



Ingeniería de software I

SHP

Diagramas de Estados

Diagramas de colaboración

Diagramas de secuencia

Sosa Rincón Daniel Isaac

Matrícula:

7331

Índice

Índice de figuras	3
Diagramas de estados.....	4
¿Qué es un diagrama de estado?	4
Simbología	4
Adición de detalles al icono de estado.....	4
Sucesos y acciones	5
Condiciones de seguridad	6
Subestados (secuenciales y concurrentes)	6
Estados históricos.....	7
Mensajes y señales.....	8
Por qué son importantes los diagramas de estado.....	8
Diagramas de colaboraciones	8
¿Qué es un diagrama de colaboraciones?	8
Cambios de estado	10
La máquina de gaseosas.....	10
Creación de un objeto	11
Varios objetos receptores de una clase	11
Representación de los resultados	12
Objetos activos.....	13
Sincronización	13
Adiciones al panorama	14
Diagramas de secuencia.....	15
¿Qué es un diagrama de secuencia?	15
Objetos	15
Mensaje.....	15
Tiempo	16
LA GUI.....	16
La secuencia.	16
El diagrama de secuencia.	17
El caso de uso.	17
Instancias y genéricos.	18

Un diagrama de secuencia de instancia	18
Un diagrama de secuencia genérico.	19
Escenario “Monto incorrecto”.	19
Escenario sin gaseosa.....	20
Creación de un objeto en la secuencia.	20
Como representar la recursividad.....	22
Bibliografía.....	23

Índice de figuras

Figura 1	4
Figura 2	4
Figura 3	5
Figura 4	6
Figura 5	6
Figura 6	7
Figura 7	7
Figura 8	9
Figura 9	9
Figura 10	10
Figura 11	10
Figura 12	11
Figura 13	11
Figura 14	12
Figura 15	12
Figura 16	13
Figura 17	13
Figura 18	14
Figura 19	15
Figura 20	16
Figura 21	16
Figura 22	17
Figura 23	17
Figura 24	18
Figura 25	18
Figura 26	19
Figura 27	20
Figura 28	21
Figura 29	22

Diagramas de estados

¿Qué es un diagrama de estado?

Sirven para modelar la vida de un objeto muestra el flujo de control entre estados (en qué estados posibles puede estar y como se producen los cambios entre dichos 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. En los diagramas de estado, como su nombre lo dice se presentan 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

Los símbolos UML en un diagrama de estados, el icono para el estado es un rectángulo de vértices redondeados, y el símbolo de una línea continua y una punta de flecha. Un círculo relleno se interpreta como el punto inicial de una secuencia de estados, y una diana que representa al punto final

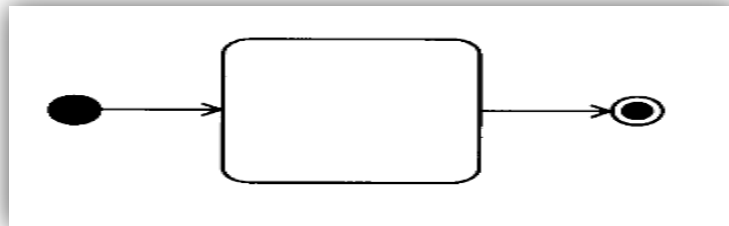


Figura 1

Adición de detalles al icono de estado

En UML tenemos la opción de agregar detalles a la simbología. Así como es posible dividir un símbolo en tres áreas (nombre, atributos y operaciones), puede dividir el símbolo de estado en áreas que muestren el nombre, variables y actividades del estado.

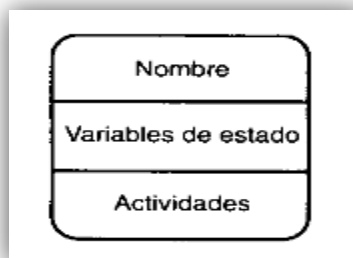


Figura 2

Las actividades constan de sucesos y acciones tres de las principales o más usadas son:

- Entrada: sucede cuando el sistema entra en el estado.
- Salida: sucede cuando el sistema sale del estado.
- Hacer: sucede cuando el sistema está en el estado.

Sucesos y acciones

También se puede indicar un suceso que provoque una transición (desencadenar un suceso), y la actividad de computo (la acción) que se ejecute y haga que suceda la modificación del estado. En ocasiones un evento causara 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.

Ejemplo.

El diagrama de estados de la interfaz gráfica de usuario de Windows XP, asume tres estados:

- Inicialización
- Operación
- Apagar

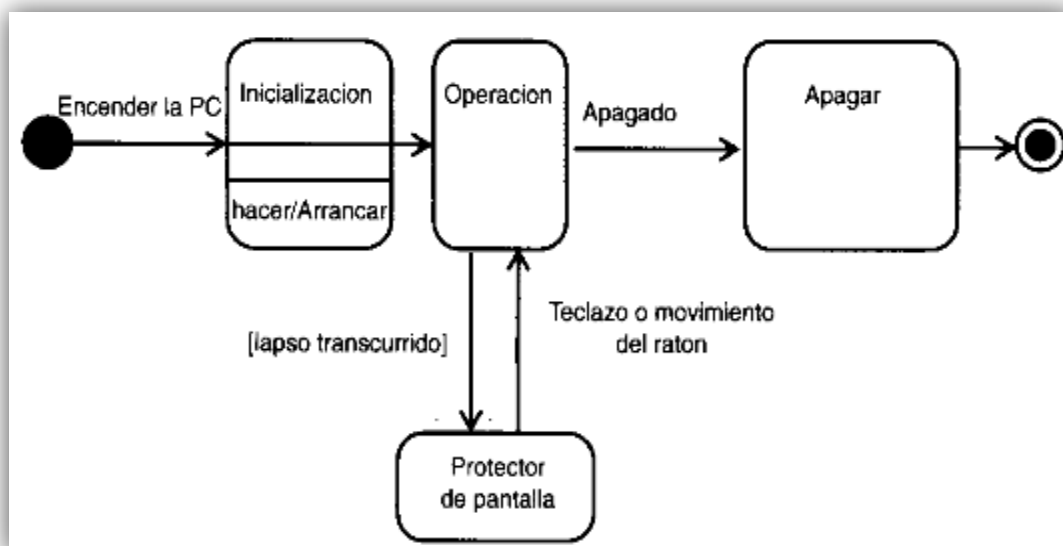


Figura 3

Los estados y transiciones de una interfaz gráfica de usuario incluyen el desencadenamiento de eventos, acciones y transiciones no desencadenadas.

