

Ejemplo de Examen

iSAQB[®] Certified Professional for Software Architecture – Foundation Level (CPSA-F[®])

Versión del documento: 2020.1-ES-rev6

Basado en el Programa de Estudio - versión V5.1-ES; 2 de enero de 2020



Descripción del ejemplo de examen de Certified Professional for Software Architecture - Foundation Level (CPSA-F®)

Este examen es un ejemplo de examen, basado en el examen de certificación de Certified Professional for Software Architecture - Foundation Level (CPSA-F®) en su forma y alcance. Sirve para ilustrar el examen real iSAQB® CPSA® así como para preparar el examen correspondiente.

El ejemplo de examen consta de 39 preguntas de selección múltiple, que pueden evaluarse con 1 ó 2 puntos según el nivel de dificultad. Para aprobar el examen se debe alcanzar, como mínimo, el 60 por ciento de la puntuación. En este ejemplo de examen puedes obtener 50,0 puntos, necesitas 30,0 puntos para aprobar.

Las reglas generales para la evaluación de las respuestas son las siguientes: las respuestas correctas suponen puntos positivos, las respuestas incorrectas suponen una deducción de puntos, pero sólo con respecto a la pregunta correspondiente. Si la respuesta incorrecta a una pregunta lleva a una puntuación negativa, esta pregunta se evalúa con un total de 0 puntos.

Las preguntas de selección múltiple del ejemplo de examen están estructuradas en tres tipos de preguntas:

Preguntas Tipo A (Una única opción, una única respuesta correcta):

Seleccionar la única respuesta correcta a una pregunta de la lista de posibles respuestas. Sólo hay una respuesta correcta. Usted obtendrá la puntuación especificada por seleccionar la respuesta correcta. La puntuación es de 1-2 puntos dependiendo del nivel de dificultad.

Preguntas Tipo P (Seleccionar de entre muchos, selección múltiple):

Seleccionar el número de respuestas correctas dadas en el texto de la lista de posibles respuestas a una pregunta. Seleccionar tantas respuestas como se requieran en el texto introductorio. Usted obtendrá 1/n del total de puntos por cada respuesta correcta. Por cada respuesta incorrecta, se deduce 1/n de los puntos. La puntuación es de 1-2 puntos dependiendo del nivel de dificultad.

Preguntas Tipo K (Preguntas de asignación, elegir la categoría):

Asignar a cada respuesta la opción correcta entre dos alternativas ("correcto" o "incorrecto", "aplica" o "no aplica"). Usted obtendrá 1/n de los puntos por cada alternativa que haya seleccionado de forma correcta. Por cada alternativa incorrecta, se deducirán 1/n de los puntos. Si no selecciona una respuesta en una línea, no hay ni puntos ni deducciones. La puntuación es de 1-2 puntos dependiendo del nivel de dificultad.

Para obtener información más detallada sobre los tipos de preguntas y el sistema de puntuación, se puede consultar las [normas de examen CPSA-F](#).

El tiempo disponible para realizar el examen es de 75 minutos para los hablantes nativos y 90 minutos para los no nativos. A fin de asegurar que la preparación del examen es lo más auténtica posible, se debe respetar el tiempo de examen establecido y no utilizar ningún tipo de recurso auxiliar (como material de seminario, libros, Internet, etc.).

El ejemplo de examen puede ser corregido utilizando las soluciones al ejemplo de examen.

Siempre que iSAQB® e. V. se indique como fuente y titular de los derechos de autor, el presente ejemplo de examen se podrá utilizar en el contexto de cursos de formación, para la preparación de exámenes o se podrá distribuir de forma gratuita.

Sin embargo, se prohíbe explícitamente utilizar estas preguntas de examen en un examen real.

Pregunta 1

Pregunta A: Seleccione una respuesta.

1 punto

ID: Q-20-04-01

¿Cuántas definiciones de "arquitectura de software" hay?

- (a) Hay exactamente una definición para todos los tipos de sistemas.
- (b) Hay una definición para cada tipo de sistema de software (por ejemplo, "embebido", "en tiempo real", "de apoyo a la toma de decisiones", "web", "por lotes", ...).
- (c) Hay, por lo menos, una docena de definiciones diferentes.

