

**UNIVERSIDAD CATOLICA BOLIVIANA “SAN PABLO”  
SEDE TARIJA**

**DEPARTAMENTO DE CIENCIAS DE LA TECNOLOGIA E  
INNOVACION**

**CARRERA INGENIERIA DE SISTEMAS**



**Diplomado en Diseño y Desarrollo de Software 1ra. Versión**

**“Sistema Web para la Administración de Reservas en el  
Hotel Viña del Sur”**

**ESTUDIANTE: CARLOS ANTONIO CRUZ PORTAL.**

**DOCENTE: CHRISTIAN FELIPE VIRREIRA SCHOLVIN.**

*Trabajo Final de Diplomado presentado a consideración de la Universidad  
Católica Boliviana San Pablo Sede Tarija, como requisito para optar al Diploma en  
Diseño y Desarrollo de Software*

**TARIJA-BOLIVIA**

**2025**

## INDICE

|       |   |    |
|-------|---|----|
| 1     | Resumen Ejecutivo .....                                 | 1  |
| 2     | Introducción.....                                       | 2  |
| 2.1   | Planteamiento del problema .....                        | 2  |
| 2.2   | Justificación técnica del sistema a desarrollar.....    | 3  |
| 2.3   | Objetivos .....   | 4  |
| 2.3.1 | Objetivo General .....                                  | 4  |
| 2.3.2 | Objetivos Específicos .....                             | 4  |
| 2.4   | Alcances y limitaciones del proyecto .....              | 4  |
| 2.4.1 | Alcances .....  | 4  |
| 2.4.2 | Limitaciones del proyecto .....                         | 6  |
| 3     | Marco Teórico .....                                     | 8  |
| 3.1   | Introducción .....                                      | 8  |
| 3.2   | Introducción a la Ingeniería de Software .....          | 8  |
| 3.3   | Metodologías ágiles y Prácticas de desarrollo .....     | 8  |
| 3.4   | Domain-Driven Design (DDD).....                         | 8  |
| 3.5   | Arquitectura de aplicaciones y Diseño de sistemas ..... | 9  |
| 3.6   | Principio y Patrones de diseño de software.....         | 9  |
| 3.7   | Verificación y pruebas de software .....                | 10 |
| 3.8   | Tecnologías seleccionadas .....                         | 10 |
| 4     | Desarrollo .....  | 11 |
| 4.1   | Metodología de trabajo.....                             | 11 |
| 4.1.1 | Actores del sistema.....                                | 11 |
| 4.2   | Análisis del dominio.....                               | 12 |
| 4.3   | Artefactos generados con DDD.....                       | 14 |
| 4.3.1 | Lenguaje Ubicuo (Ubiquitous Language) .....             | 14 |

|       |   |    |
|-------|---|----|
| 4.3.2 | Subdominios.....  | 14 |
| 4.3.3 | Bounded Context.....  | 15 |
| 4.3.4 | Entidades, objetos de valor y agregados .....                       | 17 |
| 4.4   | Diseño del sistema.....   | 17 |
| 4.4.1 | Arquitectura del sistema.....                                       | 17 |
| 4.4.2 | Diagrama de clases del Dominio (ReservaContext) .....               | 18 |
| 4.4.3 | Diagrama de secuencias. Crear Reserva.....                          | 18 |
| 4.5   | Patrones de diseño aplicados.....                                   | 19 |
| 4.6   | Base de datos.....  | 21 |
| 4.6.1 | Diseño de la Base de datos .....                                    | 21 |
| 4.6.2 | Relaciones .....  | 21 |
| 4.6.3 | Diagrama de clases del dominio (DDD – Core Boundesd Contexts) ..... | 23 |
| 4.7   | Desarrollo e implementación .....                                   | 28 |
| 4.8   | Pruebas de software.....  | 29 |
| 4.8.1 | Pruebas implementadas .....   | 29 |
| 5     | Funcionalidades implementadas.....                                  | 33 |
| 5.1   | Gestión de usuarios .....   | 33 |
| 5.2   | Gestión de huéspedes .....  | 33 |
| 5.3   | Gestión de habitaciones, tipos de habitaciones .....                | 33 |
| 5.4   | Gestión Reservas .....  | 33 |
| 5.5   | Gestion Servicios.....  | 34 |
| 5.6   | CheckIn / CheckOut .....  | 34 |
| 5.7   | Reportes Recibo .....   | 34 |
| 5.8   | Reportes diarios.....   | 34 |
| 6     | Resultados.....   | 35 |
| 6.1   | Sistema funcional y estable .....                                   | 35 |
| 6.2   | Arquitectura clara y mantenible .....                               | 35 |

|        |                                     |    |
|--------|-------------------------------------|----|
| 6.3    | Calidad de código.....              | 36 |
| 7      | Conclusiones y recomendaciones..... | 37 |
| 7.1    | Conclusiones .....                  | 37 |
| 7.2    | Recomendaciones.....                | 37 |
| 8      | Referencias .....                   | 39 |
| 9      | Anexos.....                         | 40 |
| 10     | Manual de usuario.....              | 40 |
| 10.1   | A quien va dirigido.....            | 40 |
| 10.2   | Propósito.....                      | 40 |
| 10.3   | Contenido.....                      | 40 |
| 10.3.1 | Inicio de sesión.....               | 40 |
| 10.3.2 | Gestión de reservas.....            | 43 |
| 10.3.3 | Check-in .....                      | 45 |
| 10.3.4 | Check-out .....                     | 46 |
| 10.3.5 | Registro de Consumos.....           | 47 |
| 10.3.6 | Emisión de recibos .....            | 47 |

## Índice de figuras

|  |    |
|--|----|
| Figura 1 Diagrama de contexto. Reserva Habitación .....              | 17 |
| Figura 2 Diagrama de Clases. Reserva habitación .....                | 18 |
| Figura 3 Diagrama de secuencia. Crear reserva .....                  | 19 |
| Figura 4 Diagrama E-R. Hotel Viña del Sur .....                      | 21 |
| Figura 5 Modelo E-R Crear Reservas de huésped.....                   | 22 |
| Figura 6 Diagrama de clases ReservasContext y ServiciosContext ..... | 23 |
| Figura 7 Diagrama de Contexto.....                                   | 26 |
| Figura 8 Diagrama de contenedores .....                              | 26 |
| Figura 9 Relaciones entre ReservasContext y ServiciosContext.....    | 27 |
| Figura 10 Prueba unitaria de software .....                          | 29 |
| Figura 11 Prueba de integración (Ingreso) .....                      | 30 |
| Figura 12 Prueba de integración (Confirmación) .....                 | 31 |
| Figura 13 Prueba de Integración (Base de datos) .....                | 31 |

**Índice de tablas**

|   |    |
|---|----|
| Tabla 1 Actores del sistema.....                        | 12 |
| Tabla 2 Lenguaje ubicuo generados con DDD.....          | 14 |
| Tabla 3 Subdominios DDD .....                           | 15 |
| Tabla 4 Patrones de diseño .....                        | 20 |
| Tabla 5 Cardinalidad de tablas .....                    | 21 |
| Tabla 6 Bounded Context. Hotel Viña del Sur.....        | 24 |
| Tabla 7 Tecnologías utilizadas Hotel Viña del Sur ..... | 28 |
| Tabla 8 Pruebas de software.....                        | 29 |

## **1 Resumen Ejecutivo**

Este proyecto se centra en el diseño e implementación de un sistema web destinado a la administración de reservas y servicios en un hotel ubicado en la ciudad de Tarija. La idea surgió de la urgencia por actualizar los procesos de gestión, que en la actualidad se realizan de manera manual a través de llamadas telefónicas, mensajes por redes sociales y registros en planillas. Esta forma de operar no solo restringe la competitividad del establecimiento en un entorno cada vez más digital, sino que también propicia errores durante las épocas de mayor demanda y limita la posibilidad de atraer a un público más amplio.

En su desarrollo, se recurrió a tecnologías ampliamente reconocidas en el sector: para el frontend, HTML5, CSS3 y Bootstrap, con el fin de crear una interfaz responsiva y amigable que se adapte a diversos dispositivos; en el backend, PHP junto al framework CodeIgniter, complementado con MariaDB para el manejo de la base de datos. Esta selección garantiza una solución escalable, fácil de mantener y con un desempeño óptimo para las necesidades hoteleras.

Entre sus principales aportes, el sistema facilitará la centralización de reservas en tiempo real, el registro de consumos por parte de los huéspedes, la emisión de comprobantes digitales y la generación de informes administrativos precisos. De este modo, se pretende minimizar los fallos inherentes al registro manual, acelerar la atención al cliente, mejorar el seguimiento de los ingresos y extender el alcance comercial del hotel a un público internacional.

En última instancia, esta iniciativa no solo pone en práctica los conceptos aprendidos durante el Diplomado en Diseño y Desarrollo de Software, sino que también proporciona una herramienta tecnológica tangible, con un impacto directo y positivo en la eficiencia de la gestión hotelera.

## **2 Introducción**

En la ciudad de Tarija, una de sus principales fortalezas es el turismo. En fechas específicas, como feriados o eventos locales, la demanda de visitantes aumenta considerablemente, y muchos de ellos acuden a plataformas digitales populares entre los viajeros, tales como Airbnb o Booking, para consultar la disponibilidad de habitaciones, los precios y la ubicación de los alojamientos. Sin embargo, no todos los establecimientos hoteleros han logrado adaptarse a esta modernización tecnológica. Hay hoteles y hostales que aún operan de manera tradicional y manual, lo que les impide captar una porción significativa de clientes potenciales, ya que no aprovechan las herramientas digitales para promocionarse y facilitar las reservas.

En este contexto, el Hotel Viña del Sur no solo necesita posicionarse al nivel de su competencia, sino también destacar los múltiples espacios que ofrece a sus huéspedes. Además de las habitaciones disponibles, cuenta con áreas sociales versátiles que podrían destinarse a eventos de gran o pequeña escala. Por ello, surge la necesidad de implementar un sistema web propio que permita promocionar y reservar estos espacios de forma integrada. A diferencia de plataformas como Airbnb o Booking, que se limitan principalmente a la oferta de habitaciones, un sitio web dedicado al hotel podría cubrir esta brecha y atender mejor las expectativas de los clientes, fortaleciendo así su presencia en el mercado turístico local.

### **2.1 Planteamiento del problema**

El Hotel Viña del Sur ofrece una amplia gama de servicios turísticos en la ciudad de Tarija, que incluyen 25 habitaciones en diversas categorías, espacios sociales ideales para eventos, una piscina, sauna, gastronomía de primer nivel y opciones de transporte. Estas instalaciones lo posicionan como un establecimiento con un gran potencial para destacar en el competitivo mercado hotelero local.

Aun así, toda esta operación se lleva a cabo de forma manual, sin el respaldo de un sistema digital unificado. Las reservas se manejan principalmente mediante llamadas telefónicas, mensajes de WhatsApp y registros en planillas, mientras que la promoción se reduce a publicaciones simples en redes sociales. El hotel no se beneficia de plataformas en línea como ser Airbnb o Booking, ni cuenta con un sitio web propio que permita exhibir y gestionar sus servicios de manera eficiente.



Esta dependencia de procesos manuales genera errores en la asignación de habitaciones, pérdida de información en temporadas altas, inconsistencias en el control de ingresos y una visibilidad reducida frente a la competencia. Como resultado, el hotel pierde oportunidades de negocio, debilita su posicionamiento en el sector y compromete la experiencia de sus clientes.

## **2.2 Justificación técnica del sistema a desarrollar**

La gestión manual de reservas y servicios presenta limitaciones: genera errores en temporadas de alta demanda y dificulta la captación de clientes a nivel nacional e internacional. Frente a esta situación, un sistema web resulta la alternativa más adecuada, ya que permite a los usuarios consultar disponibilidad, realizar o consultar reservas.

Para el frontend del sistema Hotel Viña del Sur se han seleccionado tecnologías estándar y ampliamente adoptadas en la industria: HTML5, CSS3 y Bootstrap 5. Estas herramientas permiten construir interfaces web responsivas, accesibles y compatibles con múltiples dispositivos y navegadores modernos. Bootstrap, en particular, facilita la creación de un diseño visual coherente, moderno y centrado en la experiencia del usuario, lo cual resulta fundamental tanto para el personal interno del hotel (recepción, administración) como para futuras extensiones orientadas al cliente. Con estas herramientas es posible mantener un diseño moderno y uniforme, lo cual es clave para captar y fidelizar clientes.

En el backend, se emplea PHP 8.2+ en conjunto con el framework CodeIgniter 4, y como gestor de base de datos, MariaDB. PHP se elige por su madurez, estabilidad, amplia adopción en aplicaciones web empresariales y su excelente rendimiento en entornos de hosting convencionales. CodeIgniter 4, por su parte, implementa de forma clara el patrón arquitectónico Modelo-Vista-Controlador (MVC), lo que favorece una separación sólida de responsabilidades, facilita el mantenimiento del código y promueve la aplicación de buenas prácticas de seguridad (como filtrado de entradas, protección contra inyecciones SQL entre otros).

Esta selección de tecnologías se alinea directamente con el propósito de desarrollar una solución segura, escalable y de bajo mantenimiento, que no solo simplifique la administración interna del hotel, sino que también amplíe su alcance comercial al ofrecer un acceso global y fluido a sus ofertas.

