1. Las cualidades más relevantes de un producto de software concebido para asistir a un visitante de un museo, brindándole información de recorridos, visitas guiadas, contenido de las salas, programa de muestras y otras actividades, a ser instalado en PCs autónomos, son:

a. seguridad de la información (security), eficiencia en el uso de recursos, escalabilidad
b. seguridad de la integridad física de las personas (safety), interfaz de usuario atractiva, mantenibilidad
c. interfaz de usuario atractiva, facilidad de uso, facilidad de aprendizaje
d. eficiencia en el uso de recursos, tiempo de respuesta, portabilidad

---

2. Le encomiendan llevar adelante un proyecto que consiste en construir un producto para asistir en la gestión de proyectos de software. El producto debe permitir manejar tareas y subtareas sin límite de niveles, asignar recursos a las tareas, facilitar la imputación de horas trabajadas a las tareas planificadas, permitir registrar el estado de las tareas, proveer interfaces con las herramientas de mayor difusión para la gestión de proyectos, soportar el enfoque de Valor Ganado (para un proyecto completo o para un subconjunto de tareas) , debe soportar múltiples usuarios y posibilitar acceso vía Web, control de acceso por usuario con manejo de perfiles, con un perfil administrador encargado de administrar los permisos de los perfiles y los perfiles asociados a cada usuario. Se le aclara con especial énfasis que se espera disponer de resultados de valor (útiles), en un máximo de 3 meses. Para encarar la planificación del proyecto usted decide partir del modelo de proceso:

a. Especificación operacional
b. (a), complementado con el modelo Transformacional, para lo relacionado con el enfoque del Valor Ganado
c. En Fases
d. En Cascada

---

3. Una empresa de telecomunicaciones con varios cientos de miles de clientes, que brinda servicios en varias ciudades, está encarando la sustitución de su sistema de gestión comercial. Acaba de iniciar un proyecto para la adaptación del sistema utilizado por una empresa de telecomunicaciones de otro país. Los responsables de ese proyecto esperan disponer en 14 meses del sistema adaptado, pronto para comenzar la implantación. A usted le encomiendan realizar la planificación de la implantación del nuevo sistema.

a. Dado que la implantación no va a empezar hasta dentro de 14 meses, y como resulta conveniente realizar una planificación muy precisa de la implantación, estima disponer de una primer versión de plan en un plazo de 3 meses.
b. Dado el plazo disponible, no cabe duda de que es completamente factible comenzar la implantación en 14 meses.
c. Entre las principales tareas a considerar están la conversión de datos del sistema actual al nuevo, la capacitación a los involucrados, la elaboración de la documentación para los usuarios y preparar mecanismos para la solución de problemas durante la explotación.
d. (c) y Big-Bang es la estrategia de implantación más conveniente en un proyecto de estas características

---

4. Una organización productora de software desarrolló una línea de productos para la gestión financiera. Productos de la línea están siendo instalados en un conjunto creciente de clientes, lo que está generando un aumento significativo en las solicitudes de cambios y reporte de problemas de parte de los clientes. Hasta el momento la organización ha priorizado la rapidez en la solución de los problemas de los clientes por encima de otras consideraciones. Este enfoque está volviendo difícil de manejar la diversidad de versiones en los distintos clientes, existiendo casos de versiones que en algunos aspectos presentan características incompatibles entre sí. Para tratar de reducir la cantidad de versiones de productos de la línea se podría:

a. Establecer un Comité de Control de Cambios (CCC) encargado de autorizar y priorizar las modificaciones, con una visión global de las características de los productos de los distintos clientes, y que tome en consideración de forma adecuada las eventuales incompatibilidades
b. (a) y en la priorización el CCC debiera definir en qué versión programada se incluye una modificación
c. (b) y el procedimiento debiera prever mecanismos para resolver situaciones de emergencia de forma de solucionar problemas puntuales, en uno o más clientes
d. (c) y establecer una política de liberación de nuevas versiones, con una cadencia pre-establecida, buscando que los distintos clientes se mantengan sincronizados en la misma versión

---

5. Dados dos archivos lógicos:
Empleados (#empleado, CI, nombre, fecha_nacimiento, #sección)
Sección (#sección, descripción)
al contar los DETs para el ILF Empleados, se deben contar:

a. 3 DETs
b. 4 DETs
c. 5 DETs
d. 6 DETs

---

6. En la Línea Base se debe poner:

a. solo el código
b. las últimas versiones de todos los elementos que se tengan en un momento dado.
c. las últimas versiones solo de aquellos elementos que se hayan definido deben ir a la línea base.

