- 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. Con respecto a las cualidades del software:

a) Una adecuada modularidad incide favorablemente en la verificabilidad.
b) (a), y también en la eficiencia, en la facilidad de uso y de aprendizaje, y en la eficiencia en el uso.
c) (a) y en la mantenibilidad.
d) Todas son correctas.

2. Considerando los modelos de proceso:

a) En cascada se caracteriza porque no se debe comenzar una actividad hasta tanto la anterior no haya concluido satisfactoriamente.
b) (a) y es particularmente adecuado para proyectos con requerimientos no del todo claros y/o cambiantes.
c) RUP (Rational Unified Process) es incremental e iterativo, está centrado en la arquitectura y guiado por Casos de Uso, y es una adaptación del modelo en espiral para proyectos de corta duración.
d) Todas son correctas.

3. Una empresa solicitó que se desarrollara una aplicación consistente en registrar un formulario de encuesta por Internet y la transferencia de la información de las encuestas de un período a una planilla electrónica. Considerando los ponderadores siguientes y que tanto las transacciones como los archivos involucrados son de complejidad baja y que el coeficiente de ajuste tiene un valor de 0.8, la aplicación tiene un tamaño de:

Característica\Complejidad	Baja	Media	Alta
Entradas Externas (EI)	3	4	6
Salidas Externas (EO)	4	5	7
Consultas Externas (EQ)	3	4	6
Archivos Lógicos Internos (ILF)	7	10	15
Archivos de Interfaz Externa (EIF)	5	7	10

a) 6 puntos de función sin ajustar
b) 4.8 puntos de función ajustados
c) 13 puntos de función sin ajustar
d) 9.6 puntos de función ajustados

4. Una organización de desarrollo lleva un registro de los defectos mayores detectados en desarrollo y en explotación para cada uno de sus productos y del esfuerzo requerido para su solución. Para uno de ellos se tienen los datos siguientes referidos a dos versiones distintas.

Versión	Tamaño	Unidad de Medida	Lenguaje	Defectos Mayores en desarrollo	Esfuerzo (hs.) corrección desarrollo	Defectos Mayores reportados (1er. Año de explot.)	Esfuerzo (hs.) corrección durante 1er. Año explot.	Total
2.0	30	KloC	Java	50	100	6	60	56
3.0	600	PF	Java	200	200	3	250	203

Considerando que cada Punto de Función corresponde en promedio a 50 LoC Java, y sólo tomando en cuenta los defectos y/o el esfuerzo de corrección de estos, se puede decir que:

a) 3.0 es de mejor calidad para el usuario que 2.0 porque la densidad de defectos detectada durante el 1er. año de explotación resultó menor.
b) 2.0 es de mejor calidad que 3.0 porque considerando desarrollo y explotación se detectaron menos defectos.
c) 2.0 es de mejor calidad (desde el punto de vista estricto de la organización de desarrollo) que 3.0. porque el esfuerzo necesario para corregir los defectos detectados en explotación resultó menor.
d) Se cumplen a) y c)

5. Con respecto a la Gestión de Comunicaciones en un proyecto:

a) Una alternativa para disminuir la cantidad de enlaces de comunicación en un grupo consiste en tener un coordinador central por grupo, que gestione y centralice las comunicaciones.
b) a) y cuanto menor es el grupo, mayor es el número de enlaces de comunicación que existen entre sus miembros.
c) b) y siendo *n* la cantidad de miembros en un grupo, existen *n*(n-1)/2* líneas de comunicación.
d) c) y el ambiente físico de trabajo (luminosidad, ruido, etc.) no es un factor que afecte la comunicación en un grupo.

6. Con respecto a las Identificación de los Riesgos:

a) Durante esta etapa se decide cómo se llevarán a cabo las actividades de gestión de riesgos, y por quiénes.
b) La identificación de nuevos riesgos es una tarea exclusiva de la gerencia del proyecto.
c) Durante la actividad de seguimiento y control de riesgos se pueden detectar riesgos nuevos o cambios en los ya identificados.
d) La identificación de riesgos se realiza solamente al comienzo del proyecto, ya que es el momento en el cual se identifica la mayor cantidad.

