

DIAGRAMA DE INTERACCIÓN

Los diagramas de interacción describen el intercambio de secuencias de mensajes entre las partes de un sistema. Proporciona una visión integral del comportamiento de un sistema, es decir, muestra el flujo de control a través de varios objetos.

Dentro de la categoría de diagramas de interacción tenemos dos diagramas que se centran en aspectos distintos:

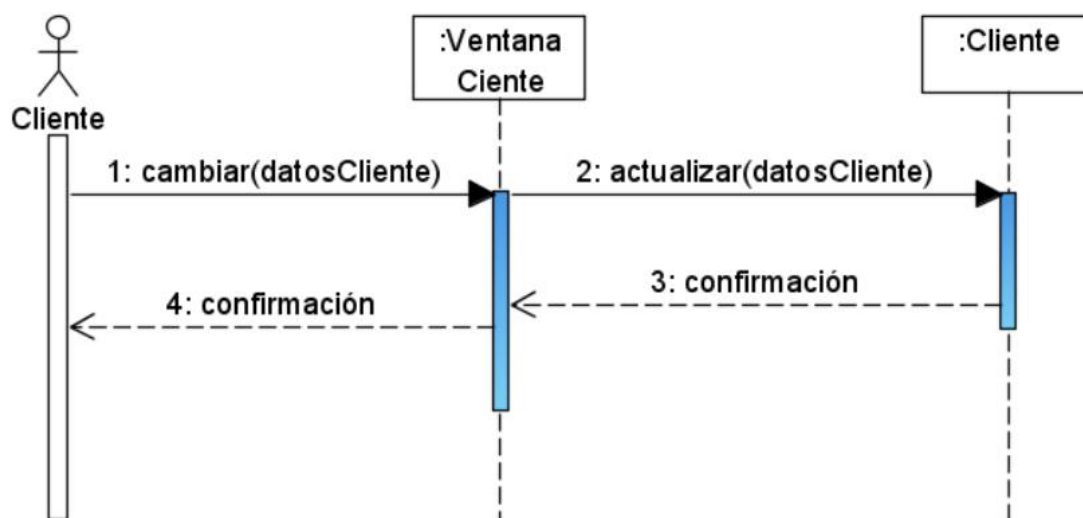
- Diagrama de secuencia, centrado en la ordenación temporal de los mensajes.
- Diagrama de comunicación (a partir de UML 2.x) o colaboración (en UML 1.x), centrado en la organización estructural de los objetos que envían y reciben mensajes.

Diagrama de secuencia

Muestra gráficamente los envíos que fluyen de los actores al sistema.

Para su elaboración se parte de los casos de uso elaborados durante la etapa de análisis. Un diagrama de secuencia tiene dos dimensiones:


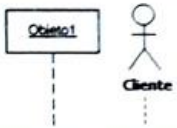


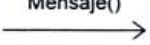


- La dimensión vertical: representa el tiempo.
- La dimensión horizontal: La dimensión horizontal muestra los roles (actores u objetos) que intervienen en la interacción. Cada rol se representa mediante una columna vertical, llamada línea de vida (cuando un objeto se destruye, se finaliza su línea de vida con una cruz). Durante el tiempo que existe un objeto, este se muestra con una línea discontinua. Durante el tiempo que dura una activación de un procedimiento en el objeto, la línea de vida se dibuja como una línea doble.



De cada línea vertical salen diferentes mensajes representados mediante flechas que muestran la interacción, encima de las cuales se muestra el correspondiente mensaje. Es importante la secuencia de los mensajes en esta línea, ya que indican el orden en que ocurren los eventos.

En la figura anterior se muestran horizontalmente los roles que participan en la interacción: el cliente, el objeto “Ventana Cliente” y el objeto “Cliente”, con sus líneas de vida. Se muestran también varios mensajes de distintos tipos (se verán más adelante), el primero de los cuales parte del cliente. El diagrama muestra el proceso de cambio de datos de un cliente en una aplicación.

