

# Diseño de Aplicaciones

El modelado en el desarrollo de Software

# Índice

## El modelado en el desarrollo de software

- ¿Qué es un modelo?
- Principios de modelado.

## Lenguaje de Modelado Unificado (UML)

- Presentación de los diagramas de UML.
- Diagrama de Caso de uso.
- Componentes del Diagrama de Caso de uso.
- Casos de Uso concretos y abstractos.
- Descripción del Caso de Uso.

# El modelado en el desarrollo de Software

**El producto principal del trabajo de un equipo de desarrollo es un software que satisfaga las necesidades de sus usuarios.**

**Para desarrollar software de calidad, debemos definir una arquitectura sólida que le de soporte.**

**Para desarrollar software de un forma rápida y eficiente, hay que contar con un equipo de desarrollo adecuado, herramientas y el enfoque apropiado.**

# ¿Qué es un modelo?

Un modelo es una simplificación de la realidad.

Un modelo proporciona los planos del sistema.

Todo sistema debe ser descrito desde distintas perspectivas utilizando distintos modelos.

Nosotros utilizaremos modelos de estructura y de comportamiento.

# La importancia de modelar

**Construimos modelos para comprender mejor el sistema que estamos desarrollando.**

**Ayudan a visualizar cómo es un sistema.**

**Proporcionan plantillas que nos guían en la construcción de un sistema.**

**Documentan las decisiones que hemos adoptado.**

**Son útiles siempre, no sólo para sistemas grandes.**

# Principios de modelado

La elección de los modelos impacta sobre cómo se soluciona un problema.

Todo modelo puede ser expresado a diferentes niveles de precisión.

Los mejores modelos están ligados a la realidad.

Un único modelo no es suficiente.

Un modelo es una abstracción incompleta de la realidad.

# Principios de modelado

El modelado está sujeto a diferentes interpretaciones de la realidad.

La realidad es cambiante, por lo que el modelo debe cambiar en consecuencia.

El no comprender las notaciones o el proceso puede volver inútil al modelo frente al programador.

La formalidad en el modelado es necesaria para su éxito.

Los procesos juegan un rol significativo en dirigir las actividades de modelado.

# Principios de modelado

El modelado está sujeto a diferentes interpretaciones de la realidad.

La realidad es cambiante, por lo que el modelo debe cambiar en consecuencia.

El no comprender las notaciones o el proceso puede volver inútil al modelo frente al programador.

La formalidad en el modelado es necesaria para su éxito.

Los procesos juegan un rol significativo en dirigir las actividades de modelado.



# Lenguaje de Modelado Unificado (UML).

Es un lenguaje de modelado, de propósito general, usado para la visualización, especificación, construcción y documentación de sistemas Orientados a Objetos.

Lenguaje: cuenta con una sintaxis y una semántica.

