📘 Documento — Sistema de Gestión de Turnos Médicos

Estilo minimalista, colores suaves y secciones colapsables (Headings).

# 🟦 BE — Entidades (Dominio)

Personas y cosas del mundo real que el sistema maneja. Lo explicamos en criollo y con ejemplos.

• Paciente: La persona que pide atención médica. Ej: “Ana Pérez, DNI 30111222”. Se usa para saber quién tiene cada turno.

• Profesional: Médico/a que atiende. Ej: “Dra. Carla Suárez, Matrícula M-1234”. Permite asignar turnos a un médico concreto.

• Especialidad: Área médica. Ej: “Cardiología”, “Pediatría”. Ayuda a filtrar profesionales y tipos de turnos.

• Turno: La cita médica con día y hora. Ej: “Ana tiene turno con la Dra. Suárez, lun 10:00, Clínica Médica”. Es el corazón del sistema.

• HistorialClinico: Anotaciones de cada atención. Ej: “Control anual, presión 120/80”. Queda vinculado al paciente y profesional.

• Usuario: Persona que usa el sistema (no confundir con Paciente). Ej: recepcionista, admin, médico. Sirve para login.

• Rol: Conjunto de permisos. Ej: Admin, Recepcionista, Médico. Determina qué puede hacer un Usuario.

• Permiso: Acciones específicas. Ej: TURNOS\_CREAR, PACIENTES\_VER. Los roles agrupan permisos.

# 🟨 DTOs / ViewModels

Paquetes de datos para transportar información entre capas o preparar datos para pantallas.

* PacienteDTO — Datos simples listos para mostrar o enviar (sin reglas).
* ProfesionalDTO — Datos simples listos para mostrar o enviar (sin reglas).
* EspecialidadDTO — Datos simples listos para mostrar o enviar (sin reglas).
* TurnoDTO — Datos simples listos para mostrar o enviar (sin reglas).
* HistorialClinicoDTO — Datos simples listos para mostrar o enviar (sin reglas).
* UsuarioDTO — Datos simples listos para mostrar o enviar (sin reglas).
* RolDTO — Datos simples listos para mostrar o enviar (sin reglas).
* PermisoDTO — Datos simples listos para mostrar o enviar (sin reglas).
* ReporteTurnosDTO — Datos simples listos para mostrar o enviar (sin reglas).
* PaginacionRequest/Response — Datos simples listos para mostrar o enviar (sin reglas).

# 🟧 BLL — Servicios / Reglas de Negocio

• IPacienteService / PacienteService: Reglas para altas/bajas/modificaciones y búsqueda de pacientes.

• IProfesionalService / ProfesionalService: Gestión de profesionales y sus especialidades.

• IEspecialidadService / EspecialidadService: Catálogo de especialidades.

• ITurnoService / TurnoService: Agenda y validaciones: horarios válidos, conflicto de turnos, cambio de estado.

• IHistorialClinicoService / HistorialClinicoService: Carga y consulta del historial clínico del paciente.

• IAutenticacionService / AutenticacionService: Login/logout con contraseñas seguras.

• IAutorizacionService / AutorizacionService: Chequeo de permisos según rol del usuario.

# 🟨 MPP — Mapeo entre Capas/BD

* IMapper<TOrigen, TDestino> — Interfaz genérica para traducir de BD/DTO a Entidad y viceversa.
* PacienteMapper — Traductor de datos: BD ⇄ Entidad.
* ProfesionalMapper — Traductor de datos: BD ⇄ Entidad.
* EspecialidadMapper — Traductor de datos: BD ⇄ Entidad.
* TurnoMapper — Traductor de datos: BD ⇄ Entidad.
* HistorialClinicoMapper — Traductor de datos: BD ⇄ Entidad.
* UsuarioMapper — Traductor de datos: BD ⇄ Entidad.

# 🟩 DAL — Acceso a Datos / Repositorios

• IPacienteRepository / PacienteRepository: Guardar y buscar pacientes en SQL.