Pregunta 2

Pregunta P: Seleccione los tres aspectos más adecuados.

1 punto

ID: Q-20-04-02

¿Cuáles son los **TRES** aspectos que abarca el término "arquitectura de software"?

- (a) Componentes.
- (b) Conceptos transversales.
- (c) Interfaces (internas y externas).
- (d) Esquemas de bases de datos.
- (e) Dimensionamiento del hardware.

Pregunta 3 *Pregunta P: Seleccione las cuatro respuestas más adecuadas.* 2 puntos
ID: Q-17-13-01

¿Cuáles son las **CUATRO** afirmaciones más adecuadas relativas a conceptos (transversales)?

- (a) El uso uniforme de conceptos reduce el acoplamiento entre bloques estructurales.
- (b) La definición de los conceptos adecuados asegura la integridad conceptual de la arquitectura.
- (c) Se consigue más fácilmente un tratamiento de excepciones uniforme cuando los arquitectos se ponen de acuerdo con los desarrolladores sobre un concepto adecuado antes de la implementación.
- (d) Debe haber un concepto documentado de forma explícita para cada objetivo de calidad.
- (e) Los conceptos son un medio para aumentar la consistencia.
- (f) Un concepto puede definir restricciones para la implementación de muchos bloques estructurales.
- (g) Un concepto podría ser implementado por un solo bloque estructural.

Pregunta 4 *Pregunta K: Seleccione «adecuado» o «no adecuado» en cada caso.* 2 puntos
ID: Q-17-13-02

En su proyecto, están trabajando en la documentación de la arquitectura del software tres arquitectos y siete desarrolladores. ¿Qué métodos son idóneos para conseguir una documentación consistente y adecuada, y cuáles no lo son?

- | Adecuado | No adecuado | |
|--------------------------|--------------------------|---|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) El arquitecto principal crea la documentación. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) Se utilizan plantillas idénticas para la documentación. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) Todas las partes de la documentación son extraídas de forma automática del código fuente. |

Pregunta 5 *Pregunta P: Seleccione las cuatro opciones más adecuadas.* 1 punto
ID: Q-17-13-03

¿Cuáles **CUATRO** de las siguientes técnicas son las más adecuadas para representar la interacción de bloques estructurales en tiempo de ejecución?

- (a) Diagramas de flujo.
- (b) Diagramas de actividad.
- (c) Representación de los flujos de pantalla (secuencia de interacciones con el usuario).
- (d) Diagrama de secuencia.
- (e) Diagrama de Venn lineal.
- (f) Lista numerada de pasos secuenciales.
- (g) Descripción tabular de las interfaces.
- (h) Diagramas de clase.

Pregunta 6 *Pregunta P: Seleccione las tres opciones más adecuadas.* 1 punto
ID: Q-17-13-04

¿Cuáles **TRES** de los siguientes principios aplican a la prueba?

- (a) En general, no es posible realizar pruebas exhaustivas.
- (b) En los componentes con muchos errores previos conocidos, las posibilidades de que se produzcan errores adicionales son altas.
- (c) Probar de forma suficiente puede demostrar que un programa está libre de errores.
- (d) Probar sólo puede mostrar la existencia de errores.
- (e) La programación funcional no permite realizar pruebas automatizadas.

Pregunta 7 *Pregunta K: Seleccione «verdadero» o «falso» en cada caso.* **2 puntos**
ID: Q-17-13-05

¿Cuáles de los siguientes enunciados relativos al principio de diseño "ocultación de información" son verdaderos y cuáles son falsos?

Verda-

dero Falso

- | | | | |
|--------------------------|--------------------------|-----|--|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | La adhesión al principio de "ocultación de información" aumenta la flexibilidad para las modificaciones. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | La ocultación de información implica ocultar deliberadamente la información a los que invocan o a los consumidores del bloque estructural. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | La ocultación de información hace más difícil distinguir entre la interfaz y la implementación. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | La ocultación de información es un derivado del enfoque de refinamiento incremental a lo largo del flujo de control. |
| <input type="checkbox"/> | <input type="checkbox"/> | (e) | En el desarrollo orientado a objetos, la ocultación de información es principalmente relevante a nivel de clase. |

Pregunta 8 *Pregunta P: Seleccione las dos opciones más adecuadas.* **1 punto**
ID: Q-20-04-03

¿Cuáles son los **DOS** objetivos más importantes de la arquitectura de software?

