

Relatório Projeto Lab Conversão PDF2TXT e Redução de Resolução

Nome	TIA
Lucca Eiki Amarante Millian	10390794
Rafael de Toledo Navarro	10389955

Decisões do projeto

Containers:

Em todos containers, foram utilizados os repositórios de debian:stable, devido a grande disponibilidade de pacotes.

Ghostscript e PDFtoText:

Os códigos contidos nos containers de ghostscript e pdftotext foram criados usando a linguagem BASH, para maior facilidade de integração com as aplicações numa máquina Linux.

Logs:

O container de Log foi feito utilizando da linguagem de programação Python, a decisão foi feita com base na facilidade da utilização de Sockets para a comunicação com os outros Containers Docker.

Funcionamento da aplicação

Para o funcionamento da aplicação, é necessário instalar Python3 e Flask e executar o “comandos.sh” contido no arquivo a fim da satisfação de todas dependencias.

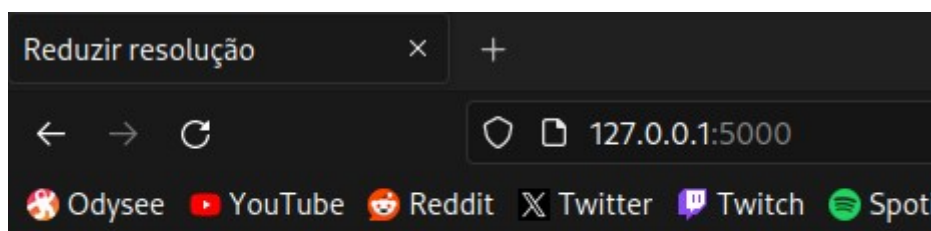
Com o “comandos.sh”, o Docker utilizado para Log também já é iniciado em Daemon.

```
docker run --rm -v $(pwd)/log/logs:/app -d --net teste --name log log
```

A aplicação Flask cuida de quaisquer outros comandos necessários para o funcionamento da aplicação, bastando apenas executar o seguinte comando:

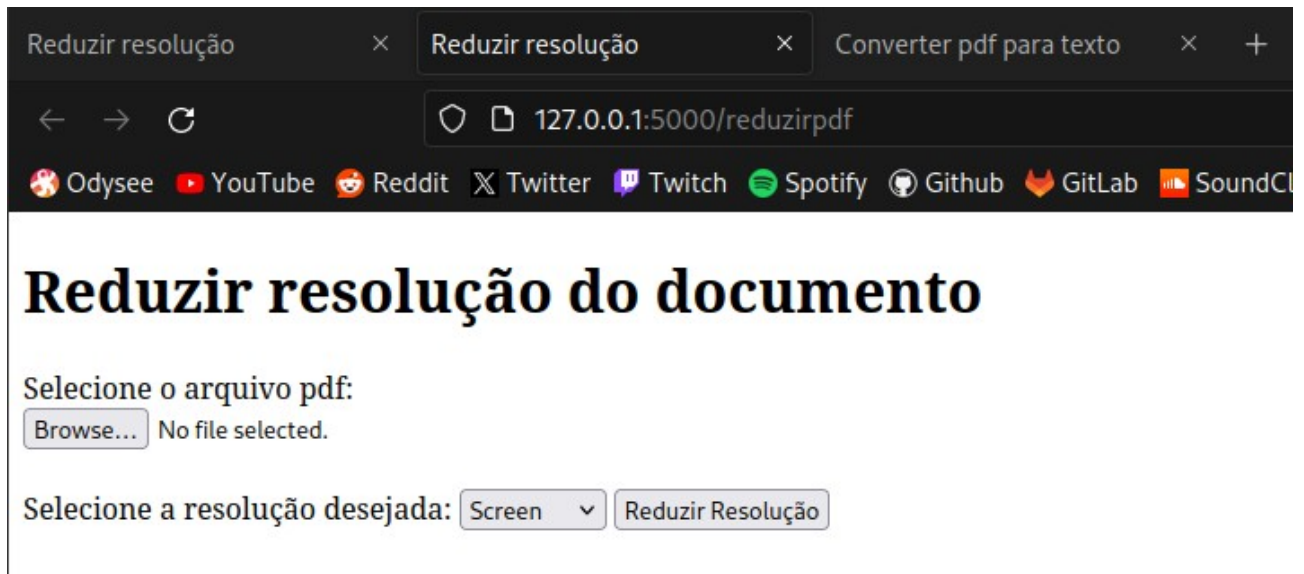
```
[mel@meowpc trabalhoDist]$ cd site/
[mel@meowpc site]$ ls
main.py  templates
[mel@meowpc site]$ python3 main.py
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 452-383-416
```

Agora, acessando a url apresentada pelo programa, o usuário é levado até a página principal, onde pode escolher entre ir para as páginas de Redução de tamanho do pdf ou de Transformação de pdf em texto.