7. Durante la etapa de obtención de requerimientos:

a) Cuando existen muchos usuarios, podemos sustituir la entrevista grupal por la generación de cuestionarios ya que es una forma barata de obtener requerimientos.
b) Las sesiones de trabajo son mejores que las entrevistas individuales, ya que se realizan con varios participantes ahorrando tiempo y no requieren trabajo previo de preparación.
c) Al realizar las entrevistas el principal objetivo es lograr conocer qué es lo que el cliente considera una buena solución, siendo importante seleccionar los participantes con cuidado y prepararla adecuadamente.
d) La técnica de prototipado no es útil en el caso de requerimientos cambiantes pero sí lo es para relevar detalles en los casos de uso.

8. Con respecto a 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) Es el proceso de comprender y controlar los cambios en los requerimientos del sistema.
c) Debe permitir conocer el impacto de un cambio de los requerimientos en el proyecto.
d) Se cumplen b) y c)

9. La construcción de prototipos:

a) Sólo es adecuada para validar requerimientos, buscando confirmar qué es lo que espera el cliente o usuario.
b) Permite reducir ciertos riesgos asociados al diseño de la interfaz de usuario o la funcionalidad requerida.
c) (b) y/o a la aplicabilidad/factibilidad de una arquitectura.
d) Normalmente se lleva a cabo al final del proceso de desarrollo para facilitar la prueba de aceptación.

10. Evaluar la adecuación de distintos estilos de arquitectura a una aplicación en particular:

a) Implica un gasto de horas innecesario, ya que si luego de la implementación se detecta que la arquitectura no es correcta, esta puede arreglarse fácilmente.
b) Permite ver las ventajas y desventajas que aportan las distintas soluciones a los requerimientos funcionales y no funcionales del sistema.
c) a) y en general las características de la arquitectura que afectan las características del producto final, son bastante difíciles de evaluar en forma previa.
d) b) y muchas veces se tienen requerimientos encontrados por lo cual se debe buscar una solución intermedia que contemple en parte los distintos intereses.

11. El estilo de arquitectura en capas:

a) Permite ver y comprender el sistema fácilmente, y distribuir las distintas capas en los niveles de hardware que sea necesario.
b) a) y puede ser estricto o relajado, en el primer caso una capa ofrece servicios solo a la capa inmediata inferior y pide servicios solo a la inmediata superior, si no, se está en el segundo caso.
c) b) y facilita la reutilización.
d) c) y si se definen demasiados niveles puede verse afectada la performance del sistema.

12. Con respecto al principio de bajo acoplamiento aplicado al diseño de software:

a) Aporta a la obtención de un sistema modular en el cual varios módulos podrían ocuparse de los mismos aspectos, lo cual facilita la comprensión del mismo.
b) Cuanto más acoplados se encuentran los módulos, más independientes son.
c) La cantidad de información que un módulo requiere conocer de otro es variable, pero en todos los casos debería ser la mayor posible.
d) Todas son falsas.

13. Se dice que el software falla cuando:

a) El mismo no hace lo requerido.
b) a) y/o hace algo que no debería hacer.
c) b) y las fallas solo se detectan luego de que el producto está en producción (el producto ya está liberado) ya que es el único momento en que se hacen notorias.
d) c) y un error humano introdujo una falta en el código del software.

14. Según la definición dada en el curso, ¿cuál de estas técnicas no es una técnica estática de verificación?

a) Análisis de código.
b) Análisis automatizado de código fuente.
c) Verificación formal.
d) Las técnicas descritas en a), b) y c) son todas técnicas estáticas de verificación.

