



## Especificaciones Sistema de Gestión de Hoteles

Murcia, a 12 de marzo de 2024

© AUNNA IT – Reservados todos los derechos.

[www.aunnait.es](http://www.aunnait.es)

## Contenido

1.	INTRODUCCIÓN.....	3
1.1.	OBJETIVO .....	3
1.2.	ALCANCE.....	3
2.	REQUISITOS FUNCIONALES .....	3
3.	BASE DE DATOS.....	4
3.1.	MODELO DE DATOS.....	4
3.2.	DIAGRAMA E-R .....	5
4.	ENTIDADES .....	6
5.	TECNOLOGÍAS Y HERRAMIENTAS .....	7

# 1. INTRODUCCIÓN

## 1.1. OBJETIVO

Este documento tiene como objetivo proporcionar una visión detallada de las especificaciones técnicas del sistema de gestión de hoteles. Este sistema administrará hoteles que disponen de varios servicios (como gimnasio, piscina, etc.), habitaciones dentro de estos hoteles, y huéspedes alojados en las habitaciones.

Este documento servirá como guía para el desarrollo e implementación del servicio.

## 1.2. ALCANCE

Para el desarrollo de este proyecto se implementará la BDD, los servicios, los controladores, y los tests unitarios mediante mockito. Toda la funcionalidad del sistema debe ser accesible mediante swagger.

Además de los tests unitarios, también se realizarán pruebas funcionales localmente a través de postman y a través de swagger.

Para poder llevar un control y seguimiento del proyecto, se trabajará con GIT y GitHub. El repositorio de GitHub debe ser público. Se utilizará la opción de GitHub gh-pages, para publicar el swagger del proyecto y que este sea accesible a través de una URL pública.

Las entidades sobre las que se construirá el sistema son las siguientes:

- **Hoteles:** entidad que se dedica al hospedaje para personas que se encuentran en tránsito.
- **Servicios:** son los servicios que puede disponer un hotel (gimnasio, lavandería, bar, etc.).
- **Habitaciones:** habitaciones disponibles en el hotel para el alojamiento de huéspedes.
- **Huéspedes:** son los clientes del hotel que realizan una estancia en alguna de las habitaciones.

# 2. REQUISITOS FUNCIONALES

El servicio cumplirá con los siguientes requisitos funcionales:

## 1. Gestión de Hoteles:

- a. **RF1.1:** el servicio permitirá registrar nuevos hoteles, al menos 2 atributos serán obligatorios.
- b. **RF1.2:** deberá ser posible actualizar la información de los hoteles existentes.
- c. **RF1.3:** el sistema permitirá eliminar hoteles. Esta acción eliminará las habitaciones y huéspedes relacionados en cascada.
- d. **RF1.4:** se podrá consultar los hoteles disponibles, filtrando por cualquiera de sus atributos.
- e. **RF1.5:** el servicio devolverá los resultados de forma paginada. Es decir, se podrán elegir tantos los elementos por página, como la página que devuelva.

## 2. Gestión de Servicios:

- a. **RF2.1:** el servicio permitirá registrar nuevos servicios, al menos 2 atributos serán obligatorios.
- b. **RF2.2:** deberá ser posible actualizar la información de los servicios existentes.
- c. **RF2.3:** el sistema permitirá eliminar servicios.
- d. **RF2.4:** se podrá consultar los servicios disponibles, filtrando por cualquiera de sus atributos.
- e. **RF2.5:** el servicio devolverá los resultados de forma paginada. Es decir, se podrán elegir tantos los elementos por página, como la página que devuelva.

## 3. Gestión de Habitaciones:

- a. **RF2.1:** el servicio permitirá registrar nuevas habitaciones, al menos 2 atributos serán obligatorios.
- b. **RF2.2:** deberá ser posible actualizar la información de las habitaciones existentes.
- c. **RF2.3:** el sistema permitirá eliminar habitaciones. Esta acción eliminará todos los huéspedes asociados.
- d. **RF2.4:** se podrán consultar las habitaciones disponibles, filtrando por cualquiera de sus atributos.
- e. **RF2.5:** el servicio devolverá los resultados de forma paginada. Es decir, se podrán elegir tantos los elementos por página, como la página que devuelva.

## 4. Gestión de Huéspedes:

- a. **RF2.1:** el servicio permitirá registrar nuevos huéspedes, al menos 2 atributos serán obligatorios.
- b. **RF2.2:** deberá ser posible actualizar la información de los huéspedes existentes.
- c. **RF2.3:** el sistema permitirá eliminar huéspedes.
- d. **RF2.4:** se podrá consultar los huéspedes disponibles, filtrando por cualquiera de sus atributos.
- e. **RF2.5:** el servicio devolverá los resultados de forma paginada. Es decir, se podrán elegir tantos los elementos por página, como la página que devuelva.

## 3. BASE DE DATOS

La base de datos deberá estar diseñada para almacenar, organizar y gestionar de forma eficiente toda la información relacionada con los hoteles, servicios, habitaciones y huéspedes.

Los atributos de cada una de las entidades estarán definidos siguiendo el estilo *camelCase*.

### 3.1. MODELO DE DATOS

#### • HOTELES:

- **Atributos:** ID, Nombre, Dirección, Teléfono, Email, Sitio Web.
- **Relaciones:** un hotel puede tener asociados varios servicios y muchas habitaciones.

- **SERVICIOS:**

- **Atributos:** ID, Nombre, Descripción, Categoría.
- **Relaciones:** un servicio puede estar asociado a varios hoteles.

