

**Neri Lizardo Juárez Velasquez**

## **Gestor de Notas en Python**

### **Manual técnico de gestor de notas**

El presente proyecto consiste en el desarrollo de un gestor de notas básico, realizado en el lenguaje de programación Python. Su objetivo principal es facilitar el control y administración de cursos junto con sus respectivas calificaciones. El programa permite agregar, modificar, eliminar y buscar cursos, así como calcular promedios, ordenar información y mantener un historial de los cambios realizados mediante funciones.

El sistema se basa en el uso de listas paralelas, donde una lista almacena los nombres de los cursos (cursos) y otra lista almacena las notas correspondientes (notas). De esta forma, cada curso mantiene una relación directa con su calificación mediante su posición dentro de las listas.

Además, se implementan estructuras adicionales para simular procesos reales:

#### **Estructura general del código**

- Una cola (**cola**) para gestionar las solicitudes de revisión de notas, funcionando bajo la lógica *FIFO* (First In, First Out), es decir, el primer curso en solicitar revisión es el primero en ser atendido.
- Una Pila (**historial**) para guardar el registro de todos los cambios efectuados en el sistema, funcionando con la lógica *LIFO* (Last In, First Out), mostrando los cambios más recientes primero.

El programa presenta un **menú interactivo** en consola que permite al usuario seleccionar entre trece opciones distintas, entre las cuales se incluyen:

1. Registrar nuevos cursos y notas.
2. Mostrar todos los cursos registrados.
3. Calcular el promedio general de notas.
4. Mostrar los cursos aprobados y reprobados.
5. Buscar un curso por su nombre (búsqueda lineal).
6. Actualizar o cambiar la nota de un curso.
7. Eliminar cursos del registro.
8. Ordenar los cursos por nota utilizando el método de burbuja.
9. Ordenar los cursos por nombre mediante el método de inserción.
10. Buscar un curso usando búsqueda binaria (requiere lista ordenada).
11. Simular la cola de solicitudes de revisión de notas.
12. Mostrar el historial de modificaciones.
13. Salir del programa.

## Funciones y como trabajan.

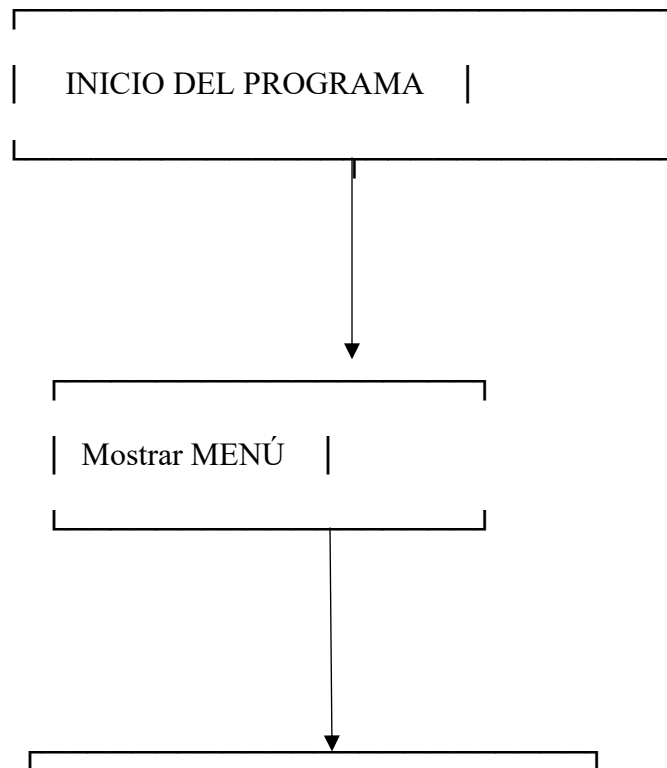
Cada función fue diseñada de manera que pueda cumplir su proposito, permitiendo mantener un código organizado y fácil de entender. Por ejemplo, la función `agregar()` permite registrar nuevos cursos verificando que no se repitan y que las notas estén dentro de un rango válido (de 0 a 100). La función `promedio()` calcula el promedio general utilizando la operación  $\text{sum}(\text{notas}) / \text{len}(\text{notas})$ . Por otro lado, las funciones de ordenamiento permiten reorganizar la información según la necesidad del usuario, ya sea por nombre o por calificación.

