UNIVERSIDAD SAN SEBASTIAN
VOCACIÓN POR LA EXCELENCIA

ADVANCE USS

GUÍA DEMOSTRATIVA
Ejemplo de uso de una API REST para consultar indicadores económicos con Python

TALLER DE PROGRAMACIÓN II

Guía demostrativa. Ejemplo de uso de una API REST para consultar indicadores económicos con Python

Indicaciones:

La siguiente actividad tiene como propósito el que puedas ejercitar lo aprendido durante esta semana relacionado con el API REST en Python.

El trabajo es de carácter *individual* y *formativo*.

Ejercicio:

1. Objetivo.

- **Este código utiliza la API de mindicador.cl para consultar indicadores económicos como la UF, el dólar, el euro, etc., proporcionando datos históricos en formato JSON.**
- **Código Python propuesto:**

```
import json
import requests
import pandas as pd
import matplotlib.pyplot as plt


# Solicitar al usuario el tipo de indicador
ind = input("Ingrese el indicador (e.g., uf, dolar, euro): ")
# valores posibles: [uf, ivp, dolar, dolar_intercambio, euro, ipc,
utm, imacec, tpm, libra_cobre, tasa_desempleo, bitcoin]


def InfoApi(ind, fech='2024'):
    # URL de la API con el indicador y año
    url = f'https://mindicador.cl/api/{ind}/{fech}'


    # Realizar la solicitud GET a la API
    response = requests.get(url)


    # Convertir la respuesta a un diccionario Python
    data = response.json()
```

```
    # Verificar si el indicador tiene datos
    if "serie" not in data:
        print("No se encontraron datos para el indicador o la fecha proporcionada.")
        return None

    # Extraer los datos de la serie
    serie = data["serie"]

    # Ordenar los datos por fecha (de más reciente a más antigua)
    serie_ordenada = sorted(serie, key=lambda x: x["fecha"], reverse=True)

    # Crear un DataFrame a partir de la serie de datos
    df = pd.DataFrame(serie_ordenada)

    # Convertir la columna de fecha a tipo datetime
    df["fecha"] = pd.to_datetime(df["fecha"])

    # Definir la columna 'fecha' como índice del DataFrame
    df.set_index("fecha", inplace=True)

    # Mostrar el DataFrame
    print("\nDatos del indicador (primeros 5 registros):")
    print(df.head())

    # Calcular estadísticas básicas
    promedio = df["valor"].mean()
    maximo = df["valor"].max()
    minimo = df["valor"].min()
    varianza = df["valor"].var()
```

```
    desviacion_estandar = df["valor"].std()


    print(f"\nEstadísticas del indicador {ind}:")
    print(f"Promedio: {round(promedio,2)}")
    print(f"Valor máximo: {round(maximo,2)}")
    print(f"Valor mínimo: {round(minimo,2)}")
    print(f"Varianza: {round(varianza,2)}")
    print(f"Desviación estándar: {round(desviacion_estandar,2)}")


    # Graficar la evolución del indicador a lo largo del tiempo
    plt.figure(figsize=(10, 6))
    plt.plot(df.index, df["valor"], marker='o', linestyle='-',
color='b')
    plt.title(f"Evolución del Indicador {ind} durante {fech}")
    plt.xlabel("Fecha")
    plt.ylabel("Valor")
    plt.grid(True)
    plt.xticks(rotation=45)
    plt.tight_layout()


    # Mostrar el gráfico
    plt.show()


    return df


# Llamar a la función y obtener los datos
resultado = InfoApi(ind)
```

2. Objetivo del código.

El objetivo del código es:

1. **Consultar indicadores económicos específicos desde una API REST.**
2. **Procesar los datos obtenidos en un *dataframe* de Pandas.**

3. **Calcular estadísticas descriptivas y visualizarlas en la consola.**
4. **Graficar la evolución del indicador a lo largo del tiempo.**

3. *API Key* y por qué este código no la requiere.

Algunas API requieren una clave de autenticación (*API Key*) para controlar el acceso. Sin embargo, mindicador.cl es una API abierta y no requiere una clave para realizar solicitudes, lo que simplifica su uso.

Mindicador.cl es un servicio *open source* (*web service*) que entrega los principales indicadores económicos para Chile en formato JSON, tanto indicadores diarios como históricos para que desarrolladores puedan utilizarlos en sus aplicaciones o sitios web.

La aplicación mapea constantemente el sitio del Banco Central de Chile, manteniendo así nuestra base de datos actualizada con los últimos valores del día.

La referencia completa se puede encontrar en https://mindicador.cl/

4. Análisis del funcionamiento del código

Código proporcionado:

```
import json

import requests

import pandas as pd

import matplotlib.pyplot as plt
```

1. **Importación de bibliotecas:**

- **Json: para procesar datos en formato JSON.**
- **Requests: para realizar solicitudes HTTP.**
- **Pandas: para procesar y analizar datos en forma de tablas.**
- **Matplotlib.pyplot: para graficar la evolución del indicador.**

Función InfoApi(ind, fech):

La función recibe dos parámetros:

- **ind: el indicador económico a consultar (ejemplo: 'uf', 'dolar').**
- **fech: el año de los datos a consultar (por defecto '2024').**

Flujo de ejecución de la función:

1. **Construcción de la URL:**

- **Se forma una URL dinámica basada en el indicador y el año:**

```
url = f'https://mindicador.cl/api/{ind}/{fech}'
```

2. **Solicitud GET a la API:**

- **response = requests.get(url): realiza la solicitud a la API.**
- **data = response.json(): convierte la respuesta JSON en un diccionario de Python.**

3. **Validación de datos:**

- **Si el indicador no tiene datos o es incorrecto:**

```
if "serie" not in data:
    print("No se encontraron datos para el indicador o la fecha proporcionada.")
    return None
```

