

DATOS GENERALES DE LA ASIGNATURA

ASIGNATURA: Fundamentos de Programación

PLAN DE ESTUDIOS: Grado en Ingeniería Informática

GRUPO: 1718-M

CENTRO: Escuela Politécnica Superior

CARÁCTER DE LA ASIGNATURA: Básico

ECTS: 6,0

CURSO: 1º

SEMESTRE: 1º Semestre

IDIOMA EN QUE SE IMPARTE:

Castellano, excepto las asignaturas de idiomas que se impartirán en el idioma correspondiente

DATOS DEL PROFESOR

NOMBRE Y APELLIDOS: Francisco José Palacios Burgos

EMAIL: fjpalacios@uemc.es

TELÉFONO: 983 00 10 00

HORARIO DE TUTORÍAS: Miércoles a las 13:00 horas

BREVE CV:

Doctor en Ciencias Físicas e Ingeniero Técnico en Informática de Sistemas. He desarrollado trabajos de investigación centrados en la simulación por computadora de sistemas físicos a nivel molecular. Dentro de estas líneas de investigación he tenido la oportunidad de trabajar en el desarrollo de aplicaciones software paralelas de alto rendimiento.

Profesionalmente he sido responsable por más de 10 años del diseño e implementación de varias aplicaciones de gestión con arquitectura web en el ámbito universitario, trabajando con variadas tecnologías (HTML, CSS, JS, PHP-Symfony y .NET)

Algunos intereses en el campo de la tecnología incluyen el Software Libre en general, Linux como sistema operativo, el stack LAMP para el desarrollo de aplicaciones y las arquitecturas software orientadas a servicios (SOA).

DATOS ESPECÍFICOS DE LA ASIGNATURA

DESCRIPTOR DE LA ASIGNATURA:

Esta asignatura se centra en los principios de la programación estructurada y modular, que es la base de multitud de lenguajes de programación. También, se ven a modo de introducción las diferentes técnicas de diseño de programas y algoritmos, así como el uso de técnicas de análisis y verificación de algoritmos.

La asignatura se plantea como el medio para que el alumno adquiriera las nociones básicas sobre técnicas de programación. Es importante superar la asignatura para afrontar posteriormente con éxito asignaturas relacionadas con el área de la programación de aplicaciones software.

La importancia de la misma, para todos aquellos Ingenieros en Informática que se vayan a dedicar profesionalmente al área del desarrollo del software, radica en que una buena asimilación de los contenidos fundamentales de esta asignatura es necesaria para poder presentar el día de mañana habilidades claras en dicho campo, ya que en esta asignatura se abordan los contenidos y habilidades relacionados con las aptitudes necesarias para programar en cualquier lenguaje de programación moderno.

CONTENIDOS DE LA ASIGNATURA:

El temario de la asignatura se estructura en temas que se presentan a continuación:

Tema 1. Introducción a la programación

Tratamiento de información y resolución de problemas mediante computadora. Lenguajes de programación: tipos e evolución. Etapas en el ciclo de vida del software.

Tema 2. Elementos de la programación estructurada. Variables, constantes, operadores

Almacenamiento de información: variables, constantes y operadores. Teoremas básicos de la programación estructurada. Estructuras de control. Colecciones simples de datos: arrays.

Tema 3. Algoritmos. Representación y análisis de algoritmos

Noción y estructura de un algoritmo. Técnicas de representación de algoritmos. Documentación de algoritmos. Análisis de la complejidad temporal y espacial de algoritmos. Verificación de algoritmos.

Tema 4. Elementos de la programación modular. Diagrama de estructura

Descomposición modular de algoritmos: funciones y procedimientos. Técnicas de diseño modular de programas. Recursividad. Representación de la estructura modular de programas.

Tema 5. Técnicas de diseño de algoritmos. Algoritmos de Ordenación y Búsqueda

Principales técnicas para el diseño de algoritmos. Algoritmos de ordenación. Algoritmos de búsqueda

Tema 6. Ficheros

Noción y tipos de ficheros. Principales operaciones con ficheros. Ficheros temporales.

