UML

Unified Modeling Language

(Lenguaje de Modelamiento unificado)


Vista Implementación
Vista Estructural
Vista de Usuario
Vista Comportamiento
Vista de Ambiente
IMPLEMENTACIÓN

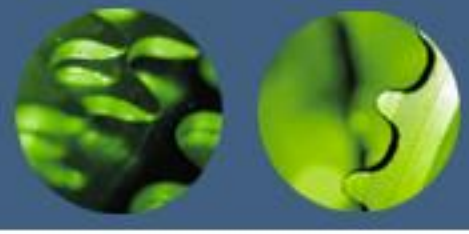


Diagrama de Componentes

- Un Componente de Software es una parte física de un Sistema y se encuentra en la Computadora y no en la mente del Analista.
- Se puede tomar como Componente: tabla, archivo de datos, html, ejecutable, biblioteca de vínculos dinámicos, documentos, etc.

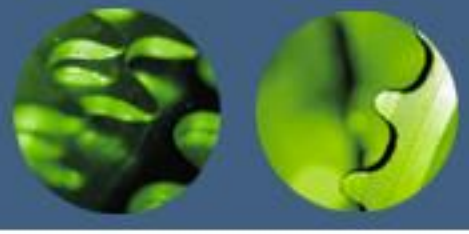


Diagrama de Componentes

- Los Diagramas de Componentes se utilizan para:
  - Los Clientes puedan ver la estructura del Sistema finalizado.
  - Los Desarrolladores cuenten con una estructura con la cual trabajar en adelante.
  - Quienes escriban las notas técnicas y la documentación puedan entender lo que escriben.
  - Ustedes se alisten para volver a utilizar los Componentes.

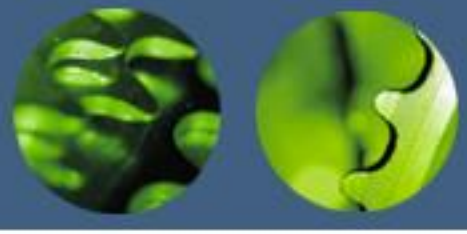


Diagrama de Componentes

- Los Diagramas de Componentes se utilizan para:
  - Modelar Código Fuentes.
  - Modelar Versiones Ejecutables.
  - Modelar Base de Datos Físicas.
  - Modelar Sistemas Adaptables.
- Los componentes representan todos los tipos de elementos software que entran en la Fabricación de aplicaciones informáticas.

Diagrama de Componentes

- Muestra la organización y las Dependencias entre un conjunto de Componentes.
- Cubren la vista de la Implementación Estática y se relacionan con los Diagramas de Clases ya que en un Componente suele tener una o mas Clases, interfaces o Colaboraciones.
- Cuando se habla del Diagrama de Componentes, se trata obviamente de, Componentes, Interfaces y Relaciones.

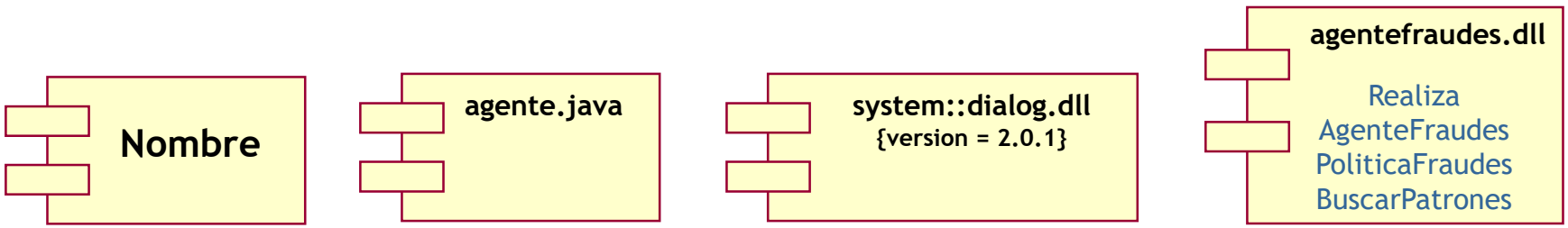
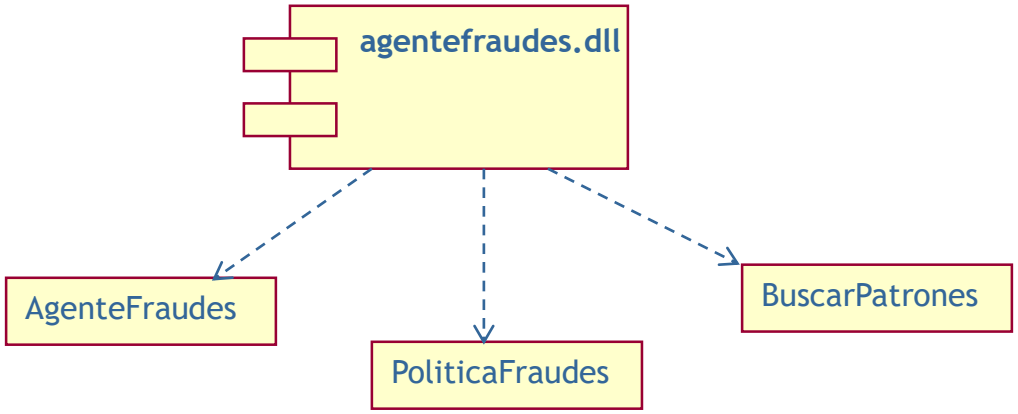


Diagrama de Componentes

Componentes y Clases

Las clases representan abstracciones lógicas. Los componentes son elementos físicos del mundo real. Un componente es la implementación física de un conjunto de otros elementos lógicos, como clases y colaboraciones.



