



Universidad Tecnológica del Centro de Veracruz

Programa educativo de:

TÉCNICO SUPERIOR UNIVERSITARIO EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN.

Materia:

“INGENIERA DE SOFTWARE”

Área:

“Sistemas Computacionales”

Presenta:

Luis Ángel Fernández Vásquez Luis Ángel.

Grado y grupo: 5A TICS I DZ

Periodo cuatrimestral: Mayo-Agosto 2015.

Cuitláhuac, Veracruz, 28 de Mayo del 2015

Contenido

1	Hora 8	2
1.1	Diagramas de estados	2
1.1.1	Qué es un diagrama de estados	2
1.1.2	Simbología	2
1.1.3	Adición de detalles al icono de estado	2
1.1.4	Sucesos y acciones	3
1.1.5	Copia de seguridad	4
1.1.6	Subestados	4
1.1.7	Subestados secuenciales	4
1.1.8	Subestados concurrentes	5
1.1.9	Estados históricos	6
1.1.10	Mensajes y señales	6
1.1.11	Por que son importantes los diagramas de estados	6
1.1.12	Adiciones al panorama	7
2	Hora 9	7
2.1	Diagramas de secuencias	7
2.1.1	Que es un diagrama de secuencias	7
2.1.2	Objetos	8
2.1.3	Mensaje	8
2.1.4	Tiempo	8
2.1.5	La GUI	9
2.1.6	La secuencia	9
2.1.7	El diagrama de secuencias	9
2.1.8	El caso de uso	11
2.1.9	Instancias y genéricos	11
2.1.10	Un diagrama de secuencias de instancias	11
2.1.11	Diagrama de secuencias genérico	12
2.1.12	Creación de un objeto en la secuencia	14
3	Hora 10	14
3.1	Diagramas de colaboraciones	14
3.1.1	Que es un diagrama de colaboraciones	15
3.1.2	La GUI	15

1 Hora 8

1.1 Diagramas de estados

1.1.1 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.

El diagrama de estado 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.

1.1.2 Simbología

En la figura 1.1 se muestra un 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.

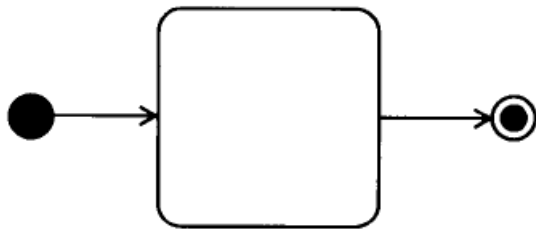
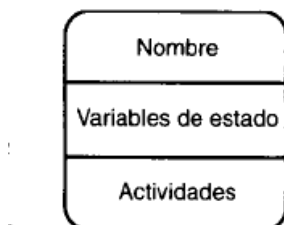


Imagen 1-1

1.1.3 Adición de detalles al icono de estado

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, el área central contendrá las variables de estado, y el área inferior las actividades se pueden apreciar en la figura 1.2.



Las actividades constan de sucesos y acciones: tres de las más utilizadas son entrada, salida y hacer. Puede agregar otras conforme sea necesario.

Imagen 1-2

Un ejemplo es muy claro es una máquina de fax que puede pasar por diversas variables y actividades de estado. Cuando se envía un fax este se encuentra en estado de envío de fax, posteriormente la máquina de fax anota la hora y fecha de salida en el que se envió el fax como también se graba en el número telefónico así como el nombre del 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 estado, la máquina jalara las hojas, paginara el fax y finalizara la transmisión como continuación lo presentamos en la figura 1.3.

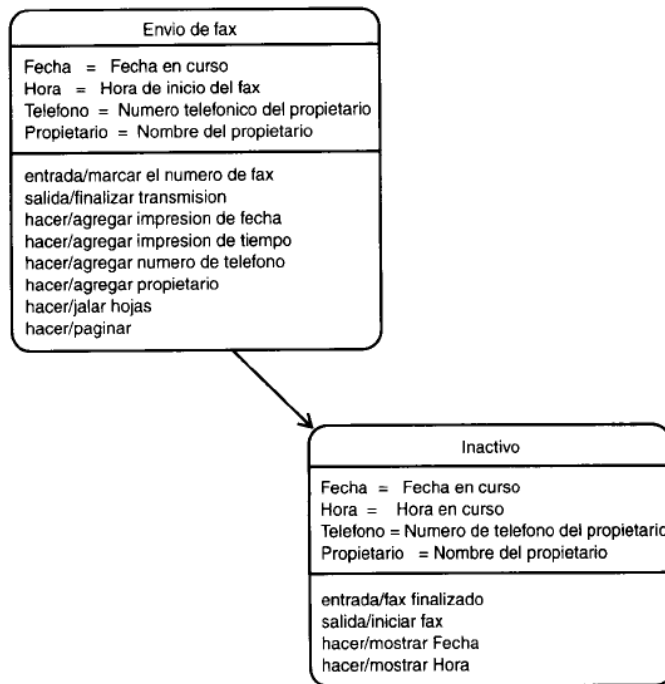


Imagen 1-3

1.1.4 Sucesos y acciones

En ocasiones un evento causara una transición sin una acción asociada, y algunas veces una transición sucederá dado que un estado finalizara una actividad. A este tipo de transición se le conoce como transición no desencadenada. La GUI con que interactúe le dará ejemplos de detalles de la transición. Por el momento, asumamos que la GUI puede establecer uno de tres estados.

Inicialización.

Operación.

Apagar.

Cuando se enciende un equipo de cómputo se desencadena un suceso que provoca que la GUI aparezca luego de una transición desde el estado de inicialización, y el arranque es una acción que se realiza durante tal transición.

