

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 una hora y media.
- El puntaje total del examen es 100 puntos.

1. Seleccione la afirmación correcta:

- a) En los últimos años se ha visto un incremento en el desarrollo de software monolítico, de soporte general, y paulatino (es decir, con incrementos parciales) y, casi siempre, en espiral de las organizaciones.
- b) Actualmente, debido al impulso del desarrollo ágil, la mantenibilidad del software es un aspecto de calidad que se podría clasificar como de "importancia menor". Esto se debe a que en las distintas iteraciones o *sprints* se ajustan los problemas de mantenibilidad.
- c) En la actualidad, las pruebas unitarias deben cumplir con el 100% del cubrimiento de decisiones en el código. Una empresa de desarrollo de software que entregue software que no cumpla con dicho criterio puede verse envuelta en demandas y juicios.
- d) (a), (b) y (c) son falsas.

2. Seleccione la afirmación correcta:

- a) Un proceso de desarrollo de software puede estar compuesto, entre otras cosas, por roles que desempeñan el equipo de software, el cliente y otros involucrados (*stakeholders*)
- b) (a) y siempre los procesos tienen actividades que debe desempeñar la gerencia de la empresa de desarrollo para garantizar que el negocio siga adelante.
- c) (a) y los métodos ágiles permiten el diseño arquitectónico lento, conllevando esto a un menor retrabajo.
- d) Los procesos ágiles no incluyen planes ni se centran en la arquitectura de software, por eso mismo son ágiles.

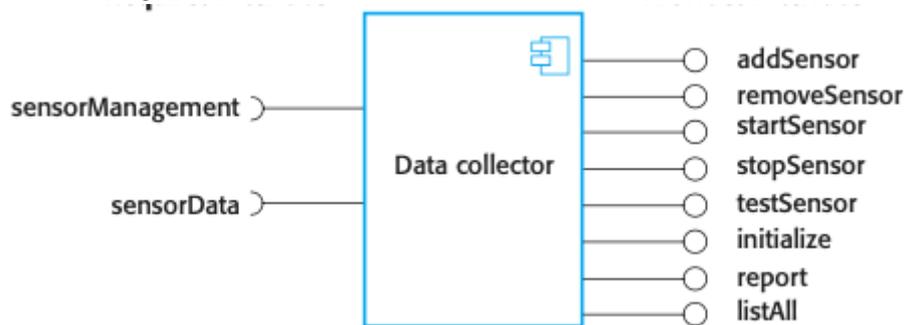
3. Seleccione la afirmación correcta:

- a) En el modelo de procesos "Integración y configuración", se integran componentes preexistentes. Los procesos ágiles pueden incluir actividades de gestión entre las actividades propuestas y tienen que contar con un Scrum master. Los procesos de desarrollo de software pueden modificarse.
- b) Un problema del modelo en cascada es la dificultad de responder a cambios pedidos por el cliente. Los métodos ágiles, generalmente, buscan gestionar de forma efectiva los cambios solicitados por clientes o usuarios.
- c) El modelo de proceso en espiral considera en sus primeras iteraciones del proceso la verificación automatizada. Con esto busca conseguir productos de calidad y altamente verificados.
- d) Cuando se utiliza un modelo de proceso incremental la arquitectura de software tiende a mejorar sin tener riesgos de degradación que afecten la estructura del sistema.