## **2.3 Objetivos**

### **2.3.1 Objetivo General**

Diseñar e implementar un sistema web que optimice la gestión de reservas en el Hotel Viña del Sur, automatizando el control de habitaciones, registros de huéspedes y emisión de comprobantes, con el fin de mejorar la eficiencia operativa y la calidad del servicio ofrecido.

### **2.3.2 Objetivos Específicos**

- Crear una interfaz web clara, fácil de usar y que se vea bien en cualquier dispositivo, usando HTML5, CSS3 y Bootstrap, para que el personal de recepción y administración trabaje sin problemas.
- Mostrar en todo momento qué habitaciones y servicios están disponibles, reservados u ocupados, evitando errores al asignar una habitación que ya está en uso.
- Automatizar todo el proceso de reservas y estancias, desde la selección de fechas hasta el registro del check-in y check-out, actualizando automáticamente los estados y calculando el monto total a pagar.
- Permitir registrar servicios adicionales y consumos durante la estadía (como desayunos, masajes o bebidas) y que estos se sumen de forma automática al total de la cuenta del huésped.
- Desarrollar el sistema con buenas prácticas (usando DDD y XP) y asegurar que los datos se guarden de forma correcta y segura, con validaciones en el código y en la base de datos, para evitar inconsistencias o errores.

## **2.4 Alcances y limitaciones del proyecto**

### **2.4.1 Alcances**

#### **2.4.1.1 Roles de usuarios**

- **Administrador**
  - Registrar, modificar y eliminar habitaciones, espacios sociales y servicios ofrecidos.
  - Crear perfiles de empleados con diferentes roles (recepción, servicio, limpieza, entre otros).

- Gestionar la disponibilidad general del hotel.
- Monitorear la actividad de los usuarios del sistema mediante reportes de auditoría.
- Acceso completo a todos los módulos de la plataforma.
- **Recepción**
  - Registrar check-in y check-out de huéspedes.
  - Asignar habitaciones según disponibilidad.
  - Gestionar reservas confirmadas, cancelaciones y reprogramaciones.
  - Registrar los servicios consumidos durante la estadía.
  - Generar comprobantes de pago y facturas.
- **Personal de servicio**
  - Cargar en el sistema los servicios adicionales utilizados por los huéspedes (alimentación, lavandería, spa, transporte, etc.).
  - Reportar incidencias relacionadas con habitaciones o espacios sociales.
  - Acceso limitado únicamente a las funciones relacionadas con el registro de consumo de servicios.

#### **2.4.1.2 Funcionalidades del sistema**

- Gestión de reservas en línea con verificación de disponibilidad.
- Administración de habitaciones y categorización (estándar, matrimonial, suite, presidencial).
- Registro de espacios sociales (salones, piscina, sauna, salas de reuniones).
- Control de servicios adicionales ofrecidos.
- Gestión de usuarios y control de roles.
- Reportes administrativos: ocupación por fecha, ingresos por servicio, historial de clientes.

### 2.4.2 Limitaciones del proyecto

El desarrollo del sistema web se ha delimitado considerando las condiciones técnicas, económicas y regulatorias actuales, así como el alcance establecido para el trabajo final del diplomado. En este sentido, las principales restricciones son las siguientes:

- a) Procesamiento de pagos en línea: no se habilitarán transacciones en dólares debido a las restricciones impuestas por el sistema bancario nacional respecto a operaciones electrónicas internacionales. Asimismo, no se contempla la integración de pasarelas de pago externas durante la etapa de desarrollo.
- b) El sistema se implementa como una aplicación web responsiva, accesible desde cualquier navegador moderno en computadoras, tabletas o teléfonos móviles. Quedan fuera del alcance del presente proyecto el desarrollo de aplicaciones nativas para Android, iOS u otras plataformas móviles, así como la publicación en tiendas digitales (Google Play, App Store, etc.).
- c) No se implementa módulos de notificación de envío automático ni de correos electrónicos ni mensajes SMS. Las confirmaciones de reservas y comprobantes se entregarán únicamente dentro del sistema web.
- d) Integración con sistemas externos: no se incorporará la conexión con el sistema oficial de facturación en línea regulado por la ASFI<sup>1</sup> ni con otros servicios de SINI<sup>2</sup>.
- e) Módulos administrativos adicionales: funciones relacionadas con inventario de cocina, control de suministros u otros procesos internos ajenos a la gestión de reservas quedarán fuera del alcance de la presente aplicación.

Estas limitaciones no afectan el propósito principal del proyecto, que es proporcionar una plataforma segura y eficiente para la administración de reservas, espacios sociales y

---

<sup>1</sup> ASFI. La Autoridad de Supervisión del Sistema Financiero, es la institución del Estado Plurinacional de Bolivia que regula y supervisa el funcionamiento de las entidades bancarias, cooperativas, mutuales, fondos financieros, y entidades que operan con valores y seguros, las que conforman el Sistema Financiero Boliviano.

<sup>2</sup> SIN. El proceso de fiscalización de impuestos es un procedimiento que lleva a cabo el Servicio de Impuestos Nacionales en Bolivia. Su objetivo es verificar que los contribuyentes cumplan con las responsabilidades tributarias de manera precisa.

servicios complementarios, sino que marcan el marco realista bajo el cual se ha diseñado la solución.

### **3 Marco Teórico**

#### **3.1 Introducción**

El sistema web de reservas para el Hotel Viña del Sur no se da por azar sino que es el fruto de un proceso de aprendizaje y aplicación práctica de los contenidos revisados en el Diplomado en Diseño y Desarrollo de Software. En este capítulo, se describen los principales fundamentos teóricos, metódicos y tecnológicos que sustentan el desarrollo del proyecto. El objetivo es vincular el aprendizaje teórico y aplicado por el diplomado a la realidad del caso, con la finalidad de que la solución encontrada sea congruente, útil y de largo plazo.

#### **3.2 Introducción a la Ingeniería de Software**

La Ingeniería de Software es la disciplina que estudia y aplica principios, métodos y herramientas para el desarrollo de sistemas de calidad. Pressman define a este proceso como la aplicación de un enfoque sistemático y disciplinado para el desarrollo, operación y mantenimiento del software. (Pressman, 2008)

En el caso del hotel, la Ingeniería de Software permitirá organizar los requerimientos y entender qué es lo que requiere el negocio. Para ello, se estudia cuál es la evolución y relación de las técnicas que se puede aplicar con más eficacia para minimizar el riesgo de fracaso de trabajar.

#### **3.3 Metodologías ágiles y Prácticas de desarrollo**

En la actualidad, las metodologías ágiles representan un estándar en la gestión de proyectos de software. Frameworks como Scrum, Kanban y también XP facilitan trabajar con entregables parciales, adaptarse a cambios y mantener una comunicación fluida entre desarrolladores y usuarios (Beck, Grenning, Beedle, & Highsmith, 2001).

Para el Hotel Viña del Sur, aplicar una metodología ágil significa que el sistema puede construirse de manera iterativa, recibiendo retroalimentación temprana del personal (recepcionistas y administradores), y ajustando las funcionalidades según sus necesidades reales.

#### **3.4 Domain-Driven Design (DDD)**

El Diseño Guiado por el Dominio, planteado por Evans, propone que el software debe construirse tomando como base el modelo de negocio. Esto implica identificar las entidades centrales, relaciones y procesos claves del dominio. (Evans, 2004)

En el sistema web hotel Viña del Sur, las entidades principales que se identificaron son:

Cliente: persona que realiza una reserva.

Habitación: unidad que se asigna a un cliente.

Reserva: proceso de asignación y control de disponibilidad.

Tarifa: valor económico según tipo de habitación o temporada.

Rol de usuario: administrador, recepcionista o cliente.

Al aplicar DDD, se construye un lenguaje común entre el personal del hotel y los desarrolladores, lo cual asegura que el sistema refleje fielmente la realidad del negocio.

### **3.5 Arquitectura de aplicaciones y Diseño de sistemas**

El diseño arquitectónico define la estructura global del sistema. Para este proyecto se empleará la arquitectura en capas, en concreto el patrón Modelo-Vista-Controlador (MVC), que separa la lógica de negocio, la interfaz de usuario y el acceso a datos. (Alonso Amo, y otros, 2005)

Además, se integran patrones de diseño que permiten mejorar la calidad del software:

Singleton para la gestión de la conexión a la base de datos.

Repositorio para el manejo centralizado de la persistencia de datos.

Factory Method para la creación controlada de objetos complejos. (Gamma, 2013)

Estos enfoques aseguran que el sistema sea escalable, mantenible y adaptable a futuras necesidades del hotel.

### **3.6 Principio y Patrones de diseño de software**

El diseño de software no solo busca que el sistema funcione, sino que sea robusto y fácil de extender. Los principios SOLID, por ejemplo, ayudan a organizar las clases y módulos de manera que sean reutilizables y fáciles de mantener.

Aplicados al sistema del hotel, estos principios permiten que módulos como “reservas” o “gestión de clientes” puedan ampliarse en el futuro sin afectar al resto del sistema.

### 3.7 Verificación y pruebas de software

Un sistema de reservas debe ser confiable, ya que cualquier error podría causar sobreventas de habitaciones o pérdida de datos de clientes. Por ello, las pruebas de software son esenciales. (Myers , 2004)

De acuerdo con Myers, existen diferentes tipos de pruebas:

- **Pruebas unitarias:** validan funciones específicas, como calcular el costo de una reserva.
- **Pruebas de integración:** verifican la interacción entre módulos, por ejemplo, reservas y notificaciones.
- **Pruebas funcionales:** comprueban que el sistema cumpla con los requerimientos planteados.

Estas pruebas serán aplicadas durante todo el proceso de desarrollo, garantizando calidad y confianza en la implementación.

### 3.8 Tecnologías seleccionadas

Las tecnologías utilizadas en el proyecto responden a criterios de accesibilidad, documentación y compatibilidad:

- **Frontend:** HTML5, CSS3, JavaScript y Bootstrap, para crear interfaces amigables.
- **Backend:** PHP con framework CodeIgniter o Laravel, que permiten trabajar con MVC y aplicar buenas prácticas.
- **Base de datos:** MariaDB , una de las más usadas en entornos web.
- **Servidor web:** Apache en XAMPP o MAMP, para pruebas locales.
- **Control de versiones:** Git y GitHub, para llevar registro del avance y permitir colaboración.

Estas herramientas son ampliamente conocidas y facilitan el desarrollo de un sistema funcional en el tiempo disponible del diplomado.



## 4 Desarrollo

### 4.1 Metodología de trabajo

Para asegurar una perspectiva enfocada en el ámbito de los negocios hoteleros, se ha implementado el Extreme Programming (XP) como método ágil de desarrollo, que se complementa con principios de Domain-Driven Design (DDD).

**XP** se seleccionó por su énfasis en:

- Entregas regulares de valor funcional.
- Ejercicios de desarrollo de alta calidad (refactorización continua, integración permanente, TDD).
- Colaboración cercana entre el equipo técnico y los especialistas del sector (los interesados en el hotel).

Principales prácticas XP aplicadas:

- Programación en pareja (Pair Programming)
- Desarrollo enfocado en las pruebas (Test-Driven Development - TDD)
- Integración continua (CI)
- Refactorización ininterrumpida
- Historias de usuario como unidad de trabajo
- Metáfora del sistema (con el objetivo de armonizar el lenguaje técnico y el lenguaje de negocio)

Colaboración con DDD:

- Se realiza modelado colaborativo con expertos del dominio (dueños administradores del hotel, recepcionistas, gerentes, personal de limpieza, personal de servicio).
- Se define y mantiene el Lenguaje Ubicuo (Ubiquitous Language) que se refleja tanto en el código como en la documentación y en las conversaciones del equipo.