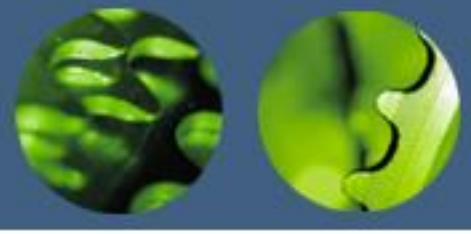


Diagrama de Componentes

Componentes y Clases

UML definen 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 una Archivo de Código Fuente o Archivo de Datos.
- **Document:** Especifica un Componente que representa un documento.

Diagrama de Componentes

Dependencias entre Componentes

La *dependencia* entre dos componentes se muestra como una flecha punteada. La *dependencia* quiere decir que una componente necesita de la otra para completar su definición, ósea, los Servicios ofrecidos por otro Componente .

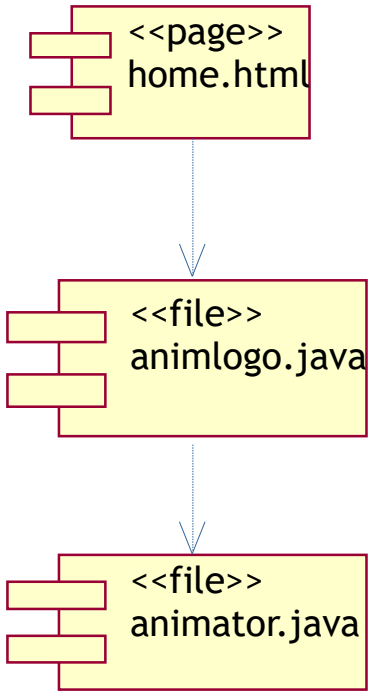


Diagrama de Componentes

Ejemplo

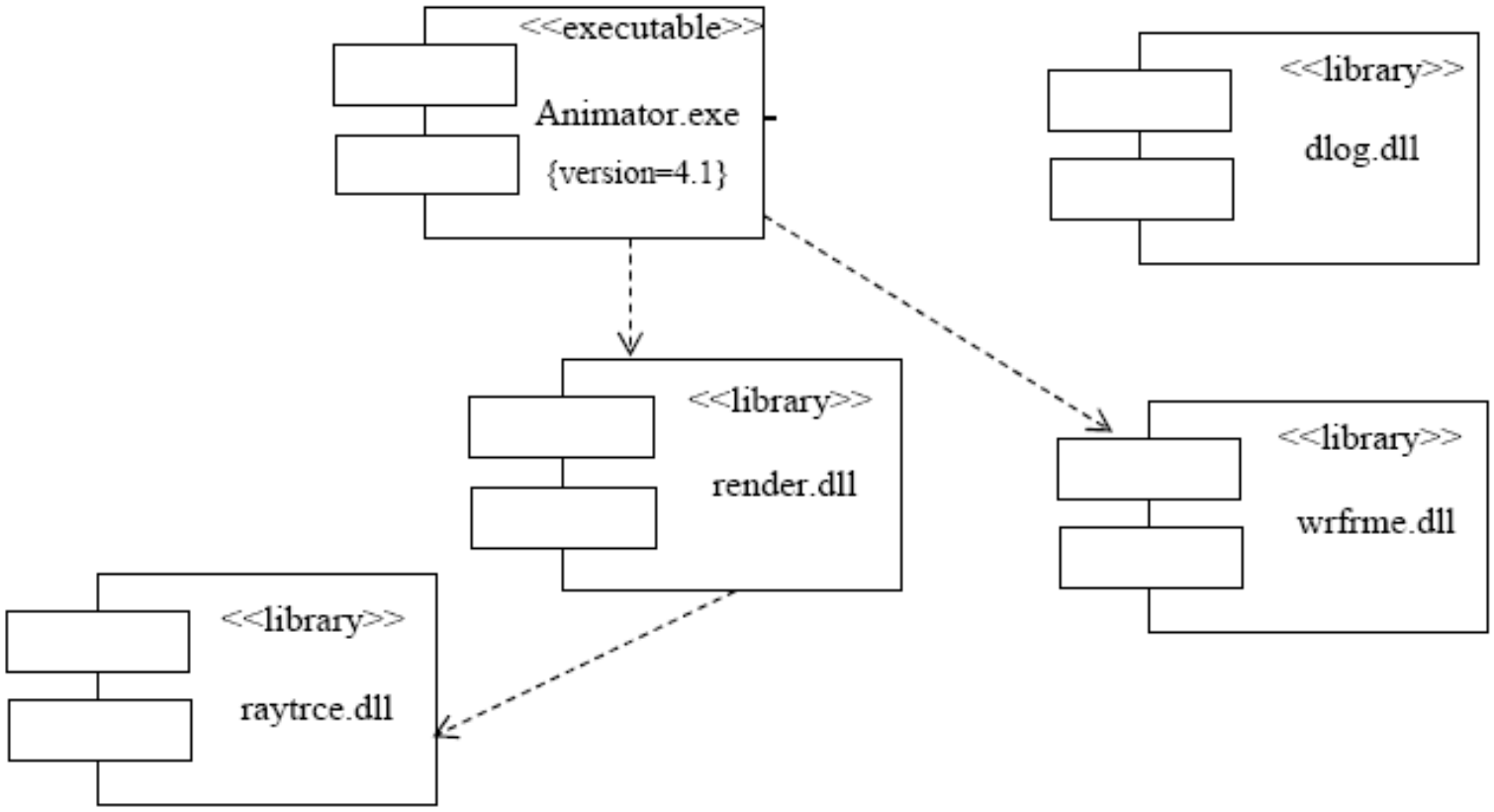


Diagrama de Componentes

Ejemplo

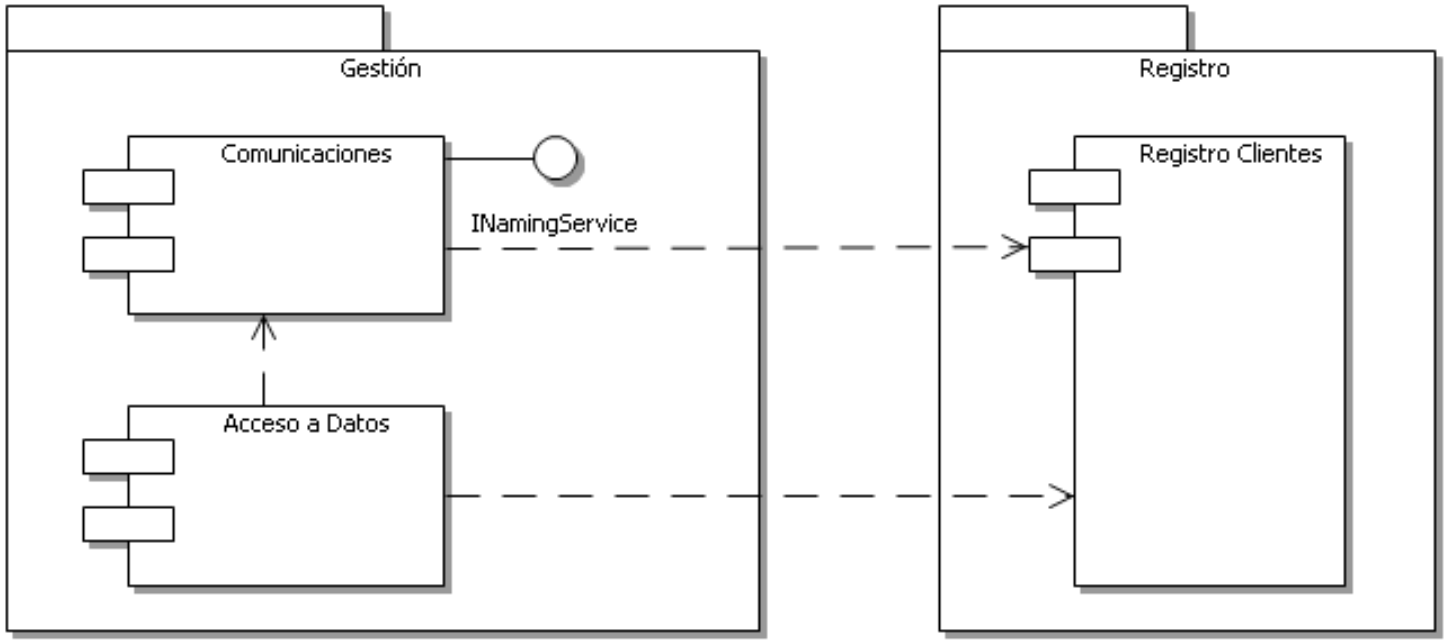
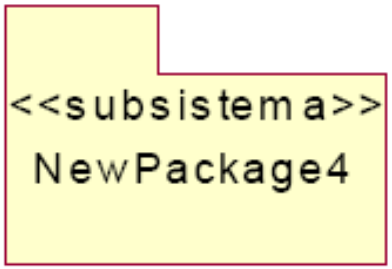


Diagrama de Componentes
Sub Sistemas

- Los distintos componentes pueden agruparse en paquetes según un criterio lógico y con vistas a simplificar la implementación.
- Son paquetes estereotipados en <<subsistemas>> para incorporar la noción de biblioteca de compilación.



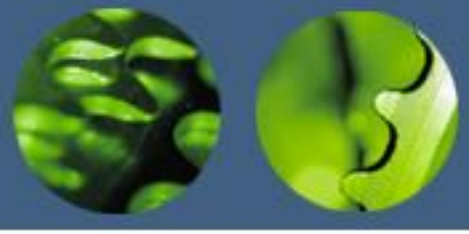
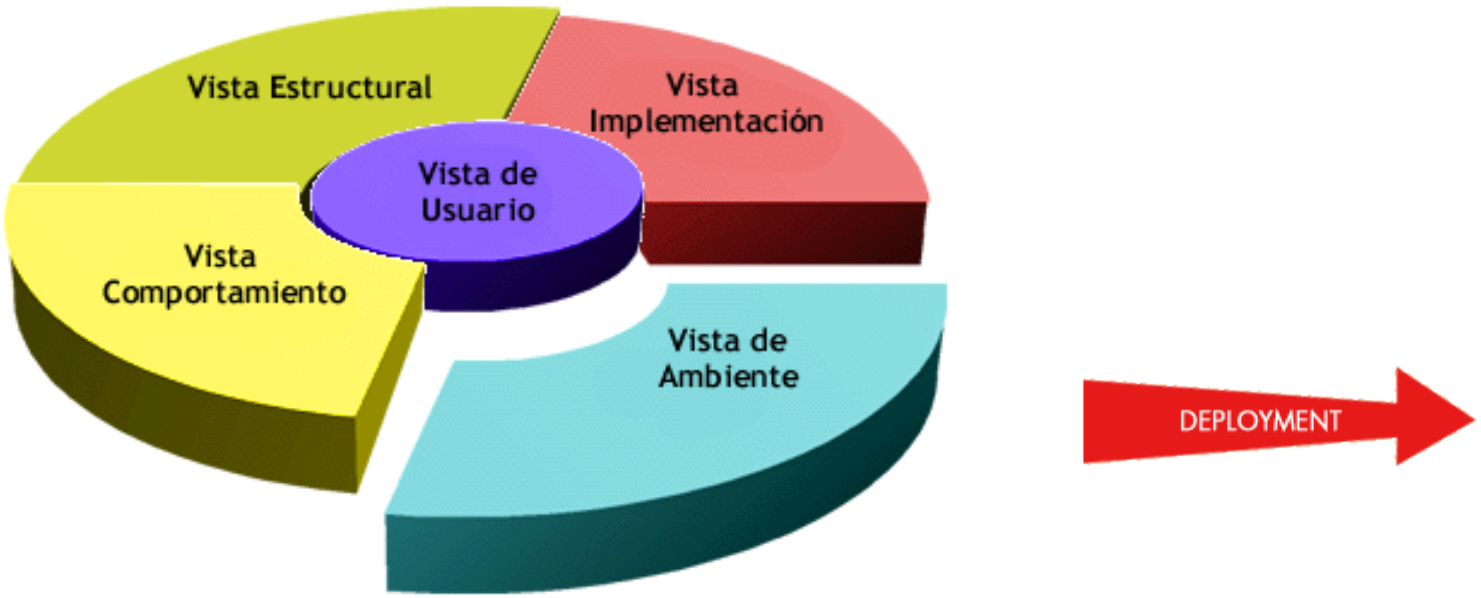


