

- Cada pregunta **múltiple opción** contestada correctamente tiene un valor de 3,13 puntos. Cada pregunta incorrecta de la múltiple opción resta 1,04 punto.
- **El puntaje total del examen es 100 puntos y se aprueba con 60 o más puntos.**

Múltiple Opción

1. Con respecto a la ingeniería de requisitos:
 - a) La administración de requisitos comienza una vez se ha finalizado de manera exitosa con el proceso de requisitos.
 - b) La administración del cambio comienza cuando se cuenta con una primera versión del documento de requerimientos.
 - c) La validación de requisitos forma parte del proceso de requisitos, mientras que la verificación forma parte de la administración de requisitos.
 - d) Ninguna de las anteriores.
-
2. Respecto a la ingeniería de software:
 - I. Es una disciplina que ha presentado muy pocos cambios durante los últimos veinte años.
 - II. Guarda relación con la Administración General y en especial con las áreas de 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 la construcción de software es una rama industrial que hace un uso intensivo de mano de obra.
 - 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.
 - 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 construcción de código.
 - VI. Entre las nociones fundamentales que han sobrevivido a la prueba de tiempo cabe mencionar: prototipación, proceso de software, mediciones y arquitectura de software.

Son correctas las respuestas:

- a) I,II,III, IV,V,VI
 - b) I,V
 - c) II,III,V,VI
 - d) II,III,IV,VI
-
3. Respecto al mantenimiento
 - a) Los sistemas legados son sistemas construidos hace muchos años, a menudo con herramientas y tecnologías hoy obsoletas
 - b) (a) y suelen tener costos de mantenimiento elevados y resultar críticos para las organizaciones, pero tardan en ser sustituidos porque justamente al ser críticos su sustitución resulta a la vez compleja y riesgosa.
 - c) La re-ingeniería consiste en rejuvenecer software que ha perdido atributos de calidad relevantes y en particular su facilidad de mantenimiento, a través de una re-documentación de software.
 - d) La re-ingeniería, que a menudo se aplica en los casos de sistemas legados, consiste en aplicar una ingeniería directa seguida de una ingeniería reversa.
-
4. Entre las cualidades más relevantes para un software que permita manipular mapas de calles en un celular inteligente están:
 - a) Modularidad, Facilidad de ser probado, Facilidad de mantenimiento, Legibilidad del código.
 - b) Portabilidad, Disponibilidad, Adecuación al uso, Calidad de la documentación técnica.
 - c) Calidad de la documentación técnica, Tiempo de respuesta, Seguridad de la integridad física (safety), Eficiencia en el uso de recursos.
 - d) Interfaz de usuario atractiva, Adecuación al uso, Facilidad de mantenimiento, Tiempo de respuesta.
-
5. Con respecto al mantenimiento de software
 - a) La gestión de la configuración que resulta esencial en las etapas finales del desarrollo de software, va perdiendo importancia de forma gradual a medida que avanza el mantenimiento.
 - b) Durante el mantenimiento se espera que, de forma análoga a lo que ocurre durante el desarrollo, la calidad del producto vaya mejorando como resultado de la corrección de los defectos residuales del software.
 - c) El costo de mantenimiento de un producto normalmente decrece a medida que se van acumulando intervenciones de mantenimiento correctivo sobre el mismo.
 - d) El costo total de mantenimiento a lo largo de todo el ciclo de vida de un producto normalmente representa una porción de orden comparable a la del costo de desarrollo.

6. 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) A1, A2(1), B1, B2(1), C1, C4
- b) A2(1), A4, B2(1), B4, C2(1), C4
- c) A3(2(1)), A3(1), B3(1), B3(2(1)), C1, C4
- d) A3(2(1)), A3(1), B4, B2(1), C1, C2(3(1))

7. Usted ha sido designado Arquitecto de software de un proyecto de desarrollo de software de gran magnitud:

- a) Será su responsabilidad la definición de las pautas de arquitectura y diseño de la solución.
- b) (a) y es aconsejable su participación en el relevamiento de los requerimientos no funcionales ya que esto son altamente relevantes para definir los aspectos arquitectónicos del producto.
- c) (a) y será necesario que defina, en conjunto con el Jefe de Proyecto, los procedimientos de gestión de la configuración para los artefactos que genere.
- d) (b) y deberá estimar, en conjunto con el resto del equipo de proyecto, el esfuerzo de implementación del producto.

