



Universidad
Internacional
de Valencia

Guía didáctica

ASIGNATURA: *Metodología de Programación*

Título: *Grado en Ingeniería Informática*

Materia: *Fundamentos de Informática*

Créditos: 6 ECTS

Código: 08GIIN

Índice

1. Organización general.....	3
1.1. Datos de la asignatura.....	3
1.2. Equipo docente	3
1.3. Introducción a la asignatura.....	3
1.4. Competencias y resultados de aprendizaje	4
2. Contenidos/temario	5
3. Metodología	5
4. Actividades formativas	5
5. Evaluación.....	7
5.1. Sistema de evaluación.....	7
5.2. Sistema de calificación	7
6. Bibliografía.....	8
6.1. Bibliografía de referencia.....	8
6.2. Bibliografía complementaria.....	8

1. Organización general

1.1. Datos de la asignatura

MÓDULO	Formación Básica
MATERIA	Fundamentos de Informática
ASIGNATURA	Metodología de Programación 6 ECTS
Carácter	Básica
Curso	Primero
Cuatrimestre	Segundo
Idioma en que se imparte	Castellano
Requisitos previos	Se recomienda la adquisición de los conocimientos y competencias de las materias de formación básica, aunque es especialmente importante la superación de la asignatura Fundamentos de programación.
Dedicación al estudio por ECTS	25 horas

1.2. Equipo docente

Profesor	Dr. Roger Clotet Martínez roger.clotet@professor.universidadviu.com
-----------------	--

1.3. 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.

1.4. Competencias y resultados de aprendizaje

COMPETENCIAS GENERALES

- CG.8. - Conocimiento de las materias básicas y tecnologías, que capaciten para El aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les dotende una gran versatilidad para adaptarse a nuevas situaciones.

COMPETENCIAS ESPECÍFICAS DE LA ASIGNATURA

- CE.3.- 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.
- CE.4.- Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
- CE.5. - 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.
- CE.9. - 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 bajonivel de dicha información.
- RA.2.- Distinguir correctamente las referencias y los objetos referenciados.
- RA.3.- Desarrollar nuevos tipos de datos, realizando una correcta separación entreinterfaz 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.

2. Contenidos/temario

1. Tipos de datos del lenguaje de alto nivel y su representación interna
2. Referencias de memoria y memoria dinámica
3. Encapsulamiento y ocultamiento de la información
4. Diseño modular y creación de bibliotecas
5. Herramientas de depuración, pruebas y validación
6. Gestión de errores
7. Mantenimiento del software
8. I/O, ficheros
9. Proyecto informático de programación

3. Metodología

La metodología de la Universidad Internacional de Valencia (VIU) se caracteriza por una apuesta decidida en un modelo de carácter e-presencial. Así, siguiendo lo estipulado en el calendario de actividades docentes del Título, se impartirán en directo un conjunto de sesiones, que, además, quedarán grabadas para su posterior visionado por parte de aquellos estudiantes que lo necesitasen. En todo caso, se recomienda acudir, en la medida de lo posible, a dichas sesiones, facilitando así el intercambio de experiencias y dudas con el docente.

En lo que se refiere a las metodologías específicas de enseñanza-aprendizaje, serán aplicadas por el docente en función de los contenidos de la asignatura y de las necesidades pedagógicas de los estudiantes. De manera general, se impartirán contenidos teóricos y, en el ámbito de las clases prácticas se podrá realizar la resolución de problemas, el estudio de casos y/o la simulación.

Por otro lado, la Universidad y sus docentes ofrecen un acompañamiento continuo al estudiante, poniendo a su disposición foros de dudas y tutorías para resolver las consultas de carácter académico que el estudiante pueda tener. Es importante señalar que resulta fundamental el trabajo autónomo del estudiante para lograr una adecuada consecución de los objetivos formativos previstos para la asignatura.

4. Actividades formativas

Durante el desarrollo de cada una de las asignaturas se programan una serie de actividades de aprendizaje que ayudan a los estudiantes a consolidar los conocimientos trabajados.

A continuación, se relacionan las actividades que forman parte de la asignatura:

1. Actividades de carácter teórico

Se trata de un conjunto de actividades guiadas por el profesor de la asignatura destinadas a la adquisición por parte de los estudiantes de los contenidos teóricos de la misma. Estas actividades, diseñadas de manera integral, se complementan entre sí y están directamente relacionadas con los materiales teóricos que se ponen a disposición del estudiante (manual, SCORM y material complementario). Estas actividades se desglosan en las siguientes categorías:

- a. Clases expositivas
- b. Sesiones con expertos en el aula
- c. Observación y evaluación de recursos didácticos audiovisuales
- d. Estudio y seguimiento de material interactivo

2. Actividades de carácter práctico

Se trata de un conjunto de actividades guiadas y supervisadas por el profesor de la asignatura vinculadas con la adquisición por parte de los estudiantes de los resultados de aprendizaje y competencias de carácter más práctico. Estas actividades, diseñadas con visión de conjunto, están relacionadas entre sí para ofrecer al estudiante una formación completa e integral.

