



SHP

Resumen Horas 8, 9 y 10 del libro aprendiendo
UML en 24 horas.

Alumno: Orbezo Hernández Gustavo
Matricula: 7576

Contenido

| | |
|--|----|
| Índice de figuras. | 2 |
| Hora 8 Diagramas de estado | 3 |
| Qué es un diagrama de estados..... | 3 |
| Simbología | 3 |
| Adición de detalles al icono de estado..... | 4 |
| Sucesos y acciones..... | 5 |
| Condiciones de seguridad..... | 6 |
| Subestados | 6 |
| Subestados secuenciales..... | 6 |
| Subestados concurrentes..... | 7 |
| Estados históricos | 7 |
| Mensajes y señales..... | 8 |
| Por qué son importantes los diagramas de estados | 8 |
| Hora 9 Diagramas de secuencias | 9 |
| Qué es un diagrama de secuencias | 9 |
| Objetos..... | 9 |
| Mensaje | 10 |
| Tiempo | 10 |
| La GUI..... | 10 |
| La secuencia | 11 |
| El diagrama de secuencias | 11 |
| El caso de uso..... | 11 |
| Instancias y genéricos..... | 12 |
| Un diagrama de secuencias de instancias | 12 |
| Un diagrama de secuencias genérico..... | 12 |
| Creación de un objeto en la secuencia..... | 13 |
| Cómo representar la recursividad..... | 13 |
| Hora 10 Diagramas de colaboraciones | 14 |
| Qué es un diagrama de colaboraciones | 15 |
| La GUI..... | 15 |
| Cambios de estado | 16 |
| Creación de un objeto | 17 |
| Algunos conceptos más | 18 |
| Varios objetos receptores en una clase..... | 18 |

| | |
|--|----|
| Representación de los resultados | 19 |
| Objetos activos..... | 19 |
| Sincronización | 19 |
| Adiciones al panorama..... | 20 |
| Bibliografía..... | 21 |

Índice de figuras.

| | |
|--|----|
| Figura 1 Símbolo de estado | 4 |
| Figura 2 Subdividir el símbolo de estado | 4 |
| Figura 3 La máquina de fax es un buen ejemplo de un estado con variables y actividades | 5 |
| Figura 4 Los estados y transiciones de una interfaz gráfica del usuario..... | 5 |
| Figura 5 Subestados secuenciales | 6 |
| Figura 6 Los subestados concurrentes suceden al mismo tiempo. | 7 |
| Figura 7 Un estado compuesto recuerda su subestado activo cuando el objeto trasciende fuera de tal estado compuesto..... | 8 |
| Figura 8 Representación de un objeto en un diagrama de secuencias. | 9 |
| Figura 9 Símbolo para los mensajes en los diagramas de secuencias. | 10 |
| Figura 10 Conjunto básico de elementos de un diagrama de secuencias..... | 10 |
| Figura 11 Diagrama de secuencias..... | 11 |
| Figura 12 Diagrama de caso de uso | 12 |
| Figura 13 Representación de la recursividad en un diagrama de secuencias. | 13 |
| Figura 14 El panorama del UML | 14 |
| Figura 15 Simbología del diagrama de colaboración..... | 15 |
| Figura 16 Diagrama de ejemplo para la GUI | 16 |
| Figura 17 Un diagrama de colaboración puede incluir cambios de estado..... | 17 |
| Figura 18 Diagrama de colaboraciones "crear una propuesta" | 18 |
| Figura 19 Un objeto que envía un mensaje a diversos objetos de una clase. | 18 |
| Figura 20 La sincronización de mensajes. | 20 |
| Figura 21 El panorama del UML. | 20 |

Hora 8 Diagramas de estado

- Qué es un diagrama de estados
- Sucesos, acciones y condiciones de seguridad
- Subestados: secuenciales y concurrentes
- Estados históricos
- Por qué son importantes los diagramas de estados
- Adición del diagrama de estados al panorama del UML

Al finalizar la hora anterior, dije que aquí trataría una nueva categoría de elementos con la cual no había trabajado, el elemento de comportamiento, éste muestra la forma en que las partes de un modelo UML cambian con el tiempo. Verá un miembro en particular de esta categoría, el diagrama de estados.

Esto también se aplica en cualquier sistema. Conforme el sistema interactúa con los usuarios y (posiblemente) con otros sistemas, los objetos que lo conforman pasarán por cambios necesarios para ajustar las interacciones. Si va a modelar sistemas, necesitará contar con un mecanismo para los cambios en el modelo.

Qué es un diagrama de estados

Una manera para caracterizar un cambio en un sistema es decir que los objetos que lo componen modificaron su estado como respuesta a los sucesos y al tiempo. He aquí algunos ejemplos rápidos:

Cuando acciona el interruptor, la fuente de luz cambia su estado de apagada a encendida.

Cuando presiona un botón de un control remoto, una televisión cambia su estado para mostrarle un canal u otro.

Luego de un lapso adecuado, una lavadora cambia su estado de "lavar" a "enjuagar".

El diagrama de estados UML captura este tipo de cambios. Presenta los estados en los que puede encontrarse un objeto junto con las transiciones entre los estados, y muestra los puntos inicial y final de una secuencia de cambios de estado.

Simbología

La figura 1 le muestra el rectángulo de vértices redondeados que representa a un estado, junto con una línea continua y una punta de flecha, mismas que representan a una transición. La punta de la flecha apunta hacia el estado donde se hará la transición. La figura también muestra un círculo relleno que simboliza un punto inicial y la diana que representa a un punto final.

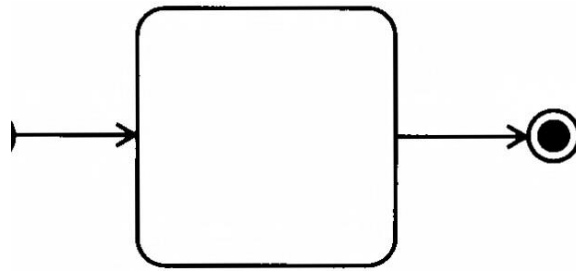


Figura 1 Símbolo de estado

Adición de detalles al icono de estado

El UML le da la opción de agregar detalles a la simbología. Así cómo es posible dividir un símbolo de clase en tres áreas (nombre, atributos y operaciones), puede dividir el icono de estado de igual forma. El área superior contendrá el nombre del estado (que tiene que establecer ya sea que haya la subdivisión o no), el área central contendrá las variables de estado, y el área inferior las actividades. La figura 2 le muestra estos detalles.

