

Fundamentos de Programación

Código: 102154
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2501232 Empresa y Tecnología	FB	1	2

Contacto

Nombre: Roger Malet Munté

Correo electrónico: Roger.Malet@uab.cat

Uso de idiomas

Lengua vehicular mayoritaria: catalán (cat)

Algún grupo íntegramente en inglés: No

Algún grupo íntegramente en catalán: Sí

Algún grupo íntegramente en español: No

Prerequisitos

Para poder seguir correctamente esta asignatura se necesitan unos conocimientos previos de algoritmica y resolución de problemas. Estos conocimientos se alcanzan en la asignatura "Introducción a la Resolución de Problemas y Algoritmos" (102.151), que se da el primer semestre y que da los mecanismos necesarios para saber afrontar los problemas que deberán resolverse mediante la programación en un lenguaje de alto nivel.

Objetivos y contextualización

Los objetivos de esta asignatura son los de introducir los conceptos básicos de la programación de aplicaciones informáticas para resolver problemas que se pueden dar en el ámbito empresarial. La idea de este curso es poder implementar algoritmos en un lenguaje de programación para poder generar soluciones finales que resuelvan problemas concretos y automatizar ciertos procesos informáticos que se dan en el entorno de las empresas. Además, entender el proceso de generación de aplicaciones debe permitir una mayor comprensión de los procesos informáticos más complejas que se dan en este mismo ámbito.

Así, los objetivos formativos de la asignatura se pueden resumir en:

Entender y utilizar las estructuras básicas de la programación estructurada
Implementar algoritmos en un lenguaje de programación de alto nivel

Competencias

- Capacidad de análisis y de síntesis, de organizar, de planificar, de resolver problemas y tomar decisiones.
- Demostrar la motivación por la calidad en los objetivos y en el desarrollo de su trabajo.
- Encontrar soluciones algorítmicas y utilizar las herramientas de programación adecuadas para su implementación en el entorno de una organización.
- Trabajar en equipo, compartiendo los conocimientos y sabiéndolos comunicar al resto del equipo y la organización.

Resultados de aprendizaje

1. Capacidad de análisis y de síntesis, de organizar, de planificar, de resolver problemas y tomar decisiones.
2. Codificar el algoritmo, utilizando las estructuras de programación estructurada.
3. Demostrar la motivación por la calidad en los objetivos y en el desarrollo de su trabajo.
4. Plantear el algoritmo más eficiente para resolver el problema.
5. Proponer soluciones basadas en programación orientada a objetos.
6. Trabajar en equipo, compartiendo los conocimientos y sabiéndolos comunicar al resto del equipo y la organización.
7. Utilizar las estructuras básicas de programación estructurada.

Contenido

Tema 1: Introducción

- Objetivo:

Introducir la forma en que los ordenadores procesan la información, el papel de los lenguajes de programación y el proceso de desarrollar aplicaciones.

- Contenido:

- 1.1. Ordenadores y algoritmos: Procesamiento de la información
- 1.2. Lenguajes de programación y programas
- 1.3. Desarrollo de software con el ordenador
- 1.4. Almacenamiento de la información en el ordenador

Tema 2: Iniciación a la programación en C

- Objetivo:

Primer contacto con un lenguaje de programación, y sus características principales.

- Contenido:

- 2.1. Mi primer programa
- 2.2. Elementos del lenguaje de programación
- 2.3. Los datos el lenguaje de programación
- 2.4. Las variables
- 2.5. Operando con datos
- 2.6. La entrada / salida
- 2.7. Ejemplos

Tema 3: Estructuras de control

- Objetivo:

Introducción a las estructuras que permiten a un programa tomar decisiones y aplicar algoritmos básicos.

- Contenido:

- 3.1. Estructura general de un programa: estructuras de control
- 3.2. Estructuras secuencia
- 3.3. Estructuras de selección
- 3.4. Estructuras de iteración
- 3.5. Más sobre el operador incremento / decremento
- 3.6. Ejemplos

Tema 4: Programación estructurada

- Objetivo:

Introducción a las estructuras que permiten organizar el código para que las aplicaciones sean legibles y escalables.

- Contenido:

- 4.1. Programación estructurada y modular
- 4.2. Procedimientos y funciones
- 4.3. Funciones
- 4.4. Procedimientos
- 4.5. Variables y paso de parámetros
- 4.6. La recursividad
- 4.7. Ejemplos

Tema 5: Vectores (arrays): Listas y tablas

- Objetivo:

Estudio de los elementos que permiten crear listas y tablas de elementos en el lenguaje C #.

