

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.
- **El puntaje total del examen es 100 puntos y se aprueba con 60 o más puntos.**

## Múltiple Opción

1. El proceso de ingeniería de requisitos:
  - a) Tiene 3 actividades principales. Estas son la elicitación y análisis de requisitos, la especificación de requisitos y la validación de requisitos.
  - b) **(a) y estas actividades se pueden realizar de forma iterativa e incremental.**
  - c) Los requisitos de usuario deben establecer qué es lo que se desea que el sistema provea o realice y un detalle de cómo se debe implementar.
  - d) (c) y en los métodos ágiles la especificación de requisitos no se ve como una actividad separada, sino que se ve como parte del desarrollo. Los requisitos son especificados para cada incremento antes de que el incremento sea desarrollado.
2. Con respecto a los procesos de software:
  - I. Al describir un proceso de software es importante especificar qué roles están involucrados, qué se produce y qué condiciones influyen la secuencia de actividades.
  - II. En los procesos basados en planes, las actividades del proceso se planifican por adelantado y el progreso se mide con respecto al plan.
  - III. Los modelos de procesos son una representación simplificada de los procesos de software. Se los puede pensar como *frameworks*, que pueden ser extendidos y adaptados para crear procesos de ingeniería de software más específicos.
  - IV. Para los sistemas de gran porte, por lo general resulta conveniente combinar características de distintos modelos de procesos. También suele ser necesario encontrar un balance entre los procesos basados en planes y los ágiles.
  - V. El desarrollo incremental es un enfoque muy utilizado. Este enfoque surge a partir de la creación de las metodologías ágiles y sólo es utilizado por estas metodologías.

Seleccione la opción correcta:

- a) Todas son correctas.
  - b) Sólo (I), (II), (III) y (V) son correctas.
  - c) Sólo tres opciones son correctas.
  - d) **Sólo (I), (II), (III) y (IV) son correctas.**
3. Con respecto a los métodos ágiles:
    - I. Uno de los objetivos de los métodos ágiles es evitar generar documentación que a la larga seguramente no se use.
    - II. Las metodologías ágiles esperan que los requisitos cambien. Por tal motivo es una buena práctica diseñar el sistema para que se adapte a los cambios.
    - III. Los procesos ágiles no necesitan ser gestionados. Esto es una de las principales diferencias con respecto a los procesos basados en planes.
    - IV. Las historias de usuario se pueden utilizar como base para la planificación de las iteraciones del sistema.
    - V. Scrum indica que debe utilizarse en conjunto con las técnicas de programación por pares y desarrollo guiado por las pruebas.

Seleccione la opción correcta:

- a) Todas son correctas.
  - b) **Sólo (I), (II) y (IV) son correctas.**
  - c) Sólo cuatro opciones son correctas.
  - d) Sólo (II), (IV) y (V) son correctas.
4. Respecto a los procesos de software
    - a) Los procesos de software comprenden: actividades, roles, responsabilidades y productos de trabajo
    - b) **a) y la organización de esas actividades.**
    - c) b) y los procesos tienen como único objetivo mejorar la calidad del software construido.
    - d) c) y la mejora de la calidad incurre en una mejora de los costos en los que se incurre al producirlo.

5. Seleccione la opción correcta:
- a) La reutilización de componentes rara vez se utiliza en los procesos de desarrollo de software.
  - b) Los productos genéricos (COT's) tienen la ventaja de que no necesitan mantenimiento, solamente basta con configurarlos adecuadamente.
  - c) Cuando se utiliza una estrategia de integración y configuración de COT's, ésta puede requerir que los requisitos sean flexibles o que puedan ser re-negociados para que las necesidades del cliente sean satisfechas, pero no de la forma inicialmente prevista.
  - d) Los desarrollos a medida tienen un altísimo costo de parametrización y configuración en relación a la integración y reutilización de COT's.
6. En relación a los tipos de pruebas:
- I. En la técnica de partición en clases de equivalencia, una clase de equivalencia representa un conjunto de valores de datos que se supone deberían comportarse de igual forma. Entonces, se asume que la prueba sobre un elemento de una clase de equivalencia debería mostrar la misma reacción o comportamiento que los demás elementos de la misma clase.
  - II. La técnica análisis de valores límites extiende la técnica de partición en clases de equivalencia, agregando casos de prueba en los bordes de cada clase de equivalencia.
  - III. Las técnicas de caja negra son muy útiles para verificar la estructura interna del código y el diseño del objeto de prueba.
  - IV. Para poder realizar técnicas de caja blanca se necesita contar con el código fuente. El cubrimiento de sentencias es una de las técnicas existentes de caja blanca.
  - V. A la hora de seleccionar qué técnicas y tipos de prueba utilizar para probar, es necesario considerar varios factores como por ejemplo: la disponibilidad de la especificación y la documentación con que se cuenta, el tiempo y presupuesto disponible y la experiencia del equipo de testing.