8. Respecto a la liberación de software

- a) En los casos en los que se requiere la conversión de datos de un sistema anterior y capacitar a usuarios, la planificación de la liberación conviene que comience cuando ya se cuenta con una versión estable del producto. Esto asegura que la capacitación se realice sobre un producto que es cercano al que finalmente se va a poner en producción y que la forma de cargar los datos a convertir presenta bajo riesgo de modificaciones.
- b) Las actividades que hay que considerar asociadas a una implantación de software son la capacitación a usuarios y la documentación del sistema.
- c) Durante el período inicial de uso de un sistema nuevo recién liberado no es frecuente que se presenten problemas si se realizaron pruebas del sistema adecuadas y si se realizó un adecuado entrenamiento a los usuarios.
- d) En casos en los que un sistema nuevo sustituye a uno anterior, las principales ventajas de una estrategia de implantación big-bang son que se obtienen los beneficios completos del sistema nuevo desde el inicio y se evitan los problemas asociados a la convivencia del sistema nuevo con el anterior. Y la principal ventaja de las estrategias de implantación paulatinas es que permiten gestionar mejor los riesgos asociados, pero presentan como inconvenientes la convivencia del sistema viejo y nuevo y que los beneficios de la incorporación del sistema nuevo se van obteniendo también de forma paulatina.

9. Durante el relevamiento de requerimientos:

- a) Es más importante contar con un cliente que pueda aportar requerimientos técnicos en lugar de un cliente con perfil funcional, el conocimiento técnico del primero aportará más información útil para el proyecto.
- b) Deben identificarse correctamente los stakeholders y relevar sus necesidades.
- c) (b). A su vez, es necesario definir cuáles requisitos a implementar se pondrán en la línea base y cuáles no formarán parte de esta.

- d) En proyectos pequeños da mejores resultados asignar tareas individualmente que el que estas sean llevadas adelante por todo el equipo de forma colaborativa.

10. Kent Beck presenta el desarrollo de software como una ecuación con varias variables. Estas variables, decisivas para evaluar y mejorar la construcción del software, son:
- Las tecnologías y las herramientas de gestión utilizadas.
 - El costo y el alcance de lo que se desea construir.
 - b) y la calidad esperada así como el tiempo disponible para lograrlo.
 - a) y el ambiente de trabajo así como el estilo de dirección.

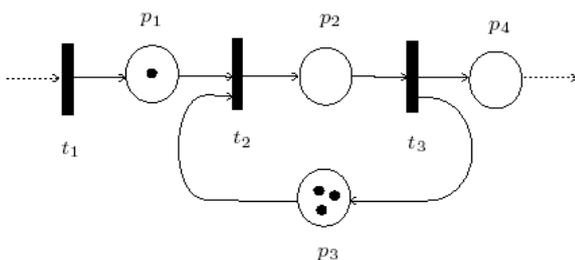
11. En la etapa de obtención de requisitos:
- Se define la arquitectura del sistema.
 - La importancia de cada requisito puede variar con el tiempo, hasta que son validados por el cliente, momento en el cual la importancia de los requisitos es estable.
 - La mejor forma de obtener los requisitos es mediante casos de uso.
 - Es importante clasificar, priorizar los requisitos y validarlos con el cliente.

12. Sobre el diseño de un sistema:
- Durante el diseño se refina la arquitectura del sistema.
 - Un sistema es modular si cumple con el patrón Model-View-Controller.
 - El diseño debería ser verificable y rastreable, pero sin embargo, no debería diseñarse toda la especificación (completo).
 - El diseño incide en el costo del mantenimiento del sistema.

Marque la opción válida:

- Sólo una opción es correcta.
- Sólo dos opciones son correctas.
- Sólo tres opciones son correctas.
- Todas son correctas.

13. La siguiente red de Petri



- Representa un buffer con capacidad ilimitada.
- Representa un buffer con capacidad limitada.
- Representa un recurso compartido.
- Representa un sistema productor-consumidor.

