

**MANUAL TECNICO**

**Sistema Web de Atención al Huésped**

**Fecha: Junio del 2025**

**INDICE**

[1. Introducción 1](#_Toc201868720)

[1.1. Objetivo del manual 1](#_Toc201868721)

[1.2. Audiencia objetivo 1](#_Toc201868722)

[1.3. Alcance del sistema 1](#_Toc201868723)

[1.4. Convenciones y notación utilizada 1](#_Toc201868724)

[2. Requerimientos Técnicos 2](#_Toc201868725)

[2.1. Requerimientos mínimos de hardware 2](#_Toc201868726)

[2.2. Requerimientos mínimos de software 2](#_Toc201868727)

[3. Tecnologías Utilizadas 3](#_Toc201868728)

[4. Arquitectura del Sistema 4](#_Toc201868729)

[4.1. Descripción General 4](#_Toc201868730)

[4.2. Módulos principales 4](#_Toc201868731)

[4.3. Relaciones entre componentes 5](#_Toc201868732)

[5. Diagramas de Modelamiento 6](#_Toc201868733)

[5.1. Diagrama de Caso de Uso 6](#_Toc201868734)

[5.2. Diagrama de Actividades 7](#_Toc201868735)

[5.2.1. Huésped 7](#_Toc201868736)

[5.2.2. Personal de Atención al Cliente 8](#_Toc201868737)

[5.2.3. Administrador del Sistema 9](#_Toc201868738)

[5.3. Diagrama de contexto 9](#_Toc201868739)

[5.4. Modelo Relacional – Base de Datos 10](#_Toc201868740)

[6. Diccionario de Datos 10](#_Toc201868741)

[7. Diagrama Web 20](#_Toc201868742)

[8. Instalación y Configuración del Sistema 21](#_Toc201868743)

[8.1. Instalación del Sistema 21](#_Toc201868744)

[8.2. Configuración del Entorno 21](#_Toc201868745)

[9. Administración del Sistema 22](#_Toc201868746)

[9.1. Mantenimiento 22](#_Toc201868747)

[9.2. Respaldo y Recuperación 22](#_Toc201868748)

[9.3. Monitoreo 23](#_Toc201868749)

**TABLA DE ILUSTRACIONES**

[**Ilustración 1** Diagrama de Caso de Uso 6](#_Toc201868772)

[**Ilustración 2** Diagrama de Actividades - Huésped 7](#_Toc201868773)

[**Ilustración 3** Diagrama de Actividades - Personal de Atención al Cliente 8](#_Toc201868774)

[**Ilustración 4** Diagrama de Actividades - Administrador del Sistema 9](#_Toc201868775)

[**Ilustración 5** Diagrama de Contexto 9](#_Toc201868776)

[**Ilustración 6** Modelo Relacional - Base de Datos 10](#_Toc201868777)

[**Ilustración 7** Diagrama Web 20](#_Toc201868778)

**TABLA DE TABLAS**

[**Tabla 1** Requerimientos Mínimos de Hardware 2](#_Toc201868882)

[**Tabla 2** Requerimientos Mínimos de Software 3](#_Toc201868883)

[**Tabla 3** Tecnologías Utilizadas 3](#_Toc201868884)

[**Tabla 4** Cliente 10](#_Toc201868885)

[**Tabla 5** Empleado 11](#_Toc201868886)

[**Tabla 6** Función 12](#_Toc201868887)

[**Tabla 7** Habitación 12](#_Toc201868888)

[**Tabla 8** Categoría Servicio 13](#_Toc201868889)

[**Tabla 9** Servicio 13](#_Toc201868890)

[**Tabla 10** Reseña 14](#_Toc201868891)

[**Tabla 11** Actividad 14](#_Toc201868892)

[**Tabla 12** Función Servicio 15](#_Toc201868893)

[**Tabla 13** Categoría Comida 15](#_Toc201868894)

[**Tabla 14** Horario Actividad 16](#_Toc201868895)

