

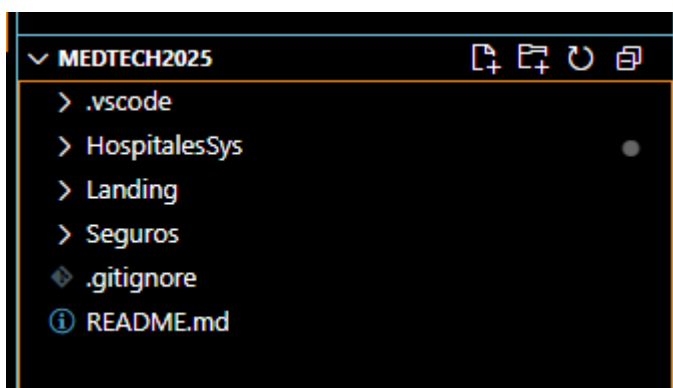
Manual Técnico Hospitales

El Sistema de Hospitales (HospitalesSys) es un módulo de la plataforma MedTech diseñado para gestionar información y procesos de un entorno hospitalario. Su propósito es centralizar la administración de pacientes, médicos, citas y expedientes clínicos.

Este sistema sigue una arquitectura de tres niveles, separando la capa de presentación (interfaz de usuario web), la capa de aplicación (lógica de negocio en el servidor) y la capa de datos (base de datos), esto permite que la interfaz web desarrollada con Svelte en el navegador actúe como nivel de presentación, el servidor Java maneje la lógica de negocio y el esquema de base de datos en Oracle almacene la información persistente. Dentro de la plataforma, HospitalesSys funciona como un servicio independiente pero complementario al sistema de Seguros. Ambos módulos pueden comunicarse a través de APIs para intercambiar información relevante, entre ellos, los servicios que ofrece el seguro por parte del hospital, las citas médicas que se crean en cualquiera de los dos sistemas o los usuarios.

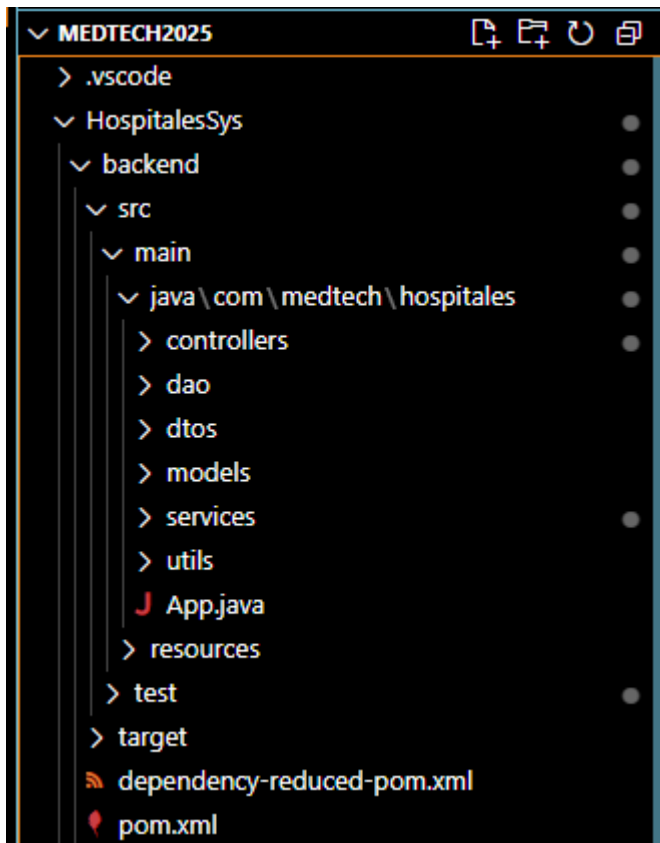
El desarrollo de HospitalesSys involucra varios lenguajes y herramientas modernas. En el backend se utiliza Javelin para facilitar la creación de servicios RESTful y la conexión con la base de datos. La capa de persistencia usa JPA para mapear las entidades a tablas en la base de datos. En el frontend, la aplicación utiliza Svelte como framework principal de la interfaz de usuario. Para el control de versiones se utiliza GitHub y Maven para compilar el proyecto Java. El acceso a la base de datos Oracle se realiza} vía JPA y models o entidades desde Java.

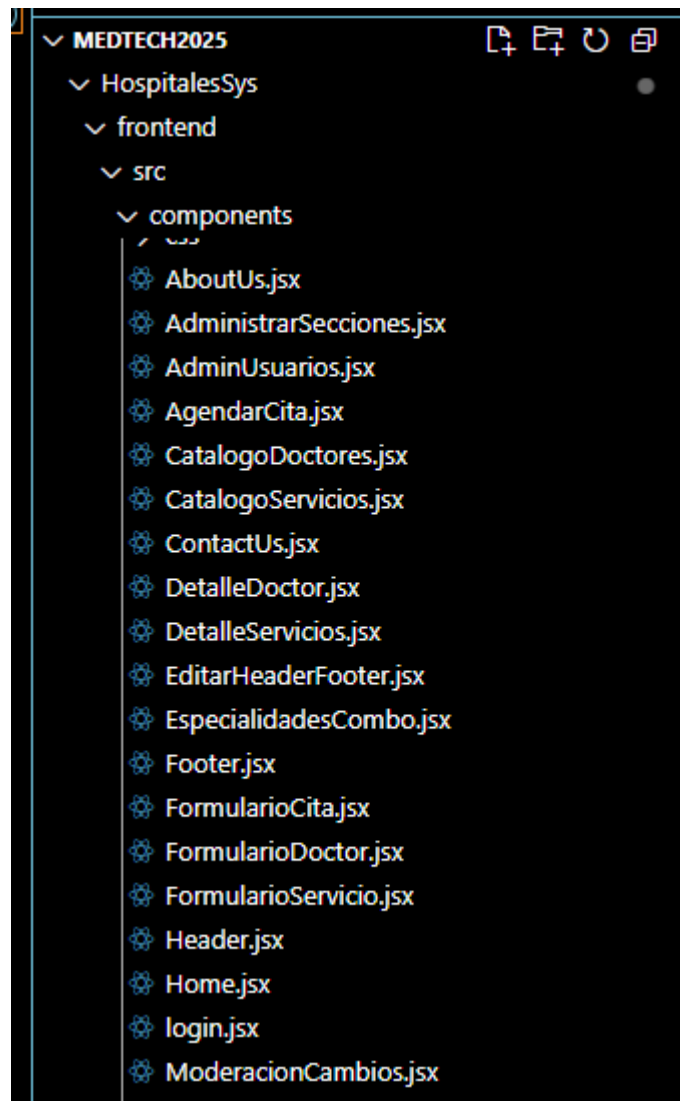
El repositorio está organizado en directorios separados para cada módulo de la plataforma. Los dos módulos principales se encuentran en:

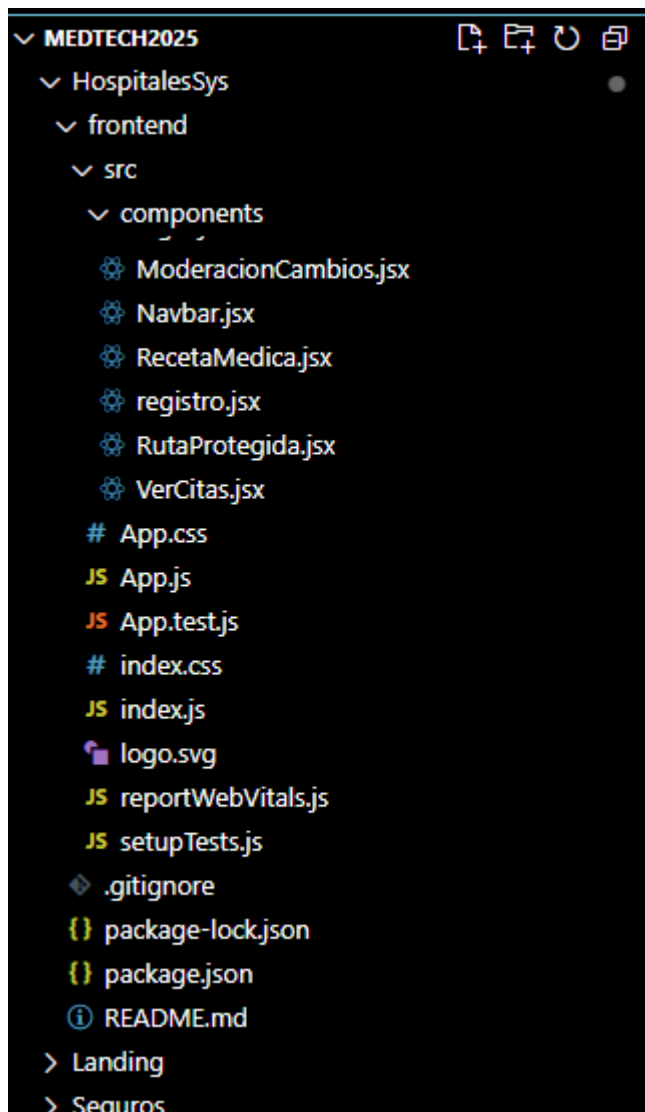


HospitalesSys que contiene el código fuente del sistema de Hospitales. Seguros que contiene el código fuente del sistema de Seguros y Landing que correspondía a una interfaz para unificar el acceso a Hospitales y Seguros.

HospitalesSys está estructurado de la siguiente manera:







Los controladores REST definen las APIs para pacientes, médicos, citas, etc. Los servicios implementan la lógica. Las clases de modelo de datos y entidades JPA mapean las tablas de la base de datos.

HospitalesSys/src/main/java/test muestra las pruebas unitarias o de integración para el módulo hospital.

A continuación, se detallan los pasos para instalar y poner en funcionamiento el sistema de Hospitales en un entorno de desarrollo o producción:

1. Prerrequisitos del entorno Java: Asegurarse de tener instalado Java Development Kit (JDK) en una versión compatible para ejecutar el backend. Verificar con `java -version` en la terminal.
2. Node.js y npm: Instalar [Node.js](#) lo cual incluye npm, necesario para compilar y ejecutar la aplicación Svelte de la interfaz.
3. Herramientas de compilación: Instalar Apache Maven (3.x).
4. Base de datos: Instalar Oracle.

Compilación y construcción del backend

1. En la terminal, en el directorio HospitalesSys/backend ejecutar `mvn clean install` (es necesario hacerlo cada vez que el backend se modifique), con este comando, Maven descargará todas las dependencias declaradas en el pom y se compilará el código Java, incluidas las pruebas unitarias, generando un archivo target en HospitalesSys
2. Para lanzar el servidor de la aplicación, ejecutar seguidamente del `mvn clean install` el comando `java -jar target/hospitales-javalin.jar`, este comando iniciará la aplicación Javelin.
3. Para lanzar el servidor se ejecuta `npm start`, esto hará que corra el sistema y lo podamos ver en el navegador.

El hospital funciona de la siguiente manera:

Gestión de pacientes: Permite crear, editar, visualizar y eliminar registros de pacientes en el sistema hospitalario. Existe una entidad Usuario que representa a cada paciente con atributos como identificador (ID), nombre, apellido, fecha de nacimiento, número de identificación, etc. Un controller REST UsuarioController , expone endpoints como: GET/pacientes para retornar la lista de pacientes registrados, GET/pacientes/{id} para obtener detalles de un paciente específico, POST/pacientes para crear un nuevo paciente, PUT/pacientes/{id} para actualizar la información de un paciente existente, etc. Estos endpoints invocan métodos del UsuarioService que contienen la lógica de negocio, por ejemplo verificar que no exista duplicado de paciente antes de guardar.