4. **Creación del *dataframe*:**

- **Los datos se organizan en un *dataframe*:**

```
serie = data["serie"]
df = pd.DataFrame(serie)
```

- **La columna 'fecha' se convierte al formato *datetime* para facilitar el análisis:**

```
df["fecha"] = pd.to_datetime(df["fecha"])
df.set_index("fecha", inplace=True)
```

5. **Cálculo de estadísticas:**

- **Se calculan estadísticas descriptivas del indicador, como promedio, máximo, mínimo, varianza y desviación estándar.**

6. **Gráfico de evolución del indicador:**

- **Se genera un gráfico de líneas que muestra la evolución del indicador durante el año.**

5. Ejemplo de ejecución del código

Entrada:

El usuario ingresa el indicador económico:

- **Ingrese el indicador (e.g., uf, dolar, euro): dolar.**

Salida en consola:

```
Datos del indicador (primeros 5 registros):

fecha        valor
2024-01-05   800.5
2024-01-04   798.2
2024-01-03   797.9
2024-01-02   798.0
2024-01-01   799.0

Estadísticas del indicador dolar:
Promedio: 798.32
Valor máximo: 800.5
Valor mínimo: 797.9
Varianza: 0.62
Desviación estándar: 0.79
```

Gráfico de salida:

- **Evolución del indicador dólar durante 2024: muestra cómo ha variado el valor del dólar a lo largo del tiempo.**

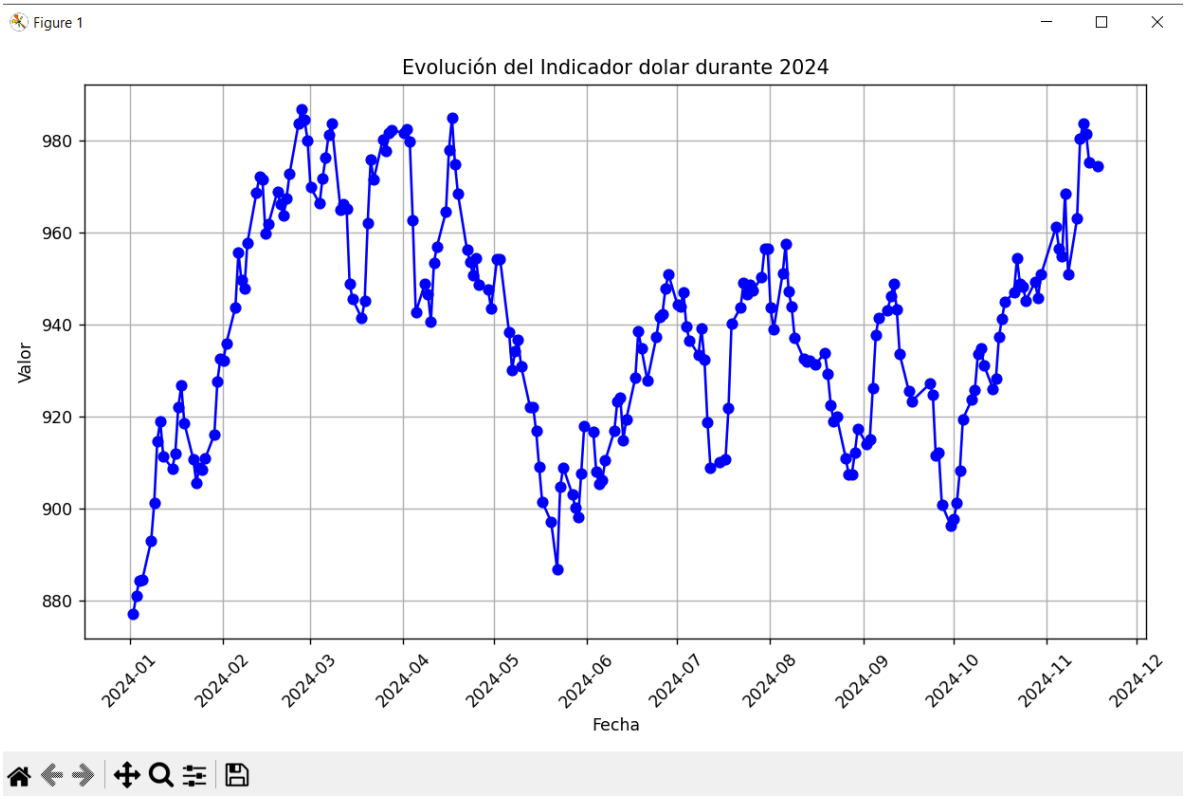

Figura 1: Evolución del indicador del dólar durante 2024

Fuente: Convera.com (s.f.)

6. Aplicaciones del código.

1. **Análisis de indicadores económicos:**
   - **Consultar y analizar indicadores como la UF, IPC, dólar, entre otros.**
2. **Visualización de tendencias:**
   - **Generar gráficos para comprender mejor la evolución de los datos económicos.**
3. **Ampliación del código:**
   - **Adaptar el código para trabajar con múltiples indicadores simultáneamente.**
   - **Exportar las estadísticas a un archivo CSV para su posterior análisis.**

7. Actividades sugeridas.

1. **Consultar indicadores diferentes:**
   - **Probar con indicadores como 'uf', 'ipc', 'bitcoin' y analizar sus estadísticas.**
2. **Modificar el período de consulta:**
   - **Cambiar el año en el parámetro fech para obtener datos históricos.**
3. **Personalizar el gráfico:**
   - **Cambiar colores, estilos de línea o agregar más información al gráfico.**
4. **Guardar los datos:**
   - **Adaptar el código para guardar el *dataframe* en un archivo CSV:**

```
df.to_csv(f'{ind}_{fech}.csv')
```