## Bucle para menú principal

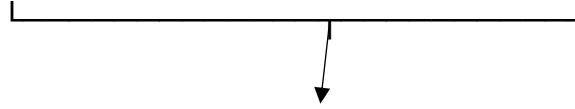
El flujo general del programa se basa en un **bucle principal** que muestra el menú recibe la opción seleccionada por el usuario y ejecuta la función correspondiente. El ciclo se repite hasta que el usuario decida salir del sistema.

Este proyecto demuestra la aplicación práctica de **estructuras de datos básicas (listas, pilas y colas)**, **búsquedas (lineal y binaria)** y **algoritmos de ordenamiento (burbuja e inserción)** dentro de un entorno interactivo. Además fomenta el desarrollo de lógica de programación, control de flujo y manejo de datos, los cuales son de bastante ayuda para realizar programas en Python.

## Diagrama general



| Leer opción del usuario |



| Según la opción elegida: |

| 1 → Agregar curso

| 2 → Mostrar cursos

| 3 → Calcular promedio

| 4 → Ver aprobados/reprobados

| 5 → Buscar curso

| 6 → Cambiar nota

| 7 → Borrar curso

| 8 → Ordenar por nota

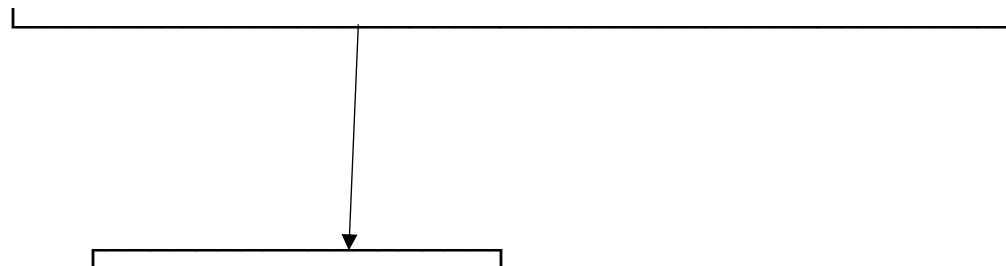
| 9 → Ordenar por nombre

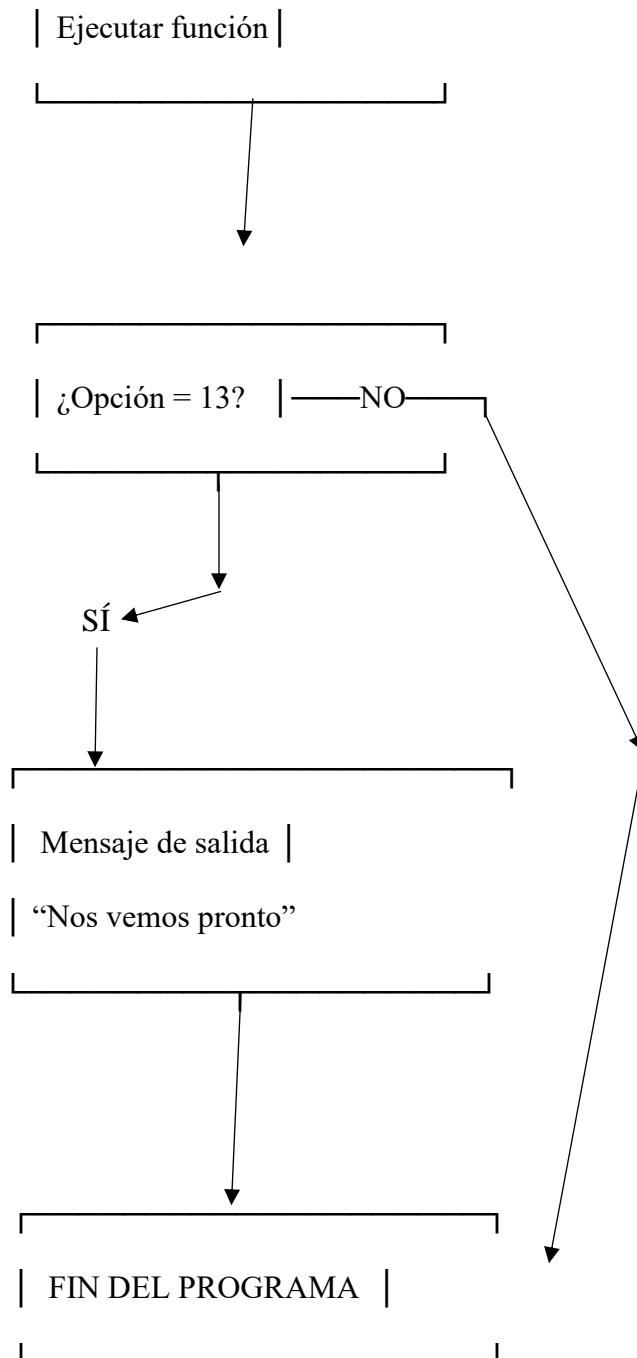
| 10 → Buscar binario

| 11 → Revisiones

| 12 → Ver historial

| 13 → Salir





**Como trabaja cada función.**

### **1. Agregar curso**

Permite añadir nuevos cursos al sistema.

El programa pedirá cuántos cursos quieres agregar, el nombre y la nota (de 0 a 100).

## **2. Mostrar cursos**

Muestra todos los cursos con sus notas actuales.

## **3. Promedio general**

Calcula el promedio de todas las notas registradas.

## **4. Aprobados / Reprobados**

Indica si cada curso está aprobado (nota  $\geq 61$ ) o reprobado (nota  $< 61$ ).

## **5. Buscar curso**

Se escribe el nombre del curso y muestra su nota si es que ya existe.

## **6. Cambiar nota**

Permite modificar la nota de un curso específico y cada cambio queda registrado en el historial.

## **7. Borrar curso**

Elimina un curso y su nota del sistema y también se guarda en el historial.

## **8. Ordenar por nota**

Ordena todos los cursos de menor a mayor según su nota.

## **9. Ordenar por nombre**

Ordena los cursos alfabéticamente de la A a la Z.

