decide cambiar los filtros de búsqueda</li> <li>2. Regresa al paso 1</li> </ol>
Excepciones	<p><b>EX-1</b> No hay registros para los criterios consultados</p> <ol style="list-style-type: none"> <li>1. El sistema informa que no existen registros de acceso para los criterios especificados</li> <li>2. Regresa al paso 1</li> </ol>
Reglas de negocio	<p>RN-1 Se considera actividad sospechosa cuando: . Hay acceso al sistema fuera del horario laboral sin justificación . Hay múltiples intentos de acceso fallidos . Hay acceso a módulos sin permisos asignados</p> <p>RN-2 La bitácora debe registrar para cada acceso: . Usuario (empleado) . Fecha y hora . Módulo/funcionalidad accedida . Acción realizada . Terminal desde donde accedió . Resultado</p>

### **CU25 - Consultar informe de inventario**

<b>Nombre</b>	<b>CU25 - Consultar informe de inventario</b>
Actor(es)	Auditor (AC04)
Objetivo	El auditor necesita obtener un reporte del estado actual de las habitaciones en el sistema para verificarlos contra el estado físico real
Precondiciones	PRE-1. Existen habitaciones y reservaciones registradas y accesibles en el sistema
Postcondiciones	POST-1 El Auditor tiene el reporte del estado del inventario en sistema para comparación manual

Nombre	<b>CU25 - Consultar informe de inventario</b>
Flujo normal	<p>1. El sistema solicita criterios de consulta como fecha, hotel y tipo de habitación</p> <p>2. El auditor especifica los criterios de búsqueda (ver FA 2.1)</p> <p>3. El sistema recupera los registros de las habitaciones y su estado para generar y presentar el reporte (ver FA 3.1, ver EX-1)</p> <p>4. Termina caso de uso</p>
Flujo alterno	<p><b>FA 2.1</b> El Auditor cancela la consulta</p> <p>1. El Auditor decide no continuar con la consulta</p> <p>2. Termina caso de uso</p> <p><b>FA 3.1</b> El Auditor modifica los criterios</p> <p>1. El Auditor decide cambiar los filtros de búsqueda</p> <p>2. Regresa al paso 1</p>
Excepciones	<p><b>EX-1</b> No hay registros para los criterios consultados</p> <p>1. El sistema informa que no existen habitaciones registradas para los criterios especificados</p> <p>2. Regresa al paso 1</p>
Reglas de negocio	RN-1 El reporte debe incluir para cada habitación: . Número de habitación . Estado actual de la habitación . Tipo de habitación . Última actualización de estado

#### **CU26 - Validar uniformidad de políticas y precios por región**

Nombre	<b>CU26 - Validar uniformidad de políticas y precios por región</b>
Actor(es)	Auditor (AC04)
Objetivo	El auditor genera reportes que comparan tarifas, políticas de cancelación y promociones aplicadas en diferentes hoteles, regiones o países para asegurar que sigan las directrices de la marca
Precondiciones	PRE-1. Existe políticas, tarifas y promociones registradas para los hoteles de la cadena
Postcondiciones	POST-1 El Auditor tiene el reporte de comparación entre diferentes hoteles o regiones

Nombre	<b>CU26 - Validar uniformidad de políticas y precios por región</b>
Flujo normal	<p>1. El sistema solicita criterios de consulta como fecha, hoteles/regiones, tipo de habitación y tipo de política (cancelación/overbooking/entre otros)</p> <p>2. El auditor especifica los criterios de búsqueda (ver FA 2.1)</p> <p>3. El sistema recupera y compara las políticas, tarifas y promociones de los hoteles/regiones seleccionados e identifica discrepancias respecto a las directrices para presentarlas en un reporte comparativo (ver FA 3.1, ver EX-1, ver EX-2)</p> <p>4. Termina caso de uso</p>
Flujo alterno	<p><b>FA 2.1</b> El Auditor cancela la consulta</p> <p>1. El Auditor decide no continuar con la consulta</p> <p>2. Termina caso de uso</p> <p><b>FA 3.1</b> El Auditor modifica los criterios</p> <p>1. El Auditor decide cambiar los filtros de búsqueda</p> <p>2. Regresa al paso 1</p>
Excepciones	<p><b>EX-1</b> No hay registros para los criterios consultados</p> <p>1. El sistema informa que no existen políticas o tarifas para los criterios especificados</p> <p>2. Regresa al paso 1</p> <p><b>EX-2</b> Información incompleta de algún hotel o región</p> <p>1. El sistema presenta el reporte con los datos disponibles e indica qué hoteles no tienen información completa</p> <p>2. Continúa con paso 4</p>
Reglas de negocio	<p>RN-1 Se considera una discrepancia cuando: . Las tarifas base para el mismo tipo de habitación/hotel/región varían más de cierto porcentaje . Hay políticas que difieren entre hoteles sin justificación aprobada . Los porcentajes de overbooking exceden o están por debajo del estándar corporativo</p> <p>RN-2 El reporte debe comparar: . Tarifas base por tipo de habitación/hotel/región . Políticas de cancelación . Políticas de overbooking . Promociones activas</p>

# **Chapter 3. Impulsores Arquitectónicos (ASRs) y Priorización**

## **3.1. Utility Tree**

Usabilidad	Facilidad de uso	(H,M) QAS-01 Huésped consulta hoteles	Un huésped busca hoteles en "Cancún"; el sistema usa su historial de búsquedas y muestra primero los hoteles más relevantes en ≤ 2s.
		(H,M) QAS-02 Huésped consulta la disponibilidad de una habitación	El huésped consulta habitaciones dobles; el sistema muestra solo las disponibles en el inventario actualizado evitando que seleccione una ocupada.
		(H,M) QAS-04 Huésped selecciona una habitación y confirma la reserva	El huésped confirma reserva y antes de pagar recibe un resumen con fechas, precios y políticas de cancelación para minimizar errores.
	Minimizar errores	(M,L) QAS-09 Huésped cancela una reserva	El huésped da clic en "Cancelar"; el sistema muestra una advertencia con la política de cancelación y un botón de confirmación.
		(M,M) QAS-07 Huésped modifica alguna característica de su reserva	El huésped entra a "Mis reservas", cambia la fecha de check-in en un solo paso con validación de disponibilidad inmediata.
	Simplicidad de interfaz	(M,L) QAS-10 Huésped revisa todas las reservas pasadas y activas realizadas por él	El huésped ve todas sus reservas pasadas y activas organizadas cronológicamente con filtros por estado.
		(H,M) QAS-12 Recepcionista registra la entrada del huésped	Un recepcionista novato registra un check-in en menos de 1 minuto gracias a una interfaz guiada por pasos.
	Eficiencia operativa	(H,M) QAS-14 Recepcionista registra la salida del huésped	El recepcionista finaliza el check-out en ≤ 1 minuto con un resumen automático de consumos y pagos pendientes.
		(M,M) QAS-16 Recepcionista realiza el cobro al huésped terminada su estancia	El recepcionista cobra con tarjeta en un solo paso desde la misma pantalla de check-out, sin menús adicionales.
		(M,M) QAS-20 Recepcionista cambia al huésped de habitación	El recepcionista puede mover al huésped a otra habitación en un solo paso, sin tener que cancelar y crear una nueva reserva.
Jutility tree	Disponibilidad continua	(H,H) QAS-03 Huésped consulta la disponibilidad de una habitación	La BD principal falla; el sistema redirige a una réplica y responde en ≤ 2s con la disponibilidad correcta.
		(H,H) QAS-21 Recepcionista cambia al huésped de habitación	Si falla la BD en medio del cambio, la reserva permanece en la habitación original sin corrupción de datos.
		(M,L) QAS-22 Recepcionista marca una habitación como fuera de servicio	El recepcionista marca una habitación como fuera de servicio; si la BD falla, el cambio no se aplica y se mantiene la disponibilidad original.
		(H,L) QAS-23 Recepcionista cancela la reserva del huésped	El recepcionista cancela una reserva, pero si la BD falla, la reserva sigue activa sin pérdida de información.
		(H,M) QAS-24 Administrador realiza altas y modificaciones de hoteles	El administrador agrega un hotel; si la BD falla, el cambio no se guarda y se notifica al usuario.
	Disponibilidad	(H,H) QAS-30 Administrador ajusta hasta el overbooking	Si falla la BD, las políticas no se aplican y el sistema conserva la configuración previa.
		(H,H) QAS-05 Huésped selecciona una habitación y confirma la reserva	Si ocurre una caída de red en el momento del pago, la habitación se libera automáticamente en ≤ 1s y no queda bloqueada.
		(H,H) QAS-08 Huésped modifica alguna característica de su reserva	Si la modificación de fecha falla por error de BD, la reserva original permanece intacta y el cliente recibe notificación.
	Tolerancia a fallos	(H,M) QAS-13 Recepcionista registra la entrada del huésped	Si la BD se desconecta, el check-in queda en cola y se sincroniza automáticamente cuando el servicio se restablece.
		(H,H) QAS-15 Recepcionista registra la salida del huésped	Si falla el sistema en medio del check-out, no se pierde información; la reserva sigue activa hasta completar la operación.
Seguridad	Fallo en sistema externo	(H,M) QAS-18 Recepcionista realiza el cobro al huésped terminada su estancia	Si falla la pasarela de pagos, el sistema guarda el estado como "pendiente de pago" sin perder la reserva.
		(H,H) QAS-19 Recepcionista registra consumos adicionales del huésped	Si el recepcionista registra consumos de minibar y ocurre un fallo, la transacción se revierte para no duplicar cargos.
	Corrupción de datos	(H,H) QAS-26 Administrador configura habitaciones disponibles mediante categorización, amenidades y capacidades	Si ocurre un error al guardar los cambios, se reivierta la operación para mantener la consistencia.
		(H,H) QAS-27 Administrador define precios diarios, temporadas altas y promociones	Si al registrar precios ocurre un error, se reivierten todos los cambios y no se guardan valores parciales.
	Integridad de pagos	(H,H) QAS-06 Huésped selecciona una habitación y confirma la reserva	Un pago con tarjeta se procesa una sola vez; no puede registrarse doble cargo ni doble reserva.
		(H,H) QAS-17 Recepcionista realiza el cobro al huésped terminada su estancia	El pago se procesa como atómico; o se confirma cobro + cierre de reserva, o no se aplica nada.
	Autenticación	(H,M) QAS-28 Administrador define precios diarios, temporadas altas y promociones	Solo un administrador con rol válido puede actualizar precios; intentos no autorizados se bloquen y registran.
		(H,M) QAS-32 Un auditor busca reservas canceladas poco después de su creación	Un auditor detecta que la reserva #321 fue cancelada 3 minutos después de crearse, con registro de usuario e IP.
	Trazabilidad de operaciones	(H,M) QAS-40 El auditor pude ver un registro de cada inicio de sesión y la actividad del personal	Un auditor consulta la bitácora de accesos y ve que el recepcionista X inició sesión a las 09:02 desde la terminal Y.
		(H,H) QAS-33 Un auditor rastrea cambios en el precio de una reserva después de confirmar	Un auditor revisa que el precio de la reserva #456 cambió después de confirmarse y ve quién lo hizo, cuándo y desde qué dispositivo.
Rendimiento	Autenticidad de datos	(H,H) QAS-35 El auditor sigue reembolsos y descuentos aplicados, asegurando que estén justificados	Un reembolso solo se aprueba si se asocia a una política válida (ej. "cancelación justificada").
		(H,H) QAS-37 Genera un reporte para verificar que los pagos en línea se hayan procesado correctamente	Se genera reporte y se valida que cada reserva confirmada tenga un pago en línea correspondiente en BD.
	Latencia en consultas	(M,M) QAS-11 Consultar historial de reservas	La consulta de historial con más de 50 reservas responde en ≤ 2s.
		(M,M) QAS-25 Administrador configura habitaciones disponibles mediante, categorización, amenidades y capacidades	El administrador modifica categorías de habitación y el catálogo se actualiza en ≤ 2s.
	Alta velocidad de escritura de datos	(M,M) QAS-31 Administrador consulta disponibilidad y ocupación	El administrador consulta ocupación general y recibe resultados en ≤ 2s.
		(M,M) QAS-29 Administrador ajusta hasta el overbooking	El administrador define un overbooking del 10% en varios hoteles y la escritura en BD ocurre en ≤ 1s.
	Cuellos de botella	(M,M) QAS-38 Genera un reporte para verificar que los pagos en línea se hayan procesado correctamente	El auditor consulta cobros de todo un mes y recibe respuesta en ≤ 5s gracias a índices en BD.
		(M,M) QAS-42 El auditor compara el estado de las habitaciones en el sistema con reportes físicos	El auditor consulta disponibilidad de todas las habitaciones de la cadena (5000 hoteles) y recibe resultados en ≤ 5s.
Confidencialidad	Consistencia de datos	(H,H) QAS-34 El auditor compra ingresos de servicios adicionales con registros de venta	El auditor compra ingresos por minibar contra facturación; los totales coinciden exactamente.
		(H,L) QAS-41 El auditor compra el estado de las habitaciones en el sistema con reportes físicos	El auditor compara reporte físico de habitaciones contra el sistema y ambas fuentes coinciden.
	Precisión de resultados	(H,H) QAS-36 El auditor sigue reembolsos y descuentos aplicados, asegurando que estén justificados	El auditor confirma que los montos devueltos al cliente son exactamente iguales a los registrados en el sistema.
		(H,M) QAS-39 Genera un reporte para verificar que los pagos en línea se hayan procesado correctamente	El reporte de pagos refleja datos exactos y consistentes con la pasarela de pagos externa.
	Veracidad de datos	(H,M) QAS-43 El auditor compara tarifas y políticas entre hoteles de diferentes regiones	El auditor genera reporte comparando precios de habitaciones estándar en 3 regiones distintas; todos cumplen con políticas corporativas.

## 3.2. Escenarios de atributos de calidad

### 3.2.1. Rendimiento

ID y descripción corta	<b>QAS-11 Huésped revisa todas las reservas pasadas y activas realizadas por él</b>
Escenario	El huésped consulta su historial de reservas y el sistema debe responder rápidamente.
Atributo	Rendimiento
Preocupación del atributo	Latencia
Refinamiento del escenario	
Fuente del estímulo	Huésped
Estímulo	Solicitud de consulta de reservas.
Ambiente	Alta concurrencia (picos de demanda).
Artefacto	Módulo de reservas, base de datos transaccional.
Respuesta	El sistema retorna la lista completa de reservas pasadas y activas.
Medida de la respuesta	Latencia ≤ 500 ms para P95 bajo 300 QPS
ID y descripción corta	<b>QAS-25 Administrador configura habitaciones disponibles mediante categorización, amenidades y capacidades</b>
Escenario	El administrador gestiona la disponibilidad y características de habitaciones.
Atributo	Rendimiento
Preocupación del atributo	Latencia del catálogo.
Refinamiento del escenario	
Fuente del estímulo	Administración
Estímulo	Registro o actualización de habitaciones y sus atributos.
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de catálogo.
Respuesta	Actualización reflejada en el sistema de manera inmediata.
Medida de la respuesta	Confirmación ≤ 2.5 s (P95)
ID y descripción corta	<b>QAS-29 Administrador ajusta hasta el overbooking</b>
Escenario	El administrador define política de sobreventa hasta el 10% por tipo de habitación.

<b>ID y descripción corta</b>	<b>QAS-29 Administrador ajusta hasta el overbooking</b>
Atributo	Rendimiento
Preocupación del atributo	Velocidad en la escritura de datos
Refinamiento del escenario	
Fuente del estímulo	Administración
Estímulo	Ajuste de parámetros de overbooking.
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Motor de reglas de disponibilidad.
Respuesta	El sistema actualiza inmediatamente la política y la refleja en las reservas.
Medida de la respuesta	Latencia ≤ 2.5 s en la confirmación
<b>ID y descripción corta</b>	<b>QAS-31 Administrador consulta disponibilidad y ocupación</b>
Escenario	El administrador consulta el estado de ocupación de todas las habitaciones.
Atributo	Rendimiento
Preocupación del atributo	Eficiencia de la consulta a la base de datos y latencia de la red
Refinamiento del escenario	
Fuente del estímulo	Administración
Estímulo	Petición de reporte de ocupación/disponibilidad.
Ambiente	Operación con carga media, con una media de 2000 solicitudes simultáneas
Artefacto	Motor de consultas de disponibilidad.
Respuesta	El sistema retorna resultados precisos en tiempo real.
Medida de la respuesta	Latencia ≤ 500 ms (P95) bajo 300 QPS
<b>ID y descripción corta</b>	<b>QAS-38 Genera un reporte para verificar que los pagos en línea se hayan procesado correctamente.</b>
Escenario	El auditor genera un reporte masivo sobre el estado de pagos.
Atributo	Rendimiento
Preocupación del atributo	Cuellos de botella de la base de datos
Refinamiento del escenario	
Fuente del estímulo	Auditoría

<b>ID y descripción corta</b>	<b>QAS-38 Genera un reporte para verificar que los pagos en línea se hayan procesado correctamente.</b>
Estímulo	Solicitud de reporte de conciliación de pagos.
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Subsistema de pagos y motor de reportes.
Respuesta	Generación del reporte con todos los pagos confirmados y rechazados.
Medida de la respuesta	Tiempo de respuesta $\leq 4$ s (P99)

<b>ID y descripción corta</b>	<b>QAS-42 El auditor compara el estado de las habitaciones en el sistema con reportes físicos</b>
Escenario	El auditor solicita datos de ocupación registrados en el sistema.
Atributo	Rendimiento
Preocupación del atributo	Cuellos de botella de la base de datos
Refinamiento del escenario	
Fuente del estímulo	Auditoría
Estímulo	Solicitud de consulta histórica de habitaciones.
Ambiente	Carga de auditoría con múltiples registros.
Artefacto	Base de datos de reservas.
Respuesta	Listado completo de habitaciones y su estado.
Medida de la respuesta	Consulta resuelta en $\leq 2$ s bajo 200 QPS.

### 3.2.2. Disponibilidad

<b>ID y descripción corta</b>	<b>QAS-03 Huésped consulta la disponibilidad de una habitación</b>
Escenario	El sistema debe mantener la disponibilidad del servicio de consulta de habitaciones incluso ante fallos de base de datos
Atributo	Disponibilidad
Preocupación del atributo	Falla de la base de datos
Refinamiento del escenario	
Fuente del estímulo	Huésped
Estímulo	El huésped realiza una búsqueda de habitaciones disponibles
Ambiente	Operación con carga normal, con una media de 300 consultas por segundo
Artefacto	Módulo de consulta de disponibilidad y base de datos de inventario

<b>ID y descripción corta</b>	<b>QAS-03 Huésped consulta la disponibilidad de una habitación</b>
Respuesta	El sistema responde con datos de disponibilidad, usando caché o réplicas si la base de datos principal no está disponible
Medida de la respuesta	Latencia P95 ≤ 500ms bajo carga de 300 QPS. Disponibilidad del servicio ≥ 99.9%. En caso de falla de BD principal, el sistema responde usando datos en caché con degradación máxima de 200ms adicionales

<b>ID y descripción corta</b>	<b>QAS-05 Huésped selecciona una habitación y confirma la reserva</b>
Escenario	El huésped selecciona una habitación y procede a confirmar la reserva de la misma
Atributo	Disponibilidad
Preocupación del atributo	Tolerancia a fallos

#### Refinamiento del escenario

Fuente del estímulo	Huésped
Estímulo	El huésped hace clic en "Confirmar reserva"
Ambiente	Carga de trabajo media a alta, con múltiples transacciones simultáneas, con una media de 5000 solicitudes a la vez
Artefacto	Módulo de reservaciones y módulo de inventario
Respuesta	El sistema muestra una página de confirmación con los detalles de la reserva y un mensaje claro para el huésped. El sistema debe garantizar que cada intento de cobro sea único
Medida de la respuesta	99.9% de disponibilidad del servicio, con un tiempo de procesamiento menor a 3 segundos. Solo un huésped obtiene confirmación; el otro recibe un rechazo en ≤ 1s. El inventario retenido para una reserva fallida debe liberarse automáticamente en ≤ 500 ms

<b>ID y descripción corta</b>	<b>QAS-08 Huésped modifica alguna característica de su reserva</b>
Escenario	El sistema debe gestionar cambios en una reserva, actualizando el inventario de habitaciones
Atributo	Disponibilidad
Preocupación del atributo	Tolerancia a fallos

#### Refinamiento del escenario

Fuente del estímulo	Huésped
Estímulo	El huésped navega a la sección de "Mis reservas" y selecciona la opción para modificar una reserva
Ambiente	El sistema opera con carga normal con una media de 1000 solicitudes a la vez

<b>ID y descripción corta</b>	<b>QAS-08 Huésped modifica alguna característica de su reserva</b>
Artefacto	Módulo de modificación de reservas e inventario de habitaciones
Respuesta	El sistema presenta un formulario pre-llenado que permite al huésped cambiar los campos editables y muestra un resumen de los posibles cargos adicionales. Al final, libera la habitación original y, si la nueva habitación está disponible, la asigna, actualizando el inventario
Medida de la respuesta	La actualización de disponibilidad de la habitación original debe ser efectiva en menos de 200ms después de la confirmación de modificación

<b>ID y descripción corta</b>	<b>QAS-13 Recepcionista registra la entrada del huésped</b>
Escenario	El sistema registra el check-in de un huésped y refleja el cambio en el estado de la habitación
Atributo	Disponibilidad
Preocupación del atributo	Tolerancia a fallos, sincronización de datos e integridad de la base de datos
Refinamiento del escenario	
Fuente del estímulo	Recepcionista
Estímulo	El recepcionista ejecuta la función de check-in para la reserva de un huésped
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de check-in y módulo de gestión de habitaciones
Respuesta	El sistema cambia el estado de la habitación de "reservada" a "ocupada"
Medida de la respuesta	El estado de la habitación debe actualizarse en el sistema en menos de 200ms después de que el recepcionista complete el check-in. La información de ocupación debe ser visible de inmediato para otros recepcionistas

<b>ID y descripción corta</b>	<b>QAS-15 Recepcionista registra la salida del huésped</b>
Escenario	El sistema debe marcar una habitación como desocupada y disponible para la siguiente reserva después de que un huésped se marche
Atributo	Disponibilidad
Preocupación del atributo	Tolerancia a fallos
Refinamiento del escenario	
Fuente del estímulo	Recepcionista
Estímulo	El recepcionista ejecuta la función de check-out para una reserva
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas

<b>ID y descripción corta</b>	<b>QAS-15 Recepcionista registra la salida del huésped</b>
Artefacto	Módulo de check-out y módulo de gestión de habitaciones
Respuesta	La habitación ocupada se marca como "limpieza necesaria" o "disponible" de inmediato, lo que la hace visible para futuras asignaciones
Medida de la respuesta	El estado de la habitación debe actualizarse en el sistema en menos de 200 ms después de la confirmación del check-out. El 100% de las habitaciones deben reflejar su estado correcto

<b>ID y descripción corta</b>	<b>QAS-18 Recepcionista realiza el cobro al huésped terminada su estancia</b>
Escenario	El sistema debe permitir el cobro final, incluyendo consumos adicionales, manteniendo la disponibilidad del servicio incluso ante fallos en la pasarela de pago
Atributo	Disponibilidad
Preocupación del atributo	Fallo en el servicio de un sistema externo

#### Refinamiento del escenario

Fuente del estímulo	Recepcionista
Estímulo	El recepcionista inicia y procesa un pago final para una reserva
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de pagos y facturación
Respuesta	El sistema registra el intento de pago y emite la factura. Si la pasarela de pago falla, el sistema reintenta automáticamente o permite completar el proceso manualmente. La disponibilidad de la habitación se actualiza solo después del check-out exitoso
Medida de la respuesta	El sistema debe procesar el cobro en menos de 500ms. En caso de fallo de la pasarela, el sistema reintenta hasta 3 veces con backoff exponencial. La disponibilidad del módulo de pagos debe ser $\geq 99.5\%$

<b>ID y descripción corta</b>	<b>QAS-19 Recepcionista registra consumos adicionales del huésped</b>
Escenario	El sistema debe permitir al recepcionista añadir cargos a la cuenta de un huésped sin afectar la disponibilidad de su habitación
Atributo	Disponibilidad
Preocupación del atributo	Tolerancia a fallos ante concurrencia

#### Refinamiento del escenario

Fuente del estímulo	Recepcionista
Estímulo	El recepcionista selecciona la reserva y añade un nuevo consumo

ID y descripción corta	<b>QAS-19 Recepcionista registra consumos adicionales del huésped</b>
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de consumos
Respuesta	El sistema registra el consumo en la cuenta del huésped sin bloquear otras operaciones sobre la misma reserva
Medida de la respuesta	El registro del consumo debe ser exitoso en menos de 200ms. El estado de la habitación debe permanecer como "ocupado". El sistema debe manejar múltiples registros de consumo concurrentes sin pérdida de datos

ID y descripción corta	<b>QAS-21 Recepcionista cambia al huésped de habitación</b>
Escenario	El sistema debe gestionar el cambio de habitación, liberando la original y asignando una nueva
Atributo	Disponibilidad
Preocupación del atributo	Falla de la base de datos y tolerancia a fallos
Refinamiento del escenario	
Fuente del estímulo	Recepcionista
Estímulo	El recepcionista ejecuta la función de cambio de habitación, seleccionando una nueva habitación disponible
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de gestión de habitaciones y reservas
Respuesta	El sistema bloquea ambas habitaciones, realiza la reasignación, actualiza la reserva con la nueva habitación y libera la habitación original
Medida de la respuesta	La transacción completa debe realizarse en menos de 500ms. La habitación original no debe estar disponible para nuevas reservas hasta que la nueva asignación esté confirmada

ID y descripción corta	<b>QAS-22 Recepcionista marca una habitación como fuera de servicio</b>
Escenario	El sistema debe excluir una habitación del inventario disponible para reservas
Atributo	Disponibilidad
Preocupación del atributo	Falla de la base de datos
Refinamiento del escenario	
Fuente del estímulo	Recepcionista
Estímulo	El recepcionista cambia el estado de una habitación a "fuera de servicio"

<b>ID y descripción corta</b>	<b>QAS-22 Recepcionista marca una habitación como fuera de servicio</b>
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de gestión de habitaciones
Respuesta	La habitación es removida de todas las listas de disponibilidad
Medida de la respuesta	Actualización del estado en menos de 100ms

<b>ID y descripción corta</b>	<b>QAS-23 Recepcionista cancela la reserva del huésped</b>
Escenario	El sistema debe liberar la habitación de una reserva cancelada, haciéndola disponible nuevamente
Atributo	Disponibilidad
Preocupación del atributo	Falla de la base de datos y tolerancia a fallos
Refinamiento del escenario	
Fuente del estímulo	Recepcionista
Estímulo	El recepcionista ejecuta la función de cancelación de reserva
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de cancelación de reservas y módulo de inventario
Respuesta	El sistema cancela la reserva y la habitación es regresada al inventario disponible
Medida de la respuesta	La habitación debe estar disponible para nuevas reservas en menos de 500 ms

<b>ID y descripción corta</b>	<b>QAS-24 Administrador realiza altas y modificaciones de hoteles</b>
Escenario	El sistema debe permitir al administrador agregar nuevos hoteles o modificar la información de los existentes, y reflejar el cambio en la disponibilidad general
Atributo	Disponibilidad
Preocupación del atributo	Falla de la base de datos
Refinamiento del escenario	
Fuente del estímulo	Administrador
Estímulo	El administrador envía una solicitud para dar de alta o modificar la información de un hotel
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de administración de hoteles

<b>ID y descripción corta</b>	<b>QAS-24 Administrador realiza altas y modificaciones de hoteles</b>
Respuesta	El sistema registra el nuevo hotel o actualiza los datos del existente, y su inventario de habitaciones se hace visible y disponible para su gestión
Medida de la respuesta	La información del hotel debe ser persistida en menos de 2 segundos. La disponibilidad de las habitaciones debe reflejarse correctamente en el sistema de forma inmediata

<b>ID y descripción corta</b>	<b>QAS-26 Administrador configura habitaciones disponibles mediante categorización, amenidades y capacidades</b>
Escenario	El sistema debe permitir al administrador configurar los tipos de habitaciones, sus amenidades y sus capacidades, y hacerlas visibles para la reserva
Atributo	Disponibilidad
Preocupación del atributo	Tolerancia a fallos ante actualizaciones concurrentes

#### Refinamiento del escenario

Fuente del estímulo	Administrador
Estímulo	El administrador envía una solicitud para configurar las características de un tipo de habitación
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de administración de habitaciones
Respuesta	El sistema actualiza la base de datos con la nueva configuración, haciendo que la disponibilidad de esas habitaciones se muestre con las características correctas en la interfaz del cliente
Medida de la respuesta	Los cambios deben reflejarse en las interfaces de reserva en menos de 100 ms. La información de capacidad y amenidades debe ser 100% precisa

<b>ID y descripción corta</b>	<b>QAS-27 Administrador define precios diarios, temporadas altas y promociones</b>
Escenario	El sistema debe permitir al administrador definir precios por día, temporadas altas y promociones que afecten la disponibilidad
Atributo	Disponibilidad
Preocupación del atributo	Tolerancia a fallos ante actualizaciones concurrentes
Refinamiento del escenario	
Fuente del estímulo	Administrador
Estímulo	El administrador introduce las nuevas reglas de precios y promociones

<b>ID y descripción corta</b>	<b>QAS-27 Administrador define precios diarios, temporadas altas y promociones</b>
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de gestión de tarifas
Respuesta	El sistema aplica las nuevas reglas de precios y promociones a las habitaciones disponibles, haciendo que se muestren en el sistema de reservas
Medida de la respuesta	La actualización de precios y promociones debe ser efectiva en menos de 200 ms. El sistema debe aplicar las reglas de precios correctamente en el 100% de los casos

<b>ID y descripción corta</b>	<b>QAS-30 Administrador ajusta hasta el overbooking</b>
Escenario	El sistema debe permitir al administrador configurar los límites de overbooking de habitaciones para maximizar la ocupación
Atributo	Disponibilidad
Preocupación del atributo	Fallo en la base de datos

#### Refinamiento del escenario

Fuente del estímulo	Administrador
Estímulo	El administrador modifica la política de overbooking para un tipo de habitación
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de gestión de overbooking
Respuesta	El sistema permite que se realicen reservas por encima del número de habitaciones físicas disponibles, hasta el límite definido
Medida de la respuesta	La política de overbooking debe ser activada en el sistema en menos de 100 ms después de su configuración. El sistema no debe permitir que las reservas superen el límite de overbooking establecido

### 3.2.3. Usabilidad

<b>ID y descripción corta</b>	<b>QAS-01 Huésped consulta hoteles</b>
Escenario	Un huésped consulta los hoteles de la cadena de forma intuitiva
Atributo	Usabilidad
Preocupación del atributo	Facilidad de uso y eficiencia.
Refinamiento del escenario	
Fuente del estímulo	Huésped

<b>ID y descripción corta</b>	<b>QAS-01 Huésped consulta hoteles</b>
Estímulo	El huésped ingresa criterios de búsqueda y hace clic en "Buscar"
Ambiente	El usuario se encuentra en un entorno típico con una conexión a internet estable
Artefacto	Interfaz de búsqueda y sus respectivos resultados
Respuesta	El sistema muestra una lista clara y organizada de hoteles que cumplen con los criterios de búsqueda, incluyendo fotos, precios y calificaciones
Medida de la respuesta	El tiempo de carga de la página no debe exceder los 2 segundos. Al menos el 95% de los usuarios de prueba deben poder completar la búsqueda sin asistencia

<b>ID y descripción corta</b>	<b>QAS-02 Huésped consulta la disponibilidad de una habitación</b>
Escenario	El huésped busca una habitación sin fricción y puede completar la consulta de disponibilidad con éxito.
Atributo	Usabilidad
Preocupación del atributo	Complejidad de la interfaz

#### Refinamiento del escenario

Fuente del estímulo	Huésped
Estímulo	El huésped realiza una búsqueda en el sitio web del hotel
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Página de resultados del motor de búsqueda
Respuesta	El sistema muestra en una lista los hoteles que en aquel momento estén disponibles
Medida de la respuesta	Un usuario puede completar la consulta de disponibilidad en menos de 15 segundos

<b>ID y descripción corta</b>	<b>QAS-04 Huésped selecciona una habitación y confirma la reserva</b>
Escenario	El proceso de reserva debe ser simple, directo y sin ambigüedades
Atributo	Usabilidad
Preocupación del atributo	Intuitividad y minimización de errores.

