**PREGUNTAS UNIDAD I**

**¿Porque es importante la Arquitectura DE Software durante el desarrollo del Sistema?**

1. Documento sobre el que se puede discutir
2. Distribución Física.
3. Permite detectar errores de diseño en fases tempranas.
4. Aspectos de Seguridad.

**¿Cuándo se diseña la Arquitectura en la Arquitectura de software?**

1. En el análisis del Problema.
2. El diseño arquitectónico es una actividad continua.
3. En el análisis previo.

**La Arquitectura de Software es a grandes rasgos, una vista del sistema que incluye lAs \_\_\_\_\_\_\_\_\_\_ principales del mismo.**

1. Aplicaciones
2. Estrategias
3. Normativas
4. Componentes

**¿La Arquitectura no es…?**

1. Una normativa madura.
2. Es la organización fundamental de un sistemas encarnada en sus componentes
3. Es la aplicación de una estrategia sistemática
4. Vista estructural de alto nivel.

**La consecuencia de la Evolución Acelerada del Hardware es:**

1. Reducción de costes para su producción
2. Avance de las comunidades entre sistemas
3. Que puede llevarse en el bolsillo
4. Incremento constante de la capacidad de operación.

**Las tecnologías viejas no mueren, solo se ocultan en \_\_\_\_\_\_\_ más profundas del software.**

1. Capas
2. Módulos
3. Servicios
4. Vistas.

**De acuerdo a la perspectiva de la arquitectura propuesta por Microsoft, cual Describe la estructura de hardware y software de base que da soporte informático a la organización.**

1. Negocio
2. Aplicación
3. Información
4. Tecnología.

**Es usada para definir los requerimientos funcionales y la visión que usuarios del negocio tienen de la aplicación y describir el modelo los de los la negocio que la arquitectura debe cubrir.**

1. Vista conceptual
2. Vista Lógica
3. Vista Física
4. Vista Organizacional.

**Muestra los componentes principales de diseño y sus relaciones de forma independiente de los detalles técnicos y de cómo la funcionalidad será implementada en la plataforma de ejecución.**

1. Vista conceptual
2. Vista Lógica
3. Vista Física
4. Vista Organizacional.

**Ilustra la distribución del procesamiento entre los distintos equipos que conforman la solución, incluyendo los servicios y procesos de base.**

1. Vista conceptual
2. Vista Lógica
3. Vista Física
4. Vista Organizacional.

**PREGUNTAS UNIDAD II**

**1.- Ventajas de diseñar y documentar explícitamente una arquitectura de software**:

* **Comunicación entre stakeholders, Decisiones tempranas de diseño, Reuso a gran escala**
* Solución de compromiso, Decisiones tempranas de diseño
* Requerimientos del sistema, Diseño detallado, sistemas complejos

**2.- Conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software**

* **Arquitectura de software**
* Arquitectura de diseño
* Arquitectura de patrones
* Arquitectura de modelado

**3.- Son piezas importantes de un patrón.**

1. Nombre
2. Contexto
3. Solución
4. Proposito
5. Diseño
6. 1,2,5
7. 2,5,4
8. **1,2,3**
9. 2,3,4

4**.- De la lista seleccione lo que considere que son estilos arquitectónicos.**

1. Banco de ayudas
2. **Tubos y Filtros**
3. **Control Centralizado**
4. **Control Basado en Eventos**
5. Descomposición de sistemas
6. Todos

**5.- Sistemas descentralizados donde las computaciones pueden ser realizadas en cualquier nodo de la red.**

1. Cliente-Servidor
2. **Peer-to-Peer**
3. Service Oriented Architecture
4. Sistemas Distribuidos

**6.- Provee un subconjunto de subsistemas predefinidos, incluyendo reglas y pautas para su organización.**

1. Patrón diseño
2. **Patrón arquitectónico**
3. Estilo Arquitectónico
4. Patrón de calidad

**7.- Permitir adaptar una interfaz de un tipo de datos para poder acceder a otro distinto, también permite Convertir un tipo de dato en otro mediante adaptación de sus interfaces.**

