

DIAGRAMAS DE SECUENCIA

Ejemplo - Figura 6.39. (pag. 335)

Realiza haciendo uso de ArgoUML el siguiente diagrama de secuencia correspondiente al caso de uso ComprarProductos.

Identifica los distintos elementos que forman parte del diagrama de secuencia y explícalo.

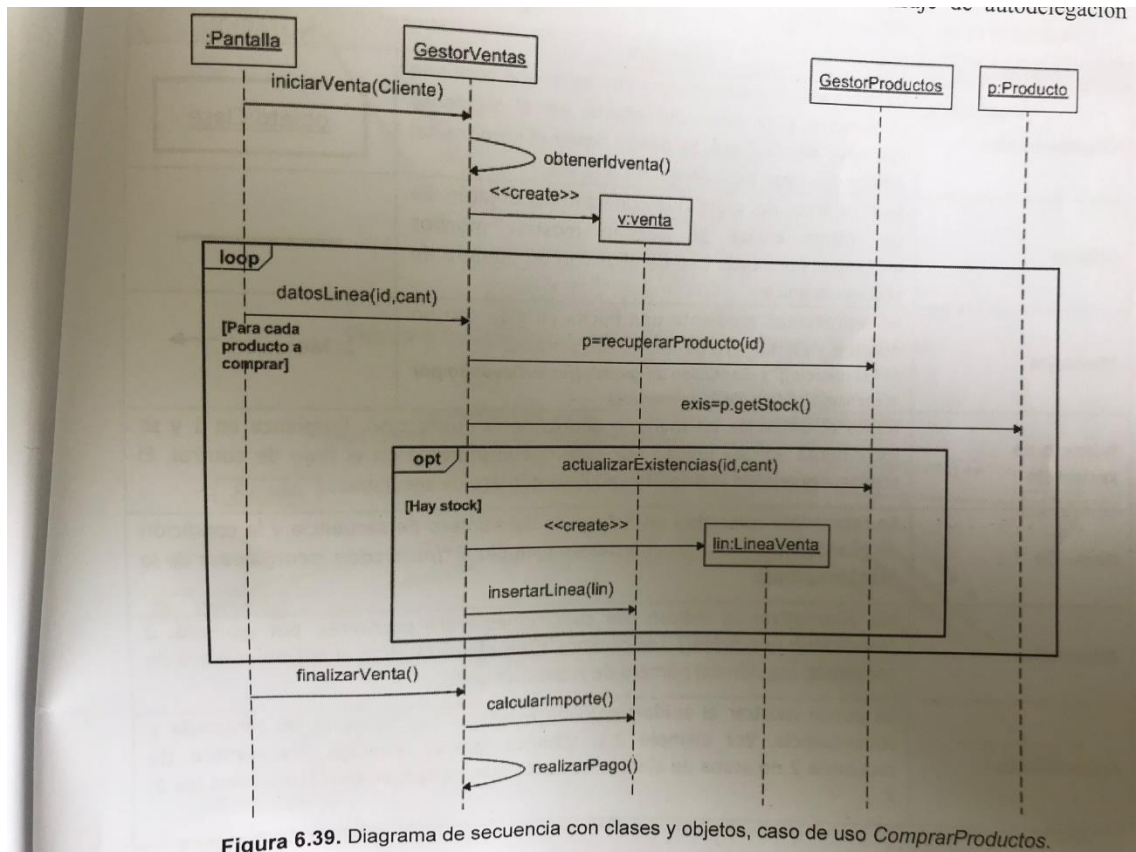


Figura 6.39. Diagrama de secuencia con clases y objetos, caso de uso ComprarProductos.

Nota: No se adjunta captura del ArgoUML, porque al reducir las dimensiones del diagrama para su ajuste en la captura, los textos se vuelven ilegibles.

1. En primer lugar, debemos **seleccionar la vista de Diagrama de Secuencia** para poder elaborar correctamente nuestro documento y obtener las opciones y elementos propios de los DS.



