

- 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. Si se quiere calcular los PFSA de la siguiente transacción:  
Se selecciona de un menú la opción de ingreso de datos de factura de proveedor:
- código proveedor (E)
  - nombre proveedor (S)
  - fecha factura (E)
  - importe total (S) (se calcula)
    - \* ( código artículo (E)
    - precio unitario (S) (se saca del archivo Precios)
    - cantidad (E)
    - Importe (S)) (se calcula)

Se ingresan todos los datos de entrada y se clikea sobre un botón Guardar, y se calcula y despliega el importe total. ¿Cuál de las siguientes afirmaciones es la correcta?

- a) **Se cuentan 8 DET.**
  - b) Se cuentan 4 DET.
  - c) La cantidad de DET dependerá de la cantidad de artículos que se ingresen.
  - d) Se cuentan 9 DET.
- 
2. En cuanto al requisito “La interfaz debe ser amigable”.
- a) Esta descripción es perfectamente válida y suficiente para documentar como requisito.
  - b) No es válida como descripción de un requisito en el SRS, porque no es verificable.
  - c) Es una meta y aporta información sobre las intenciones del usuario
  - d) **Se cumplen b) y c)**

3. Dadas las siguientes afirmaciones respecto a pruebas de software:
- I. Los términos verificación y validación son el mismo concepto.
  - II. Las pruebas de software pueden asegurar la ausencia de defectos.
  - III. Las inspecciones no pueden sustituir las pruebas de software.
  - IV. Una de las principales desventajas de TDD es que aumenta los costos de las pruebas de regresión.

¿Cuál de las siguientes opciones es correcta?

- a) Las afirmaciones I) y II) son correctas.
  - b) Las afirmaciones III) y IV) son correctas.
  - c) Las afirmaciones II y III) son correctas.
  - d) **La afirmación III es correcta.**
- 
4. La planificación del proyecto según el MUM se debe realizar según el siguiente enfoque:
- a) Planificación ágil.
  - b) Desarrollo basado en planes, tal cual lo describe Sommerville.
  - c) Se planifica todo al comienzo y se gestionan los cambios durante el resto del proyecto.
  - d) **Se realiza una planificación macro al comienzo y se planifica en detalle al comienzo de cada iteración, además de lo que se deba ajustar en razón de los cambios que surjan durante el proyecto.**

5. En relación a la Gestión de Riesgos, y de acuerdo al PMBoK (visto en clase), un riesgo :
- a) Es un evento o condición inciertos, que, en caso de producirse, afecta positiva o negativamente todos los objetivos del proyecto.
  - b) **Es un evento o condición inciertos, que, en caso de producirse, afecta positiva o negativamente al menos algún objetivo del proyecto.**
  - c) Es un evento o condición inciertos, que, en caso de producirse, afecta negativamente todos los objetivos del proyecto.
  - d) Es un evento o condición inciertos, que, en caso de producirse, afecta negativamente al menos algún objetivo del proyecto.

6. Sobre la verificación y validación de software:
- Se ha mostrado mediante experimentos controlados y en la práctica industrial que las Inspecciones de software son normalmente más efectivas (porcentaje de defectos detectados) que las pruebas de software.
  - Uno de los posibles motivos por los cuales las Inspecciones no son adoptadas por la industria de software es porque las mismas requieren costos adicionales en etapas tempranas del desarrollo de software y los gerentes tienen miedo de tomar el riesgo de que luego (en etapas finales del desarrollo) no se logren los beneficios esperados por el uso de las Inspecciones.
  - Los métodos formales son un método teórico de desarrollo de software que no ha sido utilizado en la industria de software.

¿Cuál de las siguientes opciones es correcta?

- La afirmación I es correcta. Las afirmaciones II y III no lo son.
- Las afirmaciones I y II son correctas. La afirmación III no lo es.
- Las afirmaciones I y III son correctas. La afirmación II no lo es.
- Las tres afirmaciones son falsas.

7. Respecto a la ingeniería de software:
- Es una disciplina que ha tenido una evolución acelerada poco después de su surgimiento hacia fines de la década de 1960, pero esa evolución se ha ido ralentizando paulatinamente y se encuentra desde hace más de 10 años en una situación de relativa estabilidad.
  - Guarda relación con la Administración General y en especial con el área de Gestión de Proyectos. La Gestión de Recursos Humanos ha perdido importancia relativa para la ingeniería de software durante los últimos veinte años debido a que en ese período disminuyó la importancia de los recursos humanos en el proceso de producción de software.
  - Guarda relación con la ingeniería de sistemas dado que al construir software este termina formando parte de un sistema compuesto al menos por hardware, software y personas con las que interactúa.
  - Guarda relación con la programación. Construir un producto de software requiere codificar el programa que lo soporta, pero su construcción involucra más que la programación. Asimismo el esfuerzo necesario para construir un producto de software que va a ser utilizado por distintos usuarios suele resultar mayor que construir un programa que va a ser utilizado por el propio programador y construir un sistema de programas que interactúan entre sí suele requerir más esfuerzo que la suma de los esfuerzos necesarios para construir cada uno de los programas que lo componen.
  - Persigue la mejora de la productividad y calidad en la producción de software fundamentalmente a través de la introducción de herramientas que automaticen los procesos de definición de requisitos, diseño, construcción y verificación de código.
  - Entre las nociones fundamentales que han pasado la prueba del tiempo cabe mencionar: abstracción, prototipación, proceso de software, reuso (reutilización) y arquitectura de software.

