



Guía Didáctica - GRADO

ASIGNATURA: Metodología de Programación

Título: Grado en Ingeniería Informática

Módulo: Fundamentos de Informática

Créditos: 6 ECTS

Código: 08GIIN

Índice

1. Organización general.....	3
1.1. Datos de la asignatura.....	3
1.2. Introducción a la asignatura.....	3
1.3. Competencias y resultados de aprendizaje	3
2. Contenidos/temario	6
4. Metodologías Docentes	8
5. Evaluación	9
5.1. Sistema de evaluación.....	9
5.2. Sistema de Calificación.....	10
6. Bibliografía	11

1. Organización general

1.1. Datos de la asignatura

MÓDULO	Fundamentos de Informática
ASIGNATURA	08GIIN - Metodología de Programación 6 ECTS
Carácter	Obligatorio
Curso	Primero
Cuatrimestre	Segundo
Idioma en que se imparte	Castellano
Requisitos previos	04GIIN - Fundamentos de Programación o conocimientos en programación básicos
Dedicación al estudio recomendada por ECTS	25 horas

1.2. Introducción a la asignatura

En esta asignatura profundizaremos los conocimientos vistos en Fundamentos de programación (o conocimientos básicos previos en programación). El alumno seguirá mejorando sus competencias en la programación mediante la aplicación de las mismas al desarrollo de un pequeño proyecto. En paralelo se introducirán nuevos conceptos de programación para afrontar cada vez problemas más complejos.

Las competencias de programación son parte fundamental de todo futuro Ingeniero en Informática.

1.3. Competencias y resultados de aprendizaje

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.

CB2. *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.*

CB3. *Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.*

CB4. *Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.*

CB5. *Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.*

COMPETENCIAS GENERALES DE GRADO

CG8 - Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

COMPETENCIAS ESPECÍFICAS DE LA ASIGNATURA

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.

FB4 - Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

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.

R9 - Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.

RESULTADOS DE APRENDIZAJE

Al finalizar esta asignatura se espera que el estudiante sea capaz de:

RA.1.- Explicar la relación de tipos de lenguaje de alto nivel y la representación a bajo nivel de dicha información.

RA.2.- Distinguir correctamente las referencias y los objetos referenciados.

RA.3.- Desarrollar nuevos tipos de datos, realizando una correcta separación entre interfaz e implementación.

RA.4.- Manejar correctamente herramientas de depuración, pruebas y validación.

RA.5.- Realizar una correcta gestión de la E/S, especialmente motivada por la necesidad de manejar grandes cantidades de información almacenada en ficheros.

RA.6. - Usar Python para la programación de ordenadores.

RA.7. - Aplicar metodologías de programación para la resolución de problemas.

2. Contenidos/temario

Unidad Competencial 1

- 1.1. Tipos de datos del lenguaje de alto nivel y su representación interna.
- 1.2. Referencias de memoria y memoria dinámica.
- 1.3. Primer componente del proyecto informático de programación.

Unidad Competencial 2

- 2.1. Encapsulamiento y ocultamiento de la información.
- 2.2. Diseño modular y creación de bibliotecas.
- 2.3. Segundo componente del proyecto informático de programación.

Unidad Competencial 3

- 3.1. Herramientas de depuración, pruebas y validación.
- 3.2. Gestión de errores.
- 3.3. Tercer componente del proyecto informático de programación.

Unidad Competencial 4

- 4.1. Mantenimiento del software.
- 4.2. E/S (Input/Output), ficheros.
- 4.3. Integración componentes del proyecto informático de programación.

3. Actividades Formativas

ACTIVIDAD FORMATIVA	HORAS	PRESENCIALIDAD
Clases expositivas	60	60
Resolución de ejercicios prácticos	80	30
Prácticas de laboratorios virtuales	100	20
Tutorías	60	0
Trabajo Autónomo	300	0

4. Metodologías Docentes

Clases teóricas impartidas como lecciones magistrales o exposiciones, en las que además de presentar el contenido de la asignatura, se explican los conceptos fundamentales y se desarrolla el contenido teórico.

Colección de tareas que el alumnado llevará a cabo a lo largo de toda la asignatura, entre las que podemos encontrar: análisis de casos, resolución de problemas, prácticas de laboratorios, comentarios críticos de textos, análisis de lecturas, etc.

