

- Cada pregunta **múltiple opción** contestada correctamente tiene un valor de 2 puntos. Cada pregunta incorrecta de la múltiple opción resta 0,66 puntos.
- La evaluación es de carácter individual y la duración es de 2 horas.
- **El puntaje total del parcial es de 40 puntos.**

Múltiple Opción

1. En lo visto en el curso sobre los productos genéricos y los productos específicos de software:
 - a) Los productos genéricos corresponden a sistemas autónomos que generalmente se comercializan o distribuyen a cualquier cliente que desee comprarlos o usarlos.
 - b) Los productos personalizados corresponden a software que es encargado por un cliente específico para satisfacer sus propias necesidades.
 - c) La distinción entre productos genéricos y productos específicos radica en quién controla la especificación.
 - d) Todas las anteriores son correctas.
2. En relación a los requisitos:
 - a) Un atributo de calidad es un tipo de requisito no funcional que describe una característica de servicio o desempeño de un producto.
 - b) Los requisitos no son completos cuando existen conflictos o definiciones contradictorias entre ellos.
 - c) Un ejemplo de requisito no funcional verificable es: La aplicación tiene que ser amigable y fácil de usar.
 - d) Dos de las anteriores son correctas.
3. Respecto a la realidad planteada en el Obligatorio 1:
 - a) El caso de uso “Ingresar a la aplicación” es iniciado por el usuario estudiante.
 - b) (a) y tiene otro actor que es el Sistema a construir.
 - c) (b) El caso de uso “Aplicar a una oferta de empleo” extiende al caso de uso “Buscar ofertas de empleo”.
 - d) Todas las opciones son correctas.
4. Respecto a la realidad planteada en el Obligatorio 1:
 - I. La siguiente especificación: “El sistema debe tener una usabilidad muy buena (alguien que no lo conozca puede usarlo sin manuales o entrenamiento)” describe un requisito no funcional no verificable.
 - II. Algunos requisitos funcionales son: Agregar carrera del estudiante para recibir ofertas de empleo, registrar empresa y crear una oferta de empleo.
 - III. El envío automático de ofertas de empleo a través de correo electrónico describe un requisito no funcional verificable.
 - IV. Dado que se planea una primera liberación, no es necesario validar los requisitos.
 - V. La técnica de prototipado es adecuada para relevar los requisitos.

Seleccione la opción correcta:

 - a) Sólo dos opciones son correctas.
 - b) Sólo (I), (II), (III) y (IV).
 - c) Sólo (I), (II) y (V).
 - d) Sólo (III), (IV) y (V).
5. Respecto a las historias de usuario:
 - a) Incluir los beneficios en la redacción de una historia de usuario (ficha) permite al equipo de desarrollo plantear alternativas que cumplan con el mismo propósito.
 - b) Uno de los motivos por el cual una historia de usuario es difícil de estimar puede ser la falta de conocimiento técnico.
 - c) Una historia épica es INVEST
 - d) Dos opciones de las anteriores son correctas.
6. Respecto a los requisitos en el marco de metodologías ágiles:
 - a) Generalmente se escriben casos de uso porque de esta forma el usuario visualiza su interacción con el sistema, como una estructura de pasos ordenados que cubre tanto escenarios de éxito como condiciones de excepción.
 - b) Generalmente se escriben historias de usuario porque permiten al desarrollador abordar pequeñas porciones de una funcionalidad y a la vez derivar sus pruebas de aceptación.
 - c) (b) y ayudan al desarrollador a pensar en posibles variaciones de la historia de usuario, condiciones de excepción y requerimientos no funcionales asociados.
 - d) Generalmente se escriben historias de usuario porque permiten al desarrollador abordar en pequeñas porciones de una funcionalidad sin distraer la atención en condiciones de excepción.

