

# Trabajo Práctico 3

## Desarrollo de Aplicación de Turnos Médicos

Empresa: **NovaTech**

Proyecto solicitado por el cliente: **Clínica Salud+**

Tecnologías base: **Node.js, Express, Pug, EJS, JWT**

---

La empresa NovaTech, especializada en soluciones tecnológicas para el sector salud, ha sido contratada por la Clínica Salud+ para desarrollar una aplicación web que permita la gestión de turnos médicos.

El proyecto ya cuenta con una base de código fuente desarrollada con Node.js y se encuentra disponible en el siguiente repositorio de GitHub.



<https://github.com/santozzi/practico3progra3>

,env

```
PORT=3000
```

```
HOST=http://localhost
```

```
TEMPLATE=ejs
```

Trabjarán sobre ese código para completar y mejorar la funcionalidad, según los siguientes requerimientos del cliente.

---

El **objetivo** del trabajo es implementar una solución para la gestión de turnos médicos en dos contextos distintos, según el requerimiento de la clínica:

### Versión online (turnos vía web – backend con Express)

En esta versión se trabajará únicamente con el backend, utilizando Express para construir una **API REST**. Esta API permitirá consultar y gestionar turnos de pacientes, devolviendo la información en formato JSON. No se utilizan vistas renderizadas, ya que el frontend se desarrollará en etapas posteriores.

### Funcionalidades obligatorias:

Consultar turnos por identificador

- Endpoint: **GET /api/v1/turnos/:idPaciente**
- Permite consultar los turnos registrados para un paciente.
- La respuesta debe ser un JSON con los turnos asociados a ese identificador.

## Cancelar un turno

- Endpoint: **DELETE /api/v1/turnos/:idTurno**
- Permite cancelar un turno específico a partir de su identificador.
- La respuesta debe confirmar la cancelación o indicar si ocurrió un error.

### Funcionalidades opcionales (para sumar complejidad):

#### Cargar un nuevo turno

- Endpoint: **POST /api/v1/turnos**
- Permite registrar un nuevo turno, enviando la fecha y hora deseadas.
- No es necesario elegir médico, ya que hay uno solo disponible.
- Esta acción debe requerir autenticación mediante JWT.

#### Sistema de autenticación JWT

Para las acciones de cargar o editar turnos, es necesario implementar un sistema de login que genere un token JWT al autenticarse. Este token se enviará en el encabezado de las solicitudes (Authorization) para acceder a los endpoints protegidos.

## Versión local (turnos en la clínica – vistas con Pug o EJS)

Esta versión sí incluye vistas renderizadas y debe permitir al personal de la clínica:

- Visualizar los turnos disponibles o reservados.
- Registrar nuevos turnos para pacientes.
- Cancelar turnos y dar de baja pacientes desde una interfaz amigable.

---

## Criterios de evaluación

- Correcta separación entre las vistas para turnos online y locales.
- Ruteo claro y estructurado.
- Funcionalidad completa del sistema de turnos.
- Estilo visual adecuado (aunque no se evaluará en profundidad el diseño).
- Uso correcto de plantillas (pug/ejs) y envío de datos desde el backend.
- Código subido en GitHub y documentación mínima en el README
  - Endpoints para acceder a las diferentes rutas
  - Cómo se instala el proyecto
  - Integrantes del grupo
  - Capturas de pantalla
  - Y lo que quieran agregar.
- Entrega a tiempo del práctico que será el 8/6 23:59 como máximo.