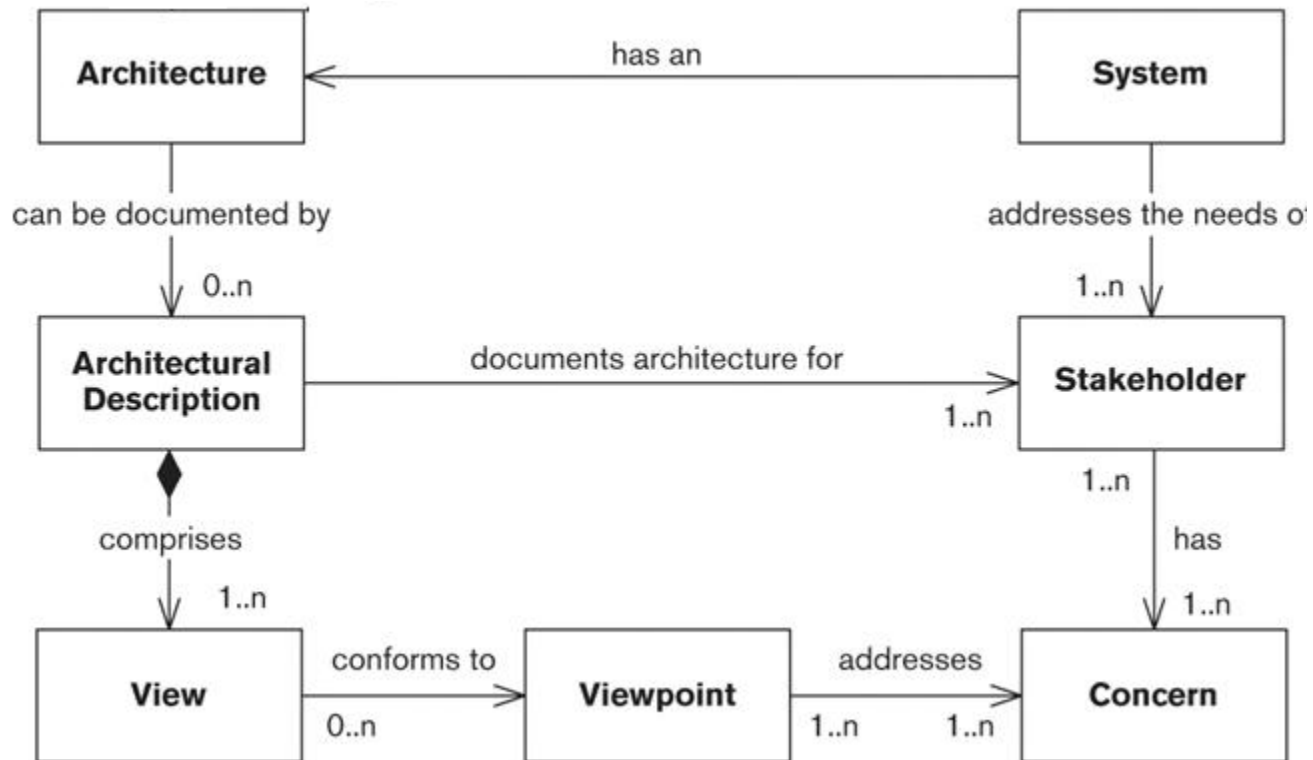
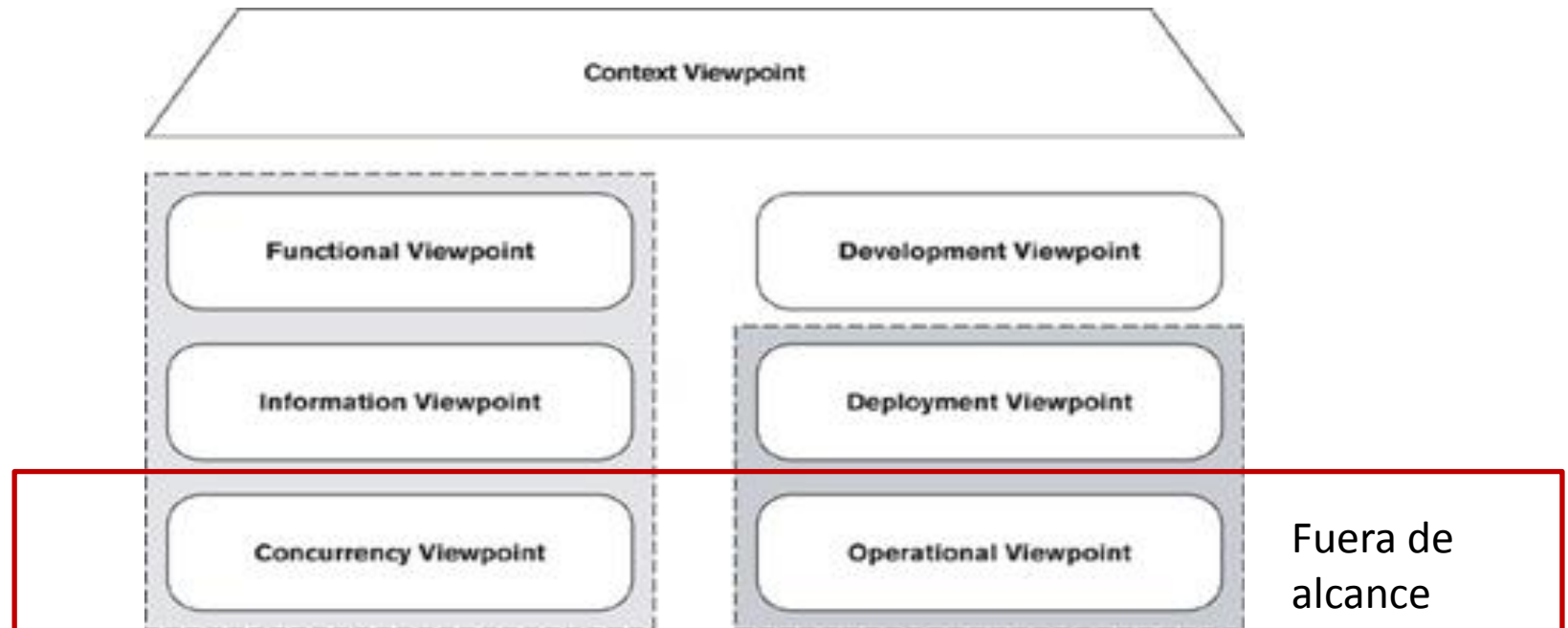