Condiciones de seguridad

Si se deja solo el equipo, sin mover el mouse o el teclado, podría aparecer un protector de pantalla. Supongamos que está configurado en un intervalo de 15 minutos:

- El intervalo de 15 minutos es una condición de seguridad, cuando se cumple una condición, se realiza la transición.

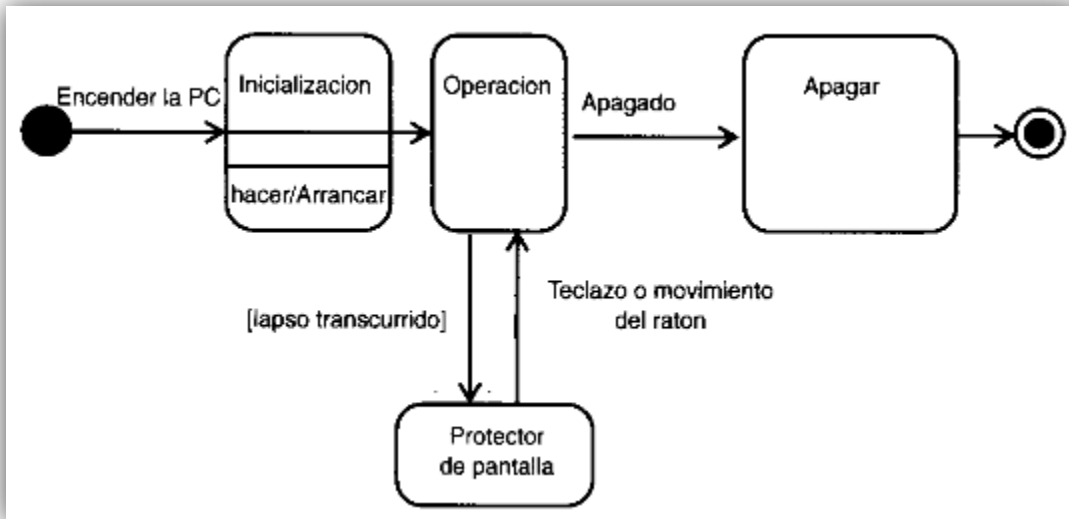


Figura 4

Subestados (secuenciales y concurrentes)

Un estado que consta de Subestados se conoce como estado compuesto. En ocasiones un estado consta de Subestados, esto es, que un estado se encuentra dentro de otros, estos pueden ser:

- Secuenciales. Ocurre unos después del otro.

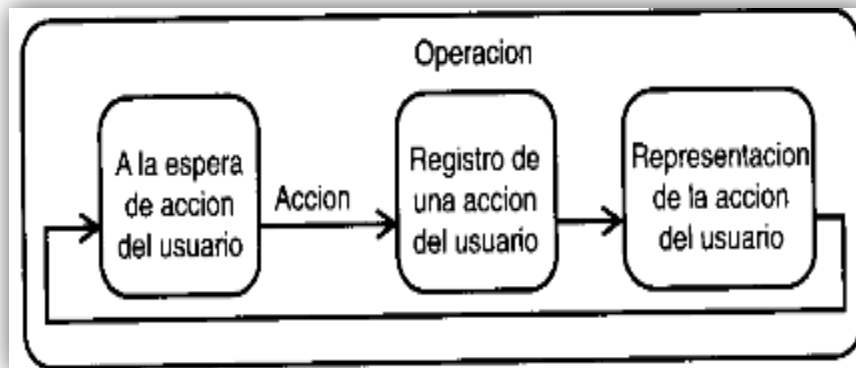


Figura 5

- Concurrentes. Ocurre al mismo tiempo.

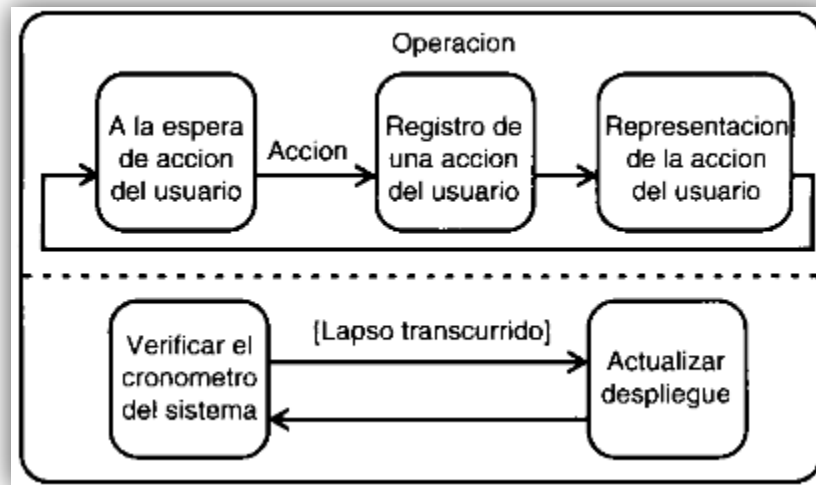


Figura 6

Estados históricos

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.

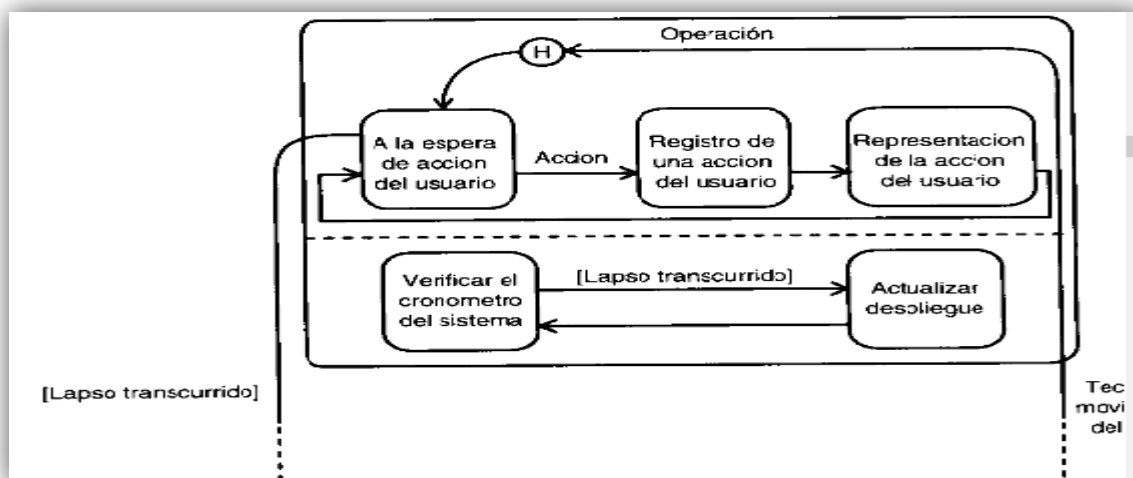


Figura 7

Cuando un estado histórico recuerda los Subestados en todos los niveles de anidación, el estado histórico es profundo. Si solo recuerda el subestado principal, el estado histórico

