



**Universidad tecnológica del centro de Veracruz**

**Programa Educativo:**

Tecnologías de la información y comunicación

**Área:**

Sistemas

**Materia:**

Ingeniería de software I

**Tema:**

**Resumen digital:**

Hora 8 – diagrama de estado

Hora 9 – diagrama de secuencia

Hora 10 – diagrama de colaboración

**Alumna:**

Elizabeth Ramírez medina

**Matricula:**

6825



## Índice

I HORA OCHO.....	5
1.1 Diagramas de estado .....	5
1.1.1 Simbología.....	5
1.2 Adición de detalles al icono de estado.....	5
1.3 Sucesos y acciones.....	6
1.4 Condiciones de seguridad .....	6
1.5 Subestados.....	7
1.6 Subestados secuenciales .....	7
1.7 Subestados concurrentes .....	8
1.8 Estados históricos.....	8
1.9 Mensajes y señales .....	9
1.10 Adiciones al panorama .....	9
II HORA NUEVE .....	9
2.1 Diagramas de secuencias .....	10
2.2 Que es un diagrama de secuencia.....	10
2.3 Objetos .....	10
2.4 Mensaje.....	10
2.5 Tiempo.....	11
2.6 El diagrama de secuencias .....	12
2.7 El caso de uso.....	13
2.8 Instancia y genéricos.....	14
2.9 Un diagrama de secuencias de instancias .....	14
2.10 Un diagrama de secuencias genérico.....	15
2.11 Creación de un objeto en la secuencia .....	15
2.12 Como representar la cursiva .....	15
III HORA DIEZ .....	16
3.1 Diagrama de colaboración.....	16
3.2 La GUI .....	16
3.3 Cambios de estado.....	17
3.4 Creación de un objeto .....	17
3.5 Algunos conceptos más.....	18

3.6 Varios objetos receptores en una clase .....	18
3.7 Objetos activos .....	18
3.8 Sincronización .....	18

## I HORA OCHO

### 1.1 Diagramas de estado

Una manera de caracterizar un cambio en un sistema que los objetos que lo componen modificó su estado como respuesta a los sucesos y al tiempo. Un diagrama de estado UML captura ese tipo de cambios mostrando el punto inicial y final de una secuencia de cambio de estado, un diagrama de estado es conocido como un motor de estado teniendo en cuenta que muestra las condiciones de un solo objeto.

#### 1.1.1 Simbología

La figura 1.1 le muestra el rectángulo de vértices redondeados que representa a un estado, junto con una línea continua y una punta de flecha misma que representa una transacción. 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.

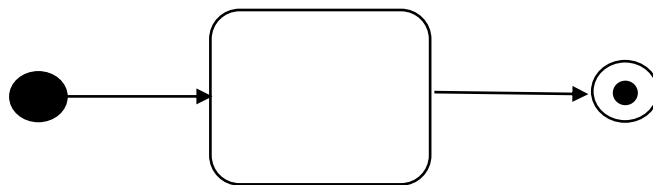


Figura 1.1

Figura 1.1 símbolos de UML en un diagrama de estado. El icono para el estado es un rectángulo de vértices redondeados, y el símbolo de una transición es una línea continua y una punta de flecha. Un círculo relleno se interpreta como el punto inicial de secuencia de estado, y un diana representa el punto final.

### 1.2 Adición de detalles al icono de estado

El UML le da la opción de agregar detalles de simbología. Puede subdividir el símbolo del estado en áreas que muestren el nombre, la variable y actividades del estado. Como lo muestra la figura 1.2.

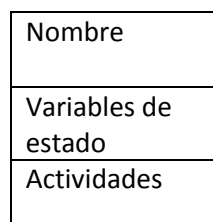


Figura 1.2

Las variables de estado como cronometro o contadores son, en ocasiones, de ayuda. Las actividades constan de sucesos y acciones las tres 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 si es necesario.

### 1.3 Sucesos y acciones

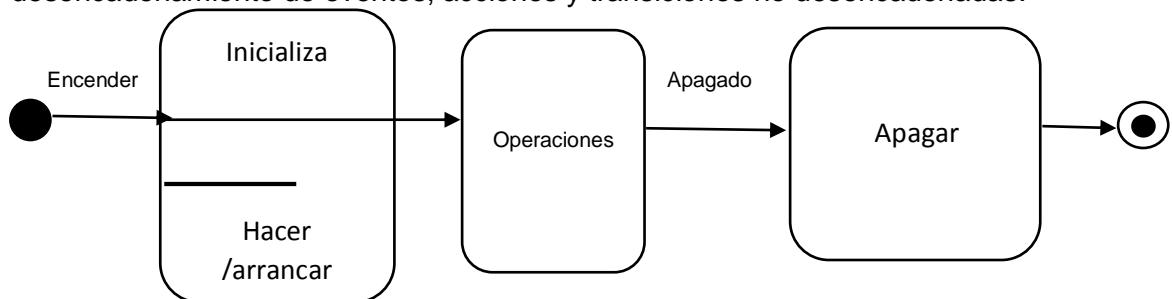
También se puede agregar ciertos detalles a las líneas de transición. Pueden indicar un suceso que provoque una transacción (desencadenar un secuso), ya la actividad de cómputo (la acción) que se ejecuta y haga que suceda la modificación del estado.