Como resultado la GUI entra al modo de Operación. Cuando desea apagar su equipo de cómputo, desencadena un suceso que provoca la transición hacia el estado de Apagado, y con ello el equipo de cómputo se apaga. En la figura 1.4 se muestra un diagrama de cómo funciona.

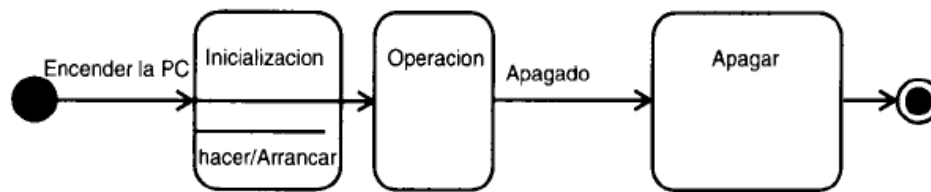


Imagen 1-4

1.1.5 Copia de seguridad

En ciertos términos de de cambio de estado si una PC pasa cierto tiempo sin interacción con el usuario la GUI hará una transición del estado Operación a un estado de Protector de pantallas.

El intervalo se especifica en el panel de control de su sistema operativo. Por lo general es de 15min. Cualquier opresión de una tecla o movimiento del ratón provocara una transición del estado Protector de pantallas o estado Operación.

El intervalo de 15min. Es una condición de seguridad cuando se llega a ella se realiza la transición como se muestra en el diagrama de la figura 1.5.

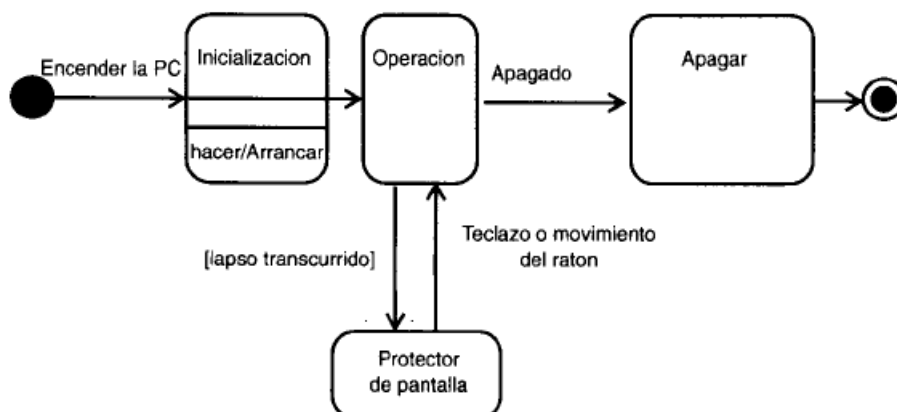


Imagen 1-5

1.1.6 Subestados

Cuando la GUI esta e estado de operación hay muchas cosas que ocurren tras bambalinas, aunque no sean particularmente evidentes en nuestra pantalla. La GUI aguarda de forma constante a que usted haga algo. Luego, deberá registrar tales acciones y modificar lo que se despliega para reflejarlas en la pantalla.

Con ello la GUI atravesara por varios cambios mientras se encuentra en el estado Operación. Tales cambios serán cambios de estado. Dado que estos estados se encuentran dentro de otros, se conocerán como subestados.

1.1.7 Subestados secuenciales

Los subestados suceden uno detrás de otro. Si retornamos los subestados 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 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 este, la GUI vuelve a iniciar a la espera de acción del usuario. La figura 1.6 muestra como representar los subestados.

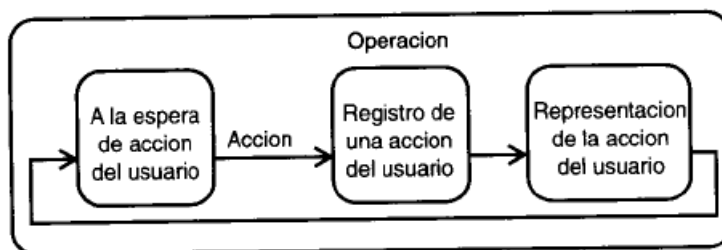


Imagen 1-6

1.1.8 Subestados concurrentes

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

Todo esto sucede al mismo tiempo que la secuencia que ya indique. Aunque cada secuencia es, claro, un conjunto de subestados secuenciales son concurrentes entre sí. Puede representar la concurrencia con una línea discontinua entre los estados concurrentes, como en la figura 1.7

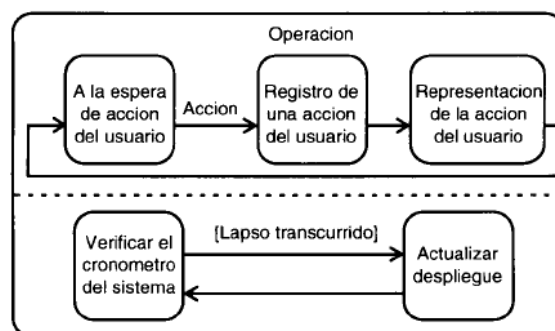


Imagen 1-7

1.1.9 Estados históricos

Cuando se activa su protector de pantallas y mueve su ratón para regresar al estado Operación ¿Qué ocurre? ¿Acaso su pantalla retorna 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 UML proporciona un símbolo que muestra que un estado compuesto recuerda su subestado activo cuando el objeto trasciende fuera del del estado compuesto. El símbolo es la letra “H” encerrada en un círculo que se conecta por una línea al subestado por recordar, con una punta de flecha que apunta a tal subestado como en la figura 1.9

