

Examen parcial 7 de marzo de 2012

Apellidos y Nombres: _____

DNI: _____

Grupo: _____

Consideraciones básicas: *por favor, léanse detenidamente antes del inicio del examen*

1. El estudiante escribirá su nombre y apellidos en TODAS las hojas de examen que se le entreguen y deberá entregar TODAS LAS HOJAS utilizadas al terminar el examen con su nombre y apellidos.
2. Se recuerda que el estudiante TIENE LA OBLIGACIÓN de custodiar ACTIVAMENTE las hojas y otros materiales suyos con los que trabaje en el examen, manteniéndolos fuera del alcance visual o físico de otros estudiantes. El no hacerlo así se considerará como indicio de participación en copia.
3. De detectarse casos de copia, se aplicará la normativa vigente de Evaluación de Asignaturas de la EPS, lo que supone en primera instancia el suspenso de todos los implicados, bien sean fuentes o receptores, sin perjuicio de otras medidas disciplinarias que puedan aplicarse.
4. Antes de responder a cada pregunta, se han de leer atentamente los enunciados de los problemas planteados. Para cada pregunta se indica su puntuación correspondiente.
5. No se pueden utilizar libros ni apuntes ni aparatos electrónicos de ningún tipo.
6. Si el estudiante realiza alguna suposición, anotará en el examen cualquier consideración que haya tenido en cuenta.
7. El tiempo de realización del examen será de dos horas.
- 8. El ejercicio número 2 se realizará en la propia hoja de enunciado.**

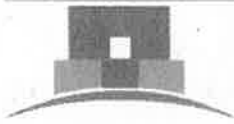
Ejercicio 1 (2,1 puntos)

Responde brevemente (máximo cuatro líneas por pregunta) a las siguientes cuestiones:

- a) ¿En qué casos se puede prescindir de la definición del ciclo de vida al comienzo de un proyecto software?
- b) ¿Es posible realizar maquetas en modelos de ciclo de vida no iterativos, como el cascada sin realimentación? ¿Por qué?
- c) La selección de un modelo de ciclo de vida, ¿engloba la selección de un determinado modelo de proceso?

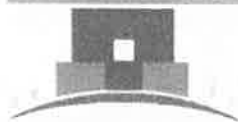
Ejercicio 2 (2,8 puntos)

Señala, en cada pregunta, la respuesta verdadera de las tres posibles opciones (0,4 puntos por respuesta verdadera y -0,1 por respuesta errónea)



Nombre:

1. ¿Cuál de estas definiciones es correcta y completa?
 - a. El objetivo principal de la Ingeniería del Software es obtener software que funcione bien.
 - b. El objetivo principal de la Ingeniería del Software es obtener software que utilice adecuadamente los recursos hardware.
 - ☒ c. El objetivo principal de la Ingeniería del Software es obtener software eficiente y fiable.
2. Se considera una guía de buen diseño:
 - ☒ a. Dividir los módulos de forma que sean lo más independientes posible (mínimo acoplamiento) y de forma que cada módulo lleve a cabo una sola operación (máxima cohesión).
 - b. Unificar los módulos de forma que sean lo más dependientes posible (mínimo acoplamiento) y de forma que cada módulo lleve a cabo varias operaciones (máxima cohesión).
 - c. Llevar a cabo una codificación uniforme que dé lugar a buenos módulos de diseño, en general.
3. En software, las metodologías normalmente:
 - a. Cubren todo el ciclo de vida de mantenimiento.
 - ☒ b. Cubren todo el ciclo de vida de un proyecto software.
 - c. Cubren todo el ciclo de vida de desarrollo.
4. Los elementos que define un modelo de ciclo de vida son:
 - a. Las fases y sus relaciones.
 - ☒ b. Las fases, sus relaciones y los criterios para la transición de fases.
 - c. Los procedimientos y herramientas para la ejecución de la fase.
5. ¿Cuándo se deben definir los criterios de aceptación de las pruebas?
 - ☒ a. Al final de la fase de análisis.
 - b. Al principio de la fase de pruebas.
 - c. Al final de la fase de pruebas.
6. Las técnicas de pruebas permiten:
 - a. Validar y verificar todos los productos intermedios y finales.
 - b. Validar y verificar el software cuando ya está en forma de código ejecutable.
 - ☒ c. Validar y verificar el software a lo largo de todo el ciclo de vida.
7. Un caso de prueba está formado por:
 - a. Los datos de entrada y la salida esperada.
 - b. Los datos de entrada y la salida generada.
 - ☒ c. Los datos de entrada y las salidas esperada y generada.