PROGRAMA DE PRÁCTICAS

Las prácticas de la asignatura se han concebido para reforzar los conocimientos teóricos de la asignatura y para garantizar las competencias de tipo *Saber hacer*. Hay dos tipos de prácticas distintas:

Prácticas cortas: En cada tema, se realizarán una serie de ejercicios prácticos de programación con el fin de ilustrar los contenidos teóricos impartidos. Para ello, se hará uso tanto de lenguajes descriptivos de tipo pseudocódigo como de algún lenguaje de programación estructurado.

Proyecto de programación: Se planteará un proyecto de programación que englobe todos los contenidos teóricos de la asignatura. Dicho proyecto se entregará y se defenderá al finalizar el semestre.

RECURSOS DE APRENDIZAJE:

Para la docencia de la asignatura se utilizarán diversos recursos electrónicos que se irán dejando en el curso habilitado para tal fin en la plataforma eCampus.

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE A ADQUIRIR POR EL ALUMNO

COMPETENCIAS BÁSICAS:

- CB1. 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

COMPETENCIAS GENERALES:

- CG01. Capacidad de análisis y síntesis
- CG02. Capacidad de organización y planificación
- CG06. Capacidad de gestión de la información

- CG07. Resolución de problemas
- CG14. Razonamiento crítico
- CG16. Aprendizaje autónomo
- CG18. Creatividad
- CG22. Motivación por la calidad
- CG26. Pensamiento lógico

COMPETENCIAS ESPECÍFICAS:

- CE04. Conocer los principios de la programación estructurada y modular, así como las principales técnicas para el análisis y diseño de algoritmos
- CE27. Saber construir programas siguiendo los paradigmas de la programación estructurada, modular y de orientación a objetos

RESULTADOS DE APRENDIZAJE:

El alumno será capaz de:

- Analizar y diseñar algoritmos siguiendo los principios de la programación estructurada y modular.
- Construir programas simples y complejos, atendiendo a criterios de modularidad, reusabilidad y eficiencia.

BIBLIOGRAFÍA Y RECURSOS DE REFERENCIA GENERALES

BIBLIOGRAFÍA BÁSICA:

Luis Joyanes Aguilar, Ignacio Zahonero Martínez (1998): *Estructura de Datos. Algoritmos, abstracción y objetos*. Editorial McGraw-Hill. [\[Link\]](#)

Luis Joyanes Aguilar (2003): *Algoritmos, estructuras de datos y objetos*. Editorial McGraw-Hill, 3ª Edición. [\[Link\]](#)

Ricardo Peña Marí (1998): *Diseño de programas. Formalismo y abstracción*. Editorial Prentice-Hall, 2ª Edición [\[Link\]](#)

Javier Galve, Juan C. González, Ángel Sánchez y J. Ángel Velázquez (1993): *Algorítmica. Diseño y análisis de algoritmos funcionales e imperativos*. Editorial RA-MA, 1ª Edición. [\[Link\]](#)

L. Joyanes (1990): *Problemas de la metodología de la programación*. Editorial McGraw-Hill. [\[Link\]](#)

BIBLIOGRAFÍA COMPLEMENTARIA:

Gregory L. Heileman (1998): *Estructuras de datos, algoritmos y programación orientada a objetos*. Editorial McGraw-Hill, 1ª Edición. [\[Link\]](#)

M. Santos, I. Patiño y R. Carrasco (2005): *Fundamentos de programación*. Editorial Ra-Ma [\[Link\]](#)

PLANIFICACIÓN DEL PROCESO DE ENSEÑANZA-APRENDIZAJE DE LA ASIGNATURA

METODOLOGÍAS:

MÉTODO DIDÁCTICO:

Los contenidos teóricos de la asignatura se impartirán mediante clases presenciales en las que se explicarán los fundamentos de programación que permiten la construcción de programas software que resuelven problemas por computadora.

MÉTODO HEURÍSTICO:

Con el fin de afianzar la comprensión y asimilación de los contenidos prácticos se realizarán una serie de clases prácticas en las que se resolverán pequeños problemas de programación.

Asimismo, en clases de laboratorio el alumno trabajará una serie de problemas propuestos con el fin de resolverlos bien de forma individual o bien de forma grupal, adquiriendo así destreza en el uso de las herramientas y lenguajes de programación reales.

