

- Cada pregunta de la **parte múltiple opción** contestada correctamente tiene un valor de 3 puntos. Cada pregunta incorrecta de la múltiple opción resta 1 punto. Esta parte consta de 25 preguntas por lo que vale en total 75 puntos.
- El ejercicio 26 vale 10 puntos.
- El ejercicio 27 vale 15 puntos. Cada parte de este ejercicio vale 5 puntos.

Parte 1 Múltiple Opción

- 1) Las cualidades más relevantes para un producto de software destinado a asistir a decoradores de interiores en el diseño de ambientes, de forma tal que puedan mostrar a sus clientes distintas alternativas de decoración y amueblamiento son:
 - a. Seguridad de la integridad física (safety), seguridad de la información (security), interfaz de usuario atractiva
 - b. Interoperabilidad, portabilidad, eficiencia en el uso de los recursos,
 - c. Mantenibilidad, facilidad para ser verificado, adecuación al uso
 - d. Interfaz de usuario atractiva, eficiencia en el uso de los recursos, adecuación al uso

 - 2) Respecto a los modelos de proceso en fases
 - a. en un tipo, al final de cada fase se libera software para su uso en explotación
 - b. (a) y en otro tipo, al final de cada fase se libera software internamente, para su evaluación por el equipo de proyecto, cliente y usuarios
 - c. (b) y una ventaja de las fases con liberación interna es que permiten reducir el tiempo de salida al mercado (time to market)
 - d. una de las principales ventajas de los modelos en fases es que facilitan la evaluación temprana de los atributos de calidad internos del producto en construcción, por parte de cliente y usuarios

 - 3) Un plan no es factible si:
 - a. hay recursos ociosos
 - b. hay recursos subutilizados
 - c. hay recursos sobreutilizados
 - d. todas las anteriores

 - 4) En los Puntos de Función, el problema con los coeficientes de ajuste es que:
 - a. no está claro si los coeficientes de ajuste son independientes entre sí
 - b. todas las características tienen igual peso, porque todos inciden igual en el esfuerzo
 - c. no son dependientes de la tecnología del momento
 - d. la combinación de factores es cercana a 0 y por lo tanto son inocuos

 - 5) Para que un grupo de trabajo funcione, su composición debe ser tal que:
 - a. el líder debe estar “orientado a la relación”
 - b. todos los miembros deben estar “orientados a la tarea”
 - c. no pase que todos los miembros están “orientados a sí mismos”
 - d. dentro del equipo necesariamente coexisten las tres distintas personalidades

 - 6) La gestión de la configuración consiste en:
 - a. el control de versiones
 - b. registro y control de cambios
 - c. generación y control de líneas base
 - d. todos los anteriores

 - 7) El documento de requisitos debe reflejar:
 - a. lo que el usuario pide
 - b. lo que el usuario espera
 - c. lo que el usuario necesita
 - d. todas las anteriores

 - 8) 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
-

-
- 9) Un diagrama de actividad sirve para modelar:
- procesos de negocio
 - casos de uso
 - el flujo de control de un sistema
 - todas las anteriores
-
- 10) La gestión de la configuración de los requisitos comienza:
- cuando terminé la especificación de los mismos
 - cuando llegué a una versión estable a partir de la cuál puedo comenzar el diseño
 - cuando comienza el proceso de requisitos
 - cuando tracé la primera línea base
-
- 11) Los DFDs:
- son descripciones estáticas
 - permiten definir procesos
 - permiten diferentes niveles de abstracción
 - son suficientes para precisar el comportamiento del sistema
-
- 12) Dadas las siguientes afirmaciones indique cuál opción es correcta
- Descubrir defectos es uno de los objetivos de la verificación
 - Evaluar la calidad de los productos es uno de los objetivos de la verificación
 - Clasificar los defectos considerando la fase en la cuál se introducen sirve para introducir mejoras al proceso de desarrollo
- Se cumple 1, no se cumple 2 ni 3
 - Se cumplen 1 y 2 y no se cumple 3
 - Se cumple 2 y 3 pero no se cumple 1
 - Se cumplen 1, 2 y 3
-
- 13) Dadas las siguientes afirmaciones indique cuál opción es correcta
- Es normal en la industria de software que las pruebas unitarias las realice un equipo especializado de testing y no el equipo de desarrollo
 - El conocimiento de las interfaces y la forma en la que interactúan las componentes de un sistema es importante para realizar las pruebas de integración
 - Es imposible realizar pruebas de software de forma exhaustiva
 - Normalmente el autor de un programa tiende a cometer los mismos errores al construirlo que al verificarlo
 - La Recorrida es una técnica dinámica de verificación
- Se cumplen 1, 2 y 3. No se cumplen 4 y 5
 - Se cumplen 2, 3 y 4. No se cumplen 1 y 5
 - Se cumple 2, 3, 4 y 5. No se cumple 1.
 - Se cumple 2 y 3. No se cumplen 1, 4 y 5.
-
- 14) Dadas las siguientes afirmaciones indique cuál opción es correcta
- Las técnicas estáticas buscan ejecutar el producto de software para corregir fallas
 - Las técnicas estáticas son infalibles durante la prueba de sistema
 - Las técnicas estáticas se pueden utilizar para distintos productos de software tales como requerimientos, diseño, código fuente, casos de prueba
 - En las pruebas de caja blanca se obtienen los casos de prueba únicamente a partir del código fuente.
 - En el proceso de pruebas que se dio en el curso, durante la prueba funcional el enfoque generalmente es de caja negra y se busca probar cada funcionalidad de forma individual, así como también distintas combinaciones de funcionalidades.
- Se cumple 3 y 5. No se cumplen 1, 2 y 4.
 - Se cumple 3, 4 y 5. No se cumplen 1 y 2
 - Se cumple 1, 3, 4. No se cumplen 2 y 5
 - Se cumple 4 y 5. No se cumplen 1, 2 y 3
-

