

- Cada pregunta de la **parte múltiple opción** contestada correctamente tiene un valor de **3,28** puntos. Cada pregunta incorrecta de la múltiple opción resta **1,093** puntos. Esta parte consta de 25 preguntas por lo que vale en total **82** puntos.
- El **ejercicio 1** vale **12** puntos y el **ejercicio 2** vale **6** puntos
- Se aprueba con el 70% del puntaje.

Parte 1 Múltiple Opción

- 1) Un módulo crítico es inspeccionado por 2 personas. La primera encuentra 10 defectos y la segunda 8. Los defectos encontrados en común son 7. Se estima la cantidad de defectos remanentes utilizando la técnica de "pruebas independientes" y el resultado es:
 - a. $(10 \times 7) / 8$
 - b. $(10 \times 8) / 7$
 - c. $(8 \times 7) / 10$
 - d. No se debería haber usado dicha técnica ya que se necesitan al menos 3 personas (o grupos) que realicen la prueba o inspección

- 2) ¿Cuál de las siguientes no es una estrategia de pruebas de integración?
 - a. Pruebas en reversa.
 - b. Big-bang
 - c. Por disponibilidad
 - d. Top-down

- 3) Algunos tipos de faltas que puede contener un software son:
 - a. Las faltas de sintaxis, como por ejemplo: confundir un "0" con la letra "o".
 - b. a) y las de algoritmo como por ejemplo: no inicializar variables.
 - c. b) las de recuperación, como por ejemplo: que no termine el trabajo en el tiempo requerido
 - d. c) y las de estándares o procedimientos, como por ejemplo: no cumplir con un estándar de programación.

- 4) Respecto a las pruebas de regresión:
 - a. Con ellas se intenta detectar si alguna modificación en el software provocó que partes que funcionaban (según las pruebas) dejaron de hacerlo.
 - b. Se clasifican dentro de las pruebas de caja blanca.
 - c. Sirven para detectar defectos que no podrían ser detectados con otro tipo de pruebas.
 - d. Ninguna de las anteriores.

- 5) Respecto de la verificación y validación:
 - a. Uno de los objetivos de la verificación es evaluar la calidad de los productos de software.
 - b. La validación responde a la pregunta "¿Estamos construyendo el producto correctamente?".
 - c. Las técnicas de verificación de caja blanca se clasifican dentro de las técnicas estáticas ya que se detectan defectos sin necesidad de ejecutar el código y las técnicas de caja negra se clasifican dentro de las dinámicas ya que es necesario ejecutar el código para encontrar fallas (y luego detectar los defectos que las originaron).
 - d. Ninguna de las anteriores.

- 6) Indique cuál de las siguientes NO es una ventaja del sistema arquitectónico Tubos y Filtros
 - a. Es intuitivo pensar las aplicaciones como secuencias de procesamientos de datos.
 - b. Es sencillo agregar nuevas transformaciones de forma que el sistema evolucione.
 - c. Es posible reutilizar los filtros
 - d. Está especialmente indicado para utilizarse en la construcción de Sistemas Interactivos.

- 7) Son aspectos de un buen diseño de software:
 - a. Completo
 - b. a), rastreable
 - c. b), amigable
 - d. c) y verificable

