UAB
Universitat Autònoma de Barcelona

# Fundamentos de Programación

Código: 104337
Créditos ECTS: 6

**2024/2025**

| Titulación | Tipo | Curso |
|---|---|---|
| 2503758 Ingeniería de Datos | FB | 1 |

## Contacto

Nombre: Roberto Benavente Vidal

Correo electrónico: robert.benavente@uab.cat

## Equipo docente

Ivan Erill Sagales

## Idiomas de los grupos

Puede consultar esta información al final del documento.

## Prerrequisitos

Dado que la asignatura es de introducción, se asumirá que no se posee ningún tipo de conocimiento previo sobre la materia.

A pesar de ello es recomendable tener conocimientos a nivel de usuario de algún tipo de plataforma (windows, mac o linux).

En las actividades presenciales del aula es imprescindible disponer de un ordenador portátil para realizar las tareas de la asignatura.

## Objetivos y contextualización

Esta asignatura tiene un carácter general e introductorio a la programación. Se profundizará en el estudio de los aspectos metodológicos de la programación y el aprendizaje de un lenguaje de alto nivel. Por tanto, los objetivos generales que se proponen para la asignatura son los siguientes:

- Entender el ciclo de vida del software: análisis del problema (entender lo que se nos pide), diseño (proponer una solución al problema), implementación (codificación en un lenguaje de programación de la solución elegida), y prueba (realización de un test de manera sistemática para asegurar la corrección de la solución implementada).
- Dotar al estudiante de la capacidad de diseño de algoritmos para la resolución de problemas, introduciendo de manera progresiva y sistemática una metodología rigurosa y estructurada de programación, basada fundamentalmente en la técnica del diseño descendente de algoritmos.
- Introducir al estudiante en un lenguaje de programación real. Se pretende que el estudiante perciba la diferencia entre la flexibilidad de la notación pseudo-algorítmica que se emplea en los primeros temas y

la estricta sintaxis de un lenguaje de programación real, en sus aspectos léxicos (palabras válidas del lenguaje), sintácticos (reglas para combinarlas) y semánticos (significado de las mismas).
- Habituar al estudiante a desarrollar programas siguiendo unas normas de estilo tendentes a conseguir programas de calidad. Dentro de estas normas de estilo engloban aquellas que facilitan la comprensión del código, como pueden ser el uso de comentarios, la indentación del código, la utilización de nombres adecuados para los tipos de datos, etc.

## Competencias

- Diseñar soluciones algorítmicas eficientes para problemas computacionales, implementarlas en forma de desarrollo de software robustos, estructurados y fáciles de mantener, y verificar su validez.
- Evaluar de manera crítica el trabajo realizado.
- Planificar y gestionar el tiempo y los recursos disponibles.
- 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.

## Resultados de aprendizaje

1. Aplicar estrategias de depuración, prueba y corrección de programas.
2. Aplicar los principios básicos de la estructura y la programación de los computadores.
3. Desarrollar programas con un buen estilo de programación y bien documentados.
4. Evaluar de manera crítica el trabajo realizado.
5. Implementar en un lenguaje de programación problemas algorítmicos de dificultad media.
6. Planificar y gestionar el tiempo y los recursos disponibles.
7. 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.

## Contenido

Tema 1: Introducción a la informática

Historia. Estructura funcional del ordenador. Programas / instrucciones. Niveles conceptuales del ordenador.

Tema 2: Resolución de problemas: introducción a la algorítmica y la programación.

Introducción a la resolución de problemas. Concepto de algoritmo. Fases en el desarrollo de algoritmos. La programación como disciplina de ingeniería. Ciclo de vida del software. Elementos básicos de un algoritmo. Herramientas para la representación de algoritmos. Lenguajes de programación. Clasificación. Traductores de lenguajes: Compiladores e intérpretes.

Tema 3: Conceptos básicos y estructuras de control

Definición de variables y constantes. Tipo de datos fundamentales. Estructura Secuencial. Estructuras de selección o condicionales. Estructuras iterativas o repetitivas.

Tema 4: Estructuras de datos

Arrays unidimensionales: strings, tuplas i listas. Búsquedas i recorridos.

Tema 5: Subprogramas

El concepto de subprograma como abstracción de operaciones. Localidad, anidamiento, ámbito y visibilidad. Definición de funciones y procedimientos. Llamadas a funciones y procedimientos. Diseño modular descendente.

Tema 6: Ficheros

Definiciones básicas. Entrada / salida de datos en ficheros. Tipo de accesos a ficheros.