Documentación de arquitectura de software con UML 2

Vistas y puntos de vista



Catálogo de puntos de vista

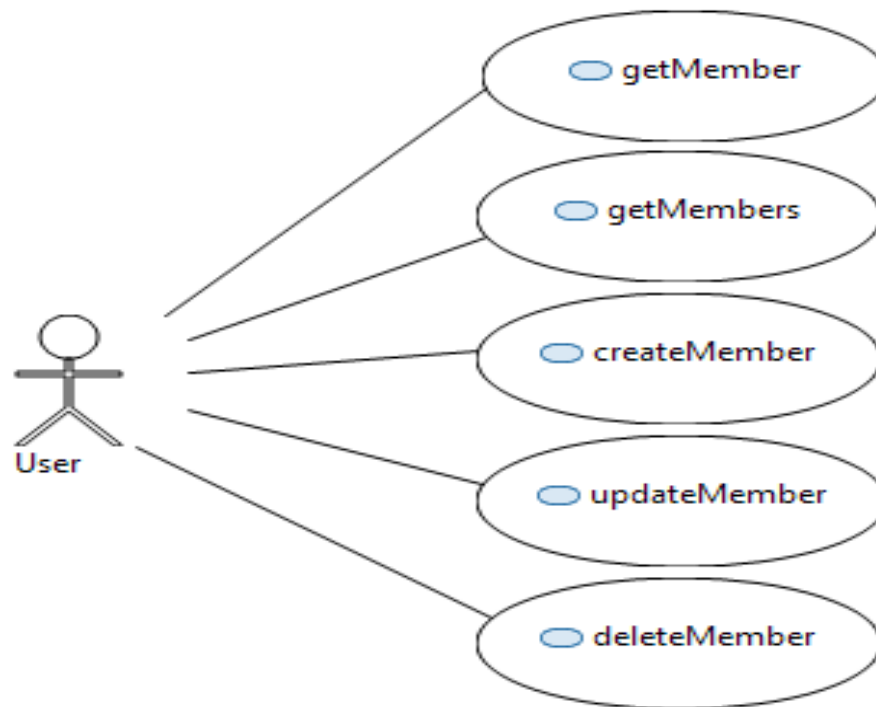


Punto de vista funcional

Definition	Describes the system's runtime functional elements and their responsibilities, interfaces, and primary interactions
Concerns	Functional capabilities, external interfaces, internal structure, and functional design philosophy
Models	Functional structure model
Problems and Pitfalls	Poorly defined interfaces, poorly understood responsibilities, infrastructure modeled as functional elements, overloaded view, diagrams without element definitions, difficulty in reconciling the needs of multiple stakeholders, wrong level of detail, "God elements," and too many dependencies
Stakeholders	All stakeholders
Applicability	All systems

Tomado de [13]

Punto de vista funcional (Casos de uso)



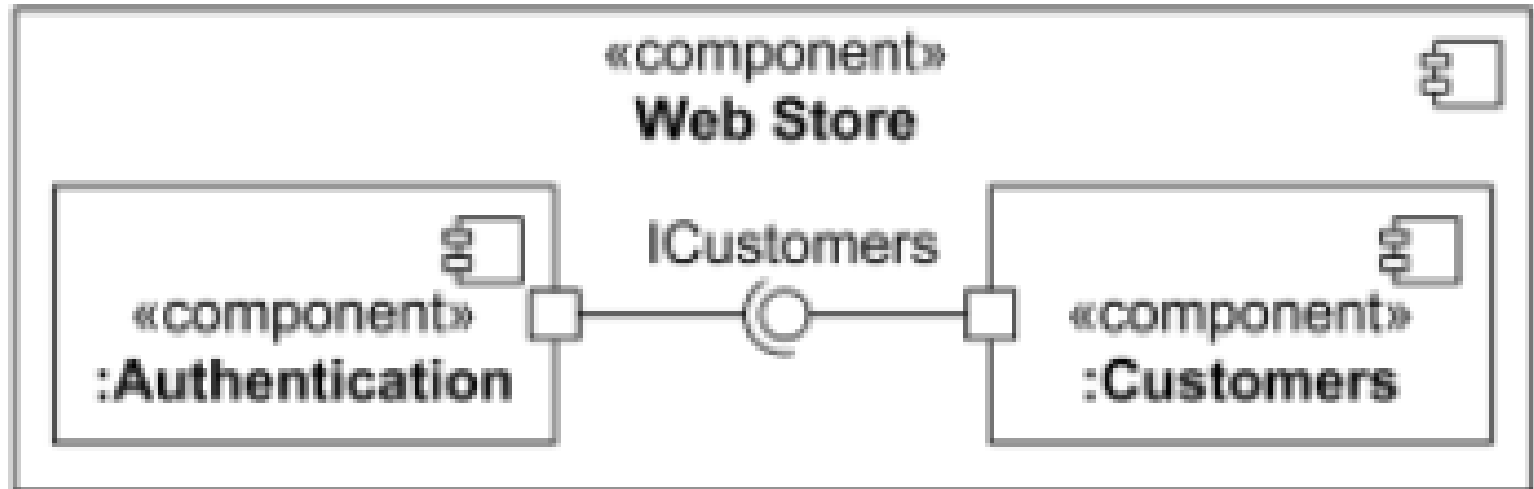
Punto de vista funcional

(Diagrama de componentes)

