

# Subrutinas en Python (Ejercicios)

Materia: Algoritmos y Programación

Prof. D.Sc. BARSEKH-ONJI Aboud

Facultad de Ingeniería  
Universidad Anáhuac México

3 de noviembre de 2025

# Agenda

1. Ejercicio 1

2. Ejercicio 2

3. Ejercicio 3

# Ejercicio 1: El Validador de Números Primos

## Objetivo

Escribir un programa que le pida al usuario un número entero y determine si es un número primo o no.

1. Deben crear una **función** llamada `es_primo(numero)`.
2. Esta función debe recibir un número como **parámetro**.
3. La función debe **devolver** un valor Booleano: True si el número es primo, o False si no lo es.
4. **Lógica (Pista):** Un número es primo si solo es divisible por 1 y por sí mismo. Pueden usar un ciclo `for` que vaya desde 2 hasta `numero - 1` y revisar si el residuo (%) de la división es 0.
5. El **programa principal** debe pedir el número, llamar a la función `es_primo()` y, basado en el resultado que *retornó*, imprimir el mensaje final (ej. "El 17 ES primo" o "El 18 NO es primo").

# Agenda

1. Ejercicio 1

2. Ejercicio 2

3. Ejercicio 3

## Ejercicio 2: El Buscador de Coordenadas

### Objetivo

Dada una matriz (lista de listas) y un valor, encontrar la posición (fila y columna) de ese valor.

1. Deben crear una **función** llamada `buscar_valor(matriz, valor_buscado)`.
2. Esta función debe recibir la matriz (lista 2D) y el valor a buscar como **parámetros**.
3. **Lógica (Pista):** Deben usar **ciclos anidados** (`for`) para recorrer cada fila y cada columna.
4. Si encuentra el valor, la función debe **devolver** las coordenadas. (Pista: pueden devolver una tupla (`fila, columna`)).
5. Si el valor **no se encuentra** después de recorrer toda la matriz, la función debe **devolver** `None`.
6. El **programa principal** debe definir una matriz, llamar a la función y luego usar un `if` para revisar el valor devuelto e imprimir un mensaje (ej. "Valor encontrado en (2, 1).º "Valor no encontrado").

# Agenda

1. Ejercicio 1

2. Ejercicio 2

3. Ejercicio 3

## Ejercicio 3: Mini Analizador de Texto (Menú Interactivo)

### Objetivo

Crear un programa que ofrezca un menú para realizar diferentes operaciones sobre una frase que ingrese el usuario.

1. El **programa principal** debe pedir al usuario una frase **una sola vez** al inicio.
2. Luego, debe entrar en un ciclo `while` que muestre un menú de opciones (ej. 1. Contar Vocales, 2. Contar Palabras, 3. Invertir Frase, 4. Salir).
3. Deben usar una estructura `match` (o `if/elif`) para manejar la opción del usuario.
4. Deben crear **funciones separadas** para cada operación:
  - `contar_vocales(frase)` → debe **devolver** un número.
  - `contar_palabras(frase)` → debe **devolver** un número (Pista: `frase.split()`).
  - `invertir_frase(frase)` → debe **devolver** un nuevo string (Pista: `frase[::-1]`).
5. El ciclo `while` principal llamará a la función correspondiente e imprimirá el resultado **devuelto** por ella.
6. El ciclo debe terminar si el usuario elige la opción "4. Salir".