

```

pip install requests
# -----
import requests
from requests.exceptions import RequestException

def verificar_estado_sitio(url):
    """
    Intenta acceder a una URL y verifica su estado.

    Args:
        url (str): La URL del sitio web a verificar.
    """
    print(f"Verificando: {url}...")

    # Aseguramos que la URL comience con un esquema HTTP
    if not url.startswith(('http://', 'https://')):
        url = 'https://' + url

    try:
        # Hacemos una solicitud HEAD para mayor eficiencia,
        # ya que solo necesitamos el código de estado, no el contenido completo.
        # Establecemos un tiempo de espera (timeout) para evitar que se cuelgue.
        respuesta = requests.head(url, timeout=5)

        # Un código de estado 200 (OK) generalmente indica que el sitio está
        # activo.
        if respuesta.status_code == 200:
            print(f"✓ ¡El sitio está ACTIVO! (Código de estado: {respuesta.status_code})")
        else:
            print(f"⚠ ¡El sitio está CAÍDO o inaccesible! (Código de estado: {respuesta.status_code})")

    except RequestException as e:
        # Captura errores de conexión, DNS, timeouts, etc.
        print(f"✗ ¡El sitio está CAÍDO! Ocurrió un error de conexión.")
        print(f"    Detalles del error: {e}")
    except Exception as e:
        # Captura cualquier otro error inesperado
        print(f"✗ Ocurrió un error inesperado al verificar la URL!")
        print(f"    Detalles del error: {e}")

if __name__ == "__main__":
    # URL de ejemplo. Pide al usuario que ingrese la suya.
    url_a_verificar = input("Por favor, introduce la URL que deseas verificar\n(ej. google.com o https://bing.com): ")

    # Quitamos espacios en blanco accidentales
    url_a_verificar = url_a_verificar.strip()

    if url_a_verificar:
        verificar_estado_sitio(url_a_verificar)
    else:
        print("No se introdujo ninguna URL. Saliendo.")

```

# -----