#### 4.1.1 Actores del sistema

| Actor   | Interacción con el sistema   |
|---------|--|
| Huésped | Realiza reservas online<br>Consulta disponibilidad<br>Paga reservas<br>Solicita servicios durante la estadía |

|                         |   |
|-------------------------|---|
| Recepcionista           | Registra check-in/check-out<br>Modifica/cancela reservas<br>Asigna habitaciones<br>Registra consumos manuales           |
| Administrador           | Configura tipos de habitación y precios<br>Gestiona usuarios del sistema<br>Genera reportes (ocupación, ingresos, etc.) |
| Personal de Servicios   | Registra consumos en tiempo real (ej.: masaje, cena)<br>Actualiza estado de servicios (disponible/ocupado)              |
| Pasarela de Pagos       | Autoriza/captura pagos<br>Notifica estado de transacciones  |
| Sistema de Contabilidad | Recibe datos de facturas y pagos diarios  |

Tabla 1 Actores del sistema

## 4.2 Análisis del dominio

El análisis del dominio se realizó mediante talleres con gente relacionada con la administración y manejo del hotel, identificando los conceptos clave del dominio hotelero: reservas, habitaciones, huéspedes, servicios, facturación, etc. Los cuadros siguiente reflejan el análisis de los dominios establecidos para realizar las reservas de habitaciones.

|                      |  |
|----------------------|--|
| Reserva              | Representa el compromiso inicial entre el hotel y un cliente para ocupar una o más habitaciones en un rango específico de fechas.  |
| Reglas de negocio    | No se puede reservar una habitación ya asignada en fechas superpuestas.<br>Toda reserva debe estar asociada a una habitación válida y a un usuario del sistema (quien la registra).<br>El estado de la reserva evoluciona: <i>pendiente</i> → <i>confirmada</i> → <i>finalizada</i> (o <i>cancelada</i> ). |
| Atributos relevantes | Fechas de inicio y fin, cantidad de personas, monto total, método de pago, estado.   |

|                      |   |
|----------------------|---|
| Habitación           | Es el recurso físico principal que ofrece el hotel. Cada habitación pertenece a un tipo de habitación (Suite Tarijeña, Doble Andina, etc.), lo que determina su tarifa y capacidad.                               |
| Reglas de negocio    | El estado de la habitación cambia dinámicamente: <i>disponible</i> → <i>reservada</i> → <i>ocupada</i> → <i>disponible</i> (o <i>mantenimiento</i> ).<br><br>No se puede asignar una habitación en mantenimiento. |
| Atributos relevantes | Número, piso, tipo, estado actual, observaciones.   |

|                   |   |
|-------------------|---|
| Huésped           | Persona que se hospeda en el hotel. A diferencia de un “cliente ocasional”, un huésped se registra formalmente al realizar su primer pago               |
| Reglas de negocio | Un huésped se crea a partir de una reserva confirmada y pagada. Puede tener datos de identidad y lugar de origen (importante para reportes turísticos). |
| Relación          | vinculado a una estancia (no directamente a la reserva).  |

|                   |  |
|-------------------|--|
| Estancia          | Representa la ejecución real de una reserva: el periodo efectivo en que el huésped ocupa la habitación   |
| Reglas de negocio | Comienza con el check-in y termina con el check-out.<br><br>Durante la estancia, se pueden registrar consumos adicionales (servicios, productos).<br><br>Solo puede existir una estancia activa por huésped en un momento dado |
| <b>Atributos</b>  | fechas reales de ingreso y salida, estado, monto final.  |

|                   |  |
|-------------------|--|
| Servicios         | Ofertas complementarias que el hotel brinda, tanto dentro como fuera de la habitación (spa, salones, desayunos, traslados, etc.).  |
| Reglas de negocio | Algunos servicios tienen disponibilidad horaria por día de la semana (ej: sauna solo de 10:00 a 22:00).<br><br>Pueden asociarse a una reserva (antes del check-in) o a una estancia (durante la estadía).<br><br>Se miden en unidades: <i>noche, hora, sesión, plato</i> . |
| Tipos             | Masaje Relajante (sesión), Salón Grande (hora), Desayuno Premium (plato).  |

|                              |   |
|------------------------------|---|
| Consumos durante la estancia | Registro de servicios o productos utilizados por el huésped después del check-in. |
| Reglas de negocio            | Todo consumo debe estar vinculado a una estancia activa.                          |

|  |   |
|--|---|
|  | <p>El subtotal se calcula automáticamente según el precio unitario y la cantidad o duración.</p> <p>Es registrado por un usuario del sistema (repcionista, mesero, etc.).</p> |
|  |   |

|                          |  |
|--------------------------|--|
| Facturación /<br>Recibos | <p>Al finalizar la estancia, se genera un recibo que resume todos los cargos Tarifa de la habitación</p> <ul style="list-style-type: none"> <li>• Servicios reservados</li> <li>• Consumos durante la estadía</li> <li>• Impuestos y descuentos (si aplica)</li> </ul> |
| Regla clave              | el recibo se emite una sola vez por estancia y no puede modificarse  |

### 4.3 Artefactos generados con DDD

#### 4.3.1 Lenguaje Ubicuo (Ubiquitous Language)

| Artefacto             | Descripción   |
|-----------------------|---|
| Reserva               | Acuerdo entre el huésped y el hotel para ocupar una o más habitaciones en un rango de fechas.                           |
| Habitación            | Unidad física del hotel con características específicas (tipo, capacidad, precio).                                      |
| Huésped               | Persona que se aloja o reserva en el hotel.   |
| Check-in              | Proceso de registro de llegada del huésped y asignación de habitación.  |
| Check-out             | Proceso de salida del huésped y cierre de la estadía.   |
| Servicio<br>Adicional | Consumo extra (spa, restaurante, lavandería) asociado a una reserva, salones de reuniones, de fiestas, transporte, etc. |

Tabla 2 Lenguaje ubicuo generados con DDD

#### 4.3.2 Subdominios

Siguiendo los principios de Domain-Driven Design (Evans, 2003), se identificaron y delimitaron los contextos principales del sistema, agrupando funcionalidades según su

relevancia en el dominio hotelero. Cada contexto posee su propio modelo, reglas de negocio y lenguaje ubicuo, lo que permite mantener la cohesión interna y reducir el acoplamiento entre módulos.

| Subdominio               | Contexto   | Descripción  |
|--------------------------|------------|--|
| Gestión de Reservas      | Core       | Lógica principal del negocio: creación, modificación, cancelación de reservas. |
| Gestión de Habitaciones  | Core       | Disponibilidad, asignación y mantenimiento de habitaciones.                    |
| Emisión de recibos       | Supporting | Generación de recibos y cálculo de montos.                                     |
| Autenticación y Usuarios | Generic    | Control de acceso al sistema (recepción, administración).                      |

Tabla 3 Subdominios DDD

#### 4.3.3 Bounded Context

En base a los subdominios establecidos en el punto anterior, se puede realizar 1 contexto delimitado para ReservasContext, HabitacionesContext y principalmente para la emisión de RecibosContext.

- **ReservasContext (Core)**

- Entidades:** Reserva, Huésped, Estancia
- Servicios del dominio**
  - ReservaService, gestiona la creación, modificación y cancelación de reservas.
  - DisponibilidadService, valida la disponibilidad de habitaciones en un rango de fechas.
- Responsabilidades**
  - Garantizar que no se asignen habitaciones ya reservadas u ocupadas.
  - Vincular cada reserva confirmada con un huésped y, posteriormente, con una estancia al hacer check-in.

- **HabitacionesContext (Core)**

- Entidades:** Habitacion, TipoHabitacion
- Servicios del dominio**

- a. DisponibilidadService, gestiona el estado de las habitaciones (disponible, reservada, ocupada, mantenimiento) y su asignación.
- c. **Responsabilidades**
  - a. Mantener actualizado el estado de cada habitación según el flujo de reservas y estancias.
  - b. Proveer información en tiempo real sobre disponibilidad para el contexto de reservas.
- d. **RecibosContext (Supporting)**
  - a. **Entidades:** Recibo, Estancia (como fuente de datos), ConsumoEstancia
  - b. **Servicios del dominio**
    - RecibosService: genera el recibo al finalizar una estancia, calculando el monto total a partir de:
      - Tarifa de la habitación (según tipo y duración)
      - Servicios reservados (reserva\_servicio)
      - Consumos registrados durante la estadía (consumo\_estancia)
  - c. **Responsabilidades**
    - Emitir un recibo único por estancia, con desglose de conceptos y método de pago.
    - Asegurar la trazabilidad del monto final y su consistencia con los registros del sistema.

Asimismo se elaboró el diagrama de contexto (Context Map) siguiendo los lineamientos establecidos por DDD, lo que permite establecer principalmente la reserva de habitaciones del Hotel Viña del Sur.

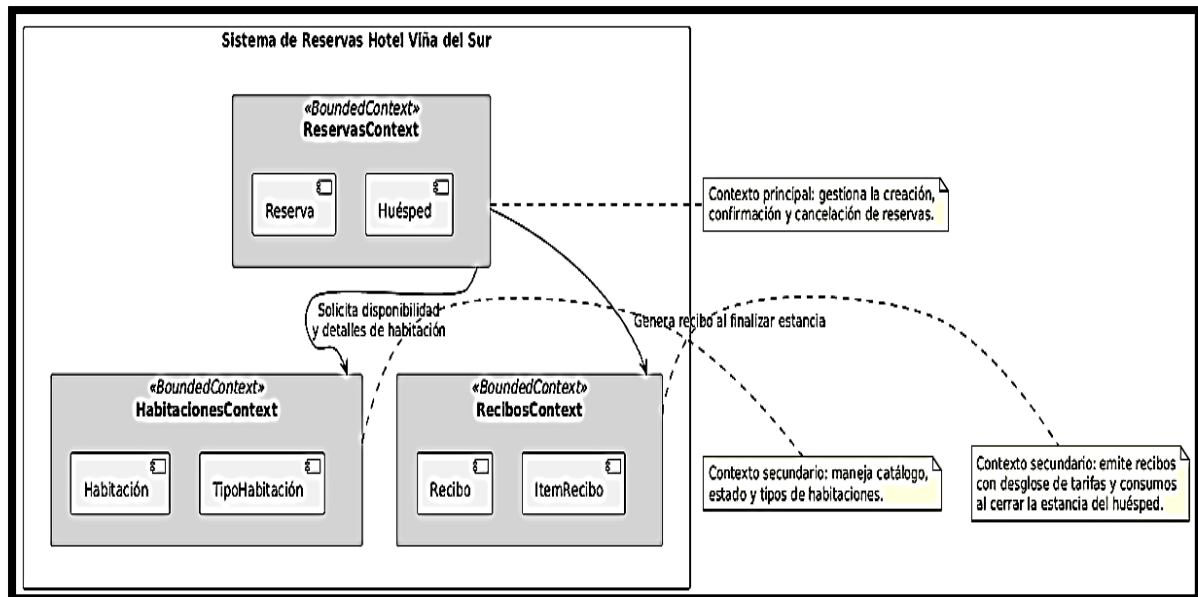


Figura 1 Diagrama de contexto. Reserva Habitación

#### 4.3.4 Entidades, objetos de valor y agregados

A partir de los contextos establecidos, se puede establecer las características de los campos de la tabla de reservas que deben cumplir las siguientes restricciones.

- **Entidad Reserva**
  - ID: reservaId
  - Propiedades: fechas, estado, huésped, habitación asignada
  - Comportamientos: confirmar(), cancelar(), realizarCheckIn()
- **Objeto de Valor RangoFechas**
  - Inmutable, compara fechas de entrada y salida
- **Agregado Raíz: Reserva** (controla la coherencia de una reserva y que no exista solapamientos)

### 4.4 Diseño del sistema

#### 4.4.1 Arquitectura del sistema

El sistema sigue una arquitectura basada en el patrón MVC de CodeIgniter 4, enriquecida con principios de Domain-Driven Design (DDD). Se ha organizado el código en

capas lógicas: dominio, aplicación, infraestructura y presentación y para separar la lógica de negocio del código técnico.

Se han aplicado buenas prácticas como la extracción de servicios, validación de reglas de negocio en el dominio y encapsulamiento de entidades, en línea con los objetivos de DDD y las prácticas de XP.

#### 4.4.2 Diagrama de clases del Dominio (ReservaContext)

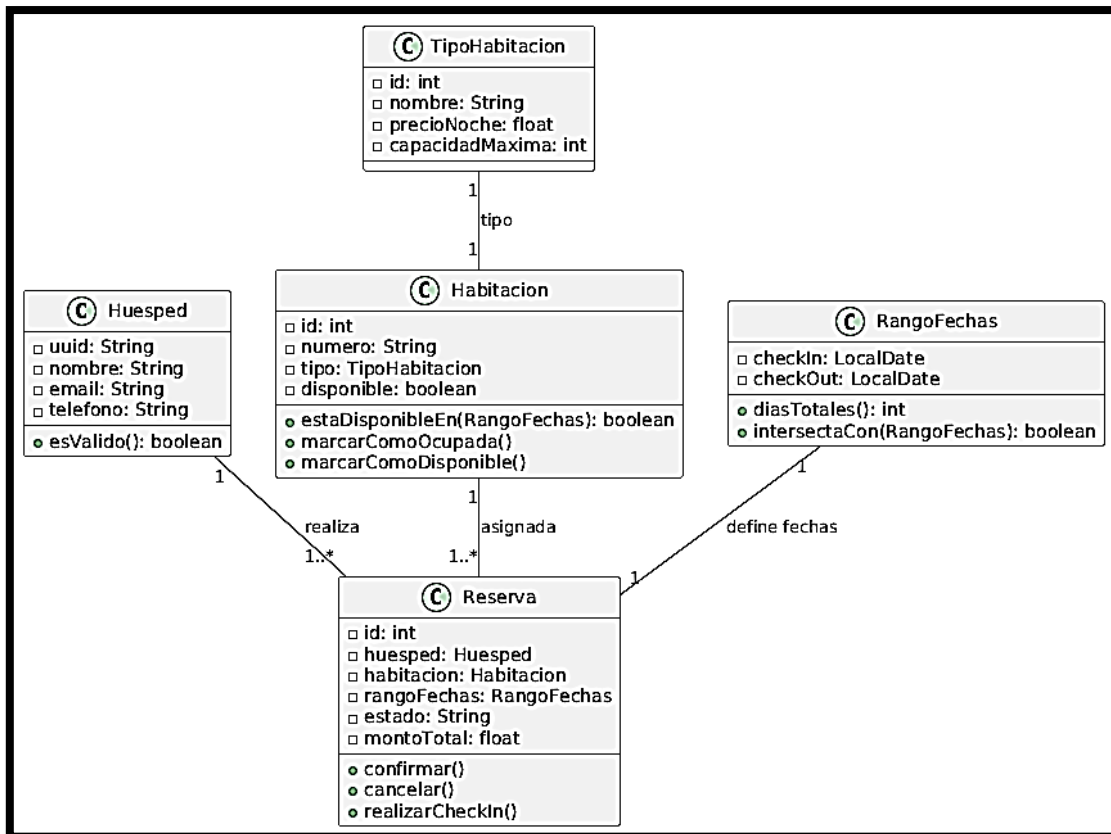


Figura 2 Diagrama de Clases. Reserva habitación

#### 4.4.3 Diagrama de secuencias. Crear Reserva

Según el desarrollo del sistema Hotel Viña del Sur, se han identificado los siguientes actores y componentes que ingresan a la secuencia de Creación de Reservas, los que han sido implementados para que se puedan realizar Reservas.

