

Universidad Mariano Gálvez de Guatemala
Facultad de Ingeniería en Sistemas
Ingeniería en Sistema de Información y Ciencias de la Computación
Curso: Algoritmos



Manual Técnico

Gestor de Notas Académicas

Nombre del Estudiante

Maria José Vásquez Blanco

Catedrático

Ing. Miguel Catalán

San Juan Sacatepéquez, Guatemala

2025

Descripción técnica general del sistema

El Gestor de Notas Académicas es un programa hecho en Python. Sirve para que un estudiante maneje sus calificaciones. El sistema tiene como objetivo ser simple y fácil de usar. Permite registrar, buscar, cambiar, borrar, consultar y analizar notas. La interfaz funciona en una línea de comandos.

El sistema usa una estructura que se basa en funciones. Esto hace que sea fácil entender, mantener y agregar nuevas funciones. Cada tarea principal, como agregar cursos, calcular promedios, ordenar o buscar datos, está en una función diferente. Esto ayuda a que el código sea y se pueda usar otra vez si es necesario.

Los datos se guardan en listas, pilas y colas. Estas estructuras de datos son propias de Python. Permiten manejar la información de forma rápida y fácil. No se usan bases de datos ni librerías externas.

Estructura general del código

Variables:

- cursos: lista principal donde se almacenan los cursos y sus notas.
- historial_cambios: pila que guarda las modificaciones o eliminaciones.

Funciones:

- Validación de notas (validar_nota)
- Visualización de cursos (mostrar_cursos)

Funciones principales:

- Registro, actualización, eliminación y búsqueda de cursos.
- Cálculo de promedio y conteo de aprobados/reprobados.
- Ordenamientos (burbuja e inserción).
- Simulación de cola de revisión y pila de historial.

Menú principal:

- Control de flujo mediante un ciclo while True.
- Permite la navegación interactiva del usuario.

El sistema está hecho con un enfoque modular, donde cada función realiza una tarea independiente. Esto ayuda a que el código sea más claro, fácil de mantener y reutilizar. La interacción con el usuario se hace a través de un menú principal que se muestra una y otra vez. El usuario puede elegir la opción que quiera hasta que decida salir del programa. El flujo de trabajo usa un ciclo while, que mantiene abierto el menú hasta que el usuario elige salir. Cada opción del menú tiene una función específica, lo que hace que el sistema sea sencillo de entender y usar.

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

- Lista de cursos (Cursos)

Es la estructura principal del sistema. Consiste en una lista de diccionarios, donde cada uno representa un curso con dos claves: Nombre (cadena de texto) y Nota (entero entre 0 y 100). Esta lista permite guardar los cursos que el usuario registra y sirve como base para mostrar, buscar, ordenar y analizar.

- Pila de historial de cambios (Historial)

Esta estructura simula una pila (LIFO: Last In, First Out) usando una lista de Python. Cada vez que se actualiza o elimina un curso, se guarda un mensaje en esta pila. La opción del menú permite ver los cambios más recientes en orden inverso, ayudando a seguir las modificaciones.

- Cola de solicitudes de revisión (Cola y revisión)

Se implementa como una lista que funciona como una cola (FIFO: First In, First Out). El usuario puede ingresar los nombres de los cursos para enviarlos a revisión, y el sistema los procesa en orden. Esto es solo una simulación y no cambia los datos reales del sistema.

- Menú interactivo (While)

El menú principal aparece en la consola y permite al usuario escoger entre trece opciones numeradas. Cada opción activa una función que realiza la tarea correspondiente. El menú se repite hasta que el usuario elige la opción salir.

Justificación de los algoritmos de ordenamiento implementados

El gestor de notas académico cuenta con la implementación manual de dos algoritmos de ordenamiento y dos métodos de búsqueda:

- **Ordenamiento burbuja:** Este algoritmo se utiliza para organizar los cursos según la nota en orden descendente. Aunque es bastante simple, resulta efectivo para listas pequeñas y ayuda a entender el concepto de intercambio entre elementos adyacentes.
- **Ordenamiento por inserción:** Se aplica para ordenar los cursos alfabéticamente por su nombre. Este método es bastante eficiente para listas que ya están parcialmente ordenadas y es fácil de implementar, lo que lo convierte en una opción ideal para esta aplicación.

