

Diagramas de Secuencia

ECOS

Juan Pablo Quiroga

Dpto. de Ingeniería de Sistemas y
Computación

Universidad de los Andes

1



Referencia

- The Unified Modeling Language, User Guide. Grady Booch, James Rumbaugh e Ivar Jacobson. Addison Wesley, 1999.
 - Capítulos 15 y 18
- The Unified Modeling Language, Reference Manual. Grady Booch, James Rumbaugh e Ivar Jacobson. Addison Wesley, 1999.
 - Págs. 86-89,

2

Agenda

- Conceptos de interacción
- Conceptos de Diagramas de Secuencia

3

Interacción

- Concepto primordial para denotar el aspecto dinámico de un sistema
- Concepto
 - Una interacción es un comportamiento que compromete un conjunto de mensajes intercambiados entre un conjunto de objetos dentro de un contexto para lograr un propósito
- Ejemplo
 - Para inscribirEstudiante en el objeto sistema se envían mensajes como crear un objeto estudiante y añadirlo a un objeto contenedor de estudiantes

4

Mensaje

■ Concepto

- Es la especificación de una comunicación entre objetos en la que se transmite información con la expectativa que la actividad se lleve a cabo.

5

Aspectos comunes en una interacción

■ Objetos

- Participantes en la interacción

■ Roles

- Jugados por los objetos

■ Enlaces

- Conexión semántica entre objetos

■ Mensajes

- Comunicación entre objetos

■ Secuenciación

- Orden de los mensajes

6

Diagramas de interacción

- Concepto
 - Muestra una interacción, que consiste en un conjunto de objetos y las relaciones entre ellos, incluyendo los mensajes que deben ser enviados entre ellos.
- Es un macroalgoritmo descrito gráficamente

7

Tipos de Diagramas de interacción

- Diagramas de secuencia
 - Enfatiza el tiempo que indica el orden de los mensajes
 - Por lo general es útil para describir escenarios donde existe interacción con el usuario.
- Diagrama de colaboración
 - Enfatiza la organización estructural de los objetos que envían y reciben información
 - Por lo general es más utilizado para mostrar el diseño detallado de un procedimiento (método)

8

Diagramas de secuencia

■ Características

- Se muestran los objetos que interactúan
- Se muestra el tiempo de vida de un objeto
- Se muestran los mensajes que se envían los objetos
- Se muestra el tiempo durante el cual un objeto se encuentra activo (completando el llamado del mensaje) – Foco de control
- Se muestra el envío y retorno de información de un mensaje.
- Se muestra el flujo de control de los mensajes.

9

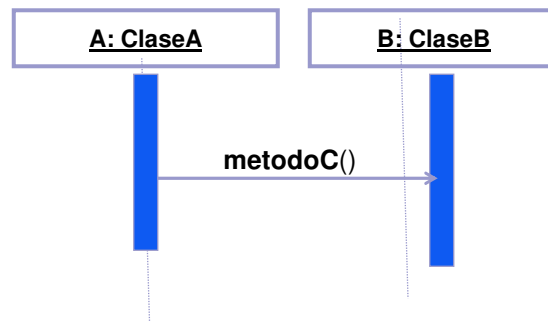
Diagramas de secuencia

- “Macroalgoritmo gráfico”
- Herramienta para visualizar y estudiar las interacciones entre los objetos
- Se hace énfasis en el orden en el que ocurren las interacciones
- Permite ver la realización de un requerimiento, o la realización a un método
- El primer objeto que aparece es el receptor del estímulo

