

Iniciación a la Programación

Código: 104383
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2503740 Matemática Computacional y Analítica de Datos	FB	1	1

Contacto

Nombre: Francesc Xavier Roca Marva

Correo electrónico: xavier.roca@uab.cat

Idiomas de los grupos

Puede consultarlo 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.

Equipo docente

Roberto Benavente Vidal

Prerrequisitos

Dado que la asignatura es de introducción asumirá que no se posee ningún tipo de conocimiento previo sobre la materia. Es responsabilidad de la propia asignatura proporcionar a los alumnos un medio para adquirir los conocimientos descritos en el apartado de contenidos de la asignatura.

A pesar de ello es recomendable:

- tener conocimientos a nivel de usuario de algún tipo de plataforma (windows, mac o linux)
- disponer de un ordenador, preferiblemente un portátil

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: analizar el problema (entender lo que se nos pide), diseño (proponer una solución al problema), implementación (codificación en un lenguaje de programación de la solución elegida), 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, tanto en sus aspectos léxicos (palabras válidas del lenguaje), sintácticos (reglas para combinarlas) y semánticos (significado de las mismas).
- Habituarse 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

- Aplicar conocimientos básicos sobre la estructura, el uso y la programación de ordenadores, sistemas operativos y programas informáticos para solucionar problemas de distintos ámbitos.
- Diseñar, desarrollar y evaluar soluciones algorítmicas eficientes para problemas computacionales de acuerdo con los requisitos establecidos.
- Evaluar de manera crítica y con criterios de calidad el trabajo realizado.
- 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 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.
- Trabajar cooperativamente en un contexto multidisciplinar asumiendo y respetando el rol de los diferentes miembros del equipo.
- Utilizar eficazmente bibliografía y recursos electrónicos para obtener información.

Resultados de aprendizaje

1. Conocer los conceptos básicos de la estructura y la programación de los computadores.
2. Evaluar de manera crítica y con criterios de calidad el trabajo realizado.
3. Evaluar y analizar la complejidad computacional de las soluciones algorítmicas para poder desarrollar e implementar aquella que garantice el mejor rendimiento.
4. 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.
5. Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
6. 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.
7. Seleccionar y utilizar las estructuras algorítmicas y de representación de los datos apropiadas para la resolución de un problema.
8. Trabajar cooperativamente en un contexto multidisciplinar asumiendo y respetando el rol de los diferentes miembros del equipo.
9. Utilizar eficazmente bibliografía y recursos electrónicos para obtener información.
10. Verificar y asegurar el funcionamiento correcto de una solución algorítmica de acuerdo con los requisitos del problema a resolver.

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 y listas. Búsquedas y recorridos.

Tema 5: Subprogramas

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

Tema 6: Control de errores

Tipos de errores. Excepciones y asserts. Programación preventiva. Depuración de programas.

Tema 7: Archivos

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

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.

Metodología

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 los profesores, etc. Para poder utilizarlo es necesario hacer los siguientes pasos:

1. Darse de alta como usuario dando el nombre, NIU, y una foto carnet 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 "Iniciación a la Programación", dando como código de asignatura "IniProg" (sin las comillas).

La docencia de la asignatura se desarrolla en dos sesiones de dos horas cada una. En estas sesiones, se podrán diferenciar cinco tipos de actividades docentes:

MD1 Exposición de contenidos en clase - Clase magistral: La estructura típica de una clase magistral de este tipo será la siguiente: en primer lugar se hará una introducción donde se presentarán brevemente los objetivos

de la exposición y los contenidos a tratar. Con el fin de proporcionar el contexto adecuado, la presentación se hará referencia al material expuesto en clases precedentes, por lo que se clarifique la posición de estos contenidos dentro del marco general de la asignatura.

MD2 Clases participativas: Resolución en común del conjunto de problemas propuesto a los estudiantes. Todos los temas irán acompañados de una relación de problemas que el alumno debe intentar resolver. En este sentido, ya medida que el alumno 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 para el estudiante para consultas sobre aspectos que necesite ayuda adicional por parte del profesorado

MD4 Realización de proyectos cortos: Realización de prácticas, problemas más amplios, para profundizar en aspectos aplicados de la teoría. La parte práctica de cada tema quedará completada con al menos una sesión práctica, donde los estudiantes deberán resolver problemas concretos de cierta complejidad. Estos proyectos se resolverán en pequeños grupos. Cada miembro del grupo deberá hacer una parte y ponerla en común con el resto para tener la solución final.

