



Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Licenciatura en Ciencias Computacionales

Programación Básica (073)

Diego González Flores – 2086068

Adrián Morales Vázquez - 2149220

Docente: Perla Marlene Viera Gonzalez

E4. Gráficas y Excel

17/5/25

Para la función `convertirMonedas()`, se agregó una manera de almacenar los datos de conversión a un archivo Excel. Se creará un archivo Excel llamado conversiones (y si ya existe, agrega más datos sin sobrescribir. Las celdas son Fecha (fecha y hora de la consulta), Moneda Fuente, Moneda Destino, Cantidad, Tasa (la cantidad de conversión entre las dos monedas escritas) y Resultado.

```
def convertirMonedas():
    try:
        fuente = input("Ingrese el código de moneda fuente: ").upper()
        blanco = input("Ingrese el código de moneda de destino: ").upper()
        cant = float(input("Ingrese la cantidad a convertir: "))
        url = f"https://api.currencyfreaks.com/v2.0/rates/latest?apikey={apikey}&symbols={blanco}&base={fuente}"
        respuesta = requests.get(url)
        if respuesta.status_code == 200:
            datos = respuesta.json()
            rate = float(datos['rates'][blanco])
            convertido = cant * rate
            if convertido:
                print(f'{cant} {fuente} es igual a {convertido} {blanco}')
                fecha_hora = datetime.now().isoformat()
                with open("conversion.txt", "a") as archivo:
                    archivo.write(json.dumps({
                        "fecha": fecha_hora,
                        "fuente": fuente,
                        "destino": blanco,
                        "tasa": rate
                    }) + "\n")
                excel = "conversiones.xlsx"
                if os.path.exists(excel):
                    libro = load_workbook(excel)
                    hoja = libro.active
                else:
                    libro = Workbook()
                    hoja = libro.active
                    hoja.append(["Fecha", "Moneda Fuente", "Moneda Destino", "Cantidad", "Tasa de Cambio", "Convertido"])
                hoja.append([fecha_hora, fuente, blanco, cant, rate, convertido])
                libro.save(excel)
            else:
                print("Error: No se encontraron datos de conversión en la respuesta del API")
        else:
            print(f"Error: Código de estado: {respuesta.status_code}")
    except ValueError as e:
        print("Error: ", e)
    print("\n")
```

19/5/25

```
def graficarComparacionMonedas():
    monedas = ['USD', 'EUR', 'MXN', 'JPY', 'ARS']
    colores = ['orange', 'red', 'darkorange', 'magenta', 'skyblue']
    valores = {}

    url = f"https://api.currencyfreaks.com/v2.0/rates/latest?apikey={apikey}"
    respuesta = requests.get(url)

    if respuesta.status_code != 200:
        print(f"Error al obtener datos. Código HTTP: {respuesta.status_code}")
        return
    datos = respuesta.json()
    rates = datos.get('rates', {})
```

```

try:
    mxn_rate = float(rates['MXN'])
except KeyError:
    print("No se encontró la tasa de MXN.")
    return
for moneda in monedas:
    try:
        if moneda == 'MXN':
            valores[moneda] = 1.0
        else:
            tasa = float(rates[moneda])
            valores[moneda] = mxn_rate / tasa
    except KeyError:
        print(f"No se encontró la tasa para {moneda}")
    except ValueError:
        print(f"Tasa inválida para {moneda}")
if len(valores) < 2:
    print("No hay suficientes tasas válidas para graficar.")
    return
print("\nOpciones de visualización:")
print("1. Gráfica de barras")
print("2. Diagrama de dispersión")
print("3. Gráfica de pastel")
print("4. Gráfica de líneas")
opcion = input("Elige el tipo de gráfico (1–4): ").strip()
plt.figure(figsize=(12, 6))
if opcion == '1':
    plt.bar(valores.keys(), valores.values(), color=colores)
    plt.title("Valor de monedas extranjeras en MXN (barra)")
    plt.ylabel("Valor en pesos mexicanos (MXN)")
elif opcion == '2':
    cantidades = list(range(1, 101))
    for i, moneda in enumerate(monedas):
        if moneda in valores:
            conversiones = [cantidad * valores[moneda] for cantidad in cantidades]
            plt.scatter(cantidades, conversiones, label=moneda, color=colores[i], s=10)
    plt.title("Diagrama de dispersión: Valor de monedas en MXN")
    plt.xlabel("Cantidad de moneda extranjera")
    plt.ylabel("Valor en MXN")
elif opcion == '3':
    plt.pie(valores.values(), labels=valores.keys(), colors=colores, autopct='%1.1f%%')
    plt.title("Proporción relativa de monedas en MXN")
elif opcion == '4':
    cantidades = list(range(1, 101))
    for i, moneda in enumerate(monedas):
        if moneda in valores:
            conversiones = [cantidad * valores[moneda] for cantidad in cantidades]
            plt.plot(cantidades, conversiones, label=moneda, color=colores[i])
    plt.title("Comparación del valor de monedas extranjeras en MXN")
    plt.xlabel("Cantidad de moneda extranjera")
    plt.ylabel("Valor en pesos mexicanos")
    plt.grid(True)
else:
    print("Error: Opción inválida.")
plt.tight_layout()
plt.show()
def salir():
    print("Adios!")
    return False

```

Esto obtiene los valores de varias monedas (USD, EUR, MXN, JPY, ARS) respecto al peso mexicano, y permite elegir cómo visualizar la comparación: barras, líneas, dispersión o pastel