[**Tabla 15** Inscripción 16](#_Toc201868896)

[**Tabla 16** Producto 17](#_Toc201868897)

[**Tabla 17** Reserva Habitación 17](#_Toc201868898)

[**Tabla 18** Servicio Comida 18](#_Toc201868899)

[**Tabla 19** Tipo Documento 18](#_Toc201868900)

[**Tabla 20** Usuario 18](#_Toc201868901)

[**Tabla 21** Tipo de Habitaciones 19](#_Toc201868902)

[**Tabla 22** Asignación Servicio 19](#_Toc201868903)

# Introducción

## Objetivo del manual

Este manual técnico tiene como objetivo proporcionar una guía detallada para la instalación, configuración, despliegue y mantenimiento del sistema web de gestión hotelera Casa Andina Premium. Está diseñado para documentar el funcionamiento técnico del sistema y facilitar su correcta implementación, asegurando su operatividad, seguridad y escalabilidad en el entorno real del hotel.

## Audiencia objetivo

Este documento está dirigido principalmente a:

* Personal técnico responsable del despliegue y mantenimiento del sistema.
* Administradores del sistema encargados de la configuración, monitoreo y respaldo de la plataforma.
* Desarrolladores encargados de futuras mejoras, ajustes o migraciones del software.
  1. Alcance del sistema

El sistema web desarrollado para Casa Andina Premium San Isidro abarca funcionalidades orientadas a digitalizar y automatizar la experiencia del huésped y los procesos internos del hotel. Incluye:

* Gestión de reservas de habitaciones.
* Registro y seguimiento de solicitudes de servicios por parte del huésped.
* Coordinación de actividades internas y espacios comunes.
* Administración de productos, personal y turnos.
* Interacción por medio de un chatbot automatizado.
* Generación de reportes, control de accesos y retroalimentación en tiempo real.

El sistema es accesible desde dispositivos móviles y navegadores web modernos, con funcionalidades diferenciadas según el rol del usuario (huésped, empleado, administrador).

* 1. Convenciones y notación utilizada

Para una mejor comprensión del documento se utilizan las siguientes convenciones:

* Los nombres de módulos, funciones o entidades del sistema se presentan en negrita (por ejemplo: Reservas, Chatbot).
* El código o comandos de configuración se mostrarán en tipografía monoespaciada: config.php
* Las acciones del usuario se expresan en infinitivo (por ejemplo: “seleccionar habitación”, “ingresar credenciales”).
* Se emplea lenguaje técnico estándar y preciso, evitando ambigüedades.

# Requerimientos Técnicos

## Requerimientos mínimos de hardware

Características físicas mínimas que deben tener los dispositivos (computadoras, servidores, móviles, etc.) para que el sistema funcione de manera adecuada.

**Tabla 1**   
Requerimientos Mínimos de Hardware

|  |  |
| --- | --- |
| **Equipo** | **Especificación** |
| Computadora (cliente o servidor local) | - Procesador Intel Core i3 o superior (≥ 2.0 GHz)  - Memoria RAM de 4 GB o superior  - Disco duro con al menos 50 GB disponibles  - Conexión a internet estable (≥ 10 Mbps) |
| Dispositivo móvil (huésped) | - Smartphone Android/iOS gama media o superior  - Procesador ≥ 1.8 GHz  - Memoria RAM de 2 GB o más  - Almacenamiento interno ≥ 8 GB |
| Servidor de hosting (entorno en producción) | - CPU virtual ≥ 2 vCPU (equivalente a 2.0 GHz)  - Memoria RAM ≥ 8 GB  - Almacenamiento SSD ≥ 50 GB  - Ancho de banda mensual ≥ 100 GB |

## Requerimientos mínimos de software

Hace referencia a los programas o plataformas mínimas que deben estar instaladas en el equipo para que el sistema funcione correctamente.

