DISEÑO DE LA ARQUITECTURA

Introducción

Herramientas de representación

Estilos arquitectónicos

Actividades del diseño arquitectónico

DISEÑO DE LA ARQUITECTURA

Introducción

Definición

El **diseño de la arquitectura** es un proceso creativo que se interesa por entender cómo se debe organizar un sistema y cómo se tiene que diseñar la estructura global de ese sistema

Características

- Es la primera etapa en el proceso de diseño del software
- Es el enlace entre el diseño y la ingeniería de requisitos
- Proporciona un modelo arquitectónico que describe cómo se organiza el sistema en un conjunto de componentes (subsistemas) de comunicación
- Es la influencia dominante para los requisitos no funcionales

DISEÑO DE LA ARQUITECTURA

Importancia de la arquitectura

- Facilita la comprensión de la estructura global del sistema
- Permite trabajar en los componente de forma independiente
- Facilita las posibles extensiones del sistema
- Facilita la reutilización de los distintos componentes

Decisiones estructurales

- Cómo se va a dividir el sistema en componentes
- Cómo deben interactuar los componentes
- Cuál va a ser la interfaz de cada componente
- Qué estilo arquitectónica se va a utilizar

DISEÑO DE LA ARQUITECTURA

El estilo arquitectónico depende de los requisitos no funcionales

- **Rendimiento**
  La arquitectura se diseña para localizar operaciones críticas dentro de un pequeño número de componentes desplegados en la misma computadora
- **Seguridad**
  La arquitectura se diseña con una estructura en capas, con los activos más críticos en las capas más internas, y con un alto nivel de validación de seguridad para esas capas
- **Protección**
  La arquitectura se diseña para que las operaciones relacionadas con la protección se ubiquen en un componente individual o en un pequeño número de componentes
- **Disponibilidad**
  La arquitectura se diseña para incluir componentes redundantes
- **Facilidad de mantenimiento**
  La arquitectura se diseña usando componentes auto contenidos de grano fino que se puedan cambiar con facilidad

DISEÑO DE LA ARQUITECTURA

Herramientas de representación

Diagrama de paquetes

Describe el sistema en torno a agrupaciones lógicas y proporciona una primera estructuración del sistema

Diagrama de componentes

Representa una estructuración concreta del sistema a partir de los componentes software (sistemas) y su interrelación (interfaces)

Diagrama de despliegue

Especifica el hardware físico sobre el que se ejecutará el sistema software y cómo cada subsistema software se despliega en ese hardware

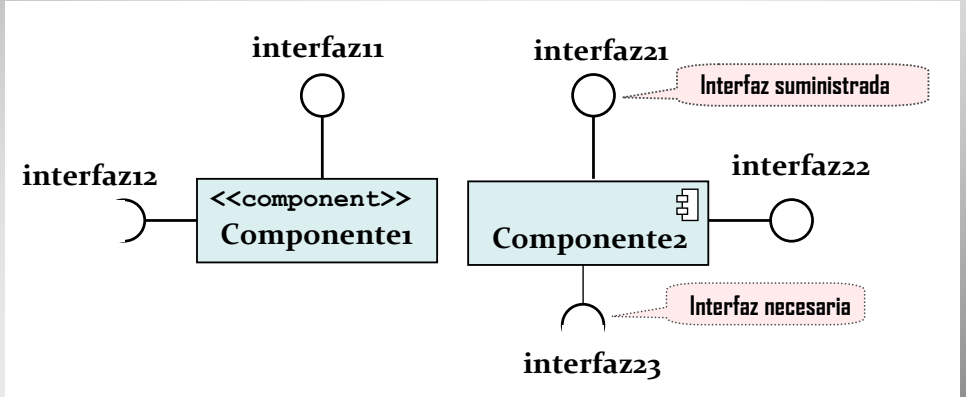
DISEÑO DE LA ARQUITECTURA

Diagrama de componentes

Componente

Unidad de software que ofrece una serie de servicios a través de una o varias interfaces

