

- Cada pregunta **múltiple opción** contestada correctamente tiene un valor de 2 puntos. Cada pregunta incorrecta de la múltiple opción resta 0,66 puntos.
- **El puntaje total del parcial es 50 puntos.**

Múltiple Opción

1. Con respecto a las métricas de software:
 - a) El objetivo a largo plazo es utilizarlas para evaluar la calidad de un producto de software.
 - b) (a) y se puede definir como una característica de un producto, documento o proceso de software que puede ser medida objetivamente.
 - c) (b) y por ejemplo el número de líneas es una métrica.
 - d) (c) y el número de personas/horas requeridos para construir un componente de software no es una métrica.

2. Con respecto al mantenimiento del software:
 - a) La duración del mantenimiento es típicamente inferior al tiempo dedicado al desarrollo.
 - b) El mantenimiento es menor en los sistemas legados ya que se posee un conocimiento grande del sistema.
 - c) El esfuerzo de hacer mantenimiento perfectivo es mayor que el dedicado al correctivo.
 - d) Con el paso del tiempo los programas se vuelven mas fáciles de entender y modificar.

3. El rejuvenecimiento de software
 - a) Tiene por objetivo mejorar los atributos de calidad de un producto de software.
 - b) (a), en general, se aplica sobre sistemas legados.
 - c) (b), y una forma de rejuvenecimiento está dada por la reingeniería que consiste en generar documentación del diseño a partir del código fuente.
 - d) Ninguna de las anteriores.

4. El desempeño en la ejecución de un proceso es una medida de los resultados reales obtenidos durante su ejecución:
 - a) El desempeño es el mismo cada vez que se ejecuta el proceso.
 - b) El desempeño y la capacidad de un proceso son concepto antagónicos.
 - c) El desempeño suele ser diferente cada vez que se ejecuta el proceso.
 - d) c) y se desea controlar y predecir su desempeño.

5. Dadas las siguientes afirmaciones sobre Verificación y Validación:
 - I. Las inspecciones y revisiones de software analizan y verifican los requerimientos del sistema, modelos de diseño, código fuente del programa e incluso las pruebas propuestas para el sistema.
 - II. Dos posibles estrategias para elegir casos de prueba de unidad son las pruebas basadas en lineamientos y las pruebas de partición.
 - III. Las pruebas del sistema incluyen la integración de los diferentes componentes del mismo y la puesta a prueba del sistema integrado.
 - IV. Uno de los beneficios más importantes del desarrollo dirigido por pruebas es que reduce los costos de las pruebas de regresión.

Seleccione la opción correcta:

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

6. ¿Qué items contiene un caso de prueba?
 - I. Datos de entrada.
 - II. Condiciones de ejecución.
 - III. Escenarios alternativos.
 - IV. Responsable de prueba.
 - V. Si es de caja negra o caja blanca.

Seleccione la opción correcta:

 - a) I, V, IV.

- b) I, II, III, IV, V.
- c) I, II, III.
- d) Ninguna de las anteriores.

7. Las empresas se proponen evaluar y mejorar sus procesos para:

- a) elevar la calidad, reducir costos o acelerar el tiempo en la construcción de productos de software.
- b) obtener una certificación internacional solamente.
- c) b) así como la satisfacción del cliente.
- d) Todas las anteriores.

8. La liberación del sistema comprende varias actividades que deben ser debidamente planificadas. Algunas de ellas son el entrenamiento, la documentación, la estrategia de conversión y la instalación. Sobre la estrategia de conversión, puede decirse que:

- a) no es una actividad que corresponda a la liberación del sistema ya que solo aplica en el mantenimiento de un sistema en ocasiones en que hay que realizar una conversión de datos.
- b) se tiene en cuenta cuando se debe sustituir un sistema anterior por uno nuevo y solamente incluye la carga inicial de los datos básicos para el funcionamiento del sistema.
- c) la estrategia más apropiada para realizar la conversión es siempre la top-down.
- d) debe tener que elegirse la más adecuada según la realidad. Bigbang, conversión paulatina y procesamiento en paralelo son estrategias de conversión.

9. En un proceso de desarrollo de software:

- a) La verificación es imprescindible, no así la validación.
- b) La validación es imprescindible, no así la verificación.
- c) La verificación y validación evalúan los mismos aspectos del software, solo que desde diferentes puntos de vista.
- d) Ninguna de las anteriores es correcta.

10. Respecto a las pruebas de software

- a) La validación del sistema tiene como principal objetivo asegurar que se ha construido correctamente el producto.
- b) La validación del sistema tiene como principal objetivo asegurar que se ha construido el producto correcto.
- c) Las pruebas de aceptación se negocian con el cliente a partir de los criterios de aceptación de las pruebas unitarias, de integración y de sistema.
- d) Ninguna de las anteriores.

