FINANTO

ANTONIO LUIS PEREIRA CANDIOTO

FINANTO CONVERTER

PIRASSUNUNGA

2022

SUMÁRIO

1. INTRODUÇÃO

A aplicação conta com o objetivo de facilitar a conversão de tipo de arquivos que possuem tabelas como conteúdo. Com o foco em arquivos Excel, existe a possibilidade de paginá-los e retorná-los em um ou mais arquivos.

Além de seu objetivo principal, a possibilidade de tratamento dessas colunas também existe, incluindo filtros e formatação dos valores, todos apontados pelo usuário.

2. RECURSOS UTILIZADOS

2.1. **Visual Studio Code:** Editor de código simples que suporta várias linguagens de programação, é customizável e repleto de extensões que facilitam o desenvolvimento e construção dos projetos.

2.2. **Python:** Linguagem de programação predominante em todo o projeto. É versátil e possui uma sintaxe sem complexidade, inclui também bibliotecas nativas que foram essenciais para o bom funcionamento da aplicação, como:

2.2.1. **os, shutil e io**: são bibliotecas que possibilitam a manipulação de pastas e arquivos trabalhados no projeto.

2.2.2. **Base64:** auxilia na segurança do sistema de autenticação ajudando a converter e desconverter a base dos valores.

2.2.3. **Typing:** possui atributos como Union, List e Optional que ajudam na forma de tratar os valores recebidos pelo usuário.

2.2.4. **re:** possibita o uso de expressões regulares (regex).

2.3. **FastAPI**: *framework web* que possibilita a construção da aplicação de forma rápida e simples, possuindo uma vasta quantidade de recursos para a entrada e saída de dados, além de modelos e funções para segurança, tratamento de erros e muitas outras opções.

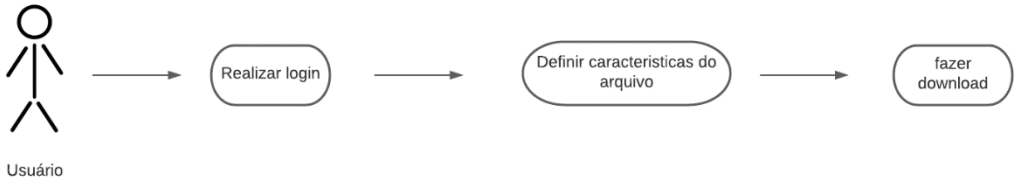
2.4. **Uvicorn**: é uma Interface de Porta de Entrada do Servidor Web (ASGI) para auxiliar a execução da aplicação.

2.5. **Pandas:** biblioteca essencial em lidar com dados através de *dataframes* (quadros de dados) e séries de dados. Na aplicação, é o que possibilita a leitura e tratamento dos dados tabulados dos arquivos. Tratando e manipulando os valores de forma simples e possibilitando o retorno no formato de arquivo desejado.

2.6. **Pydantic:** utilizada para construção dos modelos através de um modelo base, sendo uma ótima biblioteca para validar os dados recebidos de acordo com o tipo que foi atribuído na classe.

