



Programación II  
Grado en Análisis de  
Negocio



UNIVERSIDAD  
NEBRIJA

## GUÍA DOCENTE

**Asignatura:** Programación II

**Titulación:** Grado en Análisis de Negocio

**Carácter:** Obligatorio

**Idioma:** Español

**Modalidad:** Presencial

**Créditos:** 6

**Curso:** 2º

**Semestre:** 1º

**Profesores/Equipo Docente:** Rubén García Maezo

### 1. RESULTADOS DEL PROCESO DE FORMACIÓN Y APRENDIZAJE

#### 1.1. Resultados de aprendizaje

##### 1.1.1. Conocimientos

- **K5-** Conocer el funcionamiento de los lenguajes de programación más utilizados en el contexto de análisis de datos para los negocios.

##### 1.1.2. Habilidades

- **H5-** Producir informes y visualizaciones eficientes para la toma de decisiones en la empresa a partir de los datos.
- **H6-** Construir programas en los lenguajes más utilizados en el análisis de negocios, adecuados para los distintos problemas de modelización, análisis y visualización de datos.
- **H9-** Solucionar un problema identificando las opciones y ventajas de utilizar técnicas de procesamiento del lenguaje para afrontarlo dentro del análisis de negocios.

##### 1.1.3. Competencias

- **C1-** Recurrir en la práctica profesional a las herramientas e instrumentos de naturaleza cuantitativa precisos para la obtención, diagnóstico y análisis de la información empresarial en un entorno económico y social.
- **C2-** Integrar los conocimientos adquiridos a lo largo del título y demostrar una comprensión de los aspectos teóricos y prácticos de relevancia para un problema real del ámbito empresarial.
- **C3-** Desenvolverse en situaciones que requieran el desarrollo de nuevas soluciones que puedan articularse a través del tratamiento de grandes cantidades de datos en el ámbito profesional de la empresa.
- **C4-** Comunicar a todo tipo de audiencias, especializadas o no, de manera clara y precisa conocimientos, metodologías y soluciones para los problemas que se plantean en el mundo empresarial.

#### Competencias Transversales:

- **CT1:** Actuar con ética y responsabilidad profesional ante los desafíos sociales, ambientales y económicos, teniendo como referentes los principios y valores democráticos y los Objetivos de Desarrollo Sostenible.
- **CT2:** Valorar las consecuencias éticas de las decisiones a tomar en una situación concreta, considerando el impacto en la sociedad y la responsabilidad en la práctica profesional.
- **CT3:** Emitir juicios informados sobre el tratamiento de la sostenibilidad y del cambio climático.

- CT4: Demostrar concienciación sobre el respeto a la diversidad y a los principios de accesibilidad universal y diseño para todas las personas.
- CT5: Contribuir en el diseño, desarrollo y ejecución de soluciones que den respuesta a demandas sociales, teniendo en cuenta como referente los Objetivos de Desarrollo Sostenible.

## **2. CONTENIDOS**

### **2.1. Requisitos previos**

No hay requisitos previos

### **2.2. Descripción de los contenidos**

El curso de "Programación II" para Business Analytics está diseñado para dotar a los estudiantes con un dominio profundo de Python, enfocándose en su aplicación para el análisis de datos y la toma de decisiones empresariales. Es ideal para aquellos con una base en programación que desean especializarse en las dinámicas analíticas de Python dentro de contextos comerciales reales.

Iniciamos con un módulo sobre entornos de programación, donde se estudian las configuraciones de Python, instalaciones y entornos virtuales, esenciales para cualquier proyecto analítico, además de una revisión de IDEs como PyCharm, VSCode o Jupyter Notebooks, complementado con librerías clave como Pandas y NumPy para el manejo avanzado de datos. Seguimos con un módulo sobre funciones en Python, enfocándonos en la escritura de código limpio y eficiente a través del uso de funciones, incluyendo lambdas y técnicas de manejo avanzado de datos como la recursión.

El curso también aborda estructuras de datos fundamentales y avanzadas necesarias para la manipulación eficiente de grandes volúmenes de datos, y técnicas de manejo de excepciones para mejorar la fiabilidad de las aplicaciones de software. Profundizamos en la programación orientada a objetos, explorando desde la encapsulación hasta relaciones complejas entre clases, lo que es vital para el diseño de sistemas robustos y escalables.

Además, se examinan los iteradores y generadores para el manejo eficiente de datos, y se discuten técnicas avanzadas de procesamiento de datos mediante map, filter y reduce, especialmente útiles en pipelines de datos. El curso introduce también la concurrencia y el paralelismo para optimizar el rendimiento de las aplicaciones de análisis de datos.

