Universitat Autònoma de Barcelona

**2022/2023**

# Fundamentos de Programación II

Código: 106554
Créditos ECTS: 6

| Titulación | Tipo | Curso | Semestre |
|---|---|---|---|
| 2504392 Inteligencia Artificial / Artificial Intelligence | FB | 1 | 2 |

## Contacto

Nombre: Anna Bàrbara Sikora
Correo electrónico: anna.sikora@uab.cat

## Uso de idiomas

Lengua vehicular mayoritaria: inglés (eng)
Algún grupo íntegramente en inglés: Sí
Algún grupo íntegramente en catalán: No
Algún grupo íntegramente en español: No

## Prerequisitos

El curso no tiene requisitos previos oficiales. Sin embargo, se basa en Fundamentos de Programación I. Por lo tanto, se recomienda que los/las estudiantes tengan un buen conocimiento de los conceptos básicos de programación y Python.

## Objetivos y contextualización

Este curso es la continuación de Fundamentos de Programación I, y los objetivos generales son:

- Comprender diferentes estructuras de datos, como pilas, colas y listas.
- Comprender diferentes estructuras de datos más complejas, como tablas hash, árboles y gráficos.
- Comprender la memoria dinámica y los punteros.
- Comprender diferentes algoritmos para ordenar y buscar, y las limitaciones de esos algoritmos.
- Ser capaz de analizar un problema complejo, diseñar una solución eficiente, implementarla, calcular su coste y probarla.
- Introducir el concepto de recursividad y su aplicación, así como ser capaz de analizar la complejidad de los algoritmos recursivos.

## Competencias

- Analizar y resolver problemas de forma efectiva, generando propuestas innovadoras y creativas para alcanzar los objetivos.
- Conocer y utilizar de forma eficiente las técnicas y herramientas de representación, manipulación, análisis y gestión de datos a gran escala.
- Desarrollar pensamiento crítico para analizar de forma fundamentada y argumentada alternativas y propuestas tanto propias como ajenas.
- Diseñar, implementar, analizar y validar soluciones algorítmicas eficientes y robustas a problemas computacionales derivados del diseño de sistemas inteligentes.
- Introducir cambios en los métodos y los procesos del ámbito de conocimiento para dar respuestas innovadoras a las necesidades y demandas de la sociedad.

## Resultados de aprendizaje

1. Analizar y resolver problemas de forma efectiva, generando propuestas innovadoras y creativas para alcanzar los objetivos.
2. Aplicar estrategias de depuración, prueba y corrección para verificar el funcionamiento correcto de una solución algorítmica de acuerdo con los requisitos del problema a resolver.
3. Aplicar los principios de la programación orientada a objetos en el desarrollo de programas.
4. Comprender las estructuras de datos comunes y los algoritmos que las utilizan y manipulan.
5. Conocimiento de las limitaciones y ventajas de diferentes representaciones de datos y capacidad para elegir las estructuras más eficientes que faciliten el posterior análisis y explotación de los datos.
6. Desarrollar pensamiento crítico para analizar de forma fundamentada y argumentada alternativas y propuestas tanto propias como ajenas.
7. Desarrollar programas con un buen estilo de programación y bien documentados.
8. Evaluar y analizar la complejidad computacional de las soluciones algorítmicas para poder desarrollar e implementar aquella que garantice el mejor rendimiento.
9. Implementar soluciones recursivas a problemas de programación.
10. Proponer nuevos métodos o soluciones alternativas fundamentadas.
11. Seleccionar y utilizar las estructuras algorítmicas y de representación de los datos apropiadas para la resolución de un problema.

## Contenido

Estructuras de datos dinámicas:

- Punteros, estructuras de datos dinámicas (listas, pilas, colas).
- Estructuras de datos no lineales. Técnicas de hashing. Matrices hash y listas hash.
- Recursividad.
- Algoritmos de ordenación.
- Algoritmos de búsqueda.
- Grafos.
- Árboles.
- Análisis de la complejidad de los algoritmos.

Programación avanzada de diferentes estructuras de datos dinámicas en lenguaje C.
Programación básica de diferentes estructuras de datos dinámicas en lenguaje C++.
Importación de código C desde Python.
Programación de diferentes estructuras de datos de Python en C.
Lenguajes de programación:

- Lenguaje estructurado C (básico y avanzado)
- Lenguaje orientado a objetos C++ (básico)
- Lenguaje interpretado Python (conocimientos de Fundamentos de Programación 1)

## Metodología

Usaremos la plataforma Caronte (http://caronte.uab.cat/) para compartir materiales, entregar proyectos y otras actividades administrativas.

