

- Cada pregunta **múltiple opción** contestada correctamente tiene un valor de **2 puntos**. Esta parte consta de **20 preguntas**, haciendo un total de **40 puntos**.
- Los ejercicios de desarrollo tienen un valor total de **60 puntos**.

1. Las cualidades más relevantes para un sistema de servicios financieros minoristas por Internet son:

- Seguridad de la información, adecuación al uso, facilidad de uso.
- Eficiencia, modularidad, verificabilidad, interoperabilidad.
- Mantenibilidad, visibilidad, legibilidad (del código).
- Adaptabilidad, conformidad, seguridad de las personas (safety), facilidad de aprendizaje.

2. Considerando los modelos de proceso:

- RUP (Rational Unified Process) es incremental e iterativo, está centrado en la arquitectura y guiado por Casos de Uso.
- Los incrementales no son adecuados para proyectos cuyos productos sean críticos respecto a la seguridad.
- En espiral es de los más adecuados para proyectos de menos de 2 meses, con requerimientos definidos y estables.
- El modelo en cascada está especialmente adaptado para soportar cambios en los requerimientos en etapas avanzadas del proyecto.

3. Para medir el tamaño de la funcionalidad de una aplicación en Puntos de Función se elaboró el cuadro siguiente en el que se indica la cantidad de cada una de las características que aportan Puntos de Función y a su vez se discrimina cada una de acuerdo al grado de complejidad.

Evaluación					Ponderadores		
Característica\Complejidad	Cantidad	Baja	Media	Alta	Baja	Media	Alta
Entradas Externas (EI)	13	2	5	6	3	4	6
Salidas Externas (EO)	8	5	3		4	5	7
Consultas Externas (EQ)	17	5	5	7	3	4	6
Archivos Lógicos Internos (ILF)	3	1	2		7	10	15
Archivos de Interfaz Externa (EIF)	3	1	2		5	7	10

Siendo 1,1 el coeficiente de ajuste obtenido considerando los distintos factores de ajuste:

- los puntos de función sin ajustar son 220.
- (a) y los puntos de función (ajustados) son 242.
- (a) y los puntos de función (ajustados) son 200.
- los puntos de función (ajustados) son 220.

4. En un proyecto de software, los riesgos:

- identificados se deben reducir sistemáticamente, esto es desarrollar acciones que permitan aumentar la probabilidad de que ocurran y/o disminuir el impacto negativo que producirían.
- deben ser identificados y evaluados esencialmente por el cliente, que es quien dispone de la visión requerida del negocio.
- relevantes suelen cambiar a lo largo del mismo, por lo que en instancias predefinidas conviene rehacer las evaluaciones de riesgos.
- suelen ser estables a lo largo del mismo, por lo que conviene identificarlos con precisión al comienzo del proyecto.

5. Respeto al manejo de conflictos en un proyecto:

- Los conflictos deberían evitarse siempre que sea posible, ya que distorsionan el buen ambiente de trabajo de todo proyecto.
- Una buena gestión y manejo de conflictos es clave para el éxito del proyecto.
- a) y una de las técnicas para la resolución de conflictos es "suavizarlo".
- b) y la técnica más efectiva para la resolución de conflictos es lograr una solución de compromiso.

6. La Especificación de Requerimientos:

- a) sólo debe incluir requerimientos funcionales.
- b) debe incluir requerimientos funcionales, no funcionales y alternativas de diseño.
- c) debe incluir requerimientos funcionales, no funcionales y no debe especificar opciones de diseño.
- d) sólo debe incluir requerimientos no funcionales.

7. Durante la etapa de requerimientos:

- a) No se realizan prototipos, ya que estos son útiles recién en la etapa de Diseño, para ver aspectos de la arquitectura como pueden ser los riesgos tecnológicos.
- b) Se pueden realizar prototipos desechables, para explorar la factibilidad y acotar los riesgos
- c) Se puede desarrollar un prototipo evolutivo, el cual se convierte en parte del producto final.
- d) Son correctas (b) y (c).

8. Una arquitectura:

- a) en capas favorece la portabilidad pero presenta como inconveniente que es difícil de entender y generalmente se ve afectada la performance del sistema si se definen muchas capas.
- b) basada en tubos y filtros favorece la reutilización de componentes, es fácil de entender en base a sus componentes, es adecuada para aplicaciones interactivas, pero se generan ineficiencias por la repetición de procesos entre filtros.
- c) a) y b) son correctas.
- d) ninguna es correcta.