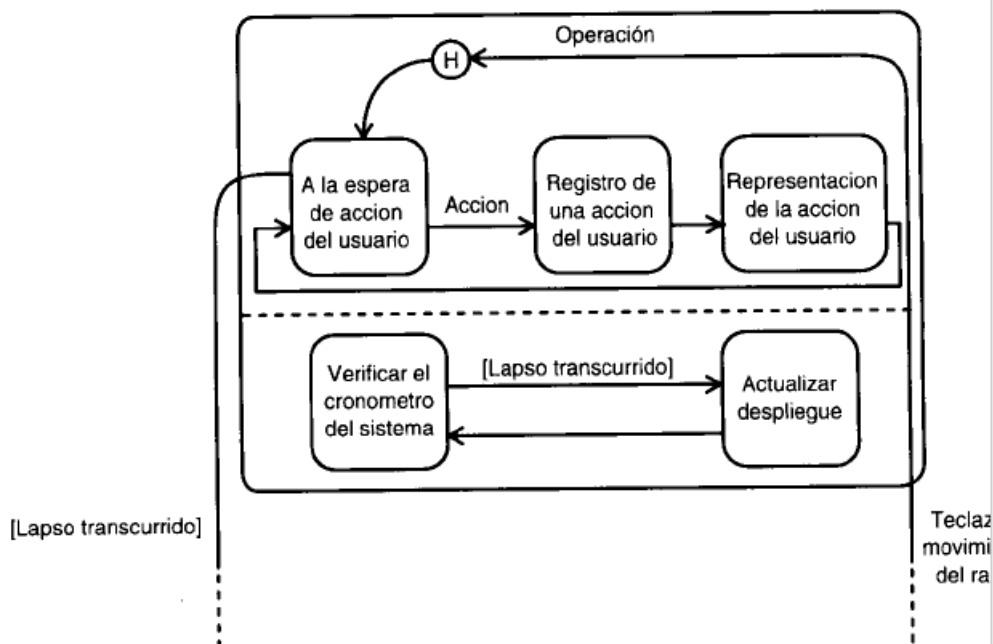


Imagen 1-8

1.1.10 Mensajes y señales

El suceso desencadenado que provoco 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. Esto es un concepto importante dado que los objetos se comunican mediante el envío de mensajes entre sí.

1.1.11 Por que son importantes los diagramas de estados

El diagrama de estado del UML proporciona una gran variedad de símbolos y abarca varias ideas. Este tipo de diagrama tiene el potencial de convertirse en algo complejo con pasmosa rapidez ¿En verdad es necesario?

Pues claro que es necesario contar con los diagramas de estados dado que permiten a los analistas, diseñadores y desarrolladores comprender el comportamiento de los objetos de un sistema.

1.1.12 Adiciones al panorama

Ahora puede agregar los “Elementos de comportamiento” al panorama del UML. La figura 1.9 muestra la imagen con el diagrama de estados incluido.

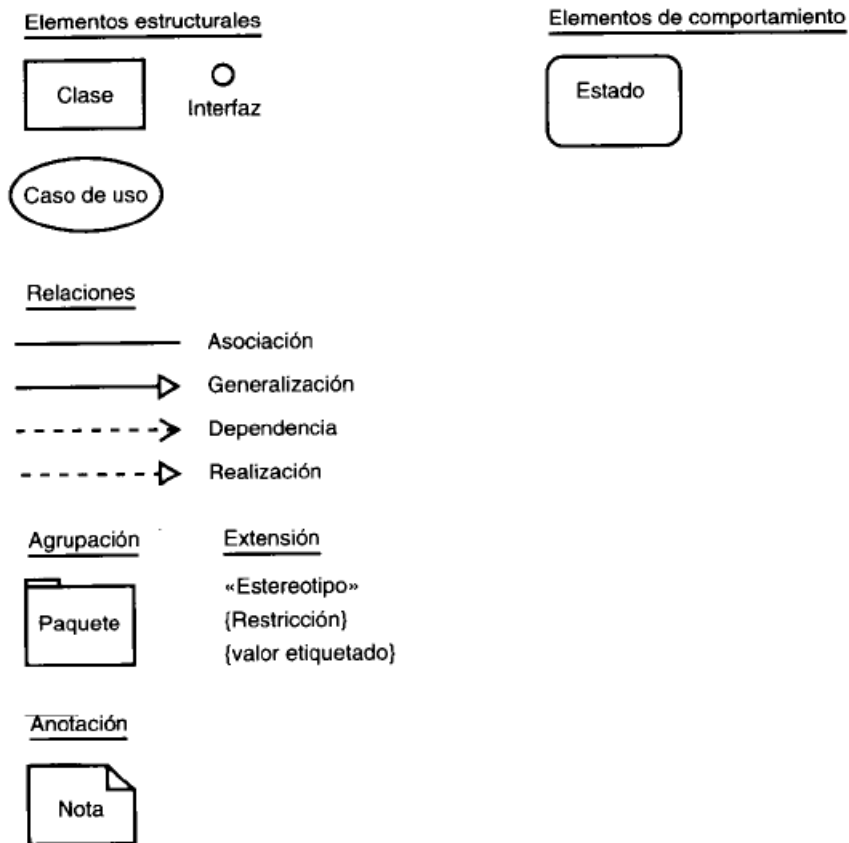


Imagen 1-9

2 Hora 9

2.1 Diagramas de secuencias

Los diagramas de estados se enfocan a los de diferentes estados de un objeto. Esto es solo una parte pequeña parte del cuadro. El diagrama de secuencias del UML establecen el siguiente paso y les muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo.