## **10. Buscar binario**

Hace una **búsqueda binaria**, que es una forma más rápida de encontrar un curso, ordenándolos por nombre.

## **11. Revisiones**

Permite manejar una “cola” de revisiones, donde se puede:

- Agregar cursos que necesiten revisión.
- Ver revisiones cambiando notas.
- Ver la cola actual.

## 12. Ver historial

Muestra los cambios que se han hecho (notas cambiadas, cursos borrados, revisiones, etc.).

## 13. Salir

Al no elegir alguna otra opción se cierra el programa.

## Ilustración del código ejecutable

---

```
1  # GESTOR DE CURSOS Y NOTAS BY NERI JUAREZ. :)
2  cursos = ["Contabilidad","Mate Discreta" , "PreCálculo", "Algoritmos", "Algebra lineal"]
3  notas = [83, 68, 61, 74, 80]
4  cola = []
5  historial = []
6
7  def menu():
8      print("""
9      >>>> MENU <<<<<
10     1. Agregar curso
11     2. Mostrar cursos
12     3. Promedio general
13     4. Aprobados / Reprobados
14     5. Buscar curso
15     6. Cambiar nota
16     7. Borrar curso
17     8. Ordenar por nota
18     9. Ordenar por nombre
19     10. Buscar binario
20     11. Revisiones
21     12. Ver historial
22     13. Salir
23     -----
24     """)
25
26 #agregar un curso>>>>
27 def agregar():
28     try:
29         n = int(input("¿Cuántos cursos quiere agregar?: "))
30     except:
```

```

31     print("Solo números por favor")
32     return
33     for i in range(n):
34         nombre = input(f"Curso {i+1}: ").strip()
35         if not nombre or nombre in cursos:
36             print("Ya existe o está vacío")
37             continue
38         try:
39             nota = float(input("Nota: "))
40             if nota < 0 or nota > 100:
41                 print("Nota fuera de rango (0-100)")
42                 continue
43         except:
44             print("Error, debe ser número")
45             continue
46         cursos.append(nombre)
47         notas.append(nota)
48     print("Cursos agregados :D")
49
50
51     #Mostrar los cursos>>>
52     def mostrar():
53         if not cursos:
54             print("No hay cursos aún")
55             return
56         for i in range(len(cursos)):
57             print(f"{i+1}. {cursos[i]} - {notas[i]} pts")
58