Diagrama de Componentes

Sub Sistemas

- Los subsistemas organizan la vista de realización de un sistema.
- Cada subsistema puede contener componentes y otros subsistemas.
- La descomposición en subsistemas no es una descomposición funcional.
- La relación entre paquetes y clases en el nivel lógico es el que existe entre subsistemas y componentes en el nivel físico.


Vista Estructural
Vista Implementación
Vista de Usuario
Vista Comportamiento
Vista de Ambiente
DEPLOYMENT

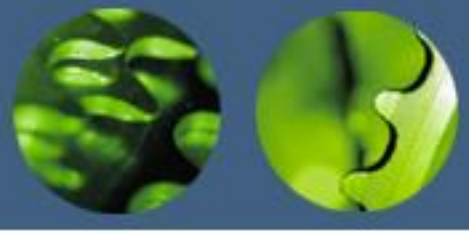


Diagrama de Despliegue o Distribución

- Los Diagramas de Despliegue o Distribución muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.
- Los Diagramas de Despliegue o Distribución modelan la topología del hardware sobre el que se ejecuta el Sistema Software.
- Este tipo de diagramas suele utilizarse para modelar Sistemas Distribuidos o Sistemas Empotrados. En los sistemas monolíticos, generalmente, resultan innecesarios.

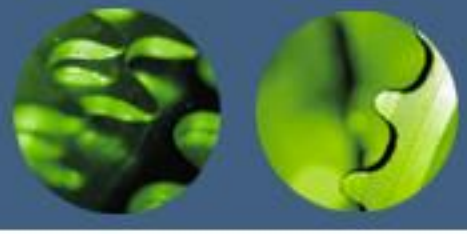


Diagrama de Despliegue o Distribución

- Representa los Dispositivos y Equipos, mostrar sus interconexiones y el Software que se encuentra en cada maquina.
- Modela la distribución en tiempo de ejecución de los elementos de procesamiento y componentes de software, junto a los procesos y objetos asociados.

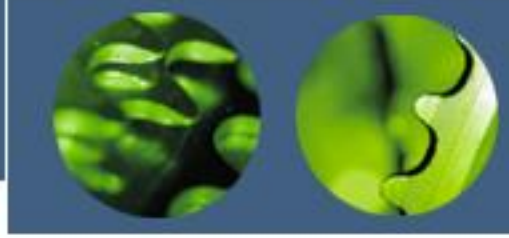


Diagrama de Despliegue o Distribución

- Un nodo es un recurso de ejecución, representa un recurso de ejecución tal como:
  - Dispositivos
  - Procesadores
  - Memoria
  - Sistema Operativos
  - Bases de Datos

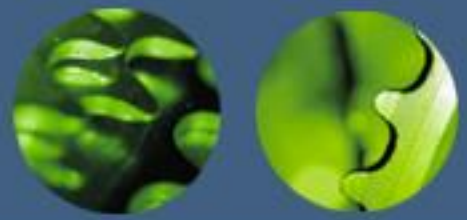


Diagrama de Despliegue o Distribución

- Un Nodo es un elemento físico, que existe en tiempo de ejecución y representa un recurso computacional que generalmente tiene alguna memoria y, a menudo, capacidad de procesamiento.
- Cada nodo puede contener instancias de componentes.
- Los nodos se interconectan mediante soportes bidireccionales que pueden a su vez estereotiparse.
- Esta vista permite determinar las consecuencias de la distribución y la asignación de recursos.

