

FERRAMENTA DE AUTOMAÇÃO PARA CAPTURA DE DADOS DE NFS-E

Bruno Coletti Marano¹, Kayke Matheus Galdiano da Silva Miguel¹, Fabrício Gustavo Henrique¹

¹Faculdade de Tecnologia de Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

bruno.marano@fatec.sp.gov.br, kayke.miguel@fatec.sp.gov.br,
fabricio.henrique@fatec.sp.gov.br

Resumo. *Este projeto tem como objetivo realizar a navegação web de forma automatizada em um site de consulta de nota fiscal de serviço para acessar e extrair informações alvos de arquivos PDF's por meio de uma ferramenta Python e gravar esses dados em um banco de dados, assim como também o arquivo PDF.*

Abstract. *This project aims to perform web browsing in an automated system on a service invoice consultation website to access and extract target information from PDF files using Python and save this data in a database, as well as the PDF file.*

1. Introdução

Como forma de proporcionar um adequado entendimento do processo que será realizado neste projeto, é necessário conceituar alguns pontos importantes como NFS-e e as bibliotecas envolvidas no processo de extração de dados.

2. Referencial Teórico

2.1. Definição e obrigatoriedade da emissão de NFS-e

Com o objetivo de regulamentar os tributos municipais, a Constituição Federal de 1988 concedeu a competência tributária aos municípios para instituir e realizar a cobrança do imposto ISSQN, o qual incide sobre os serviços prestados por pessoas jurídicas. Porém, foi a lei complementar 116/2003 que determinou a lista de serviços e suas determinadas alíquotas as quais seriam desde então cobradas a população brasileira (BRASIL, 2003).

A NFS-e, como arquivo digital, foi introduzida em 2012 pela Receita Federal do Brasil (RFB) em parceria com Associação Brasileira das Secretarias da Finanças das Capitais (ABRASF) pela lei complementar de número 687. Já a validade jurídica da NFS-e é assegurada pela Infraestrutura e Chaves Públicas Brasileiras (ICP Brasil). Diferentemente da NF-e, documento fiscal de produtos, a NFS-e documenta os tributos e impostos a serem pagos a partir das prestações de Serviço. Vale ressaltar que a obrigatoriedade deste documento se faz presente na maior parte dos municípios, levando em consideração que nem todos obrigam e/ou disponibilizam Web Services estruturados para transmissão dos arquivos.

Devido a disparidade de recursos entre os mais de 5000 municípios brasileiros, atualmente não há um padrão federal deste arquivo, assim como ocorre para NF-e. Desta forma, muitos municípios optam por padrões já bem desenvolvidos como Ginfes (Santos,

Maceió, Diadema) enquanto outros municípios, devido a segurança de dados e facilidade de manutenção optam por desenvolver seus próprios padrões, assim como ocorre em cidades como São Paulo ou até mesmo o município de Belém, distrito este o qual estaremos nos baseando para execução deste projeto.

2.2. Estrutura do arquivo e emissão da NFS-e

Uma nota fiscal de serviço pode ser definida como um documento fiscal utilizado para o registro das prestações de serviços, sendo gerada e armazenada eletronicamente em arquivos do tipo XML. Este documento é composto por informações relacionadas a todos os objetos pertencentes ao processo, por exemplo, prestador, tomador, serviço prestado, dados do município assim como também os de identificação do documento. (DE ANGELI NETO, 2016).

Este título pode ser emitido pelo prestador de serviço de dois modos, sendo eles: manual (direto na prefeitura) ou via Web Service. Para os dois casos citados, o prestador deve possuir cadastros na prefeitura de interesse e também no site de emissão de NFS-e da respectiva cidade. Em posse dos devidos cadastros, o emissor deverá selecionar o melhor método de emissão levando em consideração sua própria demanda.

O objetivo deste trabalho é desenvolver a automação de um serviço de notas fiscais o qual não é disponibilizado pela prefeitura de Belém-PA. Essa prefeitura disponibiliza um Web Service para realizar comunicação entre prefeitura e aplicações dos tomadores. Entre os métodos compostos por este Web Service (doravante WS), pode ser ressaltado os métodos de envio via lote WS. Porém, há um cenário em que essa prefeitura, em meio ao seu fluxo de envio e retorno do processamento do Xml, não devolve adequadamente o Xml de autorização contendo as informações de autorização (Figura 1) à aplicação transmissora. Uma possível solução para este problema seria o método de consulta de status, o qual é implementado eficazmente por diversos padrões de emissão de NFS-e e adotado por diversos municípios. Porém, essa prefeitura não possui esse processo implementado e então, se acaso este cenário ocorra, é necessário realizar a consulta manual pelo usuário para que haja a correta escrituração do documento.