**Tabla 2**   
Requerimientos Mínimos de Software

|  |  |
| --- | --- |
| **Componente** | **Requisitos Mínimos** |
| Sistema Operativo | Windows 10, Linux Ubuntu 20.04 o superior |
| Navegador web (usuario final) | Google Chrome, Firefox, Microsoft Edge (últimas versiones) |
| Servidor Web | Apache 2.4 o superior |

# Tecnologías Utilizadas

**Tabla 3**  
Tecnologías Utilizadas

|  |  |
| --- | --- |
| **Tecnología / Herramienta** | **Uso en el Proyecto** |
| **Node.js** | Entorno de ejecución para el desarrollo del servidor backend. Permite ejecutar código JavaScript fuera del navegador. |
| **Express.js** | Framework minimalista para Node.js. Facilita el manejo de rutas, middleware y controladores para la lógica del negocio. |
| **MySQL** | Sistema de gestión de bases de datos relacional utilizado para almacenar la información del sistema, como usuarios, reservas, productos, etc. |
| **EJS (Embedded JavaScript)** | Motor de plantillas que permite generar vistas dinámicas renderizadas desde el servidor, combinando HTML con lógica embebida. |
| **Dotenv** | Biblioteca que permite la gestión de variables de entorno a través de archivos .env, facilitando la configuración del entorno sin exponer datos sensibles. |
| **Nodemon** | Herramienta de desarrollo que reinicia automáticamente el servidor cada vez que se detectan cambios en el código fuente. |
| **Bootstrap / CSS** | **Bootstrap** proporciona una estructura de diseño responsive y componentes visuales predefinidos. **CSS** complementa con estilos personalizados para la interfaz gráfica del sistema. |

# Arquitectura del Sistema

## Descripción General

El sistema está construido bajo una arquitectura cliente-servidor de tres capas, lo que permite una clara separación entre la interfaz de usuario, la lógica del negocio y el acceso a datos. Este enfoque mejora la escalabilidad, mantenibilidad y seguridad de la aplicación.

Las capas que conforman esta arquitectura son:

* **Frontend (Capa de Presentación):**

Desarrollado utilizando EJS como motor de plantillas, junto con HTML, CSS y JavaScript, permite generar dinámicamente las vistas que el usuario final visualiza e interactúa mediante un navegador web.

* **Backend (Capa de Lógica de Negocio):**

Implementado con Node.js y el framework Express, gestiona las rutas del sistema, ejecuta los controladores correspondientes y aplica las reglas de negocio para procesar las solicitudes de los usuarios.

* **Base de Datos (Capa de Persistencia):**

Utiliza MySQL como sistema de gestión de base de datos relacional. Almacena toda la información del sistema (usuarios, reservas, habitaciones, productos, etc.) y utiliza procedimientos almacenados para realizar operaciones CRUD de forma eficiente y estructurada.

## Módulos principales

El sistema se organiza funcionalmente en los siguientes módulos principales:

* **Gestión de Usuarios:**

Registro, autenticación y administración de usuarios (empleados y administradores). Incluye control de permisos y roles.

* **Reservas de Servicios:**

Permite a los huéspedes realizar reservas de habitaciones, actividades, comidas y otros servicios ofrecidos por el hotel.

* **Administración de Inventario:**

Gestiona habitaciones, productos, actividades, categorías y disponibilidad, orientado a los usuarios administrativos del sistema.

* **Control de Roles y Accesos:**

Define los permisos asociados a cada tipo de usuario, restringiendo el acceso a funcionalidades según su perfil.

* **Visualización de Reportes:**

Módulo que presenta estadísticas y reportes sobre la operación del sistema, útil para fines administrativos y de análisis.

## Relaciones entre componentes

Las interacciones entre componentes se basan en el modelo cliente-servidor:

* El cliente (navegador del usuario) envía solicitudes HTTP al **servidor backend**.
* El servidor procesa la solicitud, accede a la **base de datos** si es necesario, y responde con la información requerida o con una vista renderizada.
* Las vistas EJS combinan datos provenientes del servidor con **plantillas HTML** para generar dinámicamente las páginas mostradas al usuario.
* Las operaciones entre los módulos se gestionan de manera coordinada a través del backend, que **actúa como** **intermediario** entre la interfaz de usuario y la base de datos.

