

Instrucciones

- Cada pregunta múltiple opción contestada correctamente tiene un valor de 3,34 puntos.
- Cada pregunta incorrecta de la múltiple opción resta 1,11 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 2 horas.
- **El puntaje total del examen es 100 puntos y se aprueba con 60 o más puntos.**

Múltiple Opción

1. ¿Cuál de las siguientes afirmaciones es correcta sobre la Ingeniería de Software?
- a) Está dedicada a apoyar el desarrollo de software profesional más que la programación individual.
 - b) (a) y técnicas para soportar la evolución no son habitualmente relevantes para el desarrollo individual.
 - c) **(b) y el software no es solo un programa, sino también su documentación, bibliotecas, datos, entre otros.**
 - d) (c) y comprende los aspectos de la denominada ingeniería de sistemas, ej. la construcción de hardware.

La opción d) es incorrecta porque la ingeniería de software es parte del proceso más general provisto por la ingeniería de sistemas y no al revés.

2. El ciclo de mejora de procesos implica:
- a) Medir el proceso de software.
 - b) (a) y medir el producto de software.
 - c) (b) y analizar dichas mediciones y reflexionar sobre ellas
 - d) **(c) y proponer cambios para mejorar las mediciones e implantarlos.**

El ciclo de mejora de procesos involucra medir tanto el proceso como el producto y no basta solamente con analizar y reflexionar sobre las medidas obtenidas, es necesario proponer cambios e implantarlos.

3. Respecto de los cambios en el proceso de desarrollo de software:
- I. Hoy en día son prácticamente inevitables y debemos gestionarlos adecuadamente cuando ocurren.
 - II. Los cambios son mucho más frecuentes cuando la metodología de desarrollo es orientada a planes que cuando es ágil.
 - III. Los cambios solamente ocurren cuando el software se encuentra en la etapa de evolución.
 - IV. Algunas estrategias para hacer frente al cambio son: prototipación, arquitecturas SOA y entregas incrementales.
 - V. Algunas ventajas de las entregas incrementales es que las entregas tempranas a su vez funcionan como “prototipos” que ayudan a relevar y validar requisitos para los próximos incrementos.

Seleccione la opción correcta:

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

La afirmación (II) es incorrecta porque una metodología orientada a planes a priori no debería fomentar un aumento en los cambios del sistema en comparación con una metodología ágil. Incluso una metodología ágil podría llegar a promoverlos. La afirmación (III) es incorrecta porque los cambios ocurren durante todo el ciclo de vida del software, no solamente cuando éste está en la etapa de evolución. Además, la afirmación (IV) es incorrecta porque las arquitecturas SOA no se mencionan como estrategias para hacer frente al cambio.

4. ¿Qué significa que un equipo sea auto-gestionado?
- I. Que no tiene líder.
 - II. Que los integrantes del equipo son responsables y se comprometen a cumplir con los objetivos planteados.
 - III. Que el equipo no da cuentas a la gerencia y es independiente.
 - IV. Que la distribución y asignación de tareas se realiza de forma interna dentro del equipo.
 - V. Que el equipo asume la responsabilidad por los fracasos (no cumplimiento de objetivos) de forma colectiva y no individual.

Seleccione la opción correcta:

- a) Solo las opciones II y IV son correctas.
- b) Solo las opciones I y V son correctas.
- c) **Solo las opciones II, IV y V son correctas.**
- d) Solo las opciones III y IV son correctas.

La opción (I) es incorrecta porque los equipos auto-gestionados sí pueden tener un líder, que oficie de facilitador y que ayude en la gestión del equipo (más que realizarla únicamente él). La opción (III) es incorrecta porque los equipos auto-gestionados dan cuenta a la gerencia como cualquier otro equipo, no por ser auto-gestionados están eximidos de esta responsabilidad.

5. Una organización acaba de concretar un proyecto con un nuevo cliente. El proyecto tiene las siguientes características: un representante del cliente estará íntegramente dedicado al proyecto, se conoce el objetivo general pero las funcionalidades y requisitos específicos aún están por descubrirse, la organización dispone de un equipo interdisciplinario de cuatro personas que ya han trabajado juntas y, por último, se debe entregar un incremento productivo para validación por la directiva dentro de tres meses.