Gestión de médicos: El sistema maneja un catálogo de médicos o doctores. Habrá una entidad con datos como ID, nombre, especialidad, etc. y un conjunto de servicios REST: GET para listar todos los doctores disponibles, POST para registrar un nuevo médico, etc. Se permite asociar doctores con su especialidad y servicios que atienden lo cual es esencial para las citas médicas.

Gestión de citas médicas: Es una funcionalidad central del sistema hospitalario: administrar las citas o consultas entre pacientes y médicos en horarios determinados. Aquí interviene la entidad CitaMedica.java que tiene campos como ID, fecha y hora de la cita, referencia al paciente, referencia al médico, motivo o notas. Programar cita a través de un endpoint POST recibe un id de paciente, id de doctor y fecha/hora deseada. El servicio de citas verifica la disponibilidad: por ejemplo, puede comprobar que el médico no tenga otra cita en ese horario, y que la fecha sea futura. Si es válida, guarda la cita en la base. Listar citas: GET devuelve todas las citas programadas por paciente, etc. La implementación en la base de datos puede establecer relaciones: la tabla de Citas tendría claves foráneas hacia la tabla de Usuario y la de Doctoes. En JPA, la entidad CitaMedica tiene @ManyToOne

para reflejar estas relaciones. Al consultar citas, el sistema puede devolver los datos anidados (con JPA) y utilizar DTOs para enviar solo la información necesaria.

Autenticación y roles de usuario: El sistema permite restringir el acceso a la información y su modificación según el rol con el que se cuenta dentro de la página.

Administrador	Puede ver todo, asignar roles, activar usuarios
Doctor	Solo puede ver su agenda, agregar diagnósticos y recetas
Empleado	Gestión operativa, no accede a datos clínicos
Paciente	Visualiza su información y citas

El sistema usa autenticación con tokens JWT.

Integración con el módulo de Seguros: El backend de hospitales cuenta con configuraciones para que a través de endpoints el módulo de seguros pueda acceder a su información que obtiene de la base de datos de Oracle.

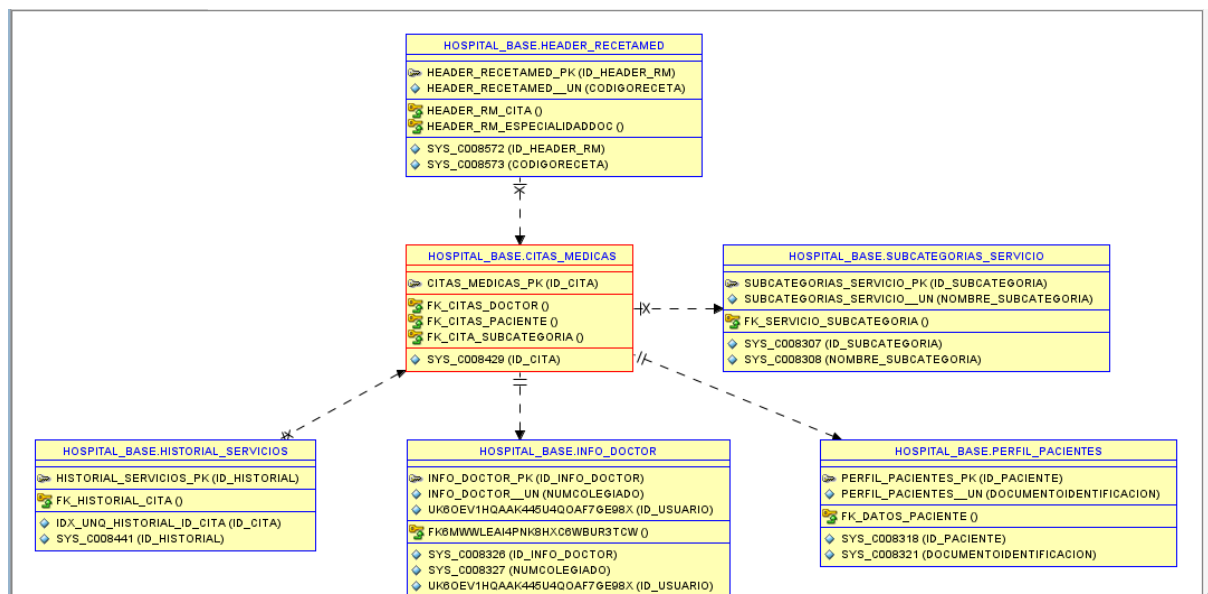
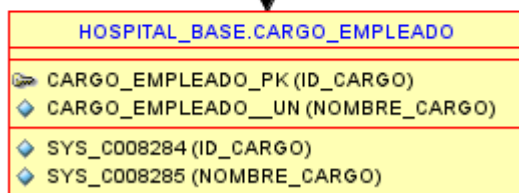
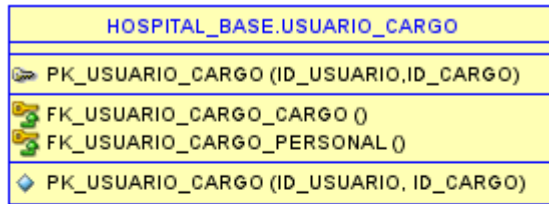
Moderación de contenido: El sistema hospitalario cuenta con un módulo de moderación que garantiza que los cambios en el contenido público del portal no se publiquen automáticamente y se aprueben antes por un administrador.

