



# Profundiza más

## Recurso de Profundización

### **Clase 12: Estructuras de control en conjunto y arreglos (Parte 2)**

**Autor:** Damián Nicolalde Rodríguez

#### **Objetivo del Documento:**

El objetivo de este recurso es consolidar el aprendizaje de los temas desarrollados a lo largo de las clases previas, integrando todo lo aprendido desde los conceptos básicos de programación hasta el manejo de estructuras de control y arreglos multidimensionales (matrices) en Python. A través de un ejercicio práctico, los estudiantes podrán aplicar estos conocimientos y crear un sistema funcional que utilice arreglos multidimensionales, bucles, condicionales y métodos de listas para resolver un problema real de gestión de datos.

#### **Descripción del Ejercicio:**

En este recurso, vamos a construir un programa completo que gestione los registros de estudiantes en una escuela. El sistema almacenará los datos de los estudiantes en una matriz bidimensional (matriz de calificaciones) y proporcionará funcionalidades como ingresar nuevos estudiantes, visualizar las calificaciones, calcular promedios y filtrar estudiantes por su desempeño. A través de este ejercicio, los estudiantes reforzarán el uso de matrices, bucles anidados, condicionales y métodos comunes de arreglos.

#### **Ejercicio: Sistema de Gestión de Estudiantes**

Este ejercicio se basa en el uso de matrices y listas para gestionar información tabular. El programa permitirá ingresar la información de varios estudiantes y sus calificaciones, calcular los promedios, clasificar a los estudiantes según su rendimiento, y mostrar el estado de los estudiantes (Aprobado, Reprobado, etc.).



# Profundiza más

## Requerimientos del Ejercicio:

1. Crear una matriz de 5x5 que contenga las calificaciones de 5 estudiantes en 5 asignaturas.
2. El programa debe permitir ingresar los datos de los estudiantes (nombre y calificaciones) y almacenarlos en la matriz.
3. El programa debe calcular el promedio de cada estudiante.
4. Clasificar a los estudiantes según su promedio: Aprobado (mayor o igual a 6), Reprobado (menor a 6).
5. Permitir mostrar la lista de estudiantes aprobados o reprobados.
6. Incluir métodos para agregar nuevos estudiantes, eliminar un estudiante, y mostrar las calificaciones de todos los estudiantes.

```
# Definir la matriz de calificaciones inicial
```

```
estudiantes = [  
    ["Juan", [8, 7, 6, 9, 7]],  
    ["Ana", [9, 10, 8, 9, 10]],  
    ["Luis", [5, 4, 6, 4, 5]],  
    ["Carlos", [6, 6, 7, 5, 6]],  
    ["Marta", [7, 8, 6, 7, 7]]  
]
```

```
# Imprimir las calificaciones de todos los estudiantes
```

```
print("Sistema de Gestión de Estudiantes")  
print("Calificaciones actuales de los estudiantes:")  
for estudiante in estudiantes:  
    nombre = estudiante[0]  
    calificaciones = estudiante[1]  
    print(f" {nombre}: {calificaciones}")
```



# Profundiza más

```
# Calcular y clasificar el rendimiento de los estudiantes
print("\nClasificación de los estudiantes:")
for estudiante in estudiantes:
    nombre = estudiante[0]
    calificaciones = estudiante[1]
    promedio = sum(calificaciones) / len(calificaciones) # Calcular el
    promedio
    estado = "Aprobado" if promedio >= 6 else "Reprobado" # Clasificar
    según el promedio
    print(f"{nombre} tiene un promedio de {promedio:.2f} y está
    {estado}.")

# Ingreso de un nuevo estudiante
print("\nAgregar un nuevo estudiante")
nuevo_nombre = input("Ingrese el nombre del nuevo estudiante: ")

# Validar las calificaciones ingresadas por el usuario
while True:
    try:
        nuevas_calificaciones = input(f"Ingrese las calificaciones de
        {nuevo_nombre} separadas por comas: ")
        nuevas_calificaciones = [float(i) for i in
        nuevas_calificaciones.split(",")] # Convertir en lista de números
        break # Si no ocurre ningún error, salir del bucle
    except ValueError:
        print("Error: Ingrese solo números separados por comas. Intente
        nuevamente.")

# Agregar el nuevo estudiante a la lista
estudiantes.append([nuevo_nombre, nuevas_calificaciones])

# Mostrar el nuevo estudiante y sus calificaciones
print(f"\nNuevo Estudiante Ingresado: {nuevo_nombre}")
for estudiante in estudiantes:
    nombre = estudiante[0]
    calificaciones = estudiante[1]
    print(f"{nombre}: {calificaciones}")
```



