

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. 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.
- 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) Los problemas éticos que pueden surgir en el desarrollo de software (por el tipo de trabajo de un ingeniero de software) son menores en comparación a los que surgen en el resto de las ingenierías.

2. 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.

- a) Dado que el trabajo va a durar un año y medio, se debería utilizar un método ágil, ya que al durar ese tiempo van a haber muchos cambios en los requisitos durante el desarrollo.
- b) Los datos que nos da el enunciado de esta pregunta son insuficientes para poder indicar qué proceso de desarrollo de software sería más adecuado utilizar.
- 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 es adecuado utilizar Scrum.
- d) Dos de las opciones anteriores son correctas.

3. 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. Buscan definir quién (con qué rol) hace qué (actividades, tareas, etc.).
- c) b) y el desarrollo iterativo e incremental propone que cada incremento alcanzado (que surge de cada iteración) sea entregado al cliente para que pueda comenzar a utilizar el incremento construido y detectar posibles cambios o problemas.
- d) Los métodos ágiles se basan fuertemente en el concepto de velocidad. Este concepto indica que el desarrollo de software no debe ser postergado o suspendido por actividades externas al equipo; es decir, mantener la velocidad siempre. Se busca registrar esta velocidad para luego poder planificar las iteraciones o *Sprints*.

4. Seleccione la afirmación correcta:
- El modelo CMMI, por sus características, puede ser utilizado para la mejora de procesos de desarrollo de software (incluso procesos ágiles).
 - (a) y las actividades de un proceso de desarrollo de software, de una empresa particular, deberían reflejar la mejor forma (que la empresa conoce) para desarrollar sus productos. Por eso, los procesos de las empresas deben seguirse de forma disciplinada y no realizar cambios durante el transcurso de un proyecto.
 - (a) y en Scrum, las retrospectivas del *sprint* (*sprint retrospective*) sirven para mejorar los procesos, herramientas, etc. que el equipo de desarrollo de software esté utilizando.
 - 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.
5. De acuerdo a las siguientes afirmaciones respecto de la información relevante a tener en cuenta para los casos de prueba:
- Además de los datos de entrada, es importante tener en cuenta el contexto previo del sistema (precondiciones).
 - Además de las salidas esperadas del sistema, es importante tener en cuenta el contexto del sistema luego de la ejecución del caso de prueba (post-condiciones).
 - No es necesario tener en cuenta los parámetros de las interfaces, los valores y estados internos del sistema, así como otras condiciones secundarias.
 - Los eventos del sistema, siempre deben formar parte de las precondiciones o de las post-condiciones.

Seleccione la opción correcta:

- Sólo las afirmaciones I y II y III son correctas
 - Sólo las afirmaciones I y II son correctas
 - Sólo las afirmaciones II, III y IV son correctas
 - Todas las afirmaciones son correctas.
6. Dado el siguiente fragmento de programa
- ```
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 se pregunta: ¿Cuál de estos criterios de cubrimiento de caja blanca es el **menos fino** que asegura que siempre se va a provocar una falla al ejecutar una división entre cero?
- Criterio de cubrimiento de sentencias
  - Criterio de partición en clases de equivalencia
  - Criterio de cubrimiento de condición múltiple
  - Ninguna de las opciones anteriores es correcta

7. Un equipo de desarrollo de 5 personas realiza un software a medida para un cliente local. El cliente no tiene apuro en poner en producción el producto, pero quiere estar presente durante el proyecto, evaluando y validando pequeños incrementos del producto. Solo cuando el software esté completo y validado por el cliente será puesto en producción. En este contexto:
- Solo será necesario generar pruebas de regresión luego de que la primera versión del software sea puesta en producción.
  - Las pruebas de regresión se deben generar al comienzo del proyecto ya que éstas tienen sentido únicamente a nivel de componentes y es deseable que estén automatizadas.
  - Las pruebas de regresión son pruebas únicamente funcionales y pueden referir a cualquier nivel de prueba (unitario, de integración o de sistema).
