

UML: Lenguaje de Modelado Unificado

Rocío Segovia

MSC. Ingeniería de Sistemas



Qué es UML?

- El Lenguaje Unificado de Modelado (**Unified Modeling Language UML**), es un **lenguaje estándar** para escribir planos de software.
- UML prescribe un **conjunto de notaciones y diagramas** estándar para modelar sistemas **orientados a objetos**.
- Empezó como una consolidación del trabajo de **Grade Booch, James Rumbaugh, e Ivar Jacobson**, creadores de tres de las **metodologías orientadas a objetos** más **populares**.
- El Object Management Group (OMG), publicó una petición con propósito de un metamodelo orientado estándar. UML, en su versión 1.0, fue una respuesta en enero de 1997.

Creadores de UML



Grady Booch

Es un diseñador de software, un metodologista de software y entusiasta de diseño de patrones. Es director científico de Rational Software y editor de una serie de Benjamin/Cummings. (http://es.wikipedia.org/wiki/Grady_Booch)

Fecha de nacimiento: 27 de febrero de 1955 (edad 58), Texas, Estados Unidos

Educación: Universidad de California en Santa Bárbara, Academia de la



Ivar Jacobson

Ingeniero sueco en Ciencias de la computación. Inventó el diagrama de secuencia y desarrolló los diagramas de colaboración. (http://es.wikipedia.org/wiki/Ivar_Jacobson)

Fecha de nacimiento: 2 de septiembre de 1939 (edad 73), Ystad, Suecia

Educación: Real Instituto de Tecnología, Chalmers University of Technology.



James Rumbaugh

Científico de la computación y un metodologista de objeto. Es mejor conocido por su trabajo en la creación de la Técnica de Modelado de Objetos y el Lenguaje Unificado de Modelado. http://es.wikipedia.org/wiki/James_Rumbaugh

Fecha de nacimiento: 24 de septiembre de 1947 (edad 65), Bethlehem, Pensilvania, Estados Unidos

Educación: Instituto Tecnológico de Massachusetts, Instituto de Tecnología de California

Conceptos generales

- **Modelo:**

Un modelo representa a un sistema software desde una perspectiva específica. Al igual que la planta y el alzado de una figura en dibujo técnico nos muestran la misma figura vista desde distintos ángulos, cada modelo nos permite fijarnos en un aspecto distinto del sistema.

- **Diagrama:**

Un diagrama es una presentación gráfica de un conjunto de elementos, que la mayoría de las veces se dibuja como un conjunto de elementos relacionados. Los diagramas se utilizan para representar un sistema desde diferentes perspectivas.

Modelos UML

Con el modelado de sistemas se obtienen diferentes tipos de vistas, las vistas estáticas de los sistemas en UML se representan con 4 tipos de diagramas que se describen a continuación:

- **Diagramas Estructurales**
 - Diagramas de Clases
 - Diagramas de Objetos
 - Diagramas de Componentes
 - Diagramas de despliegue
- **Diagramas de Comportamiento**
 - Diagramas de casos de uso
 - Diagramas de Secuencia
 - Diagramas de colaboración
 - Diagramas de estado
 - Diagrama de Actividades

UML Define los siguientes diagramas:

- **Diagramas de clases:**

Muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones, son los más comunes en el modelo orientado a objetos y cubren las vistas de diseño estático de un sistema.

- **Diagramas de Objetos:**

Muestra un conjunto de objetos y sus relaciones, representan instancias de los elementos encontrados en los diagramas de clases, representado cubren las vistas de diseño estático de un sistema desde la perspectiva de casos reales.

- **Diagrama de Actividades:**

Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema, cubren la vista dinámica de un sistema. Son especialmente importantes al modelar el funcionamiento de un sistema y resaltan el flujo de control de objetos.

UML Define los siguientes diagramas:

- **Diagrama de Componentes:**

Muestra la organización y las dependencias entre un conjunto de componentes, cubren la vista estática de un sistema.

- **Diagrama de Despliegue:**

Muestra la configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos, cubren la vista de despliegue estática de una arquitectura.