14. Respecto a los modelos de proceso:
- Entre las principales ventajas del modelo de proceso en cascada están su simplicidad y que aporta orden al proceso.
 - El modelo de proceso V consiste básicamente en un modelo de proceso en cascada pero en el que la prueba del sistema se lleva a cabo de forma previa al diseño del sistema, así como la prueba unitaria se lleva a cabo de forma previa a la construcción de cada programa, de forma análoga a lo que ocurre en eXtreme Programming.
 - El modelo de proceso en espiral se puede ver como un proceso en cascada al que se le agrega la gestión de riesgos.
 - La principal ventaja de modelo de proceso en fases con evaluaciones parciales internas 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).

-
15. Uno de los riesgos identificados en un proyecto, es que un proveedor no pueda entregar a tiempo el servidor que le fue solicitado. El equipo de proyecto ha iniciado contacto con otros dos posibles proveedores que pueden ofrecer el mismo equipo aunque a un precio 10% más caro. ¿Qué tipo de estrategia de respuesta al riesgo es esta?
- Como el riesgo es de hardware y no de software el equipo no debe gestionarlo.
 - Transferir el riesgo.
 - Evitar el riesgo.
 - Aceptar el riesgo pasivamente.
-
16. El propósito principal de la estimación de tamaño y esfuerzo es:
- Predecir con exactitud la duración del proyecto y el esfuerzo necesario para llevarlo a cabo.
 - Definir metas que sean lo suficientemente realistas para poder ser alcanzadas con una buena gestión del proyecto.
 - Poder saber cuánto cobrar al cliente.
 - Cumplir con una etapa del proceso de software, para poder mostrar algo al sponsor del proyecto, aunque las cifras estimadas estén lejos de las cifras que luego se alcanzarán.
-
17. Dado un equipo de 10 integrantes, las líneas de comunicación son menos numerosas en:
- Un equipo jerárquico de 3 grupos de 3 integrantes cada uno.
 - Un equipo democrático.
 - Un equipo chief-programmer.
 - Todas las anteriores son prácticamente equivalentes en cuanto a líneas de comunicación.
-
18. De los estilos arquitectónicos vistos en el curso:
- Si un sistema tiene una arquitectura en capas, entonces ésta no puede ser cliente servidor.
 - Si un sistema tiene una arquitectura SOA, entonces necesariamente aplica el estilo de pizarrón (blackboard).
 - Una arquitectura SOA generalmente es mejor que una cliente servidor.
 - En el diseño de la arquitectura de un sistema, puede ser que varios estilos arquitectónicos apliquen a la vez. Generalmente, los estilos se deben adaptar al contexto y al tipo de sistema que se está construyendo.
-
19. Sobre verificación y validación:
- Clasificar los defectos que se van encontrando durante el desarrollo y el mantenimiento de software sirve de guía para mejorar la verificación.
 - (a) y, si se utilizan métodos estáticos, al finalizar la verificación unitaria para cierta unidad, la misma es correcta. Sin embargo, no se puede garantizar el correcto funcionamiento de esta unidad interactuando con otras (integración).
 - (a) y los tipos de pruebas de desempeño están normalmente relacionados con los requisitos no funcionales (estrés, volumen, rendimiento, etc.)
-
20. Sobre la disciplina diseño:
- La interfaz de usuario debe considerarse en fases avanzadas de un proceso de desarrollo en fases. Por ejemplo, en el RUP sería la fase de Construcción una fase correcta para comenzar a diseñar la interfaz de usuario ya que los requisitos están suficientemente estables.
 - El diseño de bajo nivel no debe ser verificado ya que no existen técnicas para verificar ese tipo de producto.
 - Durante el mantenimiento de un producto de software la Arquitectura de software no debe cambiar ya que este tipo de cambios es altamente costoso.
 - Ninguna de las anteriores es verdadera.
-
21. De acuerdo a los datos recolectados de los defectos detectados durante el mantenimiento de un sistema en el período de 1 mes se obtienen las siguientes estadísticas:
- | Gravedad del defecto | Porcentaje |
|----------------------|------------|
| Alta | 30% |
| Media | 20% |
| Baja | 50% |
- Cantidad total de defectos encontrados: 98
- Indique cuál de las opciones es correcta:
- El testing durante el desarrollo de este sistema no ha sido bueno porque se han detectado pocos defectos.
 - El testing durante el desarrollo de este sistema ha sido bueno porque se detectó un alto porcentaje de defectos de gravedad alta.
 - Que el mayor porcentaje de defectos detectados haya sido de gravedad baja indica que el software es de buena calidad.
 - Hacen falta más datos para poder sacar conclusiones acerca del testing durante el desarrollo de este sistema.
-