- **Búsqueda lineal:** Este método permite localizar un curso mediante una coincidencia parcial del nombre, sin importar si se usan mayúsculas o minúsculas. Es especialmente útil cuando el usuario no recuerda el nombre exacto del curso.
- **Búsqueda binaria:** Se utiliza para encontrar un curso mediante una coincidencia exacta del nombre, pero solo después de haber ordenado previamente la lista por nombre. Este método es más eficiente que la búsqueda lineal en listas que ya están ordenadas.

Documentación breve de cada función o módulo

- registrar curso: Solicita al usuario el nombre del curso y la nota obtenida. Valida los datos y agrega el curso a la lista principal cursos.
- mostrar cursos: Muestra todos los cursos registrados con sus respectivas notas. Si no hay cursos, informa al usuario.
- calcular promedio: Calcula el promedio general de todas las notas registradas y lo muestra redondeado a dos decimales.
- contar aprobados: Cuenta cuántos cursos tienen nota mayor o igual a 60 (aprobados) y cuántos están por debajo (reprobados).
- buscar lineal: Realiza una búsqueda por coincidencia parcial del nombre del curso (sin importar mayúsculas/minúsculas) y muestra el resultado si lo encuentra.
- actualizar nota: Permite modificar la nota de un curso existente. Valida la nueva nota y guarda el cambio en la pila historial.
- eliminar curso: Elimina un curso de la lista tras confirmar con el usuario. Registra la eliminación en la pila historial.
- Ordenar por nota: Ordena los cursos en orden descendente según la nota utilizando el algoritmo de ordenamiento burbuja.
- Ordenar por nombre: Ordena los cursos alfabéticamente por nombre utilizando el algoritmo de ordenamiento por inserción.
- buscar binaria: Realiza una búsqueda binaria exacta del nombre del curso. Requiere que la lista esté previamente ordenada por nombre.
- Simular cola: Simula una cola de solicitudes de revisión. El usuario ingresa cursos y el sistema los procesa en orden de ingreso.
- Mostrar historial: Muestra los últimos cambios realizados (actualizaciones y eliminaciones) en orden inverso, simulando una pila.
- menú: Controla el flujo principal del programa. Muestra el menú interactivo y llama a la función correspondiente según la opción seleccionada.

Diagrama general del sistema o pseudocódigo principal

INICIO

REPETIR

Mostrar menú principal

Leer opción del usuario Hacer

SEGÚN OPCIÓN: 1 → registrar_curso()

2 → mostrar_cursos()

3 → calcular_promedio()

4 → contar_aprobados()

5 → buscar_lineal()

6 → actualizar_nota()

7 → eliminar_curso()

8 → ordenar_por_nota()

9 → ordenar_por_nombre()

10 → buscar_binaria()

11 → simular_cola()

12 → mostrar_historial()

13 → SALIR

FIN SEGÚN

HASTA que opción = 13

// Solicitarle al usuario cuántas notas desea ingresar

ESCRIBIR "¿Cuántas notas desea ingresar?"

LEER cantidadNotas

// Crear una lista vacía para almacenar las notas que el usuario desea ingresar

DEFINIR listaNotas COMO LISTA VACÍA

// Validar que la cantidad ingresada sea un número positivo y no ingrese un numero negativo

MIENTRAS NO ES_NUMERO(cantidadNotas) O cantidadNotas <= 0 HACER

ESCRIBIR "Por favor, ingrese un número mayor que cero."

LEER cantidadNotas

FIN MIENTRAS

// Repetir el proceso de ingreso de notas que desea el usuario

PARA i DESDE 1 HASTA cantidadNotas HACER

ESCRIBIR "Ingrese la nota #" + i

LEER nota

// Validar que ingrese una nota y sea numérica

MIENTRAS nota ES VACÍA O NO ES_NUMERO(nota) HACER

ESCRIBIR " Ingrese un valor numérico."

LEER nota

FIN MIENTRAS

// Que la nota ingresada por el usuario agregarla a la lista

AGREGAR nota A listaNotas

FIN PARA

// Calcular el promedio de las notas ingresadas

suma \leftarrow 0

PARA CADA nota EN listaNotas HACER

suma \leftarrow suma + nota

FIN PARA

promedio \leftarrow suma / LONGITUD(listaNotas)

// Mostrar el promedio de las notas al usuario

Fin