

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 Dadas las siguientes afirmaciones respecto a las cualidades del software (según Ghezzi):
 - a) El atributo visibilidad es un atributo de calidad del proceso y refiere a que se documentan con claridad sus pasos y estado actual.
 - b) a) y oportunidad también es un atributo de calidad del proceso y refiere a la habilidad de entregar un producto a tiempo.
 - c) b) y productividad es una cualidad del proceso de producción del software, mide la eficiencia del proceso.
 - d) c) y portabilidad consiste en que el sistema puede ser ejecutado en distintos ambientes, tanto sean estos plataformas de hardware como ambientes de software.
2. Respecto a los modelos de proceso:
 - a) Cuando los requisitos no son claros, no es una buena oportunidad para aplicar metodologías ágiles.
 - b) Si la aplicación es crítica, se recomienda tener un mayor grado de formalismo.
 - c) Al trabajar con una metodología ágil se pone énfasis en un proceso de desarrollo que permita la libertad de opinión del equipo por sobre la documentación.
 - d) El Manifiesto por el Desarrollo Ágil de Software destaca el contar con resultados de valor por sobre las personas y sus interacciones.
3. En cuanto a los distintos modelos de proceso:
 - a) El proceso en cascada resulta adecuado para proyectos de larga duración y alto riesgo.
 - b) Entre las principales ventajas del modelo de proceso en fases con evaluaciones internas parciales están que permite reducir el plazo de salida al mercado (*time to market*).
 - c) Una de las principales ventajas de un proceso en fases con evaluaciones parciales es que los usuarios puede contar con resultados útiles en un plazo más corto.
 - d) Un modelo de proceso en fases con liberaciones parciales en producción tiene como ventajas que usuarios y cliente tienen la oportunidad de evaluar antes el producto, el cliente puede contar con resultados útiles en un plazo más corto.
4. Respecto al estudio de factibilidad:
 - a) El equipo de desarrollo estudia si es posible y conveniente construir el producto solo con respecto a restricciones de costo y tiempo.
 - b) a) y en el documento de informe de factibilidad se recomienda si seguir con el sistema proponiendo, por ejemplo, cambiar el alcance y el presupuesto.
 - c) Se evalúan, para una omás alternativas, distintos aspectos, como, por ejemplo, el marco jurídico.
 - d) Ninguna opción es correcta.
5. Con respecto a los requisitos:
 - I. Por lo general, se suele describir los requisitos en diferentes niveles de detalle ya que diferentes lectores pueden usarlos de distinta forma.
 - II. Los requisitos de usuario se presentan en un documento estructurado que establece descripciones detalladas de las funciones, servicios y restricciones operacionales del sistema.
 - III. La imprecisión en la especificación de los requisitos es una de las principales causas por las cuales muchos proyectos de software fracasan.
 - IV. En sistemas grandes y complejos resulta difícil lograr que los requisitos sean consistentes y completos. Esto se debe a que, por ejemplo, es fácil cometer algún error u omisión. También suele suceder que diferentes *stakeholders* (interesados) pueden tener necesidades distintas e incluso inconsistentes entre sí.
 - V. Un requisito no funcional, como por ejemplo un requisito de seguridad, puede generar un conjunto de requisitos funcionales relacionados, así como restringir otros requisitos existentes.

Seleccione la opción correcta:

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

6. Respecto a las métricas de software:
- Son una característica de un producto o proceso de software que puede ser medida objetivamente, pero no es una característica de un documento
 - a) y un ejemplo de métrica es el número de incidentes detectados.
 - Las métricas del producto se pueden categorizar en estáticas y dinámicas
 - c) y las métricas estáticas son mediciones sobre aspectos del producto como el diseño, el código y la documentación.
7. Sobre los estándares de calidad de productos de software indique la opción incorrecta:
- Son útiles para evaluar y mejorar el proceso de construcción del software
 - Ayudan a definir los requisitos de calidad de un producto de software
 - Su aplicación puede considerar adecuaciones según el contexto
 - Son útiles para evaluar un producto de software antes y luego de su liberación
8. Con respecto al Documento de Requisitos de Software (SRS):
- El nivel de detalle del SRS depende del modelo de proceso utilizado.
 - Si se construye un sistema crítico, entonces el SRS debería ser detallado ya que hay que considerar los aspectos de seguridad y protección (*safety*).
 - Si el sistema va a ser desarrollado por equipos geográficamente distribuidos de una empresa, es recomendable que el SRS sea detallado y preciso.
 - Si el sistema se desarrolla en forma incremental, el SRS típicamente tendrá menos detalle que si se desarrollara sin incrementos.
 - Si el software a desarrollar forma parte de un sistema que incluye interacción con distintos sistemas de software y hardware, por lo general resulta importante definir los requisitos de forma detallada.
 - Las metodologías ágiles argumentan que, dado que los requisitos cambian muy rápido, el SRS deja de estar actualizado en cuanto se termina de escribir el mismo.
- Seleccione la opción correcta:
- Todas son correctas.
 - Sólo I, III, V, VI son correctas
 - Sólo I, III, IV, y VI son correctas
 - Sólo II, IV, V son correctas
9. Con respecto a los requisitos:
- Los requisitos no funcionales tienen un alto impacto en la definición de la arquitectura del sistema.
 - No cumplir con los requisitos no funcionales críticos incluidos en el alcance puede resultar en un software que no se pueda usar.
 - El tiempo necesario de entrenamiento puede ser utilizado para medir la confiabilidad.
 - El requisito: «El lenguaje de programación debe ser PHP», es un requisito funcional.
 - La técnica de casos de uso resulta de gran ayuda para identificar requisitos funcionales.
- Seleccione la opción correcta:
- Todas son correctas.
 - Solo (I), (II) y (V) son correctas.
 - Solo (III) y (V) son correctas.
 - Solo (I) y (IV) son correctas.
10. Dadas las siguientes afirmaciones respecto a la construcción de software:
- Se debe tratar de minimizar la complejidad del software para que sea más fácil de entender, así como para probarlo.
 - Una buena práctica es considerar que el código puede ser reutilizable, así como reutilizar código ya existente.
 - Algunas técnicas tales como: análisis estático, revisiones de código e inspecciones; pueden ayudar a mejorar la calidad del código.
 - Todas las anteriores son correctas.
11. Acerca de la evolución del software indique la opción correcta.
- Tiene lugar cuando cambian los sistemas de software existentes para satisfacer nuevos requerimientos. Los cambios son continuos y el software debe evolucionar para seguir siendo útil.
 - Es el proceso de comprobar que el sistema cumple con su especificación y que satisface las necesidades reales de los usuarios del sistema.
 - Es un modelo de proceso genérico moderno que está organizado en fases (concepción, elaboración, construcción y transición).
 - El cambio es totalmente evitable en los sistemas complejos. Para esto, debe aplicarse una metodología que incluya creación de prototipos y entregas incrementales.

12. Con respecto a los requisitos
- Es necesario gestionar los cambios de los requisitos a partir de que son relevados, sin que esto implique un control formal en todo momento.
 - Los requisitos de software son importantes únicamente al inicio del desarrollo de un sistema.
 - Entre las técnicas de obtención de requisitos se tienen: casos de uso, tormenta de ideas, revisiones y encuestas.
 - Un requisito es verificable si se puede demostrar formalmente.

13. Con respecto a las estimaciones en los proyectos de software:

- Las técnicas de estimación basadas en la experiencia permiten estimar el esfuerzo de un proyecto en base a las experiencias previas del estimador y a su conocimiento del dominio de la aplicación.
- Cuando se utilizan técnicas de estimación basadas en la experiencia, por lo general resulta conveniente que las estimaciones sean realizadas por un grupo de personas. Esto se debe a que, si cada uno explica su estimación al resto del grupo, muchas veces se encuentran factores que otros no habían considerado.
- En la fase inicial de los proyectos, las estimaciones por lo general tienen un alto margen de incertidumbre. Sin embargo, en la medida que el proyecto avanza, dicha incertidumbre irá disminuyendo, incluso en proyectos con requisitos inestables o tecnología desconocida.
- Se puede estimar con precisión aún si no se sabe quién va a trabajar en el proyecto. Esto se debe a que la productividad personal por lo general no varía en más de un 15 %.
- Los proyectos de software presentan el problema de deseconomía de escala. O sea, el esfuerzo por lo general crece exponencialmente con el tamaño.