será superficial. Un estado histórico profundo se representa agregando un asterisco (*) a la "H" en el círculo (H*).

El estado histórico y el estado inicial (representado por el círculo relleno) son conocidos como pseudoestados. No tienen variables de estados ni actividades, por lo que no son estados completos.

Mensajes y señales

Un mensaje que desencadena una transición en el diagrama de estados del objeto receptor se le conoce como señal. En orientación a objetos el envío de una señal es lo mismo que crear un objeto señal y transmitirlo al objeto receptor. El objeto señal cuenta con propiedades que se representan como atributos. Dado que una señal es un objeto, es posible crear jerarquías de herencia de señales.

Por qué son importantes los diagramas de estado

- Los diagramas de estado proporcionan una gran variedad de símbolos y abarca varias ideas (todas para modelar los cambios por los que pasa un objeto).
- Permiten a los analistas, diseñadores y desarrolladores comprender el comportamiento de los objetos de un sistema.
- Muestran las jerarquías y asociaciones.
- Indican que son las operaciones.
- Muestran una clara representación del comportamiento del objeto.
- Aumenta la probabilidad de que el equipo de desarrollo produzca un sistema que cumpla con los requerimientos.
-

Diagramas de colaboraciones

Los diagramas de colaboraciones muestran la forma en que los objetos colaboran entre sí.

¿Qué es un diagrama de colaboraciones?

Es una extensión de un objeto, además 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, dibujara 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.

Simbología del diagrama de secuencia:

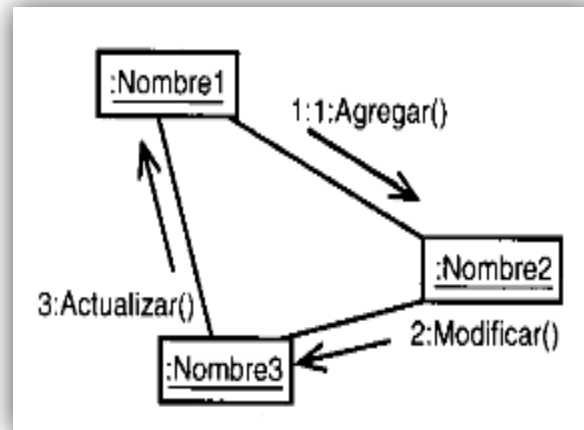


Figura 8

Ejemplo de la interfaz gráfica de usuario:

Un actor inicia la secuencia de interacción al oprimir una tecla, con lo que los mensajes ocurrirán de manera secuencial. 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.

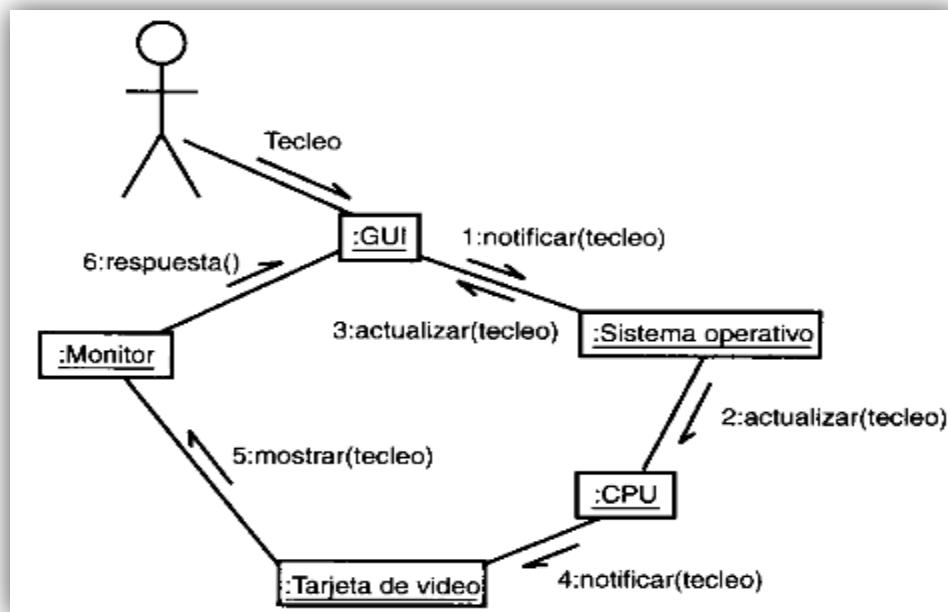


Figura 9

Cambios de estado

Puede mostrar los cambios de estado en un objeto en un diagrama de colaboraciones. En el rectángulo del objeto se indica su estado. Se agrega otro rectángulo al diagrama que haga las veces del objeto e indique el estado modificado. Se conecta a los dos con una línea discontinua y etiqueta la línea con un estereotipo << se torna >>.

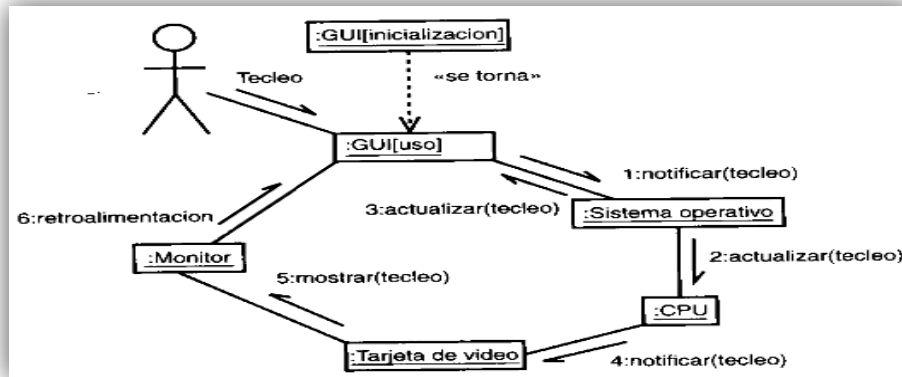


Figura 10

La máquina de gaseosas

En este caso se aplican condiciones a una situación real “comprar gaseosa” don la secuencia es:

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 la mejor situación, asumimos que si 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.

