

## **Grado en Ingeniería Informática**

### **30209 - Programación II**

**Guía docente para el curso 2014 - 2015**

**Curso: 1, Semestre: 2, Créditos: 6.0**

---

## **Información básica**

---

### **Profesores**

- **Jorge Raul Bernad Lusilla** jbernad@unizar.es
- **Joaquín Antonio Ezpeleta Mateo** ezpeleta@unizar.es
- **Carlos Catalán Cantero** ccatalan@unizar.es
- **Jesús Gallardo Casero** jesus.gallardo@unizar.es
- **Francisco Javier Martínez Rodríguez** jamarro@unizar.es

### **Recomendaciones para cursar esta asignatura**

El alumno que curse esta asignatura ha de contar con una formación en programación del nivel de la asignatura de Programación I . Por otra parte una adecuada formación matemática en los estudios previos resulta muy conveniente, así como conocimientos básicos de lógica de predicados.

### **Actividades y fechas clave de la asignatura**

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

---

## **Inicio**

---

## **Resultados de aprendizaje que definen la asignatura**

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

**1:**

Desarrolla programas robustos de tamaño medio de forma modular y orientada a objetos, dotándoles de robustez.

**2:**

Especifica formalmente algoritmos.

**3:**

Diseña algoritmos recursivos e iterativos correctos.

**4:**  
Analiza el coste de algoritmos iterativos y recursivos.

**5:**  
Prueba formalmente la corrección de algoritmos simples.

## Introducción

### Breve presentación de la asignatura

Esta asignatura persigue que el alumno mejore su capacidad para diseñar programas de ordenador haciendo énfasis en su corrección y su eficiencia. Para ello el alumno va a aprender a especificar formalmente algoritmos, a diseñar algoritmos iterativos y recursivos correctos, a probar de forma rigurosa la corrección de estos algoritmos y a analizar su coste. A la vez profundizará en el aprendizaje de la tecnología asociada a la programación orientada a objetos.

---

## 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:

Tras haber cursado una primera asignatura de programación en la que se ha aprendido a diseñar pequeños programas, en esta asignatura el alumno va a aprender la tecnología y las metodologías a aplicar para diseñar programas de mayor entidad y, de forma prioritaria, velar porque los diseños realizados sean correctos, robustos y eficientes.

La asignatura tiene un marcado carácter aplicado. El alumno aprenderá los conceptos necesarios sobre especificación, corrección, diseño y evaluación del coste y, sobre todo, aprenderá a aplicarlos en el diseño de una variedad de problemas de tratamiento de información.

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

Programación II toma el relevo de la asignatura Programación I y constituye el segundo peldaño en la formación en programación de un estudiante de Ingeniería Informática. Es una asignatura obligatoria englobada en la materia de formación común en Programación y Computación.

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

**1:**  
Conocer y aplicar los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

**2:**  
Analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

**3:**  
Concebir, diseñar y desarrollar proyectos de Ingeniería.

**4:**  
Resolver problemas y tomar decisiones con iniciativa, creatividad y razonamiento crítico.

**5:**

Usar las técnicas, habilidades y herramientas de la Ingeniería necesarias para la práctica de la misma.

- 6:** Aprender de forma continuada y desarrollar estrategias de aprendizaje autónomo.
- 7:** Aplicar las tecnologías de la información y las comunicaciones en la Ingeniería.

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

Aprender a programar es esencial para un ingeniero informático. Lo que aprenda en esta asignatura, que complementa a lo ya aprendido en Programación I, será un importante paso adelante en su formación como programador de computadores.

---

## **Evaluación**

---

### **Actividades de evaluación**

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

- 1:** **En la Escuela de Ingeniería y Arquitectura del Campus Rio Ebro:**

**Convocatoria de Junio.** 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 4.0 puntos en esta prueba para aprobar la asignatura. Si se obtiene esta nota mínima, entonces la prueba pondera un 70% en la nota de la asignatura y, si no se alcanza este mínimo, entonces la calificación en la asignatura es la de esta prueba.

**P2.** Trabajos y prueba de programación con computador. Esta prueba pondera un 30% en la nota de la asignatura. Cada alumno deberá entregar los trabajos de programación que se indiquen en las prácticas de la asignatura y presentarse a una prueba de programación en laboratorio donde deberán realizar un trabajo de programación relacionado con una parte del trabajo entregado. Se valorará tanto el trabajo entregado como el realizado en la prueba, con pesos análogos. El alumno que no presente los trabajos de programación que se indiquen o que no se presente a la prueba será calificado con un cero.

**Pruebas con carácter voluntario.** A lo largo del cuatrimestre se plantearán varias pruebas voluntarias consistentes en la resolución, por escrito en clase o en casa, de ejercicios y problemas. El 10% de la calificación de estas pruebas, es decir, entre 0 y 1.0 puntos, se sumará a la calificación obtenida por el alumno en la convocatoria de Junio de la asignatura (no en la de Septiembre), siempre y cuando haya aprobado la asignatura ( $0.7*P1+0.3*P2 \geq 5.0$ ).