-
22. Al estimar la duración de una actividad del proyecto se define que su duración más probable es 10 días, la más optimista es 6 y la pesimista es 14. Utilizando PERT, ¿cuál es la duración estimada para dicha actividad?
- 12 días.
 - 10 días.
 - 9 días.
 - Falta información sobre el riesgo de dicha actividad para poder calcular su duración.
-
23. Sobre verificación y validación:
- Los objetivos de la verificación son descubrir defectos y/o evaluar la calidad de cierto producto.
 - Las pruebas de integración se conocen como pruebas big-bang ya que buscan probar cómo funciona el software cuando se juntan las distintas partes (módulos, componentes, subsistemas, etc).
 - Las pruebas de regresión son pruebas de caja blanca, ya que para ver si algo que funcionaba antes ha dejado de funcionar se necesita conocer qué porciones del código se ejecutan (y esto lo brinda el testing de caja blanca).
 - Las técnicas estáticas deben ser utilizadas luego de utilizar técnicas dinámicas de verificación.
-
24. 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.
 - Metas específicas y genéricas que se logran con las prácticas correspondientes.
 - Las prácticas específicas requieren productos de trabajo típicos mientras que las genéricas la elaboración en detalle de la práctica y ambas tienen subprácticas.
 - Todas las anteriores.
-
25. Las especificaciones formales de los requisitos:
- Son fáciles de entender para el cliente y los desarrolladores.
 - Sirven para realizar productos grandes.
 - Permiten detectar omisiones e inconsistencias en los requisitos.
 - Son muy usadas en sistemas de gestión.
-
26. Sobre las técnicas de pruebas:
- Las técnicas estáticas normalmente logran detectar un mayor porcentaje de defectos (del total de defectos) que las dinámicas.
 - Las técnicas dinámicas normalmente logran detectar un mayor porcentaje de defectos (del total de defectos) que las estáticas.
 - (a) y la Inspección es una técnica individual estática de detección de defectos que se basa en una *checklist*.
 - (b) y la Inspección es una técnica individual estática de detección de defectos que se basa en una *checklist*.

Ejercicio

Introducción



Como es de público conocimiento, el Estado se encuentra en un proceso de transformación de sus organismos y procesos. En este contexto se encuentra el proyecto *Ventanilla Única* el cual tiene como propósito unificar y mejorar los procesos de contratación de funcionarios para la Administración Central. El objetivo del proyecto es desarrollar *IngresoUY*, un portal que permitirá la gestión de llamados públicos por parte del Estado así como la postulación a estos por parte de la ciudadanía. Su equipo de trabajo ha sido asignado al proceso de desarrollo de *IngresoUY*, a continuación encontrará la descripción del producto a construir

Descripción del Producto

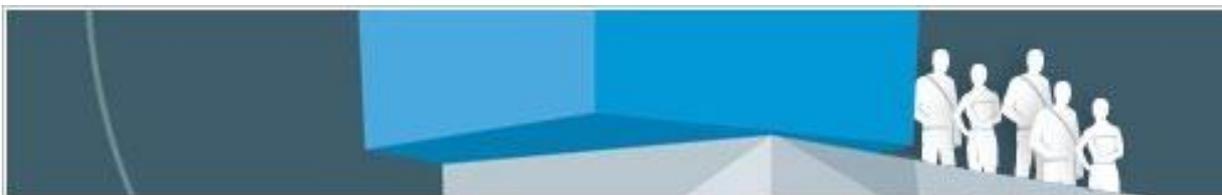
IngresoUY permitirá publicar llamados y postularse a estos, los cuales serán evaluados por tribunales especializados. La oficina del Servicio Civil (ONSC), en coordinación con los ministerios, será el organismo encargado de determinar cuántos funcionarios hacen falta y qué puestos. Esto permitirá contar con un método más ágil, eficaz, transparente y coherente, orientándose a mejorar el sistema de ingreso a la Administración Central.