• IProfesionalRepository / ProfesionalRepository: Guardar y buscar profesionales en SQL.

• IEspecialidadRepository / EspecialidadRepository: Catálogo de especialidades en SQL.

• ITurnoRepository / TurnoRepository: CRUD de turnos + consultas por fecha/estado.

• IHistorialClinicoRepository / HistorialClinicoRepository: Guardar y listar entradas de historial clínico.

# 🟪 Seguridad — Login, Roles y Permisos

• UsuarioService / AutenticacionService: Maneja login seguro (hash de contraseñas).

• AutorizacionService / PermissionChecker: Verifica si un usuario puede hacer una acción.

# 🟫 Validación y Excepciones

* TurnoValidator — valida fecha futura, rango horario y conflictos
* UsuarioValidator — valida formato de usuario/contraseña
* DominioException — errores de reglas de negocio
* RepositorioException — errores de acceso a datos

# ⬛ Utilidades / Cross-cutting

* ILogger / FileLogger — registro de acciones y errores
* DateTimeProvider — para testear con fechas controladas
* ConfigurationProvider — lee connection strings
* Csv/Excel/PdfExporter — exportación de reportes

# 🧩 Tabla de Roles y Permisos

Qué puede hacer cada rol dentro del sistema. ✅ permitido / ❌ no permitido.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rol / Permiso | Login | Ver pacientes | ABM pacientes | Dar turnos | Cancelar turnos | Ver historial | Cargar historial | ABM usuarios/roles |
| Admin | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |
| Recepcionista | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ❌ | ❌ |
| Médico | ✅ | ❌ | ❌ | ✅ (solo propios) | ✅ (solo propios) | ✅ (solo propios) | ✅ (solo propios) | ❌ |
| Supervisor | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ❌ | ❌ |

# 🧭 Diagramas (estilo simple)

## 🎭 Diagrama de Roles

[Admin] —— gestiona usuarios/roles  
[Recepcionista] —— gestiona turnos y pacientes  
[Médico] —— atiende y registra historial  
[Supervisor] —— consulta reportes

## 🔀 Diagrama de Secuencia de Roles — Caso: Dar Turno

Paciente Recepcionista Sistema Profesional  
 | | | |  
 | solicita | | |  
 |--------------->| | |  
 | | busca agenda | |  
 | |----------------->| |  
 | | | verifica disp. |  
 | | |------------------->|  
 | | | confirma |  
 | | crea Turno |<-------------------|  
 | |<-----------------| |  
 | recibe conf. | | |  
 |<---------------| | |

## 🧩 Diagrama Entidad-Relación (DER) con cardinalidad

Paciente (1) ── (0..\*) Turno (0..\*) ── (1) Profesional  
 │  
 └── (1) Especialidad  
Paciente (1) ── (0..\*) HistorialClinico (1) ── Profesional  
Profesional (1) ── (1..\*) Agenda ── (1..\*) Disponibilidad (días/horarios)  
Profesional (M) ── (N) Especialidad => tabla intermedia ProfesionalEspecialidad

## 🎬 Diagrama de Casos de Uso (listado visual)

Actores: Paciente, Recepcionista, Médico, Admin  
Casos:  
 - Gestionar Pacientes (Recepcionista, Admin)  
 - Dar/Cancelar Turno (Recepcionista)  
 - Ver Agenda (Recepcionista, Médico)  
 - Atender Turno y Cargar Historial (Médico)  
 - Generar Reportes (Supervisor/Admin)  
 - Login/Logout (Todos los usuarios internos)

## 🔎 Entrada — Comportamiento — Salida (E-C-S)

|  |  |  |  |
| --- | --- | --- | --- |
| Proceso | Entrada | Comportamiento | Salida |
| Dar Turno | Entrada: DNI paciente, especialidad, profesional (opcional), fecha/hora deseada. | Comportamiento: validar paciente, buscar disponibilidad, evitar solapamiento, crear turno en estado Pendiente. | Salida: turno creado con ID y confirmación. |
| Atender Turno | Entrada: ID turno, observaciones del médico. | Comportamiento: cambiar estado a Atendido, crear entrada en HistorialClinico. | Salida: confirmación y registro en historial. |
| Cancelar Turno | Entrada: ID turno, motivo. | Comportamiento: reglas de cancelación (por tiempo, por rol), cambio de estado a Cancelado. | Salida: confirmación y liberación de la agenda. |

