UNIVERSIDAD DE ORIENTE.

NÚCLEO DE ANZOÁTEGUI.

ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS.

DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS.

CÁTEDRA: DESARROLLO DE SOFTWARE (072-4712)



**Diagramas de UML:  
Diagrama de estructura compuesta  
Diagrama de casos de uso  
Diagrama de secuencia**

Autores:

|  |  |  |
| --- | --- | --- |
| Br. Francisco Noriega | Br. Yudimar Misel | Br. Marian Piñango |

Br. Osman Villegas

Barcelona, Mayo 2015

**INDICE**

**Pág.**

**1.** **Diagramas de secuencia** 2

**2.** **Tipos de mensajes** 3

**3.** **Estructura** 3

.

## **Diagramas de secuencia**

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el [diagrama de casos de uso](http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso) permite el modelado de una vista *business* del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

Típicamente se examina la descripción de un [caso de uso](http://es.wikipedia.org/wiki/Caso_de_uso) para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada [caso de uso](http://es.wikipedia.org/wiki/Caso_de_uso) como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos. Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

## **Tipos de mensajes**

Existen dos tipos de mensajes: sincrónicos y asincrónicos. Los mensajes sincrónicos se corresponden con llamadas a métodos del objeto que recibe el mensaje. El objeto que envía el mensaje queda bloqueado hasta que termina la llamada. Este tipo de mensajes se representan con flechas con la cabeza llena. Los mensajes asincrónicos terminan inmediatamente, y crean un nuevo hilo de ejecución dentro de la secuencia. Se representan con flechas con la cabeza abierta.

También se representa la respuesta a un mensaje con una flecha discontinua.

## **Estructura**

**•** El diagrama de secuencias consta de objetos que se representan del modo usual: rectángulos con nombre (subrayado), mensajes entre los objetos representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

• Los objetos se colocan cerca de la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen el diagrama.

• La extensión que está debajo (y en forma descendente) de cada objeto será una línea discontinua conocida como la línea de vida de un objeto.

• Junto con la línea de vida de un objeto se encuentra un pequeño rectángulo conocido como activación, el cual representa la ejecución de una operación que realiza el objeto. La longitud del rectángulo se interpreta como la duración de la activación.

• Un mensaje que va de un objeto a otro pasa de la línea de vida de un objeto a la de otro. Un objeto puede enviarse un objeto a sí mismo (es decir, de su línea de vida a su propia línea de vida).

• Un mensaje puede ser simple, síncrono o asíncrono.

• Un mensaje simple es la transferencia del control de un objeto a otro.

• En el diagrama de secuencias, los símbolos del mensaje varían. Por ejemplo, la punta de la flecha de un mensaje simple está compuesta por dos líneas, la punta de flecha de un mensaje síncrono es un triángulo relleno, y la de uno asíncrono solo tiene una sola línea.

El diagrama representa al tiempo en dirección vertical. El tiempo se inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno que esté cerca la parte inferior.

Con ello el diagrama de secuencias tiene dos dimensiones. La dimensión horizontal es la disposición de los objetos, y la dimensión vertical muestra el paso del tiempo.

En ocasiones un objeto posee una operación que se invoca a sí misma. A esto se le conoce como recursividad, y es una característica fundamental de varios lenguajes de programación. Por ejemplo, supongamos que una calculadora forma parte de los objetos de nuestro sistema y que una de sus operaciones sea el cálculo de intereses. Para calcular el interés compuesto para un periodo que incluya otros periodos, la operación cálculo de intereses del objeto tendrá que invocarse a si misma varias veces. Para representar esto en UML, dibujaremos una flecha de mensaje fuera de la activación, y un pequeño rectángulo encima de la activación.