Indique cuál es el proceso de desarrollo más adecuado bajo esas circunstancias:

- Un proceso en cascada con liberaciones internas parciales.
- Dadas todas las certezas identificadas, lo aconsejable es utilizar un proceso basado en planes
- Lo conveniente es utilizar un proceso ágil, por ejemplo, aplicando Scrum.
- Lo conveniente es utilizar un proceso ágil, por ejemplo, Cascada.

Las características mencionadas del proyecto en cuestión son las típicas para aplicar un proceso ágil.

6. MUM es un modelo basado en planes que tiene las siguientes características:
- Es un proceso iterativo e incremental, orientado a la arquitectura y con 4 fases.
 - Es un proceso orientado a la arquitectura y con las siguientes fases: Inicial, Construcción y Transición.
 - Es un proceso iterativo e incremental y con las siguientes fases: Requisitos, Diseño e Implementación.
 - No está orientado a la arquitectura y tiene 3 fases: inicial, construcción y cierre.

MUM es un modelo orientado a planes, interactivo e incremental, orientado a la arquitectura y con 4 fases (Inicial, Elaboración, Construcción, Transición).

7. Respecto a las técnicas para obtener requisitos:
- Las entrevistas son útiles para entender el problema de negocio y el ambiente de operación.
 - Las entrevistas son útiles para entender los requisitos de dominio.
 - La técnica de Focus Groups son útiles para explorar actitudes, impresiones, preferencias y necesidades de los usuarios.
 - Los cuestionarios (encuestas) son útiles para estudiar a grandes grupos de usuarios con el objetivo de entender sus necesidades.
 - No es necesario que los prototipos evolutivos sean construidos con calidad en el código y robustez desde el principio, si no que el objetivo es que se vayan mejorando en el transcurso del desarrollo.

Seleccione la opción correcta:

- Solo las afirmaciones I, II y V son correctas.
- Solo las afirmaciones I, III y IV son correctas.
- Solo las afirmaciones II, III y IV son correctas.
- Solo las afirmaciones I, IV y V son correctas.

La afirmación (II) es incorrecta porque las entrevistas no son buenas para entender los requisitos de dominio: los ingenieros de requisitos no pueden entender la terminología específica del dominio; para algunas personas algunos de los conocimientos del dominio son tan familiares que les resulta difícil explicarlos o piensan que no vale la pena. Además, la afirmación (V) es incorrecta porque los prototipos evolutivos proveen una base arquitectónica sólida para desarrollar el producto de forma incremental mientras los requisitos se vuelven más claros con el tiempo; deben ser construidos con calidad en el código y robustez desde el principio; además deben ser diseñados para contemplar un rápido crecimiento y una mejora frecuente; son una buena elección para aplicaciones que sabemos que crecerán a lo largo del tiempo (por ejemplo, sitios web).

8. En relación a los requisitos:
- Los requisitos funcionales describen cómo el sistema debería reaccionar a entradas particulares y cómo debería comportarse en situaciones particulares. Además pueden describir lo que el sistema no debe hacer.
 - Un atributo de calidad es un tipo de requisito no funcional que describe una característica de servicio o desempeño de un producto.
 - Los requisitos no son completos cuando existen conflictos o definiciones contradictorias entre ellos.
 - Un ejemplo de requisito no funcional verificable es: La aplicación tiene que ser amigable y fácil de usar.
 - Un único requisito no funcional puede generar varios requisitos funcionales que definen servicios requeridos para el sistema.

Seleccione la opción correcta:

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

La afirmación (III) es incorrecta porque completo se refiere a si todos los requisitos requeridos por el usuario están definidos, en tanto consistente se refiere a si existen conflictos o definiciones contradictorias entre ellos. Además, la afirmación (IV) es incorrecta porque para que sea verificable debería de presentar alguna medida que pueda ser probada objetivamente.

9. Respecto a los Casos de Uso:

- I. Son una técnica de relevamiento y modelado de requisitos.
- II. Describen la interacción del sistema con otros actores (se enfoca más que nada en la visión del usuario).
- III. Los casos de uso NO son adecuados para modelar requisitos no funcionales.
- IV. Son un modelo bastante complejo y que en general no se muestra a los usuarios.
- V. En general, son una buena base para construir conjuntos de casos de prueba aunque no contemos aún con el sistema construido.

