



Manual Técnico:

Gestor de Notas Académicas

Nombre:
Marcos Samuel Choc Gómez

Introducción

El gestor de Notas académicas es un programa desarrollado en el lenguaje de programación Python, diseñada para facilitar la administración de cursos y sus respectivas calificaciones de forma práctica y eficiente, este sistema permite registrar, buscar, actualizar y eliminar cursos, así como aplicar diferentes tipos de ordenamiento, calcular promedios generales y determinar cursos aprobados o reprobados, además incluye funciones de simulación de solicitudes y un historial de cambios, lo que garantiza un mejor control y seguimiento del estado académico.

Estructura general de código:

La lista principal “cursos” almacena toda la información correspondiente a los cursos registrados juntos con sus calificaciones.

Cada función esta definida de forma independiente, lo que permite mantener una estructura ordenada.

El menú principal se ejecuta dentro de un **bucle while**, asegurando que el programa permanezca activo hasta que el usuario seleccione la opción de salir.

Uso de colas, pilas y listas:

Cola (FIFO): Se utiliza para simular solicitudes de revision, las cuales se gestionan siguiendo el principio First In, First Out (primero en entrar, primero en salir), esto garantiza que las peticiones sean atendidas en el mismo orden en que fueron ingresadas por los usuarios.

Pila (LIFO): Se emplea para registrar los cambios realizados en el sistema, tales como el registro, la actualización o la eliminación de cursos, su comportamiento Last In, First Out(último en entrar, primero en salir) permite llevar un control ordenado del historial de modificaciones, facilitando la revisión de las acciones más recientes.

Lista (cursos): Se utiliza para almacenar y recorrer todos los cursos registrados junto con sus respectivas calificaciones, gracias a esta estructura, es posible acceder fácilmente a los elementos mediante su posición y realizar operación como búsqueda, actualización o eliminación.

El uso combinado de estas estructuras de datos (listas, pilas y colas) optimiza el manejo de la información dentro del programa, permitiendo un flujo lógico y organizado de los datos, además mejora la eficiencia en la operación de registro, control de historial y gestión de solicitudes.

Justificación de los algoritmos de ordenamiento:

Burbuja: Se emplea para ordenar los cursos según sus calificaciones de manera sencilla. Este algoritmo es fácil de implementar y resulta adecuado para listas pequeñas, donde su eficiencia es suficiente y permite observar de forma clara el proceso de comparación e intercambio de elementos.

Inserción: se utiliza para ordenar los cursos alfabéticamente por nombre. Este método es eficiente para listas que ya están parcialmente ordenadas, ya que inserta cada elemento en la posición correcta sin necesidad de recorrer toda la lista en cada paso, optimizando así el rendimiento en casos comunes del programa.

La selección de estos algoritmos se fundamenta en la simplicidad y claridad que aportan, lo que facilita la comprensión y mantenimiento del código, además permiten demostrar de manera práctica los conceptos de ordenamiento de estructuras de datos básicas.

Documentación brece de cada función:

registrar_curso ():

Registra cursos y notas, validando que la nota esté entre 0-100.

mostrar_curso():

Muestra todos los cursos con sus notas.

Promedio_general():

Calcula el promedio general de todas las notas.

cursos_aprobados_reprobados():

Cuenta cursos aprobados y reprobados.

buscar_curso ():

Búsqueda lineal por nombre.

actualizar_nota ():

Actualiza la nota de un curso.

eliminar_curso():

Elimina un curso de la lista tras confirmación.

ordenar_burbuja():

Ordena cursos por nota.

ordena_por_insercion ():

Ordena cursos por nombre.

buscar_curso_binario ():

Búsqueda binaria por nombre en lista ordenada.

simular_solicitudes ():

Simula una cola FIFO para solicitudes de revisión.

mostrar_historial ():

Muestra historial de cambios usando una pila LIFO.

Diagrama general del sistema o pseudocódigo principal

INICIO

```
lista cursos = []  
ciclo while True  
mostrar menú  
leer opción  
SI opción == 1: registrar_curso()  
SI opción == 2: mostrar_curso()  
SI opción == 3: promedio_general ()  
SI opción == 4: cursos_aprobados_reprobados()  
SI opción == 5: buscar_curso ()  
SI opción == 6: actualizar_nota()  
SI opción == 7: eliminar_curso()  
SI opción == 8: ordenar_burbuja()  
SI opción == 9: ordenar_insercion()  
SI opción == 10: buscar_curso_binario ()  
SI opción == 11: simular_solicitudes ()  
SI opción == 12: mostrar_historial ()  
SI opción == 13: Salir  
FIN
```