d. las versiones aprobadas de aquellos elementos estables, para cambiar los cuáles se requiere un procedimiento de control de cambios

---

7. En el proceso genérico de diseño visto en el curso se definen distintos niveles donde
a. En el nivel 1 se realiza el diseño arquitectónico o definición de la Arquitectura del Software que consiste en especificar los subsistemas que componen la solución y los algoritmos y estructuras de datos correspondientes
b. (a) y en el nivel 2 se definen las interfaces de cada subsistema y las relaciones entre estos, que permiten entender en alto nivel las comunicaciones que existirán para realizar cada requerimiento especificado
c. (b) y en el tercer nivel se puede evaluar tiempo de ejecución de los algoritmos y realizar optimizaciones de código
d. ninguna de las anteriores

---

8. Entre los estilos de Arquitectura vistos en el curso
a. el de tubos y filtros permite comprender el comportamiento total de entrada/salida del sistema a partir de los efectos de cada filtro, y es particularmente apropiado para sistemas interactivos
b. el de invocación implícita permite cambiar fácilmente los componentes que atienden un evento determinado, ya que el registro indica para cada evento únicamente qué componentes son capaces de procesarlo
c. el de capas facilita el mantenimiento y la reutilización, dado que es posible desde cualquier capa requerir los servicios especificados por otra capa superior o inferior a ella
d. se cumplen todas las anteriores

---

9. La reutilización de COTS y frameworks
a. consiste en reutilizar componentes que proveen un comportamiento determinado en el caso de los COTS, y que proveen un esqueleto de comportamiento determinado en el caso de los frameworks
b. (a) sin embargo, el enfoque de reutilización es bastante distinto ya que para los frameworks es de caja blanca al tener que estudiar el código fuente, mientras que los COTS tienen enfoque de reutilización de caja negra
c. (a) y el enfoque de reutilización es el mismo, se debe conocer qué funcionalidades proveen tanto el framework como el COTS e insertar las llamadas correspondientes en el código que se está desarrollando
d. (a) y en general resulta más fácil reutilizar un framework que un COTS, ya que tienen como ventaja la posibilidad de estudiar el código cosa que permite entender rapidamente el funcionamiento del mismo

---

10. Entre las características de un buen diseño
a. el tratamiento de anomalías mediante diseño defensivo refiere a la anticipación de situaciones que podrían llevar a problemas en el sistema, las que pueden ser tratadas con distintas técnicas como reintentar, corregir o informar
b. la independencia de componentes permite una mejor comprensión y mantenimiento del sistema, siendo la medida de esta independencia la baja cohesión y el alto acoplamiento
c. la prevención y tolerancia a faltas refiere a la capacidad del software de anticipar fallas y manejarlas de forma de minimizar los efectos negativos y maximizar la seguridad, una vez que la falla ocurre
d. ninguna de las anteriores

---

11. En la etapa de obtención de requerimientos:
a. La importancia de cada requerimiento puede variar con el tiempo, hasta que son validados por el cliente, momento en el cual la importancia de los requerimientos es estable
b. La mejor forma de obtener los requerimientos es mediante casos de uso
c. Es importante clasificar y priorizar los requerimientos junto con el cliente
d. Se cumplen todas las anteriores

---

12. La administración de los requerimientos:
a. La administración del cambio comienza una vez que el sistema es puesto en producción, cuando el cliente requiere un cambio en su funcionalidad
b. Se planifica una vez que se terminó con el análisis de requerimientos
c. Cuando los requerimientos cambian, permite conocer el impacto de ese cambio en el proyecto.
d. Permite al gerente del proyecto conocer la cantidad de requerimientos ya implementados y de esa forma saber el grado de avance del proyecto

---

13. La validación de requerimientos es un proceso donde se determina:
a. si los requerimientos fueron diseñados correctamente
b. si los requerimientos cumplen las necesidades del cliente
c. si los requerimientos fueron implementados correctamente
d. Se cumplen todas las anteriores

---

14. El modelo de madurez de la capacidad de los procesos CMMI procura encontrar solución a los siguientes problemas del modelo CMM:
a. Que no incluía los procesos de desarrollo y verificación de sistemas críticos

b. Que las organizaciones no tenían la posibilidad de elegir los procesos que deseaban mejorar según sus necesidades y requerimientos
c. (b) y que no incluía procesos de las ramas de ingeniería de sistemas e ingeniería de software
d. Su escasa aplicabilidad a las pequeñas y medianas empresas productoras de software

