



Universitat de Lleida

GUÍA DOCENTE **PROGRAMACIÓN I**

Coordinación: ARGELICH ROMA, JOSEP

Año académico 2021-22

Información general de la asignatura

Denominación	PROGRAMACIÓN I			
Código	102000			
Semestre de impartición	1R Q(SEMESTRE) EVALUACIÓN CONTINUADA			
Carácter	Grado/Máster	Curso	Carácter	Modalidad
	Doble titulación: Grado en Ingeniería Informática y Grado en Administración y Dirección de Empresas	1	TRONCAL	Presencial
	Grado en Ingeniería Informática	1	TRONCAL	Presencial
Número de créditos de la asignatura (ECTS)	6			
Tipo de actividad, créditos y grupos	Tipo de actividad	PRALAB	TEORIA	
	Número de créditos	3	3	
	Número de grupos	5	2	
Coordinación	ARGELICH ROMA, JOSEP			
Departamento/s	INFORMATICA E INGENIERIA INDUSTRIAL			
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	GEI: 6 ECTS = 25x6 = 150 horas de trabajo 40% --> 60 horas presenciales 60% --> 90 horas de trabajo autónomo del estudiante			
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.			
Idioma/es de impartición	Catalán			

Profesor/a (es/as)	Dirección electrónica\nprofesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
ALSINET BERNADÓ, MARIA TERESA	teresa.alsinet@udl.cat	3	
ARGELICH ROMA, JOSEP	josep.argelich@udl.cat	12	
ARIÑO CAGIGÓS, RAUL		3	
PLANES CID, JORDI	jordi.planes@udl.cat	3	

Información complementaria de la asignatura

Para abordar la asignatura es recomendable mostrar interés en plantear y analizar problemas reales, para los que se deberá buscar y desarrollar una solución tecnológica que los resuelva. Es también recomendable mostrar capacidad de análisis, de razonamiento lógico, trabajo metódico y organizado, y capacidad crítica.

Asignatura que se imparte durante el 1r semestre del 1r curso de la titulación.

Los conocimientos y competencias adquiridas en esta asignatura serán de utilidad en asignaturas posteriores con contenidos de programación, estructuras de datos y algorítmica.

Objetivos académicos de la asignatura

Los objetivos de aprendizaje de la asignatura son analizar y diseñar algoritmos para, posteriormente, implementar en un lenguaje de programación compilable. Concretamente, el lenguaje imperativo escogido es ANSI C/C++ y los problemas a resolver son principalmente de tratamiento de secuencias.

En concreto los objetivos de aprendizaje del estudiante son los siguientes:

- Diseñar e implementar estructuras algorítmicas adecuadas para resolver las diferentes tipologías de problemas.
- Diseñar e implementar estructuras de datos adecuadas para representar la información propia de cada problema.
- Diseñar e implementar algoritmos iterativos.
- Identificar la tipología del problema y aplica la estrategia algorítmica adecuada.
- Diseñar e implementar algoritmos para resolver problemas complejos de forma estructurada.
- Diseñar e implementar soluciones algoritmos básicas utilizando la técnica de diseño descendente.
- Utilizar un entorno de desarrollo de programas basado en un lenguaje de programación de alto nivel.

Competencias

Competencias Estratégicas de la UdL:

- EPS1. Capacidad de resolución de problemas y elaboración y defensa de argumentos dentro de su área de estudios.
- EPS5. Capacidad para la abstracción y el razonamiento crítico, lógico y matemático.
- EPS9. Capacidad de trabajo en equipo, tanto unidisciplinar como multidisciplinar.
- EPS12. Tener motivación por la calidad y la mejora continua.

Competencias específicas en el grado en Ingeniería Informática:

- GII-FB3. Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.
- GII-FB4. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
- GII-FB5. Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.
- GII-CRI7. Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
- GII-CRI9. Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.

Contenidos fundamentales de la asignatura

Introducción: Procesos, algoritmos y programas.

