

Modelo de Arquitectura de Software

UNIDAD 10

Análisis y Diseño de Sistemas de Información

Arquitectura de Software

- Forma parte del diseño del sistema a partir de los Requerimientos
- Es una de las etapas, incluyendo el Diseño Detallado, la Implementación y la Verificación

Definición Formal

- Una arquitectura de software se entiende como la estructura de un sistema, que contiene componentes, sus propiedades visibles a otros componentes y las relaciones entre ellos

Componentes de un Sistema

- Un sistema complejo estará formado por sub sistemas
- La arquitectura del Software se encarga de:
 - Los sub sistemas que componen el sistema
 - Las interfaces y las formas en que se comunican entre ellos

Utilidad

- Permite una comunicación entre los interesados en el sistema
- Permitirá tomar decisiones tempranas de diseño
- Facilita el reuso de componentes

Arquitectura y Calidad

- La arquitectura impacta en la forma en que un sistema alcanza sus requerimientos de calidad
 - Las cualidades del producto forman parte del diseño de la arquitectura
 - La arquitectura, permite, pero no garantiza que cualquier requisito de calidad se alcance

Arquitectura y Funcionalidad

- La funcionalidad está relacionada con los requisitos de calidad
- La funcionalidad indica que el sistema debe hacer
- La arquitectura permite tener calidad en el sistema al momento que implementa la funcionalidad

Construcción del Modelo

La Importancia de la Calidad

- Los atributos de calidad del software impactan en el diseño de la arquitectura
 - Inter operabilidad
 - Disponibilidad
 - Seguridad
 - Portabilidad entre otros

Impacto de los Interesados

- Los requisitos de calidad que solicita un cliente difícilmente se documentan
- Esto provoca:
 - Objetivos que no se alcanzan
 - Conflictos entre los interesados
- Los interesados impactan en:
 - Entender las restricciones reales del sistema
 - Toma de decisiones basados en compromisos
 - Negociación en base a compromisos

Creación de una Arquitectura

- Existen diversos métodos y guías para la definición de una arquitectura que consideran los requisitos funcionales y también los atributos de calidad

Definición de Requerimientos

- Creación de un modelo a partir de los requerimientos que serán la base del diseño de la arquitectura basado en atributos de calidad

Diseño de la Arquitectura

- Define la estructura y las funciones o responsabilidades de los componentes que formarán la arquitectura

Validación

- Realizar pruebas a la arquitectura validando el diseño contra los requerimientos actuales y los que pudieran existir a futuro.

Identificación de Requerimientos

- La identificación se basa en requerimientos no funcionales
 - Como se comporta el software, cómo lo hace, no qué es lo que hace
 - Pueden contener restricciones
 - Corresponden a los requerimientos URPS+

Requerimientos No Funcionales

- Facilidad de Uso (Usability). Factores humanos, ayuda del sistema, documentación, claridad de las interfaces
- Fiabilidad (Reliability). Frecuencia de fallos, capacidad de recuperación de las fallas
- Rendimiento (Performance). Tiempo de Respuesta, productividad, disponibilidad
- Soporte (Support). Adaptabilidad, facilidad de mantenimiento, internacionalización, facilidad de modificación
- + . Seguridad, manejo de recursos

Restricciones

- Una restricción es una condición que no es negociable e impacta en el diseño de la arquitectura
- Pueden ser de negocio de desarrollo, de tiempo, costos, etc.
- Por ejemplo:
 - El almacenamiento usará Oracle ya que es la Base de Datos institucional
 - Los encargados de soporte de la empresa manejan C#, por lo que será el lenguaje a desarrollar

Diseño de la Arquitectura

- Una vez identificados los requerimientos de la arquitectura y las restricciones:
 - Elegir la Arquitectura de Referencia
 - Definir los componentes que formarán parte del diseño
 - Generar las vistas que representan el diseño de la arquitectura
 - Documentar el diseño de la arquitectura

Consideraciones

- Elegir los estilos y patrones más apropiados que permitan lograr la calidad esperada
- Basarse en patrones o arquitecturas reconocidas
- Considerar el tamaño de la aplicación objetivo
 - Una pequeña puede basarse en un patrón
 - Complejas pueden requerir una mezcla

Patrones

- Solución que puede utilizarse de manera repetida a problemas comunes
- Solución a un problema no trivial que es efectiva y reusable
 - Ya ha resuelto un problema similar de manera correcta
 - Se puede aplicar a problemas de diseño similares

Tipos de Patrones

- Patrones de Alto Nivel
 - Patrones de Arquitectura
- Patrones de Nivel Medio
 - Patrones de Diseño
- Patrones de Bajo Nivel
 - Idioms

Selección de la Arquitectura

Patrones de Alto Nivel

- Se refieren a una forma de resolver problemas de arquitectura
- Proporciona un conjunto de tipos de elementos ya establecidos
- Especifica como se organizan las relaciones entre estos elementos
- Es común que se combinen varios patrones al momento de diseñar una arquitectura

Estilos Arquitectónicos

- Describen:
 - Los componentes (módulos, bases de datos, etc) que tienen una función específica
 - Conectores que permiten la colaboración entre componentes
 - Restricciones en la integración de componentes

Estilos Arquitectónicos

- Arquitectura Centrada en Datos
- Arquitectura de Flujo de Datos
- Arquitecturas de Llamada y Retorno
- Arquitecturas Orientadas a Objetos
- Arquitecturas en Capas

Llamadas y Retorno

- Se basa en una comunicación de llamadas y retornos entre módulos
- Se divide en arquitectura de programa principal y sub programas y Llamadas a Procedimientos Remotos (basado en el modelo Cliente - Servidor)

