Instrucciones

- Cada pregunta múltiple opción respondida correctamente tiene un valor de 4,762 puntos.
- Cada pregunta múltiple opción respondida incorrectamente resta 1,587 puntos.
- Ante dos opciones correctas en una pregunta, se debe seleccionar la opción más completa.
- La evaluación es de carácter individual y la duración es de dos horas.
- El puntaje total es de 100 puntos y se aprueba con 60 o más puntos.

1. Sobre los *drivers* y *stubs* en las pruebas de software:

I. Son componentes de software con defectos inyectados que ayudan a determinar la calidad del conjunto de casos de prueba.
II. Viabilizan ciertas pruebas de software.
III. Son componentes de software que forman parte del software funcional que se entrega al cliente.
IV. Pueden ser usados en diferentes niveles de prueba, en particular en las pruebas unitarias y de integración.
V. No son componentes de software, forman parte de la documentación de ciertos tipos de pruebas.

Seleccione la opción correcta:

a) Sólo las afirmaciones I, II, y V son correctas.
b) Sólo las afirmaciones II, III y IV son correctas.
c) Sólo las afirmaciones II y IV son correctas.
d) Sólo las afirmaciones II y V son correctas.

2. Dado el siguiente fragmento de código

```
if a < b and b > 0
    if (a <> 0)
        return a div b;
    else return b div a;
else return 1;
```

en un lenguaje en el cual si se ejecuta una división por cero se aborta la ejecución. Seleccione la opción correcta:

a) El código no contiene defectos.
b) El código contiene defectos y es posible detectarlos con un conjunto de casos de prueba que cumpla con el criterio de condición múltiple.
c) El código contiene defectos y no es posible detectarlos realizando una inspección de código.
d) El código contiene defectos y no es posible detectarlos con pruebas estáticas.

3. Usted es el líder de un equipo de desarrollo de 5 personas que realiza un software a medida para un cliente local. El producto es pequeño y utilizan un modelo de proceso en cascada para su construcción. Seleccione la opción correcta:

   a) No será necesario generar pruebas de regresión ya que éstas son útiles únicamente en procesos iterativos incrementales.
   b) Podría resultar útil realizar algunas pruebas de regresión, aunque se podría prescindir de parte (o toda) su automatización.
   c) Al ser un proceso en cascada, solo sería necesario realizar pruebas a nivel de sistema y de aceptación.
   d) Ninguna de las opciones anteriores es correcta.

4. Sobre la gestión de los cambios, seleccione la opción **INCORRECTA**.

   a) Durante el desarrollo la decisión de un cambio podría ser tomada directamente por el desarrollador.
   b) El cliente podría estar directamente involucrado en la toma de decisiones de la gestión de los cambios.
   c) En ciertos contextos/situaciones es posible omitir el comité de control de cambios.
   d) El costo de implementación de un cambio no incluye el retrabajo asociado a ese cambio.

