- 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. Es una disciplina que ha presentado muchos cambios durante los últimos veinte años, impulsados entre otros factores por la sostenida baja relativa del costo del software respecto al costo del hardware.

   II. 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 relativa de los recursos humanos en el proceso de producción de software.

   III. Desde el punto de vista de la ingeniería de software lo esencial es el software. Tanto el hardware sobre el que funcione como las personas que lo utilicen constituyen elementos no esenciales y poco relevantes.

   IV. Consiste esencialmente en la programación. Construir un producto de software consiste en codificar el programa que lo soporta. Desde el punto de vista del esfuerzo necesario en relación al tamaño, resulta lo mismo construir un producto de software que va a ser utilizado por distintos usuarios que construir un programa que va a ser utilizado por el propio programador y construir un sistema de programas requiere el mismo esfuerzo que la suma de los esfuerzos necesarios para construir cada uno de los programas que lo componen.

   V. Persigue la mejora de la productividad y calidad en la producción de software fundamentalmente a través de la introducción de herramientas que automaticen el proceso de recopilación de requisitos, diseño, construcción y verificación de código.

   VI. Entre las nociones fundamentales que han sobrevivido a la prueba de tiempo cabe mencionar: abstracción, prototipación, proceso de software, reuso (reutilización) y arquitectura de software.

   Son correctas las respuestas:

   a) I, II, III, IV, V
   b) I, III, V, VI
   c) II, IV, VI
   d) VI

---

2. Usar un prototipo de la interfaz de usuario para validar requisitos es apropiado:
   a) en sistemas que tendrán pocas interacciones con el usuario.
   b) en sistemas donde la interacción hombre-máquina es un factor relevante.
   c) en sistemas de procesamiento por lotes.
   d) en sistemas que, en su mayor parte, hacen cálculos.

---

3. Sobre la gestión y seguimiento de los riesgos:
   a) En el documento de riesgos hay que poner solamente los riesgos referidos al producto que se está construyendo y no aquellos que tienen que ver más con el proceso que con el producto específico que se desarrolla.
   b) Se debe determinar la probabilidad de ocurrencia y el impacto de un riesgo al comienzo del proyecto. Estos valores se mantienen constantes a lo largo del proyecto.
   c) En el documento de riesgos se deben incluir solo aquellos riesgos que tengan un impacto negativo en el proyecto o el producto, y no los eventos posibles que puedan tener un impacto positivo en el proyecto o el producto.
   d) Puede ocurrir que a lo largo de un proyecto algunos riesgos dejen de existir (porque ya no aplican) y surjan otros nuevos.

4. 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.

   a) A3(2(1), A4, B2(1), B3(1), C2(1), C3(1)
   b) A1, A4, B1, B4, C1, C4
   c) A3(2(1)), A3(1), B4, B2(1), C1, C2(3(1))
   d) A3(2(1)), A3(1), B3(2(4)), B3(2(1)), C1

---

5. Respecto a la liberación de software:
   a) Entre las desventajas de una estrategia de implantación paulatina se encuentra que se tarda más en obtener todos los beneficios del sistema nuevo y que en caso de que el sistema nuevo sustituya a uno anterior, la convivencia de ambos sistemas durante un período suele plantear problemas e inconvenientes.
   b) La principal ventaja de una estrategia de implantación paulatina es que se obtienen los beneficios completos del nuevo sistema desde el inicio mismo de la implantación.
   c) No es buena práctica asignar recursos a la solución de los eventuales problemas que se puedan presentar en la liberación, ya que es preferible asignar esos recursos a la mejora de la calidad y prevención de defectos y problemas.
   d) El momento más adecuado para comenzar la planificación de las actividades asociadas a la liberación de un producto de software es cuando se cuenta con requerimientos y arquitectura del producto estables, lo que reduce significativamente el riesgo de tener que ajustar a posteriori dicha planificación.

---

6. En cuanto a la verificación de software, indique cuál de las siguientes afirmaciones NO es correcta:
   a) La Inspección de software ha mostrado resultados muy malos en lo que refiere a su eficacia (porcentaje de detección de defectos sobre defectos totales). Debido a esto no es muy utilizada en la industria.
   b) Considerando las pruebas de caja blanca, las técnicas de pruebas basadas en el flujo de datos son menos usadas que las técnicas de pruebas basadas en el flujo de control.
   c) Existe una técnica de pruebas de integración llamada sándwich.
   d) La prueba de programas puede usarse para mostrar la presencia de defectos, pero no para demostrar su ausencia.