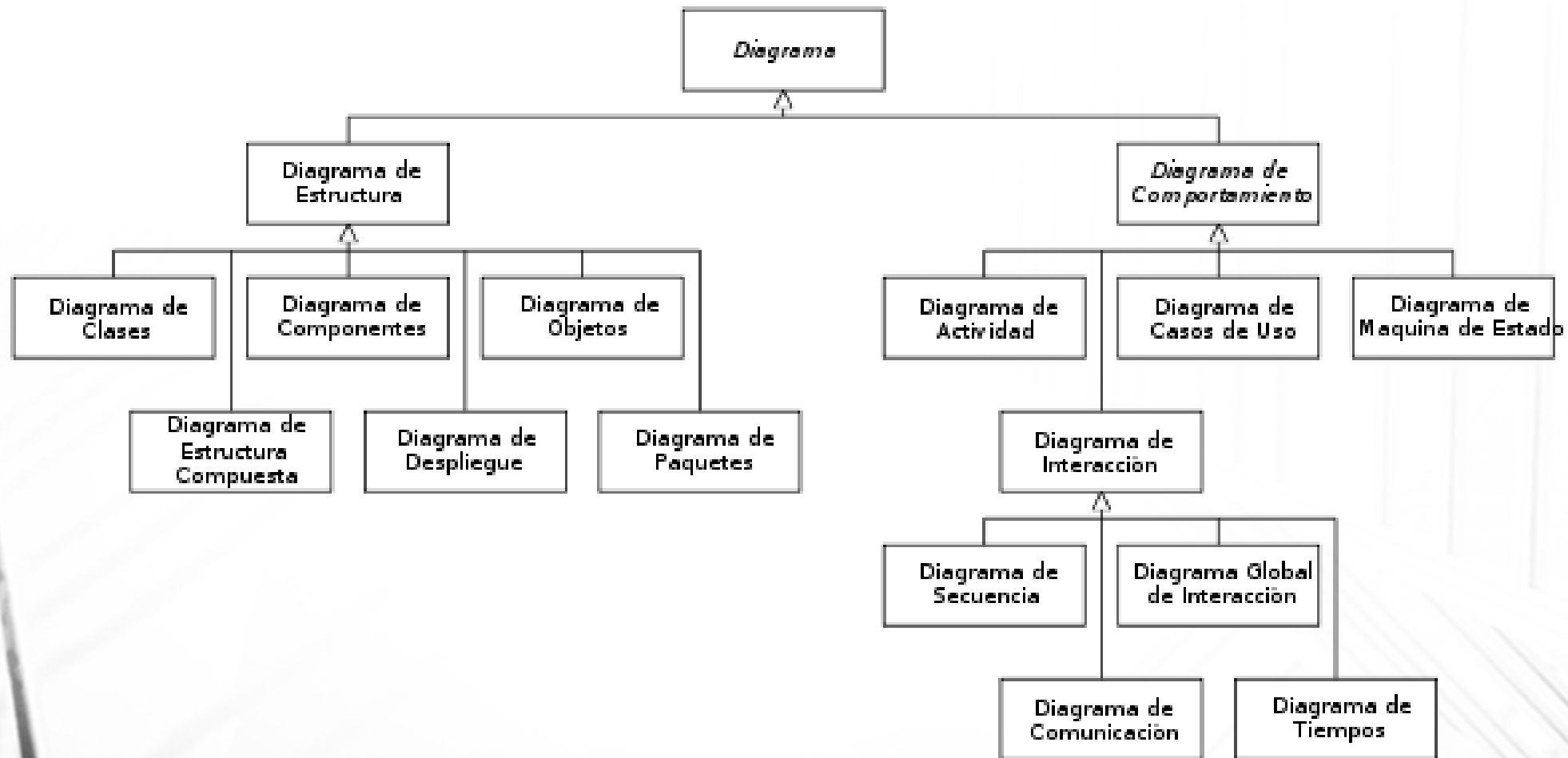
Modelado: Se modelan distintos aspectos del mundo real.

Unificado: unifica varias técnicas de modelado.

UML da soporte al desarrollo de software pero no especifica qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de los sistemas que se modelan.

# Diagramas de UML.



# Lenguaje de Modelado Unificado (UML).

## DIAGRAMAS DE ESTRUCTURA:

### 1. Diagrama de Clases

Muestra una colección de elementos de modelado, tales como clases, tipos y sus contenidos y relaciones.

### 2. Diagrama de Componentes

Muestra los componentes, sus relaciones, interacciones y sus interfaces públicas. Un componente es una pieza de código de cualquier tipo.

### 3. Diagrama de Objetos

Presenta los objetos y sus relaciones en momento del tiempo.

# Lenguaje de Modelado Unificado (UML).

## DIAGRAMAS DE ESTRUCTURA:

### 4. Diagrama de Estructura Compuesta

Representa la estructura interna de un clasificador (clase, componente o CU), y los puntos de interacción con otras partes del sistema.

### 5. Diagrama de Despliegue

Muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos.

### 6. Diagrama de Paquetes

Presenta cómo se organizan los elementos de modelado en paquetes y las dependencias entre ellos.

