



## Grado en Ingeniería de Organización Industrial 30105 - Fundamentos de informática

Guía docente para el curso 2013 - 2014

Curso: 1, Semestre: 1, Créditos: 6.0

---

### Información básica

---

#### Profesores

- Ruben Martinez Cantin -
- María Teresa Lozano Albalate maytelo@unizar.es
- Eduardo Montijano Muñoz emonti@unizar.es
- Alen Tomas Cortes Arcos -

#### Recomendaciones para cursar esta asignatura

Las habituales para acceder a los estudios de cualquier grado de ingeniería, fundamentalmente haber cursado el bachillerato científico-tecnológico.

Esta asignatura no exige ningún conocimiento previo de programación ni tratamiento automático de datos. Sin embargo, el alumnado deberá tener ciertos conocimientos de informática a nivel de usuario para un mejor aprovechamiento de las clases prácticas.

#### Actividades y fechas clave de la asignatura

Las actividades de la asignatura dependen del centro de impartición (Centro Universitario de la Defensa o Escuela Politécnica de la Almunia) y se pueden consultar en el apartado *Actividades y recursos*.

---

### Inicio

---

#### Resultados de aprendizaje que definen la asignatura

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

- 1: Analiza problemas, y diseña e implementa soluciones algorítmicas a dichos problemas.
- 2:

- Resuelve problemas de forma disciplinada, obteniendo una implementación correcta, eficaz y eficiente.
- 3: Utiliza el ordenador a nivel usuario, manejando sistemas operativos y entornos de programación.
  - 4: Conoce el equipamiento informático tanto a nivel físico como lógico.
  - 5: Identifica las necesidades de información para resolver problemas, la recupera, la interpreta y la aplica a la resolución.

## Introducción

### Breve presentación de la asignatura

La asignatura tiene por objeto que el alumnado comprenda el papel del computador en la resolución de problemas del ámbito de la Ingeniería, y sea capaz de crear, analizar, depurar y mantener pequeños/medianos programas que le permitan resolver estos problemas.

---

## Contexto y competencias

---

### Sentido, contexto, relevancia y objetivos generales de la asignatura

#### La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:

- 1 Capacitar al estudiante para que pueda plantear la solución de un problema creando sencillos programas. Por lo tanto su contenido básico y nuclear es la programación y, en particular, la especificación de los problemas, el planteamiento de un abanico de soluciones como algoritmos alternativos posibles, la elección de la mejor solución basada en la experimentación o en experiencias previas, y la traducción de estas soluciones en programas ejecutables por un computador en un lenguaje de programación de propósito general.
- 2 Que el alumnado conozca los elementos constitutivos de un computador, comprenda su funcionamiento básico, sea capaz de buscar información y de aplicar los conocimientos de programación y de resolución de problemas en las herramientas y aplicaciones software disponibles.

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

La Informática es una asignatura de formación básica. Esta particular ubicación en las primeras etapas de la titulación permite que el alumnado pueda aplicar en el resto de asignaturas de la titulación los conocimientos adquiridos en ésta, la mayoría de las cuales, en mayor o menor medida, necesitan apoyarse en herramientas informáticas para la resolución de problemas.

A pesar de que esta titulación no habilita para el ejercicio de la profesión regulada de Ingeniero Técnico Industrial, en el diseño de la titulación en IOI se han incorporado buena parte de las competencias y módulos definidos por la orden CIN 351/2009, de 9 de febrero, por la que se establecen los requisitos para la verificación de los títulos universitarios oficiales que habiliten para el ejercicio de la profesión de Ingeniero Técnico Industrial. En particular, esta asignatura pertenece al módulo de formación básica para abordar, además de las competencias genéricas del Ingeniero Técnico Industrial, conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

#### Al superar la asignatura, el estudiante será más competente para...

- 1: Resolver problemas y tomar decisiones con iniciativa, creatividad y razonamiento crítico.
- 2: Aplicar las tecnologías de la información y las comunicaciones en la Ingeniería.

- 3:** Utilizar a nivel básico ordenadores, sistemas operativos, entornos de programación, bases de datos y programas informáticos con aplicación en ingeniería.