-
- 15) Dadas las siguientes afirmaciones indique cuál opción es correcta
1. Las técnicas de verificación estática son efectivas en la detección temprana de defectos.
 2. Las técnicas de verificación estática están sujetas a los errores de nuestro razonamiento.
 3. Mientras se detecten fallas durante el testing este no se debe detener
 4. Un criterio razonable para terminar las pruebas unitarias es que todos los casos de prueba ejecuten sin provocar fallas y que se haya cumplido con algún criterio de cubrimiento de código previamente establecido
- a. Se cumplen 1 y 2. No se cumplen 3 y 4
 - b. Se cumplen todas
 - c. Se cumple 1, 2 y 3. No se cumple 4
 - d. Se cumple 1,2 y 4. No se cumple 3.
-
- 16)
- a. La V&V es normalmente un proceso poco costoso en comparación con el resto de los procesos de desarrollo de software
 - b. Es importante tener fechas asignadas para las pruebas en el Plan de Pruebas. Esto permite asignar recursos y vincular las pruebas con el plan general de desarrollo.
 - c. (b) y uno de los mayores errores que se cometen al realizar el cronograma es suponer que no se van a encontrar fallas.
 - d. Normalmente cuánto más faltas encuentro en un módulo quedan menos por encontrar, es decir, reduzco los defectos remanentes.
-
- 17) Dadas las siguientes afirmaciones indique cuál opción es correcta
1. Un beneficio de realizar una definición explícita de la Arquitectura de Software es el de mejorar la comunicación entre los interesados: cliente – diseñadores, diseñadores – desarrolladores
 2. Un beneficio de realizar una definición explícita de la Arquitectura es clarificar las intenciones de diseño evitando que la Arquitectura concebida se pierda
 3. El patrón Model-View-Controller se plantea como una solución al problema de cambiar la forma de distribución del software
 4. Entre las características de un buen diseño se encuentra la independencia de los elementos de software definidos donde cuanto mayor es la independencia se facilitan aspectos como la comprensión y el mantenimiento del diseño
- a. Se cumplen todas
 - b. Se cumple 1,2 y 4. No se cumple 3
 - c. Se cumple 1 y 4. No se cumplen 2 y 3.
 - d. Se cumple 4. No se cumplen 1, 2 y 3.
-
- 18) Dadas las siguientes afirmaciones indique cuál opción es correcta
1. Las arquitecturas de sistemas distribuidas tienen como desventaja que son complejas y que son difíciles de gestionar.
 2. Una arquitectura peer-to-peer es una arquitectura distribuida
 3. Una desventaja de una arquitectura SOA es que el enlace a los Servicios no se puede realizar en tiempo de ejecución.
 4. Una arquitectura de objetos distribuidos se compone de objetos que proveen servicios a otros objetos y que usan servicios de otros objetos
- a. Se cumplen 1 y 2. No se cumplen 3 y 4.
 - b. Se cumplen 1 y 4. No se cumplen 2 y 3.
 - c. Se cumplen todas
 - d. Se cumplen 1, 2 y 4. No se cumple 3.
-
- 19) Dadas las siguientes afirmaciones indique cuál opción es correcta
1. La Arquitectura de Software de un sistema son los subsistemas que lo componen, las interfaces de estos y las reglas de interacción entre ellos
 2. Cambiar la arquitectura de software de un sistema ya construido es relativamente sencillo si se usa el estilo en capas o de tubos y filtros ya que cada componente (capa o filtro) es independiente del resto.
 3. La elección del estilo y la estructura de la arquitectura más adecuados en un caso concreto, depende fuertemente de los requerimientos no funcionales
 4. Los métodos de evaluación de arquitecturas permiten conocer si la arquitectura cumple con los requisitos de calidad establecidos para el producto
- a. Se cumplen todas
 - b. Se cumplen 1, 2 y 3. No se cumple 4
 - c. Se cumplen 3 y 4. No se cumplen 1 y 2.
 - d. Se cumplen 1, 3 y 4. No se cumple 2
-

