



Universidad Andrés Bello.
Facultad de ingeniería.
Escuela de informática.
Sistemas de Información.

SISTEMAS DE INFORMACION II

AYUDANTE
Carlos Duque Jáuregui

SANTIAGO – CHILE
INVIERNO 2010



SIA

¿Qué es un SIA?

Tipos de SIA:

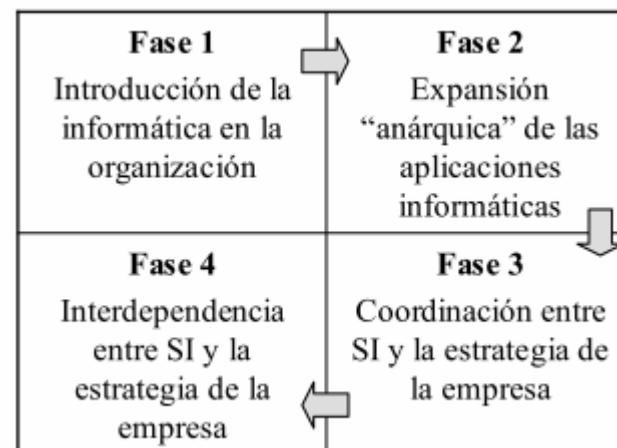
Transaccionales

Ayuda a la toma de decisiones

Estrategicos

Planificación de Recursos ERP

Fidelización de Clientes





UML

¿Qué es UML (Unified Modeling Language)?

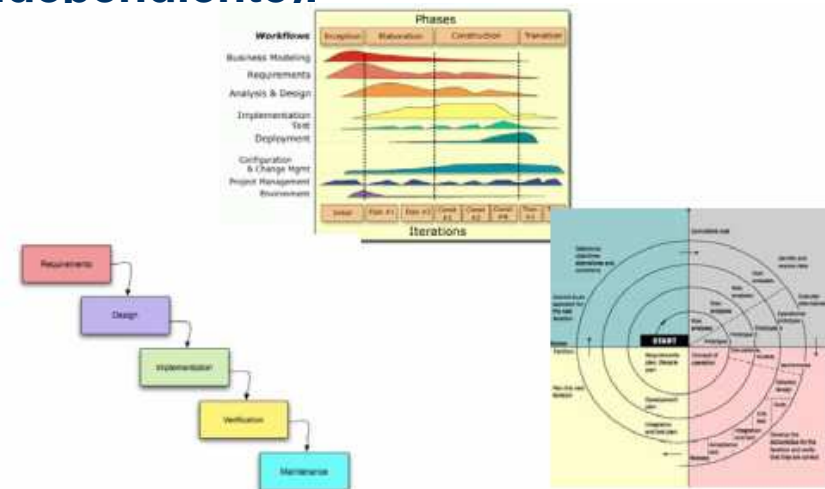
UML es el lenguaje estándar para especificar, visualizar, construir, y documentar todos los artefactos de un sistema de software.

UML es un lenguaje de para “especificar“ y no un metodo o Procedimiento.

Puede ser usado en la mayoría de las metodologías de desarrollo de software. (es independiente).

Ejemplos de metodologías (

- Incremental Model (~1985)
- Transformation Model (~1975)
- Spiral Model (~1988)
- Component Model (~1992)
- RUP (1997)
- WebE (~1998)
- Extreme Programming (~1998)





UML

Estructura de UML

UML define la notación y semántica para los siguientes dominios:

El modelo de Interacción o Modelo de Caso de Uso

El modelo de Clases o Lógico

El modelo dinámico o de Estado

El modelo de Componentes

El modelo de Despliegue físico

El modelo de Despliegue físico



UML

Estructura de UML

UML define la notación y semántica para los siguientes dominios:

El modelo de Interacción o Modelo de Caso de Uso

El modelo de Clases o Lógico

El modelo dinámico o de Estado

El modelo de Componentes

El modelo de Despliegue físico

El modelo de Despliegue físico



UML





UML – casos de uso

- ❖ **Modela la funcionalidad del sistema.**
- ❖ **Entendibles para los Clientes y los Usuarios.**
- ❖ **Conectan todos los otros modelos.**





UML – diagrama de componentes

- ❖ Modelan las piezas de software.
- ❖ Contienen información precisa de la implementación.