### **Importancia de los resultados de aprendizaje que se obtienen en la asignatura:**

Esta materia supone el primer contacto con los conceptos y habilidades que constituyen la *forma de pensar del ingeniero*, y que permiten ponerlas en práctica con problemas reales desde el principio. Si atendemos a la resolución de problemas, la Informática trata del conocimiento, diseño y explotación de la computación y la tecnología de computadores, constituyendo una disciplina que:

- 1 Desarrolla la capacidad de expresar soluciones como algoritmos, y el papel de estos para aproximarse a áreas como el diseño de sistemas, la resolución de problemas, la simulación y el modelado.
- 2 Requiere una aproximación disciplinada a la resolución de problemas, de la que se espera soluciones de calidad.
- 3 Controla la complejidad de los problemas, primero a través de la abstracción y la simplificación, para diseñar a continuación soluciones mediante la integración de componentes.
- 4 Facilita la comprensión de las oportunidades que ofrece la automatización de los procesos, y como las personas interaccionan con los computadores.
- 5 Facilita el aprendizaje, a través de la experimentación, de principios básicos como la concisión y la elegancia, así como a reconocer las malas prácticas.

---

## **Evaluación**

---

### **Actividades de evaluación**

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

- 1:** Trabajos prácticos (30%). Se evaluará la soltura en el manejo del computador para resolver problemas (RA 3). También se evaluarán las soluciones implementadas para los ejercicios planteados (RA 1), atendiendo a la calidad de los procedimientos y estrategias de resolución eficaz y eficiente en el computador, así como la calidad del programa que implementa dicha estrategia (RA 2), todo ello según el temario. Se evaluará la capacidad para identificar las necesidades de información para resolver los problemas planteados a lo largo del cuatrimestre y su utilización en su resolución (RA 5). También se valorará la capacidad crítica a la hora de seleccionar alternativas y el grado de justificación de la solución alcanzada. En su caso, se solicitará defensa
- 2:** Prueba teórico-práctica (70%)(RA 4) en la que se plantearán cuestiones y/o problemas del ámbito de la ingeniería a resolver mediante programación, de tipología y nivel de complejidad similar al utilizado durante el curso. Se valorará la calidad y claridad de la estrategia de resolución, así como su eficiencia (RA 2). También se valorará la calidad del programa, escrito en el lenguaje de programación de propósito general utilizado durante el curso, que realiza dicha estrategia. Los errores semánticos graves (desconocimiento de las reglas básicas de construcción y codificación de algoritmos) podrán suponer la penalización total del ejercicio. De no superar unos mínimos en esta parte, la asignatura no podrá ser aprobada.

---

## **Actividades y recursos**

### **Perfil empresa**

---

## Presentación metodológica general

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

La asignatura es eminentemente práctica, por tanto se ha planteado con una fuerte carga práctica tanto de elaboración de problemas en clase como realizando prácticas en el aula. También hay una parte teórica.

Además el alumno deberá trabajar de manera individual sin la presencia del profesor para conseguir el nivel práctico necesario. También se plantea un problema de tamaño mayor a los presentados en el aula a realizar en grupo.

El proceso evaluativo incluirá dos tipos de actuación:

- **Un sistema de evaluación continua**, que se realizará a lo largo de todo el período de aprendizaje.
- **Una prueba global de evaluación**, que refleje la consecución de los resultados de aprendizaje, al término del período de enseñanza.

Estos procesos valorativos se realizarán a través de:

- Observación directa del alumno para conocer su actitud frente a la asignatura y el trabajo que esta exige (atención en clase, realización de trabajos encomendados, resolución de cuestiones y problemas, participación activa en el aula, etc.).
- Observación directa de las habilidades y destrezas en el trabajo de laboratorio.
- Comprobación de sus avances en el campo conceptual (preguntas en clase, comentarios en el aula, realización de exámenes, etc.).
- Realización periódica de pruebas orales y/o escritas para valorar el grado de conocimientos adquiridos, así como las cualidades de expresión que, a este nivel educativo, debe manifestar con amplia corrección.

Para conciliar la evaluación continua y la no continua, se divide la teoría en dos partes:

- bloques I y II
- bloques III y IV

Cada una de ellas se podrá aprobar en pruebas escritas parciales realizadas durante el curso. En caso de aprobarse cualquier parte se mantendrá la nota de dicha parte durante todo el curso.

#### 1-Sistema de evaluación continua.

Siguiendo el espíritu de Bolonia, en cuanto al grado de implicación y trabajo continuado del alumno a lo largo del curso, la evaluación de la asignatura contempla el sistema de evaluación continua como el más acorde para estar en consonancia con las directrices marcadas por el nuevo marco del EEES.