A los sucesos y acciones los escribirán cerca de la línea de transición mediante un diagonal para esperar un suceso desencadenado de una acción.

La GUI (interfaz gráfica de usuario) con que interactúa le dará ejemplos de detalles de la transacción. La GUI puede establecerse en uno de tres estados:

- Inicialización
- Operación
- Apagar

Figura 1.3 Los estados y transiciones de una interfaz gráfica del usuario incluyen l desencadenamiento de eventos, acciones y transiciones no desencadenadas.



### 1.4 Condiciones de seguridad

Una transacción puede suceder como respuesta a un suceso desencadenado, e implicar una respuesta o acción, una transición también puede ocurrir por la actividad en un estado: una transición que ocurre de esta forma se conoce como transición no desencadenada. Una transición puede ocurrir cuando se cumple una condición particular, o condición de seguridad.

Una condición de seguridad: cuando se llega a ella, se realiza la transición. La figura 1.4 muestra el diagrama de estado de la GUI con el estado Protector de pantalla y la condición de seguridad añadida. Vea que la condición de seguridad se establece como expresión booleana.

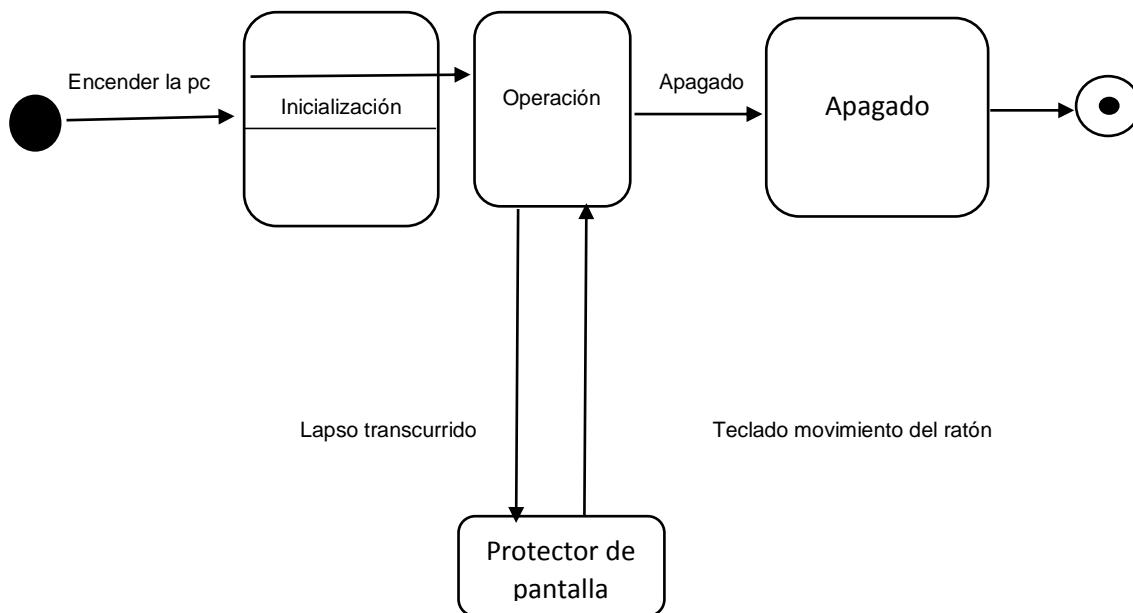


Figura 1.4

### 1.5 Subestados

Cuando GUI está en el estado Operación, hay muchas cosas que ocurren tras bambalinas, aunque no sean particularmente evidentes en la pantalla. Con ello la GUI atravesará por varios cambios mientras se encuentre en el estado Operación. Tales cambios serán cambios de estado. Dado que estos estados se encuentran entre otros, se conocerán como subestados. Hay dos tipos de subestados: secuencial y concurrente.

### 1.6 Subestados secuenciales

Los subestados secuenciales suceden uno detrás de otro. Si retomamos los subestados mencionando con anterioridad dentro del estado Operación de la GUI, tendrá la siguiente secuencia:

A la espera de acción del usuario

Registro de la acción del usuario

Representación de la acción del usuario

La figura 1.6 le muestra cómo representar los subestados secuenciales dentro del estado Operación.