Los principales elementos de un diagrama de secuencia son los siguientes:

SÍMBOLO	FUNCIÓN	NOTACIÓN
Marco	Se utiliza para dar un borde visual al diagrama de secuencia. A la izquierda del marco se escribe la etiqueta sd seguida de un nombre	
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 diagrama de secuencia tendrá una línea de vida con un símbolo de actor en la parte superior	
Actor	Representa el papel desempeñado por un usuario	
Mensaje	Mensaje síncrono	Mensaje() 
	Mensaje asíncrono	Mensaje() 
	Mensaje de retorno	Retorno 
Activación	Son opcionales. Representan el tiempo durante el que se ejecuta una función. Se suelen poner cuando está activo un método, ya sea porque está efectuando operaciones o porque se encuentra esperando la devolución de otro método	

Los mensajes representan la comunicación entre los participantes. Se dibujan como flechas dirigidas desde el participante que lo envía hasta el que lo ejecuta. Se etiquetan con un nombre acompañado o no de parámetros.

Pueden ser de varios tipos:

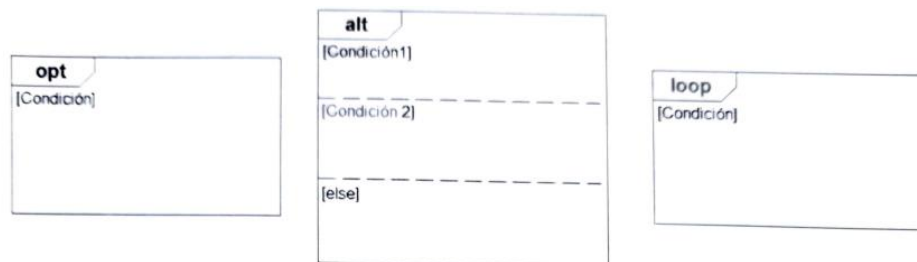
- *Mensaje síncrono*: cuando se envía un mensaje a un objeto, no se recibe el control hasta que el objeto receptor ha finalizado la ejecución.

- *Mensaje asíncrono*: en este caso, quien envía el mensaje continúa con su trabajo después de enviado, sin esperar a la finalización del mismo en el objeto receptor.
- *Mensaje de retorno*: representa un mensaje de confirmación. Su uso es opcional.

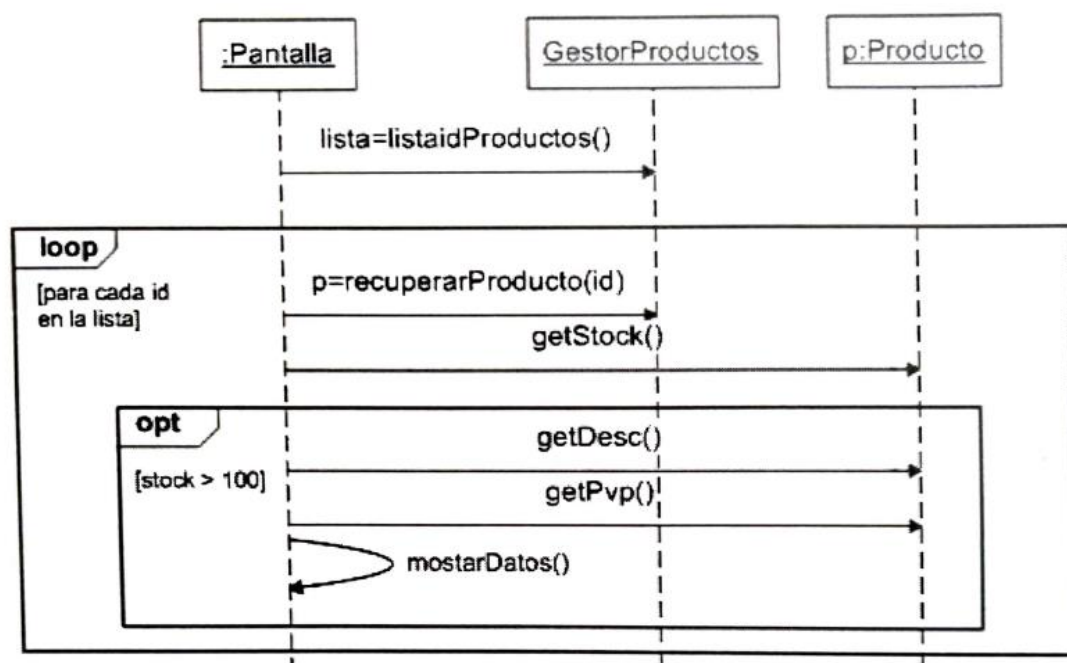
Para representar bucles y alternativas en los diagramas de secuencia, se introducen unas extensiones denominadas **fragmentos combinados**, que se etiquetan con las palabras clave loop, opt y alt:

- Opt: Se emplea seguido de una condición. Si la condición se cumple, el contenido del marco se ejecuta.
- Alt: va seguido de varias condiciones y la palabra clave else al final. El marco se divide en varias zonas, cada una de las cuales se ejecutará cuando se cumpla su condición asociada.
- Loop: va seguido de una condición. El contenido del marco se ejecutará mientras se cumpla la condición.