3. ESTRUTURA DE ROTAS

3.1. Funcionamento do sistema:

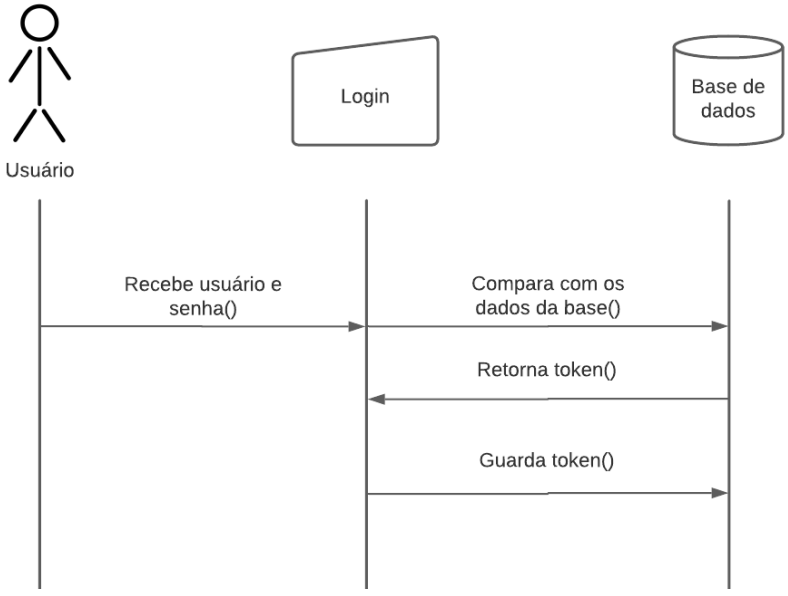


3.2. Rotas

3.2.1. Rota de Login:

Na rota de login, serão informados usuário e senha do sistema. Não existe opção de cadastro, ambos os valores são únicos e definidos pelo desenvolvedor.

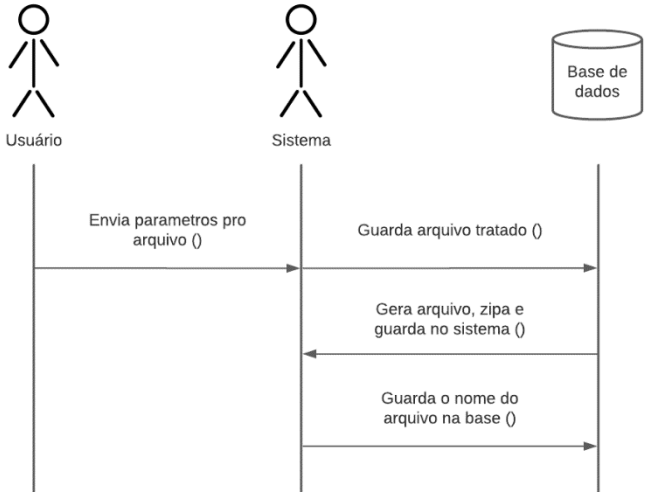
As informações serão recebidas em base64, convertidas e serão comparadas com os dados do sistema e, se corresponderem, um *token* será gerado, retornado e guardado na base de dados, senão, uma mensagem de erro 401 será retornada.



3.2.2. Rota de Conversão:

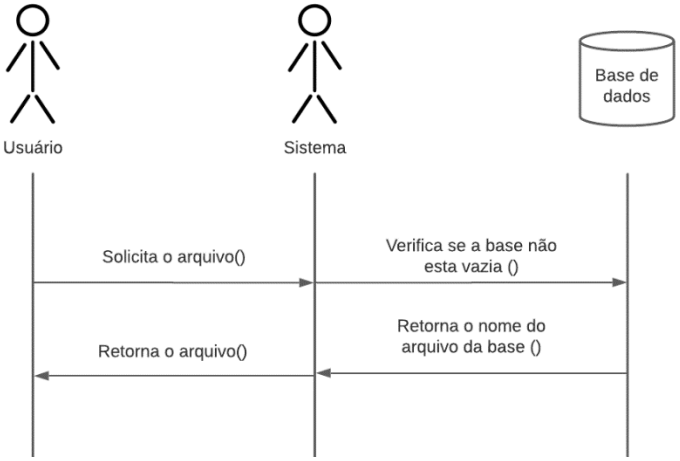
Na hora da conversão, como informações essenciais teremos o *link* ou arquivo, a informação sobre qual será o formato retornado, o nome do arquivo retornado, forma a qual a tabela será paginada e, se paginada em mais de uma página ou arquivo, quantas linhas cada página terá.

Como dados opcionais teremos os de formatação da tabela, como mudança de caixa, expressões regulares, corte de caracteres, além de também opções de filtragem, ordenação e agrupamento.