- Huésped (cliente web)
- ReservaController (controlador MVC)
- DisponibilidadService (servicio de dominio)
- ReservaService (servicio de dominio)
- HabitacionRepository (repositorio DDD)
- ReservaRepository (repositorio DDD)



Y las tablas involucradas para registrar los datos relacionados con la reserva de habitaciones y de servicios son:

- Base de datos (reserva, habitacion, tipo\_habitacion, usuario)

La siguiente figura muestra la secuencia de acciones que deben realizarse para establecer una reserva de habitaciones y de servicios

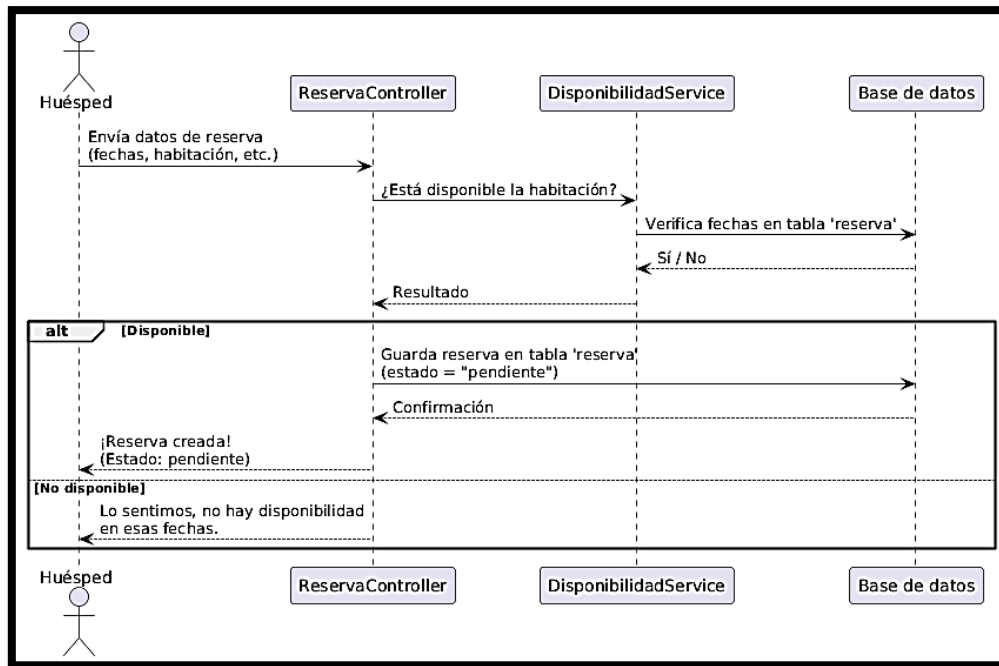


Figura 3 Diagrama de secuencia. Crear reserva

#### 4.5 Patrones de diseño aplicados

La aplicación de los patrones de diseño responde a una decisión de resolver problemas concretos del dominio hotelero con elegancia y escalabilidad. El patrón **Repository** evita que la lógica de negocio dependa de consultas SQL directas, facilitando futuras migraciones de base de datos. El **Factory** asegura que toda reserva se cree en un estado válido, reduciendo errores por datos incompletos. El **Specification** centraliza reglas complejas como la disponibilidad de habitaciones, haciéndolas reutilizables y fáciles de probar. El **CQRS** mejora el rendimiento al optimizar consultas frecuentes (como la disponibilidad) por separado de las operaciones transaccionales (como crear una reserva). Y finalmente, el **Observer** permite extender funcionalidades futuras como el envío de correos sin modificar el núcleo del sistema. (Gonzalez Candia, 2015)

Estos patrones permiten una generación de código limpio, testeable y preparado para evolucionar, cumpliendo con los estándares esperados en un proyecto académico de alto nivel.

La tabla siguiente muestra la descripción de cada uno de los patrones elegidos en el desarrollo del sistema Hotel Viña del Sur.

| <b>Patrón</b>        | <b>Aplicación en el Sistema</b>  |
|----------------------|--|
| <b>Repository</b>    | Abstrae el acceso a la base de datos. El controlador, ReservaController encapsula operaciones CRUD sobre la tabla reserva, permitiendo que la lógica de negocio no dependa directamente de SQL.                            |
| <b>Factory</b>       | Centraliza la creación de objetos complejos. El ReservaController valida fechas, calcula montos y asegura que toda reserva se inicialice con un estado consistente (pendiente) y datos completos antes de persistirse.     |
| <b>Specification</b> | Modela reglas de negocio reutilizables. La clase ReservaModel encapsula la lógica para verificar si una habitación está libre en un rango de fechas, facilitando su uso en validaciones y búsquedas.                       |
| <b>CQRS</b>          | Separa operaciones de escritura y lectura: CrearReserva, CancelarReserva (modifican estado). Consultas: ListarReservasPorHuésped, ObtenerDisponibilidad (solo leen datos). Esto mejora el rendimiento y la mantenibilidad. |
| <b>Observer</b>      | Notifica eventos de forma desacoplada. Al confirmarse una reserva, se dispara una notificación (por ejemplo, envío de correo electrónico al huésped) sin que el servicio de reservas conozca los detalles del envío.       |

Tabla 4 Patrones de diseño

## 4.6 Base de datos

### 4.6.1 Diseño de la Base de datos

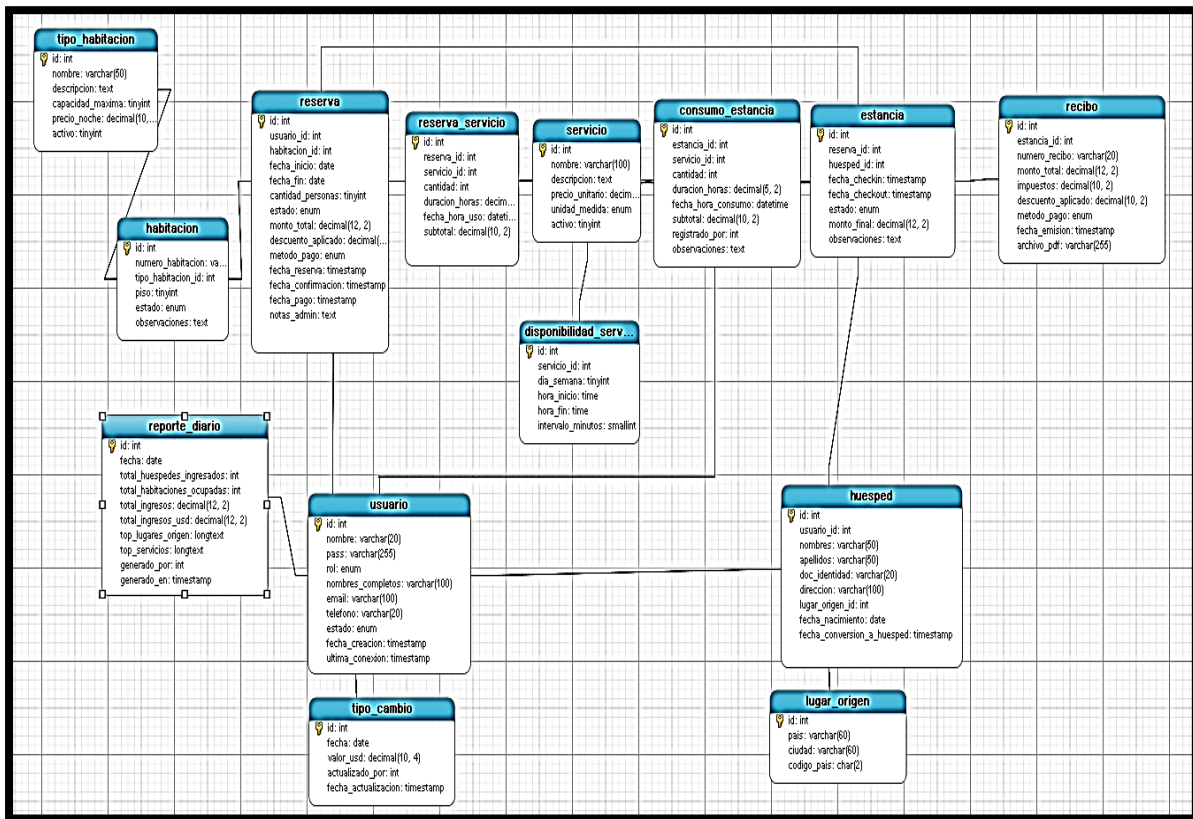


Figura 4 Diagrama E-R. Hotel Viña del Sur

### 4.6.2 Relaciones

| Relación                    | Cardinalidad | Descripción  |
|-----------------------------|--------------|--|
| <b>Huésped → Reserva</b>    | 1:N          | Un huésped puede realizar muchas reservas; cada reserva pertenece a un solo huésped.   |
| <b>Reserva → Habitación</b> | N:1          | Una reserva asigna exactamente una habitación; una habitación puede estar en muchas reservas (en distintas fechas).  |
| <b>Reserva → Recibo</b>     | 1:0..1       | Una reserva confirmada y finalizada genera un recibo al hacer check-out. No toda reserva tiene recibo (ej: canceladas).  |
| <b>Reserva ↔ Servicio</b>   | N:M          | Una reserva puede incluir múltiples servicios (masajes, salones, etc.), y un servicio puede estar en múltiples reservas. Esta relación se implementa en la tabla intermedia reserva_servicio (implícita en el diagrama). |

Tabla 5 Cardinalidad de tablas

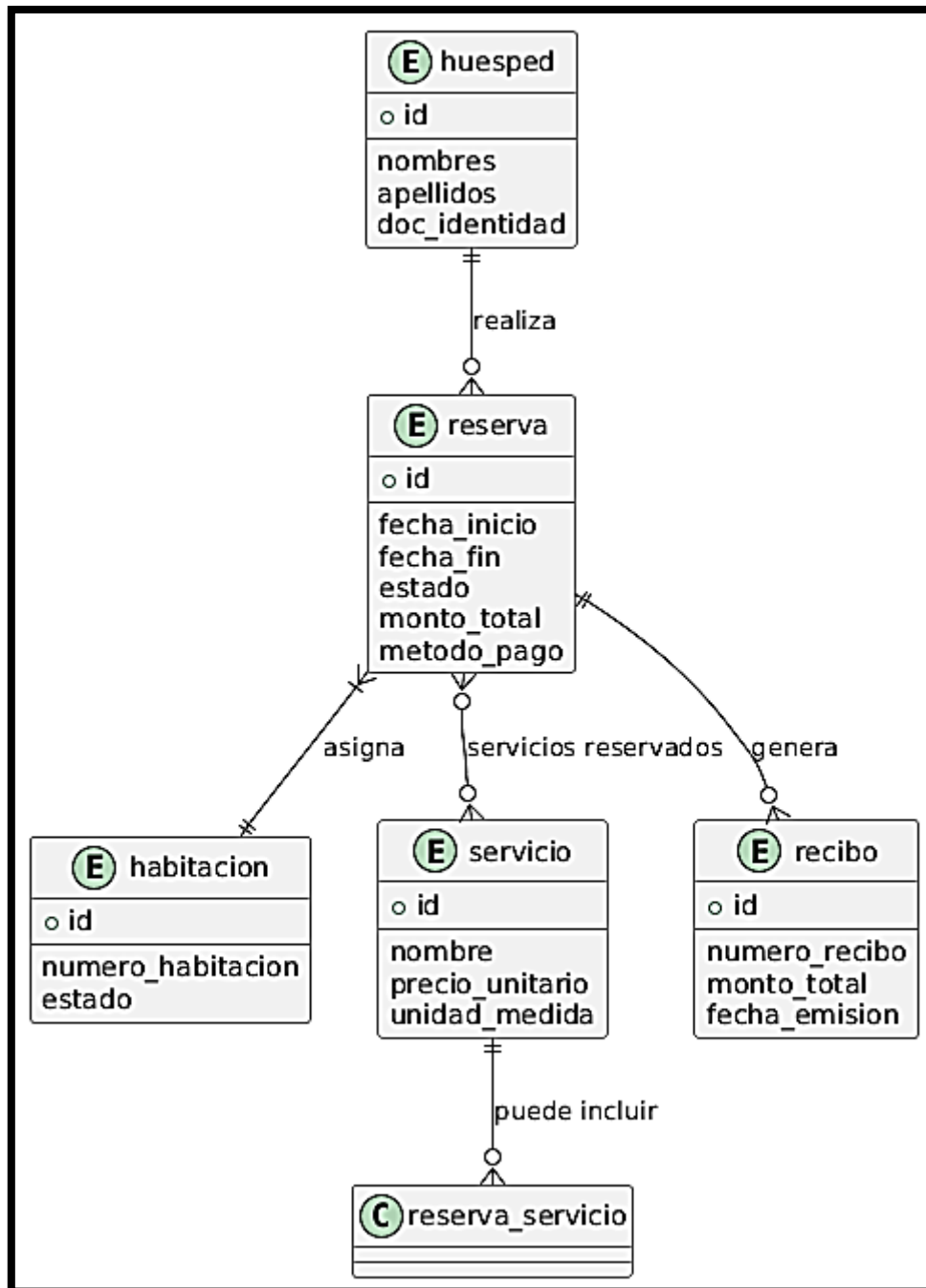


Figura 5 Modelo E-R Crear Reservas de huésped

A continuación, se presenta la documentación técnica del sistema del Hotel Viña del Sur, desarrollada bajo los lineamientos de Domain-Driven Design (DDD) y la metodología ágil Extreme Programming (XP). Cada sección incluye una descripción detallada y los diagramas relevantes, siguiendo buenas prácticas de arquitectura de software y diseño orientado al dominio.