Seleccione la opción correcta:

- a) Solo hay 2 afirmaciones correctas.
- b) Solo hay 3 afirmaciones correctas.
- c) Sólo hay 4 afirmaciones correctas.
- d) Todas las afirmaciones son correctas.

La afirmación (IV) es incorrecta porque por definición, los casos de uso son un formato simple y estructurado donde los usuarios y desarrolladores pueden trabajar juntos.

10. ¿Cuál de las siguientes afirmaciones es correcta sobre el proceso de ingeniería de requisitos?

- a) Consiste en la obtención, análisis, especificación, verificación y validación de los requisitos del sistema y termina cuando los requisitos son validados.
- b) Termina cuando el Documento de Especificación de Requisitos del Software está validado por el cliente y en línea base.
- c) Termina cuando se tiene una versión estable del Documento de Especificación de Requisitos del Software.
- d) Ninguna de las afirmaciones es correcta.

Ninguna de las afirmaciones está considerando que el proceso de requisitos se mantiene durante todo el proyecto, incluyendo la gestión de cambios.

11. ¿Cuál de las siguientes afirmaciones es INCORRECTA sobre el modelado de requisitos?

- a) Los modelos de comportamiento reflejan qué sucede o qué debe suceder cuando un sistema responde a ciertos estímulos. Un ejemplo de este modelo son los diagramas de dominio.
- b) Los modelos de contexto reflejan y ayudan a identificar lo que está dentro y fuera del sistema. Un ejemplo de este tipo de modelos es el diagrama de casos de uso.
- c) Los modelos de interacción reflejan la interacción del sistema con los usuarios, otros sistemas o entre componentes. Un ejemplo de este tipo de modelo son los diagramas de secuencia.
- d) Los modelos estructurales reflejan la organización del sistema considerando componentes y relaciones entre los mismos. Un ejemplo de este tipo de modelo es el diagrama de clases.

La opción a) es incorrecta porque el diagrama de dominio es un modelo estructural.

12. Respecto al diseño de software:

- I. El diseño arquitectónico comprende la identificación de los principales componentes del sistema y sus relaciones.
- II. Un patrón arquitectónico es una descripción abstracta de buena práctica, que se ensayó y puso a prueba en diferentes sistemas y entornos.
- III. Algunos aspectos importantes en el diseño son: concurrencia, seguridad y manejo de excepciones y errores.
- IV. El principio de diseño "Encapsulación y ocultamiento de información" significa agrupar y empaquetar los detalles internos de una abstracción y hacer que esos detalles sean inaccesibles para las entidades externas.
- V. Cuanto mayor acoplamiento más difícil resulta comprender y modificar un módulo.

Seleccione la opción correcta:

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

13. Una organización requiere implementar un módulo con las siguientes responsabilidades:

- Debe ser un proceso *batch*.
- Debe ejecutarse diariamente a las 11 pm.
- Debe realizar la conciliación entre todas las facturas pagadas en el día y los movimientos de ingreso de dinero por ese concepto.
- Todas las facturas y movimientos conciliados quedarán en un estado OK.
- Para las diferencias encontradas, se generará una notificación al departamento contable conteniendo el detalle correspondiente.

¿Cuál es el estilo arquitectónico más apropiado?

- a) Modelo-Vista-Controlador es el estilo típicamente adecuado para los procesamientos en lote.
- b) El estilo Pizarrón es el adecuado debido a que existen varios procesos observadores.
- c) **Es tubos y filtros. Es el estilo típicamente utilizado en aplicaciones que procesan información y procesos *batch*.**
- d) El patrón Observer.

Este es el caso que usa el libro para poner un ejemplo de aplicación de tubos y filtros.

14. Para describir la arquitectura de un sistema es posible utilizar un modelo de múltiples vistas. ¿Qué vistas comprende el modelo "4+1"?

- a) Funcional, administrativa, capas, física y del usuario.
- b) Funcional, del arquitecto, comunicación, física y del usuario.
- c) **Lógica, de procesos, física, de desarrollo y escenarios.**
- d) Lógica, de procesos, despliegue, comunicación y del usuario.

El modelo 4+1 presenta una vista Lógica, una vista Física, una de desarrollo y una de casos de uso o de escenarios.

15. ¿Cuál de las siguientes afirmaciones es INCORRECTA sobre los patrones arquitectónicos?