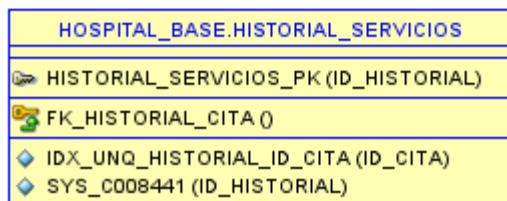
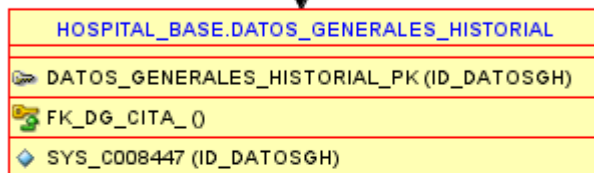
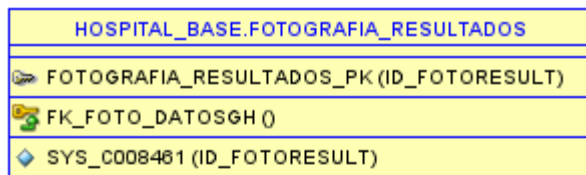
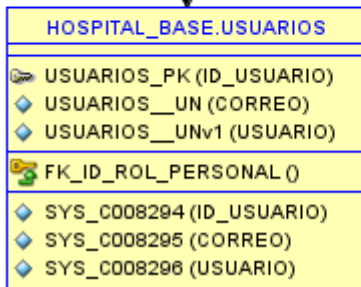
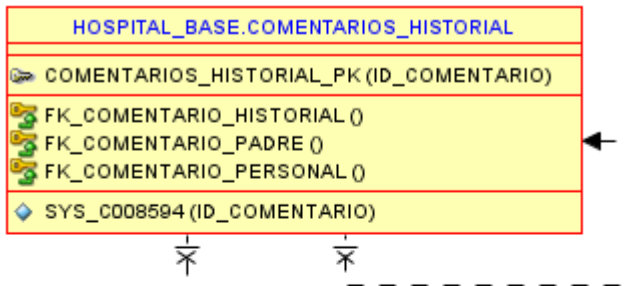
Un empleado con permisos de edición realiza un cambio en una sección y el contenido se guarda en estado draft, el administrador accede al módulo de moderación donde puede visualizar el cambio propuesto comparado con la versión actual, aprobar el contenido o rechazarlo, si se rechaza, se registra un comentario de rechazo, se envía un correo al editor indicando los motivos y el editor podrá modificar la versión en estado borrador y volver a enviarla a revisión. Se tienen los endpoints

GET	/moderacion/cambios	Lista de cambios pendientes
POST	/moderacion/aprobar/{id}	Aprueba un cambio
POST	/moderacion/rechazar/{id}	Rechaza un cambio con comentario
GET	/moderacion/draft/{usuario}	Vista de borradores de ese usuario

El sistema usa CorreoUtils para enviar correos automáticos al editor si el cambio fue rechazado.

La estructura del esquema de base de datos es la siguiente:





HOSPITAL_BASE.ESPECIALIDAD_X_DOCTOR
ESPECIALIDAD_X_DOCTOR_PK (ID_ESPDOC)
FK_ESPECIALIDADDOCTOR ()
FK_INFODOCTOR_ESPECIALIDAD ()
SYS_C008481 (ID_ESPDOC)



HOSPITAL_BASE.ESPECIALIDADES
ESPECIALIDADES_PK (ID_ESPECIALIDAD)
ESPECIALIDADES__UN (NOMBRE_ESPECIALIDAD)
SYS_C008473 (ID_ESPECIALIDAD)
SYS_C008474 (NOMBRE_ESPECIALIDAD)

HOSPITAL_BASE.FOTOGRAFIA_RESULTADOS
FOTOGRAFIA_RESULTADOS_PK (ID_FOTORESULT)
FK_FOTO_DATOSGH ()
SYS_C008461 (ID_FOTORESULT)



HOSPITAL_BASE.DATOS_GENERALES_HISTORIAL
DATOS_GENERALES_HISTORIAL_PK (ID_DATOSGH)
FK_DG_CITA_ ()
SYS_C008447 (ID_DATOSGH)



HOSPITAL_BASE.HEADER_RECETAMED
HEADER_RECETAMED_PK (ID_HEADER_RM)
HEADER_RECETAMED__UN (CODIGORECETA)
HEADER_RM_CITA ()
HEADER_RM_ESPECIALIDADDOC ()
SYS_C008572 (ID_HEADER_RM)
SYS_C008573 (CODIGORECETA)








HOSPITAL_BASE.CITAS_MEDICAS
CITAS_MEDICAS_PK (ID_CITA)
FK_CITAS_DOCTOR ()
FK_CITAS_PACIENTE ()
FK_CITA_SUBCATEGORIA ()
SYS_C008429 (ID_CITA)








HOSPITAL_BASE.HISTORIAL_SERVICIOS
HISTORIAL_SERVICIOS_PK (ID_HISTORIAL)
FK_HISTORIAL_CITA ()
IDX_UNQ_HISTORIAL_ID_CITA (ID_CITA)
SYS_C008441 (ID_HISTORIAL)









HOSPITAL_BASE.CITAS_MEDICAS
CITAS_MEDICAS_PK (ID_CITA)
FK_CITAS_DOCTOR ()
FK_CITAS_PACIENTE ()
FK_CITA_SUBCATEGORIA ()
SYS_C008429 (ID_CITA)





HOSPITAL_BASE.CITAS_MEDICAS
 CITAS_MEDICAS_PK (ID_CITA)
 FK_CITAS_DOCTOR ()
 FK_CITAS_PACIENTE ()
 FK_CITA_SUBCATEGORIA ()
 SYS_C008429 (ID_CITA)














HOSPITAL_BASE.INFO_DOCTOR
 INFO_DOCTOR_PK (ID_INFO_DOCTOR)
 INFO_DOCTOR__UN (NUMCOLEGIADO)
 UK60EV1HQAAK445U4QOAF7GE98X (ID_USUARIO)
 FK6MWWLEAI4PNK8HXC6WBUR3TCW ()
 SYS_C008326 (ID_INFO_DOCTOR)
 SYS_C008327 (NUMCOLEGIADO)
 UK60EV1HQAAK445U4QOAF7GE98X (ID_USUARIO)

HOSPITAL_BASE.MEDICAMENTOS
 MEDICAMENTOS_PK (ID_MEDICAMENTO)
 SYS_C008581 (ID_MEDICAMENTO)

HOSPITAL_BASE.RECETA_MEDICA
 RECETA_MEDICA_PK (ID_HEADER_RM, ID_MEDICAMENTO)
 SYS_C009655 ()
 SYS_C009656 ()
 SYS_C008587 (ID_HEADER_RM, ID_MEDICAMENTO)

