Universitat Autònoma de Barcelona

**2023/2024**

# Programación avanzada

Código: 100126
Créditos ECTS: 6

| Titulación | Tipo | Curso | Semestre |
|---|---|---|---|
| 2500149 Matemáticas | OT | 4 | 1 |

## Contacto

Nombre: Vicente Soler Ruíz

Correo electrónico: vicenc.soler@uab.cat

## Idiomas de los grupos

Puede consutarlo a través de este enlace. Para consultar el idioma necesitará introducir el CÓDIGO de la asignatura. Tenga en cuenta que la información es provisional hasta el 30 de noviembre del 2023.

## Prerrequisitos

Conviene haber aprendido los conocimientos en programación básica en C en los primeros cursos del Grado o en otras asignaturas, aunque no es imprescindible.

## Objetivos y contextualización

La asignatura tiene dos partes bien diferenciadas. Un primer objetivo es conseguir un buen dominio de programación en C, de manera que los alumnos sean el capaces de estructurar una aplicación y de programar un algoritmo complejo. Un segundo objetivo es conseguir una formación en programación orientada a objetos (con Java), creación de librerías y que sirva de base para el aprendizaje de cualquier lenguage orientado a objetos.

Las Prácticas se basan en hacer dos juegos: uno en C y el otro en Java.
El desarrollo de juegos como base de las prácticas de la asignatura proporciona la obligatoriedad de utilizar muchos aspectos diferentes de los lenguajes de programación y hace que las prácticas sean más atractivas para el alumno. Todo esto motiva para que el alumno esté más interesado por el desarrollo de la asignatura.

## Competencias

- Demostrar de forma activa una elevada preocupación por la calidad en el momento de argumentar o hacer públicas las conclusiones de sus trabajos.
- Desarrollar un pensamiento y un razonamiento crítico y saber comunicarlo de manera efectiva, tanto en las lenguas propias como en una tercera lengua.
- Distinguir, ante un problema o situación, lo que es sustancial de lo que es puramente ocasional o circunstancial.

- Generar propuestas innovadoras y competitivas en la investigación y en la actividad profesional.
- Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.
- Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
- Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
- Reconocer la presencia de las Matemáticas en otras disciplinas.
- Trabajar en equipo.
- Utilizar aplicaciones informáticas de análisis estadístico, cálculo numérico y simbólico, visualización gráfica, optimización u otras para experimentar en Matemáticas y resolver problemas.
- Utilizar eficazmente bibliografía y recursos electrónicos para obtener información.

## Resultados de aprendizaje

1. Conocer la arquitectura y las especificaciones de funcionamiento de los productos software, conociendo la naturaleza y las posibilidades que los distintos lenguajes de codificación le proporcionan.
2. Conocer las distintas actividades implicadas en las fases del ciclo de vida de productos software y sus aplicaciones.
3. Demostrar de forma activa una elevada preocupación por la calidad en el momento de argumentar o hacer públicas las conclusiones de sus trabajos.
4. Desarrollar un pensamiento y un razonamiento crítico y saber comunicarlo de manera efectiva, tanto en las lenguas propias como en una tercera lengua.
5. Dominar las metodologías y herramientas de desarrollo de los sistemas de información, de los sistemas de gestión de bases de datos y de las herramientas para la automatización del desarrollo de los productos software.
6. Generar propuestas innovadoras y competitivas en la investigación y en la actividad profesional.
7. Poder realizar la implementación total o parcial del producto software y poder diseñar los métodos de verificación y validación.
8. Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.
9. Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
10. Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
11. Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
12. Saber analizar, diseñar e implementar sistemas basados en computadores en general, utilizando técnicas y métodos que aseguren su eficacia y eficiencia.
13. Saber aportar soluciones científicamente válidas y tecnológicamente actuales.
14. Tener conocimientos de hardware y sistemas de comunicaciones.
15. Trabajar en equipo.
16. Utilizar eficazmente bibliografía y recursos electrónicos para obtener información.

## Contenido

Primera parte.

1. Recordatorio de programación estructurada y modular aplicada al C.

1.1. Recordatorio básico

1.1.1. Estructuras y tipos de datos

1.1.2. Estructuras de secuencia, control e iteración

1.1.3. Procedimientos y funciones

1.2. Estructuras dinámicas

1.2.1 Organización de punteros y direcciones de memoria

1.2.2. Estructuras básicas: pilas, listas y colas.

1.2.3. Árboles

1.2.4. Matrices dinámicas

2. Recursividad.

2.1. Fundamentos de la recursividad

2.2. Algoritmos de búsquedade soluciones mediante árboles:

2.2.1. Amplitud prioritaria y profundidad prioritaria.

2.2.2. Backtraking.

Segunda parte

3.Programación orientada a objetos

3.1. Conceptos generales y características.

3.2. Clases y objetos

3.3. Análisis y diseño orientado a objetos.

4. Java

4.1. Características del lenguage.

4.2 Clases, bucles, funciones y estructuras del lenguage

5. Python en Programación Orientada a Objetos

**A menos que las restricciones impuestas por las autoridades sanitarias obliguen a una priorizacion o reduccion de estos contenidos.*

## Metodología

Clases teóricas, de problemas y prácticas:

Se presentarán los conceptos de la asignatura. Se hará énfasis en la interpretación de los resultados y en la relación entre estos conceptos y sus aplicaciones. Se presentarán ejemplos que permitan a los alumnos abordar de forma autónoma la resolución de problemas.

Se discutirá la resolución de los problemas propuestos.

La base de la asignatura será absolutamente práctica. Esto implica que las clases se hacen totalmente en ordenador. Los conceptos teóricos serán explicados en cada sesión y serán complementados con ejercicios propuestos por el profesor, que se deberán programar directamente en el ordenador bajo su supervisión.

Tutorías:

El profesor supervisará y evaluará el trabajo de los estudiantes. Estos plantearán las dudas que les hayan surgido en la resolución de losejercicios planteados.

En las primeras sesiones de tutoría el profesor recogerá información sobre la formación de cada estudiante y la asesorará sobre cómo corregir posibles carencias. En el resto de sesiones de tutoría se resolverán dudas y se discutirá la resolución de los problemas.

Actividades autónomas:

Los ejercicios y prácticas deberán ser resueltos de manera individual. No tiene ningún sentido, en esta asignatura, el formar grupos de prácticas de más de un alumno para resolver las prácticas.

**La metodologia docente propuesta puede experimentar alguna modificacion en funcion de las restricciones a la presencialidad que impongan las autoridades sanitarias.*

Nota: se reservarán 15 minutos de una clase dentro del calendario establecido por el centro o por la titulación para que el alumnado rellene las encuestas de evaluación de la actuación del profesorado y de evaluación de la asignatura o módulo.

## Actividades

| Título | Horas | ECTS | Resultados de aprendizaje |
|---|---|---|---|
| Tipo: Dirigidas | | | |
| Clases teóricas, de problemas y prácticas | 45 | 1,8 | 7 |
| Tipo: Autónomas | | | |
| Estudio personal y desarrollo de prácticas | 97 | 3,88 | 7 |

## Evaluación

Módulo de teoría y problemas (peso 33%).

Exámenes parcial y final, haciendo que el alumno responda los conceptos teóricos a partir de problemas de programación que se plantean.

Para aprobar la asignatura se debe obtener un mínimo de 2.5 puntos en la parte de los exámenes. El examen parcial tiene un peso del 20% y el final del 80% de la nota de los exámenes.

Módulo de prácticas (peso 66%)
Las dos prácticas de la asignatura contarán cada una como un 33% de la nota final.

Exámenes

Las Matrículas de Honor no se concederán a los estudiantes que opten por hacer el examen de recuperación.

Se aplicará la calificación "No evaluable" a los estudiantes que no se presenten al examen.

El examen de recuperación se aplica sólo a los estudiantes que no han aprobado la asignatura.

Evaluación única

Programació Avançada no tiene evaluación única.

**La evaluacion propuesta puede experimentar alguna modificacion en funcion de las restricciones a la presencialidad que impongan las autoridades sanitarias.*

## Actividades de evaluación continuada

| Título | Peso | Horas | ECTS | Resultados de aprendizaje |
|---|---|---|---|---|
| Entrega de prácticas | 66% | 0 | 0 | 1, 2, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 |
| Examen | 33% | 4 | 0,16 | 3, 4, 5, 8, 11 |
| Examen de recuperación | 33% | 4 | 0,16 | 3, 4, 5, 8, 11 |

## Bibliografía


C

**C/C++. Curso de programación. 4ª Edición.**

Fco. Javier Ceballos Sierra;Fco. Javier Ceballos Sierra

eBook | 2015

JAVA

- **Learning Java : An Introduction to Real-World Programming with Java**

Marc Loy;Patrick Niemeyer;Daniel Leuck;Marc Loy;Patrick Niemeyer;Daniel Leu...

eBook | 2020

- **Java Cookbook : Problems and Solutions for Java Developers**

Ian F. Darwin;Ian F. Darwin

eBook | 2020

- **Foundational Java : KeyElements and Practical Programming**

David Parsons;David Parsons

eBook | 2020

PYTHON

- **Python : an introduction to programming** / James Parker

Parker, James, autor

Llibre en línia | 2017

- **High Performance Python : Practical Performant Programming for Humans**

Micha Gorelick;Ian Ozsvald;Micha Gorelick;Ian Ozsvald

eBook | 2020

- **Python : An Introduction to Programming**

Parker, James R.;Parker, James R.

eBook | 2021

Guido van Rossum, "El Tutorial de Python" (Una introducción informal a Pyhon),

- Python 2: http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf
- Python 3: http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf

- Mark Lutz, "Learning Python", Ed. O'Reilly

- Raúl González Duque, "Python para todos", http://mundogeek.net/tutorial-python/

-"Python tutorial", https://www.tutorialspoint.com/python/

Otros libros

- "The C++ Programming Language", B.Stroustrup, 3ªed.,Addison-Wesley, 1997.

- "Programación en C/C++", M.Alfonseca, A.Sierra, Anaya-Multimedia, 1999.

- "El lenguaje de programación Java", K.Arnold, J.Gosling, Addison-Wesley, 1997

- "Programación en JAVA 2", J.Zukowski, Anaya-Multimedia, 1999

- "Descubre Java 1.2", M.Morgan, Prentice Hall, 1999.

- "Java2: The Complete Reference", P.Naughton, H.Schildt, Osborne/McGraw-Hill, 2000

- "Core Java" Volúmenes I y II, C.S.Horstmann, G.Cornell, Sun Microsystems Press.

## Software

https://www.codeblocks.org/downloads/binaries/

https://www.eclipse.org/downloads/

https://code.visualstudio.com/download