

Universidad Mariano Gálvez de Guatemala  
Sede San Juan Sacatepéquez



Curso: Algoritmos  
Manual Técnico  
Inge: Miguel Catalán  
Facultad de Ingeniería en Sistemas

Kleyver Josué Lapoyeu Martínez  
Carné. 7590-24-18646

Fecha: viernes 17/10/2025

## **Manual Técnico - Gestor de Notas Académicas**

### **Descripción técnica general del sistema**

El Gestor de Notas Académicas es un sistema desarrollado en Python que permite al usuario registrar, visualizar, modificar, eliminar, analizar y organizar las calificaciones de los cursos que ha cursado. Su ejecución se realiza completamente en consola, utilizando estructuras de datos básicas como listas, pilas y colas, y algoritmos de búsqueda y ordenamiento.

### **Estructura general del código**

El código está estructurado en funciones, cada una responsable de una operación específica del menú. El archivo principal del programa es 'gestor\_notas.py', el cual contiene:

- Funciones para el manejo de cursos (agregar, mostrar, eliminar, actualizar).
- Funciones de análisis (promedio, conteo de aprobados/reprobados).
- Funciones para ordenamiento (burbuja e inserción).
- Funciones de búsqueda (lineal y binaria).
- Simulación de estructuras de datos (pila y cola).

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

- Lista principal: almacena los cursos y sus notas como pares [nombre, nota].
- Pila: se usa para almacenar los cambios realizados, permitiendo visualizar el historial en orden inverso.
- Cola: simula el proceso de revisión de notas, atendiendo las solicitudes en el orden en que fueron ingresadas.

### **Justificación de los algoritmos de ordenamiento implementados**

Se implementaron dos algoritmos clásicos:

- Ordenamiento Burbuja: utilizado para ordenar los cursos por nota de forma descendente, por su facilidad de implementación.
- Ordenamiento por Inserción: utilizado para ordenar los cursos alfabéticamente, eficiente para listas pequeñas y casi ordenadas.

### **Documentación breve de cada función o módulo**

- registrar\_curso(): agrega un nuevo curso a la lista validando los datos.
- mostrar\_cursos(): muestra todos los cursos registrados.
- calcular\_promedio(): calcula el promedio general de las notas.
- contar\_aprobados(): cuenta cuántos cursos fueron aprobados o reprobados.
- buscar\_curso\_lineal(): busca un curso mediante coincidencia parcial.
- actualizar\_nota(): modifica la nota de un curso existente.
- eliminar\_curso(): elimina un curso tras confirmación del usuario.
- ordenar\_burbuja(): ordena los cursos por nota.
- ordenar\_insercion(): ordena los cursos por nombre.
- buscar\_binaria(): realiza una búsqueda eficiente en una lista ordenada.

- `simularCola()`: simula el envío de solicitudes de revisión.
- `mostrarHistorial()`: muestra las últimas modificaciones guardadas en la pila.

### Diagrama general del sistema o pseudocódigo principal

INICIO

DEFINIR `lista_cursos` COMO LISTA VACÍA  
 DEFINIR `pila_historial` COMO LISTA VACÍA  
 DEFINIR `cola_revision` COMO LISTA VACÍA

MIENTRAS VERDADERO HACER

IMPRIMIR "===== GESTOR DE NOTAS ACADÉMICAS ====="

IMPRIMIR "1. Registrar nuevo curso"

IMPRIMIR "2. Mostrar todos los cursos y notas"

IMPRIMIR "3. Calcular promedio general"

IMPRIMIR "4. Contar cursos aprobados y reprobados"

IMPRIMIR "5. Buscar curso por nombre (búsqueda lineal)"

IMPRIMIR "6. Actualizar nota de un curso"

IMPRIMIR "7. Eliminar un curso"

IMPRIMIR "8. Ordenar cursos por nota (burbuja)"

IMPRIMIR "9. Ordenar cursos por nombre (inserción)"

IMPRIMIR "10. Buscar curso por nombre (búsqueda binaria)"

IMPRIMIR "11. Simular cola de revisión de cursos"

IMPRIMIR "12. Mostrar historial de cambios (pila)"

IMPRIMIR "13. Salir"

IMPRIMIR "Seleccione una opción: "

LEER `opcion`

SEGÚN `opcion` HACER

CASO 1:

IMPRIMIR "Ingrese el nombre del curso:"

LEER `nombre`

IMPRIMIR "Ingrese la nota obtenida (0 a 100):"

LEER `nota`

SI `nota >= 0` Y `nota <= 100` ENTONCES

AGREGAR {`nombre`, `nota`} A `lista_cursos`

IMPRIMIR "Curso registrado con éxito."

SINO

IMPRIMIR "Nota inválida."

FIN\_SI

CASO 2:

SI `lista_cursos` ESTÁ VACÍA ENTONCES

```
    IMPRIMIR "No hay cursos registrados."  
SINO  
    PARA cada curso EN lista_cursos HACER  
        IMPRIMIR curso.nombre, "-", curso.nota  
    FIN_PARA  
FIN_SI
```

CASO 3:

```
SI lista_cursos ESTÁ VACÍA ENTONCES  
    IMPRIMIR "No hay cursos para calcular promedio."  
SINO  
    suma ← 0  
    PARA cada curso EN lista_cursos HACER  
        suma ← suma + curso.nota  
    FIN_PARA  
    promedio ← suma / LONGITUD(lista_cursos)  
    IMPRIMIR "Promedio general:", promedio  
FIN_SI
```

