==========

SUBMITTED CODE

```
"""Displays information about IP addresses connected to the computer."""

import argparse
import subprocess
from datetime import datetime
import requests

API_KEY =
'd2355d03d9feff0470a4dc72e739ad028c5c9b309a4abbf9e1f0cda4c8f8d42f1d66fda25f83bc0a'
API_URL = 'https://api.abuseipdb.com/api/v2/check'
LOG_FILE = 'log_ips.txt'



def registrar_log(texto):
    """Escribe una línea de texto en el archivo de log."""
    with open(LOG_FILE, 'a', encoding='utf-8') as f:
        f.write(texto + '\n')



def obtener_ips_activas():
    """
    Ejecuta un script de PowerShell externo para obtener IPs activas.

    Returns:
        list: Lista de IPs encontradas por el script.
    """
    try:
        print("Ejecutando script de PowerShell para obtener IPs activas...")
```

```
        resultado = subprocess.run(
            ["powershell", "-ExecutionPolicy", "Bypass", "-File", "MostrarIPs.ps1"],
            capture_output=True, text=True, check=True
        )
        salida = resultado.stdout.strip()
        print("\nListado de IPs encontradas:")
        print(salida if salida else "No se encontró ninguna IP.")

        ips = [ip.strip() for ip in salida.splitlines() if ip.strip()]
        registrar_log("IPs detectadas:\n" + "\n".join(ips) + "\n")
        return ips

    except subprocess.CalledProcessError as e:
        error_msg = f"Error al ejecutar el script: {e.stderr or str(e)}"
        print(error_msg)
        registrar_log(error_msg)
        return []

    except Exception as e:
        error_msg = f"Error inesperado al obtener IPs activas: {str(e)}"
        print(error_msg)
        registrar_log(error_msg)
        return []


def checar_ip(ip):
    """
    Consulta una IP en la API de AbuseIPDB.

    Args:
        ip (str): Dirección IP a consultar.
```

```
Returns:
    str: Resultado de la consulta o mensaje de error.
"""
try:
    resp = requests.get(
        API_URL,
        headers={
            'Accept': 'application/json',
            'Key': API_KEY
        },
        params={
            'ipAddress': ip,
            'maxAgeInDays': '90'
        },
        timeout=10
    )

    if resp.status_code != 200:
        error = f"Error al consultar IP {ip}: {resp.status_code} - {resp.text}"
        registrar_log(error)
        return error

    data = resp.json().get('data', {})
    score = data.get('abuseConfidenceScore', 0)
    resultado = (
        f"\nInformación de la IP {ip}:\n"
        f"- País: {data.get('countryCode', 'Desconocido')}\n"
        f"- ISP: {data.get('isp', 'Desconocido')}\n"
        f"- Reportes: {data.get('totalReports', 0)}\n"
        f"- Confianza en abuso: {score}%\n"
```

```
            f"- {'Maliciosa' if score >= 50 else 'No maliciosa'}\n"
        )

        registrar_log(resultado)
        return resultado

    except requests.exceptions.RequestException as e:
        error = f"Error de red al consultar IP {ip}: {str(e)}"
        print(error)
        registrar_log(error)
        return error

    except ValueError as e:
        error = f"Error procesando JSON para IP {ip}: {str(e)}"
        print(error)
        registrar_log(error)
        return error

    except Exception as e:
        error = f"Error inesperado al consultar IP {ip}: {str(e)}"
        print(error)
        registrar_log(error)
        return error


def main():
    """Función principal que orquesta la ejecución del script."""
    parser = argparse.ArgumentParser(description='Consulta de IPs en AbuseIPDB')
    parser.add_argument('ips', nargs='*', help='Lista opcional de IPs para consultar')
    args = parser.parse_args()
```

```
    fecha_actual = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    registrar_log(f"\n===== EJECUCIÓN DEL SCRIPT: {fecha_actual} =====")

    if args.ips:
        ips = args.ips
        print(f"IPs recibidas desde argumentos: {ips}")
    else:
        ips = obtener_ips_activas()

    if not ips:
        print("No se encontraron IPs.")
        registrar_log("No se encontraron IPs.\n")
        return

    print("\nConsultando máximo 3 IPs en AbuseIPDB...\n")
    for ip in ips[:3]:
        resultado = checar_ip(ip)
        print(resultado)



if __name__ == "__main__":
    try:
        main()
    except Exception as e:
        error = f"Error fatal en la ejecución del script: {str(e)}"
        print(error)
        registrar_log(error)
```

==========

==========

5 STYLE ISSUES FOUND

ISSUE 1

Line 8 - Issue code: E501

Line is longer than 79 characters.

You should rewrite your long line of code by breaking it down across multiple lines.

By making sure your lines of code are not too complicated means it's easier to understand by other people. Also by limiting the line width makes it possible to have several files open side-by-side, and works well when using code review tools that present the two versions in adjacent columns.----------
---

ISSUE 2

Line 29 - Issue code: E501

Line is longer than 79 characters.

You should rewrite your long line of code by breaking it down across multiple lines.

By making sure your lines of code are not too complicated means it's easier to understand by other people. Also by limiting the line width makes it possible to have several files open side-by-side, and works well when using code review tools that present the two versions in adjacent columns.----------
---

ISSUE 3

Line 78 - Issue code: E501

Line is longer than 79 characters.

You should rewrite your long line of code by breaking it down across multiple lines.

By making sure your lines of code are not too complicated means it's easier to understand by other people. Also by limiting the line width makes it possible to have several files open side-by-side, and works well when using code review tools that present the two versions in adjacent columns.----------
---

ISSUE 4

Line 117 - Issue code: E501

Line is longer than 79 characters.

You should rewrite your long line of code by breaking it down across multiple lines.

By making sure your lines of code are not too complicated means it's easier to understand by other people. Also by limiting the line width makes it possible to have several files open side-by-side, and works well when using code review tools that present the two versions in adjacent columns.----------
---

ISSUE 5

Line 118 - Issue code: E501

Line is longer than 79 characters.

You should rewrite your long line of code by breaking it down across multiple lines.

By making sure your lines of code are not too complicated means it's easier to understand by other people. Also by limiting the line width makes it possible to have several files open side-by-side, and works well when using code review tools that present the two versions in adjacent columns.==========
===