



INGENIERÍA DEL SOFTWARE II

CONVOCATORIA SEPTIEMBRE 2008

(TIPOS A y B)



TIPO A

Aspectos prácticos y de aplicación

(3.50 puntos. Mínimo exigido: 1,25)

1. Dado el siguiente código implementado en Fortran:

```
Program Test
  USE NumTypes
  USE Integration

  Real (kind=DP) :: Tol, In(2)

  Interface
    Function Feuler(X, Y) Result (Func)
      USE NumTypes

      Real (kind=DP), Intent (in) :: X, Y(:)
      Real (kind=DP) :: Func(Size(Y))
    End Function Feuler
  End Interface

  Tol = 1.0E-2_DP
  In(1) = 0.0_DP
  In(2) = 1.0_DP
  Write(*,*)'Valor de seno(1) y coseno(1): '
  Write(*,*)Euler(In, 0.0_DP, 1.0_DP, Feuler, Tol)
  Write(*,*)Sin(1.0_DP), Cos(1.0_DP)

  Stop
End Program Test

Function FEuler(X, Y) Result (Func)
  Real (kind=8), Intent (in) :: X, Y(:)
  Real (kind=8) :: Func(Size(Y))

  Func(1) = Y(2)
  Func(2) = -Y(2)

  Return
End Function FEuler
```

- Realice el conteo de operandos y operadores siguiendo los criterios de Fitts, Christensen y Smith.
- Aplique la *métrica de Halstead* para obtener D, E, I y λ . Razone los resultados.
- ¿Cómo se vería afectado el resultado del apartado B si se utilizara el *dit* (*Decimal digIT*) como unidad de información? Esto es, cada unidad básica de información toma 10 valores, y no 2. Razone detalladamente sus conclusiones.

(2.00 puntos)

2. Indique cómo se ven afectados los distintos atributos de calidad (rendimiento, seguridad, facilidad de mantenimiento, interoperabilidad, capacidad de prueba y portabilidad) según se escoja cada uno de los siguientes estilos arquitectónicos (arquitectura basada en flujo de datos, SOA, arquitectura basada en capas) frente a modificaciones en alguno de sus elementos, funcionamiento, etc. Razone detalladamente la respuesta para cada estilo y atributo de calidad.

(1.50 puntos)

TIPO B

Ejercicios prácticos

(4.50 puntos. Mínimo exigido: 1,75 puntos)

1. Se desea facilitar la gestión y automatizar el control de una red de aparcamientos gestionados por la misma empresa municipal, de acuerdo a los siguientes requisitos:

- Los usuarios del aparcamiento dispondrán de una tarjeta magnética donde figura registrado su código de identificación. Todo usuario es considerado como cliente.
- A su llegada a uno de los aparcamientos, el usuario introducirá la tarjeta en el lector correspondiente, lo que hace que se eleve la barrera situada en la entrada. Esta barrera permanece levantada un cierto tiempo, descendiendo luego automáticamente.
- Para salir del aparcamiento se procede de igual forma con la barrera situada a la salida.
- Tanto las entradas como las salidas deben quedar registradas con objeto de realizar periódicamente una facturación a los usuarios, según el tiempo de aparcamiento consumido. Estas facturas se emitirán a petición del operador.
- Se mantiene registro de los usos que realiza cada cliente, así como gestión de las facturaciones a los clientes.
- Un cliente es dado de alta cuando obtiene su tarjeta de usuario, que es prerequisite para hacer uso de cualquier instalación de la empresa.

Además, el sistema debe tener en cuenta la ocupación de cada aparcamiento, controlando un semáforo situado a la entrada. Si hay plazas libres, el semáforo debe estar verde, pasando a rojo si el aparcamiento se llena. Cuando el aparcamiento esté lleno no debe permitirse la entrada de nuevos vehículos.

a) Realice el diseño arquitectónico del sistema propuesto:

- a) Identificando los subsistemas que componen el sistema.
- b) Especificando los servicios y restricciones bajo los que funcionará el sistema.
- c) Asignando los servicios a los componentes y diseñando las interfaces.

(2.00 puntos)

b) Represente el diseño arquitectónico propuesto utilizando un diagrama de componentes en UML.

(1.00 puntos)

Nota: Se deberá justificar todo el proceso realizado. Todo aquella parte del diseño que no esté suficientemente justificado y razonado, se considerará nulo.

2. Se conoce que el volumen mínimo de un programa Y escrito en PL/1 es de 20 bits. Sin embargo, si se utilizase el lenguaje ensamblador, este volumen mínimo se reduciría a 15 bits. Sabiendo que un programador experimentado en ensamblador es capaz de realizar 18 discriminaciones mentales elementales por segundo, ¿cuántas decisiones elementales por segundo debería tomar un programador de Pascal para tardar el mismo tiempo que su compañero en idear el programa Y?

(1.50 puntos)