Notación gráfica



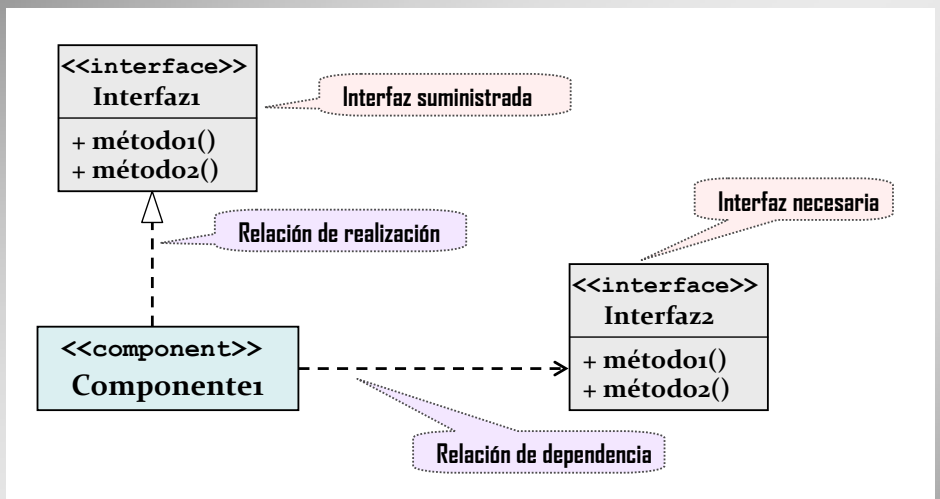
DISEÑO DE LA ARQUITECTURA

Estereotipos estándar de componentes

Estereotipo	Semántica
<<buildcomponent>>	Componente que define un conjunto de elementos para fines organizativos
<<entity>>	Componente de información persistente que representa un concepto de negocio
<<implementation>>	Componente que no tiene especificación
<<specification>>	Componente que especifica un dominio de objetos sin definir su implementación
<<process>>	Componente basado en transacción
<<service>>	Componente funcional sin estado que computa un valor
<<subsystem>>	Unidad de descomposición jerárquica para grandes sistemas

DISEÑO DE LA ARQUITECTURA

Representación alternativa de las interfaces



DISEÑO DE LA ARQUITECTURA

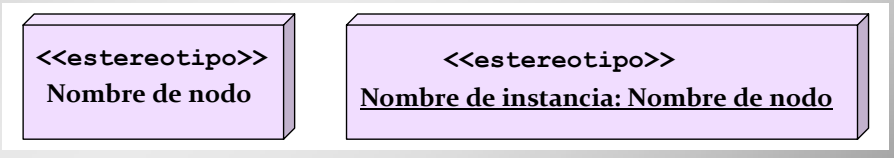
Diagrama de despliegue

Elementos de un diagrama de despliegue

- **Nodos**
  - Representan tipos de recursos computacionales sobre los que se pueden desplegar los artefactos para su ejecución
  - Los nodos se pueden anidar en nodos
  - Una instancia de nodo representa un recurso computacional específico

Estereotipo	Semántica
`<<device>>`	Tipo de dispositivo físico como un PC o un servidor
`<<execution enviroment>>`	Tipo de entorno de ejecución para software

Notación gráfica



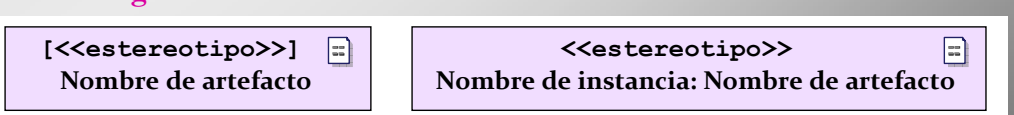
- **Asociaciones entre nodos**
  - Representan canales de comunicación a través de los cuales se puede pasar información

DISEÑO DE LA ARQUITECTURA

- **Artefactos**
  - Representan especificaciones de elementos concretos del mundo real
  - Los artefactos se despliegan en nodos
  - Ejemplos: archivos fuente, archivos ejecutables, tablas de base de datos, ...
  - Una instancia de artefacto representa una instancia específica de un artefacto determinado

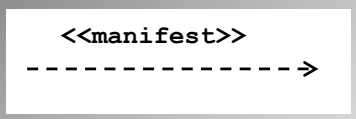
Estereotipo	Semántica
`<<file>>`	Archivo físico
`<<deployment spec>>`	Especificación de detalles de despliegue
`<<document>>`	Archivo genérico que contiene cierta información
`<<executable>>`	Archivo de programa ejecutable
`<<library>>`	Biblioteca estática o dinámica
`<<script>>`	Script que se puede ejecutar por un intérprete
`<<source>>`	Archivo fuente que se puede compilar en un archivo ejecutable

