

Instituto Tecnológico de Pabellón de Arteaga

Desarrollo, Aplicación y Consultoría de Sistemas de Información

Tics 9no.

Prof. Eduardo Flores Gallegos

Denise Kirstie Martínez Santana

14 de septiembre del 2018

HORA 1. Introducción al UML

El UML es una de las herramientas más emocionantes en el mundo actual del desarrollo, esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en forma convencional y fácil de comprender.

El UML es necesario porque al principio de la computación, los programadores no realizaban análisis muy profundos sobre el problema a resolver, actualmente es necesario contar con un plan bien analizado, la clave está en organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprenden y lo convengan con él y esta organización la proporciona UML.

El UML es la creación de Grandy Booch, James Rumbaugh e Ivan Jacobson, en 1994 Rumbaugh ingreso al Rational Software Corporation donde ya trabajaba Booch una año después ingresa Jacobson. Los anteproyectos del UML empezaron a circular en la industria del software y las reacciones resultantes trajeron consigo considerables modificaciones. Conforme diversos corporativos vieron que UML era útil a sus propósitos, se conformó un consorcio del UML. Entre los miembros se encuentran DEC, HP, Intellicorp Microsoft, Oracle, Texas Instrument y Rational.

Diagramas de clases

Es probable que muchas cosas tengan atributos y que realicen determinadas acciones. Una clase es una categoría o grupo de cosas que tienen atributos y acciones similares. Un ejemplo es una lavadora que es la clase y cuenta con atributos como marca, modelo, el número de serie y la capacidad.

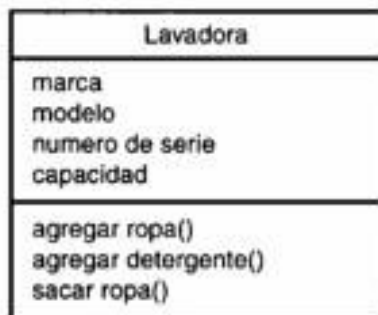


Diagrama de objetos

Un objeto es una instancia de clase, por ejemplo la lavadora podría tener la marca Laundatorium, el modelo Washmeister, el número de serie GL47774 y una capacidad de 7kg.

UML representa a un objeto en un rectángulo, con una clase, pero el nombre esta subrayado. El nombre de la instancia especifica se encuentra a la izquierda de los puntos (:), y el nombre de la clase a la derecha.

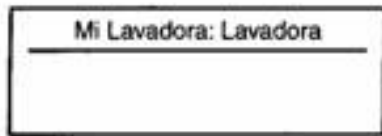


Diagrama de caso de uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario, para los programadores esta es una herramienta valiosa ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema.

Este es el modo de representar el diagrama de uso con el ejemplo de una lavadora.

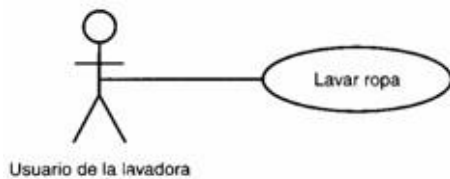


Diagrama de estados

En cualquier momento un objeto se encuentra en un estado particular, una lavadora puede estar en fase de remojo, lavado, enjuague, centrifugado o apagada.



Diagrama de secuencias

Este diagrama muestra la mecánica de la interacción con base a tiempos.

En este diagrama de secuencias se muestra las interacciones que se realizan a través del tiempo entre el abastecimiento de agua, el tambor y el drenaje, el tiempo se da de arriba hacia abajo.



Diagrama de actividades

Las actividades que ocurren dentro de un caso de uso del comportamiento de un objeto se dan en secuencia.

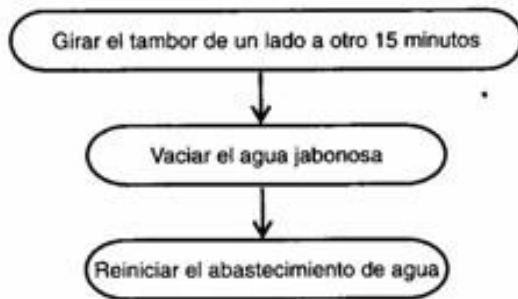


Diagrama de colaboraciones

Los elementos de un sistema trabajan en conjunto para cumplir con los objetivos del sistema.