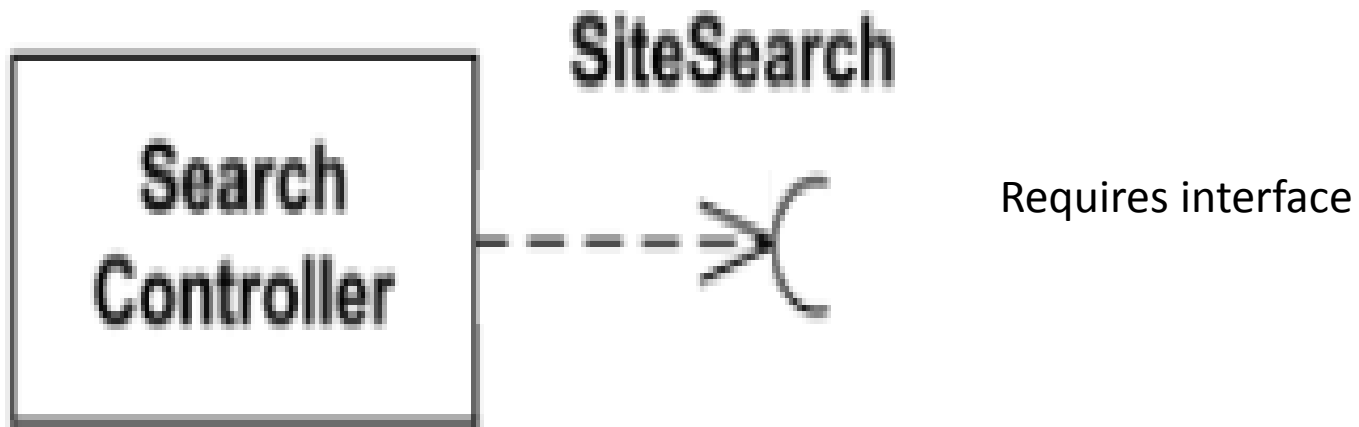
- Posee componentes y asociaciones
- Un componente es una parte modular del sistema con contenido encapsulado
- Su comportamiento está definido en términos de interfaces provistas y requeridas
- Para los componentes usar un nombre significativo que coincida con el nombre de los paquetes en el diagrama de desarrollo
- Este tipo de diagramas se puede usar con todos los stakeholders excepto con los usuarios finales

Tipos de asociaciones (Diagrama de componentes)

Assembly

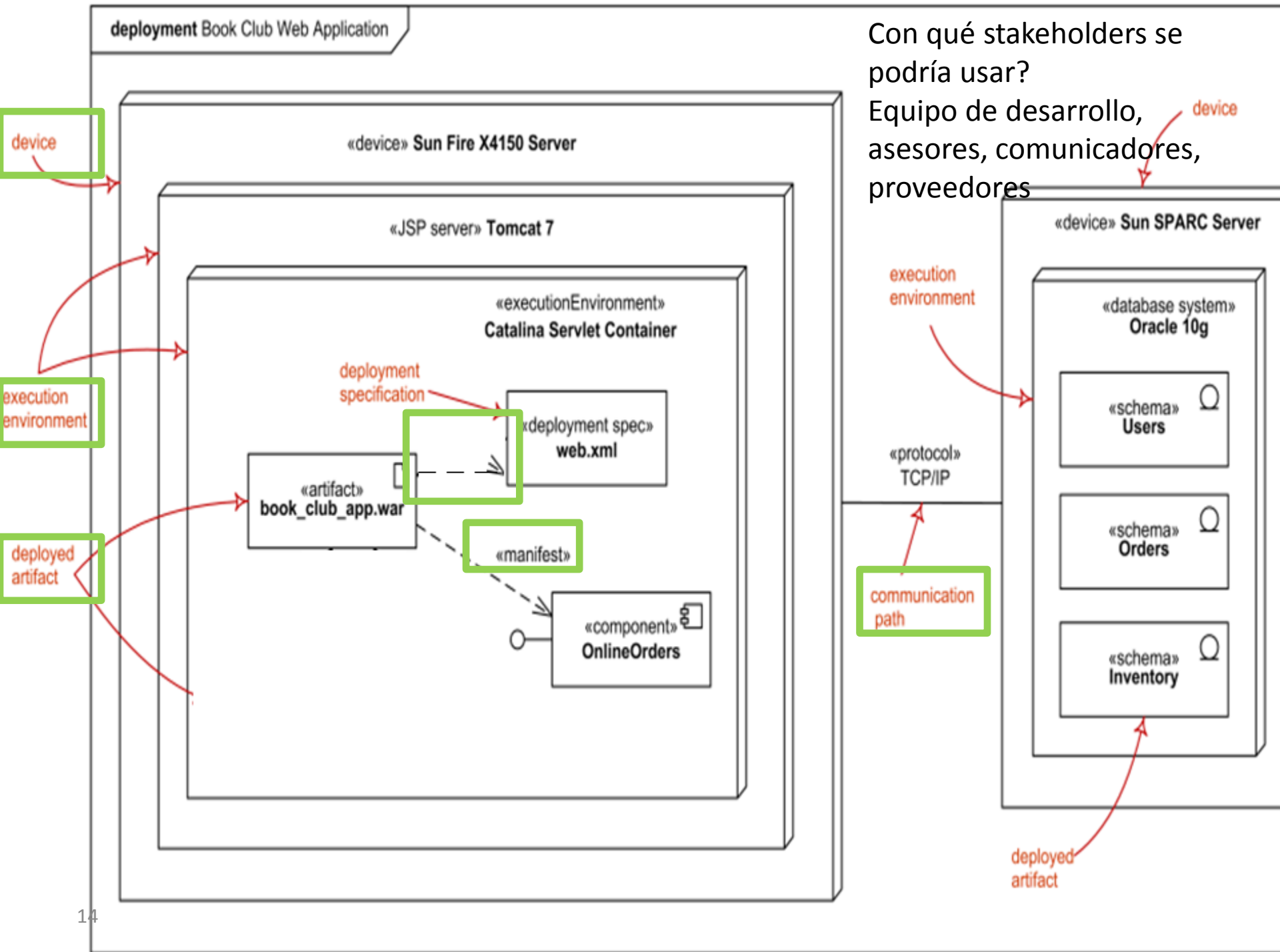


Tipos de asociaciones (Diagrama de componentes)