11. Sobre estrategias de verificación de integración:

- I. Big-Bang es recomendable para proyectos de mediano y gran porte ya que permite una gran paralelización del trabajo.
- II. Cuando se utiliza la estrategia Bottom-Up se requiere de Drivers para probar los módulos.
- III. Para las técnicas Bottom-Up y Top-Down no es tan importante la planificación, ya que cada módulo puede ser probado por separado utilizando Drivers y/o Stubs.
- IV. Top-down permite hacer demostraciones tempranas del producto.

Seleccione la opción correcta:

- a) II.
- b) I, III y IV.
- c) I y IV.
- d) II y IV.

12. En cuanto a la liberación de un sistema:

- a) Puede comenzar a realizarse esta actividad mientras aún se está verificando el software, para ganar tiempo.
- b) Luego de liberado el sistema todos los recursos humanos involucrados en su desarrollo pueden reasignarse a otros proyectos.
- c) Se debe tener en cuenta los distintos tipos de usuarios que el sistema pueda tener, para entrenarlos y explicarles como utilizar el sistema.
- d) Todas las afirmaciones son correctas.

-
13. El modelo de Collier, DeMarco y Fearey para el análisis post-mortem de un proyecto propone:
- Recopilar información objetiva del proyecto sin comprometer su confidencialidad.
 - a) e identificar los responsables de las dificultades detectadas.
 - a) y mantener reuniones para completar la información faltante y reflexionar sobre todos los datos disponibles.
 - c) y dejar pasar por lo menos seis meses desde la culminación del proyecto.

-
14. Para el tema verificación y validación, dado el siguiente fragmento de código y afirmaciones, seleccione la opción correcta:

```
If (a > 0) and (b = 0) and (c != a) {  
  x = x / a * c;  
}
```

```
If (a = 4) or (x > 1) {  
  x = x*c;  
}
```

- El caso de prueba $a=3, b=0, c=-1, x=4$ cumple el criterio de cubrimiento de sentencias.
 - Los casos de prueba $a=3, b=0, c=-1, x=4$ y $a=3, b=0, c=0, x=2$ cumplen el criterio de cubrimiento de decisión.
 - Los casos de prueba $a=4, b=0, c=1, x=4$ y $a=-3, b=1, c=3, x=0$ cumplen el criterio de cubrimiento de condición.
 - El caso de prueba $a=4, b=0, c=1, x=4$ cumple el criterio de condición múltiple.
-
15. Para el tema liberación y mantenimiento, seleccione la opción correcta:
- La planificación de la liberación debe realizarse luego de terminar el desarrollo y no antes.
 - En general, los costos de un proyecto no se ven afectados si la liberación está incluido en el alcance del proyecto o no.
 - Si una solicitud de cambio es realizada por el interesado principal del sistema, esta no debe analizarse, debe realizarse y liberarse de inmediato.
 - En general es mas costoso incorporar una funcionalidad luego de que el sistema está liberado respecto a hacerlo en la etapa de desarrollo.
-
16. Sobre el mantenimiento de un sistema:
- No es necesario realizar mantenimiento alguno si el sistema fue construido y verificado correctamente.
 - Su costo puede verse afectado por la estabilidad que tenga en cuanto a personal, el equipo encargado de realizarlo.
 - Conviene que quienes realicen esta tarea no hayan participado en el desarrollo del sistema.
 - Es una tarea que requiere menos dedicación con el paso del tiempo.

-
17. La liberación de un sistema:

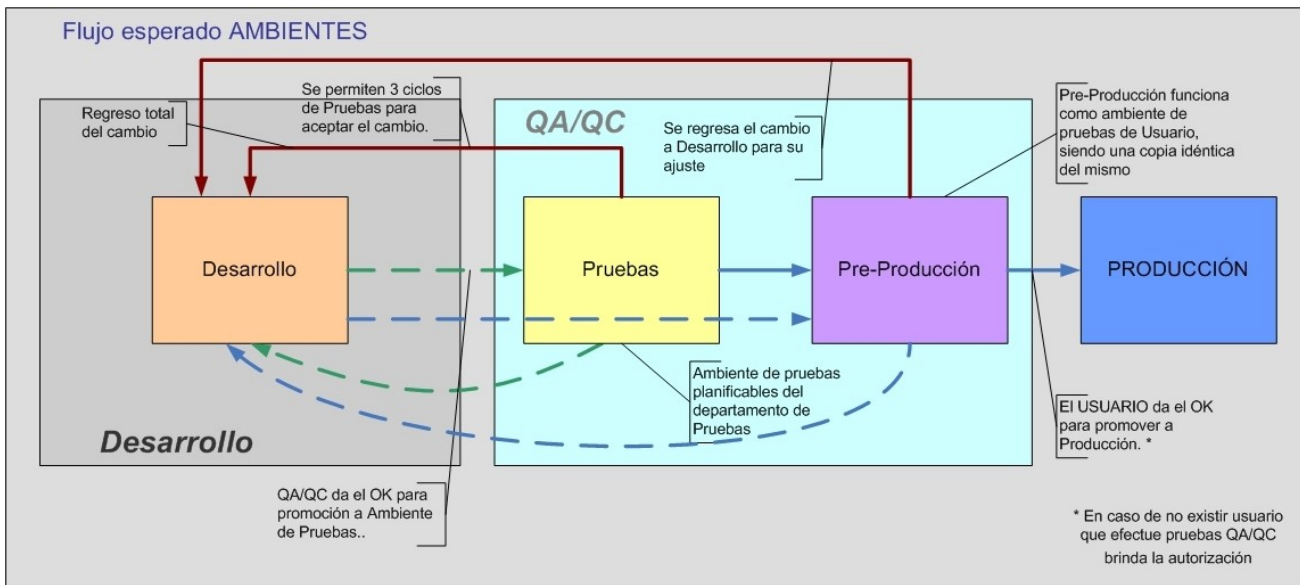
- Su complejidad depende en cierta medida de la tecnología utilizada.
- Es responsabilidad exclusiva de los usuarios del sistema llevar a cabo esta tarea.
- No es necesario planificarla ya que se llevará a cabo cuando el sistema supere las pruebas de aceptación.
- Todas las afirmaciones son falsas.