Aquí se muestra como el agregar un cronometro interno al conjunto de clases que constituyen a una lavadora.



Diagrama de componentes

El desarrollo de software se realiza mediante componentes, importante en los procesos de desarrollo de equipo.

UML así representa un componente de software.

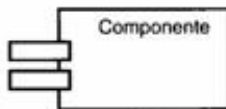
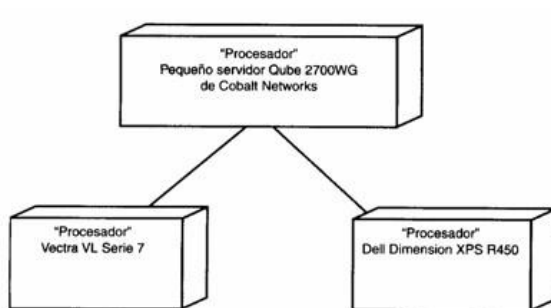


Diagrama de distribución

Muestra la arquitectura física de un sistema informático, puede representar los equipos y dispositivos, mostrar interconexiones y el software.

Cada computadora está representada por un cubo y las interacciones entre las computadoras por líneas que conectan los cubos.



Hora 2. Orientación a objetos

La orientación a objetos es tan importante para el diseño que el OMG (Grupo de administración de objetos), una corporación no lucrativa que establece las normas para el desarrollo orientado a objetos. Un objeto es una instancia de una clase.

En el mundo orientado a objetos, una clase tiene otro propósito además de la categorización, es una plantilla para fabricar objetos. La orientación a objetos se refiere a algo más que tan solo atributos y acciones, también considera otros aspectos, dichos aspectos se conocen como abstracción, herencia, polimorfismo y encapsulamiento o encapsulación, otros aspectos importantes de la orientación a objetos son el envío de mensajes, las asociaciones y la agregación.

Abstracción

Se refiere a quitar las propiedades y acciones de un objeto, para dejar solo aquellas que son diferentes.

Herencia

Un objeto tiene todas las características de la clase de la que proviene. Un objeto no solo hereda de una clase, si no que de una clase puede heredar otra clase.

Polimorfismo

Se le llama polimorfismo cuando una operación tiene el mismo nombre en diferentes clases, por ejemplo, podrá abrir una puerta, una ventana, un periódico, un regalo, pero en cada uno de estos casos se realiza una operación diferente.

Encapsulamiento

Es cuando un objeto trae consigo su funcionalidad, es cuando uno ve un objeto pero no ve las operaciones que realiza para lograrlo.

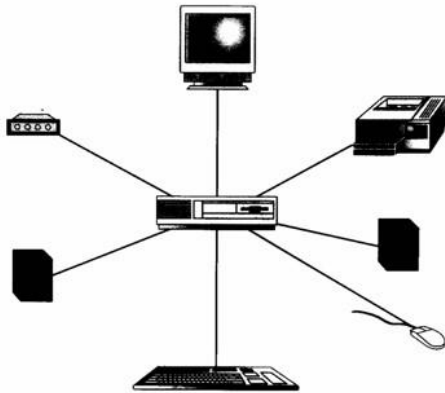
Envío de mensajes

El envío de mensajes se logra cuando en un sistema los objetos trabajan en conjunto, un objeto envía a otro mensaje para realizar una operación y el objeto receptor ejecutara la operación.

Agregación

Son los tipos de componentes que constituyen un equipo

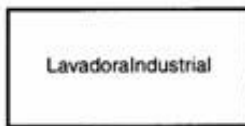
Ejemplo:



Hora 3. Uso de la orientación de objetos

Concepción de una clase

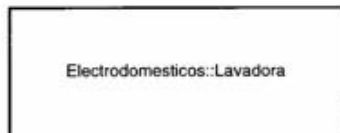
Un rectángulo representa una clase, el nombre es una palabra con la primera letra mayúscula.



UML representa un paquete como una carpeta tabular cuyo nombre es una cadena de texto.



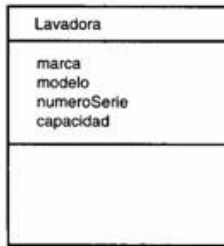
“Electrodomesticos::Lavadora”, el par de puntos separa al nombre del paquete, que está a la izquierda, del nombre de la clase que va a la derecha, a este tipo de nombre se le conoce como nombre de ruta.