Son correctas las respuestas:

- I, II, III, IV, V, VI
- II, III, IV, V, VI
- III, IV, VI
- III, IV

8. Si Ud. desea estimar el esfuerzo y la duración adecuada para un proyecto utilizando COCOMO II, para un desarrollo que parte de cero, no utiliza componentes reusables, una vez que ya tiene un diseño inicial (objetos y módulos), el modelo más apropiado sería:
- el modelo de composición de aplicaciones
  - el modelo de diseño temprano
  - el modelo de reuso
  - el modelo posarquitectura

9. Las revisiones son actividades relacionadas al aseguramiento de la calidad que se pueden aplicar a:
- Todos los artefactos generados durante el proceso de desarrollo de software.
  - Solamente a los documentos de especificación de requerimientos, plan de proyecto y manual de usuario.
  - Únicamente a los entregables del proyecto.
  - (a) y debe realizarse de forma planificada de forma de priorizar los artefactos más relevantes o sensibles al proyecto.

10. Los analizadores estáticos de código son herramientas utilizadas para encontrar ciertos tipos de defectos en los programas. Algunas de sus características son:
- No son capaces de identificar porciones de código inalcanzables.
  - Son capaces de identificar defectos en el mal uso de las variables, código inalcanzable, y defectos en los parámetros de llamada a procedimientos y funciones.
  - (b) Y si bien generan falsos positivos, son una herramienta efectiva para encontrar defectos en los programas antes de su ejecución.**
  - Identifican código inalcanzable, variables que no fueron inicializadas y no generan falsos positivos.
- 
11. La evaluación de factibilidad:
- Sommerville la presenta como parte del proceso de los requisitos, pero otros autores la colocan fuera, como una entrada al proceso de los requisitos.**
  - Se hace solo al comienzo del proyecto.
  - Responde a la pregunta de si el proyecto es viable con la tecnología existente, independientemente de las restricciones de tiempo y costo.
  - Culmina con un informe de factibilidad donde solo se presenta el análisis realizado, sin hacer recomendaciones.
- 
12. Dados los siguientes proyectos y modelos de proceso.
- Proyecto:
- A - de alto riesgo, muy complejo, alcance no del todo definido, subconjuntos de funcionalidad definidos que resultan útiles para los usuarios, larga duración (dos años o más).
- B – Riesgo medio, alcance definido, subconjunto de funcionalidad definido y útil para los usuarios, duración media (seis meses).
- C – Riesgo bajo, simple, alcance definido, corta duración (seis semanas).
- Proceso:
- 1 – Cascada
- 2 – En Fases con Evaluaciones Parciales Internas
- 3 – En Fases con Liberaciones Parciales en Producción
- 4 – De Prototipación
- Marque la asignación que le parezca más adecuada. Si un mismo proyecto aparece asignado a más de un proceso, debe entenderse que cualquiera de los dos resultaría igualmente adecuado.
- Para los procesos 2 y 3 se puede especificar además entre paréntesis el modelo de proceso a utilizar en cada fase. Por ejemplo 2(1) significa “Proceso en Fases con Evaluaciones Parciales Internas” y en cada fase se aplica Cascada. 3(2(1)) significa que en este caso en cada fase de 3 se aplica “En Fases con Evaluaciones Parciales Internas” y en cada fase de 2 se aplica Cascada.
- A3(2(1)), A3(1), B3(2(4)), B3(2(1)), C1**
  - A3(2(1)), A1, B2(1), B1, C3(2(1)), C3(1)
  - A2(1), A4, B1, B4, C1, C4
  - A3(2(1)), A4, B4, B2(3(1)), C1, C2(3(1))
- 
13. Respecto a modelos de proceso:
- Las principales ventajas del modelo de proceso en fases con liberaciones parciales en producción son que el cliente puede contar con resultados de utilidad en un plazo más corto, lo que permite reducir el plazo de salida al mercado (time to market) y que usuarios y cliente tienen oportunidad de evaluar antes atributos de calidad externos de al menos una parte del producto.**
  - Entre las principales ventajas del modelo de proceso en cascada están su simplicidad, que aporta orden al proceso y que facilita la gestión de los riesgos asociados a problemas con la arquitectura de software o del sistema.
  - Entre las principales ventajas del modelo de proceso en fases con evaluaciones internas parciales están que el cliente puede contar con resultados de utilidad en un plazo más corto, lo que permite reducir el plazo de salida al mercado (time to market).
  - El “Manifiesto por el Desarrollo Ágil de Software” enfatiza la conveniencia de contar con un proceso de desarrollo definido que sea cumplido de forma estricta con herramientas adecuadas, por encima del papel de los individuos y de sus interacciones en la construcción de software.
- 
14. Un prototipo sirve para:
- Evaluar riesgos técnicos.
  - Recabar requisitos.
  - Validar requisitos con los clientes.
  - Todas las anteriores.**