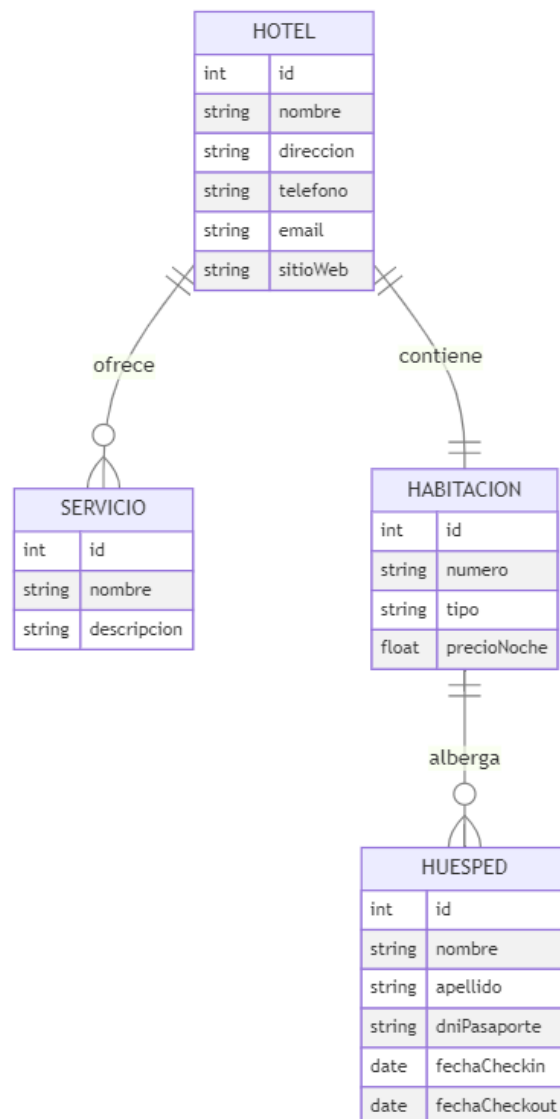
- **HABITACIONES:**

- **Atributos:** ID, Número, Tipo, Precio por noche.
- **Relaciones:** una habitación está asociada a un único hotel. Una habitación puede albergar varios huéspedes.

- **HUÉSPEDES:**

- **Atributos:** ID, Nombre, Apellido, DNI/Pasaporte, Email, Teléfono, Fecha de Check-in, Fecha de Check-out.
- **Relaciones:** está asociado a una única habitación.

### 3.2. DIAGRAMA E-R



## 4. ENTIDADES

- HOTELES

Atributo	Tipo de Dato	Descripción	Tipo Clave
id	int	Identificador único del hotel	PK
nombre	string	Nombre del hotel	
direccion	string	Dirección del hotel	
telefono	string	Número de teléfono del hotel	
email	string	Correo electrónico del hotel	
sitioWeb	string	URL del sitio web del hotel	

- SERVICIOS

Atributo	Tipo de Dato	Descripción	Tipo Clave
id	int	Identificador único del servicio	PK
nombre	string	Nombre del servicio	
descripcion	string	Descripción detallada del servicio	

- HABITACIONES

Atributo	Tipo de Dato	Descripción	Tipo Clave
id	int	Identificador único de la habitación	PK
idHotel	int	Identificador único del hotel	FK
numero	string	Número o código de la habitación	
tipo	string	Tipo de habitación (ej: simple, doble)	
precioNoche	float	Precio por noche de estancia	

- HUÉSPEDES

Atributo	Tipo de Dato	Descripción	Tipo Clave
id	int	Identificador único de la habitación	PK
nombre	string	Nombre del huésped	Nombre
apellido	string	Apellido del huésped	Apellido
dniPasaporte	string	Documento de identidad o pasaporte del huésped	DNI/Pasaporte
fechaCheckIn	date	Fecha de inicio de la estancia	Fecha de Check-in
fechaCheckOut	date	Fecha de fin de la estancia	Fecha de Check-out
nombre	string	Nombre del huésped	Nombre

## 5. TECNOLOGÍAS Y HERRAMIENTAS

- **Lenguaje de Programación:** PHP y JavaScript
- **Framework de Desarrollo Web:** Laravel (back) y Vue.js (front)
- **Base de Datos:** MariaDB
- **Gestor de BBDD:** DBeaver
- **Control de Versiones:** GIT
- **Repositorio:** GitHub
- **Editor:** Visual Studio Code





En AUNNA IT creemos en la tecnología e innovación que potencia y genera el cambio, desde una relación de atención cercana, continuada y de confianza con sus clientes.

**José Esparza Muñoz**  
*Project Manager*

M: +34 600 58 79 20  
C: [jesparza@aunnait.es](mailto:jesparza@aunnait.es)  
W: <http://www.aunnait.es>

**AUNNA IT**

C/ Central nº17. Torre Zambudio

T: 968 97 87 29  
C: [info@aunnait.es](mailto:info@aunnait.es)  
W: <http://www.aunnait.es>

## CONFIDENCIALIDAD

Este documento contiene información confidencial propiedad de AUNNA IT. La información solo puede ser divulgada a aquellas personas autorizadas por la naturaleza de sus responsabilidades a recibir dicha información, y exclusivamente para su uso en relación con el proyecto cuyo enfoque de solución se presenta en este documento. No puede ser distribuida, duplicada o utilizada con un tercero en todo o en parte sin la expresa autorización por escrito de AUNNA IT.