Seleccione la opción correcta:

- Solo (I), (II) y (V) son correctas.
- Solo (I), (II), (IV) y (V) son correctas.
- Solo (III) y (IV) son correctas.
- Todas son correctas.

14. Con respecto a la gestión de riesgos en los proyectos de software:

- El proceso de gestión de riesgos es un proceso iterativo que se realiza únicamente en las etapas iniciales del proyecto.
- La gestión de los riesgos tiene como objetivo aumentar la probabilidad y el impacto de los eventos positivos, y disminuir los de los adversos.
- La técnica de análisis del valor monetario (EMV) sirve para realizar un análisis cuantitativo de los riesgos.
- Aceptar una amenaza implica identificar el riesgo y decidir no tomar medidas a menos que el riesgo se materialice.
- Si se agregan actividades al proyecto para reducir los riesgos, estas actividades deberían ser consideradas en el WBS.

Seleccione la opción correcta:

- Todas las opciones son correctas.
- Sólo (I), (III) y (IV) son correctas.
- Sólo (II) y (V) son correctas.
- Sólo (II), (III), (IV) y (V) son correctas.

15. Según Humphrey, el Plan de Calidad debe incluir:

- la descripción del producto, del mercado y de la calidad esperada.
- planes del producto: fechas de liberación de versiones, responsables del producto, planes de distribución del producto.
- descripción de procesos para el desarrollo y la gestión del producto.
- los atributos de calidad más importantes del producto
- la gestión de los riesgos claves que pueden afectar la calidad del producto.

Señale la opción correcta:

- Todas las opciones son correctas.
- Solo (IV) es correcta.
- Solo (III) y (V) son correctas.
- Solo (I), (II), (III) y (IV) son correctas.

16. La gestión de la configuración de un producto de software comprende cuatro actividades que están muy relacionadas entre sí:

- La gestión del cambio, la gestión de versiones, la gestión del conocimiento y la obtención de requisitos.
- La obtención de requisitos, la gestión de las entregas, el armado del sistema y la gestión de conocimiento.
- La gestión de entregas, el armado del sistema, la gestión de versiones y la gestión del cambio.
- Ninguna es correcta.

17. Dadas las siguientes afirmaciones respecto al diseño de software indique la correcta:
- Dividir y conquistar, sugiere dividir el producto a construir en partes más pequeñas para facilitar el diseño. En general esta tarea es simple.
 - Abstracción permite al diseñador considerar el sistema a más bajo nivel considerando los detalles de cada componente.
 - Modularidad es la característica de un sistema en el cual los componentes se pueden implementar separadamente y el cambio en uno tiene impactos mínimos en el resto.
 - c) y si el sistema es modular, probablemente tenga alto acoplamiento.
18. La empresa en donde trabaja fue encargada del desarrollo y mantenimiento de una aplicación web, la cual fue liberada hace un año. Hace dos días se liberó una nueva actualización del navegador Firefox y a partir de eso se percibieron en la aplicación problemas en la visualización de algunas funcionalidades. Dado el alto uso de la aplicación en Firefox, se catalogó la falla como grave.
- Respecto al mantenimiento de software:
- Para reparar la falla de forma rápida, se elige una solución rápida y viable que puede no ser la mejor solución técnica. Luego, la implementación del cambio de forma correcta debería agregarse como una petición de cambio que recorra el ciclo formal.
 - Se sigue un proceso formal que incluye actividades como análisis del impacto, planeación de la versión, implementación del cambio y liberación de nueva versión.
 - Un riesgo es que el cambio no quede documentado en los requisitos y en el diseño.
 - El mantenimiento realizado es correctivo.
 - El mantenimiento realizado es adaptativo.
- Seleccione la opción correcta:
- Solo las afirmaciones I) y IV) son correctas
 - Solo las afirmaciones II), III) y V) son correctas
 - Solo las afirmaciones I), III) y IV) son correctas
 - Solo las afirmaciones II) y IV) son correctas.
19. Son factores a considerar al momento de decidir reutilizar componentes de software:
- El costo de evaluar su utilidad y pertinencia
 - El nivel de dificultad asociado a la curva de aprendizaje de los componentes
 - El nivel de reutilización que ofrece (objeto, componente, sistema, etc.)
 - Todas las opciones son correctas
20. Para evaluar las características de un proceso pueden formularse, entre otras, las siguientes preguntas:
- ¿Está el proceso explícitamente definido y es comprensible para todos los actores?
 - ¿Son visibles todas las métricas del proceso durante su ejecución por todos los actores?
 - ¿Tiene madurez?
 - Todas las anteriores.
21. Sobre el diseño de un sistema
- Es correcto cuando considera todos los requisitos funcionales.
 - Cuando uno está utilizando metodologías ágiles no es necesario realizar un diseño de la solución, ya que el foco está puesto en el producto.
 - Son características de un buen diseño: el alto acoplamiento y baja cohesión.
 - Debe considerar las restricciones asociadas a los requisitos, así como también el ambiente donde el sistema ejecutará.
22. Un ejemplo de mantenimiento preventivo es:
- Un cambio que se solicita debido a una nueva ley.
 - Un cambio que se solicita debido a la modificación de una ley anterior.
 - Una corrección a un error encontrado por el usuario final, luego de que el sistema ya se encuentra productivo.
 - La refactorización, ya que implica hacer cambios en los programas sin alterar su funcionalidad.
23. Respecto a las pruebas de caja negra:
- Se parte de los requisitos y/o especificaciones, considerando su implementación para crear casos de prueba más completos
 - a) y una técnica es la de partición en clases de equivalencia
 - b) y una clase de equivalencia es un conjunto de entradas para las cuales suponemos que el software se comporta igual.

