



**Carrera:** Licenciatura en Ciencia de Datos

**Cátedra:** Ingeniería de Software II

**Título:** Trabajo Práctico 2: Diagrama de secuencia

**Docentes:** Valotto, Victor; Godoy, Cielo

**Alumnos:** Carrozzo, Felipe; Ré, Lautaro Caupolicán

**Año:** 2025

## Desarrollo

### 1. Repaso teórico

#### ¿Qué información brinda un diagrama de secuencia?

Un diagrama de secuencia muestra la dinámica de interacción entre objetos en un contexto temporal, mostrando cómo colaboran para llevar a cabo una funcionalidad específica.

#### ¿Qué elementos lo componen y qué significa cada uno de ellos?

- **Instancias de clases/actores:** ubicadas en la parte superior.
- **Líneas de vida:** líneas verticales que representan la existencia de cada objeto a lo largo del tiempo.
- **Mensajes:** flechas entre líneas de vida que indican la comunicación (llamadas a métodos, devoluciones de valores, señales).
- **Tiempo:** fluye de arriba hacia abajo, indicando el orden en que ocurren las interacciones.

#### ¿Qué relación existe entre los diagramas de secuencia y los casos de uso?

Los casos de uso definen qué hace el sistema desde la perspectiva del usuario, mientras que los diagramas de secuencia muestran cómo se logra esa funcionalidad, detallando las interacciones entre actores y objetos en un flujo temporal.

#### ¿Se debe realizar un diagrama de secuencia por cada caso de uso?

No es obligatorio, pero lo recomendable es hacer al menos un diagrama de secuencia para los casos de uso más relevantes, aquellos donde sea importante entender el detalle de las interacciones. Un mismo caso de uso puede tener varios diagramas de secuencia: uno para el flujo principal y otros para los flujos alternativos o excepciones. En proyectos pequeños se puede hacer solo para los escenarios clave; en proyectos grandes, se suele priorizar para los casos de negocio más importantes.

#### ¿Qué diferencia hay entre un diagrama de secuencia de instancia y uno genérico?

- **Diagrama de secuencia de instancia:** muestra un escenario concreto de ejecución, con objetos específicos y un flujo de mensajes definido.
- **Diagrama de secuencia genérico:** representa la interacción de manera abstracta o generalizada, sin centrarse en instancias concretas, sino en roles o clases.

Entonces, el genérico modela la regla, el de instancia modela un ejemplo particular.

#### ¿De qué manera se relacionan los diagramas de secuencia y la arquitectura?

Los diagramas de secuencia se relacionan con la arquitectura porque permiten visualizar cómo los casos de uso se implementan a través de los distintos componentes del sistema. Ayudan a identificar la distribución de responsabilidades entre capas (presentación, lógica de negocio, datos) y facilitan la detección de necesidades arquitectónicas, como la integración con servicios externos, colas de mensajería, bases de datos o microservicios.

## 2. Selección de casos de uso

- Elegir **dos casos de uso relevantes** de su proyecto (los más críticos o los que involucren mayor interacción entre componentes).

Completar una tarea: El usuario cumple el ciclo de la tarea. Se espera una recompensa y algunas analíticas del rendimiento.

Registrar un nuevo usuario: EL usuario completa el formulario de datos para el registro. El sistema valida los datos ingresados, verifica que el usuario no exista previamente, guarda la información de manera segura, clasifica el perfil según los datos proporcionados y envía un correo o notificación de verificación para activar la cuenta.

## 3. Diagramación

- Generar los [diagramas de secuencia](#) para cada escenario principal de los dos casos de uso seleccionados.
- Identificar qué objetos (clases instanciadas) participan y cómo colaboran.
- Reflejar en los diagramas las decisiones de diseño tomadas en la arquitectura (ejemplo: capas, servicios, repositorios, controladores).

## 4. Análisis final

- Explicar en un breve texto cómo los diagramas de secuencia obtenidos ayudan a entender la distribución de responsabilidades y cómo se relacionan con la arquitectura de su proyecto.
- Los diagramas de secuencia nos permiten visualizar cómo los distintos objetos del sistema interactúan para cumplir con un caso de uso específico, mostrando el flujo de mensajes y la secuencia de acciones. Esto ayuda a *comprender la distribución de responsabilidades*, ya que queda claro qué módulo se encarga de la validación de datos, qué objeto gestiona la persistencia, quién clasifica el perfil del usuario y quién se ocupa de las notificaciones. Además, los diagramas evidencian cómo los distintos módulos de la arquitectura se comunican entre sí, mostrando de manera concreta cómo los componentes de la arquitectura colaboran para cumplir la funcionalidad del sistema. De esta manera, facilitan tanto la *planificación del desarrollo* como la *comprensión global de la estructura del proyecto*.