Notación gráfica



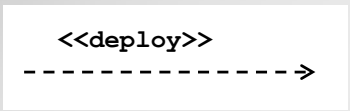
DISEÑO DE LA ARQUITECTURA

- Relaciones de dependencia



Representa la relación entre un artefacto y el elemento o elementos del modelo que implementa

Ejemplo



Representa el despliegue de un artefacto o instancia de artefacto sobre un nodo o instancia de nodo

Ejemplo

DISEÑO DE LA ARQUITECTURA

Estilos arquitectónicos

Proporcionan un conjunto de subsistemas predefinidos, especificando sus responsabilidades e incluyendo reglas y guías para organizar las relaciones entre ellos. No proporcionan la arquitectura del sistema, sino una guía de cómo obtenerla

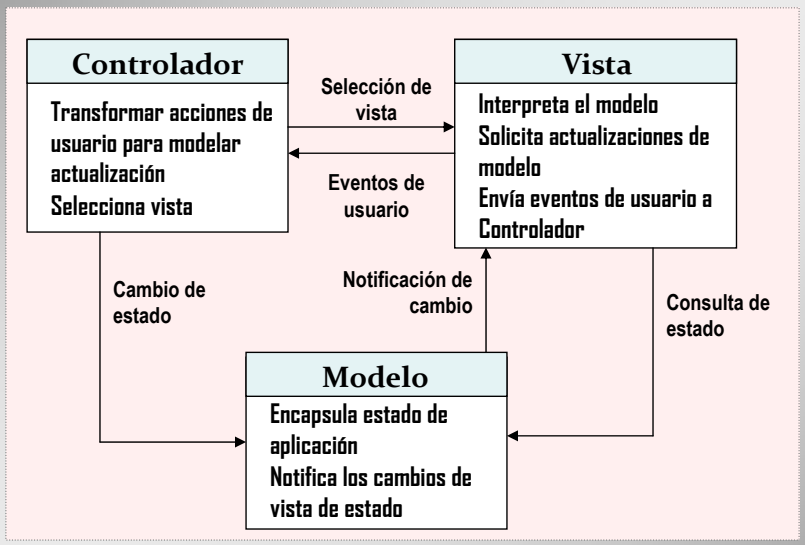
Estilos arquitectónicos más generalizados

- Arquitectura Modelo-Vista-Controlador (MVC)
- Arquitectura en capas
- Arquitectura de repositorio
- Arquitectura cliente-servidor

DISEÑO DE LA ARQUITECTURA

Arquitectura Modelo-Vista-Controlador

Separa presentación e interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interaccionan entre sí



Modelo

Maneja los datos del sistema y las operaciones asociadas a esos datos

Vista

Define y gestiona cómo se presentan los datos al usuario

Controlador

Dirige la interacción del usuario y pasa estas interacciones a Vista y Modelo

---

DISEÑO DE LA ARQUITECTURA

Principios de diseño

- Cada subsistema puede diseñarse independientemente
- Se aumenta la cohesión de los subsistemas si la Vista y el Controlador se unen en una capa de interfaz de usuario
- Se reduce el acoplamiento puesto que la comunicación entre los subsistemas es mínima

Cuándo se usa

- Cuando existen múltiples formas de interactuar con los datos
- Cuando se desconocen los requisitos futuros para la interacción y presentación

Ventajas

- Permite que los datos cambien de manera independiente de su representación (y viceversa)
- Soporta diferentes representaciones de los mismos datos
- Los cambios en una representación se muestran en todos ellos