**Convocatoria de Septiembre.** La evaluación de la asignatura se realiza en base a dos pruebas análogas a las de la convocatoria de Junio, con las mismas ponderaciones y exigencia de notas mínimas. Las calificaciones del alumno obtenidas en la convocatoria de Junio en cualquier 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 del Campus de Teruel:

**En la Escuela Universitaria Politécnica del Campus de Teruel:**

La nota final de la asignatura en la convocatoria ordinaria se divide de la siguiente forma:

- Prácticas. 30% de la nota final.** A su vez se divide en:

- Prácticas entregables: 15% de la nota. Hay tres prácticas entregables durante el curso, teniendo todas ellas el mismo peso en la nota.
- Examen de prácticas: 15% de la nota. Consiste en una prueba en laboratorio el día del examen final en la cual se pedirá realizar un programa similar al de las prácticas.

Si el alumno no llega a una nota media de 4 sobre 10 en las prácticas entregables, o si la nota del examen es superior a la media de las prácticas entregables, entonces toda la nota de prácticas (30%) corresponde al examen de prácticas. En cualquier caso, debe alcanzarse una nota de 4 sobre 10 en el examen de prácticas para poder aprobar la asignatura.

- **Examen escrito. 70% de la nota final.** Tendrá una parte de ejercicios de programación (identificar errores, completar programas o métodos, identificar la salida de un programa, realizar un diseño de clases, etc.) y otra de ejercicios de algorítmica (especificar precondiciones y postcondiciones, realizar el diseño recursivo de un algoritmo, calcular el coste de un algoritmo, etc.) Debe alcanzarse una nota de 3 sobre 7 para poder aprobar la asignatura.

En cuanto a la **convocatoria extraordinaria (septiembre)**, la nota final será la nota del examen extraordinario, teniendo en cuenta que ese examen tendrá una parte de prácticas en laboratorio que valdrá el 30% de la nota total. Esa parte no tendrán que hacerla aquellos alumnos que llegaran a una nota media de 4 en la parte de prácticas de la convocatoria ordinaria, ya que se les guardaría la nota, aunque pueden hacerla para mejorar la nota. En ese caso, se quedarían con la nota mayor entre las dos obtenidas.

---

## Actividades y recursos

---

### 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 y el diseño de programas correctos y eficientes a través de las clases magistrales, en las que se favorecerá la participación de los alumnos.
3. La aplicación de tales conocimientos al diseño y análisis de algoritmos y programas en las clases de problemas. En estas clases los alumnos desempeñarán un papel activo en la discusión y resolución de los problemas. En algunas de estas clases el trabajo de cada alumno será evaluado, mediante la realización de pruebas con carácter voluntario.
4. Las clases de prácticas en laboratorio en las que el alumno aprenderá la tecnología necesaria para desarrollar pequeños proyectos de programación utilizando un lenguaje de programación determinado.
5. El trabajo en equipo desarrollando algún pequeño proyecto de programación cuyo resultado se plasma en la entrega de programas resultantes convenientemente diseñados y documentados.
6. Un trabajo continuado en el que se conjugue la comprensión de conceptos, el análisis y la resolución de problemas de programación utilizando "lápiz y papel" y la puesta a punto en computador de algunos pequeños proyectos de programación.

### 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:**

En las clases impartidas en el aula se desarrollará el temario de la asignatura.

**2:**

En las clases de problemas se resolverán problemas de aplicación de los conceptos y técnicas presentadas en el programa de la asignatura.

**3:**

Las sesiones de prácticas se desarrollan en un laboratorio informático. A lo largo de sus sesiones cada alumno deberá realizar, individualmente o en equipo, trabajos de programación directamente relacionados con los temas estudiados en la asignatura.

# Planificación y calendario

## Calendario de sesiones presenciales y presentación de trabajos

La organización docente de la asignatura prevista en el Campus Río Ebro 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 programación en laboratorio, tuteladas por un profesor.

La organización docente de la asignatura prevista en el campus de Teruel es la siguiente.

- Clases teóricas (2 horas semanales)
- Clases de problemas y de prácticas de laboratorio (2 horas semanales)

Presentación de trabajos objeto de evaluación:

- En los problemas y trabajos de programación que se propongan se informará de su fecha de entrega al ser propuestos.

# Programa de la asignatura

## Temario

- Especificación formal de algoritmos.
- Diseño de algoritmos recursivos.
- Análisis del coste de un algoritmo.
- Corrección de algoritmos iterativos y recursivos.
- Diseño de algoritmos correctos.
- Programación modular y orientación a objetos.
- Desarrollo de un proyecto de programación.