Sendo assim, como forma de agilizar um processo oneroso de extração de dados de autorização de notas emitidas, foi adotada essa solução a qual será explorada neste trabalho.

 <p>PREFEITURA MUNICIPAL DE BELÉM SECRETARIA MUNICIPAL DE FINANÇAS NOTA FISCAL DE SERVIÇOS ELETRÔNICA - NFS-e</p>	Número da Nota 00019513			
	Data e Hora da Emissão 29/01/2021 00:00:00			
	Código de Verificação 001A.C7FF.6EC7.FF1D.87EF.8ADA.E5F4.E9F8			
PRESTADOR DE SERVIÇOS				
Nome / Razão Social: ASSOCIACAO OBJETIVO DE ENSINO SUPERIOR - ASSOBEs CPF / CNPJ: 01.711.282/0005-21 Inscrição 1606540 Endereço: DOS MUNDURUCUS BAIRRO CREMAÇÃO CEP 66073-000 Município: BELEM UF: PA Email: fiscalpaulista@unip.br Telefone: (91) 3249-8290				
TOMADOR DE SERVIÇOS				
Nome / Razão Social: VITOR FARIAS BARBOSA CPF/CNPJ: 018.228.582-08 Endereço: AVENIDA GOVERNADOR HELIO DA MOTA GUEIROS BAIRRO QUARENTA HORAS CEP 67120- Município: ANANINDEUA UF: PA Email: barbosavitor188@gmail.com				
DESCRIÇÃO DOS SERVIÇOS				
SERVIÇOS EDUCACIONAIS REFERENTE A COBRANCA 28705267				
Tributável	Serviços	QTDe	Val. Uni. (R\$)	Val. Total
SIM	SERVICO EDUCACIONAL	1	0,01	0,01

Figura 1. Exemplo de PDF de uma nota fiscal de serviço (NFS-e)
Fonte: (Prefeitura de Belém, padrão SIAT - 2021)

3. Materiais e métodos

Para desenvolver esta solução de forma prática, o método escolhido para este projeto foi o Web Scraping. Este método é fundamentado na extração de dados da web utilizando-se de técnicas de programação. O método de scraping pode ser definido como a simulação da ação humana navegando e manipulando dados estáticos e dinâmicos presentes na estrutura HTML.

3.1. Ferramentas

Bibliotecas Python

Para realizar cada etapa do processo de captura de dados, algumas bibliotecas Python foram selecionadas de forma pontual com a proposta de unificar suas funcionalidades em um só sistema complexo. Dentre elas, a biblioteca Selenium.

Selenium

Selenium é uma biblioteca disponibilizada a diversas linguagens de programação (assim como Python) para realizar testes automatizados de front-end utilizando-se de uma API chamada de WebDriver que se conecta diretamente ao browser, permitindo navegar pelas páginas do navegador, assim como também controlar as propriedades e funções do arquivo DOM completo da página HTML (DevMedia, 2016)

O método de scraping neste projeto ocorrerá em dois momentos, os quais definiremos como anterior e posterior. O momento anterior será a etapa de navegação no site. Para este, realizamos todo o mapeamento das tags HTML do site. Possuindo todas as tags mapeadas, conseguimos realizar a navegação entre as páginas de interesse. Já o momento posterior se trata da manipulação e extração de dados do PDF já baixado do site de Belém.

Como materiais de processo de criação dessa ferramenta, utilizamos o editor de textos VSCode, o qual proporciona uma boa visibilidade do código, assim como também proporciona uma praticidade fundamental na escrita devido a suas diversas funcionalidades e pacotes que foram criados para serem utilizados nesta ferramenta. Além do editor de texto, podemos citar como material, o próprio site de consulta e emissão de NFS-e de Belém: <http://siat.belem.pa.gov.br:8081/aceso/pages/geral/homeNfe.jsf>.