Para matricularse en Caronte es necesario registrarse (con nombre, NIU y foto) y matricularse a la asignatura (el código de la asignatura se facilitará el primer día de clase).

La metodología docente se centrará principalmente en el trabajo práctico. Se organizarán sesiones presenciales para la discusión de los contenidos teóricos de la asignatura, seguidas de ejercicios y problemas de programación. Más concretamente, los diferentes tipos de actividades docentes son los siguientes:

- Clases teóricas: Explicación en pizarra de la parte teórica de cada tema del programa. La estructura típica de una clase será la siguiente: en primer lugar se realizará una introducción donde se expondrán brevemente los objetivos de la clase y los contenidos a tratar. A continuación, se abordarán los contenidos de un tema, incluyendo exposiciones narrativas, desarrollos formales que aporten fundamentos teóricos, e intercalando ejemplos prácticos, que ilustren la aplicación de los contenidos expuestos. Finalmente, el profesorado

expondrá las conclusiones de los contenidos tratados. Las clases teóricas, a pesar de ser principalmente una explicación por parte del profesorado, serán participativas para los/las estudiantes dándoles la oportunidad de preguntar aquellos puntos que no terminan de entender. A los/las estudiantes se les plantearán constantemente preguntas y problemas para comprobar la buena comprensión del tema expuesto. Habrá 2 exámenes para evaluar esta parte de la asignatura.

- Clases de resolución de problemas: Resolución conjunta de problemas. Todos los temas irán acompañados de un conjunto de problemas que el/la estudiante deberá intentar resolver. En este sentido, y a medida que el/la estudiante progrese y profundice en sus conocimientos, estos problemas serán progresivamente más complejos. Las clases serán el foro natural en el que se pueda discutir en común el desarrollo del trabajo práctico, aportando los conocimientos que le faltan al alumnado para llevarlo adelante. Las clases de resolución de problemas tienen como misión servir de puente entre las clases teóricas y las clases de laboratorio, lo que fomentará la capacidad de análisis y síntesis, el desarrollo del razonamiento crítico, y que capacitará al/la estudiante en la resolución de problemas. Durante el curso, los/las estudiantes (en grupos de 2 personas) realizarán varios ejercicios cortos y básicos que serán evaluados. Cada grupo resolverá estos ejercicios básicos y realizará la entrega en un plazo corto correspondiente. Este ejercicio busca a que los/las estudiantes puedan resolver teóricamente los conceptos trabajados en las clases teóricas para luego poder aplicarlos al caso práctico de laboratorio. Los ejercicios estarán disponibles en Caronte.

- Proyectos prácticos: los/las estudiantes trabajarán en parejas y recorrerán los diferentes pasos del ciclo de vida del desarrollo de software para implementar la solución a un problema. Estos proyectos prácticos completarán las asignaturas teóricas, donde el/la estudiante desarrollará varios programas y tendrá que intentar resolver un problema concreto propuesto. Estos proyectos deberán ser entregados en las fechas indicadas.

Este enfoque de trabajo está dirigido a promover el aprendizaje activo y desarrollar competencias de organización y planificación, comunicación oral y escrita, trabajo en equipo y razonamiento crítico. Se valorará especialmente la calidad de los ejercicios realizados, de su presentación y de su funcionamiento.

Todos los detalles del funcionamiento, el calendario y las fechas de evaluación se informarán al principio del curso y aparecerá en la plataforma de comunicación.

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 | | | |
| Proyectos prácticos | 18 | 0,72 | 1, 3, 2, 8, 4, 5, 6, 7, 9, 10, 11 |
| Realización de ejercicios básicos | 5 | 0,2 | 3, 2, 4, 9, 11 |
| Soluciones de problemas | 10 | 0,4 | 3, 2, 4, 6, 9, 11 |
| Teoría | 10 | 0,4 | 3, 4, 5, 9, 11 |
| Tipo: Autónomas | | | |
| Estudios autónomos | 56 | 2,24 | 1, 3, 2, 8, 4, 5, 6, 7, 9, 10, 11 |
| Preparación de problemes i ejercicios prácticos | 10 | 0,4 | 3, 2, 8, 4, 5, 9, 11 |
| Preparación proyectos | 30 | 1,2 | 1, 3, 2, 8, 4, 5, 6, 7, 9, 10, 11 |

## Evaluación

El objetivo del proceso de evaluación es comprobar que el alumno ha alcanzado los conocimientos y habilidades definidos en los objetivos de la asignatura, así como las competencias asociadas.

