

# Manual Técnico

**Programa:** Gestor de Notas Académicas

**Lenguaje:** Python

**Desarrollador:** Alex Juan Pablo

## Estructura general del programa:

El código está dividido por funciones, cada una manejando una opción del menú principal. Se basa principalmente en listas y diccionarios.

## Variables principales:

- **cursos:** lista de diccionarios, donde cada diccionario tiene nombre y nota.
- **Historial cambios:** lista que actúa como pila para guardar los cambios de notas.
- **Cola solicitudes:** lista usada como cola para la simulación de solicitudes.

## Funciones principales y su propósito:

FunciónDescripción:

menu()	Muestra las opciones principales del programa.
Registrar_curso ()	Permite ingresar nuevos cursos con su nota
mostrar_cursos()	Muestra todos los cursos registrados.
calcular_promedio()	Calcula el promedio general de notas.
cursos_aprobados_reprobados()	Clasifica los cursos en aprobados o reprobados.
buscar_curso_por_nombre	Busca un curso usando búsqueda lineal.
actualizar_notas()	Permite modificar una nota y guarda el cambio en el historial.
eliminar_curso()	Elimina un curso de la lista.
ordenar_cursos_por_notas()	Ordena por nota usando el método de la burbuja.
ordenar_cursos_por_nombres()	Ordena alfabéticamente usando el método de inserción.
buscar_curso_por_nombre_binaria	Busca un curso con búsqueda binaria (requiere lista ordenada).
simular_cola_solicitudes()	Administra una cola de solicitudes usando una lista.
mostrar_historial_cambios()	Muestra los cambios guardados en la pila.

## Estructuras de datos usadas:

- **Lista (list)** = usada para almacenar cursos, historial y cola.
- **Diccionario (dict)** = para guardar la información de cada curso (nombre, nota).
- **Cola simulada** = usando append() y pop(0) para entrada/salida.
- **Pila simulada** = usando append() y reversed() para mostrar el historial en orden inverso.

**Algoritmos aplicados:**

- **Búsqueda lineal:** para buscar por nombre.
- **Búsqueda binaria:** versión optimizada, requiere orden previo.
- **Ordenamiento burbuja:** para ordenar por notas.
- **Ordenamiento por inserción:** para ordenar por nombre.

**Flujo principal del programa:**

1. Se muestra el menú con menu().
  2. El usuario elige una opción numérica.
  3. El programa ejecuta la función correspondiente.
  4. Al finalizar, vuelve al menú hasta que se elija
- [0] Salir.

**Requisitos técnicos:**

Python 3.x

Consola o terminal (no requiere entorno gráfico)

No necesita librerías externas