- d) Ninguna opción es correcta.
24. Con respecto a la gestión de la configuración:
- a) En proyectos que son llevados a cabo por una única persona no se necesita hacer un manejo de la gestión de la configuración.
 - b) (a) y gestionar la configuración es esencial para proyectos con equipos, ya que varios desarrolladores pueden estar trabajando al mismo tiempo en el sistema de software.
 - c) (b) y la gestión de la configuración de un sistema de software implica el control de cambios como una actividad fundamental.
 - d) Al gestionar la configuración de un sistema de software por lo general se tiene que tratar con un gran volumen de información. Por tal motivo, se han desarrollado muchas herramientas para dar soporte a esta actividad.
25. Según lo visto en el curso, verificación puede definirse cómo:
- a) Comprobar que el sistema cumple con los requisitos especificados (funcionales y no funcionales)
 - b) Lograr que el software deba cumplir una especificación.
 - c) Proceso de evaluar un sistema o componente, con el fin de determinar si los productos de cierta fase de desarrollo, satisfacen las condiciones impuestas al comienzo de dicha fase
 - d) Todas las opciones con correctas
26. El testeo puede demostrar presencia de errores, pero no la ausencia de ellos
- a) Verdadero
 - b) Falso
 - c) Solo cuando se definen pruebas exhaustivas
 - d) Solo a través de una combinación determinada de pruebas de caja negra y caja blanca
27. ¿Cuál de las siguientes no es un tipo de falta?:
- a) De capacidad de procesamiento o de desempeño
 - b) De recuperación
 - c) De documentación
 - d) De estimación de esfuerzo
28. Sobre el proceso de Verificación y Validación:
- a) Considera pruebas unitarias, de módulos, de integración, de aceptación, entre otras
 - b) Se comienza una vez definido el alcance del proyecto
 - c) (a). Donde las pruebas de integración ponen foco en verificar las funciones de los componentes
 - d) (a) y (b) son correctas
29. Respecto a gestión de proyectos, dada la siguiente planificación (en recuadro en blanco) y avance (en gris), habiendo finalizado la semana 3, y con las tareas A y C en el camino crítico,

Actividad	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7
A							
B							
C							
D							

- Seleccione la opción correcta:
- a) El proyecto está adelantando.
 - b) El proyecto está atrasado
 - c) No se pueden sacar conclusiones con esta información.
 - d) El adelanto en C compensa por el atraso en A.
30. Relacionado al diseño de software:
- a) Debe ser verificable, completo y amigable para el desarrollador
 - b) Puede verse como una tarea básicamente creativa, donde suelen tomarse decisiones de compromiso (*trade-off*)
 - c) La baja cohesión y el alto acoplamiento son dos de los conceptos claves para lograr un diseño de calidad
 - d) Dos opciones son correctas