



centre adscrit a:



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

# **GRADO EN CIENCIAS Y TECNOLOGÍAS APLICADAS AL DEPORTE I AL ACONDICIONAMIENTO FÍSICO**

**GUÍA DOCENTE INFORMÁTICA  
2020-21**

## DATOS GENERALES

ASIGNATURA:	INFORMÀTICA
CÓDIGO:	801716
CURSO:	1R CURSO
CRÉDITOS (ECTS):	6
PROFESOR COORDINADOR:	Dr. Javier Panadero
FECHA ÚLTIMA REVISIÓN:	20/07/2020

## DESCRIPCIÓN GENERAL DEL CURSO

---

Actualmente, el conocimiento y manejo de la informática es imprescindible para poder extraer el máximo rendimiento a todas las tecnologías que nos rodean. Entender cómo funcionan estas tecnologías y cómo se programan es fundamental para la vida profesional en el sector de las ciencias y tecnologías aplicadas.

Conocimientos y habilidades de programación son imprescindibles para analizar, extraer conocimiento y representar de forma visual la gran cantidad de datos que se generan en cualquier evento deportivo y/o de actividad física. En esta asignatura se estudiarán los conceptos fundamentales de los computadores, así como las herramientas de programación necesarias, las cuales permitirán a los estudiantes desarrollar sus propios programas.

## OBJETIVOS GENERALES

---

El objetivo de esta asignatura es que el alumno adquiera los conocimientos y habilidades necesarias para afrontar la resolución de problemas en forma algorítmica, e implantarlas en un computador. Primeramente, se hará un acercamiento a los fundamentos de computadores, con el fin de comprender el funcionamiento básico de un computador, para posteriormente centrarse en la programación funcional, mediante el uso del lenguaje de programación Python. Al finalizar el curso, los alumnos habrán adquirido las competencias necesarias para desarrollar programas complejos, los cuales les permitan solventar problemas del mundo real.

Por otro lado, se potenciará la resolución de trabajos prácticos relacionados con el deporte. De esta manera, los alumnos podrán incorporar los conocimientos adquiridos en esta asignatura a sus futuros proyectos en el mundo laboral.

## CONTENIDOS DEL CURSO

### TEMA 1: FUNDAMENTOS DE COMPUTADORES

---

#### Resultados del aprendizaje

El estudiante después de estudiar el capítulo y realizar los ejercicios, será capaz de:

- Identificar los componentes básicos de un computador y su funcionamiento.
- Entender cómo interpreta y transforma la información un computador.
- Diseñar e interpretar algoritmos y diagramas de flujo sencillos.
- Conocer los diferentes tipos de lenguajes programación existentes.

## Contenido

- 1.1 Componentes básicos de un computador.
- 1.2 Codificación de la información.
- 1.3 Algoritmos y diagramas de flujo.
- 1.4 Lenguajes de programación

## TEMA 2: INTRODUCCIÓN A PYTHON

---

### Resultados del aprendizaje

El estudiante después de estudiar el capítulo y realizar los ejercicios, será capaz de:

- Conocer los principios básicos del lenguaje de programación Python.
- Seleccionar el entorno de desarrollo según las necesidades.
- Codificar y ejecutar programas utilizando el entorno web Jupyter.

### Contenido

- 2.1 ¿Qué es Python?
- 2.2 Entornos de desarrollo para Python.
- 2.3 Jupyter e IPython.
- 2.4 Palabras reservadas y comentarios.
- 2.5 Estructura básica de programa Python
- 2.6 Codificación y ejecución de programas Python utilizando Jupyter.

## TEMA 3: ELEMENTOS BÁSICOS DEL LENGUAJE

---

### Resultados del aprendizaje

El estudiante después de estudiar el capítulo y realizar los ejercicios, será capaz de:

- Definir variables y constantes.
- Realizar operaciones aritméticas y lógicas.
- Utilizar la librería matemática Math
- Operar con cadenas de caracteres.

### Contenido

- 3.1 Variables y constantes.
- 3.2 Tipos de datos simples.
- 3.3 Operadores aritméticos.
- 3.4 Operadores unitarios.
- 3.5 Operadores lógicos.
- 3.6 Librería Math
- 3.7 Cadenas.
- 3.8 Operaciones con cadenas.

## TEMA 4: ESTRUCTURAS DE CONTROL DE FLUJO

---

### Resultados del aprendizaje

El estudiante después de estudiar el capítulo y realizar los ejercicios, será capaz de:

- Identificar los diferentes tipos de sentencias condicionales e iterativas.
- Desarrollar programas que contengan estructuras de flujo condicionales e iterativas.