El sistema de evaluación continua va a contar con el siguiente grupo de actividades calificables:

- **Prácticas de laboratorio:** En las prácticas realizadas a lo largo del curso se solicitará entregar la solución a las preguntas planteadas en ellas. Se valorarán la corrección y calidad de los resultados. Dichas prácticas se efectuarán en grupos de como máximo 20 alumnos/as.

— **Proyecto:** Las habilidades de diseño y programación se evaluarán mediante un proyecto. Los alumnos de cada clase formarán grupos de tres personas que, en casos excepcionales, podrían ser de dos o cuatro personas tras aprobación del profesor. Estos grupos deberán realizar un proyecto de programación de tamaño medio cuyo enunciado se publicará a lo largo del curso. El proyecto deberá defenderse en persona ante el profesor. Se valorarán la corrección y calidad de los resultados, pero también las ideas presentadas en la defensa y la claridad de la presentación. Si los componentes del grupo muestran distintos grados de conocimiento del proyecto podrán recibir una nota diferente.

— **Pruebas evaluatorias escritas:** Hay dos a lo largo del curso. En ellas hay una parte teórica evaluada mediante preguntas de tipo test, y otra parte de programación. Deben conseguirse la mitad de los puntos de programación y la mitad de los puntos de teoría para aprobar el examen. En la parte de programación se valorará la adecuación de la respuesta a lo pedido, así como la calidad de la solución.

Es necesario superar por separado las prácticas de laboratorio, el proyecto y las pruebas escritas para que puedan contribuir al promedio de la nota final.

Como resumen a lo anteriormente expuesto se ha diseñado la siguiente tabla de ponderación del proceso de calificación de las diferentes actividades en la que se ha estructurado el proceso de evaluación continua de la asignatura.

Actividad de evaluación	Ponderación
Prácticas de laboratorio	15 %
Proyecto	15 %
Pruebas evaluatorias escritas de teoría (2)	2x24 %
Pruebas evaluatorias escritas de programación (2)	2x11 %

## 2-Prueba global de evaluación final.

El alumno deberá optar por esta modalidad cuando, por su coyuntura personal, no pueda adaptarse al ritmo de trabajo requerido en el sistema de evaluación continua, haya suspendido o quisiera subir nota habiendo sido partícipe de dicha metodología de evaluación.

La prueba global de evaluación final va a contar con el siguiente grupo de actividades calificables:

— **Examen escrito:** Se realiza en las convocatorias oficiales. Siempre se podrá seguir esta opción a pesar de que el alumno haya usado el sistema de evaluación continua. La prueba estará dividida en dos partes, una por cada bloque teórico. En cada parte habrá preguntas de test y problemas de programación. Deben conseguirse la mitad de los puntos de

programación y la mitad de los puntos de teoría para aprobar el examen. En caso de que el proyecto no haya sido entregado y revisado con el profesor antes de la última prueba parcial, pero también si no ha superado las prácticas, se deberá realizar un examen de prácticas en la misma sesión que el examen. Este examen de prácticas sustituye a todas las preguntas de programación del examen. En caso de requerir el examen de prácticas, será obligatorio aprobarlo para superar la asignatura. En las respuestas de programación se valorará la adecuación de la respuesta a lo pedido, así como la calidad de la solución.

Como resumen a lo anteriormente expuesto se ha diseñado la siguiente tabla de ponderación del proceso de calificación de las diferentes actividades en la que se ha estructurado el proceso de evaluación final de la asignatura.

Actividad de evaluación	Superados prácticas y proyecto	Sin superar prácticas o proyecto
Prácticas de laboratorio	15 %	
Proyecto	15 %	
Pruebas evaluatorias escritas de teoría (2)	2x24 %	2x24 %
Pruebas evaluatorias escritas de programación (2)	2x11 %	
Prueba evaluatoria escrita de prácticas		52 %

Se habrá superado la asignatura cuando estén aprobadas todas las partes tal y como se ha explicado anteriormente y el promedio con los pesos de todas supere o iguale el 50 %.