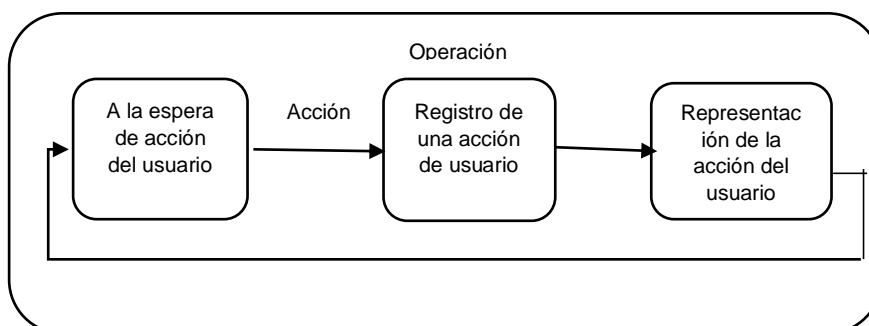


Figura 1.5

### 1.7 Subestados concurrentes

Dentro de la operación, la GUI no sólo aguarda a que usted haga algo. También verifica el cronograma del sistema y (posiblemente) actualiza el despliegue de una aplicación luego de un intervalo específico. Aunque cada secuencia es, un conjunto de subestados secuenciales, las dos secuencias son concurrentes entre sí. Puede representarse la concurrencia con una línea discontinua entre los estados concurrentes, como en la figura 1.7.

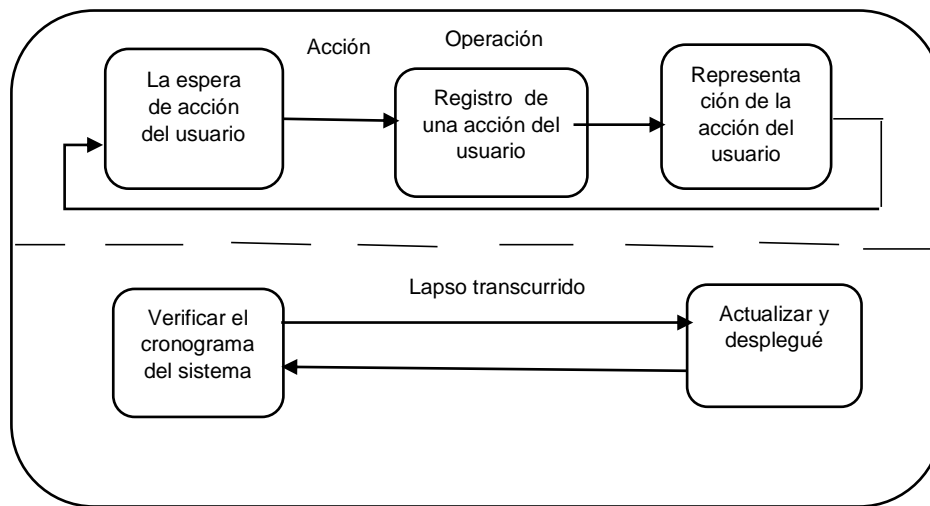


Figura 1.7

Las partes ocurrentes 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.

### 1.8 Estados históricos

El diagrama de estados históricos captura la idea. El UML proporciona un símbolo que muestra que un estado compuesto recuerda su subestados activos cuando objetos trasciendes fuera del estado compuesto. El símbolo de la "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 1.8 muestra este símbolo en el estado Operación.

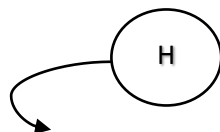


Figura 1.8

El estado histórico



Simbolizado con la “H” dentro del

Círculo, le muestra que un estado compuesto recuerda su subestados

Activos cuando el objeto trasciende fuera de tal estado compuesto.

El estado histórico y el estado inicial (representados por el círculo relleno) son conocidos como pseudoestados.

No tienen variables de estado ni actividades, por lo que no son estados “completos”.

### 1.9 Mensajes y señales

Un mensaje que desencadena una transición en el diagrama de estado del objeto receptor se conoce como señal. En el mundo de la orientación a objetos señales y transmitir al objeto receptor. El objeto señala cuenta con propiedades que se representan como atributos. Dado que una señal es un objeto, es posibles crear jerarquía de herencia de señales.

Por qué son importantes los diagramas de estado

Es necesario contar con los diagrama de estado dado que permiten a los analistas, diseñadores y desarrolladores comprender el comportamiento delos objetos de un sistema.

Los diagramas de estado se aseguran que no tendrán que adivinar lo que se supone que harán los objetos.

### 1.10 Adiciones al panorama

Ahora puede agregar los “Elementos de comportamiento” al panorama de UML.