5. De las siguientes figuras, ¿cuál o cuáles podrían representar un diagrama de despliegue?:

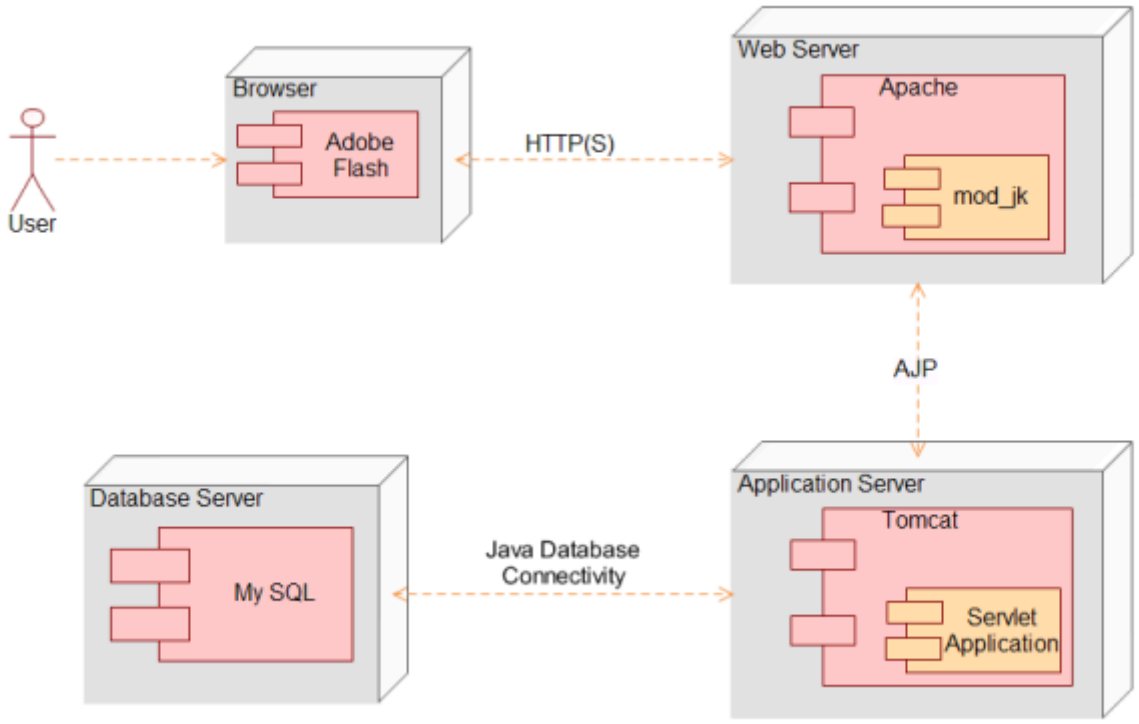

Figura 1

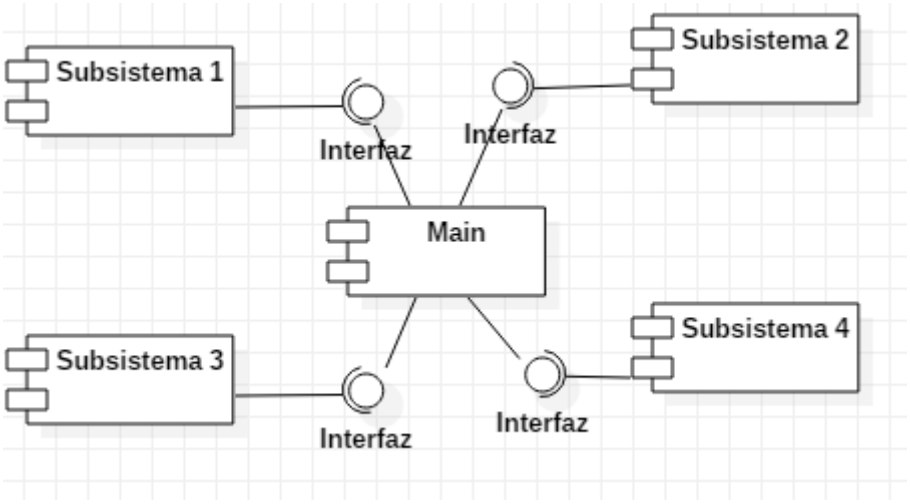

Figura 2

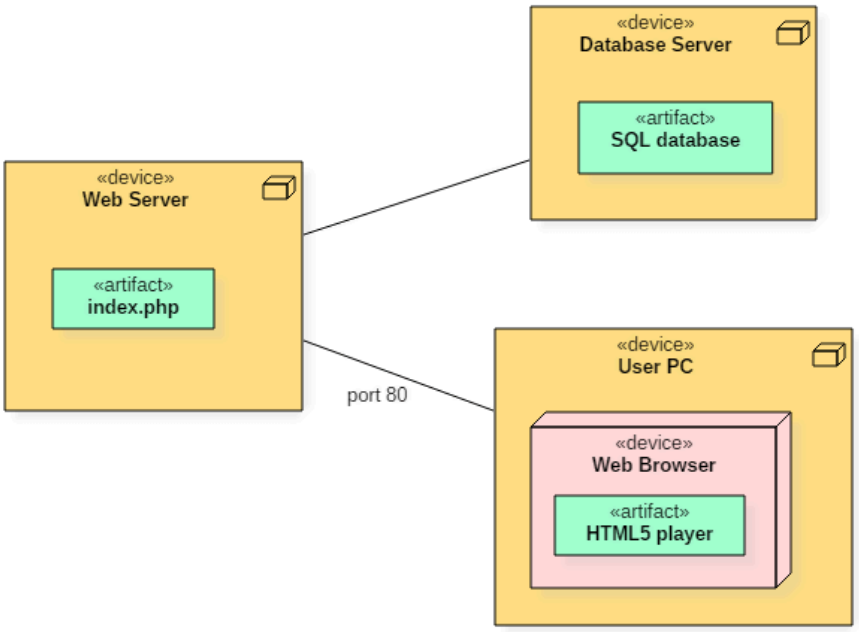

Figura 3

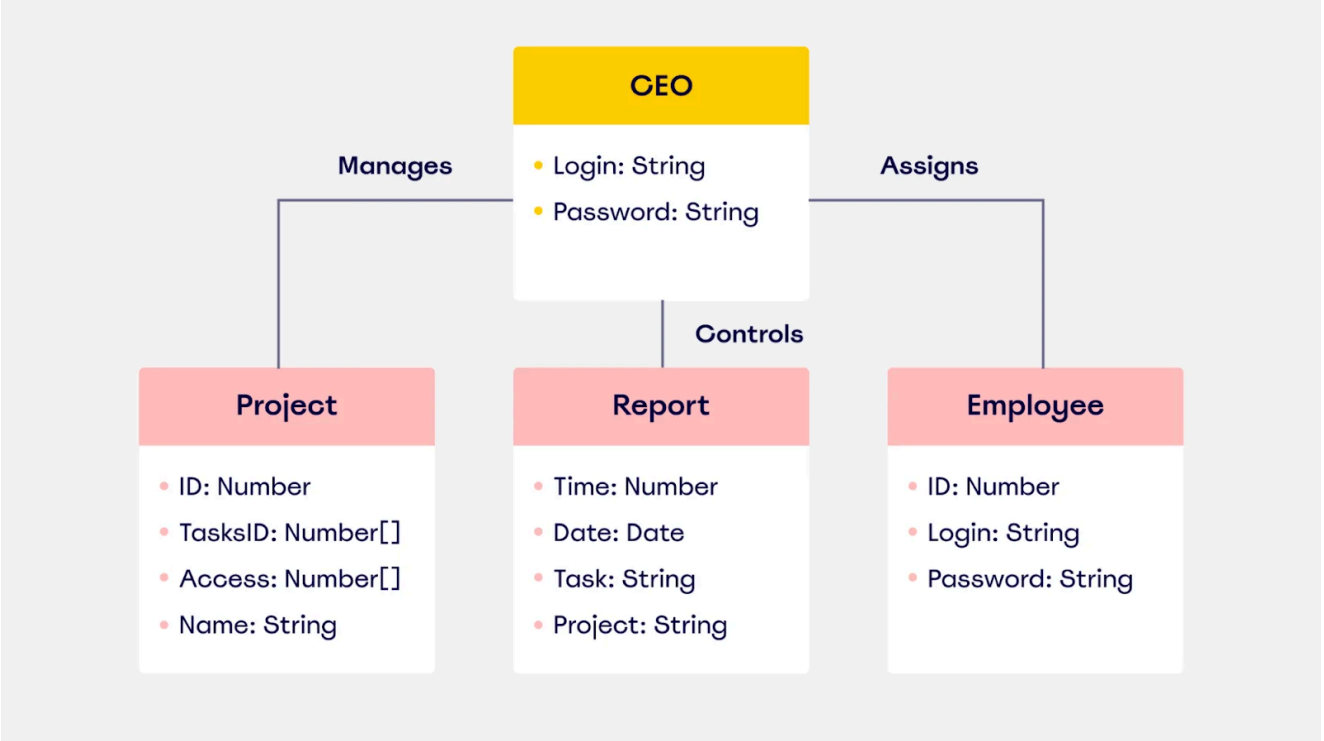

Figura 4

a. Sólo Figura 1 y Figura 2
b. Sólo Figura 2
c. Sólo Figura 1 y Figura 3
d. Sólo Figura 4.

6. Indique en qué aspectos brinda ventajas explicitar la arquitectura de software de un sistema:

I. Comunicación con las partes interesadas (stakeholders).
II. Análisis del sistema.
III. Reutilización a gran escala.
IV. Ninguna de las anteriores.

a) Solo I, II y III son correctas.
b) Solo II y III son correctas.
c) Solo IV es correcta.
d) Solo I y II son correctas.

7. Indique qué diagramas se podrían utilizar para documentar las siguientes vistas arquitectónicas:

I. Vista de desarrollo → Diagrama de componentes.
II. Vista física → Diagrama de despliegue (deploy).
III. Vista lógica → Diagramas de clases, secuencia, y actividad.
IV. Ninguno de los anteriores.

a. Solo II y III son correctas.
b. Solo II y IV son correctas.
c. Sólo IV es correcta.
d. Solo I, II y III son correctas.

8. Indique cuáles de las siguientes afirmaciones sobre la liberación de un sistema son correctas.

I. Comprende todas las actividades que son necesarias, luego de que se ha construido y probado, para comenzar el uso de un software.
II. No se trata solamente de desplegar el sistema.
III. Incluye: instalación y configuración, adopción (o conversión) del software, entrenamiento y apoyo en el uso.
IV. Ninguna de las afirmaciones anteriores es correcta.

a) Solo las afirmaciones I y II son correctas.
b) Solo las afirmaciones II y III son correctas.
c) Solo las afirmaciones I, II y III son correctas.
d) Solo la afirmación IV es correcta.