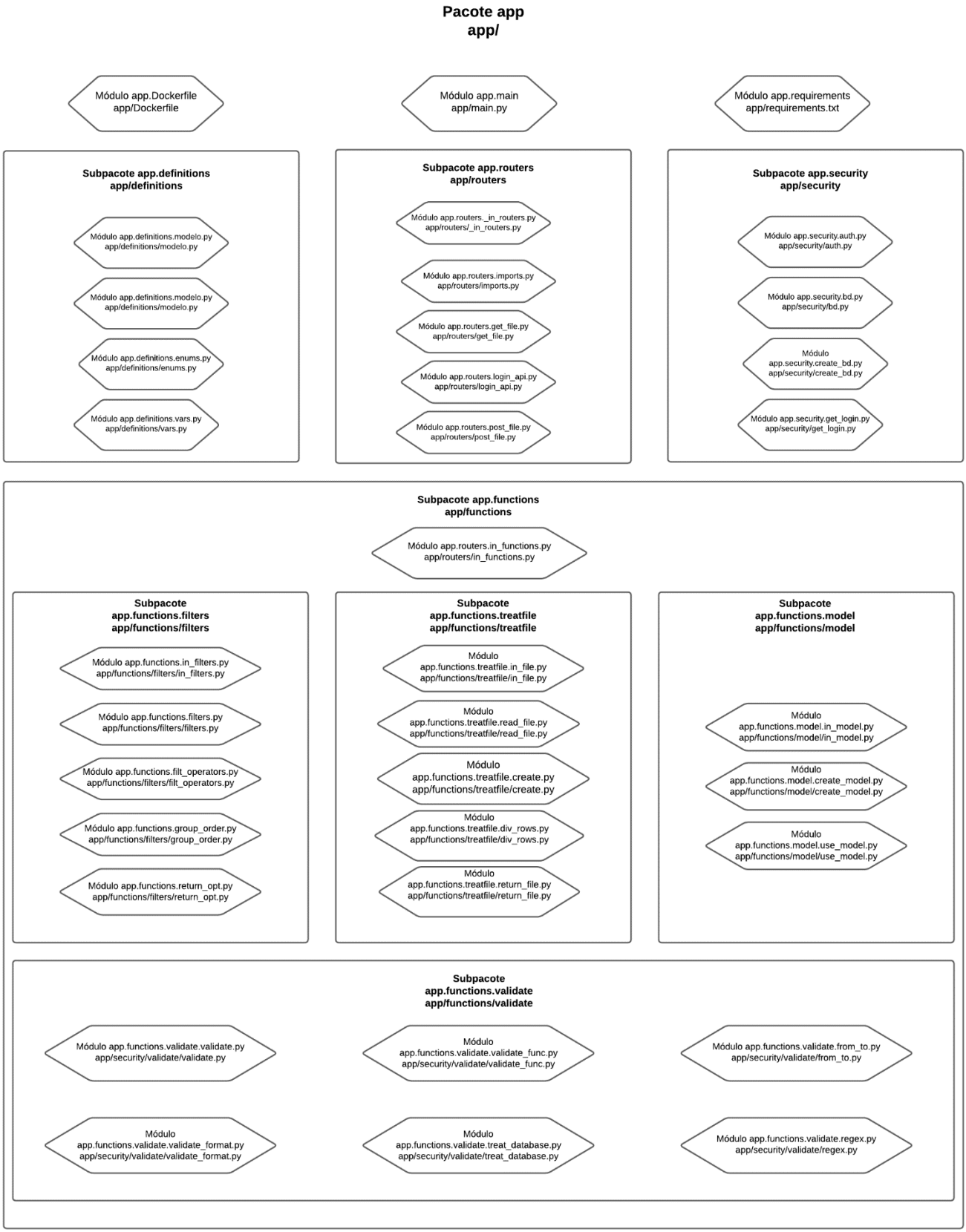


3.2.3. Rota de *download*:

Nessa rota será verificado se a base de dados está vazia, se sim, um erro será retornado, senão, o nome do arquivo armazenado será retornado ao sistema, que retornará ao usuário o arquivo correspondente que estará dentro da pasta do projeto.