2. Lo primero, será **crear los distintos roles** principales que van a intervenir en el DS, los cuales presentan un diseño de “columna con cabecera”.

Cada rol, ya sea actor u objeto, **se representa mediante un rectángulo distribuido horizontalmente** en la zona superior del diagrama. **A cada uno se le asocia una línea vertical** llamada **línea de vida**, donde se describe la interacción a lo largo del tiempo.

De cada línea vertical **salen diferentes mensajes representados mediante flechas que muestran la interacción**, encima de la flecha que se muestra el mensaje.

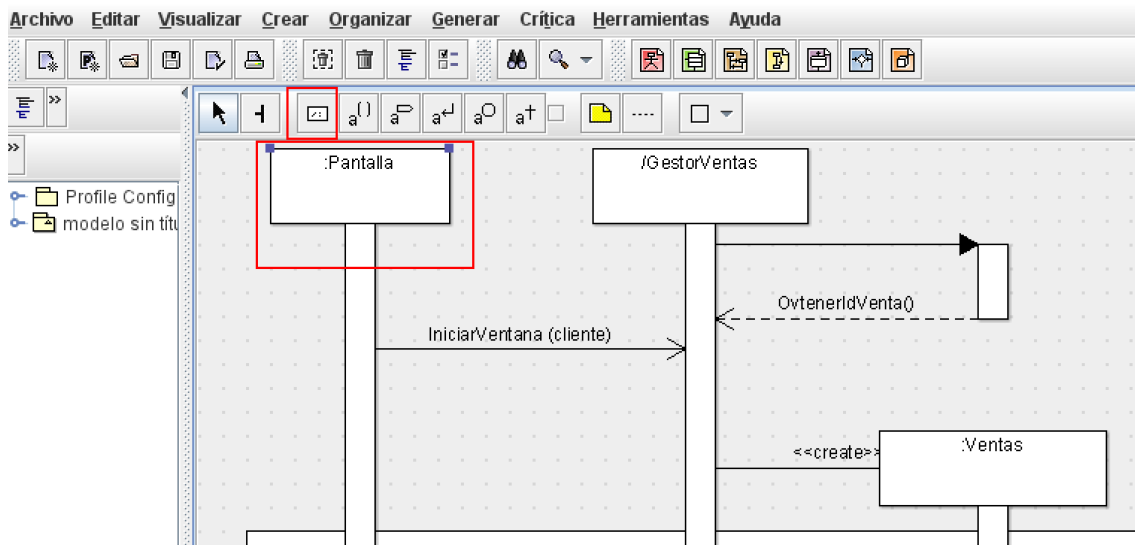
Es importante la secuencia de los mensajes en esta línea ya que indican el orden en que ocurren los eventos.

A menudo el DS va acompañado de la descripción del curso normal de eventos del caso de uso.

La línea de vida representa a un participante durante la interacción.

Normalmente contiene un rectángulo con el nombre del objeto y una línea punteada.

Algunas veces un DS tendrá una línea de vida con un símbolo de actor en la parte superior.



Nota: Existe una nomenclatura a la hora de nombrar los rectángulos que representan los objetos:

nombre:tipo

Tanto el nombre como el tipo pueden omitirse, pero no ambos a la vez.