---

7. Durante la planificación de un proyecto de desarrollo de software:
   a) Es conveniente tener en cuenta las lecciones aprendidas de proyectos similares.
   b) Debe involucrarse a los interesados para relevar y validar hitos críticos que puedan afectar la planificación.
   c) Puede definirse qué metodologías y procesos se aplicarán en las distintas fases.
   d) Todas las anteriores.

8. Respecto al mantenimiento :
   a) 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.
   b) Los sistemas legados son sistemas construidos hace muchos años, a menudo con herramientas y tecnologías hoy obsoletas y suelen tener costos de mantenimiento muy bajos dado que en general abunda personal con conocimiento y expeiencia en esas tecnologías.
   c) En caso de que la documentación de un producto de software haya quedado desactualizada respecto a las modificaciones que se le incorporaron, lo más conveniente es llevar a cabo una reingeniería del producto.
   d) La utilidad de un producto de software de tipo E (que modela la realidad y termina formando parte de esa realidad), no se pierde siempre que se le realice un adecuado mantenimiento correctivo.

---

9. El diseño de un sistema de software se centra en proporcionar la funcionalidad del sistema a través de sus diferentes componentes. Algunas de las actividades que se realizan en este proceso son:
   I. dividir requerimientos: analizar los requerimientos y organizarlos en grupos afines.
   II. identificar subsistemas: identificar diferentes subsistemas que pueden individual o colectivamente cumplir con los requerimientos.
   III. definir las interfaces de los subsistemas: definir las interfaces necesarias y requeridas por cada subsistema.

   Marque la opción más correcta:
   a) I) y II) son correctas
   b) I) y III) son correctas
   c) I), II) y III) son correctas
   d) Ni I), ni II), ni III) son afirmaciones ciertas

---

10. Sobre los requisitos funcionales y no funcionales:
   a) Los requisitos funcionales describen la interacción entre el sistema y su entorno.
   b) Los requisitos no funcionales describen restricciones a los servicios ofrecidos por el sistema.
   c) Se cumplen (a) y (b) y los no funcionales limitan las soluciones posibles y las decisiones de diseño.
   d) Se cumple (a) y, según Sommerville, los no funcionales se refieren solo al producto.

---

11. Sobre la planificación y definición del diseño de un producto de software:
   a) Para minimizar el retrabajo, debe iniciarse una vez aprobado el documento de requerimientos por parte del cliente
   b) Para proyectos de gran magnitud, debe realizarse el diseño a la misma vez que se realizan los casos de prueba basados en casos de uso.
   c) Incluye actividades que implican cómo se realizará el diseño del producto y la definición de interfaces.
   d) Ninguna de las anteriores es verdadera

---

12. Respecto de la verificación:
   a) La gestión de la configuración no es necesaria para los elementos (ítems) de verificación ya que éstos están relacionados directamente con el código. Tener el código bajo gestión de la configuración ya es suficiente.
   b) De la información que se debe registrar de la ejecución de los casos de prueba es suficiente con saber si pasó/falló la ejecución. No es necesario registrar la versión del software que se ejecutó, plataforma u otros detalles técnicos.
   c) Al momento de registrar un defecto, éste debe ser categorizado en su criticidad; por ejemplo, extremadamente crítico, muy crítico, no crítico, cosmético. No es recomendable utilizar otra categoría además de la criticidad ya que más de una categoría puede dar lugar a confusión al tipo de tratamiento que debe recibir el defecto encontrado.
   d) Es recomendable el uso de algún tipo de herramienta para el registro y seguimiento de los defectos, aunque no es obligatorio.

---

13. Sobre casos de prueba a partir de casos de uso (CU)
   a) En la generación de casos de prueba a partir de casos de uso se deben tener en cuenta los flujos alternativos del CU.
   b) (a) y muchas veces es necesario considerar los flujos alternativos más de una vez (en distintos escenarios) y en distintas combinaciones con otros flujos alternativos
   c) (b) y los flujos alternativos globales, que pueden activarse en cualquier punto del CU (p.e. "G1. Cancelar la operación") deben probarse también.
   d) No es razonable generar casos de prueba a partir de casos de uso si no está completamente desarrollado el producto software.

