📘 Sistema de Gestión de Turnos Médicos — Documento Rediseñado

Estilo minimalista, colores suaves, encabezados colapsables. Sin MPP ni DTO. Diagramas con flechas ASCII.

# 🟦 BE — Entidades (Dominio)

Objetos reales que maneja el sistema. Explicados en lenguaje simple y con ejemplos.

• Paciente: Persona que pide atención. Ej: “Ana Pérez, DNI 30111222”. Sirve para identificar a quién pertenece cada turno.

• Profesional: Médico/a que atiende. Ej: “Dra. Carla Suárez (M-1234)”. Permite asignar turnos a un profesional concreto.

• Especialidad: Área médica. Ej: “Cardiología”, “Pediatría”. Filtra profesionales y turnos por tipo de atención.

• Turno: Cita médica con fecha y hora. Ej: “Ana con la Dra. Suárez, lunes 10:00”. Es el núcleo del sistema.

• HistorialClinico: Registro de lo atendido. Ej: “Control anual, presión 120/80”. Queda vinculado a paciente y profesional.

• Usuario: Persona que usa el sistema (recepcionista, médico, admin). Se usa para iniciar sesión (login).

• Rol: Grupo de permisos. Ej: Admin, Recepcionista, Médico. Define qué puede hacer cada usuario.

• Permiso: Acciones concretas (TURNOS\_CREAR, PACIENTES\_VER). Los roles combinan permisos.

# 🟦 UI — Interfaz de Usuario (WinForms)

Pantallas con las que interactúa la gente. Deben ser simples, claras y coherentes con los permisos del usuario.

• LoginForm: Pantalla para ingresar usuario y contraseña. Si es válido, abre el sistema con su rol.

• MainForm (Menú): Ventana principal con accesos a módulos (Pacientes, Turnos, Agenda, Usuarios, Reportes).

• PacientesForm: ABM de pacientes: alta, búsqueda, edición. Muestra DNI, nombre, contacto.

• ProfesionalesForm: ABM de profesionales y sus especialidades.

• EspecialidadesForm: Catálogo de especialidades disponibles.

• TurnosForm: Agenda y gestión de turnos por fecha, profesional y estado.

• HistorialClinicoForm: Carga y consulta de atenciones realizadas.

• UsuariosForm / RolesPermisosForm: Administración de usuarios, roles y permisos.

• ReportesForm: Indicadores (ocupación, cancelaciones) y exportación a CSV/Excel/PDF.

# 🟧 BLL — Servicios / Reglas de Negocio

• IPacienteService / PacienteService: Altas/bajas/modificaciones y búsqueda de pacientes.

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

• IEspecialidadService / EspecialidadService: Catálogo de especialidades.

• ITurnoService / TurnoService: Agenda y validaciones: horarios válidos, evitar solapamientos, cambio de estado.

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

• IAutenticacionService / AutenticacionService: Login/Logout con hashes de contraseña.

• IAutorizacionService / AutorizacionService: Chequea permisos por rol/acción.

# 🟩 DAL — Acceso a Datos / Repositorios

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

• IProfesionalRepository / ProfesionalRepository: Guardar y buscar profesionales.

• IEspecialidadRepository / EspecialidadRepository: Catálogo de especialidades.

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

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

• UnitOfWork (opcional): Coordina transacciones cuando hay varias escrituras relacionadas.

• SqlConnectionFactory / SqlExecutor: Crea conexiones y ejecuta SQL parametrizado (ADO.NET).

# 🟪 Seguridad — Login, Roles y Permisos

• AutenticacionService: Valida credenciales con contraseña hasheada.

• AutorizacionService / PermissionChecker: Permite o bloquea acciones según rol/permisos.

• PasswordPolicy / LoginAttemptTracker: Política de contraseñas y bloqueo por intentos fallidos.

# 🟫 Validación y Excepciones

• TurnoValidator: Valida fecha futura, rango horario, no solapamiento.

• UsuarioValidator: Valida usuario/contraseña con política mínima.

• DominioException: Errores de reglas de negocio (lo que no se debe permitir).

• RepositorioException: Errores de acceso a datos (SQL, conexión).

# ⬛ Utilidades / Cross-cutting

• ILogger / FileLogger: Registro de operaciones y errores para auditoría.

• DateTimeProvider: Centraliza la hora actual para pruebas consistentes.

• ConfigurationProvider: Lee connection strings y configuraciones.

• Csv/Excel/PdfExporter: Exporta listados/indicadores para informes.

# 🧩 Tabla de Roles y Permisos

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

# 🧭 Diagramas (ASCII con flechas)

## 🎭 Diagrama de Roles

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

## 🔀 Secuencia de Roles — Caso: Dar Turno

Paciente -> Recepcionista: solicita turno  
Recepcionista -> Sistema: busca agenda (especialidad/profesional/fecha)  
Sistema -> Profesional: verifica disponibilidad  
Profesional -> Sistema: confirma disponibilidad  
Sistema -> Recepcionista: crea Turno (Pendiente)  
Recepcionista -> Paciente: envía confirmación

## 🧩 DER — Entidad-Relación con cardinalidad

Paciente (1) --> (0..\*) Turno (0..\*) --> (1) Profesional  
 |  
 +--> (1) Especialidad  
Paciente (1) --> (0..\*) HistorialClinico (1) --> (1) Profesional  
Profesional (M) --> (N) Especialidad (tabla intermedia: ProfesionalEspecialidad)

## 🎬 Casos de Uso (listado con flechas)

Actores: Paciente, Recepcionista, Médico, Admin  
Recepcionista -> Gestionar Pacientes  
Recepcionista -> Dar/Cancelar Turno  
Recepcionista -> Ver Agenda  
Médico -> Ver Agenda  
Médico -> Atender Turno y Cargar Historial  
Admin -> Generar Reportes  
Todos -> Login/Logout

# 📚 Glosario de términos

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

Repositorio: Cajón donde guardamos/buscamos entidades en la BD.

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

Autenticación: Verificar identidad del usuario (login).

Autorización: Comprobar permisos para una acción.

# 📖 Teoría con ejemplos

## POO — Programación Orientada a Objetos

• Abstracción: Quedarse con lo importante. Ej: Paciente tiene nombre/DNI; no guardamos datos irrelevantes.

• Encapsulamiento: Cada objeto cuida sus datos/reglas. Ej: Turno no permite fechas pasadas.

• Herencia: Compartir lo común. Ej: (opcional) UsuarioBase → Admin/Médico.

• Polimorfismo: Usar distintos objetos de la misma forma. Ej: ITurnoRepository con distintas implementaciones (SQL/memoria).

## SOLID — Principios

• S — Responsabilidad Única: Una clase, un motivo de cambio. TurnoService no envía emails.

• O — Abierto/Cerrado: Extender sin modificar. Podés agregar otro repositorio sin tocar la interfaz.

• L — Sustitución de Liskov: Si promete ser ITurnoRepository, debe comportarse como tal.

• I — Segregación de Interfaces: Interfaces chicas y específicas para evitar obligar a implementar de más.

• D — Inversión de Dependencias: Depender de interfaces, no de clases concretas. BLL depende de IRepos, no de SQL directo.

## DAS — Arquitectura en Capas

• UI → Pantallas y experiencia

• BLL → Reglas de negocio

• DAL → Acceso a datos (ADO.NET)

• Seguridad/Validación/Utils → funciones transversales