
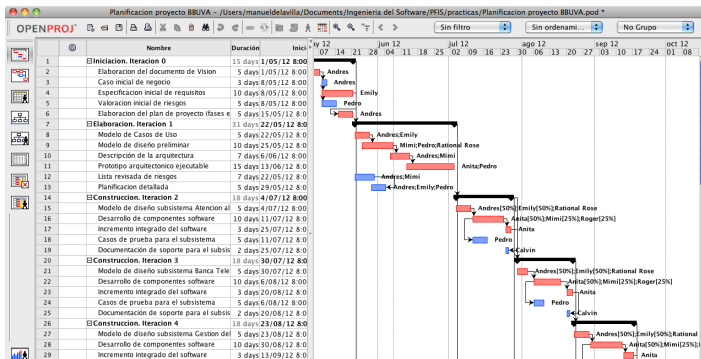
	<b>Principios y Fundamentos de la Ingeniería del Software</b>	
	<b>Examen convocatoria de Junio 2012</b>	<b>2º Grado en Ing. Informática</b> <b>21/6/2012</b>

## Problema (60% nota). GesPro, Gestión de Proyectos

**Duración: 2 horas**



La Gestión de Proyectos de Software es una rama especializada de la Ingeniería del Software. La gestión de proyectos de software es la disciplina de organizar y administrar recursos y tareas en el tiempo de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro los parámetros de recursos, tiempo, y coste definidos.

Una empresa de desarrollo de software, compuesta por un gerente, varios jefes de proyecto y numerosos desarrolladores, quiere disponer de un sistema que se encargue de la planificación de los proyectos que desarrolla para sus clientes y del seguimiento de empleados y sus tareas.

Un proyecto de desarrollo software se estructura en **Fases** y cada fase en **Actividades**. Cada Actividad se descompone en **Tareas individuales** que son asignadas a un desarrollador en particular. *P.ej., el proyecto Artemisa tiene varias fases, una fase de Pruebas, donde hay una actividad denominada Pruebas de integración que incluye la siguiente tarea individual: el desarrollador Juan Rodríguez ha de probar el formulario de Entrada de Pedidos.*

El Gerente y los Jefes de Proyecto pueden **gestionar** (crear y modificar) **proyectos, fases y actividades**. La gestión de tareas individuales (que incluye la asignación de tareas a desarrolladores) es una actividad propia y exclusiva del Jefe de Proyecto.

Cada desarrollador rellena diariamente, previa identificación, una ficha (llamada Parte de Trabajo) con información genérica de cada tarea que ha realizado y el tiempo que ha dedicado (día, hora de inicio, hora de fin, trabajo realizado, porcentaje de tarea acabado). El sistema comprueba incoherencias y, en su caso, avisa del error. Las tareas habitualmente tienen una duración de varios días.

Posteriormente, el Jefe de Proyecto habrá de revisar cada parte y normalmente lo aprobará. Excepcionalmente se solicitará una modificación del parte al empleado.

### Veamos en detalle el caso de uso 'rellenar parte de trabajo':

El desarrollador, con la condición previa de haberse identificado, solicita al sistema introducir un nuevo parte. El sistema pide el proyecto, que una vez introducido revisará para ver que existe, que no está cerrado y que el empleado pertenece al proyecto. En caso de error, muestra un mensaje de error y se vuelve al menú principal.

A continuación muestra un listado de tareas pendientes (esta consulta habrá de mostrarse en menos de 2 segundos, para no ralentizar el proceso), el desarrollador pulsa sobre la que desee introducir un parte. Cuando termina de rellenar, pulsa el botón de salvar y el sistema pide confirmación y almacena el parte.