## Actividades de aprendizaje programadas (Se incluye programa)

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

**1:**

Actividades genéricas presenciales:

- Clases teóricas: Se explicarán los conceptos teóricos de la asignatura y se desarrollarán ejemplos prácticos ilustrativos como apoyo a la teoría cuando se crea necesario.
- Clases prácticas: Se realizarán problemas y casos prácticos como complemento a los conceptos teóricos estudiados.
- Prácticas de laboratorio: Los alumnos serán divididos en varios grupos de no más de 20 alumnos/as, estando tutorizados por el profesor.

## 2: Actividades genéricas no presenciales

- Estudio y asimilación de la teoría expuesta en las clases magistrales.
- Comprensión y asimilación de problemas y casos prácticos resueltos en las clases prácticas.
- Resolución de problemas propuestos, proyecto, etc.
- Preparación de las prácticas de laboratorio, elaboración de los guiones e informes correspondientes.
- Preparación de las pruebas escritas de evaluación continua y exámenes finales.

## 3:

Actividades autónomas tutorizadas: Aunque tendrán más bien un carácter presencial se han tenido en cuenta a parte por su idiosincrasia, estarán enfocadas principalmente a seminarios y tutorías bajo la supervisión del profesor.

## 4:

Actividades de refuerzo: De marcado carácter no presencial, a través de un portal virtual de enseñanza (Moodle) se dirigirán diversas actividades que refuercen los contenidos básicos de la asignatura. Estas actividades podrán ser personalizadas o no, controlándose su realización a través del mismo.

# Planificación y calendario

## Calendario de sesiones presenciales y presentación de trabajos

La asignatura consta de 6 créditos ECTS, lo cual representa 150 horas de trabajo del alumno/a en la asignatura durante el semestre, es decir, 10 horas semanales durante 15 semanas lectivas.

Un resumen de la distribución temporal orientativa de una semana lectiva puede verse en la tabla siguiente.

Actividad	Horas semana lectiva
Clases magistrales	3
Prácticas de laboratorio	1
Otras actividades	6

No obstante la tabla anterior podrá quedar más detallada, teniéndose en cuenta la distribución global siguiente:

- 42 horas de clase magistral, con un 40 % de exposición teórica y un 60 % de resolución de problemas tipo.
- 16 horas de prácticas de laboratorio, en sesiones de 2 horas.
- 2 horas de pruebas evaluatorias escritas, a razón de una hora por prueba.
- 45 horas de ejercicios y trabajos tutelados, repartidas a largo de las 15 semanas de duración del semestre.

— 45 horas de estudio personal, repartidas a lo largo de las 15 semanas de duración del semestre.

El horario semanal de la asignatura estará accesible al público en la web de la Escuela. Consta de 6 horas reservadas a la semana, de las cuales 4 se dedicarán a actividades presenciales. Las otras dos se podrán usar para adaptar el horario debido a variaciones del calendario lectivo o para otras actividades como tutorías en grupo.

En la tabla siguiente, se muestra el cronograma orientativo que recoge el desarrollo de las actividades presentadas con anterioridad, pudiendo variar en función del desarrollo de la actividad docente.

Actividad	Semana lectiva															Horas	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	P	T
Tema 1	2	2														4	42
Tema 2		2	2	2	4	2	2									14	
Tema 3							1	2	4	2	4					13	
Tema 4												2	4	2	3	11	
Práctica 1	2															2	16
Práctica 2			2													2	
Práctica 3				2												2	
Práctica 4						2										2	
Práctica 5								2								2	
Práctica 6										2						2	
Práctica 7												2				2	
Práctica 8														2		2	
Prueba 1							1									1	2
Prueba 2															1	1	
Ejercicios/Trabajos	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	45	45
Estudio personal	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	45	45
<b>Total</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>150</b>	<b>150</b>

Las fechas de los exámenes finales serán las publicadas de forma oficial en <http://www.eupla.es/secretaria/academica/examenes.html>.

El proyecto deberá entregarse como tarde la última semana.

Las fechas de los exámenes finales serán las publicadas de forma oficial en <http://www.eupla.es/secretaria/academica/examenes.html>.

El proyecto deberá entregarse como tarde la última semana.

## Contenidos

### Contenidos de la asignatura indispensables para la obtención de los resultados de aprendizaje.

El programa de la asignatura se estructura en torno a dos componentes de contenidos complementarios:

— Teóricos.

— Prácticos.