- a) Cliente-servidor multinivel: las diferentes capas del sistema son procesos separados que pueden ejecutarse en diferentes procesadores.
- b) Maestro-esclavo: el proceso "maestro" generalmente es responsable del cálculo, la coordinación y las comunicaciones, y controla los procesos "esclavos".
- c) Componentes distribuidos: cada entidad distribuida es un componente que proporciona servicios a otros componentes y recibe servicios de otros componentes.
- d) **Entre pares: sistema centralizado en un nodo de la red.**

La opción d) es incorrecta porque son sistemas descentralizados, donde los cálculos pueden ser llevados a cabo por cualquier nodo en la red.

16. ¿Cuál de las siguientes afirmaciones es correcta sobre construcción de software?

- a) Los límites entre diseño, construcción y testing están bien definidos.
- b) **Suele producir la mayor cantidad de ítems de configuración a gestionar en un proyecto.**
- c) Minimizar la complejidad y anticipar el cambio son intereses exclusivos de la construcción.
- d) La reutilización se enfoca en aprovechar los componentes construidos dentro del mismo proyecto.

Salvo la opción correcta, el resto de las opciones son incorrectas porque: los límites entre las tres varían dependiendo del proceso utilizado (opción a); ambos intereses son compartidos con el diseño de software (opción c); y la reutilización cruza las fronteras de un proyecto, e incluso puede enfocarse en construir para la reutilización (en un futuro proyecto) y no con reutilización (opción d).

17. ¿Cuál de las siguientes afirmaciones es correcta sobre las pruebas de regresión?

- a) Deben estar siempre automatizadas.
- b) Refieren a los niveles de componente y de integración.
- c) Son necesarias desde el momento en el cual el software está en producción.
- d) **Ninguna de las anteriores es correcta.**

Todas las opciones son incorrectas porque: las pruebas de regresión pueden ser tanto automatizadas como manuales (opción a); refieren a todos los niveles de pruebas, incluidos el nivel de sistema (opción b); su necesidad varía dependiendo de varios factores, pero pueden ser necesarias antes de que el sistema entre en producción, de hecho, son deseables desde el momento en que hay una segunda versión del software (opción c).

18. ¿Cuál de las siguientes afirmaciones es correcta sobre las revisiones de software **informales**?
- Sigue el proceso general de las revisiones de una forma simplificada.
 - Sigue un proceso diferente (etapas diferentes) al de las revisiones generales.
 - (a) y en general se desarrolla como una revisión cruzada entre uno o más colegas.
 - (c) y en algunas ocasiones el revisor es el propio autor del artefacto que se está revisando.

La opción b) es incorrecta porque el proceso es similar. La opción d) es incorrecta porque en una revisión carece de sentido que sea el propio autor del artefacto quien realice la revisión.

19. ¿Cuál de las siguientes afirmaciones es correcta sobre la gestión de las pruebas?
- Es importante priorizar las pruebas, ya que puede pasar que no dé el tiempo de ejecutar todas las planificadas.
 - Si no da el tiempo de ejecutar todas las pruebas planificadas, quiere decir que hubo una mala planificación.
 - La priorización de las pruebas debe realizarse en base al impacto que las fallas pueden tener en el negocio del cliente.
 - Quien planifica las pruebas tiene que tener en cuenta las fechas comprometidas con el cliente más que el estado de avance en el desarrollo del producto.

Salvo la opción correcta, el resto de las opciones son incorrectas porque: no tener tiempo suficiente para ejecutar todas las pruebas puede tener como causa una mala planificación, pero no necesariamente (opción a); la priorización debe realizarse en base a varios diversos factores y no únicamente el impacto en el negocio, por ejemplo, pueden estar en riesgo vidas humanas (opción c); si la planificación tiene más en cuenta los compromisos con el cliente, puede planificar realizar pruebas que no van a poder realizarse si, por ejemplo, el equipo de desarrollo está muy atrasado y no planificar pruebas para lo que sí se está desarrollando (opción d).

20. ¿Cuál de las siguientes afirmaciones es correcta sobre la ejecución de pruebas?
- Permite probar que el producto no tiene defectos.
 - La ausencia de fallas no indica que no haya defectos.
 - Los defectos generalmente están diseminados de forma pareja en el código.
 - Encontrar fallas y reparar defectos asegura que el sistema cumpla con las expectativas y las necesidades de los usuarios.

