



ALUNO: Emanuelle Neves do Nascimento	MATRÍCULA 2015672
AVALIADOR: Bruno Lopes	

Introdução

Após entender como a estrutura do servidor funcionaria basicamente, comecei a esboçar estratégias de como entregar para o navegador o esperado (uma página web). Cheguei a uma conclusão que poderia fazer essa entrega através da biblioteca python chamada [os](#) onde o proposito dessa biblioteca é buscar informações do sistema. Sabendo como fazer, vamos para a execução.

Andamento

Depois de decidir o que mostraria no exemplo, comecei a desenvolver a página HTML estilizando com CSS, para facilitar a estilização e não perder tanto tempo, utilizei o framework Bootstrap.

A página consiste em uma barra de menu, conteúdo, um formulário e um rodapé simples que servira para o exemplo de entrega do servidor web.

Depois de desenvolver a página web, é hora de pedir ao servidor para enviá-la ao browser. Sendo assim, criei uma classe no webserver.py chamada pageWeb onde definiria o diretório do projeto com os CSS, HTML e imagens utilizadas na página sendo possível agora o servidor ler a pasta com os arquivos.

Ficou assim

```
class pageWeb:

    os.chdir('src')

    try:
        print("Conexão iniciada na porta %i" % PORT)
        http.serve_forever()
    except KeyboardInterrupt:
        pass
    except socket.error as error:
        print("Algo deu errado!", error)

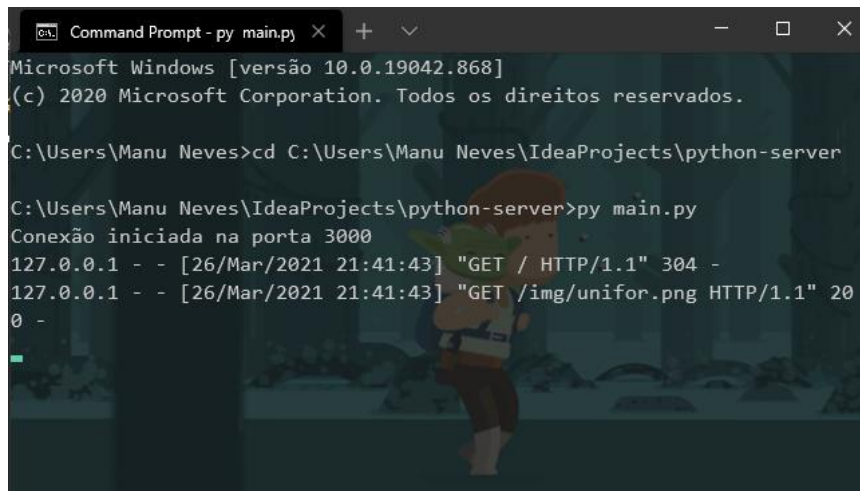
    http.server_close()
```

Uma vez feito isso, agora precisaria passar para classe HttpResponse dentro da função processResponse a classe criada para que o servidor receba os arquivos e envia para a porta selecionada.

Agora, ao inserir o endereço <http://localhost:3000/> no browser a página HTML dentro de src será renderizada. Estou definindo a porta como variável global para conseguir utiliza-la em toda a parte do código e também estou utilizando o método `os.getenv()` para nos retornar os valores das requisições ao iniciar o servidor no terminal.

Dessa forma temos uma String com o valor do endereço IP, a data time da conexão e o tipo de requisição (GET/POST) no caso abaixo, foram realizadas somente requisições GET.

Dessa forma



```
Command Prompt - py main.py
Microsoft Windows [versão 10.0.19042.868]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Manu Neves>cd C:\Users\Manu Neves\IdeaProjects\python-server

C:\Users\Manu Neves\IdeaProjects\python-server>py main.py
Conexão iniciada na porta 3000
127.0.0.1 - - [26/Mar/2021 21:41:43] "GET / HTTP/1.1" 304 -
127.0.0.1 - - [26/Mar/2021 21:41:43] "GET /img/unifor.png HTTP/1.1" 200 -
```

Conclusão

O servidor é capaz de entregar uma página web HTML, imagens e até mesmo arquivos CSS, não foi utilizado arquivos JavaScript no projeto. Estou utilizando a classe [http.server](#) para despachar as requisições para o manipulador e armazenar o endereço do servidor e porta.

Repositório do GitHub: <https://github.com/May199/python-server>