

Instrucciones

- Cada pregunta múltiple opción respondida correctamente tiene un valor de 4,762 puntos.
- Cada pregunta múltiple opción respondida incorrectamente resta 1,587 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 dos horas y quince minutos.
- El puntaje total es de 100 puntos.

1. Seleccione la respuesta correcta:

- a) Todos los procesos de desarrollo de software surgen del CMMI (anteriormente CMM) donde se encuentran todas las prácticas de la ingeniería de software.
- b) Un proceso de desarrollo de software normalmente está compuesto, entre otras cosas, por un conjunto de actividades ordenadas de alguna forma, roles y fases.
- c) **b) y puede incluir actividades de mejora de proceso.**
- d) Una contra de los procesos que se basan fuertemente en planes es que realizan el plan del proyecto una única vez y hasta que no se termine el proyecto el plan no debe cambiar.

2. Seleccione la respuesta correcta:

- a) Los proyectos que usan procesos ágiles no necesitan actividades de gestión.
- b) La complejidad de los sistemas de software, que ha incrementado en los últimos años, no tiene relación con las fallas que se encuentran en los productos de software. Estas fallas se deben a las malas prácticas de la ingeniería de software.
- c) Cuando se utiliza Scrum, en conjunto con alguna práctica técnica de XP, las historias de usuario deben estar todas escritas, al menos a alto nivel en los primeros Sprints para poder estimar el cronograma total del proyecto.
- d) **Al utilizar métodos ágiles, no es siempre necesario especificar los requisitos mediante historias de usuario. Pudiendo utilizar, por ejemplo, casos de uso (Use cases).**

3. La construcción de software tiene diferencias con la programación (codificación) de software.

- a) En la construcción únicamente se programa lo que se diseñó en una etapa anterior y se realizan pruebas unitarias. Esto es considerado como suficiente y eficiente por los métodos modernos de desarrollo de software.
- b) Al ser una la construcción de software más completa (de alguna forma) que la programación, esta incluye otras actividades. Entre ellas, test de sistema, los test de performance, los test beta y alfa (estos solo en caso de que se usen en la organización o proyecto) y los test de liberación y aceptación de parte del cliente y usuario.
- c) Durante la construcción, en general, es una buena práctica construir con reuso (es decir, cuando se construye pensar qué se puede reutilizar) así como construir para el reuso (es decir, construir pensando en que luego lo que se está construyendo puede ser reutilizado).
- d) **c) y durante la construcción es normal realizar actividades de diseño detallado y de pruebas unitarias.**

4. Según Sommerville (en el libro utilizado en el curso), todos los métodos ágiles comparten ciertas características.
- I) Los procesos de especificación de requisitos, diseño e implementación se van intercalando a medida que avanza el proyecto. Es decir, no se cuenta con una especificación completa de los requisitos, para luego hacer un diseño completo y luego implementar.
 - II) El sistema es desarrollado en una serie de incrementos.
 - III) Usuarios finales y otros interesados (*stakeholder*) en el proyecto están involucrados en especificar y en evaluar cada incremento.
 - IV) Las iteraciones tienen siempre la misma duración en semanas luego de establecidas al comienzo del proyecto.
 - V) Las retrospectivas no se realizan si no se cuenta con un moderador y un *coach* que pertenecen al equipo de desarrollo.
- a) Solamente I, II y III son verdaderas.
 b) Solamente III es verdadera.
 c) Solamente II y IV son verdaderas.
 d) Todas son verdaderas.
5. Según Sommerville (en el libro utilizado en el curso), los métodos ágiles pueden tener algunos problemas en el mantenimiento de software. Siendo uno de estos el siguiente:
- a) No poder tener iteraciones variables en tiempo. Es decir, la propuesta de Scrum de tener iteraciones de siempre la misma duración durante un proyecto no es adecuada para el mantenimiento.
 - b) La falta (o poca) documentación del producto y de los distintos sub-productos de software (requisitos, arquitectura, diseño detallado, etc.). La documentación, justamente, se supone que describe el sistema y facilita la comprensión del mismo.
 - c) El uso de la técnica *test-first* (probar primero), propuesta por XP, lleva a que en el mantenimiento se tengan pruebas automatizadas que no se pueden ejecutar.
 - d) Las retrospectivas son un problema para el mantenimiento porque normalmente no hay tiempo para realizarlas. Sobre todo, en el mantenimiento correctivo.
6. Supongamos que usted quiere probar el siguiente método.

```

public void verificarEdad(int edadCandidato)
{
1   if (edadCandidato < 18) {
2       System.out.println("Menor");
3   }
4   else {
5       if (edadCandidato > 30) {
6           System.out.println("adulto");
7       }
8   }
9   else {
10      System.out.println("Joven");
11  }
12  }
13  }