El Sistema tendrá dos grandes componentes: el Portal de consulta y postulación a llamados y un Backoffice administrativo. Este Backoffice (también web) se utilizará para la gestión de los llamados, permitirá que los usuarios puedan; crear nuevos llamados, determinar su plazo de validez, términos de referencia y tribunal, así como las actividades que el proceso de selección deba ejecutar. Una vez cumplido el plazo de inscripciones al llamado, el sistema procederá a cerrar el mismo, enviando por correo electrónico la información de los postulantes a cada uno de los miembros del tribunal. El Backoffice proveerá distintos reportes como ser: llamados en curso, llamados cerrados, resultados de un llamado, lista de usuarios postulantes del sistema. También contará con el “módulo de subutilización”. Por ejemplo, si hay un funcionario que es arquitecto y trabaja de portero en la Dirección de Loterías y Quinielas, esa persona está subutilizada y probablemente sería más útil en el Ministerio de Vivienda. A través de este módulo el sistema sugerirá su reasignación, contribuyendo a la optimización de los recursos humanos de la Administración. Por otro lado, los postulantes podrán utilizar el Portal Público, herramienta web a través de la cual se podrán visualizar los distintos llamados (tanto los que se encuentran en curso como los cerrados o adjudicados) y podrán realizar su postulación. Para esto, deberán ser usuarios registrados en el sistema; lo cual podrán lograr ingresando su número de cédula y contraseña de ingreso. En la pantalla de ingreso, los usuarios podrán visualizar los llamados en curso, indicándose el período de postulación, cargo requerido, organismo, carga horaria y remuneración. A su vez, cada llamado permitirá acceder a sus Términos de Referencia (TDR) en formato PDF donde se podrá consultar información adicional al llamado. El usuario podrá postularse al llamado, para lo cual deberá completar el formulario de postulación (nombre completo, número de cédula, correo y teléfono de contacto, experiencia laboral, estudios cursados, etc.). De forma adicional, una vez que el llamado sea adjudicado, cada postulante recibirá una notificación con el resultado del mismo (lista de personas seleccionadas). Además, como los usuarios quedan registrados, cada vez que surja un nuevo llamado le llegará un aviso al correo electrónico del interesado, siempre y cuando se adapte a su perfil previamente determinado.

Otro requerimiento importante está relacionado a la accesibilidad del sistema: se deberá asegurar que se pueda ingresar desde todos los sistemas operativos y navegadores, incluso de las XO del Plan Ceibal. También se pretende que el software almacene todos los llamados que se vayan generando en el correr del tiempo, y que los altos volúmenes de información no degraden la performance del sistema. Deberá considerarse que existirán períodos de alta demanda de cargos donde el uso del sistema será exhaustivo debiéndose garantizar la disponibilidad 24x7 del software.

El responsable del proyecto planteó la necesidad de contar con el módulo de Backoffice en primera instancia lo que permitirá comenzar con la carga de llamados. Luego, se pretende poder mostrar resultados del proceso de construcción a los directores de la ONSC tan pronto como sea posible, lo que permitirá ganar confianza y respaldo de la alta dirección.

Fuentes para la elaboración de este documento: Semanario Búsqueda y Diario El Observador.



27. En relación a la Ingeniería de Requisitos:
- La descripción con la que cuenta ya es suficiente para comenzar a diseñar e implementar el producto.
 - Los requisitos no están suficientemente detallados como para cerrar la etapa de relevamiento, en particular los requisitos no funcionales. Estos requisitos son fundamentales al momento de determinar los grados de calidad del producto (por ejemplo usabilidad). Esto refiere al estándar ISO 9126-1.
 - Se cumple a). Además b) no es correcta porque aunque la Ingeniería de Software pretende brindar un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; en general no aplica para proyectos o productos pequeños como el expuesto antes.
 - Ninguna de las anteriores es correcta.
28. Siguiendo con la Ingeniería de Requisitos:
- De la letra se desprenden varios requerimientos no funcionales que deberán ser relevados con mayor detalle.
 - Se cumple a). Esos requisitos no funcionales serán muy importantes al momento de establecer las arquitecturas de software candidatas.
 - Se cumple a). Esos requisitos no funcionales serán muy importantes al momento de establecer criterios de verificación y planes de pruebas. En este caso la arquitectura es clara, el cliente quiere el sistema full web. No es necesario realizar un estudio de arquitectura.
 - Ninguna de las anteriores es correcta porque a) no es correcta. Sí se desprenden varios requisitos no funcionales, pero estos podrían no ser relevados en mayor detalle. Ya se cuenta con información suficiente para establecer las arquitecturas candidatas o los criterios de verificación del producto.

