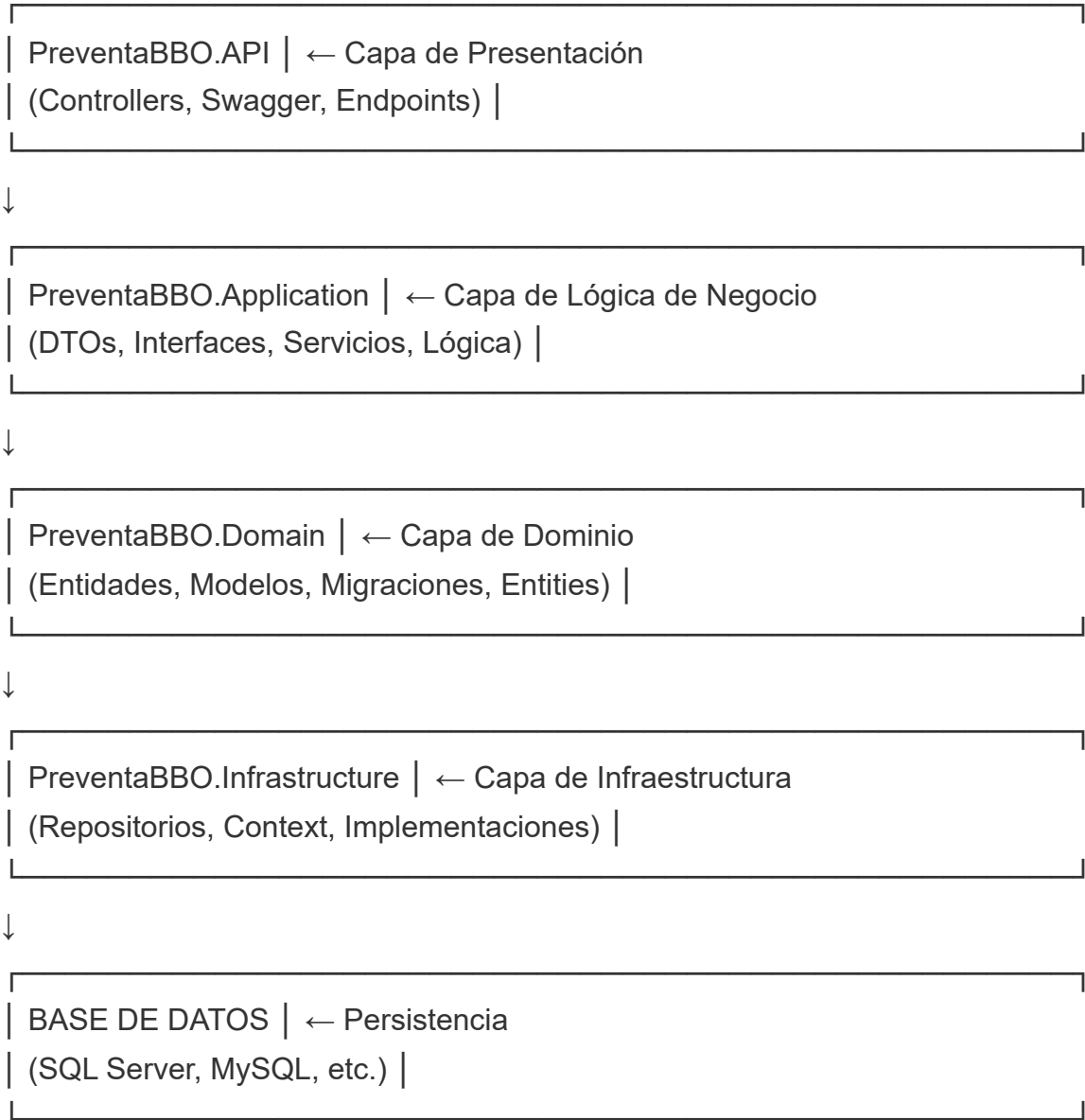


# Documentación - Arquitectura por Capas

## PreventaBBO

### Arquitectura General

#### Diagrama de Capas



# Descripción de Cada Capa

## 1. PreventaBBO.API - Capa de Presentación

**Propósito:** Punto de entrada de la aplicación - maneja las solicitudes HTTP y respuestas.

**Responsabilidades:**

- Exponer endpoints RESTful
- Documentación automática con Swagger
- Autenticación y autorización HTTP
- Serialización/Deserialización JSON
- Validación básica de requests
- Manejo de códigos de estado HTTP

**Componentes típicos:**

- Controladores (Controllers)
- Configuración de Swagger
- Middleware personalizado
- Filtros de acción

## 2. PreventaBBO.Application - Capa de Lógica de Negocio

**Propósito:** Contiene toda la lógica de negocio y reglas de la aplicación.

**Responsabilidades:**

- Implementar casos de uso
- Validaciones de negocio
- Transformación de datos (DTOs ↔ Entidades)
- Coordinación entre diferentes componentes
- Aplicación de reglas empresariales

**Componentes típicos:**

- **DTOs:** Objetos para transferir datos entre capas
- **Interfaces:** Contratos para los repositorios y servicios
- **Servicios:** Clases que implementan lógica de negocio compleja
- **Validadores:** Reglas de validación específicas del negocio

### 3. PreventaBBO.Domain - Capa de Dominio

**Propósito:** Representa el corazón del negocio - entidades y modelos centrales.

**Responsabilidades:**

- Definir entidades del dominio
- Configurar relaciones entre entidades
- Manejar migraciones de base de datos
- Contener enums y value objects del dominio
- Definir contratos de repositorio (interfaces)

**Componentes típicos:**

- **Entidades:** Clases que representan tablas de la base de datos
- **Context:** DbContext de Entity Framework
- **Migraciones:** Scripts de evolución de la base de datos
- **Configuraciones:** Fluent API para mapeo de entidades

### 4. PreventaBBO.Infrastructure - Capa de Infraestructura

**Propósito:** Implementa el acceso a datos y concerns técnicos.

**Responsabilidades:**

- Implementar repositorios concretos
- Acceso a base de datos
- Configuración de Entity Framework
- Implementación de patrones de datos
- Conexiones externas (APIs, servicios)

**Componentes típicos:**

- **Repositorios:** Implementaciones concretas de las interfaces
- **Configuraciones:** Conexión strings, configuraciones técnicas
- **Context Implementation:** Implementación del DbContext
- **Servicios externos:** Clientes HTTP, integraciones

# Conexión entre Capas

## Flujo de Dependencias

API → Application → Domain ← Infrastructure

**Regla:** Las dependencias siempre van hacia adentro

- API depende de Application
- Application depende de Domain
- Infrastructure depende de Domain

## Conexión a Base de Datos

### Configuración de Entity Framework

**En Domain:**

- Define el DbContext y las Entidades
- Configura relaciones con Fluent API
- Maneja migraciones de base de datos

**En Infrastructure:**

- Implementa los Repositorios concretos
- Inyecta el DbContext via Dependency Injection
- Maneja transacciones y conexiones

**En API:**

- Registra dependencias en el contenedor DI
- Configura connection strings
- Aplica migraciones al iniciar