

Curso : 2018/19

30209 - Programación II

Información del Plan Docente

Año académico:	2018/19
Asignatura:	30209 - Programación II
Centro académico:	110 - Escuela de Ingeniería y Arquitectura 326 - Escuela Universitaria Politécnica de Teruel
Titulación:	439 - Graduado en Ingeniería Informática 443 - Graduado en Ingeniería Informática
Créditos:	6.0
Curso:	443 - Graduado en Ingeniería Informática: 1 439 - Graduado en Ingeniería Informática: 1
Periodo de impartición:	Segundo Semestre
Clase de asignatura:	Obligatoria
Módulo:	Materia básica de grado

Información Básica

Objetivos 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 .

Recomendaciones para cursar la 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.

Competencias y resultados de aprendizaje

Competencias

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

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.

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

Concebir, diseñar y desarrollar proyectos de Ingeniería.

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

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

Aprender de forma continuada y desarrollar estrategias de aprendizaje autónomo.

Aplicar las tecnologías de la información y las comunicaciones en la Ingeniería.

Resultados de aprendizaje

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

Desarrolla programas robustos de tamaño medio de forma modular, dotándoles de robustez.

Especifica formalmente algoritmos.

Diseña algoritmos recursivos e iterativos correctos.

Analiza el coste de algoritmos iterativos y recursivos.

Prueba formalmente la corrección de algoritmos simples.

Importancia de los resultados de aprendizaje

Aprender a programar es esencial para un ingeniero informático. Lo que aprenda en esta asignatura, que complementa lo ya aprendido en Programación I, será fundamental para la formación en los aspectos relativos al desarrollo de aplicaciones y para sentar unas bases robustas para progresar a lo largo de los estudios del grado.

Evaluación

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 del Campus Rio Ebro:

La evaluación de la asignatura en la convocatoria de junio consta de tres pruebas:

- **Examen escrito** en el que el alumno ha de responder, en su caso, preguntas conceptuales y resolver problemas. Pondera un 70% en la nota de la asignatura. Es necesario obtener en este examen escrito una nota mínima de 4.0 puntos sobre 10 para optar a aprobar la asignatura. En caso de no alcanzar dicha puntuación mínima, la calificación de este examen escrito es la que constará en el acta de la asignatura.
- **Examen práctico** en el que el alumno ha de realizar individualmente un pequeño trabajo de programación en el laboratorio. Pondera un 15% en la nota de la asignatura.
- **Trabajo** que será propuesto por los profesores de la asignatura con un plazo de entrega determinado. Los profesores indicarán si este trabajo debe realizarse individualmente o en equipo. Pondera un 15% en la nota de la asignatura.

Se considerarán no presentados en esta convocatoria, exclusivamente, los alumnos que no se presenten al examen escrito.

Opcionalmente, a lo largo del cuatrimestre, los profesores de la asignatura podrán proponer una o más pruebas voluntarias de evaluación. Si su calificación total es igual o superior a 5 sobre 10, un 10% de la misma, hasta un máximo de 1 punto, se sumará a la calificación obtenida en la convocatoria de junio en el caso de haber aprobado la asignatura. Esta mejora de la calificación no se aplica a quienes no hayan aprobado la asignatura en febrero ni se conserva para la convocatoria de septiembre.

La evaluación de la asignatura en la convocatoria de septiembre consta de dos pruebas:

- **Examen escrito** en el que el alumno ha de responder, en su caso, a preguntas conceptuales y resolver problemas. Pondera un 70% en la nota de la asignatura. Es necesario obtener en este examen escrito una nota mínima de 4.0 puntos sobre 10 para optar a aprobar la asignatura. En caso de no alcanzar dicha puntuación mínima, la calificación de este examen escrito será la que constará en el acta de la asignatura.
- **Examen práctico** en el que el alumno ha de realizar individualmente un pequeño trabajo de programación en el laboratorio. Pondera un 30% en la nota de la asignatura.

Calificaciones obtenidas en junio válidas en septiembre.