10

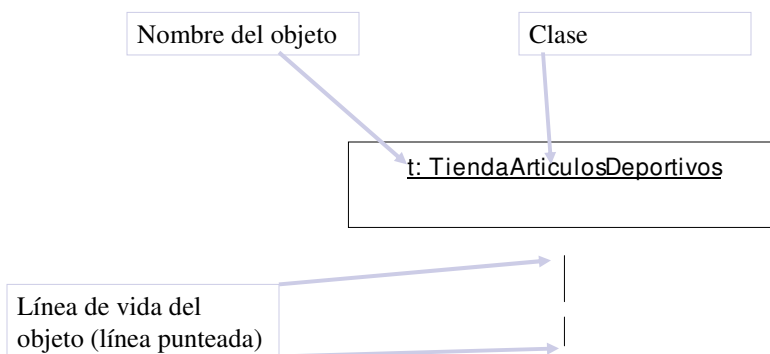
Diagramas de secuencia...

- El objeto al que se le envía el mensaje (B) debe ser visible por el objeto que envía el mensaje (A) y el mensaje debe corresponder a un método de la clase de B



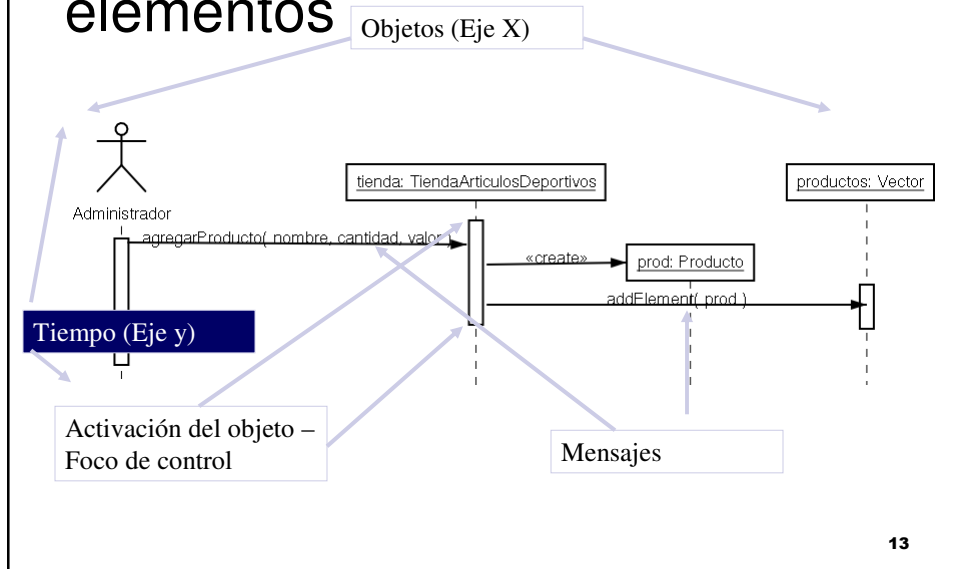
11

Diagramas de secuencia

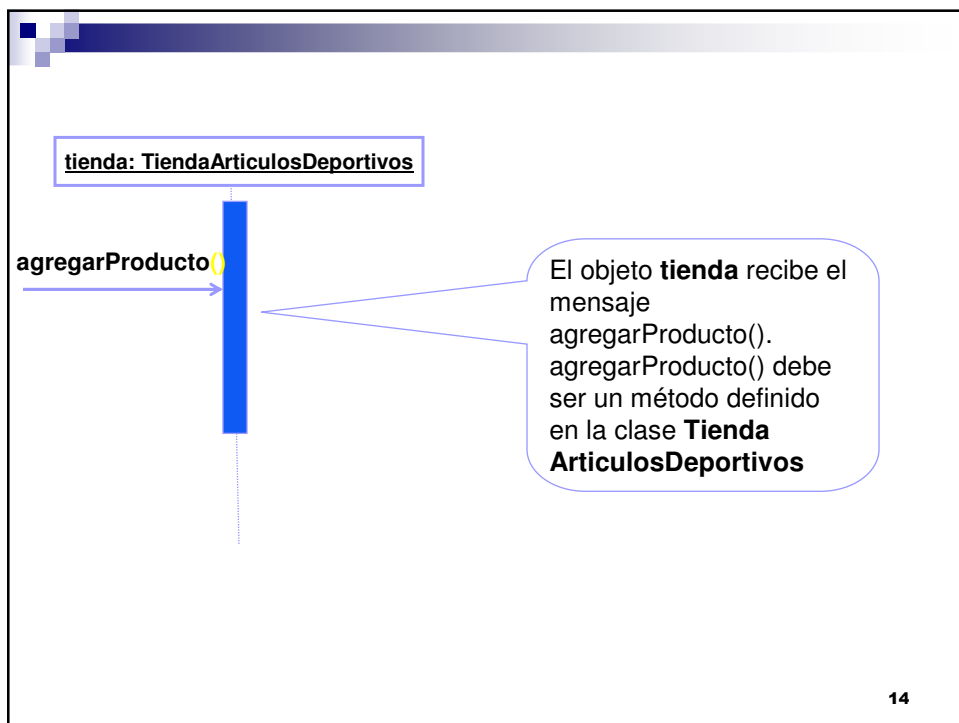


12

Diagramas de secuencia - elementos



13



14

Acciones

- Es una instrucción ejecutable que forma la abstracción de un procedimiento computacional
