

Manual Técnico

Gestor de Notas Académicas | Herber Geovanny Mares Velásquez

Descripción técnica general del sistema

- El sistema **Gestor de Notas Académicas** es una aplicación de consola desarrollada en **Python**, diseñada para que un estudiante pueda:
- Registrar cursos y notas obtenidas.
- Visualizar, actualizar y eliminar cursos.
- Analizar calificaciones con funciones de promedio, conteo de aprobados/reprobados y ordenamiento.
- Mantener un historial de cambios mediante una pila.
- Simular solicitudes de revisión mediante una cola.
- El sistema funciona mediante un **menú interactivo** que se repite hasta que el usuario decide salir.
- **Objetivo:** Proporcionar un control completo y fácil de usar de las notas académicas, utilizando estructuras de datos y algoritmos eficientes.

Estructura general del código

- **Variables globales:**

- cursos: lista principal que guarda cada curso como un diccionario {nombre, nota}.
- historial: pila que almacena los últimos cambios realizados.

- **Funciones principales:**

- Registro, visualización, actualización y eliminación de cursos.
- Cálculo de promedio general y conteo de aprobados/reprobados.
- Ordenamiento por nota (burbuja) y por nombre (inserción).
- Búsqueda lineal y binaria de cursos.
- Simulación de cola para solicitudes de revisión.
- Mostrar historial de cambios.

- **Menú interactivo:**

- Controla el flujo del programa, permitiendo al usuario seleccionar la acción deseada.
- Se repite hasta que se selecciona la opción “Salir”.

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

- **Listas:**
Se usan para almacenar cursos y notas, permitiendo agregar, eliminar y modificar elementos fácilmente.
- **Pila (historial):**
Permite registrar los últimos cambios y mostrarlos en orden inverso, usando `append()` y `reversed()`.
- **Cola (simulación de revisión):**
Gestiona solicitudes de revisión en orden de llegada. Los elementos se agregan al final (`append`) y se procesan en orden secuencial.
- **Ventaja:** Estas estructuras permiten un manejo eficiente y organizado de la información del estudiante.

Justificación de los algoritmos de ordenamiento implementados

- **Burbuja**
- **Objetivo:** Ordenar cursos por **nota descendente**.
- **Cómo funciona:**
 - Compara elementos adyacentes de la lista y los intercambia si están en el orden incorrecto.
 - Se repite este proceso varias veces hasta que no se necesiten más intercambios.
 - Cada pasada “burbujea” el valor más grande hacia la posición correcta.
- **Ventajas:**
 - Fácil de implementar y entender.
 - Ideal para listas pequeñas o cuando se quiere un método simple sin
- **Inserción**
- **Objetivo:** Ordenar cursos **alfabéticamente por nombre**.
- **Cómo funciona:**
 - Toma cada elemento de la lista y lo inserta en su posición correcta dentro de la sub lista ya ordenada.
 - Comienza desde el segundo elemento y compara hacia atrás hasta encontrar el lugar adecuado.
- **Ventajas:**
 - Muy eficiente para listas parcialmente ordenadas.
 - Estable y fácil de implementar.
 - Permite ordenar en tiempo relativamente rápido si la lista no es muy grande.

Documentación breve de cada función o módulo

Gestión de cursos

- `registrar_curso()`:
Solicita nombre y nota del curso, valida los datos y lo agrega a la lista de cursos.
- `mostrar_cursos()`:
Muestra todos los cursos registrados con sus notas.
- `calcular_promedio()`:
Calcula el promedio general de todas las notas.
- `contar_aprobados_reprobados()`:
Cuenta los cursos aprobados (≥ 60) y reprobados (< 60).
- `actualizar_nota()`:
Modifica la nota de un curso existente y guarda el cambio en el historial.
- `eliminar_curso()`:
Elimina un curso de la lista, previa confirmación, y guarda el cambio en el historial.

Ordenamiento y búsqueda

- `ordenar_por_nota()`:
Ordena los cursos por nota usando burbuja (descendente).
- `ordenar_por_nombre()`:
Ordena los cursos alfabéticamente usando inserción.
- `busqueda_lineal(nombre)`:
Busca un curso por nombre secuencialmente.
- `busqueda_binaria()`:
Busca un curso por nombre en una lista ordenada por nombre.

Estructuras adicionales

- `simularColaRevision()`:
Simula la cola de solicitudes de revisión de cursos.
- `mostrar_historial()`:
Muestra los últimos cambios registrados en la pila de historial.

Pseudocódigo General del sistema

- INICIO

- Mientras el usuario no seleccione 'Salir':

- Mostrar menú principal

- Leer opción seleccionada

- Segun la opción:

- 1 -> Registrar curso

- 2 -> Mostrar cursos

- 3 -> Calcular promedio

- 4 -> Contar aprobados y reprobados

- 5 -> Buscar curso

- 6 -> Actualizar nota

- 7 -> Eliminar curso

- 8 -> Ordenar por nota

- 9 -> Ordenar por nombre

- 10 -> Buscar binaria

- 11 -> Simular cola de revisión

- 12 -> Mostrar historial

- FIN Mientras

- FIN