Modele el sistema e incluya todas las funcionalidades y entidades que hayan sido nombradas en el texto, atendiendo a las siguientes peticiones:



Se le pide que analice, diseñe e implemente los siguientes entregables:

#### 1. Diagrama de Casos de Uso

(2 puntos)

2. **Especificación textual del Caso de Uso** “Rellenar parte de trabajo” usando la plantilla adjunta  
(1 punto)
3. **Diagrama de Clases** (con atributos y métodos derivados de los CU en clases, así como todos los adornos en las relaciones)  
(2 puntos)
4. **Diagrama de Secuencia** para el CU “Rellenar parte de trabajo”  
(1 punto)


<b>UC-</b>		
<b>Versión</b>		
<b>Autores</b>		
<b>Fuentes</b>		
<b>Objetivos asociados</b>		
<b>Descripción</b>		
<b>Precondición</b>		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
<b>Frecuencia esperada</b>		
<b>Importancia</b>		
<b>Urgencia</b>		
<b>Comentarios</b>		

	<b>Principios y Fundamentos de la Ingeniería del Software</b>	
	Examen convocatoria de Junio 2012	2º Grado en Ing. Informática 21/6/2012

**Teoría (40% nota)**

**Duración: 45 minutos**

**Indicar si son Verdaderas (V) o Falsas (F) las siguientes afirmaciones** (cada pregunta incorrecta la cuarta parte del valor de una correcta)

 <p>El diagrama muestra un círculo central etiquetado como 'Calidad del Software (McCall et al.)'. Desde este círculo, tres flechas apuntan hacia tres rectángulos de color naranja que representan diferentes aspectos de la calidad:</p> <ul style="list-style-type: none"> <li><b>UTILIZACIÓN DEL PRODUCTO.-</b> <ul style="list-style-type: none"> <li>Corrección – Confiabilidad – Utilizable – Integridad – Eficiencia</li> </ul> </li> <li><b>REVISIÓN DEL PRODUCTO.-</b> <ul style="list-style-type: none"> <li>Facilidad de mantenimiento – Flexibilidad – Facilidad de prueba</li> </ul> </li> <li><b>TRANSICIÓN DEL PRODUCTO.-</b> <ul style="list-style-type: none"> <li>Portabilidad – Reutilizable – Interoperatividad</li> </ul> </li> </ul>	<p>McCall, Richards y Walters en su estudio “<i>Factors in Software Quality</i>” (1977), establecieron los factores que afectaban a la calidad del software del siguiente modo:</p>
<p><b>Corrección</b> se define como la probabilidad que tiene un programa de operar sin fallas (falta de conformidad con los requerimientos del software) en un ambiente específico por un tiempo específico. (0,1 pts.)</p>	
<p><b>Integridad</b> es el grado en el que es posible controlar el acceso de personas no autorizadas al software o a los datos. (0,1 pts.)</p>	
<p>Para obtener un software satisfactorio, fácil de modificar y usar, se han de tener en cuenta todos los factores de calidad, independientemente del tipo de software que estemos elaborando. (0,1 pts.)</p>	
<p>El <b>Modelo de Proceso Incremental</b> es aplicable cuando los requisitos iniciales están suficientemente bien definidos; cuando es imprescindible entregar rápidamente un software con funcionalidad limitada; cuando los desarrollos son de gran tamaño y con poco personal inicialmente. (0,1 pts.)</p>	
<p><b>CMM</b> dirige su enfoque a la mejora de procesos en una organización, estudia los procesos de desarrollo y produce una evaluación de la madurez (indicador para medir la capacidad para construir un software de calidad). En el nivel 2 (de los 5), Definido, se considera que el proceso es conocido y bien entendido. (0,1 pts.)</p>	
<p>El enfoque de <b>control del proceso</b> “tradicional” ha demostrado ser efectivo y necesario en proyectos de gran tamaño, pero no resulta adecuado para muchos proyectos actuales. (0,1 pts.)</p>	
<p>El conjunto formado por los Requisitos del Producto, los Requisitos de Organizaciones y los Requisitos Externos, constituyen los denominados <b>Requisitos de Ámbito</b>. (0,1 pts.)</p>	
<p>La <b>Ingeniería de Requisitos</b> es un conjunto de tareas y técnicas de ingeniería que ayudan entre otras cosas a entender cuál será el efecto del software en el negocio, con qué restricciones de funcionamiento habrá que contar, cómo interactuarán los usuarios finales con el software, etc..., generando un <i>Documento de Requisitos del Sistema</i> como objetivo general. (0,1 pts.)</p>	
<p>Un <b>stakeholder</b> es el componente de un equipo de ingenieros software encargado de la elicitación de requisitos. (0,1 pts.)</p>	
<p>Los <b>Casos de Uso</b> son una herramienta para el análisis útil solamente para proyectos orientados a objeto. (0,1 pts.)</p>	
<p>Un <b>actor</b> en un Diagrama de Casos de Uso es una persona, sistema externo, organización o proceso ejecutable, que se comunica con el sistema y que es externo a él. (0,1 pts.)</p>	
<p>Un <b>modelo del dominio</b> es una representación visual de las clases conceptuales del mundo real,</p>	