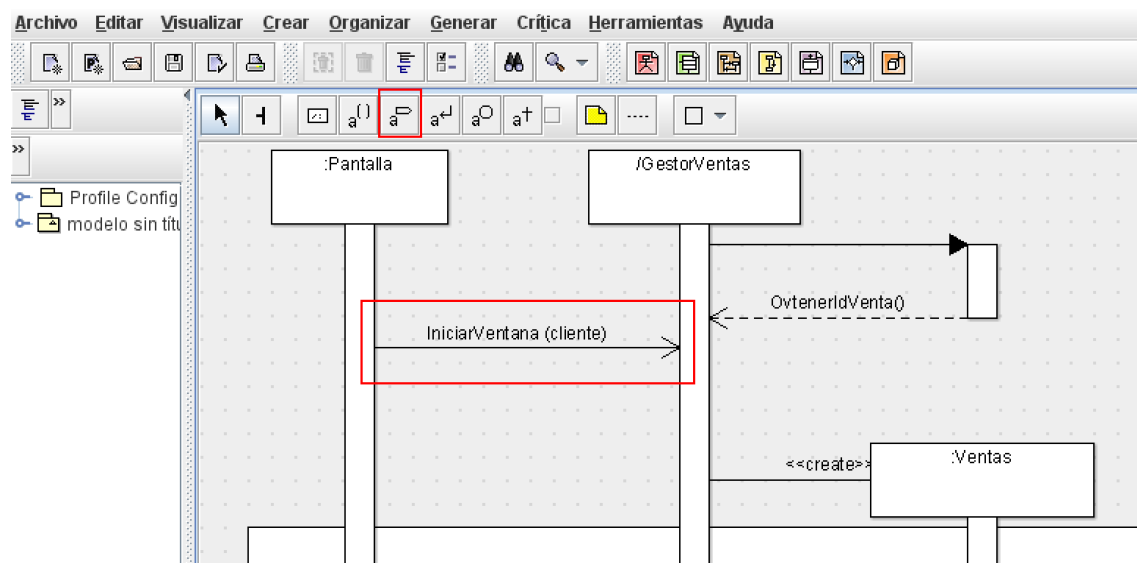
El nombre representa el objeto dentro de la interacción, y el tipo es el tipo de objeto (es decir, la clase).

3. **Para crear una Acción de Envío**, la cual tiene un diseño de flecha continua y acabada en punta sin relleno, pulsar el botón que se indica a continuación y al seleccionar un punto de nuestro DS, mantener pulsado “click” y arrastrar hasta la columna del rol al que queramos llevar la Acción de Envío.

Esta acción es la que se conoce como **mensaje asíncrono**.

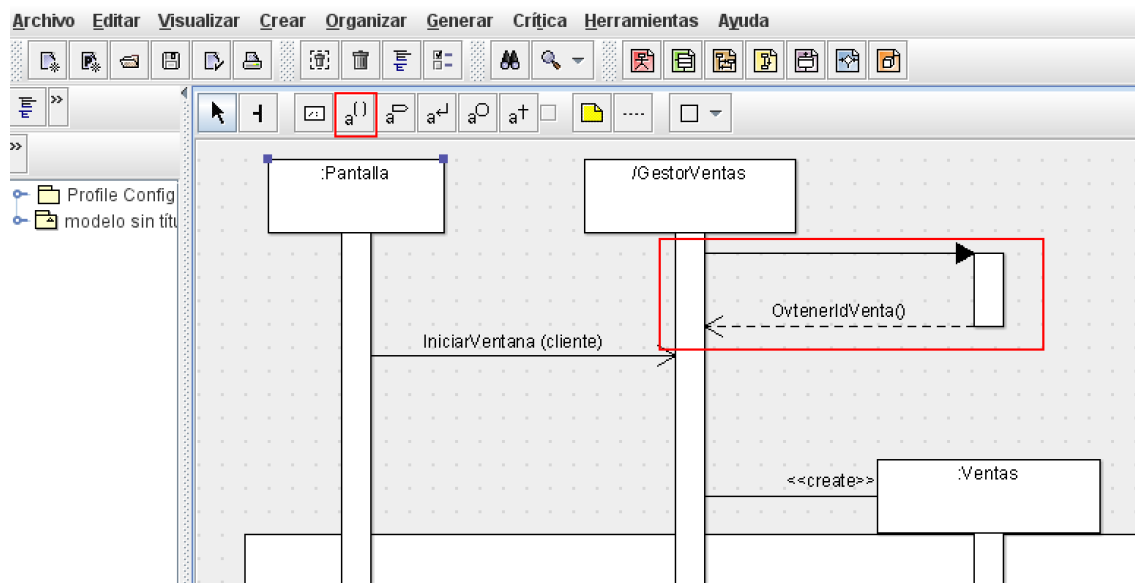
Éste representa un flujo de control asíncrono.