- | | | |
|--------------------------|-----|--|
| <input type="checkbox"/> | (a) | Mejorar la precisión de los patrones en la estructura e implementación. |
| <input type="checkbox"/> | (b) | Alcanzar los requisitos de calidad de forma comprensible. |
| <input type="checkbox"/> | (c) | Hacer posible que las pruebas de integración y aceptación de sistema sean Rentables. |
| <input type="checkbox"/> | (d) | Hacer posible una comprensión básica de las estructuras y conceptos para el equipo de desarrollo y otros implicados. |

Pregunta 9

Pregunta P: Seleccione «verdadero» o «falso» en cada caso.

2 puntos

ID: Q-20-04-12

Póngase en la posición de un arquitecto de software para una gran aplicación de negocio en el dominio de la banca o los seguros. ¿Cuál de las siguientes afirmaciones es verdadera y cuál es falsa?

Verda-

dero Falso

- | | | | |
|--------------------------|--------------------------|-----|--|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | Su arquitectura debe estar estructurada de forma que permita realizar cambios en los procesos de negocio correspondientes sin requerir una amplia reestructuración de la arquitectura de software. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | Las cualidades requeridas del producto deberían impulsar sus decisiones arquitectónicas. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | Para ser independiente de la infraestructura debe decidir su estructura de arquitectura de software clave antes de que los arquitectos de la infraestructura seleccionen el hardware o la infraestructura para un producto. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | Su arquitectura de software debería prever cambios en la tecnología (es decir, nuevos marcos de interfaz de usuario, diferentes estrategias de despliegue, nuevos dispositivos periféricos) y sólo requerir una adaptación local cuando se produzca. |

Pregunta 10

Pregunta P: Seleccione las tres responsabilidades más importantes. 2 puntos

ID: Q-20-04-06

¿Cuáles son sus **TRES** responsabilidades más importantes como arquitecto de software con respecto a los requisitos?

- (a) Ayudar al personal de negocio a expresar requisitos de calidad de forma que puedan ser probados.
- (b) Ayudar a identificar nuevas oportunidades de negocio basadas en sus conocimientos tecnológicos.
- (c) Rechazar los requisitos de negocio que contengan riesgos técnicos.
- (d) Reescribir los requisitos de negocio en unos términos que puedan ser comprendidos por su equipo de desarrollo.
- (e) Comprobar la viabilidad tecnológica de los requisitos.

Pregunta 11 *Pregunta P: Seleccione las tres acciones más importantes.* **1 punto**

ID: Q-20-04-07

Como arquitecto, usted es responsable de mantener en funcionamiento un sistema legado de acuerdo con los requisitos actuales de su negocio. ¿Cuáles son los **TRES** puntos de acción más importantes de su agenda?

- (a) Negociar el presupuesto de mantenimiento para su equipo.
- (b) Asegurar una documentación actualizada del sistema desplegado.
- (c) Analizar el impacto de nuevos requisitos en el sistema actual.
- (d) Animar a los miembros del equipo a aprender nuevos lenguajes de programación.
- (e) Sugerir a su dirección actualizaciones tecnológicas además de los requisitos del negocio.

Pregunta 12 *Pregunta K: Seleccione «verdadero» o «falso» en cada caso.* **2 puntos**

ID: Q-20-04-08

Usted es el arquitecto responsable de un producto de una familia de productos. La familia de productos tiene un arquitecto general de la familia de productos. ¿Cuáles de las siguientes afirmaciones son verdaderas y cuáles son falsas?

Verda-
dero Falso

- (a) Usted tiene que aceptar las restricciones que se aplican a toda la familia de productos también para su producto.
- (b) Dado que las partes de esta familia de productos son productos que se pueden vender por separado, su producto no está sujeto a las restricciones del conjunto.
- (c) Debe mantener reuniones periódicas con sus compañeros arquitectos de producto y con el arquitecto de la familia para negociar los requisitos de calidad y las restricciones comunes.
- (d) Puede negociar con el arquitecto de la familia las desviaciones de los requisitos de calidad que se han definido para el conjunto.

