| Nome do Arquivo |
| --- |
| saving_handler.py |

| Documentação |
| --- |

```
#1[
#1 TITULO: SAVINGHANDLER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: GERENCIAR A CRIAÇÃO DE DIRETÓRIOS E O SALVAMENTO DE
INFORMAÇÕES DE REQUISIÇÕES EM ARQUIVOS JSON
#1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), DICIONÁRIO REQUEST_INFO
(NA FUNÇÃO SAVE_REQUEST_INFO)
#1 SAIDAS: CAMINHO DO ARQUIVO JSON GERADO COM AS INFORMAÇÕES DA
REQUISIÇÃO
#1 ROTINAS CHAMADAS: _CREATE_DIRECTORY, SAVE_REQUEST_INFO
#1]
    #1[
    #1 ROTINA: __INIT__
    #1 FINALIDADE: INICIALIZA A CLASSE SAVINGHANDLER E CRIA O DIRETÓRIO
DE SALVAMENTO SE NÃO EXISTIR
    #1 ENTRADAS: NOME DO DIRETÓRIO (STRING)
    #1 DEPENDENCIAS: LOGGINGHANDLER, OS
    #1 CHAMADO POR: SAVINGHANDLER
    #1 CHAMA: LOGGINGHANDLER.__INIT__, _CREATE_DIRECTORY
    #1]
    #2[
    #2 PSEUDOCODIGO DE: __init__
        #2 CHAMA O CONSTRUTOR DA CLASSE PAI E INICIALIZA O DIRETÓRIO
        #2 ARMAZENA O NOME DO DIRETÓRIO
        #2 CHAMA O MÉTODO PARA CRIAR O DIRETÓRIO SE ELE NÃO EXISTIR
    #2]
    #1[
    #1 ROTINA: _CREATE_DIRECTORY
    #1 FINALIDADE: VERIFICA SE O DIRETÓRIO EXISTE E O CRIA SE NECESSÁRIO
    #1 ENTRADAS: NENHUMA
    #1 DEPENDENCIAS: OS
    #1 CHAMADO POR: __INIT__
    #1 CHAMA: OS.MAKEDIRS
    #1]
    #2[
    #2 PSEUDOCODIGO DE: _create_directory
        #2 VERIFICA SE O DIRETÓRIO EXISTE
            #2 CRIA O DIRETÓRIO SE ELE NÃO EXISTIR
    #2]
    #1[
    #1 ROTINA: SAVE_REQUEST_INFO
    #1 FINALIDADE: SALVA AS INFORMAÇÕES DA REQUISIÇÃO EM UM ARQUIVO JSON
DENTRO DO DIRETÓRIO ESPECIFICADO
    #1 ENTRADAS: DICIONÁRIO REQUEST_INFO (CONTENDO DADOS DA REQUISIÇÃO)
    #1 DEPENDENCIAS: JSON, OS, DATETIME
    #1 CHAMADO POR: USUÁRIO
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: save_request_info
        #2 GERA UM TIMESTAMP PARA O NOME DO ARQUIVO
        #2 DEFINE O NOME DO ARQUIVO JSON COM BASE NO TIMESTAMP
        #2 DEFINE O CAMINHO COMPLETO DO ARQUIVO A SER SALVO
        #2 ABRE O ARQUIVO JSON EM MODO DE ESCRITA E SALVA O DICIONÁRIO
REQUEST_INFO
        #2 RETORNA O CAMINHO DO ARQUIVO SALVO
    #2]
```