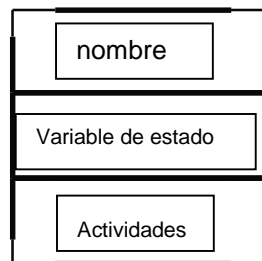


Figura 2 Subdividir el símbolo de estado

Las variables de estado como cronómetros o contadores son, en ocasiones, de ayuda. Las actividades constan de sucesos y acciones: tres de las más utilizadas son entrada (qué sucede cuando el sistema entra al estado), salida (qué sucede cuando el sistema sale del estado), y hacer (qué sucede cuando el sistema está en el estado). Puede agregar otros conforme sea necesario.

Una máquina de fax sirve como ejemplo de un objeto que puede pasar por diversas variables y actividades de estado. Cuando se envía un fax -esto es, cuando se encuentra en estado de envío de fax- la máquina de fax anota la fecha y hora en que inició el envío (los valores de las variables de estado "fecha" y "hora"), y también anota su número telefónico así como el nombre del propietario (los valores de las variables de estado "teléfono" y "propietario"). Al encontrarse en este estado, la máquina se encarga de agregar un registro de fecha y hora al fax, número telefónico y nombre del propietario. En otras actividades de este estado, la máquina jalará las hojas, paginará el fax y finalizará la transmisión.

Mientras se encuentre en el estado de inactividad, la máquina de fax mostrará la fecha y la hora en una pantalla. La figura 3 le muestra el diagrama de estados.

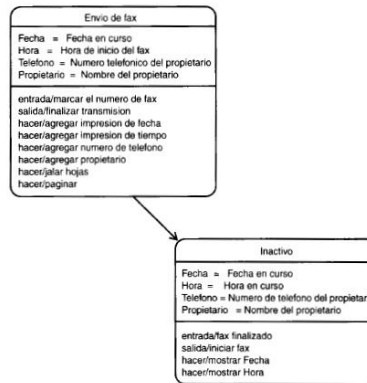


Figura 3 La máquina de fax es un buen ejemplo de un estado con variables y actividades

Sucesos y acciones

También puede agregar ciertos detalles a las líneas de transición. Puede indicar un suceso que provoque una transición (desencadenar un suceso), y la actividad de cómputo (la acción) que se ejecute y haga que suceda la modificación del estado.

A los sucesos y acciones los escribirá cerca de la línea de transición mediante una diagonal para separar un suceso desencadenado de una acción. En ocasiones un evento causará una transición sin una acción asociada, y algunas veces una transición sucederá dado que un estado finalizará una actividad (en lugar de hacerlo por un suceso). A este tipo de transición se le conoce como transición no desencadenada. La GUI (interfaz gráfica de usuario) con que interactúe le dará ejemplos de detalles de la transición. Por el momento, asumamos que la GUI puede establecerse en uno de tres estados:

- Inicialización
- Operación
- Apagar

Como resultado de las actividades en el estado de inicialización, la GUI entra al modo de Operación. Cuando desea apagar su PC, desencadena un suceso que provoca la transición hacia el estado de Apagado, y con ello la PC se apaga. La figura 8.4 muestra el diagrama de estados que captura los estados y transiciones en la GUI. La figura 4 muestra el diagrama de estados que captura los estados y transiciones en la GUI.

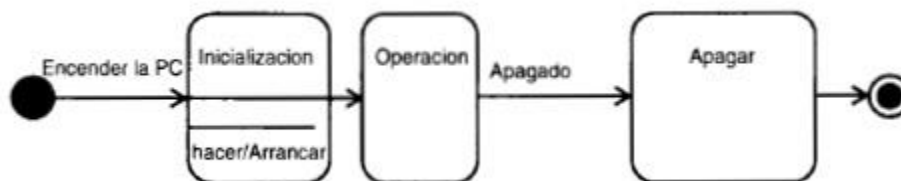


Figura 4 Los estados y transiciones de una interfaz gráfica del usuario.

Condiciones de seguridad

La anterior secuencia de estados de la GUI deja mucho que desear. Por ejemplo: si deja solo su equipo o si realizara alguna actividad en la que no tocará al ratón o al teclado, podría aparecer un protector de pantallas que evitaría el desgaste de su pantalla. Para decir esto en términos de cambio de estado, si ha pasado cierto tiempo sin que haya interacción con el usuario, la GUI hará una transición del estado.

Subestados

Cuando la GUI está en el estado Operación, hay muchas cosas que ocurren tras bambalinas, aunque no sean particularmente evidentes en la pantalla. La GUI aguarda de forma constante a que usted haga algo (oprimir una tecla, mover el ratón u oprimir uno de sus botones). Luego, deberá registrar tales acciones y modificar lo que se despliega para reflejarlas en la pantalla (como mover el cursor cuando usted mueva el ratón, o mostrar una "a" cuando usted oprima la tecla "a").

Con ello la GUI atravesará por varios cambios mientras se estado Operación. Tales cambios serán cambios de estado. Dado que estos estados se encuentran dentro de otros, se conocerán como subestados. Hay dos tipos de subestados: secuencial y concurrente.

Subestados secuenciales

Como su nombre lo indica, los subestados secuenciales suceden uno detrás de otro. Si retomamos los subestados mencionados con anterioridad dentro del estado Operación de la GUI, tendrá la siguiente secuencia:

- A la espera de acción del usuario
- Registro de una acción del usuario
- Representación de la acción del usuario

La acción del usuario desencadena la transición a partir de A la espera de acción del usuario hacia Registro de una acción del usuario. Las actividades dentro del Registro trascienden de la GUI hacia la Representación de la acción del usuario. Después del tercer estado, la GUI vuelve a iniciar a la espera de acción del usuario. La figura 5 le muestra cómo representar los subestados secuenciales dentro del estado Operación.