Diagrama de Despliegue o Distribución

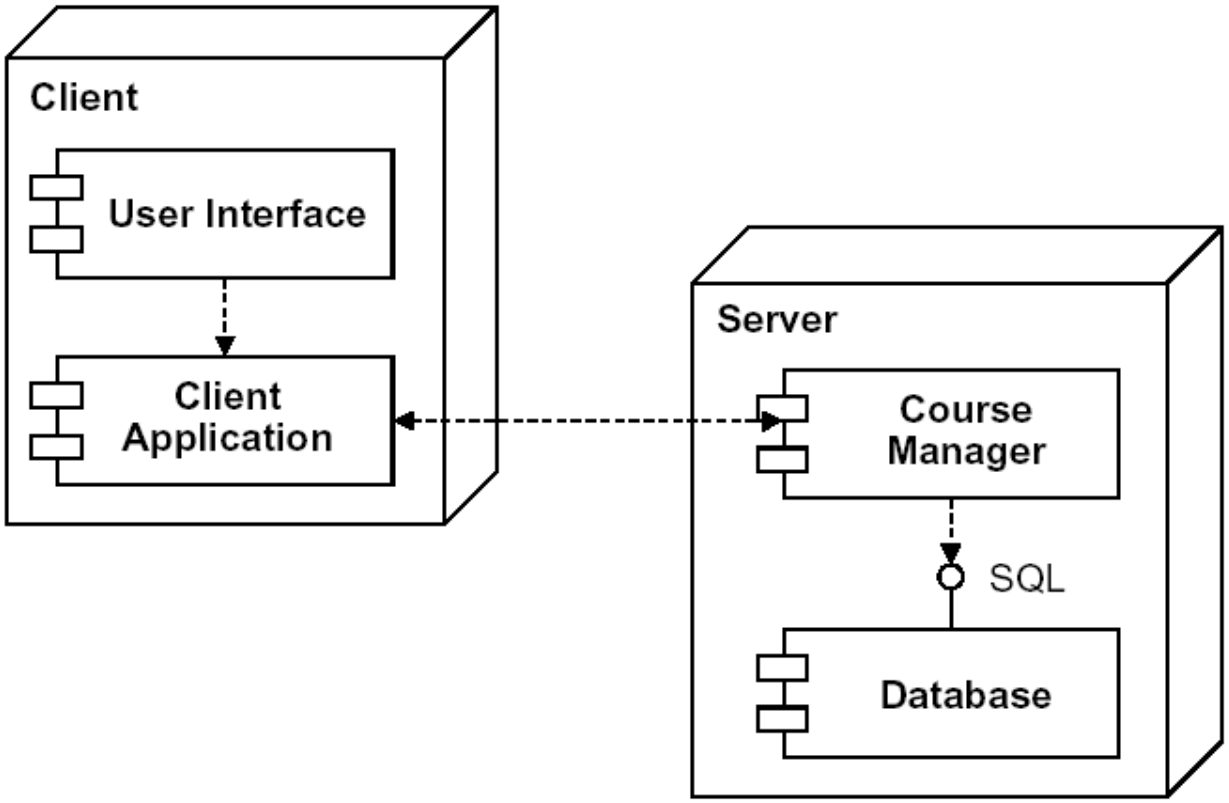


Diagrama de Despliegue o Distribución

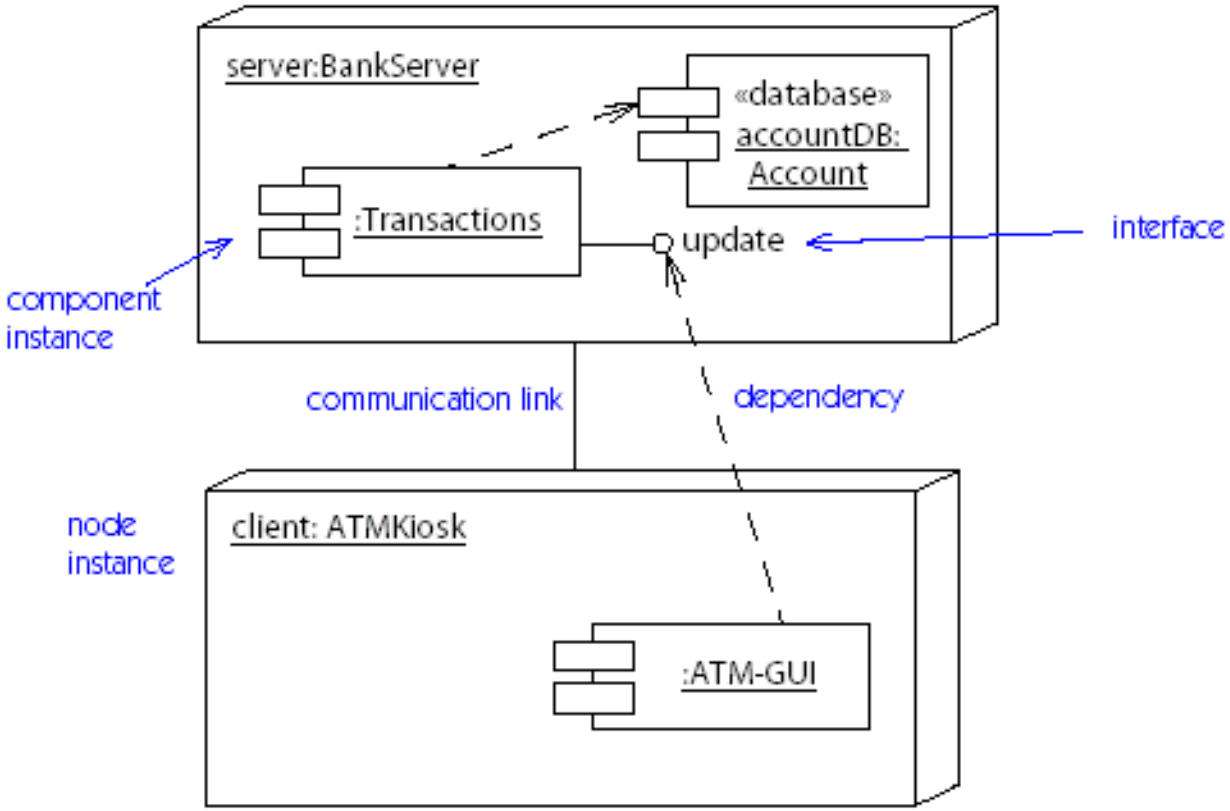
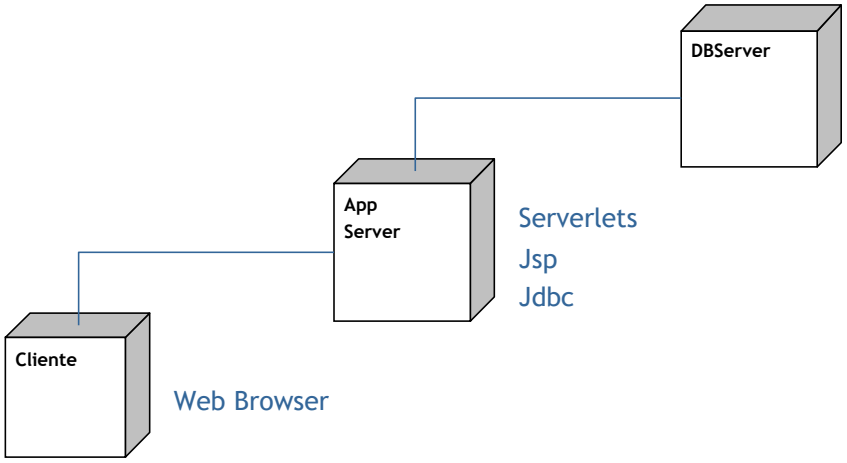


