

# Guía Docente

## DATOS DE IDENTIFICACIÓN

Titulación:	Grado en Análisis de Negocios
-------------	-------------------------------

Rama de Conocimiento:	Ciencias Sociales y Jurídicas
-----------------------	-------------------------------

Facultad/Escuela:	Derecho, Empresa y Gobierno
-------------------	-----------------------------

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

Tipo:	Optativa
-------	----------

Créditos ECTS:	6
----------------	---

Curso:	3
--------	---

Código:	5352
---------	------

Periodo docente:	Quinto semestre
------------------	-----------------

Materia:	Informática Aplicada al Análisis de Negocios / IT applied to Business Analytics
----------	---

Módulo:	Formación disciplinar
---------	-----------------------

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

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

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

Equipo Docente	Correo Electrónico
Roberto José Vázquez Lucerga	roberto.vazquez@ufv.es

## DESCRIPCIÓN DE LA ASIGNATURA

Como continuación de la asignatura de Programación I, esta asignatura profundiza en el desarrollo de los conocimientos de programación y la resolución de problemas mediante el desarrollo de programas informáticos.

Se cubrirán aspectos como el uso avanzado de estructuras de datos objetos, funciones de orden superior, clases, errores y gestión de excepciones (continuación del Programación I), pruebas, concurrencia y paralelismo, conexiones a bases de datos, programación reactiva y de red.

Se trabajará en lenguaje de programación Python 3

## OBJETIVO

El principal objetivo de esta asignatura consiste en que el estudiante profundice en el análisis, diseño, implementación y prueba de software usando el paradigma de orientación a objetos. Al finalizar la asignatura, el alumno será capaz de emplear métodos, técnicas y herramientas para la construcción de aplicaciones eficientes y robustas.

## CONOCIMIENTOS PREVIOS

Los propios de acceso a Grado. Es altamente recomendable haber superado la asignatura de Introducción a la Programación I, y al menos conocer la estructura de un programa en Python 3, (operadores, identificadores, sentencias de control, funciones), tener conocimientos básicos sobre las clases, herencia, polimorfismo, y del manejo básico de las interfaces gráficas.

## CONTENIDOS

01. Modelo de ejecución y datos en Python
02. Funciones II: Funciones de orden superior, decoradores, recursión
03. Clases II: propiedades, descriptores, sobrecarga de operadores
04. Estructuras de datos II: Listas y diccionarios II (comprensión, doblemente enlazadas, colas de prioridad, ChainMaps, defaultdict, OrderedDict), contadores, tuplas nombradas.
05. Iteradores y generadores
06. Mapas, filtros y reducciones
07. Gestión de errores II: Excepciones II, pruebas y registros (logging)
08. Concurrencia y paralelismo
09. Módulos y paquetes
10. Librerías de uso común (pandas, numpy, scipy, matplotlib/seaborn)
11. Acceso a base de datos y programación de red

## ACTIVIDADES FORMATIVAS

### Presenciales:

- Combinación de lecciones expositivas con clases prácticas, de manera que se favorezca la participación del alumno y la interacción alumno- profesor y alumno-alumno como vía para fomentar el aprendizaje colaborativo y la capacidad de autoaprendizaje, todo ello mediante estrategias de resolución de problemas y metodologías de intervención (aprendizaje basado en problemas, ABP)

### No presenciales:

- Estudio individual, que permitirá trabajar en la fijación de los conceptos teóricos abordados en las clases expositivas correspondientes. El profesor planteará problemas para que se piensen fuera de clase y se discutan posteriormente en el aula (Flipped Learning)
- Prácticas intercuatrimestrales mediante las cuales el alumno podrá poner en práctica los conocimientos adquiridos y recibirá feedback continuo de su progreso.

El trabajo colaborativo se hará usando herramientas colaborativas tanto en Office 365 como en entorno Git (sistema de código abierto para el trabajo y control de versión para el desarrollo de software). Para ello el alumno deberá crearse una cuenta en gitlab.com. Si los requerimientos de las autoridades sanitarias exigen el mantenimiento de distancias en el aula, será necesario utilizar también estas herramientas colaborativas en clase.