  - Ninguna de las opciones anteriores es correcta.**
8. Una de las actividades de la gestión de la configuración es la gestión del cambio. Sobre la gestión del cambio, indique la opción **INCORRECTA**.
- En ciertos contextos es posible omitir el comité de control de cambios. El mismo está integrado (generalmente) por representantes del cliente, los usuarios y los desarrolladores, cuya función es evaluar y autorizar o rechazar las solicitudes de cambios al sistema, ya sean mejoras, nuevas funcionalidades o correcciones de defectos.
  - En algunos casos, la decisión sobre la implementación de un cambio podría ser tomada directamente por el desarrollador.
  - Durante el mantenimiento, los cambios se deben introducir de forma que se intente asegurar la continuidad del correcto funcionamiento del software en producción y del sistema que este conforma (hardware en el que corre, las personas que lo utilizan, etc). Esto implica intentar asegurar que el cambio es pertinente y que su introducción no va a afectar de forma negativa el funcionamiento del sistema.
  - El costo de implementación de un cambio incluye el retrabajo asociado a ese cambio, y los pedidos de cambios comienzan a surgir cuando el cliente experimenta una primera versión del sistema.**
9. La arquitectura cliente-servidor:
- de dos niveles con clientes ligeros, es aplicable a una realidad donde se requiere cómputo intensivo del lado del cliente, como por ejemplo, de visualización de datos.
  - de dos niveles con clientes gruesos, es aplicable a una realidad donde el cliente debe ser capaz de computar lógica de negocio en modo "off-line" y sincronizar la información una vez recupere la conexión con el servidor.
  - multinivel es una arquitectura donde las diferentes capas (por ejemplo, presentación, lógica, acceso a datos) son procesos separados y pueden ejecutarse en diferentes procesadores.