Finalmente, se enseña cómo organizar código en módulos y paquetes, gestionando dependencias efectivamente, y cómo crear interfaces gráficas y visualizaciones avanzadas utilizando herramientas como Tkinter, PyQt, Matplotlib y Seaborn, además de plataformas interactivas como Dash y Bokeh para el desarrollo de dashboards. Este enfoque integral prepara a los estudiantes para aplicar técnicas de programación avanzadas en situaciones analíticas complejas, equipándolos para enfrentar desafíos en diversas industrias y roles analíticos.

### **2.3. Contenido detallado Programacion ii**

Presentación de la asignatura.  
Explicación de la Guía Docente.

1. Módulo 1: Entornos de Programación en Python
  - a) Introducción a Python.
  - b) Instalación y configuración: Instalación de Python, configuración de entornos virtuales.
  - c) IDE y Editores: Comparación y uso de IDEs como PyCharm, VSCode o Jupyter Notebooks.
  - d) Herramientas para Business Analytics: Librerías y frameworks específicos (Pandas, NumPy).
2. Módulo 2. Funciones en Python
  - a) Definición y uso de funciones: Sintaxis básica, argumentos y retorno de valores.
  - b) Funciones anónimas (lambda): Uso y casos de uso.
  - c) Funciones integradas: Principales funciones predefinidas.
  - d) Ámbito de las variables y recursión: Conceptos de alcance y ejemplos de recursión.
3. Módulo 3. Estructuras de Datos
  - a) Listas y Tuplas: Definición, operaciones básicas, comprensión de listas.
  - b) Diccionarios y Conjuntos: Definición, operaciones básicas, comprensión de diccionarios.
  - c) Estructuras de datos avanzadas: Pilas, colas y árboles.
4. Módulo 4. Tratamiento de Excepciones
  - a) Manejo de errores: Bloques try, except, else y finally.
  - b) Creación de excepciones personalizadas: Definición y uso de excepciones propias.
  - c) Buenas prácticas: Técnicas y estrategias para el manejo adecuado de excepciones.
5. Módulo 5. El Paradigma Orientado a Objetos
  - a) Conceptos básicos: Clases y objetos, atributos y métodos.
  - b) Encapsulación: Propiedades y métodos privados y públicos.
  - c) Abstracción: Definición y ejemplos.
  - d) Relaciones entre clases: Asociación, agregación y composición.
6. Módulo 6. Clases, Encapsulación, Herencia y Polimorfismo
  - a) Definición de clases y objetos: Sintaxis y ejemplos prácticos.
  - b) Encapsulación: Acceso y modificación de atributos y métodos.
  - c) Herencia: Tipos de herencia, superclase y subclase, uso de super().
  - d) Polimorfismo: Sobrecarga y sobrescritura de métodos.
7. Módulo 7. Iteradores y Generadores
  - a) Concepto de iterador: Implementación de iter y next.
  - b) Generadores: Uso de la palabra clave yield, creación de generadores.
  - c) Comparación entre iteradores y generadores: Casos de uso y eficiencia.
8. Módulo 8. Mapas, Filtros y Reducciones
  - a) Funciones map, filter y reduce: Definición, sintaxis y ejemplos.
  - b) Uso de funciones lambda: Aplicación en map, filter y reduce.
  - c) Comparación con comprensiones de listas: Casos de uso y eficiencia.
9. Módulo 9. Concurrencia y Paralelismo
  - a) Introducción a la concurrencia: Threads y procesos.
  - b) Librerías y herramientas: threading, multiprocessing, concurrent.futures.
  - c) Sincronización y bloqueo: Locks, Semaphores, Events.
  - d) Programación asíncrona: Introducción a async y await.
10. Módulo 10. Módulos y Paquetes
  - a) Importación de módulos: Importación absoluta y relativa.
  - b) Creación de módulos: Estructura y organización de código en módulos.
  - c) Paquetes en Python: Definición y estructura de paquetes.

- d) Gestión de dependencias: Uso de pip y virtualenv.
11. Módulo 11. Interfaces, Gráficas y Visualización
- a) Introducción a interfaces gráficas: Tkinter, PyQt.
- b) Diseño de interfaces: Componentes básicos, eventos y manejo de eventos.
- c) Visualización de datos: Uso de Matplotlib, Seaborn, Plotly.
- d) Dashboards y aplicaciones interactivas: Introducción a Dash y Bokeh.

## 2.4. Actividades Dirigidas

Durante el curso se podrán desarrollar actividades, prácticas, memorias o proyectos.

Actividad Dirigida 1 (AAD 1): Al finalizar cada unidad didáctica, se podrá chequear los conocimientos relacionados con el temario, así como solicitar la entrega de ejercicios teóricos y prácticos relacionados con el contenido del temario.

Actividad Dirigida 2 (AAD 2): Se realizarán prácticas de laboratorio. El laboratorio dispondrá del material necesario (PC y programas) para el desarrollo de la actividad. También estará permitido el uso de un PC personal.