9. Indique por qué puede ser necesaria la evolución del software:

I. Porque necesita adaptarse al entorno operativo.
II. Porque necesita adaptarse al usuario.
III. Porque necesita mejorar en el desempeño y/o en funcionalidades.
IV. Porque necesita corregir errores.

a) Solo las afirmaciones I y III son correctas.
b) Solo las afirmaciones II, III y IV son correctas.
c) Solo las afirmaciones I y II son correctas.
d) Todas las afirmaciones son correctas.

10. En cuanto a los requisitos no funcionales:

a) Siempre es posible identificar los componentes que implementan un requisito no funcional específico.
b) Los requisitos no funcionales pueden afectar a la arquitectura general del sistema en lugar de a componentes individuales.
c) Un único requisito no funcional puede generar varios requisitos no funcionales, pero no puede generar requisitos funcionales.
d) Los requisitos no funcionales definen servicios requeridos para el sistema.

11. Las pruebas de concepto son

a) prototipos que implementan completamente una capa de la arquitectura
b) a), por lo que permiten resolver incertidumbres sobre la factibilidad de la arquitectura propuesta u otros riesgos técnicos.
c) prototipos verticales, que implementan una porción funcional de la aplicación, atravesando todas las capas o niveles de la arquitectura.
d) prototipos que son evolutivos y sirven para responder preguntas, resolver incertidumbres y mejorar la calidad de los requisitos.

12. Respecto a la relación de inclusión de un caso de uso (caso incluido) en otros casos de uso (casos base):

    a) Permite extraer cierta parte de la funcionalidad y referenciarla desde uno o más casos base, para evitar repetirla en más de un lugar, o para distinguir cierto comportamiento y hacerlo más entendible.
    b) Solo se pueden incluir otros casos de uso en el flujo principal de un caso de uso base, pero no en sus flujos alternativos.
    c) Una vez que se termina la ejecución del caso de uso incluido se continúa la ejecución del caso base, tanto si el caso incluido se ejecutó con éxito como si no.
    d) El caso de uso incluido puede indicar a qué paso del flujo principal del caso base (que lo incluye) debe volverse.

13. La gestión de un proyecto de software

    a) siempre se realiza de la misma forma en los distintos proyectos de software.
    b) puede variar dependiendo de múltiples factores, entre los que se cuentan el tamaño de la empresa, las características del cliente, el tamaño del software, el tipo del software, la cultura organizacional, el proceso de desarrollo de software
    c) b), y, en esos casos, todas las actividades varían de un proyecto a otro.
    d) b), pero aún en esos casos hay actividades imprescindibles, propias de la gestión de proyectos, tales como planificación, gestión de recursos humanos, gestión de riesgos, seguimiento y control, etc.

14. Respecto al enfoque de valor ganado:

    a) Si el CPI (*cost performance index* o índice de desempeño del costo, calculado como EV/AC (valor ganado / costo actual)) < 1, el proyecto está costando más de los planificado
    b) a) y el proyecto está atrasado.
    c) Si CPI < 1, el proyecto está costando menos de lo planificado
    d) c) y, si SPI (*schedule performance index* o índice de desempeño del cronograma, calculado como EV/PV (valor ganado / valor planificado)) < 1, el proyecto está atrasado.

15. Si, una vez armado el cronograma de un proyecto, se necesita acortarlo,

    a) Siempre es posible aplicar la técnica de *fast-tracking* paralelizando la ejecución de actividades que tenía planificado ejecutar de forma secuencial.
    b) Siempre es posible aplicar la técnica de *fast-tracking* asignando más recursos a algunas actividades.
    c) Si se puede aplicar la técnica de *fast-tracking*, esta producirá siempre alguna reducción en la duración del proyecto y acortará (en mayor o menor medida) el cronograma.
    d) Puede suceder que utilice la técnica de *fast-tracking* y que la duración del proyecto sea incluso mayor a la que estaba planificada originalmente.

16. Su organización está trabajando en el desarrollo de plataforma para el transporte de carga, que se considera clave para el Ministerio de Trabajo y Obras Públicas. Algunos de los componentes a desarrollarse se integran con sistemas externos, entre ellos, el sistema de balanzas, la interacción con el cual es compleja y solo uno de los programadores conoce bien. Existe el riesgo de que, si esta persona falta por alguna razón, esta interacción no se pueda implementar debidamente y el proyecto se retrase. Se decide no hacer nada por ahora, y, si la persona llegara a faltar, ahí se verá qué hacer. Esto es, según el enfoque del PMBoK visto en el curso,

    a) una estrategia de evitar el riesgo
    b) una estrategia de transferir el riesgo
    c) una estrategia de aceptación del riesgo
    d) un plan de contingencia.