- Puede resultar en un cambio del estado del objeto

15

Acciones - Tipos

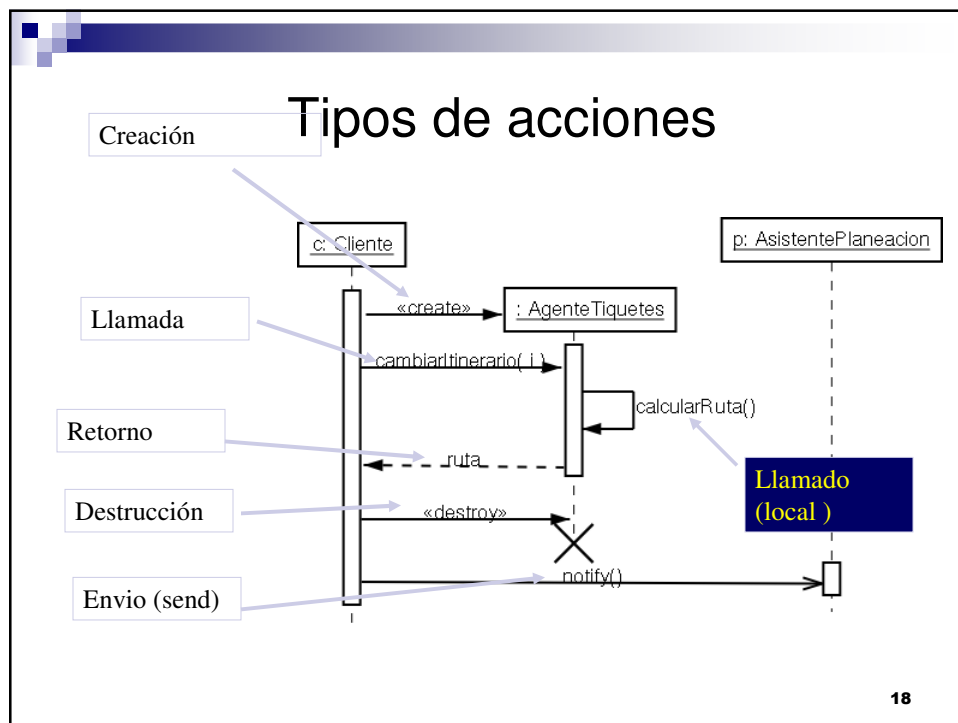
- Llamado (Call)
 - Invoca una operación de un objeto
 - Un objeto puede enviarse un mensaje a si mismo (invocación local de una operación)
- Retorno (return)
 - Retorna un valor al objeto que realizó la llamada
- Envio (send)
 - Enviar una señal al objeto
 - Una señal es un mensaje asincrónico