```

¿Cuántos casos de prueba son necesarios (la mínima cantidad) para asegurar un cubrimiento de decisión?

- a) 2
- b) 3
- c) 4
- d) 5

7. Suponga que está trabajando en un proyecto de software que sigue un enfoque ágil. Acaba de terminar el noveno sprint, el cual incluía el desarrollo de 2 nuevas funcionalidades y cambios en 4 funcionalidades ya existentes. Como resultado de las pruebas realizadas por el equipo de desarrollo sobre el incremento, se encontraron solamente 2 defectos triviales: un texto mal alineado y una palabra que le faltaba tilde. Se realizaron un total de 257 pruebas, de las cuales 233 estaban automatizadas. Dentro de las pruebas automatizadas, 5 pruebas eran de regresión. Los defectos fueron encontrados por las pruebas manuales. Al momento de poner en producción el incremento, se reporta una falla menor, relacionada a la precisión con la cual se muestra un resultado.

De acuerdo al contexto planteado, considere las siguientes afirmaciones:

- I. Se puede asegurar que el incremento liberado es de muy buena calidad.
- II. El resultado de las pruebas podría indicar que el conjunto de casos de prueba sufre de “la paradoja del pesticida”.
- III. La cantidad de pruebas realizadas es muy poca como para asegurar una buena calidad del software.
- IV. Se puede asegurar que no se introdujeron nuevos defectos, ya que se hubieran detectado mediante las pruebas de regresión.
- V. Se puede asegurar que no se degradó la performance del sistema.
- VI. Las pruebas realizadas no fueron buenas, el equipo de desarrollo no tiene la independencia necesaria para realizar las pruebas, se necesita de un equipo externo.

Seleccione la opción correcta:

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

8. En relación a los criterios de entrada y salida del proceso de ejecución de las pruebas (que se define en el Plan de Pruebas):
- I. Un criterio de salida podría ser que los casos de prueba cumplan con un cubrimiento del 90% del código.
 - II. Un criterio de salida podría ser que se ejecuten todas las pruebas planificadas y que ninguna falle.
 - III. Un criterio de entrada podría ser que el ambiente de pruebas esté disponible.
 - IV. Un criterio de entrada podría ser que el software no presente fallas.
 - V. Un criterio de entrada podría ser que el software esté instalado y configurado en el ambiente de pruebas.

Seleccione la opción correcta:

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

9. Imagine que es parte del equipo de mantenimiento del software de un autoservice, que hace 10 años está en producción. En el último mes se realizaron las siguientes tareas:
- Se migraron datos de clientes desde una planilla excel.
 - Se realizó una refactorización de código porque algunos reportes demoraban mucho más de lo esperado, ya que la facturación del autoservice creció un 600% en el último año.
 - Se agregó información de los productos marcados con octógonos para realizar reportes al MSP de las ventas de los mismos según la nueva normativa.
 - Se solucionó un problema de impresión de tickets cuando se facturaban bebidas alcohólicas con una descripción muy larga.