# Lenguaje de Modelado Unificado (UML).

## DIAGRAMAS DE COMPORTAMIENTO:

### 1. Diagrama de Casos de Uso

Un diagrama que muestra las relaciones entre los actores (que representan el ambiente de interés para el sistema), y los casos de uso.

### 2. Diagrama de Máquinas de Estado

Un diagrama que ilustra cómo los objetos de una clase, se puede mover entre estados que clasifican su comportamiento, de acuerdo con eventos disparadores de transiciones.

### 3. Diagrama de Actividades

Representa los procesos de negocios de alto nivel, incluidos el flujo de datos.

# Lenguaje de Modelado Unificado (UML).

## DIAGRAMAS DE COMPORTAMIENTO:

### 4. Diagrama de Comunicación

Enfoca la interacción entre líneas de vida y el pasaje de mensajes.

### 5. Diagrama de Secuencias

Representa una interacción, con foco en la secuencia de los mensajes que se intercambian, junto con sus correspondientes ocurrencias de eventos en las Líneas de Vida.

### 6. Diagrama de Tiempos

Muestra el cambio de estado de un objeto a lo largo del tiempo, en respuesta a los eventos o estímulos aceptados.

# Lenguaje de Modelado Unificado (UML).

## DIAGRAMAS DE COMPORTAMIENTO:

### 7. Diagrama de Descripción de Interacción.

Enfocan la revisión del flujo de control, donde los nodos son Interacciones u Ocurrencias de Interacciones. Las Líneas de Vida los Mensajes no aparecen en este nivel de revisión.

# Diagrama de Casos de Uso (CU).

**Muestra a los actores involucrados en un sistema, las funciones presentes en el sistema y cómo éstas interactúan con dichos actores.**

**Muestra actores, casos de uso, paquetes de casos de uso y sus relaciones.**



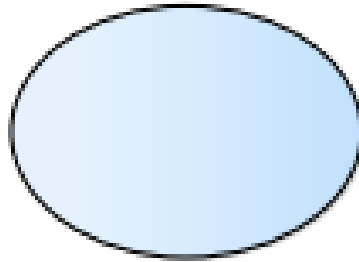
# Componentes del Diagrama de Casos de Uso.

La funcionalidad del sistema es definida por diferentes casos de uso, cada uno representa un flujo de eventos específico.

La descripción de un caso de uso define qué ocurre en el sistema cuando el caso de uso es ejecutado.

La colección de casos de uso constituye todas las maneras posibles de usar el sistema.

# Componentes del Diagrama de Casos de Uso.



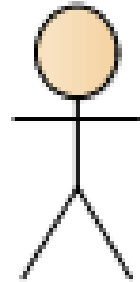
Nombre del Caso de Uso

Deberá indicar lo que se alcanza por su interacción con el actor.  
Es una frase verbal, que inicia con un verbo en infinitivo.  
No puede haber dos casos de uso con el mismo nombre.

Ejemplos:

- Registrar venta
- Generar comprobante.