no de componentes software, pues no se define ninguna operación (método). Se crea durante el Análisis Orientado a Objetos. (0,1 ptos.)	
<b>UML</b> es un método que marca las pautas para la realización de Análisis y Diseño Orientado a objetos (0,1 ptos.)	
En una <b>agregación</b> (◇), los componentes no pueden ser compartidos por varios compuestos y la destrucción del compuesto no conduce a la destrucción de los componentes. (0,1 ptos.)	
Todos los <b>modelos</b> simplifican la realidad, lo fundamental es no enmascarar ningún detalle importante, y para ello es fundamental centrarse en un solo modelo (punto de vista), pues con uno es suficiente. (0,1 ptos.)	
El <b>Diagrama de Clases</b> es una representación estática del sistema, mientras que el Diagrama de Objetos muestra las instancias creadas y sus vínculos cuando el sistema está activo. (0,1 ptos.)	
La <b>gestión del cambio</b> tiene por objeto garantizar que la evolución del sistema es un proceso global y que se regulariza la priorización de los cambios. (0,1 ptos.)	
El <b>proceso de gestión del cambio</b> tiene que ver con el análisis de los costos y beneficios de los cambios propuestos. (0,1 ptos.)	
Las <b>líneas de base</b> o baseline son versiones que cuentan con un status de aprobación formal por parte del cliente (0,1 ptos.)	

Las **metodologías Ágiles** surgen como respuesta a los estándares basados en proceso y ponen el énfasis en (señale la incorrecta):

- a) la satisfacción del cliente
- b) la entrega rápida de software incremental,
- c) los métodos formales y orientados al cambio,
- d) los equipos pequeños

(0,15 ptos.)

**Empareje** los elementos de las columnas:

Método formal		<i>Test-Driven Development</i> , programación por pares, refactorización, integración continua...	
eXtreme Programming (XP)		Sprint e incrementos. Reunión diaria	
SCRUM		Serial a lo grande, iterativo a lo pequeño.	
Desarrollo Adaptativo de Software		Técnicas definidas algebraicamente. Sistemas confiables	
Proceso Ágil Unificado (AUP)		Especulación, Colaboración y Aprendizaje	

(0,2 ptos.)

El **Plan de Proyecto** debe definir un conjunto de tareas, coordinadas en el tiempo, así como los recursos necesarios para cumplir objetivos marcados a cada tarea. ¿Cuál de estos elementos **NO** forma parte del Plan de Proyecto?:

- a) Personal y sus sueldos netos
- b) Presupuesto de esfuerzo y dinero
- c) Estándares a aplicar
- d) Lista de entregables

(0,15 ptos.)

### Análisis Coste-Beneficio

Empresa que actualmente lleva la facturación mediante sistema cliente/servidor decide implantar un sistema informático basado en "Computación en la nube" para un proceso más rápido y fiable.

Actualmente dispone de un ordenador servidor (renting renovable anualmente, a razón de 1100€/año durante próximos 3 años) con un contrato de mantenimiento anual de 350€/año, una persona que dedica 60 horas/mes a su gestión (sueldo global 1600€/mes).

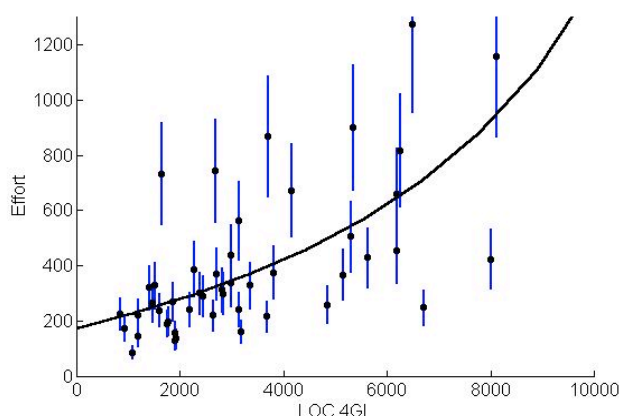
El nuevo sistema requeriría un upgrade de la línea de datos (80€/mes) y un coste del proveedor del servicio

de 170€ / año. La misma persona dedicada a gestión habrá de dedicar un 50% más de tiempo, con una formación del nuevo sistema de una semana.

