

# Arquitectura del Sistema - Reserva Hotel

Este documento describe la arquitectura elegida para el sistema de reservas de hotel. La arquitectura combina Clean Architecture con principios Hexagonales (Ports & Adapters) para asegurar escalabilidad, mantenibilidad y desacoplamiento del negocio respecto a frameworks y tecnologías.

## Comparativa de Arquitecturas

Criterio	Clean (Uncle Bob)	Onion	Hexagonal
Idea principal	Capas concentricas	Dominios en el centro	Entradas/Salidas via Ports & Adapters
Dependencias	Solo hacia adentro	Igual que Clean	Core define Ports
Uso comun y sistemas multicanal	APIs REST escalables	Apps con logica fuerte	Microservicios
Complejidad inicial	Media	Media-alta	Alta

## Arquitectura Elegida

La arquitectura elegida para el proyecto es una combinacion de Clean Architecture con principios Hexagonales (Ports & Adapters). Esta mezcla permite mantener el core del negocio independiente de frameworks como NestJS o Prisma.

- Clean Architecture para capas bien definidas y dependencias hacia el core.
- Hexagonal para manejar entradas y salidas desacopladas con adaptadores (REST, Websockets, CronJobs, etc.).

Esta eleccion permite crear un sistema profesional y preparado para escalar a microservicios en el

futuro.

## **Ventajas de Clean + Hexagonal**

- Escalable: permite separar facilmente en microservicios.
- Testable: logica core puede probarse sin frameworks externos.
- Desacoplado: no depende de NestJS, Prisma ni JWT.

Ideal para sistemas con multiples roles y posibles canales de entrada (REST, Websockets, CLI).