| Nome do Arquivo |
| --- |
| leis_municipais_camara_sp.py |

| Documentação |
| --- |

```
#1[
#1 TITULO: LEISMUNICIPAISCAMARASP SCRAPPER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: REALIZAR O SCRAPING DE DECRETOS NO SITE
```

```
LEISMUNICIPAIS.COM.BR REFERENTE À CÂMARA DE SÃO PAULO, COM BASE EM
DIVERSOS ASSUNTOS
#1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), ASSUNTO (NA FUNÇÃO
SCRAPE)
#1 SAIDAS: DADOS PROCESSADOS, LINKS EXTRAÍDOS, DECRETOS VISITADOS
#1 ROTINAS CHAMADAS: SCRAPE, BUILD_URL, PARSE_CONTENT, EXTRACT_LINKS,
SCRAPE_LINKS, SCRAPE_ALL_SUBJECTS
#1]
    #1[
    #1 ROTINA: __INIT__
    #1 FINALIDADE: INICIALIZA A CLASSE LEISMUNICIPAISCAMARASP SCRAPPER E
CONFIGURA AS VARIÁVEIS INICIAIS E O DIRETÓRIO DE SALVAMENTO
    #1 ENTRADAS: NENHUMA
    #1 DEPENDENCIAS: LOGGINGHANDLER, SCRAPINGHANDLER
    #1 CHAMADO POR: LEISMUNICIPAISCAMARASP SCRAPPER
    #1 CHAMA: SCRAPINGHANDLER.__INIT__
    #1]
    #2[
    #2 PSEUDOCODIGO DE: __INIT__
        #2 CHAMA O CONSTRUTOR DA CLASSE PAI E CONFIGURA O DIRETÓRIO
        #2 INICIALIZA UM CONJUNTO PARA ARMAZENAR AS URLS JÁ VISITADAS
    #2]
    #1[
    #1 ROTINA: SCRAPE
    #1 FINALIDADE: REALIZAR O SCRAPING DE DECRETOS COM BASE NO ASSUNTO
ESPECIFICADO
    #1 ENTRADAS: ASSUNTO (STRING)
    #1 DEPENDENCIAS: TIME, REQUESTINGHANDLER
    #1 CHAMADO POR: USUÁRIO, SCRAPE_ALL_SUBJECTS
    #1 CHAMA: BUILD_URL, MAKE_REQUEST (REQUESTINGHANDLER),
PROCESS_RESPONSE, PARSE_CONTENT, EXTRACT_LINKS, SCRAPE_LINKS
    #1]
    #2[
    #2 PSEUDOCODIGO DE: SCRAPE
        #2 DEFINE O NÚMERO DA PÁGINA E O CONTADOR DE ERROS
        #2 ENQUANTO NÃO ATINGIR O LIMITE DE ERROS, CONTINUA FAZENDO O
SCRAPING
            #2 CONSTROI A URL BASEADA NO ASSUNTO E NÚMERO DA PÁGINA
            #2 FAZ A REQUISIÇÃO HTTP USANDO O HANDLER DE REQUISIÇÃO
            #2 PROCESSA A RESPOSTA RECEBIDA
            #2 SE NÃO HOUVER ERRO NA REQUISIÇÃO, PROSSEGUE COM O PARSE E
EXTRAÇÃO DE LINKS
                #2 ANALISA O CONTEÚDO HTML RECEBIDO
                #2 EXTRAI OS LINKS RELEVANTES DO CONTEÚDO HTML
                #2 REALIZA O SCRAPING DOS LINKS ENCONTRADOS
                #2 REINICIA O CONTADOR DE ERROS
                #2 INCREMENTA O CONTADOR DE ERROS SE OCORRER UM ERRO NA
REQUISIÇÃO
            #2 AUMENTA O NÚMERO DA PÁGINA E AGUARDA UM TEMPO ANTES DE
CONTINUAR
    #2]
    #1[
    #1 ROTINA: BUILD_URL
    #1 FINALIDADE: CONSTRUIR A URL PARA REALIZAR A REQUISIÇÃO HTTP
BASEADO NO ASSUNTO E NO NÚMERO DA PÁGINA
    #1 ENTRADAS: ASSUNTO (STRING), NÚMERO DA PÁGINA (INT)
    #1 DEPENDENCIAS: NENHUMA
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: BUILD_URL
        #2 RETORNA A URL CONSTRUÍDA COM O ASSUNTO E NÚMERO DA PÁGINA PARA
A BUSCA
    #2]
    #1[
    #1 ROTINA: PARSE_CONTENT
    #1 FINALIDADE: ANALISAR O CONTEÚDO HTML OBTIDO NA REQUISIÇÃO E
TRANSFORMÁ-LO EM UM OBJETO BEAUTIFULSOUP
    #1 ENTRADAS: CONTEÚDO HTML (STRING)
    #1 DEPENDENCIAS: BEAUTIFULSOUP
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: PARSE_CONTENT
        #2 TRANSFORMA O CONTEÚDO HTML EM UM OBJETO BEAUTIFULSOUP PARA
FACILITAR A EXTRAÇÃO DE DADOS
    #2]
    #1[
    #1 ROTINA: EXTRACT_LINKS
    #1 FINALIDADE: EXTRAIR LINKS RELEVANTES DO CONTEÚDO HTML ANALISADO
    #1 ENTRADAS: CONTEÚDO HTML ANALISADO (OBJETO BEAUTIFULSOUP)
```

```
    #1 DEPENDENCIAS: BEAUTIFULSOUP
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: EXTRACT_LINKS
        #2 EXTRAI TODOS OS LINKS QUE CONTÊM 'DECRETO' NO CAMINHO DO HREF
    #2]
    #1[
    #1 ROTINA: SCRAPE_LINKS
    #1 FINALIDADE: REALIZAR O SCRAPING DE CADA LINK EXTRAÍDO, SEGUINDO OS
LINKS E PROCESSANDO AS RESPOSTAS
    #1 ENTRADAS: LISTA DE LINKS (LISTA DE STRINGS)
    #1 DEPENDENCIAS: TIME, REQUESTINGHANDLER
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: MAKE_REQUEST (REQUESTINGHANDLER), PROCESS_RESPONSE
    #1]
    #2[
    #2 PSEUDOCODIGO DE: SCRAPE_LINKS
        #2 ITERA SOBRE CADA LINK EXTRAÍDO
            #2 CRIA A URL COMPLETA CONCATENANDO O LINK COM A
DECRETO_BASE_URL
            #2 VERIFICA SE A URL JÁ FOI VISITADA
                #2 ADICIONA A URL AO CONJUNTO DE URLs VISITADAS
                #2 FAZ A REQUISIÇÃO PARA A URL DO LINK
                #2 PROCESSA A RESPOSTA RECEBIDA
                #2 AGUARDA O TEMPO CONFIGURADO ENTRE REQUISIÇÕES
    #2]
    #1[
    #1 ROTINA: SCRAPE_ALL_SUBJECTS
    #1 FINALIDADE: REALIZAR O SCRAPING DE TODOS OS ASSUNTOS DEFINIDOS
SIMULTANEAMENTE UTILIZANDO MÚLTIPLAS THREADS
    #1 ENTRADAS: NENHUMA (UTILIZA OS ASSUNTOS DA VARIÁVEL DE CLASSE
SUBJECTS)
    #1 DEPENDENCIAS: THREADPOOLEXECUTOR
    #1 CHAMADO POR: USUÁRIO
    #1 CHAMA: SCRAPE (PARA CADA ASSUNTO)
    #1]
    #2[
    #2 PSEUDOCODIGO DE: SCRAPE_ALL_SUBJECTS
        #2 UTILIZA UM THREADPOOLEXECUTOR PARA PARALELIZAR O PROCESSO DE
SCRAPING PARA CADA ASSUNTO EM SUBJECTS
            #2 EXECUTA A FUNÇÃO SCRAPE PARA CADA ASSUNTO DA LISTA
    #2]
```

**Nome do Arquivo**

logging_handler.py

**Documentação**

```
#1[
#1 TITULO: LOGGING HANDLER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: GERENCIA O LOGGING PARA UMA CLASSE, PERMITINDO A CRIACAO
DE LOGS E FORMATO ESPECIFICO, E LOGAR EXECUCAO DE METODOS COM PARAMETROS
E RESULTADOS
#1 ENTRADAS: NOME DO DIRETORIO PARA ARMAZENAR OS LOGS, NOME DA CLASSE,
NOME DO METODO, OPCOES PARA LOGAR PARAMETROS E RESULTADOS
#1 SAIDAS: LOGS EM ARQUIVO DE TEXTO
#1 ROTINAS CHAMADAS: __INIT__, LOG_METHOD
#1 DEPENDENCIAS: LOGGING, OS
#1]
#1[
#1 ROTINA: __INIT__
#1 FINALIDADE: INICIALIZA O MANIPULADOR DE LOGS E CONFIGURA O FORMATO DE
LOGS PARA A CLASSE LOGGINGHANDLER
#1 ENTRADAS: NOME DO DIRETORIO PARA ARMAZENAR OS LOGS
#1 DEPENDENCIAS: LOGGING, OS
#1 CHAMADO POR: LOGGINGHANDLER
#1 CHAMA: OS.MAKEDIRS (SE O DIRETORIO NAO EXISTIR), LOGGING.GETLOGGER,
LOGGING.FILEHANDLER, LOGGING.FORMATTER
#1]
#2[
#2 PSEUDOCODIGO DE: __INIT__
        #2 OBTEM UM LOGGER ASSOCIADO AO NOME DA CLASSE
        #2 VERIFICA SE O DIRETORIO EXISTE, SE NAO, CRIA O DIRETORIO
        #2 DEFINE O CAMINHO DO ARQUIVO DE LOGS COMO LOGS.TXT
```

```
        #2 CRIA UM MANIPULADOR DE LOG PARA O ARQUIVO
        #2 DEFINE O FORMATO DO LOG COM O NOME DA CLASSE E O NOME DO
METODO
        #2 ADICIONA O MANIPULADOR AO LOGGER
        #2 DEFINE O NIVEL DE LOG PARA INFO
#2]
#1[
#1 ROTINA: LOG_METHOD
#1 FINALIDADE: DECORADOR PARA LOGAR A EXECUCAO DE METODOS, COM OPCOES
PARA EXIBIR PARAMETROS E RESULTADOS
#1 ENTRADAS: NOME DA CLASSE, NOME DO METODO, OPCOES DE EXIBIR PARAMETROS
E EXIBIR RESULTADO
#1 DEPENDENCIAS: LOGGING
#1 CHAMADO POR: LOGGINGHANDLER
#1 CHAMA: FUNC (METODO DECORADO)
#1]
#2[
#2 PSEUDOCODIGO DE: LOG_METHOD
        #2 DEFINE UM DECORADOR PARA O METODO
            #2 CRIA O WRAPPER QUE ENVOLVE O METODO ORIGINAL
                #2 SE EXIBIR PARAMETROS, GERA UMA STRING COM OS
PARAMETROS
                #2 LOGA OS PARAMETROS UTILIZANDO O LOGGER
                #2 EXECUTA O METODO ORIGINAL E OBTEM O RESULTADO
                #2 SE EXIBIR O RESULTADO, GERA UMA MENSAGEM DE LOG COM O
RESULTADO
                #2 RETORNA O RESULTADO DA EXECUCAO DO METODO ORIGINAL
            #2 RETORNA O WRAPPER COMO O NOVO METODO DECORADO
        #2 RETORNA O DECORADOR
#2]
```

