

# Guía Docente

## DATOS DE IDENTIFICACIÓN

Titulación:	Grado en Ingeniería Biomédica		
Rama de Conocimiento:	Ciencias		
Facultad/Escuela:	Ciencias Experimentales		
Asignatura:	Programación y Algoritmia		
Tipo:	Formación Básica	Créditos ECTS:	6
Curso:	1	Código:	2405
Periodo docente:	Segundo semestre		
Materia:	Informática		
Módulo:	Fundamentos de Bioingeniería		
Tipo de enseñanza:	Presencial		
Idioma:	Castellano		
Total de horas de dedicación del alumno:	150		

Equipo Docente	Correo Electrónico
Rodrigo Madurga de Lacalle	rodrigo.madurga@ufv.es

## DESCRIPCIÓN DE LA ASIGNATURA

La figura del ingeniero biomédico surge para dar respuesta a los problemas relacionados con la salud, a partir de soluciones tecnológicas innovadoras y sofisticadas, con una elevada responsabilidad social. La ingeniería biomédica se basa en la sinergia de diversas disciplinas, entre ellas la física y la biología, la química y diversas ingenierías.

Entre las diversas áreas de aplicación de la ingeniería biomédica, están las relacionadas con la inteligencia artificial y el análisis de datos, de interés en el desarrollo de diseños inteligentes y telemedicina, biosimulación, mejora en los tiempos de diagnóstico y modelización de sistemas biológicos.

La asignatura de Programación y algoritmia permite sentar las bases de la computación y el desarrollo de algoritmos, que se aplicarán en las diversas áreas antes mencionadas.

La asignatura Programación proporciona los conocimientos necesarios sobre programación para la resolución de problemas mediante la creación de programas informáticos, siguiendo la metodología de programación estructurada y el paradigma de orientación a objetos, haciendo especial hincapié en la presentación de las técnicas algorítmicas esenciales y la utilización de diferentes entornos de programación para la construcción de aplicaciones informáticas, aplicando metodologías de manera rigurosa que proporcionen al alumno buenos hábitos en el desarrollo de software.

Esta asignatura corresponde al módulo Formación Básica y, dentro de éste, a la materia Informática. Se imparte en el segundo semestre del primer curso de los estudios de Grado en Ingeniería Biomédica, y requiere de una dedicación de 150 horas por parte del alumno.

Formación Básica y, dentro de éste, a la materia Informática.

La asignatura se centra en la resolución de problemas, partiendo de una metodología estructurada y siguiendo con el paradigma de la orientación a objetos, aprendiendo las ventajas de la utilización de clases y objetos así como del beneficio que aporta la abstracción, la herencia y el polimorfismo. Para ello se llevará a cabo la implementación en el lenguaje Python 3, empleando diferentes entornos de desarrollo para su posterior depuración y documentación. Además, la asignatura presenta las principales librerías científicas de Python, que el alumno utilizará en las diferentes asignaturas en cursos superiores. La asignatura proporciona también las herramientas básicas para que el alumno se familiarice con el sistema operativo Linux y sus comandos básicos. Este sistema operativo juega un papel fundamental en el ámbito científico.

## OBJETIVO

El objetivo de la asignatura es trabajar los contenidos básicos de diseño de algoritmos y bases de datos y de técnicas de programación.

El principal objetivo de esta asignatura consiste en que el estudiante aprenda a analizar, diseñar, implementar y probar software usando el paradigma de orientación a objetos. Por ende será capaz de emplear métodos, técnicas y herramientas propias del ingeniero para la construcción de aplicaciones robustas y de fácil mantenimiento mediante el uso de este paradigma. Se hará especial énfasis en la fase de implementación, usando el lenguaje de programación Python 3.

El objetivo final de la asignatura consiste en presentar a los estudiantes los principios de la resolución sistemática de problemas a través del diseño de algoritmos y la codificación de una solución, siguiendo las reglas básicas de la computación. El lenguaje empleado es Python 3 con el IDE Spyder y la herramienta web Jupyter Notebook. De este modo, el alumno se empieza a familiarizar con un lenguaje moderno y a priori sencillo. El producto resultante, solución informática, es fruto del carácter de artista que posee el desarrollador, en este caso el alumno. Por ende, esta asignatura plantea acompañar al alumno en el descubrimiento de que su sed científica es su humanidad, produciéndose un encuentro de búsqueda entre formador y alumno, abriendo las puertas para poder afrontar así las preguntas antropológicas propias del hombre como científico e ingeniero informático, en torno al qué, el para qué y las consecuencias de los actos humanos, sociales, políticos y económicos.

