



*Modelado con UML*



# Diagrama de Componentes

**Luis de Marcos Ortega**



- Arquitectura física: Muestra cómo el sistema diseñado se empaqueta en componentes ejecutables y distribuye en los elementos físicos con capacidad para ejecutarlos.
- Elementos Hardware
  - Dispositivo (device). Elemento con capacidad de procesamiento
  - Línea de Comunicación (communication path). Conexiones entre procesadores.
  - Entorno de ejecución (execution environment). Proveen el contexto necesario para que los componentes software puedan ejecutarse. Generalmente subnodos dentro de los dispositivos.
- Elementos Software
  - Componentes. Conjunto de elementos de arquitectura lógica agrupados
  - Artefacto (artifact). Implementación de un componente en el entorno de desarrollo.
    - Artefacto Fuente. <<source>>
    - Artefacto Ejecutable. <<executable>>
  - Especificador de despliegue (deployment specifier). Conjunto de propiedades que determinan los parámetros de ejecución de un artefacto o componente.



# *Diagrama de Componentes*

## *- Diagrama de Componentes*


- El *diagrama de componentes* proporciona una *visión física* de la construcción del sistema de información. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos.
- Un *componente* es un módulo de software que puede ser código fuente, código binario, un ejecutable, o una librería con una interfaz definida. Es un elemento o grupo de elementos identificables.
- Los componentes representan elementos *reusables* (en el mismo u otro sistema) de la arquitectura
- Una *interfaz* establece las operaciones externas de un componente, las cuales determinan una parte del comportamiento del mismo. Además se representan las dependencias entre componentes o entre un componente y la interfaz de otro, es decir uno de ellos usa los servicios o facilidades del otro.



# Notación de Componentes

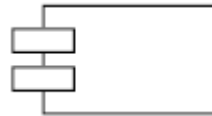
## - Diagrama de Componentes

### ■ Notación de los Componentes:

- Rectángulo con el estereotipo <<component>>, el nombre del componente y opcionalmente el icono  en la parte superior derecha.




- Notación UML 1.X (en desuso) Rectángulo, con dos pequeños rectángulos superpuestos perpendicularmente en el lado izquierdo.



Con la notación de UML 2 un componente puede poseer subcomponentes



### ■ Notación de los *Artefactos*:

- Para distinguir distintos tipos de artefactos (y distinguirlos de los componentes) se les puede asignar un estereotipo: <<artifact>>, <<source>> o <<executable>>. También se pueden especificar estereotipos para artefactos de entornos específicos: <<EJB>>, <<DLL>>, etc...
- Se puede mostrar el icono  en la parte superior derecha del artefacto






# Notación de Interfaces

## - Diagrama de Componentes

### ■ Notación de las Interfaces:

- La *interfaz implementada* por un componente se representa como un pequeño círculo situado junto al componente que lo implementa y unido a él por una línea continua. La interfaz puede tener un nombre que se escribe junto al círculo. Un componente puede proporcionar más de una interfaz.



- Las *interfaces requeridas* se muestran con el conector 
- Se pueden usar *puertos* para mostrar la interfaz requerida e implementada por cada componente formalizando sus puntos de interacción
- Los ensambles (assembler) se pueden usar para mostrar claramente las relaciones entre interfaces requeridos e implementados



### ■ Relaciones de Dependencia:

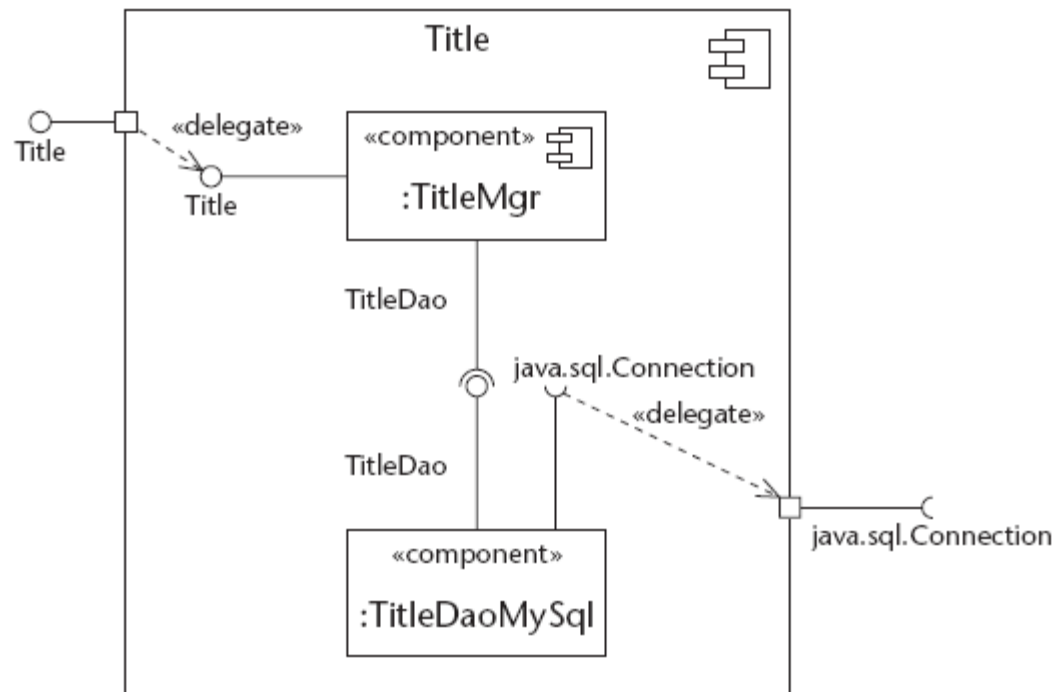
- Una relación de dependencia se representa mediante una línea discontinua con una flecha que apunta al componente o interfaz que provee del servicio o facilidad al otro. Si la dependencia ocurre entre...
  - ... artefactos de código. Al recompilar uno debe recompilarse el otro.
  - ... artefactos ejecutables. Un componente requiere de otro en tiempo de ejecución.
  - También pueden darse dependencias entre componentes / artefactos / interfaces / especificaciones
- La relación puede tener un estereotipo:
  - <<delegate>>: Para indicar que un puerto o interfaz delega la implementación del interfaz en otro componente. Se nombra responsable a otro componente de implementar el interfaz
  - <<manifest>>: Para indicar que un artefacto representa la implementación de un componente



# Ejemplo

## - Diagrama de Componentes

### ■ Ejemplo







# Ejemplo

## - Diagrama de Componentes

### ■ Ejemplo