#### 4.6.3 Diagrama de clases del dominio (DDD – Core Boundesd Contexts)

Basado en el modelo relacional y alineado con DDD, se identifican tres Bounded Contexts principales:

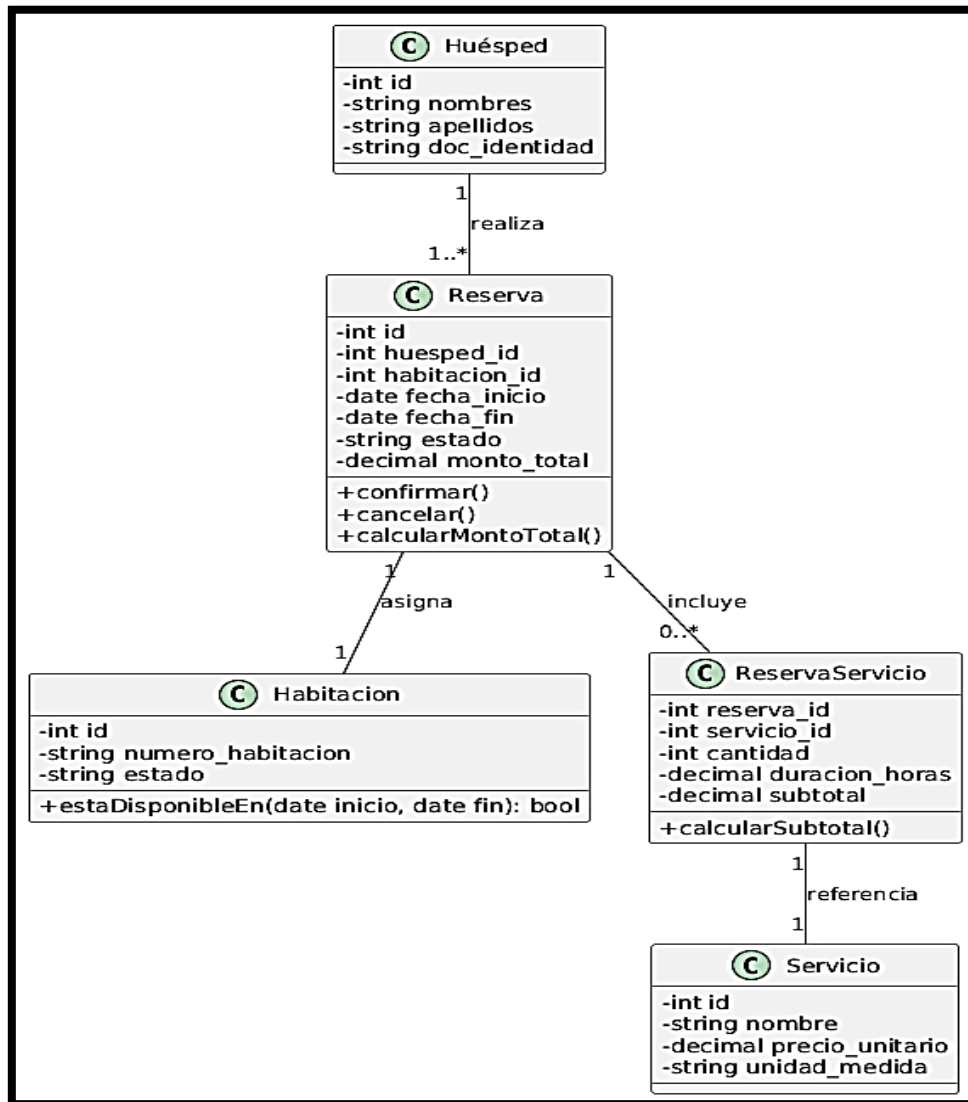


Figura 6 Diagrama de clases ReservasContext y ServiciosContext

Las clases reflejan entidades y objetos de valor del dominio. Por ejemplo, `DisponibilidadServicio` es un objeto de valor inmutable asociado a un `Servicio` de Reserva de un huésped y determina si existe una habitación del tipo solicitado en el periodo de tiempo solicitado.

El mapeo de los Bounded Contexts en el sistema del Hotel Viña del Sur consistió en dividir el modelo del dominio en partes más pequeñas y manejables, donde cada una tiene su propio significado, reglas y lenguaje claro. Esto se hizo siguiendo los principios de Domain-

Driven Design (DDD), con el objetivo de evitar confusiones y mantener el código organizado, especialmente en áreas críticas como reservas, habitaciones y emisión de recibos.

Cada Bounded Context (Contexto Delimitado) representa un subdominio funcional del hotel con su propio vocabulario técnico (que en Domain-Driven Design se conoce como lenguaje ubicuo), sus propias reglas de negocio, sus entidades y lógica específicas, y una separación clara respecto al resto del sistema para evitar ambigüedades y mantener la cohesión interna.

El sistema abarca múltiples áreas del dominio hotelero, cada una con lógicas de negocio diferenciadas: la gestión de reservas de habitaciones, la oferta y programación de servicios complementarios (como masajes, alquiler de salones y desayunos), el registro de consumos realizados durante la estancia, la emisión de recibos al finalizar la estadía, y la administración de datos de huéspedes incluyendo su lugar de origen. Esta diversidad funcional justifica la necesidad de un enfoque estructurado que permita modelar cada área de forma coherente y aislada, evitando acoplamiento innecesario entre sus componentes.

Sin esta delimitación clara, los módulos del sistema se mezclarían en un solo modelo, lo que causaría confusión, duplicación y errores.

A continuación se muestra los boundedContext establecidos para el sistema.

| Bounded Context     | Responsabilidad Principal                                   | Entidades Clave              | Lenguaje Ubicuo  |
|---------------------|---|------------------------------|--|
| ReservasContext     | Crear, confirmar y cancelar reservas de habitaciones        | Reserva, Huesped, Habitacion | “reservar”, “disponible”, “check-in pendiente”         |
| ServiciosContext    | Gestionar servicios adicionales (reservables o consumibles) | Servicio, ReservaServicio    | “sesión de sauna”, “hora de salón”, “desayuno premium” |
| HabitacionesContext | Administrar el catálogo y estado de habitaciones            | Habitación, TipoHabitacion   | “suite tarifeña”, “ocupada”, “en mantenimiento”        |
| RecibosContext      | Emitir recibos al finalizar la estancia                     | Recibo                       | “monto final”, “recibo emitido”, “método de pago”      |

Tabla 6 Bounded Context. Hotel Viña del Sur

En particular, se documenta ReservasContext y ServiciosContext ya que son las que estas referidas a las reservas de habitaciones y servicios que los huéspedes solicitan en el hotel, por tanto se detalla a continuación.

Para ReservasContext, se establece que gestiona todos los aspectos relacionados con la reserva de habitaciones y la estadía del huésped en el hotel. Las clases principales involucradas con este proceso son:

**Reserva**, representa el compromiso inicial para ocupar una habitación en un rango de fechas. Sus atributos principales son `id`, `fecha_inicio`, `fecha_fin`, `estado` (pendiente, confirmada, cancelada, etc.), `monto_total`, `cantidad_personas` y `fecha_reserva`. Cada reserva está asociada a una sola habitación (relación 1:1 con `habitacion`) y a un usuario del sistema (quien la registra). El método de pago se almacena directamente en la reserva (`metodo_pago`: tarjeta, efectivo, transferencia o sin pagar). Los servicios adicionales se gestionan en la tabla `reserva_servicio`.

**Huésped**, corresponde a la persona que se hospeda efectivamente en el hotel. Se crea a partir de una reserva confirmada y pagada. Sus datos (nombres, apellidos, `doc_identidad`, `lugar_origen_id`, etc.) se almacenan en la tabla `huesped`, la cual está vinculada a la tabla `usuario` y, posteriormente, a una. Un huésped puede tener múltiples estancias en el tiempo, pero cada estancia proviene de una única reserva.

**Habitación**, unidad física del hotel, identificada por `numero_habitacion`, `piso`, `tipo_habitacion_id` y `estado` (disponible, ocupada, reservada o mantenimiento). Su tarifa se define en la tabla `tipo_habitacion` (`precio_noche`). En un periodo determinado, una habitación puede estar asignada a una sola reserva, lo que se controla mediante validación de fechas en la lógica de negocio.

**Consumo / Servicio**, los servicios complementarios (desayunos, masajes, bebidas, etc.) se registran de dos formas:

**Antes del check-in:** en `reserva_servicio`, vinculados a la reserva.

**Durante la estadía:** en `consumo_estancia`, asociados a una estancia activa.

Cada servicio tiene `precio_unitario`, `unidad_medida` (hora, sesión, plato, noche) y puede tener disponibilidad horaria definida en `disponibilidad_servicio`.

El contexto `ReservasContext` garantiza la integridad temporal, es decir una habitación no puede estar asignada a dos reservas con fechas simultáneas o paralelas lo que impide que no se puede dar en reserva una misma habitación o espacio a dos huéspedes.

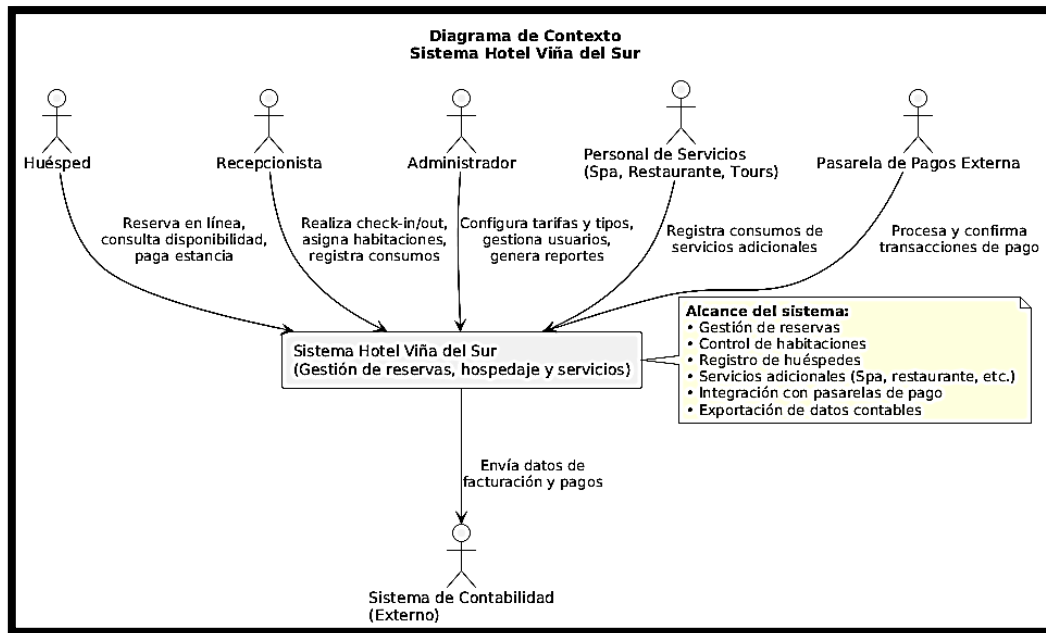


Figura 7 Diagrama de Contexto

Diagrama de contenedores

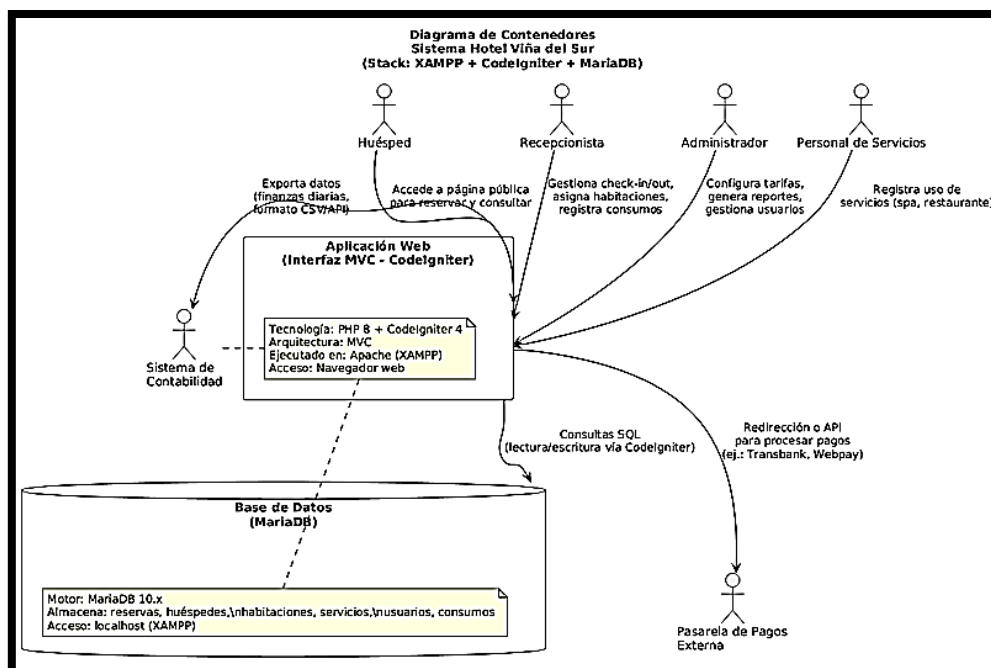


Figura 8 Diagrama de contenedores

Dado que se describió ReservasContext, a continuación se muestra la estructura de ServiciosContext



## ServiciosContext

Gestiona los servicios adicionales que el huésped consume durante su estadía (spa, restaurante, lavandería, tours, etc.).

