Documentación Proyecto: Turnero Médico Digital

# 1. Introducción

El presente proyecto consiste en el desarrollo de un sistema de gestión de turnos médicos, que permite registrar pacientes y médicos, así como asignar y administrar turnos. La solución se construyó utilizando Node.js + Express como servidor backend, PostgreSQL como gestor de base de datos, Postman para la validación de endpoints y pruebas de los servicios, y HTML + JavaScript (Visual Studio Code) como parte del entorno de desarrollo. El objetivo del sistema es facilitar la administración de turnos médicos, simplificando la relación entre pacientes y profesionales de la salud.

# 2. Objetivos

## Objetivo General

Desarrollar un sistema web de turnos médicos que permita gestionar pacientes, médicos y turnos, aplicando un modelo relacional de base de datos y validando la comunicación mediante servicios REST.

## Objetivos Específicos

- Diseñar una base de datos relacional en PostgreSQL con 3 tablas: pacientes, medicos, turnos.  
- Implementar un servidor Node.js que exponga endpoints RESTful (GET, POST, PUT, DELETE).  
- Validar la correcta comunicación y persistencia de datos con Postman.  
- Documentar el proceso de desarrollo, modelo relacional y pruebas realizadas.

# 3. Modelo Relacional

El modelo relacional del sistema se diseñó considerando las entidades principales involucradas en la gestión de turnos médicos: Pacientes, Médicos y Turnos. Cada entidad contiene atributos específicos y se relaciona con las demás a través de claves primarias (Primary Keys) y claves foráneas (Foreign Keys).

## Tablas del sistema:

1. PACIENTES  
 - id\_paciente (PK, serial)  
 - nombre (varchar)  
 - apellido (varchar)  
 - dni (varchar, unique)  
 - telefono (varchar)  
  
2. MEDICOS  
 - id\_medico (PK, serial)  
 - nombre (varchar)  
 - apellido (varchar)  
 - especialidad (varchar)  
  
3. TURNOS  
 - id\_turno (PK, serial)  
 - id\_paciente (FK → pacientes.id\_paciente)  
 - id\_medico (FK → medicos.id\_medico)  
 - fecha (date)  
 - hora (time)  
 - estado (varchar: pendiente, confirmado, cancelado)

## Relaciones entre tablas:

- Un paciente puede tener muchos turnos (relación 1 a muchos).  
- Un médico puede tener muchos turnos (relación 1 a muchos).  
- Cada turno pertenece a un único paciente y a un único médico.  
- De manera indirecta, se establece una relación muchos a muchos entre pacientes y médicos, ya que un paciente puede atenderse con diferentes médicos y un médico puede atender a distintos pacientes, relación que se materializa a través de la tabla TURNOS.

# 4. Conclusiones

Este proyecto demuestra una aplicación práctica de los conceptos de programación, bases de datos y servicios web, y puede ser expandido fácilmente con una interfaz gráfica para el uso real en clínicas u hospitales.