

Fundamentos de Programación

Código: 104337

Créditos ECTS: 6

2025/2026

Titulación	Tipo	Curso
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.

Sin embargo, es recomendable tener conocimientos a nivel de usuario de algún tipo de plataforma (Windows, Mac o Linux).

En las actividades presenciales en el 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 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. 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 prácticas	10	0,4	2, 1, 4, 3, 5, 6, 7
Clases de problemas	30	1,2	2, 1, 4, 3, 5, 7
Clases de teoría	10	0,4	2, 1, 4, 3, 5, 7
Tipo: Supervisadas			
Proyectos de programación	25	1	2, 1, 4, 3, 5, 6, 7
Tipo: Autónomas			
Preparación clases y estudio personal	30	1,2	2, 1, 3, 5, 6, 7
Resolución de problemas auto-evaluables	41	1,64	2, 1, 4, 3, 5, 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.

En el desarrollo de la asignatura se seguirá la metodología de aula inversa. Esto significa que el alumnado tendrá que trabajar algunos conocimientos antes de las sesiones presenciales en las que principalmente se realizarán actividades prácticas con el objetivo de tratar el máximo de casos posibles para entender la complejidad de lo que implica plantear una solución a un problema de programación.

Antes de clase. Los conceptos básicos se trabajan antes de las sesiones presenciales a partir de los materiales proporcionados con una semana de antelación en la plataforma Caronte. El material a trabajar será documentación escrita, contenido multimedia y actividades de aprendizaje.

Cada estudiante debe prepararse las clases antes de asistir, revisando los materiales tantas veces como sea necesario para asimilar los conceptos. Como orientación, se recomienda dedicar semanalmente un número de horas equivalente al tiempo de clase aproximadamente entre 2 y 3 horas semanalmente. Para cada tema, se propondrá un cuestionario que permitirá evaluar el grado de consecución de los contenidos trabajados

La clase. El objetivo de las sesiones presenciales es pasar de la información al conocimiento. Pondremos el énfasis en crear conocimientos aplicables al análisis y solución de problemas concretos. La estructura de las clases será:

- Revisión de los conceptos básicos del tema a tratar (15 min). El objetivo es resolver las dudas que puedan existir sobre aspectos teóricos no entendidos y hacer seguimiento de la adquisición de los conceptos básicos.
- Solución de problemas (resto del tiempo de clase). El objetivo es experimentar realizando una tarea práctica que permita aplicar los conceptos aprendidos para resolver retos y así compartir y crear conocimiento.

Después de clase. El objetivo es consolidar el conocimiento. En la plataforma Caronte se podrán encontrar un conjunto de problemas evaluables de autocorrección que deben permitir al alumnado profundizar en la comprensión de los temas y personalizar su conocimiento. El hecho de que sea autoevaluable permite adecuar el ritmo de consolidación a cada estudiante y que se produzca una reflexión sobre el propio aprendizaje.

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
Actividades de clase evaluables	10%	0	0	2, 1, 4, 3, 5
Entrega de problemas	15%	0	0	2, 1, 3, 5
Examen final	45%	2	0,08	2, 1, 4, 3, 5, 7
Examen intrasemestral	15%	2	0,08	2, 1, 4, 3, 5, 7
Prácticas	15%	0	0	2, 1, 3, 5, 6

Actividades de evaluación programadas

La asignatura consta de las siguientes actividades de evaluación:

Actividad	Tipo	Peso	Nota mínima	Recuperable
Examen intrasemestral	individual	15%	No	Sí
Examen final	individual	45%	5	Sí
Prácticas	grupal	15%	5	Sí
Resolución de problemas	individual	15%	No	Sí
Actividades de clase evaluables	individual	10%	No	No

Para poder aprobar la asignatura, será necesario tener una nota mínima de 5 en el examen final y las prácticas.

Programación de actividades de evaluación

El calendario de las actividades de evaluación se dará el primer día de la asignatura y se hará público a través de Caronte, y en el apartado de exámenes de la web del grado.