3.2. Requisitos Funcionais e Não Funcionais

Nesta seção será demonstrada as funcionalidades primárias e secundárias que o sistema fornecerá ao usuário durante a execução dos processos do sistema.

Requisitos Funcionais:

- a) O sistema deverá apresentar uma interface intuitiva, para que seja possível de o usuário assimilar da melhor forma a sua utilização.
- b) Deverá haver uma interface de login e senha para que haja segurança na sua utilização.
- c) O sistema deverá fornecer filtros de pesquisa de RPS, NFS-E, CNPJ e Razão social, filtros estes que são também apresentados na interface web da prefeitura.
- d) Após a escolha dos filtros de interesse, o programa deverá ser capaz de identificar as estruturas de HTML pré-determinadas a fim de que seja possível caminhar por todo o site de forma automática e sem erros.
- e) O sistema deverá apresentar um sistema de Logs eficiente que registrará todas as informações importantes durante a execução dos processos.
- f) Durante a execução do software, em certo momento o sistema deve criar uma pasta temporária para armazenamento de PDF's baixados da Prefeitura
- g) Ao final da execução, o sistema deve ser capaz de apagar todos os PDF's já processados e apagar o diretório temporário onde foram armazenados durante sua execução.
- h) Após a exclusão dos dados já processados, o sistema deverá gerar um excel com todos os dados raspados dos PDF's baixados da prefeitura com o objetivo de realizar o upload no banco de dados.

Requisitos não-funcionais:

- a) O sistema deverá executar de forma contínua e sem travamentos em todos os tipos de máquinas, tendo elas discos em HD ou SSD.
- b) Se acaso houver instabilidades na rede do usuário, o sistema deverá manter por certo tempo pré-determinado os seus processos.
- c) O sistema deverá apresentar um mecanismo de criptografia para autenticação do usuário no momento do login
- d) Todos os dados modificados no sistema devem ter sua integridade garantida a fim de eliminar qualquer inconsistência existente no banco de dados.

3.3. Diagrama de Caso de Uso

Como é possível verificar nesse diagrama, o usuário irá colocar as informações necessárias na interface (RPS, data, nome) e confirmar. Em seguida, o programa começará o Scraping, onde irá percorrer pelas páginas, baixar o PDF, extrair informações do mesmo e inserir em um arquivo csv. Opcionalmente, será possível armazenar essas informações em um banco de dados.

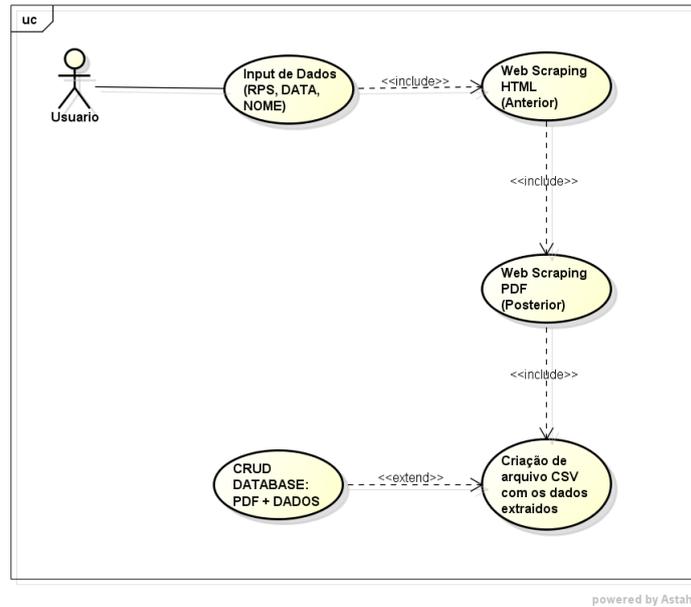


Figura 2. UML mostrando o funcionamento do sistema.
 Fonte: (Autoria própria, 2022)

4. Resultados

Este projeto está em fase de desenvolvimento. Algumas telas e funcionalidades já se encontram prontas. Dentre elas, a tela de Login, a qual já possui conexão com o Banco de Dados PostgreSQL