# Componentes del Diagrama de Casos de Uso.



**Nombre del Actor**

**Representa un rol**

**Es cualquier cosa que intercambia datos con el sistema, puede ser un usuario, hardware u otro sistema (software).**

**Varios usuarios pueden jugar el mismo rol.**

**En tal caso, cada usuario constituye una instancia de actor.**

**El mismo usuario puede tomar diferentes roles.**

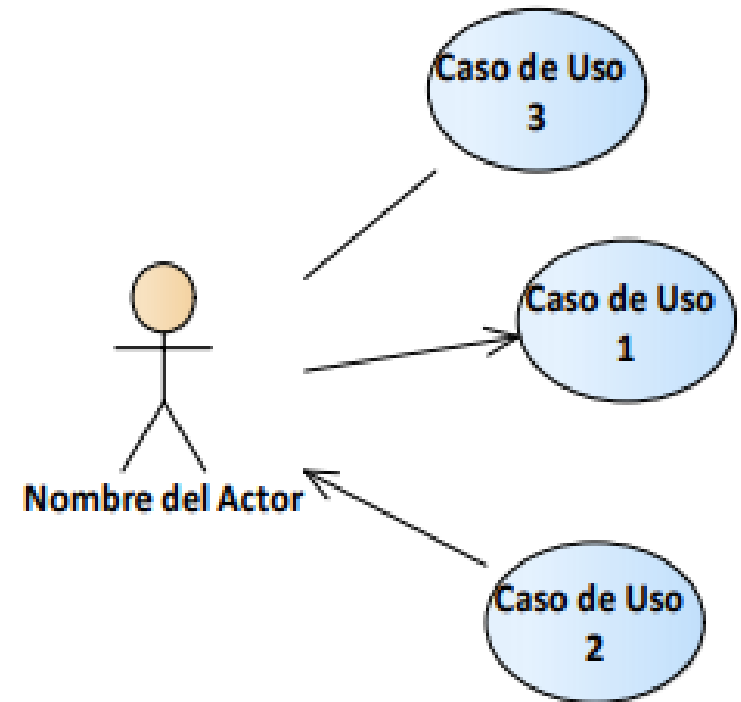
# Componentes del Diagrama de Casos de Uso.

## Asociación

Es una relación que vincula a un actor con un caso de uso.

La asociación es una relación que puede plantearse:

- **Actor-Caso de Uso 1:** esta navegabilidad se utiliza para indicar que es el actor el que inicia la relación de interacción con el caso de uso.
- **Actor-Caso de Uso 2:** esta navegabilidad se utiliza para indicar que es el Caso de Uso el que inicia la relación de interacción con el Actor.
- **Actor-Caso de uso 3.** Se utiliza cuando no está definido aún quien de los dos iniciar la relación de interacción.



# Descripción del Caso de Uso.

El propósito de la descripción de un caso de uso es contar la historia de cómo el sistema y sus actores trabajan en conjunto para alcanzar una meta específica.

Describen una secuencia de acciones, incluyendo variantes, que un sistema y sus actores pueden ejecutar para alcanzar una meta.

Se representan como un conjunto de flujos que describen cómo un actor emplea un sistema para alcanzar una meta y lo que el sistema hace para ayudar al actor a alcanzar esa meta.

Las narrativas de los casos de uso se pueden desarrollar a distintos niveles de detalle.

# Descripción del Caso de Uso.

Nosotros utilizaremos:

## Narrativa con Resumen Esencial

- Algunas veces es necesario aclarar las responsabilidades del sistema y sus actores
- mientras se trabaja en el caso de uso.
- Aquí la narrativa es una descripción del diálogo entre el sistema y sus actores.

# Descripción del Caso de Uso.

Nosotros utilizaremos:

## Narrativa con Resumen Esencial

- Algunas veces es necesario aclarar las responsabilidades del sistema y sus actores
- mientras se trabaja en el caso de uso.
- Aquí la narrativa es una descripción del diálogo entre el sistema y sus actores.

# Aclaraciones sobre Casos de Uso.

No todos los casos de uso se deben especificar al mismo nivel de detalle; lo mismo aplica para las secciones dentro de un caso de uso.

Un caso de uso puede seguir distintos caminos.

Estos caminos representan escenarios que están dentro del caso de uso

Los escenarios pueden agruparse, según si cumplen con el objetivo, en escenarios de éxito y escenarios de fracaso.



# Aclaraciones sobre Casos de Uso.

**¿Cómo describo un caso de uso?**

**debería representar lo que el sistema hace (en interacción con el usuario),  
no cómo está diseñado el sistema para ejecutar el comportamiento requerido**

**¿Cómo encuentro actores?**

**Buscar entidades externas que tienen interacciones con el sistema.**

**Solo aquellos que se comunican directamente con el sistema necesitan ser  
considerados actores.**

# Aclaraciones sobre Casos de Uso.

## Generalización de Actores

Una generalización de un tipo de actor indica que el descendiente hereda el rol que el ancestro puede jugar en un caso de uso.

Un usuario puede jugar varios roles en relación con el sistema, lo que significa que un usuario puede corresponder a varios actores.

# Aclaraciones sobre Casos de Uso.

## Casos de uso Concretos y Abstractos

Un caso de uso concreto es iniciado por un actor.

