

Fecha Límite de entrega: domingo 25 de enero, 2026, 23:59:00.

La secretaría del IES Pío Baroja nos ha entregado un archivo llamado `notasAlumnos.csv` que contiene las calificaciones de DAW de varios módulos profesionales. Lamentablemente, los datos han sido introducidos manualmente y contienen múltiples errores de formato que impiden su procesamiento automático.

Errores que pueden aparecer: espacios en blanco, nombres no normalizados ("Alejandro", "alejandro" y "ALEJANDRO" deben ser tratados como la misma persona), formatos numéricos. Si una nota contiene texto (ej. "pendiente" o "no presentado"), el programa no debe detenerse; simplemente ignorará esa entrada y continuará con la siguiente. Ignoraremos también aquellas notas que no tengan un formato decimal válido "." (ej. "4,5" en lugar de "4.5").

Además del archivo de notas, el programa debe leer un segundo archivo llamado `becas.txt`. Debes cruzar la información de ambos archivos. En el informe final y en la gráfica, se debe indicar claramente si el alumno cuenta con beca o no.

Modificación de los archivos proporcionados

1. Incluye tu propio nombre en el archivo `notasAlumnos.csv` con notas en 3 módulos
2. Incluye tu propio nombre en el archivo `becas.txt` indicando si tienes beca o no.

Debes desarrollar un programa en Python que limpie, organice y visualice esta información aplicando tipos avanzados de datos.

No se permite el uso de librerías externas de gestión de datos como Pandas.

Debes utilizar un Diccionario principal donde la clave sea el nombre del alumno.

El valor de cada clave debe ser una Lista que almacene todas las notas numéricas encontradas para ese alumno.

El informe por consola ahora tendrá 4 columnas: Nombre, Promedio, Estado y Beca.

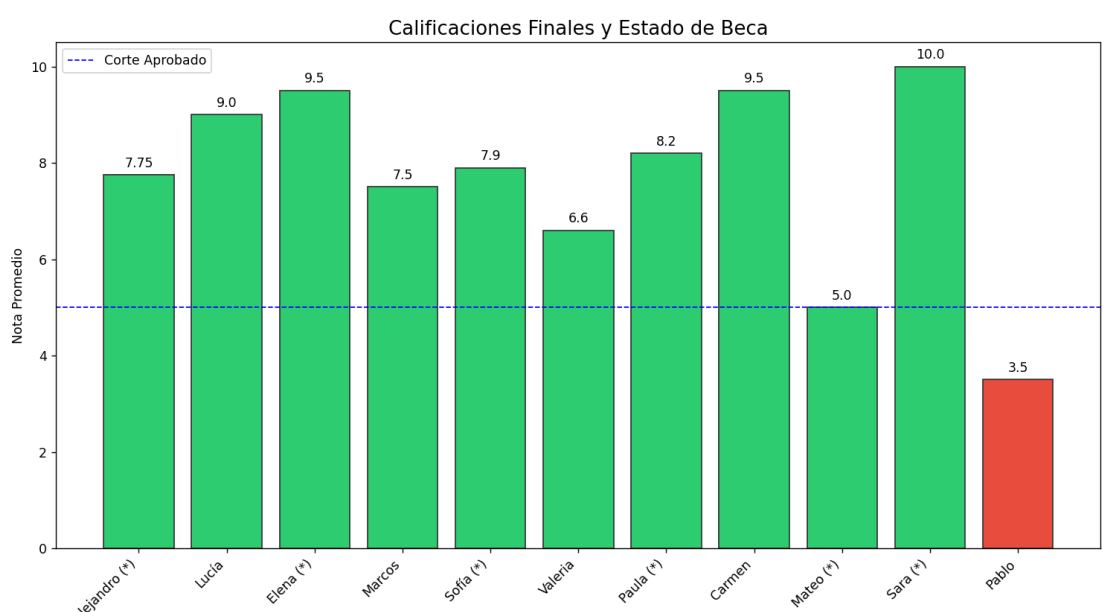
Salida por consola:

ALUMNO	NOTA	ESTADO	BECA
Alejandro	7.75	APROBADO	SI
Lucía	9.00	APROBADO	NO
Elena	9.50	APROBADO	SI
Marcos	7.50	APROBADO	NO
Sofía	7.90	APROBADO	SI
Valeria	6.60	APROBADO	NO
Paula	8.20	APROBADO	SI
Carmen	9.50	APROBADO	NO
Mateo	5.00	APROBADO	SI
Sara	10.00	APROBADO	SI
Pablo	3.50	SUSPENSO	NO

Visualización de Datos

Utiliza la librería matplotlib para generar una gráfica de barras:

- **Eje X:** Nombres de los alumnos.
- **Eje Y:** Nota media.
- **Lógica de color:** Las barras de los alumnos aprobados deben ser de color **verde** y las de los suspensos de color **rojo**.
- Debe incluir una línea horizontal discontinua en el valor 5.0 para marcar el límite de aprobado.



Instrucciones para la entrega:

- Debes crear un repositorio en GitHub con tu usuario, que será el enlace que entregarás para la corrección. Cualquier commit que se realice pasada la fecha de entrega, la anulará por completo y la nota será 0.
- Debes crear un entorno virtual (virtualenv) con venv para la instalación de las librerías incluidas en el fichero requirements.txt que se facilita. *Las instrucciones de creación se explican más abajo.
- Debes crear un fichero README.md donde se reflejen los comandos para crear y activar el virtualenv así como para la instalación de las librerías. Se valorará que el README.md incluya también una breve explicación del código de la práctica.
- El flujo principal del código debe estar incluido en un fichero denominado "main.py". Se valorará el uso de funciones en este archivo para una mayor legibilidad.

Ejemplo de jerarquía de ficheros que debes generar en tu repositorio:

```

diccionarios
├── .gitignore
├── README.md
├── ej30becas.txt
├── ej30notasdaw.csv
├── main.py
├── notes_graph.png
└── requirements.txt

```

***Instrucciones para la creación del entorno virtual:**

Un entorno virtual (venv) es un directorio aislado que contiene una instalación de Python y unas dependencias específicas para un proyecto. Esto evita conflictos entre las librerías de diferentes proyectos.

Una vez dentro del directorio del proyecto, crea el entorno virtual:

```
python -m venv practica_diccionarios
```

La activación varía según el sistema operativo y la terminal que uses:

- En macOS y Linux (Bash/Zsh):

```
source practica_diccionarios/bin/activate
```

- En Windows (Command Prompt - CMD):

```
practica_diccionarios\Scripts\activate.bat
```

- En Windows (PowerShell, terminal por defecto en VS Code o PyCharm):

```
.\practica_diccionarios\Scripts\Activate.ps1
```

Si al ejecutar este comando recibes un error sobre que la ejecución de scripts está deshabilitada, abre una nueva terminal de PowerShell y ejecuta lo siguiente (solo necesitas hacerlo una vez en tu equipo):

```
> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

```
> Confirma la acción con `S` o `Y` y vuelve a intentar la activación.
```

Con el entorno activado, instala las librerías necesarias.

```
python -m pip install -r requirements.txt
```