

- 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. Respecto a la ingeniería de software:
 - I. Guarda relación con la disciplina “programación”, sin embargo, la ingeniería de software se ocupa de la construcción y uso de programas construidos para ser usados por otras personas y, más que de programas individuales se ocupa de sistemas de programas (conjunto de programas que interactúan entre sí) y de productos de software que pueden ser utilizados por personas ajenas al proceso de construcción.
 - II. Guarda relación con la disciplina Administración General y en especial con las sub-disciplinas Gestión de Proyectos y Gestión de Recursos Humanos.
 - III. La Gestión de Recursos Humanos resulta de gran importancia en la ingeniería de software dado que los recursos humanos constituyen el factor de producción que tiene mayor peso en la construcción de software.
 - IV. Está relacionada con la ingeniería de sistemas, dado que al construir software en general estamos construyendo a la vez un sistema compuesto al menos por software, hardware sobre el que ejecuta el software y personas que lo utilizan y es este sistema el que debe funcionar de forma adecuada.
 - V. Persigue la mejora de la productividad y calidad en la producción de software a través de la verificación exhaustiva del código antes de su puesta en producción. La verificación del código constituye una garantía de la calidad del mismo.
 - VI. Entre las nociones fundamentales que han sobrevivido la prueba del paso del tiempo cabe mencionar: prototipación, proceso de software, diseño minimal, mediciones, programación por pares y arquitectura de software.

Seleccione la opción correcta:

- a) Todas son correctas.
- b) I, IV.
- c) **I, II, III, IV.**
- d) II, III, IV, VI.

-
2. Un Comité de Control de Cambios es:
 - a) Un grupo de usuarios clave del cliente, quienes tienen la responsabilidad de aprobar los cambios.
 - b) Un grupo conformado por usuarios clave del cliente, cuya función es analizar el impacto de realizar un cambio sobre un sistema ya liberado.
 - c) Un grupo compuesto íntegramente por miembros del equipo de desarrollo, quienes tienen la potestad de proponer cambios que deberán ser aprobados por el Jefe de Proyecto.
 - d) **Un grupo integrado por representantes del cliente, los usuarios y los desarrolladores, cuya función es evaluar y autorizar o rechazar las solicitudes de cambios al sistema.**
-
3. Sobre cuándo comenzar la verificación de software:
 - a) En la última fase del desarrollo de software ya que antes no es posible realizar tareas de verificación.
 - b) **Desde el comienzo del proyecto de software se deberían realizar tareas de verificación de software.**
 - c) La verificación estática debería comenzar cuando se tiene el primer programa (código) escrito. La verificación dinámica (pruebas) debería comenzar luego de la primera compilación.
 - d) Cuando se libera la primera versión al equipo de pruebas de software.

-
4. La entrevista individual, como técnica de relevamiento de requisitos, se usa para:
 - I. Establecer una mejor relación con el cliente durante el relevamiento.
 - II. Entender el problema del negocio.
 - III. Entender el ambiente operativo.
 - IV. Evitar omisiones de requisitos.
 - V. Obtener datos estadísticos sobre preferencias del cliente.

Seleccione la opción correcta:

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

5. 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

5 - Especificación Operacional

6 - Modelo Transformacional

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.

- a) A3(2(1)), A3(1), A5, B3(1), B3(2(1)), C1, C4, C6
- b) A3(2(1)), A3(1), A4, B3(1), B3(2(1)), C1
- c) A1, A2(1), B1, B2(1), C1, C4
- d) **A3(2(1)), A3(1), B3(1), B2(1), C1, C2(1), C4**

6. Respecto a la liberación de software

a) **Entre las ventajas de las estrategias de implantación paulatina están que permiten:**

- **gestionar mejor los riesgos asociados a la implantación,**
- **ajustar los procedimientos asociados al sistema también de forma paulatina y**
- **distribuir el calendario de capacitación.**

Y entre sus desventajas están que:

- **las ventajas del uso del nuevo sistema se logran de forma paulatina y**
- **la coexistencia del sistema nuevo con el viejo suele plantear problemas.**