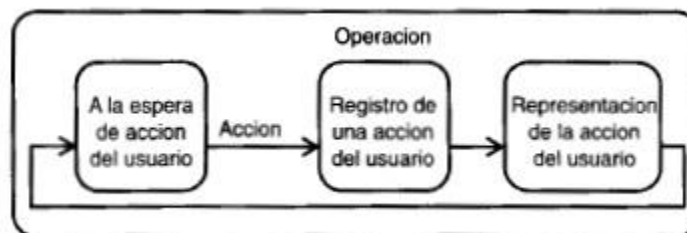


Figura 5 Subestados secuenciales

Subestados concurrentes

Dentro del estado Operación, la GUI no sólo aguarda a que usted haga algo. También verifica el cronómetro del sistema y (posiblemente) actualiza el despliegue de una aplicación luego de un intervalo específico. Por ejemplo, una aplicación podría incluir un reloj en pantalla que tuviera que actualizar la GUI.

Todo esto sucede al mismo tiempo que la secuencia que ya indiqué. Aunque cada secuencia es, claro, un conjunto de subestados secuenciales, las dos secuencias son concurrentes entre sí. Puede representar la concurrencia con una línea discontinua entre los estados concurrentes, como en la figura 6

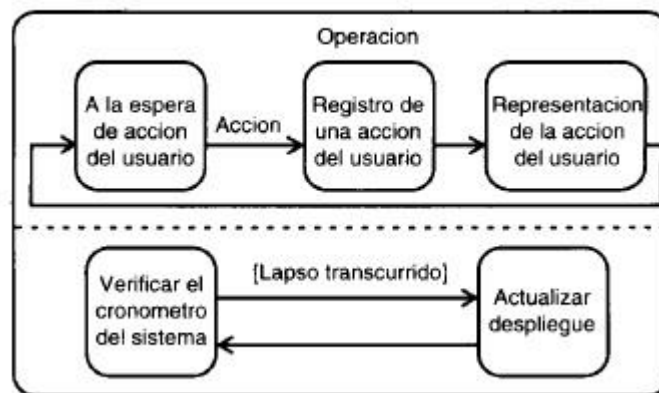


Figura 6 Los subestados concurrentes suceden al mismo tiempo.

La separación del estado Operación en dos componentes podría recordarle algo.

¿Recuerda cuando traté el tema de las adiciones y composiciones? Cuando cada componente sea parte de un "todo", tratará con una composición. Las partes concurrentes del estado Operación tienen el mismo tipo de relación con él. Por ello, el estado Operación es un estado compuesto. Un estado que consta sólo de subestados secuenciales, también es un estado compuesto.

Estados históricos

Cuando se activa su protector de pantallas y mueve su ratón para regresar al estado Operación ¿qué ocurre? ¿Acaso su pantalla retoma el estado inicial, como si apenas se hubiera encendido? ¿O lucirá tal como la dejó antes de que se activara el protector de pantallas?

Es obvio, si el protector de pantallas provocara que la pantalla regresara a su estado inicial de operación, la idea del protector de pantallas sería contraproducente. Los usuarios podrían perder su trabajo y tendrían que reiniciar una sesión nuevamente.

El diagrama de estados históricos captura esta idea. El UML proporciona un símbolo que muestra que un estado compuesto recuerda su subestado activo cuando el objeto trasciende fuera del estado compuesto. El símbolo es la letra "H" encerrada en un círculo

que se conecta por una línea continua al subestado por recordar, con una punta de flecha que apunta a tal subestado. La figura 7 muestra este símbolo en el estado Operación.

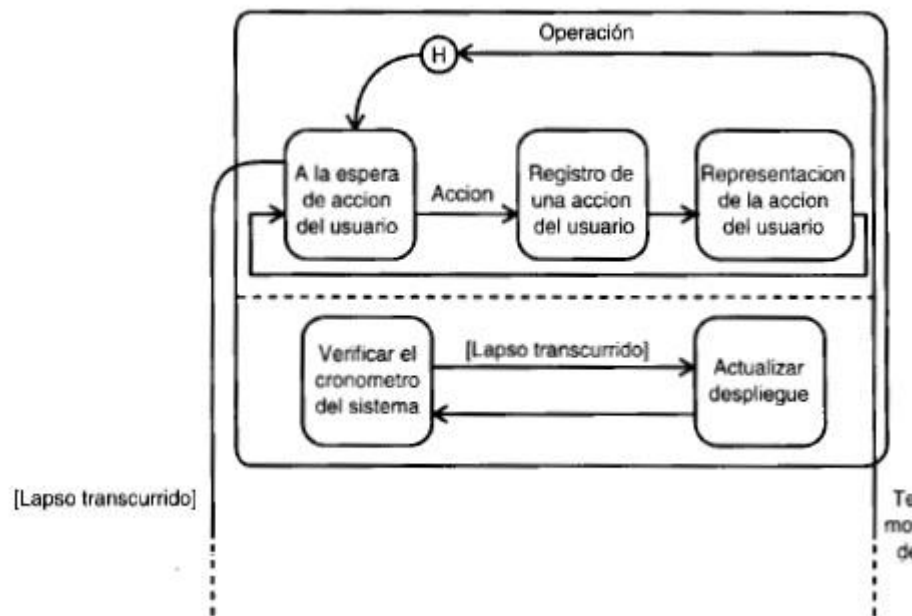


Figura 7 Un estado compuesto recuerda su subestado activo cuando el objeto trasciende fuera de tal estado compuesto.

Mensajes y señales

En el ejemplo, el suceso desencadenado que provocó la transición de Protector de pantalla a Operación es la opresión de una tecla, un movimiento del ratón o una opresión de uno de sus botones. Cualquiera de estos sucesos es, en efecto, un mensaje del usuario a la GUI. Esto es un concepto importante dado que los objetos se comunican mediante el envío de mensajes entre sí. En este caso, el suceso desencadenado es un mensaje de un objeto (el usuario) a otro (la GUI).

Por qué son importantes los diagramas de estados

El diagrama de estados del UML proporciona una gran variedad de símbolos y abarca varias ideas (todas para modelar los cambios por los que pasa un objeto). Este tipo de diagrama tiene el potencial de convertirse en algo complejo con pasmosa rapidez. ¿En verdad es necesario?

De hecho, así es. Es necesario contar con diagramas de estados dado que permiten a los analistas, diseñadores y desarrolladores comprender el comportamiento de los objetos de un sistema. Un diagrama de clases y el diagrama de objetos correspondiente sólo muestran los aspectos estáticos de un sistema. Muestran las jerarquías y asociaciones, y le indican qué son las operaciones. Pero no le muestran los detalles dinámicos de las operaciones.