Esta arquitectura modular permite mantener una alta cohesión entre funcionalidades relacionadas, mientras se reduce el acoplamiento entre capas, lo que facilita futuras ampliaciones o mantenimientos del sistema.

# Diagramas de Modelamiento

## Diagrama de Caso de Uso

Diagrama

El contenido generado por IA puede ser incorrecto.

**Ilustración 1** Diagrama de Caso de Uso

## Diagrama de Actividades

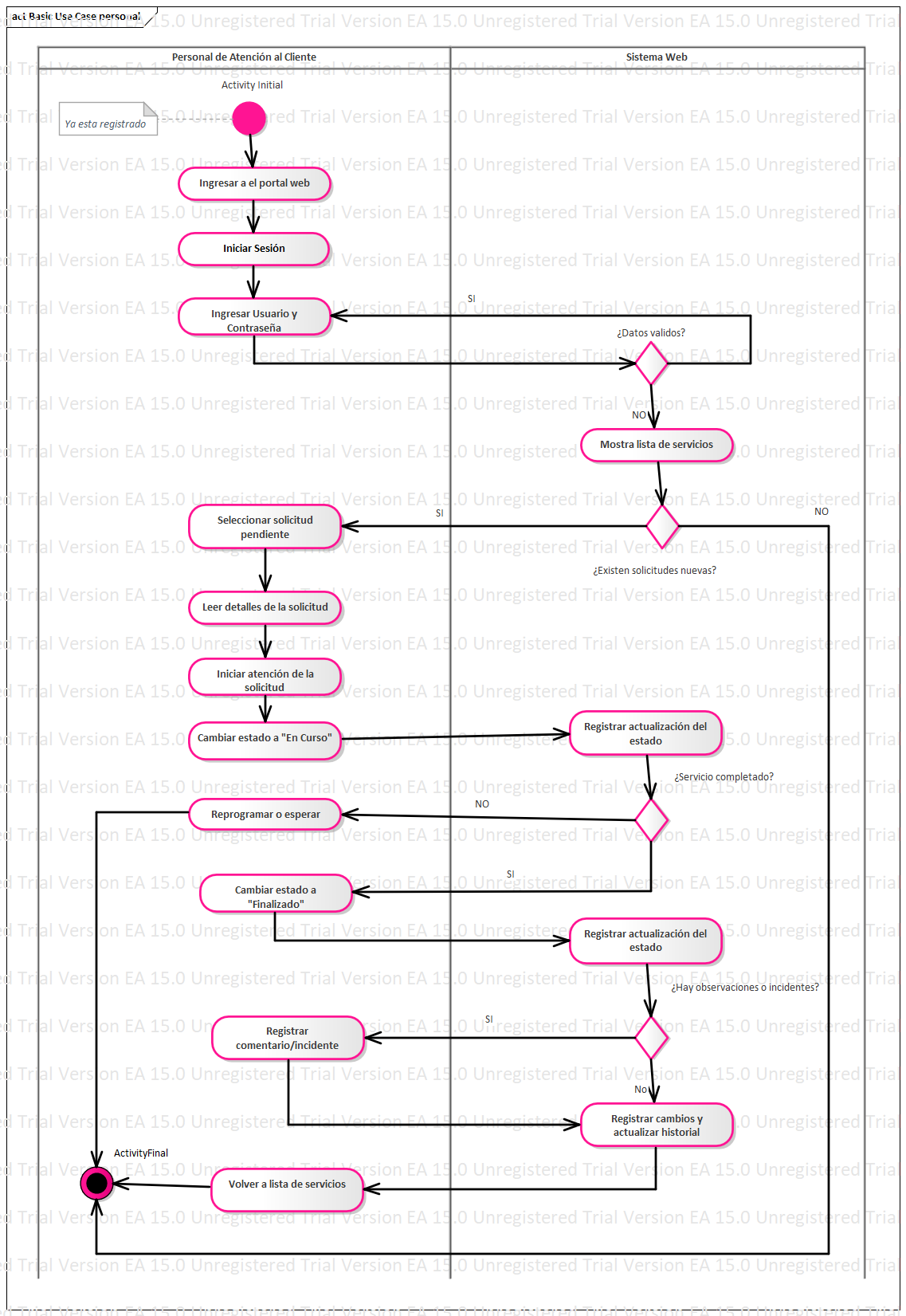
### Huésped

Diagrama

El contenido generado por IA puede ser incorrecto.

**Ilustración 2** Diagrama de Actividades - Huésped

### Personal de Atención al Cliente



**Ilustración 3** Diagrama de Actividades - Personal de Atención al Cliente

### Administrador del Sistema

Diagrama

El contenido generado por IA puede ser incorrecto.

**Ilustración 4** Diagrama de Actividades - Administrador del Sistema

## Diagrama de contexto

Diagrama

El contenido generado por IA puede ser incorrecto.

**Ilustración 5** Diagrama de Contexto

## Modelo Relacional – Base de Datos

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Ilustración 6** Modelo Relacional - Base de Datos

# Diccionario de Datos

**Tabla 4**   
Cliente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idCliente** | int | — | PK, NOT NULL | Identificador único del cliente. |
| **nombre** | varchar | 50 | NOT NULL | Nombre del cliente. |
| **apellido** | varchar | 50 | NOT NULL | Apellido del cliente. |
| **email** | varchar | 100 | NOT NULL | Correo electrónico del cliente. |
| **telefono** | varchar | 15 | NOT NULL | Número de teléfono del cliente. |
| **idTipoDoc** | int | 11 | NOT NULL | Identificador de tipo de Documento del cliente |
