

- Cada pregunta de la **parte 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) Una organización dedicada al mantenimiento de software registra información asociada a la actividad de mantenimiento de los distintos productos. Para uno de los productos se ha dedicado en el último año un esfuerzo total de 30690 horas-persona y el 68% de ese esfuerzo se concentra en 8 componentes de similar complejidad, tamaño y nivel de uso. En el cuadro siguiente se presenta el esfuerzo en horas-persona dedicado a cada uno de esos componentes durante ese período, discriminado por tipo de mantenimiento.

Tipo de Mantenimiento	Correctivo	Perfectivo	Adaptativo	Preventivo	Total
Componente					
A	0	0	1122	0	1122
B	3198	0	350	0	3548
C	54	0	3021	0	3075
D	1104	298	54	876	2332
E	221	0	0	0	221
F	0	0	1234	0	1234
G	0	0	3907	0	3907
H	87	3245	2098	0	5430
Total	4664	3543	11786	876	20869

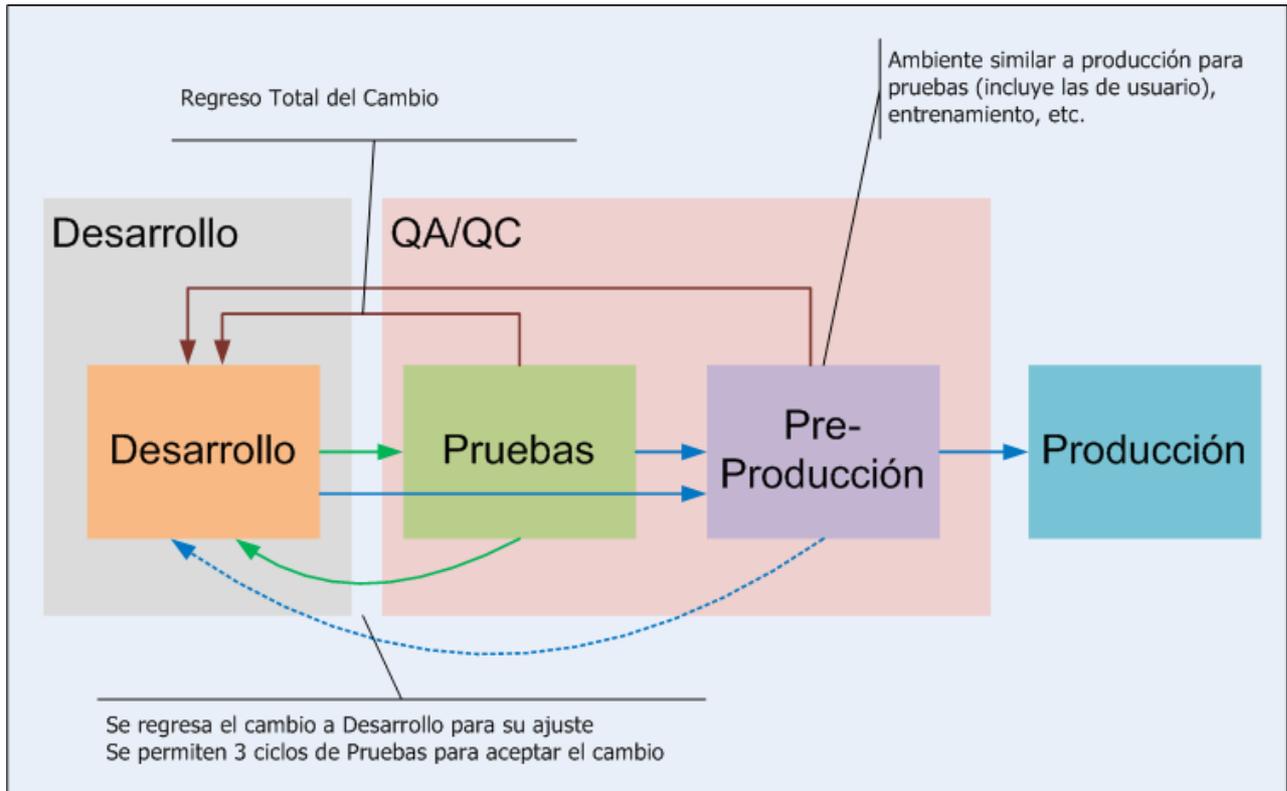
A partir de los datos del cuadro, sería conveniente considerar y evaluar con más detenimiento:

- Destinar esfuerzo al mantenimiento perfectivo y adaptativo para el componente E que solo ha recibido mantenimiento correctivo. No corresponde en este caso realizar mantenimiento preventivo.
 - Reescribir el componente B que de los 8 es el que ha concentrado la mayor parte del esfuerzo de mantenimiento correctivo y corresponde a más del 10% del esfuerzo total de mantenimiento. El esfuerzo de reescritura sería de tipo preventivo.
 - Reescribir el componente H que es al que se ha destinado mayor esfuerzo de mantenimiento total. El esfuerzo de reescritura sería de tipo preventivo.
 - Reescribir el componente D que es el único al que se ha destinado esfuerzo de mantenimiento preventivo. El esfuerzo de reescritura sería de tipo perfectivo.
- 2) Las lecciones aprendidas:
- Deben documentarse al final del proyecto en conjunto con todo el equipo de trabajo (de ser posible también considerar al cliente).
 - Son un activo importante de la organización ya que permiten compartir conocimiento entre distintos proyectos.
 - Son elaboradas por el Jefe del Proyecto y el Cliente ya que son quienes tienen una visión global sobre todas las actividades del proyecto.
 - No deben ser manejadas bajo el proceso de Gestión de la Configuración ya que el documento se mantiene vivo durante todo el proyecto.
- 3) Durante un proyecto de desarrollo de software de gran porte:
- Es conveniente comenzar a planificar la verificación tan pronto como sea posible
 - (a) y para no generar malos entendidos, es altamente recomendable no involucrar al cliente hasta no tener el plan completamente definido.
 - (b) y las tareas de verificación deben incluir también la revisión de todos los entregables que se le hagan llegar al cliente.
 - Son sumamente relevantes las pruebas de integración, en particular si las pruebas del sistema no arrojaron buenos resultados.