- 8) De los principios de diseño vistos:
- Un diseño modular implica que el cambio de un componente no impacta en los demás componentes.
 - El proceso de dividir y conquistar en algunas ocasiones agrega más complejidad y en estos casos no es conveniente seguir con el mismo.**
 - Cuanto más abstracto sea el diseño, más eficiente y simple.
 - Se cumple a y b
-
- 9) Al diseñar una interfaz de Usuario:
- En la mayoría de los casos es importante considerar a los usuarios, quiénes son, de qué cultura, qué habilidades tienen.
 - Un buen diseño de los mensajes de error puede ser fundamental para que el usuario se sienta cómodo con el sistema.
 - Es conveniente estar atento a la usabilidad, esto quiere decir, entre otras cosas, que la interfaz de usuario sea amigable, fácil de comprender, aprender, operar el software.
 - Todas las anteriores**
-
- 10) La línea base de un proyecto
- Se compone de todos los elementos del proyecto ya revisados y acordados
 - Sirve como base para desarrollos posteriores y debe cambiarse sólo a través de los procedimientos de control de cambios.**
 - (a) y se entiende como "cambio" el paso de una línea base a la otra
 - (b). Un elemento de configuración se convierte en línea base cuando fue revisado y aprobado por el Jefe de Proyecto.
-
- 11) Cuando se evalúan los resultados obtenidos luego de la aplicación de un proceso, se está evaluando:
- La madurez de la organización.
 - El desempeño del proceso**
 - La capacidad de un proceso
 - (a) y (c)
-
- 12) La característica del software confiabilidad, definida en el estándar ISO/IEC 9126, implica los siguientes atributos:
- Madurez y tolerancia a fallas
 - Adecuación, precisión y tolerancia a fallas
 - (a) y estabilidad
 - (a) y recuperabilidad**
-
- 13) Una precondition en un caso de uso
- Establece que cosas ocurren al completar el caso de uso.
 - Establece que cosas deben ser siempre verdaderas antes de comenzar un caso de uso.**
 - Es una condición que debe verificarse dentro del caso de uso, pero que no se asume verdaderas desde el comienzo.
 - Es un requerimiento no funcional, un atributo de calidad o una restricción específica relacionada con el caso de uso.
-
- 14) El objetivo de la actividad "Validación de Requerimientos" es
- Asegurar que se esté construyendo el sistema correcto.**
 - Asegurar que se esté construyendo el sistema correctamente.
 - Asegurar que el cliente está de acuerdo con la arquitectura propuesta
 - a y c son correctas
-
- 15) Todos los cambios que ocurran antes de establecer la línea base de los requisitos
- deben ser autorizados por el comité de control de cambios
 - se deben especificar en detalle
 - impactan en la Arquitectura de Software
 - deben ser reflejados en los artefactos necesarios**
-

-
- 16) Medir los requisitos puede ser útil para:
- estimar el tamaño del producto
 - evaluar la calidad de los requisitos
 - hacer un seguimiento del proyecto
 - todos los anteriores**
-
- 17) La administración de los requisitos
- se hace en paralelo al proceso de los requisitos**
 - termina una vez establecida la línea base de los requisitos
 - comienza una vez que se tiene una primera versión del documento de requisitos
 - se realiza siempre que aparezcan cambios
-
- 18) Los Stakeholders:
- Pueden ser personas activamente involucradas en el proyecto
 - a) y también pueden ser organizaciones cuyos intereses pueden verse afectados por el resultado del proyecto**
 - Son aquellas personas u organizaciones que deben determinar los requerimientos del sistema
 - c) y sus diferentes intereses pueden estar en conflicto.
-
- 19) De las técnicas de planificación se sabe que:
- WBS organiza y define el alcance total del proyecto (del producto y del proyecto).
 - a) y CPM introduce tiempo y precedencias, permitiendo calcular la duración total del proyecto.
 - b) y el diagrama de Gantt muestra el origen y el final de diferentes actividades, las cuáles requieren la utilización de ciertos recursos (personas, maquinaria) para su cumplimiento.**
 - c) y PERT es muy usado en Software, no tanto en la construcción.
-
- 20) No es sencillo estimar el esfuerzo en general cuando:
- los integrantes del equipo de trabajo son desconocidos
 - los proyectos usan tecnología de punta
 - la definición inicial de requisitos por parte del usuario es imprecisa
 - todas las anteriores**
-
- 21) La determinación de la frontera del sistema, en el cálculo de Puntos de Función:
- Es independiente de consideraciones técnicas o de implementación.
 - Es independiente de la visión del negocio y externa del usuario.
 - a) y es muy importante determinarla correctamente para que el conteo de PFs sea coherente.**
 - b) y es muy importante determinarla correctamente para que el conteo de PFs sea coherente.
-
- 22) Entre los principales atributos de calidad relevantes para un producto de software para diseño gráfico para arquitectos, en 2 y 3 dimensiones, están:
- seguridad de la información (security), facilidad de mantenimiento, facilidad de verificación, facilidad de adaptación
 - tiempo de respuesta, adecuación al uso, amigabilidad de la interfaz de usuario, consumo de recursos**
 - seguridad de la integridad física (safety), modularidad, amigabilidad de la interfaz de usuario, portabilidad
 - adecuación al uso, facilidad de mantenimiento, legibilidad del código, modularidad
-
- 23) Respecto a modelos de proceso de software:
- El esquema ETVX no es adecuado para especificar modelos de proceso prescriptivos.
 - Entre las ventajas de un proceso en fases con liberaciones parciales en producción está que permite acortar el plazo para poder contar con resultados de utilidad.**
 - El proceso en cascada al día de hoy prácticamente ha dejado de ser utilizado en la industria, habiendo sido sustituido por el modelo transformacional.
 - En el caso de un proceso en fases con liberaciones parciales en producción, la única estrategia de liberación adecuada para cada una de esas fases es la big-bang.
-

