

```
<!--Proyecto Notas Academicas-->
```

Manual
Técnico {

```
<Por="Jairo Omar León"/>
```

}



Contenidos

01

-Descripción técnica

02

-Estructura general del código

03

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

04

-Justificación de los algoritmos
de ordenamiento implementados.

05

-Documentación breve de cada

06

función o módulo.

07

Base de datos

Servidor

Planificación

1. Descripción técnica general {

Este programa funciona en la línea de comandos y sirve para gestionar cursos y sus notas. Está hecho en Python y no usa bibliotecas externas, solo herramientas básicas del lenguaje.

2. Estructura general del código {

- `proyectofuncionalcompleto.py`: archivo principal donde están todas las funciones y el menú..
- `pseudocodigo.txt`: contiene el pseudocódigo con el flujo de todo el programa.

}

3. Uso de estructuras {

LISTAS

nombres y notas van juntas, el mismo índice representa un curso y su nota.

PILA

historial_cambios guarda los cambios realizados, mostrando los últimos primero.

COLA

en
simularColaRevisión()
se usa una lista cola que
procesa los cursos en
orden de llegada (FIFO).

}

4. Algoritmos de ordenamiento implementados

BURBUJA

(ordenar_por_nota): ordena las notas de mayor a menor. Es sencillo y fácil de entender, ideal para aprender.

INSERCIÓN

(ordenar_por_nombre): ordena los nombres de forma alfabética. Funciona bien para listas pequeñas y mantiene el orden de elementos iguales.

}

5. Funciones principales {

- `mostrar_cursos()`: muestra todos los cursos y sus notas.
- `registrar_curso()`: pide nombre y nota y los agrega a las listas.
- `actualizar_nota()`: cambia la nota de un curso existente.
- `eliminar_curso()`: elimina un curso de la lista.
- `ordenar_por_nota()`: ordena los cursos de mayor a menor según la nota.
- `ordenar_por_nombre()`: ordena los cursos alfabéticamente.
- `buscar_curso_binario()`: busca un curso por nombre usando búsqueda binaria (la lista debe estar ordenada).
- `simular_cola_revision()`: procesa una cola de cursos en orden de llegada.
- `mostrar_historial()`: muestra los últimos cambios realizados, desde el más reciente.
- `salir()`: muestra un mensaje de despedida y cierra el programa.

}

6. Diagrama general / Pseudocódigo {

Revisar pseudocodigo.txt para ver el flujo completo del programa.

7. Consideraciones adicionales {

- Todas las entradas se hacen por línea de comandos.
- No guarda los datos en archivos.

}

```
<!--Algoritmos-->
```

Gracias {

```
<Por="JAIRO OMAR LEÓN"/>
```

}