

GRAT1

Las responsabilidades de un arquitecto de software son:

Seleccione una o más de una:

- ☒ a. conciencia operativa ✓
- ☒ b. habilidades técnicas ✓
- ☒ c. habilidades blandas ✓

¿Cuál es la palabra clave que se repite en las conceptualizaciones de arquitectura de software?

- ☐ a. estructura
- ☐ b. componentes
- ☒ c. decisiones

¿En qué se parece la arquitectura de software al arte?

- ☐ a. requiere mucha creatividad
- ☐ b. la comprende solo el que la creó
- ☒ c. solo puede ser comprendida en su contexto

¿Qué integra la arquitectura de software?

Seleccione una o más de una:

- ☒ a.
Decisiones ✓
- ☒ b.
Principios de diseño ✓
- ☒ c.
Estructura ✓
- ☒ d.
Características ✓

Asocie los componentes de una arquitectura con lo que integra.

Requerimientos no funcionales (disponibilidad, confiabilidad, etc.)	Características	✓
Reglas de cómo se debe construir el sistema.	Decisiones	✓
Estilos arquitectónicos (capas, microservicios)	Estructura	✓
Guía de cómo construir el sistema, más que una regla fija.	Principios de diseño	✓

La varianza explica que es posible que una parte del sistema no siga las reglas establecidas.

R//VERDADERO

La varianza la pueden aplicar en el equipo de desarrollo con plena autonomía.

R//FALSO

Asocie las funciones esperadas del arquitecto de software con su descripción

Estar expuesto a múltiples tecnologías, marcos de referencia, plataformas y ambientes. No significa ser experto, sino estar familiarizado.	Experiencia y exposición a la diversidad
Comprobar que el equipo de desarrollo esté siguiendo las decisiones y principios definidos.	Asegurar el cumplimiento
Conocer las tendencias y tecnologías relevantes para el contexto actual.	Mantenerse actualizado
Trabajo en equipo, facilitación, liderazgo.	Habilidades interpersonales
Estrategias de negociación y políticas de la empresas.	Comprender y navegar en la política
Evaluar la viabilidad y vigencia de las arquitecturas existentes	Analizar constantemente
Comprender el problema de negocio que se está resolviendo y los requerimientos funcionales.	Conocimiento del negocio
Establecer guías para elegir las tecnologías correctas.	Toma de decisiones

Hay empresas que subcontratan el diseño de arquitecturas para atributos de calidad específicos.

R//VERDADERO

El arquitecto de software es un líder técnico

R//VERDADERO

La arquitectura de software debe evolucionar

R//VERDADERO

El código y los datos tienen una relación simbiótica.

R//VERDADERO

GRAT2

¿Cuáles son las funciones de las características arquitectónicas?

R//TODAS LAS OPCIONES

¿Qué sinónimos tienen las características arquitectónicas?

Aclaración: independientemente si el autor está de acuerdo con ellos o no.

R//Requerimientos no funcionales y atributos de calidad

Una arquitectura de software debe considerar la mayor cantidad de características posibles.

R//falso

Clasifique las características por nivel.

escalabilidad	alto nivel	✓
modularidad	bajo nivel	✓
elasticidad	alto nivel	✓

Clasifique las características por tipo.

extensibilidad, mantenimiento, portabilidad	estructura	✓
disponibilidad, continuidad, rendimiento	operativas	✓
accesibilidad, usabilidad, seguridad	transversales	✓

¿Qué estándares proponen definiciones para las características arquitectónicas?

R// ISO

Existe un estándar de definición de las características claramente definido para la industria del software que es utilizado por todas (o la mayoría) de instituciones.

R//FALSO

Cada característica implica varias decisiones de diseño.

r//VERDADERO

Una buena práctica es diseñar una arquitectura genérica que aplique a todos los casos de negocio.

R//falso

Es responsabilidad del equipo de arquitectos el extraer las características de los requerimientos de negocio.

R//verdadero

Asocie la traducción a "lenguaje de negocios" de las características

tiempo de liberación al mercado	agilidad + pruebas + distribución	✓
interoperabilidad, escalabilidad, adaptabilidad	fusiones y adquisiciones	✓
simplicidad y factibilidad	tiempo y presupuesto	✓
rendimiento, disponibilidad, tolerancia a fallas, seguridad	satisfacción del usuario	✓

GRAT3

Se debe analizar constantemente el ambiente para comprobar si la arquitectura sigue vigente.

R//verdadero

Las arquitecturas se deterioran porque:

R// se maneja mal la escalabilidad y falta de análisis constante en el ambiente

¿Cuáles son los aspectos a considerar de una arquitectura?

R//todos

¿A qué nivel se puede identificar si la arquitectura tiene un deterioro estructural?

r// Analizando los componentes de la arquitectura y el código fuente

Los bloques que constituyen a una aplicación se llaman (todo en minúsculas y plural):

r// componentes

Un componente está compuesto por (una palabra en minúsculas y plural):

r// clases

Como arquitecto, se debe analizar:

r//interacción entre componentes

asocie los tipos de acoplamiento a su definición:

el grado que otros componentes dependen de un componente particular

aferente ▾ ✓

el grado en que un componente particular depende de otros componentes

eferente ▾ ✓

componentes acoplados con dependencias no estáticas

temporal ▾ ✓

Asocie el nivel de acoplamiento con su definición

el grado en que los componentes están obligados a compartir un contexto de datos

de datos ▾ ✓

un componente confía su funcionalidad en otro componente

patológico ▾ ✓

un componente le pasa información a otro componente de qué hacer

control ▾ ✓

cuando muchos componentes comparten un protocolo o formato de datos externo e impuesto

externo ▾ ✓

El acoplamiento permite analizar la "salud" de las características arquitectónicas.