Hora 9 Diagramas de secuencias

Los diagramas de estados se enfocan a los diferentes estados de un objeto.

Esto es sólo una pequeña parte del cuadro. El diagrama de secuencias del UML establece el siguiente paso y le muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo.

En esta hora se tratarán los siguientes temas:

- Qué es un diagrama de secuencias
- La GUI
- Diagramas de instancias y diagramas genéricos
- Uso de "sí" y "mientras"
- Creación de un objeto en la secuencia
- Representación de la recursividad
- Diagramas de secuencias en el panorama del UML

Los diagramas de estados que vio en la hora anterior se centran en un objeto y muestran los cambios por los que pasa dicho objeto.

Qué es un diagrama de secuencias

El diagrama de secuencias consta de objetos que se representan del modo usual:

Rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

Objetos

Los objetos se colocan cerca de la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al 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. La figura 8 muestra un objeto, su línea de vida y su activación.



Figura 8 Representación de un objeto en un diagrama de secuencias.

Mensaje

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 mensaje a sí mismo (es decir, desde su línea de vida hacia su propia línea de vida). Un mensaje puede ser simple, sincrónico, o asincrónico. Un mensaje simple es la transferencia del control de un objeto a otro. Si un objeto envía un mensaje sincrónico, esperará la respuesta a tal mensaje antes de continuar con su trabajo. Si un objeto envía un mensaje asincrónico, no esperará una respuesta antes de continuar. En el diagrama de secuencias, los símbolos del mensaje varían, por ejemplo, la punta de la flecha de un mensaje simple está formada por dos líneas, la punta de la flecha de un mensaje sincrónico está rellena y la de un asincrónico tiene una sola línea, como se aprecia en la figura 9.

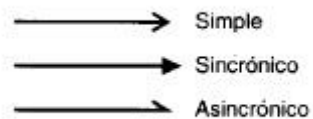


Figura 9 Símbolo para los mensajes en los diagramas de secuencias.

Tiempo

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 de 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. La figura 10 muestra al conjunto básico de símbolos del diagrama de secuencias, con los símbolos en funcionamiento conjunto. La figura muestra a un actor que inicia la secuencia, aunque, en sentido estricto, la figura adjunta no es parte del conjunto de símbolos del diagrama de secuencias.

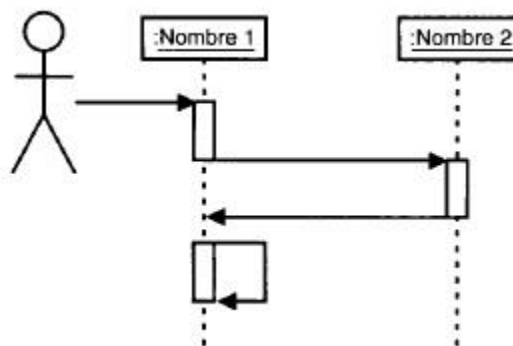


Figura 10 Conjunto básico de elementos de un diagrama de secuencias.

La GUI

En la hora anterior desarrolló los diagramas de estados que muestran los cambios por los que pasa una GUI. Ahora dibujará un diagrama de secuencias que represente las interactividades de la GUI con otros objetos.

La secuencia

Suponga que el usuario de una GUI presiona una tecla alfanumérica; si asumimos que utiliza una aplicación como un procesador de textos, el carácter correspondiente deberá aparecer de inmediato en la pantalla. ¿Qué ocurre tras bambalinas para que esto suceda?

1. La GUI notifica al sistema operativo que se oprimió una tecla.
2. El sistema operativo le notifica a la CPU.
3. El sistema operativo actualiza la GUI.
4. La CPU notifica a la tarjeta de vídeo.
5. La tarjeta de vídeo envía un mensaje al monitor.
6. El monitor presenta el carácter alfanumérico en la pantalla, con lo que se hará evidente al usuario.

El diagrama de secuencias

La figura 11 representa el diagrama de secuencias de la GUI. Como ve, los mensajes son asíncronos: ninguno de los componentes aguarda nada antes de continuar. Al trabajar con algunas aplicaciones de Windows, tal vez haya sentido algunos de los efectos de la comunicación asíncrona, particularmente en una máquina lenta. Cuando teclea en un procesador de textos, en ocasiones no ve aparecer en la pantalla el carácter correspondiente a la tecla que haya oprimido sino hasta después de haber oprimido algunas más.

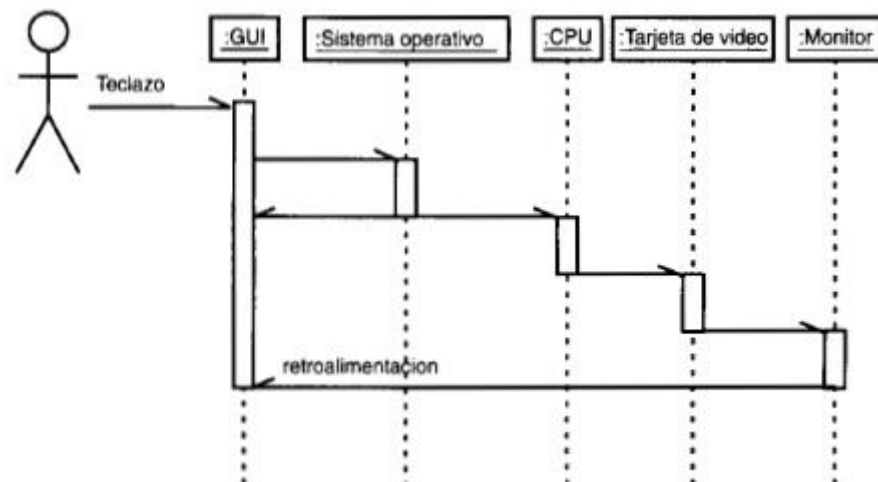


Figura 11 Diagrama de secuencias

El caso de uso

¿Qué es exactamente lo que representa un diagrama de secuencias? En este ejemplo, muestra las interacciones de objetos que se realizan durante un escenario sencillo: la opresión de una tecla. Este escenario podría ser parte de un caso de uso llamado "Ejecutar la opresión de una tecla" (vea la figura 12). Al representar gráficamente las interacciones

del sistema en el caso de uso, el diagrama de secuencias habrá, en efecto, "delineado" el caso de uso dentro del sistema.



