

Desafio Tkinter

Desafio 1: Gestor de notas con Tkinter y almacenamiento en archivo CSV

Descripción:

Crear una interfaz gráfica con tkinter que permita al usuario ingresar su nombre y una nota en un campo de texto, y luego guardar estos datos en un archivo CSV.

El código debe:

- Verificar que los campos no estén vacíos.
- Validar que la nota sea un número.
- Almacenar los datos en un archivo CSV llamado 'notas.csv'.
- Mostrar un mensaje de éxito o error según sea necesario.
- Si el archivo CSV no existe, debe inicializarlo con encabezados.

Requisitos:

- Utilizar tkinter para la interfaz gráfica.
- Utilizar el módulo csv para manejar el archivo de notas.

```
"""
Desafio 1: Gestor de notas con Tkinter y almacenamiento en archivo CSV

Descripción:
Crear una interfaz gráfica con tkinter que permita al usuario ingresar su nombre y una nota en un campo de texto,
y luego guardar estos datos en un archivo CSV.

El código debe:
- Verificar que los campos no estén vacíos.
- Validar que la nota sea un número.
- Almacenar los datos en un archivo CSV llamado 'notas.csv'.
- Mostrar un mensaje de éxito o error según sea necesario.
- Si el archivo CSV no existe, debe inicializarlo con encabezados.

Requisitos:
- Utilizar tkinter para la interfaz gráfica.
- Utilizar el módulo csv para manejar el archivo de notas.
"""

import tkinter as tk
import csv
import tkinter.messagebox
import os

def save_data():
    name = entry_name.get()
    grade = entry_grade.get()

    if name == "" or grade == "":
        tkinter.messagebox.showerror("Error",
        "Los campos no pueden estar vacíos.")
        return

    try:
        grade = int(grade)
    except ValueError:
        tkinter.messagebox.showerror("Error",
        "La nota debe ser un número.")
        return

    with open('notas.csv', 'a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([name, grade])

    entry_name.delete(0, tk.END)
    entry_grade.delete(0, tk.END)

    tkinter.messagebox.showinfo("Éxito",
    "Datos guardados correctamente.")

root = tk.Tk()
root.title("Gestor de Notas")

label_name = tk.Label(root, text="Nombre:")
label_name.pack()

entry_name = tk.Entry(root)
entry_name.pack()

label_grade = tk.Label(root, text="Nota:")
label_grade.pack()


entry_grade = tk.Entry(root)
entry_grade.pack()

save_button = tk.Button(root, text="Guardar", command=save_data)
save_button.pack(pady=10)

root.geometry("300x200")

if not os.path.exists('notas.csv'):
    with open('notas.csv', 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(["Nombre"] + ["Nota"])

root.mainloop()
```

 Gestor de Notas — □ ×

Nombre:

Nota:

Desafio Openpyxl

```
'''
Desafio 2: Generación y modificación de un archivo Excel con openpyxl

Descripción:
Crear un script que cargue un archivo Excel existente,
agregue algunos datos de forma programática,
y luego guarde el archivo modificado.

El código debe:
- Comprobar si el archivo Excel existe.
- Si el archivo no existe, debe crear uno nuevo y agregar un encabezado.
- Si el archivo ya existe, debe agregar una nueva fila con datos.
- Ajustar el tamaño de las columnas de acuerdo con el contenido.
- Mostrar mensajes de error adecuados si hay problemas al guardar el archivo o si el archivo tiene una extensión incorrecta.

Requisitos:
- Utilizar openpyxl para manejar el archivo Excel.
- Utilizar os para verificar la existencia del archivo.

'''

import openpyxl
import os
from openpyxl.utils import get_column_letter

def add_notas(nomb, nota, mate, sheet):
    sheet.append([nomb, nota, mate])

def update_excel(file_path):
    if not os.path.exists(file_path):
        print("El archivo no existe.")
        wb = openpyxl.Workbook()
        sheet = wb.active
        sheet.append(["Nombre", "Nota", "Asignatura"])
    else:
        wb = openpyxl.load_workbook(file_path)
        sheet = wb.active

        new_row = ["Juan", "80", "Matemáticas"]
        sheet.append(new_row)

    print(
        "Los datos fueron agregados correctamente al archivo Excel."
    )

    mas_notas = []
    while True:
        print("\nIngrese El nombre del estudiante, con su nota y materia de la misma")
        nomb = input()
        "Ingrese el Nombre del Estudiante: "
        nota = input("Ingrese su nota: ")
        mate = input("Ingrese la materia: ")

        mas_notas.append([nomb, nota, mate])

        add_notas(nomb, nota, mate, sheet)

        continuar = input(
            "¿Deseas agregar otra nota? (si/no): ").lower()
        if continuar != 'si':
            break

    try:
        wb.save(file_path)
        print("Se han guardado las notas")
    except PermissionError:
        print(
            "Error: No se puede guardar el archivo. Asegúrate de que esté cerrado."
        )

    for col in sheet.columns:
        max_length = 0
        column = col[0].column_letter
        for cell in col:
            try:
                if len(str(cell.value)) > max_length:
                    max_length = len(cell.value)
            except:
                pass
        adjusted_width = (max_length + 2)
        sheet.column_dimensions[column].width = adjusted_width
    wb.close()
    if not file_path.endswith('.xlsx'):
        print(
            "Error: El archivo debe tener la extensión '.xlsx'."
        )
        return

file_path = "notas.xlsx"
update_excel(file_path)
```

```
/migue/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/mi  
gue/OneDrive/Desktop/Examen final Algoritmos/excel_challenge.py"  
Los datos fueron agregados correctamente al archivo Excel.
```

```
Ingrese El nombre del estudiante, con su nota y matria de la misma
```

```
Ingrese el Nombre del Estudiante: Juan
```

```
Ingrese su nota: 85
```

```
Ingrese la materia: Matematica
```

```
¿Deseas agregar otra nota? (si/no): no
```

```
Se han guardado las notas
```

```
PS C:\Users\migue\OneDrive\Desktop\Examen final Algoritmos> |
```

>