Seleccione la opción correcta:

- a) Todas son correctas.
  - b) Sólo (I), (II) y (IV) y (V) son correctas.
  - c) Sólo (I), (II), (III) y (IV) son correctas.
  - d) Sólo (II), (III), (IV) y (V) son correctas.
7. En relación a las pruebas de regresión:
- a) Son pruebas que buscan verificar que las nuevas funcionalidades (o cambios en funcionalidades ya existentes) no hayan introducido nuevos defectos en funcionalidades ya probadas (y aceptadas)
  - b) a) y que no hayan degradado la performance del sistema
  - c) b) y son muy útiles en los procesos iterativos e incrementales
  - d) c) y son muy útiles en la evolución y mantenimiento del sistema.
8. En relación a los principios del testing:
- I. El testing no puede demostrar que un programa no tiene defectos.
  - II. Por lo general se recomienda que las actividades de verificación comiencen lo más temprano que se pueda en el ciclo de vida del software. Esto contribuye a que se comiencen a detectar defectos en forma temprana.
  - III. Por lo general los defectos no están distribuidos uniformemente. La mayoría de los defectos se encuentran agrupados en algunos objetos de prueba. Por lo tanto, si muchos defectos son detectados en un lugar, es probable que haya más defectos cercanos a dicho lugar (por ejemplo: en el mismo objeto de prueba o componentes relacionados).
  - IV. Las pruebas deben adaptarse a los riesgos específicos de cada sistema. Por lo tanto, no se deben probar dos sistemas distintos de igual forma. La intensidad de las pruebas así como los criterios de salida de las pruebas deben decidirse de acuerdo a cada sistema de software. Por ejemplo, sistemas críticos para la seguridad requieren diferentes pruebas que las aplicaciones de comercio electrónico.
  - V. La eficacia en la detección de defectos de un mismo conjunto de casos de prueba disminuye con el paso del tiempo. Esto se conoce comúnmente como "la paradoja del pesticida". Por tal motivo, las pruebas tienen que evolucionar y adaptarse para seguir siendo útiles.

Seleccione la opción correcta:

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

9. Sobre los roles y actividades del testing:
- En general conviene que todas las pruebas las realice un equipo externo al equipo de desarrollo.
  - Un futuro usuario del sistema sería un buen candidato para realizar pruebas de aceptación.
  - En los equipos ágiles, los integrantes del equipo realizan pruebas de múltiples niveles.
  - Un experto en el negocio es un buen candidato para realizar pruebas de integración.
  - Que el tester tenga conocimientos de programación no remite ningún valor para las actividades que realiza.

Seleccione la opción correcta:

- Sólo (II), (III) y (IV) son correctas.
- Sólo (II) y (III) son correctas.**
- Sólo (II), (III), (IV) y (V) son correctas.
- Sólo (I), (III) y (IV) son correctas.

---

La empresa *InqSolutions*, que tiene una amplia trayectoria en el uso de Scrum, fue seleccionada para realizar una aplicación que gestione activos y eventos sobre dichos activos. Los integrantes del equipo de desarrollo encargados de realizar el proyecto han venido trabajando juntos desde hace algunos años. Luego de comenzado el proyecto, se define una primera versión del *Product Backlog*.