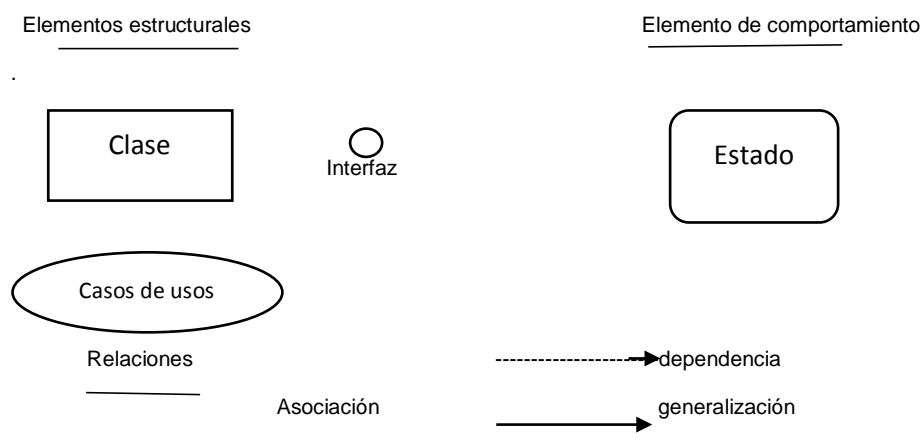


Figura 1.10

## II HORA NUEVE

## 2.1 Diagramas de secuencias

Los diagramas de estados se enfocan a los diferentes estados de uno objeto. El diagrama de secuencia del UML establece el siguiente paso y le muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo.

## 2.2 Que es un diagrama de secuencia

El diagrama de secuencia consta de objetos que se representan del mundo usual: rectángula con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

## 2.3 Objetos

Los objetos se colocan cerca de las partes superior del diagrama de izquierda a derecha y se acomoda de manera que explique el diagrama. Línea de vida es la extensión que ésta debajo (y en forma descendente) de cada objeto será una línea descontinua. La figura 2.3 muestra un objeto, su línea de vida y su activación.

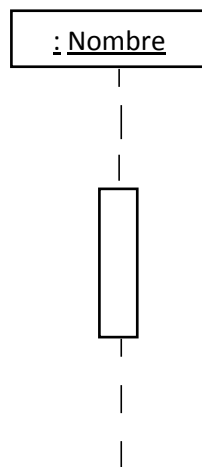


Figura 2.3

## 2.4 Mensaje

Un mensaje que va de objetos a otro pasa de la línea de vida de un objeto a la otra. Un mensaje puede ser simple, sincrónico, o asincrónico. Un mensaje simple es la transferencia del control de un objeto a otro. Un mensaje sincrónico, esperará la respuesta a tal mensaje antes de continuar con su trabajo. En el diagrama de secuencias. Los símbolos del mensaje varían, 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 2.4

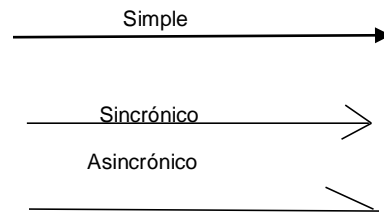


Figura 2.4

## 2.5 Tiempo

El diagrama representa al tiempo en dirección vertical. El diagrama de secuencias tiene dos dimensiones. La figura 2.3 en un diagrama de secuencia los objetos se colocan de izquierda a derecha en la parte superior. Cada línea de vida de un objeto es una línea discontinua que se desplaza hacia abajo del objeto. Una línea continua con una punta de flecha conectada a una línea de vida con otra, y representa un mensaje de un objeto a otro. El tiempo se inicia en la parte superior y continúa hacia abajo. aunque un actor es el que normalmente inicia la secuencia, su símbolo no es parte del diagrama de secuencia.

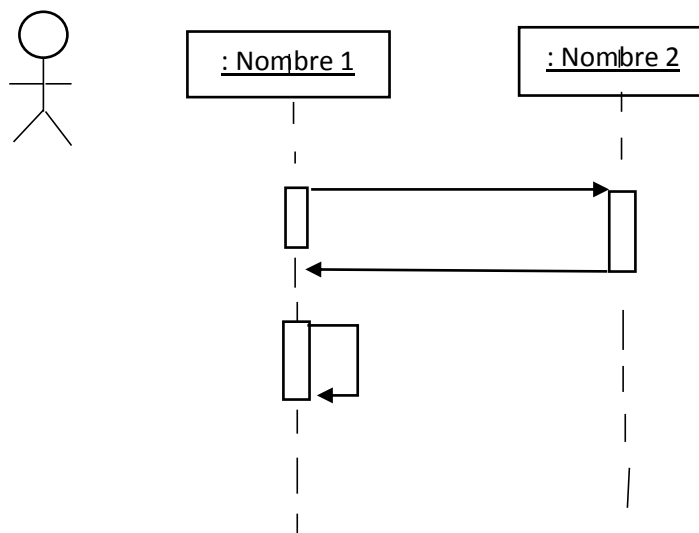


Figura 2.5

## 2.6 El diagrama de secuencias

La figura 2.4 representa el diagrama de secuencias de la GUI. Como ve, los mensajes son asíncronos: ninguno de los componentes aguardan nada antes de continuar.

