

/api/atencion/registrar

- Método: POST.
- Ruta: /api/atencion/registrar.
- Autenticación: requerida. Router aplica `authMiddleware`; controlador exige rol `medico`.
- Controlador: `AtencionController.registrarAtencion` → Servicio: `UrgenciaService.registrarAtencion`.

Request

- Headers: `Authorization: Bearer <jwt>`, `Content-Type: application/json`.
- Body:
 - `ingresoCuil`: `string` obligatorio, CUIL del ingreso asignado al médico.
 - `informe`: `string` opcional con detalle de la atención.
 - `nombre, apellido`: `strings` opcionales del médico; defaults derivados del email o `"medico"/"Cuenta"`.
 - `cuil`: `string` opcional del médico; default `"20123456789"`, validado como VO Cuil.
 - `matricula`: `string` opcional; default `"MAT-DEFAULT"`.

Validaciones y reglas

- 403 si el usuario autenticado no es `medico`.
- 400 si falta `ingresoCuil`.
- El servicio verifica:
 - Que el ingreso exista.
 - Que tenga doctor asignado y que coincida con el del JWT.
 - Que el ingreso esté en estado `EN_PROCESO`.
 - Que no esté ya finalizado.
- Si pasa las validaciones, cambia estado a `FINALIZADO` y crea `Atencion`.

Flujo interno

1. Construye `Doctor` con datos del JWT/body.
2. Llama a `registrarAtencion` con CUIL del ingreso y el informe.
3. Persiste la atención en memoria y actualiza estado del ingreso.

Respuestas

- 201 { `"message": "Atención registrada exitosamente"`, `"atencion": { ... } }`.
- 400 por ingreso inexistente, sin médico asignado, asignado a otro médico, no en proceso o ya finalizado.
- 401 si el token falta/es inválido; 403 si el rol no es médico.

Consideraciones

- Completa el flujo iniciado en `/api/reclamo/reclamar`; debe llamarse después de reclamar.
- Persistencia en memoria; se pierde al reiniciar.