14. Entre las cualidades más relevantes para un software encargado de controlar de forma centralizada los semáforos de una ciudad están:
    a) Portabilidad, Disponibilidad, Adecuación al uso, Calidad de la documentación técnica.
    b) Seguridad de la integridad física (safety), Adecuación al uso, Tolerancia a fallas, Tiempo de respuesta.
    c) Interfaz de usuario atractiva, Adecuación al uso, Facilidad de mantenimiento, Tiempo de respuesta.
    d) Modularidad, Facilidad de ser probado, Facilidad de mantenimiento, Legibilidad del código.

---

15. Los firmantes del Manifiesto Ágil expresaron que:
    a) Estaban descubriendo "mejores formas" de desarrollar software, tanto a través de su experiencia como ayudando a terceros.
    b) Los elementos que conformaban las metodologías tradicionales ya no tenían ningún valor porque no eran suficientemente ágiles.
    c) La documentación y el seguimiento del plan son más importantes que la negociación de los contratos.
    d) Que el término ágil surgía de la necesidad de liberar rápidamente los productos al mercado.

---

16. La técnica de relevamiento de requisitos llamada "de observación":
    a) consiste en observar la operativa en el ambiente de trabajo.
    b) debe ser aplicada solamente en situaciones o períodos normales.
    c) brinda datos que pueden ser generalizados a todas las situaciones / localidades.
    d) debe ser utilizada sin que los involucrados lo sepan.

---

17. Respecto a la verificación y validación:
    a) El proceso de verificación y validación es poco costoso respecto a los otros sub-procesos del desarrollo de software.
    b) La planificación de la V&V conviene comenzarla cuando el producto está en una fase de codificación ya que antes es totalmente innecesario (se está muy lejos de la fase de verificación).
    c) Un criterio que resulta en general muy malo para la terminación de las pruebas de sistema es culminarlas cuando el tiempo adjudicado en el plan de V&V para estas pruebas se ha terminado.
    d) Una forma de estimar la cantidad de defectos remanentes en el software es utilizar a dos grupos independientes para que prueben el software. La estimación de la cantidad de defectos remanentes es igual a la cantidad de defectos encontrada por el grupo que encontró más defectos.

---

18. Respecto a modelos de proceso:
    a) Las principales ventajas del 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) y que usuarios y cliente tienen oportunidad de evaluar antes atributos de calidad externos de al menos una parte del producto.
    b) 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.
    c) El esquema ETVX (Entry, Task, Verification, eXit) se utiliza para especificar modelos de proceso prescriptivos, pero no resulta adecuado para especificar modelos de proceso descriptivos.
    d) 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.

---

19. La normas de calidad ISO/IEC 9126 o 25000 son de suma utilidad para:
    a) Guiar el proceso de construcción de un producto de software.
    b) Establecer los objetivos de calidad de un producto de software.
    c) b) y evaluar un producto de software para su liberación y posterior aceptación.
    d) Ninguna de las anteriores.

---

20. En relación a la verificación, indique cual de las siguientes afirmaciones es correcta:
    a) Las técnicas de caja blanca utilizan conocimiento sobre el código fuente para crear casos de prueba interesantes.
    b) a) es correcta y se cumple que todas las técnicas incluidas en caja blanca cubren todo el código fuente. Cuando cubren solo una parte son llamadas de caja gris.
    c) b) es correcta y la cobertura del código no es algo interesante y de lo cual sea necesario tratar durante la verificación.
    d) Ninguna de las anteriores.

21. Sobre la gestión del alcance en un proyecto de desarrollo de software:
    a) Tanto el alcance del producto como del proyecto se deben medir contra el plan de proyecto, ya que en la mayoría de los casos ambos alcances coinciden.
    b) Mientras que el alcance del producto se mide contra el plan del proyecto, el alcance del proyecto se mide contra el documento de requisitos.
    c) Al momento de balancear las variables (alcance, plazos, recursos y calidad) que se pueden manejar desde la gestión del proyecto, a partir de las necesidades y expectativas de los *stakeholders*, el alcance constituye la variable más importante que debe siempre considerarse de forma prioritaria frente al resto de las variables (plazos, recursos, calidad).
    d) Ninguna de las anteriores es correcta.

---

22. El análisis post-mortem de un proyecto consiste en:
    a) Evaluar el producto construido, el proceso utilizado y los recursos.
    b) a) de un proyecto ya culminado.
    c) Investigar quiénes son los responsable de los problemas surgidos durante el proyecto.
    d) b) para aprender de los problemas surgidos y los logros obtenidos.

