Planteamiento del Problema (API elegida: REST COUNTRIES)

Contexto

La API REST Countries proporciona acceso a datos geográficos, demográficos y económicos de todos los países del mundo. Sin embargo, el uso directo de esta API presenta desafíos para usuarios que necesitan procesar, analizar o visualizar la información de manera eficiente. Los datos crudos suelen estar en formato JSON, lo que requiere transformación para extraer valor útil. Además, manipular grandes volúmenes de datos manualmente es inviable, especialmente para aplicaciones científicas, educativas o empresariales que exigen automatización y análisis estructurado.

Problema

¿Cómo integrar, procesar y analizar datos de países desde la API REST Countries de manera modular, reutilizable y escalable, para facilitar la toma de decisiones basada en información geográfica y demográfica?

Desafíos Clave:

1. Acceso y Conexión con la API:
    - La API devuelve datos en formato JSON, pero no está estructurada para análisis directo.
    - Requiere manejo de errores (ej.: conexión fallida, límites de tasa).
2. Transformación de Datos:
    - Los datos crudos incluyen campos anidados (ej.: idiomas, monedas) que deben normalizarse para análisis.
    - Es necesario calcular métricas derivadas (ej.: densidad poblacional).
3. Filtrado y Búsqueda Personalizada:
    - La API permite filtrar por nombre, región o moneda, pero no ofrece búsquedas avanzadas mediante expresiones regulares.
4. Análisis Estadístico y Visualización:
    - Falta herramientas integradas para calcular estadísticas (media, mediana) o generar gráficos.
5. Persistencia y Exportación:

- No hay funcionalidad nativa para guardar datos en formatos como Excel o JSON.

Objetivos

1. Automatizar la conexión a la API REST Countries para descargar y estructurar datos en listas/diccionarios.
2. Procesar y limpiar datos mediante expresiones regulares para filtrar información relevante (ej.: países que comienzan con una letra).
3. Realizar análisis estadístico básico (media, mediana, varianza) sobre campos numéricos (población, área).
4. Visualizar datos con gráficos (barras, líneas) usando matplotlib.
5. Exportar datos a Excel para compartir resultados en entornos colaborativos.
6. Interpretar resultados para relacionar estadísticas con contexto geográfico/demográfico.

Alcance

La solución desarrollada abarca las siguientes etapas:

1. Conexión y Descarga de Datos:
    - Usar requests para acceder a endpoints como /v3.1/all o /v3.1/name/{nombre}.
    - Manejar errores HTTP y límites de tasa.
2. Estructuración de Datos:
    - Transformar JSON en listas de diccionarios con campos clave (nombre, población, región, idiomas, monedas).
    - Calcular métricas derivadas (ej.: densidad poblacional).
3. Filtrado con Expresiones Regulares:
    - Filtrar países por patrones en sus nombres (ej.: "^A" para países que empiezan con "A").
4. Análisis Estadístico:
    - Calcular media, mediana, moda, varianza y desviación estándar para campos numéricos.
    - Relacionar resultados con contextos geográficos (ej.: país con mayor/menor población).

5. Visualización Gráfica:
    - Generar gráficos de barras (ej.: top 10 países por población) y líneas (ej.: densidad de países filtrados).
6. Exportación y Persistencia:
    - Guardar datos en JSON y Excel (usando pandas).
7. Interpretación de Resultados:
    - Relacionar estadísticas con realidades geográficas (ej.: alta dispersión en población debido a países como China o India).

Justificación

1. Automatización de Procesos:
    - Elimina la necesidad de extracción manual de datos, reduciendo tiempo y errores humanos.
2. Integración de Herramientas Modernas:
    - Combina bibliotecas como requests, pandas, matplotlib y statistics, siguiendo buenas prácticas de programación modular y reutilizable.
3. Aplicabilidad Práctica:
    - Educación : Facilita el aprendizaje de programación básica (condicionales, ciclos) y avanzada (APIs, análisis de datos).
    - Investigación Científica : Permite estudiar patrones demográficos, económicos o geográficos.
    - Empresas : Apoyar decisiones basadas en datos (ej.: planificación urbana, políticas sociales).
4. Escalabilidad y Modularidad:
    - El diseño en módulos separados (modulo.py y script.py) permite agregar nuevas funcionalidades sin alterar código existente.

Conexión con los Apuntes de Programación Básica en Python

1. Fase I (Fundamentos):
    - Uso de variables, ciclos (for, while) y condicionales (if-else) para procesar datos.
    - Ejemplo:

```
for pais in datos_estructurados[:5]:
```

```
print(f"- {pais['Nombre']}: {pais['Población']} habitantes")
```

2. Fase II (Funciones y Algoritmos):
    - Modularización de funciones (obtener_datos_paises, analizar_estadisticas) para reutilizar código.
    - Implementación de algoritmos de ordenamiento implícito (ej.: sorted(..., key=...) para top 10 países).
3. Fase III (Aplicación en Ciencia y Tecnología):
    - Integración con APIs, manejo de JSON, análisis estadístico y visualización gráfica, como se menciona en los apuntes.

Ejemplo de Aplicación Práctica

Caso de Estudio: Análisis de Población Mundial

- Problema Real:
  ¿Cuáles son los 10 países más poblados y cómo se distribuye su densidad poblacional?
- Solución Implementada:
    1. Descargar datos con obtener_datos_paises().
    2. Filtrar y estructurar con estructurar_datos_paises().
    3. Calcular estadísticas con analizar_estadisticas().
    4. Graficar con graficar_datos().
    5. Interpretar resultados con interpretar_resultados().
- Resultado:
    - Gráfico de barras con los 10 países más poblados (China, India, EE.UU.).
    - Análisis estadístico que muestra alta dispersión en población debido a países extremos.

Conclusión

La solución propuesta resuelve el problema de acceso, procesamiento y análisis de datos geográficos mediante un enfoque modular, escalable y basado en buenas prácticas de programación. Alineada con los apuntes de programación básica, combina fundamentos teóricos (ciclos, funciones) con aplicaciones avanzadas (APIs, visualización), demostrando cómo Python puede transformar datos complejos en información útil para múltiples disciplinas.