

RELATÓRIO DO PROJETO

João Paulo Agostinho da Silva
Yuri Dimitri Ramos Costa

Principais funcionalidades da aplicação

Para a nossa aplicação, utilizamos um protocolo de comunicação contendo um único comando (IMAGEM) e 2 parâmetros (PESSOA e COR DO TEXTO). Utilizamos da arquitetura cliente-servidor com o protocolo de transmissão TCP. O objetivo final da aplicação é gerar no servidor e receber no cliente imagens cômicas contendo frases desejadas. A imagem é composta por uma foto de uma personalidade famosa ao lado de uma mensagem.

A comunicação é feita seguindo os passos:

1. Estabelecimento da conexão
2. Envio do protocolo pelo cliente
3. Recepção do protocolo no servidor
4. Geração da imagem no servidor
5. Envio da resposta de status da aplicação no servidor
6. Envio da imagem pelo servidor
7. Recepção do status da aplicação no cliente
8. Recepção da imagem pelo cliente
9. Escolha do nome do arquivo para a imagem no cliente

Um exemplo de protocolo é como segue:

IMAGEM
PESSOA:BILL GATES
COR DO TEXTO:BRANCO
Texto desejado 1
...
Texto desejado n
(Linha em branco)

Procedimentos de uso

Requerimentos: python 3 e pip

Extraia o arquivo .zip fornecido em uma pasta desejada. (se preferir use a página home).

Em seguida, acesse a pasta do projeto utilizando o terminal. Por meio do comando pip abaixo, instale todas as bibliotecas necessárias para o funcionamento do projeto.

```
cd projetoredes/src/  
pip install -r requirements.txt
```

Para rodar o servidor, utilize o seguinte comando

```
python3 server.py
```

Em seguida, em uma nova janela do terminal, acesse novamente a pasta do projeto e para rodar um cliente digite o seguinte comando:

```
python3 client.py
```

Informações adicionais

1. Durante a conexão com o servidor, o cliente pode inserir o protocolo ou se desconectar. Para desconectar deve-se digitar “!SAIR”.
2. O protocolo é formado pelos seguintes campos: comando, pessoa, cor do texto e linha vazia. Cada campo é seguido de uma quebra de linha.
3. Como a única funcionalidade de nossa aplicação é gerar imagem. O campo comando só aceita a entrada *imagem*. E, respeitando o seguinte formato:
3.1. IMAGEM
4. Em *pessoa* são possíveis as seguintes opções: BILL GATES, CHURCHILL, TIRINGA e KANT. Segue o exemplo:
4.1. PESSOA:CHURCHILL
5. Em *cor do texto* são possíveis as seguintes opções: BRANCO, VERMELHO, PRETO, AZUL e VERDE. Segue o exemplo:
5.1. COR DO TEXTO:VERMELHO
6. O campo da mensagem pode utilizar várias linhas. Porém, a linha vazia é entendida como o fim da mensagem.
7. Se algum dos campos for preenchido incorretamente, o erro específico será exibido.
8. Se todos os campos forem preenchidos corretamente, será necessário nomear a imagem.

O que poderia ter sido implementado a mais

A ideia original foi concretizada com sucesso. Porém, algumas funcionalidades podem ser acrescentadas futuramente. Uma sugestão é permitir que o cliente entre apenas com um comando, uma pessoa e uma cor do texto e o servidor se encarregue de escolher aleatoriamente uma mensagem. Haveria à disposição do servidor um repertório de mensagens humorísticas.

Dificuldades encontradas no desenvolvimento do projeto

No início do projeto foi utilizado o módulo *OpenCV* para a geração e tratamento de imagens, mas foram encontradas dificuldades no uso da acentuação nas palavras e na disposição das quebras de linha no texto da imagem. Após muitas tentativas, a solução foi substituir o módulo *OpenCV* pelo pacote *Pillow*. A partir de então, as imagens puderam ser geradas conforme as especificações do projeto.