Se evaluarán tres tipos de actividades de forma independiente, y la suma ponderada de ellas dará la nota final. Estas tres actividades son:

- Teoría (T)
- Solución de los proyectos prácticos (PP)
- Resolución de ejercicios básicos (EXER).

La parte de Teoría (T) se evaluará con dos exámenes parciales individuales a lo largo del curso. La nota final de Teoría resultará de la suma ponderada de los dos exámenes (0,5 * Control 1 + 0,5 * Control 2). Habrá una segunda oportunidad para recuperar esa parte el día de junio que tenemos asignado. Las partes que no hayan sido superadas en los exámenes parciales de teoría podrán recuperarse por separado. La nota mínima para aprobar esta parte de Teoría es >= 5. En la recuperación sólo se podrán recuperar los exámenes suspendidos, no pudiendo subir la nota obtenida (en el caso de aprobar la parte de Teoría). La nota máxima que se puede obtener en recuperación es un 7.

La parte de proyectos prácticos (PP) se evaluará en grupos de 2 personas. Hay dos entregas. La nota final saldrá de la suma ponderada de las dos entregas (0,5 * Entrega 1 + 0,5 * Entrega 2). Para aprobar el PP la nota mínima tendrá que ser >= 5. Solo hay una oportunidad para aprobar esta parte (esta parte no se puede recuperar).

Los ejercicios prácticos (EXER) se realizarán en grupos de 2 personas y consistirán en trabajar problemas de programación muy concretos, breves y relacionados con los que se encuentran en las prácticas de laboratorio. Queremos que cada estudiante estudie la solución de un conjuntode problemas específicos de forma aislada en el caso más general de la práctica. El valor de estos ejercicios es el 20% de la nota final y dada su naturaleza y finalidad no son recuperables.

La nota final de la asignatura será la suma ponderada de las notas de cada una de las tres actividades: 40% de Teoría, 40% de trabajos prácticos y 20% de Resolución de ejercicios básicos. El resultado tendrá que ser >= 5.

En caso de que un/a estudiante no apruebe la asignatura por no alcanzar la nota mínima en alguna de las partes obligatorias (Teoría o Proyectos Prácticos), aunque la media ponderada sea igual o superior a 5, la nota final de la asignatura será de 4,5 .

En el caso de que la media no llegue a 5, la nota oficial será la nota media obtenida numéricamente.

Si el/la estudiante entrega alguna actividad, se entenderá que participa en la materia y será evaluado. Si no entrega ninguna actividad, entonces puede considerarse No evaluable.

La concesión de un título de matrícula d'honor (MH) es una decisión del responsable de la materia. La normativa de la UAB indica que sólo se puede conceder MH a los/las estudiantes que hayan obtenido una nota final igual o superior a 9,00. Se puede otorgar hasta el 5% de MH del total de estudiantes matriculados.

Las fechas de evaluación continua y entrega de trabajos estarán publicadas en la plataforma de comunicación y podrán estar sujetas a posibles cambios para adaptarse a posibles incidencias; Caronte siempre informará sobre estos cambios ya que se entiende que es el mecanismo habitual de intercambio de información entre profesores y alumnat.

Para cada actividad de evaluación se indicará un lugar, fecha y hora de revisión enla que el alumno podrá repasar la actividad con el profesorado. En este contexto, se pueden realizar reclamaciones sobre la calificación de la actividad, que será evaluada por el profesorado de la materia. Si el/la estudiante no se somete a esta revisión, esta actividad no será revisada más adelante.

RESUMEN

Si ((T> = 5) i (PP> = 5) entonces

NF = 0,40 * T + 0,2 * EXER + 0,40 * PP

Si (NF> = 5) entonces APROBADO

de lo contrario SUSPENSO

de lo contrario SUSPENSO

Alumnos repetidores: los/las estudiantes repetidores que hayan superado prácticas de laboratorio en ediciones anteriores de la asignatura NO PUEDEN solicitar la convalidación de esta parte de la asignatura. Todas las actividades de evaluación deben realizarse en las mismas condiciones que el resto de alumnado.