- Utilizar las diferentes sentencias de control de estructuras iterativas.

**Contenido**

- 4.1 Estructuras de control de flujo condicionales.
- 4.2 Operadores de comparación
- 4.3 Estructuras de control iterativas.
- 4.4 Control de estructuras iterativas.
- 4.5 Función *range*.

**TEMA 5: ESTRUCTURAS DE DATOS**

---

**Resultados del aprendizaje**

El estudiante después de estudiar el capítulo y realizar los ejercicios, será capaz de:

- Conocer los diferentes tipos de datos complejos de Python.
- Identificar qué tipo complejo utilizar según los datos a procesar.
- Manejar Operaciones, funciones y métodos para trabajar con listas, tuplas y diccionarios.

**Contenido**

- 5.1 Listas.
- 5.2 Operaciones con listas.
- 5.3 Listas de compresión
- 5.4 Funciones y métodos para listas.
- 5.5 Tuplas.
- 5.6 Funciones y métodos para tuplas.
- 5.7 Diccionarios.
- 5.8 Funciones y métodos para diccionarios.

**TEMA 6: FUNCIONES**

---

**Resultados del aprendizaje**

El estudiante después de estudiar el capítulo y realizar los ejercicios, será capaz de:

- Identificar cuando es necesario la creación de funciones.
- Saber definir funciones e invocarlas.
- Dominar el paso de parámetros y retorno de valores.
- Implementar funciones recursivas.

**Contenido**

- 6.1 Definición de funciones.
- 6.2 Llamada de funciones.
- 6.3 Alcance de las variables
- 6.4 Paso de parámetros.
- 6.5 Retorno de valores.
- 6.6 Funciones anónimas.
- 6.7 Llamada recursiva.

## TEMA 7: MÓDULOS Y LIBRERÍAS

---

### Resultados del aprendizaje

El estudiante después de estudiar el capítulo y realizar los ejercicios, será capaz de:

- Crear, invocar y trabajar con módulos de Python.
- Conocer y utilizar las diferentes librerías matemáticas y científicas.

### Contenido

- 7.1 Creación e invocación de módulos.
- 7.2 Librerías de análisis matemático.
- 7.3 Librerías de representación gráfica.
- 7.4 Librerías de análisis estadístico.

## TEMA 8: INTRODUCCIÓN A LA ORIENTACIÓN A OBJETOS

---

### Resultados del aprendizaje

El estudiante después de estudiar el capítulo y realizar los ejercicios, será capaz de:

- Identificar cuando es necesaria la creación y utilización de objetos.
- Definir e instanciar clases en función de necesidades del problema.

### Contenido

- 8.1 Pensar en objetos.
- 8.2 Definición de clases.
- 8.3 Propiedades y métodos de los objetos.
- 8.4 Instanciación y utilización objetos.

## METODOLOGÍA DE ENSEÑANZA Y APRENDIZAJE

---

Se basa en clases prácticas participativas complementadas con lectura por adelantado de los temas, ejercicios de clase y los trabajos en casa. Con las prácticas y los trabajos en casa se espera reafirmar los conceptos y procedimientos de cálculo.

## EVALUACIÓN

---

De acuerdo con el Plan Bolonia, el modelo premia el esfuerzo constante y continuado del estudiantado. Un 40% de la nota se obtiene de la evaluación continua de las actividades dirigidas y el 60% porcentaje restante, del examen final presencial. El examen final tiene dos convocatorias.

La nota final de la asignatura (NF) se calculará a partir de la siguiente fórmula:

- $NF = \text{Nota Examen Final} \times 60\% + \text{Nota Evaluación Continuada} \times 40\%$
- Nota mínima del examen final para calcular la NF será de 40 puntos sobre 100.
- La asignatura queda aprobada con una NF igual o superior a 50 puntos sobre 100.

## BIBLIOGRAFÍA BÁSICA

---

- [1] Nilo Ney Coutinho. Introducción a la programación con Python: Algoritmos y lógica de programación para principiantes (2017).
- [2] Jose Ojedas Rojas. Programando con Python: Introducción a Python (2017).

- [3] Nick Samoylov. Introduction to Programming: Learn to program with data structures, algorithms, and logic (2018).

### **WEBGRAFÍA / BLOGS / ETC.**

---

- [1] Código Facil. Tutorial Python. Recuperado de: <https://www.youtube.com/watch?v=CjmzDHMHxwU> (2018).
- [2] Universidad de Alicante. Introducción a Python e IPython. Recuperado de : <https://www.youtube.com/watch?v=ox09Jko1ErM> (2018)