

Universidad Mariano Gálvez de Guatemala  
Facultad de Ingeniería en Sistemas  
Ingeniería en Sistema de Información y Ciencias de la Computación  
Curso: Algoritmo  
Catedrático: Ing. Miguel Catalán  
Ciclo: Segundo siglo



**Título del trabajo: Gestor de notas**

**Nombre del alumno Nancy Paola Tubac Raxon**  
**Carné: 7590-25-27502**  
**Fecha: 24/10/2025**

## Descripción técnica general del sistema

El Gestor de Notas Académicas es una aplicación diseñada para registrar, calcular y visualizar el rendimiento académico de estudiantes. Permite ingresar asignaturas, notas parciales, promedios y determinar el estado final (aprobado/reprobado) de cada curso.

Nos permitirá llevar un mejor control de las asignaturas y sus respectivas notas que no facilitarían a llevar mejor si un estudiante, está aprobando cada curso que tiene asignado.

### Arquitectura General

El sistema sigue una estructura modular y funcional, organizada en componentes independientes que interactúan entre sí:

**Entradas de datos:** Capturan información de usuario, nombre, curso, notas parciales.

**Validación:** Verificar, que los datos ingresados sean numéricos, estén en rango y completos.

**Calculo de promedio:** Calcula el promedio ponderado o simple según las reglas definidas,

**Evaluación:** Determina si el estudiante aprueba o reprueba según el promedio obtenido.

**Visualización:** Puede guardar los datos en archivos p estructuras para revisión posterior.

Muchas más características que este gestor de notas nos da es una prevé explicación de lo que contiene nuestro gestor de notas.

## Estructura general del código

El código está estructurado en base a funciones, cada una responsable de una operación en específica del menú. El archivo principal del programa es “Gestor\_notas.py”, el cual contiene varias funciones dentro de ellas están.

- Funciones para el manejo de curso: agregar, eliminar, mostrar, actualizar.
- Funciones de análisis: promedio, aprobados, reprobados.
- Funciones para ordenamiento: burbuja e inserción.
- Funciones de búsqueda lineal y binaria.
- Funciones de estructura de datos; pilas y cola.

## Explicación del uso de listas, pilas, colas, etc.

**Las listas:** son la estructura de datos principal utilizada en el sistema. Permiten almacenar y manipular colecciones de elementos de forma ordenada y dinámica.

**Pila:** aunque no es obligatorio, una pila puede utilizarse si se desea implementar funcionalidades como deshacer la última acción (por ejemplo, eliminar la última asignatura ingresada).

**Cola:** Las colas no son esenciales en la versión básica del sistema, pero pueden ser útiles si se desea procesar estudiantes en orden de llegada o manejar múltiples usuarios.

## Justificación de los algoritmos de ordenamiento implementados

El sistema implementa algoritmos de ordenamiento para organizar los datos académicos de forma que faciliten su análisis, visualización y comprensión. Por ejemplo:

- Ordenar asignaturas alfabéticamente.
- Ordenar estudiantes por promedio (de mayor a menor).
- Ordenar notas dentro de una asignatura para análisis adicional.

## Documentación breve de cada función o módulo

- ✓ Registrar\_curso(): agregar un nuevo curso a la lista validando los datos
- ✓ Mostrar\_curso (): Muestra todos los cursos registrados
- ✓ calcular\_promedio (): calcular el promedio general de las notas
- ✓ Contar\_aprobados (): cuenta cuantos cursos fueron a probados y reprobados.
- ✓ Buscar:curso:lineal (): buscar un curso mediante coincidencia parcial.
- ✓ Actualizar\_nota (): modificar la nota de un curso dentro de la lista en existencia
- ✓ Eliminar:curso(): eliminar curso tras confirmación del usuario asignado.
- ✓ Ordenar\_burbuja(): ordena los cursos por sus notas
- ✓ Ordenar\_insercion(): ordena los cursos por nombres.
- ✓ Busqueda\_binaria(): realiza la búsqueda eficiente de una lista ordenada.
- ✓ SimulaCola(): simula el envío de solicitudes de revisión asignada.
- ✓ Mostrar\_historial(): muestra las últimas modificaciones realizadas y guardadas en la pila.