  - con un único servidor es siempre recomendable para entornos donde la alta disponibilidad es crítica.
- Solo las afirmaciones II, III y IV son correctas.
  - Solo las afirmaciones II y III son correctas.**
  - Solo la afirmación II es correcta.
  - Solo las afirmaciones I, II y III son correctas.

10. Acerca de las vistas arquitectónicas del modelo 4 + 1, ¿qué opción u opciones no son correctas?:

- i) Para la vista de desarrollo se suele utilizar un diagrama de componentes
- ii) Para la vista física se suele utilizar un diagrama de despliegue (deploy)
- iii) Para la vista de casos de uso/escenarios se suele utilizar un diagrama de casos de uso
- iv) Para la vista física se suele utilizar un diagrama de casos de uso

- a. Sólo la afirmación iv
- b. Sólo las afirmaciones i y iii
- c. Sólo las afirmaciones i, iii y iv
- d. Ninguna afirmación es correcta

11. De las siguientes figuras, ¿cuál o cuáles podrían representar un diagrama de despliegue (deploy)?:

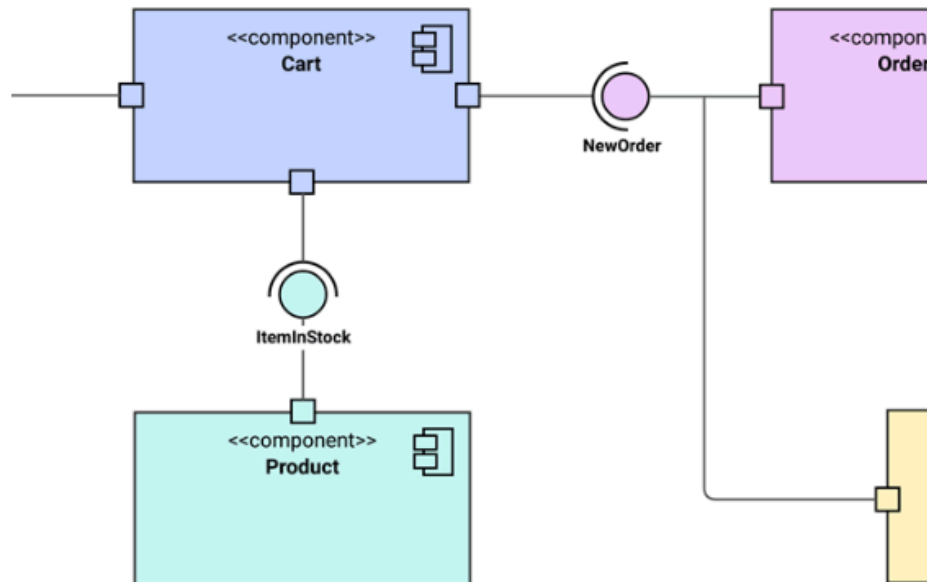


Figura 1

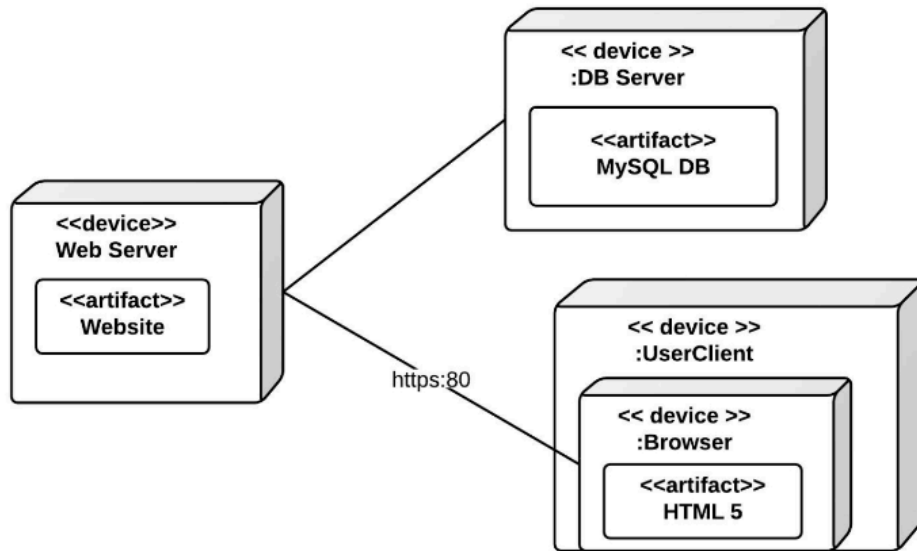


Figura 2

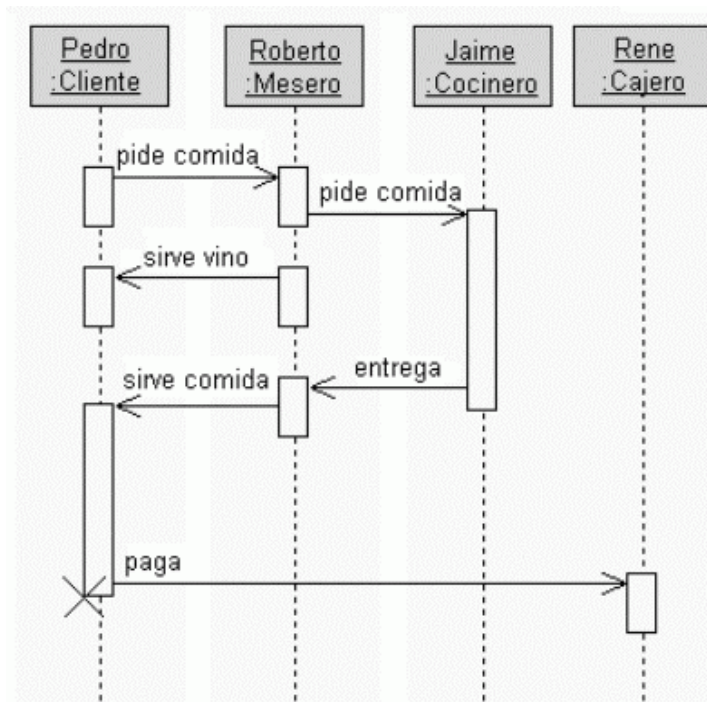


Figura 3

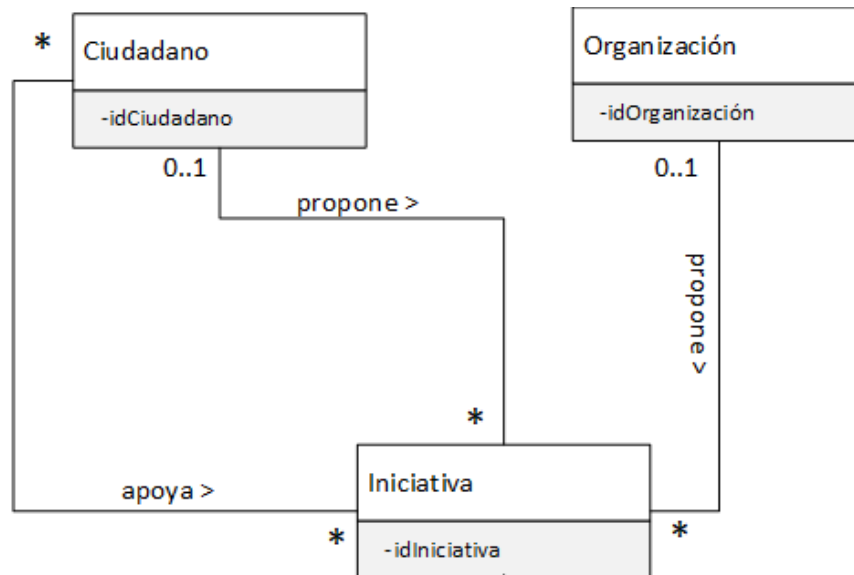


Figura 4

- Sólo Figura 3
- Sólo Figura 2
- Sólo Figura 1 y Figura 4
- Ninguna de las figuras presentadas podría representar un diagrama de despliegue.

12. Indique cuáles de las siguientes afirmaciones sobre la deuda técnica y la refactorización en el desarrollo de software son correctas.

- La refactorización apunta a mejorar un programa para mejorar su estructura, reducir su complejidad o hacerlo más fácil de entender.
- La deuda técnica implica costos adicionales para el futuro, mientras que la refactorización apunta a mejorar la calidad del código existente sin cambiar su comportamiento externo.