Atributos

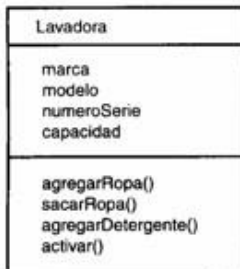
Una es propiedad o característica de una clase y describe un rango de valores que la propiedad podrá contener en los objetos de la clase, una clase podrá tener varios o ningún atributo.

Ejemplo de una clase y sus atributos.



Operaciones

Una operación es algo que la clase puede realizar o que usted puede realizar a una clase. La lista de operaciones se inicia debajo de una línea que separa a las operaciones de los atributos.



Hora 4. Uso de relaciones Asociaciones

Es una asociación cuando las clases se conectan entre sí de forma conceptual. La asociación puede funcionar de forma inversa, podrá mostrar ambas asociaciones en el mismo diagrama con un triángulo relleno que indique la dirección de cada asociación. En ocasiones una asociación entre dos clases debe seguir ciertas reglas, esta se indica al establecer una restricción junto a la línea de la asociación. Una asociación puede contener atributos y operaciones y se les llama clase de asociación.

Multiplicidad

Cuando dos clases tienen una relación uno a uno.

Asociaciones calificadas

Cuando la multiplicidad de una asociación es de uno a muchos, con frecuencia se presenta un reto muy particular: la búsqueda.

Asociaciones reflexivas

En ocasiones una clase es una asociación consigo misma. Esto puede ocurrir cuando una clase tiene objetos que pueden jugar diversos papeles.

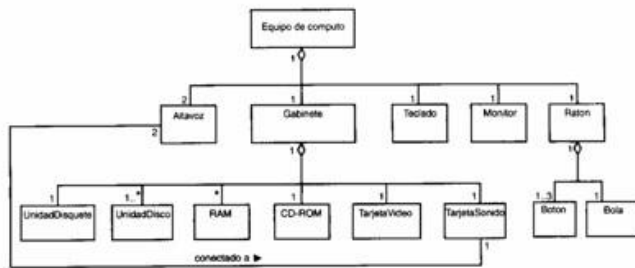
Dependencias

Una clase utiliza a otra. El uso más común de una dependencia es mostrar que una firma en la operación de una clase utiliza a otra clase.

Hora 5. Agregación, Composición, interfaces y realización.

Agregación

En ocasiones una clase consta de otras clases, los componentes y la clase que contienen son una asociación que conforma un todo. Puede representar una agregación como una jerarquía dentro de la clase completa en la parte superior, y los componentes por debajo de ella.



Composiciones

Una composición es un tipo muy representativo de una agregación. Cada componente dentro de una composición puede pertenecer tan solo a un todo. Los componentes de una mesa de café establecen una composición.

Contextos

Enfoca la atención en una clase específica dentro de un sistema. Es como un mapa detallado de un mapa mayor.

Visibilidad

Se aplica a atributos u operaciones y establece la proporción en que otras clases podrán utilizar los atributos y operaciones de una clase dada. Existen tres niveles de visibilidad: Nivel público, en el cual la funcionalidad se extiende a otras clases. En el nivel protegido la funcionalidad se otorga sólo a las clases que se heredan de la clase original. En el nivel privado sólo la clase original puede utilizar el atributo u operación.

Alcance

Existen dos tipos de alcances: el de instancia y el de archivo. En el primero cada instancia cuenta con su propio valor en un atributo u operación. En el alcance archivado, sólo habrá un valor de atributo u operación en todas las instancias de la clase.

Hora 6. Introducción a los casos de uso

El caso de uso es una estructura que ayuda a los analistas a trabajar con los usuarios para determinar la forma en que se usará un sistema.

El caso de uso es una excelente herramienta para estimular a que los usuarios potenciales hablen, de un sistema, desde sus propios puntos de vista. La idea es involucrar a los usuarios en las etapas iniciales del análisis y diseño del sistema, esto aumenta la probabilidad de que el sistema sea de mayor provecho para la gente a la que supuestamente ayudará.

Un caso de uso establece un conjunto de escenarios para realizar algo útil para un actor. Un ejemplo de un caso de uso es comprar una gaseosa.