HOSPITAL_BASE.OTRO_PERSONAL_SERVICIO
 OTRO_PERSONAL_SERVICIO_PK (ID_OTROPS)
 FK_HISTORIAL_OTROPS ()
 FK_OTRO_PERSONAL_SERVICIO ()
 SYS_C008456 (ID_OTROPS)

HOSPITAL_BASE.HISTORIAL_SERVICIOS
 HISTORIAL_SERVICIOS_PK (ID_HISTORIAL)
 FK_HISTORIAL_CITA ()
 IDX_UNQ_HISTORIAL_ID_CITA (ID_CITA)
 SYS_C008441 (ID_HISTORIAL)

HOSPITAL_BASE.USUARIOS
 USUARIOS_PK (ID_USUARIO)
 USUARIOS__UN (CORREO)
 USUARIOS__UNv1 (USUARIO)
 FK_ID_ROL_PERSONAL ()
 SYS_C008294 (ID_USUARIO)
 SYS_C008295 (CORREO)
 SYS_C008296 (USUARIO)

HOSPITAL_BASE.SECCIONES_PENDIENTES
SECCIONES_PENDIENTES_PK (ID_SECCION_PENDIENTE)
FK_SP_APROBADOR ()
FK_SP_PAGINA ()
FK_SP_SOLICITANTE ()
IDX_SP_ESTADO (ESTADO_APROBACION)
IDX_SP_PAGINA (ID_PAGINA)
IDX_SP_SOLICITANTE (ID_USUARIO_SOLICITANTE)



HOSPITAL_BASE.PAGINAS
PAGINAS_PK (ID_PAGINA)
PAGINAS__UN (NOMBRE)



HOSPITAL_BASE.SECCIONES_PAGINA
SECCIONES_PAGINA_PK (ID_SECCION)
FK_SECCION_PAGINA ()
IDX_SECCIONES_ID_PAGINA (ID_PAGINA)



HOSPITAL_BASE.CITAS_MEDICAS
CITAS_MEDICAS_PK (ID_CITA)
FK_CITAS_DOCTOR ()
FK_CITAS_PACIENTE ()
FK_CITA_SUBCATEGORIA ()
SYS_C008429 (ID_CITA)



HOSPITAL_BASE.PERFIL_PACIENTES
PERFIL_PACIENTES_PK (ID_PACIENTE)
PERFIL_PACIENTES__UN (DOCUMENTOIDENTIFICACION)
FK_DATOS_PACIENTE ()
SYS_C008318 (ID_PACIENTE)
SYS_C008321 (DOCUMENTOIDENTIFICACION)

HOSPITAL_BASE.RECETA_MEDICA
RECETA_MEDICA_PK (ID_HEADER_RM,ID_MEDICAMENTO)
SYS_C009855 ()
SYS_C009856 ()
SYS_C008587 (ID_HEADER_RM, ID_MEDICAMENTO)

HOSPITAL_BASE.HEADER_RECETAMED
HEADER_RECETAMED_PK (ID_HEADER_RM)
HEADER_RECETAMED__UN (CODIGORECETA)
HEADER_RM_CITA ()
HEADER_RM_ESPECIALIDADDOC ()
SYS_C008572 (ID_HEADER_RM)
SYS_C008573 (CODIGORECETA)

HOSPITAL_BASE.MEDICAMENTOS
MEDICAMENTOS_PK (ID_MEDICAMENTO)
SYS_C008581 (ID_MEDICAMENTO)

HOSPITAL_BASE.ROLES_SISTEMA
ROLES_SISTEMA_PK (ID_ROL)
ROLES_SISTEMA__UN (NOMBRE_ROL)
SYS_C008280 (ID_ROL)
SYS_C008281 (NOMBRE_ROL)

HOSPITAL_BASE.USUARIOS
USUARIOS_PK (ID_USUARIO)
USUARIOS__UN (CORREO)
USUARIOS__UNv1 (USUARIO)
FK_ID_ROL_PERSONAL ()
SYS_C008294 (ID_USUARIO)
SYS_C008295 (CORREO)
SYS_C008296 (USUARIO)

HOSPITAL_BASE.SECCIONES_PAGINA
SECCIONES_PAGINA_PK (ID_SECCION)
FK_SECCION_PAGINA ()
IDX_SECCIONES_ID_PAGINA (ID_PAGINA)

HOSPITAL_BASE.SECCIONES_PENDIENTES
SECCIONES_PENDIENTES_PK (ID_SECCION_PENDIENTE)
FK_SP_APROBADOR ()
FK_SP_PAGINA ()
FK_SP_SOLICITANTE ()
IDX_SP_ESTADO (ESTADO_APROBACION)
IDX_SP_PAGINA (ID_PAGINA)
IDX_SP_SOLICITANTE (ID_USUARIO_SOLICITANTE)

HOSPITAL_BASE.PAGINAS
PAGINAS_PK (ID_PAGINA)
PAGINAS__UN (NOMBRE)

HOSPITAL_BASE.PAGINAS
PAGINAS_PK (ID_PAGINA)
PAGINAS__UN (NOMBRE)

HOSPITAL_BASE.SERVICIO_X_DOCTOR
PK_SERV_DOC (ID_SERVICIO, ID_INFO_DOCTOR)
FK_SERVDOC_INFO ()
FK_SERVDOC_SERVICIO ()
PK_SERV_DOC (ID_SERVICIO, ID_INFO_DOCTOR)






HOSPITAL_BASE.INFO_DOCTOR
INFO_DOCTOR_PK (ID_INFO_DOCTOR)
INFO_DOCTOR__UN (NUMCOLEGIADO)
UK80EV1HQAAK445U4Q0AF7GE98X (ID_USUARIO)
FK8MWWLEAI4PNK8HXCBW8UR3TCW ()
SYS_C008326 (ID_INFO_DOCTOR)
SYS_C008327 (NUMCOLEGIADO)
UK80EV1HQAAK445U4Q0AF7GE98X (ID_USUARIO)