| **NroDoc** | varchar | 30 | NOT NULL | Número de documento el cliente |
| **Nacionalidad** | varchar | 50 | NOT NULL | Nacionalidad del cliente |
| **Estado** | enum (activo, inactivo, suspendido) | — | NOT NULL | estado del cliente |

**Tabla 5**   
Empleado

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idEmpleado** | int | — | PK, NOT NULL | Identificador único del empleado. |
| **nombre** | varchar | 50 | NOT NULL | Nombre del empleado. |
| **apellido** | varchar | 50 | NOT NULL | Apellido del empleado. |
| **dni** | char | 8 | UNIQUE, NOT NULL | Documento de identidad del empleado. |
| **email** | varchar | 100 | UNIQUE, NOT NULL | Correo electrónico del empleado. |
| **telefono** | varchar | 15 | — | Número de teléfono del empleado. |
| **idFuncion** | int | — | FK → Funcion  (idFuncion) | Referencia al rol o función del empleado. |
| **Estado** | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado del empleado |

**Tabla 6**   
Función

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idFuncion** | int | — | PK, NOT NULL | Identificador único de la función. |
| **detalle** | text | — | NOT NULL | Descripción del rol o función del empleado. |

**Tabla 7**   
Habitación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idHabitacion** | int | 11 | PK, NOT NULL | Identificador único de la habitación. |
| **numero** | varchar | 10 | UNIQUE, NOT NULL | Número de habitación. |
| **idTipoHbt** | int | 11 | FK → **tipohabitacion**  (**idTipoHbt**) | Tipo de habitación (simple, doble, etc.). |
| **precio** | decimal | 10.2 | NOT NULL | Precio de la habitación |
| **estado** | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado actual (disponible, ocupada, etc.). |