2.1.1 Que es un diagrama de secuencias

El diagrama de secuencias consta de objetos que representan del modo usual: rectángulos con nombre, mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

2.1.2 Objetos

Los objetos se colocan cerca de la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen el diagrama. La extensión que está debajo 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. Como lo muestra la figura 2.1



Imagen 2-1

2.1.3 Mensaje

Un mensaje que va de un objeto a otro pasa por la línea de vida de un objeto a la de otro. Un objeto puede enviarse un mensaje a sí mismo.

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, esperara la respuesta a tal mensaje antes de continuar con su trabajo. Si un objeto envía un mensaje asincrónico, esperara una respuesta antes de continuar con su trabajo como se muestra en la figura 2.2.

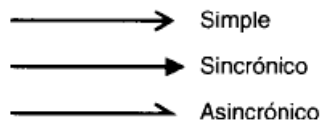


Imagen 2-2

2.1.4 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 este cerca de la parte inferior.

La dimensión horizontal es la disposición de los objetos, y la dimensión vertical muestra el paso del tiempo. La figura 2.3 muestra un actor que inicia la secuencia, aunque, en sentido estricto, la figura adjunta no es parte del conjunto de símbolos del diagrama de secuencias.

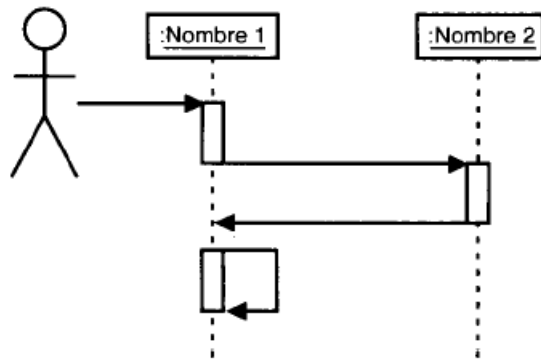


Imagen 2-3

2.1.5 La GUI

2.1.6 La secuencia

Supongamos 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 video.
5. La tarjeta de video envía un mensaje al monitor.
6. El monitor presenta el carácter alfanumérico en la pantalla, con lo que se hará evidente al usuario.

2.1.7 El diagrama de secuencias

En el diagrama de secuencias de la GUI ve si los mensajes son asincrónicos: ninguno del componente aguarda nada antes de continuar. Cuando teclea en procesador de textos, en ocasiones no ve parecer en la pantalla el carácter correspondiente a la tecla que haya oprimido sino hasta que haber oprimido algunas más, como se muestra a continuación en la figura 2.4.

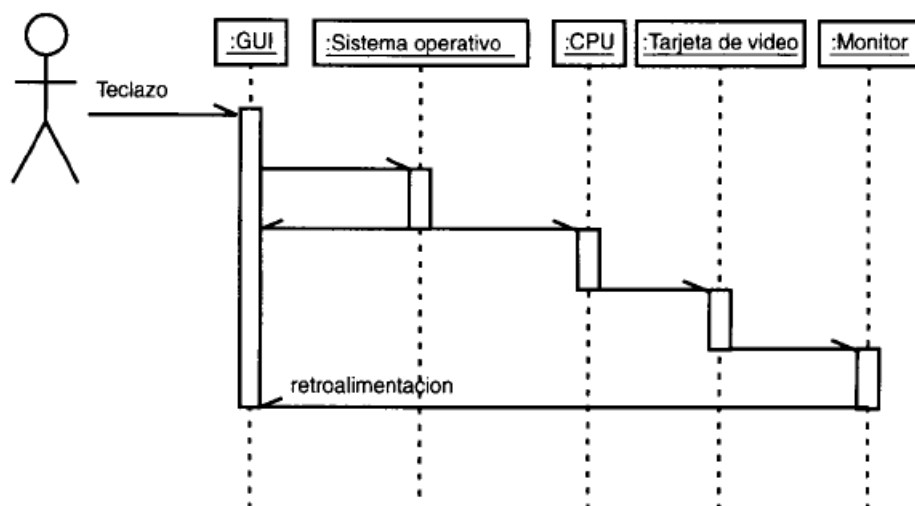


Imagen 2-4

Es muy instructivo mostrar los restos de uno o varios objetos en el diagrama de secuencias. La figura 2.5 nos muestra un híbrido: el diagrama de secuencias de la GUI con los estados de la GUI.