### 1-Contenidos teóricos.

- Bloque 1
  - Computador: Máquina que ejecuta Algoritmos. Noción de Algoritmo. Estructura del computador: Naturaleza Digital, codificación, hardware, software.
  - Sistemas operativos.
  - Bases de datos.
  - Programación: Estilos de Programación, jerarquía de lenguajes, elementos de programación.
  - Redes de computadores.
- Bloque 2
  - Abstracción con Procedimientos. Tipos de datos y esquemas de composición algorítmica. Concepto de tipo de dato.
  - Constantes y variables. Tipos de datos básicos: Booleano, carácter, entero, real.
  - Estructuras de control. Procedimientos y Funciones.
  - Técnicas de Diseño de algoritmos: Tratamiento de secuencias de E/S.
- Bloque 3
  - Abstracción con Datos. Nociones de clases y objetos.
  - Tuplas. Tablas.
  - Acceso Indexado. Ordenación.
  - Ficheros
  - Tipos Abstractos de Datos.
- Bloque 4
  - Clases, objetos y métodos. Herencia.
  - Aplicaciones típicas de la programación orientada a objetos.
  - Técnicas útiles en la programación orientada a objetos.

### 2-Contenidos prácticos.

Cada tema expuesto en la sección anterior, lleva asociadas prácticas al respecto, ya sean mediante supuestos prácticos, interpretación y comentario de lecturas asociadas a la temática y/o trabajos conducentes a la obtención de resultados y a su análisis e interpretación. Conforme se desarrollen los temas se irán planteando dichas Prácticas, bien en clase o mediante la plataforma Moodle.

- Práctica 1 **Manejo de sistema operativo GNU/Linux con lenguaje de órdenes.  
Primer programa.**
- Práctica 2 Ejercitar el uso de variables y constantes. Distinguir los usos de diferentes tipos y ver la ayuda que supone un lenguaje con una estructura de tipos estricta. Conversiones entre tipos..
- Práctica 3 **Construir algoritmos I.**
- Práctica 4 **Construir algoritmos II.**
- Práctica 5 **Uso de tablas y atributos.**
- Práctica 6 **Algoritmos con estructuras de datos.**
- Práctica 7 **Uso de ficheros. Serialización.**
- Práctica 8 **Herencia, encapsulación y polimorfismo. Colecciones e iteradores.**

## Recursos

### Materiales

Material	Soporte
Presentaciones temario Problemas temario Enlaces de interés	Digital/Moodle, Papel
Software de desarrollo	Pc's laboratorio

## Bibliografía

### Bibliografía

Además del propio texto específico de la asignatura publicado al efecto, confeccionado expresamente por el profesor, se tendrá en cuenta la siguiente bibliografía, para consulta del alumno/a.

#### Bibliografía de bloques I, II y III

Antonio Ortega y Eduardo Falces. Programación de Ordenadores bajo Component Pascal y Java. Apuntes de la EUPLA.

Allen B. Downey  
. How to Think Like a Computer Scientist  
(<http://www.greenteapress.com/thinkapjava/thinkapjava.pdf>)

*Niklaus Wirth. Algoritmos + Estructuras de Datos = Programas. Ediciones del Castillo S. A., 1984.*

Aho, Hopcroft, Ullman. Estructuras de Datos y Algoritmos. Addison Wesley Iberoamericana, 1988.

Bruce Eckel. Thinking in Java. Ed. Prentice Hall, 1998.

Durán, Guitiérrez, Pimentel. Programación Orientada a Objetos con Java. Thomson Editores Spain, 2007.

#### Bibliografía de bloque IV

Antonio Ortega y Eduardo Falces. Programación de Ordenadores bajo Component Pascal y Java. Apuntes de la EUPLA.

Allen B. Downey  
. How to Think Like a Computer Scientist  
(<http://www.greenteapress.com/thinkajava/thinkajava.pdf>)

*Bruce Eckel. Thinking in Java. Ed. Prentice Hall, 1998.*

Durán, Guitiérrez, Pimentel. Programación Orientada a Objetos con Java. Thomson Editores Spain, 2007.

Gamma, Helm, Johnson, Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley Professional, 1994.

Eric Freeman y Elisabeth Freeman. Head First Design Patterns. Ed. O'Reilly, 2004.

Martin Fowler y Kendall Scott. UML Distilled: Applying the Standard Object Modeling Language. Addison Wesley, 1997.

