

/api/urgencias/regar

- Método: POST.
- Ruta: /api/urgencias/regar.
- Autenticación: no aplicada en router; se puede invocar sin token.
- Controlador: UrgenciasController.registrarUrgencia → Servicio: UrgenciaService.registrarUrgencia.

Request

- Headers: Content-Type: application/json.
- Body:
 - cuil: string, CUIL del paciente existente (10/11 dígitos; se toleran guiones y se limpian).
 - enfermera: objeto con:
 - cuil: string, se limpia y valida como VO Cuil (default "27123456789").
 - nombre: string.
 - apellido: string.
 - email: string opcional; si falta, default <nombre>.<apellido>@example.com validado por VO Email.
 - matricula: string opcional; default "ENF00000".
 - informe: string con motivo/observaciones.
 - nivelEmergencia: string exacto en ["Critica", "Emergencia", "Urgencia", "Urgencia Menor", "Sin Urgencia"]; se mapea a instancia NivelEmergencia.
 - temperatura: number entre 30 y 45.
 - frecuenciaCardiaca: number entre 40 y 200.
 - frecuenciaRespiratoria: number entre 8 y 60.
 - frecuenciaSistolica: number entre 60 y 180.
 - frecuenciaDiastolica: number entre 40 y 120.

Validaciones y reglas

- VOs de signos vitales validan rangos y lanzan error si están fuera de rango.
- VO Cuil del paciente: si el paciente no existe en el repositorio, se lanza error.
- nivelEmergencia debe coincidir con la descripción exacta; de lo contrario se lanza error.
- No hay control de autenticación ni rol en este router.

Flujo interno

1. Normaliza datos de enfermera y construye VOs (Cuil, Email).
2. Busca paciente por CUIL en RepoPacientes.
3. Crea Ingreso con datos clínicos y nivel.
4. Inserta en listaEspera y reordena por prioridad (nivel más crítico primero) y timestamp.

Respuestas

- 201 { "message": "Urgencia registrada exitosamente" }.
- 400 si signos fuera de rango, nivel inválido, CUIL/Email inválido o paciente inexistente.

Consideraciones

- La cola se mantiene en memoria; se pierde al reiniciar.
- No se verifica token; si se requiere control de rol/enfermera debería agregarse middleware.