# 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:

En la Escuela de Ingeniería y Arquitectura:

- 50 horas, aproximadamente, de actividades presenciales (clases teóricas, de problemas y prácticas en laboratorio)
- 40 horas de trabajo de programación en equipo
- 57 horas de estudio personal efectivo (estudio de apuntes y textos, resolución de problemas, preparación clases y prácticas, desarrollo de programas)
- 3 horas de examen final escrito

En la Escuela Universitaria Politécnica de Teruel:

- 60 horas de actividades presenciales (clases teóricas, de problemas y prácticas en laboratorio)
- 30 horas de trabajo de programación tutorizado
- 55 horas de estudio personal efectivo
- 5 horas de actividades de evaluación

# Bibliografía

## Bibliografía de la asignatura

Materiales propios de la asignatura:

- MARTÍNEZ, J.: *Notas de la asignatura Programación 2*. Servicio de reprografía de la EINA. 2014
- MARTÍNEZ, J.: *Prácticas de Programación 2* . Servicio de reprografía de la EINA. 2014

Libros de consulta:

- BALCÁZAR, J.L.: *Programación metódica*. Madrid, McGraw-Hill Iberoamericana. 1993. ISBN: 84-481-1957-6.
- DEITEL, P.J. y DEITEL, H.M.: *Java. Cómo programar*. Pearson-Prentice Hall. 7<sup>a</sup> edición. 2008. ISBN: 10-970-26-1190-63.
- BRASSARD, P. y BRATLEY, P.: *Fundamentos de Algoritmia*. Ed. Prentice-Hall. 1997. ISBN: 84-311-0531-3.
- CASTRO J., CUCKER F. et al: *Curso de Programación*. McGraw Hill. 1997. ISBN: 84-481-1959-2
- PEÑA, R.: *Diseño de programas. Formalismo y Abstracción*. Tercera edición. Prentice-Hall. 2005. ISBN: 84-205-4191-5

## Referencias bibliográficas de la bibliografía recomendada

### Escuela de Ingeniería y Arquitectura

- Balcázar, José Luis. *Programación metódica* / José Luis Balcázar Madrid [etc] : McGraw-Hill, D. L. 1998
- Brassard, Gilles : *Fundamentos de algoritmia* / G. Brassard, T. Bratley ; traducción, Rafael García-Bermejo ; revisión técnica, Narciso Martí, Ricardo Peña, Luis Joyanes Aguilar . - 1<sup>a</sup> ed. en español, reimp. Madrid [etc.] : Prentice Hall, 2008
- *Curso de programación* / Jorge Castro Rabal...[et al.] . - [1a ed., reimp.] Madrid : McGraw-Hill, D.L.1994
- Deitel, Paul J. *Java : cómo programar* / P. J. Deitel, H. M. Deitel ; traducción Alfonso Vidal Romero Elizondo ; revisión técnica Gabriela Azucena Campos García, Roberto Martínez Román, Jorge Armando Aparicio Lemus. - 7<sup>a</sup> ed. Naucalpan de Juárez (Estado de México) : Pearson Educación, 2008
- Martínez Rodríguez, Javier. *Notas de la asignatura Programación 2 : Grado en Ingeniería Informática* / Javier Martínez Rodríguez . Servicio de reprografía de la EINA, 2014
- Martínez Rodríguez, Javier. *Prácticas de Programación 2 : Grado en Ingeniería Informática* / Javier Martínez Rodríguez Servicio de reprografía de la EINA, 2014
- Peña Marí, Ricardo. *Diseño de programas : formalismo y abstracción* / autor, Ricardo Peña Marí . - 3<sup>a</sup> ed. Madrid [etc.] : Pearson Educación, D.L. 2004

### Escuela Universitaria Politécnica

- Balcázar, José Luis. *Programación metódica* / José Luis Balcázar Madrid [etc] : McGraw-Hill, D. L. 1998
- Brassard, Gilles. *Fundamentos de algoritmia* / G. Brassard, T. Bratley ; traducción, Rafael García-Bermejo ; revisión técnica, Ricardo Peña, Narciso Martí . - 1a ed., 1a reimp. Madrid [etc.] : Prentice Hall, D.L. 1998
- *Curso de programación* / Jorge Castro Rabal...[et al.] . - [1a ed., reimp.] Madrid : McGraw-Hill, D.L.1994
- Deitel, Paul J.. *Java : cómo programar* / P. J. Deitel, H. M. Deitel ; traducción Alfonso Vidal Romero Elizondo ; revisión técnica Gabriela Azucena Campos García, Roberto Martínez Román, Jorge Armando Aparicio Lemus. - 7<sup>a</sup> ed. Naucalpan de Juárez (Estado de México) : Pearson Educación, 2008
- Peña Marí, Ricardo. *Diseño de programas : formalismo y abstracción* / autor, Ricardo Peña Marí . - 3<sup>a</sup> ed. Madrid [etc.] : Pearson Educación, D.L. 2004