

## 30222 - Ingeniería del Software

### Información del Plan Docente

<b>Año académico</b>	2018/19
<b>Asignatura</b>	30222 - Ingeniería del Software
<b>Centro académico</b>	110 - Escuela de Ingeniería y Arquitectura 326 - Escuela Universitaria Politécnica de Teruel
<b>Titulación</b>	439 - Graduado en Ingeniería Informática 443 - Graduado en Ingeniería Informática
<b>Créditos</b>	6.0
<b>Curso</b>	3
<b>Periodo de impartición</b>	Semestral
<b>Clase de asignatura</b>	Obligatoria
<b>Módulo</b>	---

### 1. Información Básica

#### 1.1. Objetivos de la asignatura

Tras haber cursado con aprovechamiento un conjunto de asignaturas de programación (Programación I, Programación II, Tecnología de Programación) donde se ha aprendido a diseñar la implementación de programas (y sus estructuras de datos) utilizando un paradigma orientado a objeto, en esta asignatura el alumno va a aprender a abordar la construcción de software de mayor entidad y desde una perspectiva global, no solamente centrada en aspectos de implementación. Tomando como punto de partida la descripción de un software a construir, el alumno aprenderá a determinar sus requisitos, analizarlos, realizar un diseño, y plantear un conjunto de pruebas sobre el software construido. Para ello se utilizará una metodología orientada a objeto, aunque los conceptos presentados son aplicables a cualquier otra metodología de desarrollo de software. Asimismo, se contextualizarán los conocimientos adquiridos en todas las asignaturas de Programación, Estructuras de Datos, Bases de Datos e Interacción Persona Ordenador.

La asignatura tiene un marcado carácter aplicado. El alumno aprenderá los conceptos necesarios sobre análisis, diseño y pruebas aplicándolos sobre un conjunto de ejemplos de sistemas presentados tanto en las clases de problemas, como en las distintas sesiones de las prácticas de laboratorio.

#### 1.2. Contexto y sentido de la asignatura en la titulación

La asignatura de Ingeniería del Software constituye el primer peldaño de un estudiante de Ingeniería Informática para formarse en los principios de la Ingeniería del Software, es decir, en la aplicación coordinada de técnicas, metodologías y herramientas para producir un software de alta calidad, con un determinado presupuesto y antes de una determinada fecha. Es una asignatura obligatoria englobada en la materia de formación común en Ingeniería del Software y Sistemas de Información.

#### 1.3. Recomendaciones para cursar la asignatura

El alumno que curse esta asignatura ha de contar con una formación en programación orientada a objetos del nivel de la asignatura de Tecnología de Programación.

## 2. Competencias y resultados de aprendizaje

### 2.1. Competencias

Al superar la asignatura, el estudiante será más competente para:

- Concebir, diseñar y desarrollar proyectos de Ingeniería.
- Planificar, presupuestar, organizar, dirigir y controlar tareas, personas y recursos.
- Usar las técnicas, habilidades y herramientas de la Ingeniería necesarias para la práctica de la misma.
- Combinar los conocimientos generalistas y los especializados de Ingeniería para generar propuestas innovadoras y competitivas en la actividad profesional.
- Aplicar los principios, metodologías y ciclos de vida de la ingeniería de software.

### 2.2. Resultados de aprendizaje

El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados:

- Comprende la necesidad de abordar el desarrollo de sistemas software desde un punto de vista basado en conceptos de Ingeniería del Software.
- Conoce y sabe aplicar los principios de análisis, diseño, verificación y validación de software.
- Conoce y tiene capacidad para utilizar técnicas y herramientas de ayuda al desarrollo de software (herramientas CASE).
- Comprende qué es una metodología y cómo abordar la construcción de sistemas software desde una perspectiva orientada a objetos.