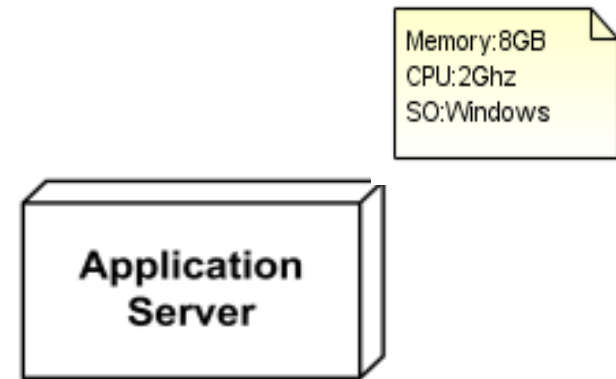
Punto de vista de despliegue

Definition	Describes the environment into which the system will be deployed and the dependencies that the system has on elements of it
Concerns	Runtime platform required, specification and quantity of hardware or hosting required, third-party software requirements, technology compatibility, network requirements, network capacity required, and physical constraints
Models	Runtime platform models, network models, technology dependency models, and intermodel relationships.
Problems and Pitfalls	Unclear or inaccurate dependencies, unproven technology, unsuitable or missing service-level agreements, lack of specialist technical knowledge, late consideration of the deployment environment, ignoring intersite complexities, inappropriate headroom provision, and not specifying a disaster recovery environment
Stakeholders	System administrators, developers, testers, communicators, and assessors
Applicability	Systems with complex or unfamiliar deployment environments



Tipos de dispositivos

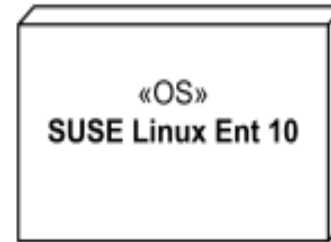
- «application server»
- «client workstation»
- «mobile device»
- «embedded device»



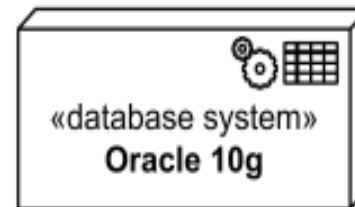
- Cada dispositivo debe tener una nota indicando las características del dispositivo, i.e., RAM, CPU, sistema operativo

Tipos de ambientes de ejecución

- «OS»
- «workflow engine»
- «database system»
- «J2EE container»
- «web server»
- «web browser»



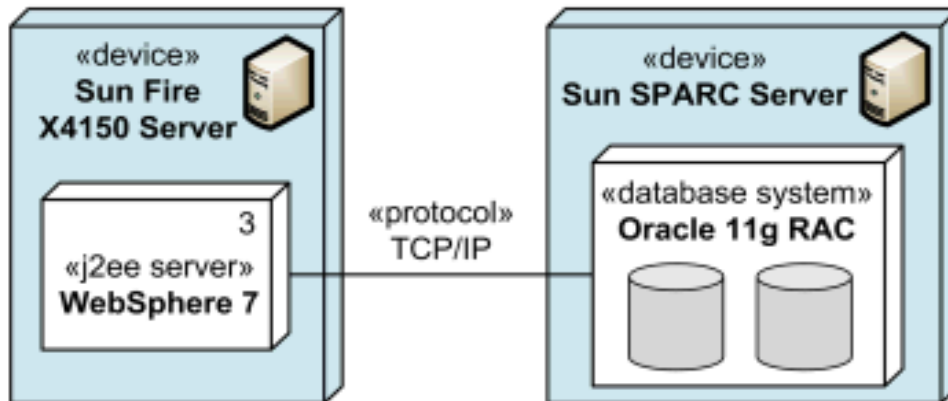
Linux Operating System Execution Environment



Oracle 10g DBMS Execution Environment

Tipos de enlace de comunicación

- Protocolos: HTTP, TCP/IP, UDP/IP, SMTP, etc.



Tipos de artefactos

«document»	A generic file that is not a «source» file or «executable».
«source»	A source file that can be compiled into an executable file.
«library»	A static or dynamic library file.
«executable»	A program file that can be executed on a computer system.
«script»	A script file that can be interpreted by a computer system.



