

---

# ARQUITECTURA: VISTAS, MODELOS Y COMUNICACIÓN

Autor: Luciano Straccia

Versión 2022.02

---

---

# Vistas de la arquitectura

---

---



## Concepto de vista

- Una vista es una presentación de un modelo. Es una descripción completa de un sistema desde una perspectiva particular (Kruchten).



# Vistas y Modelos

- Vista 4+1 (Kruchten)
- Vistas del SEI
- Modelos UML



## Vista 4+1

- Vista lógica
- Vista de procesos
- Vista de despliegue
- Vista física



## Modelos UML

- Diagrama de Paquetes: muestra cómo un sistema está dividido en agrupaciones lógicas y las dependencias entre esas agrupaciones.
- Diagrama de Componentes: muestra la organización y dependencia entre diversos componentes
- Diagrama de Despliegue: muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos.



## Vistas SEI

- Vista de módulos: describe cómo el sistema es estructurado en un conjunto de unidades de código.
- Vista de conectores y componentes: describe cómo el sistema es estructurado en un conjunto de elementos que están en tiempo de ejecución así como su interacción.
- Vista de asignación: describe la relación entre las unidades de software y los elementos del entorno como hardware, sistemas de archivos o la organización de los equipos de desarrollo de software.

---

# Comunicación de la arquitectura: Diagrama de Paquetes

---

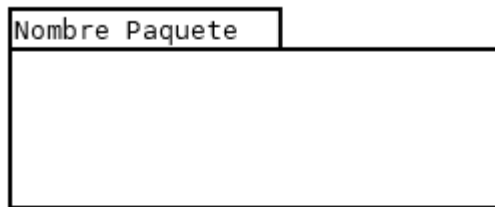
---



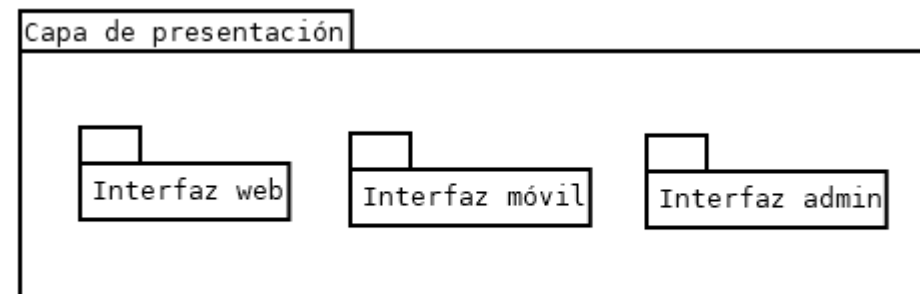


# Diagrama de Paquetes

- El **Diagrama de Paquetes** muestra cómo un sistema está dividido en agrupaciones lógicas y las dependencias entre esas agrupaciones