### 2.3. Importancia de los resultados de aprendizaje

Aprender a abordar el desarrollo de sistemas software aplicando los principios de una ingeniería es esencial para un ingeniero informático que participe en un equipo humano de desarrollo de software de tamaño medio o grande. Lo que aprenda en esta asignatura, que complementa a lo ya aprendido en las asignaturas de programación, estructuras de datos y algoritmos, bases de datos e interacción persona ordenador, será un importante paso adelante en su formación como ingeniero.

## 3. Evaluación

### 3.1. Tipo de pruebas y su valor sobre la nota final y criterios de evaluación para cada prueba

El estudiante deberá demostrar que ha alcanzado los resultados de aprendizaje previstos mediante las siguientes actividades de evaluación:

**En la Escuela de Ingeniería y Arquitectura de Zaragoza**

**Convocatoria de Febrero.** La evaluación de la asignatura se realiza en base a dos pruebas:

- **P1. Prueba escrita** en la que responder cuestiones y resolver ejercicios y problemas. Se requiere una nota mínima de 5.0 puntos en esta prueba para aprobar la asignatura. Si se obtiene esta nota mínima, entonces la prueba pondera un 80% en la nota de la asignatura. La fecha de realización de esta prueba será la que la dirección de la EINA determine para la realización de la prueba global de la asignatura.
- **P2. Trabajo de asignatura y resultados de prácticas de laboratorio en grupos.** Esta prueba pondera un 20% en la nota de la asignatura. Cada grupo, formado por 2 alumnos (salvo excepciones justificadas), deberá asistir a cada sesión de prácticas y realizar las entregas que se indiquen. Asimismo, cada grupo entregará un trabajo donde se realice el análisis, diseño y pruebas de una segunda versión de la aplicación informática abordada en las

## 30222 - Ingeniería del Software

sesiones de prácticas, pero ahora incluyendo nueva funcionalidad. Los trabajos se entregarán de forma electrónica con anterioridad a la fecha que la dirección de la EINA establezca para la realización de la prueba global de la asignatura y serán supervisados con un mínimo de 2 sesiones de tutoría. Si no se han realizado las entregas exigidas en cada sesión de prácticas o no se han concertado las tutorías obligatorias para el seguimiento del trabajo, además de enviar todos los entregables se deberá realizar un examen de la prueba P2.

Es obligatorio realizar y entregar ambas pruebas para poder aprobar la asignatura. Si en una de las pruebas, o en las dos, la nota obtenida fuera inferior a 5.0, la calificación final de la asignatura será la media ponderada de las dos calificaciones (80% P1 y 20% P2), con un máximo de 4.0.

Se considerarán como no presentados en esta convocatoria a los alumnos que no hayan realizado ninguna de las dos pruebas P1 y P2, así como a aquellos que se hayan presentado a una de las pruebas, aprobándola, pero no se hayan presentado a la otra.

**Convocatoria de Septiembre.** La evaluación de la asignatura se realiza en base a dos pruebas análogas a las de la convocatoria de Febrero, con las mismas ponderaciones y exigencia de notas mínimas. Las calificaciones del alumno obtenidas en la convocatoria de Febrero en cualquiera de las pruebas (P1 y P2) se mantienen en Septiembre, salvo que el alumno opte por presentarse a la prueba correspondiente en esta nueva convocatoria, en cuyo caso prevalecerá la nueva calificación.

### En la Escuela Universitaria Politécnica de Teruel

**Convocatoria de Febrero.** La evaluación de la asignatura se realiza en base a dos pruebas:

- **P1. Examen de problemas** en el que deberá resolverse uno o varios casos prácticos. Se requiere una nota mínima de 4.0 puntos en esta prueba para aprobar la asignatura. En esta convocatoria, esta prueba representará el 30% de la nota final.
- **P2. Trabajos asociados a las prácticas de laboratorio.** Se requiere una nota mínima de 4.0 puntos en esta prueba para poder aprobar la asignatura. En esta convocatoria esta prueba representará el 70% de la nota final de la asignatura.