HOSPITAL_BASE.SERVICIOS_HOSPITALARIOS
SERVICIOS_HOSPITALARIOS_PK (ID_SERVICIO)
SYS_C008301 (ID_SERVICIO)






HOSPITAL_BASE.SERVICIOS_HOSPITALARIOS
SERVICIOS_HOSPITALARIOS_PK (ID_SERVICIO)
SYS_C008301 (ID_SERVICIO)




HOSPITAL_BASE.SUBCATEGORIAS_SERVICIO
SUBCATEGORIAS_SERVICIO_PK (ID_SUBCATEGORIA)
SUBCATEGORIAS_SERVICIO__UN (NOMBRE_SUBCATEGORIA)
FK_SERVICIO_SUBCATEGORIA ()
SYS_C008307 (ID_SUBCATEGORIA)
SYS_C008308 (NOMBRE_SUBCATEGORIA)








HOSPITAL_BASE.SERVICIO_X_DOCTOR
PK_SERV_DOC (ID_SERVICIO, ID_INFO_DOCTOR)
FK_SERVDOC_INFO ()
FK_SERVDOC_SERVICIO ()
PK_SERV_DOC (ID_SERVICIO, ID_INFO_DOCTOR)

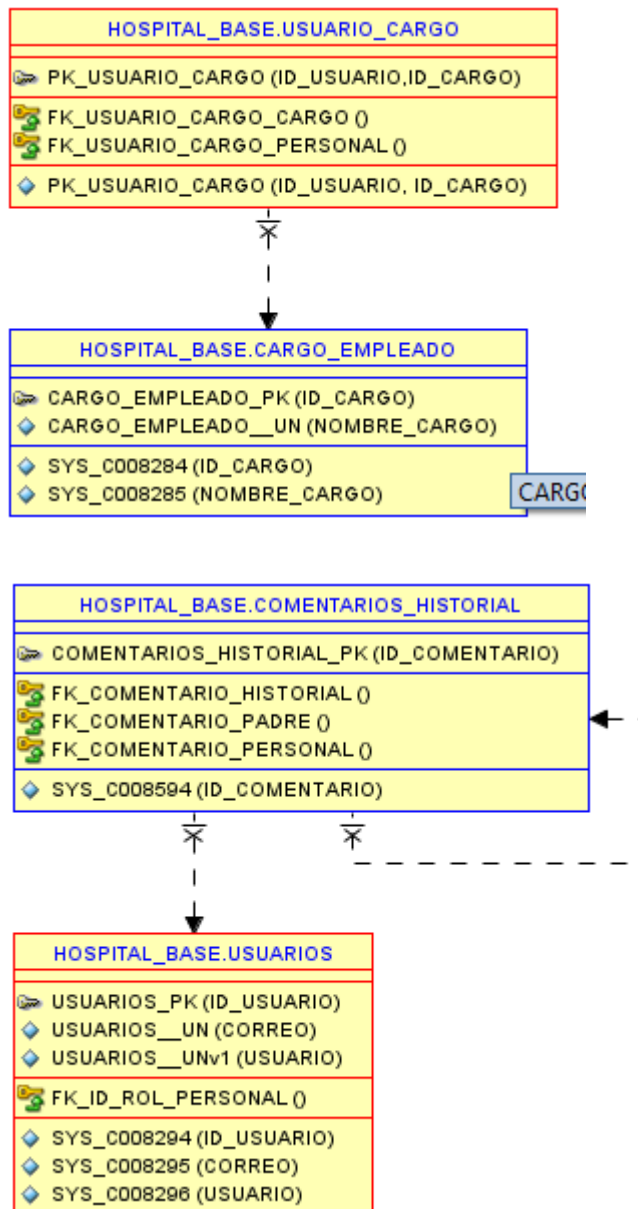
HOSPITAL_BASE.CITAS_MEDICAS
 CITAS_MEDICAS_PK (ID_CITA)
 FK_CITAS_DOCTOR ()
 FK_CITAS_PACIENTE ()
 FK_CITA_SUBCATEGORIA ()
 SYS_C008429 (ID_CITA)



HOSPITAL_BASE.SUBCATEGORIAS_SERVICIO
 SUBCATEGORIAS_SERVICIO_PK (ID_SUBCATEGORIA)
 SUBCATEGORIAS_SERVICIO__UN (NOMBRE_SUBCATEGORIA)
 FK_SERVICIO_SUBCATEGORIA ()
 SYS_C008307 (ID_SUBCATEGORIA)
 SYS_C008308 (NOMBRE_SUBCATEGORIA)

HOSPITAL_BASE.TELEFONO_DOCTOR
 TELEFONO_DOCTOR_PK (ID_TELEFONO)
 FK_INFODOCTOR_TELEFONO ()
 SYS_C008340 (ID_TELEFONO)

HOSPITAL_BASE.INFO_DOCTOR
 INFO_DOCTOR_PK (ID_INFO_DOCTOR)
 INFO_DOCTOR__UN (NUMCOLEGIADO)
 UK60EV1HQAAK446U4QOAF76E98X (ID_USUARIO)
 FK6MWWLEAI4PNK8HXC6WBUR3TCW ()
 SYS_C008326 (ID_INFO_DOCTOR)
 SYS_C008327 (NUMCOLEGIADO)
 UK60EV1HQAAK446U4QOAF76E98X (ID_USUARIO)