Todo el estudio y trabajo realizado por el alumno será supervisado y guiado por el profesor mediante tutorías, individuales o en grupo. El alumno se pondrá en contacto con el profesor para concertar citas (presenciales o virtuales).

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.

En caso de que las clases presenciales no sean posibles o sean posibles de manera limitada, se sustituirán las lecciones expositivas presenciales por sesiones síncronas a través del aula virtual, en los mismos horarios que las clases presenciales. La asistencia y participación en estas sesiones síncronas se considerará de la misma

manera que las presenciales que se han sustituido.

Las practicas intercuatrimestrales y los exámenes serán sometidos a control de plagio. A cualquier estudiante se le puede pedir, en cualquier momento, la reproducción, con datos idénticos o similares, del código enviado y/o su explicación o defensa. Se considerará que un estudiante que no es capaz de reproducir o explicar el código ha enviado un trabajo que no es de elaboración propia, contraviniendo lo establecido esta guía docente para todos los trabajos y exámenes. Quedará a discreción del Grado la aplicación de las normas relativas al plagio contenidas en la Normativa de Convivencia en estos casos.

## 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

Capacidad de organización, sistematización y planificación en la identificación de problemas, pautas y modelos en el contexto del big data

Capacidad para el cumplimiento de objetivos, resolución de problemas y toma de decisiones en un entorno de datos masivos tanto cuantitativos como cualitativos

### Competencias específicas

Conocer, comprender y aplicar los fundamentos y las herramientas de la programación para el uso y explotación de la información, garantizando su seguridad e integridad

Saber manejar herramientas cuantitativas e informáticas para la toma de decisiones

Ser capaz de conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en el ámbito del big data

## RESULTADOS DE APRENDIZAJE

Desarrolla programas avanzados basados en el paradigma de programación orientada a objetos, desde el diseño y definición de las clases y objetos que los forman, hasta su implementación y posterior depuración con un lenguaje y entorno de desarrollo determinado.

Desarrolla programas estables y fáciles de mantener: documenta, integra programas de pruebas, mantiene requerimientos de módulos, utiliza correctamente las excepciones y los registros (logging).

Conoce librerías de uso común para el manejo de tablas, arrays multidimensionales y de computación científica y técnica.

Es capaz de crear programas con que trabajan con múltiples procesos o múltiples hilos (threads) y coordinar su ejecución.

Desarrolla programas que acceden a bases de datos e interaccionan vía APIs.

## SISTEMA DE EVALUACIÓN DEL APRENDIZAJE

El sistema de evaluación contempla cuatro tipos de pruebas:

[1] Examen teórico-práctico: presenta un peso del 30% en la nota final.

[2] Examen teórico-práctico: presenta un peso del 30% en la nota final.

[3] Prácticas y otros trabajos relacionados con la asignatura: presenta un peso del 35% en la nota final.

[4] Participación en clase e implicación en la asignatura: presenta un peso del 5% en la nota final, siendo requisito imprescindible haber asistido como mínimo al 80% de las sesiones. En caso contrario este tipo de prueba se calificará con 0 puntos.

En las tres primeras pruebas es necesario obtener un mínimo de 5 puntos sobre 10 para poder aprobar la asignatura. En la [3] es necesario tener un mínimo de 5 puntos en cada práctica y/o trabajo para poder aprobar la asignatura.

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% de la participación en clase podrán obtenerlo asistiendo al menos a tres tutorías con el profesor responsable de la asignatura.

Recuperación en convocatoria ordinaria:

Los alumnos que no hayan alcanzado la nota mínima en el examen, podrán optar a una recuperación al final del semestre.

Recuperación en convocatoria extraordinaria:

Los alumnos que no hayan alcanzado la nota mínima en el examen, habiendo suspendido por tanto en la convocatoria ordinaria, podrán optar a una recuperación en la convocatoria extraordinaria.

En ambas recuperaciones (ordinaria y extraordinaria) el alumno se presentará solo a las partes que tenga evaluadas por debajo de 5.