- b) La capacitación a los usuarios conviene que se haga lo más tarde posible, justo antes de la fecha de liberación porque constituye una instancia de validación del producto. Pero a su vez, conviene que se haga lo más temprano posible para que a la fecha de salida en producción los usuarios hayan tenido tiempo suficiente de madurar y asimilar los conocimientos.
- c) Las actividades que hay que considerar asociadas a una implantación de software son la capacitación a usuarios y la documentación del sistema y en aquellos casos en los que un sistema sustituye a uno anterior, la conversión del sistema anterior al nuevo.
- d) El procesamiento en paralelo es una técnica que se utiliza cuando un sistema sustituye a otro que está en funcionamiento y consiste en tener a ambos sistemas en funcionamiento a la vez para verificar si el sistema nuevo se comporta de forma consistente con el comportamiento del sistema anterior. Esta técnica resulta muy efectiva para detectar defectos del software al punto que permite prescindir de todo otro tipo de prueba del sistema.

7. Muchas veces surgen conflictos entre el equipo de desarrollo y el de pruebas. Para intentar solucionar los mismos es una buena práctica:

a) **Trabajar en equipo y poner el foco en evaluar productos y no personas.**

- b) Establecer reglas que indican que las pruebas servirán tanto para evaluar al producto como a las personas. Si se establece claramente cómo las pruebas afectarán a quien desarrolló el software, los problemas se evitan o solucionan.
- c) Rotar ambos equipos constantemente. Por ejemplo, una vez por cada proyecto (o por cierta unidad en meses de tiempo) los equipos rotan, pasando el equipo de desarrollo a ser el de pruebas y el de pruebas a ser el de desarrollo. Esto, además de evitar los conflictos, genera otro efecto positivo, todos saben cómo construir y cómo probar software.
- d) Usualmente el conflicto se debe a un desarrollador y/o un probador de software. Asignar a estas personas a otras tareas es una forma efectiva de solucionar el conflicto.

8. Respecto al mantenimiento:
- Frecuentemente se presenta un conflicto entre la necesidad de corto plazo de incorporar modificaciones al software de forma rápida y la necesidad de largo plazo de preservar atributos de calidad relevantes del producto y en particular su facilidad de mantenimiento. Este conflicto se ve agravado en caso de que el personal responsable por llevar a cabo el mantenimiento no cuente con un conocimiento acabado del producto que está manteniendo.**
 - La gestión de cambios de un producto de software que está en producción debiera ser idéntica a la gestión de cambios de un producto en desarrollo. Los cambios debieran ser aprobados por un Comité de Control de Cambios que integre la visión de los desarrolladores y la de cliente-usuarios. No resulta buena práctica definir un procedimiento de cambio excepcional con menos controles previos para dar respuestas rápidas frente a eventuales emergencias. En lugar de ello corresponde tomar todas las medidas que aseguren que no se produzcan situaciones de emergencia, entre ellas, llevar a cabo una adecuada verificación del software de forma previa a su puesta en producción.
 - El trabajo de mantenimiento es considerado en general por los gerentes de software como más importante que el trabajo de desarrollo, lo que incide favorablemente en la motivación del personal técnico.
 - Un sistema de tipo S (*Specifiable*), como por ejemplo un software para calcular la inversa de una matriz, suele requerir una fuerte carga de mantenimiento de tipo adaptativo, para adaptarlo a cambios en el ambiente. Si no se llevan a cabo esos cambios, este tipo de software será cada vez menos útil.

9. Respecto al tema pruebas de software:
- Una de las principales ventajas de TDD es que disminuye los costos de las pruebas de regresión.
 - Una desventaja de las inspecciones respecto a las pruebas, es que un error detectado puede estar enmascarando otros errores.
 - (a) En sistemas críticos más de la mitad de los costos de desarrollo corresponden a verificación y validación.**
 - Las especificaciones formales pueden asegurar la confiabilidad de un sistema.

10. Sobre las técnicas estáticas de verificación:
- Son técnicas que no involucran la ejecución de un programa.
 - Sirven para encontrar defectos antes de que esté pronta una primera versión ejecutable.
 - Se basan en casos de prueba que son generados a partir del documento de especificación de requisitos.
 - Ejemplos de técnicas estáticas son: Inspecciones, revisión personal de código y pruebas de integración basadas en meta-modelos del software.
 - Ejemplos de técnicas estáticas son: Verificación formal y análisis automatizado de programas.
 - Las técnicas estáticas son generalmente menos efectivas que las dinámicas.

Seleccione la opción correcta:

- Todas son correctas.
- Únicamente I, II y V son correctas.**
- Únicamente II, III, IV son correctas.
- Únicamente I, II, V y VI son correctas.