Respecto a la realidad planteada, conteste las siguientes preguntas:

10. El equipo de desarrollo ha definido un conjunto de características que una historia de usuario debe cumplir para poder comprometerse a su entrega (*Definition of Ready*) y un conjunto de características que una historia de usuario debe cumplir para que el equipo pueda determinar si ha terminado de trabajar con ella (*Definition of Done*). Seleccione la opción correcta:
- El conjunto de características del *Definition of Ready* como del *Definition of Done* son generales y no es necesario adaptarlas al proyecto que se está desarrollando.
  - a) y una característica definida en *Definition of Ready* podría ser: La historia de usuario debe ser INVEST.
  - Una característica definida en *Definition of Done* podría ser: Se cumplen todos los criterios de aceptación.**
  - Todas las opciones son correctas.
11. Dadas las historias de usuario definidas en el *Product Backlog*.
- Todas las historias de usuario que conforman el *Product Backlog* tienen el mismo nivel de detalle.
  - Una historia de usuario del *Product Backlog* puede describir por ejemplo la especificación a bajo nivel de una funcionalidad indicando el flujo de control entre las pantallas.
  - Una forma de priorizar las historias de usuario que conforman el *Product Backlog* es según su valor de negocio, siendo esta priorización responsabilidad del Product Owner.
  - Si la descripción de una historia de usuario es entendida por todos los integrantes del equipo de desarrollo entonces no es necesario incluir los criterios de aceptación para dicha historia de usuario.
  - La calidad de un criterio de aceptación eficaz se puede definir bajo el método SMART.

Seleccione la opción correcta:

- Solo una afirmación es correcta.
  - Sólo (I), (II) y (IV) son correcta.
  - Sólo (II), (III) y (V) son correcta.
  - Sólo (III) y (V) son correctas.**
12. El equipo de desarrollo utilizó la técnica de planning poker para estimar las historias de usuario. Seleccione la opción correcta:
- Es importante que el equipo de desarrollo fije una escala a utilizar antes de comenzar la estimación, como por ejemplo la escala de Fibonacci.
  - a) y uno de los objetivos del Product Owner en el proceso de estimación es aclarar dudas que surjan de las historias de usuario.**
  - b) y además el Product Owner puede aceptar o rechazar la estimación dada por el equipo de desarrollo, ya que es la persona que más conoce del producto a construir y el esfuerzo necesario.
  - Cuando no se llega a un acuerdo respecto a la estimación de una historia de usuario entre los integrantes de un equipo en la primera ronda de estimación, se realiza un promedio entre las estimaciones realizadas para dicha historia de usuario.

13. Dadas las siguientes afirmaciones:

- I. Un cronograma siempre puede acortarse con las técnicas de *fast tracking* o *crashing* y sin incrementar los costos.
- II. La nivelación de recursos consiste en nivelar todos los caminos a la extensión del camino crítico.
- III. La duración del camino crítico va a determinar la fecha de finalización del cronograma y es invariable.
- IV. Luego de tener determinado el EDT ya tengo todo lo necesario para desarrollar el cronograma.
- V. Para desarrollar el cronograma no necesito conocer las dependencias de las actividades.

Seleccione la opción correcta:

- a) Solo las afirmaciones I) y II) son incorrectas.
- b) Solo las afirmaciones II) y V) son incorrectas.
- c) Solo las afirmaciones III) y IV) son incorrectas.
- d) **Todas las afirmaciones son incorrectas.**

14. Sobre la técnica de generación de casos de prueba a partir de casos de uso:

- I. La técnica indica que se deben cubrir todos los escenarios condición al menos una vez.
- II. Se recomienda complementar con valores límites para las condiciones que lo permitan.
- III. En caso de existir bucles en el flujo de control del grafo de escenarios, se recomienda probarlos tantas veces como sea posible.
- IV. La cantidad de escenarios que se pueden derivar de un caso de uso, no depende de cómo esté estructurado (escrito) el mismo.
- V. Sería una técnica válida para la estrategia Test-First-Development.

Seleccione la opción correcta:

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

15. Seleccione la opción correcta:

- a) La teoría de X y la Y presenta dos enfoques de la naturaleza humana. En la X se afirma que a las personas no les gusta el trabajo, que no tienen ambición, etc. En la Y que si son motivados cumplen altas expectativas, son autodisciplinados, etc.
- b) a) y Bass y Dunteman presentan 3 tipos de personalidades: orientados a las tareas, orientados a sí mismos y orientados último a la interacción.
- c) **b) y algunas barreras de comunicación son: personalidad, intereses, suposiciones sobre el receptor/emisor, escucha selectiva.**
- d) c) y si ante un conflicto el director de proyecto dice "Tratemos este asunto la próxima semana", la estrategia de gestión de conflictos aplicada es suavizar/adaptarse.