Figura 12 Diagrama de caso de uso

Instancias y genéricos

El ejemplo anterior comenzó con un diagrama de estados. Este otro ejemplo empieza con un caso de uso. "Comprar gaseosa" fue uno de los casos de uso del ejemplo de la máquina de gaseosas en las horas 6, "Introducción a los casos de uso", y 7, "Diagramas de casos de uso".

Un diagrama de secuencias de instancias

En el mejor escenario del caso de uso "Comprar gaseosa", recuerde que el actor es un cliente que desea adquirir una lata de gaseosa. El cliente inicia el escenario mediante la inserción de dinero en la máquina. Luego hace una selección. Dado que hablamos del mejor escenario, la máquina tiene al menos una lata de la gaseosa elegida y por lo tanto presenta una lata fría al cliente.

Asumamos que en la máquina de gaseosas hay tres objetos que realizan la tarea que nos ocupa: la fachada (la interfaz que la máquina de gaseosas presenta al usuario), el registrador de dinero (que lo recolecta), y el dispensador (que entrega la gaseosa). También daremos por hecho que el registrador de dinero controlará al dispensador. La secuencia será como sigue:

1. El cliente inserta el dinero en la alcancía que se encuentra en la fachada de la máquina.
2. El cliente hace su elección.
3. El dinero viaja hacia el registrador.
4. El registrador verifica si la gaseosa elegida está en el dispensador.
5. Dado que es el mejor escenario, asumimos que sí hay gaseosas, y el registrador actualiza su reserva de efectivo.
6. El registrador hace que el dispensador entregue la gaseosa en la fachada de la máquina.

Un diagrama de secuencias genérico

Como recordará, el caso de uso "Comprar gaseosa" tenía dos escenarios alternos. Uno de ellos se refería al hecho de que la máquina no tuviera la gaseosa seleccionada y el otro cuando el cliente no contaba con el dinero exacto. Si tomara en cuenta todos los escenarios

de un caso de uso al momento de crear un diagrama de secuencias, se trataría de un diagrama de secuencias genérico.

Para representar cada condición en la secuencia, tal condición la colocará en un "si" (un si condicional) entre corchetes. Arriba de las flechas de mensaje apropiadas, agregue [alimentación > precio], [alimentación - precio no presente] y [alimentación - precio presente].

Cada condición causará una bifurcación del control en el mensaje, que separará al mensaje en rutas distintas. Como cada ruta irá al mismo objeto, la bifurcación causará una "ramificación" del control en la línea de vida del objeto receptor, y separará las líneas de vida en rutas distintas. En algún lugar de la secuencia, las ramas del mensaje confluirán, como las bifurcaciones en las líneas de vida.

Creación de un objeto en la secuencia

En los ejemplos que hemos visto ha analizado distintos tipos de mensajes, diagramas de secuencias genérico y de instancias, así como estructuras de control. Otro concepto importante relacionado con los diagramas de secuencias, particularmente cuando diseñe software, es la creación de objetos.

Con frecuencia se da el caso de que un programa orientado a objetos debe crear un objeto. Recuerde que en términos del software, una clase es una plantilla para crear un objeto (como un molde de galletas para crear una galleta). ¿Cómo representaría la creación de un objeto cuando represente una secuencia de interacciones entre objetos?

Cómo representar la recursividad

En ocasiones un objeto cuenta con 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.

He aquí un ejemplo. Suponga que uno de los objetos en su sistema sea una calculadora, y que una de sus operaciones sea el cálculo de intereses. Para calcular el interés compuesto para un periodo que incluya a varios periodos, la operación del cálculo de intereses del objeto tendrá que invocarse a sí misma varias veces. Para representar esto en el UML, dibujará una flecha de mensaje fuera de la activación que signifique la operación, y un pequeño rectángulo sobrepuesto en la activación. Dibuje una flecha de modo que apunte al pequeño rectángulo, y una que regresa al objeto que inició la recursividad. La figura 13 muestra lo anterior.

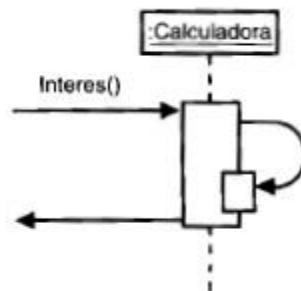


Figura 13 Representación de la recursividad en un diagrama de secuencias.

Adiciones al panorama

Ahora podrá agregar otro diagrama a su panorama del UML. Dado que se refiere al comportamiento de los objetos, el diagrama de secuencias iría bajo la categoría "Elementos de comportamiento". La figura 14 actualiza su creciente panorama.