Diagrama de Despliegue o Distribución



Relación entre Nodos y Componentes

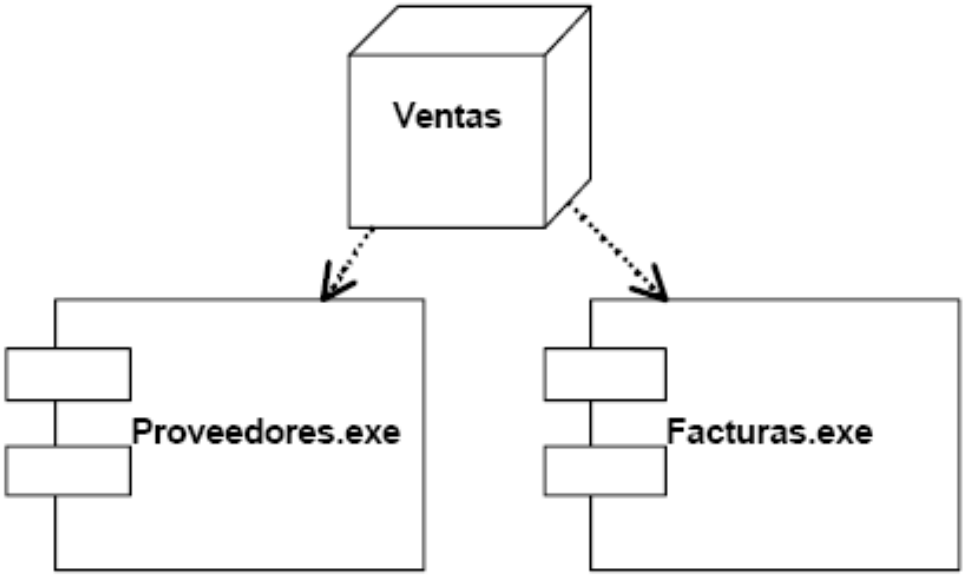
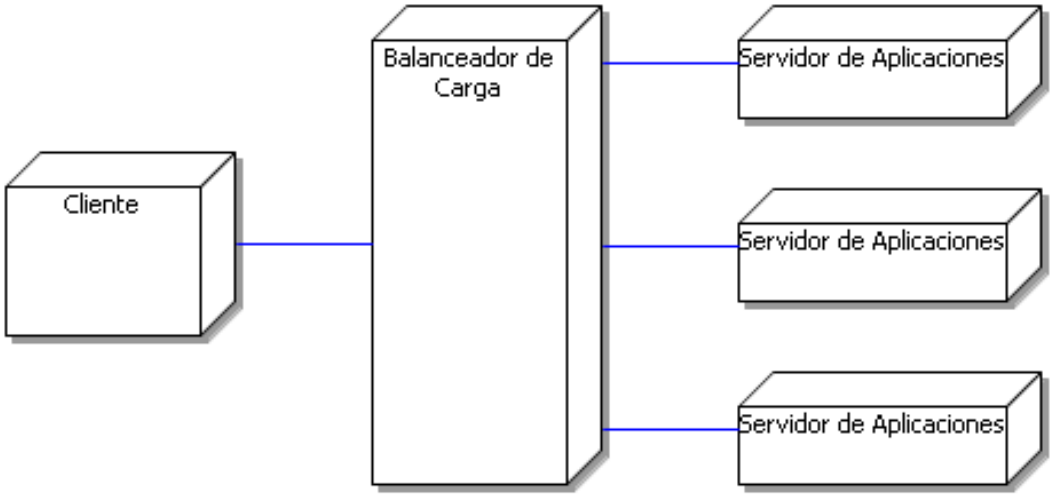