7. Respecto a los modelos estructurales en ingeniería de requisitos:
- Son muy útiles como documentación del sistema.
 - Sirven como punto de partida para la definición de la arquitectura del sistema.
 - Deben especificarse con un alto nivel de detalle para evitar errores en la implementación.
 - Se debe utilizar el estándar de notación UML.
 - Al inicio de la fase de requisitos son muy útiles para representar entidades del negocio a nivel conceptual.
- Seleccione la opción correcta:
- Todas las opciones son correctas.
 - Solo (I), (III) y (V) son correctas.
 - Solo (I), (II) y (V) son correctas.
 - Solo (I), (II), (IV) y (V) son correctas.
8. Sobre la afirmación "El diseño de software es una actividad creativa donde se identifican los componentes del software y sus relaciones, con base en los requerimientos de un cliente"
- No considera la identificación de requisitos no funcionales que son relevantes para el diseño y arquitectura de un sistema de información.
 - Y además considerando el diseño como un proceso, es una actividad que produce una descripción interna del software que será insumo para otras actividades y/o procesos.
 - (b). A su vez, el diseño correcto se logra a través de la ejecución de un procedimiento bien definido y no ambiguo.
 - La afirmación no es correcta ni incorrecta ya que presenta ambigüedades en cuanto a la definición de "diseño de software"
9. 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.
 - Abstracción permite al diseñador considerar un componente sin preocuparse por los detalles de implementación (por ej. describiendo comportamiento exterior sin describir los detalles internos).
 - 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.
 - Todas las anteriores son correctas.
10. Respecto a diseño de sistema:
- El proceso de definición de la arquitectura es un proceso creativo que varía para diferentes sistemas.
 - a) donde los requerimientos no funcionales son elementos relevantes para dicho proceso.
 - b) por lo que deben ser relevados al inicio del proceso de relevamiento de requisitos.
 - Ninguna de las anteriores es correcta.
11. En relación a la ingeniería de software basada en componentes:
- Un componente es una entidad ejecutable independiente que puede estar compuesta por uno o más objetos ejecutables.
 - (a). Por lo general, un componente es estandarizado, dependiente del contexto e implementable.
 - Las interfaces de un componente definen qué servicios deben proveerle otros componentes para que este componente opere correctamente.
 - (a) y (c) son correctas.
12. Indique cual de las siguientes NO es una ventaja del sistema arquitectónico Tubos y Filtros
- Es intuitivo para aquellos casos en que se puede pensar la aplicación como secuencias de procesamientos de datos.
 - Es sencillo agregar nuevas transformaciones de forma que el sistema evolucione.
 - Es posible reutilizar los filtros
 - Está especialmente indicado para utilizarse en la construcción de Sistemas Interactivos.
13. Según Sommerville, son ventajas de una arquitectura explícita:
- Mejorar la comunicación con los stakeholders.
 - Validar el cumplimiento de los requisitos no funcionales.
 - No favorece la construcción.
 - Dos de las anteriores son correctas.

14. En la construcción de software se hace uso de varios tipos de tecnologías. Acerca de los modelos ejecutables puede decirse que:
- corresponden a un conjunto de firmas que son exportadas y disponibles para que los usuarios de una biblioteca o framework puedan escribir sus aplicaciones.
 - (a) y sirven para poder acceder a librerías sin conocer detalles de la implementación.
 - una especificación construida en un modelo ejecutable puede ser desplegada en varios entornos de ejecución sin realizar cambios.
 - c) y un ejemplo es BPMN.
15. Del *test driven development* (TDD) podemos afirmar que :
- es una metodología de construcción de software que requiere escribir los casos de prueba antes que escribir el código del programa
 - es una estrategia de pruebas que requiere diseñar todos los casos de prueba antes de comenzar a diseñar la arquitectura del sistema
 - promueve que los programadores piensen en los requerimientos y en el diseño antes comenzar a codificar.
 - permite detectar defectos más temprano que otros estilos más tradicionales.
 - logra que los programadores, que trabajan de a pares, compartan conocimiento de forma exitosa y a bajo costo.