En este caso, **quien envía un mensaje asíncrono continua con su trabajo después de enviado**, es decir, **no necesita esperar la finalización del mismo en el objeto receptor**.
Se utiliza en sistemas multihilo donde se producen **procesos concurrentes**.



4. **Para crear una Acción de Llamada**, la cual tiene un diseño de doble flecha, cada una en sentido contrario a la otra, siendo la línea superior continua y con relleno, y la inferior discontinua y sin relleno, pulsar el botón que se indica a continuación y al seleccionar un punto de nuestro DS, mantener pulsado “click” y arrastrar hasta la columna del rol al que queramos llevar la Acción de Envío, o en caso de no arrastrar hasta una columna de un rol, se creará un pequeño recuadro a modo de “trozo de columna de rol”, como indicando que debe apuntar a algún elemento.

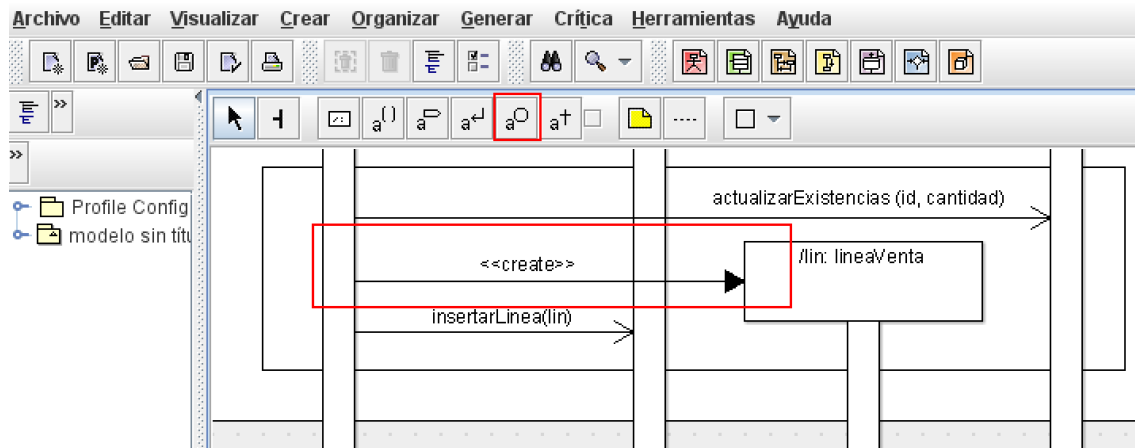
Esto se conoce como **mensaje de retorno**, el cual **representa un mensaje de confirmación**. *(Su uso es opcional)*



5. **Para crear una Acción**, la cual tiene un diseño de flecha continua y rellena, pulsar el botón que se indica a continuación y al seleccionar un punto de nuestro DS, mantener pulsado “click” y arrastrar hasta la columna del rol al que queramos llevar la nueva Acción, e inmediatamente la columna del rol al que vayamos a apuntar descenderá su cabecera para recibir la punta de la flecha de la Acción.

Esto se conoce como un **mensaje síncrono**.

Cuando se envía un mensaje a un objeto, no se recibe el control hasta que el objeto receptor ha finalizado la ejecución.



Nota: en la imagen anterior tenemos otro ejemplo de crear un Acción en la flecha <<create>> que apunta hacia la cabecera de :Ventas

En un DS se puede mostrar la creación y destrucción de objetos.