- 4) El mantenimiento "Perfectivo" implica:
- Dar respuesta a los problemas que surgen en el uso diario.
 - Dar respuesta a los cambios que ocurren en el ambiente de aplicación del software y en las reglas del negocio.
 - Mejorar algún aspecto o funcionalidad ya presente en el software
 - Introducir cambios para prevenir (o anticipar) fallas en el sistema.
- 5) Sobre el modelo CMMI:
- Para que una organización mejore sus procesos de desarrollo y mantenimiento de software, debe seguir las dos representaciones del modelo: continua y escalonada.
 - a) y la representación continua permite seleccionar en qué áreas de proceso poner foco, dependiendo del contexto de la organización.
 - En la representación escalonada del modelo, la organización sigue un camino hacia la mejora predefinido
 - c) y cada nivel de madurez contiene un conjunto de áreas de procesos específicas a implantar, donde es necesario que se cumpla con la mayoría de ellas para lograr el nivel de madurez deseado.
- 6) Los sistemas legados:
- Dan soporte al funcionamiento de una organización y fueron desarrollados por personal ajeno a la misma
 - (a) y el que la organización no cuente con personal que haya participado en el desarrollo eleva los costos de mantenimiento.
 - Son candidatos a que se les apliquen técnicas de rejuvenecimiento tales como reestructuración del código, re-documentación e ingeniería reversa.
 - Suelen ser de poca relevancia para el funcionamiento de la organización, lo que explica que no hayan sido sustituidos por sistemas que tengan un menor costo de mantenimiento.
- 7) Acerca de las pruebas de integración:
- Con la estrategia *big-bang* suele ser difícil conocer rápidamente qué módulo de los integrados es el que origina una falla que surge en las pruebas de integración.
 - Normalmente los *drivers* son más complejos de desarrollar que los *stubs*.
 - En la estrategia *top-down* puede ser muy complejo (e incluso imposible) generar determinados casos de pruebas para algunos módulos.
- Se cumplen I, II y III.
 - Se cumplen I y III. No se cumple II.
 - Se cumple I. No se cumple II y no se cumple III.
 - Se cumple II. No se cumple I y no se cumple III.
- 8) El desempeño y la capacidad de un proceso:
- Son conceptos equivalentes.
 - Son conceptos complementarios.
 - b) y el segundo es la probabilidad de que el primero coincida con las estimaciones realizadas.
 - Son conceptos contrapuestos ya que el primero es inversamente proporcional al segundo.
- 9) Considere que el primer nodo de un grafo de flujo de control es $[If\ A>1\ and\ B=0]$, entonces:
- Si aplico el Criterio de cubrimiento de condición, la condición $A>1$ debe tomar al menos una vez el valor true y al menos una vez el valor false en el conjunto de casos de pruebas.
 - Si aplico el Criterio de cubrimiento de condición, la condición $B=0$ debe tomar al menos una vez el valor true y al menos una vez el valor false en el conjunto de casos de pruebas.
 - Según el criterio de Todos los usos (Caja Blanca) en dicho nodo hay un p-uso de A y un p-uso de B.
 - Todos son correctos.
- 10) Sobre SCM indique la opción que **no** es correcta:
- Es posible y recomendable incluir en el Plan de SCM cómo se manejarán las situaciones en las que debemos saltar los procedimientos definidos para la gestión de la configuración (o sea las excepciones).
 - El alcance y el formalismo de la gestión de la configuración deberían adaptarse para cada situación.
 - Todo proyecto debe tener una Gestión de Cambios de alto grado de formalismo. Eso permite maximizar la productividad minimizando errores.
 - Es posible tener más de una línea base en un proyecto.

- 11) Los cambios durante el mantenimiento:
- 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 asegurar que el cambio es pertinente y que su introducción no va a afectar de forma negativa el funcionamiento del sistema.
 - (a) y para lograrlo, los cambios deben llevarse a cabo de acuerdo a un procedimiento de control de cambios en el que los cambios, en caso de existir un alto grado de formalismo, deben ser aprobados por el llamado Comité de Control de Cambios que aúne las visiones de Cliente/usuarios y del equipo de mantenimiento.
 - Deben ser implementados por el Comité de Control de Cambios en función de las prioridades asignadas por el cliente y de la disponibilidad de recursos del equipo de mantenimiento.
 - Se deben llevar a cabo de acuerdo a un procedimiento que incluya la evaluación de la pertinencia del cambio, su priorización, análisis de impacto, planificación de su ejecución, implementación del cambio, verificación del mismo y su puesta en producción. Todo cambio que se aplique al producto debe llevarse a cabo cumpliendo cada uno de estos pasos y en ese orden.
- 12) Respecto a la liberación de un sistema:
- En general se realiza un entrenamiento a los usuarios del sistema. Para llevarlo a cabo se debe utilizar el documento de casos de uso.
 - El entrenamiento a la totalidad de los usuarios constituye una forma de validación del producto, por lo que conviene que sea llevado a cabo de forma temprana, en lo posible al comienzo de la prueba del sistema.
 - Cuando la cantidad de usuarios a entrenar es muy grande, se puede entrenar a un grupo reducido que se encargue de entrenar al resto
 - (c) y usuarios que participen en la prueba del sistema son buenos candidatos tanto para brindar entrenamiento como para brindar soporte del sistema en producción.
- 13) La madurez (según la definición del curso) de una organización que construye y mantiene software se mide a través de:
- La capacidad del conjunto de sus procesos.
 - Su imagen y la satisfacción de sus clientes.
 - Las certificaciones internacionales obtenidas.
 - Dos de las anteriores son correctas.
- 14) Acerca de la verificación de software:
- Las técnicas de verificación estática, en comparación con las dinámicas, han mostrado ser más efectivas en la detección temprana de defectos
 - (a) y las técnicas de verificación dinámica dependen del conjunto de casos de prueba que se ejecute y por ende no tiene una validez general.
 - La verificación estática no sirve para verificar documentos de especificación de requerimientos.
 - Para un proyecto de desarrollo se debe definir qué tipo de técnica usar (o estática o dinámica). Esto se debe a que las mismas no son complementarias.
- 15) ¿Dadas las siguientes opciones, en qué momento es más aconsejable comenzar con las actividades de verificación?
- Cuando se tiene una versión validada del plan de pruebas.
 - Cuando se obtienen las primeras unidades de código ejecutable.
 - Cuando se tiene el primer borrador de especificación de requisitos.
 - Cuando se tiene la primera versión del producto a entregar al cliente.