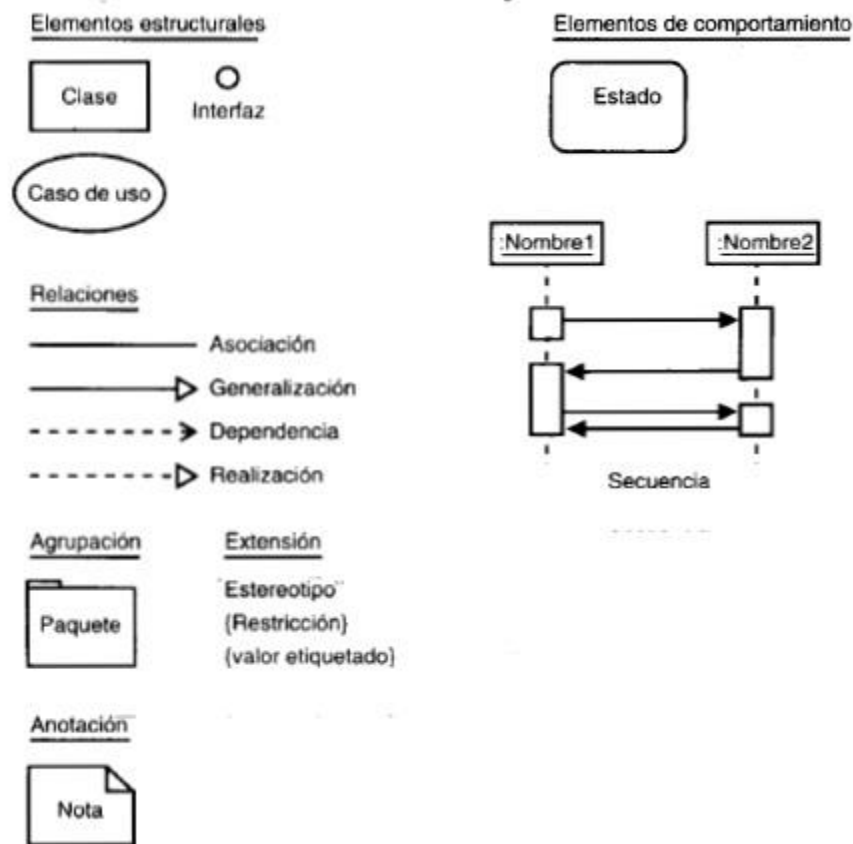


Figura 14 El panorama del UML

Hora 10 Diagramas de colaboraciones

Ahora veremos lo correspondiente a un diagrama que es similar al que vimos en la hora anterior. Este también muestra la colaboración entre los objetos, pero de una forma significativamente diferente del diagrama de secuencias

En esta hora se tratarán los siguientes temas:

- Qué es un diagrama de colaboraciones
- Cómo aplicar un diagrama de colaboraciones
- Uso de "si" y "mientras"
- Anidamiento
- Objetos activos y concurrencia
- Sincronización
- Dónde encajan los diagramas de colaboraciones en el UML

Los diagramas de colaboraciones muestran la forma en que los objetos colaboran entre sí, tal como sucede con un diagrama de secuencias. Muestran los objetos junto con los mensajes que se envían entre ellos. Si el diagrama de secuencias hace eso, ¿por qué el UML requeriría otro diagrama?, ¿qué no son lo mismo?, ¿no es una pérdida de tiempo? Ambos tipos de diagrama son similares. De hecho, son semánticamente equivalentes. Esto significa que representan la misma información, y podrá convertir un diagrama de secuencias en un diagrama de colaboraciones equivalente y viceversa.

Qué es un diagrama de colaboraciones

Un diagrama de objetos muestra a los objetos como tales y sus relaciones entre sí. Un diagrama de colaboraciones es una extensión de uno de objetos. Además de las relaciones entre objetos, el diagrama de colaboraciones muestra los mensajes que se envían los objetos entre sí. Por lo general, evitará la multiplicidad dado que podría ser fuente de confusión.

Para representar un mensaje, dibujará una flecha cerca de la línea de asociación entre dos objetos, esta flecha apunta al objeto receptor. El tipo de mensaje se mostrará en una etiqueta cerca de la flecha; por lo general, el mensaje le indicará al objeto receptor que ejecute una de sus operaciones. El mensaje finalizará con un par de paréntesis, dentro de los cuales colocará los parámetros (en caso de haber alguno) con los que funcionará la operación.

Mencioné que podrá convertir cualquier diagrama de secuencias en diagrama de colaboraciones y viceversa. Por medio de esto podrá representar la información de secuencia en un diagrama de colaboraciones. Para ello, agregará una cifra a la etiqueta de un mensaje, misma que corresponderá a la secuencia propia del mensaje. La cifra y el mensaje se separan mediante dos puntos (:).

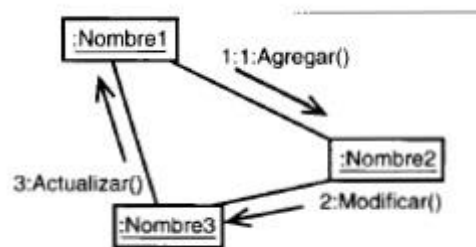


Figura 15 Simbología del diagrama de colaboración.

La GUI

Este ejemplo es el caso más directo. Un actor inicia la secuencia de interacción al oprimir una tecla, con lo que los mensajes ocurrirán de manera secuencial. Tal secuencia (a partir de la hora anterior) es:

1. La GUI notifica al sistema operativo que se oprimió una tecla.
2. El sistema operativo le notifica a la CPU.
3. El sistema operativo actualiza la GUI.

4. La CPU notifica a la tarjeta de vídeo.
5. La tarjeta de vídeo envía un mensaje al monitor.
6. El monitor presenta el carácter alfanumérico en la pantalla, con lo que se hará evidente al usuario.

La figura 16 muestra la forma de representar esta secuencia de interacciones en un diagrama de colaboraciones. El diagrama muestra la figura agregada que representa al usuario que inicia la secuencia, aunque esta figura no es parte de la simbología de este diagrama.

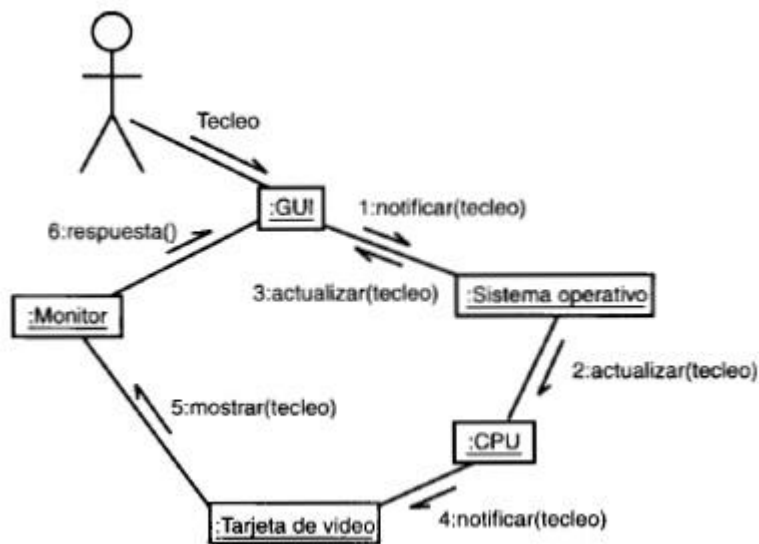


Figura 16 Diagrama de ejemplo para la GUI

Cambios de estado

Puede mostrar los cambios de estado en un objeto en un diagrama de colaboraciones.