- **Servicio**

Los atributos de Servicio son idServicio, nombre, descripcion, precioUnitario, categoría.

## Consumo

Atributos: idConsumo, cantidad, fechaHora, subtotal

- **Recibo**

Atributos: idRecibo, fechaEmision, total, estado

Los Consumos deben poder registrarse en tiempo real durante la estadía y sumarse al total de la Reserva.

Por último se muestra la relación entre ReservasContext y ServiciosContext

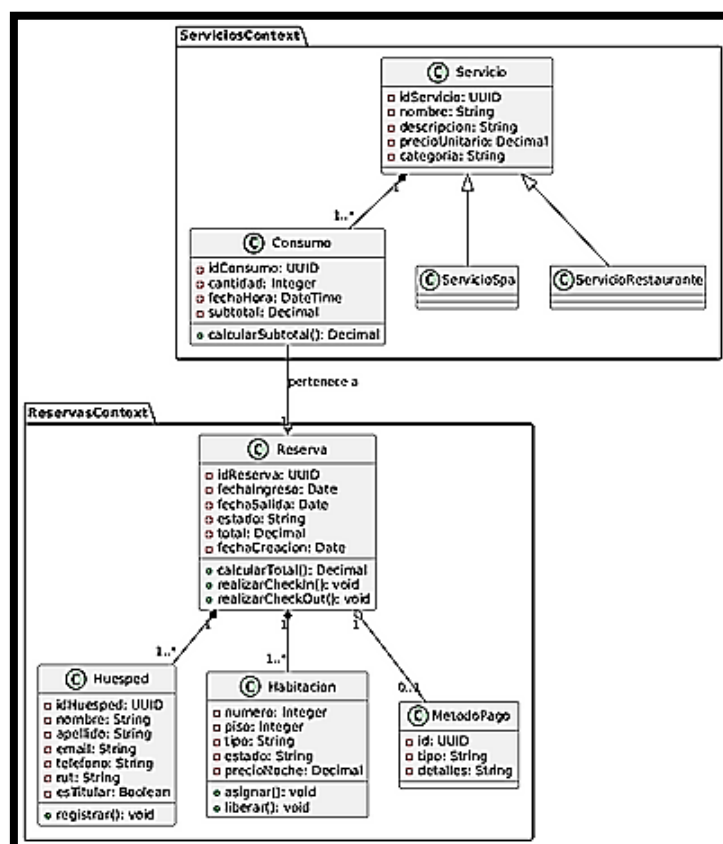


Figura 9 Relaciones entre ReservasContext y ServiciosContext

## Relaciones Clave entre Contextos

### Consumo → Reserva:

Cada `Consumo` pertenece a una y solo una `Reserva` lo que permite establecer gastos adicionales a la cuenta de la estadía.

### Reserva → Consumo:

Una `Reserva` puede tener múltiples `Consumos`. La clase `Reserva` aparece en ambos contextos, pero en el diseño físico no se duplica.

## 4.7 Desarrollo e implementación

### Tecnologías Utilizadas

| Capa                 | Tecnología  |
|----------------------|---|
| <b>Frontend</b>      | HTML5 + CSS3 + JavaScript (vanilla)<br>Plantillas con CodeIgniter Views<br>Estilos con Bootstrap 5                            |
| <b>Backend</b>       | CodeIgniter 4 (PHP 8.1+)<br>Patrón MVC<br>Validación, sesiones, autenticación nativa  |
| <b>Base de datos</b> | MariaDB 10.4+ (gestionada vía phpMyAdmin en entorno XAMPP)  |
| <b>Persistencia</b>  | CodeIgniter Query Builder + Modelos personalizados<br>(Sin ORM complejo como Hibernate; acceso directo y eficiente a MariaDB) |
| <b>API</b>           | API REST interna (opcional)<br>Endpoints JSON para AJAX (disponibilidad, listado)<br>Documentación manual                     |
| <b>Entorno</b>       | XAMPP (Apache Maria DB PHP ) para desarrollo local  |
| <b>Prácticas</b>     | DDD ligero (entidades con comportamiento)<br>XP: pruebas manuales, refactorización, historias de usuario                      |

Tabla 7 Tecnologías utilizadas Hotel Viña del Sur

### Versionamiento de Código

- Plataforma: GitHub
- Flujo de trabajo: GitFlow
  - `main`: versión estable en producción
  - `develop`: integración continua
  - `feature/xxx`: ramas por funcionalidad
  - `release/x.x`: preparación de versión
- Convenciones de commit: Conventional Commits
- Herramientas: GitHub Actions para CI/CD

## 4.8 Pruebas de software

| Tipo        | Herramienta / Enfoque                    | Propósito  |
|-------------|--|--|
| Unitarias   | PHPUnit (incluido en CI4)                | Probar servicios, entidades y lógica de negocio (ReservaService::calcularMontoTotal()).          |
| Integración | PHPUnit + Base de datos de prueba        | Probar repositorios y consultas SQL contra una base de datos de prueba (MariaDB).                |
| Frontend    | Pruebas manuales + validación HTML/JS    | Dado que no hay SPA, no se justifican Jest o RTL. Se verifica funcionalidad vía navegador.       |
| E2E         | Opcional: Codeception o pruebas manuales | Si se requiere automatización de flujos (reservar → confirmar), se puede usar Codeception (PHP). |
| Cobertura   | PHPUnit + php-code-coverage              | Generar reportes de cobertura de código en HTML.   |

Tabla 8 Pruebas de software

### 4.8.1 Pruebas implementadas

- **Pruebas Unitarias:**

Una prueba unitaria es una pequeña función que verifica que una parte muy específica de tu código funciona correctamente.

La principal prueba unitaria fue desarrollada de manera íntegra en las reservas de habitaciones y consumo de los servicios.

**Datos del Huésped**  
Nombre: CARLOS MARMOL  
Documento: 1232123

**Datos de la Habitación**  
Número: H1A  
Tipo: Suite Tarjeja Premium  
Check-in: 21/09/2025 21:27  
Noches: 1

**Resumen de Consumos**

Tarifa Base (1 noches @ Bs. 950.00/noche) **Bs. 950.00**

Consumos Durante Estancia

Desayuno Premium **Bs. 195.00**

22/09 20:49 - Administrador del Sistema

**Total a Pagar** **Bs. 1,145.00**

Pagos Previos **Bs. 950.00**

**Saldo Pendiente** **Bs. 195.00**

**Procesar Pago**

Método de Pago: Seleccione un método... Monto a Pagar (Bs.): Bs. 195.00

← Cancelar **Procesar Check-out**

Figura 10 Prueba unitaria de software

- **Pruebas de Integración:**

Para esta prueba se realizó el registro de una reserva de habitación y se verifico como quedaron afectados los registros en la tabla reservas de la base de datos hotel\_db

The screenshot displays a reservation form with the following sections:

- Fecha de Inicio \***: 01/10/2025
- Fecha de Fin \***: 02/10/2025
- Habitación \***: 222 - Suite Tarjeña Premium (Bs. 950.00/noche)
- Cantidad de Personas**: 2
- Servicios Adicionales**:
  - ☒ Sauna Premium - Bs. 80.00 / sesion
  - ☐ Salón Grande - Bodas y Conferencias - Bs. 800.00 / hora
  - ☒ Piscina Climatizada - Acceso Externo - Bs. 60.00 / sesion
  - ☐ Gimnasio Premium - Acceso Diario - Bs. 120.00 / noche
  - ☐ Sauna Privado - Bs. 80.00 / hora
  - ☐ Masaje Relajante - Bs. 220.00 / sesion
  - ☐ Salón Pequeño - Eventos - Bs. 350.00 / hora
  - ☐ Departamento VIP por Hora - Bs. 250.00 / hora
  - ☐ Masaje Relajante de 60 min - Bs. 220.00 / sesion
  - ☒ Cerveza Artesanal - Bs. 25.00 / plato
  - ☐ Desayuno Premium - Bs. 65.00 / plato
  - ☐ Traslado Aeropuerto - Bs. 150.00 / noche
- Notas / Solicitudes Especiales**: Prueba funcional

Buttons at the bottom: **Crear Reserva** and **Cancelar**.

Figura 11 Prueba de integración (Ingreso)

Al registrar la reserva se tiene el siguiente resultado unitario

**Detalles de la Reserva**

**📅 Fechas:**  
01/10/2025 - 02/10/2025

**👤 Cliente:**  
Administrador del Sistema

**💰 Monto Total Original:**  
Bs. 950.00

**🧾 Descuento por Pronto Pago:**  
¡Hasta 2% si paga hoy!

**💳 Método de Pago \***  
-- Seleccione método de pago --

**💰 Monto a Pagar \***  
Bs. 475.00

Monto mínimo requerido: Bs. 475.00 (50% del total)

**🎁 ¡Oferta Especial!** Si paga dentro de los próximos 2 días, recibirá un **2% de descuento** en el monto total.

**Confirmar Pago** **Cancelar**

Figura 12 Prueba de integración (Confirmación)

Y finalmente se tiene el registro en la base de datos, tabla reservas

reserva @hotel\_db (hotel) - Table

| id | usuario_id | habitacion_id | fecha_inicio | fecha_fin  | cantidad_personas | estado    | monto_total | descuento_aplicado | metodo_pago | fecha_reserva       | fecha_confirmacion | fecha_pago | notas_admin |
|----|------------|---------------|--------------|------------|-------------------|-----------|-------------|--------------------|-------------|---------------------|--------------------|------------|-------------|
| 15 | 5          | 9             | 2025-10-01   | 2025-10-02 | 2                 | pendiente | 950         | 0                  | sin pagar   | 2025-09-30 19:40:56 |                    |            |             |

SELECT \* FROM 'reserva' LIMIT 0, 1000 | Record 7 of 7 in page 1

Figura 13 Prueba de Integración (Base de datos)

- **Pruebas funcionales:**

Las pruebas funcionales de todo el proceso han sido verificadas, y cumplen con el 100% de las condiciones establecidas en las reservas.

## 5 Funcionalidades implementadas

### 5.1 Gestión de usuarios

- **Tabla:** usuario

#### **Funcionalidad:**

Autenticación de usuarios con roles diferenciados:

- Admin, gestión total del sistema.
- Recepcion, manejo de reservas, check-in/check-out.
- Limpieza, acceso limitado (para reporte de estado de habitaciones).
- Usuario, huéspedes registrados.

### 5.2 Gestión de huéspedes

- **Tablas:** huesped, lugar\_origen

#### **Funcionalidad:**

- Registro de huéspedes con datos personales (nombre, documento, origen).
- Normalización del lugar de origen (país + ciudad), con soporte para Bolivia y países vecinos.
- Vinculación con usuario al momento del primer pago.

### 5.3 Gestión de habitaciones, tipos de habitaciones

- **Tablas:** habitacion, tipo\_habitacion

#### **Funcionalidad:**

- Definición de tipos de habitación con nombre, descripción, capacidad y precio por noche (ej: *Suite Tarijeña Premium*, *Romántica Luna de Miel*).
- Gestión del estado de cada habitación: disponible, ocupada, reservada, mantenimiento.

### 5.4 Gestión Reservas

- **Tabla:** reserva

#### **Funcionalidad:**

- Creación de reservas con fechas de inicio y fin.

- Cálculo automático del monto total (basado en noches  $\times$  precio del tipo de habitación).
- Estados: pendiente, confirmada, cancelada, finalizada, limpieza.
- Soporte para descuentos (%), método de pago por pagos ‘Pronto pago’.