- Contenido:

- 5.1. Introducción
- 5.2. Vectores (listas)
- 5.3. Cadenas (listas) de caracteres
- 5.4. Matrices
- 5.5. Paso de parámetros
- 5.6. Operando con vectores (listas)
- 5.7. Ejemplos

Tema 6: Estructuras

- Objetivo:

Introducción a los tipos de datos definidos por el programador, que ayudan a crear tipos más complejos que los que define el propio lenguaje C.

- Contenido:

- 6.1. Introducción a las estructuras (registros)
- 6.2. Anidación de estructuras
- 6.3. Paso de estructuras
- 6.4. Ejemplos

Tema 7: Archivos

- Objetivo:

Introducción a los mecanismos que permiten tratar ficheros en C #.

- Contenido:

- 7.1. Introducción
- 7.2. Escritura de archivos
- 7.3. Lectura de ficheros
- 7.4. Funciones y opciones de lectura / escritura
- 7.5. Ficheros de acceso aleatorio
- 7.6. Paso de argumentos por el nombre de programa

Tema 8: Introducción a la programación orientada a objetos (OO)

- Objetivo:

Introducción al paradigma de programación que ve un programa como una serie de objetos o entidades que se relacionan entre ellos.

- Contenido:

8.1. ¿Qué es la programación O.O.?

8.2. Características de la programación O.O.

8.3. Clase e instancia

8.4. Ejemplos

Metodología

La metodología docente estará basada en tres tipos de actividad:

Actividad dirigida: clases teóricas, prácticas y de análisis de problemas

Actividad supervisada: asistencia a tutorías y realización de ejercicios con seguimiento pautado

Actividad autónoma: parte de estudio del alumno y resolución de casos, individualmente o en grupo

Actividades

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Clases de problemas	10	0,4	2, 4, 5, 1, 6, 7
Clases teóricas	30	1,2	2, 4, 5, 1, 7
Sesiones presenciales de laboratorio	10	0,4	2, 4, 5, 1, 6, 7
Tipo: Supervisadas			
Tutorías	15	0,6	2, 4, 5, 1, 7
Tipo: Autónomas			
Preparación y resolución de ejercicios y prácticas	81	3,24	2, 4, 5, 1, 6, 7

Evaluación

La evaluación del curso se hará de forma continua, mediante exámenes parciales, problemas relacionados con las clases de teoría y trabajos específicos de las sesiones de laboratorio. Si no se aprueba durante la evaluación continua, se deberá hacer un examen final para aprobar la asignatura.

Aprobar con evaluación continua

Durante la evaluación continua se harán 2 exámenes parciales, el primero será obligatorio para todos los alumnos, al segundo sólo se podrán presentar los alumnos que hayan sacado un 4 o más en el primer parcial, y quieran poder aprobar en evaluación continua.

El peso de cada uno de los componentes evaluados serán: 25% el primer parcial, 35% el segundo parcial, 30% las notas de prácticas, y un 10% los ejercicios propuestos en clase individualmente o por grupos.

El estudiante aprobará la asignatura si ha obtenido una nota mínima de 5, y ha sacado un mínimo de 5 en las prácticas y un 4 en el segundo parcial.

Aprobar sin evaluación continuada

Un estudiante que suspenda la evaluación continua deberá presentarse al examen final. El peso de cada uno de los componentes evaluados en el cómputo de la nota final es: 15% del primer examen parcial (o la media de los dos si se ha hecho el segundo parcial), 45% del examen final, 30% las notas de prácticas, y un 10% los ejercicios propuestos en clase individualmente o por grupos.

El estudiante superará la asignatura si logra una nota mínima de 5.0 en la media ponderada de las notas obtenidas en las actividades descritas anteriormente.

Los estudiantes que en la evaluación final hayan obtenido una nota entre 4 y 5 tendrán derecho a una re-evaluación, que se hará la última semana del semestre. El estudiante que se presente y supere la prueba con una nota mínima de 5, tendrá la asignatura aprobada con un 5.0.

Un alumno se considerará "no evaluable" en la asignatura cuando no haya participado de ninguna de las actividades de evaluación.

Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Ejercicios propuestos en clase	10%	0	0	2, 4, 5, 1, 6, 7
Entregas de las sesiones de prácticas	30%	0	0	2, 3, 4, 5, 1, 6, 7
Primer examen parcial	25%	2	0,08	2, 4, 5, 1, 7
Segundo examen parcial	35%	2	0,08	2, 4, 7

Bibliografía

"La Biblia de C #", Ferguson, B. Patterson, J. Beres, P. Boutquin, M. Gupta, Anaya Multimedia

"Algoritmos + Estructura de datos = Programas", Niklaus Wirth, Ediciones del Castillo, Madrid - 1986

Referencia online del lenguaje C#: <http://www.csharpya.com.ar/>;
[https://msdn.microsoft.com/en-us/library/aa288436\(v=vs.71\).aspx](https://msdn.microsoft.com/en-us/library/aa288436(v=vs.71).aspx)