16. Dadas las siguientes descripciones de estrategias a riesgos y las definiciones de las 8 estrategias vistas en el curso, seleccione la opción correcta:

- a) Subcontratar el trabajo difícil a una compañía más experimentada es mitigar.
- b) a) y dependiendo de la tolerancia al riesgo de cada organización, mientras una puede decidir aceptar un riesgo otra puede decidir mitigarlo.
- c) **b) y durante el monitoreo puede ocurrir que un riesgo varíe su impacto.**
- d) b) y los riesgos los identifica el director de proyectos.

17. Dadas las siguientes afirmaciones:

- I. En los proyectos, solo se planifica antes de comenzar.
- II. En la EDT los componentes del nivel inferior deben ser los necesarios y suficientes para completar el nivel superior.
- III. La EDT debe siempre descomponerse lo mas que se pueda, hasta llegar al paquete mas pequeño independientemente del proyecto en si.
- IV. Los únicos factores que influyen en las estimaciones son: tamaño, tipo de SW y factores del personal.
- V. Las estimaciones basadas en algoritmos, como por ejemplo COCOMO, deben calibrarse a la historia organizacional.

Seleccione la opción correcta:

- a) Sólo la afirmación (II) es correcta.
- b) **Sólo las afirmaciones (II) y (III) son correctas.**
- c) Sólo las afirmaciones (II), (III) y (IV) son correctas.
- d) Todas las afirmaciones son correctas.

18. Sobre los tipos de mantenimiento y sus principales objetivos, algunos de ellos son:
- (I) correctivo: mejorar funcionalidades existentes
  - (II) adaptativo: el sistema se modifica para adaptarse a cambios en el entorno
  - (III) perfectivo: control del funcionamiento diario del sistema a través de la reparación de fallas
  - (IV) preventivo: prevenir que el desempeño del software se degrade
- a) Sólo (I) y (IV) son correctas.
  - b) Sólo (II), (III) y (IV) son correctas.
  - c) **Sólo (II) y (IV) son correctas.**
  - d) Todas son correctas.
19. Sobre el mantenimiento y evolución de software:
- a) **Son relevantes ya que el software requiere continua adaptación, mantenimiento y mejoras para seguir siendo útil y valioso para la organización.**
  - b) La evolución es una etapa en la que el sistema vive en un ciclo donde se intercalan liberaciones parciales y uso operacional, donde no se incorporan nuevas funcionalidades (se pone foco en la corrección y reingeniería necesaria para mantener eficiencia en el uso de los recursos).
  - c) (a) y (b) son correctas.
  - d) Ninguna de las anteriores es correcta.
20. Según el proceso de identificación de cambios y evolución visto en el curso:
- a) La identificación de cambios y la evolución continúa a lo largo del tiempo de vida del sistema.
  - b) a) y dentro del proceso de evolución del software se tiene el proceso de implementación del cambio.
  - c) a) y el análisis de impacto de las peticiones de cambio lo realiza solamente el equipo de implementación del sistema ya que es quien conoce la estructura del programa.
  - d) **b) y durante el proceso de implementación del cambio, se analizan en detalle los requerimientos y pueden surgir modificaciones a los cambios propuestos.**
21. La gestión de versiones es el proceso de realizar el seguimiento de las diferentes versiones de los componentes de software o ítems de configuración y los sistemas de los que forman parte. Acerca de este proceso puede decirse que:
- I. Una línea base permite recrear una versión específica del sistema.
  - II. Se debe garantizar que la introducción de cambios realizados por distintos desarrolladores no genere conflictos.
  - III. Debe disponerse de un mecanismo de gestión de almacenamiento de múltiples versiones.
  - IV. No es necesario registrar los cambios históricos del sistema ya que siempre puede volverse a una línea base a partir de los procesos de merging.
- Seleccione la opción correcta:
- a) Solo las afirmaciones I, y IV) son correctas.
  - b) Solo las afirmaciones III) y IV) son correctas.
  - c) **Solo las afirmaciones I), II), III) son correctas.**
  - d) Todas son correctas.
22. En el área de ética profesional existen algunas cuestiones de responsabilidad profesional más allá del aspecto legal. Sobre estas cuestiones, indique la opción incorrecta.
- a) Confidencialidad - implica respetar la confidencialidad de los empleadores o clientes, aún cuando no haya suscrito un acuerdo de confidencialidad.
  - b) **Competencia - implica aceptar también trabajos que están fuera del nivel de competencia, pues se puede adquirir conocimientos nuevos para ampliar el nivel de competencia.**
  - c) Derechos de propiedad intelectual - implica conocer las leyes locales que rigen el uso de la propiedad intelectual, como patentes, derechos de autor, etc.
  - d) Mal uso de la computadora – implica no usar mal las computadoras de otras personas.
23. Sobre los estudios de factibilidad:
- a) **Tienen como objetivo estudiar si vale la pena implementar el sistema y si es posible implementarlo dadas las restricciones existentes (calendario, presupuesto, tecnología, etc).**
  - b) a) y reciben como entradas: los requisitos del negocio preliminares, una breve descripción del sistema y el diseño completo del sistema.
  - c) b) y brindan como salida un informe que recomienda si vale la pena o no realizar el proceso de desarrollo del sistema.
  - d) Ninguna de las anteriores es correcta.