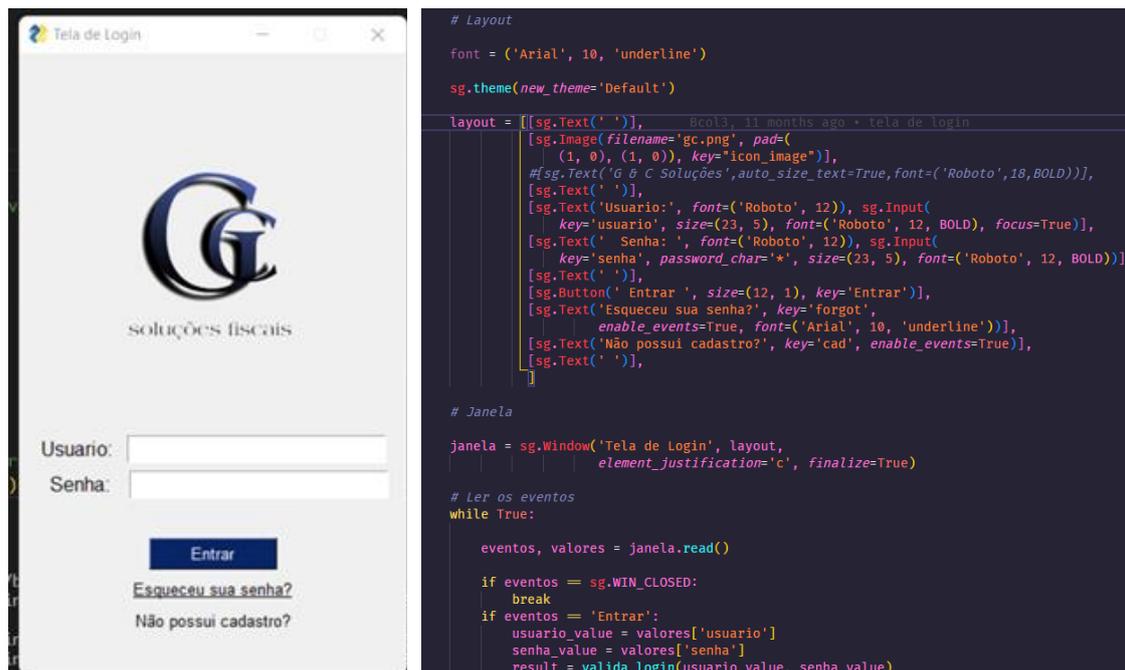


Figura 3. Tela de login do sistema Vs. Código da tela de login
Fonte: (Autoria própria, 2021)

Assim como verificado na imagem acima, após a validação do login, haverá a execução da aplicação. Segue a tela principal onde é visualizada os campos de INPUT de dados.