| **Nome do Arquivo** |
| --- |
| requesting_handler.py |
| **Documentação** |

```
#1[
#1 TITULO: REQUESTINGHANDLER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: REALIZAR REQUISIÇÕES HTTP (GET E POST) E LOGAR O TEMPO DE
RESPOSTA E OUTRAS INFORMAÇÕES RELEVANTES
#1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), MÉTODO HTTP, URL, KWARGS
(DADOS ADICIONAIS OPCIONAIS)
#1 SAIDAS: DICIONÁRIO CONTENDO DADOS DA REQUISIÇÃO E RESPOSTA, INCLUINDO
TEMPO, TAMANHO E ERROS (SE HOUVER)
#1 ROTINAS CHAMADAS: SET_METHOD_MAPPING, _REQUEST_WRAPPER, MAKE_REQUEST
#1]
    #1[
    #1 ROTINA: __INIT__
    #1 FINALIDADE: INICIALIZA A CLASSE REQUESTINGHANDLER E CONFIGURA O
MAPEAMENTO DE MÉTODOS HTTP
    #1 ENTRADAS: NOME DO DIRETÓRIO (STRING)
    #1 DEPENDENCIAS: LOGGINGHANDLER
    #1 CHAMADO POR: REQUESTINGHANDLER
    #1 CHAMA: LOGGINGHANDLER.__INIT__, SET_METHOD_MAPPING
    #1]
    #2[
    #2 PSEUDOCODIGO DE: __init__
        #2 CHAMA O CONSTRUTOR DA CLASSE PAI PASSANDO O NOME DO DIRETÓRIO
        #2 CONFIGURA O MAPEAMENTO DE MÉTODOS HTTP
    #2]
    #1[
    #1 ROTINA: SET_METHOD_MAPPING
    #1 FINALIDADE: DEFINE O MAPEAMENTO DOS MÉTODOS HTTP (GET E POST)
    #1 ENTRADAS: NENHUMA
    #1 DEPENDENCIAS: REQUESTS
    #1 CHAMADO POR: __INIT__
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: set_method_mapping
        #2 DEFINE O MAPEAMENTO DE MÉTODOS HTTP 'GET' E 'POST'
    #2]
    #1[
    #1 ROTINA: _REQUEST_WRAPPER
    #1 FINALIDADE: ENVOLVE A EXECUÇÃO DA REQUISIÇÃO HTTP E COLETA DADOS
COMO TEMPO DE RESPOSTA E TAMANHO DA RESPOSTA
```

```
    #1 ENTRADAS: FUNÇÃO DO MÉTODO HTTP, URL, DICIONÁRIO REQUEST_INFO,
KWARGS OPCIONAIS
    #1 DEPENDENCIAS: TIME, REQUESTS
    #1 CHAMADO POR: MAKE_REQUEST
    #1 CHAMA: MÉTODOS HTTP (GET OU POST), RAISE_FOR_STATUS (REQUESTS)
    #1]
    #2[
    #2 PSEUDOCODIGO DE: _request_wrapper
            #2 OBTÉM O TEMPO DE INÍCIO DA REQUISIÇÃO
            #2 EXECUTA O MÉTODO HTTP COM A URL E OS KWARGS
            #2 OBTÉM O TEMPO FINAL DA REQUISIÇÃO
            #2 CALCULA O TEMPO DE RESPOSTA
            #2 OBTÉM O TAMANHO DO CONTEÚDO DA RESPOSTA
            #2 SALVA O TEXTO DA RESPOSTA
            #2 VERIFICA SE HOUVE ALGUM ERRO NA REQUISIÇÃO
            #2 EM CASO DE ERRO, ARMAZENA A MENSAGEM DE ERRO NO DICIONÁRIO
        #2 RETORNA AS INFORMAÇÕES DA REQUISIÇÃO
    #2]
    #1[
    #1 ROTINA: MAKE_REQUEST
    #1 FINALIDADE: FAZ A REQUISIÇÃO HTTP USANDO O MÉTODO ESPECIFICADO
(GET OU POST) E RETORNA AS INFORMAÇÕES DA REQUISIÇÃO
    #1 ENTRADAS: MÉTODO (GET OU POST), URL, KWARGS OPCIONAIS
    #1 DEPENDENCIAS: DATETIME, REQUESTS, TIME
    #1 CHAMADO POR: USUÁRIO
    #1 CHAMA: _REQUEST_WRAPPER
    #1]
    #2[
    #2 PSEUDOCODIGO DE: make_request
        #2 VERIFICA SE O MÉTODO É SUPORTADO (GET OU POST)
        #2 CRIA UM DICIONÁRIO PARA ARMAZENAR AS INFORMAÇÕES DA REQUISIÇÃO
        #2 CHAMA O MÉTODO _REQUEST_WRAPPER PARA REALIZAR A REQUISIÇÃO E
RETORNAR AS INFORMAÇÕES
    #2]
```