Véase la sección 2.4 General Principles of Testing del libro para más información.

21. En relación a las pruebas de integración :
- Pueden incluir la ejecución de pruebas unitarias, pero en un entorno integrado.
 - Encuentran defectos únicamente en las interfaces y comunicación entre los componentes integrados.
 - El orden recomendado para realizar pruebas de integración es *top-down*.
 - Pruebas de integración inadecuadas generan un mayor esfuerzo y retrabajo al momento de realizar pruebas a nivel de sistema.
 - Si es posible, es recomendable automatizarlas.

Seleccione la opción correcta:

- Solo las afirmaciones I, IV y V son correctas.
- Solo las afirmaciones III y V son correctas.
- Solo las afirmaciones I y II son correctas.
- Todas las opciones son correctas.

La afirmación (II) es incorrecta porque las pruebas de integración pueden encontrar defectos también a nivel unitario (aunque no sea su objetivo principal). Además, la afirmación (III) es incorrecta porque no existe un orden recomendado para ejecutar las pruebas de integración, depende de diversos factores relacionados al producto y al proyecto.

22. Cuando se sustituye un sistema anterior por uno nuevo:
- Es necesario que el sistema anterior se deje de utilizar de forma paulatina.
 - Es necesario definir si hay que realizar migración de datos (del sistema viejo al nuevo), y en tal caso definir cómo y cuándo se debe realizar.
 - b) y en caso de tener que migrar datos, cuáles datos serían necesarios migrar.**
 - c) y definir como estrategia de contingencia que si el sistema nuevo falla en el acceso a los datos, se deben acceder a los datos del sistema viejo.

La opción a) es incorrecta porque la estrategia de liberación paulatina es una estrategia posible, pero no siempre es lo más indicado y mucho menos lo “necesario”, pueden haber otras estrategias más adecuadas según el caso. Además, la opción d) es incorrecta porque acceder a los datos del sistema viejo como estrategia de contingencia puede no ser apropiada en muchos casos, por ejemplo, cuando el sistema viejo se dejó de usar y por ende los datos a los cuales se está accediendo se encuentran desactualizados.

23. ¿Cuál de las siguientes afirmaciones es correcta sobre el apoyo durante el uso del sistema?
- Debería estar automatizada en su mayor parte (ayuda integrada y *bots*) ya que los recursos humanos son muy costosos.
 - En general, conviene que el soporte y ayuda sea dado siempre por personas, que estén accesibles a los usuarios, en lugar de documentación, ayuda integrada y *bots*.
 - Es aconsejable contar con una herramienta para la gestión de los incidentes.**
 - Ninguna de las anteriores es correcta.

Salvo la opción correcta, el resto de las opciones son incorrectas porque: la automatización por completo del soporte puede dar a lugar que los usuarios abandonen el sistema en caso de no encontrar respuesta a sus preguntas o no estar familiarizado con la interacción con *bots* o leer documentación (opción a). Por otro lado, irse al otro extremo y dar soporte personalizado a cada usuario puede ser muy costoso y no justificarse (opción b). Una combinación de ambos sería lo ideal.

24. En relación a la evolución del software:
- Hoy en día es difícil concebir un software que no sea capaz de evolucionar en el tiempo. La capacidad de evolución del software se ha convertido en una necesidad.
 - Cuando se trabaja con metodologías ágiles, la evolución del software se encuentra inmersa en el propio proceso de desarrollo.
 - Cuando se trabaja con metodologías ágiles, no es conveniente tratar los cambios relacionados a la evolución del software en conjunto con las nuevas funcionalidades.
 - Las pruebas de regresión automatizadas son deseables cuando el software se encuentra en evolución.
 - Si el desarrollo del software se realizó con un enfoque tradicional, no es posible que la evolución posterior del mismo sea abordada con un enfoque ágil.

Seleccione la opción correcta:

- Solo las afirmaciones I, II y IV son correctas.**
- Solo las afirmaciones II y IV son correctas.
- Solo las afirmaciones I, III y IV son correctas.
- Todas las afirmaciones son correctas.

