# Documentação Programa De Consulta de CNPJ

**Análise de requisitos**

**Analista: Gabriel Westerlon Pelicer**

* Condições de Uso:
  + Arquivo Excel renomeado como “CNPJ.xlxs” na mesma parta que o Executável.
  + Uma Coluna Nomeada como CNPJ na primeira linha.
  + Lista de CNJPS a baixo da coluna CNPJ.
  + Executável \_ “Alfa/dist/Consulta\_cnpj.exe”
  + Base de Dados “alfa\_transportes” Criada em servidor POStgres.
  + Tablela CAD\_Filial Criada.
  + Base de Dados em LocalHost.
  + Usuário : ‘postgres’
  + Senha : “31082”
* Diagrama de caso de uso:

Diagrama

Descrição gerada automaticamente

## IDENTIFICADOR RF001

**Requisitos:** O programa deve ter a capacidade de ler uma lista de CNPJs e os consultar na API ReceitaWS, validando se o CNPJ é valido ou não, caso ele seja invalido deve notificar o Usuário, sendo valido deve armazenar informações de Nome, Cidade e estado na tabela CAD\_FILIAL.

**História de usuário** Crie um programa em Python, que utilize a biblioteca requests para consultar os dados das filiais da Alfa Transportes listadas acima e popular automaticamente a tabela de filiais, buscando os dados na API ReceitaWS.

**Regras De negócio:**

Deve Utilizar Linguagem de programação Python com a biblioteca Requests.

Deve validar a Existência do arquivo em Excel com o CNPJ a serem consultados.

Deve Solicitar consulta para API.

Deve validar a resposta obtida pela API.

Deve notificar usuário caso o CNPJ seja invalido.

Deve validar se existe o registro na tabela, caso não exista deve inserir e caso exista deve atualizar.

# Código Fonte

import requests  
import psycopg2  
import pandas as pd  
import os  
import time  
import ctypes  
  
if os.path.exists( # valida se o Arquivo Existe na mesma pasta que o que o fonte  
 "CNPJ.xlsx"):  
  
 arquivo\_cnpj = pd.read\_excel("CNPJ.xlsx", header=0) # Le o arquivo excel que contém os cnpj  
  
else:  
 ctypes.windll.user32.MessageBoxW(0, "Arquivo excel com Cnpjs Não encontrato!"  
 , "Erro!", 0)  
 exit()  
cont = 0 # Contador  
  
for i, row in enumerate(arquivo\_cnpj.values): # estrutura de repetição para percorrer todos os cnpjs listados  
  
 cnpj = str((arquivo\_cnpj["CNPJ"][cont])) # cacputura o valor da linha e coluna CNPJ  
 cnpj = cnpj.replace('.','').replace('-','').replace('/','') # Realiza tratamento caso Usuário tenha usado caracteries  
 r = requests.get('https://www.receitaws.com.br/v1/cnpj/{}'.format(cnpj)) # realiza requisição para API  
 r.text # Capitura texto retorna pela API  
  
 while r.text == 'Too many requests, please try again later.': # Valida casos que sobre carga na API onde requer consultar mais tarde  
 time.sleep(10) # espera 10 seguntos para realizar nova consulta  
 r.close() # fecha solicitação antiga  
 r = requests.get('https://www.receitaws.com.br/v1/cnpj/{}'.format(cnpj)) # Realiza Nova consulta ate que retorne arquivo JSON  
 r.text # capitura o retono da API  
  
 arquivo = r.json() # VAriavel que armazena resulta em Json da API  
  
 if 'message' not in arquivo: # Valida casos onde a API retona uma mensagem de erro  
 nome =format(arquivo['nome'])  
 Cnpj = format(arquivo['cnpj']).replace('.','').replace('-','').replace('/','')  
 municipio = format(arquivo['municipio'])  
 UF = format(arquivo['uf'])  
  
 con = psycopg2.connect(host='localhost', database='alfa\_transportes',  
 user='postgres', password='31082') # conexão com banco  
  
 cursor = con.cursor() # cria Cursor  
  
 sql = (  
 "select \* from CAD\_FILIAL where cnpj = '" + Cnpj + "'"  
 )  
 cursor.execute(sql) # consulta se o Cnpj existe na tabela  
 result = cursor.fetchone()  
  
 if result == None : # caso não exista realiza o Insert  
 insert = (  
 "INSERT INTO CAD\_FILIAL ("  
 "cnpj"  
 ", nome"  
 ", cidade"  
 ", estado"  
 ") "   
 "VALUES ('"  
 + Cnpj +  
 "' , '" + nome +  
 "' , '" + municipio +  
 "' , '" + UF +  
 "')"  
 )  
  
 cursor.execute(insert)  
 con.commit()  
 con.close()  
 else: # caso exista realiza o Update das informações  
  
 update = (  
 "UPDATE CAD\_FILIAL "  
 "SET NOME = '" + nome +  
 "' , cidade = '" + municipio +  
 "' , estado = '" + UF +  
 "' WHERE CODIGO\_FIL = '" + str(result[0]) + "'"  
 )  
 cursor.execute(update)  
 con.commit()  
 con.close()  
  
 else: # caso de Retorno Com erro Aprenseta na tela o erro e qual foi Cnpj que apresentou o erro  
 retorno = format(arquivo['message'])  
 ctypes.windll.user32.MessageBoxW(0, " O Retorno da consulta para o CNPJ " +  
 cnpj + " Foi; --> " + retorno , "Erro!", 0)  
 cont = cont + 1 # alimenta contador  
 r.close() # Fecha requisão para API  
  
ctypes.windll.user32.MessageBoxW(0, "Registos Inseridos Com Sucesso!"  
 , "OK", 0)