- Cada pregunta de la **parte múltiple opción** contestada correctamente tiene un valor de 2,8 puntos. Cada pregunta incorrecta de la múltiple opción resta 0,93 puntos. Esta parte consta de 25 preguntas por lo que vale en total 70 puntos.

# Parte 1 Múltiple Opción

1) Usted es el arquitecto de un proyecto que consiste en el desarrollo de un sistema de software para un agente recaudador. Este sistema tiene como principal objetivo registrar el cobro de las diferentes facturas de servicios, y comunicar sobre esto a los agentes emisores (por ejemplo, UTE, ANTEL, OSE). El agente cuenta con miles de clientes en todo el país, y con sucursales en la mayoría de los departamentos. A futuro, se desea continuar creciendo de manera de poder llegar a todos los departamentos del país.
¿Cuál es el estilo arquitectónico que propondría?

   a) Arquitectura Cliente Servidor en dos niveles, con cliente grueso, ya que al contar con una gran cantidad de clientes se requiere de un intensivo procesamiento a nivel de la interfaz de usuario. La instalación del cliente grueso en cada puesto de trabajo se realizará bajando una aplicación de instalación disponible en Internet (las actualizaciones se realizarán de la misma manera).
   b) Arquitectura Cliente Servidor en tres niveles, con cliente fino. Se usará un navegador para ejecutar el cliente fino y el mismo no requiere de instalación. Se utilizarán web services para la comunicación con los agentes emisores.
   c) Arquitectura basada en pizarrón. Al momento de registrar el cobro el sistema guardará los datos en una base de datos. El pizarrón ejecutará los procesos adecuados al momento que los datos cambien en la base. La comunicación con los agentes emisores se realizará mediante un middleware que utilizará objetos distribuidos. Estos objetos estarán en servidores en cada una de las sucursales del agente recaudador. Esta arquitectura permitirá el continuo funcionamiento del sistema mediante replicación automática tanto de datos como del software.
   d) Arquitectura basada tubos y filtros. Los filtros procesarán los datos de la factura. El primer filtro identificará el agente emisor. El segundo filtro identificará el monto. El tercer filtro enviará los datos al agente emisor. El cuarto filtro identificará los posibles problemas de comunicación con el emisor. El quinto filtro imprimirá un comprobante para el cliente que ha pagado la factura.

---

2) Sobre arquitectura de software.
   a) Si la arquitectura de un sistema es mantenible en el momento en el cual se pone en producción, ésta propiedad no se pierde durante el proceso de mantenimiento del software, ya que cambiar la arquitectura de un producto ya liberado es muy difícil y requiere mucho esfuerzo.
   b) Evaluar los atributos de calidad de la arquitectura es muy difícil y se realiza muy poco ya que no existen métodos ni procedimientos para ello.
   c) La arquitectura se evalúa en el momento que se termina de definir, después ya no es necesario.
   d) Los requerimientos no funcionales de un sistema están íntimamente relacionados con atributos de calidad de la arquitectura de software del mismo.

---

3) Si se encuentra trabajando en un proyecto de desarrollo de software de grandes dimensiones con equipos de trabajo distribuidos, ¿qué aspectos debe cuidar fuertemente al momento de realizar el diseño del sistema?:
   a) Comunicar las decisiones de diseño a los verificadores ya que las pruebas de regresión son fundamentales.
   b) Definir claramente el comportamiento e interfaces de cada módulo.
   c) Minimizar la cohesión para lograr minimizar la complejidad del sistema.
   d) (b). Debido a que el proyecto es con equipos distribuidos es imprescindible utilizar metodologías ágiles.

---

4) Sobre diseño de software
   a) El diseño de la interfaz de usuario no es importante en sistemas interactivos. Esto se debe a que el software normalmente tiene muchos defectos y solamente con funcionar correctamente basta para satisfacer a un usuario.
   b) El diseño de software de bajo nivel debe de ser lo más simple y eficiente que sea posible. Tener diseños simples ayuda a obtener software con menos defectos y más fácil de mantener.
   c) Durante el desarrollo de la interfaz de usuario no es conveniente considerar la experiencia de los usuarios. La experiencia es siempre variada y considerarla no tiene ningún sentido ya que implicaría desarrollar una interfaz de usuario particular para cada usuario. Sin embargo, sí es importante considerar las expectativas de los usuarios y desarrollar la interfaz basándose en las mismas.

d) El diseño de software debe ser ambiguo ya que es imposible no tener ambigüedades cuando se está diseñando. Las ambigüedades son siempre eliminadas durante la codificación. Para codificar es muy apropiado utilizar estándares de codificación por diversos motivos.