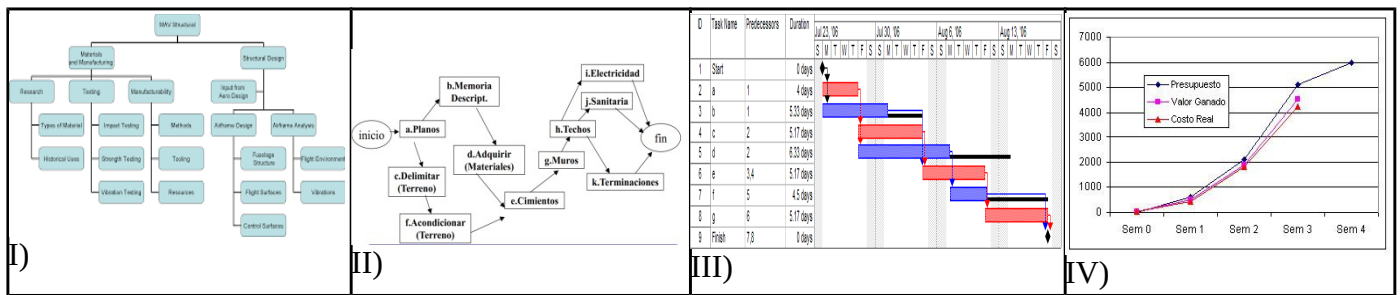
11. Con respecto al mantenimiento de software
- La gestión de la configuración resulta esencial en las etapas finales del desarrollo y es aún más importante durante el mantenimiento para, entre otras cosas, poder responder preguntas como las siguientes:**
 - ¿cuál es la versión de código fuente que corresponde al ejecutable en producción?
 - ¿cuáles son los requerimientos que cumple ese ejecutable?
 - ¿cuál es el diseño consistente con ese ejecutable?
 - ¿cuáles con los casos de prueba y los resultados esperados correspondientes a ese ejecutable?
 - ¿cuándo, quién y por qué llevó a cabo la última modificación del ejecutable en producción?
 - La calidad de un producto de software que se encuentra en producción y es sometido a intervenciones de mantenimiento tiende a crecer a medida que se acumulan intervenciones sobre el mismo ya sean estas para corregir defectos encontrados como para adaptarlo a nuevas necesidades, prevenir defectos o mejorar aspectos ya existentes.
 - El costo de mantenimiento de sistemas de tipo E (embebidos en la realidad que modelan), tiende a mantenerse estable una vez pasado el primer año de su puesta en producción.
 - Diversos estudios muestran que para sistemas de tipo E (embebidos en la realidad que modelan) el costo de mantenimiento durante toda la vida del sistema tiende a ser una pequeña fracción del costo de desarrollo.

12. En cuanto al Proceso de los Requisitos:
- Algunos atributos a considerar en la validación de requisitos son: claridad, consistencia, completitud, trazabilidad y verificabilidad.
 - a) y algunas técnicas para validar los requisitos son: revisión de requisitos, prototipación y generación de casos de prueba.**
 - a) y la obtención, análisis y especificación de los requisitos es un subproceso secuencial que comprende solamente las siguientes dos actividades: relevamiento de los requisitos y documentación..
 - Entrevistas es la técnica menos utilizada para el relevamiento de requisitos.

13. Son objetivos de la verificación y validación de software:
- Descubrir faltas, tanto sea inyectando defectos a través de pruebas o revisando el producto.
 - Evaluar la calidad del código fuente, programas, componente, sistema y/o documentos generados durante un proyecto de software para ver si el software se adecúa al uso pretendido del mismo.**
 - Establecer una aproximación de cuántos defectos cada mil líneas de código tiene el sistema.
 - Lograr que el equipo de desarrollo esté más distendido (tenga menos estrés), mediante la detección temprana de defectos,

14. La ingeniería de software empírica surge para suministrar evidencia sobre el valor que aporta la utilización de determinada metodología, para lo cual:
- Se experimenta la metodología sobre productos de software.
 - Se comparan los resultados de varios proyectos con la línea base previamente establecida.
 - a) y se recopila información en base a la cual se formulan o validan hipótesis.**
 - Todas las anteriores.