Inclusión de los casos de uso

Se le conoce como inclusión de un caso también a usar un caso de uso. En el término incluir existen dos ventajas, una es más clara: los pasos en un caso de uso, incluyen los de otro, la segunda es evitar la confusión potencial de las palabras usar y uso en un contexto estrecho, así no tenemos que decir “promover al uso mediante el uso reiterativo de un caso de uso”.

Extensión de los casos de uso

Agregamos pasos al caso de uso que ya existe.

Inicio de análisis de un caso de uso

Para iniciar un caso de uso es necesario empezar con entrevistas a los clientes, que lleven los resultados a los diagramas iniciales de clases, en las entrevistas se les pide a los usuarios que indiquen todo lo que ellos harían con el sistema que cada quien diseñará, los casos de uso ayudarán con el diseño.

Hora 7. Diagramas de casos de uso

El caso de uso es muy poderoso, pero lo es aún más cuando se visualiza por medio de UML, una de las dinámicas del proceso de análisis de un sistema es generar una colección de casos de uso.

Representación de un modelo de caso de uso

Uno de los beneficios de un caso de uso es que le muestra los confines entre el sistema y el mundo exterior, los actores están fuera del sistema, mientras que en el caso de uso están dentro de él.

En un modelo de caso de uso, una figura agregada representa a un actor, una elipse a un caso de uso y una línea asociativa representa la comunicación entre actor y caso de uso.



Concepción de las relaciones entre casos de uso

La generalización cuenta con un caso de uso que se hereda de otro, el agrupamiento es una manera sencilla de organizar los casos de uso.

Modelo de caso de uso en la máquina de gaseosas con inclusión.

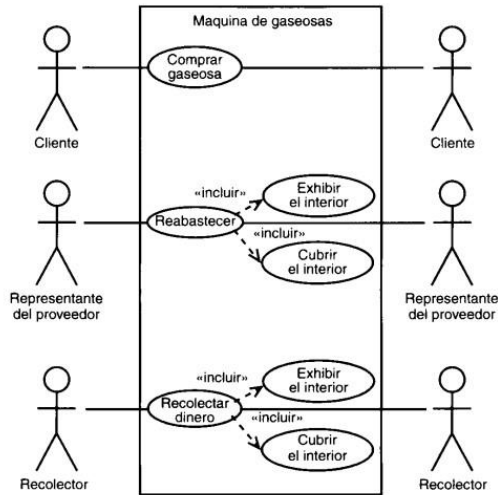
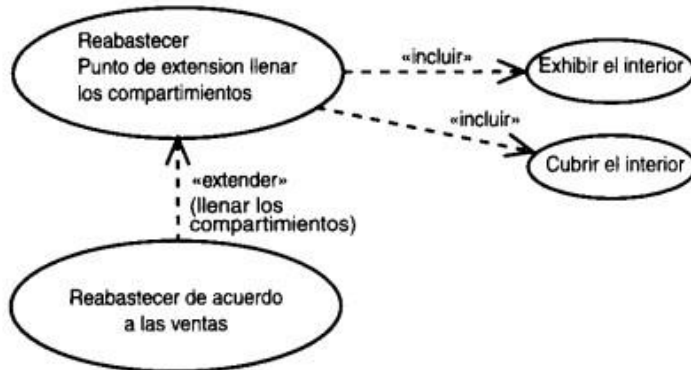
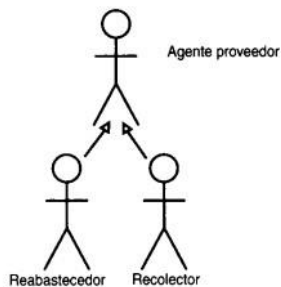


Diagrama de casos de uso que muestra la extensión y la inclusión



Como las clases y los casos de uso, los actores pueden estar en una relación de generalización.



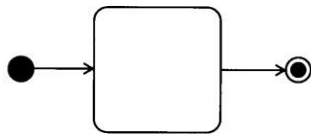
Diagramas de casos de uso en el proceso de análisis

Las entrevistas al cliente deberán iniciar el proceso, estas entrevistas producirán diagramas de clases que fungirán como las bases de su conocimiento para el dominio del sistema, las entrevistas comienzan en la terminología del dominio, los resultados iniciales de las entrevistas deberán revelar a los actores y casos de uso de alto nivel que describirán los requerimientos funcionales.