16

Acciones - Tipos

- Create
 - Crear un objeto
- Destroy
 - Destruir un objeto
 - Un objeto puede destruirse a si mismo

17



18

Llamado de mensajes

- Un objeto puede llamar un método de otro objeto cuando:
 - Association - Asociación con el objeto – atributo (asociación simple, agregación o composición)
 - Parameter - Se pasa como parámetro al método en si (asociación de dependencia)
 - Local - Se crea y se usa dentro del método (asociación de dependencia)

19

Llamado de mensajes

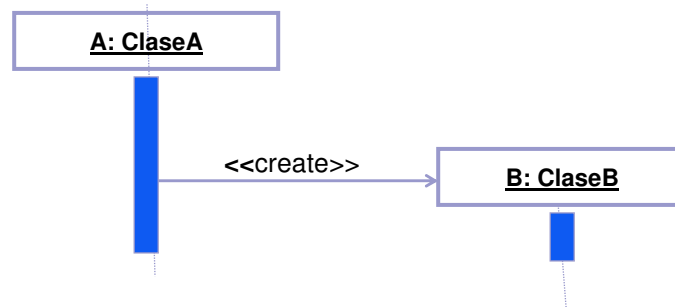
- Un objeto puede llamar un método de otro objeto cuando:
 - Local -Se recibe como resultado del llamado de una operación es decir un retorno (asociación de dependencia)
 - Self – Es el mismo objeto y está disponible porque es el despachador de la operación
 - Global – Es un objeto global que se encuentra dentro del alcance
 - Java se implementa con un atributo de clase

20

Diagramas de secuencia...

- La visibilidad ocurre cuando:

caso 1: A crea B



21

Diagramas de secuencia...

- La visibilidad ocurre cuando:

caso 2: Existe una relación entre la clase A y la Clase B con cardinalidad 1 en el lado de B

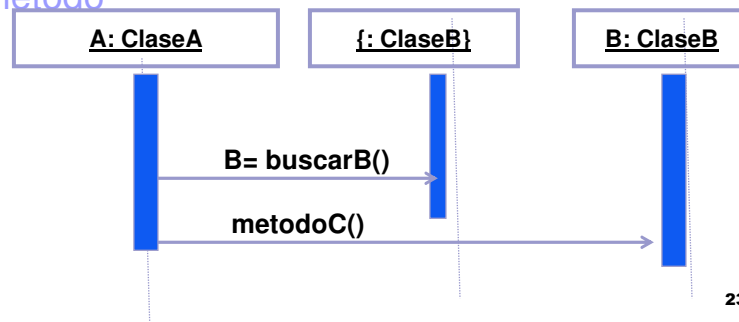


22

Diagramas de secuencia...

- La visibilidad ocurre cuando:

caso 3: El objeto B fue recibido por el objeto A como resultado de la ejecución de algún método

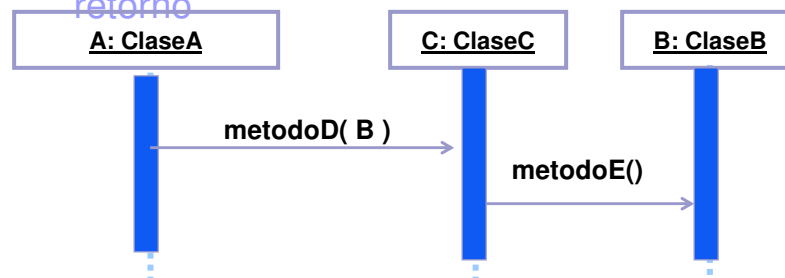


23

Diagramas de secuencia...

- La visibilidad ocurre cuando:

caso 4: El objeto B fue recibido como parámetro por el objeto C (dependencia). El objeto A lo tenía por asociación o por retorno



24

Diagramas de secuencia...