15. Los atributos de calidad de un producto de software reutilizable
a. Son el nivel de utilización y el grado de encapsulamiento
b. (a) y el número y tipo de defectos detectados
c. Están claramente identificados en el modelo de calidad ISO 9126
d. Ninguna de las anteriores

16. El análisis post-mortem de un proyecto:
a. consiste en analizar los proyectos que demoran más tiempo que el planificado
b. es recomendable siempre al final de un proyecto, para identificar lo que anduvo bien, mal y por qué
c. (b) y permitir la evolución de las estimaciones
d. (c) y detectar oportunidades de mejora del proceso

17. El testing
a. sirve tanto para verificar como para validar con el cliente
b. (a) y sólo se puede asegurar el correcto funcionamiento de los casos de prueba que se ejecutaron sin resultar en fallas.
c. (b) y el testing unitario se realiza mediante distintas técnicas de análisis de código fuente, por ejemplo, inspecciones de código
d. (b) y sólo sirve (el testing) para probar la funcionalidad del sistema y no para probar aspectos de performance

18.
a. El análisis automatizado de código fuente es una de las formas vistas dentro del curso de técnica estática.
b. (a) y una técnica híbrida es la ejecución simbólica
c. (b) y la ejecución simbólica es muy fácil de escalar y dentro del proceso de V&V visto en el curso se usa en las pruebas de integración.
d. Las inspecciones se clasifican como técnicas dinámicas y permiten evaluar a los programadores.

19. Para las pruebas unitarias
a. si uso caja negra, no necesito disponer del código fuente de la unidad a probar para obtener los casos de prueba
b. (a) y si uso caja blanca necesito el código fuente de la unidad a probar
c. (b) y estas técnicas (caja negra y caja blanca) se complementan
d. (b) y estas técnicas (caja negra y caja blanca) no se complementan y conviene en las pruebas unitarias usar sólo caja blanca.

20. El objetivo principal de la verificación es:
a. Descubrir defectos
b. Corregir defectos
c. Estipular junto con el cliente los criterios de aceptación de un producto
d. Conocer si el programa es “cero defecto”

EJERCICIOS

Ejercicio 21 (20 puntos)
Una empresa distribuidora mayorista desea implantar el paquete de software XYZ para soportar la gestión de inventarios de mercadería, pedidos a proveedores y entregas a clientes. Esta empresa cuenta actualmente con un sistema que funciona con tecnología obsoleta que soporta parcialmente estas funciones. Sus oficinas y depósito están ubicados en un mismo edificio. La empresa no cuenta con equipos informáticos con capacidad disponible como para implantar el paquete.
El paquete XYZ tiene previsto un conjunto grande de parámetros para adaptarlo a las necesidades específicas de cada organización. Sin embargo, existe cierto riesgo de que resulte necesario incorporar modificaciones al software.

Construya un WBS para el proyecto, considerando que el resultado esperado es contar con el paquete XYZ implantado y funcionando en la empresa, de forma de satisfacer sus necesidades.

Ejercicio 22 (10 puntos)

```
read(x); read(y);
while x<>y loop
        if x>y then
                x:=x-y;
        else
                y:=(y/x) - 1;
        endif;
end loop;
```

a) Dar un valor de entrada para ***x*** y otro para ***y*** de forma que la ejecución del código dado provoque una división entre cero. (2,5 puntos)
b) Indicar si alguno de los siguientes criterios asegura detectar la división entre cero: de sentencias, condición/decisión, condición múltiple. (5 puntos)
c) Proponer tres conjuntos de prueba donde cada uno satisface uno de los criterios presentados. Los conjuntos deben cumplir lo siguiente: (2,5 puntos)
    a. Ser el conjunto con menor cantidad de elementos que satisface un criterio
    b. En caso que el criterio NO asegure detectar la división entre cero (parte b) el conjunto de casos de prueba brindado NO debe ejecutar una división por cero.

Ejercicio 23 (10 puntos) – 2,5 puntos cada parte

Se tiene el siguiente procedimiento que dado dos números devuelve el menor.

```
Procedure menor (a, b: int)
Var menor : int;
Begin
    menor := 0;
    if (a < b) then menor := a else menor := b;
    return menor;
End.
```

a) ¿Una inspección de código puede detectar que la asignación del valor cero a la variable ***menor*** es innecesario?
b) ¿El análisis automatizado de código fuente puede detectar que la asignación del valor cero a la variable ***menor*** es innecesario?
c) ¿Puede dar algún caso de prueba que detecte lo innecesario de la asignación ya mencionada? En caso de responder que sí escriba al menos un caso de prueba que lo detecte.
d) Escriba un conjunto de casos de prueba que cumpla con el criterio de condición múltiple.