Artifact web-app.war

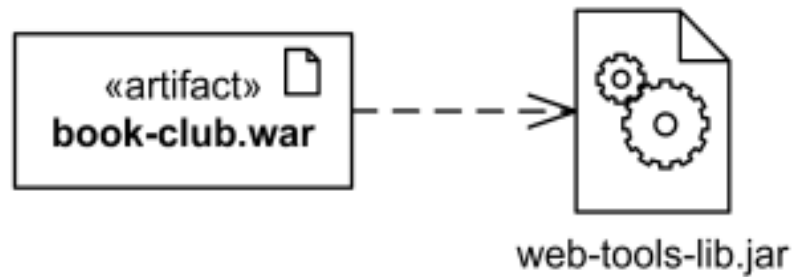


C# source file artifact UserServices.cs

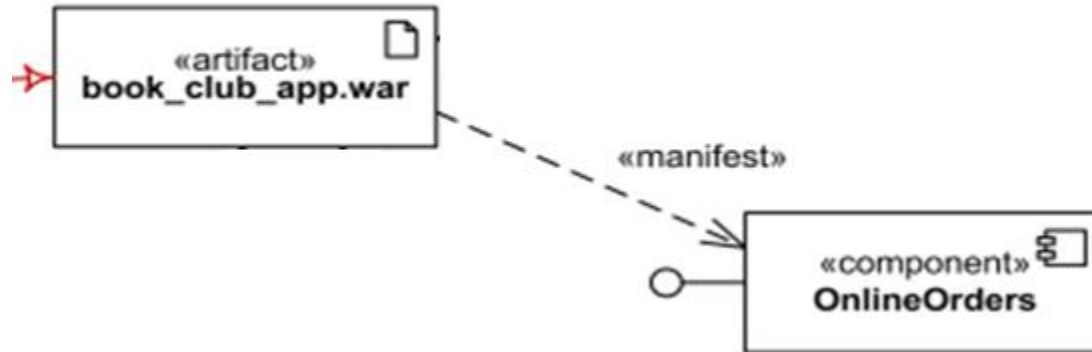


Library commons.dll

Asociación de dependencia



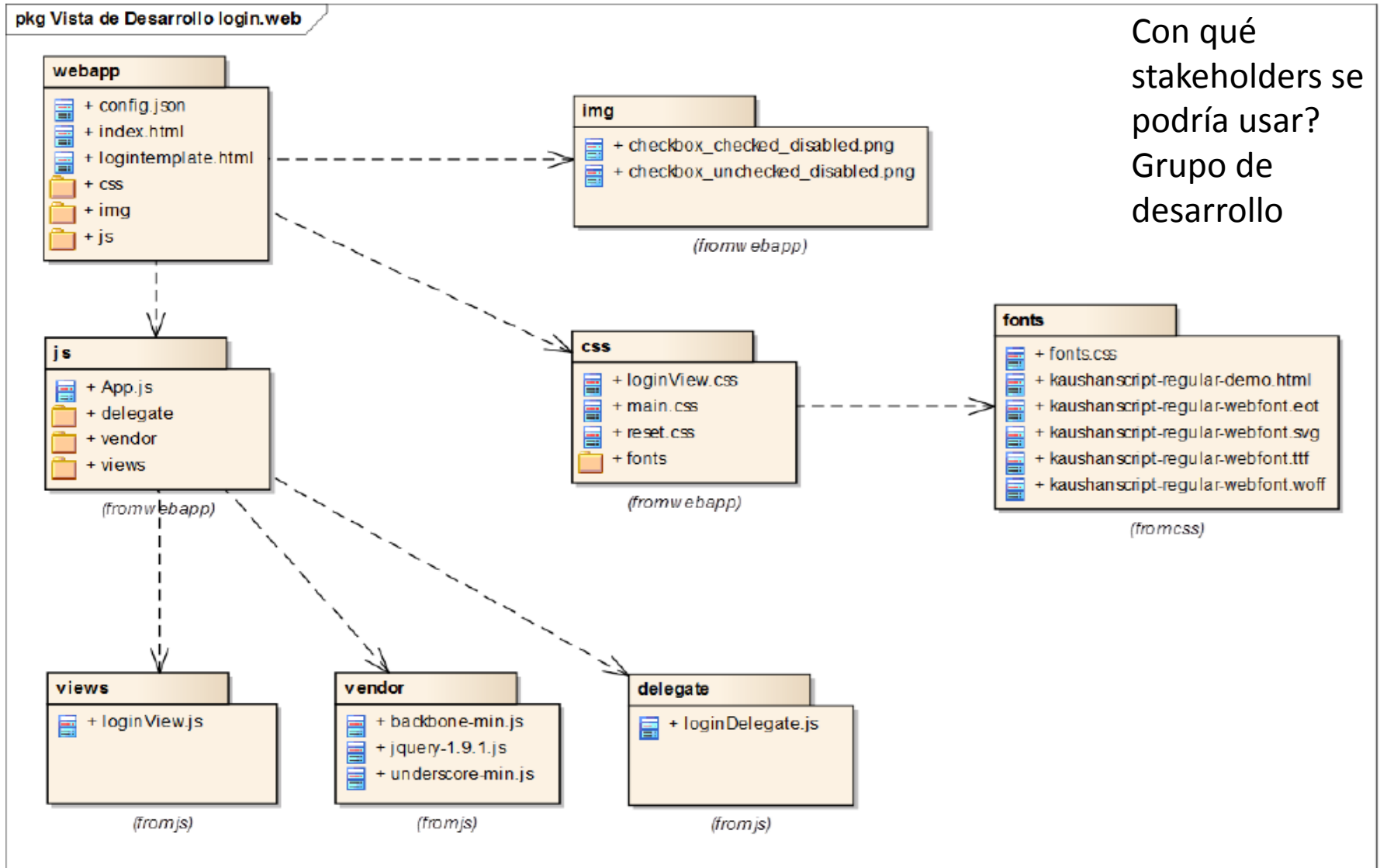
Asociación <<Manifest>>



Punto de vista de desarrollo

- Mostrar la organización del código
- Clarificar las decisiones tecnológicas y de empaquetamiento, convención de nombramiento
 - Java standalone app.
 - Clases
 - .properties
 - JEE Web app.
 - Xhtml
 - Clases
 - Descriptores (e.g., web.xml, persistence.xml)
 - Build.xml o Pom.xml
- Tener cuidado con la legibilidad del diagrama

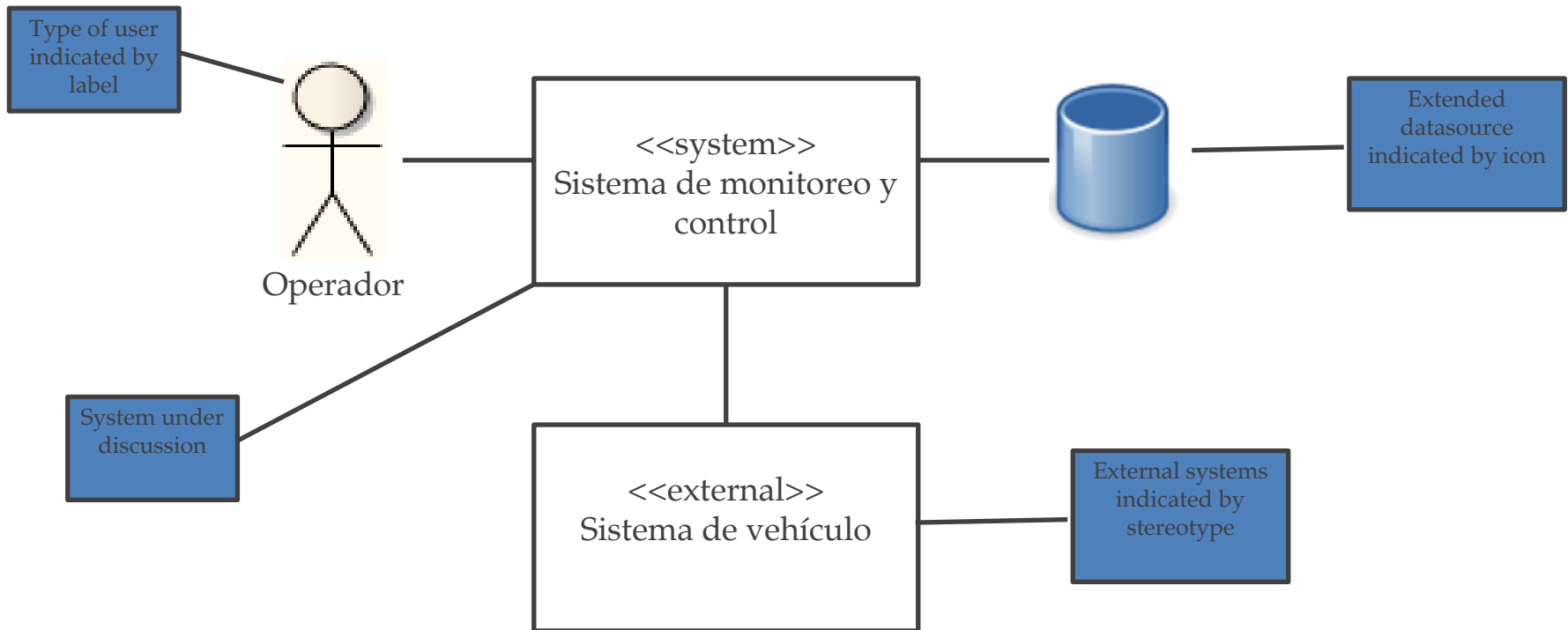
Diagrama de paquetes



Punto de vista de contexto

- Describe las relaciones, dependencias e iteraciones entre el sistema y su ambiente
 - Sistemas internos
 - Sistemas externos
- Notación ad-hoc

