

## 39811 - Programación II

### Información del Plan Docente

**Año académico:** 2022/23

**Asignatura:** 39811 - Programación II

**Centro académico:** 326 - Escuela Universitaria Politécnica de Teruel

**Titulación:** 634 - Programa conjunto en Ingeniería Informática-Administración y Dirección de Empresas

**Créditos:** 6.0

**Curso:** 1

**Periodo de impartición:** Segundo semestre

**Clase de asignatura:** Formación básica

**Materia:**

## 1. Información Básica

### 1.1. 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.

Toda la formación que aporta esta asignatura (teórica y práctica) contribuye de forma transversal a la AGENDA 2030 y ODS ya que su formación capacita al estudiante para contribuir al desarrollo y gestión de los 245 indicadores de los ODS que plantea el PNUMA.

### 1.2. 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.

### 1.3. 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.

## 2. Competencias y resultados de aprendizaje

### 2.1. 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.

## 2.2. 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.

## 2.3. 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.

# 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

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 no sumará nota y deberá realizar una prueba alternativa para alcanzar esa calificación, que tendrá lugar el día del examen final.
- **Examen 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 o igual que 5 sobre 10 para poder superar la asignatura. Si no se alcanzara esa nota, la calificación final será el mínimo entre la media ponderada de las dos partes y la nota del examen.

De cara a la **convocatoria extraordinaria**, la evaluación constará de un examen con 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.

# 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 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 **podrá ser evaluado**, mediante la realización de pruebas con carácter voluntario.
4. Las clases de prácticas **de 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 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.

## 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 **de teoría** 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.

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

## 4.3. 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.
- Trabajo con algunos esquemas algorítmicos básicos.
- Desarrollo de un proyecto de programación.

## 4.4. 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:

60 horas de actividades dirigidas por el profesor (clases teóricas, de problemas y prácticas en laboratorio)  
85 horas de estudio y trabajo personal efectivo (estudio de apuntes y textos, resolución de problemas, preparación de clases y prácticas, desarrollo de programas)  
5 horas de actividades de evaluación

### Calendario de sesiones presenciales y presentación de trabajos

La organización docente de la asignatura 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.

## 4.5. Bibliografía y recursos recomendados

<http://psfunizar10.unizar.es/br13/egAsignaturas.php?codigo=30209&Identificador=12495>