

Ingenieria en Software

Resumen redactado del material de Arquitectura y Sistemas de Software, con una muestra de 10 reactivos contestados y explicados del examen simulado.

Resumen del material

Panorama general

El material de Arquitectura y Sistemas de Software explica cómo organizar una solución informática para que sea comprensible, mantenible, escalable y verificable. Presenta temas clave como pruebas unitarias e integración, arquitectura monolítica, microservicios, cliente-servidor, MVC, modelo por capas, arquitectura de aplicaciones web, serverless y REST. La idea central es que la arquitectura no solo define cómo se ve un sistema por dentro, sino también cómo se prueba, despliega, mantiene y adapta a nuevas necesidades.

Pruebas unitarias e integración

Las pruebas unitarias verifican partes pequeñas del código, como funciones, métodos o clases, de forma aislada. Sirven para detectar errores temprano, mejorar la calidad del código y facilitar cambios futuros. Las pruebas de integración, por otro lado, revisan cómo interactúan varios módulos o servicios entre sí. Mientras las unitarias confirman que una pieza funciona correctamente, las de integración confirman que las piezas se comunican y trabajan de forma adecuada dentro del sistema completo.

Arquitectura monolítica

La arquitectura monolítica concentra toda la aplicación en un solo código base y normalmente se despliega como una unidad. Es útil en proyectos pequeños, equipos reducidos o escenarios donde se necesita avanzar rápido con una solución inicial. Su principal ventaja es la simplicidad de desarrollo y despliegue, pero conforme el sistema crece puede volverse difícil de escalar, modificar o actualizar sin afectar otras partes de la aplicación.

Microservicios

Los microservicios dividen una aplicación en servicios pequeños, independientes y especializados. Cada servicio atiende una función específica, por ejemplo usuarios, pagos, catálogo o notificaciones. Este enfoque permite escalar y desplegar componentes por separado, usar tecnologías distintas y mejorar la resiliencia del sistema. Sin embargo, también aumenta la complejidad técnica porque exige una buena comunicación entre servicios, control de datos distribuidos, monitoreo y administración de infraestructura.

Cliente-servidor, MVC y capas

La arquitectura cliente-servidor separa al cliente que solicita información del servidor que procesa y responde. Es común en aplicaciones web, correo electrónico y servicios en red. El patrón MVC divide la solución en Modelo, Vista y Controlador para separar datos, interfaz y lógica de control. El

modelo por capas organiza el sistema en niveles, como presentación, lógica de negocio y acceso a datos. Estos enfoques ayudan a ordenar responsabilidades, reducir acoplamiento y facilitar el mantenimiento.

Aplicaciones web, serverless y REST

En las aplicaciones web, el front-end se encarga de la interfaz y experiencia del usuario, mientras que el back-end procesa reglas de negocio, datos y seguridad. Serverless permite ejecutar funciones en la nube sin administrar directamente servidores, pagando según uso y facilitando escenarios por eventos. REST propone una forma estándar de diseñar servicios web usando recursos, URIs y operaciones HTTP, lo que permite construir APIs escalables y fáciles de consumir por otras aplicaciones.

Conclusión del material

El archivo muestra que elegir una arquitectura depende del tamaño del proyecto, el nivel de cambio esperado, la capacidad del equipo, los requisitos de escalabilidad y la necesidad de mantenimiento. No existe una arquitectura única para todos los casos: una solución pequeña puede beneficiarse de un monolito, mientras una plataforma grande o con múltiples equipos puede requerir capas, MVC, microservicios, REST o serverless. Para el EGEL Plus, lo importante es identificar qué necesidad plantea el caso y relacionarla con el patrón, principio o técnica adecuada.

Muestra de 10 reactivos contestados y explicados

Los siguientes reactivos retoman la lógica de los primeros ejercicios del examen simulado. Cada uno incluye respuesta correcta y una explicación breve para reforzar el criterio de solución.

Reactivo 1

Una dependencia pública contrata a una empresa para la implantación de un GRP comercial. El dueño del proceso fija los tipos de usuarios que se van a tener y sus niveles de acceso. ¿Cuál es un requerimiento de usuario?

A) Póliza de soporte B) Capacitación a usuarios C) Tipos de usuarios

Respuesta: C) Tipos de usuarios.

Explicación: Es un requerimiento de usuario porque describe quiénes usarán el sistema y con qué nivel de acceso. La póliza de soporte es una condición contractual y la capacitación es una actividad de implantación, no una característica directa del uso del sistema.

Reactivo 2

Una comercializadora internacional desarrolla una plataforma de ventas en línea. El corporativo desea operación 24/7 e información del estado de ventas por región. ¿Cuál corresponde a un requisito de negocio?