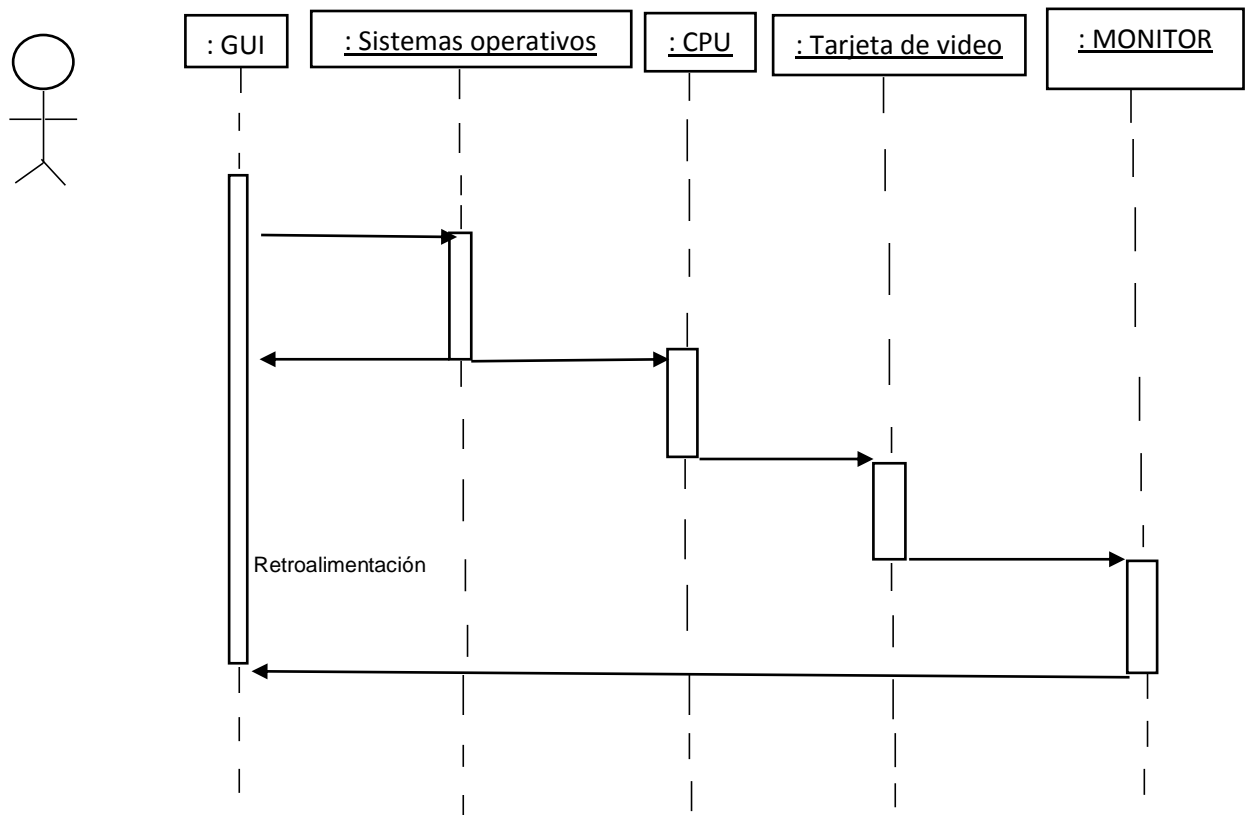


Figura 2.6

Un diagrama de secuencia puede mostrar los estados de un objeto

La figura 2.6 muestra un híbrido: el diagrama de secuencias de la GUI con los estados de la GUI. Observe que la secuencia se origina y finaliza en el estado Operativo de la GUI, como podría esperarlo.