Diseño de componentes

¿Para qué sirve el diseño de componentes?

Tiene como objetivo convertir el diseño de datos, interfaces y arquitectura en una representación intermedia que se pueda transformar fácilmente en código fuente.

El nivel de abstracción debe ser muy próximo al código.



UML - Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones.

Muestran las opciones de realización incluyendo código fuente, binario y ejecutable.

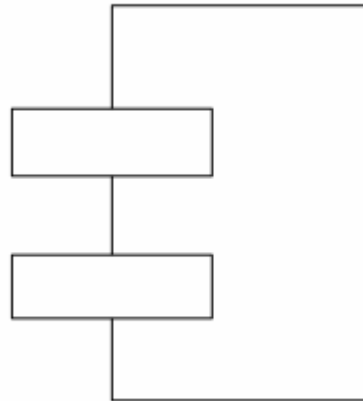
Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas

Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc.



UML - Diagrama de componentes

Representación gráfica de un componente.





UML - Diagrama de componentes

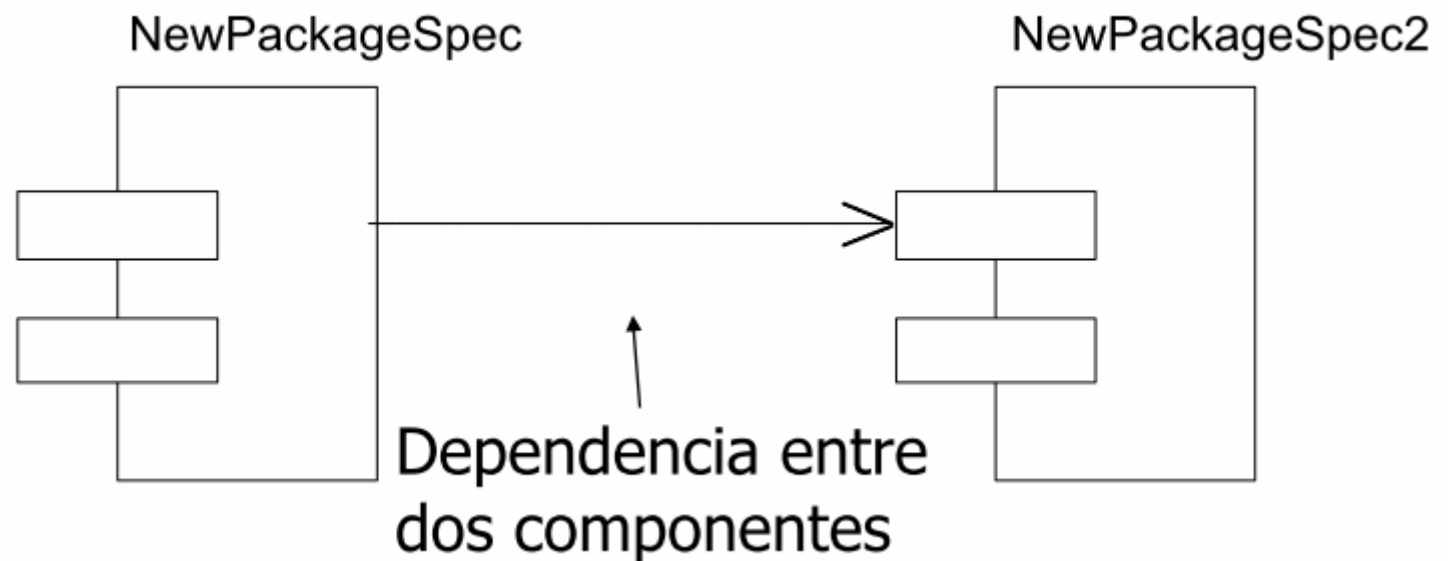
UML define cinco estereotipos estándar que se aplican a los componentes:

- ❖ **Executable:** Especifica un componente que se puede ejecutar en un nodo.
- ❖ **Library:** Especifica una biblioteca de objetos estática o dinámica.
- ❖ **Table:** Especifica un componente que representa una tabla de una base de datos.
- ❖ **File:** Especifica un componente que representa un documento que contiene código fuente o datos.
- ❖ **Document:** Especifica un componente que representa un documento.