15. Entre las cualidades más relevantes para un software encargado de controlar el funcionamiento de una bomba de insulina están:
- Portabilidad, Disponibilidad, Adecuación al uso, Calidad de la documentación técnica.
  - Seguridad de la integridad física (safety), Adecuación al uso, Tolerancia a fallas (fault tolerance), Confiabilidad.**
  - Interfaz de usuario atractiva, Adecuación al uso, Facilidad de mantenimiento, Tiempo de respuesta.
  - Modularidad, Facilidad de ser probado, Facilidad de mantenimiento, Seguridad de la información.

16. Enfoque de valor ganado.

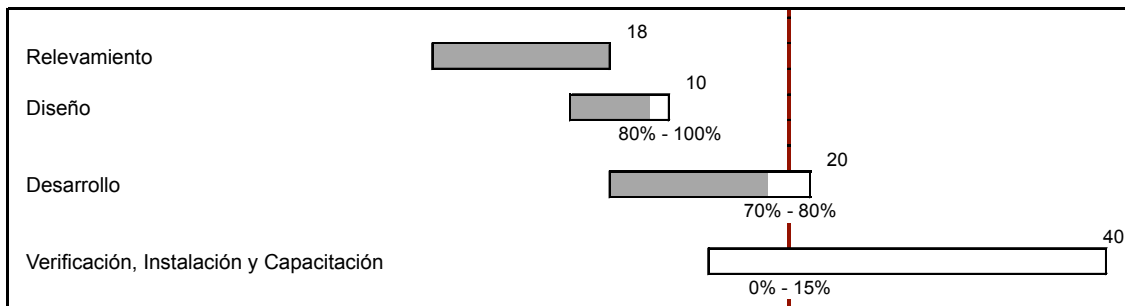
Durante el seguimiento de un proyecto Ud. como administrador aplica el enfoque de valor ganado. En la situación siguiente, donde se consideran 4 actividades con el siguiente valor total planificado (VP) y se aplican las siguientes técnicas de medición de avance para cada tarea:

- Relevamiento (VP = 18)- fórmula fija 25/75
- Diseño (VP = 10) – porcentaje de completitud.
- Desarrollo (VP = 20) – hitos con peso: (5 al comienzo, 5 a la mitad, y 10 al finalizar)
- Verificación, instalación y capacitación (VP = 40) – fórmula fija 50/50

Al día de hoy, según la planificación, las tareas de relevamiento y diseño deberían estar terminadas; la de desarrollo debería estar completada en un 80%, y la de verificación, instalación y capacitación debería estar completada en un 15%.

Sin embargo, como se indica en la figura, relevamiento está completa en un 100%, diseño en un 80%, desarrollo en un 70% y verificación, instalación y capacitación aún no ha comenzado.

Se sabe que el costo actual es de 45.



Indique si:

- El proyecto está atrasado un valor de 10 y se ha gastado más de lo planificado para el avance alcanzado.
- El proyecto está atrasado un valor de 10 y se ha gastado menos de lo planificado para el avance alcanzado.
- El proyecto está atrasado un valor de 14 y se ha gastado más de lo planificado para el avance alcanzado.**
- El proyecto está atrasado un valor de 14 y se ha gastado menos de lo planificado para el avance alcanzado.

17. Dentro de los modelos de comportamiento que menciona Sommerville, que modelan el comportamiento dinámico del sistema mientras que se ejecuta, se encuentran:

- Los diagramas de casos de uso.
- El modelo conceptual.
- El Gantt del proyecto.
- Las máquinas de estado.**

18. El precio que una empresa cobra al cliente por el software desarrollado (*software pricing*):

- Es siempre el costo de desarrollo más ciertas utilidades para la empresa desarrolladora.
- En la práctica, la relación entre el costo del proyecto y el precio cotizado debe tener en cuenta, además del costo y las ganancias, cuestiones organizacionales, económicas, políticas y del negocio, entre otras.**
- No involucra consideraciones organizacionales; por lo tanto, la definición del precio de un producto de software no debe involucrar a personal de marketing, ventas, gerentes de proyectos, etc.
- Se fija siempre utilizando la estrategia *pricing-to-win*.