La asignatura proporcionará las herramientas necesarias y una base sólida en programación para que el alumno trabaje de manera autónoma en cursos superiores.

## CONOCIMIENTOS PREVIOS

Se recomienda disponer de los conocimientos básicos de álgebra y aritmética.

## CONTENIDOS

Tema 1. El sistema operativo Linux  
- Historia  
- Comandos básicos

- Programación de scripts.
- Tema 2. Introducción a la programación: conceptos básicos.
  - Introducción a los computadores.
  - Algoritmos + Estructuras de datos = Programas.
  - Lenguajes y paradigmas de programación.
- Tema 3. Entornos de programación en Python.
- Tema 4. Elementos del lenguaje Python.
  - Estructura de un programa.
  - Variables, identificadores, expresiones y operadores.
- Tema 5. Entrada y salida de datos.
  - Entrada por teclado.
  - Salida por pantalla.
- Tema 6. Tipos de datos en Python.
  - Clases de tipado.
  - Números, cadenas, booleanos.
- Tema 7. Estructuras de control de flujo.
  - Estructuras condicionales.
  - Estructuras iterativas.
- Tema 8. Funciones en Python.
  - Librerías internas y externas.
  - Funciones internas y métodos.
  - Anotaciones y parámetros.
  - Variables locales y globales.
  - Paso de parámetros.
- Tema 9. Listas, Tuplas y Diccionarios.
  - Listas de una y varias dimensiones.
  - Creación y manejo de listas.
  - Creación y manejo de tuplas.
  - Creación y manejo de diccionarios.
  - Conversión entre estructuras.
  - Funciones y métodos.
- Tema 10. Cadenas de Caracteres.
  - Cadenas de caracteres.
  - Funciones y métodos para su tratamiento.
- Tema 11. Tratamiento de Excepciones.
  - Errores sintácticos y Excepciones.
  - Control de excepciones.
- Tema 12. Archivos.
  - Lectura básica de archivos.
  - Escritura básica en archivos.
- Tema 13. El paradigma Orientado a Objetos.
- Tema 14. Encapsulación, Herencia y Polimorfismo.
- Tema 15. Librerías científicas fundamentales:
  - Gráficos Matplotlib, Bohem y Seaborn.
  - Numérica: SciPy
  - Estructura de datos: Pandas
  - vectores y matrices: NumPy

## ACTIVIDADES FORMATIVAS

La metodología seguida en esta asignatura está enfocada en un aprendizaje basado en problemas, donde para alcanzar los resultados de aprendizaje propuestos se combinan lecciones expositivas, clases prácticas, clases invertidas y sesiones de trabajo colaborativo. El objetivo principal de esta metodología es llevar a cabo un aprendizaje basado en la resolución de pequeños ejercicios para poner en práctica los conceptos asimilados. En estas clases prácticas se fomenta la discusión y el trabajo en equipo con apoyo del profesor para facilitar el aprendizaje, donde exista un ambiente de trabajo óptimo para complementar y culminar el trabajo autónomo del alumno.

Dado que esta metodología requiere un alto grado de organización, se utilizará el Aula Virtual como plataforma de comunicación, soporte de materiales de estudio y como repositorio de prácticas y ejercicios.

--

## DISTRIBUCIÓN DE LOS TIEMPOS DE TRABAJO

ACTIVIDAD PRESENCIAL	TRABAJO AUTÓNOMO/ACTIVIDAD NO PRESENCIAL
60 horas	90 horas

## COMPETENCIAS

### Competencias básicas

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

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

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

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

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

### Competencias generales

Conocer y asimilar conocimientos científico-técnicos y su aplicación a sistemas médicos y biológicos para la identificación y comprensión de los continuos avances de las tecnologías biomédicas de manera autónoma.

### Competencias específicas

Capacidad para adquirir, obtener, formalizar y representar el conocimiento humano en una forma computable para la resolución de problemas mediante un sistema informático en el ámbito de la Ingeniería Biomédica, en particular, relacionados con aspectos de computación, percepción y actuación en ambientes o entornos inteligentes.

## RESULTADOS DE APRENDIZAJE

Emplear y comparar los aspectos generales de los diferentes paradigmas y lenguajes de programación, como el estructurado, orientado a objetos o lógico, así como de cada una de las fases del proceso de desarrollo de software, especialmente las de diseño e implementación.

