

ALUNOS: Gabriel Xavier Cabral , Mellanie Marques Fideles de Sousa ,
Taw-Ham Almeida Balbino de Paula

1. SNAPE

O Snape é um aplicativo multi-cliente, que permite ao cliente escrever em um arquivo, criar um arquivo, ler o conteúdo de um arquivo e baixar o arquivo.

O snape foi desenvolvida com a linguagem python, utilizando sockets, para a criação, estruturação da aplicação do servidor e do cliente, em ambas aplicações foi escolhido o protocolo de transporte TCP, pois o mesmo possui checagem de erro e checa se está faltando algum dado chegar para montar a informação.

1.1 Multiprocesso

Com o avanço da tecnologia, é perceptível que os servidores precisam atender várias requisições do cliente em simultâneo, em razão dessa problemática o Snape possui a característica de multi-cliente usando threads específicas para cada cliente conectado.

2. Comando Baixar(Download):

A parte do download, é responsável por baixar os arquivos existentes. Para baixá-los, é preciso digitar o comando na lista de comandos, tendo a seguinte mensagem (“baixar” + “nome do arquivo”) feito isso, é enviado para o servidor o comando download, mais o nome do arquivo, se o usuário digitar só “baixar” sem o nome do arquivo será exibida a mensagem de erro(406 : Comando incompleto). Ao baixar com sucesso o arquivo, será retornada uma

mensagem da lista de sucesso, com o seguinte enunciado (203 : download com sucesso). Caso ocorra um erro, será retornado uma mensagem da lista de erros (404 : Arquivo não existe)

2.1 Comando Criar(Create):

Comando criar, é utilizado para criar um arquivo. Para usar essa funcionalidade, será preciso digitar o comando, presente na lista de comandos criada, tendo o seguinte mensagem (“criar” + “nome do arquivo”), passado o comando, será enviado para o servidor, o comando CREATE, mais o nome do arquivo. Caso o arquivo já exista, será retornada uma mensagem da lista de erros, contendo a seguinte mensagem (444 : Arquivo já existe), caso o comando não exista, será retornado a mensagem (Comando não existe), caso ocorra tudo certo, será retornada a mensagem (201 : Arquivo criado com sucesso).

2.2 Comando Read(Ler)

Esse comando serve para ler um arquivo existente. Quando o usuário digita (Ler mais o nome do arquivo) o programa manda para o servidor o comando read e o nome do arquivo. Com isso, ele irá verificar se possui algum erro, se tiver, ele exibirá na tela a mensagem de erro (404 : Arquivo não existe), se não encontrar nenhum erro exibirá na tela o conteúdo do arquivo solicitado, caso o arquivo não tenha nenhum conteúdo, será retornada uma mensagem de erro (402 : Não possui conteúdo).

2.3 Comando Write(Escriver)

O comando Write como o próprio nome diz serve para escrever em um arquivo já existente.

Assim que o usuário digita(“escrever” + “nome do arquivo” + “conteúdo que quero adicionar no arquivo”) o programa reconhece e manda para o servidor: o comando WRITE, mais o nome do arquivo, mais o texto que vai ser escrito no arquivo.

Depois o programa decodifica a mensagem e verifica se há algum erro. Se o comando for passado de forma incompleta, no caso, faltando o nome do arquivo, ou o texto que se deseja escrever no arquivo, é exibida uma mensagem de erro 406, da lista de erros (comando incompleto), se não, ele exibirá a mensagem de sucesso 205 (Alterações salvas com sucesso) e a mensagem será escrita no arquivo passado.

OBS: Se o usuário não digitar nenhum comando e clicar em ‘Enter’ exibirá na tela o erro(409: Campo vazio)