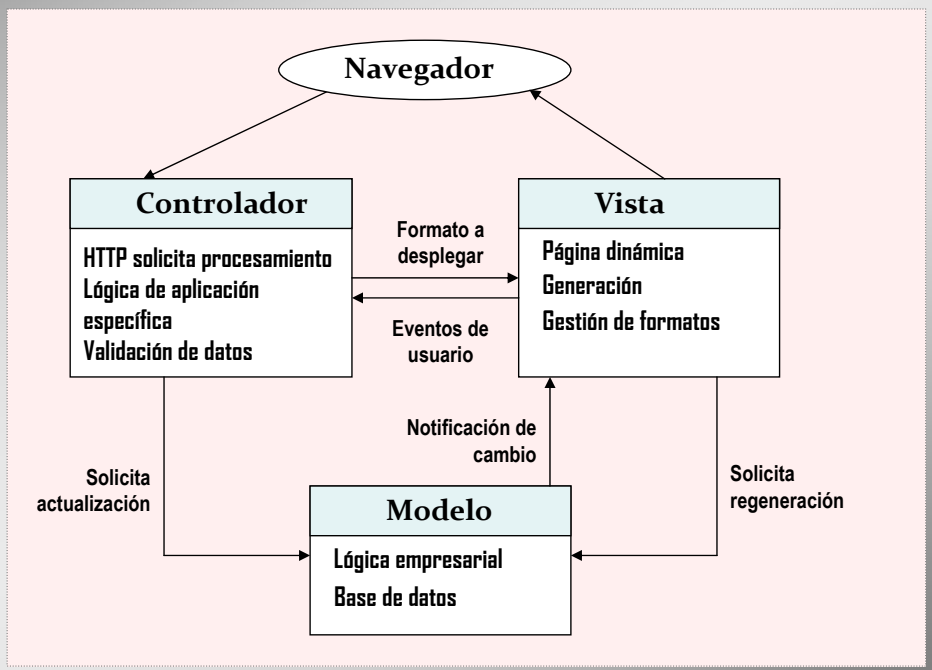
Desventajas

- Código adicional y complejidad de código cuando el modelo de datos y las interacciones son simples

DISEÑO DE LA ARQUITECTURA

Ejemplo

Arquitectura de aplicación Web con el estilo MVC



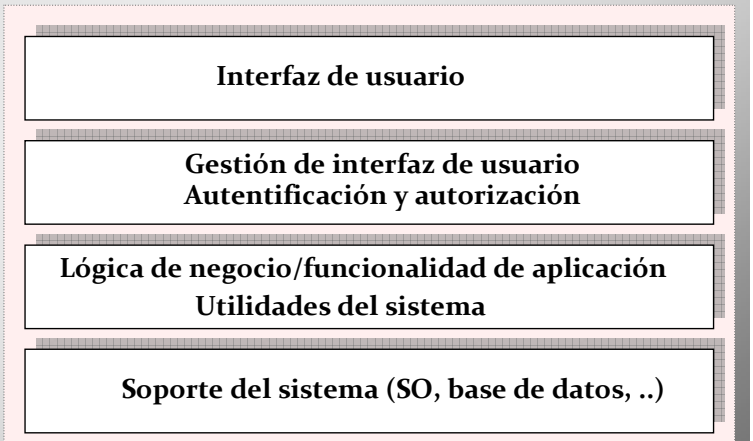
DISEÑO DE LA ARQUITECTURA

Arquitectura en capas

Organiza el sistema en capas con funcionalidad relacionada con cada capa. Una capa da servicios a la capa de encima y las capas de nivel inferior representan servicios núcleo que es probable que se utilicen a lo largo de todo el sistema

Principios de diseño

- Las capas se pueden diseñar, construir y probar independientemente
- Una capa bien diseñada presenta alta cohesión
- Una capa bien diseñada no tiene conocimiento de las capas superiores (ocultamiento de información)
- Las capas deben estar desacopladas
  - Todas las dependencias en un sentido
  - Todas las dependencias en las interfaces
- Las capas inferiores se deben diseñar para prestar servicios de bajo nivel (bajo acoplamiento)



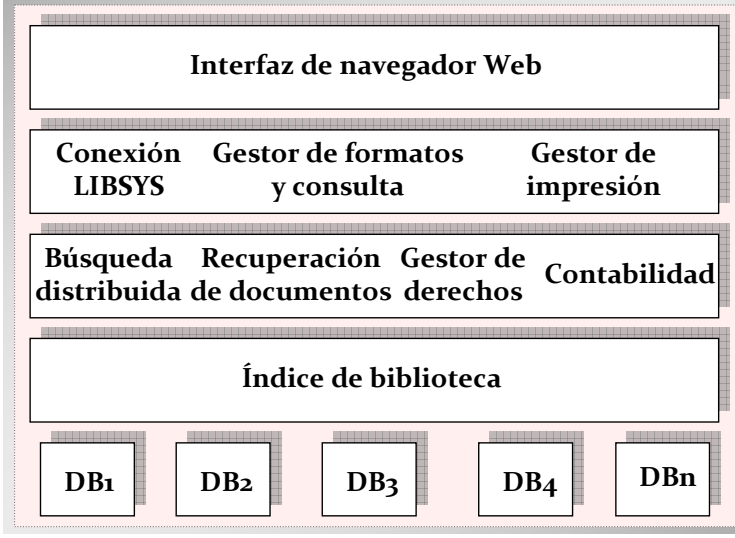
DISEÑO DE LA ARQUITECTURA

Cuándo se usa

- Al construir nuevas facilidades encima de los sistemas existentes
- Cuando el desarrollo se dispersa a través de varios equipos de trabajo
- Cuando existe un requisitos de seguridad multinivel