La nota ponderada de la evaluación continua será un valor entre 0 y 10 y se calculará como sigue:  $0,30*[1] + 0,30*[2] + 0,35*[3] + 0,05*[4]$ .

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.

[1] Examen a mitad de cuatrimestre de carácter teórico-práctico, con cuestiones cortas, preguntas de desarrollo y ejercicios prácticos de programación. Este examen representará un 30% de la calificación final y evaluará la primera mitad del temario. El examen se puntuará de 0 a 10, repartiendo esta puntuación de manera equitativa entre todos los ejercicios y apartados, salvo que se indique lo contrario. Se evaluará el planteamiento de los problemas, así como la corrección, presentación e interpretación de los resultados obtenidos.

[2] Examen de carácter teórico-práctico a final de curso con preguntas de desarrollo y ejercicios prácticos de programación que representará un 30% de la calificación final. Se evaluará la segunda mitad del temario, si bien, debido a la relación de todos los conceptos vistos en la asignatura, se recomienda encarecidamente repasar los contenidos de la primera parte. También se puntuará de 0 a 10, repartiendo esta puntuación de manera equitativa entre todos los ejercicios y apartados, salvo que se indique lo contrario. Se evaluará el planteamiento de los problemas, así como la corrección, presentación e interpretación de los resultados obtenidos. Ambas pruebas [1] y [2] se realizarán sin transparencias, apuntes, libros ni cualquier otro material relacionado con la asignatura.

[3] Realización, y entrega, de las prácticas de programación y otros trabajos (35% de la calificación final). Tareas individuales de diversa índole, incluyendo prácticas de programación y otros ejercicios relacionados con la asignatura, algunas entregas podrán ser realizadas en grupos según lo indicado por el profesor. En el caso que el profesor estime oportuno, la calificación quedará afectada por la defensa oral del trabajo, al alza o a la baja, para asegurar la autoría de los trabajos.

[4] Participación e implicación: 5% de la calificación final. Se evaluarán los ejercicios y otras actividades en grupo, el interés mostrado por el alumno, concretamente se computará el índice de asistencia a tutorías tanto individuales o grupales, el grado de participación activa en las clases mediante la respuesta a preguntas del profesor, el estudio de temas avanzados no vistos en clase, la recopilación de noticias aparecidas en los medios de comunicación relacionadas con la asignatura, etc. La calificación de este apartado será un valor numérico entre 0 y 10. Aunque esta nota sea inferior a 5, no se podrá optar a recuperación.

Todas las pruebas susceptibles de evaluación estarán supeditadas a lo establecido en la normativa de evaluación de la universidad francisco de vitoria.

Las conductas que defrauden el sistema de comprobación del rendimiento académico, tales como plagio de trabajos o copia en exámenes son consideradas faltas graves según el artículo 7 de la normativa de convivencia de la UFV y serán aplicadas las sanciones oportunas como recoge el artículo 9 del mismo documento. A estos efectos, se considerará "plagio" cualquier intento de defraudar el sistema de evaluación, como copia en ejercicios, exámenes, practicas, trabajos o cualquier otro tipo de entrega, bien de otro compañero, bien de materiales o dispositivos no autorizados, con el fin de hacer creer al profesor que son propios.

En el caso de que las recomendaciones sanitarias nos obliguen a volver a un escenario donde la docencia haya que organizarla exclusivamente en remoto, las clases se impartirán de manera síncrona siempre que sea posible, y los exámenes y las prácticas se realizarán en remoto con las herramientas previstas en al Universidad. No se prevén cambios en el sistema de evaluación o cambios en los porcentajes de cada elemento

Los exámenes se realizarán de manera presencial 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

John Hunt (2019). A Beginners Guide to Python 3 Programming (Springer). ISBN 978-3-030-20289-7

John Hunt (2019). Advanced Guide to Python 3 Programming (Springer). ISBN 978-3-030-25942-6

### Complementaria

<https://docs.python.org/3/tutorial/>

<https://docs.python.org/3/library/index.html#library-index>

<https://docs.python.org/3/reference/index.html#reference-index>