En la siguiente figura se muestran los tipos de marcos o fragmentos combinados que representan alternativas y bucles:



Y en la siguiente se muestra un ejemplo de su uso dentro de un diagrama de secuencia:



Para finalizar, se muestra un diagrama de secuencia complejo que modela el siguiente comportamiento:

- Desde el objeto pantalla se envía el mensaje `iniciarVenta(Cliente)` a la clase `GestorVentas`.
- Desde esta clase, se envía un mensaje a sí misma, `obtenerIdVenta()`, para obtener el identificador de la venta.
- Una vez obtenido el identificador se crea un objeto `Venta`.
- Se encierra en un marco etiquetado con `loop` el conjunto de operaciones que se van a repetir por cada producto que desee comprar el cliente.
- El objeto pantalla envía el mensaje `datosLinea(id, cantidad)` al gestor de ventas, con el identificador de producto y la cantidad deseada.
- Desde la clase `GestorVentas` se envía el mensaje `recuperarProducto(id)` a la clase `GestorProductos` solicitando los datos del producto cuyo identificador se envía. La operación devuelve un objeto de la clase `Producto`.
- Desde la clase `GestorVentas` se envía el mensaje `getStock()` al objeto producto recuperado para obtener el stock actual del producto.
- Se encierra en un marco etiquetado con `opt` el conjunto de operaciones que se realizan si hay stock en el producto solicitado.
- Si hay stock el gestor de ventas envía el mensaje `actualizarExistencias(id, cant)` al gestor de productos para que actualice el stock. Después crea un objeto `LineaVenta` con los datos del producto y la cantidad. Por último, envía el mensaje `insertarLinea(lin)` al objeto venta creado inicialmente para añadir una nueva línea a la venta.
- Al finalizar el proceso repetitivo, cuando el cliente no desea comprar más productos, desde el objeto pantalla se envía el mensaje `finalizarVenta()` al gestor de ventas.
- Entonces el gestor de ventas envía el mensaje `calcularImporte()` al objeto venta creado.
- Por último, desde la clase `GestorVentas` se envía el mensaje de autodelegación `realizarPago()`.