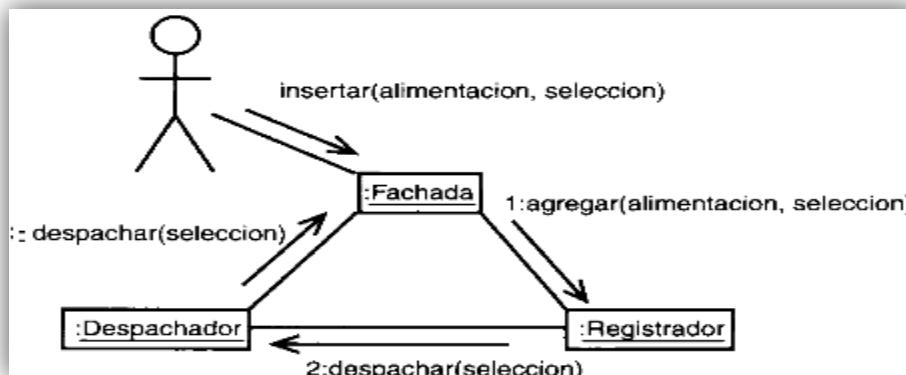


Figura 11

Creación de un objeto

Para mostrar la creación de objetos, se agregará un estereotipo “crear” al mensaje que genera al objeto. Se utilizará la instrucción “si” (if) y mensaje anidados. También se trabajará con un ciclo “mientras” (while). En el diagrama de secuencia, para representar a “mientras”, se coloca esta condición entre corchetes y antecederá al del lado izquierdo con un asterisco.

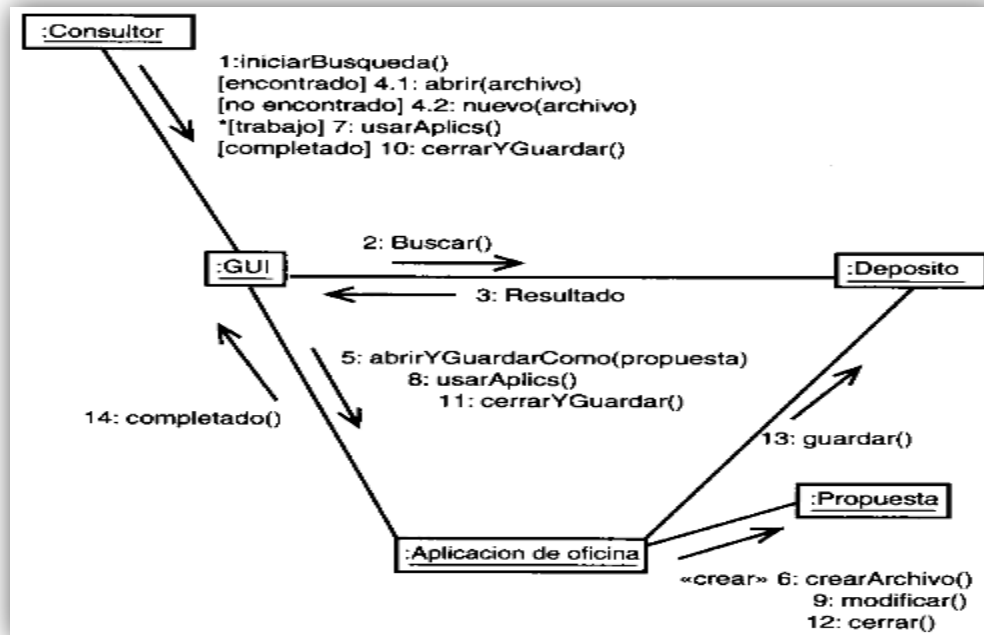


Figura 12

Varios objetos receptores de 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”. Se agrega una condición entre corchetes precedida por un asterisco para indicar que el mensaje irá a todos los objetos.

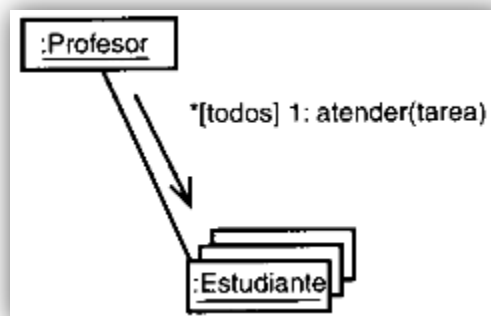


Figura 13

En algunos casos, el orden del mensaje enviado es importante. Por ejemplo, un empleado bancario dará servicio a cada cliente conforme van llegando a la fila. Esto se representa con un “mientras”.

Objeto que envía un mensaje a varios otros en orden específico.

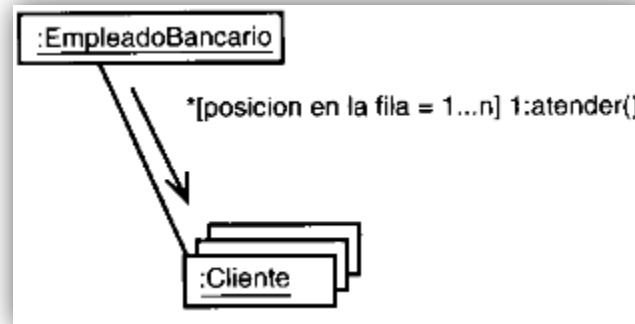


Figura 14

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. En UML se deberá escribir una expresión que tenga el nombre del valor devuelto a la izquierda, seguido de “=”, a continuación el nombre de la operación y las cantidades con que opera para producir el resultado.