Pregunta 13 *Pregunta K: Seleccione «verdadero» o «falso» en cada caso.* **1 punto**
ID: Q-20-04-09

¿Cuáles de las siguientes afirmaciones son verdaderas y cuáles son falsas?

Verda-

dero Falso

- | | | | |
|--------------------------|--------------------------|-----|--|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | Las piedras angulares de la arquitectura pueden decidirse durante el desarrollo iterativo de las prestaciones. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | El esfuerzo total invertido en el trabajo de arquitectura es mucho mayor en los proyectos iterativos en comparación con los proyectos en cascada. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | Los proyectos ágiles no necesitan documentos de arquitectura, ya que el equipo de desarrollo utiliza las reuniones diarias para comunicar las decisiones. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | Si sus sistemas consisten en un conjunto de microservicios, no es necesario un documento de arquitectura central, ya que cada servicio es libre de elegir sus tecnologías. |

Pregunta 14 *Pregunta K: Seleccione «verdadero» o «falso» en cada caso.* **2 puntos**
ID: Q-20-04-10

¿Cuáles de las siguientes afirmaciones sobre los objetivos de un proyecto y los objetivos de la arquitectura son verdaderas y cuáles son falsas?

Verda-

dero Falso

- | | | | |
|--------------------------|--------------------------|-----|--|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | Los objetivos de un proyecto pueden incluir tanto requisitos funcionales como de calidad. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | Los objetivos arquitectónicos son un derivado de los requisitos de calidad del sistema o producto. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | Los implicados del negocio deben concentrarse en los objetivos de negocio y no interferir con los objetivos de arquitectura. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | Para evitar conflictos, los objetivos de negocio y los objetivos de arquitectura deben ser conjuntos que no se superpongan. |

Pregunta 15 *Pregunta P: Seleccione las dos respuestas que mejor encajen.* 1 punto
ID: Q-20-04-11

¿Qué significa la regla "explícito, no implícito" para el trabajo de arquitectura? Seleccione las **DOS** respuestas que mejor se ajusten.

- (a) Los arquitectos deben evitar las estructuras recursivas y sustituirlas por bucles explícitos.
- (b) Los arquitectos deberían hacer explícitas las suposiciones que conducen a decisiones.
- (c) Los arquitectos deberían insistir explícitamente en explicaciones en lenguaje natural (es decir, comentarios) para cada bloque estructural.
- (d) Los arquitectos deberían insistir explícitamente en las justificaciones escritas o al menos orales de las estimaciones del esfuerzo de desarrollo de su equipo.
- (e) Los arquitectos deberían hacer explícitos los prerequisites de sus decisiones.

Pregunta 16 *Pregunta P: Seleccione las tres respuestas más adecuadas.* 1 punto
ID: Q-20-04-19

Identifique los **TRES** ejemplos más adecuados para las categorías típicas de sistemas de software.

- (a) Sistema por lotes.
- (b) Sistema interactivo en línea.
- (c) Sistema Linnés.
- (d) Sistema en tiempo real embebido.
- (e) Sistema de integración de la prueba.

Pregunta 17 *Pregunta P: Seleccione los tres que se encuentran más a menudo en la práctica.* 1 punto

ID: Q-20-04-32

Hay muchos enfoques que conducen a una arquitectura de software. ¿Cuáles de las siguientes opciones son los **TRES** enfoques más frecuentes en la práctica?

- (a) Diseño Guiado por la Interfaz de Usuario.
- (b) Diseño Guiado por el Dominio.
- (c) Desarrollo de Arquitectura basado en Vistas.
- (d) Diseño Ascendente.
- (e) Votación Mayoritaria.

Pregunta 18 *Pregunta P: Seleccione las tres vistas empleadas con mayor frecuencia.* 1 punto

ID: Q-20-04-38

Muchos métodos de desarrollo de arquitectura sugieren un enfoque basado en vistas. ¿Cuáles de las siguientes vistas son las **TRES** más utilizadas?

- (a) Vista de base de datos física.
- (b) Vista contextual.
- (c) Vista de bloque estructural/componente.
- (d) Vista guiada por la prueba.
- (e) Vista de configuración.
- (f) Vista de tiempo de ejecución.