- 16) En relación a la Gestión de la Configuración y en particular al ejemplo de Manejo de Ambientes visto en el curso. Indique la opción correcta de las siguientes afirmaciones.



- a) Las pruebas de desempeño del sistema (estres, carga, volumen, etc) se deberán realizar en el ambiente de Pre-Producción.
- b) La capacitación se debe realizar en el ambiente de Producción.
- c) Las pruebas de desempeño del sistema (estres, carga, volumen, etc) se deberán realizar en el ambiente de Pruebas.
- d) Dos de las anteriores son correctas.
- 17) La norma de calidad de software ISO/IEC 25000, difiere de la ISO/IEC 9126-1 en que:
- a) Incorpora las métricas y los criterios de evaluación de la calidad de productos de software.
- b) Establece criterios para la especificación de requisitos calidad de productos de software.
- c) Incorpora nuevos atributos y factores de calidad como la fiabilidad.
- d) Considera únicamente la calidad externa del producto de software.
- 18) Dadas las siguientes afirmaciones indique cuál respuesta es correcta:
- Descubrir defectos es uno de los objetivos de la verificación.
 - Evaluar la calidad de los productos es uno de los objetivos de la verificación.
 - La corrección de los defectos detectados es parte de la verificación.
 - Es habitual en la industria de software que las pruebas unitarias las realice un equipo especializado de testing y no el equipo de desarrollo.
 - El conocimiento de las interfaces y la forma en la que interactúan las componentes de un sistema es importante para realizar las pruebas de integración.
- a) Solo se cumplen I y III,
- b) Solo se cumplen I, II y IV,
- c) Solo se cumplen I, II, IV y V, pero no el resto.
- d) Solo se cumplen I, II y V, pero no el resto.
- 19) Dado un proyecto que incluye análisis, diseño, construcción, prueba y puesta en producción (liberación) del software, para sustituir el sistema que da soporte al funcionamiento de una organización por un sistema nuevo:
- a) La planificación de la liberación debiera comenzar junto con la planificación general del proyecto, para poder dimensionar de forma adecuada el alcance del proyecto.

- b) La estrategia de liberación big-bang es más conveniente para manejar riesgos asociados al impacto de cambios en los procedimientos de la organización en comparación con las estrategias de liberación paulatinas.
- c) La sustitución muy probablemente requerirá migrar información del sistema viejo al nuevo, lo que podría involucrar un plazo y un esfuerzo muy considerables, pero con un nivel bajo riesgo. Quienes se encarguen de llevar a cabo esta migración, fundamentalmente deberán contar con un conocimiento acabado del modelo de datos del sistema nuevo.
- d) De forma previa a la puesta en producción se podría realizar un procesamiento en paralelo durante al menos un mes, lo que constituye un mecanismo de verificación del software sumamente efectivo.