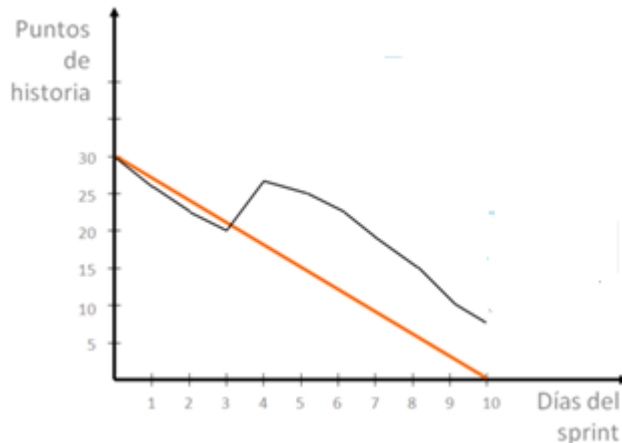
De acuerdo a la clasificación de mantenimiento de Pfleeger, seleccione la opción correcta:

- a) El mantenimiento realizado fue preventivo y correctivo.
 - b) El mantenimiento realizado fue correctivo y reactivo.
 - c) El mantenimiento realizado fue preventivo y adaptativo.
 - d) **El mantenimiento realizado fue adaptativo y correctivo.**
10. En relación a las estrategias de conversión/adopción de software
- a) **Es conveniente que una estrategia big-bang tenga un plan de contingencia ante un posible fallo del sistema nuevo. La estrategia, por ejemplo, podría ser pasar a usar el sistema viejo.**
 - b) Una estrategia big-bang permite planificar escenarios de entrenamiento a usuarios y validación del software en operación
 - c) Una estrategia de procesamiento en paralelo es incompatible con una estrategia paulatina.
 - d) Una estrategia de adopción paulatina no permite la migración de datos entre el sistema viejo y el nuevo (ya que el viejo se sigue usando).

11. Respecto a los prototipos desechables:

- a) No es necesario incluirlos en ningún plan, puesto que son desechables.
- b) **Conviene hacerlos tan rápido y barato como sea posible.**
- c) Deben ser construidos con calidad en el código y robustez desde el principio.
- d) Se debe incluir detalles como validaciones y manejo de errores.

12. Dada la siguiente gráfica de trabajo pendiente para un sprint de un proyecto que sigue Scrum (en donde la línea recta señala el trabajo pendiente planificado y la línea quebrada señala el trabajo pendiente real),



¿cuáles de las siguientes opciones podrían interpretar correctamente la situación señalada?

- I. Se estimó mal algo y se volvió reestimar.
- II. Se aumentó el alcance del sprint.
- III. Se agregó personal para trabajar en el sprint.
- IV. La velocidad estaba bastante bien estimada y se desarrolló de acuerdo con lo planificado.
- V. Veníamos bien, pero la entrega de ciertos componentes se atrasó, lo que nos detuvo unos días, y luego continuamos.

Seleccione la opción correcta:

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

13. Una de las técnicas vistas en el curso para priorizar requisitos es:

- a) comparación por pares, en la que se les pide a dos clientes que hagan un ranking general de todos los requisitos.
- b) una escala de tres dimensiones: importancia, urgencia y estabilidad.
- c) el método MoSCoW, en el que se asignan a los requisitos uno de cinco niveles posibles de prioridad: debe incluirse, debería incluirse, podría incluirse, se incluirá solo si da el tiempo, no se incluirá.
- d) priorización basada en distintas variables como beneficios, costos y riesgos, a las que se puede incluso asignar pesos relativos.

14. Con respecto al análisis de los requisitos:

- a) De las actividades que forman parte del proceso de análisis de los requisitos, solo la negociación con el cliente es aplicable en metodologías ágiles; todas las demás se utilizan solo en procesos orientados por planes.
- b) Los modelos que se construyan tienen que ser siempre correctos y completos.
- c) Las actividades de análisis de los requisitos se realizan después de obtener todos los requisitos y antes de especificarlos.
- d) El análisis de requisitos incluye descubrir las fronteras del software y cómo debe interactuar con las plataformas de hardware y software existentes, así como también con los procesos de negocio de la organización.

15. La siguiente tabla establece las actividades para un proyecto dado, con sus duraciones y las relaciones de precedencia entre ellas.

