**1. Establezca de manera sucinta la diferencia entre arquitectura empresarial y arquitectura de software.**

La arquitectura de software es una capa de la arquitectura empresarial, presenta la estructura de los componentes de un sistema/programa, sus interrelaciones, principios y guías gobernando su diseño y evolución en el tiempo.

La arquitectura empresarial se encarga de alinear procesos, datos, aplicaciones e infraestructura para lograr un objetivo común desde un punto de vista integral de la organización.

**2. Diferencie conceptualmente arquitectura de referencia descriptiva de arquitectura de referencia prescriptiva.**

* **Arquitectura prescriptiva:**  Se diseña sin tener en cuenta las tecnologías a implementar, sólo con la concepción del sistema.
* **Arquitectura descriptiva:** Se encuentra ya en desarrollo (producción) y tiene tecnologías que implementan la prescriptiva.

Si las dos arquitecturas no están acordes entre sí, se genera degradación arquitectural.

**3. Enuncie diez estilos arquitectónicos de software y explique de manera sucinta uno de ellos.**

* Centrados en los datos
* Repositorio Pasivo: Sólo consulta
* Repositorio Activo: Aparte de consultar, el estado incide en el comportamiento de los componentes (Pizarra)
* Flujo de datos
* Secuencial en lote: Se alimenta en forma secuencial
* Continuo “pipe & filter” Flujos continuos de datos entre los filtros.
* Máquina virtual
* Llamada y retorno
* Programa principal y subrutina
* Llamada a procedimientos remotos
* Orientado a objetos
* Basado en capas
* Componentes independientes
* Sistema basado en eventos
* Comunicación de procesos

**4. Enuncie tres subestilos arquitectónicos del estilo “Publicar / Suscribir”.**

* Basado en eventos
* Repositorio activo
* Repositorio pasivo

5. ASDL (Architectural Style Description Language, en el inglés) es una propuesta de Lenguaje de Descripción Arquitectural elaborada por **Rice & saidman** y se basa para la especificación estructural en **lenguaje z** y para la especificación comportamental en **algebra de procesos (csp)**.

**6. Diferencie orquestación de servicios de coreografía de servicios.**

* **Orquestación:** Hay un coordinador central que gestiona los servicios decidiendo cuando y con quien interactuar.
* **Coreografía:** Cada servicio sabe cuándo y con quién intectuar sin necesidad de algún coordinador.

**7. Explique de manera sucinta la utilidad de BPEL(Business Process Execution Language) en el logro de arquitecturas conformes a SOA(Service-Oriented Architecture).**

BPEL constituye un lenguaje estándar para la integración y automatización de procesos. Los procesos de negocio programados con BPEL serán capaces de ejecutarse en diferentes plataformas que cumplan dicho estándar, ofreciendo a los clientes una mayor libertad de elección.

**8. ¿Qué abstracción favorece el modelado modular usando lenguaje Z?. Explique sucintamente y de manera precisa.**

La abstracción de estilos de arquitectura de software porque permite ignorar los problemas de implementación de alto nivel.

**9. Enuncie los patrones de diseño creacionales. ¿Cuál es su objetivo?. Se le presentará una situación problémica de software donde pueda aplicar uno de ellos. Usted deberá identificar cuál patrón de diseño podría ser el más adecuado y sustentar su respuesta.**

* Método fabrica
* Ejemplificación única
* Factoría abstracta
* Prototype
* Builder
* Lidia con la creación de objetos.
* Soporta un uniforme, simple y controlado mecanismo de creación de objetos.
* Permite la encapsulación de los detalles sobre qué clases son ejemplificadas y cómo estas ejemplificaciones fueron creadas.

**10. Enuncie los patrones de diseño comportamentales. ¿Cuál es su objetivo?. Se le presentará una situación problémica de software donde pueda aplicar uno de ellos. Usted deberá identificar cuál patrón de diseño podría ser el más adecuado y sustentar su respuesta.**

* Comando
* Mediador
* Memento
* Observador
* Interpetrador
* Método plantilla
* Estado
* Estrategia
* Lidian con los detalles de asignar responsabilidades entre diferentes objetos
* Describen los mecanismos de comunicación entre diferentes objetos.
* Describen los mecanismos para escoger diferentes algoritmos entre objetos en tiempo de ejecución.

**11. Enuncie los patrones de diseño estructurales. ¿Cuál es su objetivo?. Se le presentará una situación problémica de software donde pueda aplicar uno de ellos. Usted deberá identificar cuál patrón de diseño podría ser el más adecuado y sustentar su respuesta.**

* Decorador
* Bridge
* Adaptador
* Cadena de responsabilidad
* Fachada
* Proxy
* Lidia con objetos delegando responsabilidades a otros objetos
* Facilita la comunicación interobjeto cuando un objeto no es accesible por otro de formas usuales o cuando sus interfaces no son compatibles.

**12. Enuncie los patrones de diseño sobre colecciones. ¿Cuál es su objetivo?. Se le presentará una situación problémica de software donde pueda aplicar uno de ellos. Usted deberá identificar cuál patrón de diseño podría ser el más adecuado y sustentar su respuesta.**

* Composite
* Iterador
* Flyweight
* Visitador
* Lidia con detalles de cómo componer clases y objetos para formar estructuras robustas.
* Permite la definición de operaciones sobre colecciones de objetos.

**13. Explique el concepto de Templates a la luz de ASDL (Architectural Style Description Language, en el inglés).**

**Plantillas:** Representa interfaces de componentes que están disponibles para su inclusión en una arquitectura de software.