MD5 Actividades de evaluación: Ver apartado evaluación de la guía docente

Competencias transversales

Las competencias transversales T01, T02 y T04 se trabajan y se evalúan a lo largo del curso en las siguientes actividades:

Competencia	MD1	MD2	MD4
T01 - Evaluar de manera crítica y con criterios calidad el trabajo realizado	Trabajo personal de preparación de los materiales Evaluación: Examen teórico (Tp y Tf)	Realización de los problemas propuestos Evaluación: Resolución problemas (P) de auto-evaluación con el Caronte	Evaluación: Practica de Laboratorio (PLG) Grupal
T02 - Trabajar cooperativamente en un contexto multidisciplinar asumiendo y respetando el rol de los diferentes miembros del equipo.			Se proporcionarán materiales de cómo trabajar en equipo de manera eficaz. Evaluación: Practica de Laboratorio (PLG) Grupal
T04 - Utilizar eficazmente la bibliografía y los recursos electrónicos para obtener información	Trabajo personal de preparación de los materiales Evaluación: Examen Teórico (Tp y Tf)		Trabajo: Búsqueda de información para resolver la práctica

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 Presenciales / Teoría	10	0,4	2, 3, 1, 4, 5, 6, 7, 9, 10
Clases Prácticas	10	0,4	2, 3, 1, 5, 7, 8, 9, 10
Clases de Problemas	26	1,04	2, 3, 1, 6, 7, 10
Tipo: Supervisadas			
Programming project	17,5	0,7	2, 3, 1, 5, 6, 7, 8, 9, 10
Tipo: Autónomas			
Preparación clases y Estudio personal	20	0,8	2, 3, 1, 4, 5, 6, 7, 9, 10
Resolució de casos pràctics (grupal)	22,5	0,9	2, 3, 1, 6, 7, 8, 9, 10
Resolución de problemas auto-evaluables (individual)	40	1,6	2, 3, 1, 6, 7, 9, 10

Evaluación

Proceso y actividades de evaluación programadas

Se realizan las siguientes actividades evaluables:

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

Resolución de Problemas (P) individual	Cada semana	No	20%	No
Práctica de Laboratorio(PLG) Grupal	Consultar Planificación	Consultar Planificación	20%	PLG > = 5
Examen Práctica de Laboratorio (PLI) individual	Consultar Planificación	Consultar Planificación	10%	pLI > = 5

La nota de cada final de la asignatura es la suma ponderada de las actividades de evaluación con su porcentaje correspondiente. El resultado de la suma ponderada debe ser ≥ 5 por considerar aprobada la asignatura.

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 la actividad Resolución 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 (PLG y PLI) 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	Consultar Planificación	50%	Tf \geq 5

Práctica de Laboratorio (PLg) Grupal	Fecha del examen teórico final	Fecha del examen de recuperación	30%	PLg >= 5
Examen de la Práctica de Laboratorio (PLi) individual	Fecha del examen teórico final	Fecha del examen de recuperación	20%	PLi >= 5

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

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 (aplicar suma ponderada y 4,5)

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.

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 siguientes reglas (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 superiores a 9 en las actividades 1 (Tp), 2 (Tf), y 5 (PLi).
3. Estudiantes con mejor nota global.

Irregularidades por parte del/la estudiante, copia y plagio

Sin perjuicio de otras medidas disciplinarias que se 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 estudiantes repetidores

Los estudiantes repetidores deben cursar la asignatura por completo. No se guarda ninguna nota de cursos anteriores.

Actividades de evaluación continuada

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Examen Teórico Final (Tf) Individual	35%	2	0,08	2, 3, 1, 4, 5, 6, 7, 9, 10

Examen Teórico Parcial (Tp) Individual	15%	1	0,04	2, 3, 1, 4, 5, 6, 7, 9, 10
Practica de Laboratorio (PLG) Grupal	20%	0,5	0,02	2, 3, 1, 5, 6, 7, 8, 10
Practica de Laboratorio (PLI) individual	10%	0,5	0,02	2, 3, 1, 5, 7, 9, 10
Resolución Problemas (P) Individual	20%	0	0	2, 3, 1, 6, 7, 9, 10

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>)