4. Seleccione la afirmación correcta:
- a) La mejora de procesos de desarrollo de software busca, por ejemplo, mejorar las prácticas utilizadas, quitar alguna práctica o agregar alguna otra.
 - b) (a) Además, las mejoras propuestas intentan concentrar el trabajo al comienzo del proceso ya que esto mejora la calidad del software. Por ejemplo, una mejora en este sentido es entender todo lo que el usuario necesita y tener una especificación de requisitos completa al comienzo del proyecto.
 - c) (a) Además, para realizar mejoras no es necesario tener mediciones ya que la percepción del equipo de desarrollo es ampliamente suficiente.
 - d) La mejora de procesos debe ser guiada por casos de uso.
5. Sobre la construcción de software.
- a) El reuso no tiene nada que ver con la construcción de software ya que el reuso debe ser considerado durante el diseño.
 - b) La construcción efectiva es aquella que es guiada por métodos ágiles.
 - c) En la ingeniería de software se denomina "Construcción segura" a la práctica de utilizar un cubrimiento de código de decisión para todas las componentes construidas.
 - d) La forma en la cual se lleva adelante la construcción de software depende, en parte, del proceso de desarrollo de software utilizado.
6. Seleccione la opción **incorrecta**:
- a) En la práctica las actividades de la ingeniería de requisitos se intercalan iterativamente
 - b) La falta de precisión en los requisitos es la causa de muchos problemas en la ingeniería de software
 - c) Las metodologías ágiles abrazan el cambio en los requisitos
 - d) (c) Lo mismo sucede en las metodologías tradicionales
7. Sobre las historias de usuario y los criterios de aceptación, seleccione la opción correcta.
- a) Las historias de usuario complejas se subdividen en épicas
 - b) Una historia de usuario está incompleta si no se formulan los criterios de aceptación
 - c) (b) que sería bueno que cubrieran tanto los casos comunes como los alternativos
 - d) La historias de usuario son formuladas por el equipo de desarrollo
8. Dadas las siguientes características deseables de un conjunto de requisitos, selecciona la opción **incorrecta**:
- a) Es Completo si no hay necesidades ausentes
 - b) Es Consistente si no hay requisitos en conflicto
 - c) Es Modificable si se pueden incluir nuevos requisitos
 - d) Es Trazable si admite ser rastreado hacia atrás a su origen y hacia adelante a elementos de diseño, código y verificación.
9. Sobre la evolución del software en las metodologías ágiles, seleccione la opción **incorrecta**.
- a) No hay diferencia con lo que sucede en las metodologías tradicionales
 - b) Los cambios pueden ser expresados como historias de usuario adicionales
 - c) La transición del desarrollo a la evolución es sencilla porque aplican un enfoque iterativo incremental
 - d) La evolución se puede entender como una simple continuación del proceso de desarrollo basado en liberaciones frecuentes del sistema.

10. Seleccione la opción correcta.
- La liberación de un producto de software consiste, entre otras cosas, en su despliegue para ser usado por el público objetivo
 - (a) y siempre Incluye también una encuesta de satisfacción para evaluar la calidad del producto
 - (a) e incluye la instalación y configuración del sistema, su adopción o conversión, así como el entrenamiento y apoyo para su uso
 - (a) y no se necesitaría realizar nada más durante la liberación, pero se sugiere tener un plan de contingencia
11. Sobre la arquitectura de software:
- Es importante porque afecta principalmente: los requerimientos funcionales.
 - Es un proceso automatizado que difiere entre un sistema y otro, incluso en sistemas con requerimientos similares.
 - Es importante porque afecta principalmente: la performance, robustez, distribuibilidad y mantenibilidad de un sistema.
 - Se documenta con diagramas de procesos de UML.
12. Indique la afirmación correcta sobre sistemas distribuidos:
- Pueden compartir recursos, por ejemplo archivos.
 - Pueden escalar cuando aumenta la demanda.
 - Pueden brindar servicios al usuario incluso si parte del sistema falla.
 - a) b) y c) son correctas.

13. Dada la siguiente imagen:



- sensorData* es una interfaz que provee el componente *Data collector* y *report* es una interfaz requerida por el componente *Data collector*.
- sensorData* es un componente requerido por el servicio *Data collector* y *report* es un componente que provee el servicio *Data collector*.
- sensorData* es una interfaz requerida por el componente *Data collector* y *report* es una interfaz que provee el componente *Data collector*.
- sensorData* es un componente que provee el servicio *Data collector* y *report* es un componente requerido por el servicio *Data collector*.