---

## Actividades y recursos

### Perfil defensa

---

## Presentación metodológica general

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

Todas las actividades que se realizan en el día a día de la asignatura cuentan para su evaluación final. Por tanto, la **evaluación** de la asignatura se realizará de forma **continuada** mediante ejercicios, prácticas, proyectos, controles y examen final.

## Actividades de aprendizaje programadas (Se incluye programa)

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

**1:**  
El programa de la asignatura incluye los siguientes temas:

**Tema 1: Arquitectura de ordenadores, hardware y software**

**Tema 2: Lenguajes y entorno de programación**

**Tema 3: Tipos de datos predefinidos**

**Tema 4: Programación modular I**

**Tema 5: Estructuras de control**

**Tema 6: Programación modular II**

**Tema 7: Tratamiento de excepciones y ficheros**

## **Tema 8: Tipos de datos estructurados I**

## **Tema 9: Tipos de datos estructurados II**

## **Tema 10: Algoritmos**

En cada uno de estos temas se hace uso de distintas actividades de entre las programadas en la asignatura para alcanzar los objetivos específicos para Fundamentos de Informática y algunos de los generales para la titulación.

# **Planificación y calendario**

## **Calendario de sesiones presenciales y presentación de trabajos**

### Actividades de evaluación

Trabajos prácticos (problemas, prácticas de laboratorio y proyectos): 30%

Prueba teórico-práctica: 70%

### Entregas/trabajos prácticos

Se distinguen tres tipos de entregas: problemas, prácticas y proyectos.

1. Los problemas o ejercicios, a desarrollar sobre papel, se solicitarán en el transcurso de las clases.

2. Al finalizar cada clase de prácticas se deberá entregar la resolución de las mismas, de modo individual.

3. De modo autónomo se resolverán dos proyectos a lo largo del curso. Estos serán desarrollados en parejas, y se entregarán vía telemática dentro de un plazo dado.

Los enunciados de prácticas se deben leer y entender antes de ir a clase. Además, en general, es recomendable comenzar los ejercicios de prácticas también con anterioridad, de modo que se pueda utilizar el tiempo en el aula para resolver los problemas que hayan surgido.

Cada proyecto entregado constará del código fuente del programa en Pascal y de una documentación que contenga los datos de quienes la realizan, el método de resolución empleado y las decisiones de diseño de programa adoptadas.

En caso de plagio en cualquier entrega para la asignatura (o cualquier otro comportamiento indebido) se obtendrá como resultado un 0 en dicha entrega y un endurecimiento progresivo en las condiciones mínimas para superar la asignatura, sin perjuicio de que se puedan tomar medidas adicionales. Todo ello tanto para el grupo que ha copiado como para el grupo que se ha dejado copiar.

### Evaluación

La evaluación de la asignatura se realizará de forma continuada mediante distintas pruebas: ejercicios, prácticas, proyectos, controles y examen final. La calificación de la asignatura se calculará considerando que el resultado obtenido de los proyectos supone el 15% (1,5 puntos de 10), el obtenido del resto de entregas y controles el 15% (1,5 puntos de 10) y la prueba final supone el 70% de la nota (7 puntos de 10).

El examen final se divide a su vez en dos partes. Ambas forman una única entidad, no guardándose partes entre convocatorias.

- Un test (con penalización para respuestas incorrectas), que evaluará los conocimientos y el saber hacer de todo lo aprendido (en clases magistrales, problemas, prácticas y proyectos). El resultado obtenido en esta parte supondrá el 25% de la nota final (2,5 puntos de 10).

- Unos problemas, en los que se deberá demostrar las habilidades de programación adquiridas durante el curso. Esta parte supondrá el 45% de la nota final (4,5 puntos de 10).

A lo largo del semestre se abordarán una serie de ejercicios, prácticas de laboratorio, proyectos y controles. Estos representarán el 30% de la nota final. Finalizadas las clases, tendrá lugar la prueba teórico-práctica, que representará el 70% de la nota final.

Más detalladamente:

### **Entregas/trabajos prácticos (30%)**

Se distinguen tres tipos de entregas: problemas, prácticas y proyectos.