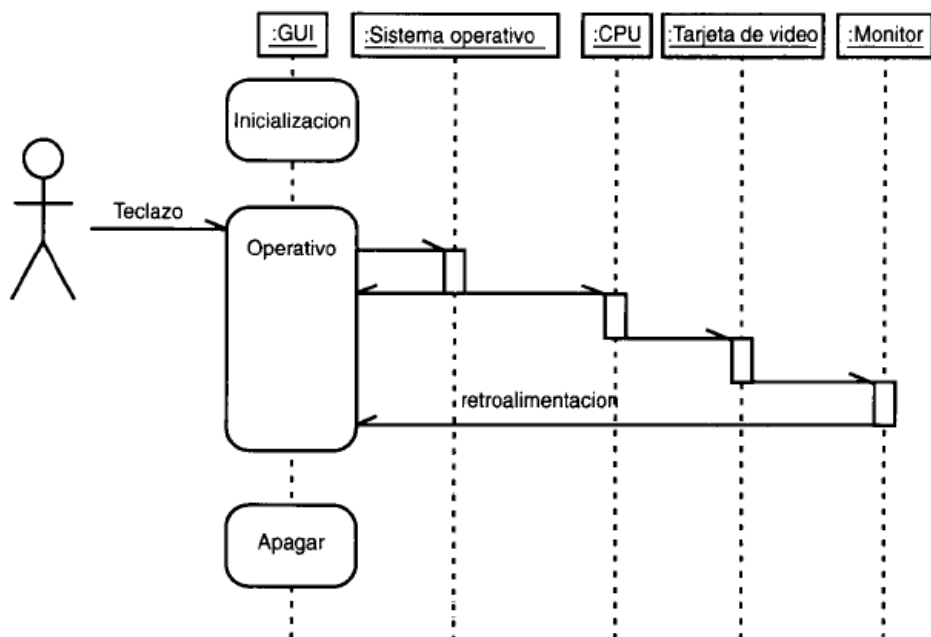


Imagen 2-5

2.1.8 El caso de uso

En la figura 2.6 no muestra las interacciones de objetos que se realizan durante un escenario sencillo: la opresión de una tecla. A l 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.

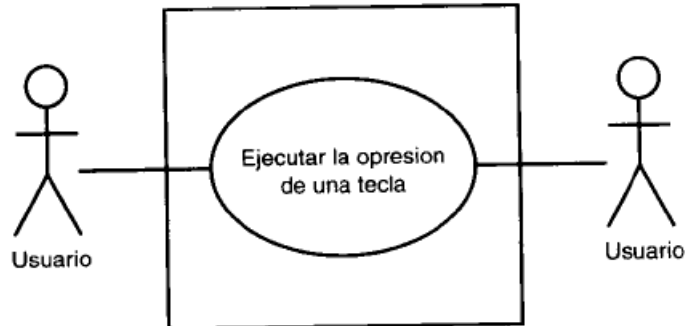


Imagen 2-6

2.1.9 Instancias y genéricos

2.1.10 Un diagrama de secuencias de instancias

El cliente inicia el escenario mediante la inserción de dinero en la máquina. Luego hace una selección. Dado que hablamos del mejor escenario, la maquina 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, el registrador de dinero, y el dispensador. También daremos por hecho que el registro de dinero controlara al dispensador. La secuencia será como sigue:

1. El cliente inserta el dinero en la alcancía que se encuentras 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, asumidos 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.

Dado el diagrama se secuencias solo se centra en un escenario en el caso de uso “Comprar gaseosa” se conoce como diagrama figura 2.7.

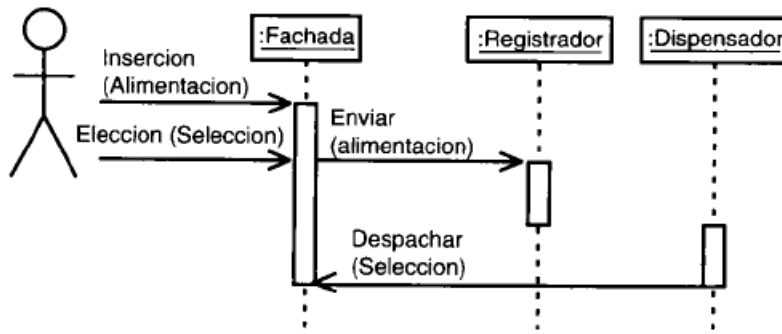


Imagen 2-7

2.1.11 Diagrama de secuencias genérico

En este caso podrá generar el diagrama de secuencias genérico a partir del diagrama de secuencias de instancias. Para ello tendrá que justificar el control del flujo. Esto es, tendrá que representar las condiciones y consecuencias de “Monto incorrecto” y “Sin gaseosa”

Para el escenario relacionado con “Monto incorrecto”

1. El registrador verifica si la alimentación del usuario concuerda con el precio de la gaseosa.
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, el registrador devuelve el cambio al cliente y todo transcurre como antes.
4. Si se puede devolver la diferencia no se encuentra en la reserva del cambio, el registrador regresará el monto alimentado y mostrará en mensaje que indique al cliente que inserte el monto exacto.
5. Si la cantidad insertada es menor al precio, el registrador no hace nada y la máquina esperará más dinero.

Para representar cada condición en la secuencia, tal condición colocará en un “si” entre corchetes. Arriba de las flechas de mensaje apropiadas agrega [alimentación > precio], [alimentación - precio no presente], y [alimentación - precio presente].

Cada condición causará una bifurcación de control en el mensaje, que separará el mensaje en rutas distintas. Como cada ruta irá del objeto, la bifurcación causará una “ramificación” del control en la línea de vida de la secuencia, las ramas del mensaje confluirán, como las bifurcaciones en las líneas de vida como se muestra en la figura 2.8.

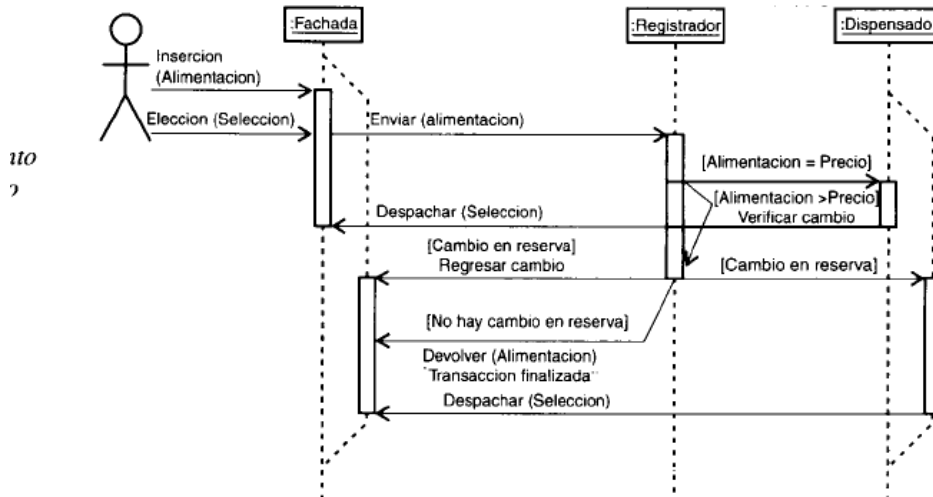


Imagen 2-8

Ahora agregaremos el escenario “Sin gaseosa”