Proceso de recuperación

El estudiante puede presentarse a la recuperación siempre que se haya presentado a un conjunto de actividades que represente un mínimo de dos terceras partes de la calificación total de la asignatura. En particular si el estudiante no se ha presentado en el examen final no podrá realizar el examen de recuperación

El examen final podrá recuperarse en la fecha fijada por la coordinación del grado. En caso de realizar el examen de recuperación, la nota de este examen también podrá sustituir a la nota del examen intrasemestral si la nota de la recuperación es superior a la nota del intrasemestral.

Las prácticas se podrán recuperar en una segunda entrega en la fecha fijada en el calendario de las actividades de evaluación que se dará el primer día de la asignatura y se hará público a través de Caronte.

Las entregas de problemas se podrán recuperar en cualquier momento del curso, entregándolas de nuevo y evaluándolas con el autocorrector.

De acuerdo con la coordinación del Grado y la dirección de la Escuela de Ingeniería, las actividades de clase evaluables no se podrán recuperar.

Procedimiento de revisión de las calificaciones

Para los exámenes (intrasemestral, final y recuperación), se indicará un lugar, fecha y hora de revisión en la que el alumnado podrá revisar la actividad con el profesorado. En este contexto, se podrán realizar reclamaciones sobre la nota de la actividad, que serán evaluadas por el profesorado responsable de la asignatura. Si el estudiante no se presenta a esta revisión, no se revisará posteriormente esta actividad.

Para el resto de actividades, si se detecta algún error en la corrección, se podrá pedir que se revise en cualquier momento, hasta la fecha de la revisión del examen de recuperación.

Calificaciones

No Evaluable (NA): Un estudiante se considerará no evaluable (NA) si no se ha presentado en ninguna de las siguientes actividades de evaluación: exámenes de la asignatura (intrasemestral, final y recuperación), prácticas y actividades evaluables en clase.

Nota final: Se calcula con la suma ponderada de las actividades de evaluación según criterios expuestos en el apartado de actividades de evaluación.

En caso de no superar la asignatura debido a que alguna de las actividades de evaluación no llega a la nota mínima requerida, la nota numérica del expediente será el menor valor entre 4.5 y la media ponderada de las notas. Con las excepciones de que se otorgará la calificación de "No Evaluable" al alumnado que no participe en ninguna de las actividades de evaluación, y de que la nota numérica del expediente será el valor menor entre 3.0 y la media ponderada de las notas en caso de que el o la estudiante haya cometido irregularidades en un acto de evaluación

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 podrán concederse a estudiantes que hayan obtenido una calificación final igual o superior a 9.00. Puede otorgarse 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 supere el 5% del total de matriculados, se aplicarán los siguientes criterios de priorización en el orden enumerado a continuación:

1. Estudiantes que hayan realizado menos actividades de recuperación.
2. Estudiantes con mayor nota final.
3. Estudiantes con nota más alta en el examen final
4. Estudiantes con nota más alta en el examen intrasemestral

Irregularidades cometidas por el alumnado

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, y de acuerdo con la normativa académica vigente, las irregularidades cometidas por un/a estudiante que puedan conducir a una variación de la calificación en una actividad evaluable se calificarán con un cero (0). 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 quedarás suspendida directamente, sin oportunidad de recuperarla en el mismo curso. Estas irregularidades incluyen, entre otras:

- la copia total o parcial de una práctica, informe, o cualquier otra actividad de evaluación;
- dejar copiar;
- presentar un trabajo de grupo no realizado íntegramente por los miembros del grupo (aplicado a todos los miembros, no sólo a los que no han trabajado);
- uso no autorizado de la IA (p. ej., Copilot, ChatGPT o equivalentes) para resolver ejercicios, prácticas y/o cualquier otra actividad evaluable;
- presentar como propios materiales elaborados por un tercero, aunque sean traducciones o adaptaciones, y por lo general trabajos con elementos no originales y exclusivos del estudiante;
- tener dispositivos de comunicación (como teléfonos móviles, smart watches, bolígrafos con cámara, etc.) accesibles durante las pruebas de evaluación teórico-prácticas individuales (exámenes);
- hablar con compañeros durante las pruebas de evaluación teórico-prácticas individuales (exámenes);
- copiar o intentar copiar de otros alumnos durante las pruebas de evaluación teórico-prácticas (exámenes);
- usar o intentar utilizar escritos relacionados con la materia durante la realización de las pruebas de evaluación teórico-prácticas (exámenes), cuando éstos no hayan sido explícitamente permitidos.