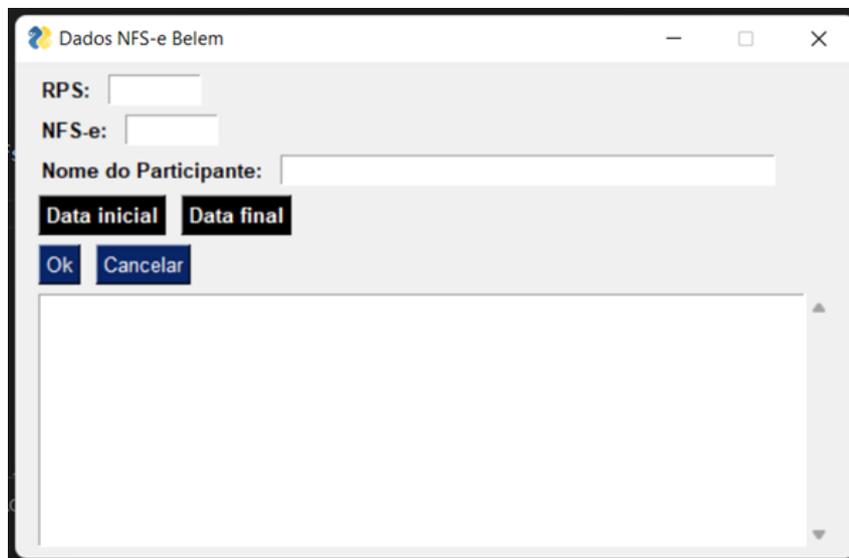


Figura 4. Tela principal do sistema.
Fonte: (Autoria própria, 2021)

```

def tela_app():
    sg.theme(new_theme='DefaultNoMoreNagging')

    layout = [
        [sg.Text("RPS:", font=('Arial', 10, 'bold'), text_color=(
            'black')), sg.InputText(size=(8, 0), key='rps'), ],
        [sg.Text("NFS-e:", font=('Arial', 10, 'bold'),
            text_color=('black')), sg.InputText(size=(8, 0), key='nfs')],
        [sg.Text("CNPJ/CPF do tomador:", font=('Arial', 10, 'bold'),
            text_color=('black')), sg.InputText(key='participante')],
        [sg.CalendarButton("Data inicial", target='_CALENDARI_', pad=None, font=('Arial', 10, 'bold'),
            button_color=('white', 'black'), key='_CALENDARI_', format='%d/%m/%Y'),
            sg.CalendarButton("Data final", target='_CALENDARF_', pad=None, font=('Arial', 10, 'bold'),
            button_color=('white', 'black'), key='_CALENDARF_', format='%d/%m/%Y')],
        [sg.Button("Ok", key='Ok'), sg.Button("Cancelar", key='Cancelar')],
        [sg.Output(size=(70, 10))]
    ]

    return sg.Window('App', layout=layout, finalize=True)

```

Figura 5. Código da tela principal do sistema.

Fonte: (Autoria própria, 2021)

Com os dados de interesse inseridos nos devidos campos, o método de scraping será ativado, e tanto o download dos PDF's quanto a extração dos dados do PDF serão realizadas de forma automática.

Após a conclusão do processo, os dados serão disponibilizados em uma planilha Csv contendo os campos a serem atualizados no banco de dados. Dentre os campos, podemos verificar os dados de autorização de um documento fiscal, que são: código de verificação, número de autorização, data de emissão e rps.

	A	B	C	D
1	codigo verificacao	numero autorizacao	data emissao	rps
2	8A36.E372.1160.5253.C0C2.EBA9.3E1B.44D1	41607	16/08/2022 00:00	55552
3	339B.C876.4FAE.312F.C878.355D.7E75.1099	41598	16/08/2022 00:00	55542
4	F29F.488C.A93F.6BF4.3972.7F69.F6F3.F12A	41507	16/08/2022 00:00	55451
5	C478.C621.1B2B.F6AA.42C2.9032.A41A.108C	41597	16/08/2022 00:00	55541
6	7419.2FEC.9FAA.826F.0D6D.FB15.3D59.A5FE	41596	16/08/2022 00:00	55540
7	E8B6.BC20.8CB0.C58D.9EDF.4253.A5C6.2A1D	41595	16/08/2022 00:00	55539
8	EBB3.E2E5.1B3A.6967.A949.A7DB.91E4.5E62	41594	16/08/2022 00:00	55529
9	D95F.7FDF.5890.2B55.A724.F9EB.2EB1.FCD4	41593	16/08/2022 00:00	55538
10	2A6E.853E.3A57.306A.B436.B736.588A.AD1D	41592	16/08/2022 00:00	55537
11	29F7.D68D.94D2.0969.454E.6433.DF2A.A0A2	41591	16/08/2022 00:00	55536
12	C37A.FCD1.3341.9599.99D3.33DD.A2C7.3C8E	41590	16/08/2022 00:00	55535
13	4930.CCE1.01F0.2DF4.C139.E5BB.5A48.25E5	41589	16/08/2022 00:00	55534
14	338E.8D70.D72F.D04C.1796.6F29.C90E.C326	41606	16/08/2022 00:00	55551
15	3948.3A34.0553.4B4E.A6AF.32A6.22E4.716F	41588	16/08/2022 00:00	55533
16	BDB3.01DE.8D8F.DF6B.5A96.D21C.FFEE.E985	41587	16/08/2022 00:00	55532
17	72DC.8C68.260B.9FA8.E96B.334A.4E62.2033	41586	16/08/2022 00:00	55531
18	EDB9.5231.04E2.EB03.7D9E.44A8.83F8.EBE9	41585	16/08/2022 00:00	55530
19	9306.BC85.05A8.9180.3408.CEFC.CD4D.4EA2	41584	16/08/2022 00:00	55528
20	E76F.CE51.875F.9EF6.27ED.309D.C127.B087	41583	16/08/2022 00:00	55527
21	82B6.04E6.2173.7F8E.FF7E.1B45.2672.3DF6	41582	16/08/2022 00:00	55526
22	6A29.6657.5B42.AF1B.C748.022F.C0D9.448A	41581	16/08/2022 00:00	55525
23	A63F.E259.AC30.BDE8.FF7C.6982.A2AD.D7AE	41580	16/08/2022 00:00	55524
24	2066.D89E.8DDF.0B7F.22D9.269D.2D8B.7B4D	41579	16/08/2022 00:00	55523
25	8DB3.6195.545E.EB7D.481B.690A.2980.4536	41605	16/08/2022 00:00	55550
26	F744.9A71.B416.2427.5BD3.0929.4AB1.9211	41578	16/08/2022 00:00	55522
27	8E30.5A9C.F280.48B7.C357.E65C.A759.D630	41577	16/08/2022 00:00	55521
28	0850.2125.C286.A58E.B071.C677.B3A5.6761	41576	16/08/2022 00:00	55520
29	9A4A.B495.977D.DE0C.5DC8.A962.28BF.BF95	41575	16/08/2022 00:00	55519