1. Una vez que el cliente elige un marca agotada, la maquina mostrara un mensaje de “Agotado”.
2. La máquina mostrara un mensaje que solicitara al cliente que haga otra selección.
3. El cliente tendrá la opción de oprimir un botón para que se le regrese su dinero.
4. Si el cliente elige una marca en existencia, todo procederá como en el mejor escenario si el monto insertado es correcto Si no lo es, la maquina seguirá por el escenario del “Monto incorrecto”.
5. Si el cliente elige otra marca agotada, el proceso se repetirá hasta que el cliente elija una marca en existencia o presione un botón que le regrese su dinero.

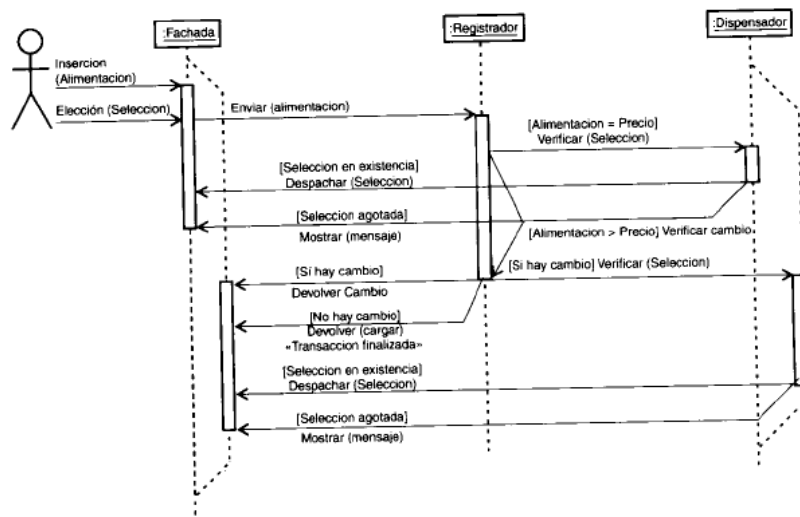


Figura 2.9

2.1.12 Creación de un objeto en la secuencia

Con frecuencia se da el caso que un programa orientado a objetos debe crear un objeto. Recordar que en términos del software, una clase es una plantilla para crear un objeto. ¿Cómo representaría la creación de un objeto cuando presente una secuencia de interacciones entre objetos?

El caso de uso “Crear una propuesta” de ejemplo de la LAN en un firma de consultoría nos muestra una instancia de la creación de objetos. En este ejemplo concebirá la LAN en el entendido que todo se realiza mediante la red. Si damos por hecho que el consultor ya ha iniciado una sesión en la LAN, la secuencia que modelara quedara como sigue:

1. El consultor querrá volver a utilizar partes de una propuesta existente y busca en un área centralizada de la des a propuesta adecuada.
2. Si el consultor encuentra una propuesta adecuada, abrirá el archivo y, en el proceso, abrirá el software integrado para la oficina relacionada.
3. Si el consultor no encuentra una propuesta, abrirá la aplicación de oficina y creara un archivo para la propuesta.
4. Al trabajar en la propuesta, el consultor utilizara las aplicación del software integrado para oficina
5. Cuando el consultor finalice la propuesta, la guardara en el área de almacenamiento centralizada.

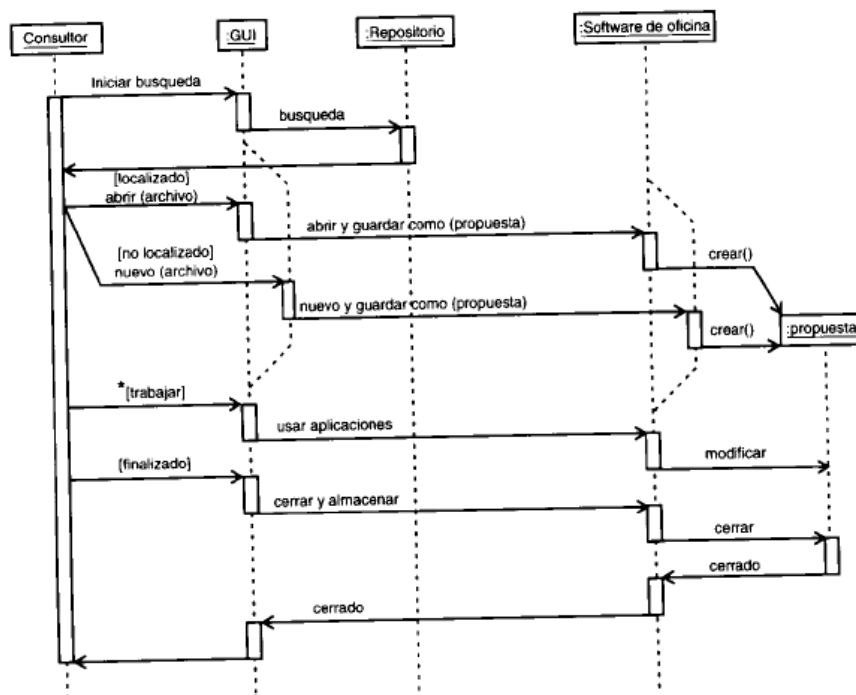


Imagen 2-9

3 Hora 10

3.1 Diagramas de colaboraciones

Los diagramas de colaboraciones muestra la forma en que los objetos colaboras entre sí, tal como sucede con un diagrama de secuencias. Muestran los objetos juntos con los mensajes que se envían e un activo entre ellos. Si el diagrama de secuencias hace eso, ¿Por qué el UML requería otro diagrama? ¿Qué son lo mismo? ¿No es una perdida?