## 2.5. Actividades formativas

Tipo de actividad modalidad presencial	Horas	Presencialidad %
A1 Clase Magistral/ Fundamentos Teóricos	45	100%
A2 Clases prácticas. Seminarios y talleres	9	100%
A3 Tutoría	9	100%
A4 Trabajos o ejercicios de los estudiantes	18	0%
A5 Actividades a través de recursos virtuales	6	0%
A6 Acceso e investigación sobre contenidos complementarios	6	0%
A7 Estudio individual	51	0%
A13 Evaluación	6	100%

## 3. SISTEMA DE EVALUACIÓN

### 3.1. Sistema de calificaciones

El sistema de calificaciones finales se expresará numéricamente del siguiente modo:

- 0 - 4,9 Suspenso (SS)
- 5,0 - 6,9 Aprobado (AP)
- 7,0 - 8,9 Notable (NT)
- 9,0 - 10 Sobresaliente (SB)

La mención de "matrícula de honor" podrá ser otorgada a alumnos que hayan obtenido una calificación igual o superior a 9,0.

### 3.2. Criterios de evaluación

Convocatoria ordinaria

Modalidad: Presencial

Sistemas de evaluación	Porcentaje
S1 Asistencia y participación en clase	10%
S2 Presentación de trabajos y proyectos (Prácticas individuales y trabajo en equipo)	30%
S3 Prueba parcial presencial (escrita/presentación trabajo)	10%
S4 Examen final o trabajo final presencial	50%

Convocatoria extraordinaria

Modalidad: Presencial

Sistemas de evaluación	Porcentaje
S2 Presentación de trabajos y proyectos (Prácticas individuales y trabajo en equipo)	30%
S4 Examen final o trabajo final presencial	70%

Restricciones y explicación de la ponderación: Para poder hacer media con las ponderaciones anteriores será necesario obtener al menos una calificación de 5 en la prueba final.

Asimismo, será potestad del profesor solicitar y evaluar de nuevo las prácticas o trabajos escritos, si estos no han sido entregados en fecha, no han sido aprobados o se desea mejorar la nota obtenida en ambas convocatorias.

En todo caso, la superación de cualquier materia/asignatura está supeditada a aprobar las pruebas finales presenciales e individuales correspondientes.

### **3.3. Restricciones**

Calificación mínima

Para poder hacer media con las ponderaciones anteriores es necesario obtener al menos una calificación de 5 en la prueba final.

Asistencia

El alumno que, injustificadamente, deje de asistir a más de un 25% de las clases presenciales podrá verse privado del derecho a examinarse en la convocatoria ordinaria.

Normas de escritura

Se prestará especial atención en los trabajos, prácticas y proyectos escritos, así como en los exámenes tanto a la presentación como al contenido, cuidando los aspectos gramaticales y ortográficos. El no cumplimiento de los mínimos aceptables ocasionará que se resten puntos en dicho trabajo o examen.

### **3.4. Advertencia sobre plagio**

La Universidad Antonio de Nebrija no tolerará en ningún caso el plagio o copia. Se considerará plagio la reproducción de párrafos a partir de textos de auditoría distinta a la del estudiante (Internet, libros, artículos, trabajos de compañeros...), cuando no se cite la fuente original de la que provienen. El uso de las citas no puede ser indiscriminado. El plagio es un delito.

En caso de detectarse este tipo de prácticas, se considerará Falta Grave y se podrá aplicar la sanción prevista en el Reglamento del Alumno.

#### 4. BIBLIOGRAFÍA

##### Bibliografía básica

Schurmann, N. (2024). Ultimate Python: de cero a experto: Un libro para principiantes fácil de seguir. Independently Published.

Jiménez, O. R. (2021). Python a fondo. Marcombo Editorial.

Severance, Charles R. (2020). Python para Todos: Explorando la información con Python 3. Independently published.

Chazallet, S. (2020). Python 3: los fundamentos del lenguaje. Eni Editorial.

Cuevas, A. (2018). Aplicaciones gráficas con python 3. RA-MA Editorial.

Hinojosa Gutiérrez, Á. (2015). Python paso a paso: ( ed.). RA-MA Editorial.

Lee, K. D. (2011). Python programming fundamentals. London: Springer.

Martin, R. C. (2009). Clean code: a handbook of agile software craftsmanship. Pearson Education.

##### Bibliografía recomendada

John V. Guttag (2021). Introduction to computation and programming using Python: With application to computational modeling and understanding data.

Torres, J. (2020) Python Deep Learning: Introducción práctica con Keras y TensorFlow 2.

Sebastian Raschka y Vahid Mirjalili (2019). Python Machine Learning: aprendizaje automático y aprendizaje profundo con Python, scikit-learn y TensorFlow.

Nolasto Valenzuela, J.S (2018) Python aplicaciones prácticas. RA-MA.

##### Software recomendado

Thonny

Visual Studio Code

Notepad++