**Parcial N° 2 - Unidades 4, 5, 6 y 7**

**Ejercicio 1:**

1. Suponga que se asigna un esfuerzo estimado "Cn" a cada uno de los N casos de uso de un sistema. El puntaje ‘Cn’ se basa en el tamaño y la complejidad de cada caso de uso (pequeños valores de C indican casos de uso simples y/o pequeños, mientras que valores altos de C suponen casos de uso más complejos y de mayor tamaño. Por ejemplo: Simple: 1 <= C < 3; Medio: 4<= C < 7; Alto: 7 <= C <= 10; y es una valoración subjetiva). Suponga además que el esfuerzo total requerido para el desarrollo de los N casos de uso es E, expresado en horas-hombre. La métrica obtenida por Y=(C1+C2+...+CN)/E es una métrica de: (1)

a. Calidad.

**b. Productividad.**

c. Costo.

d. Velocidad de desarrollo.

e. Complejidad.

2. La probabilidad de la operación libre de fallas de un software durante un lapso de tiempo específico se refiere a la: (1)

a. Usabilidad.

**b. Fiabilidad.**

c. Verificabilidad.

d. Tasa de corrección de defectos.

e. Evolución de defectos a lo largo del ciclo de vida.

3. La **validación** responde a la pregunta "¿Estamos construyendo el producto apropiado?" Y mientras que la **verificación** responde a la pregunta "¿Estamos construyendo apropiadamente el producto?".

4. El cliente/usuario está involucrado necesariamente en el proceso de validación del software.

**a. Verdadero.**

b. Falso.

5. Si se desea comprobar que un sistema satisface los requerimientos del cliente, debe utilizar: (1)

a. Pruebas de verificación.

**b. Pruebas de validación.**

c. Pruebas de defectos.

d. Pruebas de integración.

e. Pruebas de sistema.

6. El proceso de desarrollo de software que adoptamos depende principalmente del tamaño del software a construir.

a. Verdadero.

**b. Falso.**

7. Una “línea base” en gestión de la configuración: (1)

a. Es producto de una etapa del ciclo de vida del software.

**b. Es una especificación que describe un acuerdo a partir del cual deben gestionarse los cambios con procedimientos aprobados.**

c. Es un punto en el tiempo de desarrollo de software en que todos los elementos de la configuración deben quedar ‘congelados’.

8. La Gestión de la Configuración del Software (GCS) le permite: (+1)

**a. No propagar defectos en el software.**

**b. Saber cuál es la última versión de la documentación del Sistema.**

**c. Generar una versión diferente del software para cada cliente.**

d. Recuperar la información de una base de datos de un cliente de la semana anterior.

e. Mantener desarrollo paralelo del software.

9. La GCS tiene su impacto y utilidad más fuerte cuando el software inicia su fase de

Mantenimiento.

a. Verdadero

**b. Falso**

10. La relación entre ‘entregas’ y ‘versiones’ de un producto de software es constante: Cada ‘versión’ del producto implica una ‘entrega’

a. Verdadero.

**b. Falso. (X - Siempre existen más versiones de un sistema que las entregas.)**

11. La información acerca de la historia de un cambio se mantiene en:

**a. El documento de especificación del programa. X**

**b. El log de cambios al repositorio del proyecto.**

**c. El registro de solicitudes de cambio. X**

d. El código afectado.

e. Ninguno de los anteriores.

**Ejercicio 2:**

Gestión de la configuración del software (GCS): (xx puntos)

a. Nombre al menos cinco (5) productos generados o utilizados en un proceso de desarrollo

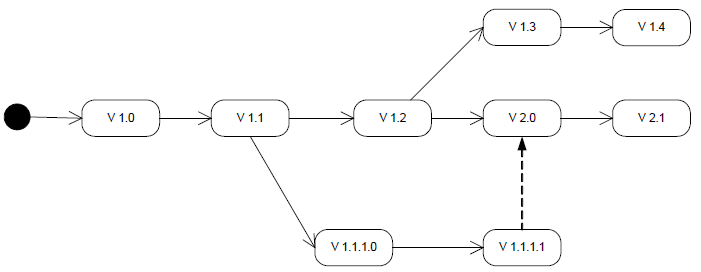
de software que podrían colocarse bajo la GCS:

* Documentos de especificación de requisitos
* Diagrama de clases
* Modelo de datos
* Documentos de casos de uso
* Diseño de módulos
* Diseño de interfaces
* Código fuente del sistema
* Casos de prueba
* Planificación de prueba

b. ¿Los elementos de configuración (EC) seleccionados para incluir bajo GCS, deben ser incorporados a la misma desde el instante de su creación? Justifique su respuesta.

No. No hay ningún parámetro para indicar cuándo es adecuado incorporar los elementos a la configuración, pero no es conveniente incluirlos demasiado antes porque genera costo y pérdida de tiempo ya que no están casi desarrollados, ni demasiado tarde porque ya están completamente desarrollados y generaría caos realizar cambios.

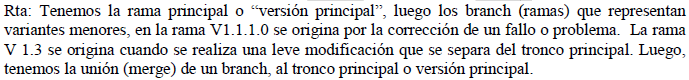
c. Dado el siguiente gráfico (grafo) que representa la evolución de un producto de software, que utiliza la convención “Id 9.9.9.9”, indique que representa cada rama del mismo.



medio: nivel -grandes cambios

abajo: instalación - pequeños cambios

arriba: versión- cambios medianos



**Ejercicio 3:**

¿Por qué motivo podríamos querer poner bajo control de configuración el software utilizado para desarrollar nuestro producto, como el compilador, el framework de desarrollo, la herramienta de modelado de Bases de datos, etc.? ¿Es recomendable esta medida? (15 puntos)

Muchas organizaciones también ponen los productos hardware y software utilizados durante el desarrollo bajo control de configuración. Como estos productos son necesarios para desarrollar el software, deberán estar disponibles cuando se realicen cambios sobre su configuración, principalmente cuando estamos hablando de software que se discontinua o tiene variantes significativas en cada versión. Aunque no es normal que haya problemas, es posible que una nueva versión de una herramienta produzca resultados diferentes que la versión usada (un compilador, por ejemplo). Es ABSOLUTAMENTE recomendable en estos casos.

**Ejercicio 4:**

**Gestión de la Calidad (xx puntos)**

Si usted debe gestionar la calidad de un proceso de desarrollo de software, en el cual el producto a desarrollar es un sitio de e-bussines para empresas exportadoras frutihortícolas, indique:

a. ¿Qué aspecto del software evaluaría para ver la calidad obtenida?

Algunos aspectos a evaluar para comprobar la calidad del software son:

* Facilidad de mantenimiento
* Flexibilidad (para introducir mejoras o modificaciones)
* Facilidad de prueba
* Corrección
* Usabilidad
* Integridad
* Eficiencia
* Confiabilidad
* Portabilidad
* Facilidad de aprendizaje

b. En este caso, el gerente de desarrollo informo que con evaluar la visión del usuario era suficiente ¿Usted considera esto correcto? ¿Por qué? Justifique.

No lo considero correcto. Si bien un software de calidad es aquel que satisface las expectativas del usuario, también es fundamental que cumpla con las expectativas del cliente, con las necesidades del equipo de desarrollo y mantenimiento y con las de otros interesados, ya sea en el producto o en el proceso.