17. Seleccione la respuesta correcta

    a) El incremento, de los últimos años, de la complejidad de los sistemas de software que se construyen es un factor más que influye en los defectos (severidad, cantidad, etc.) que los desarrolladores introducen en el software durante el desarrollo. Por eso, para los proyectos más complejos de software de la actualidad se prefiere el modelo en cascada. Esto es para evitar el desconocimiento de cómo voy a desarrollar el sistema.
    b) (a) y no existen al día de hoy procesos de desarrollo de software genéricos, en el sentido que aplican a cualquier contexto y proyecto, debido a que, entre otras cosas, diferentes tipos de software pueden requerir diferentes procesos de desarrollo de software.
    c) (a) y actualmente existen procesos de desarrollo de software generales (que aplican a cualquier contexto y proyecto) debido al avance y mejoras en los últimos años de los métodos ágiles.
    d) a), b) y c) son INCORRECTAS.

18. Una empresa es contratada para desarrollar un cierto producto de software. Luego de trabajar en recolectar al menos una idea general de los requisitos, los líderes del proyecto de desarrollo de software estiman el tamaño del proyecto y el esfuerzo. Dado el tamaño del equipo de desarrollo estiman que el proyecto para la construcción del producto de software pedido llevará alrededor de un año y medio de trabajo. La empresa contratante, expresa que no tiene todos los requisitos claros y que le gustaría abordar en etapas la construcción del software si esto fuera posible. También le gustaría contar con liberaciones parciales del producto para ir viendo cómo funciona y poder pedir cambios si fuera necesario.

    a) Dado que el trabajo va a durar un año y medio, se debería utilizar un método en espiral, ya que al durar ese tiempo van a haber muchos cambios en los requisitos durante el desarrollo.
    b) Parece razonable abordar este proyecto con algún proceso iterativo e incremental con liberaciones parciales al cliente.
    c) Se agrega la siguiente información al contexto presentado en el enunciado de la pregunta: “El equipo de desarrollo tiene experiencia en proyectos llevados adelante con Scrum.” Con esta nueva información queda claro que para este proyecto lo indicado es utilizar Scrum.
    d) Dos de las opciones anteriores son correctas.

19. Seleccione la afirmación correcta:
    a) Los procesos iterativos e incrementales deberían liberar incrementos (versiones) al cliente al menos una vez por mes.
    b) Los roles, en la especificación de un proceso de desarrollo, normalmente definen las habilidades y las responsabilidades que deben poder llevar adelante las personas de los equipos de desarrollo. Los roles definen también cómo se hace cada actividad del proceso de desarrollo de software.
    c) El modelo CMMI, por sus características, puede ser utilizado para la mejora de procesos de desarrollo de software (incluso procesos ágiles).
    d) El desarrollo con Scrum debe ser siempre con la práctica “cliente en el lugar”. De otra forma, no se está haciendo Scrum sino una variante de Scrum.

20. Seleccione la afirmación correcta:

    a) Cuando se usan procesos iterativos e incrementales en un proyecto, no se realizan liberaciones parciales al cliente. Sin embargo, estos procesos promueven las liberaciones internas (para algunos incrementos) para que el equipo de desarrollo pueda realizar pruebas de sistema.
    b) Scrum define el rol de Scrum Master, entre otros roles.
    c) b) y Scrum propone revisar en cada *Sprint* la forma en la cual el equipo de desarrollo trabajó durante el *Sprint* y reflexionar sobre cómo se podría haber trabajado mejor.
    d) c) y los procesos ágiles se centran en la arquitectura de software. En las primeras iteraciones se define y en las siguientes la arquitectura es refinada. Esto, por ejemplo, en Scrum se llama refinamiento sucesivo.

21. Sobre la construcción de software, seleccione la afirmación correcta:

    a) La planificación de la construcción incluye, entre otras cosas, cómo van a ser construidas e integradas las componentes de software.
    b) (a) y la construcción es una etapa del desarrollo de software que normalmente no se realiza en la práctica profesional (es decir, en la industria de software) ya que se pasa directo a la programación luego del diseño.
    c) La construcción efectiva y eficiente que se ha logrado mostrar a través de los últimos años y que sigue vigente es mediante el uso del lenguaje de programación Java.
    d) a) y el reuso es una actividad que comprende, entre otras cosas, utilizar (reutilizar) software ya existente como parte del desarrollo de software que se está realizando.