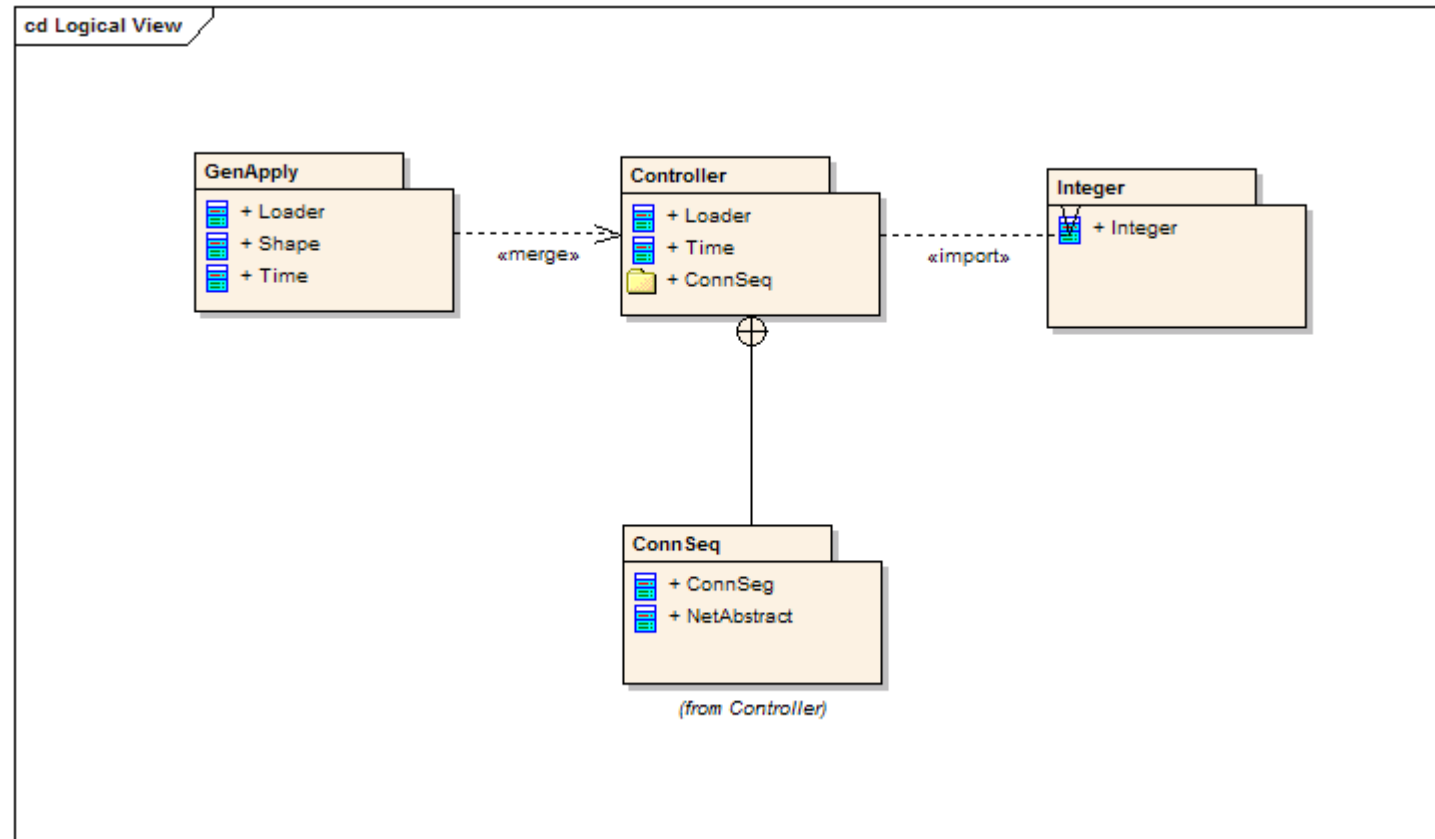
Notación de un paquete



Notación de un paquete que contiene otros paquetes



# Ejemplo



---

# Comunicación de la arquitectura: Diagrama de Componentes

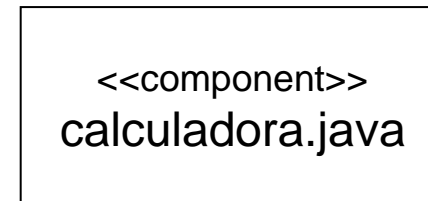
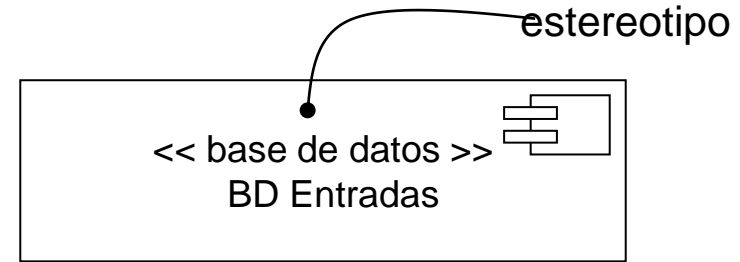
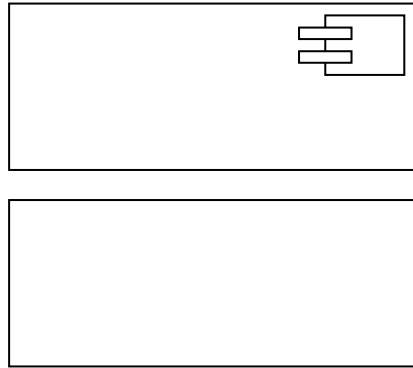


## Diagrama de Componentes

- El **Diagrama de Componentes** muestra la organización y dependencia entre diversos componentes
- Los Diagramas de Componentes ilustran las piezas del software, controladores embebidos, etc. que conformarán un sistema.
- Un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase – usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema.