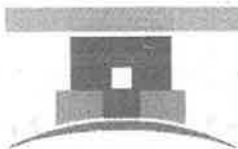
Nombre:

Ejercicio 3 (3,1 puntos)

Dada la siguiente función en lenguaje Java:

1	boolean fechaCorrecta (int dia, int mes, int año){
2	int limite;
3	boolean bisiestro;
4	if (año<2005){
5	bisiestro=((año%4==0)&&(año%100!=0)) (año%400==0) ;
6	switch (mes) {
7	case 1:
8	case 3:
9	case 5:
10	case 7:
11	case 8:
12	case 10:
13	case 12: limite=31;
14	break;
15	case 2: if (bisiestro)
16	limite=29;
17	else limite=28;
18	break;
19	case 4:
20	case 6:
21	case 9:
22	case 11: limite=30;
23	break;
24	default: return (false);
25	}
26	if ((dia>0) && (dia<=limite))
27	return (true);
28	else return (false);
29	}
30	else return (false);
31	}

- a) Dibujar el grafo de flujo y calcular la complejidad ciclomática. ¿Qué indica el valor obtenido? (2,5 puntos)
- b) ¿Existe alguna relación entre la complejidad ciclomática y la cobertura de decisión? En caso afirmativo, ¿cuál es esa relación? (0,6 puntos)



Nombre:

Ejercicio 4 (2 puntos)

Dada la siguiente especificación de una función "C":

```
typedef struct {
    int fila;
    char columna;
} Posición;

int MovimientoValido ( Posición pinicial, Posición pfinal);
/*
    Esta función comprueba si el movimiento hecho por el jugador se sale del tablero
    (8X8). Los parámetros de entrada son:

    pinicial: posición inicial de la ficha.
    Pfinal: posición final de la ficha.
    La fila es un número entero en el intervalo [1..8]. La columna es una letra en el
    intervalo [A..J]

    La función devuelve 1 en caso de que la posición final sea válida, cualquier otro
    número si no es válida. */
```

Se pide crear una tabla de clases de equivalencia en la que se indiquen:

- Dato de entrada o atributo que se analiza
- Clases válidas para ese dato de entrada
- Clases no válidas para ese dato de entrada.

INGENIERÍA DEL SOFTWARE I

Grado: Proyecto IS

EJERCICIOS PUNTOS DE FUNCIÓN CON SOLUCIÓN

①

Ejercicio puntos de función

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. FUI

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, scroll, 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).

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.

Solución:

PASO 1 -

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 sin ajustar

PASO 2 -

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: 2 (uso en varios laboratorios con XP) ✓
- 14. Facilidad de cambio: 0

$$TDI = 5+2+5+3+1+1+4+2=23$$

$$AF = TDI \times 0.01 + 0.65 = 0.23 + 0.65 = 0.88$$

PASO 3: PUNTOS DE FUNCIÓN AJUSTADOS

$$FPA = 55 \times 0.88 = 48.4 \approx 49 \text{ PF}$$

$$LDC \text{ "C++"} = 49 \times 200 = 9800 \text{ LDC en "C++"}$$

2

Ejercicio Puntos de Función

El departamento de marketing de una distribuidora de DVDs quiere automatizar y ampliar sus funciones. Este último objetivo se centra en enviar a domicilio un DVD regalo consistente en un DVD de lanzamiento a aquellos clientes que superen un determinado umbral de compras anuales. La base de datos que contiene estos clientes es interna a la empresa y la mantiene el departamento de ventas.

Normalmente la empresa tiene unos 30 títulos que son lanzamientos nuevos y unos 300 títulos que forman el fondo de catálogo. Los responsables encargados de gestionar ambos grupos son distintos. Así, los títulos de fondo de catálogo los gestionan dos personas cuya función es promocionarlos y enviar esta promoción al departamento de ventas de la empresa.

Los lanzamientos requieren más dedicación por lo que hay cuatro personas encargadas de su gestión, que se basa en la presentación de los mismos al departamento de ventas. Estos dos grupos de empleados, los encargados del fondo de catálogo y los de los lanzamientos, tienen que remitir informes semanales al director de marketing que los supervisa y archiva.

El resto del equipo de marketing lo componen dos diseñadores que elaboran el correspondiente material gráfico consultando los lanzamientos y el fondo de catálogo y teniendo en cuenta las directrices marcadas por el director de marketing. Una vez realizado el material, se lo envían al director de marketing para su supervisión. Para la función de envíos a domicilio se ha contratado a una persona nueva.