La afirmación (III) es incorrecta porque cuando se trabaja con metodologías ágiles, justamente los cambios relacionados a entregas anteriores se tratan en conjunto con las nuevas funcionalidades a desarrollar para las próximas entregas. La afirmación (V) es incorrecta porque el enfoque abordado en el desarrollo del software no imposibilita que luego en la evolución se aborde otro enfoque. Sí puede tener algunas desventajas, como ser la no existencia de pruebas automatizadas o que el código no esté refactorizado/simplificado, pero no son factores que imposibiliten el cambio de enfoque como tal.

25. En el contexto de la gestión de configuraciones:
- El control de versiones debe asegurar que los cambios realizados por los desarrolladores no interfieren entre sí.
 - El armado del sistema (*build*) implica ensamblar componentes, datos y bibliotecas en un sistema ejecutable.
 - Gestionar un cambio implica evaluar su costo e impacto y tomar la decisión de cómo y cuándo implementarlo.
 - La gestión de entregas implica preparar el software para la liberación de su uso.

Seleccione la opción correcta:

- Solo las afirmaciones I, III y IV son correctas.
- Solo las afirmaciones II y III son correctas.
- Solo las afirmaciones II y IV son correctas.
- Todas las afirmaciones son correctas.**

26. ¿Cuál de las siguientes afirmaciones es correcta sobre la *definition of done* de una historia?
- a) Es un conjunto de características que debe cumplir una historia para que el equipo de desarrollo pueda estimarla e incluirla en un sprint.
 - b) Es un conjunto de características que debe cumplir una historia para que el equipo pueda determinar si ha terminado de trabajar con ella.
 - c) No puede ser parcial, por ejemplo, no podría establecer que se tuviera terminado el back-end, pero no el front-end correspondiente a esa historia.
 - d) Es el conjunto de criterios de aceptación de esa historia.

La opción a) es incorrecta porque es la definición de *definition of ready*. La opción c) es incorrecta porque perfectamente se puede estipular que la tarea se va a considerar completada cuando se termine una parte o se llegue a un hito, por lo que no tiene por qué implementarse la historia completa. La opción d) es incorrecta porque difiere de los criterios de aceptación en que estos determinan, para cada historia, qué es lo necesario para que la historia sea aceptada por el cliente (mira a este), pero el *definition of done* mira a la organización y apunta a qué tiene que hacer esta como parte de las tareas necesarias para lograr tener esa historia implementada en el producto.

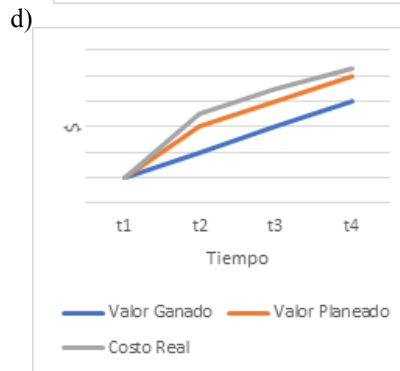
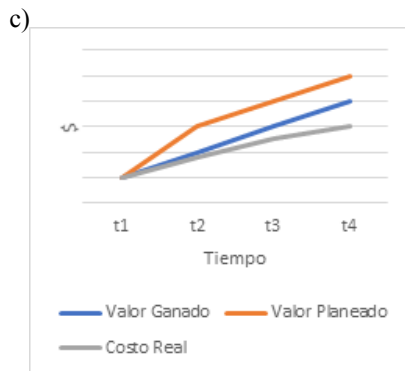
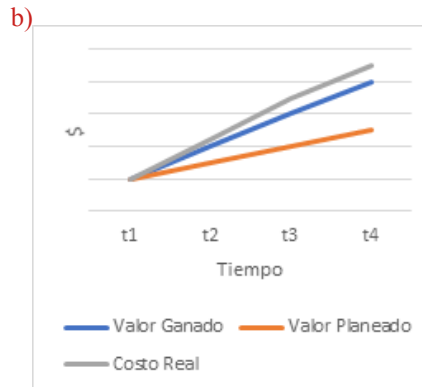
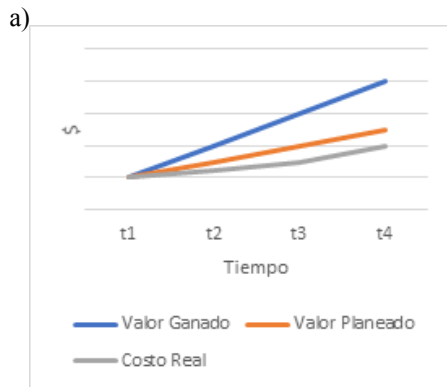
27. Dado un proyecto de software de porte pequeño con un plazo de entrega muy ajustado, llegado dos tercios del proyecto, el equipo se da cuenta de que ha subestimado la complejidad y el esfuerzo de trabajar en la interfaz gráfica, tarea a la que ha asignado una sola persona y está en el camino crítico.
- a) Este problema siempre se soluciona utilizando la técnica de *crashing*.
 - b) Puede que aplicar la técnica de *crashing* extienda el plazo de entrega en lugar de acortarlo.
 - c) Aplicar la técnica de *fast-tracking* siempre es una solución posible.
 - d) Aplicar la técnica de *fast-tracking* no es posible porque la tarea está en el camino crítico.

