iuKer

Informe de Avance: Sprint 1 - Base del Sistema Clínico iuKer

Fecha: @4 de noviembre de 2026

Enfoque del Sprint: Configuración de la Infraestructura y Base de Datos. Implementación del CRUD inicial.

1. Resumen

El objetivo principal de este sprint era sentar las bases de la arquitectura del proyecto iuKer, establecer una conexión robusta con la base de datos PostgreSQL y entregar las funcionalidades esenciales de creación, consulta, actualización y eliminación de datos para las cuatro entidades clínicas.

Meta Principal	Estado
Infraestructura (Base de datos y API's)	**Completada**
CRUD Básico (Creación, lectura, actualización y eliminación)	**Funcional**

Hemos superado los retos de configuración inicial (CRUD de las instancias y Conexión a BD) y el sistema ya permite gestionar pacientes, médicos, citas y consultorios de forma separada para cada entidad.

2. Implementación de la Arquitectura Central

Se ha implementado el diseño de **Arquitectura Hexagonal (Puertos y Adaptadores)**, lo que garantiza que la lógica de negocio esté completamente aislada de la tecnología (Fastify, PostgreSQL).

- **Dominio (Núcleo):** Se definieron las entidades (Paciente, Médico, Cita, Consultorio) y sus reglas básicas (interfaces).

- **Aplicación (Casos de Uso):** Se crearon los servicios para manejar flujos de trabajo (**Ej:** `crearPaciente()`, `agregarConsultorio()`), aplicando la regla de negocio que permiten que, por ejemplo, no se creen elementos idénticos más de una vez.
- **Infraestructura:** Se implementaron los adaptadores para la web (Controladores Fastify) y la persistencia (Repositorios PostgreSQL).

3. Entregables Funcionales (CRUD Inicial)

El siguiente conjunto de funcionalidades está operativo y listo para ser probado:

4. Gestión de Base de Datos

Entidad	Descripción Funcionalidad	Endpoints Implementados (CRUD)
Paciente	Gestión de datos de identificación y contacto del paciente. Incluye validación de unicidad por número de documento.	`GET /api/pacientes` (Listar, filtrar por ID y Límites) `POST /api/pacientes` (Crear paciente) `PUT /api/pacientes` (Modificar datos concretos) `DELETE /api/pacientes` (Borrar del sistema uno o más pacientes)
Médico	Gestión de información profesional y especialidades.	`GET /api/medicos` (Listar, filtrar por tarjetaProfesional y Límites) `POST /api/medicos` (Crear nuevos médicos) `PUT /api/medicos` (Actualizar médico por tarjetaProfesional) `DELETE /api/medicos` (Eliminar médico por tarjetaProfesional)
Consultorio	Identificación de la ubicación y disponibilidad básica.	`GET /api/consultorios` (Listar, filtrar por ID y Límites) `POST /api/consultorios` (Agregar nuevos consultorios) `PUT /api/consultorios` (Actualizar un consultorio [menos ID]) `DELETE /api/consultorios` (Eliminar un consultorio por ID)
Cita Médica	Programación simple, sin lógica compleja de disponibilidad por ahora.	`GET /api/citas-medicas` (Listar, filtrar por ID y Límites) `POST /api/citas-medicas` (agendar citas) `PUT /api/citas-medicas/reprogramar` (Reprogramar cita por ID) `PUT /api/citas-medicas/finalizar` (Finalizar cita por ID) `PUT/api/citas-medicas/cancelar` (Cancelar cita por ID) `DELETE/api/citas-medicas/eliminar` (Eliminar cita por ID)

- **Se proporcionan las consultas SQL,** que permiten crear las tablas y datos, para ejecutar y testear la funcionalidad de la aplicación.

5. Próximos Pasos (Sprint 2)

El enfoque del próximo ciclo será completar la cobertura funcional:

1. **Validaciones de CRUD:** Implementar las validaciones a las que se pueda llegar en términos de abstracción para cada entidad (`Paciente`, `Médico`, `Cita`, `Consultorio`).
2. **Implementar migración:** Esto nos permitirá que las tablas y la base de datos se cree de manera automatizada y se pueda estandarizar esta parte del proyecto.

6. Enlace video

https://youtu.be/xIsnPYPZQOc