**Tabla 8**  
Categoría Servicio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idCtgServicio** | int | — | PK, NOT NULL | Identificador único de la categoría de servicio. |
| **detalle** | text | — | NOT NULL | Detalle específico del servicio. |
| **descripcion** | text | — | — | Descripción general de la categoría. |
| **precio** | decimal | — | NOT NULL | Precio base de la categoría de servicio. |
| **estado** | char | — | NOT NULL | Estado del servicio (activo, inactivo, etc.). |

**Tabla 9**  
Servicio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| idServicio | int | 11 | PK, NOT NULL | Identificador único del servicio. |
| idHabitacion | int | 11 | FK → Habitacion  (idHabitacion) | Relaciona el servicio con una habitación. |
| idCtgServicio | int | 11 | FK → CategoriaServicio  (idCtgServicio) | Categoría del servicio ofrecido. |
| idCliente | int | 11 | FK → Cliente  (idCliente) | Cliente que solicita el servicio. |
| estado | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado del servicio (activo, finalizado, cancelado, etc.). |

**Tabla 10**  
Reseña

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idRsnServicio** | int | 11 | PK, NOT NULL | Identificador único de la reseña. |
| **idServicio** | int | 11 | FK servicios -> idServicios | identificado del servicio |
| **calificacion** | int | 11 | NOT NULL | Puntuación otorgada al servicio (ej. 1 a 5). |
| **comentario** | text | — | — | Comentario adicional del cliente. |
| **fchResenia** | datetime | —- | NOT NULL | Fecha de la reseña |

**Tabla 11**  
Actividad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| idActividad | int | 11 | PK, NOT NULL | Identificador único de la Actividad. |
| detalle | text | — | NOT NULL | Detalle específico de la actividad. |
| descripcion | text | — | — | Descripción general del evento. |
| duracion | int | 11 | — | Duración total de la actividad en días o horas. |
| precio | decimal | 10,2 | NOT NULL | Costo de la actividad. |
| estado | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado de la actividad (programado, realizado, cancelado). |

**Tabla 12**  
Función Servicio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idFunSer** | int | 11 | PK, NOT NULL | Identificador único de la FuncionserServicio. |
| **idFuncion** | int | 11 | FK → Funcion  (idFuncion) | Referencia al id del función. |
| **idCtgServicio** | int | 11 | FK → CategoriaServicio  (idCtgServicio) | Referencia al id de CategoriaServicio |

**Tabla 13**  
Categoría Comida

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idCtgComida** | int | 11 | PK, NOT NULL | Identificador único de la CategoriaComida. |
| **descripcion** | text | — | NOT NULL | Descripcion de la CategoriaComida. |
| **estado** | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado de CategoriaComida (pendiente, confirmada, etc.). |

**Tabla 14**  
Horario Actividad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idHraActividad** | int | 11 | PK, NOT NULL | Identificador único de la HorarioActividad. |
| **IdActividad** | int | 11 | FK → actividad  (idActividad) | referencia a id de actividad. |
| **cupoMax** | int | 11 | NOT NULL | Máxima cantidad de horario actividD |
| **fchInicio** | datetime | — | NOT NULL | Fecha de inicio de actividad |
| **fchFin** | datatime | — | NOT NULL | Fecha de finalización de actividad |
| **estado** | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado de HorarioActividad (pendiente, confirmada, etc.). |

**Tabla 15**  
Inscripción

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idInspripcion** | int | 11 | PK, NOT NULL | Identificador único de la inscripción. |
| **idHraActividad** | int | 11 | FK → horarioActividad  (idActividad) | Referencia a id de hora actividad. |
| **idCliente** | int | 11 | FK → cliente  (idCliente) | Referencia a id de cliente |
| **fchInscripcion** | datetime | — | NOT NULL | Fecha de inscripción |
| **estado** | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado de inscripción (pendiente, confirmada, etc.). |

**Tabla 16**  
Producto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idProducto** | int | 11 | PK, NOT NULL | Identificador único de producto. |
| **nombre** | text | — | NOT NULL | Nombre del producto |
| **descripcion** | text | — | NOT NULL | Descripción del producto |
| **precio** | decimal | 10,2 | NOT NULL | Precio del producto |
| **idCtgComida** | int | 11 | FK → categoriaComida  (idCtgComida) | Referencia a id de categoría comida |
| **estado** | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado de producto (pendiente, confirmada, etc.). |