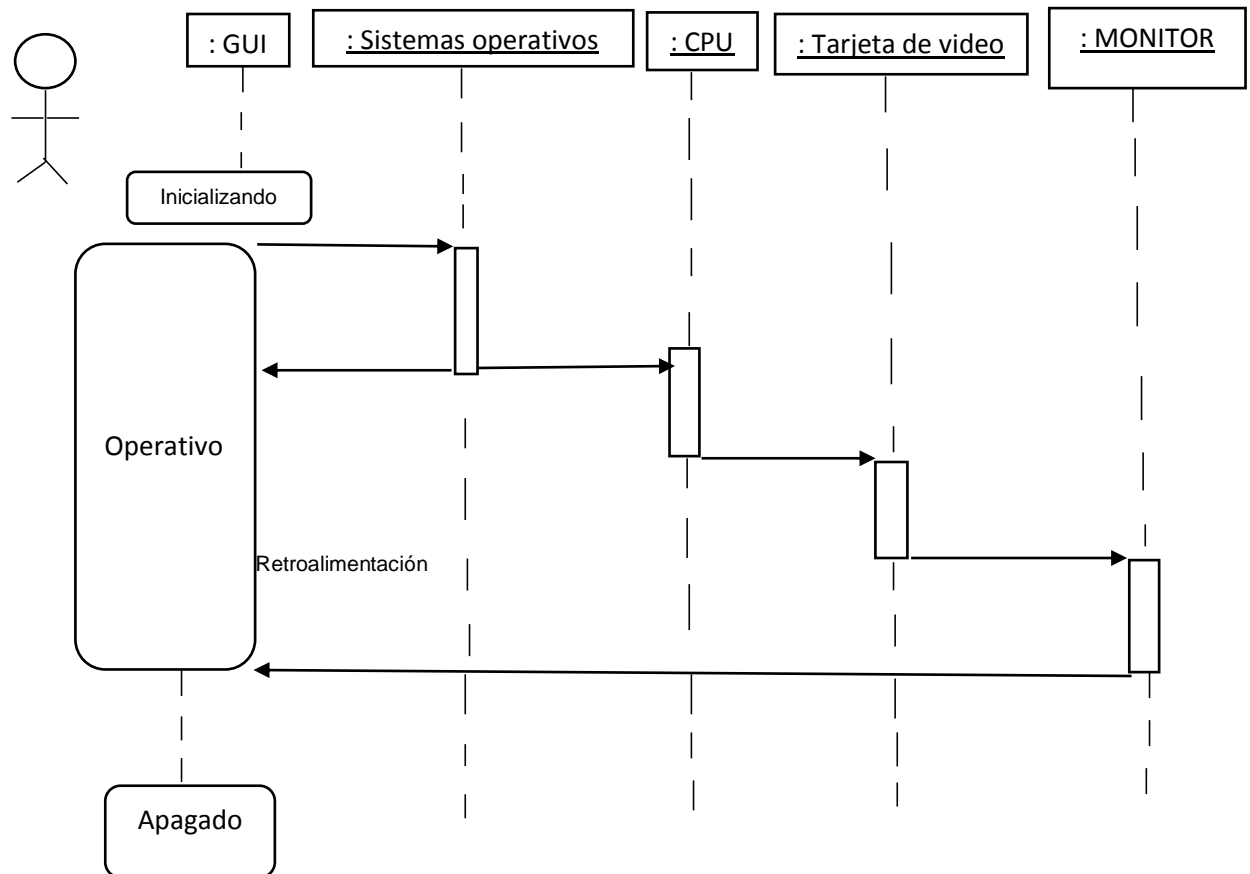


Figura 2.6

Un diagrama de secuencias, otra forma de mostrar el cambio de estado de un objeto es incluir al objeto más de una vez en el diagrama.

## 2.7 El caso de uso

¿Qué es exactamente lo que representa un diagrama de secuencia? Por ejemplo. Por 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”.

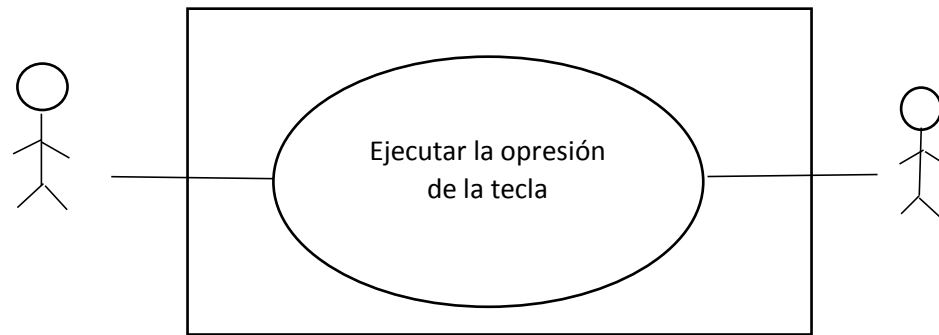


Figura 2.7

## 2.8 Instancia y genéricos

El ejemplo anterior comenzó con un diagrama de estado. Este otro ejemplo empieza con un caso de uso. “comprar gaseosa” fue uno de los casos de usos del ejemplo de las máquinas de gaseosas en las 6, “introducción a los casos de uso”, y 7 “diagramas de caso de usos”.

## 2.9 Un diagrama de secuencias de instancias

Dado que el diagrama de secuencias sólo se centra en un escenario (una instancia) en el caso de uso “comprar gaseosa”, se conoce como diagrama de secuencia de instancia. La figura 2.7 le muestra este diagrama. Vea que el diagrama muestra mensajes sencillos. Cada mensaje mueve el flujo de control de un objeto a otro.