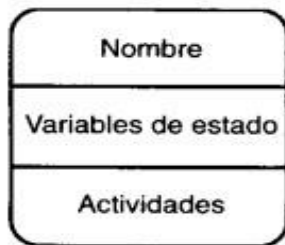
Hora 8. Diagramas de estados

Presenta los estados en los que puede encontrarse un objeto junto con las transacciones entre los estados, y muestra los puntos inicial y final de una secuencia de cambios de estado, un diagrama de estado también es conocido como un motor de estado. Un diagrama de estados muestra las condiciones de un solo objeto.

El icono para 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 una secuencia de estados, y una diana representada al punto final.



Puede subdividir el símbolo del estado en áreas que muestren el nombre, variables y actividades del estado.



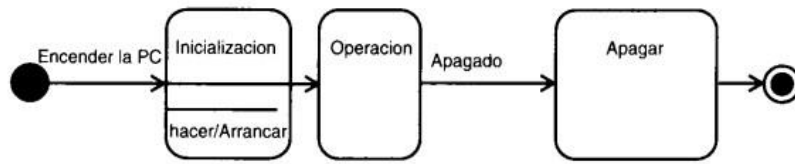
Sucesos y acciones

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. La GUI (interfaz gráfica de usuario) con que interactúe le dará ejemplos de detalles de la transición. Utilizaremos tres estados de GUI:

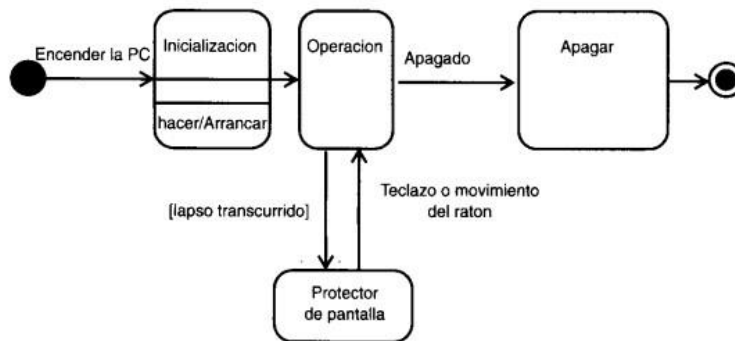
Inicialización

Operación

Apagar

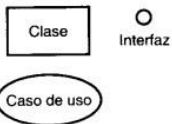


El diagrama de estados para la GUI, con el estado protector de pantalla y la condición de seguridad



Elementos de comportamiento: el diagrama de estados. Los organización del UML, en términos de los elementos que ha utilizado hasta ahora.

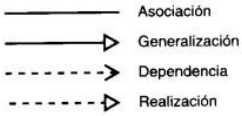
Elementos estructurales



Elementos de comportamiento



Relaciones



Agrupación



Extensión

«Estereotipo»
{Restricción}
{valor etiquetado}

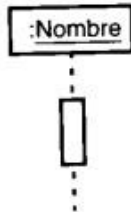
Anotación



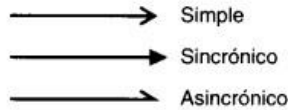
Hora 9. Diagramas de secuencias

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.

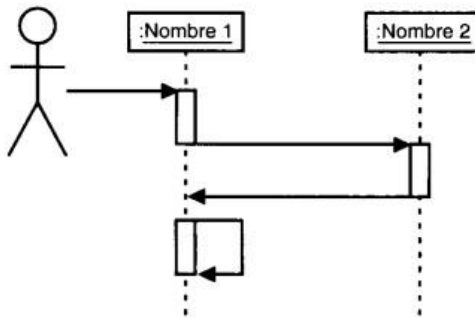
Representación de un objeto en un diagrama de secuencias