Figura 6. Planilha contendo dados extraídos dos PDF's baixados.

Fonte: (Autoria própria, 2021)

```
import csv

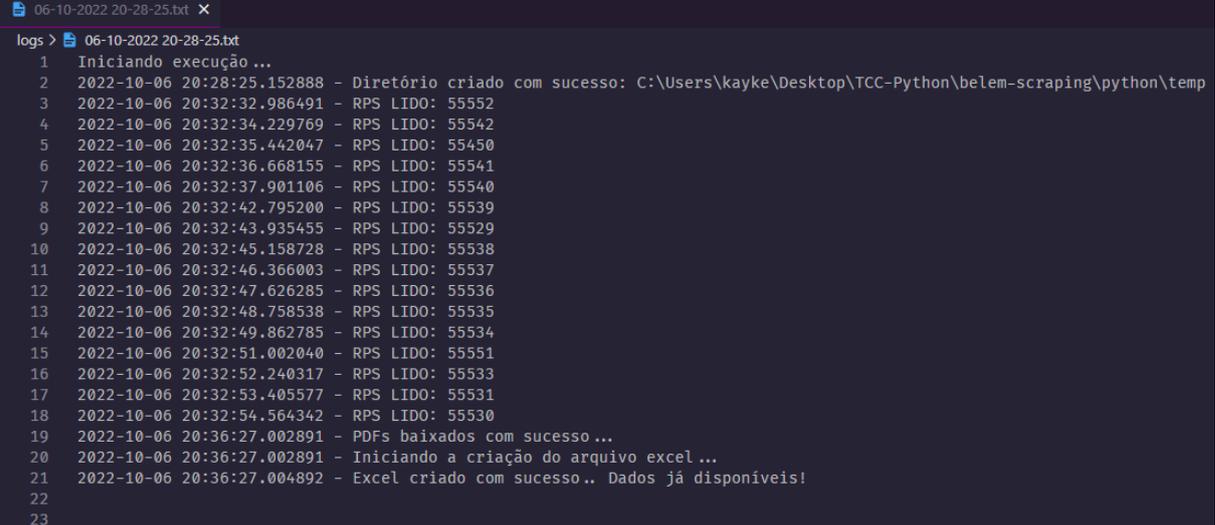
def cria_arq_excel(dados):
    header = ['codigo verificacao', 'numero autorizacao', 'data emissao', 'rps']
    with open('arquivo_excel.csv', mode='w', newline='', encoding='utf8') as arquivo_excel:
        escreve = csv.writer(arquivo_excel, delimiter=';', quotechar='"', quoting=csv.QUOTE_MINIMAL)
        escreve.writerow(header)
        escreve.writerows(dados)
```

Figura 7. Código de criação da planilha contendo os dados extraídos via Web Scraping

Fonte: (Autoria própria, 2022)

