

Curso 2004/2005

Examen Intermedio, Ingeniería del Software I

19 de Noviembre de 2004.

Ejercicio 1 (6 puntos)

Eres un jefe de proyecto de una empresa informática que se dedica al desarrollo de aplicaciones de ingeniería. Tienes que desarrollar un simulador para la modelización de materiales, así como la experimentación de sus propiedades ante la descarga de electrones en diversas geometrías. Pese a que tus clientes (una empresa del sector aeroespacial) te piden que la aplicación funcione en Windows XP, no están familiarizados con aplicaciones interactivas para simulación. Así, compruebas que no tienen claros los requisitos en cuanto a interfaz de usuario y salidas gráficas. Por tu parte, no sabes cuál es el mejor procedimiento para la simulación (dudas entre eventos discretos o un esquema de integración clásico), ya que tus clientes te piden que cada material pueda simularse en 30 minutos como máximo, así que deberás usar herramientas para evaluar el rendimiento de ambas alternativas. En ambos casos reutilizarías una librería de funciones matemáticas que se ha desarrollado en tu empresa en proyectos anteriores.

La aplicación recibe como entradas (de manera interactiva) los parámetros del material, una especificación de las condiciones iniciales de la nube de electrones, una especificación del número y tipo de experimentos a realizar, así como las condiciones de final de la simulación. El simulador produce una salida textual con los resultados de cada experimento, así como cuatro salidas gráficas. El programa debe manejar una base de datos de materiales. Esta base de datos se podrá consultar por el nombre del material.

Los procedimientos matemáticos son en general complejos. Además, el usuario te pide el uso en la aplicación de una interacción típica de Windows (ventanas, menús, scrol, uso de ratón, teclas de función, ayuda on-line, etc.), así como una herramienta para su fácil instalación, ya que el simulador debe instalarse en varios laboratorios (todos PCs con Windows XP).

Se pide:

- a) Elije razonadamente el modelo de ciclo de vida más adecuado (1 punto).
- b) Calcula los puntos de función ajustados (considera medio el valor de la complejidad de los elementos) y el tamaño del programa en líneas de código en C++ si los datos históricos de tu empresa indican que son necesarias 200 líneas de código C++ por punto de función (4 puntos).
- c) ¿Hay alguna desventaja en el uso de Puntos de Función para estimar el tipo de aplicaciones descrita en el ejemplo? (1 punto).

Solución:

- a) Son necesarias las maquetas (para validar requisitos de interfaz y salidas) y prototipos. Como ciclo de vida se podría usar casi cualquiera, aunque al haber prototipos es aconsejable un modelo iterativo e incremental, como el ciclo de vida de “cascada con realimentación” o los “iterativos”.

- b) PF sin ajustar:

Entradas = parámetros del material + condiciones iniciales de la nube de electrones + especificación del número y tipo de experimentos a realizar + condiciones de final = 4

Salidas = 1 texto + 4 gráficas = 5

Consultas = 1 consulta por el nombre del material = 1

Ficheros Internos = BD de materiales = 1

Total = $4 \times 4 + 5 \times 5 + 1 \times 4 + 1 \times 10 = 16 + 25 + 4 + 10 = 55$ PF

Factor de ajuste:

- 1. Comunicación de datos: 0
- 2. Funciones distribuidas: 0
- 3. Rendimiento: 5 (uso de herramientas para evaluar rendimiento)
- 4. Configuraciones fuertemente utilizadas: 2 (restricciones de tiempo).
- 5. Frecuencia de transacciones: 0
- 6. Entrada on-line de datos: 5 (todas entradas on-line)
- 7. Diseño para la eficiencia del usuario final: 3
- 8. Actualización on-line: 0
- 9. Procesos complejos: 1 (procesos matemáticos complejos)
- 10. Reusabilidad: 1 (se reutilizan librerías)
- 11. Facilidad de instalación: 4 (necesitan herramientas para la instalación)
- 12. Facilidad de operación: 0
- 13. Instalación de múltiples sitios: 1 (uso en varios laboratorios con XP)
- 14. Facilidad de cambio: 0

$TDI = 5+2+5+3+1+1+4+2=22$

$AF = TDI \times 0.01 + 0.65 = 0.22+0.65 = 0.87$

$FPA = 55 \times 0.87 \cong 49$ PF

$LDC \text{ “C++”} = 49 \times 200 = 9800$ LDC

- c) Los puntos de función no se ajustan demasiado bien a la evaluación de software de tipo científico, donde la complejidad está en los algoritmos, no en la interacción E/S con el usuario o en el manejo de fichero. Los PF son mejores para evaluar aplicaciones de gestión.