- La reducción de la deuda técnica apunta a mejoras en el rendimiento, mientras que la refactorización se enfoca en nuevas funcionalidades.
- La deuda técnica se acumula solo en proyectos pequeños, mientras que la refactorización solo se aplica a proyectos grandes.

- Sólo las afirmaciones i y iii son correctas
- Sólo las afirmaciones ii y iv son correctas
- Sólo las afirmaciones iii y iv son correctas
- Sólo las afirmaciones i y ii son correctas

13. Indique cuáles de las siguientes afirmaciones sobre la reingeniería en el desarrollo de software son correctas.
- i) La reingeniería permite la reestructuración o reescritura de parte o todo un sistema heredado sin cambiar su funcionalidad.
  - ii) El software puede "envejecer". En ese envejecimiento comienzan a ocurrir fallas o problemas varios de forma más frecuente. Este es un momento adecuado para pensar en una reingeniería.
  - iii) En la reingeniería debe participar el cliente, dado que es común que se modifiquen o se agreguen nuevas funcionalidades.
  - iv) La reingeniería carece de riesgos, dado que implica realizar modificaciones de código sin cambiar las funcionalidades.
- a) Sólo las afirmaciones i y iii son correctas
  - b) Sólo las afirmaciones ii y iv son correctas
  - c) **Sólo las afirmaciones i y ii son correctas**
  - d) Sólo las afirmaciones iii y iv son correctas
14. «El procesador de texto a desarrollar deberá proveer un verificador ortográfico.» Esta proposición describe:
- a) un requisito de interfaz externa
  - b) un requisito no funcional
  - c) **un *feature* o característica**
  - d) una regla de negocio
15. «El sistema debe ser fácil de usar». Esta formulación representa:
- a) un requisito del negocio
  - b) un requisito no funcional, que, así como está especificado, ya puede ser probado objetivamente.