Tema 1. Estructuras algorítmicas básicas

- 1.1 Constantes, variables, tipos elementales, y expresiones válidas
- 1.2 La asignación, la composición secuencial, la composición alternativa y la composición iterativa
- 1.3 Entorno de programación

Tema 2. Diseño de programas iterativos

2.1 Acceso secuencial

- 2.1.1 Esquemas algorítmicos de tratamiento de secuencias
- 2.1.2 Esquemas algorítmicos de búsqueda en secuencias

2.2 Acceso directo. Las tablas

- 2.2.1 Tratamiento secuencial de tablas
- 2.2.2 Tratamiento directo de tablas
- 2.2.3 Algoritmos de ordenación clásicos

Tema 3. Tratamiento de datos no elementales

- 3.1. Acciones y funciones
- 3.2 Mecanismos de paso de parámetros
- 3.3 Diseño descendente de algoritmos

Ejes metodológicos de la asignatura

Cada semana el estudiante asiste a 2 horas presenciales con Grupo Grande y 2 horas presenciales con Grupo Medio. Las sesiones con Grupo Medio imparten en el laboratorio.

Grupos Grandes: Clases Teoría y Problemas (3 créditos)

- *Parte teórica:* clases soportadas con transparencias y / o apuntes.
- *Parte de aplicación práctica:* se trabaja siempre con problemas y ejercicios de programación. Se dispone de una colección de problemas, de los que se van proporcionando soluciones a lo largo del cuatrimestre.

Grupos Medios: Clases Laboratorio (3 créditos)

- Clases dirigidas y seguimiento personalizado para grupos de prácticas. Se dispone de una colección de problemas, de los que se van proporcionando soluciones a lo largo del cuatrimestre.
- Uso de los compiladores y herramientas de edición.
- Trabajo continuado alrededor del enunciado de las dos prácticas obligatorias.

Trabajo Autónomo (no presencial):

- La práctica se completará en horas No Presenciales. En las sesiones de Grupo Medio se da apoyo a las prácticas obligatorias que debe desarrollar el estudiante a lo largo de la asignatura de forma autónoma.
- Se recomienda que el alumno resuelva por cuenta propia los problemas de la colección de problemas, a fin de practicar y obtener feedback por parte del profesor.

Plan de desarrollo de la asignatura

Sem	Descripción	Actividad Presencial GG	Actividad Presencial GM	Trabajo autónomo
1	Presentación Introducción	Presentación Asignatura Introducción: Procesos, algoritmos y programas	Utilización de un entorno de programación	Resolución de ejercicios
2	Estructuras algorítmicas básicas	T1 Constantes, variables, tipos elementals y expresiones válidas	Ejercicios de programación	Resolución de ejercicios
3	Estructuras algorítmicas básicas	T1 La asignación, la composición secuencial y la composició alternativa	Ejercicios de programación	Resolución de ejercicios
4	Estructuras algorítmicas básicas	T1 La composició iterativa	Ejercicios de programación	Resolución de ejercicios
5	Diseño de programas iterativos	T2 Tratamiento secuencial	Práctica 1: Descripción de la primera práctica obligatoria	Resolución de la Práctica 1 en grupos

6	Diseño de programas iterativos	T2 Búsqueda en secuencias	Ejercicios de programación Soporte a la Práctica 1	Resolución de ejercicios Resolución de la Práctica 1 en grupos
7	Diseño de programas iterativos	T2 Acceso directo. El tipo tabla	Ejercicios de programación Soporte a la Práctica 1	Resolución de ejercicios Resolución de la Práctica 1 en grupos
8	Diseño de programas iterativos	T2 Exercicis d'accés directe amb taules: tractament i cerca.	Ejercicios de programación Soporte a la Práctica 1	Resolución de ejercicios Resolución de la Práctica 1 en grupos
9		1ª Parcial	Entrega de la Práctica 1	Estudio Resolución de la Práctica 1 en grupos
10	Diseño de programas iterativos	T2 Tablas multidimensionales	Algoritmos de ordenación clásicos	Resolución de ejercicios
11	Tratamiento de datos no elementales	T3: Acciones y funciones	Ejercicios de programación	Resolución de ejercicios
12	Tratamiento de datos no elementales	T3: Mecanismos de paso de parámetros	Práctica 2: Descripción de la segunda práctica obligatoria	Resolución de la Práctica 2 en grupos
13	Tratamiento de datos no elementales	T3: Diseño descendente de algoritmos	Ejercicios de programación Soporte a la Práctica 2	Resolución de ejercicios Resolución de la Práctica 2 en grupos
14	Tratamiento de datos no elementales	T3 Ejercicios de diseño descendente	Ejercicios de programación Soporte a la Práctica 2	Resolución de ejercicios Resolución de la Práctica 2 en grupos