#### Refinamiento del escenario

Fuente del estímulo	Huésped
Estímulo	El huésped hace clic en "Confirmar reserva"
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas

<b>ID y descripción corta</b>	<b>QAS-04 Huésped selecciona una habitación y confirma la reserva</b>
Artefacto	Formulario de reserva y pasarela de pago
Respuesta	El sistema muestra una página de confirmación con los detalles de la reserva y envía un correo electrónico al huésped
Medida de la respuesta	El proceso de reserva debe completarse en menos de 5 pasos. La tasa de éxito de la reserva debe ser superior al 98%

<b>ID y descripción corta</b>	<b>QAS-07 Huésped modifica alguna característica de su reserva</b>
Escenario	La modificación de una reserva debe ser tan fácil como la original
Atributo	Usabilidad
Preocupación del atributo	Simplicidad de la interfaz de modificación
Refinamiento del escenario	
Fuente del estímulo	Huésped
Estímulo	El huésped navega a la sección de "Mis reservas" y selecciona la opción para modificar una reserva específica
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Interfaz de gestión de reservas
Respuesta	El sistema presenta un formulario pre-llenado que permite al huésped cambiar los campos editables, como las fechas, y muestra un resumen de los posibles cargos adicionales
Medida de la respuesta	El tiempo medio para completar la modificación debe ser menor a 1 minuto. El 100% de las modificaciones válidas deben ser procesadas correctamente

<b>ID y descripción corta</b>	<b>QAS-09 Huésped cancela su reserva</b>
Escenario	El proceso de cancelación debe ser sencillo y transparente para el huésped
Atributo	Usabilidad
Preocupación del atributo	Intuitividad y claridad en el proceso de cancelación
Refinamiento del escenario	
Fuente del estímulo	Huésped
Estímulo	El huésped hace clic en el botón de "Cancelar reserva"
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Interfaz de gestión de reservas

<b>ID y descripción corta</b>	<b>QAS-09 Huésped cancela su reserva</b>
Respuesta	El sistema muestra una ventana emergente que pide al huésped confirmar la cancelación y le informa sobre la política de cancelación y los posibles reembolsos
Medida de la respuesta	El proceso de cancelación debe completarse en un máximo de 3 clics. No debe haber ambigüedad en la confirmación de la cancelación

<b>ID y descripción corta</b>	<b>QAS-10 Huésped revisa todas las reservas pasadas y activas realizadas por él</b>
Escenario	El huésped puede ver un historial completo y organizado de sus reservas pasadas y activas
Atributo	Usabilidad
Preocupación del atributo	Claridad en la visualización de datos y navegabilidad
Refinamiento del escenario	
Fuente del estímulo	Huésped
Estímulo	El huésped navega a la sección de su historial de reservas
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Interfaz de perfil de usuario
Respuesta	El sistema muestra una lista de reservas, categorizadas como "activas" y "pasadas", con detalles clave como fechas, hotel y estado
Medida de la respuesta	El tiempo de carga de la página del historial de reservas debe ser menor a 3 segundos. El 100% de las reservas deben ser recuperadas y mostradas sin errores

<b>ID y descripción corta</b>	<b>QAS-12 Recepcionista registra la entrada del huésped</b>
Escenario	El proceso de check-in debe ser rápido y sencillo para el recepcionista
Atributo	Usabilidad
Preocupación del atributo	Eficiencia operativa y curva de aprendizaje mínima para el personal
Refinamiento del escenario	
Fuente del estímulo	Recepción
Estímulo	El recepcionista busca la reserva del huésped y hace clic en "Check-in"
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de check-in y gestión de reservas

<b>ID y descripción corta</b>	<b>QAS-12 Recepcionista registra la entrada del huésped</b>
Respuesta	El sistema guía al recepcionista a través de los pasos necesarios, como la verificación de identidad, la asignación de habitación y la activación de la llave, con mensajes claros
Medida de la respuesta	El tiempo promedio para un check-in debe ser menor a 2 minutos. El 99% de los check-ins deben completarse sin errores

<b>ID y descripción corta</b>	<b>QAS-14 Recepcionista registra la salida del huésped</b>
Escenario	El proceso de check-out debe ser rápido y sin errores para el recepcionista
Atributo	Usabilidad
Preocupación del atributo	Eficiencia y precisión en el proceso de salida

#### Refinamiento del escenario

Fuente del estímulo	Recepción
Estímulo	El recepcionista busca la reserva del huésped y hace clic en "Check-out"
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Módulo de check-out y procesamiento de pagos
Respuesta	El sistema muestra un resumen de la cuenta, procesa el pago y marca la habitación como desocupada, con una confirmación clara para el recepcionista
Medida de la respuesta	El tiempo promedio para un check-out debe ser menor a 1 minuto. La conciliación de pagos no debe presentar errores

<b>ID y descripción corta</b>	<b>QAS-16 Recepcionista realiza el cobro al huésped terminada su estancia</b>
Escenario	El sistema de cobro debe de ser eficiente para la recepcionista, minimizando errores y agilizando el proceso de check-out
Atributo	Usabilidad
Preocupación del atributo	Curva de aprendizaje mínima

#### Refinamiento del escenario

Fuente del estímulo	Recepcionista
Estímulo	Recepcionista lleva a cabo los pasos para procesar el pago final del huésped
Ambiente	Operación normal, potencialmente con una fila de huéspedes que necesitan hacer su pago final rápidamente
Artefacto	Interfaz de check-out del sistema

<b>ID y descripción corta</b>	<b>QAS-16 Recepcionista realiza el cobro al huésped terminada su estancia</b>
Respuesta	La recepcionista completa el proceso de cobro de forma rápida y sin errores. La interfaz monto a pagar, detalles del huésped y formas de pago
Medida de la respuesta	El cobro debe de hacerse en menos de 5 minutos, con una tasa de error inferior al 2%

<b>ID y descripción corta</b>	<b>QAS-20 Recepcionista cambia al huésped de habitación</b>
Escenario	El sistema debe permitir que un recepcionista reasigne una habitación a un huésped de forma rápida y sin errores
Atributo	Usabilidad
Preocupación del atributo	Flujo de trabajo ineficiente

#### Refinamiento del escenario

Fuente del estímulo	Recepción
Estímulo	El huésped solicita un cambio de habitación
Ambiente	Operación normal, posiblemente bajo presión de otros huéspedes
Artefacto	Interfaz del sistema
Respuesta	La recepcionista puede buscar otra habitación, reasignar al huésped y actualizar el sistema con los nuevos detalles con un mínimo de clics
Medida de la respuesta	Los clics promedio para completar de actividad deben ser máximo 5

### 3.2.4. Seguridad

<b>ID y descripción corta</b>	<b>QAS-06 Huésped selecciona una habitación y confirma la reserva</b>
Escenario	Un huésped reserva una habitación.
Atributo	Seguridad
Preocupación del atributo	Integridad de transacciones.
Refinamiento del escenario	
Fuente del estímulo	Huésped
Estímulo	Confirmación de la reserva con pago.
Ambiente	Operación con alta concurrencia, con una media de 10,000 solicitudes simultáneas
Artefacto	Módulo de reservas y pagos.
Respuesta	El sistema confirma una sola vez y asegura la transacción.
Medida de la respuesta	Incidentes de doble cobro = 0

<b>ID y descripción corta</b>	<b>QAS-17 Recepcionista realiza el cobro al huésped terminada su estancia</b>
Escenario	El recepcionista realiza un cargo de consumos al finalizar la estancia.
Atributo	Seguridad
Preocupación del atributo	Integridad de transacciones.

#### Refinamiento del escenario

Fuente del estímulo	Recepción
Estímulo	Solicitud de pago final.
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Respuesta	Registro único del cobro.
Medida de la respuesta	100% de consistencia en la transacción.

<b>ID y descripción corta</b>	<b>QAS-27 Administrador define precios diarios, temporadas altas y promociones.</b>
Escenario	El administrador cambia tarifas y promociones.
Atributo	Seguridad
Preocupación del atributo	Autenticación y autorización.

#### Refinamiento del escenario

Fuente del estímulo	Administración
Estímulo	Modificación de tarifas.
Ambiente	Sesión autenticada.
Artefacto	Módulo de catálogo de precios.
Respuesta	Solo usuarios autorizados pueden modificar precios.
Medida de la respuesta	Acceso no autorizado = 0.

<b>ID y descripción corta</b>	<b>QAS-32 Un auditor busca reservas canceladas poco después de su creación</b>
Escenario	Un auditor valida intentos sospechosos de cancelación.
Atributo	Seguridad
Preocupación del atributo	Trazabilidad de operaciones.

#### Refinamiento del escenario

<b>ID y descripción corta</b>	<b>QAS-32 Un auditor busca reservas canceladas poco después de su creación</b>
Fuente del estímulo	Auditoría
Estímulo	Consulta de cancelaciones recientes.
Ambiente	Carga de auditoría.
Artefacto	Log de operaciones.
Respuesta	Mostrar todas las cancelaciones con marca de tiempo y usuario.
Medida de la respuesta	100% de transacciones trazables.

<b>ID y descripción corta</b>	<b>QAS-33 Un auditor rastrea cambios en el precio de una reserva después de confirmada.</b>
Escenario	Auditor revisa modificaciones de precios post-confirmación.
Atributo	Seguridad
Preocupación del atributo	Integridad de precios.

#### Refinamiento del escenario

Fuente del estímulo	Auditoría
Estímulo	Solicitud de historial de cambios.
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Registro de precios.
Respuesta	Mostrar cada modificación con usuario y hora
Medida de la respuesta	100% de cambios registrados

<b>ID y descripción corta</b>	<b>QAS-35 El auditor sigue reembolsos y descuentos aplicados, asegurando que estén justificados.</b>
Escenario	Un auditor valida descuentos y devoluciones.
Atributo	Seguridad
Preocupación del atributo	Autenticidad y justificación de descuentos.

#### Refinamiento del escenario

Fuente del estímulo	Auditoría
Estímulo	Solicitud de reporte de reembolsos/descuentos.
Ambiente	Operación con carga normal, con media de 1000 solicitudes simultáneas
Artefacto	Subsistema de pagos.
Respuesta	Registro completo de cada descuento con justificación.
Medida de la respuesta	100% de descuentos con registro válido.

<b>ID y descripción corta</b>	<b>QAS-37 El auditor genera un reporte para verificar que los pagos en línea se hayan procesado correctamente.</b>
Escenario	Auditor valida pagos electrónicos.
Atributo	Seguridad
Preocupación del atributo	Disponibilidad e integridad de los registros de pago.

#### Refinamiento del escenario

Fuente del estímulo	Auditoría
Estímulo	Solicitud de reporte de conciliación.
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Pasarela de pagos.
Respuesta	Generación de reporte con estatus de cada pago.
Medida de la respuesta	100% de pagos procesados verificados

<b>ID y descripción corta</b>	<b>QAS-40 El auditor puede ver un registro de cada inicio de sesión y la actividad del personal.</b>
-------------------------------	--

Escenario	El auditor revisa accesos al sistema.
Atributo	Seguridad
Preocupación del atributo	Autenticidad y trazabilidad de usuarios.

#### Refinamiento del escenario

Fuente del estímulo	Auditoría
Estímulo	Solicitud de logins del personal.
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas
Artefacto	Sistema de autenticación.
Respuesta	Lista completa de sesiones e interacciones.
Medida de la respuesta	100% de accesos registrados.

### 3.2.5. Confiabilidad

<b>ID y descripción corta</b>	<b>QA34 - El auditor compara ingresos de servicios adicionales con registros de venta.</b>
Escenario	Conciliar ingresos por servicios adicionales
Atributo	Confiabilidad
Preocupación del atributo	Consistencia de datos.

<b>ID y descripción corta</b>	<b>QA34 - El auditor compara ingresos de servicios adicionales con registros de venta.</b>
Refinamiento del escenario	
Fuente del estímulo	Auditoría
Estímulo	Solicitud de conciliación de registros de venta.
Ambiente	Operación con carga normal.
Artefacto	Subsistema de contabilidad
Respuesta	El sistema compara automáticamente los datos de ingresos con los registros de venta y genera un informe de discrepancias.
Medida de la respuesta	Menos del 0.5% de discrepancia entre los registros.

<b>ID y descripción corta</b>	<b>QA36 - El auditor sigue reembolsos y descuentos aplicados, asegurando que estén justificados.</b>
Escenario	Auditor revisa reembolsos y descuentos.
Atributo	Confiabilidad
Preocupación del atributo	Precisión de los resultados.
Refinamiento del escenario	
Fuente del estímulo	Auditoría
Estímulo	Solicitud de reporte de reembolsos/descuentos.
Ambiente	Operación con carga normal, con media de 1000 solicitudes simultáneas.
Artefacto	Subsistema de pagos.
Respuesta	El sistema genera un reporte detallado que asocia cada descuento o reembolso con su justificación documentada.
Medida de la respuesta	100% de los descuentos y reembolsos en el reporte tienen un registro válido de justificación.

<b>ID y descripción corta</b>	<b>QA39 - Genera un reporte para verificar que los pagos en línea se hayan procesado correctamente.</b>
Escenario	Auditor valida pagos electrónicos.
Atributo	Confiabilidad
Preocupación del atributo	Veracidad de los datos.
Refinamiento del escenario	
Fuente del estímulo	Auditoría
Estímulo	Solicitud de reporte de conciliación de pagos en línea.
Ambiente	Operación con carga normal, con una media de 1000 solicitudes simultáneas.

<b>ID y descripción corta</b>	<b>QA39 - Genera un reporte para verificar que los pagos en línea se hayan procesado correctamente.</b>
Artefacto	Pasarela de pagos.
Respuesta	El sistema concilia los pagos registrados en la base de datos interna con los de la pasarela de pagos
Medida de la respuesta	100% de los pagos procesados son verificados y coinciden con la pasarela de pagos.

<b>ID y descripción corta</b>	<b>QA41 - El auditor compara el estado de las habitaciones en el sistema con reportes físicos</b>
Escenario	Auditor verifica el inventario.
Atributo	Confiabilidad
Preocupación del atributo	Consistencia de datos.

#### Refinamiento del escenario

Fuente del estímulo	Auditoría
Estímulo	Solicitud de un informe de estado de habitaciones.
Ambiente	Operación con carga baja.
Artefacto	Subsistema de gestión de habitaciones.
Respuesta	El sistema genera un informe detallado que puede ser comparado con el conteo físico, mostrando el estado de cada habitación (ocupada, limpia, fuera de servicio)
Medida de la respuesta	100% de los registros en el sistema concuerdan con los reportes físicos.

<b>ID y descripción corta</b>	<b>QA43 - El auditor compara tarifas y políticas entre hoteles de diferentes regiones</b>
Escenario	Auditor valida uniformidad de precios
Atributo	Confiabilidad
Preocupación del atributo	Veracidad de los datos.

#### Refinamiento del escenario

Fuente del estímulo	Auditoría
Estímulo	Solicitud de un informe de precios y políticas por región.
Ambiente	Operación con carga baja.
Artefacto	Módulo de administración.
Respuesta	El sistema genera un reporte consolidado que lista los precios y las políticas de los hoteles, agrupados por región.

ID y descripción corta	<b>QA43 - El auditor compara tarifas y políticas entre hoteles de diferentes regiones</b>
Medida de la respuesta	El reporte refleja con precisión los precios y políticas de todos los hoteles en cada región.

## 3.3. Restricciones y preocupaciones

### 3.3.1. Restricciones (Constraints)

Dentro de las restricciones del sistema podemos encontrarnos con las siguientes:

Identificador	Descripción	Origen
CON-1	El sistema de reservación no debe almacenar, procesar y transmitir directamente datos sensibles de tarjetas de crédito. Esta responsabilidad debe ser delegada completamente por una Pasarela de Pagos externa certificada.	Legal / Negocio
CON-2	El sistema debe cumplir con las regulaciones de protección de datos personales aplicables en las geografías donde opera la cadena hotelera.	Legal / Negocio
CON-3	Para cumplir con la trazabilidad y auditoría financiera, los accesos al sistema de eventos críticos (cobros, cancelaciones) deben ser almacenados en un almacén de datos <b>append-only</b> (solo escritura, sin modificación) para garantizar el no repudio.	Técnico

### 3.3.2. Concers (preocupaciones)

Las preocupaciones representan las inquietudes técnicas, operativas y de calidad que los interesados esperan que la arquitectura resuelva. Además, podemos encontrarnos que, cada actor cuenta con preocupaciones, las cuales deben ser implementadas, como lo pueden ser:

- Huéspedes: desean rapidez, exactitud en los cobros y protección de su información personal.
- Recepcionistas: necesitan una interfaz intuitiva, rápida y tolerante a errores.
- Administradores: buscan disponibilidad continua, integridad de pagos y confiabilidad en los reportes.
- Auditores: requieren trazabilidad completa de operaciones y datos verificables.
- Equipo técnico y docente: se enfocan en la mantenibilidad, modularidad y alineación del diseño con los atributos de calidad definidos.

Con esto podemos decir que las preocupaciones principales son:

<b>Identificador</b>	<b>Descripción</b>	<b>Objetivo</b>
CRN-1	Garantizar que cada módulo pueda ser probado y desplegado a producción de forma autónoma, sin causar interrupciones en otros servicios.	Extensibilidad / Testabilidad
CRN-2	Hacer uso eficiente de recursos de cómputo y bases de datos optimizadas para mantener bajos costos operativos y de infraestructura.	Portabilidad
CRN-3	Implementar patrones de diseño que hagan de los componentes críticos altamente testeables de forma aislada.	Testabilidad
CRN-4	Estandarizar la experiencia del usuario en todos los módulos para reducir la curva de aprendizaje.	Usabilidad
CRN-5	Contar con un diseño modular, es decir, cliente-servidor y capas, que facilite el desarrollo de nuevas funcionalidades.	Extensibilidad / Mantenibilidad
CRN-6	No acumular deuda técnica que comprometa la mantenibilidad del sistema.	Mantenibilidad

# Chapter 4. Método de diseño

## 4.1. Método de diseño

El diseño arquitectónico del Sistema Hotelero se desarrolló siguiendo el método Attribute-Driven Design (ADD) complementado con el enfoque Component-Based Software Engineering (CBSE) descrito por Cheesman & Daniels. Este método garantiza que la arquitectura resultante no sea producto de decisiones arbitrarias, sino una respuesta racional, trazable y justificada a los impulsores arquitectónicos identificados (drivers), entre los cuales se incluyen los atributos de calidad priorizados —usabilidad, disponibilidad, seguridad y rendimiento—, además de las restricciones y preocupaciones del proyecto.

El propósito del método fue traducir los escenarios de calidad y casos de uso en una estructura arquitectónica modular, capaz de sostener las operaciones críticas del sistema, asegurar la mantenibilidad y facilitar la validación futura del diseño mediante vistas y diagramas coherentes.

### 4.1.1. Enfoque general del proceso ADD

El proceso ADD se basa en construir la arquitectura a partir de los atributos de calidad y no exclusivamente de las funcionalidades. El diseño se desarrolla en iteraciones donde cada ciclo busca satisfacer un conjunto de drivers arquitectónicos priorizados, refinando el modelo con nuevas decisiones estructurales o tácticas.

El procedimiento aplicado constó de las siguientes fases:

**Identificación de drivers arquitectónicos:** Se elaboró un backlog que incluía los Casos de Uso, los escenarios de atributos de calidad (QAS) y las restricciones técnicas. Los atributos priorizados fueron:

- **Usabilidad:** facilidad de uso, simplicidad de interfaz y eficiencia operativa.
- **Disponibilidad:** tolerancia a fallos y continuidad del servicio.
- **Seguridad:** autenticación, integridad de pagos y trazabilidad.
- **Rendimiento:** latencia reducida y optimización de consultas.

**Selección de los drivers para la iteración inicial:** En la primera iteación se atendieron los drivers usabilidad y disponibilidad, enfocados en la interacción con el usuario y la robustez operativa. Después, se atendieron los drivers de seguridad y rendimiento, para garantizar la operación confiable del sistema.

**Descomposición del sistema en elementos lógicos:** Se definió la arquitectura cliente-servidor en capas, separando responsabilidades de presentación, lógica de negocio y persistencia. Este estilo fue elegido por su alineación con los QAS y por ofrecer un marco modular y escalable.

**Asignación de responsabilidades a componentes:** Cada caso de uso fue analizado para identificar las operaciones de sistema requeridas, las clases participantes y las interfaces involucradas. Las responsabilidades fueron distribuidas jerárquicamente desde las interfaces de usuario hasta las entidades de datos.

**Evaluación de suficiencia (“suficiente”):** Cada iteración se cerró cuando las decisiones arquitectónicas satisfacían adecuadamente los escenarios de calidad sin introducir complejidad innecesaria. Esto permitió mantener un equilibrio entre funcionalidad y mantenibilidad.

#### 4.1.2. Iteraciones del proceso ADD

##### Iteración 1: Usabilidad

###### Drivers atendidos:

QAS de usabilidad (simplicidad de interfaz, eficiencia operativa).

Concern de capacitación mínima del usuario.

###### Decisiones tomadas:

Se aplicó el patrón MVC (Model-View-Controller) en el cliente, dividiendo la presentación (UI), el control (Controller) y la comunicación (ClientService).

Las interfaces gráficas se diseñarían con flujos directos y mensajes claros, reduciendo la carga cognitiva del usuario.

Se crearían controladores especializados por caso de uso (ReservaController, PagoController, etc.), facilitando así la localización y mantenimiento del código.

**Resultado:** La arquitectura resultante mejora la experiencia del usuario final, permitiendo que el cliente, el personal de recepción y administración operen el sistema con una curva de aprendizaje mínima. La separación entre vista y lógica también incrementó la mantenibilidad y la capacidad de evolución del software.

##### Iteración 2: Disponibilidad y confiabilidad

###### Drivers atendidos:

QAS de disponibilidad (continuidad del servicio, recuperación ante fallos).

QAS de confiabilidad (consistencia de datos, precisión de resultados).

###### Decisiones tomadas:

Se establecería un manejo controlado de excepciones en todas las capas, permitiendo la recuperación ante errores sin comprometer la integridad de los datos.

Los Managers se consolidaron como capa de negocio, controlando la secuencia de operaciones y validarían los estados de las entidades antes de realizar transacciones.

Se implementó el patrón Service Layer, donde los servicios (\*ServiceImp) encapsulan los flujos de negocio expuestos al cliente.

**Resultado:** La arquitectura asegura una disponibilidad estable mediante la separación de responsabilidades y una capa de control bien definida. La confiabilidad del sistema se ve reforzada

por la validación de datos y la trazabilidad de las operaciones críticas.

### **Iteración 3: Seguridad y rendimiento**

#### **Drivers atendidos:**

QAS de seguridad (autenticación, trazabilidad, integridad de pagos).

QAS de rendimiento (latencia y eficiencia de consultas).

#### **Decisiones tomadas:**

Se estableció el patrón cliente-servidor distribuido, separando la lógica de negocio de la interfaz gráfica.

Se definieron los servicios remotos (interfaces) (ILoginService, IReservaService, etc.) y sus implementaciones en el servidor, reduciendo la exposición de datos y mejorando el control de acceso.

Se diseñaría la clase ConexionDB bajo el patrón Singleton, asegurando una única conexión activa y mejorando así la eficiencia en la gestión de recursos.

Se adoptó el patrón DAO (Data Access Object) para encapsular la lógica de persistencia, minimizando el acoplamiento entre la lógica de negocio y la base de datos.

**Resultado:** El sistema alcanzó un nivel de seguridad controlada mediante autenticación y una capa de datos aislada, junto con rendimiento optimizado por la reutilización de conexiones y consultas específicas.

### **Iteración 4: Integración de reportes y trazabilidad**

#### **Drivers atendidos:**

QAS de seguridad (trazabilidad de operaciones, integridad de registros).

QAS de disponibilidad (acceso consolidado a información histórica).

#### **Decisiones tomadas:**

Se agregó la clase ReporteManager con dependencias hacia ReservaDAO, PagoDAO y HabitacionDAO, integrando información de distintas fuentes.

Se definió el servicio IReporteService y su implementación (ReporteServiceImp) para permitir la generación remota de reportes por parte del administrador o auditor.

Se garantizó que los reportes se basaran en datos consistentes y verificables, fortaleciendo la integridad del sistema.

**Resultado:** El sistema logra visibilidad total sobre su operación, habilitando auditorías y revisiones de desempeño sin afectar la disponibilidad ni el rendimiento del sistema principal.

#### **4.1.3. Enfoque CBSE (Cheesman & Daniels)**

El método Component-Based Software Engineering (CBSE) se aplicó en paralelo al ADD para formalizar la especificación de componentes e interfaces. El proceso siguió tres fases:

**Identificación:** A partir de los casos de uso y del modelo de conceptos de negocio, se identificaron los componentes principales del dominio: Reserva, Habitación, Pago, Usuario y Reporte. Cada uno fue representado como un componente lógico con responsabilidades bien definidas.

**Interacción:** Se desarrollaron diagramas del sistema para refinar cómo los componentes colaboran entre sí. Por ejemplo, la operación “Registrar Reserva” implica la interacción entre ReservaServiceImp, ReservaManager, ReservaDAO y HabitacionDAO. Este análisis permitió definir las interfaces provistas y requeridas entre componentes.

**Especificación:** Las interfaces IReservaService, IPagoService, etc., representan las interfaces provistas del sistema, mientras que los DAO y Managers actúan como componentes requeridos dentro de la especificación.

El resultado del CBSE fue una arquitectura orientada a componentes reutilizables y sustituibles, con fronteras bien definidas y bajo acoplamiento entre capas.

#### **4.1.4. Cierre del proceso**

El método ADD garantizó que cada iteración atendiera drivers específicos, mientras que CBSE proporcionó la estructura técnica para documentar las decisiones tomadas y formalizar las interfaces entre componentes. Los criterios de “suficiente” se aplicaron cuando los escenarios de atributos de calidad fueron satisfechos sin comprometer simplicidad o mantenibilidad.

En conjunto, ambos métodos produjeron una arquitectura robusta, trazable y alineada con los impulsores arquitectónicos, permitiendo que cada vista del documento se derive lógicamente del proceso de diseño y no de decisiones improvisadas.

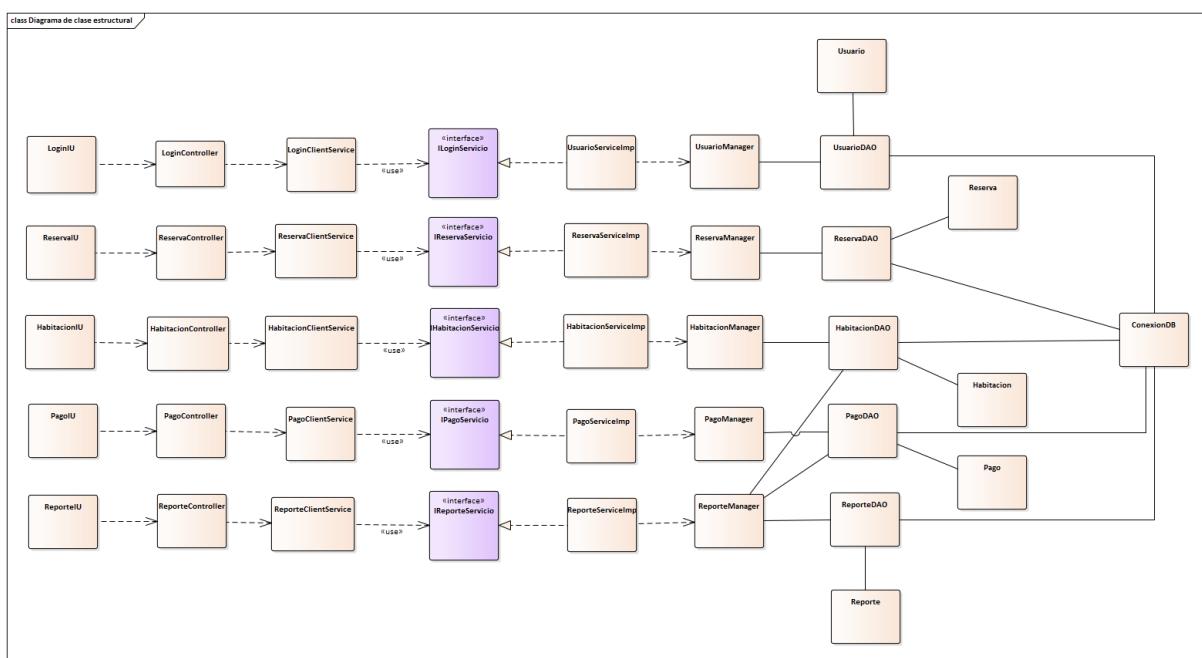
# Chapter 5. Arquitectura

## 5.1. Vista lógica

La vista lógica del Sistema Hotelero representa la organización estática del sistema y la distribución de responsabilidades funcionales a través de una arquitectura cliente-servidor en capas. Esta vista permite comprender cómo se estructura el software internamente, cómo se comunican sus componentes y cómo cada elemento contribuye a los atributos de calidad definidos en los escenarios arquitectónicos.

La estructura general está organizada en dos grandes nodos —Cliente y Servidor— cada uno dividido en subcapas que separan las responsabilidades de presentación, lógica de negocio y persistencia de datos. Este enfoque refleja el estilo arquitectónico en capas combinado con el patrón cliente-servidor, donde las dependencias fluyen de manera unidireccional, desde las interfaces gráficas hacia la base de datos, promoviendo la modularidad y la mantenibilidad.

### 5.1.1. Diagrama de Clases



#### Capa Cliente

En el extremo izquierdo del diagrama se encuentra la capa de presentación, compuesta por las clases LoginUI, ReservaUI, HabitacionUI, PagoUI y ReporteUI. Estas clases representan las ventanas o pantallas del sistema utilizadas por los diferentes actores (receptionista, administrador y auditor). Cada interfaz de usuario es responsable de mostrar información y capturar entradas, actuando como punto de contacto directo con el usuario final.

Estas clases dependen de sus respectivos controladores (LoginController, ReservaController, HabitacionController, PagoController, ReporteController), que implementan el patrón Model-View-Controller (MVC) dentro del cliente. Los controladores gestionan la lógica de interacción y validaciones básicas, delegando la comunicación con el servidor a los objetos ClientService.

Las clases LoginClientService, ReservaClientService, HabitacionClientService, PagoClientService y ReporteClientService encapsulan las llamadas remotas al servidor. Estas clases dependen de las interfaces de servicio (ILoginService, IReservaService, IHabitacionService, IPagoService, IReporteService) mediante relaciones de dependencia UML << use >>. Este diseño asegura que el cliente permanezca desacoplado de la implementación concreta del servidor, permitiendo modificar la lógica del lado servidor sin afectar el código cliente.

