🩺 ÉPICA 1 — Gestión de Autenticación y Usuarios

1.1 Login de usuario (Front y Back)

Como doctor
quiero iniciar sesión en el sistema
para acceder a las funcionalidades según mi rol.

Criterios de aceptación:

- El sistema valida credenciales desde la base de datos.
- En caso de error, se muestra el mensaje “Usuario o contraseña incorrectos”.
- Se genera un token de sesión válido (JWT).
- Se redirige al panel principal.

1.2 Recuperar contraseña (Back)

Como usuario
quiero restablecer mi contraseña
para recuperar el acceso en caso de olvido.

Criterios de aceptación:

- Se genera un token temporal de recuperación.
- Se envía un correo electrónico con link para definir nueva contraseña.
- El token expira tras 15 minutos o un solo uso.

1.3 Crear nuevo usuario (Back y Front)

Como administrador
quiero registrar un nuevo usuario del sistema
para permitir que un nuevo doctor pueda acceder.

Criterios de aceptación:

- Se almacenan nombre, email, rol y contraseña encriptada.
- No se permiten correos duplicados.
- El sistema confirma el registro.

1.4 Cerrar sesión (Front y Back)

Como usuario
quiero cerrar mi sesión
para asegurar la confidencialidad de los datos médicos.

Criterios de aceptación:

- El token de sesión se invalida en el servidor.
- Se redirige al login.
- No debe poder accederse al sistema mediante el botón “Atrás”.

🧾 ÉPICA 2 — Gestión de Pacientes e Historias Clínicas

2.1 Crear paciente (Front y Back)

Como doctor
quiero registrar un nuevo paciente
para almacenar su historia clínica digital.

Criterios de aceptación:

- Se valida que los campos obligatorios estén completos.
- Se verifica que no exista el DNI.
- Se crea un registro de historia clínica asociado.

2.2 Editar paciente (Front y Back)

Como doctor
quiero modificar datos del paciente
para mantener su información actualizada.

Criterios de aceptación:

- Los cambios se guardan en la base de datos.
- El sistema registra la fecha y usuario que editó.

2.3 Eliminar paciente (Back)

Como doctor
quiero eliminar un paciente sin consultas activas
para mantener limpia la base de datos.

Criterios de aceptación:

- No puede eliminarse si tiene consultas.
- Requiere confirmación previa.

2.4 Buscar paciente (Front + Back)

Como doctor
quiero buscar un paciente por nombre o DNI
para acceder rápidamente a su perfil.

Criterios de aceptación:

- Permite búsqueda parcial.
- Retorna lista de coincidencias.
- Al seleccionar, abre la historia clínica.

🧬 ÉPICA 3 — Gestión de Consultas Médicas

3.1 Crear consulta (Front y Back)

Como doctor
quiero registrar una nueva consulta médica
para documentar la atención del paciente.

Criterios de aceptación:

- Se guarda fecha, motivo, diagnóstico, prescripciones y archivos adjuntos.
- Se asocia automáticamente al paciente.

3.2 Editar consulta (Back)

Como doctor
quiero actualizar una consulta registrada
para corregir o ampliar información.

Criterios de aceptación:

- Solo el médico creador puede modificarla.
- Se registra la fecha de actualización.

3.3 Eliminar consulta (Back)

Como doctor
quiero eliminar una consulta
para depurar registros duplicados o erróneos.

Criterios de aceptación:

- Requiere confirmación.
- No se elimina si pertenece a un reporte emitido.

3.4 Visualizar consulta (Front)

Como doctor
quiero acceder al detalle de una consulta
para revisar la información médica registrada.

Criterios de aceptación:

- Muestra los datos de la consulta en modo lectura.
- Permite navegar por fecha.

3.5 Adjuntar archivos o imágenes (Back y Front)

Como doctor
quiero subir archivos de estudios e imágenes
para tener los registros médicos completos del paciente.

Criterios de aceptación:

- Acepta formatos JPG, PNG y PDF.
- Valida tamaño máximo antes de guardar.
- Los archivos quedan asociados a la consulta.

📅 ÉPICA 4 — Gestión de Turnos del Día

4.1 Visualizar turnos del día (Front y Back)

Como doctor o secretario
quiero ver los pacientes del día
para organizar las consultas.

Criterios de aceptación:

- Muestra pacientes con estado: “En espera”, “Atendido”, “Ausente”, “Cancelado”.
- Permite actualizar estado manualmente.
- Acceso directo al perfil del paciente.

4.2 Crear turno del día (Back)

Como secretario
quiero registrar un nuevo turno del día
para incluir pacientes no agendados.

Criterios de aceptación:

- Se vincula a un paciente existente.
- Se define fecha, hora y estado inicial.

4.3 Editar turno del día (Back)

Como secretario
quiero modificar un turno existente
para corregir horario o estado.

Criterios de aceptación:

- Solo editable si aún no fue atendido.
- Se actualiza en tiempo real en la vista del día.