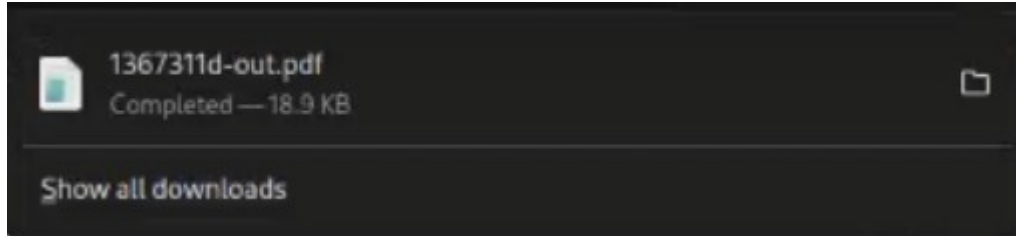
Escolha o serviço: [reduzir totxt](#)

Ao selecionar “reduzir”, o usuário é redirecionado para a página de diminuição de resolução de pdfs



O usuário então tem a capacidade de selecionar seu arquivo e selecionar a resolução desejada.

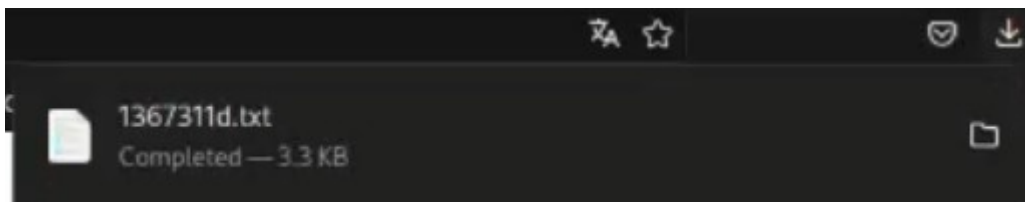
Após a seleção do PDF e resolução, quando o botão Reduzir Resolução é clicado, o documento é baixado automaticamente com um nome que consiste dos primeiros 8 dígitos do hash SHA256 do nome do arquivo + (-out.pdf).



Ao selecionar “totxt”, o usuário é redirecionado para a página de transformação de PDFs em texto.



Da mesma forma que na página de diminuição da resolução, o usuário pode selecionar o arquivo desejado e, após clicar o botão, é feito o download automático do arquivo em txt.



Logs de todas execuções são guardados em (raiz git)/log/logs/log.txt e tem o seguinte formato:
Data (Dia-Mes-Ano Hora-Minuto-Segundo); Aplicação utilizada; Código de retorno da Aplicação

```
[mel@meowpc logs]$ cat log.txt
26-04-2024 03-15-14; PDFtoTXT; 0
26-04-2024 03-15-26; GhostScript; 0
26-04-2024 03-15-57; Serviço de Log; -1
26-04-2024 03-59-03; GhostScript; 0
26-04-2024 03-59-20; PDFtoTXT; 0
26-04-2024 04-58-43; GhostScript; 0
26-04-2024 04-58-53; GhostScript; 0
26-04-2024 04-58-56; GhostScript; 0
26-04-2024 04-59-56; GhostScript; 0
26-04-2024 05-06-04; GhostScript; 0
26-04-2024 05-06-45; PDFtoTXT; 0
```

Para autenticação, é enviado um Token junto com as informações após a execução do aplicativo utilizando o comando nc do pacote netcat-traditional para Debian.

```
envio="$TOKEN; $(date +%d-%m-%Y %H-%M-%S); GhostScript; $resposta"
echo $envio
echo -n "$envio" | nc -q 0 log 12345
```

```
envio="$TOKEN; $(date +%d-%m-%Y %H-%M-%S); PDFtoTXT; $resposta"
echo $envio
echo -n "$envio" | nc -q 0 log 12345
```

O aplicativo de log recebe as informações por Socket, verifica se o token está presente na string recebida e, caso esteja, guarda as informações enviadas no arquivo de Log sem o token e, caso o token não seja encontrado, o Log deixa a tentativa falha de acesso com código -1 (Token incorreto ou inexistente).

```
with open("/app/log.txt","a+") as f:
    if(arrdata[0] == TOKEN):
        f.write(data.decode()[7:])
        f.write('\n')
    else:
        data = datetime.datetime.now().strftime("%d-%m-%Y %H-%M-%S")
        f.write(data)
        f.write("; Serviço de Log; -1")
        f.write('\n')
```

Contribuições dos membros

Lucca Millian:

Criação dos arquivos de configuração Dockerfile, configuração da Network dos dockers, criação de scripts BASH para utilização dos aplicativos pdftotext e ghostscript.

Rafael Navarro:

Criação dos scripts Python, configuração dos arquivos Flask e criação de templates HTML.