INF1407 - Programação para Web - T1

Antonio Jardim - 1610422  
Felipe Metson - 1520302

Arquitetura:

O servido foi organizado com a seguinte arquitetura:

* Servidor(servidorLinux.py): é o arquivo responsável por roda o servidor e receber as requisições. É lá que a conexão socket é estabelecida entre cliente e servidor.
* Controller(servidorController.py): possui a classe ServidorController, responsável por lidar e direcionar as requisições recebidas pelo servidor. Nela evitamos que requisições que não são GET sejam feitas e é partir dele que geramos a response para o servidor. Também é nele que é feita a chamada para os métodos do Service.
* Service(servidorLinux.py): é a classe que lida com as regras de negocio estabelecidas. Tanto pelo carregamento do arquivo pedido pelo cliente na requisição, quanto pela verificação se esse arquivo existe e está na lista de arquivos presente no Configurator.
* Configurator(arquivo.py): Responsável por armazenar informações imprescindíveis para a inicialização do servidor, como host e porta, além de guardar a lista de arquivos que serão aceitas nas requisições.

Colocando Servidor para funcionar:

Para facilitar a explicação de como rodar a explicação, escrevemos esse passo-a-passo na seção:

1) No terminal se dirija à raiz do projeto, mais precisamente no diretório do arquivo servidorLinux.py.

$ cd <caminho\_que\_foi\_colocado\_o\_projeto>

2) Executar o comando:

$ python3 servidorLinux.py

Pacotes utilizados que precisam estar instalados:

* os
* sys
* genericpath
* socket
* time

Testes de Requisição:

* Testando conexões simultâneas:

Vale destacar que, como recomendado no enunciado, utilizamos de um sleep(10) para validar que o servidor possui capacidade de lidar com a simultaneadade das requisições. Compravados nos testes que ocorrem sem problemas.

* Casos de teste:

Testamos o acesso a todos as chamadas abaixo verificando se a resposta era compatível com o teste em questão.

i) Casos de Retorno Válido:

A) Arquivos texto:

localhosthost:8080/arquivo/arquivo.html

localhosthost:8080/arquivo/arquivo.js

B) Arquivos de imagem:

localhosthost:8080/arquivo/arquivo.jpeg

localhosthost:8080/arquivo/arquivo.png

localhosthost:8080/arquivo/arquivo.gif

ii) Casos de Erro:

A) Sem caminho definido:

localhosthost:8080

B) Com caminho inexistente:

localhosthost:8080/teste