# Componente

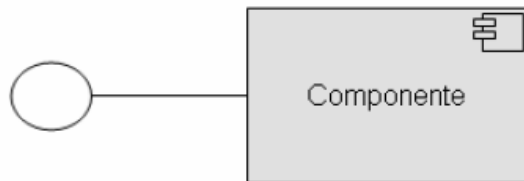




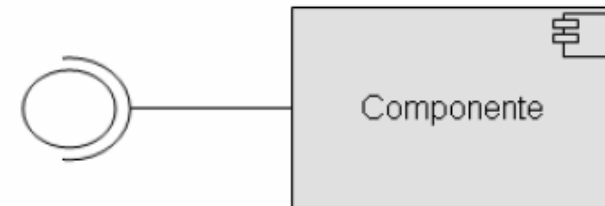
## Interfaz entre componentes

- Interfaz: lazo de unión entre componentes
- Un componente provee una interfaz, otro componente usa esa interfaz

Un componente provee una interfaz



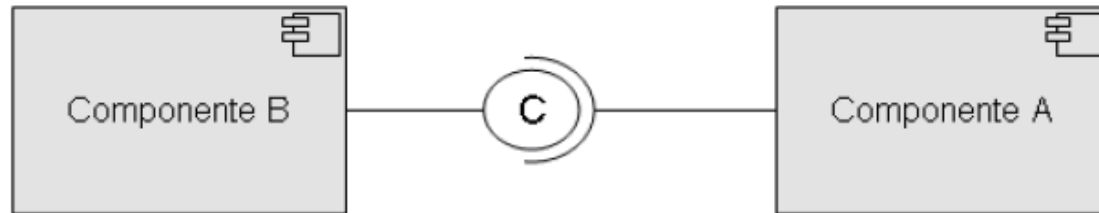
Un componente usa una interfaz





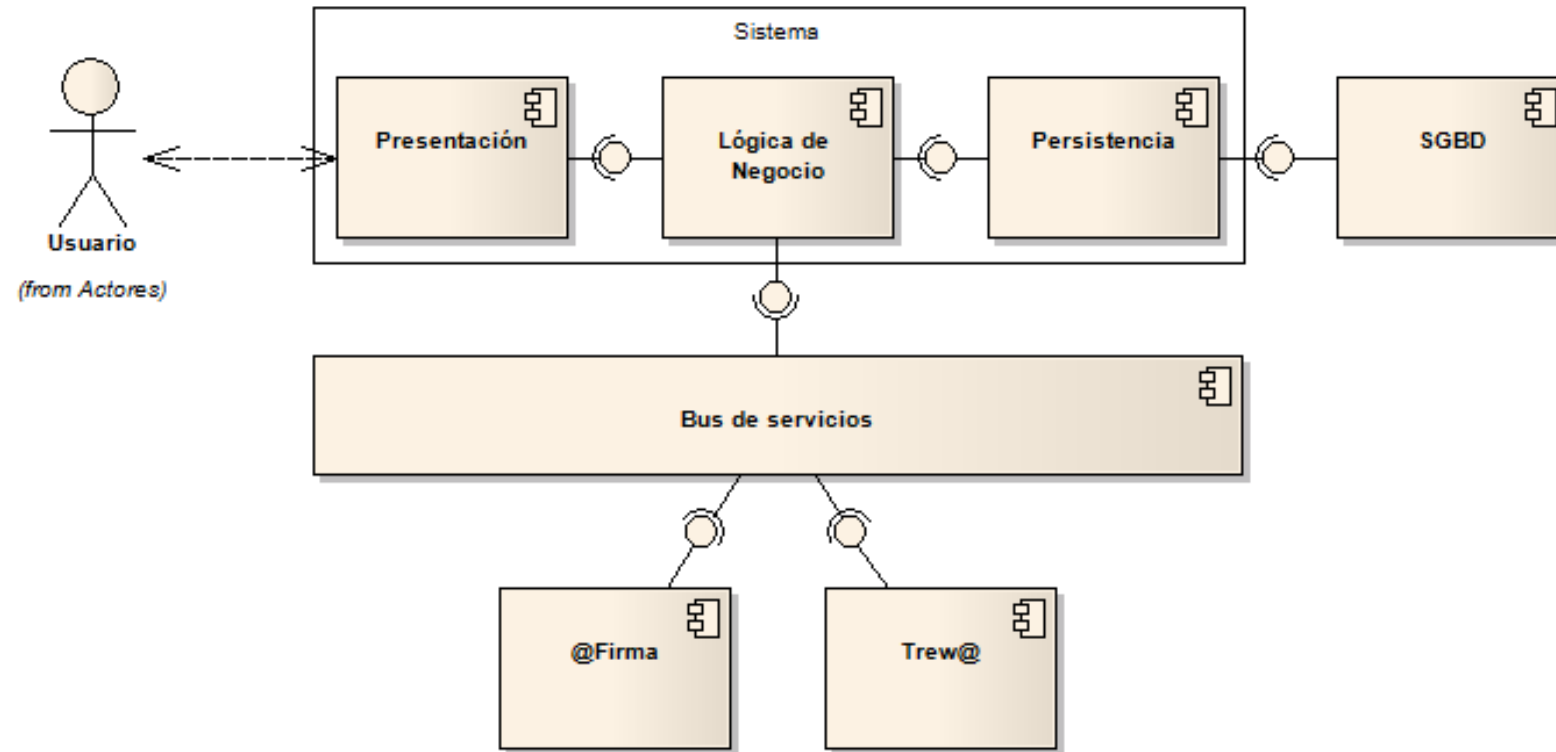
## Ejemplo

- Componente A usa una interfaz que provee el Componente B





# Ejemplo





---

# Comunicación de la arquitectura: Despliegue

---

---



## Diagrama de despliegue

- Un **Diagrama de Despliegue** muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos.