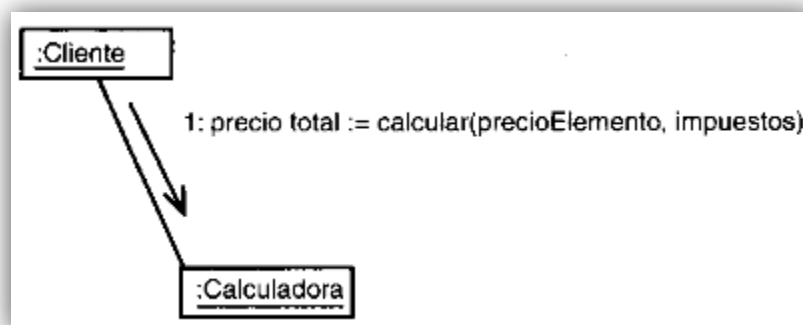


Figura 15

A la parte que está a la derecha de la expresión se le conoce como firma del mensaje.

Objetos activos

En algunas iteraciones, un objeto específico controla el flujo. Este objeto activo puede enviar mensajes a los objetos pasivos e interactuar con otros objetos activos. Al proceso de que dos o más objetos activos hagan sus tareas al mismo tiempo, se le conoce como concurrencia. El diagrama de colaboraciones representa a un objeto activo de la misma manera que a cualquier otro objeto, excepto que su borde será grueso y más oscuro.

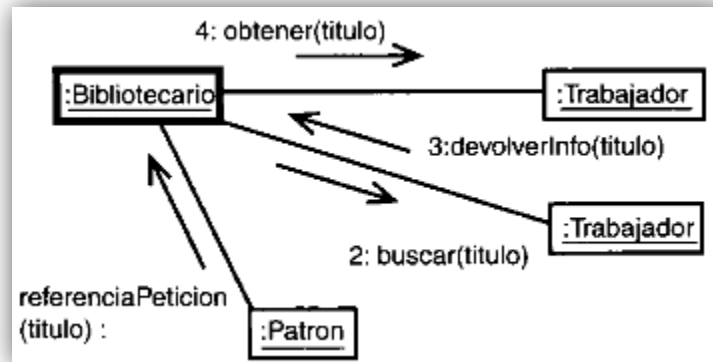


Figura 16

Sincronización

Otro caso que se puede presentar es que un objeto solo 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. La sincronización de mensajes en un diagrama de colaboraciones se representa como se muestra en la figura.

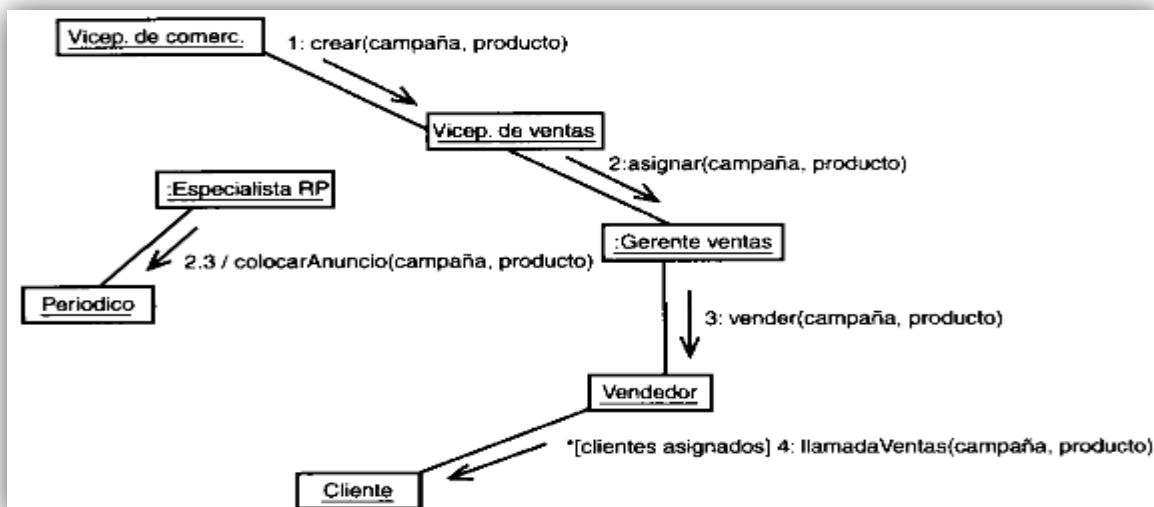


Figura 17

Adiciones al panorama

Se puede agregar el diagrama de colaboraciones a su panorama del UML. Es otro elemento de comportamiento.