  - c) **una intención general del cliente, que hay que registrar.**
  - d) un *feature* (característica) del producto a desarrollar.
16. El nivel de detalle con que se deben especificar los requisitos dependerá de
- a) el involucramiento del cliente
  - b) la dispersión geográfica del equipo de desarrollo
  - c) la precisión necesaria a alcanzar en las estimaciones de tamaño, esfuerzo y cronograma del proyecto
  - d) **todas las anteriores son correctas**
17. Suponga un proyecto que sigue una metodología ágil, para el que el cliente le pide que presente, al comienzo del proyecto, un plan para la primera release. Usted decide utilizar puntos de historia como medida de estimación:
- a) **Se deben estimar todas las historias que conforman la *release* al comienzo del proyecto**
  - b) Solo se estiman las historias que se van a desarrollar en la siguiente iteración
  - c) Una vez estimadas las historias, no se puede cambiar esa estimación, porque alteraría el compromiso para la *release*.
  - d) No es necesario conocer o estimar la velocidad del equipo para elaborar el plan de la *release*.

18. Usted está a cargo de un proyecto de desarrollo de una aplicación para un cliente. Se ha completado el desarrollo y usted quiere planificar la fase de transición. Parte del contrato comprende dar mantenimiento correctivo durante los primeros dos meses de uso. Después, el mantenimiento correrá por cuenta del equipo de informática del cliente. Para continuar con la planificación, se completó la tercera rama del siguiente WBS/EDT, usando el enfoque tradicional de división del trabajo necesario para hacer los entregables.

1. Fase inicial
2. Fase de desarrollo
3. Fase de transición
  - 3.1 Pruebas de aceptación
    - 3.1.1 Instalar en preproducción
    - 3.1.2 Correr pruebas de aceptación
  - 3.2 Migración de datos
    - 3.2.1 Preparar scripts para migración
    - 3.2.2 Correr scripts
    - 3.2.3 Verificar corrección de datos migrados
  - 3.3 Preparar documentación
  - 3.4 Capacitación
    - 3.4.1 Preparar capacitación
    - 3.4.2 Brindar capacitación
  - 3.5 Instalar en producción
  - 3.6 Puesta en producción
  - 3.7 Mantenimiento correctivo urgente

Su equipo estimó el esfuerzo necesario para cumplir con las tareas e identificó las dependencias entre ellas. El resultado se muestra en la siguiente tabla.

| Tarea                                        | Estimación<br>(semanas-persona) | Precedencias      |
|----------------------------------------------|---------------------------------|-------------------|
| 3.1.1 Instalar en preproducción              | 1                               | No tiene          |
| 3.1.2 Correr pruebas de aceptación           | 2                               | 3.1.1             |
| 3.2.1 Preparar scripts para migración        | 6                               | No tiene          |
| 3.2.2 Correr scripts                         | 2                               | 3.2.1, 3.5        |