¿Es capaz de cuantificar el beneficio al primer año del cambio?

a)	Año 1	b)	Año 1	c)	Año 1
<b>Ingresos/Ahorros</b>		<b>Ingresos/Ahorros</b>		<b>Ingresos/Ahorros</b>	
renting	1100	renting	3300	renting	3300
mantenimiento	350	mantenimiento	350	mantenimiento	350
gestion (60 h.)	600	gestion (60 h.)	600	gestion (60 h.)	600
<b>Total ingresos</b>	2050	<b>Total ingresos</b>	4250	<b>Total ingresos</b>	4250
<b>Costes/Gastos</b>		<b>Costes/Gastos</b>		<b>Costes/Gastos</b>	
upgrade	960	upgrade	80	upgrade	960
coste cloud	170	coste cloud	170	coste cloud	170
Gestión (+50%)	900	Gestión (+50%)	900	Gestión (+50%)	900
Formación (1600/4)	400	Formación (1600/4)	400	Formación (1600/4)	400
<b>Total costes</b>	2430	<b>Total costes</b>	1550	<b>Total costes</b>	2430
Beneficio	-380	Beneficio	2700	Beneficio	1820

Indique la opción correcta (a, b, c) \_\_\_\_\_



Con respecto a las **técnicas empíricas** de estimación, podemos afirmar que (**señale la falsa, todas son correctas menos una**):

- a) Que se basan en datos históricos de proyectos
- b) Que aplican técnicas de regresión lineal
- c) Que siguen la formula  $\text{Esfuerzo} = A * \text{Tamaño} * B$ , midiéndose el tamaño en LDC o PF.
- d) Que una vez calibrados, son efectivos y rápidos

**Realización de Casos de Uso** (en el Proceso Unificado). La realización de casos de uso consiste en la (0,15 pts.) elaboración secuencial de varios diagramas UML y artefactos textuales, que juntos validan que nosotros tenemos las clases, responsabilidades e interacciones de los objetos necesarios para dictaminar el comportamiento que debe tener el proceso de nuestro caso de uso.

**Ordene los pasos a seguir en la realización (numere de 1 a 5):**

- \_\_\_ Obtención de métodos, pre- y post-condiciones
- \_\_\_ Diagrama de Secuencia
- \_\_\_ Diagrama de Casos de Uso
- \_\_\_ Modificación del Diagrama de Clases
- \_\_\_ Especificación textual del CU

(0,15 pts.)

Relación entre **diagramas de interacción y clases**. Señale qué 2 elementos **NO** obtenemos a partir de la elaboración de diagramas de interacción.

- a) Obtención de diagramas de clases

b)      Obtenemos métodos (mensajes) para las clases c)      Obtenemos algoritmo a alto nivel d)      Definimos los puntos de creación de nuevas instancias e)      Definimos los puntos de eliminación de instancias f)      Modificación de Diagrama de CU
Indique de qué tipo es cada patrón, si GRASP ( <b>GR</b> ) o Gof ( <b>GOF</b> ): Singleton _____ Creador _____ Observer _____ Controlador _____ Memento _____ Experto en Información _____

(0,15 pts.)

¿Qué diagramas se corresponden con cada tipo de modelado? Pon una **X** en la casilla adecuada

	Modelado Estructural (Estático)	Modelado de Interacción (Funcional)	Modelado del Comportamiento (Dinámico)	Modelado de la Implementación (Arquitectónico)
Diagrama de Secuencia				
Diagrama de Despliegue				
Diagrama de Clases				
Diagrama de Colaboración				
Diagrama de Actividad				
Diagrama de Componentes				
Diagrama de Objetos				
Diagrama de Estados				

(0,5 pts.)