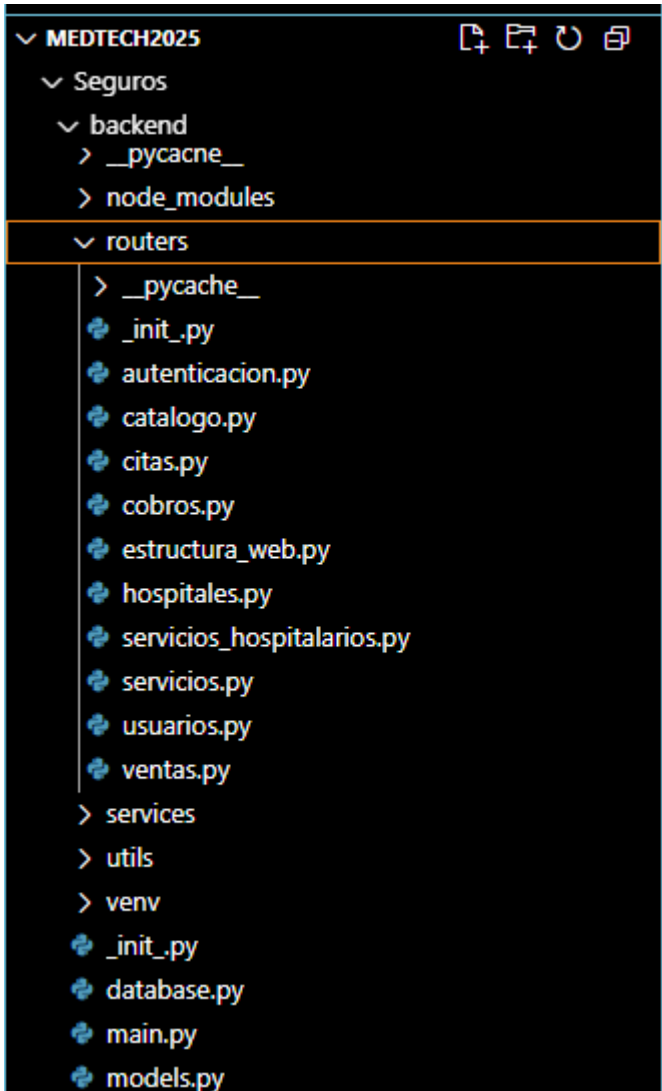
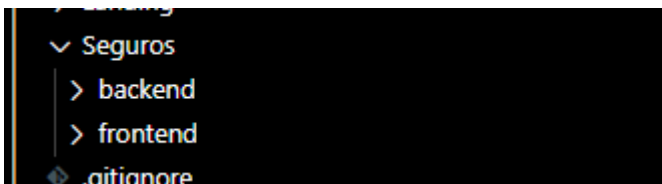
Manual Técnico Seguros

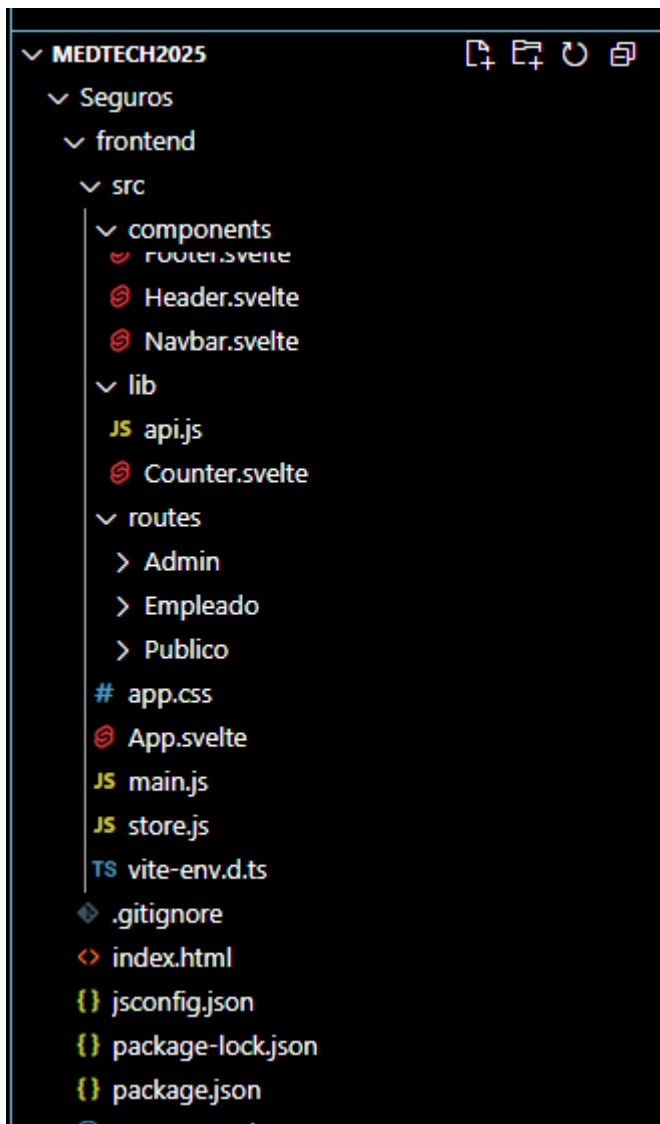
El sistema de Seguros forma parte del ecosistema ****MedTech****, el cual se compone de dos sistemas principales: Hospitales y Seguros. Este módulo está diseñado para gestionar la administración de usuarios asegurados, validar coberturas, aprobar servicios médicos y recetas, registrar ventas y cobros, y comunicarse dinámicamente con hospitales y farmacias.

