

# API REST APP TURNOS

Rios Julian

## REGISTRO

### Enlazar Cuenta de Paciente Existente

- **Método:** POST
  - **Ruta:** /api/pacientes/paciente\_existente
  - **Tipo de envío:** application/json
  - **Parámetros (Body):**
    - dni: string (Para *encontrar* al paciente existente)
    - mail: string (El email que usará para *iniciar sesión*)
    - password: string (La contraseña que usará, en texto plano)
  - **Respuesta:** Objeto JSON { message: "Cuenta creada y enlazada" }
  - **Descripción:** Busca un paciente por DNI, crea un registro de usuarios para él y enlaza las dos tablas.
- 

## AUTENTICACION

### Login

- **Método:** POST
- **Ruta:** /api/auth/login
- **Tipo de envío:** application/x-www-form-urlencoded
- **Parámetros:**
  - Usuario: string (ej. carlos@mail.com)
  - Clave: string (ej. mi\_password\_123)
- **Respuesta:** Objeto JSON { token: "string" } (Un token JWT)
- **Descripción:** Inicia sesión del usuario (paciente, admin, etc.) y devuelve un token de autorización.

---

## **PACIENTES**

### **Registro de Nuevo Paciente**

- **Método:** POST
- **Ruta:** /api/pacientes/registro
- **Tipo de envío:** application/json
- **Parámetros (Body):**
  - Un objeto JSON con los datos del paciente:

JSON

```
{  
  "nombre": "string",  
  "apellido": "string",  
  "telefono": "int",  
  "mail": "string (ej. paciente.nuevo@mail.com)",  
  "password": "string (ej. fuerte123)",  
  "dni": "int",  
  "idObraSocial": "int"  
}
```

- **Respuesta (Éxito):** Status 201 Created - Objeto JSON

JSON

```
{  
  "message": "Paciente registrado exitosamente",  
  "usuariold": ----  
}
```

- **Descripción:** Registra un paciente 100% nuevo que no existe en la base de datos. Crea el registro en la tabla usuarios (con contraseña encriptada) y en la tabla paciente simultáneamente, enlazándolos.
-

## **PACIENTES**

### **Actualizar Perfil (protegido)**

- **Método:** PATCH
- **Ruta:** /api/pacientes/mi-perfil
- **Autorización (requerida):**
  - Ttipo:** Bearer Token
    - **Cabecera (header):** Authorization: Bearer <token\_jwt>
- **Tipo de envío:** application/json
- **Parámetros (body):**
  - un objeto json con los campos que se desean cambiar (todos son opcionales):

JSON

```
{  
  "telefono": 1122334455,  
  "idObraSocial": 2,  
  "contraseña": "miNuevaClaveSegura123"  
}
```

- **respuesta (exito):** Status 200 OK - objeto json

JSON

```
{  
  "message": "perfil actualizado exitosamente"  
}
```

- **Descripción:** actualiza los datos del perfil del paciente. actualiza el teléfono en ambas tablas (usuarios y paciente) y la obra social solo en paciente. si se envía una contraseña, la encripta y la guarda.

---

### **Actualizar mi foto de perfil (protegido)**

- **Método:** POST
- **Ruta:** /api/pacientes/mi-perfil/imagen

- **Autorización (requerida):**
  - **tipo:** Bearer Token
  - **cabecera (header):** Authorization: Bearer <token\_jwt>
- **Tipo de envío:** multipart/form-data
- **Parámetros (form-data):**
  - una llave unica llamada imgperfil que contenga el archivo de imagen.
- **Respuesta (éxito):** Status 200 OK - objeto json

JSON

```
{
  "message": "imagen de perfil actualizada exitosamente",
  "ruta": "uploads/10-1762428585250-424242-foto.jpg"
}
```

- **Descripción:** sube una nueva imagen de perfil, la guarda en el servidor y actualiza la ruta en la columna imgperfil de la tabla usuarios.
- 

## TURNOS

### Obtener Mis Turnos (HISTORIAL)

- **Método:** GET
- **Ruta:** /api/turnos/mis-turnos
- **Tipo de envío:** N/A
- **Autorización (Requerida):**
  - **Tipo:** Bearer Token
  - **Cabecera (Header):** Authorization: Bearer <token\_jwt>
- **Respuesta:** Un array de objetos JSON, donde cada objeto es un turno.