Simbolos para los mensajes en un diagrama de secuencias



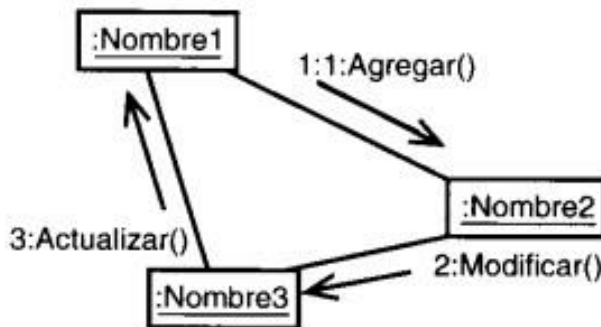
En un diagrama de secuencias los objetos se colocan de izquierda a derecha en la parte superior. Cada línea de vida de un objeto es una línea discontinua que se desplaza hacia abajo del objeto. Una línea continua con una punta de flecha conecta a una línea de vida con otra, y reprenta un mensaje de un objeto a otro. El tiempo se inicia en la parte superior y continúa hacia abajo. Aunque un actor el que normalmente inicia la secuencia, su símbolo de diagrama de secuancias.



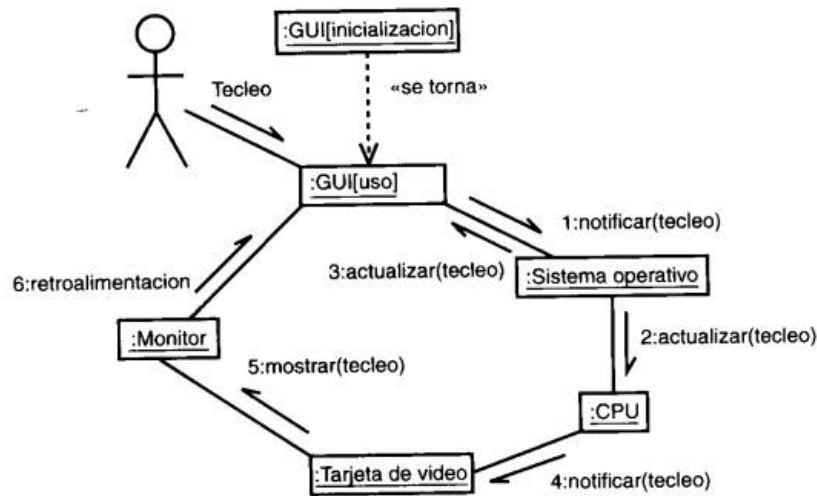
Hora 10. Diagramas de colaboraciones

Un diagrama de colaboraciones es una extensión de uno de objetos. Además de las realciones entre objetos, el diagrama de colaboraciones muestra los mensajes que se envían los objetos entre si. Por lo general, evitará la multiplicidad dado que podría ser funte de confusión.

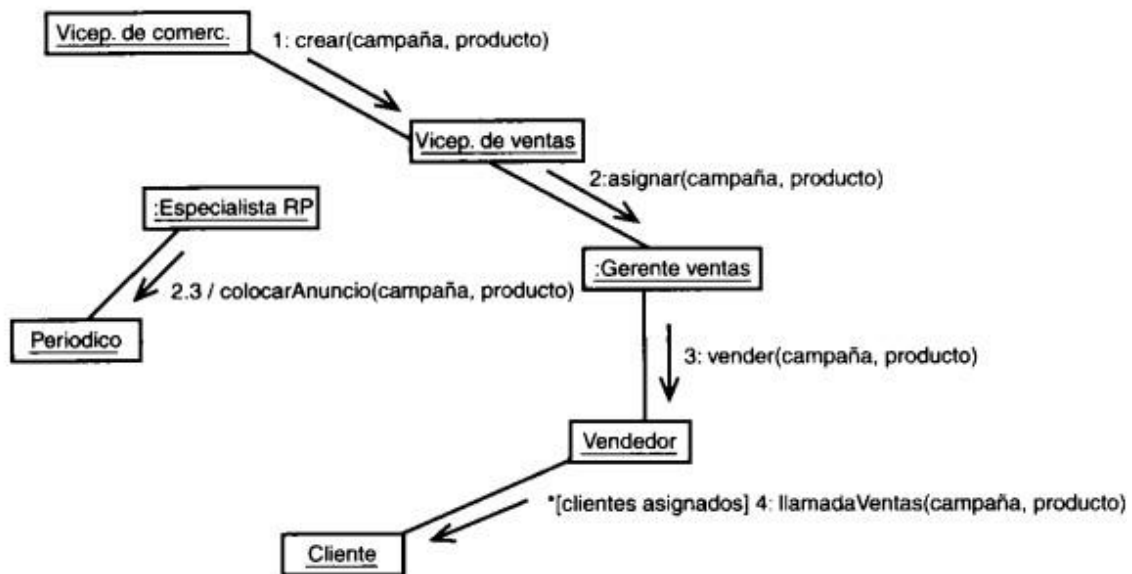
Ejemplo de un diagram de colaboraciones



Un diagrama de colaboraciones puede incorporar camnios de estado.



