

***Entregable 10 – Analista.***

Casas Vázquez Luis Américo

22140750

*Ingeniería en Sistemas Computacionales, Instituto Tecnológico de México campus  
Querétaro.*

Ingeniería de Software.

6A

SCC1007

Villeda Maldonado Julio Alejandro

24 de febrero de 2026

## Diagramas de Secuencia (Análisis)

**Caso 1:** Registrar paciente

**Actores:** Enfermero y Médico.

**Objetos:**

- **UIRegistroPaciente (Interfaz):** Captura datos del paciente (nombre, fecha de nacimiento, sexo, CURP, domicilio, estado civil, correo, ocupación, contacto de emergencia, tipo de sangre, alergias, enfermedades crónicas), muestra formulario y mensajes de éxito o error.  
**Justificación:** Se ubica en la capa de presentación y no contiene lógica de negocio, respetando la separación de responsabilidades.
- **ControladorAltaPaciente (Controlador):** Recibe solicitud de la UI, invoca validación/registro y devuelve la respuesta a la UI.  
**Justificación:** Su función es orquestrar la secuencia sin implementar reglas del dominio.
- **ServicioAltaPaciente (Servicio):** Aplica reglas del registro (validación de campos obligatorios) y decide si procede el alta.  
**Justificación:** Centraliza la lógica del dominio evitando que la interfaz o el controlador la implementen.
- **RepositorioPaciente (Repositorio):** Consulta si existe paciente (por CURP/ID) y guarda el nuevo registro cuando procede.  
**Justificación:** Se encarga exclusivamente de la persistencia, sin incluir lógica de validación.

### Identificación de fragmentos combinados

**alt:**

En el Paso 4 (D1): El sistema valida la información ingresada.

- **Sí** → Continúa al paso 5.
- **No (Datos incompletos o inválidos)** → flujo alterno A1.
- **No (ID o CURP ya existentes)** → flujo alterno A2.

Se usa alt y no opt porque no es una acción opcional, sino una decisión con múltiples posibles resultados.

**loop:**

Desde el Paso 3 (captura de datos) cuando ocurre A1 o A2, porque ambos alternos regresan al Paso 3.

Se utiliza loop porque el proceso puede repetirse hasta que se cumplan las condiciones de validación.

**Caso 2:** Dar de baja paciente

**Actor:** Médico.

**Objetos:**

- **UIConsultaBajaPaciente (UI):** Solicita criterio de búsqueda (ID, CURP, nombre), muestra resultados, muestra confirmación y resultado final.  
**Justificación:** Pertenece a la capa de presentación y no ejecuta reglas de negocio.
- **ControladorBajaPaciente (Controlador):** Coordina búsqueda, selección del paciente y solicitud de baja; maneja confirmación desde la UI.  
**Justificación:** Su función es orquestar la secuencia sin implementar reglas del dominio.
- **ServicioBajaPaciente (Servicio):** Valida la existencia del paciente y autoriza la operación.  
**Justificación:** Centraliza la lógica del dominio evitando que la interfaz o el controlador la implementen.
- **RepositorioPaciente (Repositorio):** Ejecuta consulta de pacientes y actualiza el estado a “Baja”.  
**Justificación:** Se encarga exclusivamente de la persistencia, sin incluir lógica de validación.

### Identificación de fragmentos combinados

**alt:**

- **En el Paso 4 (D1):** El sistema muestra los resultados encontrados.
  - **Sí** → Paso 5.
  - **No** → Flujo alterno A1.
- **En el Paso 8 (D2):** El médico presiona “confirmar”.
  - **Sí** → Paso 9.
  - **No** → Flujo alterno B1. (operación cancelada)

Se usa alt y no opt porque no son acciones opcionales, sino decisiones con dos posibles resultados.

**loop:**

Desde el paso 2, si se toma el flujo alterno A1.

Se utiliza loop porque el proceso puede repetirse mientras no se encuentren resultados.