

Soluciones Examen Parcial Ingeniería del Software II. 1 de Abril 2004.

Ejercicio 1 (4 puntos)

Contesta brevemente (15 líneas máximo cada):

- a) Comenta las ventajas y desventajas de las pruebas estáticas de código frente a las dinámicas.

Las pruebas estáticas van directamente sobre las causas de los errores, no sobre los síntomas. Pueden encontrar varios defectos en una sola sesión, estos pueden ser difíciles o imposibles de encontrar con pruebas dinámicas (por ejemplo, falta de comentarios, código mal estructurado, etc.) Como desventaja, son más costosas (implica a un conjunto de personas, normalmente).

- b) ¿Cuáles son las entradas y salidas de los procesos de ingeniería inversa lógica y de datos?.

De ingeniería inversa lógica: Entrada = Código Fuente de Programas, Salida=Diseño. De datos, entrada= Base de datos, Salida = Diagramas entidad relación.

- c) ¿Qué diferencias hay entre las pruebas de integración de un sistema estructurado y uno orientado a objetos? ¿Y entre las pruebas de unidad?.

El diseño de un sistema estructurado tiene estructura jerárquica, y por tanto se puede integrar empezando por arriba en la jerarquía, por abajo o en "sandwich". Esto no sucede con los sistemas orientados a objetos, en los que la integración se puede hacer por ejemplo cogiendo los escenarios de los diagramas de secuencia.

Las pruebas de unidad de los sistemas estructurados consisten en probar las funciones por separado. En los sistemas orientados a objetos se extienden para probar objetos enteros, y se suelen hacer siguiendo los caminos de los Diagramas de Estado asociados.

- d) ¿Cuáles son las posibles estrategias de mantenimiento, y cuál es la más rentable para la empresa encargada del mantenimiento?

Estructurado y no estructurado (y de emergencia). Aunque depende de cada caso particular, suele ser el estructurado, ya que permite una mejor planificación de los recursos.

Ejercicio 2 (4 puntos)

Dado el siguiente fragmento de código de impresión de nóminas de personal:

```
Begin
  read (Registro)
  while not eof() do
    IRPF = CalculaIRPF (Registro.Sueldo, Registro.IRPF)
    SS = CalculaSS (Registro.Sueldo)
    Salario = CalculaSalario (Registro.Sueldo, IRPF, SS)
    PrintNomina Registro.Persona, Salario
    if MesActual = 'Diciembre' and Registro.MesesTrabajados >= 6 then
      PagaExtra = CalculaPagaExtra (Registro.Extra, IRPF, SS)
      PrintNomina Registro.Persona, PagaExtra
    endif
    read (Registro)
  enddo
End
```

a) Explica qué es la cobertura de decisión/condición. (1 punto)

La cobertura de decisión/condición obliga a ejecutar cada condición al menos una vez con valor verdadero y al menos una vez con valor falso, garantizando además la cobertura de decisión, es decir, que la decisión de la que forman parte las condiciones sea al menos una vez verdadera y al menos una vez falsa.

b) Genera los casos de prueba necesarios para obtener una cobertura completa de decisión/condición. (3 puntos)

En el código del enunciado hay dos decisiones:

D1 -> "not eof()" está formada por una única condición

D2 -> "MesActual = 'Diciembre' and Registro.MesesTrabajados >= 6" está formada por dos condiciones.

C21 -> MesActual = 'Diciembre'

C22 -> Registro.MesesTrabajados >= 6

Hay que garantizar que cada condición tome al menos una vez el valor verdadero y otra el valor falso, garantizando además que se cumpla la cobertura de decisión. Los datos concretos para los casos de prueba podrían ser los siguientes:

		Valor Verdadero	Valor Falso
D1		Que haya registros que leer	Que sea fin de fichero
D2	C21	MesActual = 'Diciembre'	MesActual = 'Enero'
	C22	Registro.MesesTrabajados = 6	Registro.MesesTrabajados = 3

Si tomamos los datos de C21 y C22 que hacen que tomen los valores VERDADERO simultáneamente, la decisión D2 tomará también el valor VERDADERO. Si tomamos los datos de C21 y C22 que hacen que tomen los valores FALSO simultáneamente, la decisión D2 tomará también el valor FALSO. De esta forma se garantiza la cobertura de decisión/condición con el mínimo número de casos de prueba posibles (2 casos de prueba).

Si además consideramos la decisión D1, tenemos los siguientes 3 casos de prueba:

Caso de prueba 1: D1= Verdadero; D2 = Verdadero (C21 = Verdadero = C22)

MesActual = 'Diciembre'; Registro.MesesTrabajados = 6

Salida: escribe el salario y la paga extra

Caso de prueba 2: D1= Verdadero; D2 = Falso (C21= Falso = c22)

MesActual = 'Enero'; Registro.MesesTrabajados = 3

Salida: escribe el salario

Caso de prueba 3: D1= Falso (es decir, que no haya datos)

Que sea fin de fichero

Salida: ninguna

Ejercicio 3 (2 puntos)

1. Las empresas que realizan mantenimiento siempre están obligadas a dar mantenimiento correctivo.

Falso.

2. Todas las actividades posibles que mejoran la mantenibilidad del software se llevan a cabo a lo largo del ciclo de vida de desarrollo de un sistema software.

Falso.

3. Las pruebas de unidad siempre incluyen pruebas de caja blanca.

Verdadero.

4. El proceso completo de reingeniería incluye ingeniería inversa, y además es conveniente su automatización con herramientas.

Verdadero.

5. El plan de pruebas se genera al final de la fase de codificación.

Falso.

6. La gestión de la configuración en un conjunto de actividades de ingeniería del software que se realizan después de que el software se haya entregado al cliente.

Falso.

7. Las pruebas de caja negra se pueden definir cuando se han especificado los requisitos.

Verdadero.

8. El mantenimiento preventivo modifica la funcionalidad del software.

Falso.

9. En la ingeniería inversa según aumenta el nivel de abstracción disminuye el grado de interactividad requerido.

Falso.

10. Las pruebas de aceptación se realizan en el entorno del usuario.

Verdadero.