15. Las pruebas de regresión:

a) Ayudan a detectar faltas introducidas al realizar modificaciones al código, chequeando que funcionalidades que estaban funcionando correctamente lo sigan haciendo.
b) (a) y no conviene tener estas pruebas automatizadas porque al usarse luego de un cambio en el código es lógico que estás pruebas también cambien.
c) (a) y es conveniente tener estas pruebas automatizadas.
d) son pruebas concebidas específicamente para detectar fallas en el rendimiento del sistema.

16. Dado el siguiente programa que devuelve como resultado la suma de todos los elementos de un array de n+1 elementos:

```
Var lista : array[0..n] of integer;
Var index : integer;
index := 2;
while index < n do
   suma := suma + lista[index]
   index ++;
end while;
return suma;
```

y suponiendo que luego de cada instrucción es necesario un ;
¿Cuál de las siguientes faltas NO es una falta clasificada como "falta de algoritmo" (según lo visto en el curso en "tipos de faltas)?

a) La variable *index* está mal inicializada.
b) Falta un punto y coma luego de la sentencia: *suma := suma + lista[index]*.
c) El *while* bifurca a destiempo.
d) Las faltas descritas en a), b) y c) son todas del tipo falta en algoritmos.

17. Con respecto a las pruebas del software:

a) Las pruebas unitarias carecen de sentido si la integración se realiza integrando de a una en una las unidades, ya que de esta manera se testea correctamente cada unidad y al agregar una única unidad en cada integración se detecta fácilmente donde reside la falta.
b) La integración de un sistema desarrollado con un lenguaje orientado a objetos y bajo un proceso iterativo e incremental debe hacerse siempre en forma bottom-up.
c) Para una unidad conviene primero ejecutar las pruebas de caja negra y luego que todas estas sean correctas complementar con pruebas de caja blanca.
d) Todas son correctas.

18. Con respecto a 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.

19. De acuerdo con CMMI, el nivel 2 de madurez implica:

i. Organizaciones proactivas.
ii. Definición de métricas básicas.
iii. Éxito dependiendo de esfuerzos personales.
iv. Repetible: se puede repetir el éxito en proyectos similares.
v. Software de alta calidad.

a) Todas son correctas.
b) ii) y iv)
c) ii), iii) y iv)
d) i), iv) y v)

20. En el modelo de proceso MUM (Modularizado Unificado y Medible):

a) Durante la fase de elaboración se define y construye la línea base de la arquitectura.
b) Los casos de uso guían el desarrollo y es centrado en la arquitectura, cada fase tiene objetivos específicos que se deben cumplir.
c) Se cumplen a) y b)
d) Se cumplen a) y b), y la duración de cada fase es fija y no puede cambiarse.

Ejercicio 21 (22 puntos)

Considere la bedelía de la Facultad de Ingeniería. Se desea que todo usuario que ingrese a su sitio web esté registrado, cada usuario debe ser autenticado en el sistema, ya sean funcionarios o alumnos. Se le permitirá a dichos usuarios ingresar su nombre de usuario y su contraseña secreta, que luego será validada por el sistema contra los datos registrados en el momento del alta del usuario. Una vez que el usuario ingresó al sitio, puede ver el resultado de una prueba, para eso el alumno debe elegir el curso dentro de la carrera, esta opción solo es válida para usuarios alumnos. Solo se debe mostrar la nota de ese alumno y no del resto de sus compañeros. Cada alumno puede además consultar, si lo desea, la fecha de la muestra para dicha prueba.

Se pide:

a) Realizar diagrama de casos del uso.
b) Modelar el caso de uso: *Ingresar al sitio*.
c) Modelar el caso de uso: *Ver resultados.*

Ejercicio 22 (26 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.

Se pide:

a) Construya una 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.
b) Identifique los riesgos más relevantes, indicando de qué tipo (del proyecto, producto, o negocio) son.

Ejercicio 23 (12 puntos)

Defina condiciones y casos de prueba para un programa que recibe un valor entre 0 y 999,99 con hasta dos decimales y devuelve un texto con el número expresado en palabras.