R//verdadero

GRAT4

Un patrón permite organizar:

r/todos

Un sistema con código spaghetti, con signos de crecimiento irregular, sin estructura interna y llamadas directas a las bases de datos, corresponde al patrón:

r// Big ball of mud

Cuando el software se ejecuta en una sola computadora, parecido a los mainframes y computadoras personales. Resultan difíciles de extender a características arquitectónicas como rendimiento y escalamiento, corresponde al patrón:

r// unitary

Cuando un sistema de software está separado en partes o capas que pueden operar en distintos equipos, corresponde al patrón.

r//cliente servidor

¿Cuáles son variantes del patrón cliente-servidor?

r// todos

Por la lectura se puede inferir que una arquitectura monolítica corresponde al patrón Unitary

r//verdadero

Es la arquitectura donde todo el código se despliega en una sola unidad/componente (1 palabra, singular en español, todo minúsculas)

r// monolítica

Es la arquitectura donde existen unidades múltiples de código conectadas por protocolos de acceso remoto (1 palabra, minúsculas, singular en español)

r// distribuida

¿Cuál es la consideración más importante en los estilos arquitectónicos distribuidos?

r//garantizar transacciones

GRAT5

La arquitectura por capas se puede considerar un estándar de facto porque familiar para la mayoría de arquitectos de software.

r//verdadero

¿Qué anti-patrones pueden estar asociados a la arquitectura por capas?

a.

por implicación

☒ b.

sumidero (sinkhole)

☒ c.

accidental

Ubique las capas tradicionales de este estilo arquitectónico

Persistencia	3	↕	✓
Presentación	1	↕	✓
Base de datos	4	↕	✓
Negocios	2	↕	✓

Todas las capas siempre deben implementarse en la misma unidad de distribución.

Rr/falso

Una unidad de distribución se refiere a:

Rr//hardware

En un modelo cerrado de capas se establece una jerarquía vertical que debe pasar por todas las capas de manera obligatoria.

r//verdadero

Cuando alguna de las capas es abierta, puede ser omitida en la secuencia de llamadas para completar una transacción.

r//verdadero

Las leyes de aislamiento de capas significa que las capas son:

r// independiente unas de otras

las mayores ventajas de la arquitectura por capas son:

r//todas

Las mayores desventajas de la arquitectura por capas son:

☒ b.
falta de agilidad para mantenimiento, pruebas y distribución

☒ c.
se hacen complejas cuando se requiere realizar cambios rápidamente y son de "dominio"

GRAT6

La arquitectura basada en servicios es parte de las arquitecturas distribuidas

R//VERDADERO

La relación entre los servicios y el dominio de negocio regularmente es:

R//UNO A UNO

En la arquitectura basada en servicios la base de datos generalmente es:

R//CENTRALIZADA

Las variantes de la topología basada en servicios, sugieren partición en las capas:

Seleccione una o más de una:

☐

a.
dominio de negocio

☒

b. aplicación ✓

☒

c.
interfaz de usuario ✓

Las particiones en la base de datos deben ser granulares sin perder la cohesión de dominio

R//VERDADERO

La estructura de la arquitectura basada en servicios la determina:

R//EL DOMINIO DE NEGOCIO

La arquitectura basada en servicios es de las más pragmáticas (prácticas) disponibles.

R//VERDADERO

GRAT7

Todas las arquitecturas distribuidas deben tener siempre las mismas capas

R//FALSO

La propósito mayor de SOA es:

R//REUTILIZACION

Asocie las capas más comunes de SOA con su función:

no está presente en este tipo de arquitectura

resuelve un problema de negocio específico, que no será compartido por todas las capas/equipo

contiene la implementaciones compartidas y "granuladas" de un dominio de negocio particular

proporciona las funciones operativas como monitoreo, bitácoras, autenticación

es el punto de entrada, responde al dominio, contiene definiciones

es el corazón del estilo de arquitectura, coordina las transacciones y la mensajería



Cuando la división de servicios se realiza por criterios técnicos y no de dominio, se suele aumentar:

R//ACOPAMIENTO

La capa que orquesta los servicios tiene como desventaja:

R//es un gran punto de acoplamiento

El gran valor de SOA para los arquitectos de software es:

R//enseñar la dificultad de distribuir las transacciones

GRAT8

Los microservicios se basan en el diseño conducido por el dominio

R//VERDADERO

Es una unidad de software que puede ser reemplazada o actualizada independientemente

R//COMPONENTE

La arquitectura de microservicios favorece el bajo o nulo acoplamiento, implicando aumento en redundancia

R//VERDADERO

Cada microservicio opera independientemente incluyendo bases de datos y otros componentes

R//VERDADERO

Qué característica es la más importante al diseñar en nivel de granularidad de los microservicios

R//RENDIMIENTO

Microservicio implica que los servicios deben ser lo más pequeños posible

R//FALSO

Los microservicios se inspiran en el principio de diseño clásico:

R//BAJO ACOPLAMIENTO Y ALTA COHESION

Es necesario que exista un estándar de persistencia de datos en todos los microservicios de una arquitectura

R//FALSO

Agregar orquestación va en contra de lo que predica la arquitectura de microservicios

R//VERDADERO

Es posible reutilizar componentes en la arquitectura de microservicios. como una variante topológica

R//VERDADERO

La principal función de la capa de API es

R//DESCUBRIMIENTO DE SERVICIOS

¿Cómo se implementa una transacción en la arquitectura de microservicios?

R//AJUSTANDO LA GRANURALIDAD