

# Guía Docente

## DATOS DE IDENTIFICACIÓN

Titulación:	Ingeniería Matemática
-------------	-----------------------

Rama de Conocimiento:	Ingeniería y Arquitectura
-----------------------	---------------------------

Facultad/Escuela:	Escuela Politécnica Superior
-------------------	------------------------------

Asignatura:	Programación II
-------------	-----------------

Tipo:	Formación Básica
-------	------------------

Créditos ECTS:	6
----------------	---

Curso:	2
--------	---

Código:	4955
---------	------

Periodo docente:	Tercer semestre
------------------	-----------------

Materia:	Informática
----------	-------------

Módulo:	Formación Básica
---------	------------------

Tipo de enseñanza:	Presencial
--------------------	------------

Idioma:	Castellano
---------	------------

Total de horas de dedicación del alumno:	150
--	-----

Equipo Docente	Correo Electrónico
Juan Pueyo Candil	j.pueyo.prof@ufv.es

## DESCRIPCIÓN DE LA ASIGNATURA

La asignatura Programación II 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 primer semestre del segundo curso de los estudios de Grado en Ingeniería Matemática, y requiere de una

dedicación de 150 horas por parte del alumno.

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.

## OBJETIVO

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. 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.

## CONOCIMIENTOS PREVIOS

Se recomienda disponer de los conocimientos básicos de álgebra y aritmética. Es muy recomendable haber superado la asignatura Programación I.

## CONTENIDOS

- Tema 1. Entornos de programación en Python.
- Tema 2. Elementos del lenguaje Python.
  - Estructura de un programa.
  - Variables, identificadores, expresiones y operadores.
- Tema 3. Entrada y salida de datos.
  - Entrada por teclado.
  - Salida por pantalla.
- Tema 4. Tipos de datos en Python.
  - Clases de tipado.
  - Números, cadenas, booleanos.
- Tema 5. Estructuras de control de flujo.
  - Estructuras condicionales.
  - Estructuras iterativas.
- Tema 6. Funciones en Python.
  - Librerías internas y externas.
  - Funciones internas y métodos.
  - Anotaciones y parámetros.
  - Variables locales y globales.
  - Paso de parámetros.
- Tema 7. 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 8. Cadenas de Caracteres.

- Cadenas de caracteres.
  - Funciones y métodos para su tratamiento.
- Tema 9. Tratamiento de Excepciones.
- Errores sintácticos y Excepciones.
  - Control de excepciones.
- Tema 10. Archivos.
- Lectura básica de archivos.
  - Escritura básica en archivos.
- Tema 11. El paradigma Orientado a Objetos.
- Tema 12. Encapsulación, Herencia y Polimorfismo.
- Tema 13. Interfaces gráficas.

## 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.

\* Las actividades formativas, así como la distribución de los tiempos de trabajo, pueden verse modificadas y adaptadas en función de los distintos escenarios establecidos siguiendo las indicaciones de las autoridades sanitarias.

## DISTRIBUCIÓN DE LOS TIEMPOS DE TRABAJO

ACTIVIDAD PRESENCIAL	TRABAJO AUTÓNOMO/ACTIVIDAD NO PRESENCIAL
60 horas	90 horas
Lección Expositiva 12h Laboratorio 34h Presentación y Defensa de Trabajos 4h Tutorías 4h Evaluación 6h	Estudio y Trabajo Individual 60h Trabajo en Grupo 30h

## 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

Capacidad de resolver problemas con iniciativa, toma de decisiones, creatividad, razonamiento crítico y rigor de pensamiento, y de comunicar y transmitir conocimientos, habilidades y destrezas en el campo de la Ingeniería Matemática.

### Competencias específicas

Conocimientos básicos sobre el uso y programación de los ordenadores, estructuras de datos, bases de datos y programas informáticos con aplicación en ingeniería.

## 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.

## SISTEMA DE EVALUACIÓN DEL APRENDIZAJE

El sistema de evaluación contempla varios tipos de pruebas, distribuidos de la siguiente manera:

- Prueba parcial de tipo teóricopráctico: 30 %.
- Examen final de tipo teóricopráctico: 40 %.
- Realización de trabajos y prácticas: 25 %.
- Participación e implicación en la asignatura: 5 %.