**Tabla 17**  
Reserva Habitación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| idReservaHbt | int | 11 | PK, NOT NULL | Identificador único de reserva habitación. |
| idCliente | int | 11 | FK → cliente  (idCliente) | Referencia a id del cliente. |
| idHabitacion | int | 11 | FK → habitación  (idHabitacion) | Referencia a id de habitación. |
| fchInicio | datetime | — | NOT NULL | Fecha de inicio |
| fchFin | datetime | — | NOT NULL | Techa de salida |
| horaSalida | datetime | — | NOT NULL | Hora de salida |
| estado | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado de reserva (pendiente, confirmada, etc.). |

**Tabla 18**  
Servicio Comida

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| **idSerCom** | int | 11 | PK, NOT NULL | Identificador único de Servicio comida. |
| **idProducto** | int | 11 | FK → producto  (idProducto) | Referencia a id del Producto. |
| **idServicio** | int | 11 | FK → servicio  (idServicio) | Referencia a id de Servicio. |
| **cantidad** | int | 11 | NOT NULL | Cantidad consumida |

**Tabla 19**  
Tipo Documento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| idTipoDoc | int | 11 | PK, NOT NULL | Identificador único de tipo de documento. |
| nombre | varchar | 50 | NOT NULL | Nombre del documento |

**Tabla 20**  
Usuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| idCuenta | int | 11 | PK, NOT NULL | Identificador único de usuario. |
| contraseña | varchar | 250 | NOT NULL | contraseña del usuario |
| idCliente | int | 11 | FK → cliente  (idCliente) | Referencia al id del cliente |
| idEmpledado | int | 11 | FK → empleado  (idEmpleado) | Referencia al id del empleado |
| estado | enum (activo, inactivo, suspendido) | — | NOT NULL | Estado de usuario (Activo, Inactivo, etc.). |

**Tabla 21**  
Tipo de Habitaciones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| idTipoHbt | int | 11 | PK, NOT NULL | Identificador único de tipo de habitación. |
| nombre | text | — | NOT NULL | Nombre de tipo de habitación |
| descripcion | text | — | NOT NULL | Descripción del tipo de habitación |
| tamano | varchar | 10 | NOT NULL | Referencia al id del empleado |
| cama | text | — | NOT NULL | Tipo de cama |
| adultos | int | 11 | NOT NULL | Cantidad de adultos |
| ninos | int | 11 | NOT NULL | cantidad de niños |
| precio | decimal | 10.0 | NOT NULL | Precio de tipo de habitación |

**Tabla 22**  
Asignación Servicio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Longitud** | **Clave / Restricción** | **Descripción** |
| idAsigSer | int | 11 | PK, NOT NULL | Identificador único de Asignación Servicio. |
| idEmpleado | int | 11 | FK → empleado(idEmpleado) | Referencia al id del empleado |
| idServicio | int | 11 | FK → servicio(idServicio) | Referencia al id del Servicio |
| fchAsignacion | datetime | — | NOT NULL | Fecha de asignación |

# Interfaz de usuario gráfica, Aplicación, Teams El contenido generado por IA puede ser incorrecto.Diagrama Web

**Ilustración 7** Diagrama Web

# Instalación y Configuración del Sistema

## Instalación del Sistema

* **Descarga** o **clona** el repositorio del sistema desde la fuente oficial del proyecto.
* Abre una **terminal** o **consola de comandos** y accede a la carpeta principal del sistema.
* Instala todas las dependencias necesarias utilizando el gestor de paquetes de **Node.js**.
* Crea una base de datos nueva en tu servidor **MySQL** local o remoto.
* Ejecuta los **archivos SQL** proporcionados para generar la estructura de tablas y procedimientos almacenados.
* Crea un **archivo de configuración** donde se definan las variables del entorno, como el nombre de la base de datos, usuario, contraseña y puerto del servidor.
* Una vez **configurado el entorno**, ejecuta el servidor para iniciar el sistema.
* Abre un **navegador web** e ingresa a la dirección correspondiente (por ejemplo, localhost con el puerto definido) para verificar que el sistema esté funcionando correctamente.

