

```

import requests
from bs4 import BeautifulSoup
import os
import time

# Função para fazer o download do arquivo
def download_file(url, save_path):
    response = requests.get(url)
    with open(save_path, 'wb') as f:
        f.write(response.content)

# Função para fazer uma requisição HTTP com retries
def make_request(url, retries=5, delay=5):
    for i in range(retries):
        try:
            response = requests.get(url)
            response.raise_for_status()
            return response
        except requests.exceptions.RequestException as e:
            print(f"Erro ao acessar {url}: {e}")
            if i < retries - 1:
                print(f"Tentando novamente em {delay} segundos...")
                time.sleep(delay)
            else:
                raise

# URL base inicial
base_url = 'https://dados.cvm.gov.br/dados/CIA_ABERTA/'

# Diretório base de destino
dest_dir_base = 'CIA_ABERTA'

# Lista para armazenar os diretórios a serem explorados, iniciando com o URL base
dirs_to_explore = [base_url]

# Loop para explorar os diretórios
while dirs_to_explore:
    # Remove o primeiro diretório da lista
    current_dir = dirs_to_explore.pop(0)

    # Faz a requisição HTTP para obter o conteúdo da página
    response = make_request(current_dir)
    soup = BeautifulSoup(response.text, 'html.parser')

    # Encontra todos os links na página
    for link in soup.find_all('a'):
        href = link.get('href')

        # Ignora os links que são de volta ('../')
        if href == '../':
            continue

        # Constrói a URL completa do link
        full_url = current_dir + href

        # Se o link for um diretório, adiciona à lista de diretórios a serem explorados
        if href.endswith('/'):
            dirs_to_explore.append(full_url)
        else:
            # Se o link for um arquivo, verifica se é um arquivo CSV, ZIP ou TXT e faz o download
            if href.endswith('.csv') or href.endswith('.zip') or href.endswith('.txt'):
                # Obtém o caminho relativo do arquivo em relação à URL base
                relative_path = os.path.relpath(full_url, base_url)
                # Constrói o caminho completo de destino
                save_dir = os.path.join(dest_dir_base, os.path.dirname(relative_path))
                # Cria os diretórios necessários se não existirem
                os.makedirs(save_dir, exist_ok=True)
                # Constrói o caminho completo do arquivo de destino
                save_path = os.path.join(save_dir, os.path.basename(full_url))
                # Faz o download do arquivo
                download_file(full_url, save_path)
                print("Arquivo baixado:", save_path)

    # Imprime o diretório atual
    print("Diretório atual:", current_dir)

```