---

23. Al calcular puntos de función, al contar el aporte de las transacciones:
    a) no se tienen en cuenta los archivos.
    b) sí se tienen en cuenta los archivos (FTR).
    c) se tienen en cuenta los archivos e importa si se trata de un ILF o un ELF.
    d) solo se debe considerar los archivos que son mantenidos por la aplicación.

---

24. La Ingeniería de Requisitos:
    a) tiene como principal objetivo definir lo que debe brindar un sistema y sus restricciones.
    b) no tiene implicancia directa en la estimación del proyecto.
    c) no tiene relación con la calidad del producto final.
    d) Se cumplen todas las anteriores.

---

25. Sobre la Arquitectura de software:
    a) La Arquitectura en capas se puede utilizar cuando el sistema se puede organizar de forma "natural" en distintas capas jerárquicas que se comunican,
    b) (a) donde cada capa tiene tareas específicas que cumplir y para realizarlas se comunica con la capa inmediatamente inferiror (modelo estricto) o con algunas de las inferiores (modelo relajado).
    c) La Arquitectura orientada a servicios es la arquitectura a utilizar en sistemas que se comienzan a construir de cero ya que es lo más moderno en Arquitecturas de software.
    d) Los requerimientos deben estar totalmente definidos para poder comenzar a construir la Arquitectura de software. Caso contrario surgirán grandes cambios y aumentarán los costos del proyecto debido al retrabajo.

---

26. Como parte del Documento de Definición de Requisitos de un Sistema, se encuentra definido el siguiente requisito: *«Se debe proveer documentación de ayuda para los usuarios, tanto mediante un documento impreso (Manual de Usuario), así como mediante ayuda en línea.»*
    a) Corresponde a uno o más requisitos funcionales.
    b) Corresponde a uno o más requisitos no funcionales.
    c) Corresponde a un requisito no funcional (manual de usuario) y uno funcional (ayuda en línea).
    d) No debería considerarse un requisito; es sabido que para todos los sistemas se debería proveer siempre algún tipo de documentación de usuario.

---

27. Sobre el diseño:
    a) En la actualidad el diseño de los productos de software es una de las etapas más breves en el ciclo de vida de desarrollo. Esto se debe a que los problemas de elección de una buena arquitectura son solucionados por los frameworks de desarrollo.
    b) a) además se cumple que con requerimientos estables es relativamente fácil cambiar de arquitectura.
    c) En la etapa de diseño de productos de software se exploran y estudian soluciones para el software a construir.
    d) c) y algo importante es el estudio de la factibilidad de esas soluciones.

28. Con respecto al mantenimiento de software:
    a) En general conviene atender las solicitudes de cambio en el orden en el que estas se fueron recibiendo (First In First Out) de forma de minimizar el tiempo medio de espera y maximizar la satisfacción de los usuarios.
    b) Es una actividad que los gerentes en general consideran tanto o más relevante que el desarrollo, lo que consituye un factor de motivación del personal que la lleva a cabo.
    c) En general constituye una buena práctica establecer una política de versiones periódicas de forma que se libere un conjunto de modificaciones a la vez, en lugar de liberar modificaciones de forma individual. Esto permite reducir los costos promedio por modificación originados por las pruebas de integración y de regresión del sistema.
    d) De forma análoga a lo que sucede en las etapas finales del desarrollo, luego de varios años de mantenimiento en general se cumple que un producto de software mejora su su nivel de calidad como resultado de la paulatina eliminación de los defectos remanentes.

---

29. Entre las actividades de la Gestión de la Configuración se encuentran:
    a) Identificar elementos que se pondrán bajo la gestión de configuración.
    b) a) y gestionar el cambio de estos elementos.
    c) a) y ubicarlos bajo la línea base de forma de minimizar los cambios.
    d) c) y realizar auditorías de seguimiento que permitan evaluar la aplicación de los procedimientos de gestión de la configuración.

---

30. La Configuración del Software:
    a) comprende ítems generados durante del proceso de ingeniería de software.
    b) a) y algunos ejemplos pueden ser: especificación de requerimientos, conjuntos de casos de prueba, reportes de defectos, manuales de usuarios.
    c) b) y además esto quiere decir que desde que se crea un elemento de configuración es necesario controlar los cambios mediante procedimientos formales.
    d) b) y un elemento de configuración se puede incluir en línea base aún cuando no está estable.