UML - Diagrama de componentes

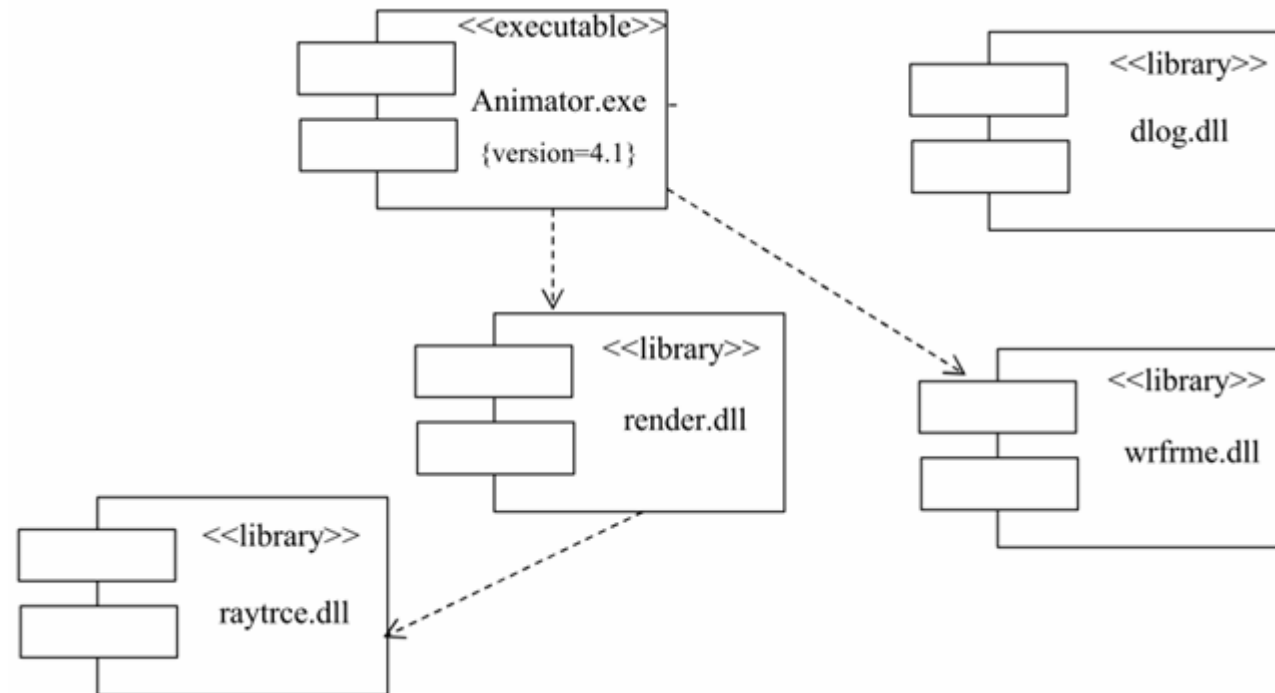
Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente se refiere a los servicios ofrecidos por otro componente.





UML - Diagrama de componentes

Ejemplo: Modelo de ejecutables y bibliotecas





Diccionario

Interfaz

En software, parte de un programa que permite el flujo de información entre un usuario y la aplicación, o entre la aplicación y otros programas o periféricos. Esa parte de un programa está constituida por un conjunto de comandos y métodos que permiten estas intercomunicaciones.

Instancia

En programación una instancia se produce con la creación de un objeto perteneciente a una clase (instanciar una clase), que hereda entonces sus atributos, propiedades y métodos para ser usados dentro de un programa, ya sea como contenedores de datos o como partes funcionales del programa al contener en su interior funcionalidades de tratamiento de datos y procesamiento de la información que ha sido programada con anterioridad en la clase a la que pertenece.



Fuente

**Extracto “Intro UML”, 2009
Profesor Guillermo Badillo.**



CONSULTAS