Todas las entradas serán interactivas. Todos los ficheros considerados deben ser actualizados de manera on line. El departamento de marketing quiere instalar la aplicación en diversas oficinas, todas con ordenadores personales con sistema operativo Windows XP. Los usuarios te han pedido un diseño para obtener la máxima eficiencia del usuario final así como la máxima facilidad de operación. Además, prevés que la gestión de los lanzamientos tendrá una lógica compleja, si bien reutilizarás código de un proyecto anterior en tu aplicación.

Nota: Supór. un valor de complejidad medio para los distintos parámetros.

Calcula razonadamente los puntos de función ajustados para la aplicación que se describe.

Ficheros lógicos internos: 3 (lanzamientos, catálogos, informes)

Ficheros lógicos externos: 1 (clientes)

Entradas externas: 1 (clientes)

Salidas externas: 3 (promociones a ventas, presentación de lanzamiento a ventas envío regalo)

Consultas: 2 (catálogo y lanzamientos por el dpto ventas). Nota: puede haber más.

Puntos de función: $10 \times 3 + 1 \times 7 + 5 \times 3 + 4 \times 1 + 2 \times 4 = 64$ *PF Sin Ajustar*

Características del sistema:

- Actualizaciones on-line: 1
- Entradas de datos on-line: 5
- Eficiencia del usuario final: 5
- Reusabilidad: 1
- Facilidad de operación: 5
- Inst. en lugares distintos: 1
- Procesos complejos: 1

TDI: $1 + 5 + 5 + 1 + 5 + 1 + 1 = 19$

FA = $(TDI \times 0,01) + 0,65 = 19 \times 0,01 + 0,65 = 0,84$

Puntos de función ajustados: $PF \times FA = 64 \times 0,84 = 53,76 \approx 54$ *PF ajustados*

4.

Ejercicio 3 - Puntos de Función

La empresa VIAJA-YA, S.L. te encarga desarrollar un proyecto software que tiene como objetivo la venta de paquetes baratos de avión+hotel de forma on-line. Los precios son muy competitivos pero el cliente debe ajustarse únicamente a las fechas que la empresa ofrece como ofertas. La empresa VIAJA-YA, S.L. gestiona una base de datos donde almacena estas ofertas que son las que presenta al usuario mediante una interfaz interactiva avanzada diseñada con un elevado número de funciones. Esta base de datos se alimenta de los vuelos que les ofrecen las compañías aéreas a través de ficheros que les llegan diariamente y de una base de datos que contiene hoteles con su precio y disponibilidad que gestiona una empresa dedicada a esto. Además las entradas on-line al sistema se estiman en alrededor de un 25% del total.

El usuario accederá a la web del sistema, verá las ofertas disponibles y podrá seleccionar ocho de ellas, de las que el sistema le responderá de forma inmediata si hay plazas libres o no. Finalmente, una vez seleccionada la opción elegida, el usuario recibe como respuesta la confirmación de su reserva.

Los usuarios pueden acceder desde cualquier PC pero la aplicación se encuentra en un único ordenador central. Dado que el tiempo de respuesta es crítico en horas punta como es por ejemplo entre las 8 de la tarde y las 12 de la noche y los fines de semana, se requiere un diseño especial que tenga en cuenta estos estrictos requisitos de rendimiento por parte del usuario.

Por otra parte, se espera ampliar en un futuro el proyecto al alquiler de coches y otras funcionalidades por lo que la aplicación se va a implementar y documentar expresamente para ser fácilmente reusable. Para garantizar el éxito del sistema, se ha hecho una labor importante de recogida de requisitos y más de un 50% de la aplicación se basará en requisitos y necesidades de varios usuarios.

La complejidad de todos los parámetros se considera media.

El lenguaje en el que se desarrollará la aplicación es JAVA y basándose en proyectos anteriores, se sabe que el ratio interno de la empresa de producción de líneas de código JAVA por punto de función es de 75. Así mismo, el coste estimado por punto de función es de 1.025 euros y a partir de estadísticas se ha establecido que una persona lleva a cabo 7 puntos de función al mes de media.

Utilizando la técnica de estimación de puntos de función, se pide:

- Calcular el número de líneas de código de la aplicación.
- Calcular el coste total de la aplicación.
- Suponiendo que la fase de pruebas consume un 40% del esfuerzo total, ¿cuál es el número total de personas-mes estimado para dicha fase?