# 📄 Documentación del código

## 🧩 Mini historias de uso (quién / qué hace / en base a qué datos)

### Proceso: Dar Turno (Recepción)

* 👤 Quién lo hace: Recepcionista, a pedido del Paciente.
* 🛠️ Qué hace: Busca disponibilidad por especialidad/profesional y agenda un turno.
* 🗂️ En base a qué datos: DNI del paciente, especialidad elegida, fecha/hora; agenda del profesional.

### Proceso: Atender Turno (Consulta)

* 👤 Quién lo hace: Médico.
* 🛠️ Qué hace: Atiende al paciente y registra la atención en el historial.
* 🗂️ En base a qué datos: Turno asignado, observaciones clínicas, diagnóstico.

### Proceso: Cancelar Turno

* 👤 Quién lo hace: Recepcionista o Paciente (por teléfono), con política definida.
* 🛠️ Qué hace: Cambia el estado a Cancelado y libera el horario.
* 🗂️ En base a qué datos: ID de turno, motivo, política de cancelación (tiempo previo).

### Proceso: Gestión de Usuarios y Roles

* 👤 Quién lo hace: Admin.
* 🛠️ Qué hace: Crea usuarios, asigna roles y permisos.
* 🗂️ En base a qué datos: Usuario, Rol, Permisos disponibles.

# 📚 Glosario de términos

Entidad: Objeto del mundo real que queremos guardar (Paciente, Turno).

Repositorio: Capa que guarda y busca entidades en la base de datos.

Servicio (BLL): Reglas de negocio que combinan entidades y repositorios.

DTO/Mapper: Formato de datos para mover info entre capas y su traductor.

Autenticación: Proceso de verificar que un usuario es quien dice ser (login).

Autorización: Revisar si tiene permiso para hacer algo (roles/permisos).

# 📖 Teoría con ejemplos

## POO — Programación Orientada a Objetos

• Abstracción: Quedarse con lo importante y ocultar lo irrelevante. Ej: Paciente tiene nombre/DNI; no guardamos datos que no usamos.

• Encapsulamiento: Cada objeto maneja sus propios datos y reglas. Ej: Turno valida su fecha/estado, no cualquiera lo cambia.

• Herencia: Compartir cosas comunes. Ej (si quisieras): UsuarioBase → MedicoUsuario / AdminUsuario.

• Polimorfismo: Poder usar distintas clases de la misma forma. Ej: distintos IRepositorio<T> con igual interfaz (SQL, memoria).

## SOLID — Principios

• S — Responsabilidad Única: Cada clase con un motivo de cambio. Ej: TurnoService gestiona turnos; no envía emails.

• O — Abierto/Cerrado: Podés extender sin modificar. Ej: agregar DapperRepository sin tocar la interfaz IRepositorio.

• L — Sustitución de Liskov: Si algo promete ser de un tipo, debe comportarse como tal. Ej: cualquier ITurnoRepository debe respetar las reglas.

• I — Segregación de Interfaces: Interfaces chicas y específicas. Ej: IBusquedaTurnos e IAltaTurnos si hace falta separarlas.

• D — Inversión de Dependencias: Dependé de abstracciones, no de concretos. Ej: BLL usa ITurnoRepository, no SqlTurnoRepository directo.

## DAS — Desarrollo y Arquitectura de Software (capas)

Organizamos el proyecto en capas para separar responsabilidades y facilitar mantenimiento:

* UI → Formularios y experiencia del usuario.
* BLL → Reglas de negocio (qué se puede o no).
* DAL → Acceso a la base de datos.
* MPP/DTO → Traducción de datos entre BD y objetos.
* Seguridad/Validación/Utils → Funciones transversales.