Diagrama de Despliegue o Distribución



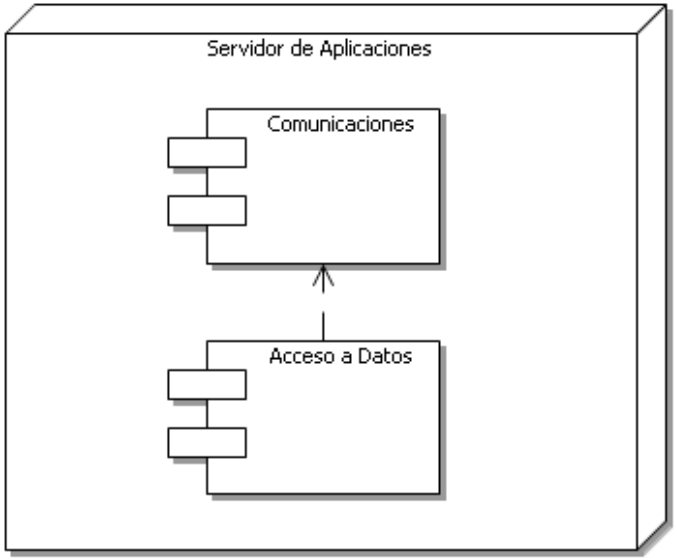
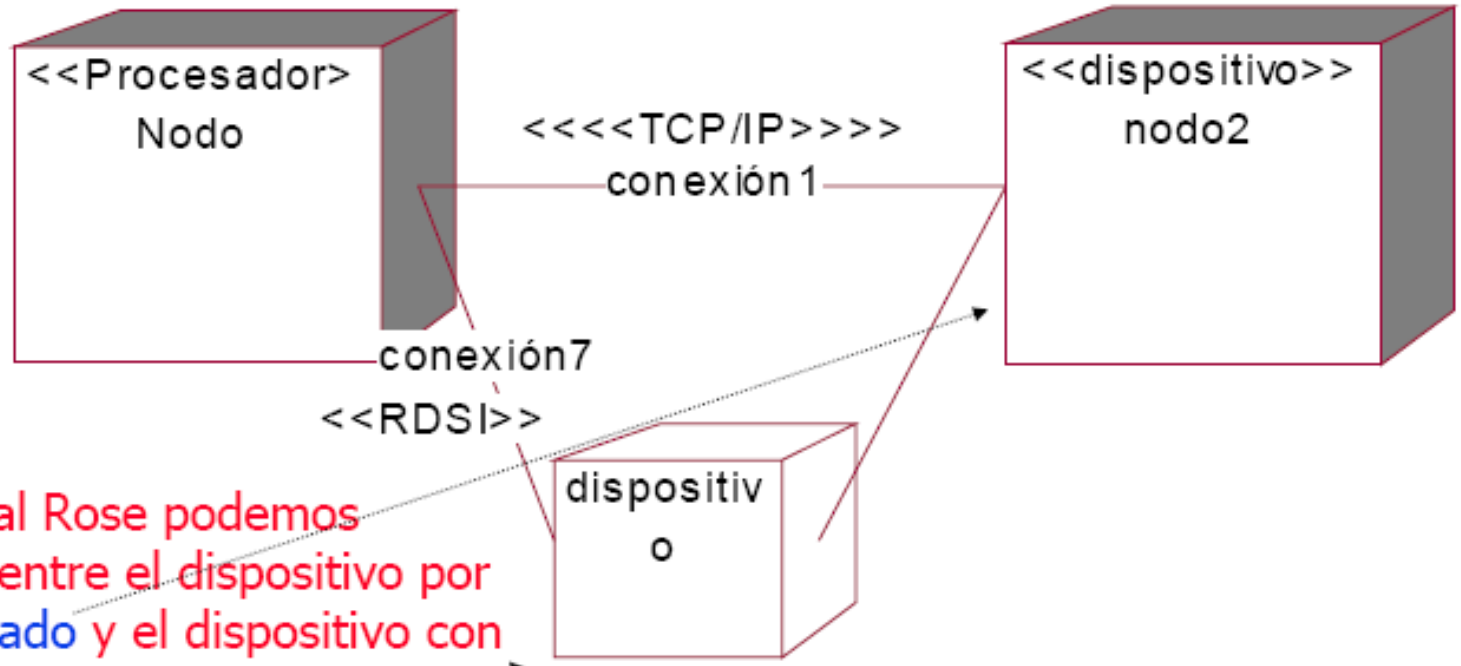
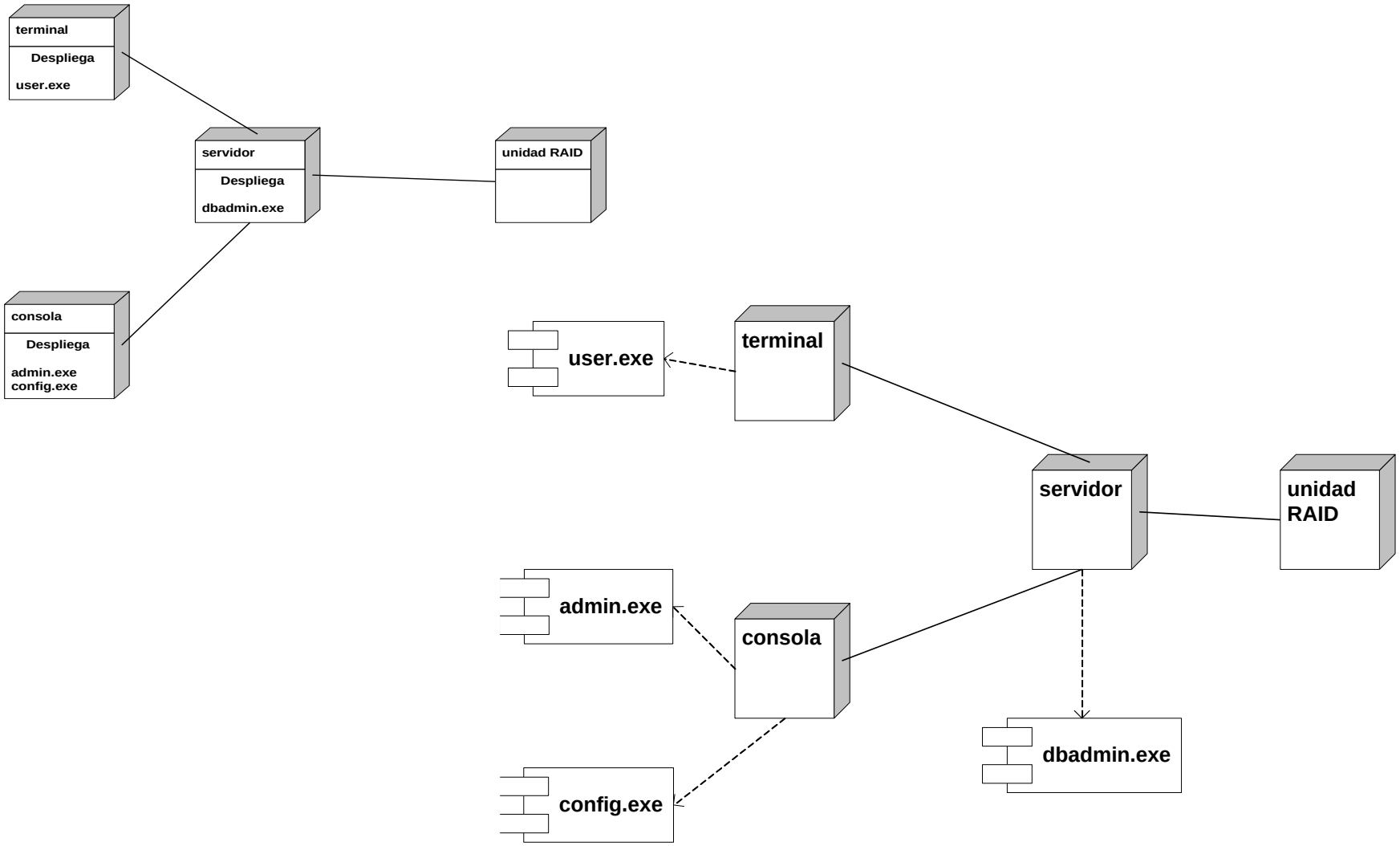