## 5.5 Gestion Servicios

- **Tablas:** servicio, reserva\_servicio, disponibilidad\_servicio

Funcionalidad:

- Catálogo de servicios (masajes, salones, desayunos, acceso a gimnasio, etc.) con precio y unidad de medida (hora, sesión, plato, noche).
- Reserva anticipada de servicios vinculados a una reserva de habitación.

## 5.6 CheckIn / CheckOut

- **Tablas:** estancia, consumo\_estancia

Funcionalidad:

- Registro de la estadía real del huésped (desde check-in hasta check-out).
- Registro de consumos durante la estancia (ej: cerveza, desayuno, masaje) por parte del staff (`registrado_por`).
- Acumulación del monto final que incluye habitación + servicios reservados + consumos.

## 5.7 Reportes Recibo

- **Tabla:** recibo

Funcionalidad:

- Generación de un recibo al finalizar la estancia.
- Incluye monto total, impuestos, descuentos, método de pago y número único (ej: REC-2025-00005).
- Compatible con la normativa (uso de solo “recibo”).

## 5.8 Reportes diarios

- **Tabla:** reporte\_diario



Funcionalidad:

Generación automática o manual de reportes por día.

Incluye:

- Total de huéspedes ingresados.
- Habitaciones ocupadas.
- Ingresos en BOB y a futuro se pretende el uso de USD (usando tipo\_cambio).
- Top de lugares de origen y servicios más usados.

El desarrollo de la base de datos del Hotel Viña del Sur no solo cubre las funcionalidades básicas de un sistema de reservas, sino que también anticipa necesidades operativas reales:

- Gestión de consumos durante la estancia,
- Reportes gerenciales detallados,
- Soporte para servicios,

Esta base de datos constituye una base sólida para la aplicación en CodeIgniter 4, y refleja un análisis de dominio y alineado con las prácticas de Domain-Driven Design.

## 6 Resultados

El desarrollo del sistema de reservas del Hotel Viña del Sur ha permitido alcanzar los siguientes resultados concretos.

### 6.1 Sistema funcional y estable

- Se implementó un flujo completo de reserva: desde la selección de fechas y habitación, hasta la confirmación y generación de un recibo.
- Los huéspedes pueden visualizar disponibilidad en tiempo real, gracias a la validación de fechas contra la tabla `reserva`.
- Los servicios adicionales (masajes, salones, desayunos) se integran de forma flexible a través de la tabla `reserva_servicio`.

### 6.2 Arquitectura clara y mantenible

- El modelo de dominio refleja de manera real el lenguaje del negocio (huésped, estancia, recibo, tipo de habitación).
- La separación en contextos delimitados (`ReservasContext`, `ServiciosContext`) evita el acoplamiento excesivo.

- El uso de patrones como Repository, Factory y Specification ha permitido encapsular la lógica de negocio y facilitar futuras modificaciones.

### **6.3 Calidad de código**

- Se aplicaron básicamente de Extreme Programming, programación en pareja (en sesiones semanales), refactorización continua y diseño guiado por pruebas.
- Se desarrollaron pruebas unitarias básicas con PHPUnit para validar cálculos de montos, fechas y estados.
- El código sigue las convenciones de CodeIgniter 4, lo que facilita la incorporación de nuevo personal de desarrollo que así quiera realizar el mantenimiento y posteriormente el desarrollo de nuevas funcionalidades.

## 7 Conclusiones y recomendaciones

### 7.1 Conclusiones

El sistema de reservas del Hotel Viña del Sur cumple con los objetivos planteados.

- Se diseñó e implementó un sistema web funcional para el Hotel que mejora la forma en que se manejan las reservas, el registro de huéspedes y la generación de recibos. Como beneficio de la automatización, el personal involucrado con la administración del hotel, ahora trabaja más rápido, con menos errores y ofreciendo un mejor servicio.
- La interfaz del sistema es limpia, intuitiva y adaptable a cualquier dispositivo, usando tecnologías como HTML5, CSS3 y Bootstrap, lo que facilita mucho el trabajo y desempeño del personal de recepción.
- Se tiene ahora un reporte en tiempo real que muestra qué habitaciones están libres, reservadas u ocupadas, lo que evita el problema de asignar una habitación que ya está en uso.
- Se logró automatizar el proceso de reserva desde que un cliente elige fechas hasta el check-in, check-out y el cálculo automático del total a pagar. Este cálculo reduce la carga manual y hace que el proceso de administración y seguimiento de habitaciones y servicios se realice de forma más ágil.
- Se desarrolló módulos para el registro de cualquier consumo adicional durante la estadía (desayunos, masajes o bebidas y otros) los que se suman de manera automática a la cuenta del huésped, sin necesidad de cálculos manuales.
- Por último, se logró desarrollar el sistema software, aplicando conceptos de Domain-Driven Design (DDD) y Extreme Programming (XP). Además, se logró incluir validaciones tanto en el frontend como en el backend para garantizar que la información se maneje de forma segura, consistente y sin errores.

### 7.2 Recomendaciones

- **Ampliar la cobertura de pruebas**  
Implementar pruebas de integración automatizadas que validen la interacción con la base de datos (DatabaseTestTrait de CI4).

- **Implementar notificaciones automáticas**

Agregar el patrón Observer para enviar correos electrónicos o mensajes al huésped al confirmar una reserva (usando una librería como PHPMailer).

- **Preparar migración a entorno de producción**

- Crear scripts de respaldo y restauración de la base de datos.
- Configurar variables de entorno (`env`) para separar configuración de desarrollo y producción.

- **Considerar un panel de reportes**

Basado en la tabla `reporte_diario`, se podría desarrollar un módulo administrativo que muestre estadísticas de ocupación, orígenes de huéspedes y servicios más demandados.

## 8 Referencias

- Alonso Amo Fernando, Martinez Normand Loic y Segovia Francisco Javier** Introducción a la Ingeniería del software [Libro]. - [s.l.] : Delta Publicaciones, 2005.
- Beck Kent [y otros]** Manifiesto por el Desarrollo Ágil de Software [Libro]. - [s.l.] : Agile Alliance, 2001.
- Evans Erick** Domain-Driven Design: Tackling Complexity in Heart of Software [Libro]. - 2004.
- Gamma J.** Advances in Software Engineering.Comprehension, Evaluation, and Evolution [Libro]. - New York : Springer New York, 2013.
- Gonzalez Candia Hernaldo** MVC 4 Desde Cero [Libro]. - [s.l.] : Hernaldo Gonzalez Candia, 2015.
- IEEE** IEEE Standards Collection: Software Engineering [Informe]. - [s.l.] : IEEE, 1993.
- Myers Jhon** “The art of software testing [Libro]. - [s.l.] : John Wiley & Sons, 2004.
- MySQL** MySQL [En línea]. - 18 de Octubre de 2018. - <https://www.mysql.com/>.
- Pressman Roger S.** Ingeniería de Software. Sexta edición [Libro]. - [s.l.] : Mc Graw Hill, 2008.
- Sommerville Ian** Ingeniería del Software. Septima edición [Libro]. - Madrid : Pearson Educación S.A., 2005.

## 9 Anexos

### 10 Manual de usuario

#### 10.1 A quien va dirigido.

Personal operativo del hotel principalmente recepcionistas, ayudantes de piso, cajeros y personal administrativo

En realidad está dirigido para aquellas personas que usan el sistema diariamente pero sin los conocimientos técnicos ni de desarrollo de sistemas.

#### 10.2 Propósito

Guiar al usuario en el uso correcto y eficiente del sistema para realizar sus tareas cotidianas: reservas, check-in, check-out, registro de consumos, estado de las habitaciones y espacios: disponible, en limpieza, en mantenimiento.

#### 10.3 Contenido

##### 10.3.1 Inicio de sesión

- Abra su navegador web (Chrome, Firefox o Edge).

En la barra de direcciones, escriba la dirección del sistema

`https://www.hotel.bo`

Si el sistema está en un servidor de la red del hotel, la URL podría ser algo como `http://localhost/hotel`.



Consulte con el administrador si no puede acceder.

Si todo está correctamente los datos en la url, se tiene la pantalla siguiente



- Ingreso de usuario y contraseña

De la pantalla que se presenta, dar clic en ‘Panel Admin’



Y luego debe ingresar el usuario y contraseña asignados por el administrador del sistema para ingresar al sistema web Hotel Viña del Sur

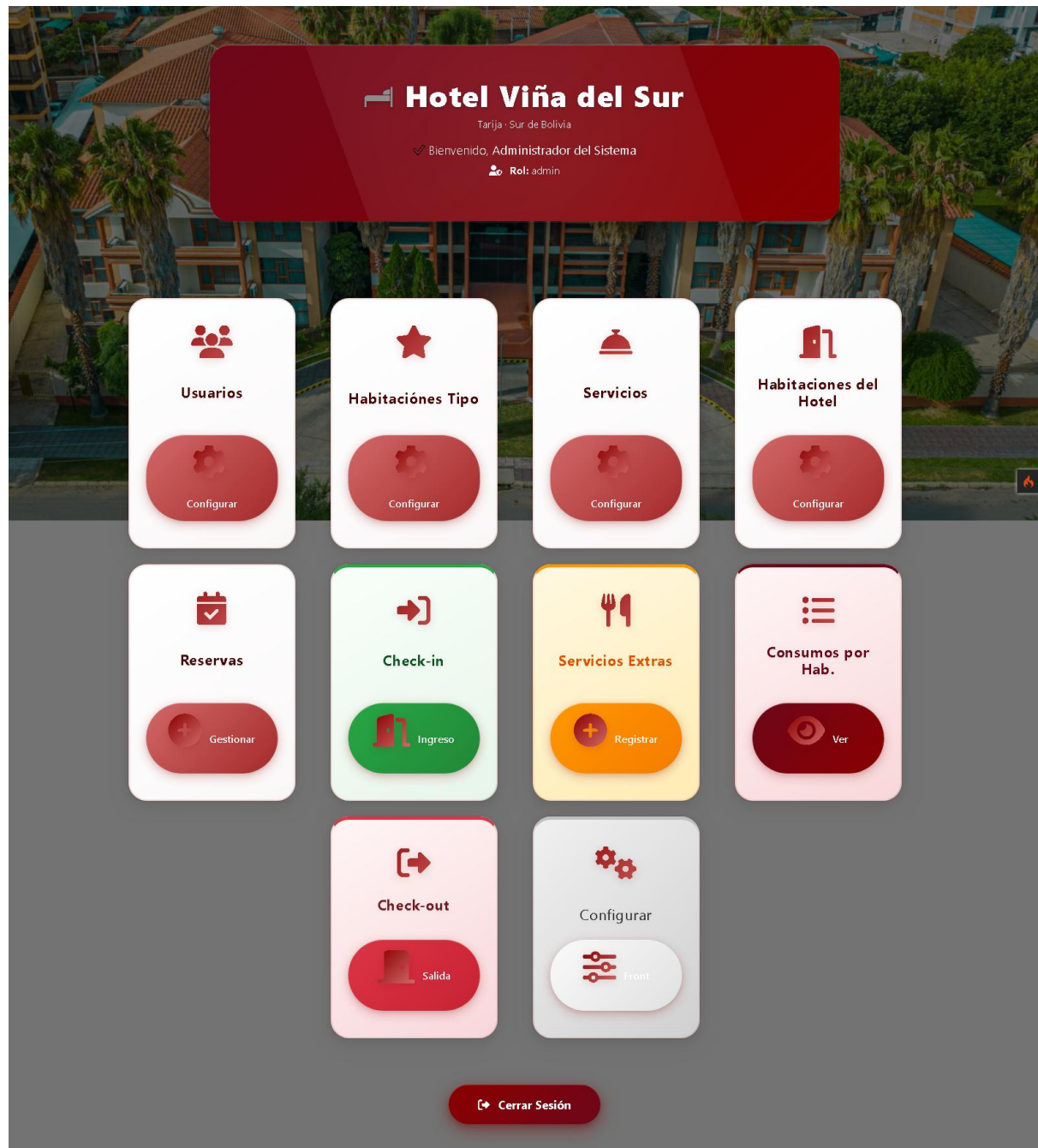
**Usuario:** su nombre de login (ej: nancy\_recep)

**Contraseña:** la clave asignada por el administrador

- Escriba sus credenciales con mayúsculas y minúsculas exactas.
- Haga clic en el botón Ingresar.
- Si los datos son correctos, será redirigido al Panel Principal.
- Si recibe un mensaje de error, verifique que su usuario esté activo y que no haya errores de escritura.

Se presenta la pantalla de administración





- Cierre de sesión.
  - En la parte inferior central se tiene el botón ‘Cerrar sesion’
  - Seleccione la opción **Cerrar Sesión**.
  - El sistema lo redirigirá a la pantalla de frontend del sistema.

### 10.3.2 Gestión de reservas

Crear una nueva reserva (selección de huésped, habitación, fechas).

Vaya al menú principal y haga clic en **Reservas > Nueva Reserva**.

**Paso 1 – Huésped:**

- Si el huésped ya está registrado, escriba su nombre o documento en el buscador y selecciónelo.
- Si es nuevo, haga clic en + Registrar Huésped y complete sus datos (nombre, apellido, documento, ciudad de origen).

**Paso 2 – Fechas:**

- Ingrese la fecha de check-in y check-out (use el calendario integrado).