Centrada en Datos

- Se enfoca en el almacenamiento de datos al que tienen acceso otros componentes
- Se basa en un almacenamiento central y los componentes tienen acceso a él
 - Arquitectura del Pizarrón (Blackboard)

Arquitectura de Flujo de Datos

- Se utiliza cuando hay transformaciones en secuencia sobre ciertos datos
- Se basa en el concepto de tuberías, los componentes (filtros) se comunican a través de conectores (*pipe*) o tuberías

Orientación a Objetos

- Los componentes de un sistema son datos y operaciones encapsulados y se comunican a través del paso de mensajes

Arquitectura en Capas

- Se tienen varias capas, cada capa se comunica con las capas inferiores o superiores a través del paso de información y servicios

Patrones de Diseño

- Es una descripción de clases y objetos y la manera en que se comunican entre ellos
- A través de modificaciones, se adaptan para resolver ciertos problemas de diseños de un mismo contexto

Componentes de un Patrón

- Nombre.
 - Describe el problema
- Problema
 - Especifica cuando aplicar el patrón
- Solución
 - Elementos que componen el diseño, como se relacionan y que responsabilidades tienen asignadas

Tipos de Patrones

- De creación. Definen como se crearán los objetos
- De estructura. Se refieren a la estructura de clases y objetos
- De Comportamiento. Indican la forma en que interactúan las clases u objetos

Idioms

- Son los patrones de más bajo nivel
- Se enfocan en problemas específicos relacionados con la implementación en un determinado lenguaje
- Ejemplos:
 - Convenciones de nombres
 - Formatos para código fuente
 - Manejo de memoria

Componentes

Componente

- Un componente de software puede ser desde un módulo de un programa hasta incluir bases de datos o sistemas de configuración entre clientes y servidores

Propiedades de los Componentes

- Se refieren a las características que hacen posible la interacción entre componentes
- No se especifican las propiedades internas de cada componente, por ejemplo, detalles de funcionamiento o algoritmos

Relaciones entre Componentes

- Las relaciones pueden ser tan sencillas como una invocación de un módulo a otro hasta el protocolo de conexión con base de datos

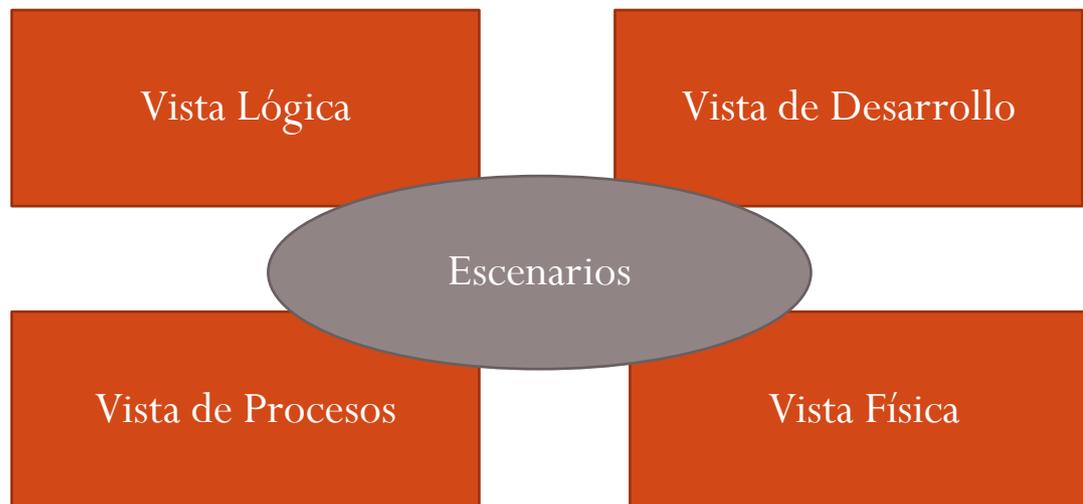
Vista de Arquitectura

Vista de Arquitectura

- Las vistas se utilizan para representar la arquitectura del software en un solo diagrama
- Una vista se define como la forma de presentar un modelo que describe un sistema de manera completa desde un cierto punto de vista

Arquitectura 4 + 1

- Describe la arquitectura de un sistema a partir de su representación en vistas



Vistas

- Vista Lógica.
- Representa los requisitos funcionales del sistema y lo que debe hacer
- Se representa a través de los diagramas:
 - Diagramas de Clases
 - Diagrama de Secuencia
- Utiliza mucho el esquema Orientado a Objetos

Vista de Desarrollo

- También llamada de Despliegue
- Se enfoca en la organización de los módulos que componen el software
- Se representa a través de los diagramas:
 - Diagrama de Componentes
 - Diagrama de Paquetes
 - Diagramas de Estado
- Es común utilizar un representación en capas

Vista de Procesos

- Representa el flujo del trabajo del sistema en tiempo de ejecución
- Se representa con los diagramas:
 - Diagramas de Actividad
- Se utilizan tuberías y filtros, llamadas y retornos
- Considera aspectos como la tolerancia a fallos

Vista Física

- Muestra como se reparten los componentes en los equipos que formarán el sistema completo
- Se representa a partir de los siguientes diagramas:
 - Diagrama de Despliegue
- Considera la disponibilidad del sistema, tolerancia a fallos y escalabilidad

Vista + 1

- También llamada Vista de Escenarios, está representada por el conjunto de los casos de uso
- Sirve como enlace entre las otras cuatro vistas
- Identifica los componentes que implementados representan una funcionalidad