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.

Como se infiere, es útil contar con ambas formas. Los diagramas de secuencias destacan la secesión de las iteraciones.

3.1.1 Que 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. A demás de las relaciones entre objetos entre sí. Por lo general, evitara la multiplicidad dado que podría ser fuente de confusión.

Mencione que podrá convertir cualquier diagrama de secuencias en un diagrama de colaboraciones y viceversa. Por medio de esto podrá representar las informaciones secuencia en un diagrama de colaboraciones. Parra ello, agregara una cifra a la etiqueta de un mensaje separadamente mediante dos puntos.

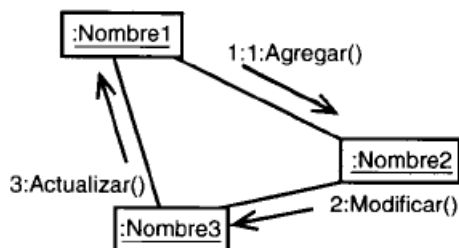


Imagen 3-1

3.1.2 La GUI

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

- La GUI notifica al sistema operativo que se oprimió una tecla.
- El sistema operativo le notifica a la CPU.
- La CPU notifica a la tarjeta de video.
- La tarjeta de video envía un mensaje alfanumérico en la pantalla, con lo que hará evidente al usuario.

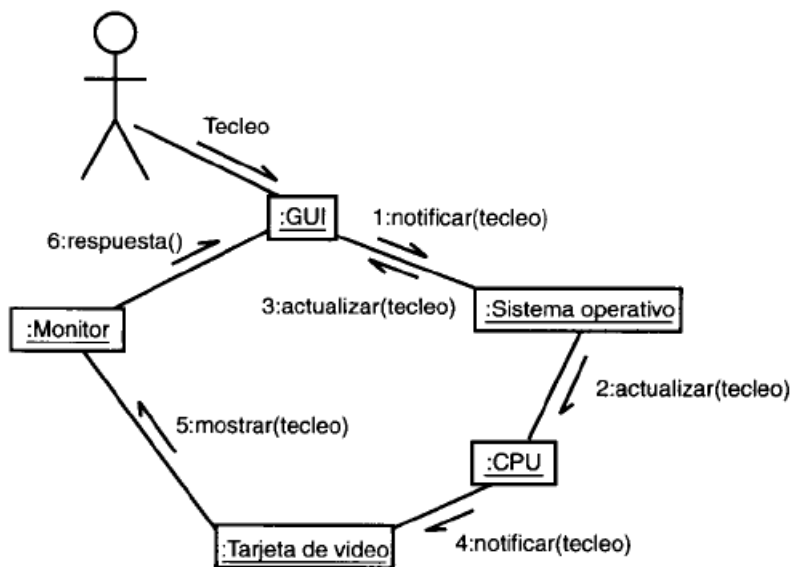


Imagen 3-2

3.2 Cambio 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.

La figura 3.3 ilustra un cambio de estado para la GUI, que muestra que el estado de inicialización se convierte en el estado operativo.

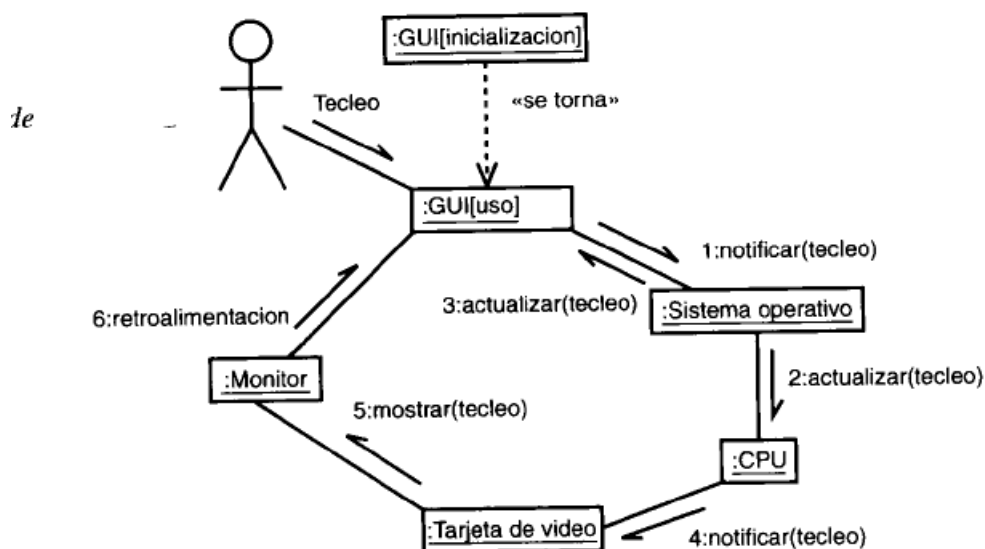


Imagen 3-3

3.3 La máquina de gaseosas

Las cosas se hacen más interesantes cuando aplicas las condiciones a una situación real.