19. En *Cleanroom* se utilizan las siguientes técnicas/métodos:

- Pruebas unitarias con cubrimiento de sentencias.
- Especificaciones formales.
- (b) y test estadísticos del sistema basados en un perfil operacional.**
- Entrevistas con los usuarios como única técnica de elicitación de requisitos.

20. Durante el proceso de validación, se deben realizar distintos tipos de verificaciones sobre el documento de requisitos:
- I. De validez, que implica ver si el sistema funciona correctamente.
  - II. De consistencia, que implica si no ha habido cambios en ese requisito desde su formulación inicial.
  - III. De completitud, que verifica si están todas las funciones requeridas por el cliente.
  - IV. De factibilidad, que mira solo a la factibilidad técnica de implementar un requisito.
  - V. De viabilidad, que identifica si el requisito a incluir resuelve las necesidades del cliente.
  - VI. Verificabilidad, que verifica si el requisito puede ser probado con un procedimiento viable, sea o no accesible económicamente.
  - VII. Comprensibilidad, que verifica si los requisitos son entendibles por los stakeholders y los desarrolladores.
  - VIII. Rastreabilidad, que verifica si está documentado el origen de cada requisito.
  - IX. No ambigüedad, que implica que la formulación de los requisitos no se presta a dobles interpretaciones.
  - X. que el documento de requisitos cumpla con los estándares de la empresa.

Se cumplen:

- a) I), III), V), VI), VII) y IX)
  - b) III), VII), VIII), IX) y X)**
  - c) I), II), IV), VIII) y X)
  - d) Todas las opciones.
- 
21. El cronograma del proyecto (*project scheduling*):
- I. Es el trabajo de decidir cómo organizar el proyecto en tareas y cómo y cuándo estas tareas se ejecutarán.
  - II. Se debe estimar el tiempo necesario para completar cada tarea, así como el esfuerzo requerido.
  - III. No es necesario estimar los recursos necesarios para cumplir cada tarea ya que se sabe que están disponibles.
  - IV. Se deben asignar responsables a cada tarea.
  - V. Existen diversas representaciones gráficas para el cronograma de proyecto. Los gráficos de barras muestran la duración de la actividad y los plazos, siendo la representación de *project scheduling* más utilizada.

Son correctas las respuestas:

- a) Se cumplen I), II) y V)
  - b) Se cumplen II), III) y V)
  - c) Se cumplen I), II), IV) y V)**
  - d) Se cumplen I), III), y IV)
- 

22. Dado el siguiente fragmento de programa:

```
int x;
...[código no mostrado]
if (x>0) {
    ...[código no mostrado]
}
else {
    z = z/x;
}
```

y suponiendo que una división entre cero causa una falla durante la ejecución conteste cuál de estas afirmaciones es correcta:

- a) Una Inspección o revisión de software puede detectar este defecto.**
  - b) (a) y Pruebas unitarias que satisfagan el cubrimiento de sentencias son suficientes para provocar la falla.
  - c) Se cumple (a). No se cumple (b). y pruebas unitarias que satisfagan el cubrimiento de caminos linealmente independientes son suficientes para provocar la falla.
  - d) Ninguna de las anteriores afirmaciones es correcta debido a que las pruebas no pueden garantizar la ausencia de defectos (Dijkstra).
- 
23. Los requerimientos no-funcionales incluyen:
- a) Restricciones que especifican el comportamiento del producto, así como los que se derivan de los factores externos al sistema.
  - b) Restricciones que se derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador.
  - c) Declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares.
  - d) a) y b) son correctas.**

24. El proceso de gestión del cambio:
- Es el proceso que se sigue en el análisis de requisitos de un cambio, e incluye las actividades que evalúan el impacto y costo del cambio.
  - Debería aplicarse a todos los cambios propuestos a los requisitos, estén o no en la línea base.
  - Debería aplicarse solo a los cambios propuestos por el cliente.
  - Debería aplicarse desde el comienzo de la administración de los requisitos.
- 
25. Uno de los objetivos de la V&V es establecer confianza en que el sistema de software es adecuado para su propósito. Esto significa que el sistema va a ser suficientemente bueno para su uso esperado. Según Sommerville, el nivel de calidad y confianza que requieren/piden/esperan clientes y usuarios para un cierto software depende entre otras cosas de:
- El propósito de uso del software.
  - El nivel de expectativa de los usuarios.
  - El plan del proyecto de desarrollo.
  - El plan de pruebas y la ejecución de todo el proceso de pruebas.
  - La arquitectura de software
- ¿Cuál de las siguientes opciones es correcta?
- Todas las afirmaciones son correctas.
  - Las afirmaciones I y II son correctas.
  - Las afirmaciones IV y V son correctas.
  - La afirmación III es correcta.