Analizar e interpretar el problema a resolver y elaborar algoritmos utilizando las estructuras adecuadas de control

y de datos.

Depurar los errores detectados y analizar las posibles alternativas de resolución del problema.

Diseñar e implementar, en distintos paradigmas de programación, algoritmos que resuelvan problemas sencillos utilizando tipos de datos elementales y estructuras de control básicas, utilizando correctamente los entornos de desarrollo y depuración.

Aplicar metodologías y buenas prácticas en la construcción de aplicaciones con diferentes entornos y herramientas de desarrollo.

Utilizar de manera eficiente librerías informáticas en diferentes ámbitos.

Trabajar de manera autónoma y con confianza sobre el sistema operativo Linux.

## SISTEMA DE EVALUACIÓN DEL APRENDIZAJE

La evaluación de la asignatura persigue valorar la adquisición y el grado de desarrollo de todas las competencias previstas en esta guía docente por parte de los alumnos. Se evaluarán los resultados de aprendizaje.

### REQUISITOS MÍNIMOS PARA APROBAR LA ASIGNATURA

En cualquier convocatoria o sistema de evaluación la asignatura se supera obteniendo una puntuación mínima de 5 en todas y cada una de las calificaciones "CAL" desglosadas en los siguientes apartados de esta Guía Docente.

### SISTEMAS DE EVALUACIÓN

#### 1) ORDINARIO

Basado en la evaluación continua. La nota final se compondrá de las siguientes calificaciones, según los porcentajes indicados:

- CAL1 (70%): Exámenes de teoría: examen final sobre los contenidos teóricos, prácticos y metodológicos tratados en la asignatura.
- CAL2 (10%): Preparación y presentación de trabajo de práctica en grupos y defensa oral.
- CAL3 (10%): Realización del trabajo práctico en laboratorio.
- CAL4 (10%): Realización y presentación de ejercicios.

El grado de aprendizaje del alumno sobre los trabajos presentados podrá ser evaluado mediante examen escrito u oral independiente de CAL1.

Convocatorias extraordinarias. En el supuesto de no haber superado CAL1, se deberá realizar un único examen escrito sobre los mismos contenidos evaluados en la convocatoria ordinaria que contará un 70%. En el supuesto de no superar CAL2 y/o CAL3 y/o CAL4, el alumno deberá presentar los ejercicios, trabajos y actividades suspensas realizados durante el semestre del año académico en curso (incluidos seminarios). Como norma general se conservarán las calificaciones de las distintas partes aprobadas en la convocatoria ordinaria.

#### 2) ALTERNATIVO (opción para alumnos de segunda matrícula o sucesivas)

No basado en la evaluación continua. El seguimiento del proceso enseñanza-aprendizaje se realizará mediante tutorías, que podrán ser obligatorias. Los alumnos en 2º o sucesivas matrículas deben contactar con el profesor para solicitar acogerse a éste sistema. La nota final se compondrá de las siguientes calificaciones, según los porcentajes indicados:

- CAL1 (70%): Exámenes de teoría: examen final sobre los contenidos teóricos, prácticos y metodológicos tratados en la asignatura.
- CAL2 (10%): Preparación y presentación de trabajo de práctica en grupos y defensa oral.
- CAL3 (10%): Realización del trabajo práctico en laboratorio.
- CAL4 (10%): Realización y presentación de ejercicios.

El grado de aprendizaje del alumno sobre los trabajos presentados podrá ser evaluado mediante examen escrito u oral independiente de CAL1.

Convocatorias extraordinarias. En el supuesto de no haber superado CAL1, se deberá realizar un único examen escrito sobre los mismos contenidos evaluados en la convocatoria ordinaria que contará un 70%. En el supuesto de no superar CAL2 y/o CAL3 y/o CAL4, el alumno deberá presentar los ejercicios, trabajos y actividades suspensas realizados durante el semestre del año académico en curso (incluidos seminarios). Como norma general se conservarán las calificaciones de las distintas partes aprobadas en la convocatoria ordinaria.

Las conductas de plagio, así como el uso de medios ilegítimos en las pruebas de evaluación, serán sancionados conforme a los establecido en la Normativa de Evaluación y la Normativa de Convivencia de la universidad.

## BIBLIOGRAFÍA Y OTROS RECURSOS

## Básica

Buttu, Marco. El gran libro de Python / 1ª ed. Barcelona :Marcombo.,2016.