CASO 4:

```
aprobados ← 0  
reprobados ← 0  
PARA cada curso EN lista_cursos HACER  
    SI curso.nota >= 60 ENTONCES  
        aprobados ← aprobados + 1  
    SINO  
        reprobados ← reprobados + 1  
    FIN_SI  
FIN_PARA  
IMPRIMIR "Cursos aprobados:", aprobados  
IMPRIMIR "Cursos reprobados:", reprobados
```

CASO 5:

```
IMPRIMIR "Ingrese el nombre del curso a buscar:"  
LEER nombre_buscar  
encontrado ← FALSO  
PARA cada curso EN lista_cursos HACER  
    SI nombre_buscar ESTÁ EN curso.nombre (ignorando mayúsculas) ENTONCES  
        IMPRIMIR "Curso encontrado:", curso.nombre, "-", curso.nota  
        encontrado ← VERDADERO  
    FIN_SI  
FIN_PARA
```

```
SI NO encontrado ENTONCES
    IMPRIMIR "Curso no encontrado."
FIN_SI
```

CASO 6:

```
IMPRIMIR "Ingrese el nombre del curso a actualizar:"
LEER nombre_actualizar
PARA cada curso EN lista_cursos HACER
    SI curso.nombre = nombre_actualizar ENTONCES
        nota_anterior ← curso.nota
        IMPRIMIR "Ingrese nueva nota:"
        LEER nueva_nota
        SI nueva_nota >= 0 Y nueva_nota <= 100 ENTONCES
            curso.nota ← nueva_nota
            AGREGAR ("Se actualizó: " + nombre_actualizar + " (" + nota_anterior + " → "
+ nueva_nota + ")") A pila_historial
            IMPRIMIR "Nota actualizada."
        SINO
            IMPRIMIR "Nota inválida."
        FIN_SI
    FIN_SI
FIN_PARA
```

CASO 7:

```
IMPRIMIR "Ingrese el nombre del curso a eliminar:"
LEER nombre_eliminar
PARA cada curso EN lista_cursos HACER
    SI curso.nombre = nombre_eliminar ENTONCES
        AGREGAR ("Se eliminó: " + curso.nombre + " - Nota: " + curso.nota) A
pila_historial
        ELIMINAR curso DE lista_cursos
        IMPRIMIR "Curso eliminado."
    FIN_SI
FIN_PARA
```

CASO 8:

```
// Ordenamiento Burbuja por nota (descendente)
PARA i DESDE 0 HASTA LONGITUD(lista_cursos)-2 HACER
    PARA j DESDE 0 HASTA LONGITUD(lista_cursos)-i-2 HACER
        SI lista_cursos[j].nota < lista_cursos[j+1].nota ENTONCES
            INTERCAMBIAR lista_cursos[j] CON lista_cursos[j+1]
        FIN_SI
    FIN_PARA
```

```
FIN_PARA
IMPRIMIR "Cursos ordenados por nota."
```

CASO 9:

```
// Ordenamiento por Inserción (por nombre)
PARA i DESDE 1 HASTA LONGITUD(lista_cursos)-1 HACER
    actual ← lista_cursos[i]
    j ← i - 1
    MIENTRAS j >= 0 Y lista_cursos[j].nombre > actual.nombre HACER
        lista_cursos[j+1] ← lista_cursos[j]
        j ← j - 1
    FIN_MIENTRAS
    lista_cursos[j+1] ← actual
FIN_PARA
IMPRIMIR "Cursos ordenados por nombre."
```

CASO 10:

```
IMPRIMIR "Ingrese el nombre del curso a buscar (búsqueda binaria):"
LEER nombre_buscar
inicio ← 0
fin ← LONGITUD(lista_cursos) - 1
encontrado ← FALSO
MIENTRAS inicio <= fin Y NO encontrado HACER
    medio ← (inicio + fin) // 2
    SI lista_cursos[medio].nombre = nombre_buscar ENTONCES
        IMPRIMIR "Curso encontrado:", lista_cursos[medio].nombre, "-",
lista_cursos[medio].nota
        encontrado ← VERDADERO
    SINO SI lista_cursos[medio].nombre > nombre_buscar ENTONCES
        fin ← medio - 1
    SINO
        inicio ← medio + 1
    FIN_SI
FIN_MIENTRAS
SI NO encontrado ENTONCES
    IMPRIMIR "Curso no encontrado."
FIN_SI
```

CASO 11:

```
LIMPIAR cola_revision
IMPRIMIR "Ingrese cursos para revisión (escriba 'fin' para terminar):"
MIENTRAS VERDADERO HACER
```

```
    LEER curso_rev
    SI curso_rev = "fin" ENTONCES
        SALIR_DEL_BUCLE
    FIN_SI
    AGREGAR curso_rev A cola_revision
FIN_MIENTRAS
IMPRIMIR "Procesando solicitudes:"
MIENTRAS cola_revision NO ESTÉ VACÍA HACER
    curso ← PRIMERO DE cola_revision
    IMPRIMIR "Revisando:", curso
    ELIMINAR PRIMERO DE cola_revision
FIN_MIENTRAS
```

CASO 12:

```
    SI pila_historial ESTÁ VACÍA ENTONCES
        IMPRIMIR "No hay cambios recientes."
    SINO
        IMPRIMIR "Historial de cambios:"
        PARA i DESDE LONGITUD(pila_historial)-1 HASTA 0 PASO -1 HACER
            IMPRIMIR pila_historial[i]
        FIN_PARA
    FIN_SI
```

CASO 13:

```
    IMPRIMIR "Gracias por usar el Gestor de Notas Académicas. ¡Hasta pronto!"
    SALIR_DEL_PROGRAMA
```

DEFECTO:

```
    IMPRIMIR "Opción inválida, intente de nuevo."
    FIN_SEGÚN
    FIN_MIENTRAS
```

FIN