

Universidad Mariano Gálvez

Facultad DE INGENIERIA EN SISTEMAS DE INFORMACION
DE LA COMPUTACION
Curso: ALGORITMOS



Borrador de Manual Técnico

NOMBRE: José Eduardo
APELLIDOS: Chavac Yoc
CARNET: 7590-25-11288
FECHA: 17/10/2025

Universidad Mariano Gálvez

INTRODUCCION

El Sistema de Gestión Académica es un programa de consola desarrollado en Python que sirve para administrar de forma sencilla los cursos, y sus calificaciones. Este sistema fue creado con el objetivo de facilitar la organización de la información dentro de un entorno académico, haciendo que todo sea más práctico y ordenado.

Dentro del proyecto se utilizaron estructuras de datos básicas, como listas y diccionarios, además de algoritmos de ordenamiento que permiten organizar los datos de manera rápida y eficiente. Gracias a esto, el sistema puede mostrar la información de forma clara y sin errores.

En pocas palabras, este sistema busca demostrar cómo con conocimientos de programación se pueden crear herramientas útiles para la educación. Además, representa una buena forma de poner en práctica los conceptos aprendidos en el curso, combinando la lógica, la programación y la organización de datos en un solo proyecto funcional.

Arquitectura del Sistema

El proyecto está desarrollado en un solo archivo el cual lo llame `PROYECTO_FINAL.py` con el fin de mantenerlo simple y fácil de ejecutar sin necesidad de instalar otros módulos o dependencias externas. Esta estructura nos permite que cualquier persona la pueda probar el programa rápidamente desde la consola de Python

La arquitectura del sistema se basa principalmente en el uso de estructuras de datos y en la organización por secciones lógicas dentro del código.

Estructuras de Datos Principales:

Para manejar la información se utilizan listas de diccionarios donde se almacenan los datos de los cursos y de los estudiantes incluyendo sus nombres, notas y demás detalles también se implementa una lista tipo pila (stack) para guardar el historial de acciones realizadas y otra lista tipo cola (queue) que se usa para gestionar las solicitudes de revisión de notas. Estas estructuras permiten que el sistema funcione de manera ordenada y eficiente.

Módulos de Funcionalidad:

El código está dividido por partes que cumplen diferentes funciones:

Gestión de Cursos: Permite crear, leer y eliminar cursos (CRUD).

Gestión de Estudiantes y Notas: Se encarga de registrar estudiantes, asignarles calificaciones y mostrarlas

Algoritmos de Ordenamiento: Se utilizan para organizar la información, por ejemplo, ordenar estudiantes por nota o por nombre.

Manejo de Pila y Cola: Permiten registrar el historial de acciones y procesar solicitudes pendientes.

Menú Principal y Control de Flujo: Es la parte que conecta todo el sistema y guía al usuario paso a paso para realizar las distintas operaciones.

Universidad Mariano Gálvez

Estructuras de Datos

El sistema utiliza diferentes estructuras de datos que permiten organizar y manejar la información de manera clara y eficiente. Cada una cumple una función específica dentro del programa y está pensada para facilitar las operaciones de búsqueda, registro y actualización de los datos.

cursos: Una lista donde cada elemento es un diccionario que representa un curso.

```
{"nombre": "Algoritmos", "estudiantes": [...]}
```

estudiantes (dentro de cada curso): Una lista donde cada elemento es un diccionario que representa a un estudiante.

```
{"nombre": "Juan Pérez", "notas": [85, 90, 78], "promedio": 84.33}
```

historial_acciones (Pila): Una lista que funciona como una pila (LIFO - Last-In, First-Out). Se utiliza `append()` para agregar acciones y se lee en orden inverso para mostrar el historial más reciente primero.

solicitudes_revision (Cola): Una lista que funciona como una cola (FIFO - First-In, First-Out). Se utiliza `append()` para agregar solicitudes y `pop(0)` para procesar la más antigua.

Algoritmos Implementados

Dentro del sistema implemente dos algoritmos de ordenamiento básicos: el Ordenamiento Burbuja y el Ordenamiento por Inserción. Ambos fueron elegidos por su sencillez y por permitir comprender mejor el funcionamiento interno de los métodos de organización de datos en Python.

Ordenamiento Burbuja (`ordenamiento_burbuja`):

Propósito: Este algoritmo se utiliza para ordenar la lista de estudiantes alfabéticamente por su nombre.

Lógica: El proceso consiste en comparar pares de elementos que están uno al lado del otro y, si no están en el orden correcto, se intercambian. Este ciclo se repite varias veces hasta que toda la lista queda ordenada.

Aunque es un método muy fácil de entender e implementar, no es muy eficiente cuando se trabaja con una gran cantidad de datos, ya que su complejidad es de $O(n^2)$. Sin embargo, para este proyecto resulta ideal, ya que las listas manejadas son pequeñas y permiten observar claramente el proceso de ordenamiento paso a paso.

Ordenamiento por Inserción (`ordenamiento_insercion`):

Propósito: Este algoritmo se emplea para ordenar a los estudiantes según su promedio, de mayor a menor.

Lógica: Funciona construyendo poco a poco una sublista ordenada. Recorre los elementos de la lista original, y cada vez que toma un nuevo valor, lo inserta en la posición correcta dentro de la parte que ya está ordenada.

Este método es más eficiente cuando las listas son pequeñas o están parcialmente ordenadas, y también tiene una complejidad de $O(n^2)$. Aun así, resulta muy útil dentro del sistema, ya que permite organizar rápidamente los promedios de los estudiantes sin necesidad de usar funciones externas.

Universidad Mariano Gálvez

En conclusión, ambos algoritmos fueron implementados con fines didácticos, demostrando cómo se pueden aplicar conceptos de lógica y estructuras de control para resolver problemas reales de organización de información dentro de un programa académico.

Despliegue y Dependencias

El script solo requiere una instalación estándar de Python 3. No tiene dependencias externas. Para ejecutarlo, simplemente corra el comando en la terminal