**14. Explique el concepto de Settings a la luz de ASDL (Architectural Style Description Language, en el inglés).**

**Configuración/Marco:** Arquitecturas que han sido construidas por plantillas ya instanciadas.

**15. Explique el concepto de Units a la luz de ASDL (Architectural Style Description Language, en el inglés).**

**Unidades:** Representa la jerarquía del sistema. Puede encapsular marcos para que sus interfaces puedan usarse a la vez como plantillas o representen interfaces diseñadas de arriba abajo.

**16. ¿Para qué sirve CSP(Communicating Sequential Processes) en ASDL?**

CSP (Communicating sequential processes) sirve para describir interacciones, patrones de interacción de sistemas concurrentes, parte del algebra de procesos.

**17. ¿A qué se denomina jerarquización y refinamiento en el área de conocimiento de arquitecturas de software?.**

**Jerarquización:** Cómo está conformado por dentro un componente en una arquitectura.

**Refinamiento:**  Mapeo desde una especificación formal a una tecnología.

**18. Diferencie los enfoques de especificación axiomática y algebraica en el área de conocimiento de arquitecturas de software.**

**Algebra:** Los valores abstractos de los universos de datos y las operaciones que se pretende representar en un programa.

**Axioma:** Una de las formas más extendida para caracterizar o especificar algebras, se basa en el uso de ecuaciones para expresar los comportamientos de valores y funciones.

**19. ¿Qué es la “Chemical Abstract Machine”?. ¿Qué utilidad tiene el operador “Airlock”?.**

**Chemical Abstract Machine** es un marco de trabajo para desarrollar especificaciones operacionales que no sesgan el sistema descrito a un modelo de computación particular, está compuesta por moléculas y soluciones que son multiconjuntos de moléculas.

El operador **Airlock** permite extraer una molécula de una solución o reabsorber una molécula a una solución sin perder su identidad.

**20. Enuncie los patrones arquitectónicos más conocidos. Explique de manera sucinta uno de ellos.**

* Patrón de capas (Presetanción-lógica de negocio-estado): Se puede utilizar para estructurar programas que se pueden descomponer en grupos de subtareas, cada una de las cuales se encuentra en un nivel particular de abstracción. Cada capa proporciona servicios a la siguiente capa.
* Patrón modelo-vista-controlador
* Patrón sensar-computar- actuar.

**21. Basándose en los modelos funcionales, estructurales y dinámicos explique el ejercicio de recuperación de diseño del ejemplo asociado al patrón de diseño “MEMENTO”.**

//Se explica a fondo en la sustentación.

Convierte los datos a inserciones SQL, donde sí se daña un dato de la conversión, se detiene el proceso, guarda el estado y en la posterior ejecución comienza donde quedó.

**22. Basándose en los modelos funcionales, estructurales y dinámicos explique el ejercicio de recuperación de diseño del ejemplo asociado al patrón de diseño OBSERVADOR.**

**//**Se explica en la sustentación.

Actualiza los reportes según la selección del usuario del departamento requerido, siendo *ReportManager* el observado, y *YDTChart* junto *MonthlyReport* los observadores. Es un observador de tipo push, donde al existir un cambio se notifica a los observadores.

**23. Basándose en los modelos funcionales, estructurales y dinámicos explique el ejercicio de recuperación de diseño del ejemplo asociado al patrón de diseño INTERPRETADOR.**

**//** Se explica en la sustentación.

Evalua una expresión algebraica mediante 3 pasos, construir infijo a postfijo, construir árbol sintático y recorrido del árbol junto evaluación según contexto.

**24. Basándose en los modelos funcionales, estructurales y dinámicos explique el ejercicio de recuperación de diseño del ejemplo asociado al patrón de diseño ESTADO.**

**//**Se explica en la sustentación.

Se evalua el estado de una tarjeta de crédito y se realiza la operación según su estado.

**25. Basándose en los modelos funcionales, estructurales y dinámicos explique el ejercicio de recuperación de diseño del ejemplo asociado al patrón de diseño MÉTODO PLANTILLA.**

**//**Se explica en la sustentación.

Se valida una tarjeta de crédito, teniendo una plantilla con los pasos a seguir.

**26. Basándose en los modelos funcionales, estructurales y dinámicos explique el ejercicio de recuperación de diseño del ejemplo asociado al patrón de diseño “ESTRATEGIA”.**

//Se explica en la sustentación.

Se cifra un mensaje según el método que se quiera, generando por cáda método un archivo txt donde se guarda el mensaje.

**27. Responda de manera sucinta: ¿Qué se puede entender por un modelo independiente de la computación? Presente un ejemplo.**

Es una descripción lógica del negocio con una visión totalmente ajena a la computación. Es un modelo de dominio que solo recalca las funciones o lo que se espera del sistema. Como lo son los diagramas de casos de uso.

**28. Responda de manera sucinta: ¿Qué se puede entender por un modelo independiente de la plataforma? Presente un ejemplo.**

Es una descripción de la funcionalidad del sistema en forma independiente de las características de plataformas específicas, pues crea los procesos y las estructuras del sistema, sin pensar en el SO, lenguaje o el hardware. Como lo son los diagramas de clases.

**29. Responda de manera sucinta: ¿Qué se puede entender por un modelo específico de la plataforma? Presente un ejemplo.**

Descripción específica del sistema a nivel de código a implementar tomando en cuenta el SO, lenguajes de programación y plataformas. Como lo es un programa ya implementando en JAVA.