Ejemplo

Sistema para compartir documentos con derechos de autor



Ventajas

- Permite la sustitución de capas completas siempre que se conserve la interfaz
- En cada capa se pueden incluir facilidades redundantes

Desventajas

- Es difícil ofrecer una separación limpia entre capas
- El rendimiento es un problema debido a los múltiples niveles de interpretación

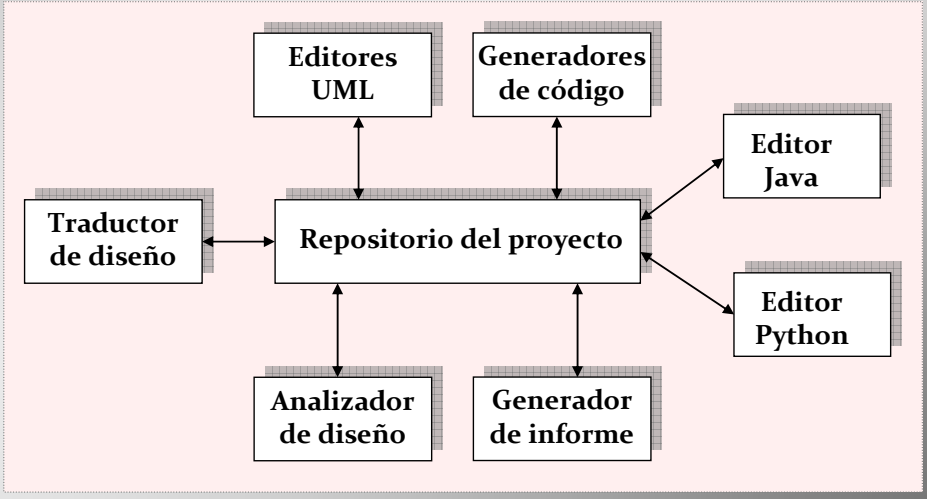
DISEÑO DE LA ARQUITECTURA

Arquitectura de repositorio

Todos los datos en un sistema se gestionan en un repositorio central, accesible a todos los componentes. Los componentes no interactúan directamente, sino sólo a través del repositorio

Ejemplo

Herramienta CASE



DISEÑO DE LA ARQUITECTURA

Principios de diseño

- Se pueden diseñar subsistemas independientes, aunque deben conocer el esquema del repositorio
- Cada subsistema tiene definida una funcionalidad específica (alta cohesión)
- El acoplamiento de los subsistemas con el repositorio es alto

Cuándo se usa

- Cuando se tiene un sistema donde los grandes volúmenes de información generados se deben almacenar durante mucho tiempo
- En sistemas en los que la inclusión de datos en el repositorio active una acción o herramienta

Ventajas

- Los componentes puedes ser independientes, no necesitan conocer la existencia de otros
- Los cambios en un componente se pueden propagar hacia todos los componentes
- La totalidad de datos se puede gestionar de manera consistente

Desventajas

- Los problemas en el repositorio afectan a todo el sistema
- Existencia de ineficiencias al organizar toda la comunicación a través del repositorio
- Quizá sea difícil distribuir el repositorio en varias computadoras

---

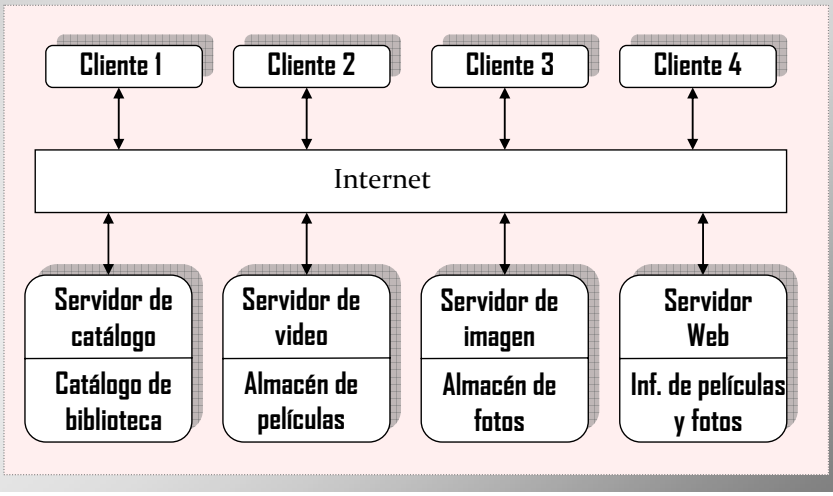
DISEÑO DE LA ARQUITECTURA

Arquitectura cliente-servidor

La funcionalidad del sistema se organiza en servicios, y cada servicio lo entrega un servidor independiente. Los clientes son usuarios de dichos servicios y para usarlos ingresan a los servidores