Tarea	Estimación (Horas)	Precedencia
A	8	No tiene
B	4	No tiene
C	2	A
D	0	C y B
E	6	D
F	14	B y D
G	24	E
H	8	G

¿Cuál es la duración mínima del proyecto?

- a) 18
- b) 24
- c) 42
- d) 48

16. Sobre los riesgos en un proyecto de software:

- a) Un riesgo es un evento que puede tener un efecto positivo o negativo sobre algún objetivo del proyecto.
- b) Cuando el riesgo es negativo y se materializa se dice que es una amenaza.
- c) La severidad de un riesgo depende del grado de tolerancia al riesgo de la organización.
- d) Una de las estrategias para un riesgo positivo es mejorar, que refiere a eliminar la incertidumbre asegurando que la oportunidad se concrete.

17. En el ámbito de la gestión de personal, la teoría de Maslow plantea

- a) 5 niveles jerárquicos, donde el más alto corresponde a las personas con mayores ingresos económicos. Si no se cuenta con ingresos altos, no se puede llegar a ese nivel.
- b) 5 niveles de necesidades, donde luego de satisfacer las necesidades de un nivel inferior se puede pasar al siguiente nivel.
- c) 5 niveles, donde debe satisfacerse el nivel más alto para cumplir el de abajo.
- d) 3 posibles estados que son: insatisfacción, no satisfacción y satisfacción; y lo que determina el moverse dentro de esos estados son los factores de higiene y los de motivación.

18. Movie FING es un proyecto que surge por iniciativa de estudiantes aficionados al cine de la Facultad de Ingeniería y el Instituto Escuela Nacional de Bellas Artes, el proyecto tiene como objetivo fomentar el acceso al cine no comercial dentro del ámbito estudiantil y académico mediante el uso de un sistema que facilite el intercambio colaborativo (no streaming) de películas entre estudiantes. Para iniciar el proyecto, el grupo de estudiantes no tiene presupuesto pero cuentan con un total de 500 películas en formato digital cada una de ellas particionada en archivos de 300 MB distribuidos en sus máquinas. Esperan luego de los 2 meses haber ampliado el grupo y que los participantes puedan intercambiar unos 1.000 títulos. Dentro de los estudiantes de Fing, hay estudiantes avanzados en Ingeniería en computación que realizan algunas sugerencias para el proyecto. Si ud. fuera parte de dicho grupo, entiende que:

- a) se debe hacer uso del patrón cliente-servidor multinivel donde se cuente con un servidor para la interfaz del sistema y otro para el almacenamiento de todas las películas para permitir el escalamiento del sistema.
- b) se debe hacer uso del patrón maestro-esclavo donde se cuente con un servidor que oficie de nodo maestro que se encargue de la coordinación y los nodos esclavo adquieren los datos para facilitar la visualización de las películas en tiempo real.
- c) se debe hacer uso del patrón peer-to-peer semicentralizado donde uno de los participantes (a veces llamado super-peer) tenga el rol de contener la información que permita establecer conexiones entre los distintos peers para que pueda dar respuesta al intercambio.
- d) Ninguna de las opciones anteriores propone un patrón razonable para ser utilizado en el problema planteado.

19. Recientemente ha ingresado como arquitecto principal en una empresa de desarrollo de software y le asignaron como proyecto la evolución de un sistema de tickets para gestionar los incidentes y solicitudes. Los incidentes y las solicitudes se registran en el sistema en formato de ticket a partir de un correo electrónico. Los tickets que se registran en el sistema deberían contener el grupo que debe atender el incidente y la organización que está reportando. Sin embargo, a partir de los datos contenidos en el mail no es posible asignar de forma determinista (por ejemplo usando palabras clave) estos datos del ticket. Para dar respuesta a esta necesidad, se creó un modelo el cual a partir del asunto, descripción y dominio del mail predice el grupo y organización. La siguiente etapa es integrar el modelo al sistema de tickets para asignar los datos faltantes y automatizar la asignación de los tickets. Como arquitecto, indique qué consideraciones y cuidados debe tener:

- I. debe tener en cuenta los requisitos no funcionales y en particular la posibilidad de que entren en conflicto al momento de definir la arquitectura.
- II. en la integración del modelo al sistema, una solución adecuada es hacer uso del patrón de tubería y filtro para identificar y luego modificar los datos del ticket.
- III. debe tener en cuenta hacer uso del Modelo 4+1 ya que es un modelo que contempla todas las vistas de una posible arquitectura de software.
- IV. debe contar con un modelo arquitectónico para facilitar las discusiones sobre el diseño del sistema y también para documentar la arquitectura diseñada.
- V. en la integración del modelo al sistema, una solución adecuada es usar una arquitectura en capas para proporcionar el agregado de la nueva funcionalidad contemplando desde la presentación hasta el acceso a los datos.
- VI. explicitar el diseño y documentar la arquitectura tiene las siguientes ventajas: favorece la comunicación entre los distintos *stakeholders*, el análisis del sistema y el reuso a gran escala en sistemas con similares requisitos.

Seleccione la opción correcta:

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

20. Sobre la ingeniería de software basada en componentes (CBSE):

- I. La ingeniería de software basada en componentes (CBSE) es el proceso de definir, implementar e integrar o componer, componentes independientes en sistemas.
- II. Al integrar componentes independientes no se deben considerar problemas de incompatibilidad entre ellos debido a que fueron diseñados para ser reutilizados.
- III. A la hora de integrar o componer componentes se pueden tomar distintas estrategias como ser: composición secuencial, composición aditiva y composición jerárquica.
- IV. Hay dos tipos de procesos de CBSE: desarrollo para el reuso que se encarga de buscar e integrar componentes reutilizables, y desarrollo con reuso que desarrolla componentes o servicios que se reutilizarán.
- V. Las interfaces provistas de un componente definen qué servicios deben ofrecer otros componentes para que este componente opere correctamente.
- VI. Se cuenta con dos componentes, uno que recolecta datos de determinados sensores y otro que se encarga de mostrar datos en un *display*. Para que los datos de los sensores se puedan ver en el *display* se recomienda realizar una composición jerárquica entre ambos.

Seleccione la opción correcta:

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

21. Imagine que es parte del equipo de SCM de una empresa enfocada en el desarrollo de sistemas de Inteligencia Artificial basados en visión por computadora (área de investigación que explora el reconocimiento y comprensión de imágenes y video). En particular se encuentra trabajando en el desarrollo de un software para el control de vehículos autónomos. El software debe procesar de forma muy rápida muchas imágenes de elementos con los que se encuentra el vehículo en su recorrido (peatones, otros vehículos, etc) para tomar decisiones de forma autónoma. Los requisitos son analizados en Inglaterra y el desarrollo del software se realiza en Uruguay. La primera versión del software será utilizada en vehículos que circularán en Uruguay. Indique cuáles de las siguientes definiciones le parecen acertadas, dado este contexto:

- I. El sistema de control de versiones debe dar soporte a un riguroso flujo de trabajo que permita tener un adecuado control de versiones de todos los componentes que conforman el software.
- II. Es aconsejable, debido a la criticidad del software, definir un comité de control de cambios (change control board - CCB) el cual dentro de sus responsabilidades tendrá que evaluar y decidir si el cambio es justificado, y priorizar los cambios para su implementación.
- III. Dadas las características del software, para su armado es recomendable utilizar un enfoque de integración continua y en lo posible de entrega (*deploy*) continuo así el vehículo siempre tiene la última versión.
- IV. Para solicitar cambios es recomendable utilizar un sistema de tickets que de soporte a un proceso de aprobación de cambios del cual participe solo el equipo de desarrollo debido a que son los expertos en el sistema a desarrollar.
- V. Es conveniente contar como ítem de la configuración con un glosario que contenga los principales términos que pueden generar confusión por las posibles interpretaciones que puedan tener en diferentes lugares.

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