En el rectángulo del objeto indique su estado. Agregue otro rectángulo al diagrama que haga las veces del objeto e indique el estado modificado. Conecte a los dos con una línea discontinua y etiquete la línea con un estereotipo «se toma».

La figura 17 ilustra un cambio de estado para la GUI, que muestra que el estado de inicialización se convierte en el estado operativo.

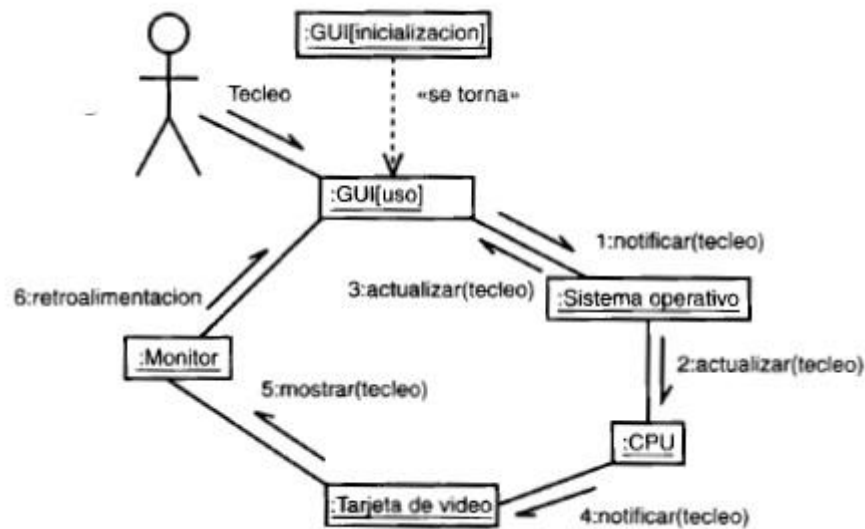


Figura 17 Un diagrama de colaboración puede incluir cambios de estado.

Creación de un objeto

Para mostrar la creación de objetos, volveré al caso de uso "Crear propuesta" de la firma de consultoría. Una vez más, la secuencia que modelará será:

1. El consultor buscará en el área de almacenamiento centralizada de la red una propuesta adecuada en la cual basarse.
2. Si el consultor localiza una propuesta adecuada, la abrirá y en el proceso abrirá la aplicación de oficina. El consultor guardará el archivo bajo un nuevo nombre, con lo que creará un nuevo archivo para la nueva propuesta.
3. Si el consultor no encuentra una propuesta, abrirá la aplicación de oficina y generará un nuevo archivo.
4. Al trabajar en la propuesta, el consultor utilizará los componentes de la aplicación de oficina.
5. Cuando el usuario finalice la propuesta, la guardará en el área de almacenamiento centralizada.

Para mostrar la creación de un objeto, agregaré un estereotipo "crear" al mensaje que genera al objeto. Una vez más, utilizaré instrucciones "si" (if) y mensaje anidados. También trabajará con un ciclo "mientras" (while). Como en el diagrama de secuencias, para representar a "mientras", colocará esta condición entre corchetes y antecederá al del lado izquierdo con un asterisco.

La figura 18 le muestra este diagrama de colaboraciones completo con la creación del objeto y "mientras".

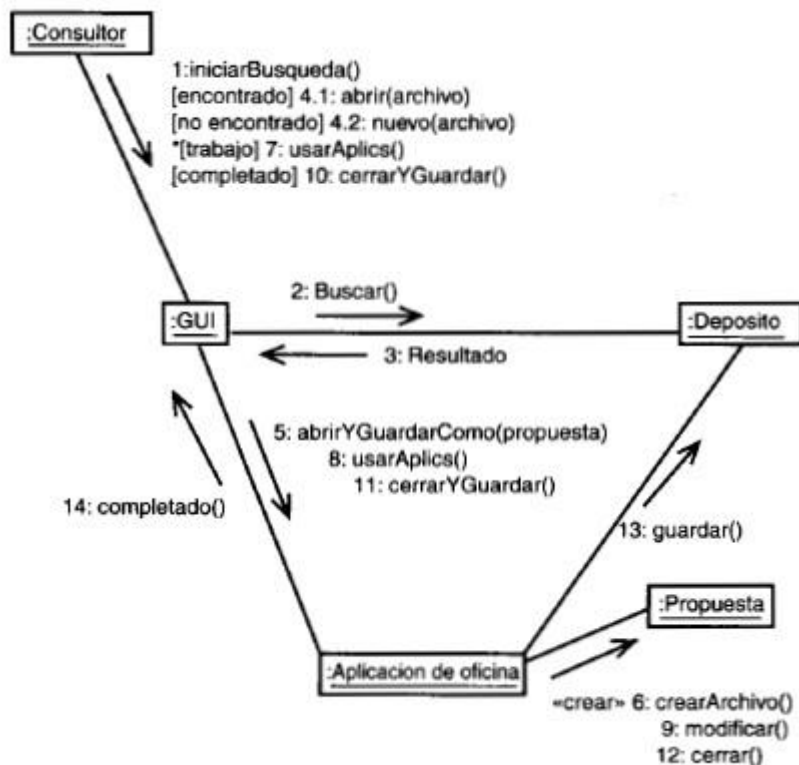


Figura 18 Diagrama de colaboraciones "crear una propuesta"

Algunos conceptos más

Aunque ha visto algunas bases, no ha visto todo lo relacionado con los diagramas de colaboraciones. Los conceptos en esta sección son un poco esotéricos, pero podrían serle útiles en sus esfuerzos para analizar sistemas.

Varios objetos receptores en una clase

En ocasiones un objeto envía un mensaje a diversos objetos de la misma clase. Por ejemplo:

Un profesor le pide a un grupo de estudiantes que entreguen una tarea. En el diagrama de colaboraciones, la representación de los diversos objetos es una pila de rectángulos que se extienden "desde atrás". Agregaré una condición entre corchetes precedida por un asterisco para indicar que el mensaje irá a todos los objetos. La figura 19 le muestra los detalles.



Figura 19 Un objeto que envía un mensaje a diversos objetos de una clase.

Representación de los resultados

Un mensaje podría ser una petición a un objeto para que realice un cálculo y devuelva un valor. Un objeto Cliente podría solicitar a un objeto Calculadora que calcule el precio total que sea la suma del precio de un elemento y el impuesto.