9. Entre los beneficios de realizar una definición explícita de la Arquitectura de Software se encuentran:

- a) mejorar la comunicación entre los interesados: cliente – diseñadores, diseñadores – desarrolladores.
- b) clarificar las intenciones de diseño evitando que la Arquitectura concebida se pierda, aportando al mantenimiento del sistema.
- c) proporcionar bases para el análisis del diseño en cuanto a evaluación de características requeridas o deseables como el desempeño.
- d) todas las anteriores.

10. Respecto a la fase de diseño:

- a) A pesar de que existen varias soluciones diferentes para diseñar un sistema de software, siempre es posible llegar a la mejor solución si se siguen los pasos establecidos por el proceso de desarrollo.
- b) La fase de diseño es fundamental ya que sienta las bases para definir la Arquitectura del Software.
- c) Los principios de diseño vistos en clase son: Modularidad, Dividir y Conquistar, y Abstracción; siendo la modularidad la conjunción de la partición y la abstracción.
- d) Los diagramas UML que se usan en esta etapa son: Diagrama de Clases, Diagrama de Colaboración, Modelo de Dominio.

11. Sobre las pruebas de caja negra y blanca:

- a) Para generar los casos de prueba utilizando la técnica de caja blanca se necesita del código.
- b) a) y si se realizan pruebas solamente basadas en caja negra parte del código puede no ser ejecutado.
- c) b) y una de las técnicas de caja negra, vista en el curso, es la técnica de cubrimiento de sentencias.
- d) Ninguna de las anteriores

12. Dado el algoritmo que se presenta a continuación indicar cuál de los siguientes criterios es el menos fino que asegura detectar la posible división entre cero. La palabra “asegura” se refiere a que cualquier conjunto de casos de prueba que cumpla con el criterio detecta el defecto.

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

- a) Criterio de condición múltiple.
 - b) Cubrimiento de sentencias.
 - c) Todos los c-usos (las sentencias read(x) y read(y) son definiciones de x e y respectivamente).
 - d) Ninguna de las anteriores.
13. Considerar la siguiente situación.
- La base de datos de una estación de servicio guarda la cantidad de litros disponible de su único surtidor (con su único tipo de combustible). Este valor es actualizado mediante el “controlador náutico” que es un pequeño detector y un software propietario asociado que está instalado en el surtidor. El “controlador náutico” se comunica con la base de datos cada vez que se termina de extraer o ingresar combustible al surtidor. La base de datos también mantiene el precio por litro del combustible que se vende. Considerando la venta de combustible en la cual el pistero ingresa la cantidad de litros a vender en el surtidor y la necesidad de generar casos de prueba para la “venta” y usando criterios de la realidad:
- a) Se deben considerar como datos de prueba la cantidad de litros disponibles y el precio por litro que están en la base de datos así como también los litros ingresados por el pistero en el surtidor.
 - b) Se debe considerar como dato de prueba los litros ingresados por el pistero en el surtidor pero no los datos de la base de datos.
 - c) a) y realizar pruebas sin la base de datos y sin el “controlador náutico” sería realizar pruebas para tener una idea de la robustez del sistema en esas situaciones.
 - d) b) y realizar pruebas sin la base de datos y sin el “controlador náutico” sería realizar pruebas para tener una idea de la robustez del sistema en esas situaciones.
14. Sobre la comparación entre técnicas estáticas y dinámicas de verificación:
- a) Las técnicas estáticas son efectivas en la temprana detección de defectos.
 - b) a) y las técnicas dinámicas consideran el ambiente de ejecución.
 - c) b) y también sirven para probar otras características además de la funcionalidad.
 - d) c) y las técnicas estáticas sirven también para verificar el diseño y la especificación de los requerimientos.
15. Respecto a los criterios de terminación de pruebas:
- a) Cumplir con cierto criterio de cubrimiento de código al ejecutar las pruebas (y que todas las pruebas ejecuten correctamente) puede ser adecuado como criterio de terminación de las pruebas unitarias.
 - b) (a) y la siembra de defectos es una técnica para estimar la cantidad remanente de defectos.
 - c) La terminación de las pruebas basada en la cantidad de defectos detectados por unidad de tiempo establece que si se detectan menos de 5 defectos por semana durante el testing de sistema las pruebas deben culminar.
 - d) Ninguna de las anteriores.

16. El mantenimiento de software:

- a) enfrenta a menudo un conflicto entre las necesidades de corto plazo (solucionar rápidamente un problema puntual) y necesidades de largo plazo (costos de mantenimiento reducidos en el futuro).
- b) a) y una estrategia posible para mejorar el servicio de mantenimiento consiste en aplicar una política de liberaciones planificadas de nuevas versiones, por ejemplo liberar una nueva versión cada 6 meses.
- c) a) y una estrategia posible para mejorar el servicio de mantenimiento consiste en incorporar al software y poner en explotación lo más pronto posible, cada solicitud de cambio.
- d) c) y el Comité de Control de Cambios es una entidad que deja de tener relevancia una vez que el software se pone en explotación.