```

```

60     #Promedio de las notas en general>>>
61     def promedio():
62         if notas:
63             prom = sum(notas)/len(notas)
64             print(f"Promedio general: {prom:.2f}")
65         else:
66             print("No hay notas todavía")
67
68
69     #Cursos aprobados>>>
70     def aprobados():
71         for i in range(len(cursos)):
72             estado = "Aprobado" if notas[i] >= 61 else "Reprobado"
73             print(f"{cursos[i]}: {notas[i]} ({estado})")
74
75
76     #Buscar por curso>>>
77     def buscar():
78         nom = input("Nombre del curso: ").strip()
79         if nom in cursos:
80             i = cursos.index(nom)
81             print(f"{cursos[i]} - Nota: {notas[i]}")
82         else:
83             print("No se encontró ese curso")
84
85
86     #Cambiar la nota de cursos>>>
87     def cambiar_nota():
88         nom = input("Curso a cambiar nota: ").strip()
89         if nom not in cursos:

```

```

89     if nom not in cursos:
90         print("No existe")
91         return
92     i = cursos.index(nom)
93     try:
94         nueva = float(input("Nueva nota: "))
95         if 0 <= nueva <= 100: (variable) notas: list[int]
96             historial.append(f"{nom}: {notas[i]} {nueva}")
97             notas[i] = nueva
98             print("Nota cambiada")
99         else:
100             print("Fuera de rango")
101     except:
102         print("Dato no válido")
103
104
105 #Borrar un curso >>>>>
106 def borrar():
107     nom = input("Curso a borrar: ").strip()
108     if nom in cursos:
109         i = cursos.index(nom)
110         historial.append(f"Se borró {nom} ({notas[i]})")
111         cursos.pop(i)
112         notas.pop(i)
113         print("Curso eliminado")
114     else:
115         print("No encontrado")

```

```

118 #Ordenar>>>
119 def ordenar_nota():
120     for i in range(len(notas)-1):
121         for j in range(len(notas)-1-i):
122             if notas[j] > notas[j+1]:
123                 notas[j], notas[j+1] = notas[j+1], notas[j]
124                 cursos[j], cursos[j+1] = cursos[j+1], cursos[j]
125     print("Ordenado por nota :)")
126
127 def ordenar_nombre():
128     for i in range(1, len(cursos)):
129         c_act, n_act = cursos[i], notas[i]
130         j = i - 1
131         while j >= 0 and cursos[j].lower() > c_act.lower():
132             cursos[j+1], notas[j+1] = cursos[j], notas[j]
133             j -= 1
134         cursos[j+1], notas[j+1] = c_act, n_act
135     print("Ordenado por nombre :)")
136
137 def buscar_binario():
138     ordenar_nombre()
139     nom = input("Buscar curso: ").lower()
140     izq, der = 0, len(cursos)-1
141     while izq <= der:
142         mid = (izq + der)//2
143         if cursos[mid].lower() == nom:
144             print(f"{cursos[mid]} - Nota: {notas[mid]}")
145             return
146         elif nom < cursos[mid].lower():
147             der = mid - 1

```



```

148     else:
149         izq = mid + 1
150     print("No encontrado")
151
152 def revisiones():
153     while True:
154         print("\n1. Agregar  2. Atender  3. Ver cola  4. Salir")
155         op = input("Opción: ")
156         if op == "1":
157             c = input("Curso: ").strip()
158             if c in cursos: cola.append(c)
159             else: print("No existe")
160         elif op == "2":
161             if not cola:
162                 print("Nada en cola")
163                 continue
164             curso = cola.pop(0)
165             i = cursos.index(curso)
166             nueva = float(input(f"Nueva nota para {curso}: "))
167             historial.append(f"Revisión {curso}: {notas[i]} → {nueva}")
168             notas[i] = nueva
169         elif op == "3":
170             print("Cola actual:", cola or "vacía")
171         elif op == "4":
172             break
173
174 def ver_historial():
175     if historial:

```

```

176         print("\n--- Historial ---")
177         for h in reversed(historial):
178             print(h)
179     else:
180         print("Sin cambios aún")
181
182 # >>> Programa principal <<<
183 while True:
184     menu()
185     try:
186         op = int(input("Opción: "))
187     except:
188         print("Error, ingrese número")
189         continue
190
191     if op == 1: agregar()
192     elif op == 2: mostrar()
193     elif op == 3: promedio()
194     elif op == 4: aprobados()
195     elif op == 5: buscar()
196     elif op == 6: cambiar_nota()
197     elif op == 7: borrar()
198     elif op == 8: ordenar_nota()
199     elif op == 9: ordenar_nombre()
200     elif op == 10: buscar_binario()
201     elif op == 11: revisiones()
202     elif op == 12: ver_historial()
203     elif op == 13:
204         print("Saliendo..."); break

```