

Fundación **VASS**

Programación con Python A2 (25 horas)

Índice

1. Contextualización
2. Objetivos de aprendizaje
3. Contenidos
4. Metodología de aprendizaje
5. Competencias
6. Calendario
7. Evaluación

1. Contextualización

En el panorama tecnológico actual, la verdadera ventaja competitiva no radica solo en escribir código, sino en la capacidad de diseñar soluciones robustas y escalables. Este nivel avanzado de Python está diseñado para transformar al usuario de un ejecutor de scripts en un arquitecto de software con total autonomía.

A través de este programa, profundizaremos en los pilares que definen el desarrollo profesional:

- **Estrategias algorítmicas y construcción de código:** optimización de procesos para resolver problemas complejos con eficiencia computacional.
- **Diseño de software y evolución:** aplicación de patrones de diseño y principios de mantenibilidad para que el software crezca sin perder calidad.
- **Representación de datos y librerías especializadas:** dominio técnico de las estructuras de datos y el ecosistema de librerías avanzadas que lideran la industria.

2. Objetivos de aprendizaje

Saber programar en Python te permitirá automatizar tareas, analizar datos y desarrollar páginas web, entre otras muchas utilidades. Estos son los objetivos de la formación:

- **Codificar con autonomía**, traduciendo lógica de negocio a código eficiente sin dependencia constante de referencias externas.
- **Implementar estructuras de datos y algoritmos** óptimos para resolver problemas complejos de computación.
- **Aplicar principios de diseño avanzado** para crear software escalable, legible y fácil de mantener.
- **Gestionar el ciclo de vida de un proyecto**, desde la manipulación de datos hasta la administración de dependencias y librerías.

3. Contenidos

- **MÓDULO 1: Procesamiento de datos en Python**
 - Introducción: preprocesamiento de datos
 - Primeros pasos
 - Análisis descriptivo
 - Identificación de tipos de variables
 - Integración de datos
 - Transformación de datos

- Limpieza de datos
- Técnicas de escalado
- Reducción de dimensiones
- Reducción de muestras
- Discretización

● **MÓDULO 2: Análisis de datos en Python**

- Introducción al aprendizaje automático (ML)
- Aprendizaje no supervisado
- Reducción de dimensionalidad
- Otros tipos de aprendizaje automático
 - Aprendizaje por refuerzo
 - Aprendizaje semisupervisado
- Selección de modelos y validación cruzada
- Ejemplo práctico: datos de flores iris
 - Análisis exploratorio de datos
 - Reducción dimensionalidad
- Clasificación
- Algoritmos de agrupamiento o clustering
- Validación de modelos

● **MÓDULO 3: Visualización de datos**

- Un mapa de las librerías de visualización de Python
 - Visualización de alto nivel
 - Visualización científica
 - Visualización de grafos
 - Visualización de geolocalización
 - Visualizaciones interactivas
- Visualización de alto nivel
 - Matplotlib
 - Altair
 - Plotnine
 - Gráficas con Seaborn

- **MÓDULO 4: Captura de datos (transversal)**

- Descarga directa de datos
 - Petición de APIs
 - Concepto de API
 - Peticiones y respuestas HTTP
 - JSON y XML
 - Acceso a API con librerías de Python
 - Obtención de datos a partir de web crawling
 - BeautifulSoup
 - Scrapy
- **IA para Python**
 - **Soft skills (irá intrínsecamente en cada uno de los apartados)**

Nota: La propuesta a nivel contenido recoge los niveles A2 y B1 de la [Escuela de Programación de la UOC](#) reduciendo la carga lectiva.

4. Metodología de aprendizaje

La UOC se caracteriza por un modelo educativo online, centrado en el estudiante y basado en una metodología flexible y personalizada. Desde 1995, ofrecemos una formación 100% online, aplicando innovaciones en e-learning gracias a nuestra investigación y desarrollo. Los estudiantes aprenden de manera activa y dinámica, utilizando recursos digitales accesibles en el Campus Virtual.

El sistema incluye elementos como trabajo colaborativo y píldoras autoformativas. Además, el modelo es asincrónico y evoluciona constantemente para adaptarse a las nuevas tecnologías y la sociedad, con un equipo docente dedicado y una comunidad global y diversa.

Como novedad en esta segunda edición de los cursos de la Cátedra UOC con FVASS, se incluye un tutor/a en cada aula que estará dedicado a guiar el proceso formativo, resolver dudas y motivar al alumnado. Además, facilitará la comunicación entre participantes y docentes, asegurando una atención personalizada y la identificación temprana de dificultades y apoyo a las necesidades del alumnado.

5. Resultados de aprendizaje

Este curso se orienta a mejorar las siguientes competencias profesionales.

Aprender a programar en Python es importante por varias razones:

- **Análisis de datos:** con Python, puedes analizar datos para tomar decisiones basadas en información.
- **Innovación:** Python te permite trabajar en áreas como la visualización de datos, la creación de juegos o el aprendizaje automático.
- **Eficiencia:** la sintaxis clara y el código conciso de Python facilitan un rápido desarrollo, esencial en entornos profesionales.
- **Demanda del mercado:** Python es uno de los lenguajes de programación más solicitados, con una amplia aplicación en industrias como la tecnología, las finanzas o la biomedicina.
- **Crecimiento profesional:** Python sirve como base para aprender otros lenguajes de programación y tecnologías, y esto posibilita el crecimiento y la adaptabilidad profesionales.

6. Calendario

El curso tendrá una **duración de 4 - 5 semanas**, en las cuales el participante irá resolviendo las diferentes actividades planteadas.

- **Inicio:** 26/10/2026
- **Fin:** 20/11/2026

7. Docente



El docente de esta formación será Joan Maynou, graduado en Física y Doctor en Ingeniería Biomédica, con más de 15 años de experiencia en el análisis de datos tanto en el entorno hospitalario como en el empresarial, a nivel nacional e internacional. También es docente colaborador en la UOC.

8. Evaluación

Este curso es de carácter **práctico** y se desarrolla mediante la ejecución de proyectos de software real. Las actividades de evaluación se plantean como **retos de ingeniería**, donde el estudiantado debe demostrar su capacidad para traducir lógica de negocio compleja en código Python eficiente y autónomo, minimizando la dependencia de referencias externas.

El curso sigue un modelo de evaluación continua. Esto implica que la única vía para superar la asignatura es la resolución progresiva de los desafíos técnicos propuestos. El diseño pedagógico asegura una evolución constante en las competencias del programador:

- **Algoritmia y estructuras:** implementación de soluciones óptimas para problemas complejos de computación.
- **Arquitectura y diseño:** aplicación de principios avanzados para crear software que no solo funcione, sino que sea escalable, legible y fácil de mantener.
- **Gestión integral:** control total del ciclo de vida del proyecto, incluyendo la manipulación técnica de datos y la administración profesional de dependencias y librerías.

Para la calificación final, se evaluarán los siguientes pilares:

1. **Autonomía técnica:** la capacidad de codificar soluciones robustas de forma independiente.
2. **Calidad del código:** el uso de estándares profesionales de limpieza y eficiencia algorítmica.
3. **Gestión de proyectos:** la correcta administración del entorno de desarrollo y sus librerías.
4. **Compromiso:** la participación activa en la comunidad de aprendizaje y la puntualidad en la entrega de las prácticas.