4. ESTRUTURA DE ARQUIVOS

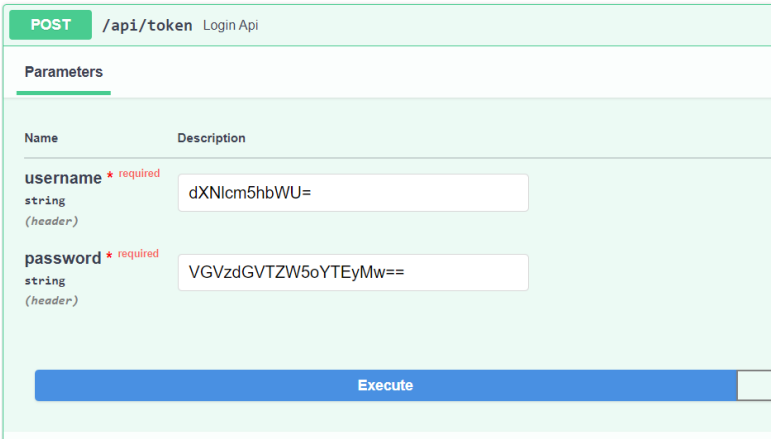


5. EXEMPLO DE EXECUÇÃO ATRAVÉS DO UVICORN

Podemos testar o sistema utilizando a seguinte tabela de um arquivo xlsx (Excel):

	A	B	C	D	E	F
1		**indice**	**Letra**	**Classificação**		
2	**0**	1	a	vogal		
3	**1**	2	b	consoante		
4	**2**	3	c	consoante		
5	**3**	4	d	consoante		
6	**4**	5	e	vogal		
7	**5**	6	f	consoante		
8	**6**	7	g	consoante		
9	**7**	8	h	consoante		
10	**8**	9	i	vogal		
11	**9**	10	j	consoante		
12	**10**	11	k	consoante		
13	**11**	12	l	consoante		
14	**12**	13	m	consoante		
15	**13**	14	n	consoante		
16	**14**	15	o	vogal		
17	**15**	16	p	consoante		
18	**16**	17	q	consoante		
19	**17**	18	r	consoante		
20	**18**	19	s	consoante		
21	**19**	20	t	consoante		
22	**20**	21	u	vogal		
23	**21**	22	v	consoante		
24	**22**	23	w	consoante		
25	**23**	24	x	consoante		
26	**24**	25	y	consoante		
27	**25**	26	z	consoante		

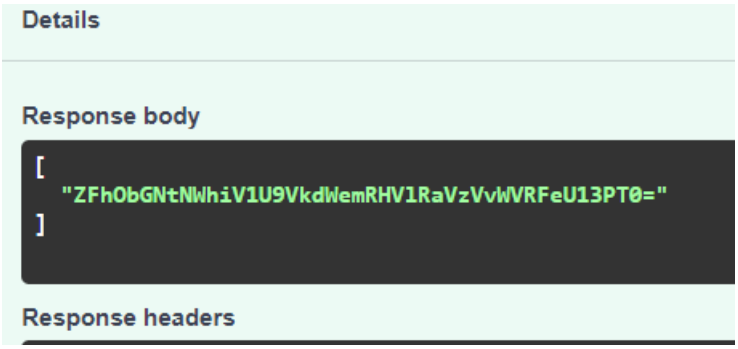
Na rota de postagem “/api/token”, iremos inserir o valor “username” no usuário e “TesteSenha123” na senha, mas em base64, pois será o formato do qual o *front-end* enviará para o *back-end*.