Desde la perspectiva de los atributos de calidad, esta separación mejora la usabilidad, ya que las interfaces gráficas pueden evolucionar independientemente de la lógica interna, y favorece la seguridad, pues las operaciones críticas se ejecutan exclusivamente del lado servidor.

## Capa Servidor

El servidor implementa las interfaces anteriores mediante clases concretas (LoginServiceImp, ReservaServiceImp, HabitacionServiceImp, PagoServiceImp y ReporteServiceImp) que realizan las interfaces definidas (<< realize >>). Estas implementaciones representan la lógica de aplicación, coordinando la ejecución de operaciones complejas y validando las reglas de negocio.

Cada servicio mantiene una dependencia hacia su correspondiente Manager, como ReservaManager, HabitacionManager, PagoManager, LoginManager y ReporteManager. Los managers encapsulan la lógica de negocio específica de cada subsistema:

- ReservaManager gestiona la asignación y modificación de reservas.
- HabitacionManager controla disponibilidad y mantenimiento.
- PagoManager administra las transacciones económicas.
- LoginManager verifica credenciales y controla accesos.
- ReporteManager integra datos de distintos DAO para generar informes de auditoría y desempeño.

Esta capa media cumple un papel esencial en la confiabilidad y consistencia del sistema, ya que asegura que todas las operaciones sigan un flujo uniforme.

## Capa de base de datos

La capa de acceso a datos está conformada por las clases ReservaDAO, HabitacionDAO, PagoDAO, ReporteDAO y LoginDAO (o UsuarioDAO), todas asociadas a la clase ConexionDB, que representa el componente de conexión a la base de datos. Cada DAO es responsable de las operaciones CRUD sobre una entidad específica (Reserva, Habitacion, Pago, Reporte, Usuario).

Las asociaciones entre DAO y POJOs (Reserva, Habitacion, Pago, Reporte, Usuario) muestran una relación de uno a muchos, donde un DAO puede gestionar múltiples instancias de su entidad correspondiente. La clase ConexionDB se modela como una asociación compartida (1 a 1) para representar una conexión única o gestionada mediante un patrón Singleton, lo cual contribuye directamente a la eficiencia y rendimiento del sistema al evitar conexiones redundantes.

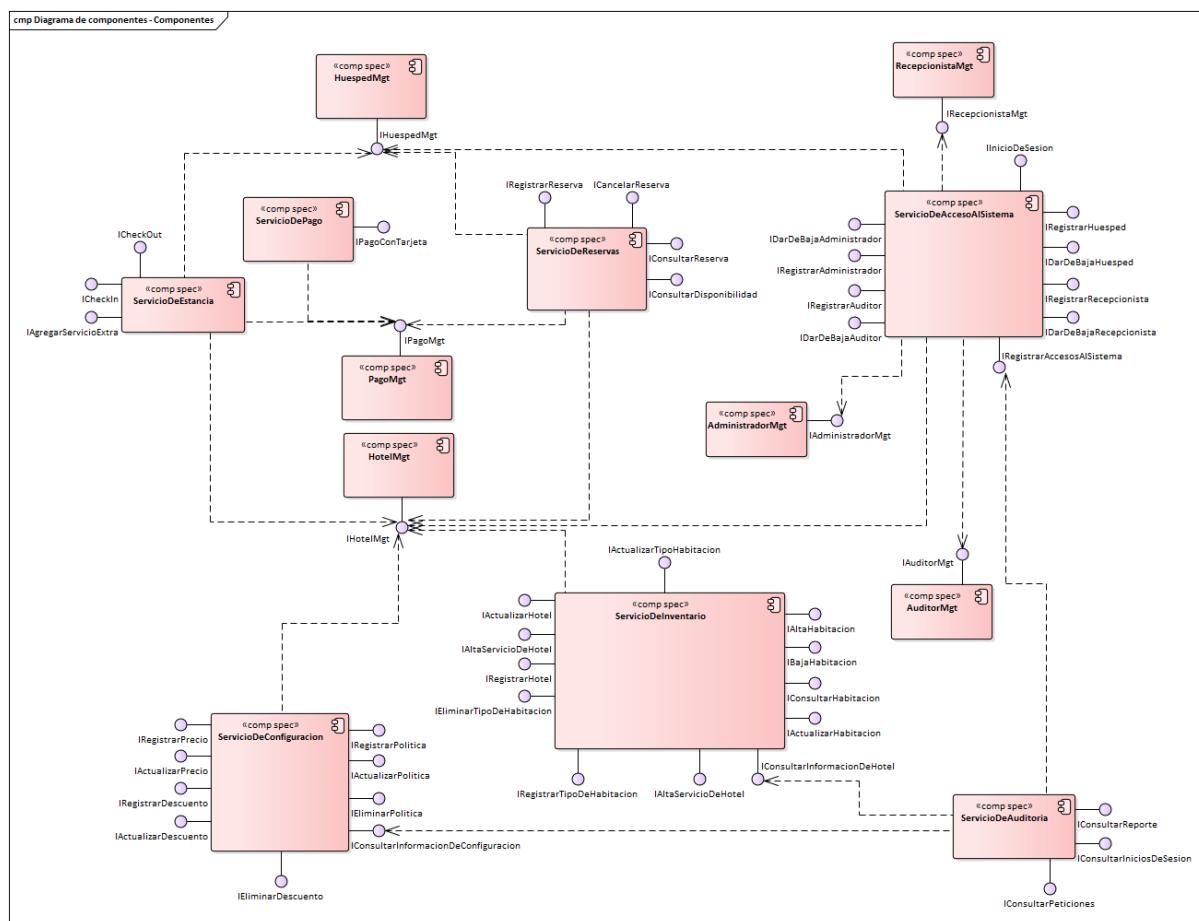
## Justificación del estilo arquitectónico

El sistema adopta el estilo arquitectónico en capas cliente-servidor, que proporciona una separación clara entre responsabilidades: presentación, lógica de negocio, servicios y persistencia.

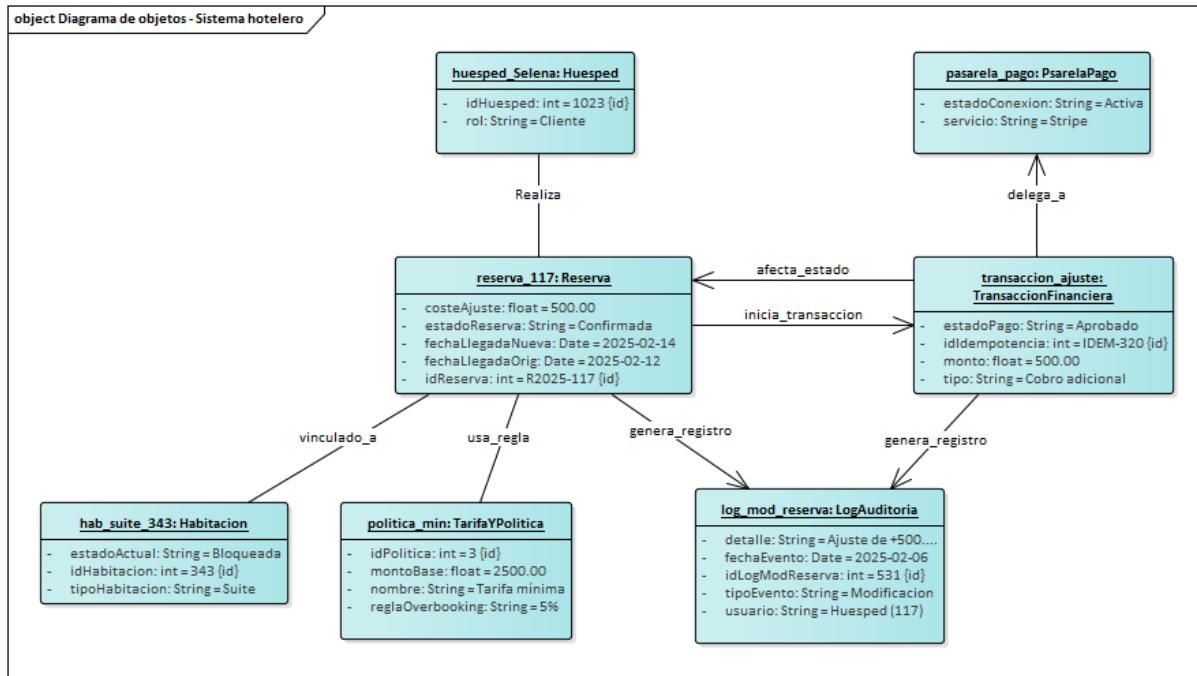
Esta estructura favorece la modularidad, la escalabilidad y la facilidad de mantenimiento, atributos esenciales en sistemas transaccionales como los de gestión hotelera.

- Usabilidad: las capas de presentación son independientes, permitiendo rediseñar interfaces sin modificar la lógica interna.
  - Disponibilidad: la separación entre cliente y servidor facilita el despliegue en diferentes máquinas, permitiendo recuperación ante fallos y operación continua.
  - Seguridad: el control de acceso y autenticación se centraliza en los servicios del servidor, minimizando riesgos en el cliente.
  - Rendimiento: la conexión única a la base de datos y la delegación de procesamiento al servidor optimizan el uso de recursos.
  - Confiabilidad: las operaciones se canalizan a través de managers que garantizan integridad y consistencia en las transacciones.

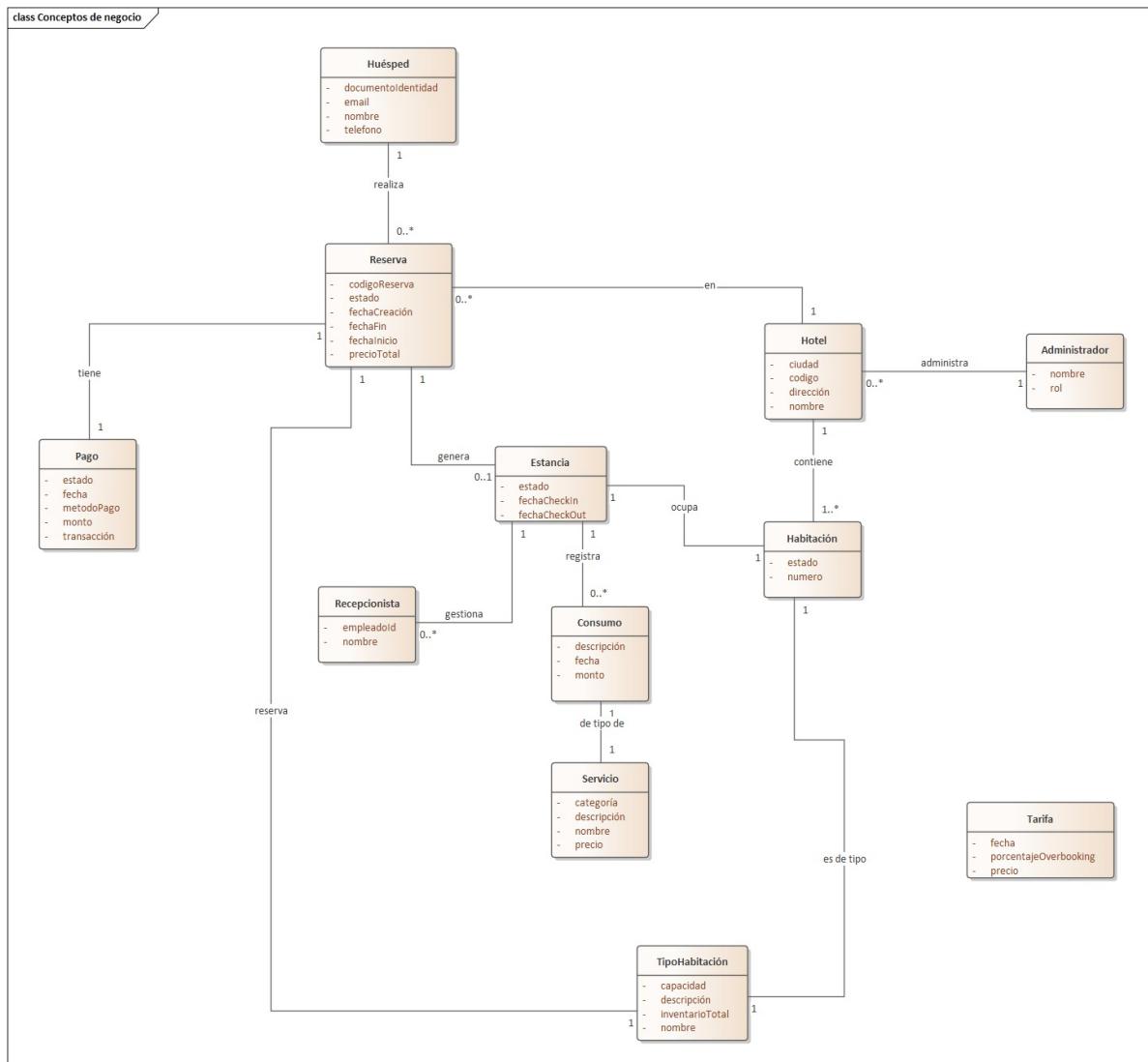
### **5.1.2. Diagrama de Componentes**



### 5.1.3. Diagrama de Objetos



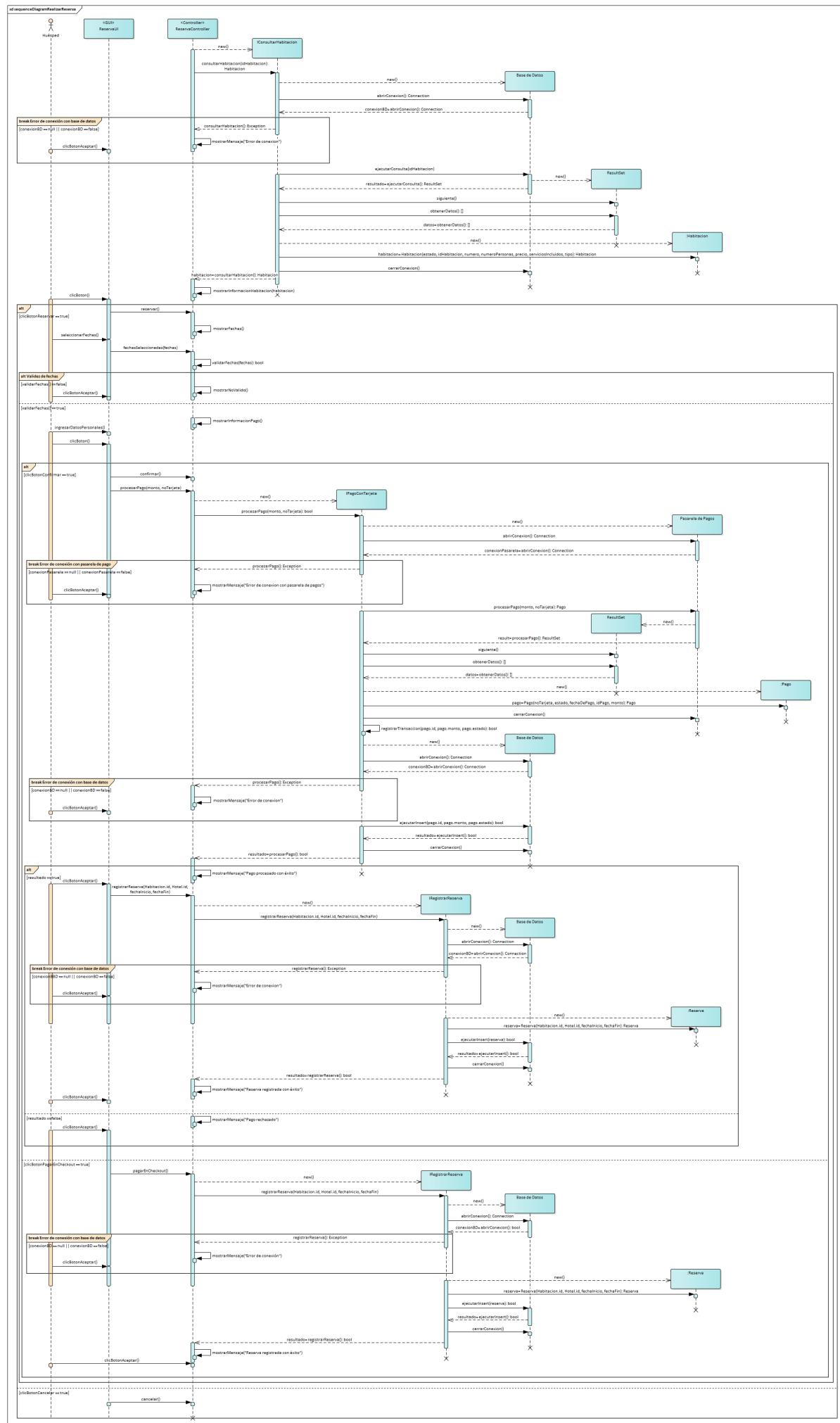
## 5.2. Modelo de conceptos de negocio



## **5.3. Vista de Proceso**

Aquí se abordan aspectos dinámicos de la arquitectura, y en cómo se sincronizan los datos críticos y cómo se distribuyen los procesos entre el cliente y el servidor para cumplir con los atributos de calidad de Disponibilidad (QAS-03), Rendimiento (QAS-05) y Seguridad/Integridad (QAS-06).

### **5.3.1. Diagrama de Secuencia de Flujo Crítico: Realizar Reserva**



## Análisis del Flujo Crítico: Realizar Reserva

El diagrama de secuencia presentado detalla el flujo de "Realizar Reserva" (CU03). Este proceso ha sido seleccionado como arquitectónicamente significativo porque es el punto donde convergen los requisitos más estrictos de integridad de datos ("tasa cero de dobles reservas") y la integración con sistemas externos (Pasarela de Pagos).

### Modelo de Interacción y Comunicación

La comunicación inicia en la capa Cliente, donde el ReservaController captura la intención del usuario. Siguiendo el patrón de desacoplamiento definido en la vista lógica, el controlador no ejecuta lógica de negocio, sino que delega la llamada a la interfaz IConsultarHabitacion. La comunicación entre la capa Cliente y la capa Servidor se realiza de forma síncrona para operaciones transaccionales críticas; el usuario debe esperar la confirmación inmediata de que su habitación ha sido asegurada.

1. Inicio de Transacción: El IConsultarHabitacion obtiene la instancia única de conexionDB (Singleton), garantizando un punto centralizado de acceso a la base de datos para optimizar recursos (QAS-11). Se inicia un bloque transaccional con propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).

### Consistencia y Manejo de Fallos

La política de consistencia aplicada es fuerte. El sistema no puede permitirse consistencia eventual en la asignación de habitaciones.

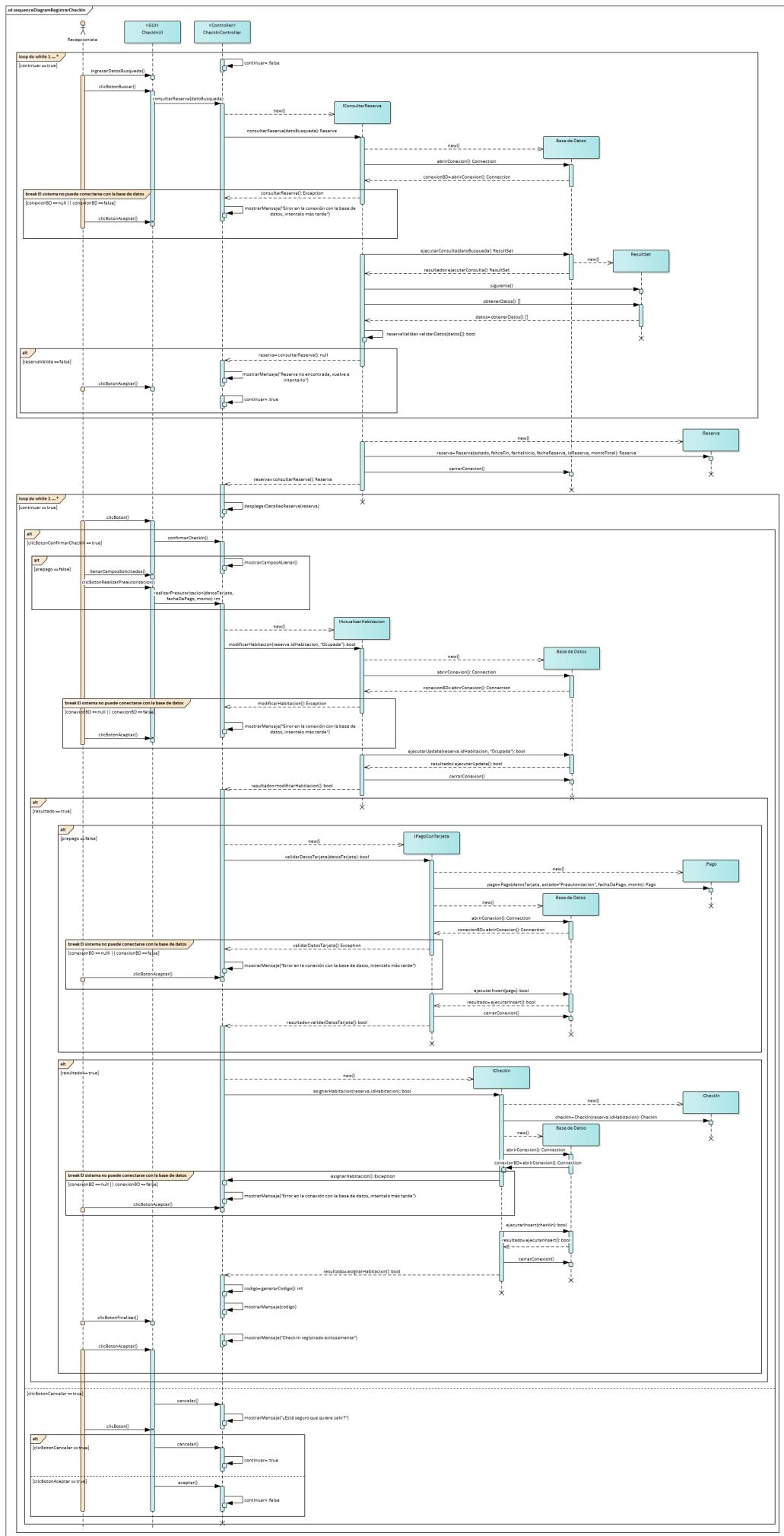
1. Integración con Pasarela de Pagos: La llamada a la PasarelaPago es una operación externa y síncrona.
2. Atomicidad (Rollback): Si la pasarela de pagos rechaza la transacción o si ocurre un error de conexión (QAS-05), se captura la excepción y ejecuta un rollback completo en la base de datos.

### Satisfacción de Atributos de Calidad

Este diseño de proceso satisface los impulsores arquitectónicos clave:

- Disponibilidad: El uso de ClientService y el manejo de excepciones en los Managers aseguran que un fallo en la pasarela de pago no tumbe el servicio, sino que retorne un mensaje de error controlado al usuario.
- Seguridad (Trazabilidad): Como parte de la transacción, el ReservaManager invoca al ReporteManager para registrar el evento en el log de auditoría inmutable (CON-3), garantizando que cada intento de reserva, exitoso o fallido, quede documentado para análisis posterior (QAS-32).

### 5.3.2. Diagrama de Secuencia de Flujo Crítico: Registrar Check-In



## Análisis del Flujo Crítico: Registrar Check-In

El diagrama de secuencia presentado detalla el flujo de "Registrar Check-In". Este proceso ha sido seleccionado como arquitectónicamente significativo porque representa la materialización del servicio, donde convergen la actualización de estado del inventario (Habitaciones), la validación financiera y la persistencia de la estancia del cliente.

### Modelo de Interacción y Comunicación

La comunicación inicia en la capa de Presentación (GUI), donde el CheckInController captura la interacción del Recepcionista. Siguiendo el patrón de desacoplamiento, el controlador orquesta el flujo delegando responsabilidades a interfaces especializadas (IConsultarReserva, IActualizarHabitacion, IPagoConTarjeta, ICheckIn) en lugar de ejecutar la lógica directamente.

La comunicación entre los componentes es estrictamente síncrona, ya que el proceso ocurre en tiempo real frente al cliente (Huésped), requiriendo confirmación inmediata de cada paso (búsqueda, pago, asignación).

### Consistencia y Manejo de Fallos

El diseño prioriza la robustez y la integridad operativa a través de mecanismos explícitos de control de errores:

1. Gestión de Conexiones y Excepciones: Se observa un patrón repetitivo de manejo de fallos mediante bloques break y try-catch implícitos. Si ocurre una Exception al intentar conectar con la base de datos (ya sea al consultar la reserva, actualizar la habitación o registrar el pago), el sistema captura el error, muestra un mensaje amigable ("Error en la conexión... intentarlo más tarde") y detiene el flujo crítico para evitar estados inconsistentes.
2. Lógica Condicional de Negocio: El flujo maneja la variabilidad del negocio mediante bloques alt. Por ejemplo, el sistema evalúa si la reserva tiene prepago == false. Si es así, fuerza la validación de datos de tarjeta y la ejecución del cobro a través de IPagoConTarjeta antes de permitir la finalización del Check-In.
3. Atomicidad Lógica: El cambio de estado de la habitación a "Ocupada" (IActualizarHabitacion) y la inserción del registro de check-in (ICheckIn) ocurren secuencialmente. El diseño asegura que no se genere un código de check-in si los pasos previos (pago o actualización de habitación) fallan.

### Satisfacción de Atributos de Calidad

Este diseño de proceso satisface los impulsores arquitectónicos clave:

- Disponibilidad y Recuperabilidad: El uso de bucles (loop) y bloques de ruptura (break) ante fallos de conexión demuestra una táctica de defensa para mantener la interfaz receptiva incluso cuando la capa de datos presenta intermitencias, evitando el colapso total de la aplicación.
- Usabilidad: El sistema provee retroalimentación constante al Recepcionista en cada punto de decisión (ej. "Reserva no encontrada", "Error de conexión", "Check-in registrado exitosamente"), facilitando la operación bajo presión.
- Modificabilidad: La separación de preocupaciones en interfaces distintas (IConsultarReserva contra IPagoConTarjeta) permite cambiar la lógica de pagos o de consultas sin afectar el flujo

principal del controlador.

### 5.3.3. Diagrama de Secuencia de Flujo Crítico:

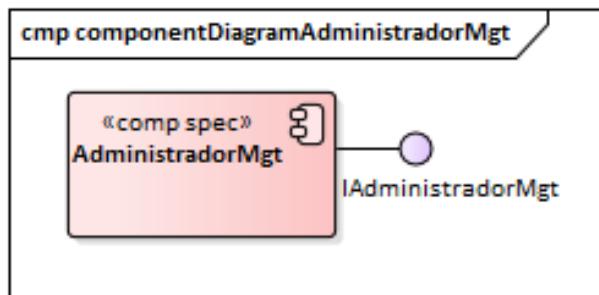
[Diagrama de Secuencia UML] | sequenceDiagram.png

Análisis del Flujo Crítico:

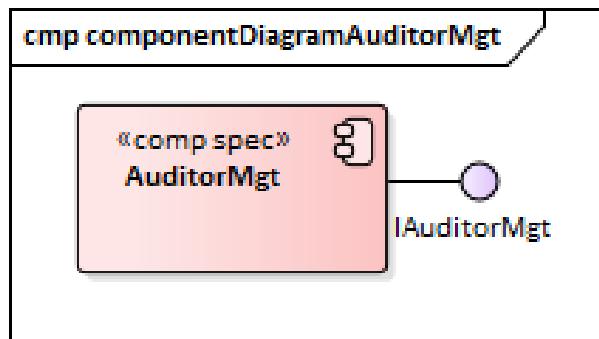
## 5.4. Vista de Implementación

### 5.4.1. Diagramas de especificación de componentes

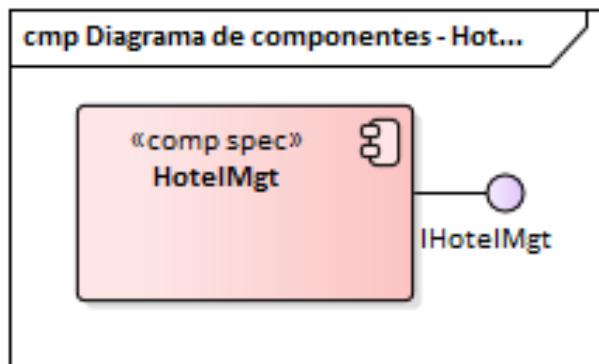
AdministradorMgt



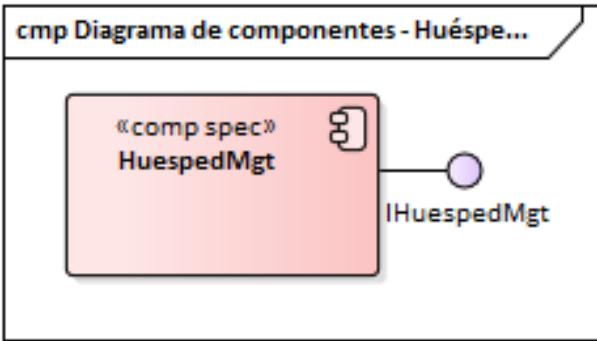
AuditorMgt



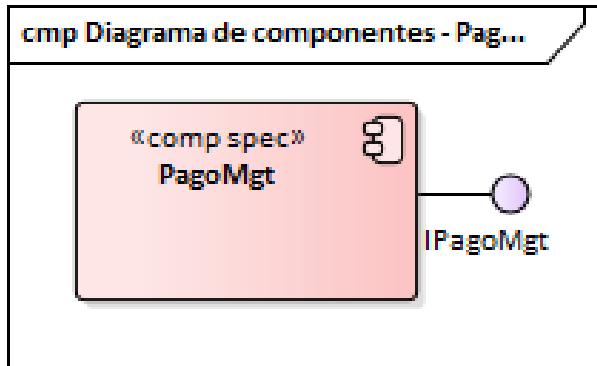
HotelMgt



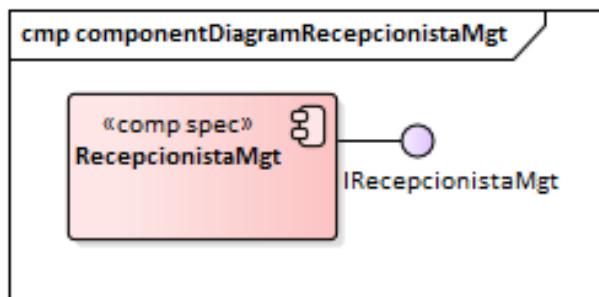
HuéspedMgt



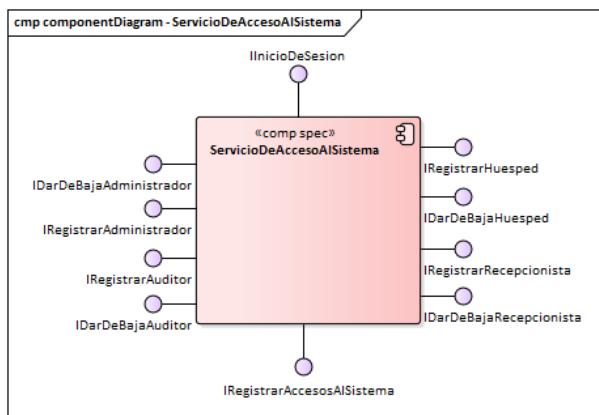
PagoMgt



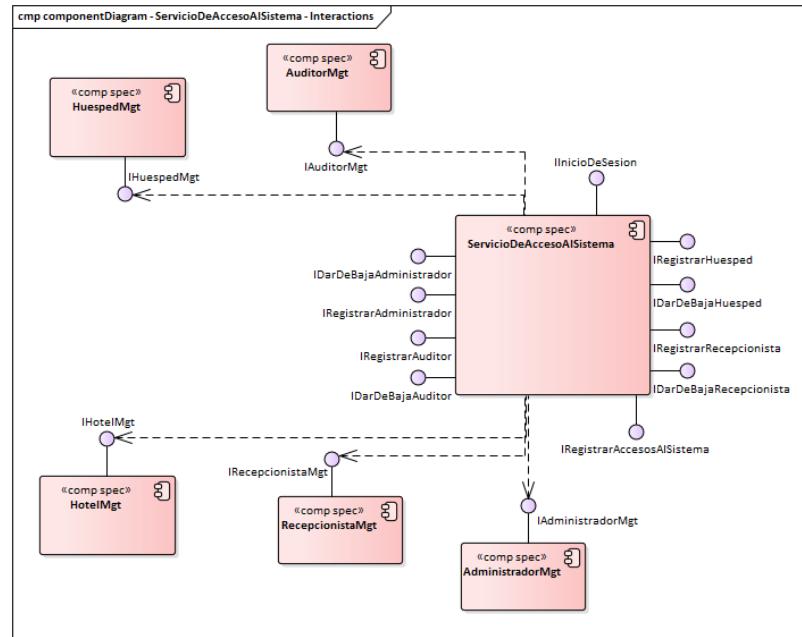
RepcionistaMgt



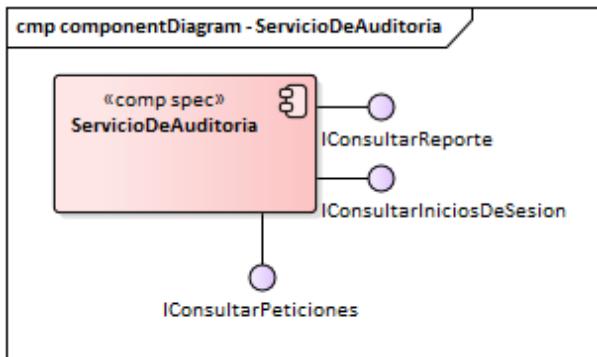
Servicio de acceso al sistema



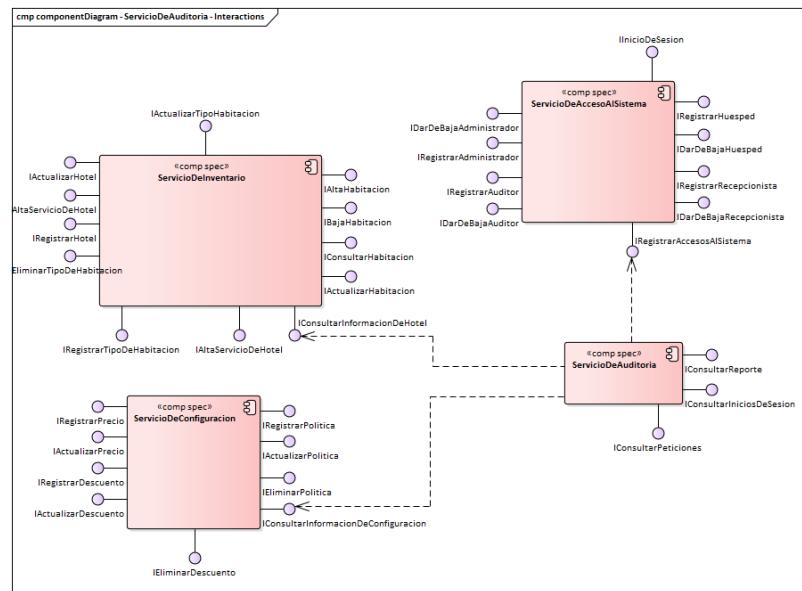
Servicio de acceso al sistema (interacciones)



