



Guía de Trabajos Prácticos N° 3: Introducción al Modelado de Interacciones

Introducción

La presente guía de trabajos prácticos se enfoca específicamente en el modelado de interacciones a través de diagramas de secuencia (DS). Contiene ejercicios que ayudan a conocer el uso de DS de forma simplificada y aislada (hasta donde es posible), para luego incorporarlos a la realización de casos de uso prevista en las guías posteriores.

Objetivos de la guía

Lograr que el alumno:

- Adquiera las nociones básicas del modelado de interacciones.
- Utilice los constructores básicos de UML para DS.
- Entienda como derivar el pseudocódigo a partir de los DS.

Modalidad de Trabajo

Grupal, de hasta 6 integrantes

Fundamentos teóricos

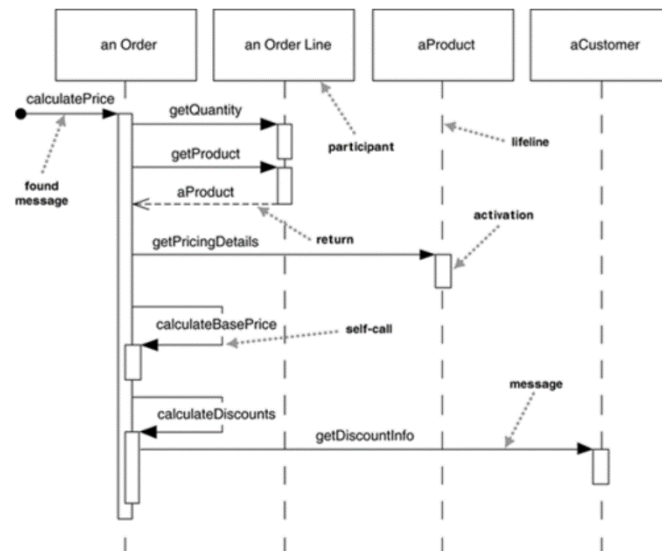
Una interacción es un conjunto de mensajes que se intercambian dentro del contexto de una colaboración por roles de clasificadores (objetos) a través de enlaces (instancias de asociación).

Un mensaje es una comunicación unidireccional entre objetos, un flujo de control con información desde un emisor a un receptor. Un mensaje puede tener parámetros. Puede ser una señal (asíncrono) o una llamada (síncrono).

La secuencia de mensajes puede representarse mediante diagramas de secuencia (se enfocan en la relación de tiempo).

Diagrama de secuencia

Es un gráfico bidimensional. La dimensión vertical es el eje de tiempo. La dimensión horizontal muestra los roles de clasificadores que representan objetos individuales en la colaboración. Cada rol de clasificador es representado por una línea vertical que representa su línea de vida. Una línea punteada representa el período de 'existencia' del objeto. Una línea doble representa una activación del objeto. Un mensaje es representado por una flecha desde la línea de vida de un objeto hacia la línea de vida de otro objeto. La secuencia de mensajes está ordenada en forma descendente en el diagrama.

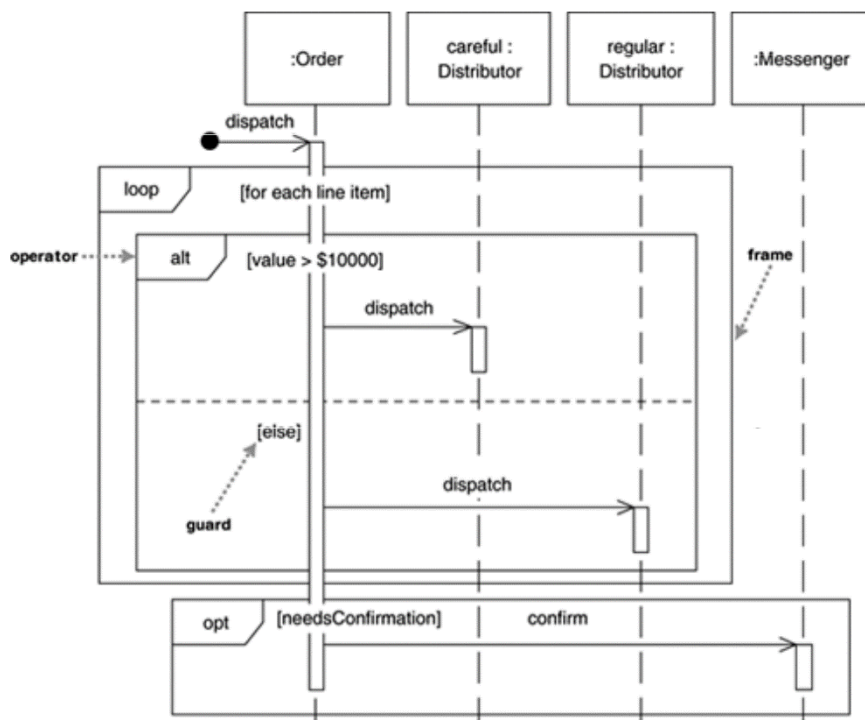


Bucles y condicionales

Para representarlos se utilizan marcos de interacción, que son formas de marcar una porción de un diagrama de secuencia.