---

5) Sobre las pruebas de desempeño:
   a) Se realizan a nivel de sistema, no se pueden realizar ni a nivel unitario, ni a nivel de integración
   b) Solamente se pueden ejecutar teniendo el sistema completo (todas las funcionalidades y CU implementados), no se pueden realizar en una versión parcial del sistema.
   c) Se deben realizar únicamente en el ambiente en producción, ya que las configuraciones de hardware, software, instalaciones de redes y otros, afectan el desempeño en el sistema.
   d) En muchas ocasiones es necesaria la utilización de herramientas y software adicionales que generen grandes volúmenes de datos o simulen numerosos accesos simultáneos al sistema. Cuando se utilizan estas herramientas hay que tener cuidado en cómo éstas influyen en el consumo de recursos, pudiendo variar las mediciones de desempeño del sistema (ya que éstas también consumen recursos como ser: procesador, disco, memoria, etc).

---

6) Las pruebas del sistema:
   a) Se deben realizar a continuación de las pruebas unitarias, independientemente de las características del proyecto y modelo de desarrollo seleccionado.
   b) Pueden ser de aceptación, funcionales, de desempeño, etc.
   c) (b). Las pruebas de desempeño implican, justamente, que el sistema se desempeñe correctamente respecto de los requerimientos funcionales definidos por el cliente.
   d) (b) La Inspección de software es menos eficiente en la detección de defectos que cualquier técnica dinámica (pruebas) de verificación de software.

---

7) Respecto a la verificación:
   a) Las pruebas unitarias solamente las puede realizar quien desarrolló la unidad que está bajo prueba.
   b) En las pruebas de integración es de suma importancia conocer las interfaces del sistema.
   c) Las pruebas de sistema es razonable que las realice un equipo especializado, aunque a veces, por el tamaño de una organización no se puede contar con un equipo de estos y quizás tampoco contratarlo.
   d) (b) y (c) son correctas.

---

8) Acerca de las estrategias de integración:
   a) Big-Bang es considerada una estrategia no incremental, mientras que Bottom-Up y Top-Down son consideradas estrategias incrementales.
   b) (a) y cuando la integración es utilizando Big-Bang no se hacen pruebas unitarias.
   c) (a) y en Bottom-Up se comienza por los módulos "base" (aquellos que no usan otros módulos) y se sigue hacia arriba según la jerarquía "usa".
   d) (a) y las pruebas de regresión solamente se utilizan cada vez que se libera (al cliente) una nueva versión del software ya que el objetivo de las mismas es conocer si hubo una regresión entre versión y versión liberada.

---

9) Respecto a la verificación
   a) Las técnicas de verificación dinámica tienen una validez general y no dependen del caso de prueba específico
   b) La verificación estática no sirve para verificar el documento de especificación de requerimientos
   c) Para un proyecto de desarrollo se debe definir qué tipo de técnica usar (o estática o dinámica). Esto se debe a que las mismas no son complementarias.
   d) Ninguna de las anteriores

---

10) Acerca de defectos (faltas) y fallas:
   a) Si un programa que acepta dos datos de entrada y tiene que devolver como resultado uno de los valores "Verdadero", "Falso", se prueba con 80.000 parejas de datos de entrada distintos y falla con la mitad de esas parejas, esto significa que el programa presenta al menos 40.000 defectos.
   b) Si luego de probar un programa con un número suficientemente grande de datos de entrada distintos no aparece ninguna falla, esto no significa que el programa carece de defectos.
   c) Si un programa tiene un defecto y se ejecuta un número suficientemente grande de veces, necesariamente se va a producir una falla (va a tener un comportamiento que no es el adecuado).

d) Durante la prueba de un programa, la razón "cantidad de casos de prueba que generaron una falla" / "cantidad de casos de prueba ejecutados" es un buen indicador de la densidad de defectos (faltas) que presenta el programa.