Pregunta 19 *Pregunta P: Seleccione las dos respuestas más adecuadas.* 1 punto
ID: Q-20-04-22

Usted documenta un componente de su arquitectura de software. ¿Qué información debería contener su descripción de la caja negra?

- (a) Interfaces.
- (b) Responsabilidad.
- (c) Estructura interna.
- (d) Pistas para la implementación.

Pregunta 20 *Pregunta P: Seleccione las dos respuestas más adecuadas.* 1 punto
ID: Q-20-04-17

¿Qué prerequisites se deben cumplir antes de desarrollar una arquitectura de software? Elija las **DOS** respuestas más adecuadas.

- (a) La especificación de requisitos para el sistema es completa, detallada y consistente.
- (b) Se conocen las cualidades más importantes del sistema.
- (c) Se conocen las restricciones de la organización.
- (d) Se ha seleccionado el lenguaje de programación.
- (e) Se dispone del hardware necesario para el equipo de desarrollo.

Pregunta 21 *Pregunta P: Seleccione las tres respuestas más adecuadas.* 1 punto
ID: Q-20-04-18

¿Qué factores pueden influir en el diseño de una arquitectura de software? Elija las **TRES** respuestas más adecuadas.

- (a) Políticos.
- (b) Organizativos.
- (c) Técnicos.
- (d) Virtuales.

Pregunta 22 *Pregunta A: Seleccione una respuesta.* 1 punto
ID: Q-20-04-28

¿Cuál de las siguientes cualidades puede mejorarse con mayor probabilidad utilizando una arquitectura en capas?

- (a) Eficiencia de ejecución (rendimiento).
- (b) Flexibilidad para modificar o cambiar el sistema.
- (c) Flexibilidad en tiempo de ejecución (configurabilidad).
- (d) No repudio.

Pregunta 23 *Pregunta A: Seleccione una respuesta.* 1 punto
ID: Q-20-04-33

¿Para qué tipo de sistema se puede utilizar el patrón de arquitectura pizarra?

- (a) Sistemas en tiempo real.
- (b) Sistemas basados en reglas.
- (c) Sistemas Linnés.
- (d) Sistemas de seguridad crítica.

Pregunta 24 *Pregunta A: Seleccione una respuesta.* 1 punto
ID: Q-20-04-20

¿Qué objetivos se pretenden lograr con el principio de inversión de la dependencia?

- (a) Los bloques estructurales grandes no deben depender de los bloques estructurales pequeños.
- (b) Los componentes deben ser capaces de crear componentes dependientes más fácilmente.
- (c) Los bloques estructurales sólo deben depender unos de otros a través de abstracciones.

Pregunta 25 *Pregunta K: Seleccione «acoplamiento fuerte» o «acoplamiento débil» en cada caso.* 1 punto

ID: Q-20-04-21

¿Cuáles son las características del acoplamiento fuerte (alto) o del acoplamiento débil (bajo)?

Acoplamiento fuerte	Acoplamiento débil
---------------------	--------------------

- | | | | |
|--------------------------|--------------------------|-----|--|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | Los bloques estructurales invocan directamente a los bloques dependientes, es decir, sin desvíos a través de interfaces o abstracciones. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | Los bloques estructurales utilizan tipos de datos comunes. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | Los bloques estructurales utilizan una base de datos común. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | Al diseñar los bloques estructurales, se ha aplicado sistemáticamente el principio de inversión de la dependencia. |

Pregunta 26 *Pregunta P: Seleccione las dos respuestas más adecuadas.* 2 puntos

ID: Q-20-04-14

¿Qué **DOS** afirmaciones sobre el principio "No te repitas" (NTR) son correctas? (En otras palabras: ¿Qué podría ocurrir si partes del código fuente o de la configuración existen en múltiples copias en el sistema?)

- (a) "No te repitas" (NTR) reduce la seguridad.
- (b) El cumplimiento estricto de "No te repitas" (NTR) podría conducir a un mayor acoplamiento.
- (c) Los componentes del sistema que contienen código redundante pueden mejorarse independientemente unos de otros.
- (d) La adhesión a "No te repitas" (NTR) conduce a una reducción de los vectores de ataque en la seguridad informática.
- (e) La aplicación de los patrones de capas permite una aplicación consistente del principio "No te repitas" (NTR).