-
- 20) Dadas las siguientes afirmaciones indique cuál opción es correcta
1. Algunos de los principios de diseño vistos en el curso son: dividir y conquistar, abstracción y modularidad.
 2. Respecto a la interfaz de usuario. La forma de presentar los datos al usuario no es relevante ya que se usan estándares de interfaz.
 3. Un sistema modular está compuesto de módulos donde estos se pueden implementar de forma separada.
- a. Se cumple 1. No se cumplen 2 y 3
 - b. Se cumple 3. No se cumplen 1 y 2
 - c. Se cumplen 1 y 2. No se cumple 3
 - d. Se cumplen 1 y 3. No se cumple 2.
-
- 21) La liberación de un producto de software para su uso
- a. corresponde al instante en que el producto de software implementado queda pronto para su uso y normalmente no requiere ningún esfuerzo especial ni involucra tareas específicas lograr que sea utilizado de forma adecuada
 - b. (a) , por lo que al definir el alcance de un proyecto de software no vale la pena considerar cuestiones relacionadas con su liberación
 - c. involucra un conjunto de actividades de planificación, preparación, seguimiento y soporte, pudiendo incluso en algunos casos requerir más esfuerzo que la propia construcción del producto de software
 - d. (c), y conviene comenzar la planificación de la liberación una vez que ya se dispone de una versión beta del producto (versión que cubra la mayoría de la funcionalidad relevante y pueda ser usada por los usuarios a los efectos de su evaluación), de forma que exista muy poca incertidumbre respecto a las actividades necesarias
-
- 22) Un sistema de software está compuesto por 824 módulos, agrupados en 4 subsistemas (A,B,C,D). La documentación de los requerimientos del sistema es completa y está actualizada a excepción del subsistema A. La documentación de diseño es pobre y no se actualiza desde hace más de dos años. El sistema está siendo utilizado por poco más de 2500 usuarios. El grupo encargado del mantenimiento recibe un promedio de 30 solicitudes diarias, de las que, también en promedio, la mitad corresponden a problemas originados por fallas del software. Quien soluciona un problema originado por defectos del software, registra al finalizar la intervención, la identificación del (o de los) componente(s) causantes del problema en una planilla electrónica. Del análisis de esa planilla para los últimos 12 meses surge que:
- o el 7% de causas de problemas están asociados al subsistema A
 - o el 17% al B
 - o el 73% al C
 - o el 3% al D
 - o 63 módulos del subsistema A participan en el 83% de los problemas de A
 - o 29 módulos del subsistema B participan en el 82% de los problemas de B
 - o 60 módulos del subsistema C participan en el 81% de los problemas de C
 - o 50 módulos del subsistema D participan en el 85% de los problemas de D.
- Elija la opción que le parezca más adecuada para este caso:
- a. Para reducir la carga total de mantenimiento preventivo, conviene realizar una redocumentación del sistema en su totalidad
 - b. Para reducir la cantidad de problemas que originan mantenimiento correctivo, conviene revisar, ajustar y quizás reescribir, los módulos que concentran los problemas en el subsistema C
 - c. Para reducir la carga total de mantenimiento, conviene realizar una reingeniería de los subsistemas A, B, D y E
 - d. Para reducir la cantidad de problemas que originan mantenimiento correctivo, conviene reescribir el subsistema A
-
- 23) Un modelo de proceso de evaluación consta, entre otras, de las siguientes actividades:
- a. Definir los criterios de evaluación y los niveles de puntuación
 - b. Establecer el proceso de evaluación
 - c. Medir, puntuar y dictaminar
 - d. Todas las anteriores
-
- 24) El modelo recomendado por Collier, DeMarco y Fearey para el análisis post-mortem de un proyecto incluye:
- a. Recopilar información objetiva del proyecto sin comprometer su confidencialidad
 - b. Comenzar inmediatamente de culminado el proyecto
 - c. a) y mantener reuniones para completar la información faltante y reflexionar sobre todos los datos disponibles
 - d. Ninguna de las anteriores
-
- 25) En la representación por niveles del modelo CMMI:
- a. Se promueve la mejora incremental de los procesos correspondientes a una o más áreas de proceso
 - b. Cada nivel estructura el proceso de mejora como un conjunto de áreas de proceso relacionadas
 - c. Se hace énfasis en los niveles de madurez de la organización en las áreas de proceso de la categoría gestión de proyectos
 - d. Se definen un conjunto de actividades y artefactos para lograr una mejor calidad en los productos de software
-

