

1. Defina arquitectura de software.

Estructura de las estructuras del sistema, que comprende componentes, propiedades de ellos y relaciones entre ellos.

2. Defina tres capacidades dadas por la arquitectura de software.

- 1) Analizar la efectividad del diseño para la consecución de requisitos fijados
- 2) Considerar las alternativas arquitectónicas en una etapa en la cual hacer cambios es fácil
- 3) Reducir riesgos asociados a construir software

3. Defina componente de una arquitectura de software y sus propiedades.

Un componente puede ser algo tan sencillo como un módulo, o algo tan complicado como incluir bases de datos y software intermedio. Las propiedades son las características necesarias para entender cómo interactúan los componentes.

4. De tres razones que fundamenten la importancia de la arquitectura de software.

- 1) Las representaciones de la arquitectura facilitan la comunicación entre todas las partes
- 2) La arquitectura destaca decisiones tempranas de diseño que tendrán un profundo impacto en todo el trabajo de ingeniería del software que sigue
- 3) Es un modelo pequeño y comprensible de cómo está estructurado el sistema y como trabajan sus componentes

5. Defina y diferencie diseño de datos y diseño arquitectónico

El diseño arquitectónico da una visión “Gestalt” del sistema, permitiendo al ingeniero de software examinarlo como un todo, y este modelo es transferible.

El diseño de datos crea un modelo de datos que se representa con un alto nivel de abstracción y se refina progresivamente

6. Defina modelado de datos, estructuras de datos, base de datos y almacén de datos.

Un almacén de datos es un entorno de datos separado que no es integrado en las aplicaciones, pero que contiene los datos utilizados por una empresa. En cierto sentido es una gran BD independiente que contiene algunos datos de las bases de datos del negocio

7. Enuncie características diferenciales entre un almacén de datos y una base de datos típica.

- **Orientación por materia:** Se organiza por materias y no por procesos o funciones del negocio.
- **Integración:** Es bueno definir unidades, medidas y convenciones más allá de que haya inconsistencia entre las bases de datos.
- **Restricciones en el tiempo:** Los datos son accedidos en un momento específico de tiempo (5 a 10 años).
- **No volatilidad:** Los datos se cargan pero nunca se cambian

8. Enuncie los principios de Wasserman para especificar y diseñar dicha estructura de datos

- 1) Los principios de análisis sistemático aplicados a función y comportamiento se deben aplicar también a los datos

- 2) Todas las estructuras de datos y operaciones a llevar a cabo deben estar claramente identificadas
- 3) Se debe establecer un diccionario de datos y usarlo para definir el diseño de datos del programa
- 4) Las decisiones de diseño de datos de bajo nivel deberían dejarse para el final del proceso de diseño
- 5) La representación de las estructuras de datos deben conocerla solo aquellos módulos que hacen uso de los datos de esa estructura
- 6) Debe desarrollarse una biblioteca de estructuras y operaciones que se pueden utilizar
- 7) El diseño del software y el lenguaje deben soportar la especificación y realización de TDA's

9. Enuncie los cuatro contenidos que tiene cada categoría de estilo arquitectónico.

- 1) Un conjunto de componentes que realizan una función requerida por el sistema
- 2) Un conjunto de conectores que posibilitan coordinación, cooperación y comunicación entre estos
- 3) Restricciones que definen como se pueden integrar los componentes del sistema
- 4) Modelos semánticos que permiten al diseñar entender propiedades del sistema

10. Defina las siguientes arquitecturas: [a] centradas de datos, [b] de flujo de datos, [c] de llamada y retorno diferenciando (c.1) programa principal y subprograma y (c.2) de llamada a procedimiento remoto, [d] orientadas a objetos y [e] estratificadas.

- **A Centradas en datos:** En el centro de la estructura hay un almacén de datos que s accedido con frecuencia por otros componentes para para modificar, borrar o actualizar informacion del almacén
- **B De flujo de datos:** Se usa cuando los datos de entrada son modificados por los componentes. Un patrón tubería y filtro, posee un grupo de componentes llamados filtros, conectados por tuberías que transmiten datos de un componentes al siguiente.
- **C De llamada y retorno diferenciado:** Permite construir una estructura de programa fácil de montar y modificar. Posee dos subestilos, arquitectura de programa principal y subprograma, donde las funciones sonllamadas por un programa <<principal>>, y arquitectura de procedimiento remoto, donde los componentes de la arquitectura principal/subprograma están en varias computadoras de un red.
- **D) Orientados a objetos:** Los componentes encapsulan datos y operaciones, y la comunicación es por paso de mensajes.
- **E) Estratificadas:** Se crean capas donde cada una realiza operaciones que progresivamente se aproximan a las instrucciones de la máquina. Capa externa, operaciones de interfaz de usuario, Capa intermedia, servicios de utilidad, Capa Interna, operaciones de interfaz de sistema

11. Enuncie preguntas sobre control y datos que proporcionan al diseñador una evaluación temprana de la calidad del diseño y sientan las bases para un análisis detallado de la arquitectura.

¿Cómo se gestiona el control dentro de la arquitectura?,

- ¿Existe una arquitectura diferente?,
- ¿Cómo transfieren el control los componentes dentro del sistema?

Preguntas de datos:

- ¿Cómo se comunican los datos entre componentes?

- ¿Existen componentes de datos, y de ser así, cuál es su papel?
- ¿Los componentes son activos o pasivos?

12. Enuncie y defina las actividades del Método de Análisis de Compromiso para la Arquitectura (MACA) desarrollado por el Instituto de Ingeniería de Software (SEI - Software Engineering Institute)

- 1) Recopilar escenarios
- 2) Elicitación de requisitos, para asegurar que todos los implicados fueron atendidos.
- 3) Describir patrones o estilos arquitectónicos elegidos para derivar escenarios y requisitos.
- 4) Evaluar atributos de calidad considerando cada atributo de forma aislada.
- 5) Identificar la sensibilidad de atributos de calidad con diferentes atributos arquitectónicos en un estilo específico.
- 6) Analizar arquitecturas del paso 3 utilizando análisis de sensibilidad del paso 5

13. Enuncie y defina tipos de dependencia entre componentes de arquitectura de software.

Dependencia de comportamiento: Representan relaciones de dependencia entre consumidores que utilizan los mismos recursos o productores que producen para los mismos consumidores.

Dependencias de flujo: Representar relaciones de dependencia entre consumidores y productores de recursos.

Dependencias Restrictivas: Representan restricciones de un relativo flujo de control entre un cuadro de actividades

14. Enuncie las tareas a completar después de haber desarrollado y refinado la arquitectura de software.

- Desarrollar descripción de procesamiento para cada módulo
- Aportar una descripción de la interfaz para cada módulo

- Definir estructuras de datos generales y locales
- Anotar limitaciones y restricciones de diseño
- Llevar a cabo una revisión del diseño
- Realizar un refinamiento si es necesario y esta justificado