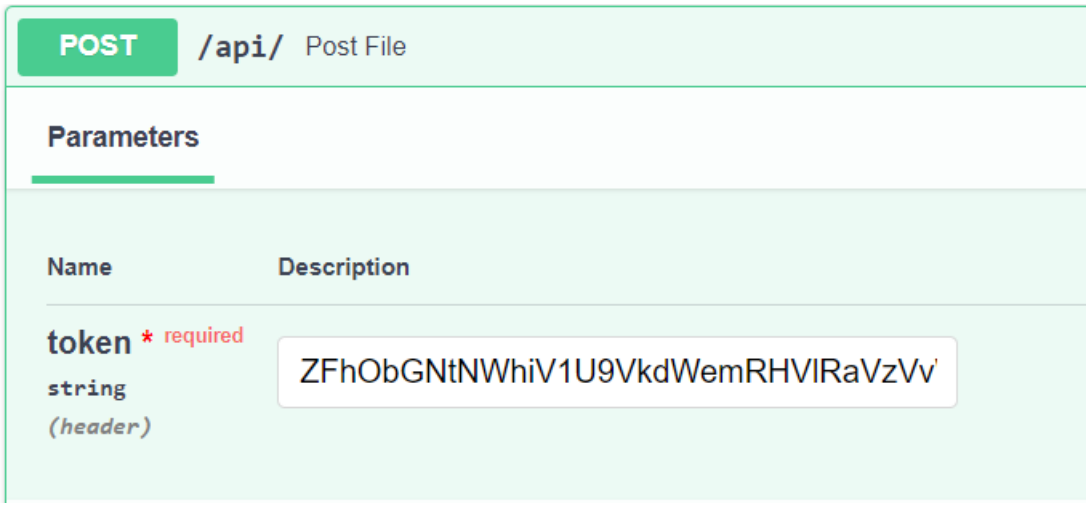
Usuário e senha definidos no sistema:

```
class Login(str):
    username = "username"
    password = "TesteSenha123"
```

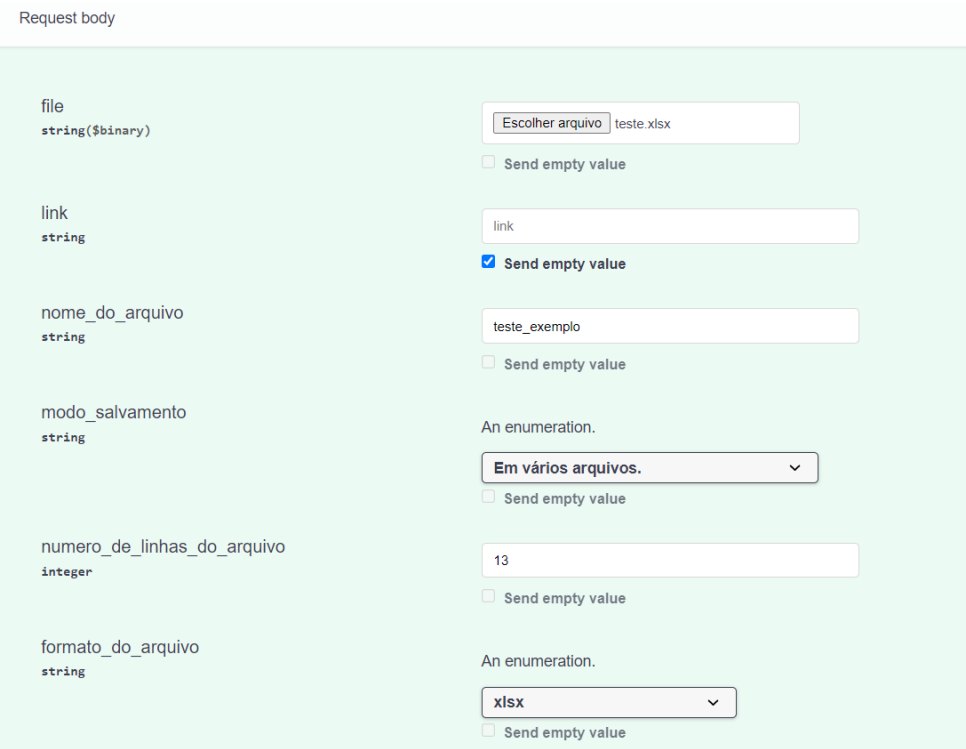
Quando correspondentes, um *token* será retornado, não ao usuário, mas ao sistema, que irá capturar o *token* e utilizá-lo na próxima rota de postagem.