## Configuración del Entorno

* Verifica que tengas instalado **Node.js** en tu equipo, ya que es necesario para ejecutar el sistema.
* Asegúrate de tener un servidor **MySQL** activo y accesible para la conexión a la base de datos.
* Crea un **archivo de configuración** donde se definan las variables de entorno necesarias, como el nombre de la base de datos, usuario, contraseña, host y puerto.
* Verifica que las **rutas internas del sistema** estén correctamente organizadas, separando vistas, controladores, archivos públicos y rutas.
* Si el sistema utiliza **imágenes** o **recursos multimedia**, asegúrate de que las carpetas correspondientes estén creadas y tengan los permisos adecuados.
* Revisa que las conexiones con la **base de datos** estén correctamente configuradas en los archivos del sistema.
* Si se desea migrar a un **entorno de pruebas o producción**, ajusta las variables de entorno y configura el servidor correspondiente según el entorno deseado.

# Administración del Sistema

Esta sección detalla los procedimientos esenciales para mantener el sistema en óptimas condiciones, incluyendo tareas de mantenimiento, respaldo y recuperación de datos, así como monitoreo continuo del estado del sistema.

## Mantenimiento

El mantenimiento debe realizarse de forma periódica para garantizar la estabilidad, el rendimiento y la seguridad del sistema. Las actividades recomendadas son:

* Verificar que el servidor y los servicios asociados (como **Node.js** y **MySQL**) estén activos y funcionando correctamente.
* **Revisar** y actualizar las dependencias del proyecto para aplicar mejoras o corregir vulnerabilidades.
* Eliminar **archivos temporales** o **registros antiguos** que puedan afectar el rendimiento o ocupar espacio innecesario.
* Comprobar que el archivo de configuración del entorno **(.env)** contenga datos correctos y actualizados.
* Realizar pruebas funcionales periódicas en los módulos principales del sistema para detectar errores a tiempo.

## Respaldo y Recuperación

Para evitar la pérdida de información, se recomienda establecer una política de respaldo y recuperación de datos. Las buenas prácticas incluyen:

**Respaldo**

* Generar copias de seguridad de la base de datos de forma **diaria o semanal**, según la frecuencia de uso del sistema.
* Almacenar los respaldos en un lugar seguro, con acceso restringido al **personal autorizado**.
* Registrar la fecha de cada respaldo para facilitar su identificación en caso de necesitar una restauración.

**Recuperación**

* En caso de **errores, fallos o pérdida de datos**, restaurar la base de datos desde el **respaldo más reciente**.
* Realizar **pruebas de restauración** en un entorno de pruebas para verificar la validez de los respaldos.
* Documentar el **procedimiento de recuperación** paso a paso, asegurando que el personal técnico pueda ejecutarlo con rapidez.

## Monitoreo

El monitoreo es clave para anticiparse a problemas, detectar fallos a tiempo y garantizar la disponibilidad del sistema. Se recomienda:

* Utilizar herramientas de supervisión **(como PM2)** para mantener el servidor activo y detectar caídas o reinicios automáticos.
* Registrar los **errores del sistema** y revisar periódicamente los logs de actividad para detectar comportamientos inusuales.
* Implementar servicios de monitoreo externo que notifiquen al personal técnico ante:
* interrupciones del servicio
* alto consumo de recursos
* errores críticos
* Configurar **alertas automáticas** para actuar con rapidez ante incidentes.
* Evaluar regularmente el estado general del sistema para asegurar que cumple con los estándares de rendimiento y estabilidad definidos.