# Componentes

- Nodo
- Artefactos
- Relación entre nodos y artefactos
- Relación entre nodos

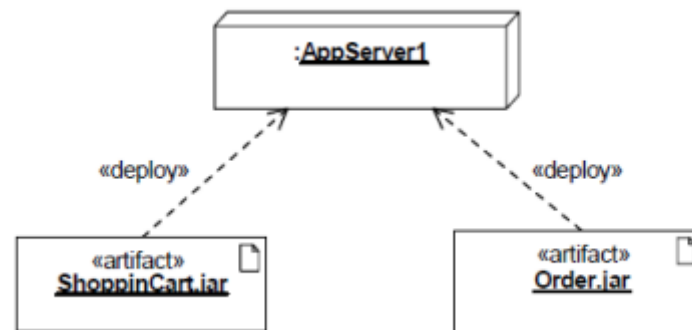
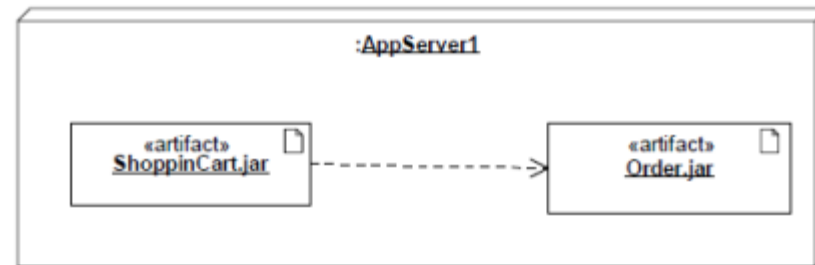


## Componentes

- **Nodo:** es un elemento de hardware o software. Es un contenedor. Un nodo puede contener otros nodos.
- Un artefacto es un producto del proceso de desarrollo de software, que puede incluir los modelos del proceso (e.g. modelos de Casos de Uso, modelos de Diseño, etc.), archivos fuente, ejecutables, documentos de diseño, reportes de prueba, prototipos, manuales de usuario y más.