Na seguinte rota, já encontramos o campo obrigatório do *token*, esse campo será o que definirá se o usuário tem ou não permissão pra utilizar a aplicação, ou seja, se o usuário está logado no sistema.

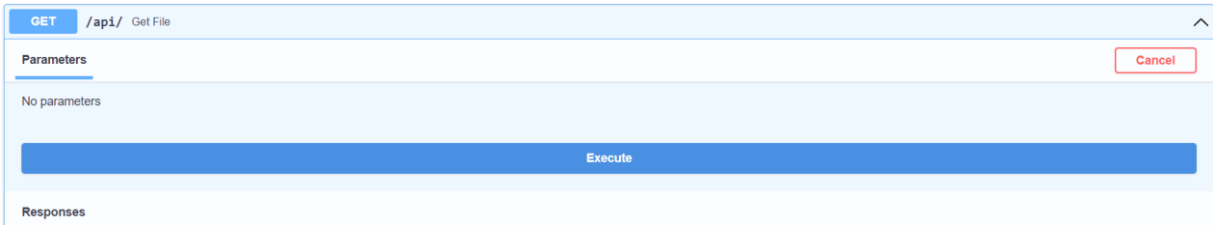


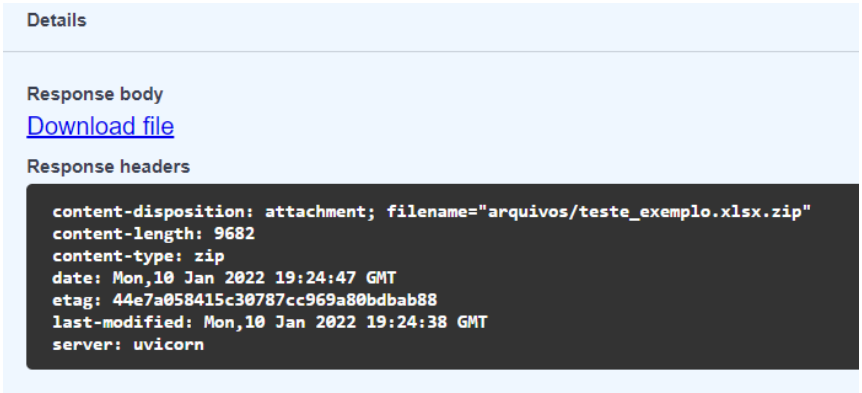
Agora iniciamos o teste do funcionamento da aplicação, onde serão inseridos os parâmetros para tratamento do arquivo desejado. Como principal, teremos o campo “file”, de arquivo, o “link”, caso o usuário queira inserir o *link* de algum *site* para leitura de formato html, o campo “nome_do_arquivo”, que terá o nome do arquivo, “modo_salvamento”, que terão as opções “Em uma página em um só arquivo.”, “Em várias páginas em um só arquivo.” e “Em vários arquivos.”, o campo “numero_de_linhas_do_arquivo”, que trabalhará em conjunto com o campo anterior e, por fim, o “formato_do_arquivo” que terão opções de “xlsx”, “json e “csv”.



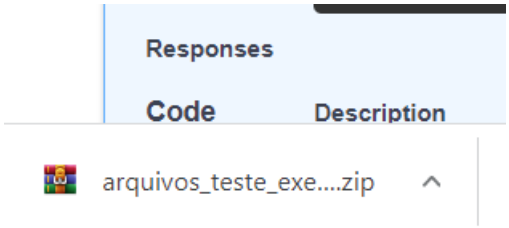
Se executarmos dessa maneira e se não ocorrer nenhum erro nos parâmetros (seja por parâmetros vazios ou inválidos), o(s) arquivo(s) será(ão) criado(s) e os nomes da pasta padrão de armazenamento do sistema e do arquivo compactado serão retornados e armazenados na base de dados.