Aquellos alumnos que estén exentos de la obligación de asistir a clase, bien por segunda matrícula en la asignatura o sucesivas, bien por contar con autorización expresa de la Dirección del Grado, serán evaluados por el mismo tipo de pruebas. El 5 % correspondiente a la participación e implicación se evaluará en base a la entrega de los ejercicios correspondientes en la fecha establecida.

Criterios para superar la asignatura:

- Obtener una nota igual o superior a 5 puntos sobre 10 en la prueba parcial que se realizará a lo largo del cuatrimestre, en cuyo caso, el día del examen oficial, solo se presentará al examen que corresponda a la última parte de la asignatura. En dicho examen también tendrá que obtener un mínimo de 5 puntos sobre 10. Esta prueba es liberatoria solo de cara al examen de la convocatoria ordinaria. En la convocatoria extraordinaria siempre tendrá que presentarse a la totalidad de la materia.

- Obtener una nota igual o superior a 5 puntos sobre 10 en la media obtenida de las prácticas y trabajos.
- La entrega de prácticas y trabajos no es obligatoria, pero tendrán una calificación de 0 de cara al cálculo de la media correspondiente.
- Las prácticas tendrán una fecha de entrega. Si se entregan fuera de fecha, tendrán una penalización de un 30%, siempre y cuando se entreguen como tarde, una semana después de la fecha establecida.
- La asignatura se considerará aprobada en la convocatoria correspondiente si la nota final es igual o superior a 5 puntos sobre 10.
- En el caso de no tener superada alguna de las partes de la asignatura (pruebas teórico-prácticas por un lado o trabajos y prácticas por otro) en la convocatoria ordinaria, el alumno tendrá que presentarse a aquellas partes que no haya superado en la convocatoria extraordinaria.
- La participación en clase solo será evaluada cuando la asistencia registrada del alumno sea de al menos el 80%.

**Notas adicionales:**

- La entrega del examen por parte del profesor el día del examen implica automáticamente la presentación a la convocatoria correspondiente.
- A efecto de cómputo de convocatorias en una asignatura, solamente se contabilizarán como consumidas aquellas en las que el alumno se haya presentado a todas las pruebas de evaluación, o a una parte de las mismas, siempre que su peso en la nota final supere el 50 %, aunque no se presente al examen final. Se entenderá que un alumno se ha presentado a una prueba aunque la abandone una vez comenzada la misma. La condición de No Presentado en la convocatoria extraordinaria estará ligada a la no asistencia o entrega de ninguna prueba, práctica o trabajo que esté pendiente.
- Cualquier tipo de fraude o plagio por parte del alumno en una actividad evaluable será sancionado e implicará un 0 en la calificación de esa parte de la asignatura, anulando la convocatoria en curso. La situación, además, será comunicada a la Dirección de la Carrera, que a su vez comunicará a Secretaría General, siguiendo el protocolo establecido en la universidad.

\* En el caso de que las recomendaciones sanitarias obliguen a volver a un escenario donde la docencia haya que impartirla exclusivamente en remoto, se mantendrá todo lo indicado anteriormente.

\*\* Los exámenes se realizarán de manera presencial con un ordenador en el aula, siempre y cuando la situación sanitaria lo permita, pudiendo ser modificados con el objetivo de cumplir las indicaciones dadas por las autoridades.

## BIBLIOGRAFÍA Y OTROS RECURSOS

### Básica

CHAZALLET, S. Python 3: los fundamentos del lenguaje (2ª edición). Ediciones ENI. 2016. ISBN: 9782409006142.

LUTZ, M. Learning Python (5TH ED.). Editorial O REILLY & ASSOCIATES. 2013. ISBN: 9781449355739.

Material disponible en el Aula Virtual.

### Complementaria

L. Joyanes Aguilar, Fundamentos de programación: algoritmos, estructuras de datos y objetos, 4ª edición. Madrid: Editorial McGraw-Hill Interamericana, 2008. ISBN: 978-84-481-6111-8.

L. Joyanes Aguilar, L. Rodríguez Baena y M. Fernández Azuela, Fundamentos de Programación: Libro de Problemas, 2ª edición. Madrid: Editorial McGraw-Hill Interamericana, 2003. ISBN: 9788448139865.

Muñoz C., Niño A., Vizcaíno A. Introducción a la programación con orientación a objetos. Pearson Educación. 2002. ISBN: 9788420534404