

Gestión de tareas

Desarrollar un programa en Python que permita al usuario gestionar una lista de tareas. Cada tarea estará representada por una tupla que contenga el nombre de la tarea y su prioridad. El programa deberá ser capaz de leer estas tareas desde un archivo al iniciar y guardar los cambios en el archivo antes de finalizar. Es crucial implementar el manejo de excepciones para controlar errores durante la apertura y manipulación del archivo.

Requisitos:

1. **Cargar Tareas:** Al iniciar, el programa intentará cargar una lista de tareas desde un archivo llamado `tareas.txt`. El formato de cada tarea en el archivo será `nombre_de_la_tarea,prioridad`. Por ejemplo:
`Comprar leche,Alta`.
2. **Manejo de Excepciones:** Si el archivo `tareas.txt` no existe o hay un problema al abrirlo, el programa deberá capturar la excepción y mostrar un mensaje de error amigable al usuario, por ejemplo, "Error al cargar el archivo de tareas. Se iniciará con una lista de tareas vacía." Luego, el programa continuará su ejecución normalmente con una lista de tareas vacía.
3. **Agregar Tareas:** El programa permitirá al usuario añadir nuevas tareas a la lista. Cada tarea se solicitará al usuario a través de la consola, pidiéndole que ingrese el nombre de la tarea y su prioridad, que luego se almacenarán como una tupla en la lista de tareas.
4. **Guardar Tareas:** Antes de finalizar, el programa deberá guardar la lista actualizada de tareas en el archivo `tareas.txt`, sobrescribiendo el contenido anterior. Cada tarea se escribirá en una nueva línea del archivo, manteniendo el formato `nombre_de_la_tarea,prioridad`.
5. **Visualizar Tareas:** El programa ofrecerá una opción para que el usuario pueda ver todas las tareas cargadas actualmente en la lista, mostrándolas en la consola.

Ejemplo fichero tareas:

Lavar la ropa,Media
Comprar leche,Alta
Enviar email al profesor,Baja
Pagar el alquiler,Alta
Estudiar para el examen,Alta
Cortar el césped,Baja
Programar la reunión,Media

Ayudas:

1. Lectura de Ficheros

Para leer el contenido de un fichero en Python y cargarlo en una lista de tuplas:

```
tareas = []
try:
    with open('tareas.txt', 'r') as archivo:
        for linea in archivo:
            partes = linea.strip().split(',')
            if len(partes) == 2: # Asegurarse de que la línea tiene 2 partes
                tareas.append((partes[0], partes[1]))
except FileNotFoundError:
    print("Archivo no encontrado. Se iniciará con una lista de tareas vacía.")
```

2. Escritura de Ficheros

Para guardar una lista de tuplas en un fichero, sobrescribiendo su contenido anterior:

```
with open('tareas.txt', 'w') as archivo:
    for tarea in tareas:
        archivo.write(','.join(tarea) + '\n')
```

3. Añadir Tareas

Para añadir una nueva tarea a la lista, solicitando al usuario el nombre de la tarea y su prioridad:

```
nombre = input("Ingrese el nombre de la tarea: ")
prioridad = input("Ingrese la prioridad de la tarea (Alta/Media/Baja): ")
tareas.append((nombre, prioridad))
```

4. Manejo de Excepciones

Es importante manejar excepciones para evitar que el programa falle inesperadamente. El manejo de la excepción **FileNotFoundError** al intentar abrir un archivo que no existe es un buen ejemplo. Otro ejemplo podría ser el manejo de errores de entrada/salida genéricos:

```
try:
    # Intento de abrir un archivo, leerlo o escribirlo
except FileNotFoundError:
    # Código para manejar un archivo que no se encuentra
except IOError:
    # Código para manejar errores de E/S generales
```

5. Visualizar Tareas

Para mostrar todas las tareas cargadas actualmente en la lista:

```
for tarea in tareas:
    print(f"Nombre: {tarea[0]}, Prioridad: {tarea[1]}")
```