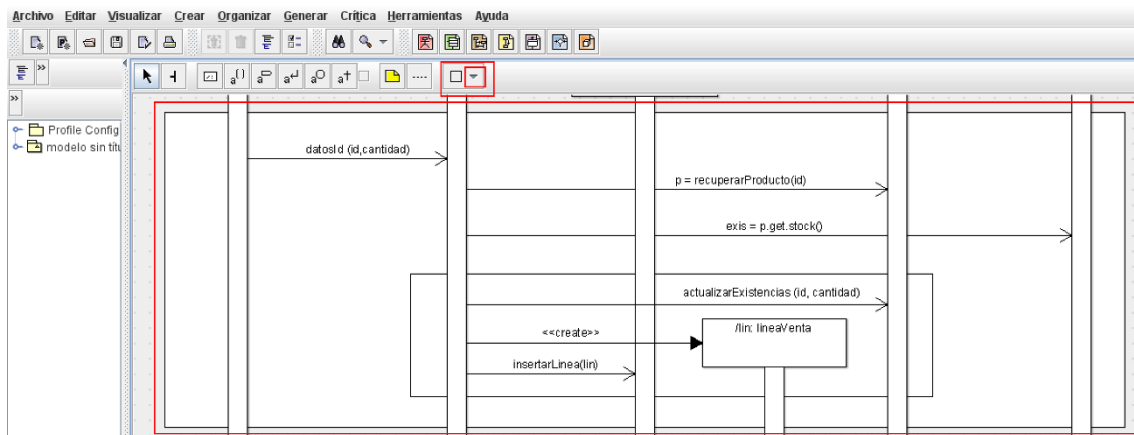
La creación se representa mediante un mensaje que termina en el objeto que va a ser creado. El mensaje puede llevar la identificación <<create>>.

La destrucción del objeto da lugar a la finalización de su línea de vida, y **se denota** mediante una “X” grande en su línea de vida. El mensaje puede llevar la identificación <<destroy>>.

6. **Para crear el espacio rectangular** que se sitúa en medio de este ejemplo de DS y que engloba distintas acciones de llamada y otros roles y elementos, pulsar el botón que se indica a continuación, o si no está ya preseleccionado, pulsar la flecha hacia abajo que indica desplegar otras opciones, y seleccionar la forma de rectángulo, y al seleccionar un punto de nuestro DS, mantener pulsado “click” y arrastrar hasta conseguir el tamaño y dimensión que se desea abarcar en el espacio de trabajo del DS. Posteriormente, hacer click derecho sobre el espacio rectangular que acabamos de crear y dentro de la opción de “Ordering”, seleccionar la opción de “al fondo”.

Esto se conoce como **marco**, y se utiliza para dar un **borde visual** al DS.

A la izquierda del marco se escribe la etiqueta “sd” seguida de un nombre.



Nota: En un DS los mensajes que se repiten se pueden incluir dentro de un marco. Por ejemplo, los mensajes se pueden encerrar en un marco etiquetado con “loop”, y debajo se escribe la condición del bucle entre corchetes [], indicando así que se repita mientras ocurre el evento.

También se pueden representar bucles y otras alternativas utilizando la extensión denominada “fragmento combinado”, los cuales utilizarán una palabra clave o etiqueta (loop, opt, alt) para indicar el tipo del que se trata.

- Utilizando el operador “**opt**” seguido de una condición, **si la condición se cumple, el contenido del marco se ejecuta.**
- Utilizando el operador “**alt**”, éste va seguido de varias condiciones **y al final, la palabra clave “else”**.
El marco **se divide en varias zonas, dependiendo de las condiciones** que haya, cuyo contenido **se ejecuta si se cumple la condición** asociada.
La parte “else” se ejecuta si no se cumplen las condiciones asociadas.
Se utiliza **para modelar estructuras “if”, “then”, “else”...**
- Para representar un bucle se utiliza el operador “**loop**” seguido de una condición.
Lo encerrado en el marco se ejecutará mientras se cumpla la condición.