CONSIDERACIONES DE LA PLANIFICACIÓN:

El desarrollo de los contenidos de la asignatura se realizará a través de las clases expositivas teóricas a las que seguirán clases prácticas de problemas. Cuando la materia impartida así lo permita, se programarán ejercicios prácticos de programación que habrá que realizar en el laboratorio. Esta es un esquema (aproximado) de cómo se desarrollará la asignatura:

T = Teoría, P = Problemas, L = Laboratorio

Semana 1. Tema 1 (T)
Semana 2. Tema 1 (T) y Tema 2 (T)
Semana 3. Tema 2 (T) y Tema 2 (P)
Semana 4. Tema 2 (P y L)
Semana 5. Prueba evaluación Temas 1 y 2 (T y P). Tema 3 (T)
Semana 6. Tema 3 (T) y Tema 3 (P)
Semana 7. Tema 3 (P) y Tema 4 (T)
Semana 8. Tema 4 (P y L)
Semana 9. Prueba evaluación Temas 3 y 4 (T y P). Tema 5 (T)
Semana 10. Tema 5 (P)
Semana 11. Tema 6 (T)
Semana 12. Tema 6 (P y L)
Semana 13. Prueba evaluación Temas 5 y 6 (T y P)
Semana 14. Laboratorio
Semana 15. Laboratorio

PROGRAMACIÓN DE ACTIVIDADES Y EVALUACIONES:

PROGRAMACIÓN DE ACTIVIDADES:

Actividad	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	¿Se evalúa?	EO	EE
Prueba evaluación Temas 1 y 2					X											X	X	X
Prueba evaluación Temas 3 y 4									X							X	X	X
Prueba evaluación Temas 5 y 6													X			X	X	X
Entrega y defensa Practicas Temas 1 y 2							X									X	X	
Entrega y defensa Practicas Temas 3 y 4											X					X	X	
Entrega y defensa Practicas Temas 5 y 6															X	X	X	
Entrega y defensa Práctica Final															X	X	X	X

CONSIDERACIONES DE LA EVALUACIÓN:

Todas las pruebas que se hagan en la asignatura y que supongan la adquisición de alguna de las competencias definidas son eliminatorias. El alumno se presentará a la prueba final únicamente con aquellas partes que no haya superado (nota aprobado, 5) a lo largo del trabajo diario y evaluación continua del semestre.

Todas las prácticas programadas son obligatorias y son indispensables para la superación de la asignatura (nota aprobado, 5). Dichas prácticas se entregarán y evaluarán de forma individual salvo que se indique para alguna de ellas lo contrario expresamente.

En la convocatoria extraordinaria el alumno se presentará con todos aquellos elementos que no haya superado (nota aprobado, 5) en el periodo de evaluación continua y convocatoria ordinaria.

La realización fraudulenta de cualquiera de las pruebas de evaluación, así como la extracción de información de las pruebas de evaluación, será sancionada según lo descrito en el Reglamento 7/2015, de 20 de noviembre, de Régimen Disciplinario de los estudiantes, Arts. 4, 5 y 7 y derivarán en la pérdida de la convocatoria correspondiente, así como en el reflejo de la falta y de su motivo en el expediente académico del alumno.

Los sistemas de evaluación descritos en esta GD son sensibles tanto a la evaluación de las competencias como de los contenidos de la asignatura.

SISTEMAS DE EVALUACIÓN:

SISTEMA DE EVALUACIÓN	PORCENTAJE (%)
Pruebas de respuesta larga, de desarrollo	20%
Trabajos y proyectos	50%
Pruebas de ejecución de tareas reales y/o simuladas	10%
Pruebas objetivas	20%

EVALUACIÓN EXCEPCIONAL:

Los estudiantes que por razones excepcionales no puedan seguir los procedimientos habituales de evaluación continua exigidos por el profesor podrán solicitar no ser incluidos en la misma y optar por una «evaluación excepcional». El estudiante podrá justificar la existencia de estas razones excepcionales mediante la cumplimentación y entrega del modelo de solicitud y documentación requerida para tal fin en la Secretaría de la Universidad Europea Miguel de Cervantes en los siguientes plazos: con carácter general, desde la formalización de la matrícula hasta el viernes de la segunda semana lectiva del curso académico para el caso de alumnos de la Universidad, y hasta el viernes de la cuarta semana lectiva del curso académico para el caso de alumnos de nuevo ingreso. En los siete días hábiles siguientes al momento en que surja esa situación excepcional si sobreviene con posterioridad a la finalización del plazo anterior.