- 20) Sobre la calidad de un producto:
- La calidad de un producto no está influenciada por el proceso con el cual se construye
 - La calidad de un producto está influenciada por el proceso con el cual se construye
 - b) y por ende, mejorar el proceso puede influenciar de forma positiva en la calidad del producto
 - b) y evaluar los procesos es una actividad que se debe realizar solamente al comienzo de cada proyecto
- 21) Sobre los casos de uso y las pruebas de software:
- En la generación de casos de prueba a partir de casos de uso se deben tener en cuenta los flujos alternativos del CU
 - (a) y no es conveniente utilizar la técnica partición en clases de equivalencia para seleccionar los distintos valores de entrada al CU.
 - (a) y muchas veces es necesario considerar los flujos alternativos más de una vez, en distintos escenarios y en distintas combinaciones con otros flujos alternativos.
 - Los casos de uso no se deben utilizar para generar casos de prueba.
- 22) En el mantenimiento de software:
- Los encargados de llevar a cabo el mantenimiento no siempre pueden tener una comprensión total del sistema que están manteniendo, por lo que gran parte del esfuerzo se dedica a entender el sistema bajo mantenimiento.
 - Realizar el análisis del impacto de un cambio representa normalmente una porción significativa del esfuerzo total dedicado a realizar cada cambio.
 - Es frecuente que la gerencia asigne más importancia al desarrollo que al mantenimiento, lo que resulta en un importante factor de desmotivación del personal encargado del mantenimiento.
 - La realización de las pruebas requiere contar con un ambiente adecuado para llevarlas a cabo, lo que suele plantear dificultades, sobre todo si se requiere contar con datos similares a los del ambiente de producción y no resulta posible duplicar los datos de producción debido a su volumen y/o por razones de confidencialidad.
 - Existe un conflicto entre la necesidad de contar rápidamente con una solución y que la solución preserve atributos de calidad relevantes del producto.
 - La secuencia de reparaciones y mejoras centradas en soluciones de corto plazo tiende a degradar la calidad general del producto y en especial las facilidades de adaptación y de mantenimiento.
 - Las sucesivas intervenciones para modificar el software en general tienden naturalmente a reducir su complejidad.
 - El costo de mantenimiento tiende a bajar con el paso del tiempo, afectado por diversos factores, entre los que sobresale la reducción relativa del costo del hardware respecto al del software.
- De las afirmaciones anteriores son correctas las identificadas por los números:
- I, II, III, IV, V, VI
 - I, III, IV, V, VII
 - II, III, IV, V, VIII
 - I, IV, VI, VII, VIII
- 23) Los experimentos formales son una técnica de evaluación utilizada para:
- Analizar las tendencias y las relaciones entre varios productos y/o proyectos
 - Validar o refutar hipótesis sobre productos y/o proyectos
 - Comparar los resultados de varios proyectos con la línea base previamente establecida
 - Ninguna de las anteriores
- 24) Las pruebas de regresión:
- Son pruebas concebidas específicamente para detectar fallas en el rendimiento del sistema.
 - Ayudan a detectar faltas introducidas al realizar modificaciones al código, chequeando que funcionalidades que estaban funcionando correctamente lo sigan haciendo.
 - (b) y únicamente se pueden aplicar en las pruebas unitarias.
 - (c) y es conveniente tener estas pruebas automatizadas.

25) Dado el siguiente ejemplo que contiene un programa con su ejecución simbólica, indicar cuál de las opciones es la más correcta. En el ejemplo las variables de entrada al programa (ent1 y ent2) son variables numéricas.

```
[ent1 = A, ent2 = B]
if (ent1 > ent2) then begin
  write ("primer then");
  [ent1 = A, ent2 = B] [A > B]
  if (ent2 > ent1) then
    write ("segundo then");
    [ent1 = A, ent2 = B] [A > B and B > A]
    [ent1 = A, ent2 = B] [false] - Esta ejecución no es posible
  else
    write ("segundo else");
  endif
end
endif
```

- a) Una evaluación "false" en una ejecución simbólica siempre indica que el camino de ejecución realizado nunca se puede ejecutar.
- b) Una evaluación "false" en una ejecución simbólica siempre indica código inalcanzable (código muerto).
- c) (a) y (b) son correctas.
- d) Ninguna de las anteriores.