



# Tecnológico Nacional de México Campus Querétaro

Entregable 10 - Unidad 1

## "Revisión de diagramas de secuencia"

Que presenta

**Carrillo Tejas Luis Antonio**

Estudiante de la carrera:

**Ingeniería en Sistemas Computacionales**

Materia:

**Ingeniería de Software**

A cargo del docente:

**Julio Alejandro Villeda Maldonado**

Periodo:

**ENERO-JUNIO 2026**



## Responsable de calidad y documentación

Revisión de los diagramas de secuencia y documentación relacionada del sistema "DIGICLIN".

### Revisión

Criterio	Caso 1	Caso 2
Arquitectura respetada	Sí	Sí
Actor correcto	Sí	Sí
Objetos coherentes	No	Sí
Orden temporal coherente	Sí	Sí
Uso correcto de alt	Sí	Sí
Uso correcto de opt	Sí	Sí
Uso correcto de loop	Sí	No
Coherente con el E9	Sí	Sí
Sin nuevas funcionalidades	Sí	Sí

### Observaciones

En el caso 1 existe un flujo llamado `existePaciente()` que va del servicio al repositorio. Este flujo está de más pues anteriormente se había usado el flujo `validarDatos()` que, en teoría, debería verificar la unicidad del registro dentro de sus procesos. Recomiendo eliminar ese flujo.

Hay un detalle técnico en el caso 1: en caso de que los datos no sean válidos, el error no debería llegar hasta la UI. En lugar de eso, recomiendo manejar el uso de excepciones para marcar códigos de error.

En el caso 2, el loop principal está bien usado para el flujo que representa, sin embargo tiene un error lógico. El loop se detiene solamente cuando hay resultados, pero este debería seguir hasta que el actor elija un paciente. De lo contrario, el ciclo terminaría con que haya resultados, incluso si los arrojados no fueron los que el actor necesitaba.

En el caso 2, en el alt de "No se encuentran resultados", parece que el controlador está implementando reglas de negocio al definir un resultado vacío como un error. La forma más correcta de implementar esto sería haciendo que el controlador interprete los resultados de la búsqueda y responda en base a eso sin llegar a considerar los resultados un error.



En el caso 2 hace falta una validación antes de dar de baja a un paciente para revisar si el paciente no fue dado de baja con anterioridad.

En el caso 2, cuando el actor cancela la baja de un paciente solo deberían ocurrir cambios en la UI y no llamar al controlador para procesar la decisión.