Cada marco tiene un operador. La tabla enumera los operadores comunes para marcos de interacción.

ALT	Fragmentos múltiples alternativos; solo aquel cuya condición es verdadera se ejecutará.
LOOP	El fragmento puede ejecutarse varias veces, y el guarda indica la base de la iteración.
OPT	Opcional; el fragmento se ejecuta solo si la condición suministrada es verdadera. Equivalente a un alt con solo un camino.
REF	Referencia; se refiere a una interacción definida en otro diagrama. El marco se dibuja para cubrir las líneas de vida involucradas en la interacción. Puede definir parámetros y un valor de retorno.
SD	Diagrama de secuencia; usado para rodear un diagrama de secuencia completo, si lo desea.





Ejercicio 1

Dados los ejercicios 1 y 2 de la Guía de TP N°5, modele las siguientes interacciones:

- Búsqueda de médico por número de matrícula.
- Búsqueda de pacientes por apellido.
- Registrar un nuevo análisis.

Ejercicio 2

Usando diagramas de secuencia modele las interacciones relacionadas con un home banking:

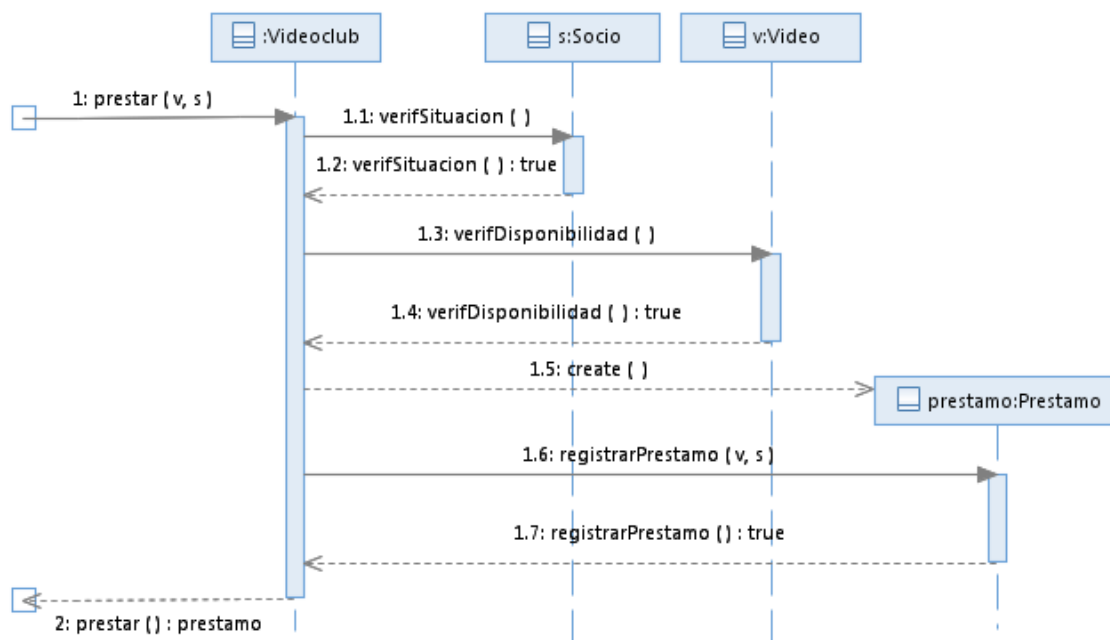
- Autenticación mediante usuario y clave.
- Consulta de saldo.
- Transferencia entre cuentas del cliente.

Considere la existencia de las clases cliente, cuenta, banco, transferencia, si lo cree necesario agregue las clases pantalla y control.

Ejercicio 3

El siguiente diagrama de secuencia muestra la operatoria para el préstamo de un videoclub.

- Confeccionar el diagrama de clases
- Realizar el correspondiente pseudocódigo.



Ejercicio 4

Dado el ejercicio N°4 de la Guía de TP N°5, confeccionar el diagrama de secuencia para registrar una venta a un cliente Premium con acumulación de puntos, para simplificar la interacción suponga que solo involucra un libro por venta.

Ejercicio 5

A partir del siguiente código escrito en lenguaje C# obtenga el diagrama de secuencia correspondiente

```
public class Venta
{
    private List<LineaVenta> lineas = new List<LineaVenta>();

    public Dinero getTotal()
    {
        Dinero total = new Dinero();
        Dinero subtotal = null;
        foreach (LineaVenta linea in lineas)
        {
            subtotal = linea.getSubtotal();
            total.Sumar(subtotal);
        }
        return total;
    }
}
```

Bibliografía

- Rumbaugh, J., & Jacobson, I. (2007). *El lenguaje unificado de modelado: Manual de referencia* (2a. ed.). Madrid: Pearson Addison Wesley.
- Torossi, G. (2003). *Apunte: "Modelado de objetos con UML"*.
- Torossi, G. (2007). *Apunte: "Introducción a la orientación a objetos"*.
- Fowler, M. (2004). *UML distilled: A brief guide to the standard object modeling language* (3rd ed.). Boston: Addison-Wesley.