**Nome do Arquivo**

alesp.py

**Documentação**

```
#1[
#1 TITULO: ALESP SCRAPPER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: REALIZAR O SCRAPING DE NORMAS DA ASSEMBLEIA LEGISLATIVA DO
ESTADO DE SÃO PAULO (ALESP) COM BASE EM DETERMINADOS ASSUNTOS
#1 ENTRADAS: NOME DO DIRETÓRIO, ASSUNTO (NA FUNÇÃO SCRAPE)
#1 SAIDAS: DADOS DE RESPOSTA PROCESSADOS, URLS VISITADAS
#1 ROTINAS CHAMADAS: SCRAPE, BUILD_URL, PARSE_CONTENT, EXTRACT_LINKS,
SCRAPE_LINKS, SCRAPE_ALL_SUBJECTS
#1]
    #1[
    #1 ROTINA: __INIT__
    #1 FINALIDADE: INICIALIZA A CLASSE ALESP SCRAPPER E CONFIGURA AS
VARIÁVEIS INICIAIS E O DIRETÓRIO DE SALVAMENTO
    #1 ENTRADAS: NENHUMA
    #1 DEPENDENCIAS: LOGGINGHANDLER, SCRAPINGHANDLER
    #1 CHAMADO POR: ALESP SCRAPPER
    #1 CHAMA: SCRAPINGHANDLER.__INIT__
    #1]
    #2[
    #2 PSEUDOCODIGO DE: __INIT__
        #2 INICIALIZA A CLASSE PAI E O DIRETÓRIO
        #2 INICIALIZA O CONJUNTO DE URLS VISITADAS
    #2]
    #1[
    #1 ROTINA: SCRAPE
    #1 FINALIDADE: REALIZAR O SCRAPING DE NORMAS COM BASE NO ASSUNTO
ESPECIFICADO
    #1 ENTRADAS: ASSUNTO (STRING)
    #1 DEPENDENCIAS: URLLIB, TIME, REQUESTINGHANDLER
    #1 CHAMADO POR: USUÁRIO, SCRAPE_ALL_SUBJECTS
    #1 CHAMA: BUILD_URL, MAKE_REQUEST (REQUESTINGHANDLER),
PROCESS_RESPONSE, PARSE_CONTENT, EXTRACT_LINKS, SCRAPE_LINKS
    #1]
    #2[
    #2 PSEUDOCODIGO DE: SCRAPE
        #2 ESCAPA O ASSUNTO PARA FORMATO DE URL E DEPOIS DECODIFICA
```

```
        #2 REALIZA O LOOP PARA TENTAR ACESSAR PÁGINAS ENQUANTO O NÚMERO
MÁXIMO DE ERROS NÃO FOR ALCANÇADO
            #2 CONSTROI A URL COM BASE NO ASSUNTO E NÚMERO DA PÁGINA
            #2 REALIZA A REQUISIÇÃO HTTP
            #2 PROCESSA A RESPOSTA
                #2 ANALISA O CONTEÚDO HTML
                #2 EXTRAI OS LINKS ENCONTRADOS NO CONTEÚDO HTML
                #2 REALIZA O SCRAPING DOS LINKS ENCONTRADOS
    #2]
    #1[
    #1 ROTINA: BUILD_URL
    #1 FINALIDADE: CONSTRUIR A URL PARA REALIZAR A REQUISIÇÃO HTTP
BASEADO NO ASSUNTO E NO NÚMERO DA PÁGINA
    #1 ENTRADAS: NÚMERO DA PÁGINA (INT), ASSUNTO ESCAPADO (STRING),
ASSUNTO DECODIFICADO (STRING)
    #1 DEPENDENCIAS: NENHUMA
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: BUILD_URL
        #2 CONSTROI A URL PARA BUSCA DE NORMAS COM BASE NO NÚMERO DA
PÁGINA E NO ASSUNTO
    #2]
    #1[
    #1 ROTINA: PARSE_CONTENT
    #1 FINALIDADE: ANALISAR O CONTEÚDO HTML OBTIDO NA REQUISIÇÃO E
TRANSFORMÁ-LO EM UM OBJETO BEAUTIFULSOUP
    #1 ENTRADAS: CONTEÚDO HTML (STRING)
    #1 DEPENDENCIAS: BEAUTIFULSOUP
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: PARSE_CONTENT
        #2 TRANSFORMA O CONTEÚDO HTML EM UM OBJETO BEAUTIFULSOUP PARA
FACILITAR A EXTRAÇÃO DE DADOS
    #2]
    #1[
    #1 ROTINA: EXTRACT_LINKS
    #1 FINALIDADE: EXTRAIR LINKS RELEVANTES DO CONTEÚDO HTML ANALISADO
    #1 ENTRADAS: CONTEÚDO HTML ANALISADO (OBJETO BEAUTIFULSOUP)
    #1 DEPENDENCIAS: BEAUTIFULSOUP
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: EXTRACT_LINKS
        #2 EXTRAI TODOS OS LINKS QUE CONTÊM 'NORMA' NO CAMINHO DO HREF
    #2]
    #1[
    #1 ROTINA: SCRAPE_LINKS
    #1 FINALIDADE: REALIZAR O SCRAPING DE CADA LINK EXTRAÍDO, SEGUINDO OS
LINKS E PROCESSANDO AS RESPOSTAS
    #1 ENTRADAS: LISTA DE LINKS (LISTA DE STRINGS)
    #1 DEPENDENCIAS: TIME, REQUESTINGHANDLER
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: MAKE_REQUEST (REQUESTINGHANDLER), PROCESS_RESPONSE
    #1]
    #2[
    #2 PSEUDOCODIGO DE: SCRAPE_LINKS
        #2 ITERA SOBRE CADA LINK EXTRAÍDO
            #2 CRIA A URL COMPLETA CONCATENANDO O LINK COM A BASE_URL DA
NORMA
            #2 VERIFICA SE A URL JÁ FOI VISITADA
                #2 ADICIONA A URL AO CONJUNTO DE URLs VISITADAS
                #2 FAZ A REQUISIÇÃO PARA A URL DO LINK
                #2 PROCESSA A RESPOSTA RECEBIDA
                #2 AGUARDA O TEMPO CONFIGURADO ENTRE REQUISIÇÕES
    #2]
    #1[
    #1 ROTINA: SCRAPE_ALL_SUBJECTS
    #1 FINALIDADE: REALIZAR O SCRAPING DE TODOS OS ASSUNTOS DEFINIDOS NA
VARIÁVEL SUBJECTS UTILIZANDO MÚLTIPLAS THREADS
    #1 ENTRADAS: NENHUMA (UTILIZA OS ASSUNTOS DA VARIÁVEL DE CLASSE
SUBJECTS)
    #1 DEPENDENCIAS: THREADPOOLEXECUTOR
    #1 CHAMADO POR: USUÁRIO, FUNÇÃO MAIN
    #1 CHAMA: SCRAPE (PARA CADA ASSUNTO)
    #1]
    #2[
    #2 PSEUDOCODIGO DE: SCRAPE_ALL_SUBJECTS
        #2 UTILIZA UM THREADPOOLEXECUTOR PARA PARALELIZAR O PROCESSO DE
```

```
SCRAPING PARA TODOS OS ASSUNTOS
            #2 EXECUTA A FUNÇÃO SCRAPE PARA CADA ASSUNTO EM SUBJECTS
SIMULTANEAMENTE
    #2]
```