Seleccione la opción correcta:

- I y V son correctas.
 - II, III y IV son correctas.
 - Ninguna es correcta.
 - I, III y IV son correctas.
16. En cuanto al desarrollo incremental:
- El desarrollo incremental se basa en crear ciclos cortos de desarrollo (unas pocas semanas), lo cual puede incluir validar el software desarrollado de cada iteración (con el cliente/usuarios/stakeholders), para luego evolucionar el software a través de múltiples versiones.
 - (a) y por lo general resulta difícil mantener la trazabilidad de documentos entre una versión a otra.
 - El desarrollo incremental no es compatible con un enfoque basado en planes.
 - (c) y el desarrollo incremental es una parte fundamental de los métodos ágiles.
17. Con respecto a las actividades del proceso de desarrollo:
- El proceso de ingeniería de requisitos tiene 3 actividades básicas: el relevamiento y análisis, la especificación y la validación de requisitos.
 - Una de las actividades del proceso de diseño es el diseño de la arquitectura. Esta implica diseñar las estructuras del sistema de datos y cómo éstos serán representados/implementados en la base de datos.
 - En las metodologías ágiles el diseño se intercala con la implementación.
 - La verificación y validación involucra los procesos de comprobación, revisión y pruebas de sistema.
 - Si cambian las circunstancias del negocio por lo general se cambian los requisitos del software. En dicho caso, es deseable que el software evolucione y cambie acompañando esos cambios para no volverse inútil y obsoleto.

Seleccione la opción correcta:

- Todas son correctas.
 - Sólo (I), (II) y (V) son correctas.
 - Sólo (I), (III), (IV) y (V) son correctas.
 - Sólo (II), (III) y (IV).
18. Con respecto a las actividades del proceso de desarrollo:
- Sólo conviene definir el proceso de software a utilizar cuando el sistema a construir es de gran porte.
 - Al día de hoy, se sabe que el modelo de proceso en cascada no es aplicable a ningún tipo de software existente.
 - En el desarrollo incremental la especificación, el desarrollo y la validación se realizan en forma intercalada.
 - El modelo de procesos de integración y configuración refiere a que el sistema es "ensamblado" configurando e integrando componentes pre-existentes ya desarrollados.
 - En el desarrollo incremental el costo de acomodar los cambios es mayor que con cascada.

Seleccione la opción correcta:

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

19. Respecto a modelos de procesos:
- I. Las etapas del desarrollo en cascada son: definición de requerimientos, diseño de software y del sistema, implementación y pruebas unitarias, y operación y mantenimiento
 - II. Una desventaja del desarrollo incremental es que no se piensa el diseño de todo el sistema desde el inicio, lo que puede llevar a cambios costosos (re-trabajo y refactorización) en el "core" del sistema si los cambios a futuro no son soportados por la arquitectura ya construida
 - III. Una desventaja del uso del desarrollo en cascada es la limitante de adaptarse rápidamente a cambios en los requisitos.
 - IV. La autonomía, autogestión y compromiso de los integrantes del equipo de desarrollo es más importante en los enfoques dirigidos por planes que en los ágiles.
 - V. Algunas ventajas de desarrollo incremental vs cascada son: el costo de la implementación de los cambios es menor, se puede obtener retroalimentación del cliente de forma temprana y el cliente puede obtener tempranamente una versión parcial del producto que le genere valor.

Seleccione la opción correcta:

- a) Solo las afirmaciones II) y III) son verdaderas.
 - b) Solo las afirmaciones I), II) y III) son verdaderas.
 - c) Solo las afirmaciones II), III) y V) son verdaderas.
 - d) Todas las afirmaciones son verdaderas.
20. Seleccione la opción correcta
- a) La práctica de refactorización planteada en XP sostiene que debe implementarse pensando en el cambio para así evitar el retrabajo de considerarlo.
 - b) Una ventaja de test first development (o test driven development) es que permite detectar ambigüedades y omisiones en los requisitos previo a comenzar la implementación.
 - c) b) y la programación por pares tiene como ventaja que al realizar una revisión informal del código al menos por dos personas, esta colabora con el sentido de pertenencia y responsabilidad compartida del software.
 - d) c) y esta práctica (programación por pares) es menos eficiente que la programación individual.