

UADE

Introducción a la algoritmia

Trabajo Práctico Integrador

Primer Cuatrimestre 2025

TP-Opción 1: Sistema de Gestión de Alumnos

Contexto del problema

Se desea desarrollar un sistema en Python que gestione las calificaciones de alumnos a partir de un archivo CSV y permita consultar, procesar y ordenar los datos.

Cada línea del archivo CSV tendrá el siguiente formato:

Nombre,Materia,Nota1,Nota2,Nota3,NotaFinal

Donde:

- **NotaFinal** es el promedio de las tres notas.
- El sistema debe calcular esta nota automáticamente al cargar datos manuales.

Requisitos del sistema

1. **Cargar datos desde un archivo CSV**, guardándolos en una **lista de listas** o **matriz**.
2. Permitir **agregar alumnos manualmente** por teclado, con cálculo automático de la nota final.
3. Generar **cinco informes**, por ejemplo:
 - Listado completo de alumnos con sus notas y promedio.
 - Promedio general por materia.
 - Alumnos con nota final mayor a un valor dado.
 - Alumnos con al menos una nota menor a 4.
 - Cantidad de aprobados y desaprobados por materia.
4. Implementar **al menos dos métodos de ordenamiento** sobre los datos:
 - Por nombre del alumno
 - Por nota final (de mayor a menor)

Organización del código

- Usar **funciones** para cada operación (carga de CSV, carga manual, informes, ordenamiento).
 - Validar los datos ingresados (ej., que las notas estén entre 0 y 10).
 - Comentar el código explicando cada bloque funcional.
-

TP-Opción 2: Tema libre con guía estructural

Desarrollar un **sistema de gestión** que permita:

1. **Cargar datos desde un archivo CSV** a una estructura de datos (lista de listas o matriz).
2. **Cargar datos manualmente**, desde el teclado, agregándolos a la misma estructura.
3. Generar **cinco informes estadísticos o resúmenes** utilizando estructuras condicionales, repetitivas y funciones propias.
4. Aplicar al menos **dos métodos de ordenamiento** (Selección, Burbuja o Inserción) sobre los datos cargados.
5. Mostrar los resultados por consola de forma clara.

Condiciones

- El sistema debe estar dividido en funciones.
- Debe incluir validaciones básicas (por ejemplo, evitar ingresar texto donde se espera un número).
- Debe presentarse un informe de lo implementado en formato PDF: descripción del sistema, estructura de datos utilizada, resumen de cada función y cómo se organizó el trabajo grupal.

Ideas de temas libres

- Sistema de gestión de películas, series o libros
- Registro de actividades deportivas o alimenticias
- Control de stock en una tienda o librería
- Agenda de contactos con múltiples campos
- Análisis de calificaciones o asistencia de estudiantes

Presentación

- El grupo debe presentar el proyecto oralmente (10 minutos).
- Cada integrante debe explicar una parte del código y responder una pregunta conceptual.

Rúbrica de calificación

Puntaje Total: 10 puntos

Criterio	Puntaje
Carga desde archivo CSV	1.0
Carga manual de datos	1.0
Implementación de 5 informes	2.0
Aplicación de dos métodos de ordenamiento	2.0
Organización del código en funciones	1.0
Presentación clara y documentación	1.0
Participación individual en la presentación	1.0

Bonificaciones

- **Bonus 1 (0.5 ptos):** uso de archivos JSON para guardar configuraciones o resultados.
- **Bonus 2 (0.5 ptos):** interfaz gráfica simple (puede ser con tkinter, PySimpleGUI, etc.).

Evaluación individual

Durante la presentación, cada estudiante deberá explicar *al menos una función del código* y responder una pregunta conceptual relacionada con estructuras de datos o control. Así se justifica la **nota individual complementaria**.