- Al alumno que opte por no realizar el examen práctico de septiembre, se le aplicará como calificación la suma de las calificaciones obtenidas en el examen práctico de junio y en el trabajo de programación de junio.
- Al alumno que opte por realizar el examen práctico de septiembre y que opte por no realizar el examen escrito de septiembre, se le aplicará como calificación del examen escrito la obtenida en el examen escrito de junio.

Se considerarán no presentados en esta convocatoria, los alumnos que no se presenten ni al examen escrito ni al examen práctico de septiembre.

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. 40% de la nota final.** Esta nota se alcanzará mediante el desarrollo por parte de los alumnos de una serie de trabajos prácticos de programación a realizar a lo largo del curso. A la hora de calificar estos trabajos se valorará su funcionamiento según las especificaciones, la calidad de su diseño y su presentación, la adecuada aplicación de los métodos de resolución, el tiempo empleado, así como la capacidad de los alumnos para explicar y justificar el diseño realizado.

Si el alumno no entrega dichos trabajos prácticos en su totalidad, o si la calificación media de los mismos es inferior a 5 sobre 10, el alumno puede optar a una prueba alternativa para alcanzar esa calificación, que tendrá lugar el día del examen final.

- **Examen escrito. 60% de la nota final.** Se trata de un examen teórico-práctico con ejercicios y problemas que abarcan todo el temario de la asignatura. En general, se valorará la calidad y claridad de las respuestas y soluciones propuestas, su adecuación a las especificaciones y restricciones planteadas, la calidad de los diseños, la adecuada aplicación de los métodos de resolución y el tiempo empleado. Deberá obtenerse una calificación mayor de 4 sobre 10 para poder superar el examen final.

De cara a la **convocatoria extraordinaria (septiembre)**, la evaluación constará de las mismas partes que en la convocatoria ordinaria, teniendo en cuenta que, de haber superado ya alguna de las mismas en su momento, no será necesario repetir dicha parte, y se guardará la nota antes obtenida.

Metodología, actividades de aprendizaje, programa 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 plasme en la entrega de los 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 un computador, de algunos pequeños proyectos de programación.

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 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.

Programa

- 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.
- Desarrollo de un proyecto de programación.

Planificación de las actividades de aprendizaje y calendario de fechas clave

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

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

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 semana). 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.

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

Bibliografía y recursos recomendados

Teruel:

- | | |
|-----------|---|
| BB | Balcázar, José Luis. Programación metódica / José Luis Balcázar. Madrid [etc] : McGraw-Hill, D. L. 1998 |
| BB | Brassard, Gilles. Fundamentos de algoritmia / G. Brassard y Robert Bratley ; traducción, Rafael García-Bermejo ; revisión técnica, Ricardo Peña, Narciso Martí . - 1a ed., 1a reimp. Madrid [etc] : Prentice Hall, D.L. 1998 |
| BB | Curso de programación / Jorge Castro Rabal...[et al.] . - [1a ed.] reimp.] Madrid : McGraw-Hill, D.L.1994 |
| BB | 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 Romero, Jorge Armando Aparicio Lemus. - 7ª ed. Naucalpan de Juárez (Estado de México) : Pearson Educación, 2008 |
| BB | Peña Marí, Ricardo. Diseño de programas : formalismo y abstracción / autor, Ricardo Peña Marí . - 3ª ed. Madrid [etc] : Pearson Educación, D.L. 2004 |
| BB | Balcázar, José Luis. Programación metódica / José Luis Balcázar. Madrid [etc] : McGraw-Hill, D. L. 1998 |
| BB | Brassard, Gilles : Fundamentos de algoritmia / G. Brassard y Robert Bratley ; traducción, Rafael García-Bermejo ; revisión técnica, Narciso Martí, Ricardo Peña, Luis Joyanes Aguilar . - 1ª ed. español, reimp. Madrid [etc.] : Prentice Hall, 2008 |
| BB | Peña Marí, Ricardo. Diseño de programas : formalismo y abstracción / autor, Ricardo Peña Marí . - 3ª ed. Madrid [etc] : Pearson Educación, D.L. 2004 |