Pregunta 27 *Pregunta K: Seleccione «verdadero» o «falso» en cada caso.* **2 puntos**
ID: Q-20-04-15

Usted puede comunicar aspectos de su arquitectura de software de forma verbal y/o por escrito. ¿Cómo se correlacionan estas variantes? ¿Cuáles de las siguientes afirmaciones son verdaderas y cuáles son falsas?

Verda-

dero Falso

- | | | | |
|--------------------------|--------------------------|-----|---|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | La comunicación verbal debería complementar la documentación escrita. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | La retroalimentación a las decisiones de arquitectura debe hacerse por escrito para asegurar la trazabilidad. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | La documentación escrita debe preceder siempre a la comunicación oral. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | Los arquitectos deben elegir una variante (oral o escrita) y atenerse a esta elección durante todo el desarrollo. |

Pregunta 28 *Pregunta K: Seleccione «verdadero» o «falso» en cada caso.* **2 puntos**
ID: Q-20-04-37

¿Cuáles de los siguientes enunciados sobre las notaciones para las vistas arquitectónicas son verdaderos y cuáles son falsos?

Verda-

dero Falso

- | | | | |
|--------------------------|--------------------------|-----|--|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | Business Process Model & Notation (BPMN) sólo debe ser utilizado por los analistas de negocio y no para la documentación de la arquitectura. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | Los modelos de despliegue UML son la única forma de documentar la correspondencia de los componentes de software con la infraestructura. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | Los diagramas de paquetes UML pueden utilizarse para capturar la vista de bloque estructural de las arquitecturas de software. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | Siempre que se explique la notación (por ejemplo, mediante una leyenda), cualquier notación puede ser suficiente para describir las estructuras de bloques estructurales y colaboración. |

Pregunta 29 *Pregunta P: Seleccione las dos respuestas más adecuadas.* 1 punto
ID: Q-20-04-13

¿Qué vistas arquitectónicas tienen aplicación práctica para el desarrollo de arquitecturas de software?

- (a) Vista de patrón.
- (b) Vista de observador.
- (c) Vista de bloque estructural (o componente).
- (d) Vista de despliegue.

Pregunta 30 *Pregunta P: Seleccione las dos respuestas más adecuadas.* 1 punto
ID: Q-20-04-23

La vista contextual incluye el contexto de negocio y el contexto técnico. Elija las **DOS** respuestas más adecuadas que correspondan al contexto técnico.

- (a) El contexto técnico contiene los canales físicos entre su sistema y su entorno.
- (b) El contexto técnico contiene toda la infraestructura en la que se despliegan los componentes de su sistema.
- (c) El contexto técnico debe incluir los precios del hardware o de los servicios en la nube utilizados como infraestructura para su arquitectura.
- (d) El contexto técnico contiene información sobre el lenguaje de programación elegido, así como todos los marcos de trabajo utilizados para implementar su arquitectura de software.
- (e) El contexto técnico puede contener elementos diferentes a los del contexto de negocio.

Pregunta 31 *Pregunta P: Seleccione las dos razones más pertinentes.* **1 punto**
ID: Q-20-04-24

La documentación de la arquitectura del software podría contener descripciones de asuntos de interés transversal. Elija las **DOS** mejores razones por las que la documentación de asuntos de interés transversal es útil.

- (a) Los conceptos transversales deben centrarse en el dominio y estar libres de información técnica.
- (b) Los aspectos o conceptos que se utilizan en múltiples partes de su arquitectura de software deben ser descritos de una manera no redundante.
- (c) Los conceptos transversales pueden reutilizarse en más productos dentro de la misma organización.
- (d) Los conceptos transversales deben ser implementados por especialistas. Por lo tanto, es útil una documentación por separado.

Pregunta 32 *Pregunta K: Seleccione «verdadero» o «falso» en cada caso.* **1 punto**
ID: Q-20-04-25

¿Cuáles son las directrices para un buen diseño de interfaz? ¿Cuál de los siguientes enunciados son verdaderos y cuáles son falsos?

Verda-
dero Falso

- (a) El uso de las interfaces debe ser fácil de aprender.
- (b) El código cliente debe ser fácil de entender.
- (c) Una interfaz debe ser definida por el proveedor del servicio adecuado.
- (d) Las especificaciones de las interfaces deben contener aspectos funcionales y no funcionales.