La opción a) es incorrecta porque la técnica de *crashing* implica agregar recursos, pero esto no siempre soluciona un atraso. En este caso, por ejemplo, quizás sea muy tarde para que la incorporación de otra persona sea eficiente (p. ej. está la curva de aprendizaje, el trabajo necesario para capacitar a la persona, el aumento en la comunicación, etc.). La opción b) es incorrecta porque puede que todo esto haga que el esfuerzo necesario sea mucho y el atraso aumente. La opción c) es incorrecta porque la técnica de *fast-tracking* implica paralelizar actividades que estén en el camino crítico, pero a veces las dependencias son físicas y no se pueden romper. La opción d) es incorrecta porque, por definición, el *fast-tracking* se aplica a tareas en el camino crítico. Paralelizar otras tareas que no estén en dicho camino no afectaría en nada la duración del proyecto.

28. Un proyecto puede definirse como:
- a) Un esfuerzo temporal, que se lleva a cabo para crear un producto, servicio o resultado único.
 - b) a) y un ejemplo clásico de proyecto es la etapa de mantenimiento del producto, etapa que, a su vez, es la que suele llevar más tiempo en la vida útil del software.
 - c) b) y a su vez, la gestión de proyectos se entiende cómo "la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos de este".
 - d) c) y entre sus principales objetivos se encuentra entregar en tiempo, cumplir las expectativas del cliente, entre otros.

La opción b) es incorrecta porque la etapa de mantenimiento de software no se considera como un proyecto, ya que no es un «esfuerzo temporal que crea un resultado único», sino que es entendido como una actividad continua en el tiempo. Esto hace también incorrectas las opciones c) y d).

29. Respecto a la gestión de proyectos, dados los siguientes gráficos donde se relaciona costo (\$) y plazo (tiempo), selección, ¿cuál de las siguientes opciones cumple el siguiente enunciado: «El proyecto está adelantado y está costando más de lo planificado», para el tiempo t4 del gráfico?



Los gráficos muestran: opción a) adelantado y menores costos; opción b) más costoso y adelantado; opción c) atrasado y menores costos; opción d) más costoso y atrasado.

30. La empresa se embarcó en un proyecto de gran visibilidad en el cual uno de los mayores riesgos es la fecha de salida al mercado, la cual es inamovible. El equipo de desarrollo designado cuenta con las personas con mayor experiencia en la empresa y cuenta con el apoyo de gerencia en lo que necesiten para llegar a los plazos. Uno de los requisitos no funcionales establece tiempos de respuesta bastante ambiciosos para la web. ¿Cuál de las siguientes opciones es correcta sobre la gestión del riesgo?

- a) Contratar una empresa especializada en pruebas de performance para que se sume al equipo de pruebas e involucrarla desde el inicio del proyecto es una estrategia de transferir el riesgo de que la aplicación no cumpla con los tiempos establecidos.
- b) Negociar con el cliente el alcance del proyecto para estar más holgados en los tiempos es aplicar la estrategia de evitar el riesgo.
- c) Definir con el equipo y cliente entregas parciales en las cuales se vaya evaluando la performance de la aplicación y validando con el cliente es una estrategia de mitigación.
- d) Dado que la fecha de salida al mercado es inamovible, no puedo hacer nada con el riesgo de no llegar a la fecha y debo aceptarlo.

Las opciones a) y b) son incorrectas porque hablan de mitigar. La opción d) es incorrecta porque se pueden aplicar otras estrategias.