---

11) Entre los atributos de calidad relevantes para un producto de software que da soporte a una central telefónica empresarial están:
   a) adecuación al uso, eficiencia en el uso de los recursos, confiabilidad, seguridad de la información (security)
   b) confiabilidad, seguridad de la información (security), seguridad de la integridad física (safety), facilidad de verificación.
   c) adecuación al uso, amigabilidad de la interfaz de usuario, modularidad, facilidad de verificación.
   d) facilidad de aprendizaje, calidad de la documentación técnica, facilidad de mantenimiento, modularidad.

---

12) Respecto a modelos de proceso de software:
   a) La planificación de un proyecto de desarrollo de software, sea de forma explícita o no, se basa en un modelo del proceso de software.
   b) (a), y contar con un modelo de proceso permite razonar sobre la forma mediante la cual construimos software, en particular para encarar la mejora del proceso, por ejemplo para generar productos de mejor calidad, a menor costo o en menor plazo.
   c) Una de las ventajas de un proceso en fases con evaluaciones parciales internas es que permite acortar el plazo para poder contar con resultados de utilidad, lo que a su vez permite atender la cada vez más buscada reducción del plazo para salir al mercado (time-to market).
   d) El esquema ETVX solo es adecuado para especificar modelos de proceso descriptivos.

---

13) Dados los siguientes proyectos y modelos de proceso.

Proyecto:

A - de alto riesgo, muy complejo, alcance no del todo definido, varios subconjuntos de funcionalidad identificados como útiles para los usuarios, larga duración (dos años o más).

B – Riesgo medio, alcance definido, subconjunto de funcionalidad definido y útil para los usuarios, duración media (seis meses).

C – Riesgo bajo, simple, alcance definido, corta duración (seis semanas).

Proceso:
1 – Cascada

2 – En Fases con Evaluaciones Parciales Internas

3 – En Fases con Liberaciones Parciales en Producción

4 – De Prototipación

5 - Cascada con prototipos

6 - Modelo V

7 – Especificación Operacional

8 - Espiral

Marque la asignación que le parezca más adecuada. Si un mismo proyecto aparece asignado a más de un proceso, debe entenderse que cualquiera de los dos resultaría adecuado.

Para los procesos 2 y 3 se puede especificar además entre paréntesis el modelo de proceso a utilizar en cada fase. Por ejemplo 2(1) significa Proceso en Fases con Evaluaciones Parciales Internas y en cada fase se aplica Cascada. 3(2(1)) significa que en este caso en cada fase de 3 se aplica En Fases con Evaluaciones Parciales Internas y en cada fase de 2 se aplica Cascada.

a) A3(2(1)), B3(2(1)), B3(1), C1, C2(1)
b) A2(3(1)), B3(2(1)), B3(6), C2(7)
c) A4, B2(1), C3(1), C1, C6
d) A1, A5, B1, B3(1), C3(2(1)), C8

---

14) Respecto a la liberación del software:
   a) durante el período inicial de implantación de un sistema tan solo se presentan problemas relacionados con defectos presentes en el producto de software y no identificados previamente en el proceso de desarrollo.
   b) involucra el entrenamiento a usuarios, la documentación, dar soporte a la solución de problemas y en aquellos casos en que el software sustituye a un sistema previo, preparar y realizar la conversión del sistema anterior al nuevo.
   c) (b) y la planificación de la liberación debe comenzar en el momento de considerar el alcance del proyecto para definir si los aspectos involucrados en la liberación de software están comprendidos en el alcance del proyecto.
   d) como regla general, la mejor estrategia de conversión de un sistema previo a uno nuevo es la big-bang, ya que evita los problemas relacionados con la convivencia del sistema previo y nuevo.

---

15) Con respecto al mantenimiento de software
   a) Es una actividad que representa una porción muy menor del costo total del software considerando la totalidad de su ciclo de vida.
   b) Suelen aparecer conflictos entre las urgencias por resolver problemas originados por defectos o por introducir cambios al software, y atender a la vez la preservación de atributos de calidad relevantes del software.
   c) El costo del mantenimiento de un producto normalmente decrece a medida que se van acumulando intervenciones de mantenimiento ya que estas normalmente tienden a mejorar la facilidad de mantenimiento.
   d) Reducciones en el número de solicitudes de mantenimiento correctivo, en el tiempo promedio para implementar una solicitud de cambio o en el número de solicitudes de cambio pendientes pueden estar indicando que hay problemas de mantenibilidad.

