



## 01 PAO25-25 - PYTHON, Data Types



*Luis Pilaguano*

### EXAMEN PRACTICO PRIMER PARCIAL

**Análisis de rendimiento de estudiantes con validación de datos y control estructurado.**

#### Objetivos

Desarrollar un programa en Python que reciba un conjunto de registros de estudiantes (nombre y lista de notas), e implemente estructuras de control para:

- Validar que las notas estén entre 0 y 10 (inclusive)
- Calcular el promedio individual
- Clasificar al estudiante con base en el promedio usando match o if-elif
- Usar control de errores (try/except) para gestionar entradas defectuosas
- Al final, mostrar un resumen con:
  - 1.- La cantidad de estudiantes aprobados, reprobados y con calificación excelente
  - 2.- El estudiante con el promedio más alto
  - 3.- Una lista de estudiantes ordenados por calificación (usando list comprehension con condiciones)

```
In [13]: print("--- Sistema de Clasificación de Estudiantes ---")
print("Escribe 'FIN' para terminar.\n")
```

```

estudiantes = []
nombre = input("Nombre del estudiante")

while nombre != "FIN":
    notas = input("Escribe las notas separadas por coma (NOTAS EN UNA RANGO DE 0
#separa las notas por un ,
    notas_lista = notas.split(",")
    notas_validas = []

    for n in notas_lista:
        try:
            if 0 <= nota <= 10:
                notas_validas.append(nota)
            else:
                print("Nota fuera de rango (0-10)")
                notas_validas = []
                break
        except ValueError:
            print("Nota no válida.")
            notas_validas = []
            break

    if not notas_validas:
        print("Estudiante omitido")
    else:
        promedio = sum(notas_validas) / len(notas_validas)

        if promedio >= 9:
            clasificacion = "Excelente"
        elif promedio >= 6:
            clasificacion = "Aprobado"
        else:
            clasificacion = "Reprobado"

        print(f"{nombre} - Promedio: {promedio} - {clasificacion}")

        estudiantes.append({
            "nombre": nombre,
            "promedio": promedio,
            "clasificacion": clasificacion
        })

    nombre = input("\nNombre del siguiente estudiante (o 'FIN') ")

print("\n--- Resumen Final ---")

if estudiantes:
    Excelente = sum(1 for e in estudiantes if e["clasificacion"] == "Excelente")
    Aprobado = sum(1 for e in estudiantes if e["clasificacion"] == "Aprobado")
    Reprobado = sum(1 for e in estudiantes if e["clasificacion"] == "Reprobado")

    print(f"Excelente: {Excelente}")
    print(f"Aprobado: {Aprobado}")
    print(f"Reprobado: {Reprobado}")
else:
    print("No se ingresaron estudiantes válidos.")

```

--- Sistema de Clasificación de Estudiantes ---  
Escribe 'FIN' para terminar.

LUIS - Promedio: 4.0 - Reprobado  
--- Resumen Final ---  
Excelente: 0  
Aprobado: 0  
Reprobado: 1

## Antecedentes/Escenario

En la programación real, los desarrolladores frecuentemente se enfrentan a la necesidad de validar y procesar datos provenientes de diversas fuentes. Una de las tareas comunes en aplicaciones académicas o administrativas es el manejo de registros de estudiantes, los cuales pueden incluir calificaciones, observaciones y clasificaciones. Estas tareas requieren no solo cálculos, sino también una adecuada estructuración del control del flujo del programa para gestionar condiciones, errores y decisiones lógicas.

El lenguaje Python proporciona herramientas sólidas para el manejo de este tipo de escenarios a través de estructuras de control como sentencias condicionales (if, elif, else), bucles iterativos (for, while), control de flujo con break y continue, el uso del bloque try/except para el tratamiento de excepciones, así como expresiones compactas como las list comprehensions. Estas herramientas permiten construir programas robustos y eficientes para el análisis y manipulación de datos. Además, Python incorpora una estructura moderna como match-case, que permite implementar una lógica condicional similar al switch en otros lenguajes, haciendo el código más limpio y expresivo al clasificar o tomar decisiones según múltiples posibles valores.

En este ejercicio práctico, se integran todas estas capacidades para desarrollar un sistema de clasificación de estudiantes basado en sus calificaciones. El objetivo es aplicar de forma conjunta los conocimientos sobre estructuras de control, manejo de errores y procesamiento de datos, abordando problemas típicos de entrada, validación, clasificación y salida de resultados. Esta actividad representa un caso aplicado de desarrollo lógico secuencial con validación de datos, ideal para consolidar los conocimientos adquiridos en la unidad sobre estructuras de control.

## Desarrollo

- El programa solicitará datos hasta que se ingrese "FIN" como nombre del estudiante.
- Por cada estudiante se ingresarán varias notas separadas por coma. Si alguna nota no es numérica o no está en el rango, se mostrará un mensaje y se omite ese estudiante.
- La clasificación de desempeño será:
  - "Excelente" si el promedio es mayor o igual a 9
  - "Aprobado" si el promedio está entre 6 y 8.99
  - "Reprobado" si el promedio es menor a 6

[Git Hub](#)

In [ ]: