

9. Lectura Diagramas de Secuencia con UML

Los **diagramas de secuencia** son un tipo de diagrama de interacción en **UML (Unified Modeling Language)** utilizados para **modelar la comunicación entre objetos en un sistema**. Representar el **flujo de mensajes** entre los objetos a lo largo del tiempo, describiendo cómo interactúan en **un escenario** específico.

- **Ordenación temporal:** Se representan de manera **vertical**, con los eventos sucediendo de arriba hacia abajo.
- **Interacción entre objetos:** Se ubican en el eje **horizontal**, mostrando los objetos involucrados en el proceso.

Tipos de Diagramas de Interacción en UML

Existen dos tipos principales de diagramas de interacción:

1. **Diagrama de Secuencia:**
 - Enfatiza la ordenación **temporal** de los eventos.
 - Representa cómo los **objetos se comunican** a lo largo del tiempo mediante mensajes.
 - Se utiliza para modelar **métodos específicos** de una clase.
 - Se modela para cada caso de uso.
2. **Diagrama de Colaboración (o de Comunicación):**
 - Enfatiza la **estructura** organizativa entre los objetos.
 - Es una colección de **nodos y arcos**.
 - Se usa cuando la disposición **estructural** de los objetos es más importante que la secuencia de mensajes.

Componentes Claves de un Diagrama de Secuencia

Objetos e Interacciones

- Cada objeto dentro del diagrama se representa con una **línea de vida**, la cual es una línea vertical discontinua que muestra la existencia del objeto a lo largo del tiempo.
- Los objetos intercambian **mensajes**, representados con flechas dirigidas. Dependiendo del tipo de mensaje, las flechas pueden ser:
 - **Síncronas:** Bloquean el flujo de ejecución hasta que el mensaje es procesado.
 - **Asíncronas:** Permiten que el emisor continúe su ejecución sin esperar respuesta.

Elementos Gráficos del Diagrama

- **Línea de Vida:** Representa la duración de existencia de un objeto en la secuencia.
- **Foco de Control:** Representado con un rectángulo delgado sobre la línea de vida, indica cuándo un objeto está ejecutando una acción.
- **Mensajes:** Representan la comunicación entre objetos. Se representan con diferentes tipos de flechas.

Creación y Destrucción de Objetos

- **Creación:** Se indica con el estereotipo `<<create>>` cuando un objeto es instanciado.
- **Destrucción:** Se representa con `<<destroy>>` , indicando el final de la vida de un objeto en el diagrama.

Tipos de Mensajes en los Diagramas de Secuencia

Mensajes Síncronos

- Representados con una flecha con **cabeza llena** (\rightarrow).
- El objeto que envía el mensaje **se bloquea** hasta recibir una respuesta.
- Son equivalentes a **llamadas de métodos** en la programación orientada a objetos.

Mensajes Asíncronos

- Representados con una flecha con **cabeza abierta** (\rightarrow).
- No bloquean al remitente; pueden generar múltiples hilos de ejecución.

Mensajes de Retorno

- Representados con una **línea discontinua** (\leftarrow).
- Indican la respuesta a un mensaje previamente enviado.

Usos de los Diagramas de Secuencia

Los diagramas de secuencia pueden representarse de dos formas:

1. **De Instancia:**
 - Modelan un caso específico de ejecución del sistema.
 - Representan una interacción concreta entre objetos.
2. **Genéricos:**
 - Describen la interacción **para un caso de uso**.
 - Incluyen estructuras de control como **ramificaciones (branches)**, **condiciones y bucles**.

Fragmentos Combinados y Operadores en Diagramas de Secuencia

Los **fragmentos combinados** son bloques de control que agrupan interacciones dentro de un marco. Se utilizan para representar **lógica condicional y de iteración**.

Tipos de Fragmentos Combinados

- **Alternativa (alt):**
 - Representa una estructura `if-then-else` , donde solo una de las interacciones se ejecuta según una condición.
- **Opción (opt):**
 - Similar a `alt` , pero con un solo bloque. Se ejecuta solo si se cumple la condición.
- **Bucle (loop):**
 - Representa una iteración (`for` , `while`). Se ejecuta varias veces según una condición.
- **Referencia (ref):**
 - Indica una referencia a otro diagrama de secuencia.

- **Paralelo** (`par`):
 - Representa interacciones que pueden ejecutarse en paralelo.
- **Región Crítica** (`critical`):
 - Define una sección del diagrama donde solo un proceso puede ejecutarse a la vez.

Ejemplo de un Diagrama de Secuencia

Un ejemplo clásico de diagrama de secuencia es la interacción entre un **usuario** y un **sistema de autenticación**:

1. El usuario ingresa sus credenciales (`login()`).
2. El sistema verifica la información (`verificar_credenciales()`).
3. Si las credenciales son correctas, se concede acceso (`acceso_autorizado()`).
4. Si no, se muestra un mensaje de error (`mostrar_error()`).

Este diagrama muestra cómo los objetos **Usuario** y **Sistema** interactúan en una secuencia temporal definida.

Dudas

- ¿Para qué se usan los fragmentos combinados en un diagrama de secuencia?
- ¿Cómo se puede representar una interacción que ocurre dentro de otro diagrama de secuencia?
- ¿Qué diferencia hay entre un diagrama de secuencia de instancia y uno genérico?