**Nome do Arquivo**

text_handler.py

**Documentação**

```
#1[
#1 TITULO: TEXTHANDLER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: REALIZAR EXTRAÇÃO E FILTRAGEM DE TEXTO BRUTO A PARTIR DE
CONTEÚDO HTML
#1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), CONTEÚDO HTML (NA FUNÇÃO
EXTRACT_RAW_TEXT)
#1 SAIDAS: TEXTO LIMPO EM MINÚSCULAS EXTRAÍDO DO CONTEÚDO HTML
#1 ROTINAS CHAMADAS: EXTRACT_RAW_TEXT, _FILTER_PORTUGUESE_TEXT
#1]
    #1[
    #1 ROTINA: __INIT__
    #1 FINALIDADE: INICIALIZA A CLASSE TEXTHANDLER E CONFIGURA O
DIRETÓRIO
    #1 ENTRADAS: NOME DO DIRETÓRIO (STRING)
    #1 DEPENDENCIAS: LOGGINGHANDLER
    #1 CHAMADO POR: TEXTHANDLER
    #1 CHAMA: LOGGINGHANDLER.__INIT__
    #1]
    #2[
    #2 PSEUDOCODIGO DE: __init__
        #2 CHAMA O CONSTRUTOR DA CLASSE PAI E INICIALIZA O DIRETÓRIO
        #2 ARMAZENA O NOME DO DIRETÓRIO
    #2]
    #1[
    #1 ROTINA: EXTRACT_RAW_TEXT
    #1 FINALIDADE: EXTRAI O TEXTO BRUTO DO CONTEÚDO HTML E O FILTRA PARA
REMOVER CARACTERES INDESEJADOS
    #1 ENTRADAS: CONTEÚDO HTML (STRING)
    #1 DEPENDENCIAS: BEAUTIFULSOUP, RE
    #1 CHAMADO POR: USUÁRIO
    #1 CHAMA: _FILTER_PORTUGUESE_TEXT
    #1]
    #2[
    #2 PSEUDOCODIGO DE: extract_raw_text
            #2 ANALISA O CONTEÚDO HTML USANDO BEAUTIFULSOUP
            #2 EXTRAI O TEXTO BRUTO DO CONTEÚDO HTML
            #2 FILTRA O TEXTO PARA REMOVER CARACTERES NÃO PERTENCENTES AO
PORTUGUÊS
            #2 RETORNA O TEXTO FILTRADO EM MINÚSCULAS
            #2 RETORNA UMA STRING VAZIA EM CASO DE FALHA
    #2]
    #1[
    #1 ROTINA: _FILTER_PORTUGUESE_TEXT
    #1 FINALIDADE: FILTRA O TEXTO PARA REMOVER CARACTERES QUE NÃO SÃO DO
PORTUGUÊS
    #1 ENTRADAS: TEXTO BRUTO (STRING)
    #1 DEPENDENCIAS: RE
    #1 CHAMADO POR: EXTRACT_RAW_TEXT
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: _filter_portuguese_text
        #2 DEFINE UM PADRÃO PARA MANTER APENAS CARACTERES PORTUGUESES,
NÚMEROS E PONTUAÇÃO
        #2 APLICA O PADRÃO PARA REMOVER CARACTERES INDESEJADOS
        #2 RETORNA O TEXTO FILTRADO
    #2]
```

**Nome do Arquivo**

dataframe_handler.py

**Documentação**

```
#1[
#1 TITULO: DATAFRAMEHANDLER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: MANIPULAR E PROCESSAR ARQUIVOS JSON, TRANSFORMANDO-OS EM
DATAFRAMES E EXTRAINDO INFORMAÇÕES ÚTEIS.
#1 ENTRADAS: NENHUMA
#1 SAIDAS: ARQUIVOS CSV GERADOS A PARTIR DE DATAFRAMES
#1 ROTINAS CHAMADAS: LOAD_JSON_FILES, _PROCESS_AND_SAVE_DATAFRAME,
EXTRACT_URL_INFORMATION, _EXTRACT_INSTITUTION_NAME, _EXTRACT_SUBJECT,
EXECUTE
#1]
    #1[
    #1 ROTINA: __INIT__
    #1 FINALIDADE: INICIALIZA AS VARIÁVEIS DA CLASSE DATAFRAMEHANDLER E
CHAMA O CONSTRUTOR DA CLASSE PAI.
    #1 ENTRADAS: NENHUMA
    #1 DEPENDENCIAS: LOGGINGHANDLER, OS
    #1 CHAMADO POR: DATAFRAMEHANDLER
    #1 CHAMA: LOGGINGHANDLER.__INIT__
    #1]
    #2[
    #2 PSEUDOCODIGO DE: __init__
        #2 CHAMA O CONSTRUTOR DA CLASSE PAI PASSANDO O NOME DO DIRETÓRIO
    #2]
    #1[
    #1 ROTINA: LOAD_JSON_FILES
    #1 FINALIDADE: CARREGA TODOS OS ARQUIVOS JSON DO DIRETÓRIO E OS
PROCESSA.
    #1 ENTRADAS: NENHUMA
    #1 DEPENDENCIAS: OS, JSON, TQDM, RE
    #1 CHAMADO POR: EXECUTE
    #1 CHAMA: _PROCESS_AND_SAVE_DATAFRAME
    #1]
    #2[
    #2 PSEUDOCODIGO DE: load_json_files
        #2 ITERA SOBRE OS ARQUIVOS NO DIRETÓRIO ATUAL
            #2 FILTRA ARQUIVOS COM EXTENSÃO .JSON
            #2 CARREGA CADA ARQUIVO JSON E LIMPA OS DADOS
            #2 SE EXISTIREM DADOS, PROCESSA E SALVA EM CSV
    #2]
    #1[
    #1 ROTINA: _PROCESS_AND_SAVE_DATAFRAME
    #1 FINALIDADE: PROCESSA OS DADOS E SALVA EM ARQUIVOS CSV NO DIRETÓRIO
CORRESPONDENTE.
    #1 ENTRADAS: LISTA DE DICIONÁRIOS COM OS DADOS JSON E O CAMINHO DO
DIRETÓRIO
    #1 DEPENDENCIAS: PANDAS, OS
    #1 CHAMADO POR: LOAD_JSON_FILES
    #1 CHAMA: EXTRACT_URL_INFORMATION
    #1]
    #2[
    #2 PSEUDOCODIGO DE: _process_and_save_dataframe
        #2 CRIA UM DATAFRAME A PARTIR DOS DADOS JSON
        #2 EXTRAI INFORMAÇÕES DA URL E ADICIONA AO DATAFRAME
        #2 OBTÉM O NOME DA INSTITUIÇÃO, OU DEFINE COMO 'DESCONHECIDO'
        #2 DEFINE O CAMINHO DE SAÍDA PARA O CSV
        #2 DEFINE O TAMANHO DE CADA LOTE DE CSV A SER GERADO
        #2 GERA E SALVA CADA PARTE DO CSV
    #2]
    #1[
    #1 ROTINA: EXTRACT_URL_INFORMATION
    #1 FINALIDADE: EXTRAI INFORMAÇÕES RELEVANTES DAS URLS PRESENTES NO
DATAFRAME.
    #1 ENTRADAS: DATAFRAME COM A COLUNA 'URL'
    #1 DEPENDENCIAS: RE
    #1 CHAMADO POR: _PROCESS_AND_SAVE_DATAFRAME
    #1 CHAMA: _EXTRACT_INSTITUTION_NAME, _EXTRACT_SUBJECT
    #1]
    #2[
    #2 PSEUDOCODIGO DE: extract_url_information
        #2 VERIFICA SE A COLUNA 'URL' EXISTE NO DATAFRAME
        #2 EXTRAI O DOMÍNIO BASE DA URL
        #2 EXTRAI O NOME DA INSTITUIÇÃO A PARTIR DO DOMÍNIO
        #2 EXTRAI O ASSUNTO RELACIONADO COM A URL
    #2]
    #1[
    #1 ROTINA: _EXTRACT_INSTITUTION_NAME
    #1 FINALIDADE: IDENTIFICA O NOME DA INSTITUIÇÃO BASEADO NO DOMÍNIO DA
URL.
```

```
    #1 ENTRADAS: BASE_URL (STRING)
    #1 DEPENDENCIAS: RE
    #1 CHAMADO POR: EXTRACT_URL_INFORMATION
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: _extract_institution_name
        #2 VERIFICA SE O DOMÍNIO CONTÉM O TEXTO 'PREFEITURA.SP'
        #2 VERIFICA SE O DOMÍNIO CONTÉM O TEXTO 'AL.SP'
        #2 VERIFICA SE O DOMÍNIO CONTÉM O TEXTO 'LEISMUNICIPAIS.COM.BR'
        #2 RETORNA 'OUTROS' SE NENHUMA CONDIÇÃO FOR ATENDIDA
    #2]
    #1[
    #1 ROTINA: _EXTRACT_SUBJECT
    #1 FINALIDADE: EXTRAI O ASSUNTO DA URL COM BASE NO NOME DA
INSTITUIÇÃO.
    #1 ENTRADAS: URL (STRING), NOME DA INSTITUIÇÃO (STRING)
    #1 DEPENDENCIAS: RE
    #1 CHAMADO POR: EXTRACT_URL_INFORMATION
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: _extract_subject
        #2 EXTRAI O ASSUNTO PARA INSTITUIÇÕES DA ALESP
        #2 EXTRAI O ASSUNTO PARA INSTITUIÇÕES DA PREFEITURA SP
        #2 EXTRAI O ASSUNTO PARA INSTITUIÇÕES DA CÂMARA SP
        #2 RETORNA STRING VAZIA SE NÃO HOUVER MATCH
    #2]
    #1[
    #1 ROTINA: EXECUTE
    #1 FINALIDADE: EXECUTA A ROTINA PRINCIPAL DE CARREGAMENTO E
PROCESSAMENTO DOS ARQUIVOS JSON.
    #1 ENTRADAS: NENHUMA
    #1 DEPENDENCIAS: LOGGINGHANDLER
    #1 CHAMADO POR: USUÁRIO
    #1 CHAMA: LOAD_JSON_FILES
    #1]
    #2[
    #2 PSEUDOCODIGO DE: execute
        #2 INICIA O PROCESSO DE CARREGAMENTO E PROCESSAMENTO DOS ARQUIVOS
JSON
    #2]
```