24. Sobre los casos de uso indique la opción incorrecta.
- Es posible modelar todos los requisitos utilizando esta técnica, lo que sucede es que se requiere mucho esfuerzo.
  - Los casos de uso no tienen por qué desarrollarse por completo de una vez.
  - No se recomienda incluir información de diseño o desarrollo, ni tampoco sobre la definición de tipos de datos.
  - No se recomienda incluir en los flujos detalles de la interfaz gráfica.
25. Según Wieggers cuales de las siguientes son características deseables de un requisito:
- Correcto — describe de forma precisa una necesidad y de forma clara la funcionalidad a desarrollar para cumplirla.
  - Implementado — está implementado en el software construido.
  - Necesario — tiene cierto valor para el negocio.
  - Priorizado — fue priorizado según su importancia utilizando la perspectiva de todos los interesados.
  - Perfecto — no admite múltiples interpretaciones y contiene toda la información para entenderlo.
  - Verificable — se pueden realizar pruebas para determinar si se encuentra presente en un producto de software o no.
- Seleccione la opción correcta:
- I, II, III, IV y VI son correctas.
  - I, III, IV, V, VI son correctas.
  - I, III, IV y VI son correctas.
  - Todas son las afirmaciones son correctas.
26. Sobre construcción de software:
- No es solo codificación, también implica, por ejemplo, pruebas unitarias y debugging.
  - Se busca minimizar la complejidad, aplicando técnicas de programación defensiva ante cambios.
  - Las pruebas del sistema no forman parte de la construcción.
  - a) y c) son correctas.
27. Sobre el software como servicio (SaaS).
- SAAS implica alojar el software de forma remota y proporcionar acceso a él a través de Internet.
  - El software es propiedad de un proveedor de software, en lugar de las organizaciones que lo utilizan.
  - Los usuarios pueden, por ejemplo, pagar por el software de acuerdo con la cantidad de uso que hacen de él o mediante una suscripción anual o mensual.
  - Todas las anteriores son correctas.
28. Sobre el patrón *Componentes Distribuidos* para sistemas distribuidos.
- Cada entidad distribuible es un componente que proporciona servicios a otros componentes y recibe servicios de otros componentes.
  - La comunicación de componentes se realiza a través de un sistema de Internet.
  - Es una arquitectura muy poco abierta ya que no permite agregar nuevos recursos según sea necesario.
  - Ninguna de las anteriores es correcta.
29. Sobre la Ingeniería de Software Basada en Componentes (CBSE):
- Se basa en la reutilización de entidades.
  - a), y contar con un estándar facilita la integración de componentes.
  - b), y es un proceso de software que está orientado a la reutilización.
  - Ninguna de las anteriores es correcta.
30. Sobre el diseño arquitectónico:
- Es una etapa temprana del proceso de diseño del sistema.
  - a) y representa la asociación entre la especificación y el diseño de las pruebas.
  - Comprende, principalmente, en considerar los requisitos funcionales para la identificación de los componentes del sistema y definir sus relaciones.
  - Todas las opciones son correctas.