Tema 7: Prevención y detección de errores

Tipos de errores. Excepciones y *assert's*. Programación preventiva. Depuración de programas.

Tema 8: Introducción a la programación orientada a objetos

Clases y objetos. Atributos y métodos. Encapsulación. Definición de clases.

Tema 9: Tipos de datos complejos

Listas: iteradores, generadores, paradigma funcional y list comprehensions. Conjuntos. Diccionarios.

## Actividades formativas y Metodología

| Título | Horas | ECTS | Resultados de aprendizaje |
|---|---|---|---|
| Tipo: Dirigidas | | | |
| Clases de problemas | 26 | 1,04 | 1, 2, 3, 5, 4, 7 |
| Clases de prácticas | 11 | 0,44 | 1, 2, 3, 5, 4, 6, 7 |
| Clases de teoría | 10 | 0,4 | 1, 2, 3, 5, 4, 7 |
| Tipo: Supervisadas | | | |
| Proyectos de programación | 30 | 1,2 | 1, 2, 3, 5, 4, 6, 7 |
| Tipo: Autónomas | | | |
| Preparación clases y estudio personal | 20 | 0,8 | 1, 2, 3, 5, 6, 7 |
| Resolución de problemas auto-evaluables | 48 | 1,92 | 1, 2, 3, 5, 4, 7 |

La gestión de la docencia de la asignatura se hará a través del gestor documental Caronte (http://caronte.uab.cat/), que servirá para poder ver los materiales, gestionar los grupos de prácticas, hacer las entregas correspondientes, ver las notas, comunicarse con el profesorado, etc. Para poder utilizarlo es necesario hacer los siguientes pasos:

1. Darse de alta dando el nombre, NIU, y una foto en formato JPG. Si ya se ha dado de alta por alguna otra asignatura, no es necesario volver a hacer; puede ir al siguiente paso.
2. Inscribirse en el tipo de docencia "Programació en Python", dando como código de asignatura el proporcionado el primer día de clase.

La docencia de la asignatura se desarrolla en dos sesiones de dos horas cada una en un aula informática. En estas sesiones, se podrán diferenciar las siguientes actividades docentes:

MD1 Exposición de contenidos en clase: Presentación de los contenidos teóricos a trabajar en la clase. Algunos de estos contenidos deberán haberse preparado antes de la clase a partir de la lectura de textos, visualización de vídeos, búsqueda de información, etc. Los contenidos presentados estarán directamente relacionados con los problemas propuestos en otras actividades docentes, de forma que serán la base sobre la que se desarrollarán otras actividades del curso.

MD2 Clases participativas: Resolución en común del conjunto de problemas propuestos a los estudiantes. Todos los temas irán acompañados de una relación de problemas que el estudiante debe resolver. En este sentido, ya medida que el estudiante vaya progresando en la profundidad de sus conocimientos, estos problemas serán poco a poco más complejos, permitiendo de esta manera apreciar claramente las ventajas de utilizar las herramientas metodológicas impartidas durante el curso. Todos los problemas desarrollados en clase y otros que se puedan proponer se podrán encontrar en la plataforma Caronte, y serán auto-evaluables. Estas actividades deben permitir al estudiante profundizar en la comprensión y personalizar el conocimiento. El hecho de que sean auto-evaluables permite adecuar el ritmo de consolidación y reflexionar sobre el propio aprendizaje.

MD3 Tutorías: Horas de libre disposición por el estudiante para consultas sobre aspectos en los que necesite ayuda adicional por parte del profesorado.

MD4 Realización de proyectos cortos: Realización de pequeños proyectos de prácticas para profundizar en aspectos aplicados de la teoría. Estos proyectos se resolverán en pequeños grupos en los que cada miembro del grupo deberá hacer una parte y ponerla en común con el resto del grupo para tener la solución final.

MD5 Actividades de evaluación: Ver apartado de evaluación de esta guía docente.

Competencias transversales

Las competencias transversales T06 y T08 se trabajan y se evaluaron a lo largo del curso en las siguientes actividades:

| Competencia | MD1 | MD2 | MD4 |
|---|---|---|---|
| T06 - Evaluar de manera crítica el trabajo realizado | Trabajo personal de preparación de los materiales<br><br>Evaluación: Examen Teórico (Tp y Tf ) | Evaluación: Resolución de Problemas (P) de auto-evaluación con Caronte | Evaluación: Proyectos de prácticas (PPg y PPi) |
| T8 - Planificar el tiempo y los recursos disponibles | | | El enunciado de la práctica dará indicaciones de cómo realizar la planificación.<br><br>Evaluación: Proyectos de prácticas grupal (PPg) |

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.