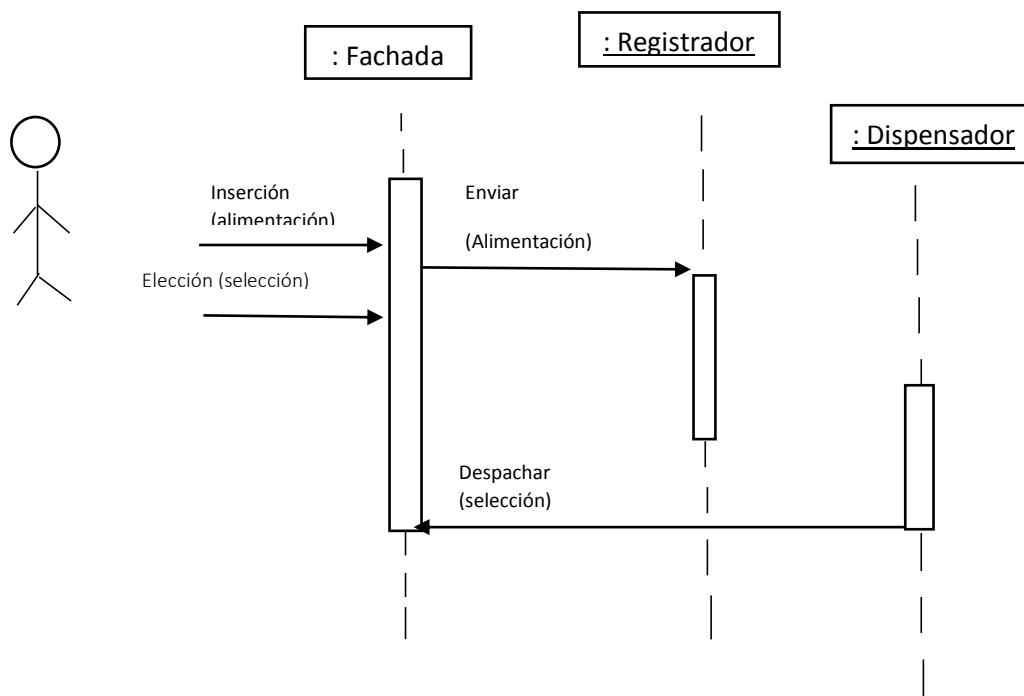


Figura 2.9

## 2.10 Un diagrama de secuencias genérico

A partir del diagrama de secuencias de instancias se puede generar el diagrama de secuencias genérico.

## 2.11 Creación de un objeto en la secuencia

Cuando una secuencia da por resultado la creación de un objeto, tal objeto se representara de la forma usual: como un rectángulo con nombre. La diferencia es que no lo colocará en la parte superior del diagrama de secuencias sino que lo colocará en la parte superior del diagrama de secuencias, sino que lo colocará junto con la dimensión vertical, de modo que su ubicación corresponda al momento que se cree. El mensaje que creara al objeto se nombrara "Crear ()". Los paréntesis implican una operación: en un lenguaje orientado a objetos, una operación constructor genera un objeto. En lugar de usar "Crear ()" o "Créate ()" para etiquetar la flecha de un mensaje de creación de un objeto, podría valerse de un estereotipo llamado <<Crear>>.

## 2.12 Como representar la cursiva

En ocasiones un objeto cuenta con una operación que se invoque a sí misma.

A esto se le conoce como recursividad, y es unas características fundamentales de varios lenguajes de programación.

Para representar esto en 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 regrese al objeto que inicio la recursividad. La figura 2.8

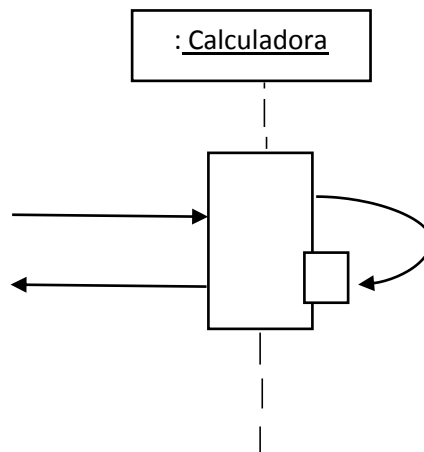


Figura 2.12

Como representar la recursividad en un diagrama de secuencias.

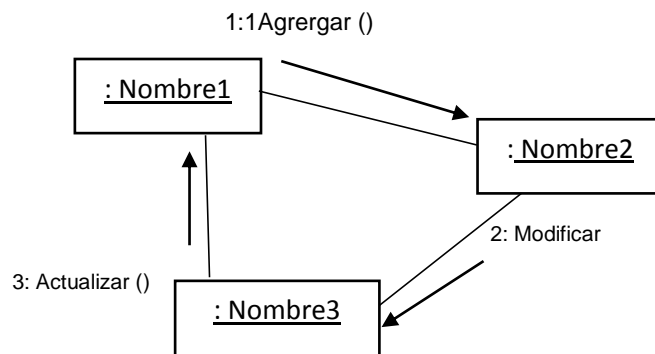
## III HORA DIEZ

### 3.1 Diagrama de colaboración

Los diagramas de colaboración muestran la forma en que los objetos colaboran entre sí, tal como sucede con un diagrama de secuencia. Muestran los objetos junto con los mensajes que se envía entre ellos. Si el diagrama de secuencia 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 muestran los mensajes que se envían los objetos entre sí.