17. La gestión de la configuración del software:

- a) en los requerimientos es importante para gestionar los cambios y durante el diseño es importante para controlar la correspondencia del diseño con los requerimientos y con su evolución.
- b) a) y durante todo momento permite mantener una línea base con los requerimientos del software acordados, el diseño correspondiente y la implementación asociada.
- c) b) y mantener la correspondencia con los casos de prueba y la correspondencia entre fuentes y ejecutables.
- d) c) y asegurar que no se producen pérdidas o alteraciones no autorizadas de los distintos componentes del software.

18. El análisis post-mortem de un proyecto:

- a) consiste en analizar los proyectos que se cancelan de forma prematura, sin alcanzar su objetivo.
- b) se realiza sobre el final del mismo pero la oportunidad más adecuada depende de las características de la organización, aplicación y proyecto.
- c) b) y es esencial para posibilitar la mejora del proceso, identificando lo que anduvo bien, mal y por qué.
- d) c) y a menudo se lleva a cabo pasado un tiempo del proyecto para tener datos del resultado en explotación, sin lo cual es aventurado evaluar los resultados.

19. Según Deming, los cuatro pasos hacia la mejora continua son (en ese orden):

- a) Planificar, Ejecutar, Actuar, Verificar
- b) Hacer, Verificar, Actuar, Evaluar
- c) Planificar, Hacer, Verificar, Actuar
- d) Evaluar, Hacer, Actuar, Re-evaluar

20. Respecto a los niveles de madurez de CMMI:

- a) Coinciden con los niveles de capacidad de la representación continua de CMMI.
- b) a) y existen 5 niveles de madurez: cuanto menor es el nivel más madura se encuentra la organización.
- c) En una organización que se encuentra en el Nivel 3 (Definido), sus procesos se encuentran gestionados y se ajustan a la política de procesos que existe en la organización.
- d) c) y una organización que se encuentra en el nivel inicial (Nivel 1), implementa prácticas en gestión de proyectos.

Ejercicio 21 (22 puntos)

La compañía área “Charrúa Airlines” desea mejorar la atención de sus pasajeros vía web. Los pasajeros que accedan al sitio web www.charruairlines.com pueden reservar los pasajes y más tarde realizar el *check in*.

Para la reserva del pasaje se le pide: ciudad de origen, ciudad de destino, fecha de ida, fecha de vuelta, clase (primera, ejecutiva, turista) y cantidad de pasajeros. El sistema presenta la lista de vuelos posibles para cubrir el pedido del pasajero, mostrando número de vuelo, fecha y hora. El pasajero debe seleccionar uno ó más de los vuelos. A continuación el sistema le pide el nombre de los pasajeros y realiza la reserva mostrándole al usuario un número de reserva.

Para hacer el *check in*, el usuario ingresa su número de reserva y el sistema le muestra los posibles asientos para el vuelo. El usuario selecciona uno ó más asientos según cantidad de pasajeros reservados. El sistema le muestra su boleto electrónico para imprimir. El *check in* electrónico sólo puede realizarse con al menos 48 horas de anticipación al vuelo.

Se pide

- a) Realizar diagrama de casos del uso.
- b) Modelar el caso de uso: *Reserva de pasaje*
- c) Modelar el caso de uso: *Check in*

Ejercicio 22 (26 puntos)

Una empresa con 4000 empleados cuenta con un sistema para la gestión de sus recursos humanos desarrollado internamente hace más de 15 años. El sistema presenta crecientes costos de mantenimiento, y ya hace tiempo que se maneja la idea de su sustitución por un paquete de aplicación, como forma de bajar los costos de mantenimiento y a la vez incorporar funcionalidad adicional como gestión de la capacitación y competencias del personal.

Por otra parte, existen cambios en la tributación que actualmente continúan realizándose de manera manual, ya que no han sido incorporados al sistema aún.

Se plantean las siguientes alternativas:

- I. realizar los cambios sobre el sistema existente, y continuar con su mantenimiento.
- II. sustituir el sistema existente por uno que la soporte los cambios requeridos.

Se le encomienda estudiar las distintas alternativas, para lo cual le piden:

- a) WBS de la alternativa I y una identificación preliminar de riesgos relevantes.
- b) WBS de la alternativa II y una identificación preliminar de riesgos relevantes.

Ejercicio 23 (12 puntos)

- a) ¿Qué entiende por “calidad del software”? ¿Qué significa que un software goce de buena calidad?
- b) ¿Cuál es la diferencia entre calidad en el producto, y calidad en el proceso?