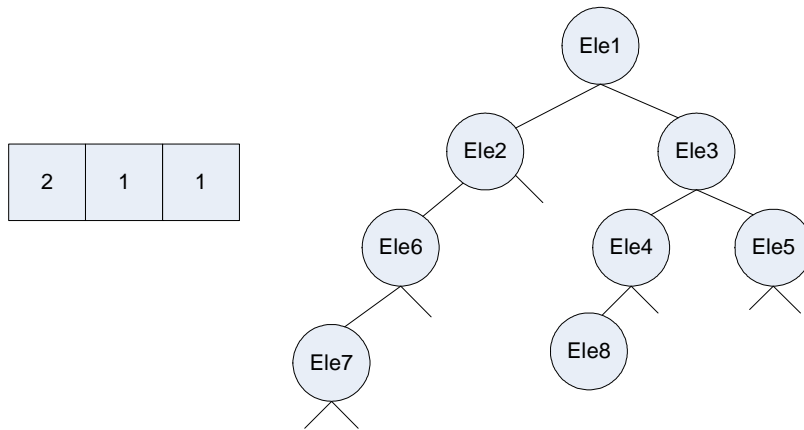
Parte 2 Ejercicios

26) Proponer un conjunto de casos de prueba de cardinalidad 8 para el siguiente método en el lenguaje Java buscando verificar el método de la mejor manera posible:

```
public Object elementAt(Vector<Integer> pos, Arbol arbol)
```

- Este método devuelve el elemento del árbol en la posición indicada por el Vector.
- La estructura “arbol” es recursiva donde cada nodo tiene tres miembros:
 - “elem” de tipo Object
 - “izq” de tipo Arbol
 - “der” de tipo Arbol
- La lista pos puede tener un “0” como primer elemento. En este caso se devuelve el elemento que está en la raíz de “arbol”.
- Los otros valores de la lista “pos” son “1” para identificar el “arbol” de la izquierda y “2” para el de la derecha.
- En caso de no existir el elemento o encontrar algún inconveniente en la ejecución del método se devuelve *null*

Ejemplo:



En el ejemplo se presenta un Vector “pos” con los valores enteros “2”, “1” y “1”. Este vector indica que el elemento que se debe devolver es el que está a la derecha (“2”), luego a la izquierda (“1”) y luego otra vez a la izquierda (“1”) del árbol. Para el ejemplo el elemento que se debe devolver es el “Ele8”.

Se debe justificar cada caso de prueba incluido en el conjunto de casos de prueba. Esta justificación puede ser desde el punto de vista del defecto que se intenta detectar o desde el punto de vista del cubrimiento funcional.

27) Dado el siguiente método:

```
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 devuelve el promedio de un ArrayList de Integer.

Parte a) Identificar las distintas decisiones y las condiciones dentro de estas. Presentar qué combinaciones de valores de verdad se tienen que ejecutar para cumplir con el Criterio de Cubrimiento de Condición Múltiple (CCCM).

Parte b) Construir un conjunto de casos de prueba de cardinalidad mínima que cumpla con el criterio de CCCM

Parte c) Suponer que la sentencia 7 es ahora la siguiente:

```
prom = prom + lista.get(0).intValue();
```

Explique y **justifique brevemente** si cumplir con el CCCM asegura que el defecto provoque una falla.