| **Nome do Arquivo** |
| --- |
| logs.txt |
| **Documentação** |
| |

| **Nome do Arquivo** |
| --- |
| main.py |
| **Documentação** |

```
#1[
#1 TITULO: MAIN
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: INICIALIZAR O HANDLER DE DATAFRAMES E EXECUTAR SUAS
FUNÇÕES
#1 ENTRADAS: NENHUMA
#1 SAIDAS: EXECUÇÃO DAS FUNÇÕES DO DATAFRAMEHANDLER
#1 ROTINAS CHAMADAS: EXECUTE (DATAFRAMEHANDLER)
#1]
#1[
#1 ROTINA: MAIN
#1 FINALIDADE: CRIA UMA INSTÂNCIA DO DATAFRAMEHANDLER E EXECUTA SUAS
FUNÇÕES
#1 ENTRADAS: NENHUMA
#1 DEPENDENCIAS: DATAFRAMEHANDLER
#1 CHAMADO POR: SCRIPT PRINCIPAL
```

```
#1 CHAMA: EXECUTE (DATAFRAMEHANDLER)
#1]
#2[
#2 PSEUDOCODIGO DE: MAIN
    #2 INICIALIZA O HANDLER DE DATAFRAMES
    #2 EXECUTA A FUNÇÃO PRINCIPAL DO HANDLER
#2]
```

**Nome do Arquivo**

toponym_handler.py

**Documentação**

```
#1[
#1 TITULO: TOPONYMHANDLER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: LEMATIZAR TEXTOS E EXTRAIR TOPÔNIMOS (NOMES DE LOCAIS)
USANDO PADRÕES DEFINIDOS
#1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), TEXTO (NA FUNÇÃO
LEMMATIZE_AND_EXTRACT_TOPONYMS)
#1 SAIDAS: TEXTO LEMATIZADO E LISTA DE TOPÔNIMOS EXTRAÍDOS
#1 ROTINAS CHAMADAS: _ADD_PATTERNS, LEMMATIZE_AND_EXTRACT_TOPONYMS
#1]
    #1[
    #1 ROTINA: __INIT__
    #1 FINALIDADE: INICIALIZA A CLASSE TOPONYMHANDLER E CONFIGURA O
PROCESSAMENTO DE LINGUAGEM NATURAL E PADRÕES DE TOPÔNIMOS
    #1 ENTRADAS: NOME DO DIRETÓRIO (STRING)
    #1 DEPENDENCIAS: SPACY, MATCHER, LOGGINGHANDLER
    #1 CHAMADO POR: TOPONYMHANDLER
    #1 CHAMA: LOGGINGHANDLER.__INIT__, SPACY.LOAD, MATCHER
    #1]
    #2[
    #2 PSEUDOCODIGO DE: __INIT__
        #2 CHAMA O CONSTRUTOR DA CLASSE PAI E INICIALIZA O DIRETÓRIO
        #2 ARMAZENA O NOME DO DIRETÓRIO
        #2 CARREGA O MODELO DE LINGUAGEM NATURAL EM PORTUGUÊS USANDO
SPACY
        #2 INICIALIZA O MATCHER (PARA DETECTAR PADRÕES DE TOPÔNIMOS)
        #2 CHAMA A FUNÇÃO PARA ADICIONAR PADRÕES AO MATCHER
    #2]
    #1[
    #1 ROTINA: _ADD_PATTERNS
    #1 FINALIDADE: ADICIONA PADRÕES DE TOPÔNIMOS AO MATCHER USANDO NOMES
PRÓPRIOS E PALAVRAS-CHAVE RELACIONADAS A LOCAIS
    #1 ENTRADAS: NENHUMA
    #1 DEPENDENCIAS: SPACY, RE
    #1 CHAMADO POR: __INIT__
    #1 CHAMA: MATCHER.ADD
    #1]
    #2[
    #2 PSEUDOCODIGO DE: _ADD_PATTERNS
        #2 DEFINE OS TERMOS RELACIONADOS A LOCAIS E SUAS ABREVIAÇÕES
        #2 DEFINE PADRÕES FIXOS QUE CORRESPONDEM A NOMES PRÓPRIOS E
ESTRUTURAS COMUNS EM NOMES DE LOCAIS
            #2 EX: SÃO PAULO
            #2 EX: RIO DE JANEIRO
            #2 EX: CIDADE DE SÃO PAULO
            #2 EX: VILA NOVA DE GAIA
            #2 EX: SÃO PEDRO E SÃO PAULO
            #2 EX: O RIO DE JANEIRO
            #2 EX: A CIDADE DE SÃO PAULO
            #2 EX: A CIDADE DE SÃO PAULO E RIO DE JANEIRO
            #2 EX: ALOYSIO NUNES
            #2 EX: PEDRO ÁLVARES CABRAL
            #2 EX: ALOYSIO NUNES FERREIRA FILHO
        #2 DEFINE PADRÕES PARA NOMES PRÓPRIOS COM HÍFEN
            #2 DOIS NOMES PRÓPRIOS COM HÍFEN EX: PEDRO ÁLVARES-CABRAL
            #2 TRÊS NOMES PRÓPRIOS COM UM HÍFEN EX: PEDRO ÁLVARES-CABRAL
FILHO
            #2 DOIS NOMES PRÓPRIOS SEGUIDOS DE UM HÍFEN E MAIS DOIS NOMES
PRÓPRIOS EX: JOÃO PAULO-SILVA
            #2 TRÊS NOMES PRÓPRIOS COM TRES HÍFENS EX: JOÃO-PAULO-SILVA
            #2 TRÊS NOMES PRÓPRIOS COM DOIS HÍFENS EX: JOÃO-PAULO-SILVA
FILHO
            #2 QUATRO NOMES PRÓPRIOS COM DOIS HÍFENS EX: JOÃO-PAULO-
SILVA-SANTOS
```

```
        #2 ADICIONA OS PADRÕES FIXOS AO MATCHER
        #2 ADICIONA VARIAÇÕES DOS PADRÕES USANDO PALAVRAS-CHAVE DE LOCAIS
                #2 CADA PADRÃO BASE É PRECEDIDO POR UMA PALAVRA-CHAVE DE
LOCAL
            #2 ADICIONA VARIAÇÕES COM HÍFENS PARA CADA PALAVRA-CHAVE
        #2 ADICIONA OS PADRÕES COM HÍFEN AO MATCHER
        #2 ADICIONA TODOS OS PADRÕES AO MATCHER
    #2]
    #1[
    #1 ROTINA: LEMMATIZE_AND_EXTRACT_TOPONYMS
    #1 FINALIDADE: LEMATIZA O TEXTO E EXTRAI OS TOPÔNIMOS USANDO O
MATCHER
    #1 ENTRADAS: TEXTO (STRING)
    #1 DEPENDENCIAS: SPACY, MATCHER
    #1 CHAMADO POR: USUÁRIO
    #1 CHAMA: MATCHER, NLP
    #1]
    #2[
    #2 PSEUDOCODIGO DE: LEMMATIZE_AND_EXTRACT_TOPONYMS
            #2 PROCESSA O TEXTO USANDO O MODELO NLP
            #2 LEMATIZA CADA TOKEN DO TEXTO
            #2 ENCONTRA OS PADRÕES CORRESPONDENTES AOS TOPÔNIMOS NO TEXTO
            #2 EXTRAI OS TOPÔNIMOS IDENTIFICADOS
            #2 RETORNA O TEXTO LEMATIZADO E OS TOPÔNIMOS COMO UMA STRING
SEPARADA POR VÍRGULAS
            #2 RETORNA STRINGS VAZIAS EM CASO DE ERRO
    #2]
```