## Evaluación

### Actividades de evaluación continuada

| Título | Peso | Horas | ECTS | Resultados de aprendizaje |
|---|---|---|---|---|
| 1. Examen teórico parcial | 15% | 2 | 0,08 | 1, 2, 3, 5, 4, 7 |
| 2. Examen teòric final | 35% | 2 | 0,08 | 1, 2, 3, 5, 4, 7 |
| 3. Entrega de problemas | 20% | 0 | 0 | 1, 2, 3, 5 |
| 4. Proyectos de programación (evaluación grupal) | 20% | 0,5 | 0,02 | 1, 2, 3, 5, 6 |
| 5. Examen Proyectos de programación (evaluación individual) | 10% | 0,5 | 0,02 | 1, 2, 3, 5, 4 |

Proceso y actividades de evaluación programadas

A lo largo del curso se realizarán las siguientes actividades de evaluación::

| Actividad | Fecha | Recuperación | Porcentaje | Nota mínima |
|---|---|---|---|---|
| Examen Teórico Parcial (Tp) Individual | Consultar Planificación | Consultar Planificación | 15% | No |
| Examen Teórico Final (Tf) Individual | Consultar Planificación | Consultar Planificación | 35% | Tf> = 5 |
| Entrega de Problemas (P) individual | Cada semana | No | 20% | No |
| Proyectos de Programación (PPg) Grupal | Consultar Planificación | Consultar Planificación | 20% | PPg> = 5 |
| Examen Proyectos de Programación (PPi) ndividual | Consultar Planificación | Consultar Planificación | 10% | PPi> = 5 |

Para poder aprobar la asignatura, el resultado de la suma ponderada de las actividades de evaluación debe ser mayor o igual que 5, y habrá que sacar una nota mínima de 5 en las actividades Tf, PPg y PPi.

Programación de actividades de evaluación

Las fechas de evaluación y entrega de trabajos se publicarán en el gestor documental Caronte y pueden estar sujetos a cambios de programación por motivos de adaptación a posibles incidencias. Siempre se informará por el gestor documental sobre estos cambios ya que se entiende que esta es la plataforma habitual de intercambio de información entre profesores y estudiantes.

Procedimiento de recuperación

El estudiante puede presentarse a la recuperación siempre que se haya presentado a un conjunto de actividades que represente al menos dos terceras partes de la calificación total de la asignatura.

Los exámenes teóricos (Tp y Tf) se podrán recuperar en un único examen de recuperación en las fechas fijadas por la coordinación del Grado. El examen de recuperación tendrá un porcentaje del 50% sobre la nota final. La nota del examen de recuperación, en caso de realizarse, sustituirá a las notas de los exámenes parcial (Tp) y final (Tf) realizados durante el curso.

De acuerdo con la coordinación del Grado y la dirección de la Escuela de Ingeniería, la actividad Entregas de Problemas (P) Individual no se podrá recuperar.

Procedimiento de revisión de las calificaciones

Los estudiantes tendrán derecho a la revisión de los exámenes teóricos (Tp y Tf). El lugar, fecha y hora de revisión se publicará el día que sean publicadas las notas. Si el estudiante no se presenta en esta revisión no se revisará posteriormente esta actividad. Sólo en casos justificados se podrá hacer una revisión a posteriori de la fecha fijada y siempre hasta un máximo de 7 días naturales.

De las notas de la práctica de laboratorio (PPg y PPi) no se realizará revisión para que la evaluación se realiza ante los estudiantes.

Evaluación única

Esta asignatura ofrece sistema de evaluación única.

La evaluación única de la asignatura constará de las siguientes actividades de evaluación:

| Actividad | Fecha | Recuperación | Porcentaje | Nota mínima |
|---|---|---|---|---|
| Examen Teórico Final (Tf) Individual | Consultar Planificació | ConsultarPlanificación | 50% | Tf >= 5 |
| Proyectos de Programación (PPg) Grupal | Fecha del examen teórico final | Fecha del examen de recuperación | 30% | PPg >= 5 |
| Examen Proyectos de Programación (PPi) individual | Fecha del examen teórico final | Fecha del examen de recuperación | 20% | PPi >= 5 |

Para poder aprobar la asignatura, el resultado de la suma ponderada de las actividades de evaluación debe ser mayor o igual que 5, y habrá que sacar una nota mínima de 5 en las actividades Tf, PPg y PPi.