La sincronización de mensajes en un diagrama colaboraciones

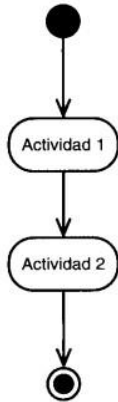


Los diagramas de colaboraciones le permiten modelar varios objetos receptores en una clase, ya sea que los objetos reciban o no los mensajes en un orden específico. También podrá representar objetos activos que controlen el flujo de los mensajes, así como los mensajes que se sincronizan con otros.

Hora 11. Diagramas de actividades

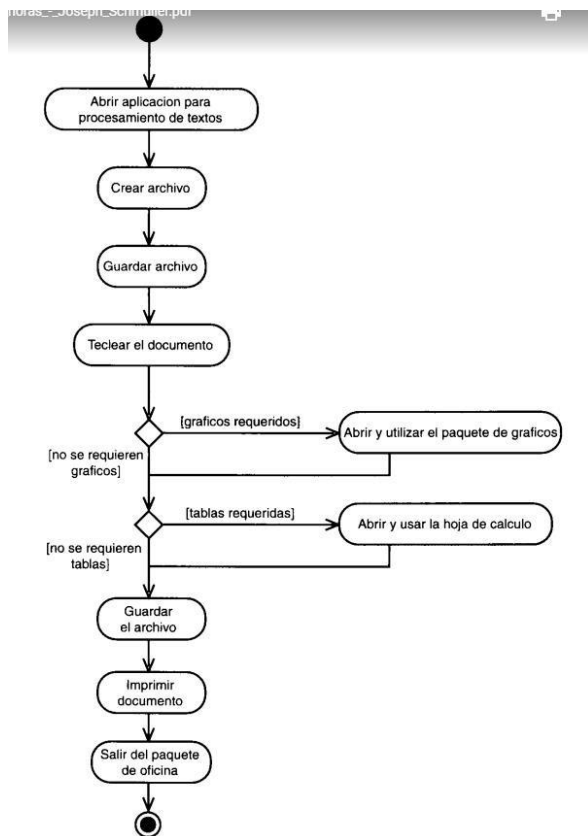
Fue diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o proceso. Es una extensión de diagramas de estados, el diagrama de actividades resalta, precisamente, a las actividades.

Muestra el punto inicial y final, así como dos actividades y una transición



Aplicación de los diagramas de actividades

Un diagrama de actividades para el proceso de creación de un documento



El diagrama de actividades agrega la dimensión de visualizar responsabilidades. Separa el diagrama en segmentos paralelos conocidos como marcos de responsabilidad. Cada marco de responsabilidad muestra el nombre de un responsable en la parte superior, y representa las actividades de cada uno.

Hora 12. Diagramas de Componentes

Un componente de software es una parte física de un sistema, y se encuentra en una computadora, no en la mente del analista, un componente se puede tomar como una tabla, archivo de datos, ejecutable, biblioteca de vínculos dinámicos, documentos.

Sólo podrá ejecutar las operaciones de un componente a través de su interfaz, se le conoce como realización a la relación entre un componente.

Un componente puede hacer disponible su interfaz para que otros componentes puedan utilizar las operaciones que contiene, un componente puede acceder a los servicios de otro componente.

Existen tres tipos de componentes:

1. Componentes de distribución, que conforman el fundamento de los sistemas ejecutables por ejemplo DLL, ejecutables, controles ActiveX y Java Beans.
2. Componentes para trabajar en el producto, a partir de los cuales se han creado los componentes de distribución como archivos de base de datos y código.
3. Componentes de ejecución, creados como resultado de un sistema de ejecución.

Un diagrama de componentes contiene, componentes, interfaces y relaciones, también pueden aparecer otros tipos de símbolos que ya haya visto.

Bibliografía

- [1] J. Schuller, Aprendiendo UML en 24 horas, Naucalpan de Juárez, edo. de México: Prentice Hall.