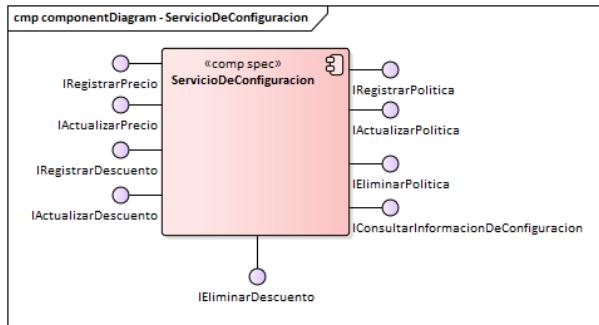
## Servicio de auditoría



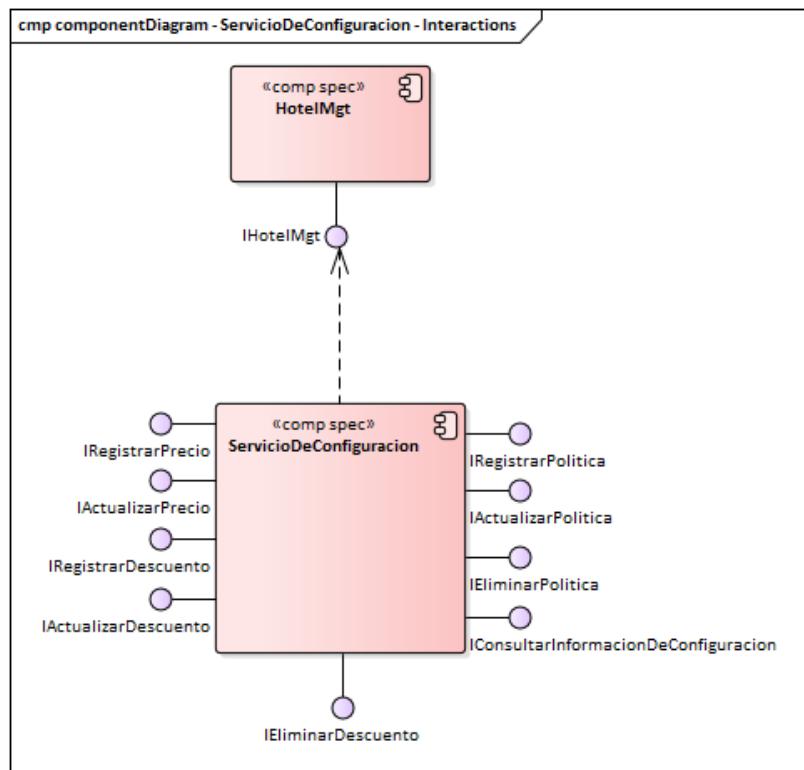
## Servicio de auditoría (interacciones)



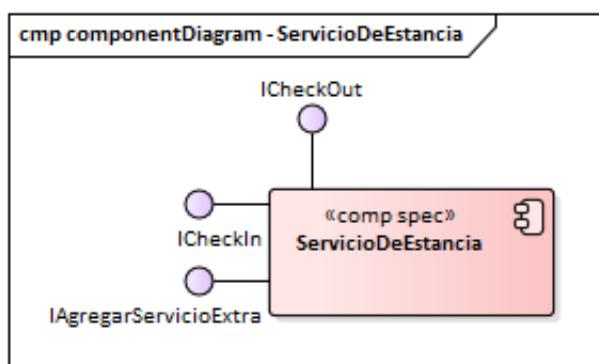
## Servicio de configuración



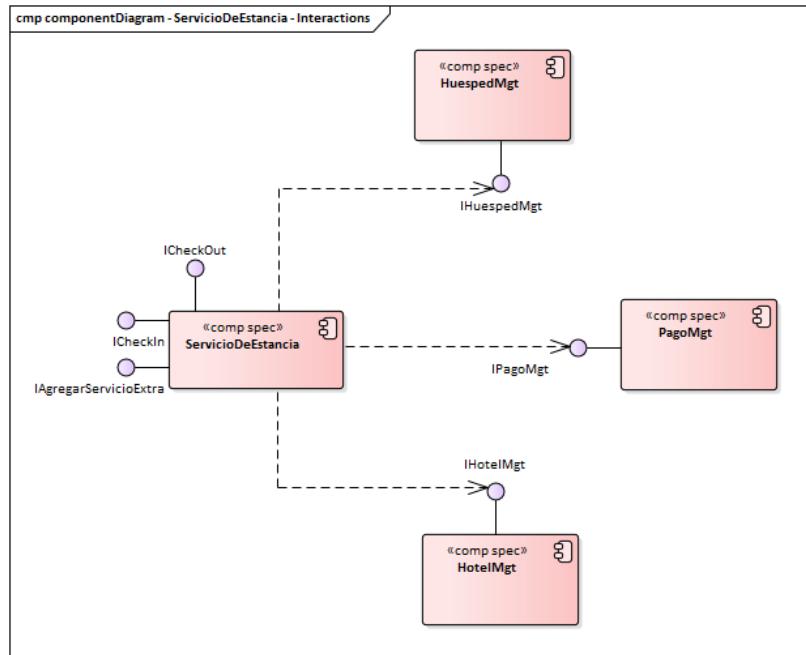
## Servicio de configuración (interacciones)



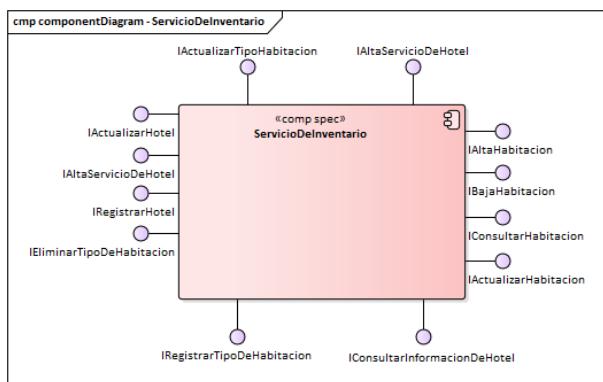
## Servicio de estancia



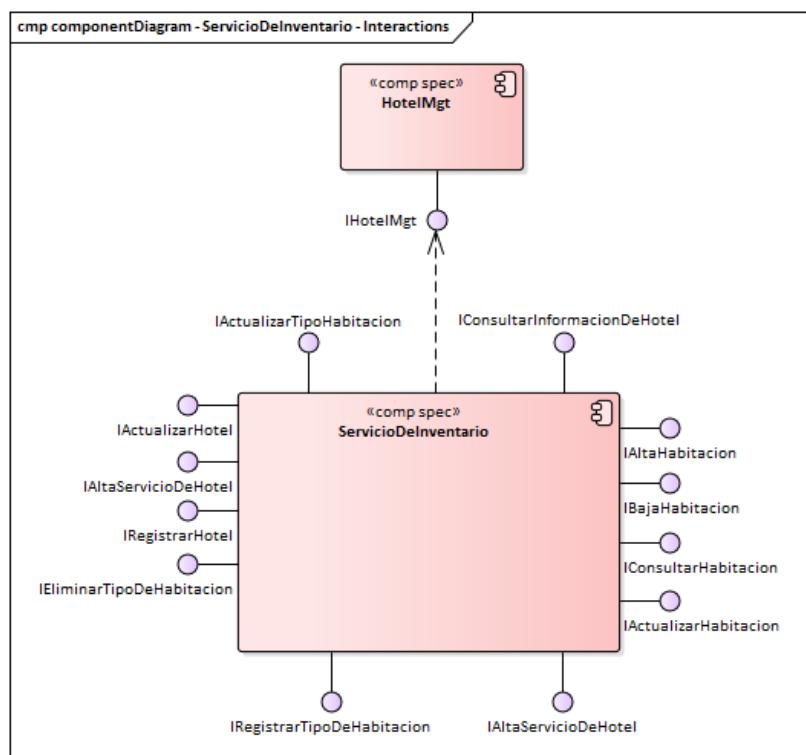
## Servicio de estancia (interacciones)



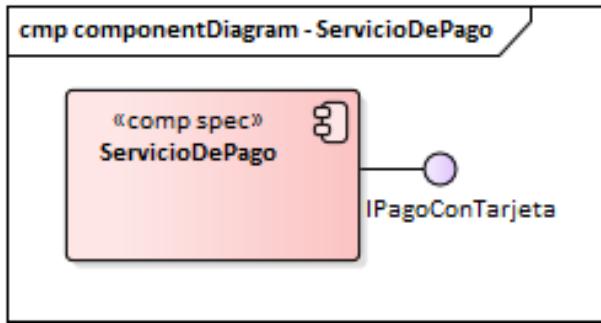
## Servicio de inventario



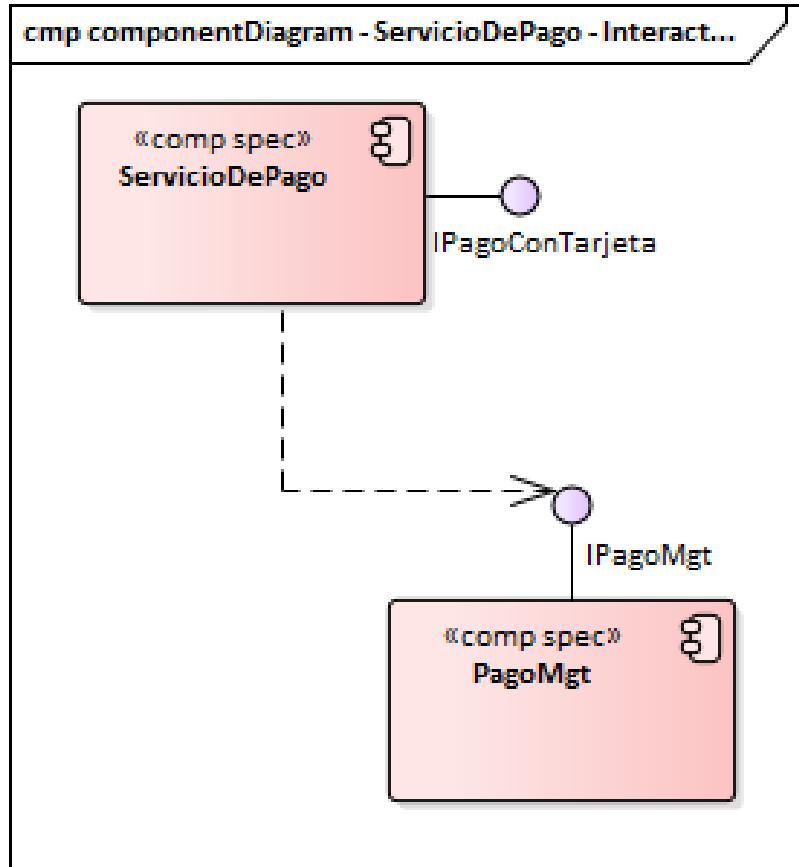
## Servicio de inventario (interacciones)



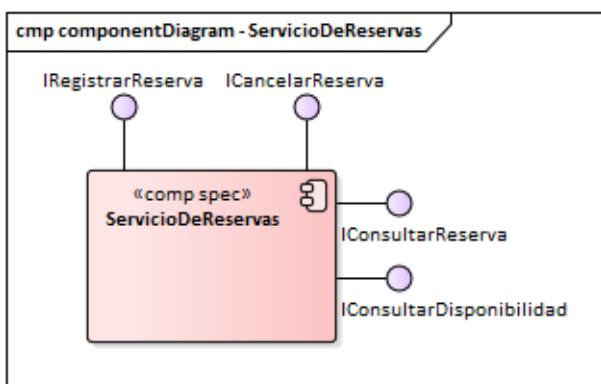
## Servicio de pago



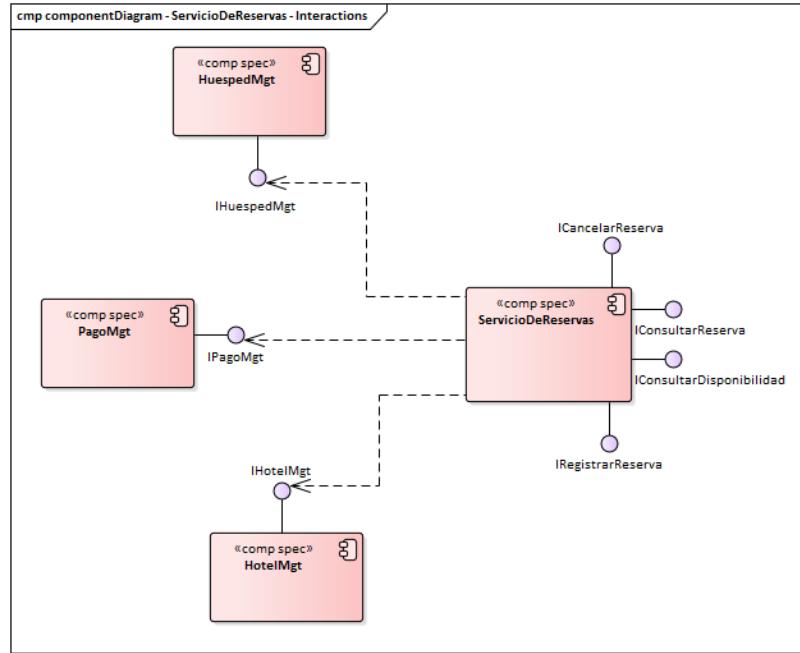
## Servicio de pago (interacciones)



## Servicio de reservas

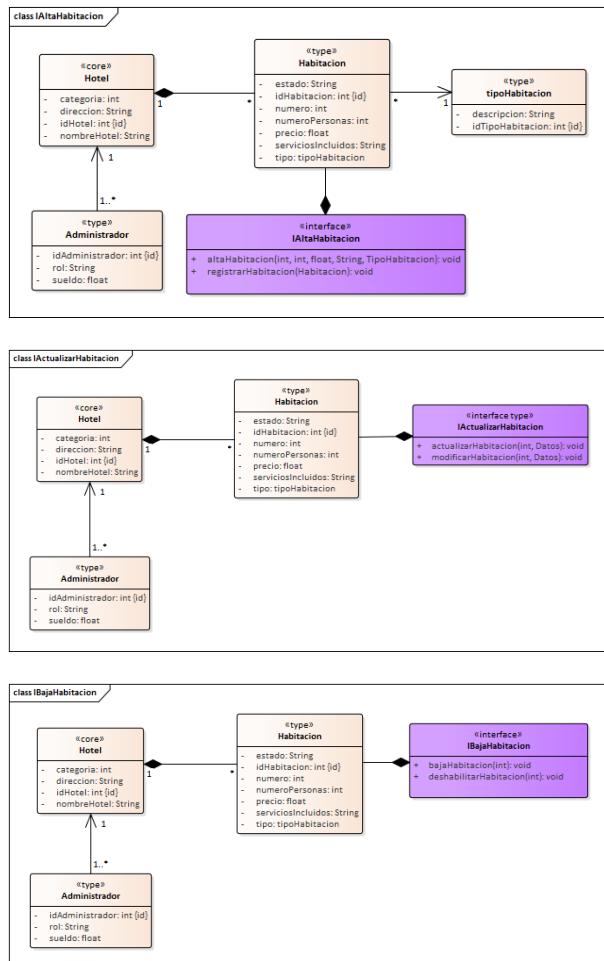


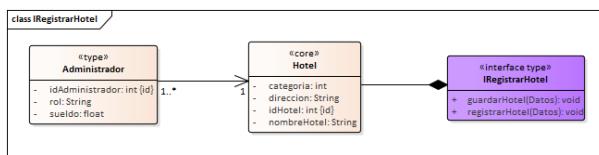
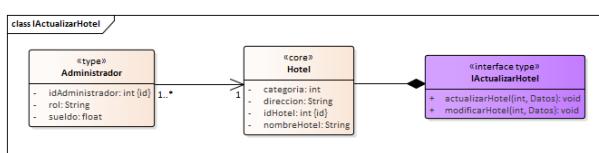
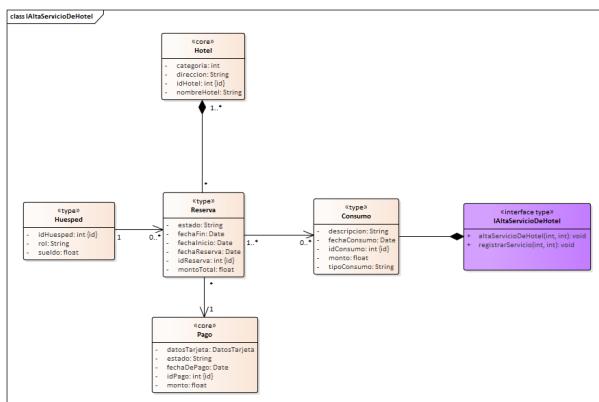
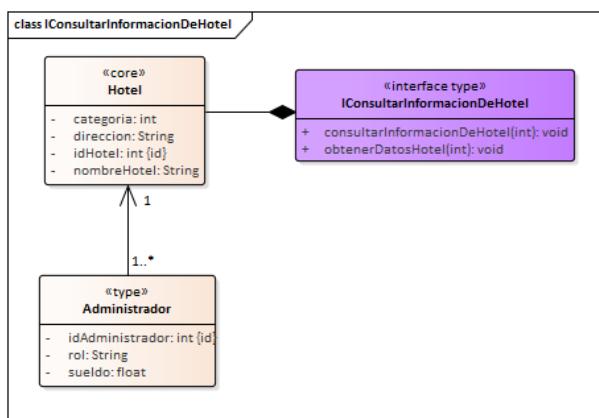
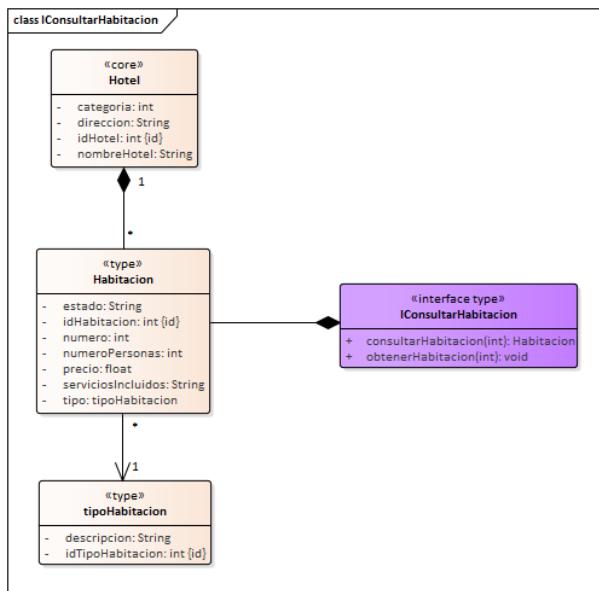
## Servicio de reservas (interacciones)

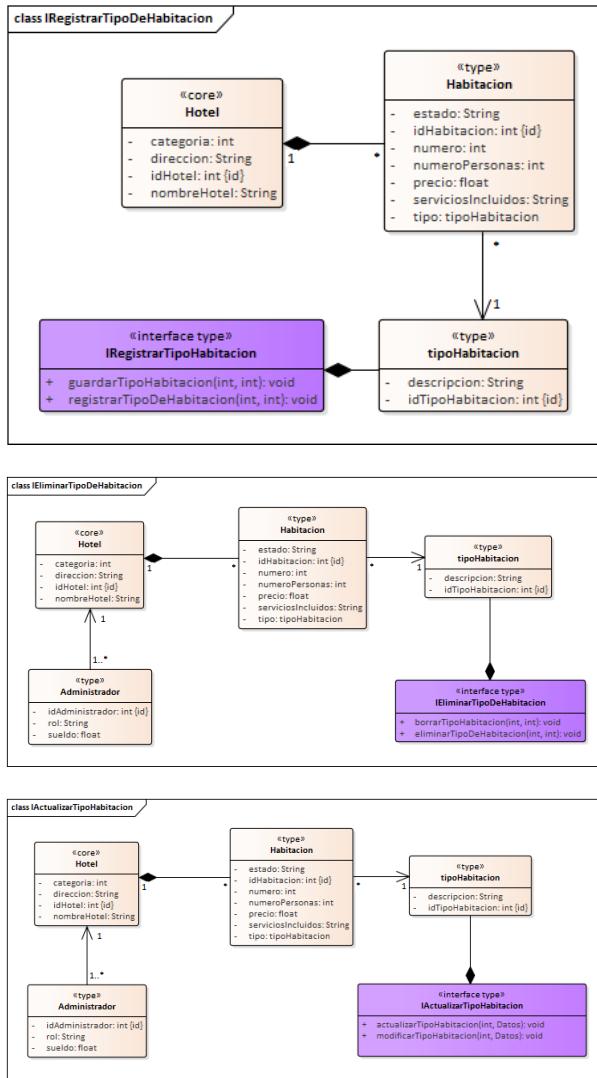


## 5.4.2. Diagramas de especificación de interfaces

### Interfaz de Inventario







## Precondiciones:

Las fechas deben ser válidas ( $\text{fechaInicio} \leq \text{fechaFin}$ ).

La habitación debe existir en el registro del sistema.

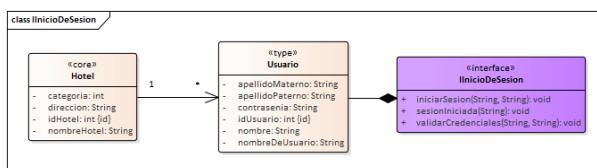
## Postcondiciones:

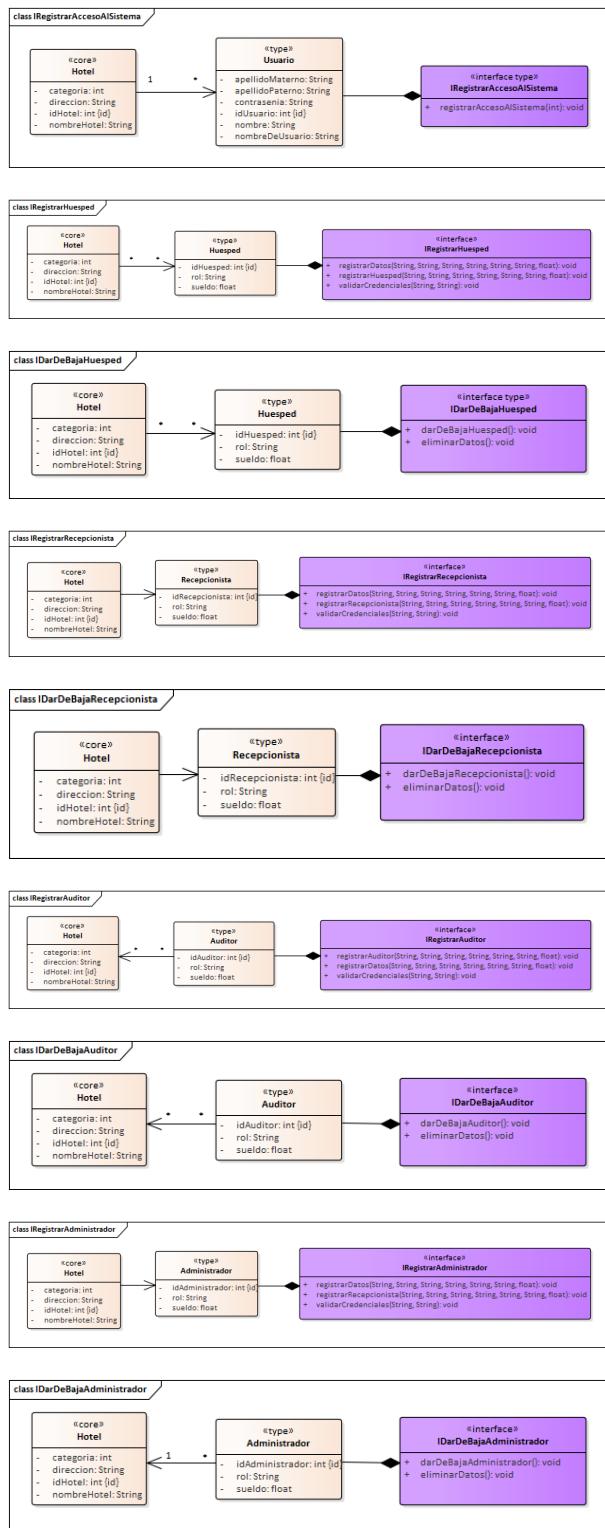
Si se actualiza el estado, este cambio se refleja en la base de datos.

Se guarda dentro de la base de datos los cambios respecto a habitaciones.

Las consultas retornan listas no nulas (pueden ser vacías si no hay resultados).

## Interfaz de Acceso al Sistema





## Precondiciones:

El nombre de usuario y contraseña no deben ser nulos ni vacíos.

El sistema debe estar conectado a la base de datos (ConexionDB activa).

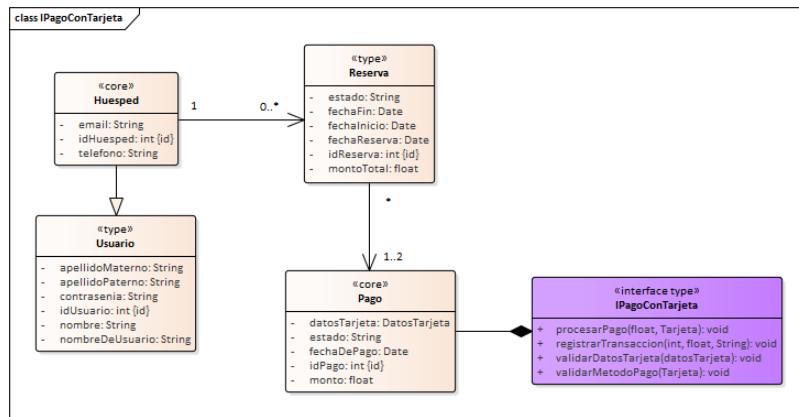
## Postcondiciones:

Si las credenciales son válidas, retorna true y carga el objeto Usuario asociado.

Si las credenciales son incorrectas, retorna false sin alterar el estado del sistema.

El sistema guarda los cambios dentro de la base de datos referentes a usuarios nuevos, eliminados o actualizados.

## Interfaz de Pago



### Precondiciones:

El usuario debe tener una reserva confirmada.

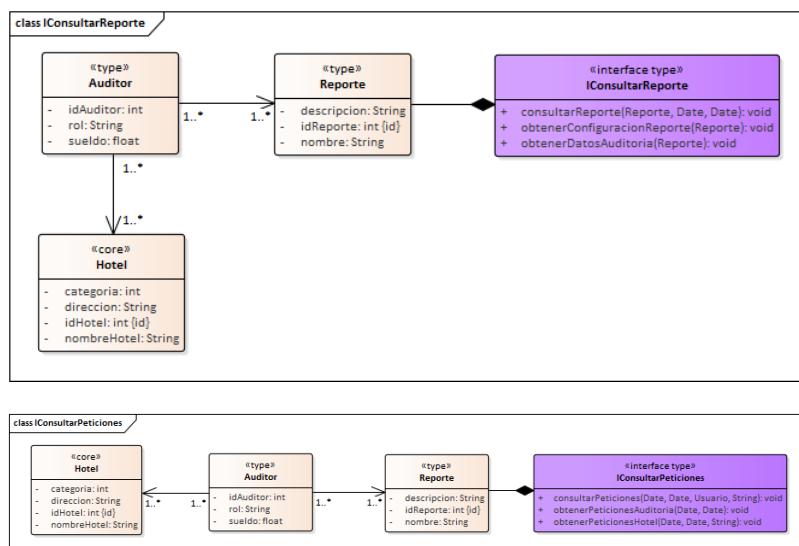
Los datos de pago deben ser válidos (método, monto, referencia).