En ediciones futuras de esta asignatura, al alumnado que haya cometido irregularidades en un acto de evaluación no se le convalidará ninguna de las actividades de evaluación realizadas ni podrá acogerse a ninguna normativa específica de evaluación de los estudiantes repetidores.

Ús de la IA

Para esta asignatura, se permite el uso de tecnologías de Inteligencia Artificial (IA) exclusivamente en labores de apoyo, como la búsqueda de información, la interpretación y explicación de programas, resolución de dudas, etc. En ningún caso se permite el uso de la IA para la realización de las tareas de programación que

deben entregarse como parte de la evaluación. Cualquier trabajo que incluya fragmentos generados con IA será considerado una falta de honestidad académica y comportará la aplicación de la normativa en lo referente a irregularidades por parte del estudiante.

Evaluación de los estudiantes repetidores

Si el curso anterior el actual se superó la asignatura "Programación Avanzada" y se aprobaron los ejercicios de repaso propuestos en aquella asignatura, se podrá optar por una evaluación que consistirá en una prueba de síntesis, sin tener que realizar todas las actividades de evaluación. La nota final en ese caso será de 5.

Para poder optar a esta evaluación diferenciada, la persona interesada debe pedirlo al profesorado mediante correo electrónico (Robert.Benavente@uab.cat o Ivan.Erill@uab.cat) a más tardar 15 días después del inicio de las clases.

Evaluación única

Esta asignatura ofrece sistema de evaluación única.

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

Actividad	Tipo	Peso	Nota mínima	Recuperable
Examen final	individual	60%	5	Sí
Prácticas	grupal	15%	5	Sí
Resolución de problemas	individual	15%	No	Sí
Cuestionario de prácticas	individual	10%	5	Sí

Para poder aprobar la asignatura, será necesario tener una nota mínima de 5 en el examen final, las prácticas y el cuestionario de prácticas.

Se aplicará el mismo sistema de recuperación que por la evaluación continua, excepto en los puntos que hacen referencia al examen intrasemestral. El cuestionario de prácticas se podrá recuperar el día del examen de recuperación.

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

Se aplicará el mismo criterio de no evaluable que para la evaluación continua.

Bibliografía

J. L. Gahete Díaz, C. M. Vallejo Fernández, A. Fernández-Pacheco Sánchez-Migallón. *Fundamentos de programación con Python*. McGraw Hill, 1ª edición, 2024. ISBN-10: 9788448645274

J. Guttag. *Introduction to Computation and Programming Using Python: With Application to Understanding Data*. MIT Press, 2nd edition, 2016. ISBN-10: 9780262529624

S. Chazallet. *Python 3: Los fundamentos del lenguaje*. Ediciones ENI, 2^a edición, 2016. ISBN-10: 2409006140

E. Matthes. *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*. No Starch Press, 1st edition, 2016. ISBN-10: 1593276036 (Existe versión online:
<https://ebookcentral.proquest.com/lib/uab/detail.action?docID=4503145>)

M. Myers. *A Smarter Way to Learn Python: Learn it faster. Remember it longer*. Createspace Independent Pub, 2017. ISBN-10: 1974431479

L. Joyanes Aguilar. *Fundamentos de Programación: Algoritmos, Estructuras de Datos y Objetos*. 5^a Edición, McGraw-Hill, 2020. ISBN-10:8448161114 (Existe versión online:
https://www-ingebok-com.eu1.proxy.openathens.net/ib/NPcd/IB_Escritorio_Visualizar?cod_primaria=1000193&l=es)

Software

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

Grupos e idiomas de la asignatura

La información proporcionada es provisional hasta el 30 de noviembre de 2025. A partir de esta fecha, podrá consultar el idioma de cada grupo a través de este [enlace](#). Para acceder a la información, será necesario introducir el CÓDIGO de la asignatura

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