**Convocatoria de Septiembre.** La evaluación de la asignatura se realiza en base a dos pruebas análogas a las de la convocatoria de Febrero, con las mismas notas mínimas pero con distintas ponderaciones. En esta convocatoria, la nota final de la asignatura será  $NF = 0.7 \cdot P1 + 0.3 \cdot P2$ .

## 4. Metodología, actividades de aprendizaje, programa y recursos

### 4.1. Presentación metodológica general

El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:

1. El estudio y trabajo continuado desde el primer día de clase.
2. El aprendizaje de conceptos y metodologías para el análisis, diseño, verificación y validación de software a través de las clases magistrales, en las que se favorecerá la participación de los alumnos.
3. La aplicación de tales conocimientos al análisis, diseño, verificación y validación de ejemplos de sistemas software en las clases de problemas. En estas clases los alumnos desempeñarán un papel activo en la discusión y resolución de los problemas.

## 30222 - Ingeniería del Software

4. Las clases de prácticas en laboratorio en las que el alumno aprenderá la tecnología necesaria para realizar el análisis, diseño, verificación y validación de una aplicación informática.
5. Sólo en EINA: El trabajo en equipo abordando una segunda versión de la aplicación informática que se ha trabajado a lo largo de las prácticas de laboratorio, pero esta vez incluyendo nuevas funcionalidades. El resultado de este trabajo se plasma en la entrega de los documentos correspondientes al análisis, diseño y pruebas, así como el código fuente modificado y los scripts de las pruebas.

### 4.2. Actividades de aprendizaje

El programa que se ofrece al estudiante para ayudarle a lograr los resultados previstos comprende las siguientes actividades:

- En las clases impartidas en el aula se desarrollará el temario de la asignatura.
- En las clases de problemas se resolverán problemas de aplicación de los conceptos y técnicas presentadas en el programa de la asignatura.
- Las sesiones de prácticas de desarrollo se desarrollan en un laboratorio informático. A lo largo de sus sesiones cada alumno deberá realizar, individualmente o en equipo, trabajos directamente relacionados con los temas estudiados en la asignatura.
- **Sólo en EINA:** Adicionalmente, se solicitará un trabajo por grupos bajo la tutela de un profesor donde se abordará una segunda versión de la aplicación abordada en las sesiones prácticas, donde se incluirá nueva funcionalidad.

### 4.3. Programa

El programa de la asignatura consta de los siguientes temas:

- Introducción a la ingeniería del software.
- Determinación de requisitos.
- Análisis: modelado de objetos, modelado dinámico.
- Diseño: diseño de sistemas, diseño de objetos.
- Prueba del producto software.

### 4.4. Planificación de las actividades de aprendizaje y calendario de fechas clave

Calendario de sesiones presenciales y presentación de trabajos

La organización docente de la asignatura prevista en la **Escuela de Ingeniería y Arquitectura de Zaragoza** es la siguiente:

- Clases teóricas (2 horas semanales)
- Clases de problemas (1 hora semanal)
- Clases prácticas de laboratorio (seis sesiones de 2 horas cada dos semanas). Son sesiones de trabajo de análisis, diseño y pruebas de software en laboratorio, tuteladas por un profesor.
- Tutorías de los trabajos en grupo bajo la supervisión de un profesor. Se realizarán al menos dos sesiones de tutoría por grupo (se deberán concertar con cita previa).
- Presentación de trabajos objeto de evaluación: Los trabajos (P2) se entregarán con anterioridad a la fecha límite para su entrega, que será la misma que la dirección de la EINA establezca para la realización de la prueba P1.

La organización docente de la asignatura prevista **en la Escuela Universitaria Politécnica de Teruel** es la siguiente:

- Clases teóricas (2 horas semanales)
- Clases prácticas de laboratorio (2 horas semanales)
- Tutorías de trabajos dirigidos (1 hora semanal). Los alumnos/as deben concertar cita previa.
- Presentación de trabajos objeto de evaluación: La práctica general de la asignatura (P2) deberá entregarse antes del comienzo del examen (P1).