Diagrama de contexto



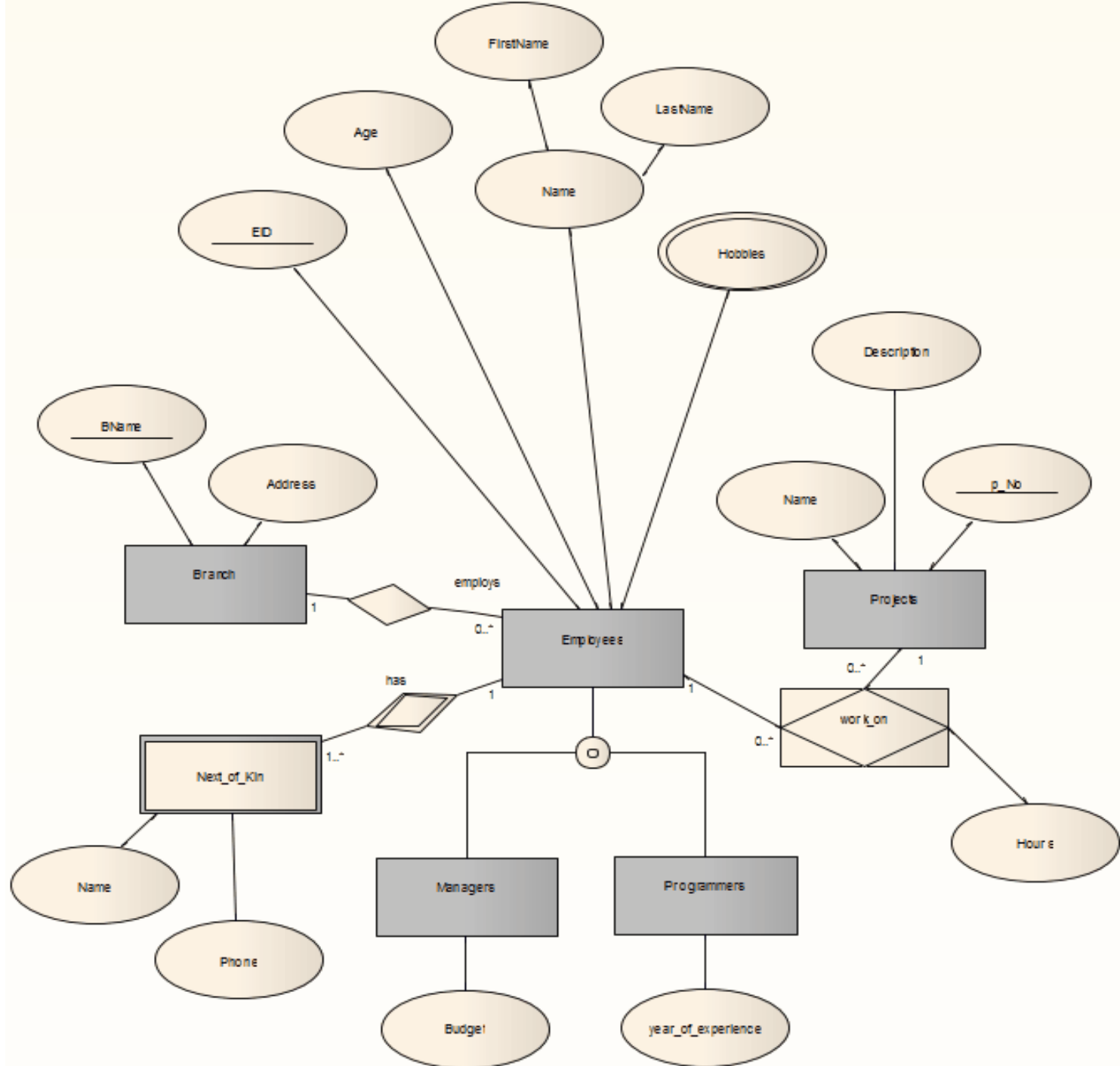
Con qué stakeholders se podría usar?
Todos

Punto de vista información

Definition	Describes the way that the system stores, manipulates, manages, and distributes information
Concerns	Information structure and content; information purpose and usage; information ownership; enterprise-owned information; identifiers and mappings; volatility of information semantics; information storage models; information flow; information consistency; information quality; timeliness, latency, and age; and archiving and information retention
Models	Static information structure models, information flow models, information lifecycle models, information ownership models, information quality analysis, metadata models, and volumetric models
Problems and Pitfalls	Representation incompatibilities, unavoidable multiple updaters, key-matching deficiencies, interface complexity, overloaded central database, inconsistent distributed databases, poor information quality, excessive information latency, and inadequate volumetrics
Stakeholders	Primarily users, acquirers, developers, testers, and maintainers, but most stakeholders have some level of interest
Applicability	Any system that has more than trivial information management needs

Diagrama entidad relación

Este es el único diagrama que no pertenece a UML2 pero se puede hacer con Enterprise Architect