15. Dados los siguientes gráficos:



Seleccione la opción correcta:

- El gráfico I) representa un WBS. Esta técnica organiza y define el alcance TOTAL del proyecto (del producto y del proyecto)
- a) y el gráfico II) representa un grafo de actividades. Esta técnica no permite calcular la duración total del proyecto.
- a) y el gráfico III) representa un diagrama de Gantt. Un diagrama de Gantt es una herramienta gráfica que permite exponer el plazo de dedicación previsto para diferentes actividades a lo largo de un período de tiempo total determinado.**
- El gráfico IV) representa una gráfica de Valor Ganado. La técnica de Valor Ganado no es buena para conocer el avance real de un proyecto.

16. Con respecto a la Ingeniería de Requisitos:

- En teoría la especificación de los requisitos funcionales de un sistema debería ser completa y consistente. Sin embargo, en la práctica cuando se trata de sistemas grandes y complejos es muy difícil lograr esto debido a que es común cometer errores y omisiones al escribir la especificación de dichos sistemas.
- La seguridad, el desempeño y la disponibilidad son ejemplos de requisitos no funcionales.
- Los requisitos no funcionales son por lo general menos críticos que los requisitos funcionales individuales.
- El documento de requisitos idealmente debería incluir detalles relacionados a la arquitectura y diseño del sistema.
- Los requisitos del SRS son una versión ampliada de los requisitos del Documento de Requisitos y son utilizados por los ingenieros de software como punto de partida para el diseño de software.

Seleccione la opción correcta:

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

-
17. Con respecto al diseño de software:
- El diseño de software es una actividad creativa en la cual se identifican los componentes del software y sus relaciones en base a los requerimientos del cliente.
 - El diseño y la implementación están estrechamente ligados. Sin embargo la decisión sobre cómo documentar el diseño a realizar nunca se ve afectada por aspectos relacionados a la implementación tales como el lenguaje de programación a utilizar.
 - Los patrones de diseño son formas de describir buenas prácticas. Estos buscan capturar la experiencia de forma tal que otros puedan reusar esa experiencia.
 - Para describir un patrón de diseño resulta esencial especificar: el nombre del patrón, una descripción del problema a resolver, una descripción de la solución planteada y las consecuencias de aplicar el patrón.
 - El diseño de software puede ser visto como un proceso secuencial a seguir.

Seleccione la opción correcta:

- Todas son verdaderas.
 - Solo (II), (II) y (IV) son verdaderas.
 - Solo (I), (III) y (IV) son verdaderas.**
 - Solo (I), (III), (IV) y (V) son verdaderas. .
-
18. Con respecto al proceso de planificación de un proyecto:
- La identificación de restricciones y riesgos que afectan el proyecto es una tarea fundamental que debe llevarse a cabo al comienzo del proceso de planificación.**
 - a), mientras que la definición de hitos y entregables del proyecto puede realizarse una vez finalizada la planificación.
 - Luego de culminar el proceso de planificación, es esperable que el plan de proyecto resultante pueda seguirse y mantenerse invariante a lo largo del mismo.
 - c) y si ocurre un cambio que afecte la planificación del proyecto, entonces será necesario elaborar un nuevo plan que contemple los impactos correspondientes.

19. La madurez de una organización que construye software es función de:

- La capacidad del conjunto de procesos de la organización.**
- El desempeño del conjunto de procesos de la organización.
- La política de calidad definida para la organización.
- Ninguna de las anteriores.

20. Le piden desarrollar un sistema para la gestión de los activos de una red de farmacias. Se trata de un desarrollo a medida desde cero en Java. Una vez que tiene una versión estable de los requisitos, desea hacer una estimación del esfuerzo necesario para desarrollar el sistema utilizando COCOMO II, para ver si es factible desarrollar el proyecto en los plazos dados. ¿Qué submodelo de COCOMO II utilizaría?

- El modelo de composición de aplicaciones.
- El modelo de diseño temprano.**
- El modelo de reuso.
- El modelo posarquitectura.

21. Indique cual de las siguientes NO es una ventaja del sistema arquitectónico Filtros y Tubos:

- Es intuitivo pensar las aplicaciones como secuencias de procesamientos de datos.
- Está especialmente indicado para utilizarse en la construcción de Sistemas Interactivos.**
- Es sencillo agregar nuevas transformaciones de forma que el sistema evolucione.
- Es posible reutilizar los filtros.