**Nome do Arquivo**

legislacao_prefeitura_sp_gov_br.py

**Documentação**

```
#1[
#1 TITULO: LEGISLACAOPREFEITURASPGOVBR SCRAPPER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: REALIZAR O SCRAPING DE LEGISLAÇÕES NO SITE DA PREFEITURA
DE SÃO PAULO COM BASE EM DIVERSOS ASSUNTOS
#1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), ASSUNTO (NA FUNÇÃO
SCRAPE)
#1 SAIDAS: DADOS PROCESSADOS, LINKS EXTRAÍDOS, LEGISLAÇÕES VISITADAS
#1 ROTINAS CHAMADAS: SCRAPE, BUILD_URL, PARSE_CONTENT, EXTRACT_LINKS,
SCRAPE_LINKS, SCRAPE_ALL_SUBJECTS
#1]
    #1[
    #1 ROTINA: __INIT__
    #1 FINALIDADE: INICIALIZA A CLASSE LEGISLACAOPREFEITURASPGOVBR
SCRAPPER E CONFIGURA AS VARIÁVEIS INICIAIS E O DIRETÓRIO DE SALVAMENTO
    #1 ENTRADAS: NENHUMA
    #1 DEPENDENCIAS: LOGGINGHANDLER, SCRAPINGHANDLER
    #1 CHAMADO POR: LEGISLACAOPREFEITURASPGOVBR SCRAPPER
    #1 CHAMA: SCRAPINGHANDLER.__INIT__
    #1]
    #2[
    #2 PSEUDOCODIGO DE: __INIT__
        #2 CHAMA O CONSTRUTOR DA CLASSE PAI E CONFIGURA O DIRETÓRIO
        #2 INICIALIZA UM CONJUNTO PARA ARMAZENAR AS URLS JÁ VISITADAS
    #2]
    #1[
    #1 ROTINA: SCRAPE
    #1 FINALIDADE: REALIZAR O SCRAPING DE LEGISLAÇÕES COM BASE NO ASSUNTO
ESPECIFICADO
    #1 ENTRADAS: ASSUNTO (STRING)
    #1 DEPENDENCIAS: TIME, REQUESTINGHANDLER
    #1 CHAMADO POR: USUÁRIO, SCRAPE_ALL_SUBJECTS
    #1 CHAMA: BUILD_URL, MAKE_REQUEST (REQUESTINGHANDLER),
PROCESS_RESPONSE, PARSE_CONTENT, EXTRACT_LINKS, SCRAPE_LINKS
    #1]
    #2[
    #2 PSEUDOCODIGO DE: SCRAPE
        #2 DEFINE O NÚMERO DA PÁGINA E O CONTADOR DE ERROS
        #2 ENQUANTO NÃO ATINGIR O LIMITE DE ERROS, CONTINUA FAZENDO O
SCRAPING
            #2 CONSTROI A URL BASEADA NO ASSUNTO E NÚMERO DA PÁGINA
            #2 FAZ A REQUISIÇÃO HTTP USANDO O HANDLER DE REQUISIÇÃO
            #2 PROCESSA A RESPOSTA RECEBIDA
            #2 SE NÃO HOUVER ERRO NA REQUISIÇÃO, PROSSEGUE COM O PARSE E
```

```
EXTRAÇÃO DE LINKS
                #2 ANALISA O CONTEÚDO HTML RECEBIDO
                #2 EXTRAI OS LINKS RELEVANTES DO CONTEÚDO HTML
                #2 REALIZA O SCRAPING DOS LINKS ENCONTRADOS
                #2 REINICIA O CONTADOR DE ERROS
                #2 INCREMENTA O CONTADOR DE ERROS SE OCORRER UM ERRO NA
REQUISIÇÃO
            #2 AUMENTA O NÚMERO DA PÁGINA E AGUARDA UM TEMPO ANTES DE
CONTINUAR
    #2]
    #1[
    #1 ROTINA: BUILD_URL
    #1 FINALIDADE: CONSTRUIR A URL PARA REALIZAR A REQUISIÇÃO HTTP
BASEADO NO ASSUNTO E NO NÚMERO DA PÁGINA
    #1 ENTRADAS: ASSUNTO (STRING), NÚMERO DA PÁGINA (INT)
    #1 DEPENDENCIAS: NENHUMA
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: BUILD_URL
        #2 RETORNA A URL CONSTRUÍDA COM O ASSUNTO E NÚMERO DA PÁGINA PARA
A BUSCA
    #2]
    #1[
    #1 ROTINA: PARSE_CONTENT
    #1 FINALIDADE: ANALISAR O CONTEÚDO HTML OBTIDO NA REQUISIÇÃO E
TRANSFORMÁ-LO EM UM OBJETO BEAUTIFULSOUP
    #1 ENTRADAS: CONTEÚDO HTML (STRING)
    #1 DEPENDENCIAS: BEAUTIFULSOUP
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: PARSE_CONTENT
        #2 TRANSFORMA O CONTEÚDO HTML EM UM OBJETO BEAUTIFULSOUP PARA
FACILITAR A EXTRAÇÃO DE DADOS
    #2]
    #1[
    #1 ROTINA: EXTRACT_LINKS
    #1 FINALIDADE: EXTRAIR LINKS RELEVANTES DO CONTEÚDO HTML ANALISADO
    #1 ENTRADAS: CONTEÚDO HTML ANALISADO (OBJETO BEAUTIFULSOUP)
    #1 DEPENDENCIAS: BEAUTIFULSOUP
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: NENHUMA
    #1]
    #2[
    #2 PSEUDOCODIGO DE: EXTRACT_LINKS
        #2 EXTRAI TODOS OS LINKS QUE CONTÊM 'LEIS' NO CAMINHO DO HREF
    #2]
    #1[
    #1 ROTINA: SCRAPE_LINKS
    #1 FINALIDADE: REALIZAR O SCRAPING DE CADA LINK EXTRAÍDO, SEGUINDO OS
LINKS E PROCESSANDO AS RESPOSTAS
    #1 ENTRADAS: LISTA DE LINKS (LISTA DE STRINGS)
    #1 DEPENDENCIAS: TIME, REQUESTINGHANDLER
    #1 CHAMADO POR: SCRAPE
    #1 CHAMA: MAKE_REQUEST (REQUESTINGHANDLER), PROCESS_RESPONSE
    #1]
    #2[
    #2 PSEUDOCODIGO DE: SCRAPE_LINKS
        #2 ITERA SOBRE CADA LINK EXTRAÍDO
            #2 CRIA A URL COMPLETA CONCATENANDO O LINK COM A BASE_URL
            #2 VERIFICA SE A URL JÁ FOI VISITADA
                #2 ADICIONA A URL AO CONJUNTO DE URLs VISITADAS
                #2 FAZ A REQUISIÇÃO PARA A URL DO LINK
                #2 PROCESSA A RESPOSTA RECEBIDA
                #2 AGUARDA O TEMPO CONFIGURADO ENTRE REQUISIÇÕES
    #2]
    #1[
    #1 ROTINA: SCRAPE_ALL_SUBJECTS
    #1 FINALIDADE: REALIZAR O SCRAPING DE TODOS OS ASSUNTOS DEFINIDOS
SIMULTANEAMENTE UTILIZANDO MÚLTIPLAS THREADS
    #1 ENTRADAS: NENHUMA (UTILIZA OS ASSUNTOS DA VARIÁVEL DE CLASSE
SUBJECTS)
    #1 DEPENDENCIAS: THREADPOOLEXECUTOR
    #1 CHAMADO POR: USUÁRIO
    #1 CHAMA: SCRAPE (PARA CADA ASSUNTO)
    #1]
    #2[
    #2 PSEUDOCODIGO DE: SCRAPE_ALL_SUBJECTS
        #2 UTILIZA UM THREADPOOLEXECUTOR PARA PARALELIZAR O PROCESSO DE
SCRAPING PARA CADA ASSUNTO EM SUBJECTS
```

```
            #2 EXECUTA A FUNÇÃO SCRAPE PARA CADA ASSUNTO DA LISTA
    #2]
```