La figura 3.1 le muestra la simbología del diagrama de colaboración.



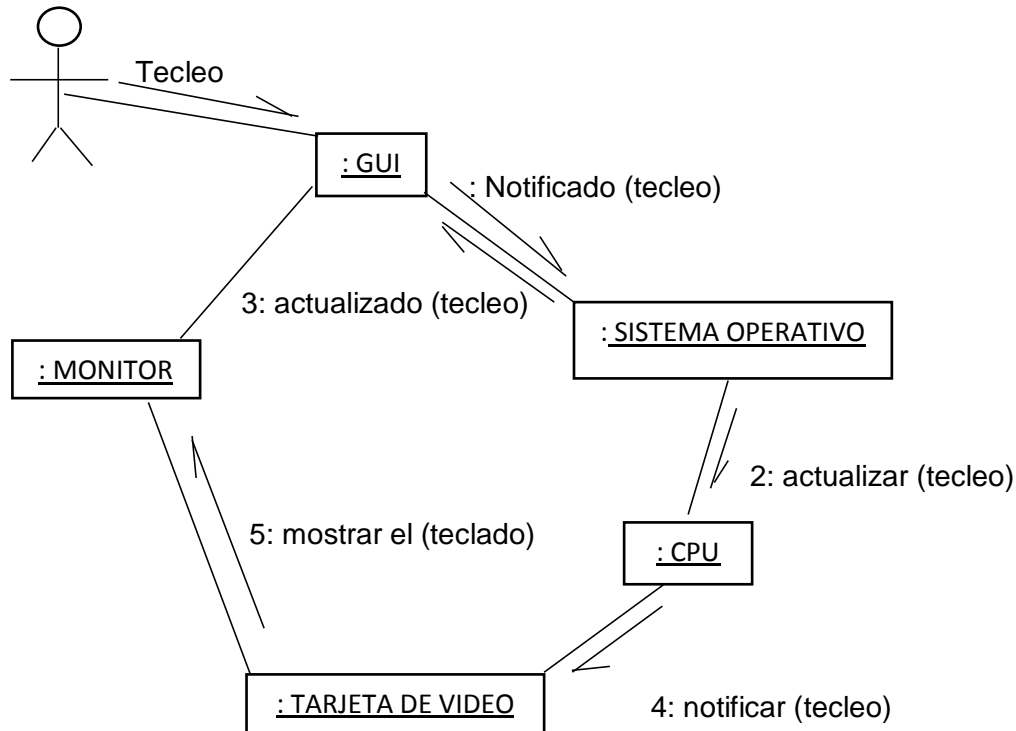
### 3.2 La GUI

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

La figura 3.2 muestra la forma de representar esta secuencia de interacción al oprimir una tecla, con lo que los mensajes ocurrirán de manera secuencial. Tal secuencia (a partir de la hora anterior).

La figura 3.2 muestra la forma de representar esta secuencia de interacciones en un diagrama de colaboración.





### 3.3 Cambios de estado

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

Un rectángulo es el objeto que indique el estado. Agregar 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 torna>>.

### 3.4 Creación de un objeto

Para mostrar la creación de objetos, volveré al caso de uso “crear propuesta” de la firma de consultaría. Una vez más, las secundarias que modelará será:

1. El consultor buscará en el área de almacenamiento centralizada de la red una propuesta adecuado en la cual basarse.
2. Si el consultor localiza una propuesta adecuada, la abrirá y en el proceso abrirá la aplicación de oficina. El consultar guardaría el archivo bajo un nuevo nombre, con lo creará un nuevo archivo para la nueva propuesta.
3. Si el consultor no encuentra una propuesta, abrirá la aplicación de oficina y genera 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 guardaría en el área de almacenamiento centralizada.

### 3.5 Algunos conceptos más

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

### 3.6 Varios objetos receptores en una clase

En ocasiones un objeto envía un mensaje de diversos objetos de la misma clase. En el diagrama de colaboraciones, la representación de los diversos objetos es una pila de rectángulo que se extiende “desde atrás”. Agregaré una condición entre corchetes precedida por asteriscos para indicar que el mensaje irá a todos los objetos.

### 3.7 Objetos activos

Los objetos activos pueden enviar mensajes a los objetos pasivos e interactuar con otros objetos activos.

Un objeto activo controla el flujo en la secuencia. se representa como un rectángulo con un borde grueso en negro.

### 3.8 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 orden debido.

## Bibliografía

schmuller, j. (2001). Aprendiendo UML 24 HORAS . Mexico .