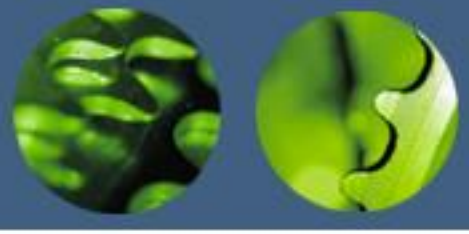
Diagrama de Despliegue o Distribución



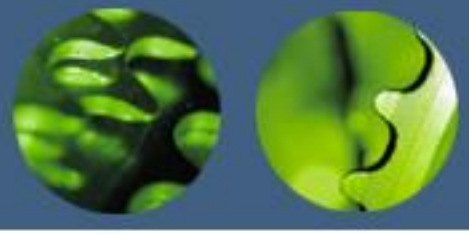
En Rational Rose podemos distinguir entre el dispositivo por estereotipado y el dispositivo con su propio símbolo

Diagrama de Despliegue o Distribución



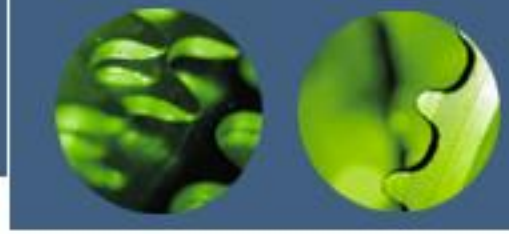


Conclusiones



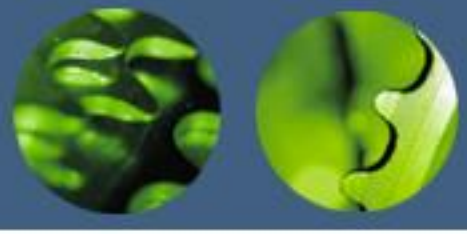
Conclusión

- El UML es un lenguaje reconocido mundialmente por la industria de construcción de software.
- El Modelamiento visual es una de las técnicas probadas que brinda mejores resultados.
- Todos los sistemas tienen una estructura estática y comportamiento dinámico.
- La estructura se describe con los diagramas de clases, componentes y despliegue.
- El comportamiento dinámico del sistema se describe con diagramas de estados, secuencias, colaboración y actividades.



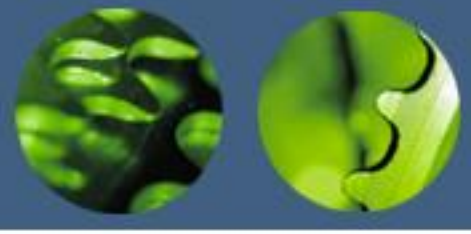
Conclusión

- UML define una notación que se expresa como diagramas sirven para representar modelos/subsistemas o partes de ellos.
- El 80% de la mayoría de los problemas pueden modelarse usando alrededor del 20% de UML– Grady Booch



Herramientas CASE

- Rational Rose (www.rational.com)
- Rational XDE (www.rational.com)
- Borland Together (www.borland.com/together/)
- Embarcadero Describe (www.embarcadero.com/)



Herramientas CASE - Libre

- Argo UML (argouml.tigris.org)
- Poseidon (www.gentleware.com)
- Dome (www.htc.honeywell.com/dome)

- Comparativa:
  - http://www.diatel.upm.es/malvarez/UML/Comparativa.html

¿Preguntas?