| Nome do Arquivo |
| --- |
| scraping_handler.py |

**Documentação**

```
#1[
#1 TITULO: SCRAPINGHANDLER
#1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA
#1 DATA: 07/10/2024
#1 VERSAO: 1
#1 FINALIDADE: REALIZAR O PROCESSO DE SCRAPING, EXTRAÇÃO DE TEXTO,
LEMATIZAÇÃO E SALVAMENTO DOS DADOS EM ARQUIVOS JSON
#1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), DICIONÁRIO DATA_DICT (NA
FUNÇÃO PROCESS_RESPONSE)
#1 SAIDAS: DICIONÁRIO PROCESSADO COM CAMPOS EXTRAÍDOS E ARQUIVO JSON COM
AS INFORMAÇÕES SALVAS
#1 ROTINAS CHAMADAS: PROCESS_RESPONSE, SAVE_REQUEST_INFO (SAVINGHANDLER),
EXTRACT_RAW_TEXT (TEXTHANDLER), LEMMATIZE_AND_EXTRACT_TOPONYMS
(TOPONYMHANDLER)
#1]
    #1[
    #1 ROTINA: __INIT__
    #1 FINALIDADE: INICIALIZA A CLASSE SCRAPINGHANDLER E CONFIGURA AS
DEPENDÊNCIAS NECESSÁRIAS
    #1 ENTRADAS: NOME DO DIRETÓRIO (STRING)
    #1 DEPENDENCIAS: REQUESTINGHANDLER, SAVINGHANDLER, TEXTHANDLER,
TOPONYMHANDLER, LOGGINGHANDLER
    #1 CHAMADO POR: SCRAPINGHANDLER
    #1 CHAMA: LOGGINGHANDLER.__INIT__, REQUESTINGHANDLER, SAVINGHANDLER,
TEXTHANDLER, TOPONYMHANDLER
    #1]
    #2[
    #2 PSEUDOCODIGO DE: __init__
        #2 CHAMA O CONSTRUTOR DA CLASSE PAI E INICIALIZA O DIRETÓRIO
        #2 INICIALIZA O HANDLER DE REQUISIÇÕES
        #2 INICIALIZA O HANDLER DE SALVAMENTO
        #2 INICIALIZA O HANDLER DE TEXTO
        #2 INICIALIZA O HANDLER DE TOPÔNIMOS
    #2]
    #1[
    #1 ROTINA: PROCESS_RESPONSE
    #1 FINALIDADE: PROCESSA O DICIONÁRIO DE RESPOSTA, EXTRAINDO TEXTO
BRUTO, LEMATIZANDO E EXTRAINDO TOPÔNIMOS, E SALVA AS INFORMAÇÕES
    #1 ENTRADAS: DICIONÁRIO DATA_DICT (COM CAMPOS COMO RESPONSE_STRING)
    #1 DEPENDENCIAS: TEXTHANDLER, TOPONYMHANDLER, SAVINGHANDLER
    #1 CHAMADO POR: USUÁRIO
    #1 CHAMA: EXTRACT_RAW_TEXT (TEXTHANDLER),
LEMMATIZE_AND_EXTRACT_TOPONYMS (TOPONYMHANDLER), SAVE_REQUEST_INFO
(SAVINGHANDLER)
    #1]
    #2[
    #2 PSEUDOCODIGO DE: process_response
        #2 EXTRAI O TEXTO BRUTO A PARTIR DA RESPOSTA
        #2 ADICIONA O TEXTO EXTRAÍDO AO DICIONÁRIO
        #2 LEMATIZA O TEXTO E EXTRAI OS TOPÔNIMOS
        #2 ADICIONA O TEXTO LEMATIZADO E OS TOPÔNIMOS AO DICIONÁRIO
        #2 SALVA O DICIONÁRIO PROCESSADO USANDO O SAVINGHANDLER
    #2]
```