24) Respecto a la liberación del software:

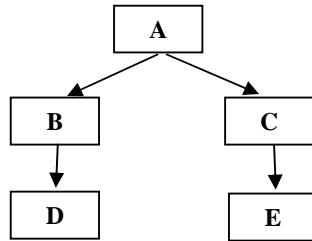
- a. La planificación de la liberación del software debiera comenzar al inicio de un proyecto para poder manejar de forma adecuada el alcance del proyecto.
 - b. La alternativa big-bang resulta especialmente adecuada para proyectos de gran envergadura (por ejemplo de más de 5000 Puntos de Función) y que resultan esenciales para el funcionamiento de organizaciones complejas con gran cantidad y diversidad de usuarios, ya que se evitan la complejidad y los riesgos asociados a la convivencia entre el sistema nuevo y el viejo.
 - c. En una organización que tiene muchas dependencias (sucursales), la única estrategia de liberación paulatina adecuada consiste en poner en producción el software incrementando de forma gradual la cantidad de dependencias en las que se utiliza dicho software.
 - d. Si una organización tiene soportadas sus funciones críticas por un sistema legado (construido hace muchos años y con tecnologías hoy en día obsoletas) que presenta costos muy elevados y crecientes de mantenimiento, no existe justificación posible para no sustituirlo por un sistema nuevo con menores costos de mantenimiento.
-

25) Respecto al mantenimiento de software:

- a. Si una organización que se dedica al desarrollo y mantenimiento de software dedica 90% del esfuerzo total de mantenimiento a mantenimiento correctivo y 10% a mantenimiento adaptativo, se puede concluir que el software bajo mantenimiento resulta muy fácil de adaptar.
- b. Uno de los problemas que presenta el mantenimiento de software es que prácticamente no existen herramientas que asistan en la ejecución y gestión del mantenimiento.
- c. Uno de los problemas que presenta el mantenimiento de software es el conflicto entre las necesidades de corto plazo (liberar los cambios lo antes posible) y las necesidades de largo plazo (mantener y mejorar los atributos de calidad del software y reducir los costos de mantenimiento).
- d. Normalmente las organizaciones de desarrollo y mantenimiento de software asignan mucha importancia al mantenimiento, lo que constituye un factor adicional de motivación del personal que se dedica a esta tarea.

Ejercicio 1 (12 puntos)

La empresa “Suárez Inc.”, cuyo eslogan es “Desarrollamos software con las manos”, está construyendo un nuevo producto de software. Este producto consiste en 5 componentes que se comunican según muestra la figura.



El proceso de desarrollo de software de la empresa tiene actividades de pruebas unitarias y de integración. La estrategia para las pruebas de integración es bottom-up.