Nota sobre el plagio:

Sin perjuicio de otras medidas disciplinarias que se consideren oportunas, y de acuerdo con la normativa académica vigente, las irregularidades cometidas por un estudiante que puedan dar lugar a una variación de la calificación en una actividad evaluable serán calificadas con cero (0). Las actividades de evaluación calificadas de esta forma y por este procedimiento no serán recuperables. Si para aprobar la asignatura es necesario superar alguna de estas actividades de evaluación, se suspenderá directamente esta asignatura, sin posibilidad de recuperarla en el mismo curso. Estas irregularidades incluyen, entre otras:

- la copia total o parcial de un ejercicio de laboratorio, informe o cualquier otra actividad de evaluación;
- dejar que otro estudiante copie;
- presentar un trabajo grupal no realizado en su totalidad por los integrantes del grupo (aplicado a todos los integrantesy no solo a los que no han trabajado);
- presentar como propios materiales elaborados por un tercero, aunque se trate de traducciones o adaptaciones, y en general obras con elementos no originales y exclusivos del alumno;
- disponerde dispositivos de comunicación (como teléfonos móviles, relojes inteligentes, bolígrafos con cámara, etc.) accesibles durante las pruebas de evaluación teórico-prácticas (exámenes individuales);
- hablar con los compañeros durante las pruebas teórico-prácticas individuales (exámenes);
- copiar o intentar copiar de otros estudiantes durante las pruebas de evaluación teórico-prácticas (exámenes);
- utilizar o intentar utilizar material escrito relacionado con la materia durante las pruebas de evaluación teórico-prácticas (exámenes), cuando éstas no hayan sido expresamente permitidas.
- Si no superas la asignatura debido a que alguna de las actividades de evaluación no alcanza la nota mínima exigida, la nota oficial numérica será el valor más bajo entre 4,5 y la media ponderada de las notas. Con las salvedades de que se otorgará la calificación de "No Valorativo" a los alumnos que no participen en ninguna de las actividades de evaluación, y que la nota oficial numérica será el valor más bajo entre 3,0 y la nota media ponderada en caso de que el alumno haya cometió irregularidades en un acto de evaluación (y por lo tanto el sujeto no puede ser aprobado por compensación). En próximas ediciones de esta asignatura, el alumno que haya cometido irregularidades en un acto de evaluación no será convalidado de ninguna de las actividades de evaluación realizadas.

En resumen: copiar, dejar copiar o plagiar(o intentar) en cualquiera de las actividadesde evaluación dará lugar a un SUSPENSO, no indemnizable y sin convalidaciones de partes de la asignatura en cursos posteriores.

## Actividades de evaluación

| Título | Peso | Horas | ECTS | Resultados de aprendizaje |
|---|---|---|---|---|
| 1er Control parcial individual | 20% | 2 | 0,08 | 8, 4, 11 |
| 2do Control parcial individual | 20% | 2 | 0,08 | 3, 8, 4, 9, 11 |
| Proyectos prácticos | 40% | 2 | 0,08 | 1, 3, 2, 8, 4, 5, 6, 7, 9, 10, 11 |
| Realización de ejercicios básicos | 20% | 5 | 0,2 | 3, 2, 8, 4, 5, 9, 11 |

## Bibliografía

Mark Allen Weiss: Data Structures and Algorithm Analysis in C (2nd Edition). Pearson, 1996.

Mark Allen Weiss: Data Structures and Algorithm Analysis in C++. 4 th edition. Pearson, 2014.

Kernighan, Brian; Ritchie, Dennis M., The C Programming Language (2nd edition). Englewood Cliffs, NJ: Prentice Hall. 1988.

K.N. King: C Programming: A Modern Approach. 2nd edition. W.W. Norton & Company. 2008

http://www.cplusplus.com/ : The C++ Resources Network

## Software

Microsoft Visual C++. Visual Studio. Available for UAB students at https://msdnaa.uab.cat/

Queue manager (SLURM)

Remote connection to the laboratory (e.g. MobaTex, Eclipse, Visual Studio, etc.). MobaXTerm for Windows [https://mobaxterm.mobatek.net/] or Visual Studio Code for all platforms [https://code.visualstudio.com/]