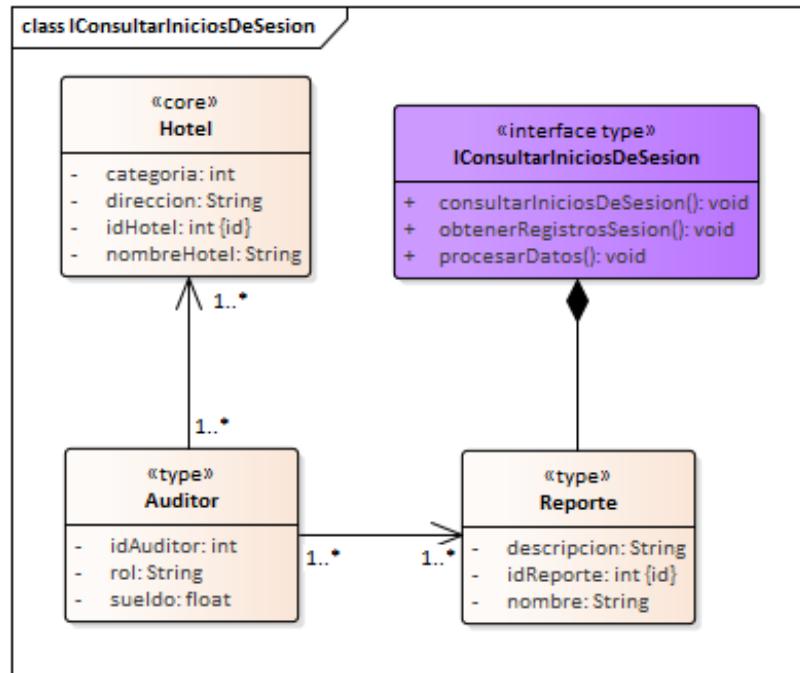
### Postcondiciones:

Si el pago se procesa correctamente, se genera un registro en la base de datos y se actualiza el estado de la reserva.

Si el pago falla, no se generan registros ni cambios de estado.

## Interfaz de Auditoría





### Precondiciones:

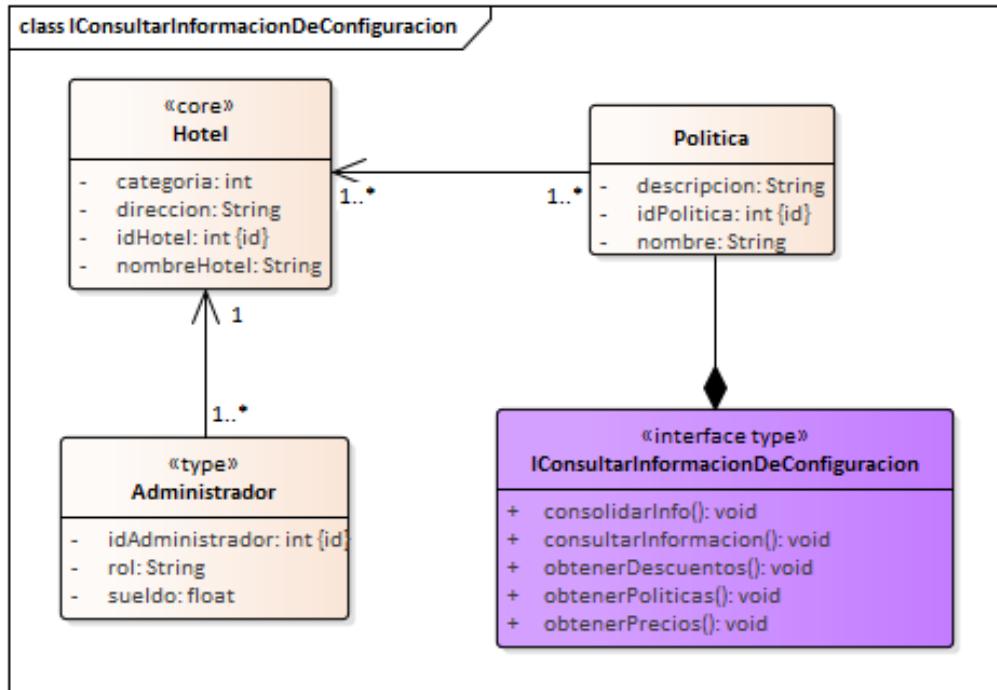
El usuario debe acceder al sistema con un rol de Auditor.

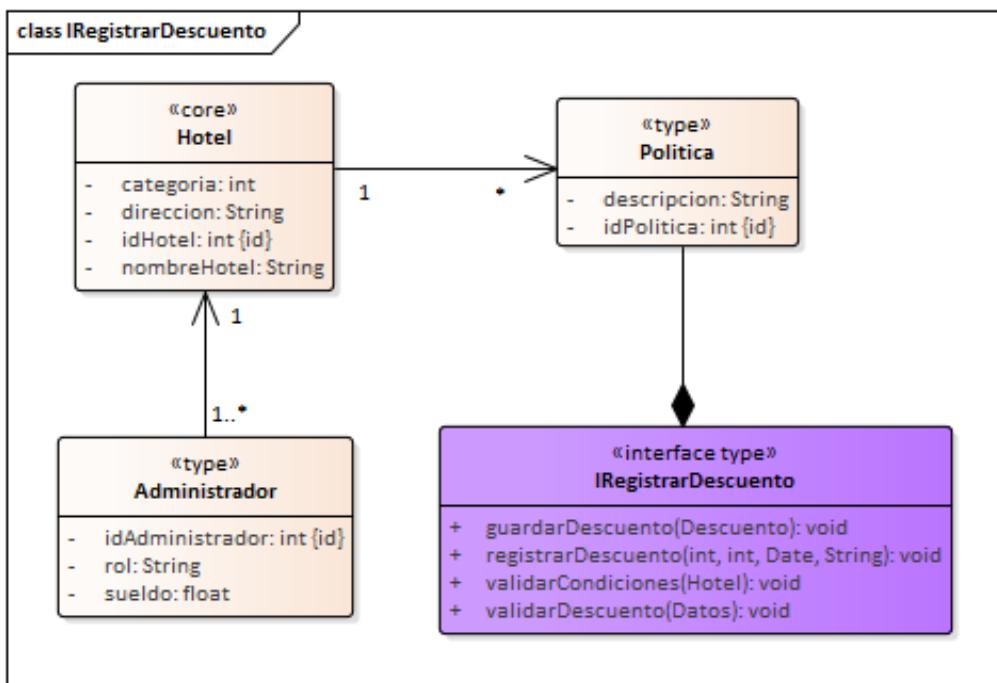
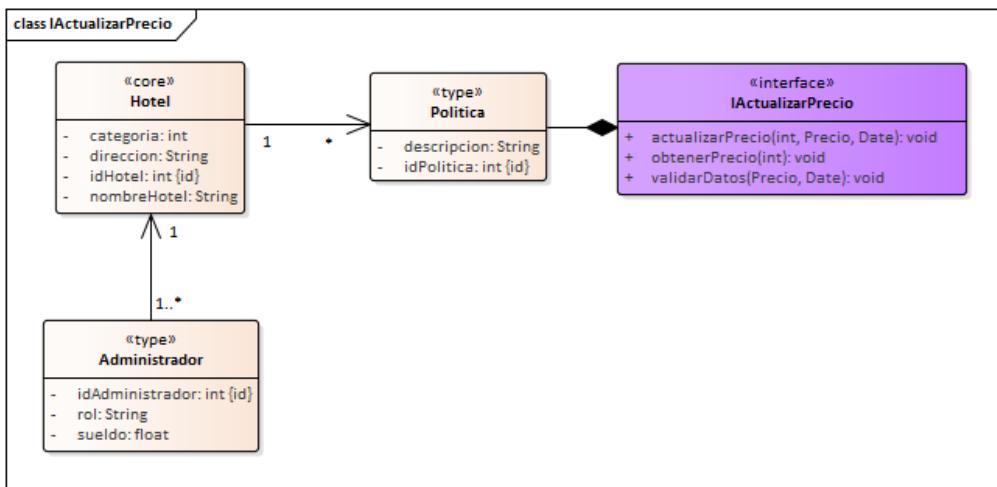
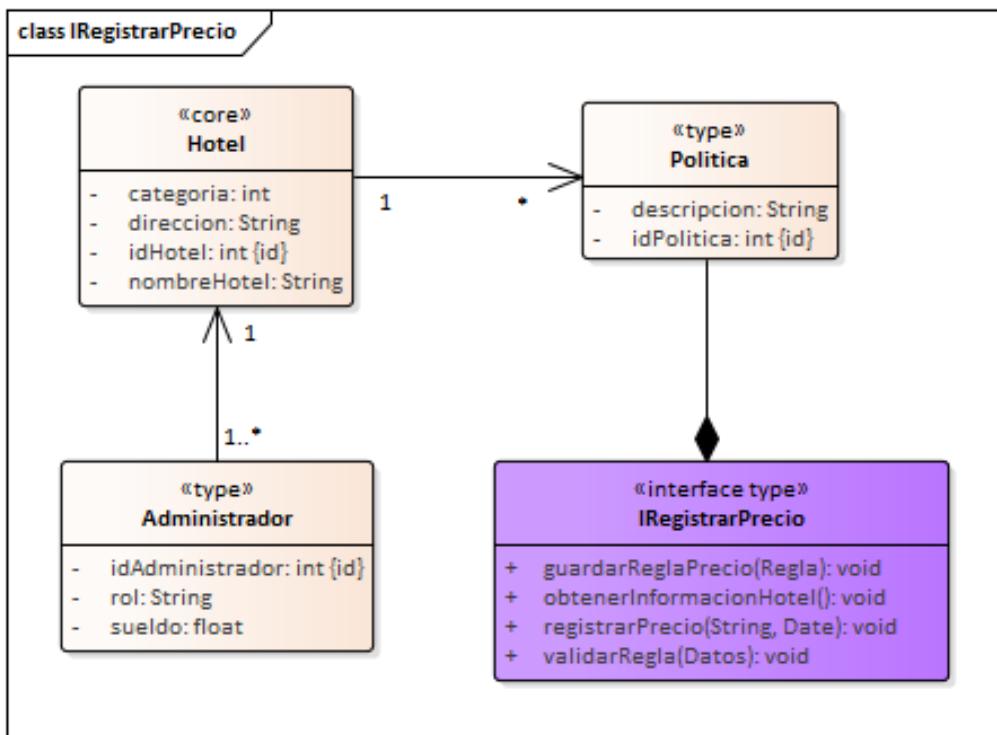
Las fechas ingresadas deben ser válidas y pertenecer a un rango cronológico coherente.

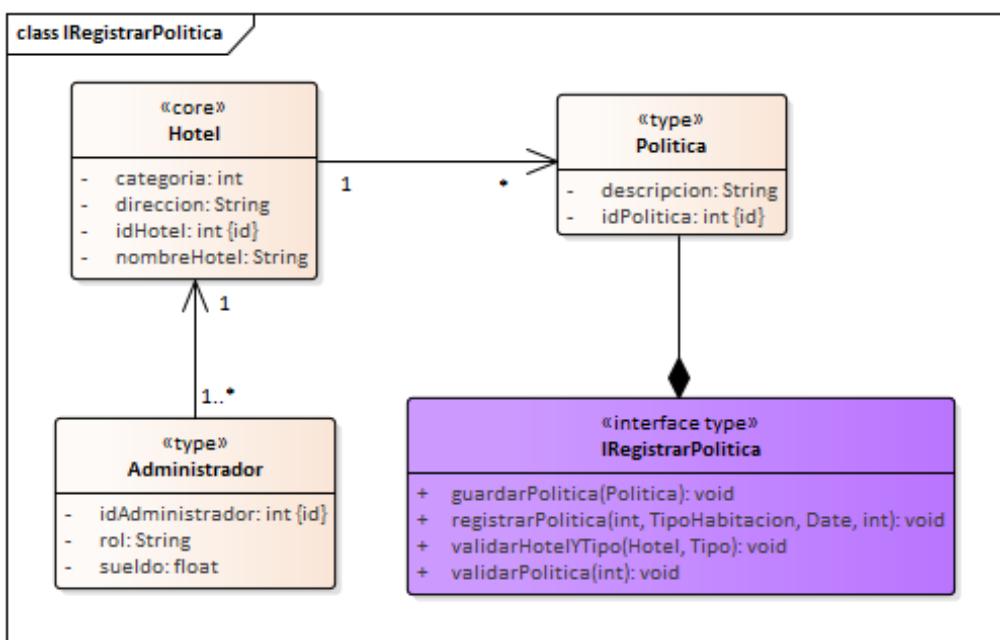
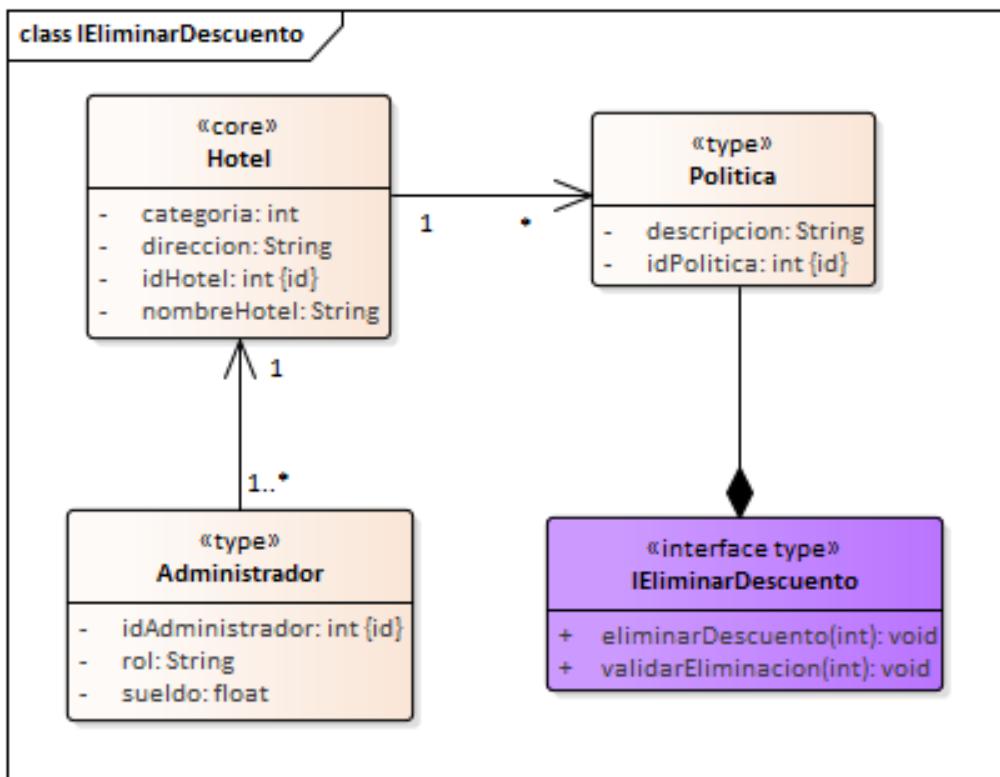
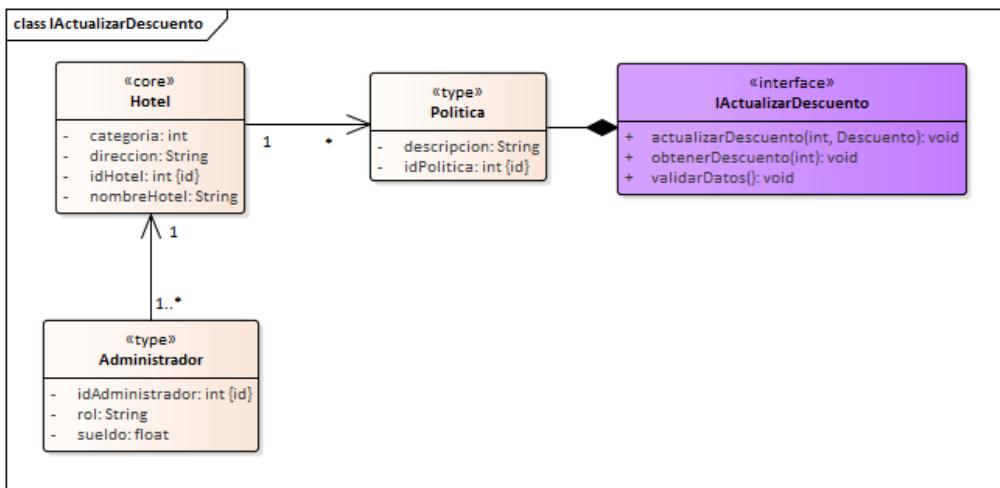
### Postcondiciones:

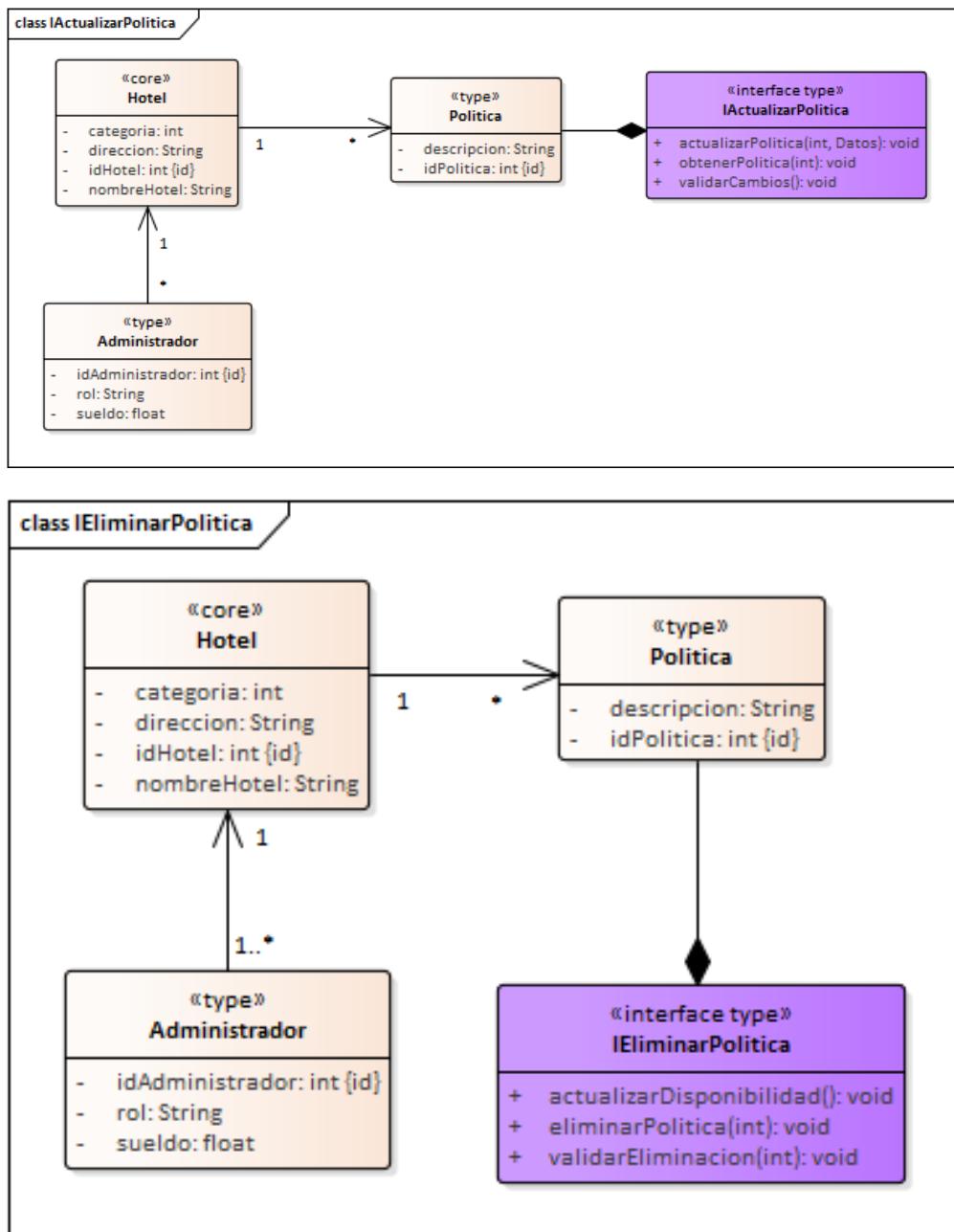
El sistema muestra los registros solicitados en función de los filtros aplicados.

### Interfaz de Configuración









### Precondiciones:

El usuario solicitante debe tener rol de administrador.

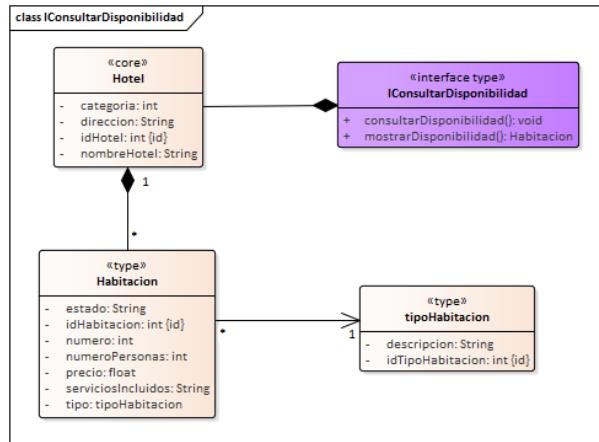
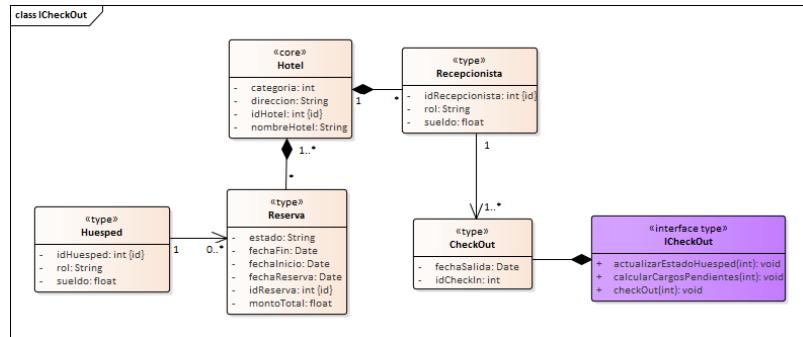
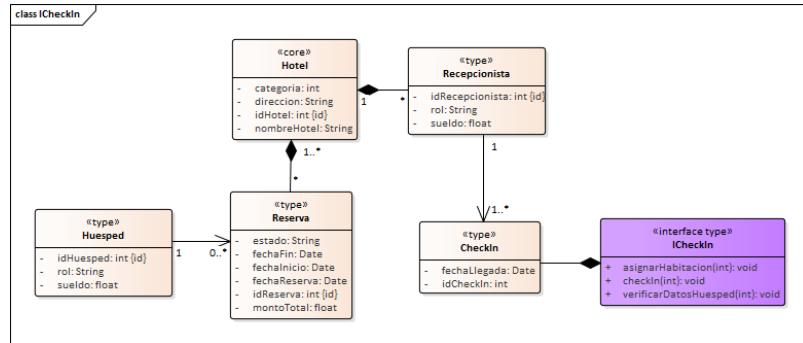
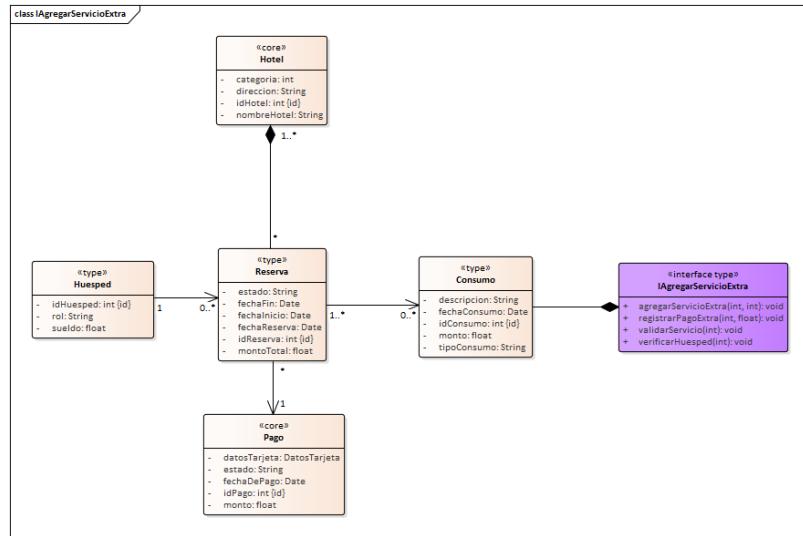
Las fechas ingresadas deben ser válidas y pertenecer a un rango cronológico coherente.

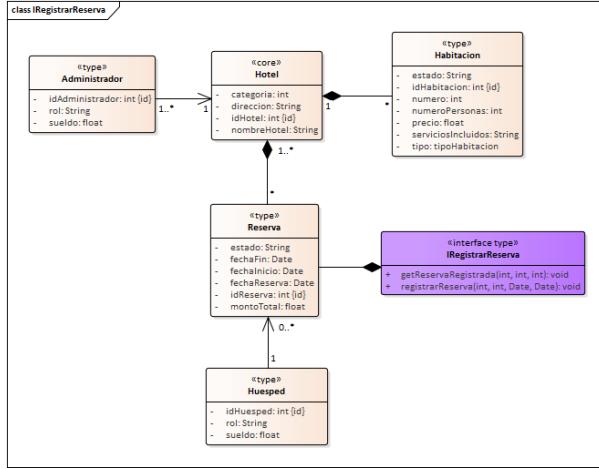
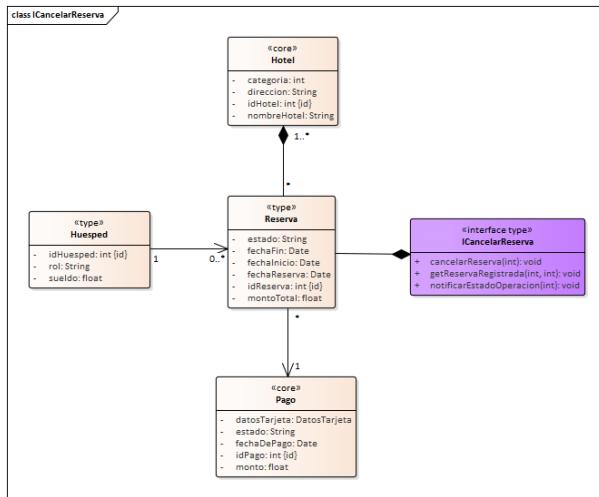
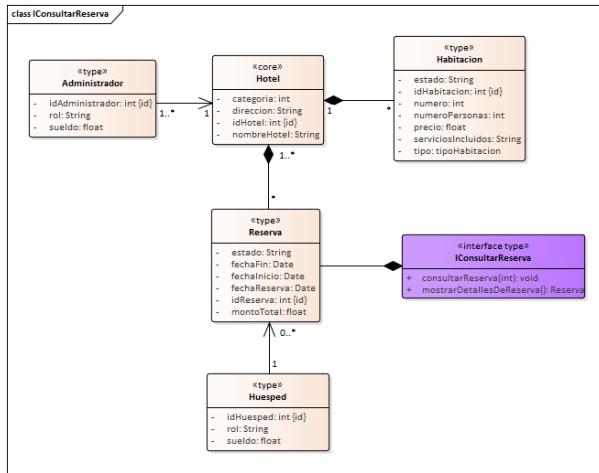
### Postcondiciones:

El sistema registra, actualiza o elimina aspectos de la configuración del sistema dentro de la base de datos.

Los datos añadidos (como descuentos) se guardan o actualizan en la tabla de historial (ReporteDAO).

### Interfaz de Reserva





## Precondiciones:

La habitación debe estar disponible en las fechas solicitadas.

El usuario debe estar autenticado en el sistema.