Existen diversas alternativas sobre la cantidad de recursos humanos a asignar a este proyecto y la empresa se encuentra estudiando cada posibilidad.

La primera posibilidad asigna 2 desarrolladores. Los mismos comenzarían por los módulos D y E y los terminarían al mismo tiempo. Luego desarrollarían B y C y también terminarían al mismo tiempo. Por último, un recurso se libera del proyecto y el otro implementa A.

Pregunta 26) ¿Cuál es la cantidad mínima de integraciones siguiendo la estrategia de la empresa?

- a) 2
- b) 3
- c) 4
- d) 5

Otra alternativa es asignar 3 recursos humanos al proyecto. Estos recursos desarrollan primero los módulos D, E y B terminando en la misma fecha. Luego desarrollan C y A también terminando en la misma fecha.

Pregunta 27) ¿Cuál es la cantidad mínima de integraciones siguiendo la estrategia de la empresa?

- a) 2
- b) 3
- c) 4
- d) 5

El proceso de la empresa incluye las pruebas unitarias de módulos como ya se mencionó. Cabe aclarar que el módulo A es la interfaz de usuario y la empresa ha decidido no realizar pruebas unitarias sobre dicho módulo.

Pregunta 28) ¿Cuántas pruebas unitarias se ejecutan?

- a) 1
- b) 2
- c) 3
- d) 4

Para probar unitariamente el módulo B la empresa ha decidido que va a usar el módulo D. Sin embargo, en la segunda alternativa esos módulos estarían terminados, incluyendo sus pruebas unitarias, a la misma vez.

Pregunta 29) Esto obliga a tener

- a) un *stub* de D
- b) un *driver* de D
- c) un *driver* de D y un *stub* de D
- d) ninguna de las anteriores

Ejercicio 2 (6 puntos)

Dado el siguiente método que devuelve el promedio de un ArrayList de Integer

```
1    public Integer promedio(ArrayList<Integer> lista){
2        if (lista==null || lista.size()==0){
3            return null;
4        }
5        int prom = 0;
6        for(int index=0; index<lista.size(); index++){
7            prom = prom + lista.get(index).intValue();
8        }
9        prom = prom/lista.size();
10       return new Integer(prom);
11    }
```

El método get no quita elementos de la lista.

Pregunta A) El criterio “también bucles” exige que para cumplirlo todas las decisiones del programa se hagan verdaderas y falsas así como también que todos los bucles ‘no se ejecuten’, ‘se ejecuten exactamente una vez’ y ‘se ejecuten más de una vez’. En caso que alguna exigencia del criterio sea imposible de satisfacer, si se satisfacen el resto de las exigencias, el criterio queda cumplido. ¿Cuál o cuáles de estos conjuntos de casos de prueba satisface el criterio?

Pregunta 30 - a) { (1, 2, 3) , (0, 123, 5666, 453, 444) , () }

Pregunta 31 - a) { null , (4, 6, 7, 22, 333) , () }

Pregunta 32 - a) { null , (3) , (4, 5, 6) , () }

Pregunta 33 - a) { () , (1) , (34, 54, 44) }

Pregunta B) Si al método se le pasa como parámetro una lista con uno de sus elementos (Integer) en null se provoca una falla al consultar el intValue. No confundir con la lista null. Ejemplo de lista (5, 76, null, 54, 33). Marque la o las respuestas correctas

Pregunta 34 - a) Si se hace una prueba con la lista (5, 4, null, 3) se provoca la falla mencionada

Pregunta 35 - a) Si el conjunto de casos de prueba con el que se prueba el método satisface el criterio de condición múltiple se provoca la falla mencionada

Pregunta 36 - a) Si dentro del conjunto de casos de prueba hay un caso de prueba que contiene una lista con un elemento null en ella, al ejecutar el conjunto de casos de prueba se provoca la falla mencionada.

Pregunta 37 - a) El defecto que provoca la falla mencionada se puede detectar mediante inspección de código