A próxima e última rota pegará o nome do arquivo que foi armazenado na base de dados gerada na ultima rota e retornará o arquivo com nome correspondente.

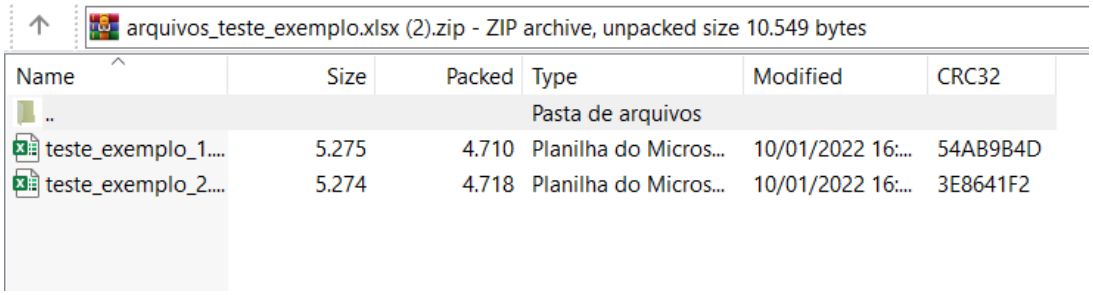




Ao clicar no botão de *download*, o arquivo será baixado e armazenado na pasta padrão de *download* do dispositivo.



Como foram selecionados os parâmetros “Em vários arquivos” e com 13 linhas, o arquivo compactado retornou dois arquivos, contendo cada um uma parte da tabela original.



Esses serão os conteúdos dos arquivos finais:

	A	B	C	D	E
1		**indice**	**Letra**	**Classificação**	
2	**0**	1	a	vogal	
3	**1**	2	b	consoante	
4	**2**	3	c	consoante	
5	**3**	4	d	consoante	
6	**4**	5	e	vogal	
7	**5**	6	f	consoante	
8	**6**	7	g	consoante	
9	**7**	8	h	consoante	
10	**8**	9	i	vogal	
11	**9**	10	j	consoante	
12	**10**	11	k	consoante	
13	**11**	12	l	consoante	
14	**12**	13	m	consoante	
15					

	A	B	C	D	E
1		**indice**	**Letra**	**Classificação**	
2	**13**	14	n	consoante	
3	**14**	15	o	vogal	
4	**15**	16	p	consoante	
5	**16**	17	q	consoante	
6	**17**	18	r	consoante	
7	**18**	19	s	consoante	
8	**19**	20	t	consoante	
9	**20**	21	u	vogal	
10	**21**	22	v	consoante	
11	**22**	23	w	consoante	
12	**23**	24	x	consoante	
13	**24**	25	y	consoante	
14	**25**	26	z	consoante	
15					

6. CONCLUSÃO

Como pudemos ver, a aplicação é simples e trabalha com o básico dos recursos utilizados, cumprindo seu objetivo de praticidade e utilidade, mas ainda podendo conter alterações de acordo com o desenvolvimento do *front-end*, de futuras necessidades que surgirem e a aplicação consiga resolver e também de atualizações em suas bibliotecas que abram portas para novas possibilidades.

7. REFERÊNCIAS BIBLIOGRÁFICAS

Python documentation, 2001. Disponível em: https://docs.python.org/3/. Acesso em 11 de janeiro de 2022.

Pandas documentation, 2008. Disponível em: https://pandas.pydata.org/docs/index.html. Acesso em 11 de janeiro de 2022.

COLVIN, Samuel. Pydantic documentation, 2017. Disponível em: https://pydantic-docs.helpmanual.io/. Acesso em 11 de janeiro de 2022.

TIANGOLO, Sebastián. FastAPI, 2018. Disponível em: https://fastapi.tiangolo.com/. Acesso em: 11 de janeiro de 2022.

ENCODE. Uvicorn documentation, 2020. Disponível em: https://www.uvicorn.org/. Acesso em 11 de janeiro de 2022.