

**DATOS DE INTERÉS DE ACTIVIDAD FORMATIVA
EDUAM SIGMA SIGMA-PDS – CURSO 2026/27**

1. Nombre de la Actividad: **Introducción a Python para la computación científica.**
2. Responsable de la actividad (Nombre y apellidos, departamento y dirección de email):
Jorge Ibáñez Gijón. Psicología Básica, jorge.ibannez@uam.es
3. Profesores/Ponentes en la actividad (Nombre y apellidos y dirección de email):
Jorge Ibáñez Gijón, jorge.ibannez@uam.es
4. Persona de contacto con los alumnos (Nombre y apellidos y dirección de email):
Jorge Ibáñez Gijón, jorge.ibannez@uam.es
5. Correo contacto cuestiones administrativas: doctorado.gestion@uam.es.
6. Fechas de matrícula: actividad formativa de SIGMA-PDS oficial - matrícula asociada a la renovación de la tutela académica. Ver [Calendario Académico y de Matrícula](#):
 - Del día 16 a 30 de septiembre de 2026: doctorandos/as que se matriculan de las tutelas 2 a 5.
 - Del día 2 al 13 de octubre: tutelas 6 a 9.
 - Del día 19 de octubre al 3 de noviembre: primera tutela.
7. Fechas de impartición: días 2, 4, 9 y 11 de marzo de 2027 (martes y jueves).
8. Horario de impartición:

Días 2, 4 y 9 de marzo de 2027, de 16:00 a 20:00h.
Día 11 de marzo de 2027, de 15:00 a 20:00h.
9. Requisitos de admisión: podrán matricularse las personas matriculadas en cualquier programa de doctorado de la UAM.
10. Número mínimo de matriculados: 15.
11. Número máximo de matriculados: 25.
12. Lugar de impartición: Facultad de Psicología (aula de informática por determinar).
13. Metodología:

a. Modalidad de impartición: **presencial**.

b. Objetivo de la actividad.

El curso propone una introducción a Python orientada específicamente a la computación científica, el análisis de datos y la investigación reproducible en contexto doctoral. En la práctica investigadora actual, la capacidad de procesar datos, automatizar análisis y construir flujos reproducibles es transversal a prácticamente todas las disciplinas, desde ciencias experimentales e ingeniería hasta ciencias sociales y humanidades digitales. Python se ha consolidado como uno de los lenguajes de referencia por su claridad sintáctica, su ecosistema de bibliotecas científicas y su amplia adopción en entornos académicos e industriales.

Este curso responde a la necesidad de dotar a doctorandos sin formación previa en programación de una base operativa suficiente para integrar herramientas computacionales en su trabajo de investigación. El planteamiento es aplicado y metodológico: no se centra en informática teórica, sino en resolución de problemas reales con datos. Se incorporan ejemplos de análisis numérico, tratamiento de datos tabulares, procesamiento básico de datos variados, de forma que estudiantes tanto de ciencias naturales como de humanidades y ciencias sociales encuentren casos directamente transferibles. El enfoque en notebooks reproducibles, modularización y buenas prácticas favorece la transparencia metodológica y la reutilización de código en publicaciones y proyectos.

c. Estructura y contenidos.

La docencia se organiza en cuatro sesiones de cuatro horas, con una progresión acumulativa desde fundamentos de programación hasta análisis científico completo con datos. Cada sesión se divide en dos bloques temáticos de dos horas.

Cada bloque combina exposición conceptual, resolución de micro-ejercicios integrados y práctica guiada con ordenador. Al final de cada bloque se plantea una actividad individual de mayor complejidad que obliga a integrar los conceptos introducidos. En la última sesión, de cinco horas, veremos qué librerías existen para realizar análisis en áreas de conocimiento específicas (en la medida de lo posible, se intentará adaptar el contenido a las propuestas hechas por los alumnos en la primera sesión) y realizarán una práctica con datos reales de su tesis que será la base para el trabajo final que deben entregar.

La organización global del contenido es:

Sesión 1

Bloque 1: Introducción a programación científica. Entornos de trabajo.

Bloque 2: Tipos de datos y estructuras básicas. Lógica y organización del código.

Sesión 2

Bloque 3: Biblioteca estándar y modularización: funciones, clases, scripts y módulos.

Bloque 4: Arrays y álgebra con NumPy. Funciones científicas con SciPy. Modelos estadísticos.

Sesión 3

Bloque 5: Datos tabulares con Pandas. Visualización científica con matplotlib/seaborn.

Bloque 6: Desarrollo basado en tests. Repositorios git. Buenas prácticas en computación científica.

Sesión 4

Bloque 7: Librerías para análisis de datos de áreas específicas (bioinformática, EEG, NLP, GIS...)

Bloque 8: Implementa un análisis de datos de tu tesis con Python.

d. Descripción de actividades de evaluación:

La evaluación combina requisito de participación y prueba aplicada integradora.

La prueba principal es un trabajo final individual en el que el estudiante debe implementar un flujo completo de análisis con Python. Se prioriza el uso de datos de su propia investigación; cuando no sea posible, se permite el uso de datos abiertos o simulados con justificación metodológica. El trabajo debe incluir: definición del problema analítico; carga o generación de datos brutos; limpieza y transformación; estructuración tabular; tratamiento de posibles series temporales si procede; generación de variables para análisis estadístico; aplicación de técnicas básicas de inferencia o ajuste de modelos; representación gráfica adecuada; e interpretación razonada de resultados.

Se valoran la corrección técnica, la trazabilidad del proceso, la claridad del código, la modularización y la reproducibilidad. La entrega se realiza en notebook ejecutable o proyecto de scripts con instrucciones de reproducción.

e. Idioma de impartición: español.

f. Equivalencia de créditos ECTS: 2.0

g. La calificación de no evaluado por falta de asistencia no justificada conllevará la penalización de no poder matricularse en actividades formativas propias de la EDUAM en el próximo curso académico.

14. Actividad de matrícula directa.