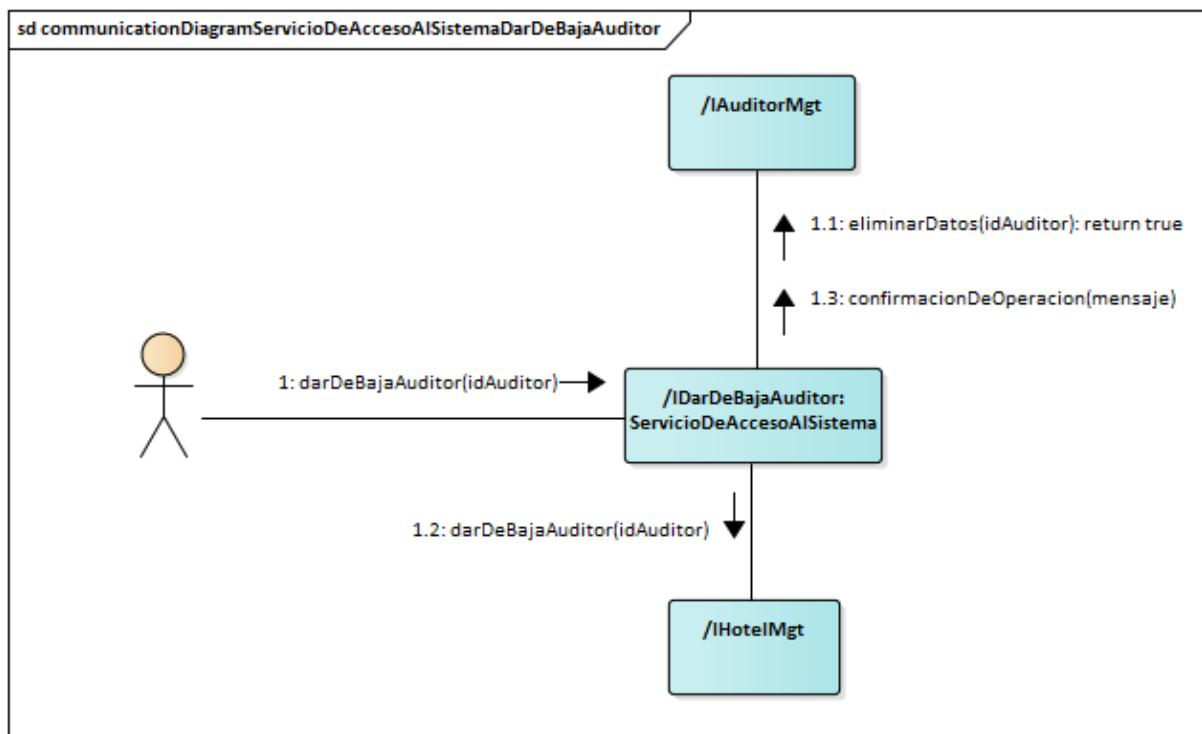
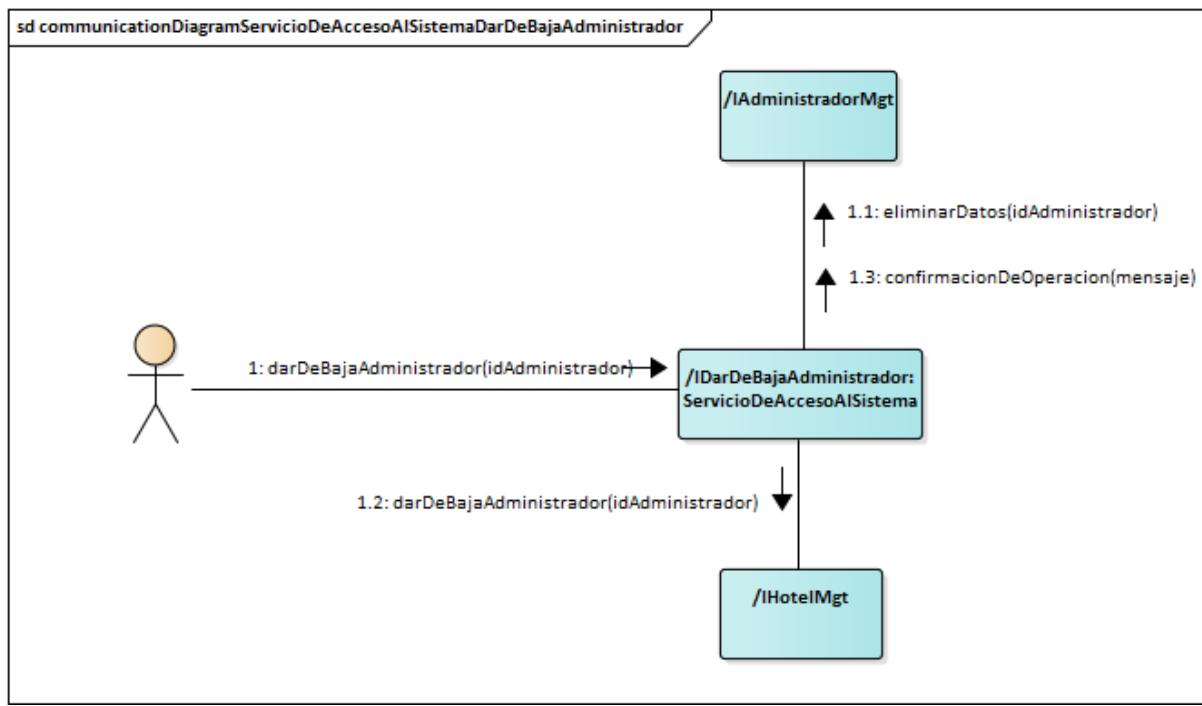
## Postcondiciones:

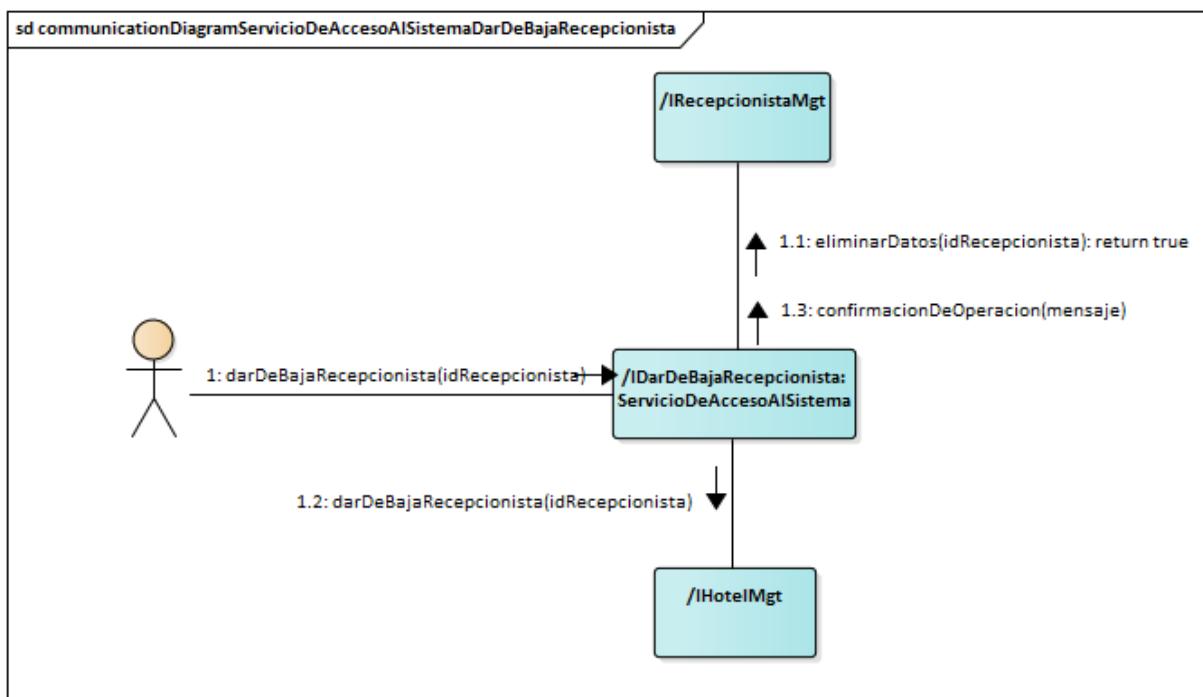
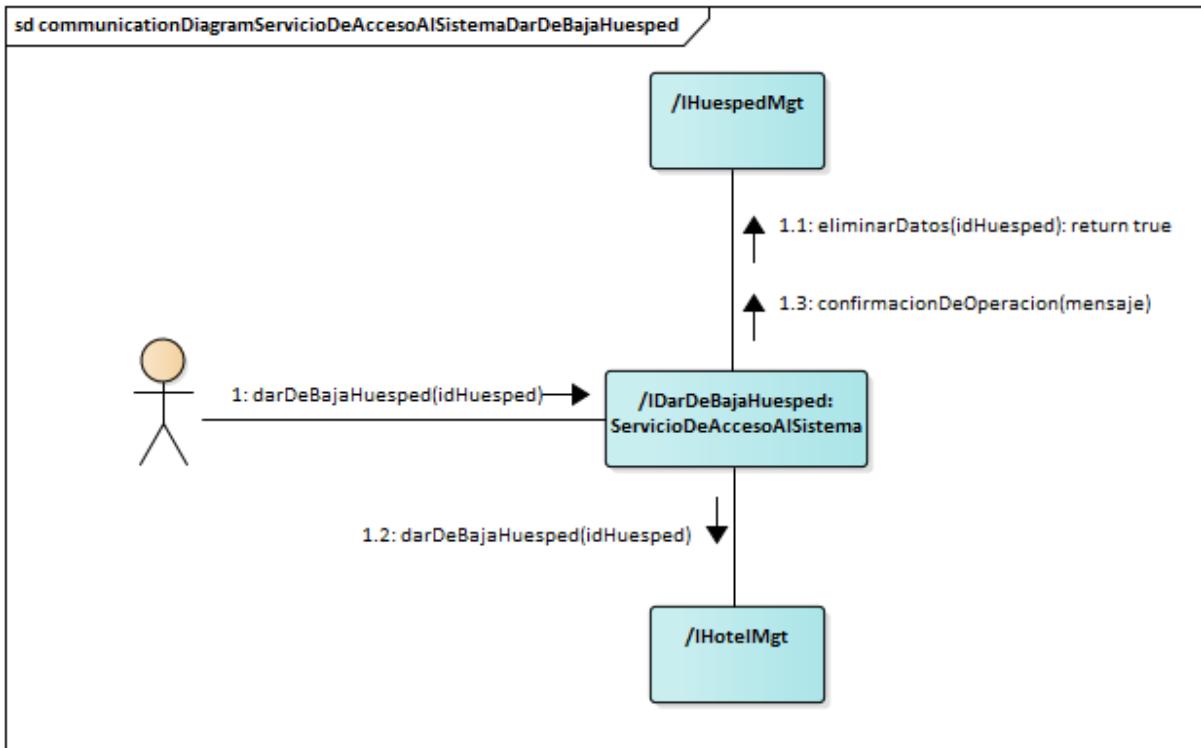
Si la reserva es válida, se almacena en la base de datos y se actualiza el estado de la habitación.

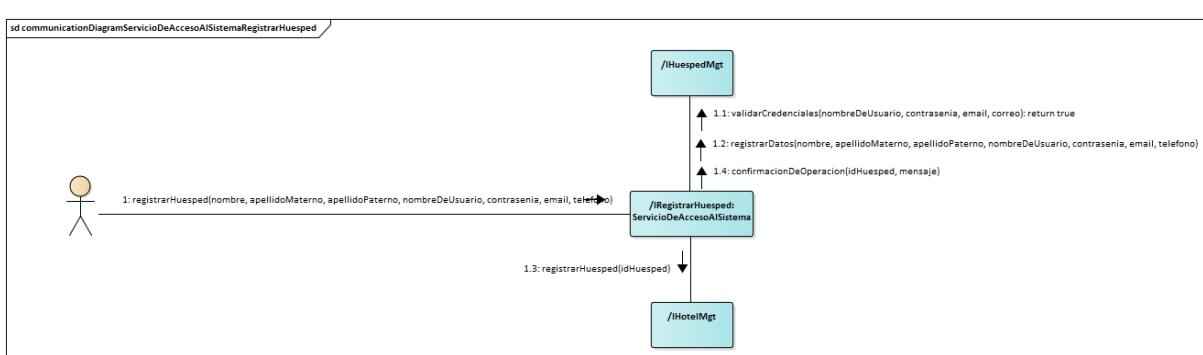
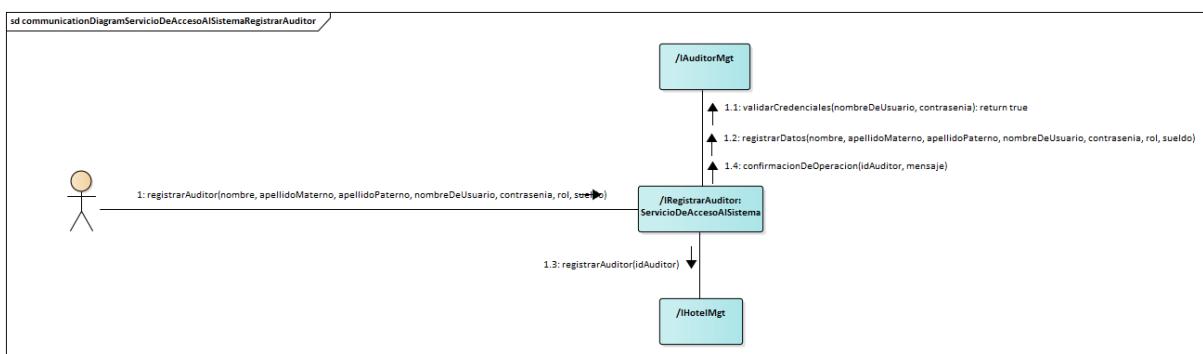
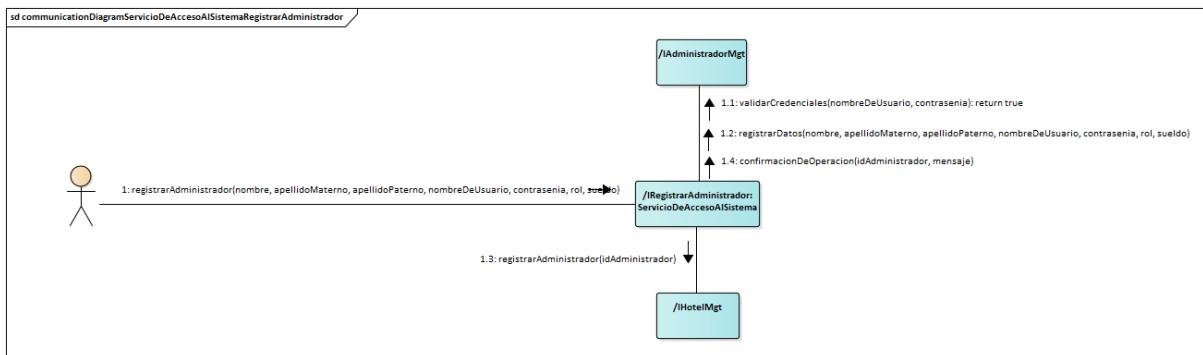
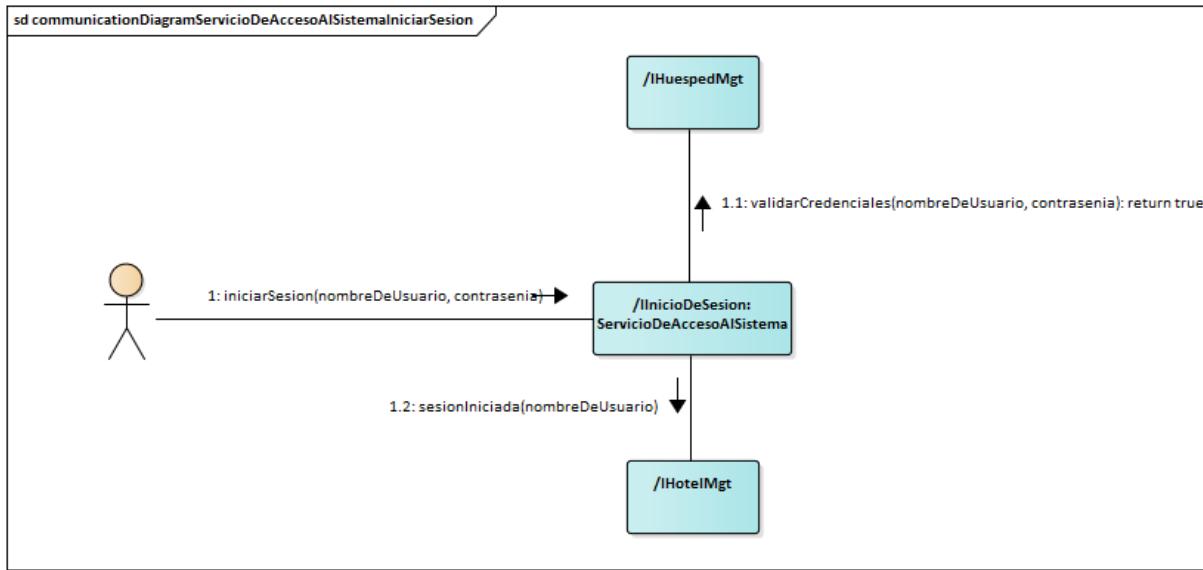
Si se cancela, la habitación vuelve a estado “disponible” y la reserva pasa a estado “cancelada”.

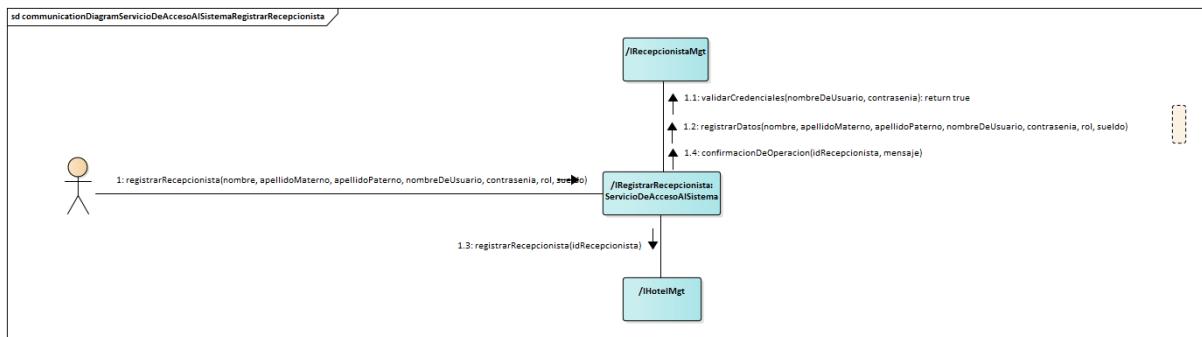
### 5.4.3. Interacción de componente

#### Servicio de Acceso al Sistema

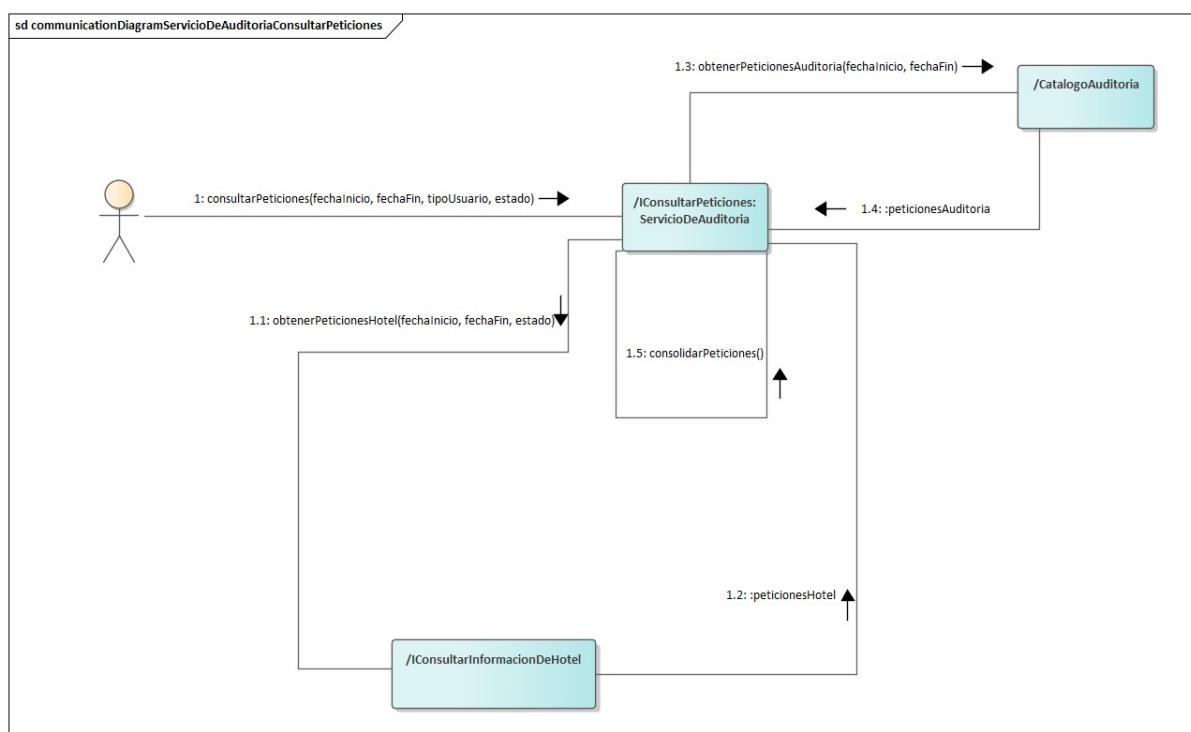
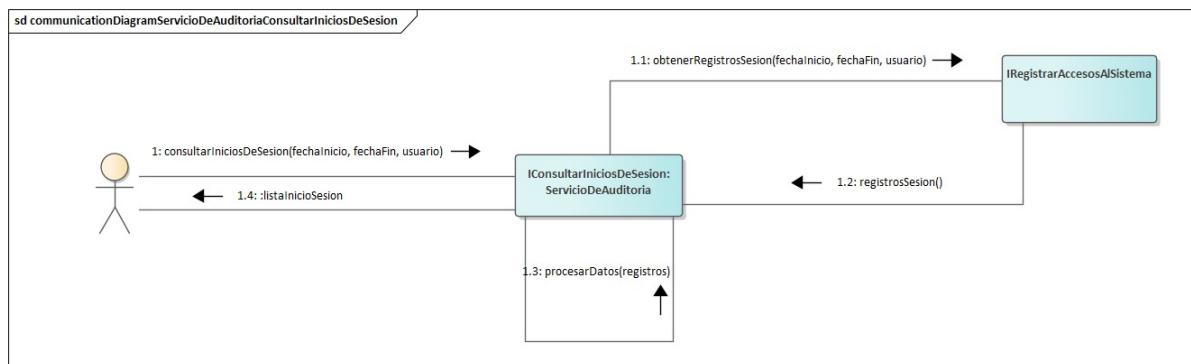


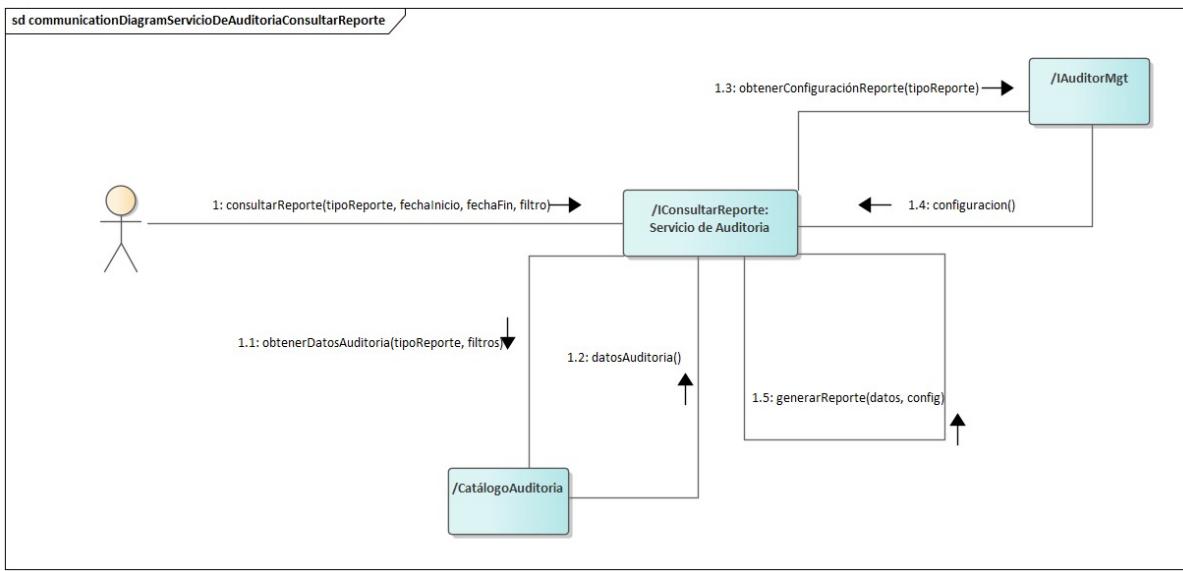




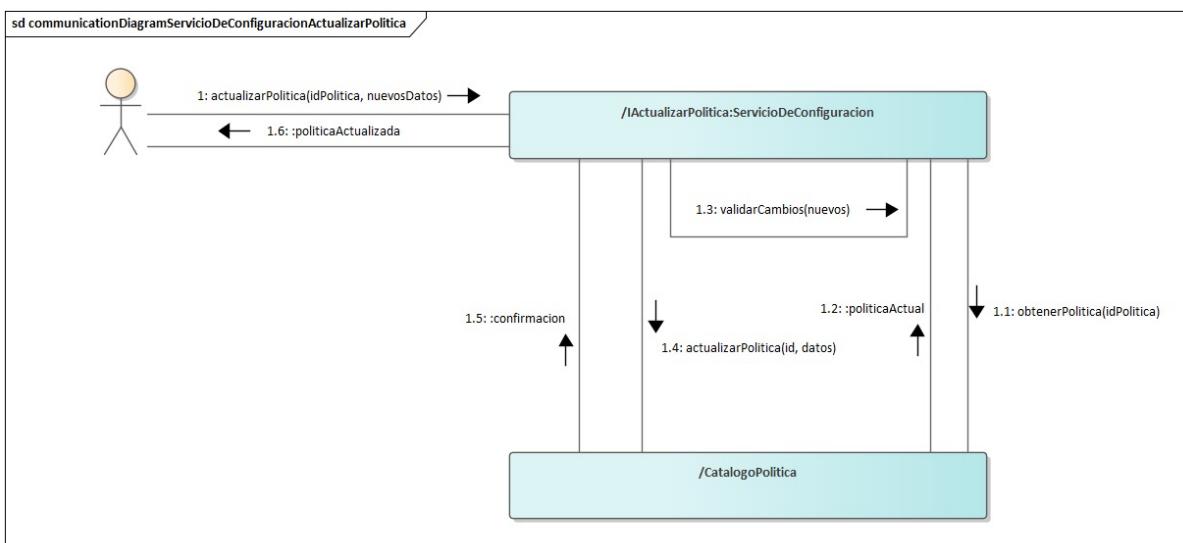
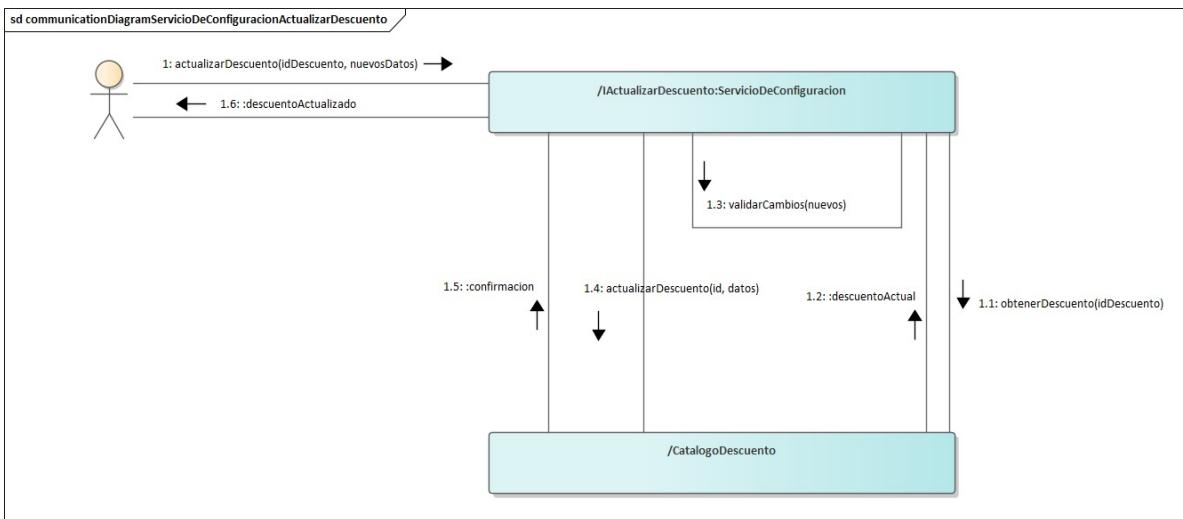


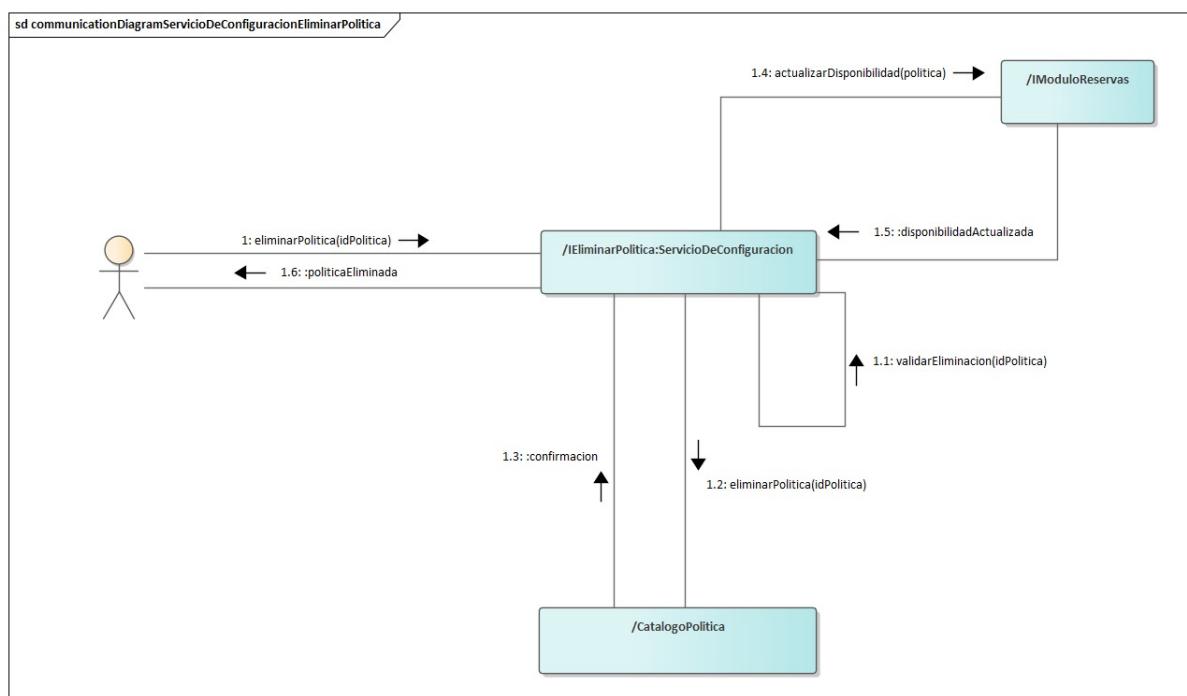
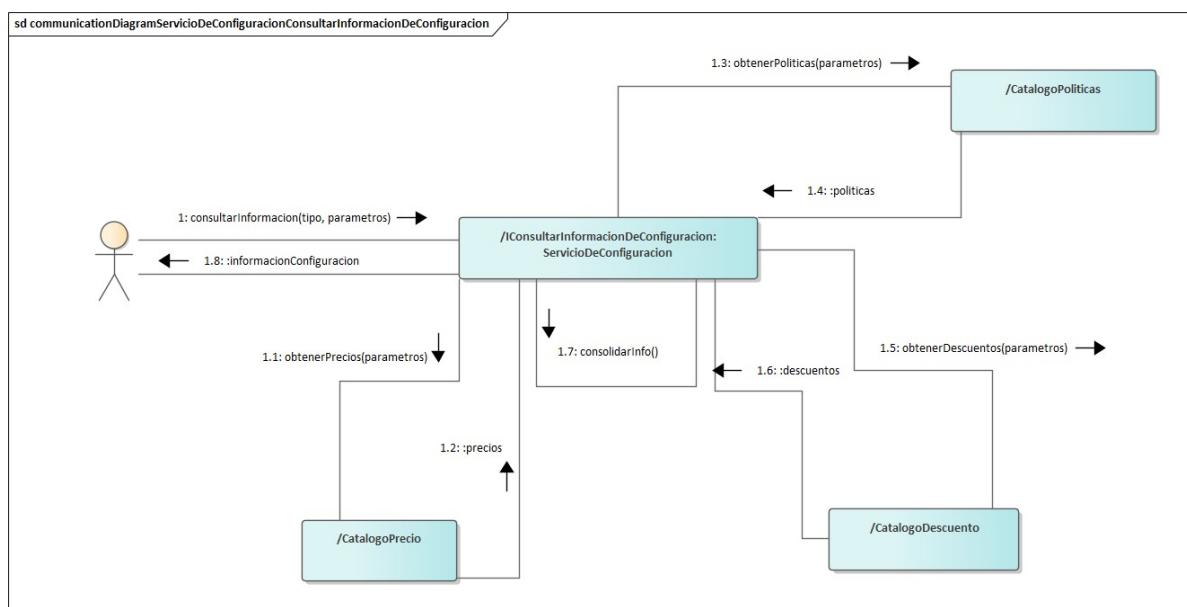
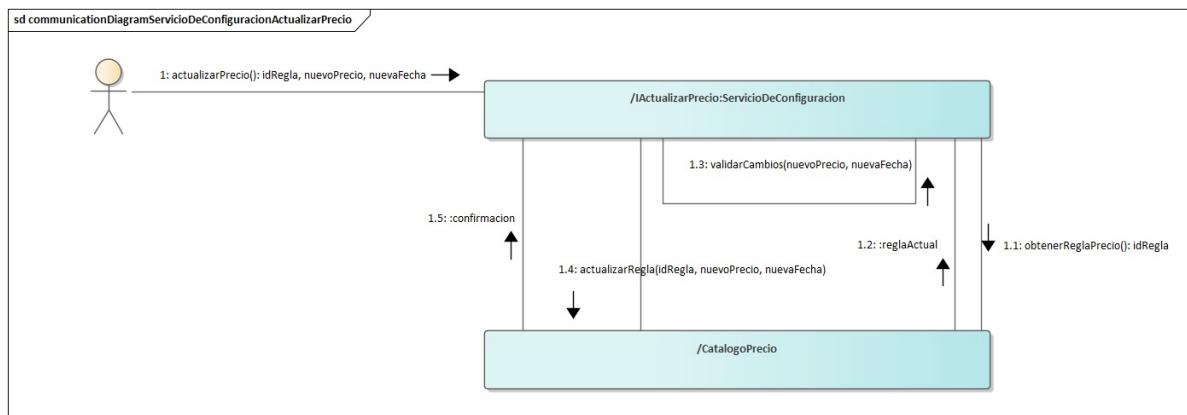
## Servicio de Auditoría