JSON

```
[
  {
    "turno_id": 1,
```

```

    "fecha": "2025-11-10",
    "horario": "10:30:00",
    "tipoconsulta": "NORMAL",
    "motivo_consulta": "Control anual",
    "estado": "confirmado",
    "medico_nombre": "Carlos",
    "medico_apellido": "Rivera",
    "especialidad_nombre": "Clínica Médica"
  },
  {
    "turno_id": 5,
    "fecha": "2025-10-20",
    "horario": "15:00:00",
    "tipoconsulta": "ESPECIAL",
    "motivo_consulta": "Dolor de rodilla",
    "estado": "atendido",
    "medico_nombre": "Ana",
    "medico_apellido": "Salas",
    "especialidad_nombre": "Traumatología"
  }
]

```

- **Descripción:** Devuelve una lista de todos los turnos (futuros y pasados) del paciente que está logueado. Requiere un token JWT válido.

---

## **TURNOS**

### **Reservar un turno**

- **Método:** POST
- **Ruta:** /api/turnos/reservar

- **Autorización (requerida):**
  - **Tipo:** Bearer Token
  - **Cabecera (header):** Authorization: Bearer <token\_jwt>
- **Tipo de envío:** application/json
- **Parámetros (body):**
  - un objeto json con los datos del turno a reservar:

JSON

```
{
  "profesional_id": 1,
  "fecha": "2025-12-20",
  "horario": "10:30:00",
  "tipoconsulta": "NORMAL",
  "motivo_consulta": "chequeo general"
}
```

- **Respuesta (exito):** Status 200 OK - objeto json

JSON

```
{
  "message": "turno reservado exitosamente."
}
```

- **Descripción:** busca un turno 'libre' que coincida con el médico, fecha y hora, y lo actualiza, asignándoselo al paciente logueado.
- 

## **TURNOS**

### **Obtener turnos disponibles**

- **Método:** GET
- **Ruta:** /api/turnos/disponibles/:idmedico
- **Parámetros (query string):** fecha (obligatorio).
  - *ejemplo de como se usa:* /api/turnos/disponibles/1?fecha=2025-12-20

- **Autorización:** ninguna (es publica).
- **Respuesta (éxito):** un array de objetos (turnos libres)

JSON

```
[
  { "turno_id": 15, "horario": "09:00:00" },
  { "turno_id": 16, "horario": "09:30:00" },
  { "turno_id": 17, "horario": "10:00:00" }
]
```

- **Descripción:** devuelve una lista de los horarios (y sus turno\_id) que están con estado 'libre' para un médico específico en una fecha específica.

## **TURNOS**

### **Cancelar turno (protegido)**

- **Método:** PUT
- **Ruta:** /api/turnos/cancelar/:idturno
  - *ejemplo de como se usa:* /api/turnos/cancelar/23
- **Autorización (requerida):**
  - **Tipo:** Bearer Token
  - **Cabecera (header):** Authorization: Bearer <token\_jwt>
- **Tipo de envío:** N/A (no se envia body)
- **Respuesta (éxito):** Status 200 OK - objeto json

JSON

```
{
  "message": "turno cancelado exitosamente"
}
```

- **Descripción:** cambia el estado de un turno a 'cancelado'. solo permite cancelar un turno si le pertenece al paciente logueado y su estado actual es 'reservado' o 'confirmado'. esto preserva el turno en el historial del paciente, pero libera el horario para que otro paciente pueda reservarlo.