1. Los problemas o ejercicios, a desarrollar en pizarra o sobre papel, se solicitarán en clase.
2. Al finalizar cada clase de prácticas se deberá entregar la resolución de las mismas, de modo individual o responder al cuestionario correspondiente.
3. De modo autónomo se resolverán dos proyectos a lo largo del curso. Estos serán desarrollados en pareja, y se entregarán por vía telemática dentro de un plazo dado. Cada proyecto entregado constará del código fuente del programa y de una documentación que contenga los datos de quienes la realizan, el método de resolución empleado y las decisiones de diseño de programa adoptadas.

En caso de plagio en cualquier entrega para la asignatura (o cualquier otro comportamiento indebido) se obtendrá como resultado un 0 en dicha entrega y un endurecimiento progresivo en las condiciones mínimas para superar la asignatura, sin perjuicio de que se puedan tomar medidas adicionales. Todo ello tanto para el grupo que ha copiado como para el grupo que se ha dejado copiar.

### **Prueba teórico-práctica (70%)**

El examen final se divide a su vez en dos partes, formando ambas una única entidad, no guardándose partes entre convocatorias.

1. Un test (con penalización para respuestas incorrectas), que evaluará los conocimientos y el saber hacer de todo lo aprendido (en clases magistrales, problemas, prácticas y proyectos). El resultado obtenido en esta parte supondrá el 25% de la nota final (2,5 puntos de 10).
2. Unos problemas, en los que se deberá demostrar las habilidades de programación adquiridas durante el curso. Esta parte supondrá el 45% de la nota final (4,5 puntos de 10).

## **Referencias bibliográficas de la bibliografía recomendada**

### **Escuela Universitaria Politécnica de La Almunia**

- Cortés Arcos, Tomás. Apuntes de Fundamentos de Informática/Tomás Cortés Arcos. 1ª edic. La Almunia:EUPLA,2012
- Design patterns : Elements of reusable object-oriented software / Erich Gamma...[et al.] . - 15th. pr. Reading, Massachusetts : Addison-Wesley, 1998
- Durán, Francisco. Programación orientada a objetos con Java / Francisco Durán, Francisco Gutiérrez, Ernesto Pimentel Madrid : Thomson Paraninfo, D. L. 2007
- Eckel, Bruce. Piensa en Java / Bruce Eckel ; traducción, Jorge González Barturen ; revisión técnica, Javier Parra Fuente, Ricardo Lozano Quesada ; coordinación general y revisión técnica, Luis Joyanes Aguilar . - 2ª ed. Madrid [etc.] : Prentice Hall, D.L. 2002
- Fowler, Martin. UML distilled : applying the standard object modeling language / Martin Fowler with Kendall Scott[]. - 1ª edición Reading, Massachusetts [etc.] : Addison-Wesley Longman, 1997
- Freeman, Eric. Head first design patterns / Eric Freeman, Elisabeth Freeman ; with Kathy Sierra, Bert Bates. - 1ª edición Beijing [etc.] : O'reilly, 2004
- John, Latham. Java:just in time/John Latham. - 1ª edic Colleague Publications, 2010.
- Ortega, Antonio. Apuntes Asignatura de Fundamentos de Informática/ Antonio Ortega;Eduardo Falces. - 1ª edic. La Almunia: EUPLA,
- Wirth, Niklaus. Algoritmos + estructuras de datos = programas / Niklaus Wirth ; versión castellana de Angel Alvarez Rodríguez y José Cuenca Bartolomé. - 1ª edic. Ediciones del Castillo,S.A.1984.

### **Centro Universitario de la Defensa**

- Desarrollo de algoritmos y técnicas de programación en Pascal / Cristobal Pareja Flores...[et al.] . - [1a. ed.] Madrid : RA-MA,

1997

- Leestma, Sanford. Programación en Pascal / Sanford Leestma, Larry Nyhoff ; traducción, Natalia López, Fernando Rubio, Clara M<sup>a</sup> Segura ; revisión técnica, Sebastián Dormido Bencomo . - 4<sup>a</sup> ed. en español, reimp. Madrid : Pearson Educación, 2006
- Prieto Espinosa, Alberto. Introducción a la informática / Alberto Prieto Espinosa, Antonio Lloris Ruiz, Juan Carlos Torres Cantero . - 4<sup>a</sup> ed. Madrid [etc.] : MacGraw-Hill, D.L. 2006
- Valls Ferrán, José María. Programación estructurada y algoritmos en PASCAL / José María Valls Ferrán, David Camacho Fernández . - [reimp.] Madrid : Pearson Educación, D.L. 2010