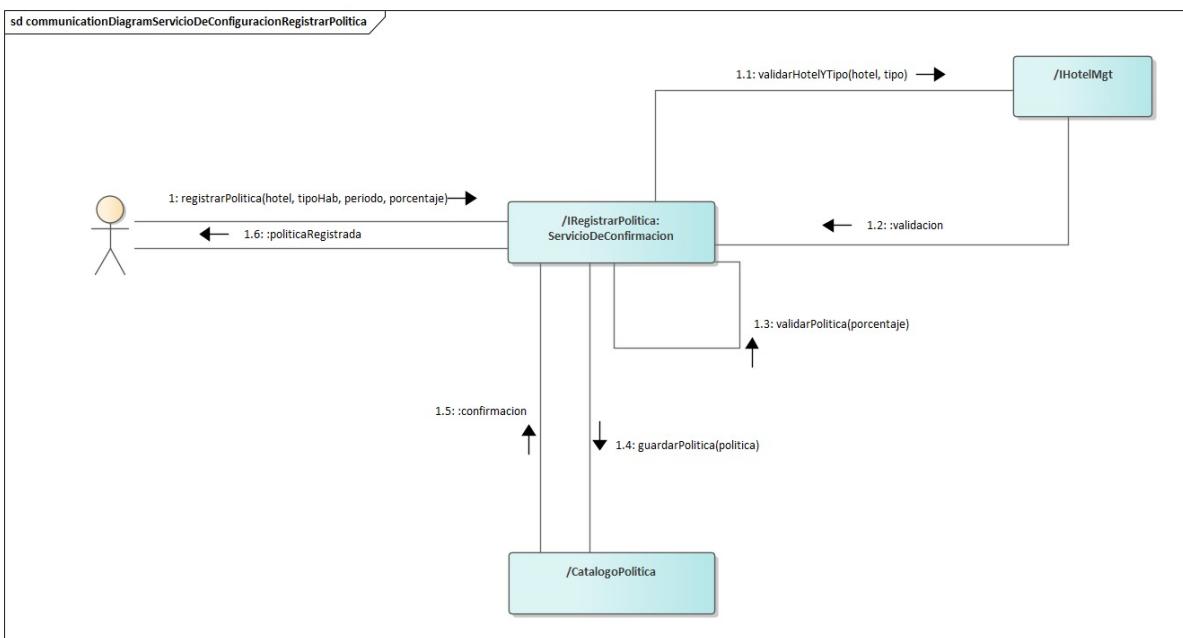
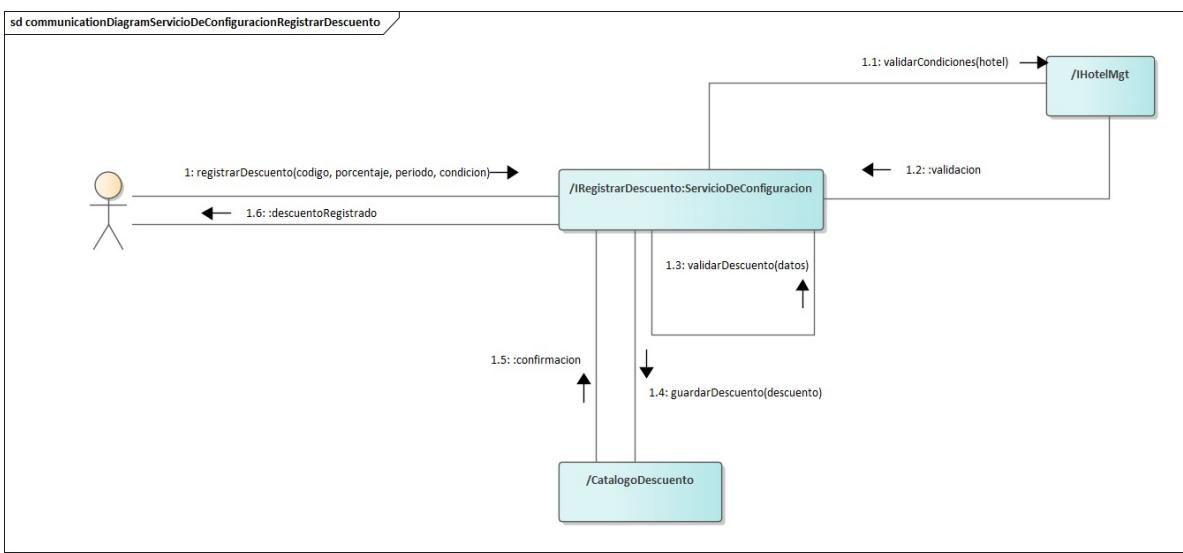
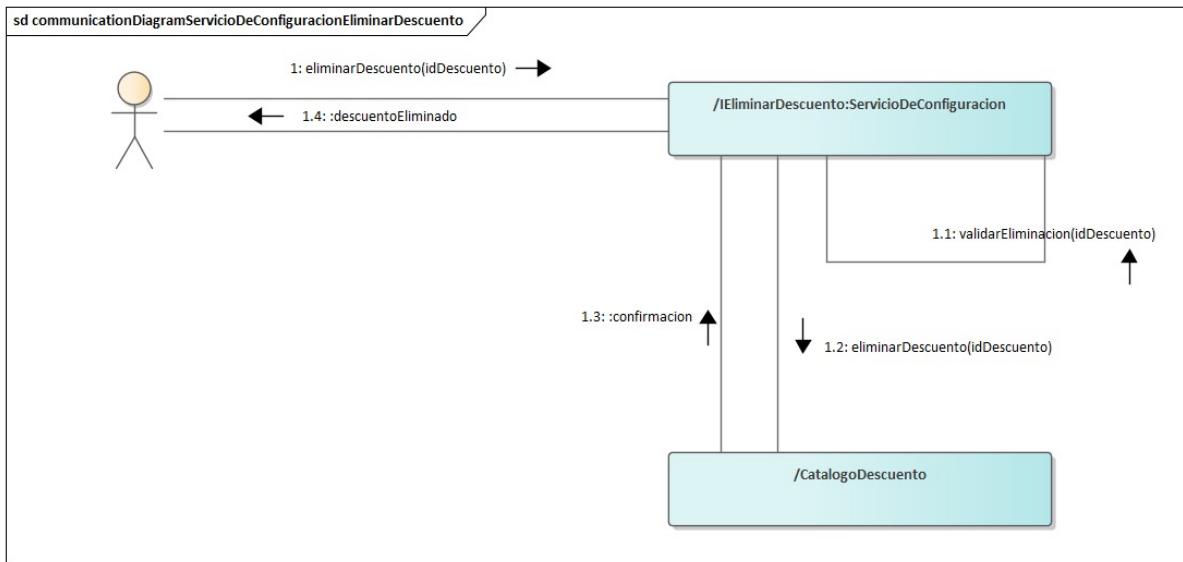


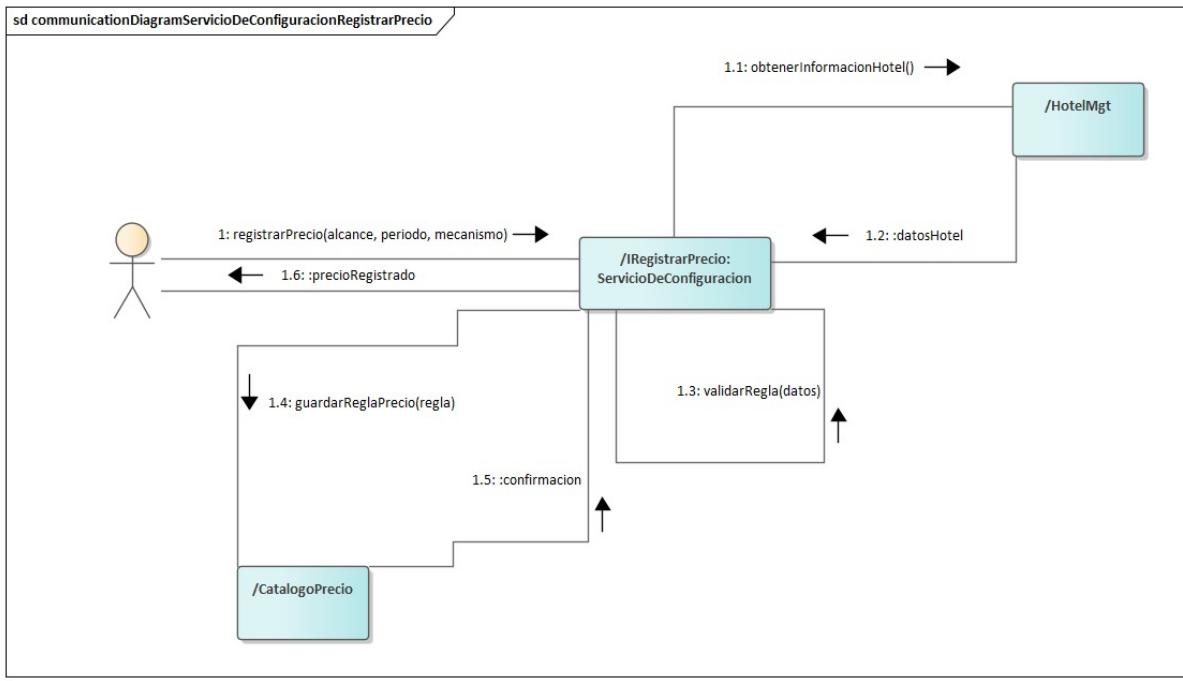


## Servicio de Configuración

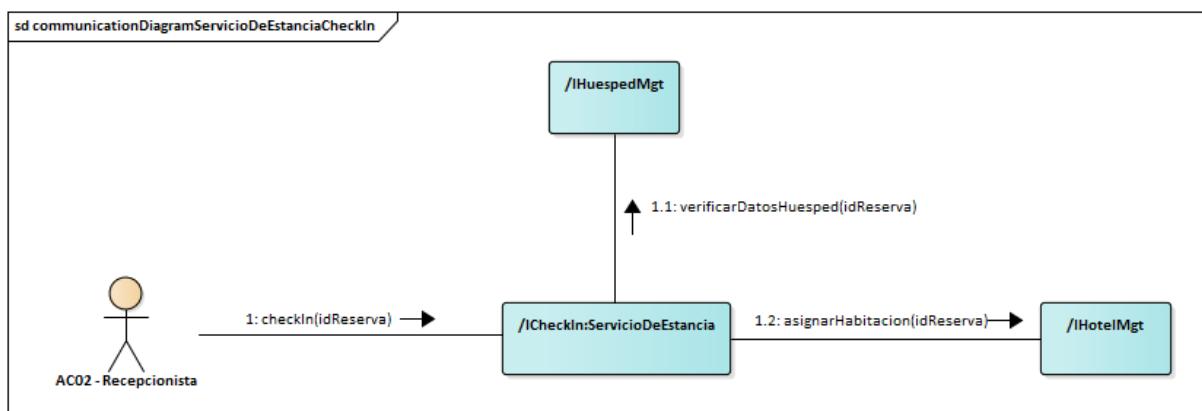
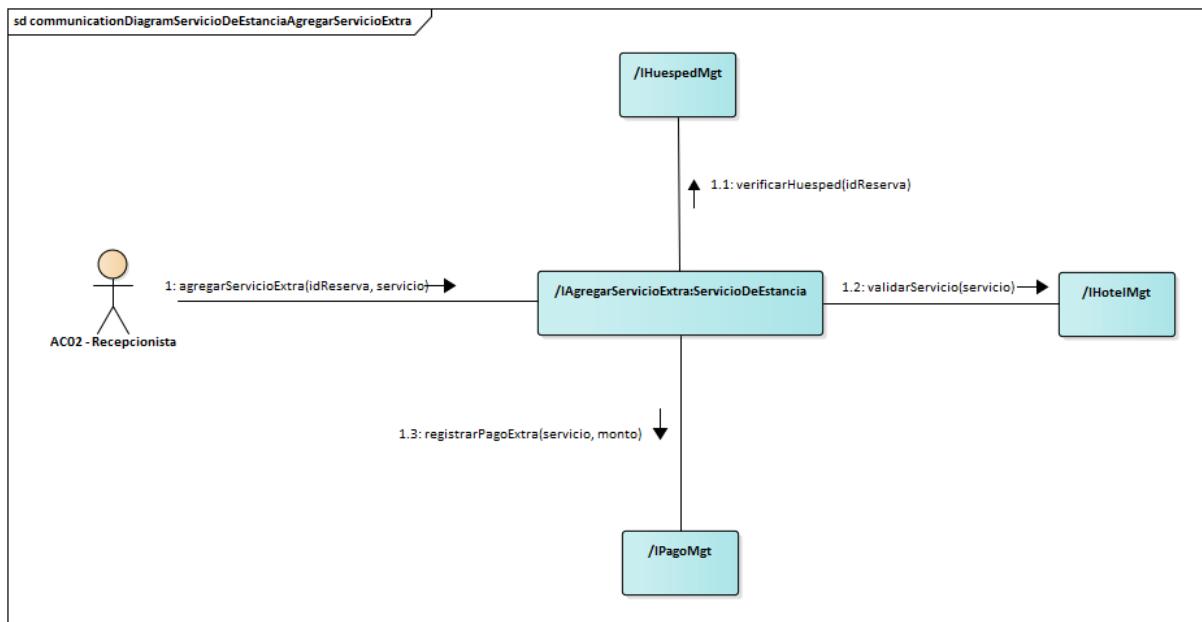


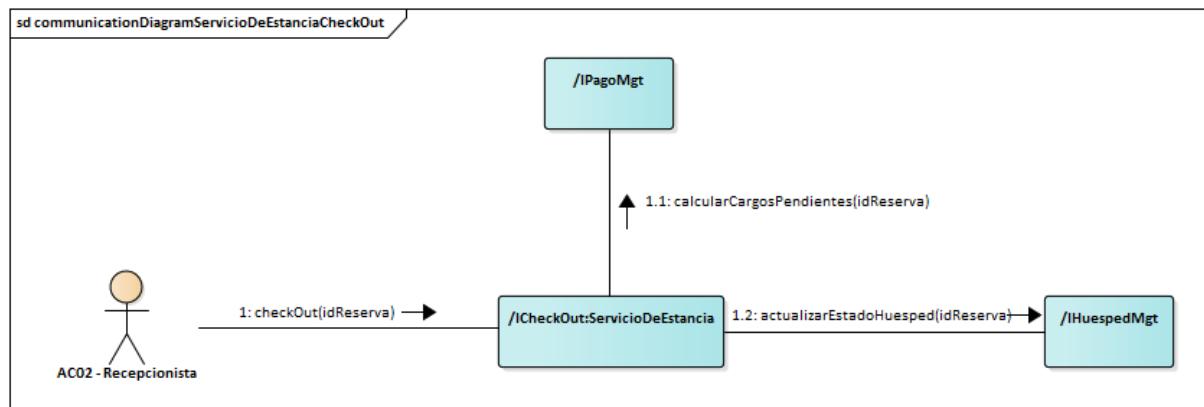




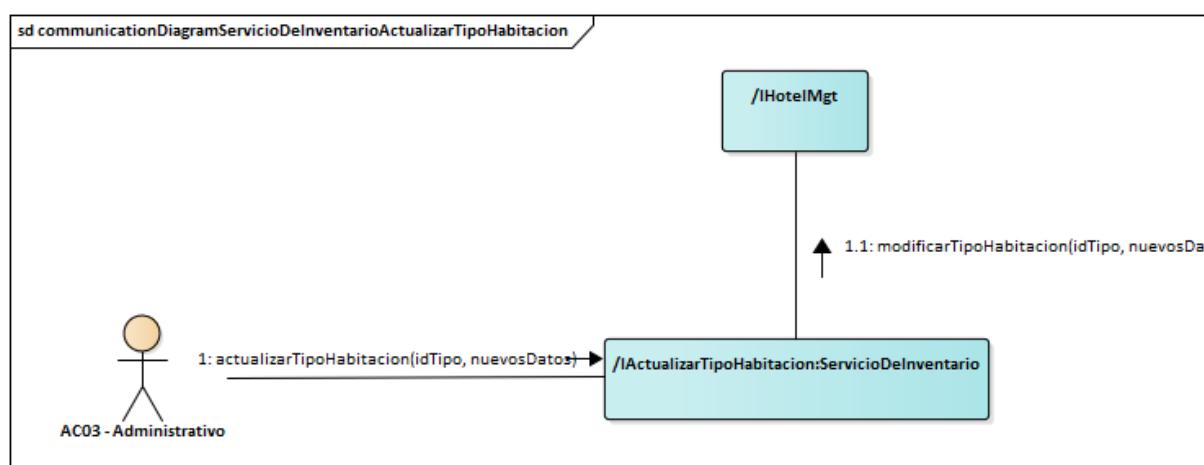
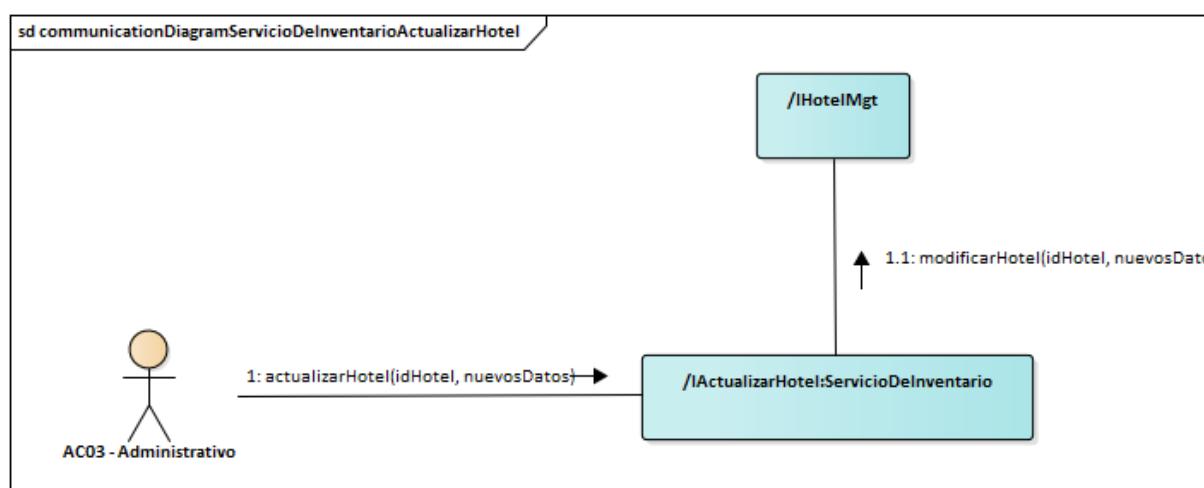
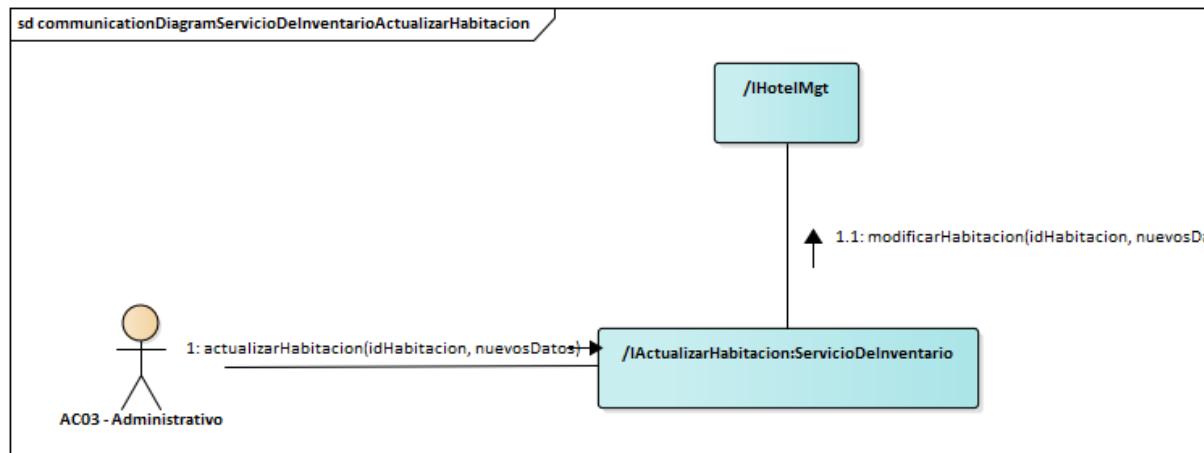


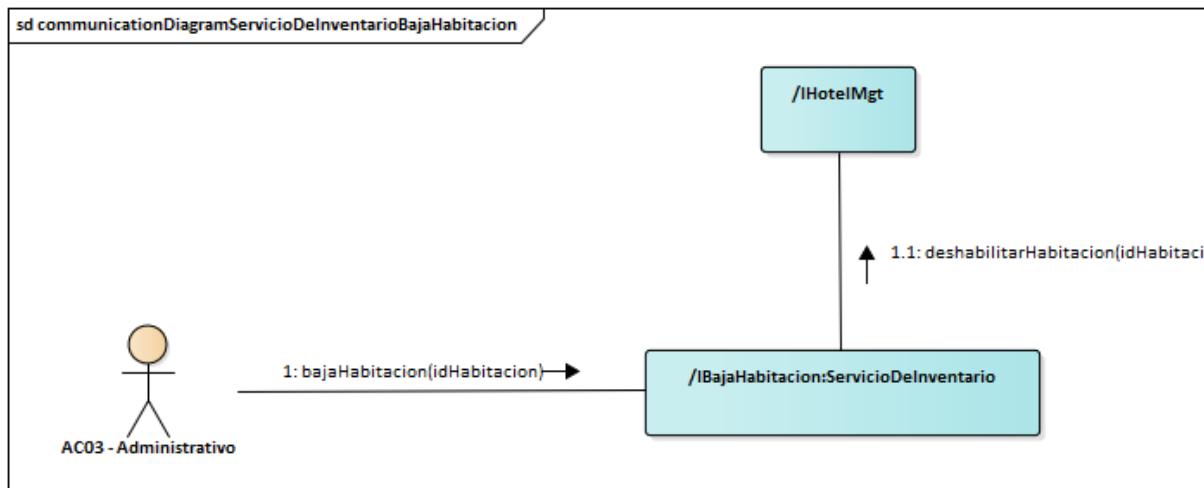
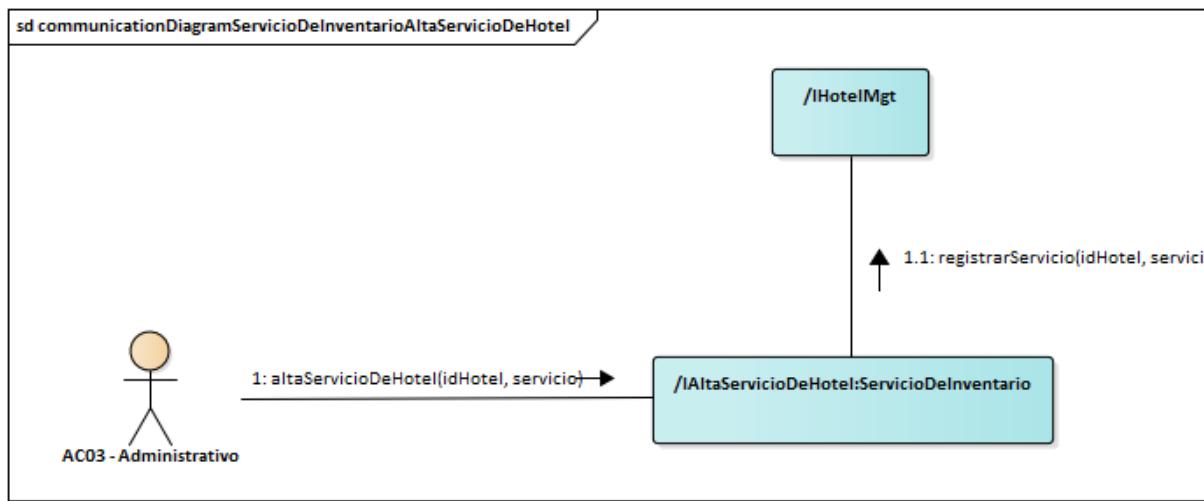
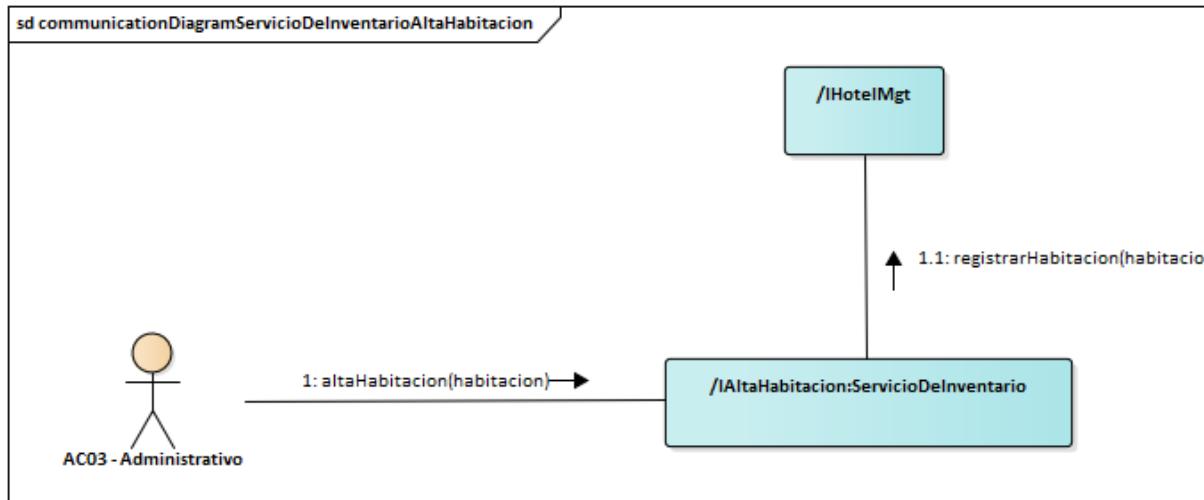
## Servicio de Estancia

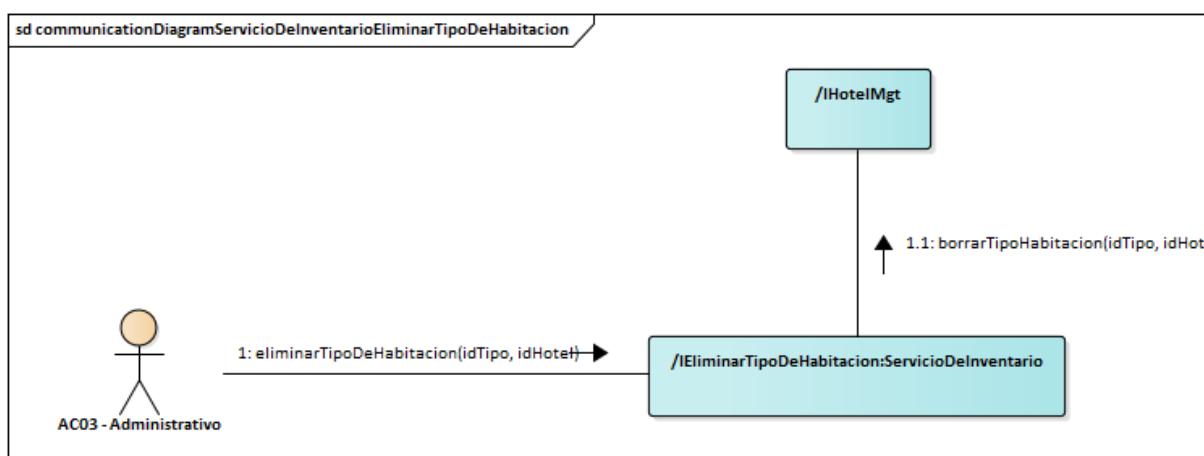
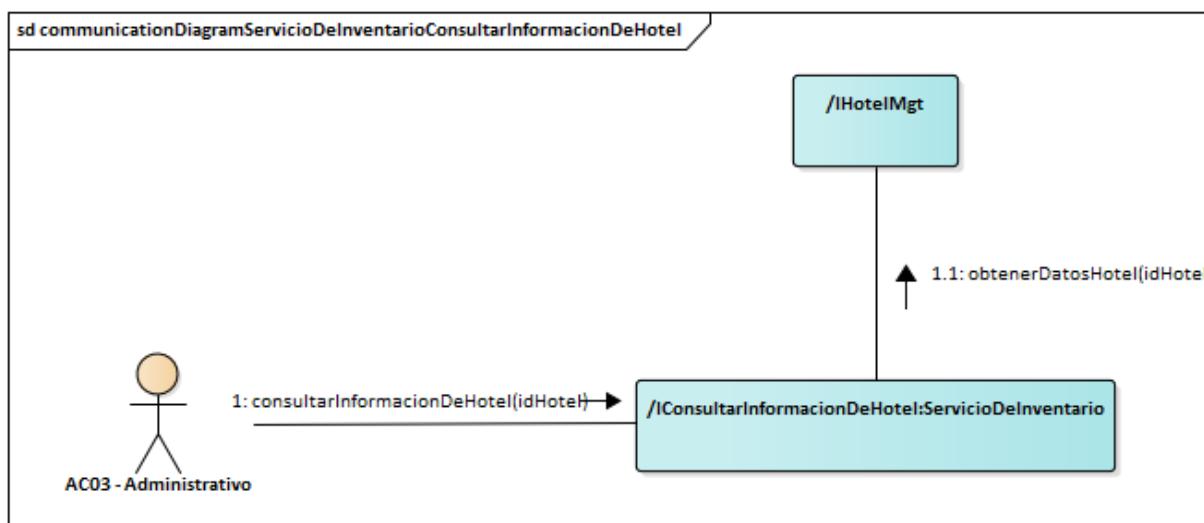
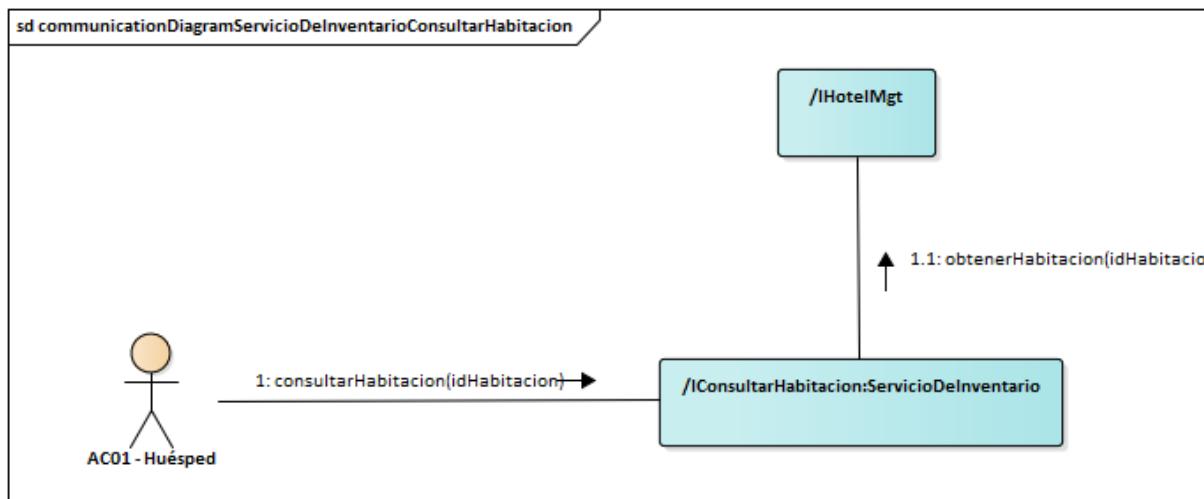