-
18. En el contexto de Verificación de Software, en cuanto a técnicas estáticas vs técnicas dinámicas:

- Ambas sirven tanto para verificar como para validar.
 - Ambas sirven para verificar cualquier producto (código, documento de requerimientos, casos de prueba, etc.).
 - La ejecución de pruebas estáticas favorece la detección temprana de defectos.
 - Todas las afirmaciones son verdaderas.
-

19. Es conveniente finalizar la verificación de un sistema cuando:
- Se hayan encontrado todos los errores.
 - El cliente solicita la liberación del sistema.
 - Todos los casos de prueba ejecuten sin error.
 - Ninguna afirmación es verdadera.
-
20. Sobre los estándares:
- Son multipropósito y existen en diversas disciplinas.
 - Muchas veces se exige su cumplimiento como requisito de un producto de software.
 - Guían en la elaboración de un determinado producto, en relación a buenas prácticas y a un lenguaje común (por ejemplo).
 - Todas las anteriores.
-
21. Las métricas de productos son utilizadas para medir atributos internos de un sistema de software. Acerca de estas métricas puede decirse que:
- todas las métricas son estáticas, ya que son tomadas a partir de representaciones del sistema, tales como el diseño, el código o la documentación. Un ejemplo es la cantidad de líneas de código no comentado.
 - las métricas dinámicas ayudan a evaluar la eficiencia y confiabilidad de un programa, por ejemplo.
 - todas las métricas son dinámicas, ya que son recolectadas a partir de medidas tomadas cuando el programa está en ejecución. Un ejemplo es el número de errores reportados.
 - la complejidad ciclomática es una medida del promedio de los largos de los id
-
22. Indique la opción correcta sobre la disciplina gestión de la configuración:
- Se ocupa de revisar los cambios e incluir los mismos en la línea base cuando corresponda.
 - Se ocupa de las políticas, procesos y herramientas para la gestión de cambios.
 - (b) y se encarga de versionar los defectos del software.
 - (a) y al controlar un cambio el encargado de la gestión de configuración deberá determinar si es un cambio válido, para luego enviar el cambio al comité de control de cambios y en caso de ser aceptado realizar los cambios pertinentes. En caso que no sea aceptado se rechaza el cambio.
-
23. Se conoce como "pruebas de caja negra":
- Pruebas de sistema automatizadas.
 - Todas las pruebas realizadas por el equipo de testing.
 - Las técnicas estáticas de verificación.
 - Ninguna de las anteriores.
-
24. Algunas de las preguntas que pueden formularse para evaluar las características de un proceso son
- ¿El proceso está definido y se basa en un proceso de referencia estándar?
 - ¿El proceso es flexible y medible?
 - ¿Existen evidencias de la aplicación del proceso? ¿Cuán visible es?
 - ¿El proceso está debidamente comunicado a todos los involucrados en su ejecución?
- Seleccione la opción correcta:
- Solo las preguntas: I) y II) IV) son correctas.
 - Solo la pregunta II) es correcta.
 - Solo las preguntas II) y III) son correctas.
 - Todas las afirmaciones son correctas.
-

25. Indique la opción correcta en relación a la siguiente imagen:



- a) La imagen muestra un esquema de ambientes recomendado para la gestión del software desde las etapas de desarrollo y mantenimiento hasta su puesta en producción.
- b) (a) y en cualquiera de estos ambientes es posible modificar el software por parte de los programadores.
- c) (a) y existen distintas políticas de acuerdo al ambiente particular. En ambiente de desarrollo por ejemplo existen menos libertades de realizar modificaciones al software que en el ambiente de producción.
- d) (a) y la gestión de la configuración recomienda definir políticas de forma explícita para cada ambiente, especificando, por ejemplo, si se van a registrar los incidentes que sucedan durante el uso del software.