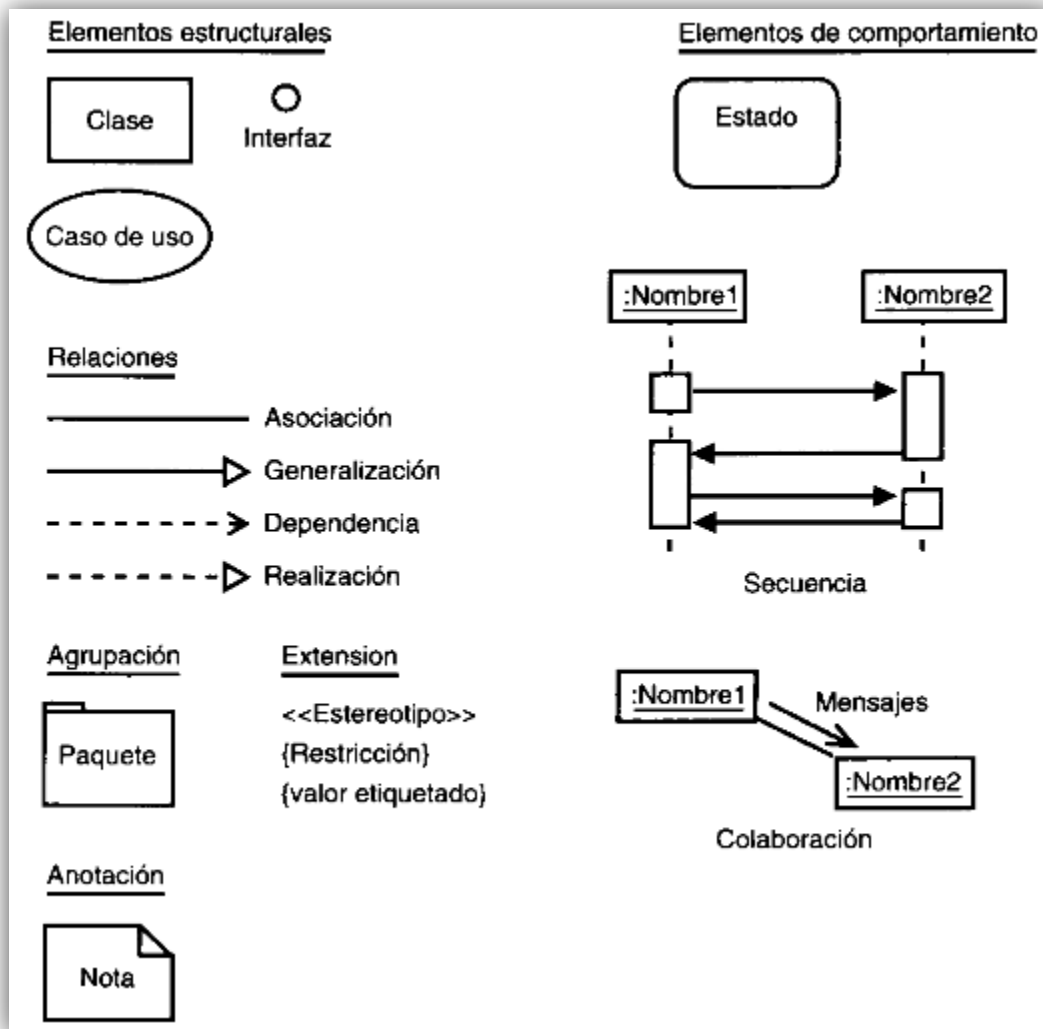


Figura 18

Diagramas de secuencia

Los diagramas de secuencia UML establece el siguiente paso y muestra cómo se comunican los objetos entre si al transcurrir el tiempo.

El UML te permite expandir tu campo de visión y le muestra la forma en que un objeto interacciona con otro. En el campo de visión expandido, incluye una importante dimensión: tiempo. La idea primordial es que las interacciones con los objetos realizan una secuencia y esta lleva su tiempo desde el inicio hasta el fin.

¿Qué es un diagrama de secuencia?

Consta de objetos que se representan de forma usual: nombre en rectángulos (subrayado), mensajes representados con líneas continuas con flecha y el tiempo como una progresión de forma vertical.

Objetos

Se colocan cerca de la parte superior del diagrama de izquierda a derecha y se acomoda de manera que se simplifique el diagrama. La extensión que se encuentra debajo de estos será una línea discontinua y es conocida como la línea de la vida de un objeto. Junto a esta línea se encuentra un rectángulo pequeño que es conocido como activación, la cual representa la ejecución de un objeto.

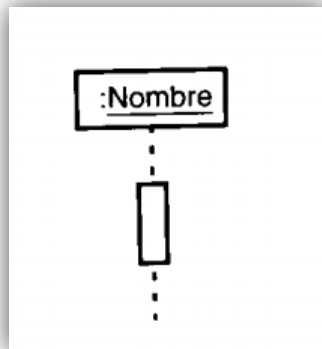


Figura 19

Mensaje

Este pasa por la línea de vida de objeto a la de otro. Un objeto se puede mandar un mensaje así mismo.

Puede ser:

- *Simple*. Transferencia del control de un objeto a otro.
- *Sincrónico*. Espera la respuesta del mensaje para continuar.
- *Asincrónico*. No espera por la respuesta para continuar.

Los símbolos del mensaje varían.

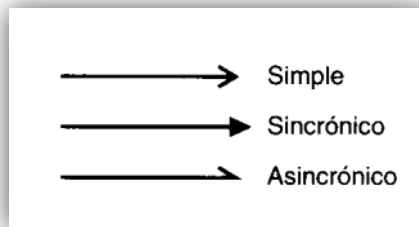


Figura 20

Tiempo

Se representan en forma vertical. El tiempo se inicia en la parte superior avanza hacia la inferior.

Tiene dos dimensiones:

- *Horizontal*. Disposición de los objetos.
- *Vertical*. Muestra el paso del tiempo.

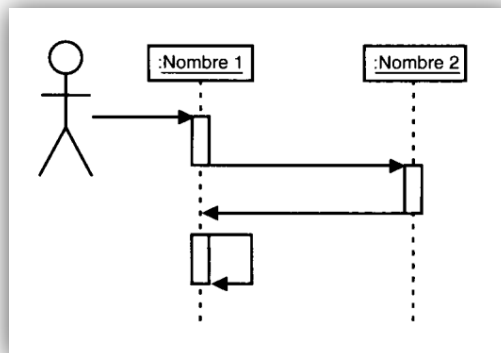


Figura 21

LA GUI

La secuencia.

Suponga que un usuario presiona una tecla alfanumérica; si asumimos que utiliza una aplicación como procesador de datos, el carácter tendrá que aparecer de inmediato. ¿Qué sucede detrás de todo eso?

1. La GUI notificara al sistema que se oprimió una tecla.
2. El SO notificara al CPU.
3. El SO actualiza la GUI.
4. La CPU notifica a la tarjeta de video.

5. La tarjeta de video envía un mensaje al monitor.
6. El monitor presenta el carácter en pantalla para ser visible al usuario.

El diagrama de secuencia.

La figura 9.4 representa un diagrama de secuencia de la GUI. Los mensajes son asíncronos. Al trabajar con Windows, tal vez haya sentido algunos efectos de este tipo de comunicación, en especial si la maquina es lenta.

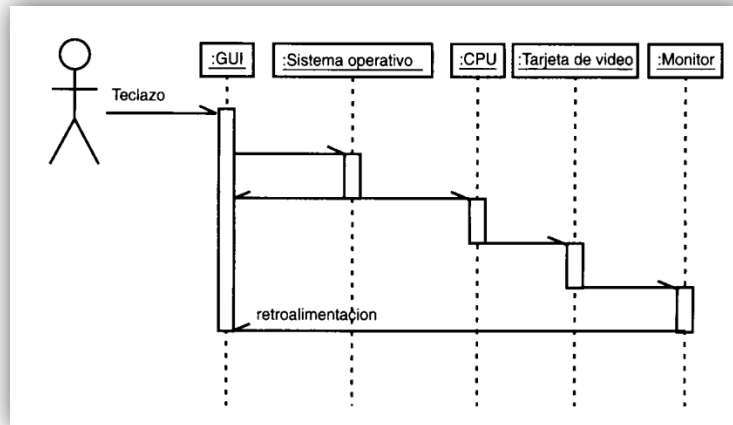


Figura 22

Es muy instructivo mostrar los datos de un uno o varios objetos en el diagrama de secuencia.