- **Diagrama de Casos de Uso:**

Muestra un conjunto de casos de uso, actores y relaciones, cubren la vista de casos de uso estática de un sistema.

UML Define los siguientes diagramas:

- **Diagrama de Secuencia:**

Es un diagrama de interacción que resalta la ordenación temporal de los mensajes.

- **Diagrama de Colaboración:**

Es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensaje.

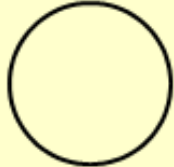
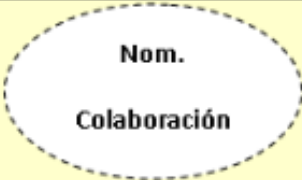
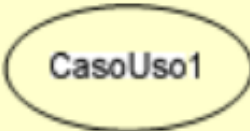
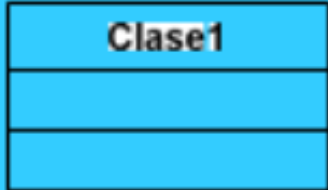
- **Diagrama de Estados:**

Muestra una máquina de estados, que consta de estados, transiciones, eventos y actividades, cubren la vista dinámica de un sistema, resaltan el comportamiento dirigido por eventos de un objeto.


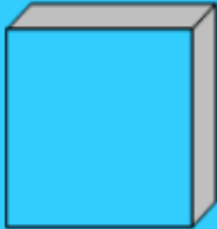
Elementos UML

- Hay 4 tipos de elementos en UML:
 - Elementos estructurales
 - Elementos de comportamiento
 - Elementos de agrupación
 - Elementos de anotación.
- Los elementos estructurales como (clases, interfaces, colaboraciones, casos de uso, clases activas, componentes y nodos) son los elementos básicos que se pueden incluir en el modelo UML. También existen variaciones de estos siete elementos, tales como actores, señales, procesos, hilos y aplicaciones, documentos, archivos, bibliotecas, páginas y tablas.

Elementos estructurales

Nombre	Descripción	Símbolo
Interfaz	Es una colección de operaciones que especifican un servicio de una clase o componente, representa el comportamiento completo de una clase o componente.	
Colaboración	Se definen como una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos las colaboración representan la implementación de patrones que forman un sistema.	
Caso de Uso	Es una descripción de secuencias de acciones que un sistema ejecuta y que produce un resultado observable de interés para un actor particular. Se utiliza para estructurar el comportamiento en un modelo.	
Clase Activa	Es un objeto que tienen uno o más procesos o hilos de ejecución y por lo tanto pueden dar origen a actividades de control. Son iguales a las clases excepto en que sus objetos representan a elementos cuyo comportamiento es concurrente con otros elementos.	

Elementos estructurales

Nombre	Descripción	Símbolo
Componente	<p>Es una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto.</p> <p>Representa típicamente el empaquetamiento físico de diferentes elementos lógicos, como clases interfaces y colaboraciones.</p>	
Nodo	<p>Es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que dispone de memoria y con frecuencia capacidad de procesamiento.</p>	 <p>Nono 1.</p>

Elementos de Comportamiento

Son las partes dinámicas de los modelos UML. Estos son los verbos de un modelo y representan comportamiento en el tiempo y en el espacio.

- **Interacción:**

Es un comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular para alcanzar un propósito específico.

- **Máquina de Estados:**

Es un comportamiento que especifica las secuencias de estados por las que pasa un objeto o una interacción durante su vida en respuesta a eventos.

Estos elementos están conectados normalmente a diversos elementos estructurales, principalmente clases, colaboraciones y objetos.

Elementos de agrupación

Son las partes organizativas de los modelos UML. Estos son las cajas en las que pueden descomponerse un modelo.

- **Paquete:**

Es un mecanismo de propósito general para organizar elementos en grupos. En los paquetes se pueden agrupar los elementos estructurales, de comportamiento e incluso otros elementos de agrupación

Los paquetes son los elementos de agrupación básicos con los cuales se puede organizar un modelo UML. También hay variaciones, tales como los framework, los modelos y los subsistemas.

- **Relaciones:**

Existen cuatro tipos de la relaciones en UML

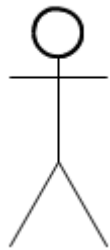
- Dependencias
- Asociaciones
- Generalizaciones
- Realización

Diagramas de Casos de Uso

- El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). Un diagrama de casos de uso consta de los siguientes elementos:
 - Actor.
 - Casos de Uso.
 - Relaciones de Uso, Herencia y Comunicación.

Diagramas de Casos de Uso

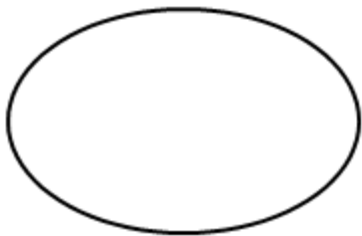
Actor



- Actor

Una definición previa, es que un **Actor** es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra **rol**, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Caso de Uso


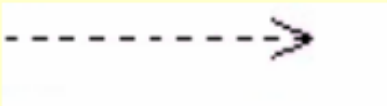
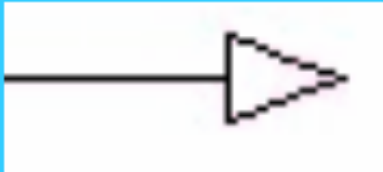


- Caso de Uso

Es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

Diagramas de Casos de Uso

- Relaciones que se utilizan en el diagrama de casos de uso:

Asociación	Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.	
Dependencia o Instanciación	Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.	
Generalización	<p>Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>).</p> <p>Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).</p>	
extends	Se recomienda utilizar cuando un caso de uso es similar a otro (características).	<<extends>>
Uses	Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.	<<uses>>

Tipos de Relaciones

- **Dependencia:** es una relación semántica entre dos elementos, en la cual un cambio a un elemento puede afectar el significado de otro elemento.
- **Asociaciones:** es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objeto y representan una relación estructural entre un todo y sus partes.
- **Generalizaciones:** representan un elemento que incluye características y comportamiento similar entre varios elementos.

Tipos de Relaciones

- **Extensión:** representa cuando un elemento tiene comportamientos que han sido generalizados para incluirse en varias operaciones.
- **Inclusión:** representa un elemento que es parte de otro y se requiere definir a parte debido a su complejidad.
- **Uso:** representa la relación entre varios elementos donde el uno requiere del otro para cumplir su objetivo pero no hace parte el uno del otro.

Como se construye un Diagrama de Casos de Uso

Para construir el Modelo de Casos de Uso en la fase de Planificación y especificación de Requisitos se siguen los siguientes pasos:

- Después de listar las funciones del sistema, se definen los límites del sistema y se identifican los actores y los casos de uso.
- Se escriben todos los casos de uso en el formato de alto nivel. Se categorizan como primarios, secundarios u opcionales.
- Se dibuja el Diagrama de Casos de Uso.
- Se detallan relaciones entre casos de uso, en caso de ser necesarias, y se ilustran tales relaciones en el Diagrama de Casos de Uso.
- Los casos de uso más críticos, importantes y que conllevan un mayor riesgo, se describen en el formato expandido esencial. Se deja la definición en formato expandido esencial del resto de casos de uso para cuando sean tratados en posteriores ciclos de desarrollo, para no tratar toda la complejidad del problema de una sola vez.
- Se crean casos de uso reales sólo cuando:
 - Descripciones más detalladas ayudan significativamente a incrementar la comprensión del problema.
 - El cliente pide que los procesos se describan de esta forma.
- Ordenar según prioridad los casos de uso (este paso se va a ver a continuación).

