
# Guía didáctica

## ASIGNATURA: *Fundamentos de Programación*

**Título**: *Grado en Ingeniería Informática*
**Materia**: *Fundamentos de Informática*
**Créditos**: *6 ECTS*
**Código**: *04GIIN*

# Índice

# 1. Organización general

## 1.1. Datos de la asignatura

| | |
|---|---|
| **MÓDULO** | ***Formación Básica*** |
| **MATERIA** | ***Fundamentos de Informática*** |
| **ASIGNATURA** | *Fundamentos de Programación*<br>**6 ECTS** |
| **Carácter** | Básica |
| **Curso** | Primero |
| **Cuatrimestre** | Primero |
| **Idioma en que se imparte** | Castellano |
| **Requisitos previos** | No existen |
| **Dedicación al estudio por ECTS** | **25 horas** |

## 1.2. Equipo docente

| | |
|---|---|
| **Profesor** | **Dr. Pedro Gomis Román**<br>*pedro.gomis@professor.universidadviu.com* |

## 1.3. Introducción a la asignatura

*Fundamentos de Programación contribuye a que los alumnos obtengan competencias en el uso de la programación para resolver problemas propios de la ingeniería, que incluyen procesos informáticos.*

*El enfoque del curso es aprender a programar a través de un lenguaje de programación. Las herramientas de los lenguajes de programación incluyen el uso de variables y operadores en expresiones para hacer cálculos, de estructuras algorítmicas secuenciales, alternativas e iterativas para manejar el control del flujo de las instrucciones, el uso de funciones o subprogramas y el manejo de estructuras de datos. El objetivo general es adquirir habilidades para hacer que el computador, a través de programas, resuelva los problemas que el usuario se enfrentará en su campo laboral.*

## 1.4. Competencias y resultados de aprendizaje

### COMPETENCIAS GENERALES

CG.8.- Conocimientos 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

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.

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 el funcionamiento de un ordenador enfatizando la necesidad de desarrollo de software por parte del programador. Diseñar expresiones algorítmicas.

RA.2.- Usar las estructuras de control básicas: secuencial, condicional e iterativa.

RA.3.- Resolver problemas aplicando una metodología de diseño modular (top-down/bottom-up).

RA.4.- Manejar correctamente los mecanismos de comunicación entre módulos (interfaces), así como las distintas formas de paso de parámetros y devolución de resultados.

# 2. Contenidos/temario

## Unidad Competencial 1

1.1 Introducción. Visión general del mundo de la informática

1.1.1 Computadores. Aspectos históricos

1.1.2 Computadores. Estructura funcional

1.1.3 Codificación de información en computadores

1.1.4 Algoritmos, programas y lenguajes

1.2 Tipos de datos básicos

1.2.1 Datos simples: enteros, reales, booleanos, carácter

*1.2.2* Dato compuesto básico: *string*

1.3 Objetos, operadores y expresiones

1.3.1 Variables y acción de asignación

1.3.2 Expresiones y sentencias

1.3.3 Operadores: aritméticos, booleanos, relacionales

1.3.4 Acciones elementales

## Unidad Competencial 2

2.1 Introducción a funciones

2.1.1 Uso de funciones. Funciones internas y de módulos

2.1.2 Funciones y procedimientos

2.1.3 Diseño de funciones

2.2 Estructuras de control

2.2.1 Estructura secuencial

2.2.2 Estructuras condicionales-alternativas: simple, doble y múltiple oanidada

*2.2.3* Estructuras iterativas. Esquemas de recorrido y búsqueda. Composiciones *while* y *for*

## Unidad Competencial 3

3.1 Programación modular. Funciones y procedimientos

3.1.1 Funciones productivas y no productivas

3.1.2 Variables locales y globales

3.1.3 Valores de argumentos por omisión (*default*)

3.1.4 Argumentos de palabra clave (*keyword arguments*)

3.1.5 Módulos: integrar funciones en una biblioteca

3.1.6 Recursividad

3.1.7 Paso de parámetros por valor y por referencia

# 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 e interactivas con demostraciones de códigos en Python.

b. Estudio y seguimiento de material interactivo, como entornos de desarrollo integrado de software (IDE): IDLE de Python, Spyder,etc.

### *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.

- Introducción a las actividades prácticas entregables de ejercicios de programación codificados en lenguaje Python.

- Ejercicios de programas en Python desarrollados durante las videoconferencias.

### *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. Los detalles de la prueba serán introducidos por el profesor en las clases. Suelen incluir

preguntas tipo test y algunas preguntas de respuesta breve o dediseño de un código de programa en Python.

# 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 %** |
| *Entrega de informes de problemas y ejercicios* | *15 %* |
| *Informes o memorias de prácticas de laboratorio* | *20 %* |
| *Participación activa en los debates, foros y otros medios* | *5 %* |
| . | |
| **Sistema de Evaluación** | **Ponderación** |
| **Prueba final*** | **60 %** |
| Preguntas tipo test de opciones de respuesta entre varias opciones y algunas preguntas de respuesta breve o de diseño de un código de programa en Python. | |

***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 cómputos 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

Gomis, Pedro, (2018). *Fundamentos de programación. Manual de la asignatura,* Universidad Internacional de Valencia. Disponible en el aula en: Recursos y materiales – 03. Materiales del profesor.

Severance, Charles, (2015). *Python para Informáticos: Explorando la información (Version 2.7.2)*. Licencia Creative Commons. Recuperado de: http://do1.dr-chuck.net/py4inf/ES-es/book.pdf

Downey, A. (2015). *Think Python. How to Think Like a Computer Scientist.* Needham, MA: Green Tea Press. Recuperado de: http://greenteapress.com/wp/think-python-2e/

## 6.2. Bibliografía complementaria

Harrington, A. N. (2015). *Hands-on Python Tutorial*. Recuperado de: http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/

Klein, B. (2016). *Python 3 Tutorial*. Bodenseo. Recuperado de: https://www.python-course.eu/python3_course.php

Marzal-Varó, A., García-Luengo, I., & García-Sevilla, P. (2014). *Introducción a la programación con Python 3*. Castellón, España: Universitat Jaume I. Recuperados de: http://repositori.uji.es/xmlui/handle/10234/102653

Python Software Foundation (2108). *The Python Tutorial*. Recuperado de: https://docs.python.org/3/tutorial/index.html

Refsnes Data. (2020). *Python Tutorial*. Recuperado de: https://www.w3schools.com/python/