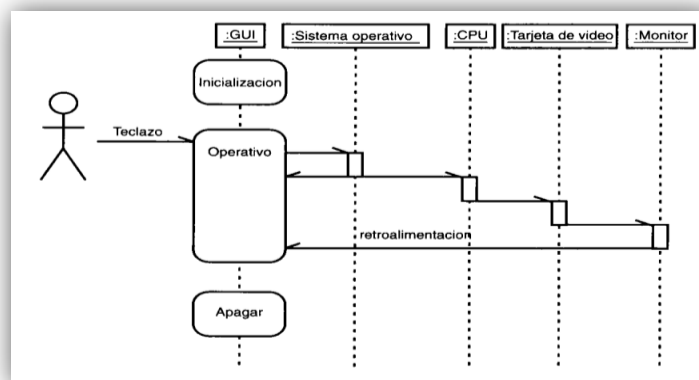


Figura 23

El caso de uso.

¿Qué es exactamente lo que representa un diagrama de secuencia? Muestra las interacciones de objetos que se realizan dentro de un escenario sencillo. Este escenario podría ser parte de un caso llamado "La opresión de una tecla". Al representar

gráficamente las interacciones del sistema, el diagrama de secuencia habrá, en efecto, “delineado” el caso de uso dentro de un sistema.

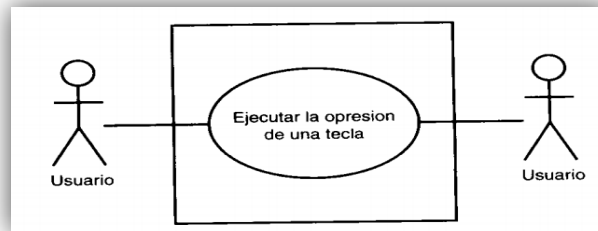


Figura 24

Instancias y genéricos.

Un diagrama de secuencia de instancia.

El cliente inicia el escenario mediante la inserción de dinero en la máquina. Luego hace una selección. Dado a que se habla del mejor escenario la maquina tiene al menos una lata gaseosa de la elegida y se la presenta al cliente fría.

Asumamos que en la máquina de gaseosas hay tres objetos: La fachada (la interfaz), el registrador del dinero y el dispensador. También se da por hecho que el registrador controla el dispipador. La secuencia será:

1. El cliente inserta el dinero.
2. El cliente hace su elección.
3. El dinero viaja al registrador.
4. El registrador verifica si la gaseosa está en el dispensador.
5. Asumimos que si hay la gaseosa y el registrador actualiza su reserva de efectivo.
6. El registrador hace que el dispensador entregue la gaseosa.

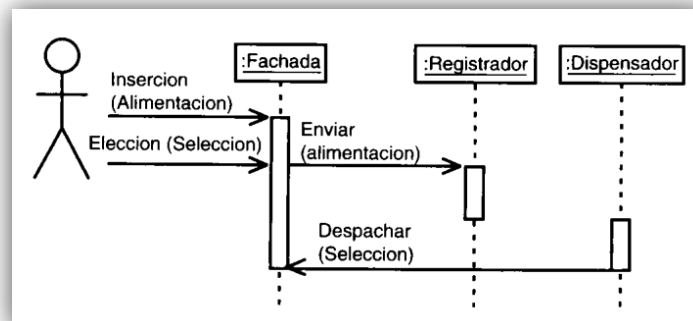


Figura 25

Un diagrama de secuencia genérico.

Si tomara en cuenta todos los escenarios dentro de un caso de uso al momento de crear un diagrama de secuencia, se trataría de un diagrama de secuencia genérico.

Se puede generar a partir del diagrama de secuencia de instancias. Para ello tendrá que justificar el control de flujo.

Escenario “Monto incorrecto”.

1. El registrador verifica si la alimentación del usuario concuerda con el precio.
2. Si el monto es mayor que el precio, el registrador calcula la diferencia y verifica si cuenta con cambio.
3. Si se puede devolver la diferencia, le devuelve el cambio al cliente y todo transcurre normal.
4. Si la diferencia no se encuentra, el registrador regresa el monto alimentado y mostrara un mensaje al cliente de que inserte el monto exacto.
5. Si la cantidad insertada es menor al precio, el registrador no hace nada y la maquina espera más dinero.

Para representar una condición en la secuencia , tal condición se colocara con un “si” entre corchetes.

Cada condición causara una bifurcación del control en el mensaje, que lo separa en rutas distintas. Cada ruta irá al mismo objeto, la bifurcación causara una “ramificación” de control en la línea de vida del objeto receptor, y separara las líneas de vida en rutas distintas.

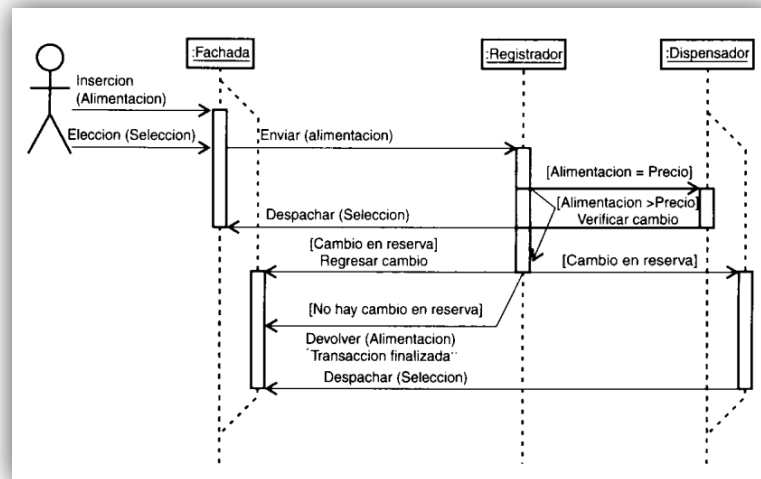


Figura 26

Escenario sin gaseosa.

1. Una vez que el cliente elija una gaseosa agotada, la maquina mostrara un mensaje de "Agotado".
2. La máquina mostrara un mensaje de que el cliente tiene que elegir otra elección.
3. El cliente tendrá la opción de un botón para que le regresen su dinero.
4. Si el cliente elige otra marca agotada, el proceso se reiniciará hasta que el cliente elija una marca en existencia u oprima el botón de que le regrese el dinero.