---

16) Respecto al mantenimiento
   a) los sistemas legados son sistemas construidos hace muchos años, a menudo con herramientas y tecnologías hoy obsoletas, suelen tener costos de mantenimiento elevados y resultar críticos para las organizaciones.
   b) (a) y la mejor solución para reducir los costos de mantenimiento de los sistemas legados es realizar una redocumentación de los mismos.
   c) entre las técnicas de rejuvenecimiento se puede mencionar la re-ingeniería que consiste en realizar una ingeniería directa seguida de una ingeniería reversa del producto.
   d) la gestión de la configuración que resulta esencial en las etapas finales del desarrollo de software, carece de importancia durante el mantenimiento.

---

17) El desempeño de un proceso es:
   a) Una medida de los resultados reales obtenidos luego de su ejecución
   b) La probabilidad de que los resultados obtenidos luego de su ejecución coincidan con los estimados
   c) Una medida de la madurez de la organización
   d) Ninguna de las anteriores

---

18) La norma ISO/IEC 25000:
   a) Proporciona una guía para el uso de las nuevas series de estándares internacionales, llamados Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE)
   b) a) y establece criterios para la especificación de requisitos de calidad de productos de software, sus métricas y su evaluación.
   c) b) y la evaluación de un producto de software debe realizarse antes de su liberación.
   d) b) y define tres vistas diferenciadas en el estudio de la calidad de un producto: interna, externa y en uso.

---

19) Existen varios tipos de evaluación de procesos de software. Los estudios retrospectivos, por ejemplo, se utilizan para:
   a) Determinar tendencias o relaciones entre diferentes indicadores
   b) Determinar los responsables de los incidentes detectados luego de la liberación
   c) a) y es necesario contar con los registros correspondientes.
   d) Validar o refutar hipótesis

---

20) La Gestión de la Configuración implica:
   a) Versionar todos los cambios que se realicen en los documentos del proyecto luego de que estos hayan sido aprobados por el cliente.
   b) Definir roles que controlen todos los cambios que se efectúan en los artefactos del proyecto.
   c) Definir un proceso de gestión de cambios que sea útil durante la ejecución del proyecto.
   d) (b) y (c) son correctas.

---

21) Siempre que es necesario realizar un cambio en determinado artefacto se debe:
   a) Notificar al responsable de la gestión de la configuración sobre el cambio e impacto.
   b) Chequear si este se encuentra en la línea base del proyecto
   c) Ejecutar el procedimiento de gestión de cambios
   d) Convocar al Comité de Gestión de Cambios.

---

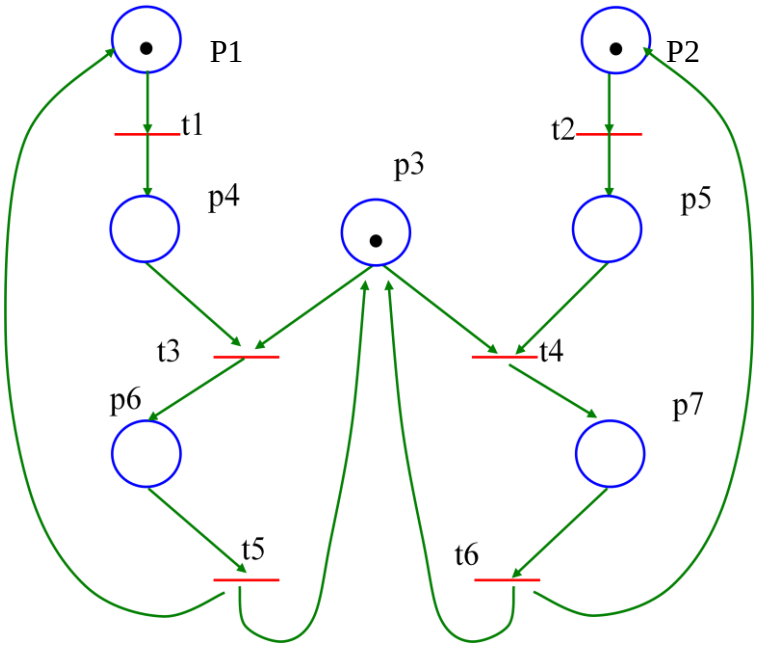
22) La empresa en la que trabaja lo ha designado para que lleve a cabo la obtención y análisis de requisitos de un proyecto muy importante:
   a) Durante la obtención de requisitos usted decide trabajar sólo
   b) Durante la obtención de requisitos usted decide trabajar en conjunto con los usuarios y clientes
   c) a) y parte de la información a relevar son las reglas de negocio, requisitos funcionales y no funcionales.
   d) b) y algunas de las técnicas que podría elegir para obtener los requisitos son entrevistas, encuestas, casos de usos y prototipado.