Objetos activos

En algunas interacciones, un objeto específico controla el flujo. Este objeto activo puede enviar mensajes a los objetos pasivos e interactuar con otros objetos activos. En una biblioteca, un bibliotecario relaciona las peticiones a partir de un patrón, verifica la información de referencia en una base de datos, devuelve una respuesta al petionario, asigna personas para reabastecer los libros, entre otras cosas. Un bibliotecario también interactúa con otros que realicen las mismas operaciones. Al proceso de que dos o más objetos activos hagan sus tareas al mismo tiempo, se le conoce como concurrencia.

Sincronización

Otro caso con el que se puede encontrar es que un objeto sólo puede enviar un mensaje después de que otros mensajes han sido enviados. Es decir, el objeto debe "sincronizar" todos los mensajes en el orden debido.

Un ejemplo aclarará esto. Suponga que sus objetos son personas en un corporativo, y que están ocupados en la campaña de un nuevo producto. He aquí la secuencia de interacciones:

1. El vicepresidente de comercialización le pide al de ventas que cree una campaña para un producto en particular.
2. El vicepresidente de ventas crea la campaña y la asigna al gerente de ventas.
3. El gerente de ventas instruye a un agente de ventas para que venda el producto de acuerdo con la campaña.
4. El agente de ventas hace llamadas para vender el producto a los clientes en potencia.
5. Luego de que el vicepresidente de ventas ha dado la comisión y el gerente de ventas ha expedido la directiva (esto es, cuando se han completado los pasos 2 y 3), un especialista en relaciones públicas de la corporación hará una llamada al periódico local y colocará un anuncio de la campaña.

¿Cómo representará la posición del paso cinco en la secuencia? Nuevamente, el UML le da una sintaxis. En lugar de anteceder este mensaje con una etiqueta numérica, lo antecederá con una lista de mensajes que tendrán que completarse antes de que se realice el paso cinco. La lista de elementos se separará mediante una coma, y finalizará con una diagonal. La figura 20 le muestra el diagrama de colaboraciones en este ejemplo.

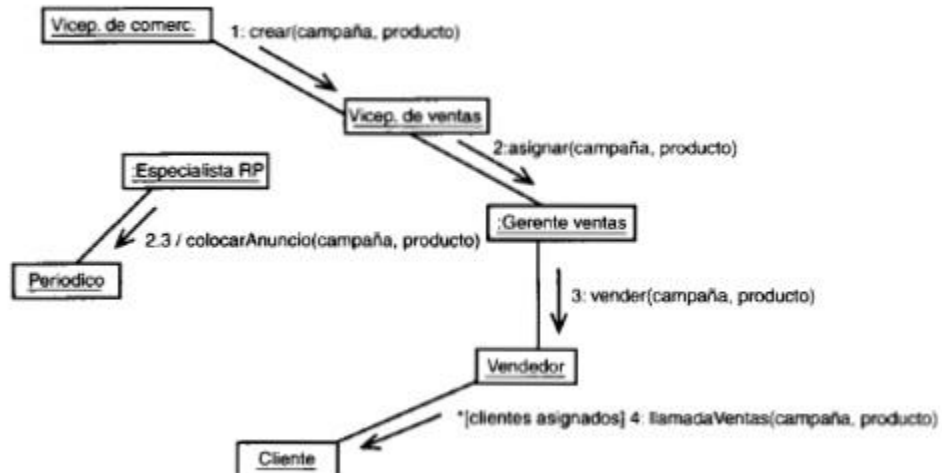


Figura 20 La sincronización de mensajes.

Adiciones al panorama

Ahora podrá agregar el diagrama de colaboraciones a su panorama del UML. Es otro elemento de comportamiento, como se aprecia en la figura 21.

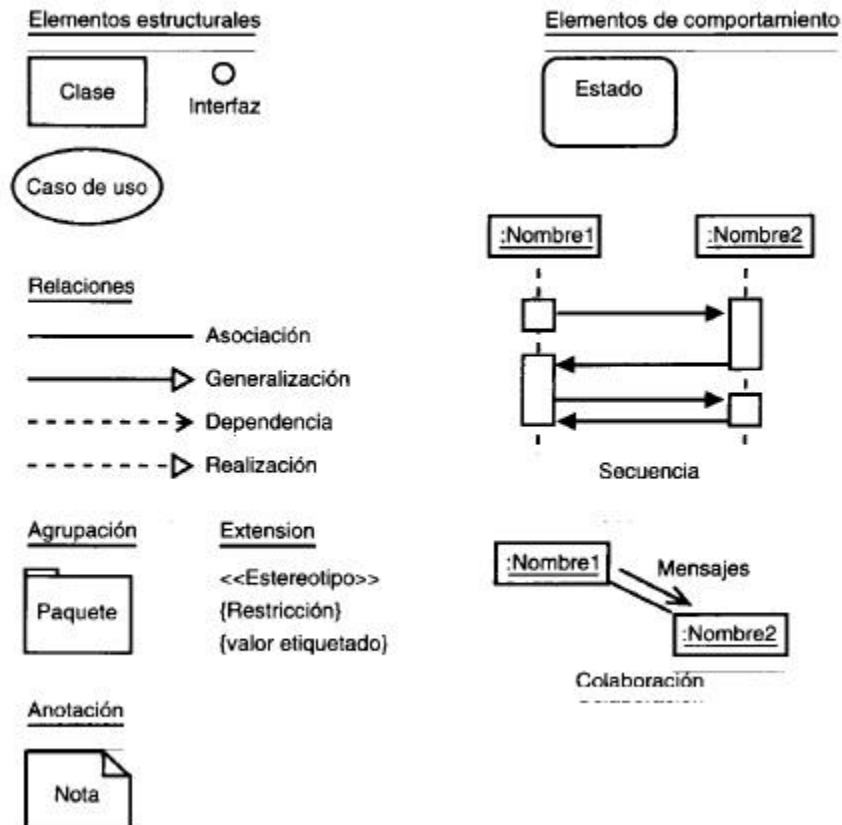


Figura 21 El panorama del UML.

Bibliografía

SCHMULLER, J. (2000). *APRENDIENDO UML EN 24 HORAS*. Mexico D.F.: Prentice-Hall.