Ejemplo

Filmoteca y videoteca



DISEÑO DE LA ARQUITECTURA

Principios de diseño

- Permite diseñar, implementar y probar servidores y clientes independientemente
- Mejora la cohesión de los subsistemas al proporcionar servicios específicos a los clientes
- Reduce el acoplamiento al establecer un canal de comunicación para el intercambio de mensajes
- Aumenta la abstracción al tener subsistemas distribuidos en nodos separados

Cuándo se usa

- Cuando desde varias ubicaciones se tiene que acceder a una base de datos compartida
- Cuando la carga de un sistema es variable

Ventajas

- Los servidores se pueden distribuir a través de una red
- Se pueden añadir fácilmente servidores o clientes extras
- Se pueden escribir clientes para nuevas plataformas sin modificar a los servidores

Desventajas

- Es susceptible a fallos del servidor
- El rendimiento es impredecible porque depende de la red
- Problemas administrativos cuando los servidores sean propiedad de distintas organizaciones

---

DISEÑO DE LA ARQUITECTURA

Actividades del diseño arquitectónico

- Identificar los objetivos del diseño
- Determinar la arquitectura software, y los correspondientes subsistemas
- Modelar la arquitectura software
- Refinar la descomposición en subsistemas

DISEÑO DE LA ARQUITECTURA

Identificar los objetivos del sistema

- Identificar las cualidades deseables del sistema
- Obtener a partir de los requisitos no funcionales
- Seleccionar un pequeño conjunto de objetivos de diseño que el sistema debe satisfacer necesariamente
- Adquirir compromisos, puesto que muchos objetivos de diseño son contrapuestos

DISEÑO DE LA ARQUITECTURA

Determinar la arquitectura software

- Seleccionar el estilo arquitectónico que mejor se adapte
- Identificar subsistemas en el dominio del problema
- Añadir subsistemas predefinidos de acuerdo al estilo arquitectónico seleccionado

Modelar la arquitectura software

- Diseñar la arquitectura en un diagrama de paquetes
- Elaborar el diagrama de componentes de la arquitectura
- Realizar el diagrama de despliegue

DISEÑO DE LA ARQUITECTURA

Diseñar la arquitectura en un diagrama de paquetes

Los diagramas de paquetes permiten modelar una primera estructuración del sistema en base a uno o más paquetes

- Cada paquete agrupa a un conjunto de clases relacionadas semánticamente
- Los paquetes pueden guiar en la identificación de los subsistemas y/o componentes reutilizables

Al identificar los paquetes hay que tener en cuenta los siguientes aspectos

- Identificar paquetes cohesivos y con poca interacción con otros paquetes de acuerdo con la arquitectura planteada
- Diseñar paquetes que puedan reutilizarse en otros proyectos
- Integrar subsistemas de proyectos anteriores
- Evitar las dependencias cíclicas entre paquetes

DISEÑO DE LA ARQUITECTURA

Elaborar el diagrama de componentes

Los diagramas de componentes permiten estructurar el sistema en subsistemas en base a componentes que pueden reemplazarse

A partir del diagrama de paquetes determinar qué partes pueden definir componentes, para ello habrá que

- Definir interfaces suministradas de cada componente para determinar las operaciones que pueden ser usadas por otros componentes
- Especificar las interfaces necesarias en cada componentes para poder desarrollar su funcionalidad
- Construir el componente de forma que su contenido interno sea independiente del exterior, salvo a través de las interfaces suministradas y necesarias

DISEÑO DE LA ARQUITECTURA

Refinar la descomposición en subsistemas

- Refinar los subsistemas hasta satisfacer los objetivos de diseño
- Establecer los siguientes aspectos genéricos
  - Correspondencia hardware-software
  - Administración de datos persistentes
  - Especificación de una política de control de accesos
  - Diseño del flujo de control
  - Diseño de la concurrencia y sincronización
  - Definición de las condiciones del entorno