22. Las revisiones, como técnica de validación de requisitos:

- Son costosas.**
- Son siempre una discusión informal entre cliente y desarrolladores.
- Solo son formales cuando se utilizan métodos formales.
- No incluyen las recorridas.

23. La estructura de un área de proceso del modelo CMMI presenta:

- Un propósito, notas relacionadas, las áreas de procesos relacionadas así como el soporte informativo correspondiente.
 - a) y Metas específicas y genéricas que se logran con las prácticas específicas y genéricas asociadas.
 - b) y las metas son requeridas u obligatorias, mientras que las prácticas asociadas son esperadas o deseadas.**
 - b) y tanto las metas como las prácticas asociadas son requeridas o sea obligatorias.
-

24. En un diagrama de Clases se describen:
- clases.
 - (a) y asociaciones entre clases.
 - (b) y relaciones de herencia.**
 - (c) y atributos de calidad.
-
25. Respecto a modelos de proceso
- Construir prototipos que no terminen formando parte del producto final constituye en general una mal práctica ya que incrementa innecesariamente el esfuerzo requerido para completar el proyecto.
 - La principal ventaja de modelo de proceso en fases con liberaciones parciales en producción es 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 modelo de proceso V consiste esencialmente en un modelo de proceso en cascada en el que la planificación de las pruebas se hace antes de la implementación y en el mismo orden en que se llevan a cabo las pruebas (primero las pruebas unitarias, luego las de integración y por último las del sistema).
 - El modelo de proceso en espiral se compone de las mismas fases que un modelo de proceso en cascada, pero cada una de estas fases está dividida en cuatro: Determinar Objetivos, Alternativas y Restricciones; Evaluar Alternativas y Riesgos; Desarrollar y Probar; Planificar, por lo que permite incorporar la Gestión de Riesgos al proceso.
-
26. El "4 + 1" es un modelo para describir la arquitectura de sistemas de software, basándose en el uso de múltiples vistas concurrentes. Acerca de este modelo podemos afirmar que:
- La vista lógica describe la organización estática del software en su ambiente de desarrollo.
 - La vista física describe el mapeo del software en el hardware y la sincronización de procesos concurrentes.
 - La vista lógica está principalmente dirigida a los responsables de la verificación.
 - La vista de procesos refleja requisitos no funcionales como la performance y la disponibilidad.**
-
27. Entre las cualidades más relevantes para un software que permita capturar firma autógrafa (a mano) en una tablet con pantalla táctil y que permita contar con información de características biométricas tales como velocidad y aceleración al momento de firmar (para que la firma tenga validez análoga a la de la firma en papel) están:
- Facilidad de uso, Adecuación al uso, Seguridad de la información (security), Tiempo de respuesta.**
 - Interfaz de usuario atractiva, Facilidad de uso, Adecuación al uso, Modularidad.
 - Interfaz de usuario atractiva, Calidad de la documentación de usuario, Seguridad de la integridad física (safety), Tiempo de respuesta.
 - Facilidad de uso, Tiempo de respuesta, Calidad de la documentación técnica, Modularidad.
-
28. La gestión de proyectos en Ingeniería de Software es particularmente desafiante y diferente a la de otras ingenierías debido a factores tales como:
- El producto (software) se debe entregar al cliente en el plazo y con la calidad establecida.
 - La naturaleza del producto (software) es intangible.
 - El producto (software) debe satisfacer los requerimientos y necesidades del cliente.
 - Los procesos (de software) varían de organización en organización, dificultando su estandarización.
 - Las lecciones aprendidas por un proyecto (de software) pueden no ser válidas o transferibles a otros proyectos.
- Son correctas las afirmaciones:
- Todas son correctas.
 - II, IV y V.**
 - I, II, III y IV.
 - I y V.
-
29. Entre las técnicas de relevamiento de requisitos se encuentran:
- entrevistas, casos de uso, escenarios, etnografía y encuestas.**
 - entrevistas, escenarios, casos de uso y diagramas de flujo.
 - casos de uso, entrevistas y diagrama de clases.
 - escenarios, etnografía, casos de prueba y casos de uso.
-
30. Son actividades de SCM:
- Controlar versiones.
 - (a) e identificar elementos de configuración.**
 - Conformar un Comité de Control de Cambios con la cantidad mínima de integrantes para maximizar su productividad.
 - Todas las anteriores.