| 3.2.3 Verificar corrección de datos migrados | 1                               | 3.2.2             |
| 3.3 Preparar documentación                   | 6                               | No tiene          |
| 3.4.1 Preparar capacitación                  | 4                               | No tiene          |
| 3.4.2 Brindar capacitación                   | 3                               | 3.3, 3.4.1        |
| 3.5 Instalar en producción                   | 1                               | 3.1.2             |
| 3.6 Puesta en producción                     | 1                               | 3.2.3, 3.4.2, 3.5 |
| 3.7 Mantenimiento correctivo urgente         | 8                               | 3.6               |

Señale la opción correcta:

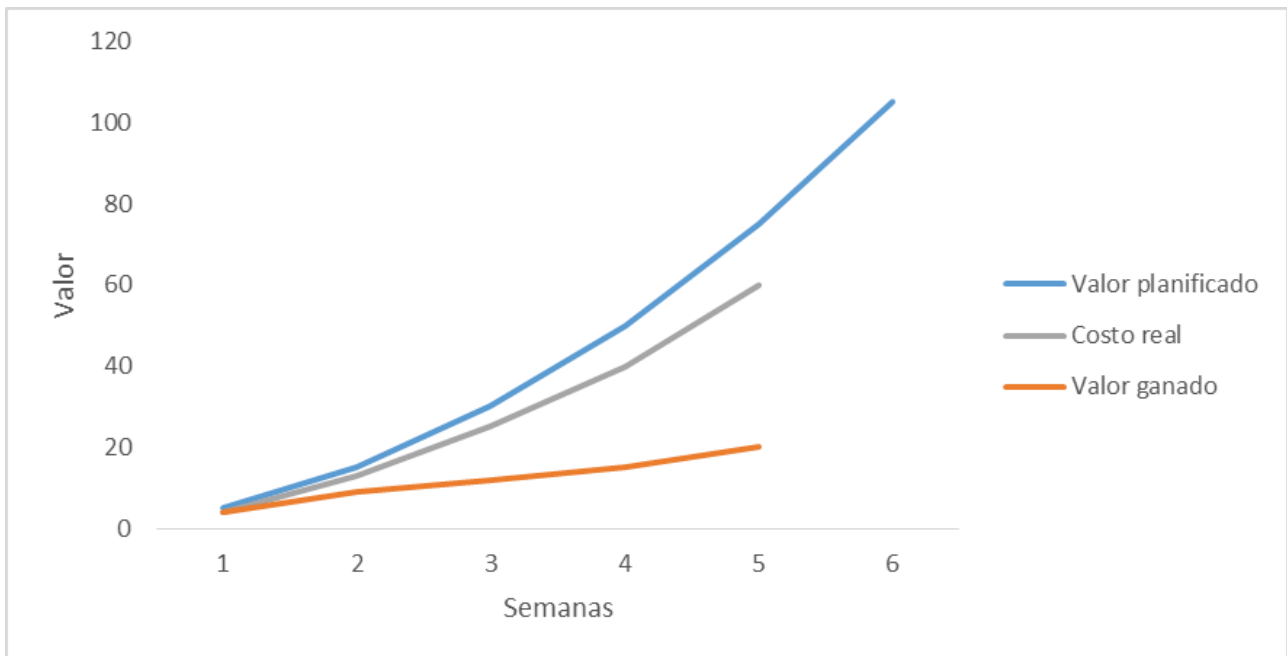
- a) Hay un solo camino crítico que dura 18 semanas.
- b) Hay dos caminos críticos que duran 12 y 18 semanas respectivamente.
- c) Hay dos caminos críticos, que se gestionan ambos como críticos.
- d) Hay dos caminos de igual duración, pero solo uno se toma como crítico.



19. Su organización está trabajando en el desarrollo de una plataforma de software para el transporte de carga, que se considera clave para el Ministerio. 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 que, si la persona llegara a faltar, se renegociará el plazo de entrega con el cliente. 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 aceptar el riesgo
- d) un plan de contingencia

20. Usted es el director de un proyecto cuya duración está estimada en 6 meses. Ha decidido medir el avance, utilizando la técnica de valor ganado, al final de cada mes. Se encuentra al final del quinto mes. La gráfica resultante es la que se muestra a continuación, donde el trazo superior (en azul) es el del valor planificado, el trazo inferior (en naranja) es el del valor ganado) y el trazo del medio (en gris) es el del costo real.



Este escenario indica que el proyecto está:

- a) Adelantado y con ahorros
- b) Adelantado, pero con sobrecostos
- c) Con ahorros, pero atrasado
- d) Atrasado y con sobrecostos

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

- a) La planificación de la construcción incluye, entre otras cosas, cómo van a ser construidos e integrados los componentes de software.
- b) (a) y en la industria de software no se aplica la construcción de software (según define esta área el SWEBOK) ya que se pasa directo a la programación luego del diseño.
- c) (a) y según el SWEBOK acerca del reuso en construcción de software: el reuso es una actividad que comprende utilizar (reutilizar) software ya existente como parte del desarrollo de software que se está realizando y también comprende el construir software que pueda ser luego reutilizado.
- d) La construcción y la programación son lo mismo. En particular, el SWEBOK define a la construcción de software como la actividad de programar.