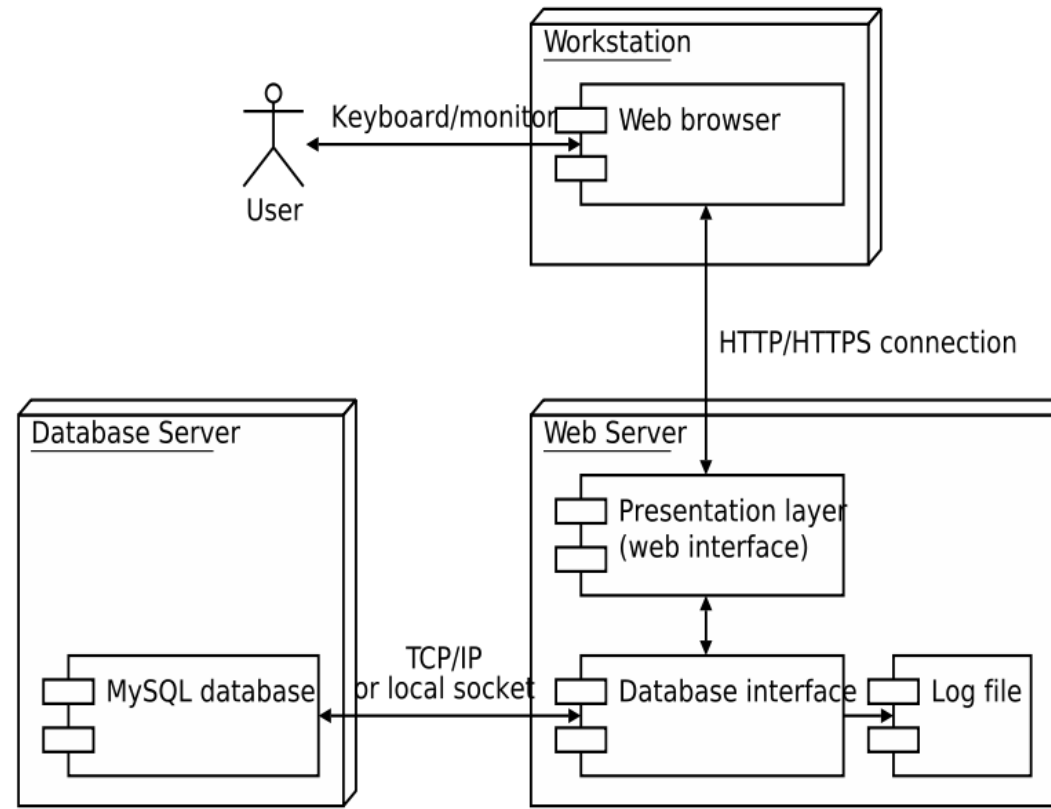


# Componentes y Diagrama



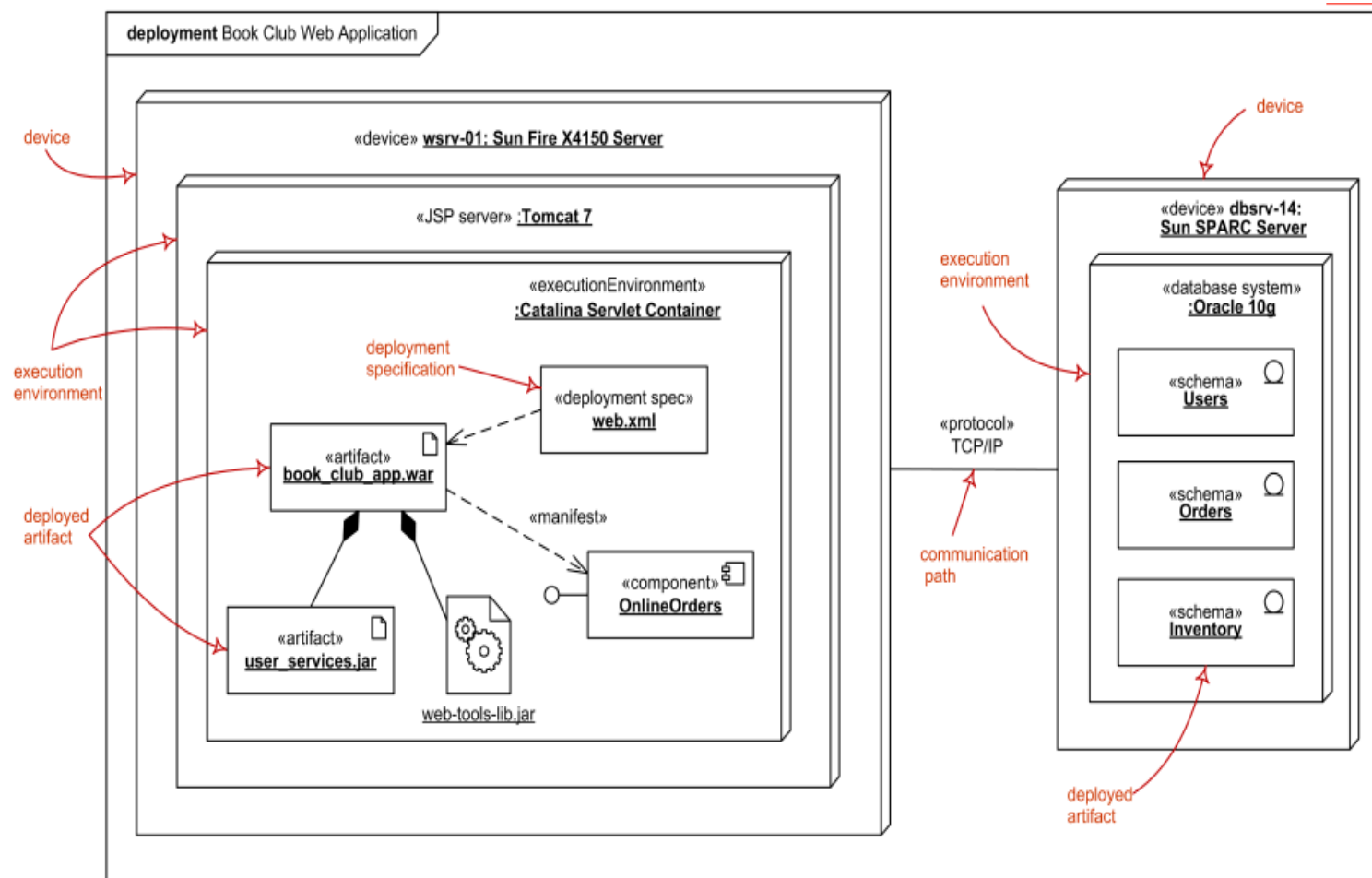


# Ejemplo





# Ejemplo



---

## Vistas 4+1 (Kruchten)





## Vistas 4+1

- 4 vistas + 1 vista integradora
- Sólo propone vistas sin modelos. Cada vista referencia a modelos UML.



# Vistas



Pilippe (*Philippe*) Kruchten define: **“Una vista es una presentación de un modelo, la cual es una descripción completa de un sistema desde una perspectiva particular”.**



# Modelo 4+1 (Kruchten)



## Vista Lógica

Requisitos funcionales del sistema y de lo que el sistema debe de hacer, las funciones y servicios que se han definido.

Enfocada a lo definido como dominio de la aplicación, lo que son las clases y objetos principales que formaran el corazón o "**core**" de la aplicación.

Esta vista se complementa con los diagramas UML:

- Diagrama de Clases
- Diagrama de Paquetes



# Modelo 4+1 (Kruchten)



## **Vista de Despliegue o Vista de Desarrollo**

Mostrar básicamente como está dividido nuestro sistema de software en componentes, y muestra las dependencias entre estos componentes.

Muestra los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Organización y las dependencias entre el conjunto de componentes, y como se comunican entre ellos.



# Modelo 4+1 (Kruchten)



## **Vista de Procesos**

Representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes que conforman el sistema. También va a mostrar algunos de los requisitos no funcionales, como son ejecución, disponibilidad, tolerancia a fallas, integridad, seguridad, confiabilidad entre otros.

Esta vista la vamos a complementar con los diagramas UML:

- Diagrama de Actividad



# Modelo 4+1 (Kruchten)



## **Vista Física**

Representa cómo están distribuidos los componentes entre los distintos equipos que conforman la solución incluyendo los servicios.

Los elementos definidos en la vista lógica se mapean a componentes de software o de hardware.

Esta vista la vamos a complementar con los diagramas UML:

- Diagrama de Deployment



## Modelo 4+1 (Kruchten)

- “+1”
- Esta vista va a ser representada por los casos de uso.
- Tiene la función de unir y relacionar las otras 4 vistas.
- Trazabilidad de componentes, clases, equipos, paquetes, etc., para realizar cada caso de uso.

---

# ARQUITECTURA: VISTAS, MODELOS Y COMUNICACIÓN

Autor: Luciano Straccia

---