14. Para probar unitariamente la siguiente función en C++:

```
double calculate_price
(double baseprice, double specialprice,
 double extraprice, int extras, double discount)
{
    double addon_discount;
    double result;

    if (extras >= 3) addon_discount = 10;
    else if (extras >= 5) addon_discount = 15;
    else addon_discount = 0;
    if (discount > addon_discount)
        addon_discount = discount;

    result = baseprice/100.0*(100-discount)
    + specialprice
    + extraprice/100.0*(100-addon_discount);
    return result;
}
```

- La estrategia Top-down es la adecuada.
- Se puede utilizar una clase que llame a la función `calculate_price()` adecuadamente y registre su resultado. Esa clase sería un \rightarrow *driver*.
- La estrategia Bottom-up es la adecuada.
- Se puede utilizar una clase que llame a la función `calculate_price()` adecuadamente y registre su resultado. Esa clase sería un \rightarrow *stub*.

15. Considere los siguientes casos de prueba para la función `calculate_price()` (del ejercicio 4)

```
// testcase 01
price = calculate_price(10000.00,2000.00,1000.00,3,0);

// testcase 02
price = calculate_price(25500.00,3450.00,6000.00,6,0);

// testcase 03
price = calculate_price(10000.00,2000.00,1000.00,0,10);

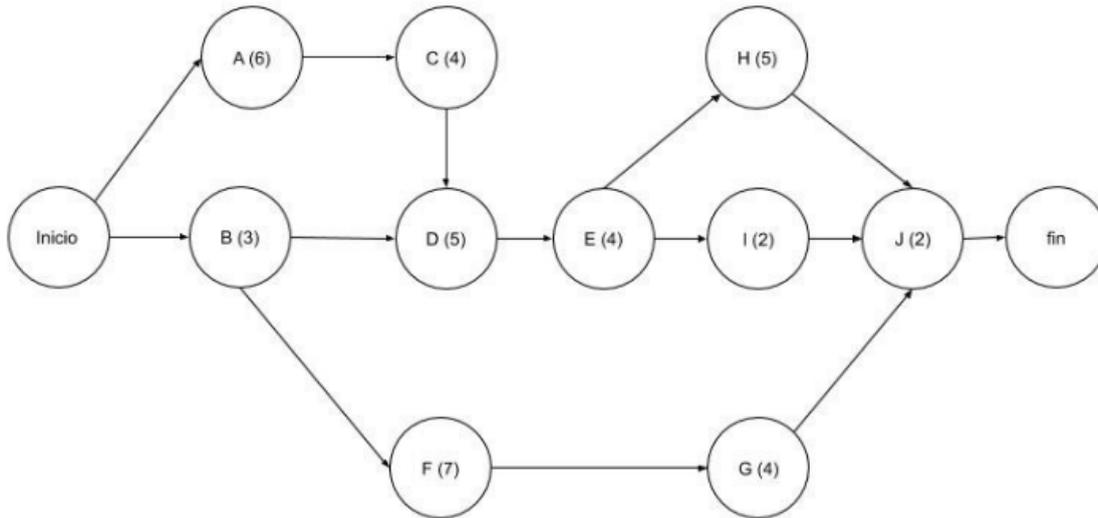
// testcase 04
price = calculate_price(25500.00,3450.00,6000.00,6,15);
```

- 4 y 1 generan el mismo cubrimiento de sentencias
- 3 y 4 generan el mismo cubrimiento de sentencias
- 2 y 3 generan el mismo cubrimiento de sentencias
- 1 y 2 generan el mismo cubrimiento de sentencias

16. Dado el siguiente requerimiento: "Para precios menores a \$15.000, no hay descuentos. Precios mayores a \$15.000 y hasta \$20.000, un 5% de descuento. Mayores a \$20.000 y hasta \$25.000, el descuento es 7%. Mayores a \$25.000, el descuento es 8,5%." qué representantes considera mejor para incluir como datos de prueba:

- 14500, 16500, 24750, 31800
- "f", MIN_INT, -123, -1, 0, 654, MAX_INT
- "f", MIN_INT, -1, 0, 1, 14999, 14500, 15000, 15001, 16500, 24750, 31800, MAX_INT
- "f", MIN_INT, -1, 0, 1, 14999, 15000, 15001, 24999, 25000, 25001, MAX_INT

17. Según Sommerville, en un sistema de control de versiones distribuido:
- los distintos desarrolladores comitean los cambios que realizan en su repositorio local
 - los desarrolladores pueden hacer push de los cambios comiteados al repositorio del proyecto.
 - los desarrolladores informan al responsable de integración que hay cambios prontos y este hace pull de los cambios desde el repositorio del proyecto.
 - todas las opciones anteriores son posibles.
18. Como gerente del proyecto, acaba de presentarle al cliente el siguiente cronograma de desarrollo, en el que la duración de las actividades se indica en días hábiles.



Luego de ver el cronograma de desarrollo, el cliente no está conforme con la fecha de entrega y solicita que el proyecto se termine 15 días hábiles antes, ya que quiere presentar el nuevo producto en un evento muy importante. Para evaluar la factibilidad de cumplir con la fecha solicitada, el equipo revisa los recursos y actividades a realizar y concluye lo siguiente:

- No es posible sumar un nuevo integrante al proyecto.
- Algunos integrantes del equipo podrían realizar horas extras.
- Revisando las tareas y el grafo de precedencias, presentado a continuación, el equipo resuelve que es viable que las tareas A y D se realicen de forma paralela, es decir, que se podrían obviar o eliminar las precedencias de A y C sobre D.

Considerando las siguientes afirmaciones:

- Si el cliente acepta que parte del equipo realice horas extras, entonces se podría, en principio analizar aplicar la técnica de crashing para acortar el cronograma.
- La cantidad de horas extras necesarias para poder terminar 15 días hábiles antes el proyecto es 120.
- Aplicar la técnica de fast-tracking para acortar el cronograma puede generar un incremento en el costo del proyecto.
- La holgura del camino crítico va a ser 15 días menor.
- Las estrategias de crashing y fast-tracking incrementan los riesgos del proyecto.

Seleccione la opción correcta:

- Solo (V) es correcta.
- Solo (II) y (V) son correctas.
- Solo (I), (III) y (V) son correctas.
- Solo (III) y (IV) son correctas.

19. En cuanto a las estimaciones de esfuerzo:
- a) Es importante tener en cuenta los datos históricos de la organización, que reflejan experiencias pasadas, si se cuenta con ellos.
 - b) El tipo de software a construir no es un factor que influya en la estimación de un proyecto.
 - c) Los únicos factores que influyen en ellas son el tamaño del producto a construir, el tipo de software y factores del personal.
 - d) Las estimaciones basadas en algoritmos, como, por ejemplo, COCOMO, deben calibrarse con la historia organizacional.
20. En un proyecto que se gestiona con Scrum, los ítems que están en el *Product Backlog*,
- a) deben estar todos especificados con el mismo nivel de detalle.
 - b) su funcionalidad se valida con los usuarios.
 - c) una forma de priorizarlas es según su valor de negocio.
 - d) la responsabilidad de priorizarlas es del equipo de desarrollo.
21. Dada las siguientes afirmaciones, señale la correcta:
- a) Algunos de los factores críticos que influyen en la relación entre el director de proyectos y el equipo son: inclusión, respetar la diferencia, honestidad y fomentar el aumento de salarios.
 - b) Respecto a conflictos, el trabajo del director de proyectos es evitarlos siempre y en caso de que ocurran, se debe seleccionar la técnica de gestión de conflictos según el caso.
 - c) Bass y Dunteman presentan 3 tipos de personalidades: orientados a las tareas, orientados a sí mismos y orientados a la interacción.
 - d) Es conveniente que el director del proyecto sea una persona orientada a la interacción.