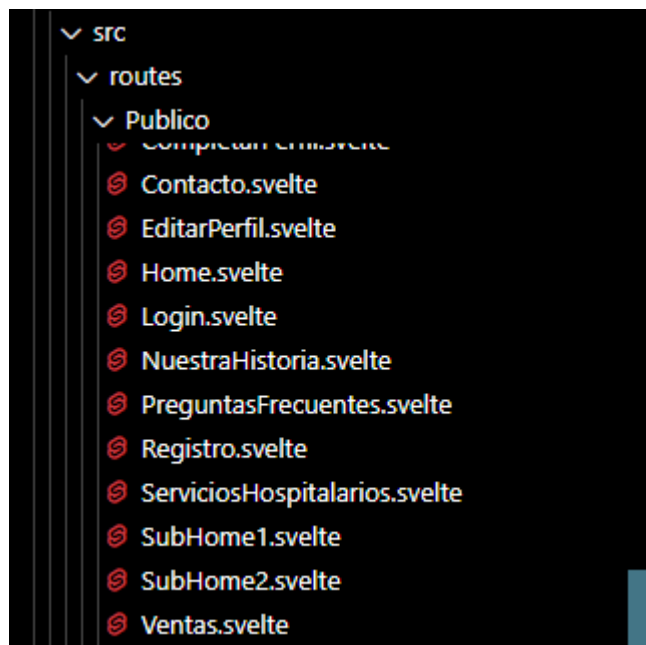
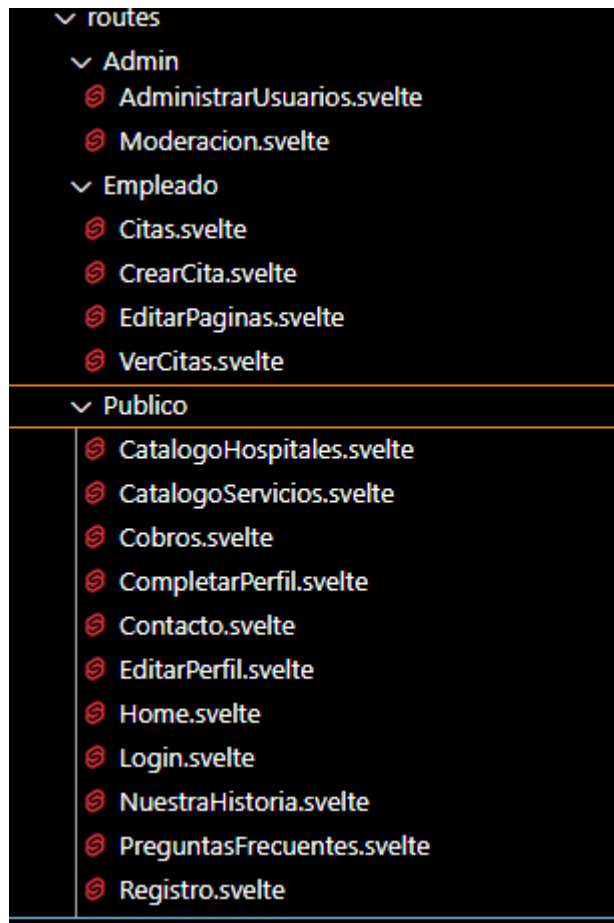
Se trata de un sistema No Relacional, desarrollado con el siguiente stack de tecnologías: Python (FastAPI) para el backend y MongoDB como base de datos. El frontend está construido con Svelte. Todo el sistema opera bajo una arquitectura de microservicios basada en REST, comunicándose exclusivamente a través de APIs y JSONs.

Arquitectura del Sistema

La estructura general de Seguros está organizada de la siguiente forma.







Los routers definen los endpoints REST

Los models definen los esquemas Pydantic de entrada/salida

database contiene la conexión a MongoDB

utils tiene las funciones auxiliares como envío de correos

services la lógica del negocio y operaciones con la base de datos

Para la instalación se debe contar con

Python 3.11 o superior

Node.js y npm

MongoDB

Para levantar el backend se utilizan los comandos

- `uvicorn main:app --reload --port 8000` para levantar el servidor python
- `mongod` para la base de datos MongoDB

y para levantar el frontend se debe ejecutar

- `npm run dev`

Modelos y Colecciones de MongoDB

Las colecciones principales que se utilizaron para Seguros fueron:

- Usuarios
- estructura_web
- estructura_drafts
- servicios_hospitalarios
- catalogo_servicios

Algunas funcionalidades clave son:

Registro de Nuevos Clientes

- * El sistema permite registrar nuevos asegurados desde un formulario de venta.
- * Se genera una contraseña automática, se hashea y se guarda en la base.
- * Se envía correo de bienvenida al usuario y correo al administrador con los datos y fecha.

Gestión de Usuarios

- * Todos los usuarios inician en estado inactivo y sin rol.
- * El administrador asigna el rol (paciente, empleado, administrador) y activa el usuario.
- * Se envía correo automático de activación.

Citas desde el módulo de seguros

- * Permite buscar disponibilidad en hospitales afiliados.
- * Permite agendar una cita en el hospital desde el seguro.
- * Si el paciente no existe en el hospital, se le crea automáticamente.

Ventas y Cobros

La página de ventas crea un nuevo usuario asegurado, permite seleccionar el tipo de póliza (70% o 90%), guarda la tarjeta de forma no enmascarada y crea un nuevo usuario, enviando un correo electrónico a este mismo con sus credenciales para el sistema.

Página de cobros

Muestra lista de todos los usuarios ordenados por fecha_vencimiento, permite renovar la póliza (registra nuevo cobro, actualiza fecha) y deshabilitar al usuario si no renueva

Seguridad y Autenticación

Autenticación por token JWT con los siguiente roles disponibles:

paciente: solo puede ver su información y póliza

empleado: gestiona ventas y cobros

administrador: puede activar usuarios, asignar roles, ver reportes

Envío de Correos Automáticos

utiliza CorreoUtils para enviar:

- Correo de bienvenida al registrarse
- Correo al administrador con datos del nuevo usuario
- Correo de activación cuando se le asigna rol
- Correo de rechazo si una receta o servicio no es aprobado

Resumen de Endpoints REST

POST/autenticacion/login: Inicia sesión y devuelve JWT

POST/autenticacion/registro: Crea nuevo usuario sin activar

POST/usuarios/activar/{id}: Activa usuario y asigna rol

GET/usuarios: Lista usuarios con filtros

POST/ventas: Registra nueva venta con tarjeta

GET/usuarios/vencimiento: Lista usuarios por vencimiento

PUT/usuarios/renovar/{id}: Renueva póliza

PUT/usuarios/desactivar/{id}. Desactiva usuario

POST/autorizaciones/solicitar: Solicita aprobación de servicio o receta

GET/autorizaciones/usuario/{id}: Ver autorizaciones del usuario

POST/citas: Agenda cita en hospital desde seguros

DELETE/citas/{id}: Cancela cita

Todas las operaciones son vía REST y JSON, no se permite acceso directo entre sistemas y toda modificación importante debe ser enviada al módulo de moderación de cambios.