Sesiones periódicas entre el profesorado y el alumnado para la resolución de dudas, orientación, supervisión, etc.

Trabajo tanto individual como grupal para la lectura crítica de la bibliografía, estudio sistemático de los temas, reflexión sobre problemas planteados, resolución de actividades propuestas, búsqueda, análisis y elaboración de información, investigación e indagación, así como trabajo colaborativo basado en principios constructivistas.

5. Evaluación

5.1. Sistema de evaluación

El Modelo de Evaluación de estudiantes en la Universidad se sustenta en los principios del Espacio Europeo de Educación Superior (EEES), y está adaptado a la estructura de formación virtual propia de esta Universidad. De este modo, se dirige a la evaluación de competencias.

Es requisito indispensable aprobar el portafolio y la prueba final con un mínimo de 5 para ponderar las calificaciones.

Sistema de Evaluación	Ponderación
Portafolio*	40 %
Participación en el foro sobre un tema indicado por el profesor. Realización de un proyecto en Python de forma incremental donde se pondrán en práctica los conceptos teóricos que se vayan viendo durante el curso.	
Sistema de Evaluación	Ponderación
Prueba final*	60 %
El examen final se realizará online y constará de preguntas teóricas, preguntas de detectar y corregir errores en el código y preguntas prácticas donde deben desarrollar un pequeño programa donde pondrán en práctica todo lo visto durante el curso sobre programación en Python.	

***Es requisito indispensable para superar la asignatura aprobar cada apartado (portafolio y prueba final).**

Atendiendo a la Normativa de Evaluación de la Universidad, se tendrá en cuenta que la utilización de **contenido de autoría ajena** al propio estudiante debe ser citada adecuadamente en los trabajos entregados. Los casos de plagio serán sancionados con suspenso (0) de la actividad en la que se detecte. Asimismo, el uso de **medios fraudulentos durante las pruebas de evaluación** implicará un suspenso (0) y podrá implicar la apertura de un expediente disciplinario.

5.2. Sistema de Calificación

La calificación de la asignatura se establecerá en los siguientes cálculos y términos:

Nivel de Competencia	Calificación Oficial	Etiqueta Oficial
Muy competente	9,0 - 10	Sobresaliente
Competente	7,0 - 8,9	Notable
Aceptable	5,0 - 6,9	Aprobado
Aún no competente	0,0 - 4,9	Suspense

El nivel de competencia en cada una de las actividades realizadas se medirá, teniendo en cuenta **criterios generales derivados de la consecución de los resultados de aprendizaje**, que en términos generales y en función de la adecuación en el planteamiento de los contenidos generales y contenidos específicos, valorarán por norma general y en trabajos escritos, la corrección de la estructura formal y organización del discurso (semántica, sintaxis y léxico) valorándose además la originalidad, creatividad y argumentación de las intervenciones utilizando referencias bibliográficas.

Sin detrimento de lo anterior, el alumnado dispondrá de una **rúbrica simplificada (en cada una de las actividades en el CAMPUS)** que mostrará los aspectos que valorará el docente, como así también los **niveles de desempeño que tendrá en cuenta para calificar las actividades vinculadas a cada resultado de aprendizaje.**

6. Bibliografía

1. Pigoski, T. M. (1996). Practical software maintenance: best practices for managing your software investment. Wiley Publishing.
2. Python Software Foundation. Python 3 documentation. Recuperado el (18-01-2021) de <https://docs.python.org/3/>
3. Python Software Foundation. Package Index. Recuperado el (18-01-2021) de <https://pypi.org/>
4. Pedro Gomis Román (2017). Manual de la asignatura 04GIIN Fundamentos de Programación. Grado de Ingeniería Informática Universidad Internacional de Valencia.
5. Python Software Foundation. Scientific PYthon Development EnviRonment (Spyder). Recuperado el (18-01-2021) de <https://pypi.org/project/spyder/>
6. Pressman, R. (2010). Ingeniería del Software-Enfoque Práctico Mc Graw Hill 7ª. Edición.
7. Almeida, M. A. R. (1993). Metodología de la programación: a través de Pseudocódigo. MacGraw-Hill.
8. Balcázar, J. L. (1993). Programación metódica (pp. I-XVI). McGraw-Hill.