A) Identificar hosting en nube B) Reportar en tiempo real ventas por región C) Cumplir ANSI TIA 942 tier IV

Respuesta: B) Reportar en tiempo real el nivel de operaciones de venta con desglose regional.

Explicación: El requisito de negocio expresa la necesidad estratégica de la organización: conocer el estado de las ventas por región. Las opciones A y C son decisiones técnicas o restricciones de infraestructura.

Reactivo 3

Una tienda desea actualizar su punto de venta. Se busca asegurar que satisfaga requerimientos de dominio para usuarios finales. ¿Cuál es un requerimiento de usuario final?

A) Permitir al cajero imprimir recibo B) Registrar una venta en menos de 3 segundos C) Dar de alta cajeros

Respuesta: A) Permitir al cajero imprimir el recibo de pago.

Explicación: El cajero es usuario final y necesita imprimir recibos como parte de su operación. La velocidad de venta es un requisito de rendimiento y el alta de cajeros corresponde más a administración del sistema.

Reactivo 4

En un sistema web, el dueño de la empresa pide obtener cualquier consulta en máximo tres clics. ¿Cuál atributo de calidad externa corresponde?

A) Fiabilidad B) Eficiencia C) Usabilidad

Respuesta: C) Usabilidad.

Explicación: La necesidad se relaciona con facilidad de uso, navegación y acceso rápido a la información. No se refiere a fallas del sistema ni al consumo técnico de recursos.

Reactivo 5

Una oficina recaudadora necesita que el sistema se actualice constantemente por cambios fiscales. Se implementa diseño orientado a objetos con nuevas clases e interfaces. ¿Qué requerimiento de calidad interna se atiende?

A) Modificabilidad B) Facilidad de prueba C) Despliegue

Respuesta: A) Modificabilidad.

Explicación: Las interfaces y el diseño orientado a objetos facilitan cambiar o extender el sistema sin alterar toda la estructura. Por eso el atributo principal atendido es la capacidad de modificación.

Reactivo 6

Un sistema de ventas en línea clasifica clientes como plata, oro o platino según compras acumuladas y asigna descuentos. ¿Qué elemento corresponde a una regla de negocio?

A) Determinar monto a pagar B) Calcular descuentos C) Asociar niveles de cliente distinguido

Respuesta: C) Asociar los niveles de cliente distinguido.

Explicación: La regla de negocio define cómo se clasifican los clientes según montos acumulados y qué categoría obtienen. Calcular descuentos o montos puede ser una operación del sistema, pero la regla central es la asociación entre compra acumulada, nivel y beneficio.

Reactivo 7

Una aseguradora necesita calcular tendencias y pronósticos de primas con gráficas sin invertir tiempo programando desde cero ni comprar componentes adicionales. ¿Qué lenguaje se requiere?

A) Python B) Java C) C++

Respuesta: A) Python.

Explicación: Python es adecuado para análisis de datos, pronósticos y visualización por su ecosistema de bibliotecas. Java y C++ pueden hacerlo, pero normalmente requieren más desarrollo para análisis estadístico y gráficas.

Reactivo 8

Un equipo desarrollará una app para venta de cuadros que se usará en celulares y tabletas con el mejor rendimiento. ¿Qué requerimiento corresponde?

A) Desarrollo para sistemas operativos nativos B) Desarrollo híbrido C) Desarrollo con HTML5

Respuesta: A) La aplicación será desarrollada para sistemas operativos nativos.

Explicación: Cuando la prioridad es el mejor rendimiento en dispositivos móviles, el desarrollo nativo suele aprovechar mejor los recursos del sistema operativo y del hardware.

Reactivo 9

Un cliente verifica que todos los estados posibles, cambios de estado, entradas, productos y restricciones estén descritos en los requerimientos. ¿A qué característica corresponde?

A) Completitud B) Correctitud C) Factibilidad

Respuesta: A) Completitud.

Explicación: La completitud exige que los requerimientos incluyan todos los escenarios, entradas, salidas, restricciones y condiciones necesarias. El caso se enfoca en que no falte información relevante.

Reactivo 10

Un cliente especifica que el software debe tener alta velocidad para comenzar a trabajar y lo marca como elemento de alta importancia. ¿Qué característica atiende?

A) No ambigüedad B) Priorizable C) Verificable

Respuesta: B) Priorizable.

Explicación: El enunciado señala la importancia del requerimiento para el cliente. Un requerimiento priorizable permite ordenar su atención según valor, urgencia o impacto en el producto.

Cierre de estudio

Para resolver este tipo de reactivos conviene identificar primero si el caso habla de requerimientos, arquitectura, calidad, pruebas, patrones o decisiones tecnológicas. Después se debe distinguir entre lo que pide el negocio, lo que necesita el usuario y lo que corresponde a una decisión técnica. Esta separación ayuda a evitar confundir objetivos de negocio con soluciones de implementación.