4.4 Eliminar turno del día (Back)

Como secretario
quiero eliminar un turno
para mantener actualizada la agenda.

Criterios de aceptación:

- Se pide confirmación.
- Se elimina el registro y actualiza la lista visible.

⚙️ ÉPICA 5 — Back End / API / Persistencia

5.1 Implementar estructura base de API

Como desarrollador
quiero crear una API RESTful con Node.js y Express
para proveer endpoints seguros para el sistema.

Criterios de aceptación:

- Rutas separadas por entidad (usuarios, pacientes, consultas, turnos).
- Responde en formato JSON.
- Maneja errores con códigos HTTP adecuados.

5.2 Configurar conexión a base de datos

Como desarrollador
quiero establecer la conexión con la base de datos
para permitir persistencia de datos.

Criterios de aceptación:

- Maneja reconexiones automáticas.
- Credenciales en variables de entorno.
- Archivo único de configuración (`db.js`).

5.3 Implementar autenticación con JWT

Como desarrollador
quiero implementar autenticación basada en tokens
para proteger los endpoints.

Criterios de aceptación:

- Genera token JWT al iniciar sesión.
- Rutas protegidas validan token.

- Tokens expiran automáticamente.

5.4 Implementar validación de roles

Como desarrollador
quiero controlar el acceso según rol (doctor, secretario, admin)
para garantizar la seguridad.

Criterios de aceptación:

- Middleware de autorización por rol.
- Los doctores solo acceden a sus pacientes.
- Los secretarios solo a turnos.

5.5 Generación de PDF

Como desarrollador
quiero generar informes PDF de historias clínicas
para exportar información médica.

Criterios de aceptación:

- El PDF incluye datos del paciente y consultas.
- Puede descargarse o enviarse.
- Se conserva una copia en base de datos.

5.6 Pruebas y validaciones del Back End

Como desarrollador
quiero crear pruebas unitarias y de integración
para garantizar correcto funcionamiento del servidor.

Criterios de aceptación:

- Tests para login, pacientes y consultas.

- Todos deben pasar antes del despliegue.

💻 ÉPICA 6 — Front End / Interfaz y Navegación (sin conexión)

6.1 Página de login

Como usuario
quiero acceder a una pantalla de inicio de sesión
para ingresar al sistema.

Criterios de aceptación:

- Validación de campos vacíos.
- Mensajes de error visibles.
- Botón “Ingresar” funcional (mock).
- Link “¿Olvidaste tu contraseña?”.

6.2 Página principal / Dashboard

Como doctor
quiero visualizar un panel principal
para acceder a las secciones principales.

Criterios de aceptación:

- Menú lateral o superior.
- Datos simulados en JSON.
- Diseño responsivo.

6.3 Lista de pacientes

Como doctor
quiero visualizar pacientes
para acceder a su historial clínico.

Criterios de aceptación:

- Tabla con Nombre, DNI, Cobertura, Acciones.
- Búsqueda y orden con JavaScript.
- Carga desde JSON o localStorage.

6.4 Interfaz de registro (Nuevo usuario)

Como nuevo usuario

quiero registrarme en el sistema mediante un formulario

para crear mi cuenta de acceso al sistema de historias clínicas.

Criterios de aceptación:

- Campos: nombre completo, correo, contraseña, confirmar contraseña, pregunta de seguridad, respuesta.
- Validaciones: correo con formato correcto, contraseñas coincidentes y mínimo de 8 caracteres.
- Botón “Crear cuenta”.
- Mensaje de éxito al completar el registro.
- Redirección al login tras el registro exitoso.

6.5 Historia clínica del paciente

Como doctor
quiero ver la historia clínica completa
para acceder a todas sus consultas.

Criterios de aceptación:

- Encabezado con datos del paciente.
- Lista de consultas ordenadas.
- Botón para nueva consulta.

6.6 Formulario de nueva consulta

Como doctor
quiero registrar una nueva consulta
para documentar la atención.

Criterios de aceptación:

- Campos: Motivo, Diagnóstico, Prescripción, Fecha.
- Validaciones básicas.
- Guardado temporal.

6.7 Página de turnos del día

Como doctor o secretario
quiero ver la lista de pacientes del día
para gestionar sus estados.

Criterios de aceptación:

- Tabla con estados actualizables.
- Cambios guardados en memoria.
- Acceso a la historia del paciente.

6.8 Exportar historia clínica a PDF

Como doctor
quiero generar PDF imprimible
para guardarlo o entregarlo al paciente.

Criterios de aceptación:

- Botón “Exportar PDF”.
- Uso de `jsPDF` o similar.
- Archivo con datos de ejemplo.

6.9 Diseño visual general

Como usuario
quiero un diseño claro y profesional
para facilitar la lectura.

Criterios de aceptación:

- Paleta azul/celeste.
- Tipografía legible.
- CSS Grid o Flexbox.

6.10 Navegación entre pantallas

Como usuario
quiero moverme entre secciones sin recargar
para tener una experiencia fluida.