---

23) Marque la opción correcta respecto de la técnica de modelado de casos de uso
   a) Son la única técnica que sirve para modelar requisitos funcionales.
   b) Especifican qué es lo que el sistema debe hacer y cómo debe ser implementado.
   c) Es la técnica más adecuada para modelar requisitos que no tienen interacción con usuarios o sistemas externos.
   d) Son usados tanto para la comunicación con el cliente, ayudando al proceso de definición del alcance del sistema, como también por diseñadores y programadores, como parte de las bases para la especificación de las funcionalidades que deben diseñar o implementar.

---

24) En la siguiente red de Petri, con una marca inicial de p1, p2 y p3




a) T5 y t6 son transacciones concurrentes
b) T3 y t4 son transacciones concurrentes
c) T1 y t2 son transacciones concurrentes
d) Todas las anteriores

---

25) Sobre las técnicas de obtención de requisitos
a) Los casos de uso son la herramienta más indicada cuando nos enfrentamos a un sistema donde predominan los requisitos no funcionales.
b) Cuando se utiliza la observación como herramienta de obtención de requisitos, si bien tiene la ventaja de ser confiable y muy rica desde el punto de vista de los resultados obtenidos, no debe perderse de vista que puede generarse el efecto Hawthorne.
c) Las encuestas o cuestionarios, si bien requieren un tiempo dedicado a la elaboración de los mismos y un análisis posterior de los resultados, se utilizan comúnmente por ser más eficientes ya que pueden sustituir las entrevistas individuales brindando los mismos resultados.
d) Todas las anteriores

---

26) La línea base de requisitos del proyecto es
a) el código de la última versión del producto liberada al cliente
b) el conjunto de productos que han sido revisados formalmente y acordados
c) a) y que solo pueden ser modificados a través de procedimientos formales de control de cambios
d) b) y que pueden ser modificados solo si el director de proyecto así lo decide.

---

27) Para el seguimiento del avance de un proyecto se decide utilizar la técnica de valor ganado. Se traza una línea base calculando el valor planificado a ganar en dos iteraciones (comenzando de 0): en la primera se planifica ganar 230 y en la segunda y última 477. Cuando termina la primera iteración se mide y se tiene que el valor ganado es de 288 y el costo real de 251. Asumiendo que las actividades del camino crítico para la primera iteración están cumplidas,
a) El proyecto está atrasado y se gastó más de lo previsto para el avance obtenido.
b) El proyecto está atrasado y se gastó menos de lo previsto para el avance obtenido.
c) El proyecto está adelantado y se gastó más de lo previsto para el avance obtenido
d) El proyecto está adelantado y se gastó menos de lo previsto para el avance obtenido.

---

28) En una organización dedicada al desarrollo de software, no existen políticas de reconocimiento y recompensas al buen desempeño de sus equipos de proyecto. Según Maslow, ¿cuál de las siguientes necesidades no está siendo satisfecha?
   a) Necesidades fisiológicas.
   b) Necesidades de estima
   c) Necesidades de seguridad.
   d) Necesidades de autorrealización.

29) Uno de los riesgos identificados en un proyecto es que un ingeniero sénior con vasta experiencia en el negocio deba abandonar el equipo de proyecto en la mitad de la ejecución para atender otros proyectos en el exterior. El equipo de proyecto ha decidido que una respuesta posible a este riesgo es designar a otro ingeniero para trabajar junto con él durante el proyecto, para capacitarlo y que pueda continuar con las tareas, si fuera necesario. ¿Qué tipo de estrategia de respuesta al riesgo es esta?
   a) Mitigar el impacto.
   b) Mitigar la probabilidad de ocurrencia
   c) Evitar el riesgo.
   d) Aceptar el riesgo pasivamente.

---

30) Elija de los siguientes métodos de estimación de tamaño el más apropiado para estimar software desarrollado en C, cuya parte central (y quizás única) son algoritmos científicos complejos
   a) Puntos de función
   b) Puntos objeto
   c) Puntos de característica
   d) GxPoints