

# Modelos Computacionales I

Presenta:

Dr. Fernando Javier Alcántara López

FACULTAD DE MEDICINA



Licenciatura en Neurociencias



# Presentación

## Formas de Evaluación General:

- Dr. Oswaldo 35%
- Python 35%
- Proyecto final 30%

## Evaluación

- Trabajos 30%
- Exámenes 70%

## Días de clase:

- Lunes 5:00-7:00pm
- Martes 5:00-7:00pm
- Miércoles 5:00-7:00pm
- Jueves 5:00-7:00pm

- Aula V-404

## Fechas de parciales:

- 1er parcial 11-Sep
- 2do parcial 23-Octubre
- 3er parcial 26-Noviembre

# Calendario semestral de la UNAM 2026-1

## AGOSTO 2025

L	M	M	J	V	S	D
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

CICLO ESCOLAR 2025-2026

## SEPTIEMBRE 2025

L	M	M	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

## OCTUBRE 2025

L	M	M	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

## NOVIEMBRE 2025

L	M	M	J	V	S	D
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

## DICIEMBRE 2025

L	M	M	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

## ENERO 2026

L	M	M	J	V	S	D
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

## Semestre 1



Inicio ciclo escolar  
11 / Agosto / 2025



Fin ciclo escolar  
28 / Noviembre / 2025



Días inhábiles

Septiembre 15 y 16

Enero 1

Marzo 16

Noviembre 1, 2 y 17

Febrero 2

Mayo 1, 10 y 15

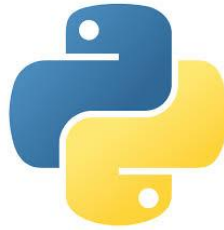
Diciembre 12 y 25

CALENDARIO ESCOLAR  
PLAN SEMESTRAL

2026

# Temario. Python

## Python

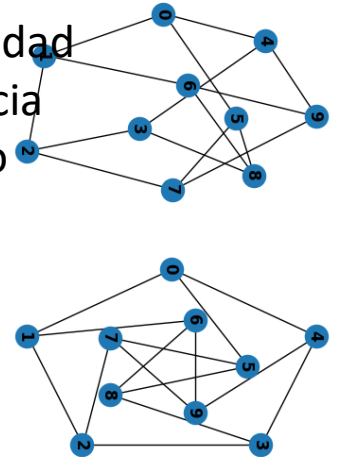


- ¿Qué es Python?
- Entornos de aprendizaje IDEs
- Dependencias (Git/GitHub, jupyter, Pycharm)
- Tipos de datos y operadores
- Expresiones
- Controladores de flujo
- Colecciones de datos
- Entradas y salidas de datos
- Funciones
- Arreglos de datos con numpy
- Bases de datos con pandas
- Funciones lambda, filter y map
- Visualización de datos con matplotlib, seaborn
- Networkx



## Teoría de grafos

- ¿Qué es la Topología?
- ¿Qué es un grafo?
- Nodos y Aristas
- Teoría de matrices
- Matriz de conectividad
- Matriz de adyacencia
- El grafo del cerebro
- Centralidad
- Hubs



## Bibliografía

Fornito, A., Zalesky, A., & Bullmore, E. (2016). *Fundamentals of brain network analysis*. Academic press.

# Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para **realizar procesos que pueden ser llevados a cabo por máquinas** como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Está formado por un conjunto de **símbolos y reglas sintácticas y semánticas** que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación. Un lenguaje de programación proporciona los elementos de lenguaje necesarios que son necesarios para traducir los pasos de un pseudocódigo en formato comprensible de la máquina. En otras palabras, el lenguaje de programación proporciona el puente para hacer la transición de pseudocódigo legible por humano instrucciones legible por máquina.

Las clasificaciones principales de lenguajes de programación son: lenguajes de bajo nivel, de nivel medio y de alto nivel.



# ¿Por qué python?

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.





# Ventajas



- Sintaxis clara y legible:** La sintaxis de Python es simple y fácil de entender. Su estilo de codificación facilita la lectura y escritura de código, lo que acelera el desarrollo y facilita la colaboración entre equipos.

- Gran comunidad y soporte:** Python cuenta con una comunidad activa de desarrolladores que contribuyen con bibliotecas, tutoriales y resuelven dudas en foros. Esto proporciona un valioso respaldo y facilita el aprendizaje y la resolución de problemas.

- Amplia biblioteca estándar:** Python incluye una biblioteca estándar extensa que cubre diversas áreas, desde manipulación de archivos hasta desarrollo web. Esto le permite a los desarrolladores aprovechar herramientas existentes sin necesidad de reinventar la rueda.

- Versatilidad y portabilidad:** Python es un lenguaje multiplataforma, lo que significa que el código escrito en Python puede ejecutarse en diferentes sistemas operativos sin modificaciones significativas. Esto facilita el desarrollo de aplicaciones que pueden ser utilizadas en una variedad de entornos.

- Desarrollo rápido de prototipos:** La facilidad de escritura y la sintaxis concisa hacen que Python sea ideal para el desarrollo rápido de prototipos. Esto le permite a los desarrolladores probar y experimentar con ideas antes de comprometerse con implementaciones más extensas.

- Integración fácil con otros lenguajes:** Python puede integrarse fácilmente con otros lenguajes, como C y C++. Esto le permite a los desarrolladores aprovechar código existente y obtener un rendimiento óptimo cuando sea necesario.

# Desventajas



- Velocidad de ejecución:** Python, al ser un lenguaje interpretado, puede ser más lento en comparación con lenguajes compilados, como C++ o Java. Esto puede ser una limitación en aplicaciones que requieren un rendimiento extremadamente rápido, como desarrollo de juegos o cómputo intensivo.

- Gestión de memoria automática:** Aunque la gestión automática de memoria es una ventaja para muchos desarrolladores, puede ser una desventaja en aplicaciones que requieren un control preciso de los recursos de memoria, ya que Python no proporciona la misma flexibilidad que lenguajes de bajo nivel.

- No es ideal para desarrollo móvil:** Aunque existen *frameworks* como Kivy o BeeWare, Python no es considerado el lenguaje principal para el desarrollo móvil. Otros lenguajes como Swift o Kotlin son preferidos en este contexto.

- Problemas de comprensión asíncrona:** Aunque Python ha mejorado en la gestión de operaciones asíncronas, algunos desarrolladores aún encuentran desafíos al trabajar con código asíncrono, especialmente en comparación con lenguajes diseñados específicamente para manejar tareas concurrentes.

- Interpretación y ejecución:** El hecho de que Python sea interpretado puede ser una desventaja en términos de velocidad de ejecución en comparación con lenguajes compilados. Sin embargo, esta desventaja se ve mitigada por las implementaciones Just-In-Time (JIT) como PyPy.

- Problemas de integración con C/C++:** Aunque Python se puede integrar con C y C++, puede haber desafíos y complejidades asociadas con la interacción entre estos lenguajes, especialmente en proyectos más grandes.



# IDE (Entorno de Desarrollo Integrado)

