

Nome do Arquivo
toponym_handler.py
Documentação
<pre> #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] #2[#2 PSEUDOCODIGO DE: _ADD_PATTERNS #2 DEFINE OS TERMOS RELACIONADOS A LOCAIS E SUAS ABREVIACÕES #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] #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] </pre>

Nome do Arquivo
leis_municipais_camara_sp.py
Documentação
<pre> #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] #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 </pre>

```
#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]
#2[
#2 PSEUDOCODIGO DE: BUILD_URL
#2 RETORNA A URL CONSTRUÍDA COM O ASSUNTO E NÚMERO DA PÁGINA PARA
A BUSCA
#2]
#2[
#2 PSEUDOCODIGO DE: PARSE_CONTENT
#2 TRANSFORMA O CONTEÚDO HTML EM UM OBJETO BEAUTIFULSOUP PARA
FACILITAR A EXTRAÇÃO DE DADOS
#2]
#2[
#2 PSEUDOCODIGO DE: EXTRACT_LINKS
#2 EXTRAI TODOS OS LINKS QUE CONTÊM 'DECRETO' NO CAMINHO DO HREF
#2]
#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]
#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
dataframe_handler.py
Documentação
<pre>#2[#2 PSEUDOCODIGO DE: __init__ #2 CHAMA O CONSTRUTOR DA CLASSE PAI PASSANDO O NOME DO DIRETÓRIO #2] #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] #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] #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] #2[#2 PSEUDOCODIGO DE: _extract_institution_name</pre>

```
#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]
#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]
#2[
#2 PSEUDOCODIGO DE: execute
#2 INICIA O PROCESSO DE CARREGAMENTO E PROCESSAMENTO DOS ARQUIVOS
JSON
#2]
```

Nome do Arquivo
main.py
Documentação
#2[#2 PSEUDOCODIGO DE: MAIN #2 INICIALIZA O HANDLER DE DATAFRAMES #2 EXECUTA A FUNÇÃO PRINCIPAL DO HANDLER #2]

Nome do Arquivo
legislacao_prefeitura_sp_gov_br.py
Documentação
#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] #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] #2[#2 PSEUDOCODIGO DE: BUILD_URL #2 RETORNA A URL CONSTRUÍDA COM O ASSUNTO E NÚMERO DA PÁGINA PARA A BUSCA #2] #2[#2 PSEUDOCODIGO DE: PARSE_CONTENT #2 TRANSFORMA O CONTEÚDO HTML EM UM OBJETO BEAUTIFULSOUP PARA FACILITAR A EXTRAÇÃO DE DADOS #2] #2[#2 PSEUDOCODIGO DE: EXTRACT_LINKS #2 EXTRAI TODOS OS LINKS QUE CONTÊM 'LEIS' NO CAMINHO DO HREF #2] #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]
#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
requesting_handler.py
Documentação
<pre>#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] #2[#2 PSEUDOCODIGO DE: set_method_mapping #2 DEFINE O MAPEAMENTO DE MÉTODOS HTTP 'GET' E 'POST' #2] #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] #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]</pre>

Nome do Arquivo
saving_handler.py
Documentação
<pre>#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] #2[#2 PSEUDOCODIGO DE: _create_directory #2 VERIFICA SE O DIRETÓRIO EXISTE #2 CRIA O DIRETÓRIO SE ELE NÃO EXISTIR #2] #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]</pre>

Nome do Arquivo
logs.txt
Documentação

Nome do Arquivo
alesp.py
Documentação
<pre> #2[#2 PSEUDOCODIGO DE: __INIT__ #2 INICIALIZA A CLASSE PAI E O DIRETÓRIO #2 INICIALIZA O CONJUNTO DE URLS VISITADAS #2] #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] #2[#2 PSEUDOCODIGO DE: BUILD_URL #2 CONSTROI A URL PARA BUSCA DE NORMAS COM BASE NO NÚMERO DA PÁGINA E NO ASSUNTO #2] #2[#2 PSEUDOCODIGO DE: PARSE_CONTENT #2 TRANSFORMA O CONTEÚDO HTML EM UM OBJETO BEAUTIFULSOUP PARA FACILITAR A EXTRAÇÃO DE DADOS #2] #2[#2 PSEUDOCODIGO DE: EXTRACT_LINKS #2 EXTRAI TODOS OS LINKS QUE CONTÊM 'NORMA' NO CAMINHO DO HREF #2] #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] #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] </pre>

Nome do Arquivo
logging_handler.py
Documentação
#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]
#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
text_handler.py
Documentação
<pre>#2[#2 PSEUDOCODIGO DE: __init__ #2 CHAMA O CONSTRUTOR DA CLASSE PAI E INICIALIZA O DIRETÓRIO #2 ARMAZENA O NOME DO DIRETÓRIO #2] #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] #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]</pre>

Nome do Arquivo
scraping_handler.py
Documentação
<pre>#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] #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]</pre>

