



ARIDIEY

**Nombre:** John Gilbert Queque Arias

**Curso:** Desarrollo en Python Intermedio

**Proyecto:** Lista de alumnos

**2024**

## Introducción:

Mi nombre es John Gilbert Queque Arias, tengo 20 años y soy un apasionado del desarrollo de aplicaciones y la gestión de proyectos. Mi meta es dedicarme profesionalmente a estas áreas, donde estaría creando soluciones innovadoras y coordinando equipos para lograr el éxito en cada proyecto. En este caso le presento mi solución del proyecto propuesto.

## Desarrollo:

- ❖ Solicitar la cantidad de alumnos N.
- ❖ Solicitar la cantidad de notas por cada alumno.

```
1  import os
2
3  def solicitar_datos_generales():
4      n_alumnos = 0
5      n_notas = 0
6      while True:
7          os.system('cls')
8          n_alumnos = int(input('Ingrese la cantidad de alumnos: '))
9          n_notas = int(input('Ingrese la cantidad de notas por alumno: '))
10         if n_alumnos > 0 and n_notas > 0:
11             return n_alumnos, n_notas
12         else:
13             print('\nValores no soportados')
14             os.system('pause')
```

- ❖ Solicitar los datos de cada alumno (nombre, apellido, sección y grado).

```
1  def solicitar_datos_alumno():
2      nombre = input("Nombre del alumno: ")
3      apellido = input("Apellido del alumno: ")
4      seccion = input("Sección del alumno: ")
5      grado = input("Grado del alumno: ")
6      print()
7      return nombre, apellido, seccion, grado
```

- ❖ Crear una función que permita saber el promedio de notas por alumno.

```
1  # + Funcion de promedio de notas
2  def hallar_promedio(notas):
3      return sum(notas) / len(notas)
```

- ❖ Crear una función que permita saber cuál es el nombre del alumno con mayor promedio.

```
1  # + Funcion de alumno con mayor promedio
2  def obtener_alumno_promedio(nombres, promedios):
3      promedio_mayor = 0
4      for promedio in promedios:
5          if promedio_mayor < promedio:
6              promedio_mayor = promedio
7
8      indice_promedio = promedios.index(promedio_mayor)
9      return nombres[indice_promedio]
```

- ❖ Validar que solamente se ingrese notas en el rango de 0 a 20. Caso contrario mostrar un mensaje indicando que "el valor de la nota no pertenece a un rango válido".

```
1  # + Validaciones (notas >0 <20, el valor de la
2  # nota no pertenece a un rango válido)
3  def validacion_notas(nota):
4      if 0 <= nota and nota <= 20:
5          return True
6      else:
7          print('El valor de la nota no pertenece a un rango válido')
8          return False
9
10 # + Nombre del alumno con mayor promedio en UPPER()
11 def validacion_mayusculas(nombre):
12     return nombre.isupper()
```

## Código completo:

```
1  # Almacenar la lista de nombres de alumnos, por cada
2  # alumno un listado de notas de todo el ciclo
3  import os
4
5  def solicitar_datos_generales():
6      n_alumnos = 0
7      n_notas = 0
8      while True:
9          os.system('cls')
10         n_alumnos = int(input('Ingrese la cantidad de alumnos: '))
11         n_notas = int(input('Ingrese la cantidad de notas por alumno: '))
12         if n_alumnos > 0 and n_notas > 0:
13             return n_alumnos, n_notas
14         else:
15             print('\nValores no soportados')
16             os.system('pause')
17
18  def solicitar_datos_alumno():
19      nombre = input("Nombre del alumno: ")
20      apellido = input("Apellido del alumno: ")
21      seccion = input("Sección del alumno: ")
22      grado = input("Grado del alumno: ")
23      print()
24      return nombre, apellido, seccion, grado
25
26  # + Funcion de promedio de notas
27  def hallar_promedio(notas):
28      return sum(notas) / len(notas)
29
30  # + Funcion de alumno con mayor promedio
31  def obtener_alumno_promedio(nombres, promedios):
32      promedio_mayor = 0
33      for promedio in promedios:
34          if promedio_mayor < promedio:
35              promedio_mayor = promedio
36
37      indice_promedio = promedios.index(promedio_mayor)
38      return nombres[indice_promedio]
39
40  # + Validaciones (notas >0 <20, el valor de la
41  # nota no pertenece a un rango válido)
42  def validacion_notas(nota):
43      if 0 <= nota and nota <= 20:
44          return True
45      else:
46          print('El valor de la nota no pertenece a un rango válido')
47          return False
```

```

48
49 # + Nombre del alumno con mayor promedio en UPPER()
50 def validacion_mayusculas(nombre):
51     return nombre.isupper()
52
53 # + Almacenar los elementos en listas
54 lista_nombres = []
55 lista_notas = []
56 # + Solicitar la cantidad
57 # + Solicitar la cantidad de notas por cada alumno
58 n_alumnos, n_notas = solicitar_datos_generales()
59
60 # + Solicitar los datos de cada alumno
61 print('-----')
62 print('\n SOLICITANDO DATOS DE ALUMNOS \n')
63 print('-----')
64 for i in range(n_alumnos):
65     print('+-----+')
66     print(f'| ALUMNO N° {i + 1} |')
67     print('+-----+')
68     nombre, apellido,_,_ = solicitar_datos_alumno()
69     notas = []
70     for j in range(n_notas):
71         while True:
72             try:
73                 nota = float(input(f'Nota {j + 1} : '))
74                 if (validacion_notas(nota)):
75                     notas.append(nota)
76                     break
77             except:
78                 print('Error: Por favor ingrese un número valido')
79
80     lista_nombres.append(nombre + ' ' + apellido)
81     lista_notas.append(notas)
82
83 promedios = [hallar_promedio(notas) for notas in lista_notas]
84 nombre_mayor_promedio = obtener_alumno_promedio(lista_nombres, promedios)
85
86 if not validacion_mayusculas(nombre_mayor_promedio):
87     indice_nombre = lista_nombres.index(nombre_mayor_promedio)
88     lista_nombres[indice_nombre] = lista_nombres[indice_nombre].upper()
89
90 print('\nPromedio de notas por alumno\n')
91 for nombre, promedio in zip(lista_nombres, promedios):
92     print(f'{nombre}: {promedio}')
93

```

## **Conclusión:**

Implementar funciones y listas en Python ofrece múltiples beneficios, las funciones permiten reutilizar y organizar código de manera eficiente, en este caso se pudo realizar el proyecto modulando o separando el código en tareas específicas que solucionan un problema en concreto, el uso de listas me facilita el manejo de datos tanto para los nombres de cada alumno como las notas, por su versatilidad para implementar una gran variedad de funciones y algoritmos.

**@Ardev**