Ejemplo de Diagrama de Casos de Uso

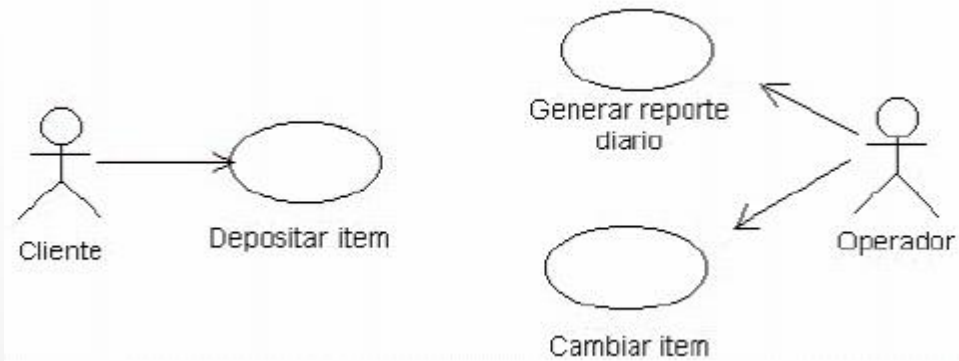
- Sistema que controla la máquina de reciclamiento de botellas, tarros y jabas. El sistema debe controlar y/o aceptar lo siguiente:
- **Registrar el número de ítems ingresados.**
- **Imprimir un recibo cuando el usuario lo solicita: El usuario/cliente presiona el botón de comienzo**
 - Describe lo depositado
 - El valor de cada ítem
 - Total
- **Existe un operador que desea saber lo siguiente:**
 - Cuantos ítems han sido retornados en el día.
 - Al final de cada día el operador solicita un resumen de todo lo depositado en el día.
- **El operador debe además poder cambiar:**
 - Información asociada a ítems.
 - Dar una alarma en el caso de que:
 - Ítem se atora.
 - No hay más papel.

Ejemplo de Diagrama de Casos de Uso

- Como una primera aproximación identificamos a los actores que interactúan con el sistema:

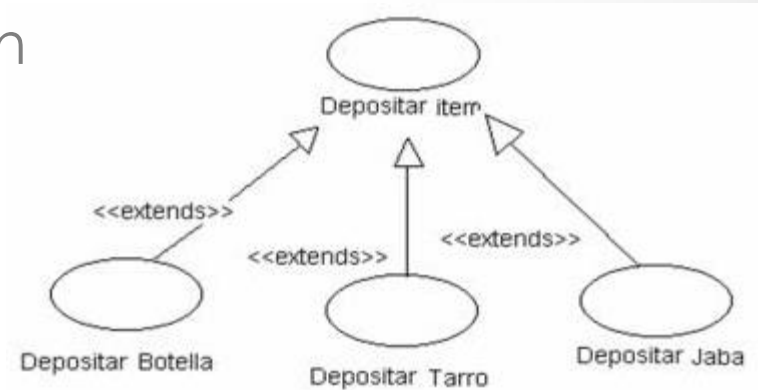


- Luego, tenemos que un Cliente puede Depositar Ítems y un Operador puede cambiar la información de un Ítem o bien puede Imprimir un informe:

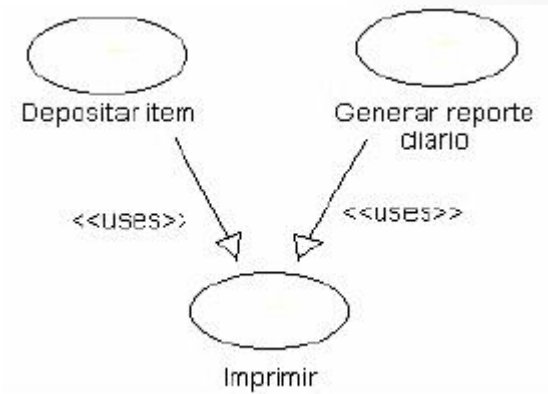


Ejemplo de Diagrama de Casos de Uso

- Además podemos notar que un ítem puede ser una Botella, un Tarro o una Jaba.



- Otro aspecto es la impresión de comprobantes, que puede ser realizada después de depositar algún ítem por un cliente o bien puede ser realizada a petición de un operador.



Ejemplo de Diagrama de Casos de Uso

- Entonces, el diseño completo del diagrama casos de uso es:

