

## GUÍA DOCENTE

### 1. DESCRIPCIÓN DE LA ASIGNATURA

<b>Grado:</b>	<b>Ingeniería Informática en Sistemas de Información</b>
<b>Doble Grado:</b>	
<b>Asignatura:</b>	<b>Ingeniería de Proyectos</b>
<b>Módulo:</b>	<b>M6: Tecnología Específica de Sistemas de Información</b>
<b>Departamento:</b>	<b>Deporte e Informática</b>
<b>Año académico:</b>	<b>2017-2018</b>
<b>Semestre:</b>	<b>Segundo semestre</b>
<b>Créditos totales:</b>	<b>6</b>
<b>Curso:</b>	<b>3º</b>
<b>Carácter:</b>	<b>Obligatoria</b>
<b>Lengua de impartición:</b>	<b>Español</b>

<b>Modelo de docencia:</b>	<b>C1</b>	
<b>a. Enseñanzas Básicas (EB):</b>		<b>50%</b>
<b>b. Enseñanzas de Prácticas y Desarrollo (EPD):</b>		<b>50%</b>
<b>c. Actividades Dirigidas (AD):</b>		

## GUÍA DOCENTE

### 2. RESPONSABLE DE LA ASIGNATURA

<b>Responsable de la asignatura</b>	
<b>Nombre:</b>	Francisco A. Gómez Vela
<b>Centro:</b>	Escuela Politécnica Superior
<b>Departamento:</b>	Deporte e Informática
<b>Área:</b>	Lenguajes y Sistemas Informáticos
<b>Categoría:</b>	Profesor Asociado
<b>Horario de tutorías:</b>	Lunes de 15:00-16:30 y 18:00-19:30 Jueves 16-17 y 18:30-20:30
<b>Número de despacho:</b>	Edificio 23 despacho 23.1.48
<b>E-mail:</b>	fgomez@upo.es
<b>Teléfono:</b>	954977865

## GUÍA DOCENTE

### 3. UBICACIÓN EN EL PLAN FORMATIVO

#### 3.1. Descripción de los objetivos

El objetivo principal de la asignatura es que los alumnos aprendan a identificar, desarrollar y trabajar con la documentación propia de un proyecto informático de envergadura. Por ello, se hace especial hincapié al proceso de análisis, diseño y gestión del mismo desde un punto de vista temporal, de costes y coordinación de equipos.

En este sentido, en la asignatura se mostrarán las ventajas e inconvenientes de las diferentes metodologías de desarrollo software para llevar a cabo una elección acorde con el proyecto a desarrollar.

Así mismo, la asignatura pretende introducir a los alumnos en la dirección y planificación de proyectos informáticos. Para ello, los alumnos deberán realizar defensas del proyecto.

#### 3.2. Aportaciones al plan formativo

Enmarcada en el módulo de Tecnología Específica de Sistemas de Información, la asignatura IP introduce a al estudiante a las tareas de gestión de proyectos informáticos, continuando su formación en materia de supervisión y desarrollo de proyectos informáticos iniciada en la asignatura Ingeniería de Software 2 (ISG2).

El conocimiento de la dirección, gestión y auditoría de proyectos es crucial para cualquier graduado en Ingeniería Informática. Así mismo, resulta de gran importancia la aportación de la asignatura IP para proporcionar al alumno una metodología útil, correcta, completa y ordenada, cuyo resultado sea la consecución de proyectos informáticos con éxito.

Esta asignatura, prepara al alumno para asumir cargos de responsabilidad en el ámbito empresarial orientando su perfil a los mismos (Jefe de proyectos o Analista de sistemas de la información). Es importante que todo futuro Ingeniero Informático esté preparado para asumir responsabilidad dentro de la jerarquía empresarial, y esta asignatura le proporciona las herramientas necesarias para ello.

#### 3.3. Recomendaciones o conocimientos previos requeridos

Es muy recomendable que el alumno haya superado la asignatura Ingeniería del

## GUÍA DOCENTE

Software II, impartida en el segundo curso del grado.

### 4. COMPETENCIAS

#### 4.1 Competencias de la Titulación que se desarrollan en la asignatura

4.1.1. Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas. (Competencia G01)

4.1.2. Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática. (Competencia G02)

4.1.3. Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática. (Competencia G03)

4.1.4. Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad. (Competencia G05)

4.1.5. Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería. (Competencia EB5)

4.1.6. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados. (Competencia EC08)

4.1.7. Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software. (Competencia EC16)

4.1.8. Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas. (Competencia EC17)

4.1.9. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas

## GUÍA DOCENTE

informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente. (Competencia EC01)

4.1.10. Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social (Competencia EC02).

### **4.2. Competencias del Módulo que se desarrollan en la asignatura**

4.2.1. Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social. (Competencia EC02)

4.2.2. Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software. (Competencia EC03)

### **4.3. Competencias particulares de la asignatura**

4.3.1. Enseñar al alumno el papel fundamental que desempeña el análisis dentro del proceso software. Comprender la importancia de identificar requisitos y realizar una gestión que disminuya los riesgos de su desarrollo.

4.3.2. Enseñar al alumno las técnicas para el análisis de problemas mediante técnicas descriptivas de modelado. Usar y elegir correctamente un tipo de proceso de desarrollo software dependiendo de los requisitos a cumplir.

4.3.3. Fomentar la disciplina en la programación, siguiendo planteamientos que desarrollen la capacidad analítica del alumno para enfrentarse a problemas reales. Ser consciente de las ventajas de identificar requisitos software en etapas tempranas del ciclo de vida software y su utilidad en el proceso de desarrollo software.

4.3.4. Capacidad para determinar los requisitos de los sistemas de información y comunicación de una organización atendiendo a aspectos de seguridad y cumplimiento de la normativa y la legislación vigente. (Competencia ET2)

4.3.5. Capacidad para comprender y aplicar los principios y prácticas de las organizaciones, de forma que puedan ejercer como enlace entre las comunidades técnica y de gestión de una organización y participar activamente en la formación de los usuarios. (Competencia ET4)

## GUÍA DOCENTE

4.3.6. Capacidad para comprender y aplicar los principios de la evaluación de riesgos y aplicarlos correctamente en la elaboración y ejecución de planes de actuación. (Competencia ET5)

### 5. CONTENIDOS DE LA ASIGNATURA (TEMARIO)

#### **Tema 1: Contextualización de proyectos**

1. Ciclo de vida de proyecto TIC
2. Valoración del marco del proyecto
3. Desarrollo del concepto del proyecto
4. Valorar viabilidad financiera del proyecto
5. Identificación de los interesados de alto nivel
6. Solución conceptual
7. Lanzamiento del proyecto

#### **Tema 2: Metodologías de desarrollo**

1. Metodologías de desarrollo software.
2. Metodologías Pesadas.
  - a. Ciclo de vida en cascada
  - b. Ciclo de vida incremental
  - c. Ciclo de vida iterativo
  - d. RUP
3. Metodologías Ágiles
  - a. Extreme programming
  - b. Scrum

#### **Tema 3: Planificación y plan de proyectos**

1. Propósito de la planificación
2. Work Breakdown Structure (WBS)
3. Estimaciones de tiempo
4. Diagrama de PERT
5. Diagrama de GANTT
6. Plan de proyecto

## GUÍA DOCENTE

### **Tema 4: Diagramas de análisis UML y Documentación de Análisis**

1. Diagramas de paquetes
2. Diagramas de componentes I
3. Diagramas de despliegue (ASI)
4. Documento de análisis (ASI)

### **Tema 5: Diagramas de Diseño y Documento de diseño DSI**

1. Patrones de diseño
2. Diagrama de robustez y secuencia.
3. Diagrama de despliegue (DSI)
4. Plan de prueba de (DSI)
5. Documento de diseño (DSI)

