



# Guía Didáctica - GRADO

## ASIGNATURA: **Fundamentos de Programación**

Título: **Grado en Ingeniería Informática**

Módulo: **Formación Básica**

Créditos: **6 ECTS**

Código: **04GIIN**

# Índice

1. Organización general.....	3
1.1. Datos de la asignatura.....	3
1.2. Introducción a la asignatura.....	3
1.4. Competencias y resultados de aprendizaje .....	3
2. Contenidos/temario .....	6
3. Evaluación .....	10
3.1. Sistema de evaluación.....	10
3.2. Sistema de Calificación.....	11
4. Bibliografía .....	12

# 1. Organización general

## 1.1. Datos de la asignatura

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

## 1.2. 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.*

## 1.3. 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**

**CG.1.-** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

**CG.2.-** 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**

**C.E.1.-** Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

**C.E.2.-** 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.

## **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.

**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.

RA.5.- Utilizar datos estructurados de tipo homogéneos, heterogéneos, dinámicos y estáticos para representar información

## 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 o anidada
- 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

3.2 Tipos de datos estructurados en Python: estáticos o inmutables y dinámicos o mutables. Datos Homogéneos y heterogéneos.

3.2.1 Datos estructurados inmutables: *strings*

3.2.2 Datos estructurados inmutables: Tuplas

#### Unidad Competencial 4

4.1 Tipos de datos estructurados en Python: dinámicos o mutables

4.1.1 Datos estructurados mutables o dinámicos: listas

4.1.2 Datos estructurados mutables o dinámicos: diccionarios

4.2 Tópicos especiales en Python

4.2.1 Asignaciones múltiples con tuplas

4.2.2 Iteraciones paralelas con la función zip

4.2.3 Copia superficial y copia profunda de objetos

4.2.4 Comprensión de listas (*list comprehension*)

### 3. Actividades Formativas

<b>ACTIVIDAD FORMATIVA</b>	<b>HORAS</b>	<b>PRESENCIALIDAD</b>
Clases expositivas	15	60
Resolución de ejercicios prácticos	20	30
Prácticas de laboratorios virtuales	25	20
Tutorías	15	0
Trabajo Autónomo	75	0



## 4. Metodologías Docentes

- 4.1. Sesiones de clases teóricas impartidas como lecciones magistrales o exposiciones que tienen por objetivo principal presentar el contenido de la asignatura, explicar los conceptos fundamentales y desarrollar el contenido teórico.
- 4.2. Sesiones de prácticas y problemas en donde el alumnado trabajará aplicando los aprendizajes alcanzados.
- 4.3. Sesiones de tutorías entre el profesorado y el alumnado para la resolución de dudas, orientación, supervisión, etc.
- 4.4. 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
<b>Portafolio*</b>	<b>40 %</b>
<p>Colección de tareas realizadas por el alumnado y establecidas por el profesorado. Las tareas incluyen: entrega de informes de problemas y ejercicios, análisis y diseño de software e informes de prácticas de laboratorio.</p> <p>Portafolio Actividad 1: 10%</p> <p>Portafolio Actividad 2: 10%</p> <p>Portafolio Actividad 3: 10%</p> <p>Portafolio Actividad 4: 10%</p>	
Sistema de Evaluación	Ponderación
<b>Prueba final*</b>	<b>60 %</b>
<p>La realización de una prueba cuyas características son definidas en cada caso por el correspondiente profesorado.</p>	

**\*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** 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**.

## 5. Bibliografía

### **Bibliografía Básica:**

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>

### **Bibliografía Opcional:**

Klein, B. (2020). Python 3 Tutorial. Bodenseo. Recuperado de: [https://www.python-course.eu/python3\\_course.php](https://www.python-course.eu/python3_course.php)

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

Marzal-Varó, A., García-Luengo, I., y 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>