- En el mensaje se coloca el nombre y la información que necesita
- La información puede provenir del inicio del diagrama o de objetos que han aparecido durante la reacción
- Entre { :Clase1 }, se coloca cuando el objeto es una contenedora de objetos de tipo Clase1
- Con * se indica que el mensaje se envía múltiples veces (contenedoras)
- Entre [] se indica una guarda (expresión booleana) que hace que el mensaje se envíe

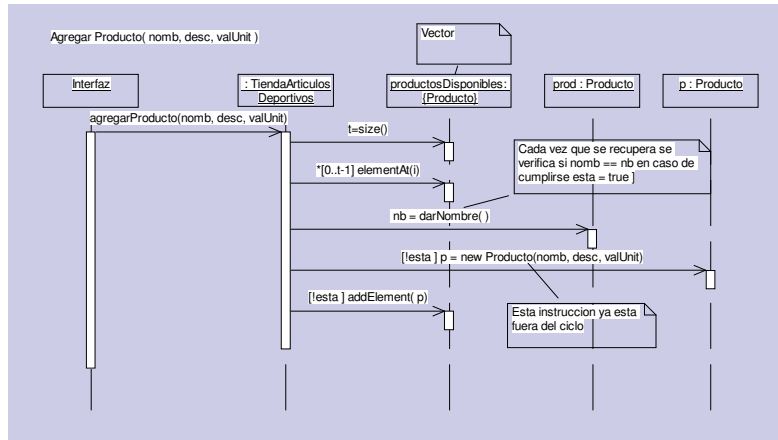
25

Diagramas de Secuencia

- Elementos
 - Objetos
 - Mensajes
 - Enlaces (diagramas de colaboración)
 - Foco
 - Tiempo de vida del objeto
 - Guardas
 - Contenedoras
 - Ciclos

26

Diagramas de Secuencia



27

Relación con el diagrama de clases

- Tiene una relación directa
 - Para reconocer que objetos pueden interactuar en el diagrama de secuencia
 - Para reconocer los métodos posibles
 - Para completar el diagrama de clases
 - Para reconocer más métodos de una clase
 - Para crear asociaciones entre clases

28

Modelar el flujo de control

- Indicar el contexto de la interacción
 - Sistema, clase o operación individual
- Indicar la etapa de la interacción identificando cuales objetos juegan un rol, colocar las propiedades iniciales, incluyendo los valores de los atributos, estado y rol.
- Si el modelo enfatiza la organización estructural, identificar los enlaces entre ellos.

29

Modelar el flujo de control

- Si enfatiza el orden en el tiempo, especificar los mensajes que pasan de objeto a objeto.
- Colocar aspectos adicionales que permitan reconocer cuál es el estado y rol del objeto en cada momento

30

Aspectos a tener en cuenta

- Una interacción bien estructurada
 - Es simple y sólo muestra los objetos que trabajan entre si para responder a un comportamiento mayor
 - Tiene un contexto claro y representa la interacción de los objetos en el contexto de la operación, clase o el sistema como un todo
 - Es eficiente y puede contener comportamiento con un balance óptimo de tiempo y recursos

31

Aspectos a tener en cuenta

- Una interacción bien estructurada
 - Es adaptable y los elementos de la interacción que son dados al cambio pueden ser identificables de forma que sean modificables fácilmente
 - Es entendible, sin cosas extrañas, efectos escondidos o semántica oscura

32

Aspectos a tener en cuenta

- Escoger un énfasis de la interacción.
 - Orden de los mensajes en el tiempo
 - Secuencia de mensajes en el contexto de una estructura organizacional de los objetos
 - No los dos al tiempo
- Mostrar solo las propiedades del objeto (atributos, roles y estado) que son importantes para entender la interacción en el contexto.

33

Aspectos a tener en cuenta

- Mostrar sólo las propiedades de cada mensaje(parámetros, semántica de concurrencia, valor retorno) que son importantes para entender la interacción en el contexto.

34