# Profundiza más

```
# Clasificar nuevamente después de agregar el nuevo estudiante
print("\nClasificación de los estudiantes después de agregar uno nuevo:")
for estudiante in estudiantes:
    nombre = estudiante[0]
    calificaciones = estudiante[1]
    promedio = sum(calificaciones) / len(calificaciones)
    estado = "Aprobado" if promedio >= 6 else "Reprobado"
    print(f"{nombre} tiene un promedio de {promedio:.2f} y está
{estado}.")
```

## Explicación detallada del flujo:

### 1. Definición inicial de estudiantes:

La lista estudiantes contiene sublistas, donde cada sublista tiene el nombre de un estudiante y sus calificaciones. Al inicio del programa, se muestra esta lista de estudiantes con sus respectivas calificaciones.

### 2. Clasificación de estudiantes:

Se recorre la lista de estudiantes y para cada uno se calcula su promedio usando  $\text{sum}(\text{calificaciones}) / \text{len}(\text{calificaciones})$  y luego se clasifica como "Aprobado" o "Reprobado" dependiendo de si el promedio es mayor o igual a 6.

### 3. Ingreso de un nuevo estudiante:

El programa pide al usuario que ingrese el nombre del nuevo estudiante y sus calificaciones. Si las calificaciones no son válidas (es decir, si el usuario no ingresa solo números separados por comas), se utiliza un bucle while para seguir pidiendo al usuario hasta que ingrese los datos correctamente.

### 4. Agregar el nuevo estudiante:

El nuevo estudiante (nombre y calificaciones) se agrega a la lista estudiantes utilizando el método `append()`. Después de agregar el estudiante, se muestra la lista completa de estudiantes, incluida la nueva entrada.

### 5. Clasificación después de agregar un estudiante:

Finalmente, el programa recorre nuevamente la lista de estudiantes para calcular y mostrar el promedio y la clasificación de cada uno, incluyendo el nuevo estudiante agregado.



# Profundiza más

Salida:

Sistema de Gestión de Estudiantes

Calificaciones actuales de los estudiantes:

Juan: [8, 7, 6, 9, 7]

Ana: [9, 10, 8, 9, 10]

Luis: [5, 4, 6, 4, 5]

Carlos: [6, 6, 7, 5, 6]

Marta: [7, 8, 6, 7, 7]

Clasificación de los estudiantes:

Juan tiene un promedio de 7.40 y está Aprobado.

Ana tiene un promedio de 9.20 y está Aprobado.

Luis tiene un promedio de 4.80 y está Reprobado.

Carlos tiene un promedio de 6.00 y está Aprobado.

Marta tiene un promedio de 7.00 y está Aprobado.

Agregar un nuevo estudiante

Ingrese el nombre del nuevo estudiante: Pedro

Ingrese las calificaciones de Pedro separadas por comas: 6, 7, 5, 8, 6

Nuevo Estudiante Ingresado: Pedro

Juan: [8, 7, 6, 9, 7]

Ana: [9, 10, 8, 9, 10]

Luis: [5, 4, 6, 4, 5]

Carlos: [6, 6, 7, 5, 6]

Marta: [7, 8, 6, 7, 7]

Pedro: [6, 7, 5, 8, 6]

Clasificación de los estudiantes después de agregar uno nuevo:

Juan tiene un promedio de 7.40 y está Aprobado.

Ana tiene un promedio de 9.20 y está Aprobado.

Luis tiene un promedio de 4.80 y está Reprobado.

Carlos tiene un promedio de 6.00 y está Aprobado.

Marta tiene un promedio de 7.00 y está Aprobado.

Pedro tiene un promedio de 6.40 y está Aprobado.