Se aplicará el mismo sistema de recuperación que para la evaluación continuada.

La revisión de la calificación final sigue el mismo procedimiento que para la evaluación continuada.

Calificaciones

No Evaluable (NA): Cualquier alumno que entregue una práctica o una evaluación programada tendrá nota. Sólo se considerará no evaluable en el caso de no entregar ninguna actividad evaluable.

Nota Final: Sale de la suma ponderada según los criterios expuestos en el apartado de actividades de evaluación. Si en alguna actividad no se alcanza la nota mínima la nota saldrá de la siguiente fórmula:

Mínimo (suma ponderada de notas, 4.5)

Matrículas de honor: Otorgar una calificación de matrícula de honor es decisión del profesorado responsable de la asignatura. La normativa de la UAB indica que las MH sólo se podrán conceder a estudiantes que hayan obtenido una calificación final igual o superior a 9.00. Se puede otorgar hasta un 5% de MH del total de estudiantes matriculados. En caso de que el número de estudiantes con nota superior o igual a 9 sean más del 5% del total de matriculados, se priorizó los estudiantes según las siguientesreglas (en orden):

1. Estudiantes que hayan conseguido la nota igual o superior a 9 haciendo menos recuperaciones de actividades (ya sea por haber suspendido la actividad o porque se ha ido a subir nota).
2. Estudiantes con más notas iguales o superiores a 9 en las actividades Tp, Tf y PPi.
3. Estudiantes con mejor nota global.

Irregularidades por parte del estudiante, copia y plagio

Sin perjuicio de otras medidas disciplinarias que estimen oportunas, se calificarán con un cero las irregularidades cometidas por el estudiante que puedan conducir a una variación de la calificación de un acto de evaluación. Por lo tanto, la copia, el plagio, el engaño, dejar copiar, etc. en cualquiera de las actividades de evaluación implicará suspender con un cero. Las actividades de evaluación calificadas de esta forma y por este procedimiento no serán recuperables. Si es necesario superar cualquiera de estas actividades de evaluación para aprobar la asignatura, esta asignatura quedará suspendida directamente, sin oportunidad de recuperarla en el mismo curso. En este caso, la nota numérica del expediente será el valor menor entre 3.0 y la media ponderada de las notas.

Evaluación de los estudiantes repetidores

Los estudiantes repetidores que el curso anterior al actual hayan aprobado la asignatura "Programación avanzada" y hayan entregado correctamente los ejercicios de repaso propuestos en aquella asignatura, podrán superar la asignatura presentando unos ejercicios de síntesi del curso. La nota final en este caso serà igual a 5. Los estudiantes que se quieran acoger a esta vía, deberán comunicarlo la primera semana de clase al profesorado de la asignatura.

El resto de estudiantes repetidores deben cursar la asignatura por completo. No se mantendrá ninguna nota de cursos anteriores.

## Bibliografía

J. Guttag. *Introduction to Computation and Programming Using Python: With Application to Understanding Data. Second Edition*. MIT Press. ISBN-10: 9780262529624

S. Chazallet *Python 3. Los fundamentos del lenguaje*. Eni, ISBN-10: 2409006140

E. Matthes. *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*. No Starch Press ISBN-10: 1593276036

M. Myers. *A Smarter Way to Learn Python: Learn it faster. Remember it longer*. Createspace Independent Pub ISBN-10: 1974431479

A. Prieto, A. Lloris, J.C. Torres. *Introducción a la Informática*. Mc Graw-Hill ISBN-10: 8448146247

A. Prieto, B. Prieto. *Conceptos de Informática*. Mc Graw-Hill, Schaum ISBN-10: 8448198573

L. Joyanes Aguilar. *Fundamentos de Programación: Algoritmos, Estructuras de Datos y Objetos*. Mc. Graw-Hill. ISBN-10:8448161114

## Software

Se utilizará la última versión del paquete Anaconda que incluye Python 3.x y el editor Spyder ( https://www.anaconda.com/products/individual).

## Lista de idiomas

| Nombre | Grupo | Idioma | Semestre | Turno |
|---|---|---|---|---|
| (PAUL) Prácticas de aula | 81 | Catalán | primer cuatrimestre | manaña-mixto |
| (PAUL) Prácticas de aula | 82 | Catalán | primer cuatrimestre | manaña-mixto |