

23. En los sistemas enterprise es típico el modelar roles y permisos para restringir las acciones que un usuario puede hacer, y así permitir a una organización el reflejar su estructura organizacional en las diferentes responsabilidades del sistema. ¿Qué táctica de seguridad se relaciona a esto?

Autorización



24. ¿Cómo podemos saber si las decisiones de diseño que tomamos fueron las correctas?

A través de métricas y umbrales, midiendo los resultados de la arquitectura implementada



25. "Documentar la arquitectura no es necesario si el arquitecto va a formar parte del equipo de desarrollo." Esta afirmación es:

Falsa. El equipo de desarrollo no es el único interesado en la arquitectura y es necesario comunicar a todas las partes interesadas



17. Luego del estreno de la última película de La Guerra de las Galaxias, el sitio oficial de la saga recibió cien veces más pedidos de compras de sus productos de los que tiene en promedio. Esto generó que el sistema de cálculo de existencias e inventario se saturara, ya que no estaba preparado para el volumen de datos que recibió. ¿Qué conector se puede implementar para evitar que esto afecte al componente de ventas?

Cola



18. El uso de Promesas o Futuros permite responder un pedido, cuando la respuesta no está todavía calculada por completo. El componente cliente decide cuándo resolver esta promesa o qué ejecutar cuando esta esté disponible. Si un componente que responde con un dato pasa a responder con una promesa, ¿qué cambio de conector hizo?

Llamado sincrónico a llamado asincrónico



19. En las tácticas de redundancia, el componente se replica y estas réplicas se sincronizan para poder así responder con cualquiera de ellas a los pedidos de sus clientes. ¿Qué atributo de calidad está favoreciendo esta táctica?

Disponibilidad



22. "Para mejorar la usabilidad tenemos que tener un _____ que nos ayude a entender su contexto específico, sus necesidades y los objetivos que intenta cumplir a través del uso de la aplicación."

Modelo del usuario



6. El lenguaje de programación Java es usado para tipos de aplicación muy variados, ya que el mismo artefacto, una vez compilado a bytecode, se puede desplegar y ejecutar en diferentes contextos. Sin embargo, las aplicaciones más exigentes como por ejemplo los videojuegos de última generación, deciden usar lenguajes como C++ para tener mejor control de la memoria y las instrucciones de video. ¿Cuál es la tensión entre atributos que contribuye a esta decisión?

Uso de recursos y adaptabilidad



8. El principio de inversión de dependencias dice: "Los módulos de alto nivel no deben depender de módulos de bajo nivel: Ambos deben depender de abstracciones. Las abstracciones no deben depender de los detalles: Los detalles deben depender de abstracciones." ¿Qué atributo de calidad favorece este principio?

Modularidad



9. Los patrones de arquitectura monolíticos se diferencian de los patrones de arquitectura distribuidos en que:

...el despliegue del monolítico es en un solo artefacto, mientras que los distribuidos se despliegan independientes.



10. Cuando un sistema se desarrolla sin prestar atención a su arquitectura y cada componente puede comunicarse con cualquier otro componente, decimos que tenemos el patrón:

Gran bola de lodo



11. El patrón MVC y los relacionados (MVVM, MVP o Flux) separan una aplicación monolítica en 3 familias de componentes importantes, y luego los interconectan de formas diferentes. Estas familias se pueden definir como:

El modelado del dominio, las acciones que el usuario puede hacer y la presentación de la información



12. El patrón de arquitectura de capas describe una estructura tal que:

Cada capa se puede probar independientemente de sus capas superiores



13. Si usamos el patrón orientado a eventos, ¿cuál de estas afirmaciones es correcta?

Los componentes se comunicarán entre si indirectamente, a través de publicación y suscripción de eventos



14. Las organizaciones que migran sus productos a microservicios buscan favorecer:

Mantenibilidad y disponibilidad



15. "Decidimos diseñar la aplicación con el patrón de arquitectura "_____", que nos permitió separar claramente nuestro modelo de dominio, los puertos que definen las operaciones que se pueden efectuar en este dominio y los adaptadores que definen las tecnologías específicas del contexto actual de desarrollo como qué base de datos necesitamos o el formato de los mensajes de la API." ¿Qué patrón de arquitectura usaron?

Hexagonal



16. En una arquitectura en capas, un componente de la capa de negocio está conectado con un componente de la capa de datos de forma tal que, cuando lo necesita, delega la ejecución al componente de la capa inferior hasta que éste termina de obtener la información solicitada y luego la devuelve, junto con la ejecución. ¿Qué tipo de conector los une?

Llamador sincrónico



30. En el modelo de vistas de arquitectura "4+1", los escenarios nos sirven para:

Conectar las cuatro vistas a través de ejemplos y casos de uso