**Paso 3 – Habitación:**

- El sistema mostrará automáticamente las habitaciones disponibles en ese rango.
- Seleccione una habitación (ej: “1A – Suite Tarijeña Premium”).

**Paso 4 – Confirmar:**

- Revise el monto total (calculado automáticamente).
- Haga clic en Guardar Reserva.
- La reserva se crea con estado “pendiente”.

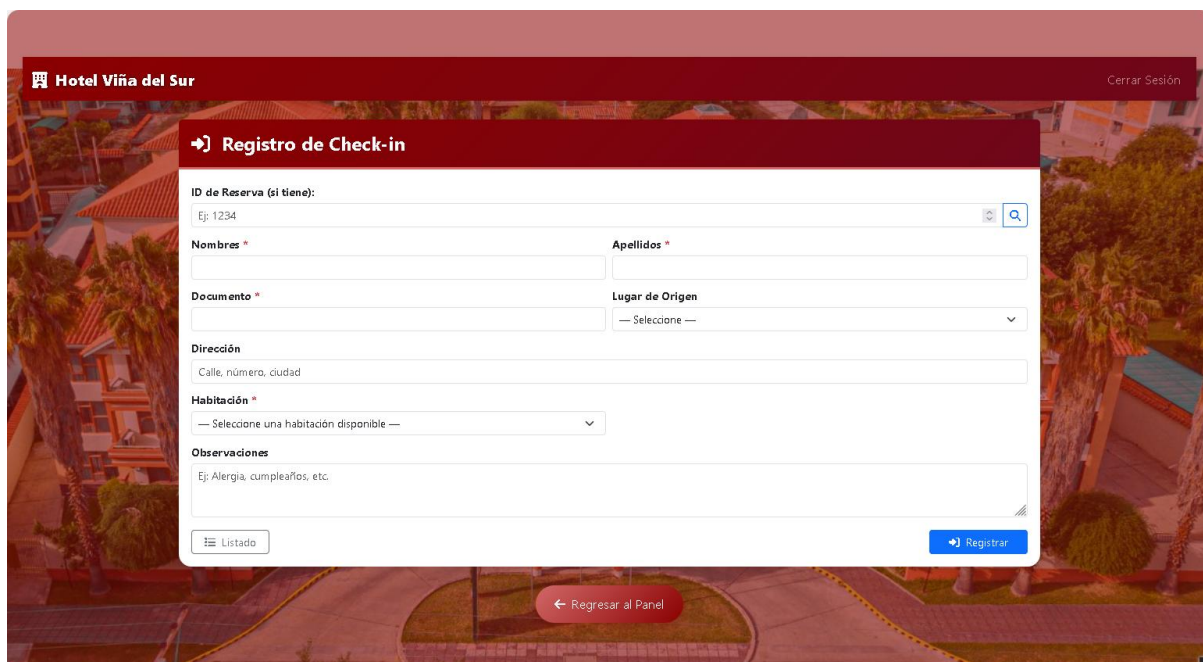
Ver reservas pendientes, confirmadas o canceladas.

- Vaya a **Reservas > Listado de Reservas**.
- Modificar o cancelar una reserva.

Modificar, solo se permite cambiar fechas u habitación si la reserva está en estado “pendiente”.

- Abra la reserva.
- Haga clic en **Editar**.
- Realice los cambios y guarde.

### 10.3.3 Check-in

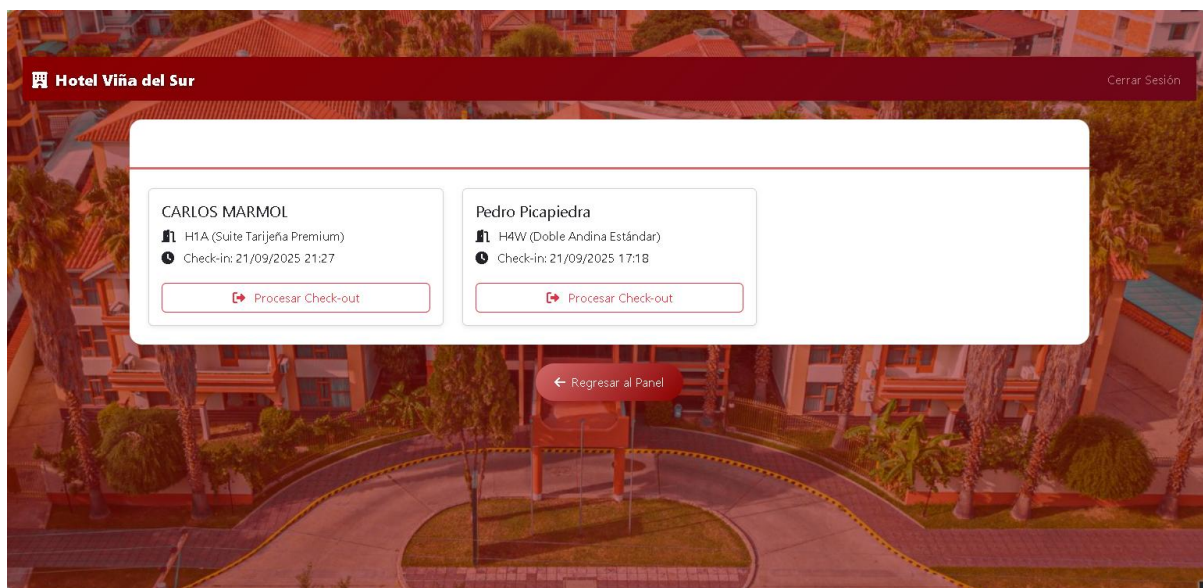


The screenshot shows a web application interface for Hotel Viña del Sur. At the top, there is a dark red header with the hotel's logo and name on the left, and a "Cerrar Sesión" link on the right. Below the header is a large, semi-transparent modal window titled "Registro de Check-in" with a right-pointing arrow icon. The form inside the modal contains several fields: "ID de Reserva (si tiene):" with a placeholder "Ej: 1234" and a search icon; "Nombres \*" and "Apellidos \*" as separate text inputs; "Documento \*" and "Lugar de Origen" (a dropdown menu with "— Seleccione —" as the selected option); "Dirección" with a placeholder "Calle, número, ciudad"; "Habitación \*" (a dropdown menu with "— Seleccione una habitación disponible —" as the selected option); and "Observaciones" with a placeholder "Ej: Alergia, cumpleaños, etc.". At the bottom left of the form is a "Listado" button with a list icon, and at the bottom right is a blue "Registrar" button with a right-pointing arrow icon. Below the modal, there is a red button with a left-pointing arrow and the text "Regresar al Panel". The background of the page shows a blurred image of a hotel building and palm trees.

Cómo registrar el ingreso del huésped (check-in).

1. Vaya a Reservas > Confirmadas.
2. Busque la reserva del huésped que llega hoy.
3. Haga clic en Realizar Check-in.
4. El sistema:
  - Crea una estancia vinculada a la reserva.
  - Cambia el estado de la habitación a “ocupada”.
  - Registra la hora exacta de ingreso.

### 10.3.4 Check-out



Cómo registrar la salida (check-out).

1. Vaya a Estancias > Activas.
2. Seleccione la estancia del huésped que se retira.
3. Haga clic en Realizar Check-out.
4. El sistema:
  - Calcula el monto final (habitación + consumos).
  - Permite seleccionar el método de pago.
  - Prepara la emisión del recibo.

Cambio automático del estado de la habitación.

- Al hacer **check-in**, la habitación pasa de “disponible” al estado “ocupada”.
- Al hacer **check-out**, la habitación pasa a “limpieza”.
- Una vez que el personal de limpieza confirma que está lista, un administrador debe cambiarla manualmente a “disponible”.

### 10.3.5 Registro de Consumos

**Hotel Viña del Sur** Cerrar Sesión

**Servicio Extra**

**HUÉSPED**  
 Seleccione huésped activo... ▼

**SERVICIO**  
 Seleccione servicio... ▼

**CANTIDAD**  
 1

**CUÁNDO**  
 02/10/2025, 08:41

**NOTAS**  
 Ej: Sin hielo, sorpresa...

**REGISTRAR SERVICIO**

**CANCELAR**

[← Regresar al Panel](#)

- Agregar servicios consumidos durante la estancia (ej: desayuno, cerveza).
- Selección del servicio, cantidad y observaciones.
- En la ventana emergente:

**Servicio:** seleccione de la lista (ej: “Desayuno Premium”, “Cerveza Artesanal”).

**Cantidad:** ingrese el número de unidades (ej: 2 desayunos).

**Duración** (si aplica): para servicios por hora (ej: sauna).

**Observaciones:** opcional (ej: “para habitación 1A”, “sin hielo”).

Haga clic en **Registrar Consumo**.

- Visualización del monto acumulado.

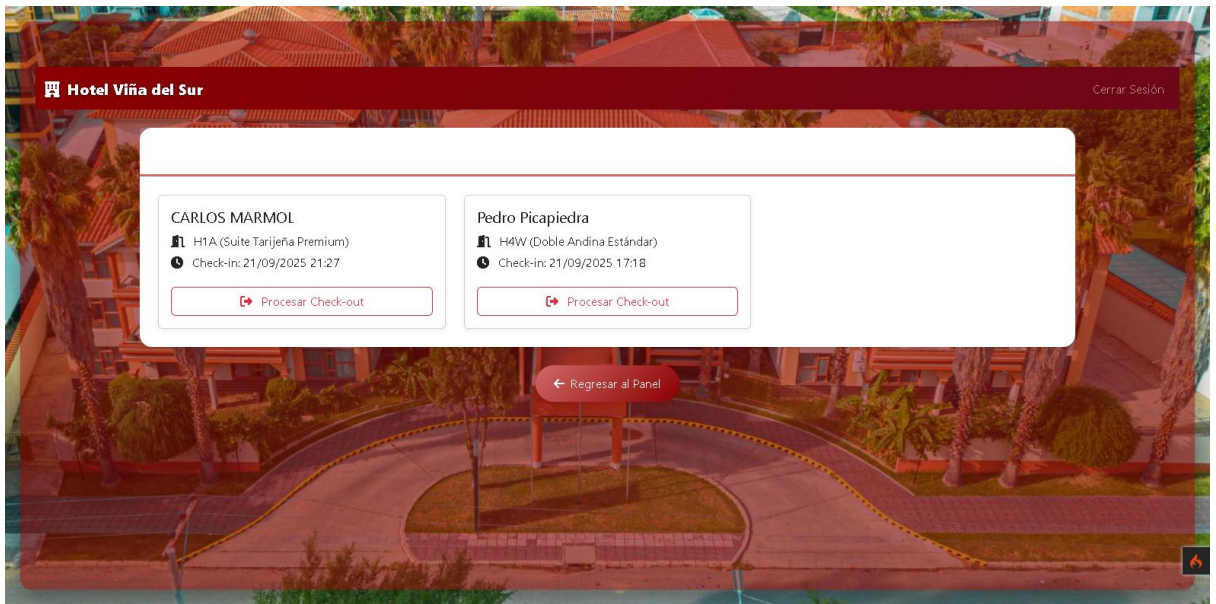
El sistema muestra en tiempo real el **subtotal de consumos**.

Al hacer check-out, este monto se suma al de la habitación para formar el **monto final** del recibo.

### 10.3.6 Emisión de recibos

Se genera principalmente cuando el huésped realiza el checkOut del hotel, por lo que el sistema de manera automática emite un recibo





Al elegir un huésped, se establece que se desea realizar el checkout de esa habitación, por lo que el sistema emite una vista preliminar de los gastos del huésped

**Hotel Viña del Sur** Cerrar Sesión

**Datos del Huésped**

**Nombre:** Pedro Picapiedra

**Documento:** 34543

**Datos de la Habitación**

**Número:** H4W

**Tipo:** Doble Andina Estándar

**Check-in:** 21/09/2025 17:18

**Noches:** 1

**Resumen de Consumos**

**Tarifa Base (1 noches @ Bs. 520.00/noche)** Bs. 520.00

Consumos Durante Estancia

**Cerveza Artesanal** 1.00h Bs. 100.00

21/09 18:23 - Administrador del Sistema

**Desayuno Premium** 1.00h Bs. 130.00

21/09 19:21 - Administrador del Sistema

**Piscina Climatizada - Acceso Externo** 1.00h Bs. 60.00

21/09 20:11 - Administrador del Sistema

**Total a Pagar** Bs. 810.00

Pagos Previos Bs. 520.00

**Saldo Pendiente** Bs. 290.00

**Procesar Pago**

**Método de Pago** Seleccione un método... **Monto a Pagar (Bs.)** Bs. 290.00

← Cancelar ✓ Procesar Check-out

← Regresar al Panel

Finalmente se tiene

1. Tras realizar el check-out, el sistema le pedirá confirmar el método de pago (efectivo, tarjeta, transferencia).
  2. Haga clic en Generar Recibo.
  3. El sistema crea un recibo.
    - Número único
    - Fecha y hora
    - Desglose de cargos
    - Monto total en BOB
- Impresión o visualización del recibo.
    - Visualizar: el recibo se muestra en pantalla en formato PDF.

- Imprimir: haga clic en el botón Imprimir (el navegador abre la ventana de impresión).