3. Tutorías

Se trata de sesiones, tanto de carácter síncrono como asíncrono (e-mail), individuales o colectivas, en las que el profesor comparte información sobre el progreso académico del estudiante y en las que se resuelven dudas y se dan orientaciones específicas ante dificultades concretas en el desarrollo de la asignatura.

4. Trabajo autónomo

Se trata de un conjunto de actividades que el estudiante desarrolla autónomamente y que están enfocadas a lograr un aprendizaje significativo y a superar la evaluación de la asignatura. La realización de estas actividades es indispensable para adquirir las competencias y se encuentran entroncadas en el aprendizaje autónomo que consagra la actual ordenación de enseñanzas universitarias. Esta actividad, por su definición, tiene carácter asíncrono.

5. Prueba objetiva final

Como parte de la evaluación de cada una de las asignaturas (a excepción de las prácticas y el Trabajo fin de título), se realiza una prueba (examen final). Esta prueba se realiza en tiempo real (con los medios de control antifraude especificados) y tiene como objetivo evidenciar el nivel de adquisición de conocimientos y desarrollo de competencias por parte de los estudiantes. Esta actividad, por su definición, tiene carácter síncrono.

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.

Sistema de Evaluación	Ponderación
Portafolio*	40 %
<i>Entrega de informes de problemas y ejercicios</i>	10%
<i>Informes o memorias de prácticas de laboratorio</i>	10%
<i>Trabajos o proyectos desarrollados en grupo o de forma individual</i>	15%
<i>Participación activa en los debates, foros y otros medios</i>	5%
Sistema de Evaluación	Ponderación
Prueba final*	60 %
<p><i>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 y desarrollar cambios sobre su proyecto.</i></p> <p><i>Las teóricas son selección simple, las de detectar y corregir errores deben redactar la solución (se valorará hasta donde lleguen, por ejemplo si solo detectan los errores pero no dan la solución correcta a los mismos tendrán el 50% de la nota) y por último las preguntas prácticas deben realizarlas en el Spyder, después deben poner todos los archivos generados en un .zip y subirlo como respuesta.</i></p>	

***Es requisito indispensable para superar la asignatura aprobar cada apartado (portafolio y prueba final) con un mínimo de 5 para ponderar las calificaciones.**

Los enunciados y especificaciones propias de las distintas actividades serán aportados por el docente, a través del Campus Virtual, a lo largo de la impartición de la asignatura.

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 aprendizaje	Calificación numérica	Calificación cualitativa
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	Suspenso

Sin detrimento de lo anterior, el estudiante dispondrá de una **rúbrica simplificada** en el aula 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**.

La mención de «**Matrícula de Honor**» podrá ser otorgada a estudiantes que hayan obtenido una calificación igual o superior a 9.0. Su número no podrá exceder del cinco por ciento de los estudiantes matriculados en una materia en el correspondiente curso académico, salvo que el número de estudiantes matriculados sea inferior a 20, en cuyo caso se podrá conceder una sola «Matrícula de Honor».

6. Bibliografía

6.1. Bibliografía de referencia

- M. J. Algar Díaz y M. Fernández de Sevilla Vellón, **Introducción práctica a la programación con Python**. Alcalá de Henares: Editorial Universidad de Alcalá, 2019.[En Línea] Disponible en: <https://elibro-net.universidadviu.idm.oclc.org/es/ereader/universidadviu/124259>
- Á. Hinojosa Gutiérrez, **Python paso a paso**. Paracuellos de Jarama, Madrid: RA-MAEditorial, 2015. [En Línea] Disponible en: <https://elibro-net.universidadviu.idm.oclc.org/es/ereader/universidadviu/107213>

6.2. Bibliografía complementaria

- Zuras, D., Cowlshaw, M., Aiken, A., Applegate, M., Bailey, D., Bass, S., ... & Canon, S. (2008). IEEE standard for floating-point arithmetic. IEEE Std 754-2008, 1-70.
- Python Software Foundation. Python 3 documentation. Recuperado el (17-01-2022) de <https://docs.python.org/3/>
- Python Software Foundation. Package Index. Recuperado el (17-01-2022) de <https://pypi.org/>
- Pedro Gomis Román (2017). Manual de la asignatura 04GIIN Fundamentos de Programación. Grado de Ingeniería Informática Universidad Internacional de Valencia.
- Python Software Foundation. Scientific PYthon Development EnviRonment (Spyder).

Recuperado el (17-01-2022) de <https://pypi.org/project/spyder/>

- Lehman, Meir M. (1980). «Programs, Life Cycles, and Laws of Software Evolution». Proc. IEEE 68 (9): 1060-1076.
- Lehman, M. M., Ramil, J. F., Wernick, P. D., Perry, D. E., & Turski, W. M. (1997, November). Metrics and laws of software evolution-the nineties view. In Software metrics symposium, 1997. proceedings., fourth international (pp. 20-32). IEEE.
- Boehm, B. W. (1988). A spiral model of software development and enhancement. Computer, 21(5), 61-72.
- Almeida, M. A. R. (1993). Metodología de la programación: a través de Pseudocódigo. MacGraw-Hill.