29. En relación al siguiente extracto:

El responsable del proyecto planteó la necesidad de contar con el módulo de Backoffice en primera instancia lo que permitirá comenzar con la carga de llamados. Luego, se pretende poder mostrar resultados del proceso de construcción a los directores de la ONSC tan pronto como sea posible, lo que permitirá ganar confianza y respaldo de la alta dirección.

Según la clasificación vista en clase (que corresponde al estándar IEEE 830) esto corresponde a:

- una restricción.
- un requisito no funcional.
- un requisito funcional.

Es algo a tener en cuenta principalmente en:

- la definición del alcance funcional del producto.
- el plan de actividades del proyecto.
- usted no debería tenerlo en cuenta, ya que corresponde a un tema del cliente y no lo afecta directamente.

Indique la afirmación correcta.

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

30. Siguiendo con la pregunta anterior, indique cuales de las siguientes acciones recomienda en relación a ese tema.

- Adaptar la definición de requisitos funcionales para cumplir con esto.
- Incluir en la Planificación un hito asociado (como puede ser una entrega parcial, o un prototipo).
- Vista la siguiente aclaración: “Luego, se pretende poder mostrar resultados del proceso de construcción a los directores de la ONSC tan pronto como sea posible, lo que permitirá ganar confianza y respaldo de la alta dirección”. Quizás sea sensato evaluar riesgo la perdida de respaldo de la alta dirección.
- Definir un criterio de aceptación más detallado, ya que el requisito no es fácil de verificar. Esto pasa comúnmente con los requisitos no funcionales.
- Para ganar la confianza de la dirección proponer darle mayor visibilidad sobre el proyecto, permitiéndoles participar en algunas actividades. Un buen ejemplo, que le permitiría no modificar la planificación de forma sustantiva, es brindarles acceso al ambiente de desarrollo y que puedan ver el avance del proyecto probando la última versión del producto en cualquier momento.

Indique cuál de las siguientes afirmaciones es correcta:

- II y III son correctas.
- III y V son correctas.
- I, IV y V son correctas.
- II, III y IV son correctas.

31. En relación a la Gestión de la Configuración.
- A primera vista y con lo que se vio en el curso, no existen necesidades de contar con actividades particulares de esta disciplina.
 - Siempre es conveniente contar con actividades de gestión de la configuración, lo que en realidad pasa es que no todo es aplicable para cada realidad particular. Usted debería ajustar alcance y formalidad de las actividades de SCM para poder contemplar las necesidades del proyecto.
 - Se cumple b) pero el alcance no es necesario ajustarlo (todo activo del proyecto se encuentra dentro del espacio de gestión de la configuración). Sí se deberían establecer actividades con un nivel de formalidad requerido.
 - Ninguna de las anteriores es correcta.
32. Suponga que en relación al producto de software usted planifica dos hitos de entrega incrementales. Ambas serán en ambiente de pruebas del cliente y el cliente se encarga de su pasaje a producción. Además es necesario destacar que no existe un sistema previo que de soporte a las funciones descritas. En relación a la Verificación y Validación, indique que pruebas sería conveniente realizar:
- Pruebas Unitarias
 - Pruebas Funcionales
 - Pruebas de Regresión
 - Pruebas de Desempeño
 - Pruebas de Aceptación
 - Pruebas en Paralelo
 - Pruebas de Plataforma
- I, II, III, IV, V y VI.
 - I, II, III, IV, V y VII.
 - I, II, IV, V, VI y VII.
 - Todas las pruebas. Lo más conveniente y sensato es probar todo el sistema.