Pregunta 33 *Pregunta K: Seleccione «verdadero» o «falso» en cada caso.* 1 punto
ID: Q-20-04-26

Según una definición: "La arquitectura del software es la suma de todas las decisiones que se toman durante el desarrollo". ¿Cuáles de los siguientes enunciados sobre las decisiones de diseño/arquitectura son verdaderos y cuáles son falsos?

Verda-

dero Falso

- | | | | |
|--------------------------|--------------------------|-----|--|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | Las decisiones arquitectónicas pueden estar implícitamente contenidas en la estructura de la vista de bloque estructural/componente. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | Los arquitectos de software deberían justificar todas las decisiones de diseño por escrito. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | Las decisiones de arquitectura pueden tener interdependencias mutuas. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | Las soluciones de compromiso entre requisitos de calidad contradictorios deberían ser decisiones explícitas. |

Pregunta 34 *Pregunta K: Seleccione si se trata de «Es una buena razón» o «No es una buena razón» en cada caso.* 1 punto
ID: Q-20-04-31

¿Cuáles de los siguientes enunciados son una buenas razones para mantener (adecuadamente) la documentación de la arquitectura y cuáles no son buenas razones?

Es una
buena razón

No es una
buena razón

- | | | | |
|--------------------------|--------------------------|-----|---|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | Para permitir la incorporación de nuevos desarrolladores. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | Para cumplir con las restricciones legales. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | Para dar soporte al trabajo de los equipos distribuidos. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | Para apoyar las futuras mejoras del producto. |

Pregunta 35 *Pregunta K: Seleccione «Presencia de conflicto» o «Ausencia de conflicto» en cada caso.* *1 punto*

ID: Q-20-04-30

¿Cuáles de los siguientes pares de cualidades se encuentran normalmente en conflicto entre sí, y cuáles no?

- | Presencia de conflicto | Ausencia de conflicto | |
|--------------------------|--------------------------|---|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) Comprensibilidad – Legibilidad. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) Usabilidad – Seguridad. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) Configurabilidad en tiempo de ejecución – Robustez. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) Seguridad – Cumplimiento de la legislación/normativa. |

Pregunta 36 *Pregunta P: Seleccione las dos alternativas más adecuadas.* *1 punto*

ID: Q-20-04-27

La norma ISO 25010 proporciona características de calidad genéricas para los sistemas de software. ¿Cómo se pueden concretar los requisitos de calidad relativos a estas características? Elija las **DOS** mejores alternativas.

- (a) Desarrollar prototipos de interfaz de usuario.
- (b) Definir interfaces explícitas.
- (c) Discutir o escribir escenarios.
- (d) Crear pruebas automatizadas.
- (e) Crear un árbol de calidad.

Pregunta 37 *Pregunta A: Seleccione una respuesta.* 1 punto
ID: Q-20-04-28

¿Cuál de las siguientes opciones no contribuye al análisis cualitativo de su arquitectura de software? Elija la única respuesta incorrecta.

- (a) Métricas.
- (b) Modelos de arquitectura.
- (c) Escenarios de calidad.
- (d) Informes de estado del proyecto.
- (e) Archivos de registro.

Pregunta 38 *Pregunta P: Seleccione los dos indicadores más adecuados.* 2 puntos
ID: Q-20-04-29

Usted intenta analizar su arquitectura cuantitativamente. ¿Cuáles son los **DOS** indicadores más adecuados para las áreas problemáticas de la arquitectura?

- (a) Acoplamiento elevado de los componentes.
- (b) Nombres inadecuados de métodos públicos.
- (c) Ausencia de comentarios.
- (d) Agrupamientos de errores.
- (e) Número de casos de prueba por componente.

Pregunta 39

Pregunta P: Seleccione dos respuestas.

1 punto

ID: Q-20-04-36

¿Cuál de las siguientes alternativas no se puede medir en su arquitectura de software? Elija las **DOS** respuestas menos factibles.

- (a) Tamaño de los bloques estructurales (por ejemplo, líneas de código).
- (b) Tasa de cambio del código fuente de los componentes.
- (c) Cohesión de los componentes de la arquitectura.
- (d) Nivel de seguridad de un componente.
- (e) Número de desarrolladores que conocen un componente específico.