15	Tratamiento de datos no elementales	T3 Ejercicios de diseño descendente	Ejercicios de programación Soporte a la Práctica 2	Resolución de ejercicios Resolución de la Práctica 2 en grupos
16		2º Parcial		Estudiar Resolución de la Práctica 2 en grupos
17		2º Parcial	Entrega de la Práctica 2	Estudiar Resolución de la Práctica 2 en grupos
18				
19		Recuperación	Recuperación de las prácticas	Estudiar Resolución de la Práctica 2 en grupos

Sistema de evaluación

Actividades de evaluación

Acrónimo	Actividad de evaluación	Ponderación	Nota Mínima	En grupo	Obligatoria	Recuperable
<i>P1</i>	<i>Examen 1º Parcial</i>	25%	4	NO	SI	SI
<i>P2</i>	<i>Examen 2º Parcial</i>	35,00%	4	NO	SI	SI
<i>PRA1</i>	<i>Práctica 1</i>	15,00%	4	SI (<=2)	SI	SI
<i>PRA2</i>	<i>Práctica 2</i>	25,00%	4	SI (<=2)	SI	SI

Para aprobar la asignatura es necesario obtener la nota mínima de 4 tanto en las pruebas escritas como en las prácticas. Además, la nota final deberá de ser >=5.

$$\text{Nota Final} = 0,25 \cdot P1 + 0,35 \cdot P2 + 0,15 \cdot PRA1 + 0,25 \cdot PRA2$$

Observaciones:

- Si la nota obtenida en la prueba escrita P2 es ≥ 4 , entonces la nota obtenida podrá actuar como recuperación / mejora de la primera prueba escrita P1, el peso de la cual es del 25%.
- Si la nota obtenida en la prueba escrita P2 es < 4 , entonces el estudiante puede optar a mejorar / recuperar el 60% que representan las pruebas escritas. La recuperación será una única prueba escrita que será evaluada sobre 10 puntos. La nota obtenida sustituirá la nota de las dos pruebas escritas del curso. La nota de la prueba representará el 60% de la nota final y para aprobar la asignatura la nota de la prueba de recuperación deberá ser ≥ 4 .
- Si la nota obtenida en la práctica *PRA2* es ≥ 4 , entonces la nota obtenida podrá actuar como mejora/recuperación de la primera práctica *PRA1*, el peso de la cual es del 15%.

- En caso que la práctica *PRA2* no se supere (nota < 4), la práctica podrá ser recuperada en el periodo de recuperación (semana número 20).

Bibliografía y recursos de información

Algorítmica

- J. Castro, F. Cucker, X. Messeguer, A. Rubio, L. Solano and B.Valles. *Curs de Programació*. McGraw-Hill, 1992.
- G. Brassard and P. Bratley. *Fundamentos de Algoritmia*. Prentice Hall, 1997.
- L. Joyanes. *Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos*. McGraw-Hill, 2003.

El lenguaje ANSI C y C++

- H.M. Deitel and P.J. Deitel. *Como Programar en C/C++*. Prentice-Hall, segunda edición, 2002.
- B. Stroustrup. *Programming -- Principles and Practice Using C++*. Addison Wesley, 2008.
- L. Joyanes. *Programación en C++*. McGraw-Hill, 2006.