Un caso de uso abstracto nunca se inicia por un actor.

Los casos de uso abstractos son incluidos en, extienden o generalizan otros casos de uso.

# Estructuración de Casos de Uso.

- Extensión
- Inclusión
- Generalización

**Se hace para:**

**Hacer a los casos de uso más fáciles de entender.**

**Particionar comportamiento común descrito en muchos casos de uso.**

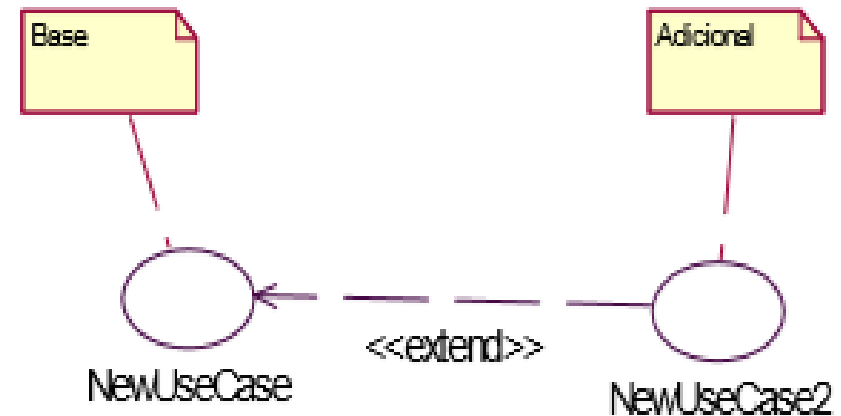
**Hacer el modelo de caso de uso más fácil de mantener.**

# Estructuración de Casos de Uso.

## Extensión

- Es un caso de uso que extiende la funcionalidad de otro.
- El caso de uso donde se inserta la extensión, debe ser un curso completo en sí mismo.
- Se usan para modelar partes alternativas de un caso de uso, con objetivo propio.
- Se dibuja con una flecha de línea cortada, cuya dirección va desde el caso de uso de extensión (adicional) al caso de uso base.

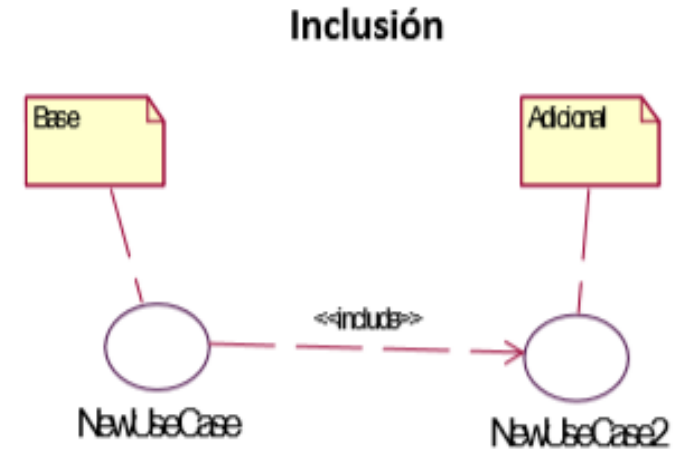
## Extensión



# Estructuración de Casos de Uso.

## Inclusión

- Específica y agrupa comportamiento similar de varios casos de usos, en un caso de uso abstracto, que otros podrán usar.
- Se usan cuando su intervención es necesaria para completar un curso completo de eventos.
- Se dibuja con una flecha desde el caso de uso concreto o base al caso de uso abstracto (adicional).



# Estructuración de Casos de Uso.

## Generalización

- Se usa cuando los hijos tienen el mismo objetivo y comportamiento que el padre, con variaciones.
- Una relación de generalización entre casos de usos implica que el caso de uso hijo contiene todos los atributos y secuencias de comportamiento y puntos de extensión definidos para el padre.
- Los casos de usos hijos pueden redefinir el comportamiento heredado del padre.
- Se dibuja con una flecha de línea entera y punta cerrada, desde el caso de uso hijo al padre.