Solución

- Ficheros Lógicos Internos: 1 (base de datos de ofertas)
- Ficheros de Interfaz Externos: 2 (ficheros vuelos y base de datos hoteles)
- Entrada: 1 (opción)
- Salidas: 2 (interfaz interactiva y respuesta reserva)
- Consultas: 8 (selección opciones)

Nota: Algunos de estos parámetros pueden variar. Por ejemplo, el fichero de vuelos se puede considerar entrada o fichero lógico externo y la respuesta reserva se puede considerar salida o consulta.

Como todos los parámetros tienen complejidad media, hallamos los puntos de función sin ajustar de la siguiente forma.

PF: $1 \times 10 + 2 \times 7 + 1 \times 4 + 2 \times 5 + 8 \times 4 = 70$ puntos de función sin ajustar.

Para hallar el TDI, miramos en las tablas los valores de las características del sistema:

- Comunicación de datos: 3
- Entradas de datos on-line: 4
- Eficiencia usuario final: 3
- Rendimiento: 4
- Reusabilidad: 3
- Frecuencia transacciones: 3

Por más de un solo de la aplicación se basará en requisitos y necesidades de varios usuarios.

Sumando obtenemos que el TDI es 20.

Factor de ajuste: $FA = (20 \times 0,01) + 0,65 = 0,85$

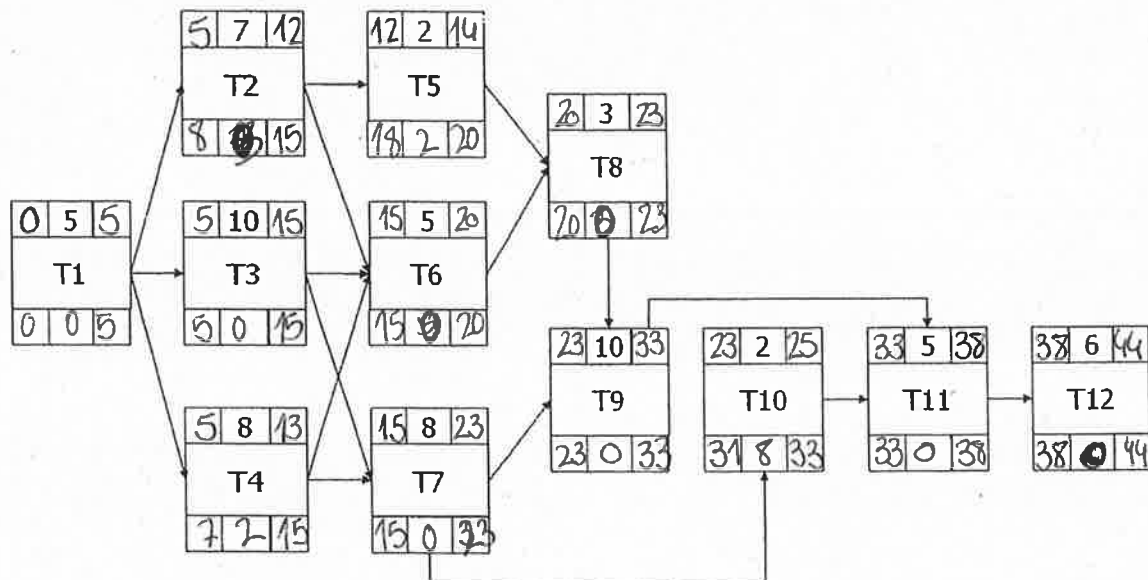
Puntos de función ajustados: $70 \times 0,85 = 59,5$ puntos de función ajustados.

- a) Líneas de código Java: $75 \times 59,5 = 4.462,5$ l.d.c. Java
- b) Coste: $1.025 \times 59,5 = 60.987,5$ €
- c) P-m fase pruebas: $(59,5 / 7) \times 0,4 = 3,4$ p-m en pruebas

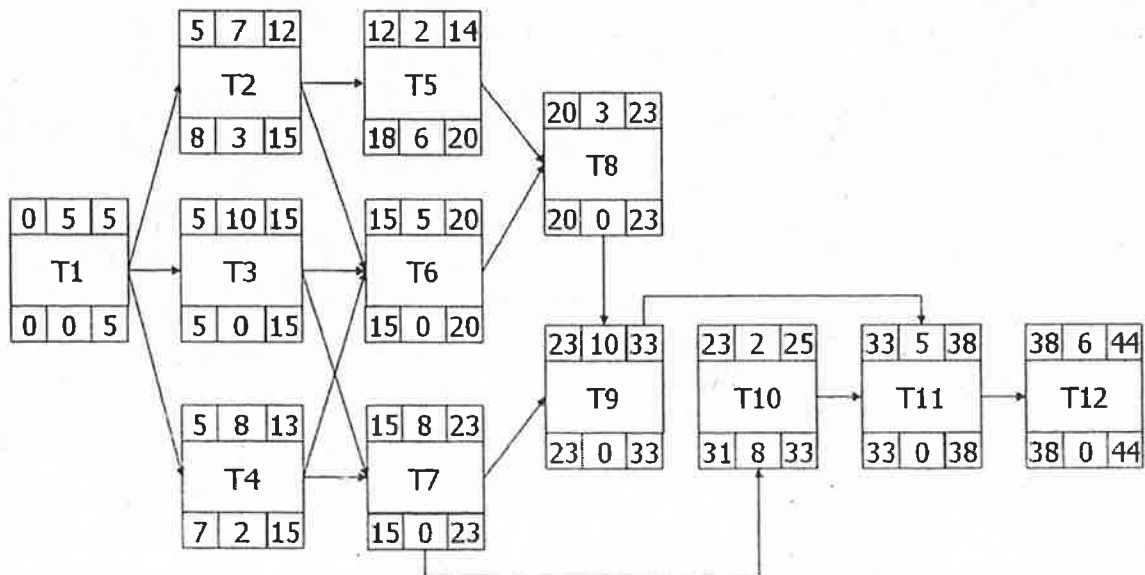
esfuerzo

EJERCICIO PERT CON SOLUCIÓN

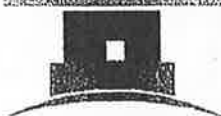
Dada la siguiente red de tareas:



1. Calcula los tiempos más tempranos/tardíos de inicio y fin de cada tarea, así como la holgura (contesta sobre el diagrama)
2. ¿Cuáles son los caminos críticos?
3. Calcula los tiempos más tempranos/tardíos de inicio y fin de cada tarea, así como la holgura (contesta sobre el diagrama)



Los caminos críticos son T1-T3-T6-T8-T9-T11-T2 y T1-T3-T7-T9-T11-T2.



CURSO: 2007/2008

EXAMEN FINAL: Febrero 2008

IDENTIFICACIÓN:

Apellidos y Nombres: _____

DNI: _____

Consideraciones básicas: *por favor, léanse detenidamente antes del inicio del examen*

1. El estudiante escribirá su nombre y apellidos en TODAS las hojas de examen que se le entreguen y deberá entregar TODAS LAS HOJAS utilizadas al terminar el examen con su nombre y apellidos.
2. Se recuerda que el estudiante TIENE LA OBLIGACIÓN de custodiar ACTIVAMENTE las hojas y otros materiales suyos con los que trabaje en el examen, manteniéndolos fuera del alcance visual o físico de otros estudiantes.
El no hacerlo así se considerará como indicio de participación en copia.
3. De detectarse casos de copia, los mismos supondrán de entrada el suspenso de todos los implicados, bien sean fuentes o receptores, sin perjuicio de otras medidas disciplinarias que puedan aplicarse.
4. Antes de responder las consignas, se han de leer atentamente los enunciados de los problemas planteados. Para cada consigna se indica su puntuación correspondiente.
5. No se pueden utilizar libros ni apuntes.
6. Si el estudiante realiza alguna suposición, anotará en el examen cualquier consideración que haya tenido en cuenta.
7. El tiempo de realización del examen será de tres horas.

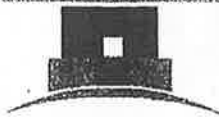
Notas: Contesta todas las preguntas en el enunciado, respeta la limitación de espacio indicado si se aplica y entrégalo al terminar.

Entrega las soluciones a los problemas en hojas separadas.

Problema 1 (2,40 puntos)

Responde razonadamente en el enunciado del examen a las siguientes preguntas. (0,4 puntos cada pregunta)

- a) Eres un jefe de proyectos software de una empresa que desarrolla aplicaciones informáticas en distintos sectores, ¿qué características, ventajas y desventajas analizarías para los tres modelos de ciclo de vida del software (MCVS) que utiliza esta empresa enumerados en la tabla siguiente?



Modelo Convencional en Cascada (con vuelta atrás)

Características:

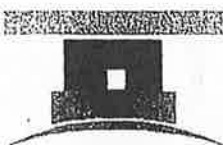
- La necesidad del usuario no se subdivide.
- El desarrollo del producto se construye completamente y de una sola vez.
- La evolución del producto software se realiza a través de una secuencia ordenada de transiciones de una fase a la siguiente según un orden lineal, aunque permite iteraciones durante el desarrollo, dentro de un mismo estado o de un estado hacia otro anterior.
- Obliga a especificar lo que el sistema debe hacer antes de construir el sistema.
- Obliga a definir cómo van a interactuar los componentes antes de construir tales componentes.
- Requiere que el proceso de desarrollo genere un conjunto de documentos que posteriormente pueden utilizarse para la validación y el mantenimiento del sistema. Es decir, es una visión del proceso de desarrollo de software como una sucesión de etapas que producen productos intermedios.
- Es el más utilizado, sencillo y más cómodo de gestionar.
- Las fases continúan hasta que los objetivos se han cumplido.

Ventajas:

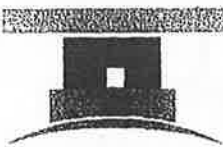
- Las etapas están organizadas de un modo lógico. Si una etapa no puede llevarse a cabo porque faltan decisiones de un nivel más alto, no se puede seguir adelante. Por ello, permite al jefe del proyecto seguir y controlar los progresos de un modo más exacto.
- Cada etapa incluye cierto proceso de revisión, y se necesita una aceptación del producto antes de que la salida de la etapa pueda usarse. Está organizado de modo que se pase el menor número de errores de una etapa a la siguiente. Por ello, permite detectar y resolver desviaciones sobre la planificación inicial de un modo más exacto.
- El ciclo es iterativo. El flujo básico es de arriba hacia abajo, no obstante, reconoce que los problemas de etapas inferiores afectan a las decisiones de las etapas superiores.

Desventajas:

- Para que el proyecto tenga éxito deben desarrollarse todas las fases. Si se cambia el orden de las fases, el producto final será de inferior calidad.
- Asume que los requisitos de un sistema pueden ser congelados al principio del proyecto y antes de comenzar el diseño, para problemas abiertos y no estructurados esto no es real.
- Cuando se entrega el sistema éste no satisface las expectativas actuales del cliente; sólo satisface las expectativas planteadas al comienzo del proyecto, pues las necesidades están en constante evolución.
- No existe un proyecto "enseñable" hasta el final del proyecto. Es decir, se entrega al cliente el primer producto cuando ya se han consumido el 99% de los recursos para el desarrollo.



Modelo Incremental Iterativo
Características:
<ul style="list-style-type: none">- Existe una decisión consciente para retrasar el desarrollo de una parte del producto.- El problema entero es subdividido en grupos antes de que cualquier producto intermedio sea desarrollado.- Se realiza el estudio del problema del usuario y a partir de allí se descompone en sub-problemas que deben ser desarrollados en diferentes etapas de manera progresiva y planificada.- Permite construir una implementación parcial del sistema global y posteriormente ir aumentando la funcionalidad del sistema.- El software se construye de modo que sólo satisfaga unos pocos requisitos, los más relevantes para el usuario y los más comprendidos por el equipo de desarrollo, de todos los que tiene el usuario.- Debe construirse de tal modo que facilite la incorporación de nuevos requisitos.
Ventajas:
<ul style="list-style-type: none">- Produce un sistema operacional mas rápidamente, reduciendo la posibilidad de que las necesidades del usuario cambien durante el desarrollo.- Se reduce el tiempo de desarrollo inicial, debido al nivel reducido de funcionalidad.- Facilita la incorporación de nuevos requisitos, por lo tanto el software es fácil de mejorar, es decir el software tiene una adaptabilidad mayor.- La mayor adaptabilidad permite que el software se desarrolle a partir de una serie de construcciones y/o mejoras claramente definidas y planificadas.
Desventajas:
<ul style="list-style-type: none">- La optimización de la primera entrega para que la demostración tenga éxito como software de exploración no podrá reflejar adecuadamente los requisitos no funcionales del sistema en versiones aumentadas (tiempo de respuesta, etc.)- Las versiones iniciales se centran en la funcionalidad y la interfaz, dejando de lado algunos temas centrales de la arquitectura que pueden ser muy costosos de desarrollar (seguridad, procesamiento distribuido, etc.)- El equipo de desarrollo sacrifica la etapa de análisis a favor de una obtención rápida de la primera versión, generando un primer producto que puede estar alejado de las necesidades del usuario.
Modelo Evolutivo en Espiral
Características:
<ul style="list-style-type: none">- Existe una decisión consciente para retrasar el desarrollo de una porción del producto.- Los productos intermedios se desarrollan antes de que la siguiente subdivisión se determine. Es decir, las evoluciones futuras del sistema no están previstas a la manera incremental, donde se debe saber qué funcionalidades se añadirán en cada incremento y cuándo se llevarán a cabo.- Representa un enfoque dirigido por el riesgo para el análisis y estructuración del proceso software.- Incorpora métodos de proceso dirigidos por las especificaciones y por los prototipos.- Desarrolla ciclos de vida iterativos en forma de espiral, mostrando: los ciclos internos del análisis y desarrollo de prototipos tempranos y los ciclos externos del modelo clásico.- El análisis de riesgo aparece en cada ciclo del espiral y busca identificar situaciones que pueden causar el fracaso o excederse de los plazos o presupuestos esperados. Por tanto, se usa en proyectos en los que se prevén riesgos.- Proporciona una combinación de los modelos existentes para un proyecto dado.
Ventajas:
<ul style="list-style-type: none">- Su rango de opciones permite utilizar los modelos de ciclo de vida del software tradicionales.- Su orientación a detectar y prevenir el riesgo evita muchas dificultades.- Incorpora opciones que permiten la reutilización del software existente.- Se centra en la eliminación de errores y alternativas poco atractivas.- No establece una diferenciación entre desarrollo de software y mantenimiento del sistema.- Proporciona un marco estable para desarrollos integrados hardware-software.
Desventajas:
<ul style="list-style-type: none">- Es complicado y consume muchos recursos.- Las etapas y sus entradas/salidas no están claramente definidas.- Es difícil de gestionar y es costoso.



- b) ¿Cuáles son los objetivos de las actividades clave de la gestión de proyectos software? ¿Qué factores impactan en la exactitud de la estimación de un proyecto informático? ¿Cómo calcularías los puntos de función a partir de la información que proporciona un diagrama de flujo de datos?

Actividades clave de la gestión de proyectos software: Objetivos

Estimación tiene como objetivo la predicción de duración, esfuerzo y costes requeridos para realizar todas las actividades y constituir todos los productos asociados con el proyecto.

Planificación tiene como objetivo la selección de una estrategia para la obtención de unos productos finales dados, la definición de actividades a realizar para lograr ese objetivo, la coordinación de concurrencia y solapamiento de dichas actividades y la asignación de recursos a las mismas.

Coordinación del Equipo de Trabajo tiene como propósito el desarrollo y seguimiento de la organización y relaciones de trabajo de los participantes en el desarrollo del proyecto software.

Análisis y gestión de riesgos tiene como objetivo especificar y realizar los pasos que ayudan al equipo del software a comprender y a gestionar la incertidumbre. Incluye: identificar el riesgo, evaluar su probabilidad de aparición, estimar su impacto, determinar su prioridad y establecer un plan de contingencia por si ocurre el problema.

Seguimiento y Control del Proyecto pretende realizar la monitorización y revisión de resultados (y proceso) contra estimaciones, planes, compromisos, etc. Además, lleva a cabo las acciones correctivas cuando la realidad no coincide con lo planificado (ajuste de planes basándose en resultados reales, mejora de la eficiencia del equipo de trabajo) y fiablemente la implantación de los cambios acordados.

Factores que afectan a la estimación

La falta de definición del proyecto (al principio tenemos menos datos, con lo que la estimación será más inexacta e imprecisa), su complejidad y su tamaño.

DFD → PF

- Almacenes que tengan flujos de entrada y salida: Ficheros Lógicos Internos.
- Almacenes que tengan flujos de salida solamente: Ficheros Lógicos Externos.
- Flujos que salgan de una entidad externa hacia un proceso, el cual actualiza uno o varios almacenes: Entradas.
- Flujos que lleguen a una entidad externa desde un proceso, el cual realiza algún tipo de procesamiento: Salidas.
- Consultas a un almacén a través de un proceso: Consultas.

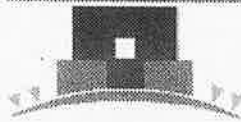
- c) Eres analista de un proyecto software, que sigue un modelo de ciclo de vida en espiral, para el cual has definido un conjunto de requisitos funcionales y no funcionales. El jefe del proyecto te solicita que enumeres todas las representaciones de requisitos disponibles, indicando en qué tipos de proyectos es adecuado su uso. Por último, debes enumerar las validaciones de la actividad de análisis de un proyecto software. Realiza todas estas tareas en la tabla que se presenta a continuación.

Representaciones – Tipo de Proyectos

- Estructurado: DFD, ERD, DTE. - En sistemas de gestión, transaccionales, que impliquen una transformación de datos.
- OO: Casos de uso. Diagramas de secuencia. Diagramas conceptuales de clases... - En sistemas con una alta interacción con el usuario, en sistemas que se puedan modelar naturalmente mediante el paradigma de orientación a objetos.
- Especificación formal: Lenguaje Z. - En sistemas en los que se tienen restricciones de rendimiento o fiabilidad.

Validaciones

- Requisitos: completos, no ambiguos, relevantes, traceables, correctos, consistentes.
- Atributos ERS: correcto, completo, consistente, no ambiguo, verificable, traceable, fácilmente modificable, comprensible por los usuarios, trazado, conciso, organizado.
- Acordado por el usuario, cliente e ingenieros de software.



Problema Tipo Examen

La empresa ANTOTIKET te encarga la construcción de un software para la venta de entradas para acontecimientos deportivos y culturales. Se quiere que la aplicación funcione en la web, de tal manera que los usuarios accedan a través de un navegador. Los usuarios podrán comprar entradas de manera interactiva en cuatro pasos. En la primera pantalla introducen las características relativas al evento. En la segunda, introducen sus datos personales. En la tercera dan sus datos bancarios. Finalmente, en la cuarta, introducen modificaciones o confirman la compra. Los usuarios podrán realizar consultas para obtener los eventos por tipo, por fecha y por localidad. Una vez obtenido el evento, se podrá realizar otra consulta sobre la disponibilidad y precio de las entradas. La disponibilidad de entradas se le proporcionará a la aplicación en un fichero externo.

Una vez realizada la compra, la aplicación permitirá imprimir las entradas en formato electrónico si los datos bancarios son válidos. En caso contrario, se debe dar un mensaje de error. La aplicación debe pues comunicarse con un sistema de gestión de medios de pago, a la que mandará la información de la tarjeta del cliente, y de la que recibe si se autoriza el pago. Esta comunicación, así como la introducción de datos bancarios debe realizarse de manera segura. La aplicación mantendrá una base de datos con la información de los clientes, que por seguridad ha de guardarse de manera encriptada. El cifrado es un proceso matemático y lógico que se considera complejo. Se esperan periodos punta diarios de uso de la aplicación (suele haber más de un evento por día), típicamente entre las 18 y las 23 horas.

Por otra parte, se desea que los usuarios puedan acceder a la aplicación desde cualquier equipo, independientemente de la calidad de su conexión, del explorador o del sistema operativo. La aplicación debe ser distribuida y la transferencia de datos on-line y en una sola dirección.

Además, la aplicación debe tener un módulo de mantenimiento, que será utilizado por los encargados de la empresa. Éstos dispondrán de la posibilidad de realizar transacciones de alta y baja de los distintos eventos de manera interactiva (esta actualización debe poder verse inmediatamente desde el módulo de los clientes), podrán obtener un listado con los clientes de cierto evento, así como un gráfico e informe con estadísticas.

El gestor de ANTOTIKET parece no tener claro el estilo de la interfaz de usuario, tanto de la aplicación cliente como del módulo de mantenimiento. No obstante te pide que la aplicación tenga las características de usabilidad de las aplicaciones típicas de Windows (uso de ratón, y presencia de menús, ventanas, scrolling, selección mediante cursor, teclas pre asignadas y ayuda on-line), así como minimizar el número de interacciones con la interfaz necesarias para adquirir las entradas. Has considerado la realización de la aplicación usando tecnología orientada a objetos (Java), y el gestor de ANTOTIKET te dice que quiere una versión básica de la aplicación lo antes posible para evaluarla.

Se pide:

- Selecciona razonadamente el modelo de ciclo de vida del software y el modelo de desarrollo de productos software más adecuado.
- Calcula los Puntos de Función ajustados, suponiendo media la complejidad de los parámetros, excepto las salidas, que se consideran de complejidad alta. Calcula el coste y el tamaño de la aplicación si los ratios de tu empresa señalan un coste de 700€ por punto de función y de 120 LDC por Punto de Función.
- Utilizando COCOMO básico, haz una estimación del esfuerzo y tiempo total del proyecto y del esfuerzo y duración de codificación, considerando un 15% de esfuerzo y un 10% de duración de esta fase. ¿Cuál sería el índice de productividad del proyecto considerado de tipo semiacoplado?