

# Universidad Mariano Gálvez

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



## Manual Técnico

NOMBRE: José Eduardo  
APELLIDOS: Chavac Yoc  
CARNET: 7590-25-11288  
FECHA: 24/10/2025

# PROYECTO: GESTOR DE NOTAS ACADÉMICAS

DESARROLLADOR: JOSÉ EDUARDO CHAVAC YOC

FECHA: 24/10/2025

## Técnica General del Sistema

El "Gestor de Notas Académicas" es una aplicación de consola desarrollada en Python cuyo propósito es simular un sistema básico de administración de cursos y notas. Su objetivo principal es demostrar la implementación práctica de diversas estructuras de datos fundamentales (listas como estructuras base, Pila, Cola) y algoritmos clásicos de búsqueda y ordenamiento (Burbuja, Inserción, Lineal, Binaria) dentro de un menú interactivo y funcional.

## Estructura General del Código

El código está organizado de manera modular y sigue una estructura de programa principal monolítico:

**Variables Globales y Constantes:** Inicialización de las listas de datos (cursos, historial\_acciones, solicitudes\_revision) y la constante de aprobación (NOTA\_APROBACION).

**Funciones de Estructuras de Datos:** Funciones para manejar la Pila (registrar\_accion, ver\_historial) y la Cola (agregar\_solicitud\_revision, procesar\_revision).

**Funciones CRUD y Cálculos:** Implementación de las operaciones básicas de gestión de cursos (Crear, Leer, Actualizar, Eliminar) y cálculos (promedio, conteo).

**Funciones de Algoritmos:** Implementación separada de los algoritmos de ordenamiento (Burbuja, Inserción) y búsqueda (Lineal, Binaria).

**Función Principal (main):** Bucle infinito que gestiona la presentación del menú, la captura de la opción del usuario y la ejecución de la función correspondiente.

## Explicación del Uso de Estructuras de Datos

estructura	Variable	Tipo de Uso	Justificación
Escuchando/Array	cursos	Almacenamiento Principal (CRUD)	Colección dinámica de diccionarios, ideal para almacenar registros heterogéneos (nombre <i>string</i> , nota <i>float</i> ) y permitir acceso por índice para las operaciones de actualización y eliminación.
Pila (LIFO)	historial_acciones	Auditoría/Historial de Acciones	Se utiliza como una Pila (Last In, First Out). Es perfecta para un historial, ya que los usuarios generalmente quieren ver la acción <b>más reciente</b> primero, que es la última en haber sido agregada.
Cola (FIFO)	solicitudes_revisio n	Gestión de Tareas Pendientes	Se utiliza como una Cola (First In, First Out). Las solicitudes de revisión deben procesarse en el orden en que fueron recibidas, garantizando equidad y no dejando revisiones antiguas olvidadas.

## *Universidad Mariano Gálvez*

Al igual Se implementaron dos algoritmos de ordenamiento sencillos, de baja complejidad, para fines didácticos y para demostrar su funcionamiento directamente en la lista principal:

Algoritmo	Propósito	Justificación de Elección
<b>Burbuja</b> (ordenar_cursos_por_nota_burbuja)	Ordenar cursos por <b>nota</b> (Descendente) .	Es uno de los algoritmos más fáciles de entender e implementar. Su ineficiencia en grandes volúmenes es aceptable para esta demostración de alcance limitado.
<b>Inserción</b> (ordenar_cursos_por_nombre_insercion) )	Ordenar cursos por <b>nombre</b> (Alfabético).	Es eficiente con listas pequeñas o casi ordenadas. Se elige para ilustrar un enfoque diferente a Burbuja, donde se construye la lista ordenada elemento por elemento. Además, prepara la lista para la Búsqueda Binaria.

## *Universidad Mariano Gálvez*

### **Documentación Breve de Funciones**

---

Función	Descripción
registrar_accion(accion)	Agrega una cadena de texto a la Pila de historial.
ver_historial()	Muestra el contenido de la Pila de acciones, del más reciente al más antiguo (LIFO).
agregar_solicitud_revision()	Recibe datos del usuario y agrega una solicitud al final de la Cola (FIFO).
procesar_revision()	Remueve y muestra la primera solicitud pendiente de la Cola (FIFO).
registrar_curso()	(CRUD C) Solicita nombre y nota, validando que la nota sea numérica.
mostrar_cursos()	(CRUD R) Itera y muestra todos los cursos con su índice.
calcular_promedio_general()	Calcula la media aritmética de todas las notas.
contar_cursos_aprobados()	Cuenta y muestra los cursos $\geq 61.0$ (Aprobados) y los Reprobados.

## *Universidad Mariano Gálvez*

Función	Descripción
actualizar_nota()	(CRUD U) Permite cambiar la nota de un curso seleccionado por índice.
eliminar_curso()	(CRUD D) Elimina un curso seleccionado por índice.
ordenar_cursos_por_nota_burbuja()	Ordena la lista cursos por nota en forma descendente.
ordenar_cursos_por_nombre_insercion()	Ordena la lista de cursos por nombre alfabéticamente.
buscar_curso_lineal()	Busca un curso por nombre iterando la lista completa ( $\mathcal{O}(n)$ ).
buscar_curso_binaria()	Busca un curso por nombre usando el método de dividir a la mitad ( $\mathcal{O}(\log n)$ ), requiriendo que esté ordenado.
main()	Función de control principal que maneja el menú y la limpieza de pantalla.

*Universidad Mariano Gálvez*

**Diagrama General del Sistema (Pseudocódigo Principal)**

---

INICIO main

DEFINIR cursos = []

DEFINIR historial\_acciones = []

DEFINIR solicitudes\_revision = []

DEFINIR NOTA\_APROBACION = 61.0

MIENTRAS VERDADERO:

  MOSTRAR MenuPrincipal

  LEER opcion

  SEGUN opcion:

    CASO "1": registrar\_curso()

    CASO "2": mostrar\_cursos()

    ... (Otros casos para opciones 3 a 12)

    CASO "13": SALIR

  DEFECTO: Imprimir "Opción no válida."

  ESPERAR tecla

  LIMPIAR pantalla

FIN MIENTRAS

FIN main