



HealthAPI - Sistema de Citas Médicas

Descripción General

HealthAPI es un sistema backend para la gestión de citas médicas desarrollado con arquitectura hexagonal (puertos y adaptadores). Permite:

- Gestionar pacientes, médicos y citas médicas
- Registrar observaciones y seguimiento de consultas

Objetivo: Proporcionar una API funcional, escalable y mantenible para sistemas de salud.





Colaboradores del proyecto

- Jerónimo Roldán Cardona
- Santiago Herrera Martínez
- Samuel Restrepo Del Valle

Arquitectura

El proyecto sigue **Arquitectura Hexagonal (Clean Architecture)** con separación clara de responsabilidades:

Ventajas de esta arquitectura:

-  Independencia de frameworks y librerías
 -  Testeable: cada capa se puede probar aisladamente
 -  Escalable: fácil agregar nuevos casos de uso
 -  Mantenible: cambios en infraestructura no afectan la lógica de negocio
-

✂ Tecnologías

Backend

- **Runtime:** Node.js
- **Lenguaje:** TypeScript
- **Framework Web:** Fastify
- **Base de Datos:** PostgreSQL

Herramientas de Desarrollo

- **API Testing:** Bruno
 - **Administración DB:** pgAdmin 4
 - **Formato:** Prettier
 - **Gestor de Paquetes:** pnpm/npm
-

API Endpoints

Base URL

http://localhost:3000/api

Citas Médicas

Crear Cita

POST /api/citas: Este endpoint espera mediante el body los siguientes parámetros:

- idPaciente
- idDisponibilidad
- fecha
- estado ("programada", "cancelada", "realizada")
- motivo
- observaciones

Seguido de pasarle el parámetro con la información correspondiente a cada uno, se debe presentar la siguiente respuesta: **Respuesta (200) OK.**

Listar Todas las Citas

GET /api/citas: Este endpoint se ejecuta mediante BRUNO directamente, lo cual nos pasa en forma de una lista todas las citas que estén registradas en la base de datos respectivamente.

Obtener Cita por ID

GET /api/citas/id: En este endpoint es necesario pasar al final de la URL el ID que deseamos buscar, seguido a esto, ejecutar la request en bruno y recibir la información de si fue encontrado o no la cita respectivamente.

Actualizar Cita

PUT /api/citas/id: En este endpoint es necesario pasar al final de la URL el ID que deseamos actualizar, y seguido a esto, en el body pasarle el o los parámetros que deseamos actualizar en la cita, sean varios cambios o solo uno.

Eliminar Cita

DELETE /api/citas/:id: En este endpoint es necesario pasar al final de la URL el ID que deseamos eliminar, si el sistema lo encuentra, procede a eliminarlo por completo de la base de datos.

Médicos

Crear Médico

POST /api/medicos: Este endpoint espera mediante el body los siguientes parámetros:

- nombreMedico
- emailMedico
- especialidadMedico

Seguido de pasarle el parámetro con la información correspondiente a cada uno, se debe presentar la siguiente respuesta: **Respuesta (200) OK.**

Listar Médicos

GET /api/medicos: Este endpoint se ejecuta mediante BRUNO directamente, lo cual nos pasa en forma de una lista todos los médicos que estén registrados en la base de datos respectivamente.

Buscar médico por ID

GET /api/medico/id: En este endpoint es necesario pasar al final de la URL el ID que deseamos buscar, seguido a esto, ejecutar la request en bruno y recibir la información de si fue encontrado o no el médico respectivamente.

Actualizar médico

PUT /api/medico/id: En este endpoint es necesario pasar al final de la URL el ID que deseamos actualizar, y seguido a esto, en el body pasarle el o los parámetros que deseamos actualizar en la cita, sean varios cambios o solo uno.

Eliminar médico

DELETE /api/medico/id: En este endpoint es necesario pasar al final de la URL el ID que deseamos eliminar, si el sistema lo encuentra, procede a eliminarlo por completo de la base de datos.

Pacientes

Crear Paciente

POST /api/pacientes: Este endpoint espera mediante el body los siguientes parámetros:

- nombre
- email
- telefono

Seguido de pasarle el parámetro con la información correspondiente a cada uno, se debe presentar la siguiente respuesta: **Respuesta (200) OK.**

Listar pacientes

GET /api/pacientes: Este endpoint se ejecuta mediante BRUNO directamente, lo cual nos pasa en forma de una lista todos los pacientes que estén registrados en la base de datos respectivamente.

Buscar pacientes por ID

GET /api/paciente/id: En este endpoint es necesario pasar al final de la URL el ID que deseamos buscar, seguido a esto, ejecutar la request en bruno y recibir la información de si fue encontrado o no el paciente respectivamente.

Actualizar paciente

PUT /api/paciente/id: En este endpoint es necesario pasar al final de la URL el ID que deseamos actualizar, y seguido a esto, en el body pasarle el o los parámetros que deseamos actualizar en el paciente, sean varios cambios o solo uno.

Eliminar paciente

DELETE /api/paciente/id: En este endpoint es necesario pasar al final de la URL el ID que deseamos eliminar, si el sistema lo encuentra, procede a eliminarlo por completo de la base de datos.

Testing con Bruno

La carpeta /bruno contiene todas las peticiones pre-configuradas:

1. Abrir Bruno
2. File → Open Collection
3. Seleccionar la carpeta /bruno dentro del proyecto HealthAPI
4. Navegar por las request ya preestablecidas para mayor facilidad de test