### Trabajo del estudiante

La dedicación del estudiante para alcanzar los resultados de aprendizaje en esta asignatura se estima en 150 horas distribuidas del siguiente modo:

- 60 horas, aproximadamente, de actividades presenciales (sesiones en el aula teóricas -30- y de problemas -15- y sesiones en el laboratorio -15-)
- 50 horas de trabajo en grupo
- 35 horas de trabajo y estudio individual efectivo (estudio de apuntes y textos, resolución de problemas, preparación clases y prácticas, desarrollo de programas)
- 5 horas dedicadas a distintas pruebas de evaluación

El calendario de exámenes y las fechas de entrega de trabajos se anunciará con suficiente antelación.

### 4.5. Bibliografía y recursos recomendados

[BB: Bibliografía básica / BC: Bibliografía complementaria]

Zaragoza:

- [BB] Booch, Grady. The Unified Modeling Language user guide / Grady Booch, James Rumbaugh, Ivar Jacobson . - 2nd ed.
- [BB] Booch, Grady. UML : el lenguaje unificado de modelado : guía del usuario / Grady Booch, James Rumbaugh, Ivar Jacobson ; traducción y revisión técnica, Jesús J. García Molina, José Sáez Martínez. - 2ª ed. Madrid [etc.] : Addison-Wesley, D.L. 2010
- [BB] Bruegge, Bernd . Object-Oriented Software Engineering Using UML, Patterns, and Java / Bruegge Bernd, Dutoit Allen H . - 3rd Edition Prentice Hall, 2009.
- [BB] Bruegge, Bernd. Ingeniería de software orientado a objetos / Bernd Bruegge, Allen H. Dutoit ; traducción, Sergio Luis María Ruiz Faudón ; revisión técnica, Rafael Gamboa Hiraes, Martha Rosa Cordero López, Marco Antonio Dorantes González. - 1a. ed. Naucalpan de Juárez, México [etc.] : Pearson Educacion, 2002
- [BB] Modelado y diseño orientado a objetos / James Rumbaugh ... [et al.] ; traducción, José Rafael García-Bermejo Giner ; con la colaboración de Antonio Reus Hungría ; coordinación general y revisión técnica Luis Joyanes Aguilar . - 1a. ed. en español, reimpr. Madrid [etc.] : Prentice Hall, 2001
- [BC] Sommerville, Ian. Software engineering / Ian Sommerville . 10th ed. Boston [etc.] : Pearson, cop. 2016
- [BC] Design patterns : Elements of reusable object-oriented software / Erich Gamma...[et al.] . - 40th. print. Boston [etc.] : Addison-Wesley, 2012

Teruel:

- [BB] Bruegge, Bernd. Ingeniería de software orientado a objetos / Bernd Bruegge, Allen H. Dutoit ; traducción, Sergio Luis María Ruiz Faudón ; revisión técnica, Rafael Gamboa Hiraes, Martha Rosa Cordero López, Marco Antonio Dorantes González. 1a. ed. Naucalpan de Juárez, México [etc.] : Pearson Educacion, 2002 [ ]
- [BB] Design patterns : Elements of reusable object-oriented software / Erich Gamma...[et al.] . - 40th. print. Boston [etc.] : Addison-Wesley, 2012
- [BB] Kruchten, Philippe. The rational unified process : an introduction / Philippe Kruchten . - 3rd ed. Boston [etc.] : Addison Wesley, cop. 2004
- [BB] Rumbaugh, James. El lenguaje unificado de modelado UML : manual de referencia / James Rumbaugh, Ivar Jacobson, Grady Booch ; traducción Héctor castán Rodríguez, Àscar Sanjuán Martínez , Mariano de la Fuente Alarcón ; coordinación general y revisión técnica Luis Joyanes Aguilar . - 2ª ed. Madrid [etc.] : Pearson Educación, D. L. 2007