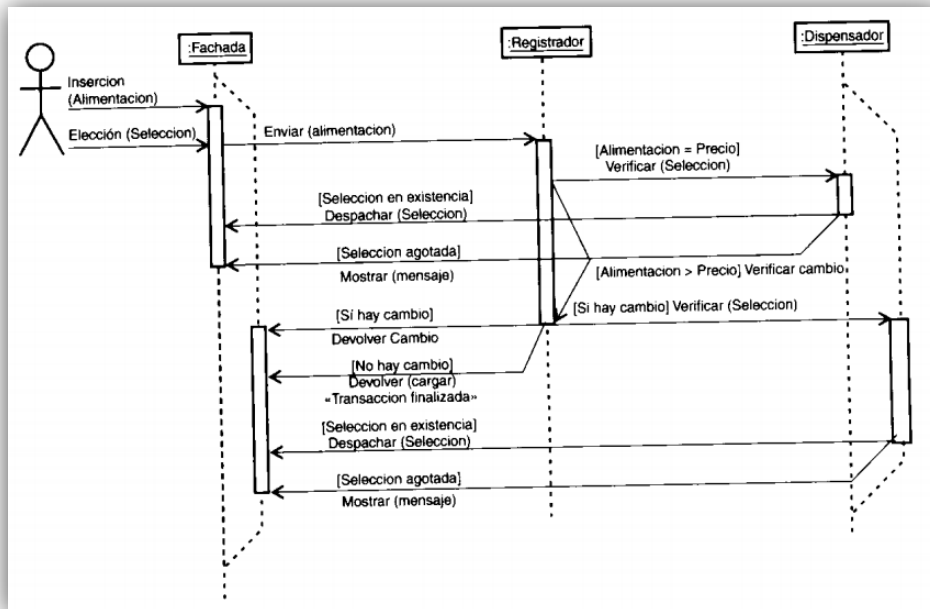


Figura 27

Creación de un objeto en la secuencia.

¿Cómo representaría la creación de un objeto cuando represente una secuencia de interacciones entre objetos?

El caso de uso de "Crear una propuesta" del ejemplo de la LAN en una firma de consultoría nos muestra una instancia de creación de objetos. Si damos por hecho que el consultor ya ha iniciado una sesión en la LAN, la secuencia modelada quedara como sigue:

1. El consultor querrá volver a utilizar una propuesta existente, y busca en un área centralizada de la red de la propuesta adecuada.
2. Si el conductor encuentra una propuesta adecuada, abrirá el archivo y, en el proceso, abrirá el software integrado para la oficina relacionada. El consultor guardara el archivo con un nuevo nombre, con lo que creara un nuevo archivo para la nueva propuesta.

3. Si el consultor no encuentre una propuesta, abrirá la aplicación de oficina y creará un archivo para la propuesta.
4. El consultor utilizará las aplicaciones de software integradas en la oficina.
5. Cuando el consultor finaliza la propuesta, lo guardará en el área de almacenamiento centralizada.

Cuando una secuencia da por resultado la creación de un objeto, se representa de la forma usual. La diferencia es que no lo colocará en la parte superior, sino que lo colocará en la parte inferior, de modo que su ubicación corresponda al momento que se cree. El mensaje que se crea al objeto se nombrará "Crear()". Los paréntesis implican una operación.

En el caso de "mientras", a este control de flujo lo representará colocando la condición mientras entre corchetes, con un asterisco (*) antes del primer corchete.

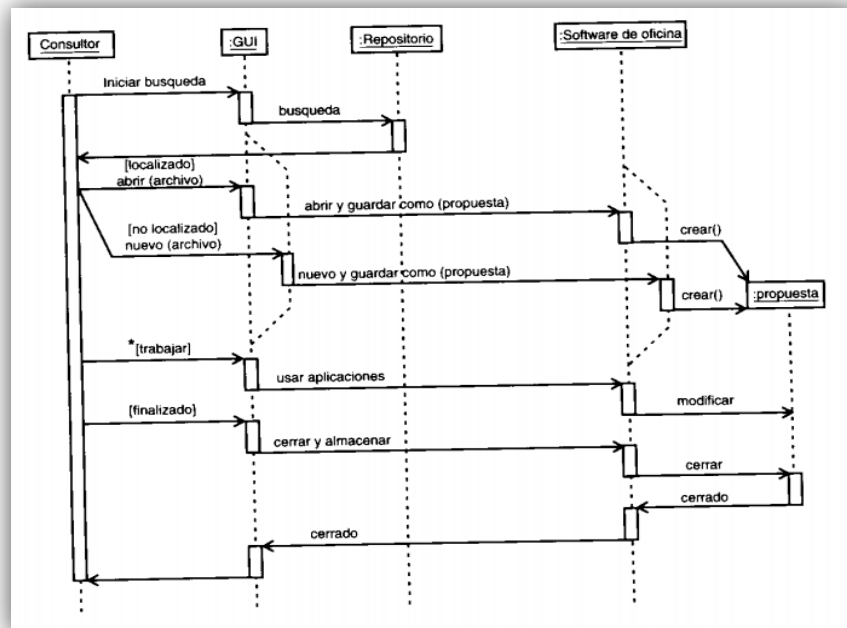


Figura 28

Como representar la recursividad.

En ocasiones un objeto cuenta con una operación que se invoca a si mismo. A esto se le conoce como recursividad, y es una característica principal en el lenguaje de la programación.

Ejemplo. Suponga que uno de los objetos sea una calculadora, y que una de sus operaciones sea el cálculo de intereses. Para calcular el interés de varios periodos el objeto tendrá que invocarse varias veces.

Para representar en UML, se dibujara una flecha de mensaje fuera de la activación que significa la operación y un pequeño rectángulo sobrepuesto que representa la activación. Dibuje la flecha a modo que dé hacia el rectángulo, y una que regresa al objeto que inicio la recursividad.

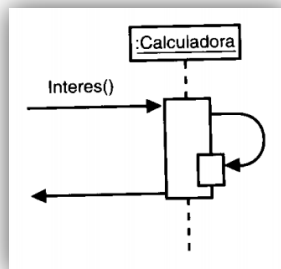


Figura 29

Bibliografía

Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. España: Prentice-Hall