

Manual Técnico Gestor de Notas

Introducción

El presente documento técnico describe el funcionamiento interno del programa “Gestor de Notas”, el cual fue desarrollado en Python para administrar asignaturas y calificaciones. El programa ofrece un menú interactivo que permite al usuario agregar, modificar, ordenar y buscar cursos, además de calcular promedios y registrar un historial de cambios.

Descripción General del Sistema

El sistema trabaja mediante listas paralelas: una para las asignaturas (“mis_asignaturas”) y otra para las

notas (“mis_notas”). Cada función cumple un propósito específico, desde registrar nuevas materias hasta realizar

ordenamientos y búsquedas. También maneja errores en las entradas de usuario y proporciona mensajes informativos

en caso de no existir registros.

Requisitos

- Lenguaje: Python 3.8 o superior.
- No requiere librerías externas.
- Plataforma: Windows, macOS o Linux.
- Ejecución: Terminal o IDE (IDLE, Thonny, VSCode, PyCharm, etc.).

Estructura del Programa

El programa se compone de tres secciones fundamentales:

1. ****Datos Iniciales:**** Se declaran las listas principales y las variables globales.
2. ****Funciones:**** Contiene la lógica de cada operación (agregar, listar, ordenar, etc.).
3. ****Bucle Principal:**** Ejecuta el menú y llama a las funciones según la opción elegida por el usuario.

Explicación Detallada de Funciones

1. Función agregar_asignatura()

Agrega una nueva asignatura y su nota. Verifica duplicados y valida que la nota esté entre 0 y 100.

Fragmento de código:

```
def agregar_asignatura():
```

```
    # Lógica interna de la función
```

```
    pass
```

2. Función listar_asignaturas()

Muestra todas las asignaturas y sus notas, o un mensaje si no hay registros.

Fragmento de código:

```
def listar_asignaturas():
```

```
    # Lógica interna de la función
```

```
    pass
```

3. Función promedio()

Calcula el promedio de todas las notas registradas y lo muestra con dos decimales.

Fragmento de código:

```
def promedio():
```

```
    # Lógica interna de la función
```

```
    pass
```

4. Función aprobadas_reprobadas()

Clasifica las asignaturas en aprobadas (≥ 61) y reprobadas (< 61).

Fragmento de código:

```
def aprobadas_reprobadas():
```

```
    # Lógica interna de la función
```

```
    pass
```

5. Función buscar_asignatura()

Permite encontrar una asignatura específica mediante búsqueda lineal.

Fragmento de código:

```
def buscar_asignatura():
```

```
    # Lógica interna de la función
```

```
    pass
```

6. Función cambiar_nota()

Modifica la nota de una asignatura y guarda el cambio en el historial.

Fragmento de código:

```
def cambiar_nota():
```

```
    # Lógica interna de la función
```

```
    pass
```

7. Función borrar_asignatura()

Elimina una asignatura y su nota asociada.

Fragmento de código:

```
def borrar_asignatura():
```

```
    # Lógica interna de la función
```

```
pass
```

8. Función ordenar_por_nota()

Ordena las asignaturas de menor a mayor nota usando el método burbuja.

Fragmento de código:

```
def ordenar_por_nota():  
    # Lógica interna de la función  
  
    pass
```

9. Función ordenar_por_nombre()

Ordena alfabéticamente las asignaturas.

Fragmento de código:

```
def ordenar_por_nombre():  
    # Lógica interna de la función  
  
    pass
```

10. Función buscar_binaria()

Realiza una búsqueda binaria para localizar una asignatura específica.

Fragmento de código:

```
def buscar_binaria():  
    # Lógica interna de la función  
  
    pass
```

11. Función simular_solicitudes()

Agrega solicitudes simuladas a una lista auxiliar.

Fragmento de código:

```
def simular_solicitudes():  
    # Lógica interna de la función  
  
    Pass
```

12. Función ver_historial()

Muestra las modificaciones registradas de las notas.

Fragmento de código:

```
def ver_historial():  
    # Lógica interna de la función  
    Pass
```

Ejemplo de Ejecución

A continuación, se muestra un ejemplo de cómo luce el programa en consola al ejecutarse:

```
***** GESTOR DE NOTAS SIMPLE *****
```

1. Agregar asignatura
2. Listar asignaturas
3. Calcular promedio
4. Aprobadas / Reprobadas
5. Buscar asignatura
6. Cambiar nota
7. Borrar asignatura
8. Ordenar por nota
9. Ordenar por nombre
10. Búsqueda binaria
11. Simular solicitudes
12. Ver historial

13. Salir

Ejemplo de interacción:

Elija opción: 1

Nombre de la asignatura: Matemática

Nota: 85

Matemática agregada con nota 85.

Elija opción: 2

--- LISTA DE ASIGNATURAS ---

Matemática -> 85

Conclusión

El “Gestor de Notas” es un proyecto educativo que integra conceptos clave de programación estructurada.

Su diseño modular y el uso de algoritmos de ordenamiento y búsqueda lo convierten en un ejemplo práctico para comprender la lógica de manejo de datos en Python. Además, su estructura clara permite futuras ampliaciones como almacenamiento en archivos o interfaz gráfica.