Todas as etapas de geração de Web Scraping foram “logadas” de forma padronizada a fim de termos um determinado controle da sessão sendo executada. Abaixo podemos

verificar um exemplo onde é possível observar que em cada execução, o sistema gera um arquivo novo com o nome contendo a data atual e os dados sendo gerados de acordo com o tempo de execução da sessão.



```
06-10-2022 20-28-25.txt X
logs > 06-10-2022 20-28-25.txt
1 Iniciando execução ...
2 2022-10-06 20:28:25.152888 - Diretório criado com sucesso: C:\Users\kayke\Desktop\TCC-Python\belem-scraping\python\temp
3 2022-10-06 20:32:32.986491 - RPS LIDO: 55552
4 2022-10-06 20:32:34.229769 - RPS LIDO: 55542
5 2022-10-06 20:32:35.442047 - RPS LIDO: 55450
6 2022-10-06 20:32:36.668155 - RPS LIDO: 55541
7 2022-10-06 20:32:37.901106 - RPS LIDO: 55540
8 2022-10-06 20:32:42.795200 - RPS LIDO: 55539
9 2022-10-06 20:32:43.935455 - RPS LIDO: 55529
10 2022-10-06 20:32:45.158728 - RPS LIDO: 55538
11 2022-10-06 20:32:46.366003 - RPS LIDO: 55537
12 2022-10-06 20:32:47.626285 - RPS LIDO: 55536
13 2022-10-06 20:32:48.758538 - RPS LIDO: 55535
14 2022-10-06 20:32:49.862785 - RPS LIDO: 55534
15 2022-10-06 20:32:51.002040 - RPS LIDO: 55551
16 2022-10-06 20:32:52.240317 - RPS LIDO: 55533
17 2022-10-06 20:32:53.405577 - RPS LIDO: 55531
18 2022-10-06 20:32:54.564342 - RPS LIDO: 55530
19 2022-10-06 20:36:27.002891 - PDFs baixados com sucesso ...
20 2022-10-06 20:36:27.002891 - Iniciando a criação do arquivo excel ...
21 2022-10-06 20:36:27.004892 - Excel criado com sucesso.. Dados já disponíveis!
22
23
```

Figura 8. Logs gerados durante a execução de uma sessão
Fonte: (Autoria própria, 2022)

Cabe informar que apesar de já estar em fase de testes, ainda há diversos ajustes a serem feitos, os quais já se encontram em nosso roadmap de desenvolvimento.

5. Considerações Finais

Neste projeto foi possível aplicar técnicas novas de desenvolvimento Web as quais não havíamos executado na graduação. Durante o desenvolvimento, foi possível superar dificuldades, como a curva de aprendizagem da linguagem Python, como utilizar a biblioteca Selenium, entre outras as quais serviram de conhecimento que levaremos para nossa carreira como desenvolvedores. Podemos concluir após todo o processo que a ferramenta do projeto se demonstrou extremamente útil, levando em consideração que o processo automatizado levou em torno de 5 minutos para processar 140 documentos. Em condições normais, segundo a experiência vivida, esta rotina manual levava em torno de 3 horas. Como próximas etapas, realizaremos o aperfeiçoamento das telas desenvolvidas com o objetivo de melhorar a experiência do usuário e inserir a ferramenta no mercado como mais uma solução automatizada e útil na rotina dos usuários que se deparam com o problema citado neste projeto.

6. Referências

BRASIL (2003). Presidência da República. Casa Civil. Subchefia para Assuntos Jurídicos. LEI COMPLEMENTAR Nº 116, DE 31 DE JULHO DE 2003. Brasília, DF.

DE ANGELI NETO H.; MARTINEZ, A. L. (2016) Nota fiscal de serviços eletrônica: Uma análise dos impactos na arrecadação em municípios brasileiros. Revista de Contabilidade e Organizações, [S. I.], v. 10, n. 26, p. 49-62, 2016. DOI: 10.11606/rco.v10i26.107117. Disponível em: <<https://www.revistas.usp.br/rco/article/view/107117/>>. Acesso em: 11 mar. 2021.

LIMA, C. E. N. M. (2015) O uso da nota fiscal de serviço no combate à evasão do imposto sobre serviços de qualquer natureza no município de Fortaleza/CE: Uma análise nas atividades de monitoramento, fiscalização e arrecadação. Orientador: Prof. Ms. Eduardo Rodrigues Linhares. 2015. TCC (Graduação) - Curso de Ciências Contábeis, Departamento de Contabilidade, Universidade Federal do Ceará, Repositório UFC, 2015. Disponível em: https://repositorio.ufc.br/bitstream/riufc/30842/1/2015_tcc_cenmlima.pdf. acesso em: 1 set. 2022.

MUTHUKADAN, BAIJU. (2018) Selenium with Python. Disponível em: <<https://selenium-python.readthedocs.io/>> Acesso em: 04 de abril de 2022