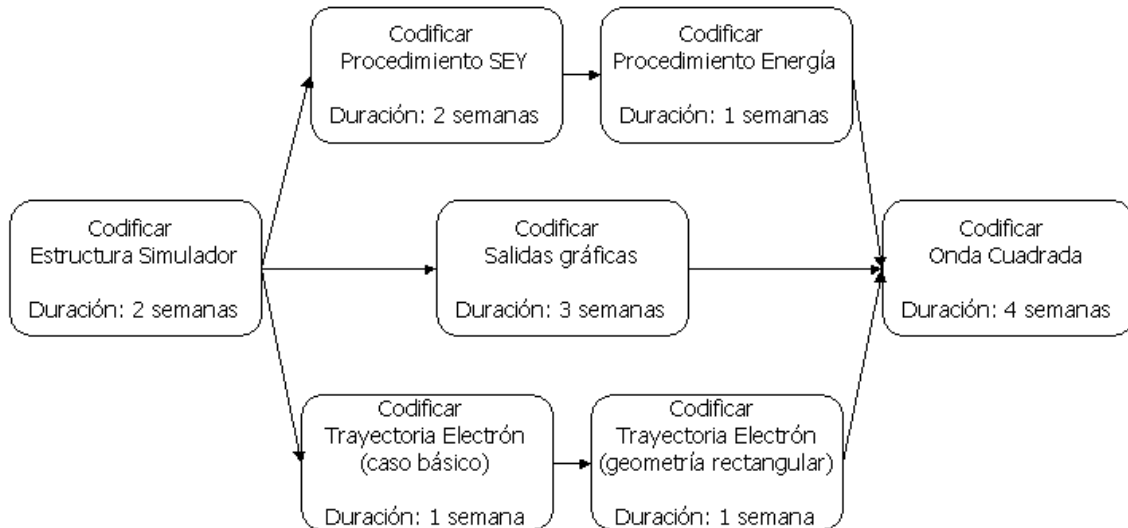
Ejercicio 2 (2,5 puntos)

Indica si las siguientes afirmaciones son verdaderas o falsas:

- a) El tamaño en puntos de función de una aplicación es independiente del lenguaje de implementación. *(V, es una estimación desde el punto de vista de los requisitos de la aplicación)*
- b) El modelo GANTT es un modelo dinámico a lo largo del ciclo de vida de un proyecto software y el modelo PERT es estático. *(F, los dos son dinámicos, evolucionan con el tiempo)*
- c) En Ingeniería del software los procedimientos definen la secuencia en la que se aplican las herramientas. *(F, los procedimientos definen, principalmente, la secuencia en la que se aplican los métodos)*
- d) Dentro del marco de la estimación, las técnicas empíricas excluyen a las de descomposición. *(F, normalmente hay que hacer primero descomposición y luego empíricas y, en cualquier caso, no se excluyen)*
- e) Las fases básicas de un ciclo de vida de un proyecto software no varían independientemente que el propósito del software sea una aplicación a medida que sea una herramienta comercial. *(V, las fases son las mismas. Las actividades en fases como requisitos, diseño, pruebas y entrega no).*

Ejercicio 3 (1,5 puntos)

- a) Dibuja el diagrama de GANTT equivalente al siguiente diagrama de PERT, correspondiente a la fase de codificación de un proyecto. **(1 punto)**
- b) Indica cuál es el camino crítico. **(0.25 puntos)**
- c) ¿Cuál es la holgura de la tarea “Codificar Procedimiento Energía”? **(0.25 puntos).**



Solución:

a)

Tarea	1	2	3	4	5	6	7	8	9
Codificar Estructura Simulador	X	X							
Codificar Procedimiento SEY			X	X					
Codificar Procedimiento Energía					X				
Codificar Salidas Gráficas			X	X	X				
Codificar Tray. Electrón (caso básico)			X						
Codificar Tray. Electrón (geom. Rectangular)				X					
Codificar Onda Cuadrada						X	X	X	X

b) hay dos:

cod.estruct.simulador – codificar proc. SEY – cod. Proc-energía – codif. Onda cuadrada
 cod.estruct.simulador – codificar Salidas gráficas – codif. Onda cuadrada

c) No tiene holgura, es parte de un camino crítico.