## Diagrama general del sistema o pseudocódigo principal

INICIO

DEFINIR listas: cursos, historial, cola\_revision

FUNCIÓN agregar\_curso(nombre, nota)

SI nombre está vacío

MOSTRAR "El nombre no puede estar vacío"

RETORNAR

SI nota no está entre 0 y 100

MOSTRAR "La nota debe estar entre 0 y 100"

RETORNAR

AGREGAR {"nombre": nombre, "nota": nota} a cursos

MOSTRAR mensaje de confirmación

FUNCIÓN listar\_cursos()

SI cursos no está vacío

MOSTRAR lista de cursos con índice y nota

SINO

MOSTRAR "No hay cursos registrados"

FUNCIÓN promedio\_notas\_cursos()

```
SI cursos no está vacío
  CALCULAR promedio de notas
  MOSTRAR promedio
SINO
  MOSTRAR "No hay cursos registrados"
```

```
FUNCIÓN contar_aprobados_reprobados()
  CONTAR cursos con nota >= 60 como aprobados
  CALCULAR reprobados = total - aprobados
  MOSTRAR cantidad de aprobados y reprobados
```

```
FUNCIÓN buscar_lineal()
  PEDIR nombre del curso
  BUSCAR coincidencias parciales en cursos
  SI se encuentra
    MOSTRAR curso(s) encontrado(s)
  SINO
    MOSTRAR "Curso no encontrado"
```

```
FUNCIÓN actualizar_nota()
  PEDIR nombre del curso
  BUSCAR coincidencia exacta
  SI se encuentra
    PEDIR nueva nota
    VALIDAR rango de nota
    GUARDAR cambio en historial
    ACTUALIZAR nota
    MOSTRAR confirmación
  SINO
    MOSTRAR "Curso no encontrado"
```

```
FUNCIÓN eliminar_curso()
  PEDIR nombre del curso
  BUSCAR coincidencia exacta
  SI se encuentra
    PEDIR confirmación
    SI confirmación es "s"
      GUARDAR en historial
      ELIMINAR curso
      MOSTRAR confirmación
    SINO
      MOSTRAR "Eliminación cancelada"
  SINO
    MOSTRAR "Curso no encontrado"
```

```
FUNCIÓN ordenar_por_nota()
```

APLICAR ordenamiento burbuja descendente por nota  
MOSTRAR cursos ordenados

FUNCIÓN ordenar\_por\_nombre()  
APLICAR ordenamiento por inserción ascendente por nombre  
MOSTRAR cursos ordenados

FUNCIÓN buscar\_binaria()  
ORDENAR cursos por nombre  
PEDIR nombre del curso  
APLICAR búsqueda binaria  
SI se encuentra  
MOSTRAR curso  
SINO  
MOSTRAR "Curso no encontrado"

FUNCIÓN simularColaRevisión()  
MOSTRAR instrucciones  
MIENTRAS entrada ≠ "fin"  
AGREGAR curso a cola\_revisión  
MIENTRAS cola\_revisión no esté vacía  
MOSTRAR curso en revisión

FUNCIÓN mostrar\_historial()  
SI historial está vacío  
MOSTRAR "No hay cambios registrados"  
SINO  
MOSTRAR historial en orden inverso

FUNCIÓN menu()  
MIENTRAS True  
MOSTRAR menú de opciones  
PEDIR opción  
SEGÚN opción:  
1 → agregar\_curso()  
2 → listar\_cursos()  
3 → promedio\_notas\_cursos()  
4 → contar\_aprobados\_reprobados()  
5 → buscar\_lineal()  
6 → actualizar\_nota()  
7 → eliminar\_curso()  
8 → ordenar\_por\_nota()  
9 → ordenar\_por\_nombre()  
10 → buscar\_binaria()  
11 → simularColaRevisión()  
12 → mostrar\_historial()

13 → SALIR del programa  
otro → MOSTRAR "Opción no válida"

LLAMAR menu()

FIN