Iniciemos con la mejor situación del caso de uso “Comprar gaseosa” donde 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 el mejor escenario, asumidos 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.

El diagrama de colaboraciones es directo como lo muestra la figura 3.4.

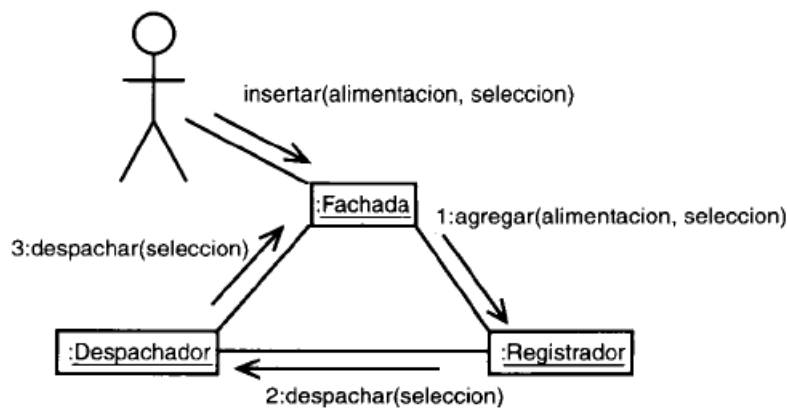


Imagen 3-4

Ahora agregamos el caso de “Cantidad incorrecta de dinero” El diagrama tiene que contabilizar varias decisiones:

- El usuario ha introducido más dinero que el necesario para comprar.
- La máquina cuenta con la cantidad adecuada de cambio.
- La máquina no tiene la cantidad correcta de cambio.

Esto podría ser algo complicado por lo que haremos el diagrama por secciones. Empezaremos con la condición donde el usuario ha insertado más dinero del indicado en el precio y el registrador cuenta con el cambio adecuado. Agregaremos el paso de la máquina al devolver cambio utilizando el mismo número del mensaje que verificara el cambio y agregaremos un punto decimal y un uno.

Cuando agreguemos esta condición, agregaremos una bifurcación en el centro de flujo. Numeraremos esta bifurcación como mensaje anidado. Dado que es el segundo mensaje anidado. Habrá un 2 luego el puente decimal. Finalmente y debido a que la transacción habrá finalizado, hará clara esta situación mediante la adición de un

estereotipo “transacción finalizada” en este mensaje, y otro mensaje que despacha la gaseosa como se muestra en la figura 3.5.

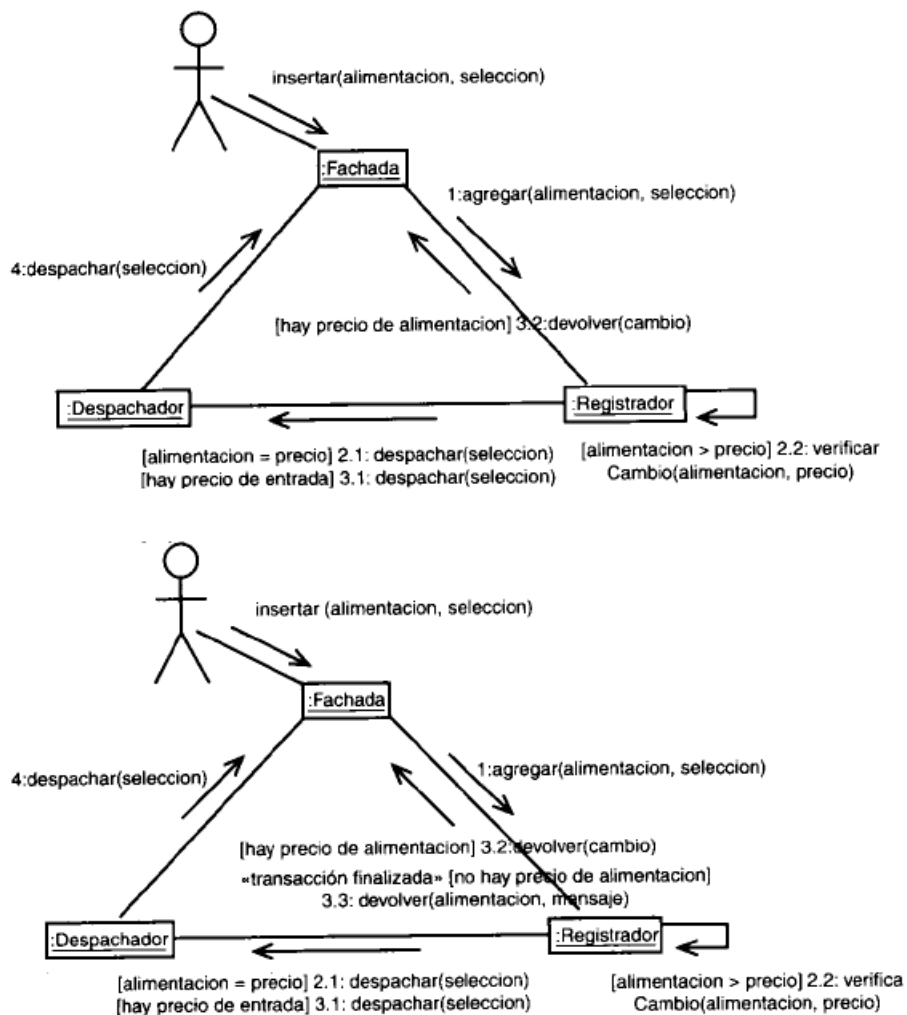


Imagen 3-5

(SCHMULLER, 2001)