

## Ejercicio 1: Manipulación de Listas y Diccionarios

Dado el siguiente dataset de empleados en formato de **lista de diccionarios**, implementa una función que:

- Devuelva los nombres de los empleados que ganan más de \$50,000.
- Calcule el **salario promedio** de los empleados.

```
empleados = [  
    {"nombre": "Carlos", "edad": 35, "salario": 55000},  
    {"nombre": "Ana", "edad": 28, "salario": 48000},  
    {"nombre": "Luis", "edad": 40, "salario": 62000},  
    {"nombre": "Sofía", "edad": 32, "salario": 70000},  
    {"nombre": "Pedro", "edad": 45, "salario": 52000}  
]
```

### Output esperado

Nombres de empleados con salario > 50,000: ['Carlos', 'Luis', 'Sofía', 'Pedro']

Salario promedio: 57400

## Ejercicio 2: Análisis de Texto

Implementa una función que reciba un **texto** y devuelva un diccionario con el **conteo de palabras**, ignorando mayúsculas y signos de puntuación.

### Output esperado

```
{'python': 3, 'es': 2, 'increíble': 1, 'poderoso': 1, 'me': 1, 'encanta': 1}
```

## Ejercicio 3: Bases de Datos y ORM

Crear un CRUD básico con **SQLite y SQLAlchemy**.

**Base de Datos SQLite**

- Crea una **base de datos SQLite** llamada empresa.db.
- Crea una tabla **Empleados** con columnas: id, nombre, edad, salario.
- Implementa funciones para **insertar, actualizar, eliminar y obtener empleados**.
- Muestra la lista de empleados después de cada operación.

#### Ejercicio 4: Creación de API

- Crea una API con FastAPI que tenga una ruta **/saludo/{nombre}** que reciba un nombre y devuelva "Hola, {nombre}!".
- Agrega una ruta **/calculadora/suma?num1=5&num2=3** que devuelva la suma de dos números.
- Agrega validaciones de entrada.
- Implementar documentación automática con **Swagger** en FastAPI

#### Ejercicio 5: Ejecución en Paralelo

- Implementa una función que reciba una lista de números y calcule el **cuadrado de cada número en paralelo** usando **concurrent.futures**.

#### Ejercicio 6: Pruebas en Python

- Escribe pruebas unitarias para la función de conteo de palabras (Ejercicio 2).
- Usa pytest y verifica que la función maneje correctamente:
  - Texto con diferentes mayúsculas y minúsculas.
  - Texto con signos de puntuación.
  - Texto vacío.