1. Patrón Fábrica
2. Patrón Fachada
3. Patrón Arquitectónico
4. **Patrón Adaptador**

**8.- El patrón plantilla es el que.**

1. **Define el esqueleto de un algoritmo en un método, refiriendo algunos pasos a subclases.**
2. Permite que la selección del algoritmo se haga según el objeto que se trate.
3. Se eliminan sentencias condicionales para seleccionar el algoritmo deseado
4. Definir una dependencia 1:n de forma que cuando el objeto 1 cambie su estado, los n objetos sean notificados y se actualicen automáticamente

**9.- Es la que se encarga de asegurar que una clase tiene una sola instancia y proporcionar un punto de acceso global a ella.**

1. Patrón Acción
2. Patrón Fachada
3. Patrón Adaptador
4. **Patrón Singlenton**

**10.- Seleccione 2 escenarios en los cuales se aplicará el patrón fachada:**

1. **Se necesite proporcionar una interfaz simple para un subsistema complejo**
2. Centralización de la creación de objetos
3. **Cuando se quiera estructurar varios subsistemas en capas**
4. Se quiere envolver código no orientado a objeto con forma de clase

**PREGUNTAS UNIDAD III**

**1.- Indique cuál de las siguientes definiciones de calidad no es correcta:**

Grado en el que un conjunto de características cumple con los requisitos.

Conjunto de actividades que nos permiten satisfacer los requisitos de un colectivo.

Satisfacción del cliente y conformidad con sus requisitos.

Grado de satisfacción que produce el cliente.

**2.- indique los pasos de la Metodología.**

1. Clasificación
2. Evaluación
3. Definición
4. Selección
   1. 3,2,1,4
   2. 1,2,3,4
   3. 4,3,1,2
   4. 3,1,4,2

**3.- Indique las distintas formas en las que están basadas las técnicas de evaluación.**

* Basada en Escenarios
* Basada en Simulación
* Basada en Modelos Matemáticos
* Basada en Experiencia

**4.-Indique cual es el objetivo de la clasificación.**

1. Colección de información por parte de las comunidades de código abierto.
2. Definir varios elementos de la tipología a ser utilizada por los 3 pasos que siguen.
3. Identificar el software que contenga y satisfaga los requerimientos de usuario, o de manera más general permita la comparación de software de una misma familia.
4. Definir los filtros que traduzcan las necesidades y restricciones relacionadas con la selección del software de código abierto en un contexto específico.

**5.- Seleccione las ventajas de la Evaluación Basada en Escenarios.**

1. Es más eficiente.
2. Son simples de crear y entender.
3. Son efectivos.
4. Son más costosos y requieren mucho entrenamiento.

**6.- Seleccione la definición de Evaluación Basada en Escenarios: Profiles.**

* Es un esquema en forma de árbol que presenta los atributos de calidad de un sistema de software, refinados hasta el establecimiento de escenarios que especifican con suficiente detalle el nivel de prioridad de cada uno.
* Es un conjunto de escenarios, generalmente con alguna importancia relativa asociada a cada uno de ellos.
* Los atributos de calidad asociados al mismo conforman el segundo nivel del árbol, los cuales se refinan hasta la obtención de un escenario lo suficientemente concreto para ser analizado y otorgarle prioridad a cada atributo considerado.
* Su uso permite hacer especificaciones más precisas del requerimiento para un atributo de calidad.

**7.- indique porque evaluar una arquitectura.**

* Arreglar un error antes y después de las fases de requerimientos o diseño.
* El producto no tiene un efecto en el sistema y en el proyecto.
* Estilos y patrones impactan negativamente los atributos de calidad del sistema.
* Manera más económica de evitar desastres.

**8.- ¿Por qué cualidades puede ser evaluada una Arquitectura?**

* Performance
* Fiabilidad
* Disponibilidad
* Seguridad

**9.- Seleccione quien o quienes están involucrados en la evaluación del software.**

* Equipo de Evaluación
* Director del proyecto
* Stakeholders
* Ninguno de los anteriores

**10.- indique cual no es un beneficio de realizar una evaluación arquitectónica.**

Resultan mejoras en las arquitecturas

Mejora la calidad de la documentación de la arquitectura

Descubre oportunidades de reúso

Ninguna de las anteriores