## **6. METODOLOGÍA Y RECURSOS**

Por un lado las Enseñanzas Básicas (EB), comúnmente llamadas clases teóricas, están dirigidas a todo el grupo y se imparten en forma de lección magistral, con los objetivos esenciales de transmitir conocimientos, ofrecer un enfoque crítico de la asignatura que lleve a los alumnos a reflexionar y descubrir las relaciones entre los diversos conceptos, y el formar una mentalidad crítica en la forma de afrontar los problemas y la existencia de un método. Aunque la lección magistral se trata principalmente de la exposición continua del profesor, los alumnos pueden tomar notas y tienen la oportunidad de preguntar. La resolución de problemas se llevará a cabo tanto en clases de teoría como en las de prácticas de laboratorios, pudiéndose trabajar tanto en pequeños grupos de trabajo como individualmente.

Por otro lado, en las prácticas de laboratorio o Enseñanzas Prácticas de Desarrollo (EPD) se exponen las ideas fundamentales desarrolladas de una manera básica, se introduce al alumno en la resolución de problemas mediante unos experimentos guiados, y se enumeran una serie de ejercicios y problemas que el alumno debe resolver. Aunque se potenciará principalmente el trabajo individual y autónomo, esta actividad permite también el trabajo en pequeños grupos de dos o tres alumnos.

Todas estas actividades formativas estarán además complementadas con soporte a través de herramientas Web como la herramienta de Aula Virtual WebCT.

## **7. EVALUACIÓN**

## GUÍA DOCENTE

### CONVOCATORIA DE CURSO:

La evaluación de la asignatura se acogerá al modelo de evaluación continua durante 3 entregas de documentación mas una presentación de defensa final. Esta evaluación sólo será aplicable para la convocatoria de curso. Aquellos alumnos que no superen la asignatura en dicha convocatoria, se podrán evaluar en la convocatoria de julio como se describe más adelante.

La evaluación se basará principalmente en los conocimientos adquiridos tanto en clase de teoría como en laboratorio de informática. La participación en las sesiones teóricas también será evaluada en forma de un cuaderno de profesor. La nota final oscilará entre 0 y 10 puntos, los cuales se acumularán en función de la calificación de las pruebas:

- 30% cada entrega documental
- 10% defensa del proyecto realizado

Así mismo, la obtención de una calificación menor de 4 en una entrega documental o dos entregas con una nota inferior a un 5, supondrá el suspenso de la asignatura en esta convocatoria. Para poder optar a realizar la defensa del proyecto, la nota media de las 3 entregas escritas deberá superar el 5.

### CONVOCATORIA DE RECUPERACIÓN:

Para la convocatoria de recuperación, las notas obtenidas superiores a un 5 en cada una de las pruebas evaluables de la evaluación continua se guardarán para esta convocatoria, teniendo que realizar el alumno la entrega de las partes que necesite recuperar.

En esta convocatoria la realización de dichas pruebas entregables podrá ser de forma individual.

## 8. BIBLIOGRAFÍA GENERAL

- Dirección y Gestión de Proyectos Informáticos: Un enfoque práctico  
Alberto Domingo Ajenjo. RA-MA .2000
- The Essence of Computing Projects. A student's Guide . Christian W.Daes  
Prentice-Hall .2001
- Rapid Development: Taming Wild Software Schedules. Steve McConnell.  
Microsoft Press,U.S. 1996





## GUÍA DOCENTE

- El proceso Unificado de Desarrollo de Software (traducida). Ivar Jacobson, Grady Boock, James Rumbaugh. Editorial Addison Wesley
- Ingeniería del Software. Un enfoque práctico (Edición 6 Traducido). Roger S. Pressman. Editorial Mac Graw Hill. ISBN 970-10-573-3