Criterios de aceptación:

- Navegación SPA simulada.
- Botones “Inicio” o “Atrás”.
- Muestra breadcrumb actual.

6.11 Interfaz de recuperación de contraseña

Como usuario registrado
quiero poder recuperar mi contraseña si la olvidé
para volver a acceder al sistema sin perder mis datos.

Criterios de aceptación:

- Paso 1: ingresar correo electrónico.
- Paso 2: mostrar la pregunta de seguridad asociada.
- Paso 3: validar respuesta.
- Paso 4: permitir definir una nueva contraseña.
- Mensaje final de confirmación.
- Validaciones visuales en cada paso.

6.12 Interfaz de configuración de cuenta

Como doctor
quiero acceder a una pantalla de configuración de mi cuenta
para poder modificar mis datos personales y de seguridad.

Criterios de aceptación:

- Campos: nombre completo, correo electrónico, contraseña actual, nueva contraseña, confirmar nueva contraseña, pregunta y respuesta de seguridad.
- Campo adicional sugerido: “Teléfono de contacto” o “Especialidad médica” (opcional).
- Validación de contraseña actual antes de permitir cambios.
- Botón “Guardar cambios”.
- Mensaje de confirmación o error.
-

---

7. Diseño y Configuración de la Base de Datos PostgreSQL

7.1 Configurar entorno de base de datos

Como desarrollador
quiero instalar y configurar PostgreSQL en el entorno del proyecto
para poder almacenar la información persistente del sistema.

Criterios de aceptación:

- Se crea una base de datos llamada `clinica_db`.
- Las credenciales se guardan en un archivo `.env`.
- La conexión se prueba correctamente desde el backend.

7.2 Crear estructura base de tablas

Como desarrollador
quiero definir las tablas `usuario`, `paciente`, `consulta` y `turno`
para representar las entidades principales del sistema.

Criterios de aceptación:

- Se crean las tablas con tipos de datos adecuados.
- Todas las tablas tienen una clave primaria (`PRIMARY KEY`).
- Se aplican las relaciones necesarias (`FOREIGN KEY`).

Tareas técnicas:

- Crear tabla `usuario` con campos: `id`, `mail`, `nombre_completo`, `contraseña`.
- Crear tabla `paciente` con todos los campos personales y médicos.
- Crear tabla `consulta` relacionada con `paciente (id_paciente)`.
- Crear tabla `turno` relacionada con `paciente` y `consulta`.

7.3 Definir relaciones entre entidades

Como desarrollador
quiero establecer las relaciones entre las tablas
para mantener la integridad referencial de los datos.

Criterios de aceptación:

- Un paciente puede tener muchas consultas (`1:N`).
- Un paciente puede tener muchos turnos (`1:N`).
- Cada consulta pertenece a un paciente.
- Cada turno puede estar asociado a una consulta (opcional).
- Las claves foráneas están correctamente indexadas.

7.4 Crear tabla auxiliar de situaciones de turno

Como desarrollador
quiero crear una tabla `situaciones_turno`
para almacenar los estados posibles de los turnos.

Criterios de aceptación:

- Contiene los valores: `llegó`, `no llegó`, `en espera`, `canceló`.
- Relacionada con `turno.situacion` mediante `id_situacion`.
- Evita valores hardcodeados.

7.5 Aplicar restricciones y validaciones

Como desarrollador
quiero aplicar restricciones de integridad y validación
para asegurar la calidad de los datos almacenados.

Criterios de aceptación:

- Campos obligatorios definidos con `NOT NULL`.
- Claves únicas en `dni`, `mail` de paciente y `mail` de usuario.
- Valores lógicos para booleanos (`1° vez`).
- Restricciones de fecha válidas (`fecha_nacimiento < fecha_actual`).

7.6 Población inicial de datos (seeds)

Como desarrollador
quiero insertar registros de prueba en las tablas
para realizar pruebas de integración.

Criterios de aceptación:

- Al menos 2 usuarios (doctor y secretario).
- 5 pacientes con datos variados.
- 5 consultas asociadas a diferentes pacientes.
- 5 turnos con diferentes situaciones.

7.7 Crear scripts SQL versionados

Como desarrollador
quiero crear scripts SQL versionados
para mantener control de los cambios en la estructura.

Criterios de aceptación:

- Scripts separados: `01_create_tables.sql`, `02_insert_seed.sql`, `03_constraints.sql`.
- Se ejecutan correctamente desde consola o desde Node.js.
- Se documenta cada versión del esquema.

7.8 Documentar el modelo entidad-relación (MER)

Como desarrollador
quiero generar un diagrama entidad-relación (ERD)
para visualizar las relaciones entre las tablas.

Criterios de aceptación:

- Se usa herramienta (PgAdmin, Draw.io, Lucidchart o DBDiagram).
- El diagrama incluye claves primarias y foráneas.
- Se guarda como imagen y archivo fuente en la carpeta `/docs`.