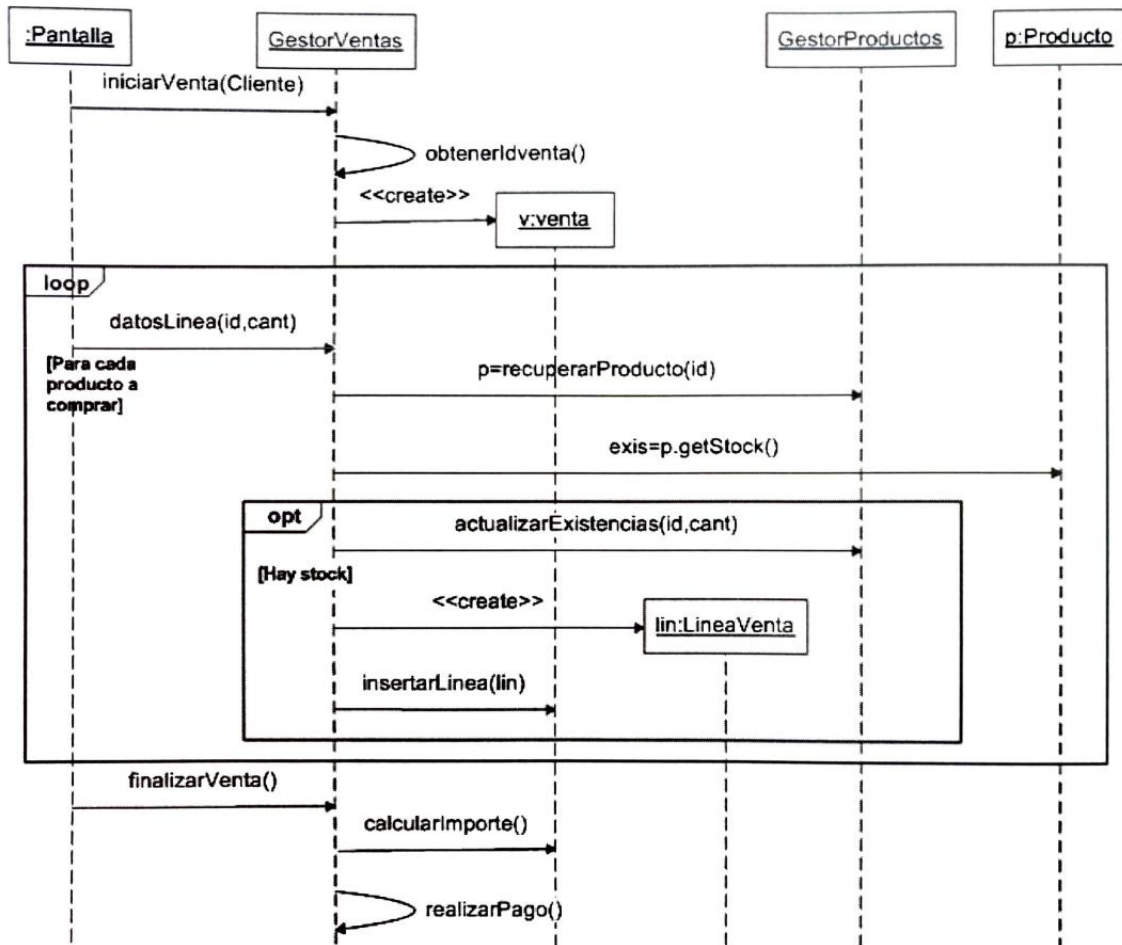


Diagrama de comunicación

El diagrama de comunicación muestra los objetos junto con los mensajes que se envían entre ellos. Representa la misma información que el diagrama de secuencia, de hecho, muchas herramientas para modelar UML permiten convertir diagramas de secuencia en diagramas de colaboración y viceversa.

A diferencia del diagrama de secuencia, centrado en el orden a través del tiempo en que ocurren los mensajes, el diagrama de comunicación se centra en el contexto y la organización general de los objetos que envían y reciben mensajes. El diagrama de objetos muestra los objetos y sus relaciones, el de colaboración es una extensión de este, ya que, a mayores de la relación entre los objetos, muestra los mensajes que se envían entre sí.

Un diagrama de colaboración es un grafo con los siguientes elementos:

SÍMBOLO	FUNCIÓN	NOTACIÓN
Objetos o roles	Se representan mediante un rectángulo que contiene el nombre y la clase del objeto en el siguiente formato: <i>objeto:Clase</i> . Se puede omitir el nombre del objeto o la clase pero no ambos a la vez	<div style="border: 1px solid black; padding: 5px; display: inline-block;"><u>objeto:Clase</u></div>
Enlaces	Son los arcos del grafo que conectan los objetos. En un mismo enlace se pueden mostrar muchos mensajes, pero cada uno de ellos con un número de secuencia único	—————
Mensajes	Se representan mediante una flecha dirigida con un nombre y un número de secuencia	1: Mensaje() →
Número de secuencia	Indica el orden de un mensaje dentro de la interacción. Comienza en 1 y se incrementa en una unidad por cada nuevo mensaje en el flujo de control. El mensaje que inicia el diagrama no lleva número de secuencia	
Iteración	Se representa colocando un * después del número de secuencia y una condición encerrada entre corchetes. Ejemplo: 2*[para cada identificador de la lista]:prepara()	
Alternativa	Las alternativas se indican con condiciones entre corchetes, por ejemplo: 2 [condición]: operación(). Los caminos alternativos tendrán el mismo número de secuencia, seguido del número de subsecuencia	
Anidamiento	Se puede mostrar el anidamiento de mensajes con números de secuencia y subsecuencia. Por ejemplo 2.1, significa que el mensaje con número de secuencia 2 no acaba de ejecutarse hasta que no se han ejecutado todos los 2. X	