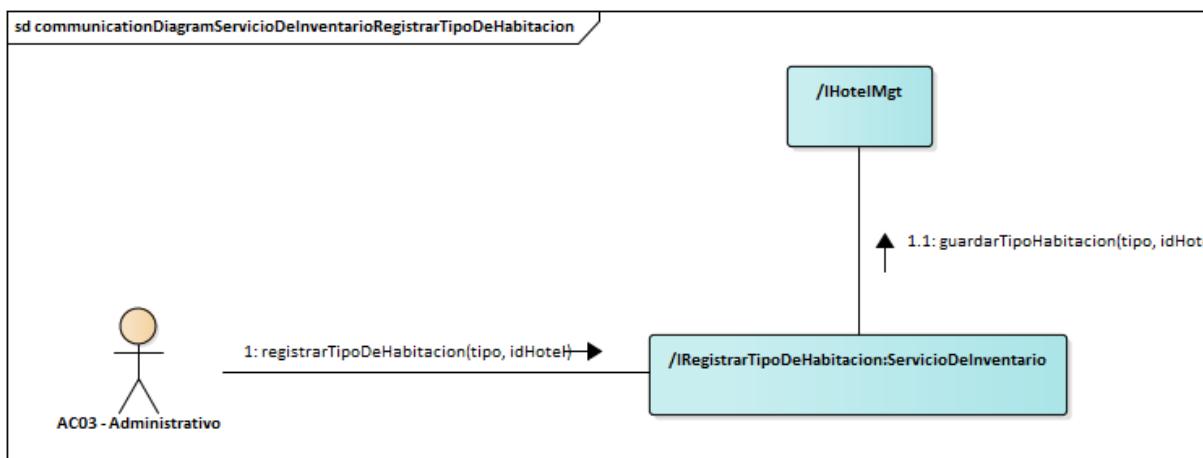
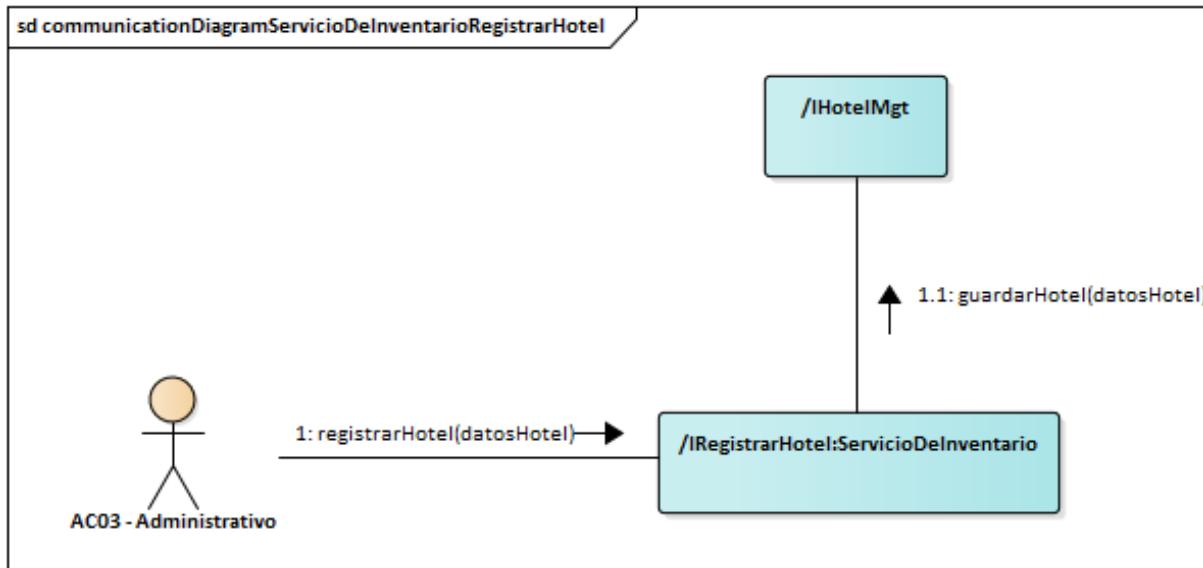


## Servicio de Inventario

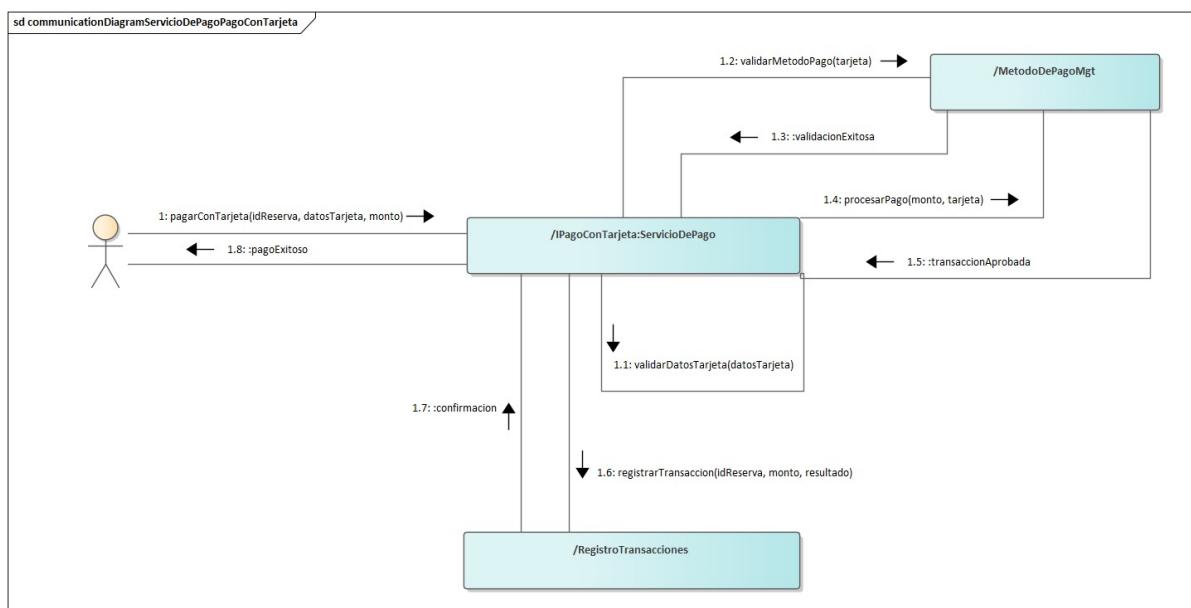




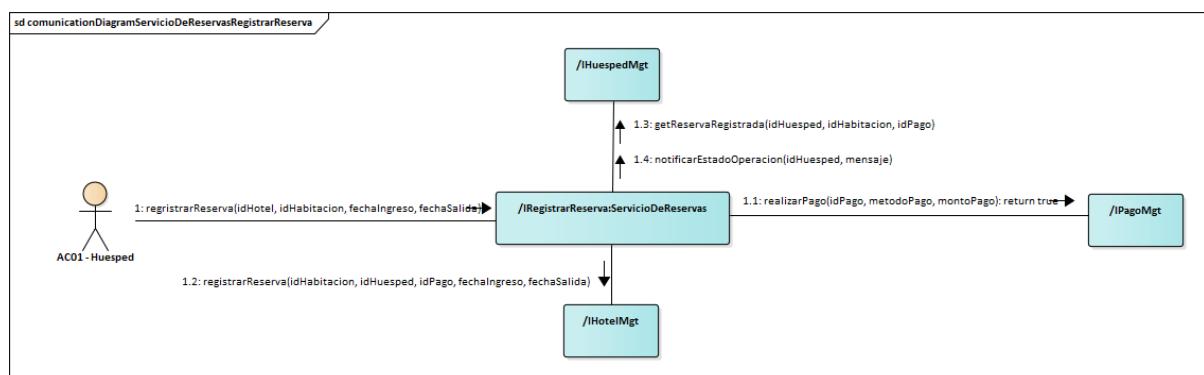
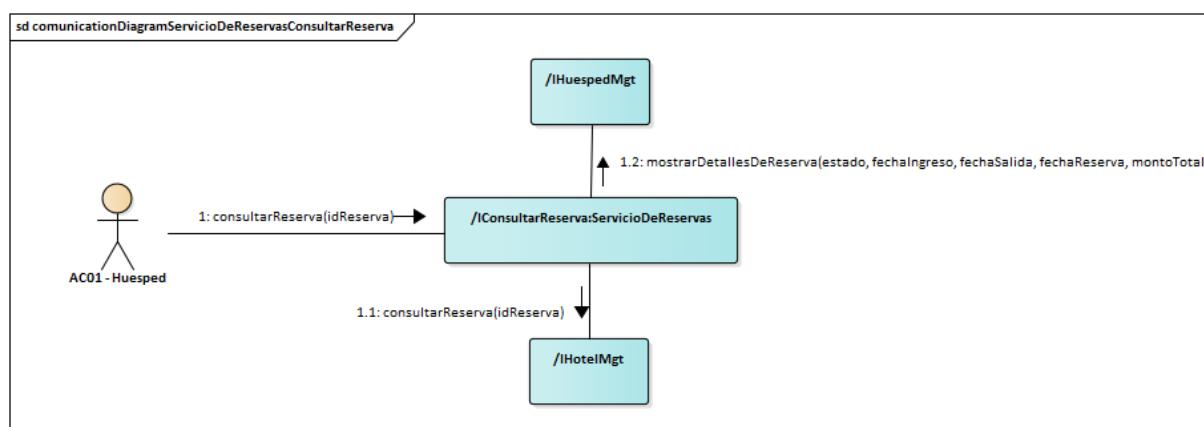
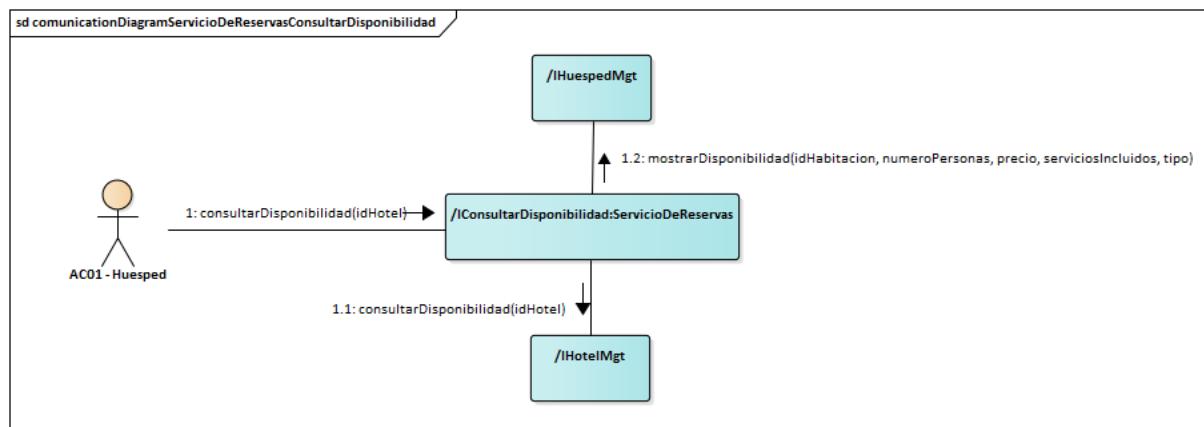
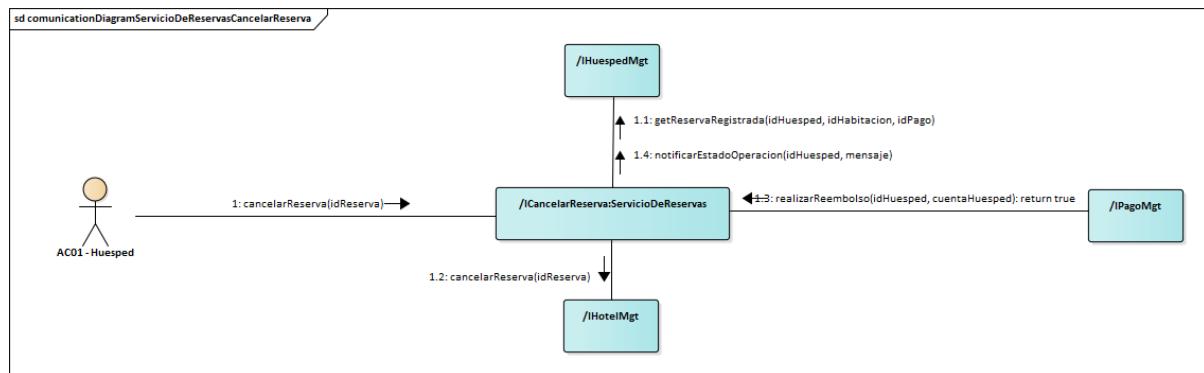




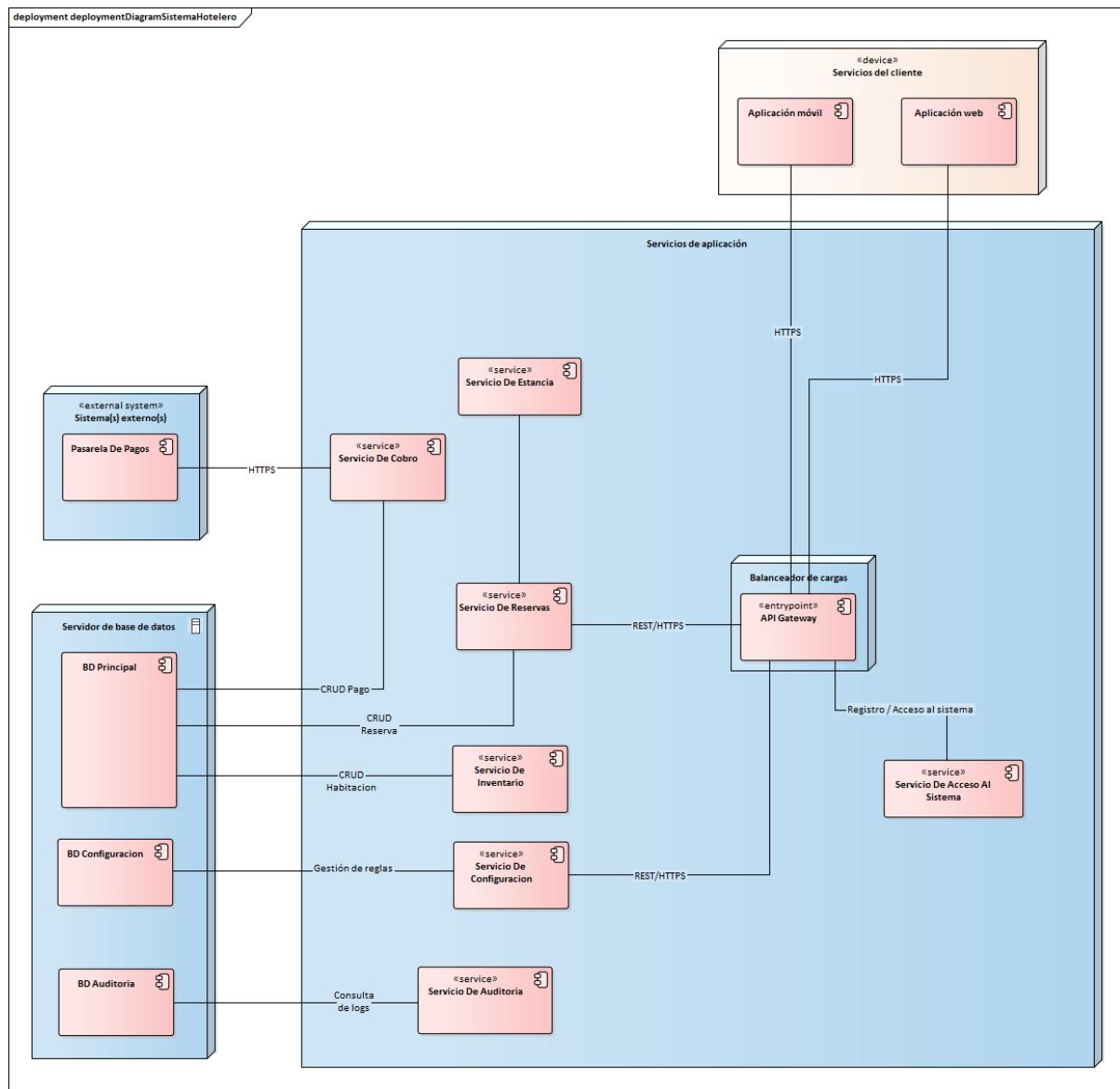
## Servicio de Pago



## Servicio de Reservas



#### 5.4.4. Vista de Despliegue



#### 5.4.5. Estructura y Topología de Despliegue

La infraestructura se organiza en tres niveles físicos principales, separados por fronteras de red (Firewalls y VLANs) para garantizar la seguridad y el rendimiento. Esta separación permite una administración independiente de los recursos y facilita el escalado granular.

1. Nivel de Cliente (Frontend): Representado por los dispositivos de los usuarios finales (Huéspedes y Personal). En este nivel residen los artefactos de presentación: la Aplicación Móvil y la Aplicación Web. Estos clientes son "ligeros" y no contienen lógica de negocio compleja; su función principal es la renderización de la interfaz de usuario y la comunicación segura mediante el protocolo HTTPS con el servidor de aplicaciones.
2. Nivel de Aplicación (Backend Server): Este es el núcleo de procesamiento del sistema, donde reside la lógica de negocio. Se ha optado por un cluster de Servidores de Aplicación gestionado mediante una plataforma de orquestación de contenedores (como Kubernetes).
  - API Gateway / Balanceador de Carga: Actúa como el único punto de entrada (Entrypoint) al sistema desde el exterior. Su función es desacoplar a los clientes de la ubicación física de los servicios, gestionar la terminación SSL (descifrado HTTPS), aplicar límites de tasa y distribuir el tráfico entrante entre los nodos disponibles para evitar cuellos de botella.

- Servicios de Negocio: Los componentes lógicos definidos en la vista anterior (Reservas, Inventario, Cobro) se despliegan aquí como unidades de ejecución independientes (Deployment Units o Contenedores). Esta separación permite que cada servicio escale vertical u horizontalmente según su demanda específica. Por ejemplo, el Servicio de Reservas puede replicarse en múltiples nodos para manejar picos de tráfico estacionales sin consumir los recursos asignados al Servicio de Configuración.
3. Nivel de Datos (Persistencia): Para garantizar la integridad y el rendimiento, la capa de datos se ha segregado físicamente en un cluster de base de datos dedicado.
- BD Principal (Transaccional): Aloja los datos operativos críticos (Reservas, Inventario) en un RDBMS configurado en modo de alta disponibilidad (Master-Replica), garantizando la consistencia ACID requerida por las transacciones de negocio.
  - BD de Auditoría: Cumpliendo con la restricción CON-4, se despliega una instancia de base de datos separada, optimizada para operaciones de escritura secuencial. Esto asegura la inmutabilidad y la trazabilidad forense de todos los registros de seguridad y transacciones financieras, aislándolos de la carga operativa diaria.
  - BD de Configuración: Almacena reglas de negocio y tarifas. Al estar segregada, se puede optimizar con estrategias de caché agresivas para lecturas frecuentes y baja latencia.

#### **5.4.6. Justificación del Estilo Arquitectónico y Satisfacción de QAS**

La elección de esta topología de despliegue distribuida no es accidental, sino que responde directamente a los impulsores arquitectónicos y escenarios de atributos de calidad (QAS) del proyecto:

- Escalabilidad (Soporte a 5,000 Hoteles): La separación física del nivel de aplicación del nivel de datos permite escalar los recursos de cómputo independientemente del almacenamiento. El uso de un Balanceador de Carga frente a los servicios permite añadir nuevos nodos de procesamiento de forma transparente al usuario, gestionando el crecimiento masivo proyectado en la visión del sistema sin interrupciones.
- Rendimiento (Baja Latencia): La comunicación interna entre los servicios dentro del Cluster de Aplicación se optimiza mediante redes locales de alta velocidad y protocolos ligeros, minimizando la latencia de red en las transacciones complejas como la generación atómica de reserva. Además, la segregación de la BD de Configuración reduce drásticamente la carga de lectura sobre la base de datos transaccional principal, liberando recursos para las operaciones de escritura críticas.
- Confiabilidad y Disponibilidad: La arquitectura elimina los puntos únicos de fallo mediante redundancia en todos los niveles. Si el Servicio de Auditoría experimenta una latencia alta o una caída, la separación física de su base de datos evita que esto bloquee las operaciones de la BD Principal. El sistema puede seguir procesando reservas en un modo degradado o poniendo en cola los logs asíncronamente, cumpliendo así con los objetivos de continuidad operativa.
- Seguridad (Integridad y PCI-DSS): El despliegue aísla el Servicio de Cobro dentro de la red interna del servidor, sin acceso público directo. La comunicación con la Pasarela de Pagos externa se realiza estrictamente desde este punto seguro, asegurando que los datos sensibles de las tarjetas nunca toquen el nivel de cliente ni sean almacenados en logs intermedios, facilitando el cumplimiento de la normativa PCI-DSS y protegiendo la integridad financiera del sistema.

negocio.

En conclusión, este mapeo de despliegue proporciona la infraestructura física necesaria para soportar la lógica distribuida del SRC, garantizando que la atomicidad lógica diseñada en los componentes se respalde con una robustez física en el servidor.

# Chapter 6. Registro de decisiones arquitectónicas

## 6.1. Record architecture decisions

Date: 2025-10-23

### 6.1.1. Status

Accepted

### 6.1.2. Context

We need to record the architectural decisions made on this project.

### 6.1.3. Decision

We will use Architecture Decision Records, as [described by Michael Nygard](<http://thinkrelevance.com/blog/2011/11/15/documenting-architecture-decisions>).

### 6.1.4. Consequences

See Michael Nygard's article, linked above. For a lightweight ADR toolset, see Nat Pryce's [adr-tools](<https://github.com/npryce/adr-tools>).

# Chapter 7. Aspectos Transversales (Cross-Cutting Concerns)

En el diseño del Sistema Hotelero, se han identificado tres preocupaciones transversales críticas que deben manejarse de manera uniforme para garantizar la integridad, seguridad y operabilidad del sistema: Seguridad y Auditoría, Gestión de Transacciones y Ma

## 7.1. Seguridad y Auditoría (Trazabilidad)

La seguridad y la trazabilidad son impulsores arquitectónicos primarios (QAS-32, QAS-33, QAS-40 y la restricción CON-3). Dado que el sistema maneja información sensible de huéspedes y operaciones financieras, la seguridad no se trata como un módulo aislado, sino como una capa de protección que envuelve todas las operaciones de negocio.

### 7.1.1. Estrategia y Tácticas

Para satisfacer estos requisitos, se implementa una estrategia de "Defensa en Profundidad" y "Logging Centralizado":

1. Autenticación y Control de Acceso (Nivel Servicio): Aunque existe un ServicioDeAccesoAlSistema y un LoginManager, la validación de permisos se aplica transversalmente. Cada método expuesto en la capa de Servicios (por ejemplo, IReservaService, IConfiguracionService) actúa como una puerta de entrada (Facade) que verifica que la sesión del usuario (Repcionista, Administrador o Auditor) sea válida y tenga los privilegios necesarios antes de delegar la llamada al Manager correspondiente. Esto evita que accesos no autorizados lleguen a la lógica de negocio.
2. Auditoría de Eventos Críticos (Append-Only): Para cumplir con CON-3, se utiliza un mecanismo de intercepción en la capa de Managers. Operaciones críticas como cancelarReserva, realizarCobro o ajustarOverbooking disparan automáticamente un evento de registro hacia el componente ReporteManager.
  - Implementación: Se utiliza el patrón Observer o llamadas directas al ReporteDAO dentro del bloque transaccional de la operación. Esto asegura que si la operación de negocio tiene éxito, el registro de auditoría se persista obligatoriamente. Si la auditoría falla, la operación se revierte, garantizando la integridad de la trazabilidad.

## 7.2. Gestión de Transacciones (Atomicidad)

La visión del sistema establece como prioridad "cero incidentes de dobles reservas o dobles cobros". Esto implica que la gestión de transacciones es un aspecto transversal vital que afecta a los módulos de Reservas, Pagos y Habitaciones simultáneamente.

### 7.2.1. Estrategia y Tácticas

La estrategia se basa en el principio ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), gestionado explícitamente en la Capa de Negocio (Managers):

1. Delimitación de la Transacción (Unit of Work): La responsabilidad de iniciar y finalizar transacciones recae en los Managers (ReservaManager, PagoManager), no en los DAOs. Un caso de uso como "Realizar Reserva" (CU03) implica múltiples escrituras: insertar el cliente, crear el registro de reserva y actualizar el estado de la habitación.
  - Táctica: El Manager abre una transacción a través de la clase ConexionDB (Patrón Singleton), ejecuta las operaciones secuenciales llamando a ReservaDAO y HabitacionDAO, y finalmente realiza el Commit.
2. Manejo de Fallos y Rollback: Si ocurre un error en cualquier punto (por ejemplo, la pasarela de pagos rechaza la tarjeta después de haber bloqueado la habitación), el sistema captura la excepción y ejecuta un Rollback completo. Esto garantiza que el sistema nunca quede en un estado inconsistente.

## 7.3. Manejo de Excepciones y Observabilidad

Para garantizar la Disponibilidad y la Usabilidad (CRN-4, CRN-6), el sistema no puede permitirse fallos silenciosos o mensajes de error crípticos (SQL Errors) mostrados al usuario final. El manejo de excepciones atraviesa desde la base de datos hasta la interfaz de usuario.

### 7.3.1. Estrategia y Tácticas

Se utiliza una estrategia de "Propagación y Enmascaramiento":

1. Captura y Traducción (Capa DAO a Manager): Los DAOs capturan excepciones técnicas de bajo nivel. Estas se envuelven y se relanzan como excepciones de dominio personalizadas (ej. ReservaException, DatabaseConnectionException). Esto desacopla la lógica de negocio de la tecnología de persistencia específica.
2. Manejo y Recuperación (Capa Service/Controller): En la capa superior (\*Controller y \*UI), se implementan bloques globales de captura (try-catch) que interceptan estas excepciones de dominio.
  - Táctica: Dependiendo del tipo de error, el sistema decide si puede recuperarse o si debe informar al usuario. Si es un error irrecuperable, se muestra un mensaje amigable y se registra el Stack Trace técnico en un log interno para el equipo de desarrollo, cumpliendo con la mantenibilidad.

# Chapter 8. Trazabilidad

<b>Driver (ID y Tipo)</b>	<b>Decisión de Diseño</b>	<b>Artefactos Afectados</b>	<b>Estado</b>
QAS-Usabilidad (QAS-01, QAS-04, QAS-12, QAS-14, QAS-16) Tipo: Atributo de Calidad	Patrón Model-View-Controller (MVC) en Cliente Controladores especializados por caso de uso.	Vista Lógica (5.1): Diagrama de Clases.  Elementos: Clases *UI (Ej. LoginUI), Clases *Controller (Ej. LoginController)	Cumple
QAS-Disponibilidad (QAS-03, QAS-05, QAS-13, QAS-21) Tipo: Atributo de Calidad	Estilo Cliente-Servidor en Capas Manejo controlado de excepciones Patrón Service Layer	Vista Lógica (5.1): Diagrama de Clases (Separación Cliente/Servidor).  Elementos: Clases *ServiceImp.  Vista de Implementación (5.4): Diagramas de Componentes.	Cumple
QAS-Rendimiento (QAS-11, QAS-31, QAS-38, QAS-42) Tipo: Atributo de Calidad	Patrón Singleton para ConexionDB (reducir conexiones). Patrón Data Access Object (DAO) (optimizar consultas)	Vista Lógica (5.1): Diagrama de Clases.  Elementos: Clases *DAO (Ej. ReservaDAO), Clase ConexionDB	Cumple
QAS-Seguridad (Integridad) (QAS-06, QAS-17) Tipo: Atributo de Calidad	Capa de Managers (Lógica de negocio centralizada). Delegación a sistema externo (Pasarela de Pagos)	Vista Lógica (5.1) : Clases *Manager (Ej. PagoManager).  Vista de Contexto (7) : DFD Nivel 0/1 (Actor 'Sistema de pago').	Cumple
QAS-Seguridad (Trazabilidad) (QAS-32, QAS-33, QAS-40) Tipo: Atributo de Calidad	Integración de reportes (Clase ReporteManager). Patrón DAO para persistencia de auditoría.	Vista Lógica (5.1) : Clases ReporteManager, ReporteDAO.  Vista de Implementación (5.4) : Componente ServicioDeAuditoria , Interfaces de Auditoría (Ej. IConsultarReporte)	Cumple

<b>Driver (ID y Tipo)</b>	<b>Decisión de Diseño</b>	<b>Artefactos Afectados</b>	<b>Estado</b>
CON-1  Tipo: Restricción	Delegación a Actor Externo (Pasarela de Pagos)	Vista de Contexto (7) : DFD Nivel 0/1.  Vista de Implementación (5.1.3): Diagrama de Objetos (Objeto pasarela_pago).  Casos de Uso (2.3) : Actor 'Sistema de pago'	Cumple
CON-3  Tipo: Restricción	Persistencia de "solo escritura" (append-only) para auditoría.	Vista Lógica (5.1) : Clase ReporteDAO.  Vista de Implementación (5.4) : Componente ServicioDeAuditoria.	Cumple
CRN-1, CRN-3, CRN-5, CRN-6  (Mantenibilidad, Testabilidad, Extensibilidad)  Tipo: Preocupación	Estilo Arquitectónico en Capas Cliente-Servidor.  Patrón DAO y Patrón Service Layer (para desacoplamiento)	Vista Lógica (5.1) (Completa).  Vista de Implementación (5.4) (Diagramas de Componentes e Interfaces)	Cumple
CU01-CU06  (Gestión Huésped)  Tipo: Caso de Uso	Patrones MVC, Service Layer, Manager y DAO.	Vista Lógica (5.1) : Clases ReservaUI, ReservaController, ReservaClientService, IReservaService, ReservaServiceImp, ReservaManager, ReservaDAO  Vista de Implementación (5.4) : Componente ServicioDeReservas , Interfaces de Reserva	Cumple
CU07-CU13  (Gestión Recepción)  Tipo: Caso de Uso	Patrones MVC, Service Layer, Manager y DAO.	Vista Lógica (5.1).  Vista de Implementación (5.4) : Componente ServicioDeEstancia , Componente RecepcionistaMgt , Interfaces ICheckIn, ICheckOut	Cumple

<b>Driver (ID y Tipo)</b>	<b>Decisión de Diseño</b>	<b>Artefactos Afectados</b>	<b>Estado</b>
CU14-CU18 (Gestión Administración )  Tipo: Caso de Uso	Patrones MVC, Service Layer, Manager y DAO	Vista Lógica (5.1).  Vista de Implementación (5.4) : Componentes ServicioDeInventario , ServicioDeConfiguracion , AdministradorMgt , Interfaces IAltaHabitacion, IRegistrarPrecio	Cumple
CU19-CU26 (Gestión Auditoría)  Tipo: Caso de Uso	Patrones MVC, Service Layer, Manager y DAO	Vista Lógica (5.1) : Clases ReporteUI, ReporteController, ReporteServiceImp, ReporteManager, ReporteDAO .  Vista de Implementación (5.4) : Componente ServicioDeAuditoria , AuditorMgt , Interfaces de Auditoría	Cumple

# Chapter 9. Apéndice

## 9.1. Vista de Contexto