Con qué stakeholders se podría usar?
Grupo de desarrollo

COHERENCIA ENTRE DIAGRAMAS

Diagrama de componentes

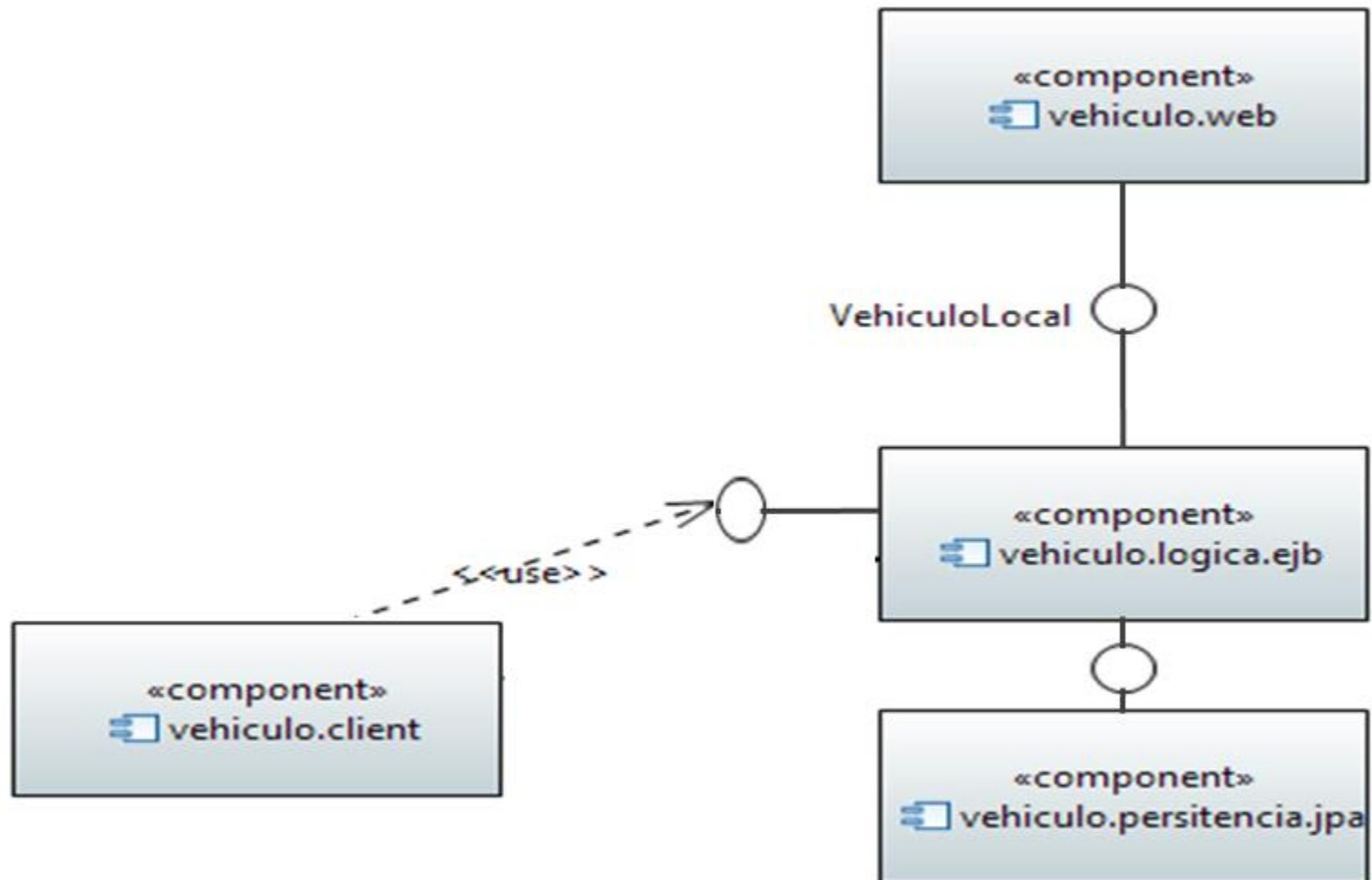


Diagrama de despliegue

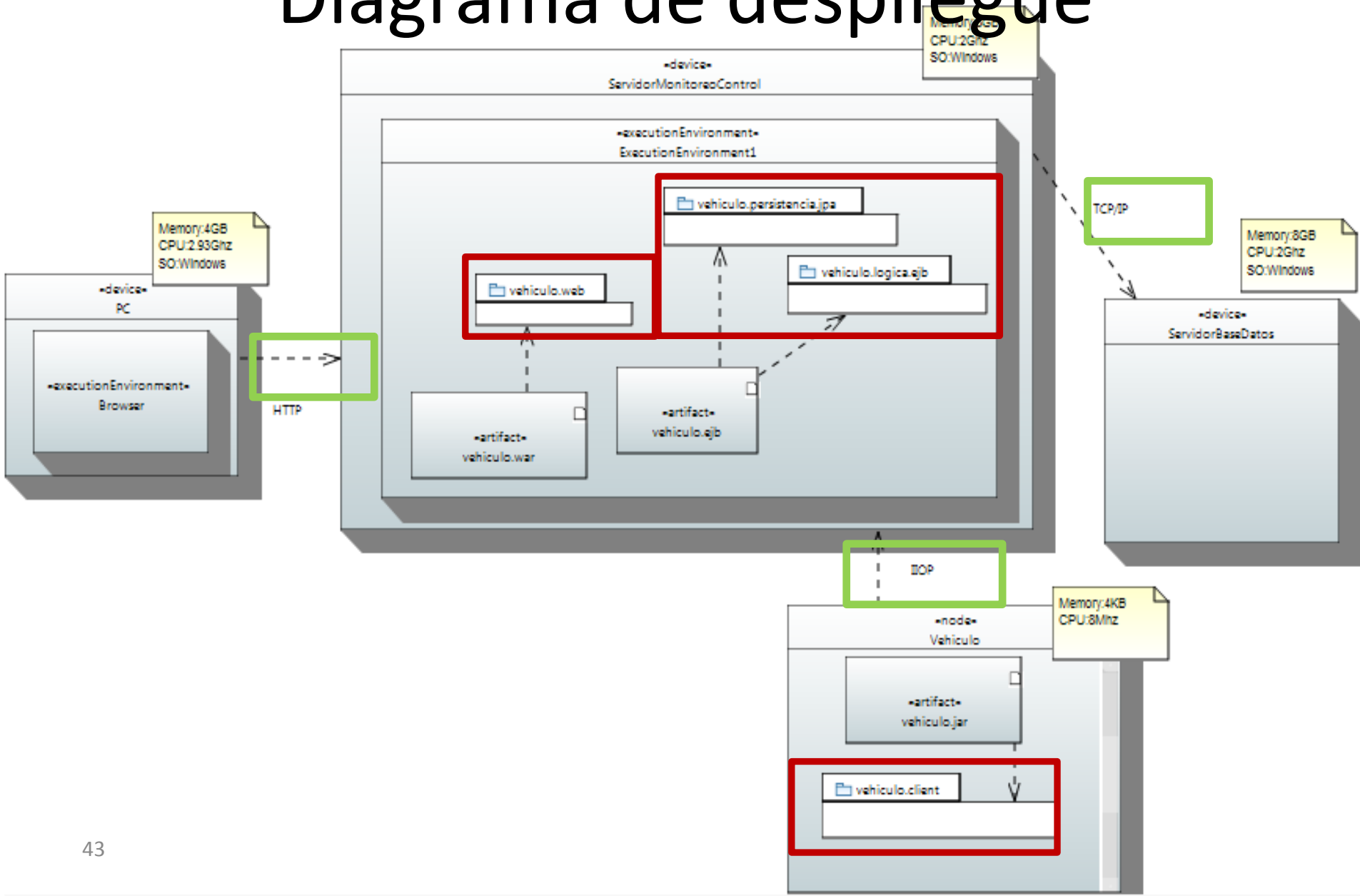
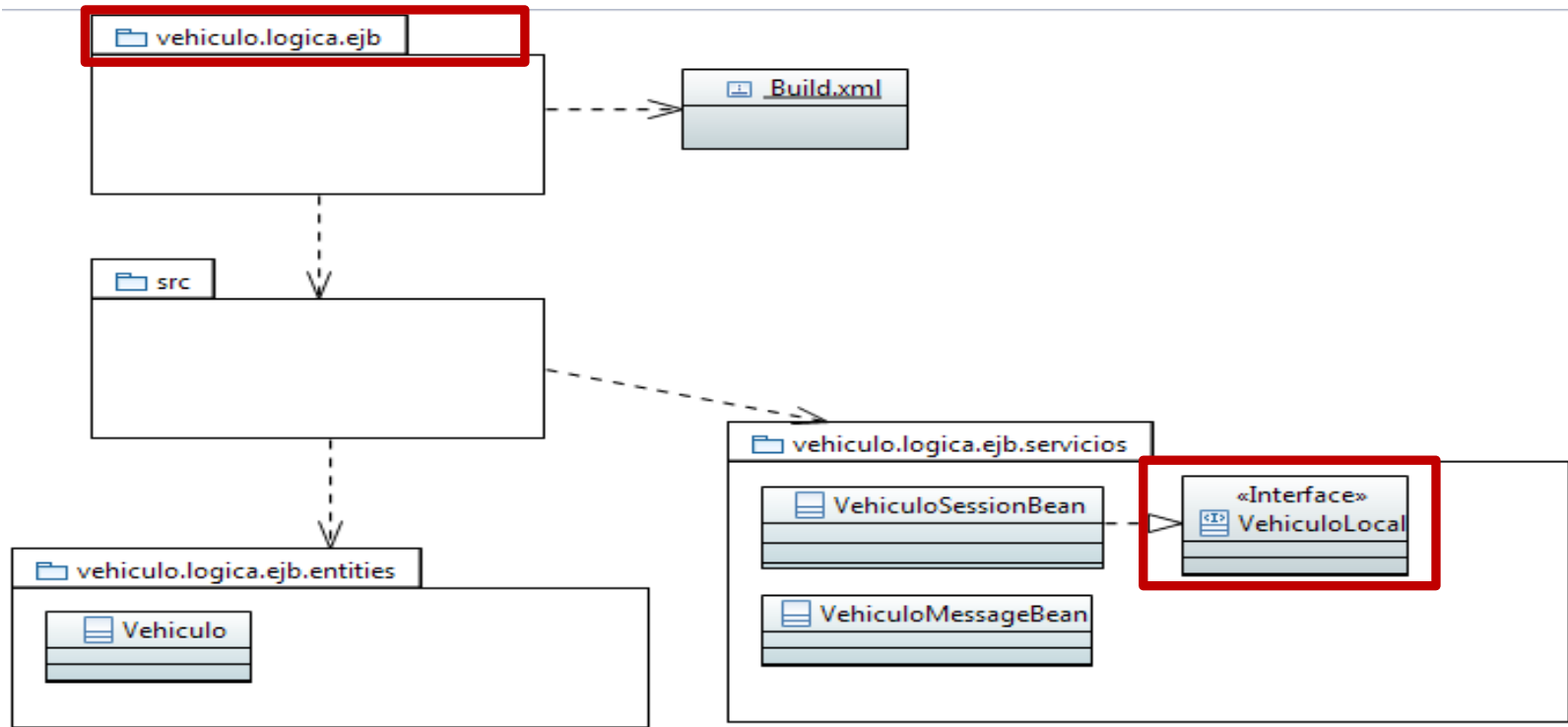


Diagrama de paquetes (desarrollo)



Nombre del paquete corresponde al nombre componente

Clasificadores sin métodos ni atributos

Errores comunes en vistas

- Vista de componentes
 - Nombre de componentes no significativo
 - Cuando la asociación es assembly o requires interface, falta especificar la interfaz o servicio
 - Uso incorrecto de las asociaciones
 - Poner a depender un componente que requiere baja latencia (e.g., gestión de alarmas, dashboard) de la base de datos

Errores comunes en vistas

- Vista de despliegue
 - Dejar <<executionEnvironment>> en el aire, es decir, sin ser incluidos en un nodo
 - Omisión de estereotipos como <<app server>>, <<embedded system>>
 - Omisión de artefactos
 - Omisión de los componentes de la vista de componentes (omisión de la asociación manifest)
 - Protocolos equívocos en las asociaciones
 - Decir en un escenario de calidad que hay disponibilidad y la vista de despliegue no muestra evidencia de ello (e.g., una nota, o un nodo <<app server>> que aparece dos veces
 - Omisión de una nota que diga las características de hw

Errores comunes en vistas

- General a las vistas de componentes y de despliegue
 - Dada una petición desde un comp. solicitante a un comp. proveedor, el último puede enviar una respuesta al primero pero eso no quiere que el proveedor deba conocer necesariamente al solicitante. Esto es, la bidireccionalidad de la asociación no se justificaría aquí.