

Imagens espalhadas em vários diretórios

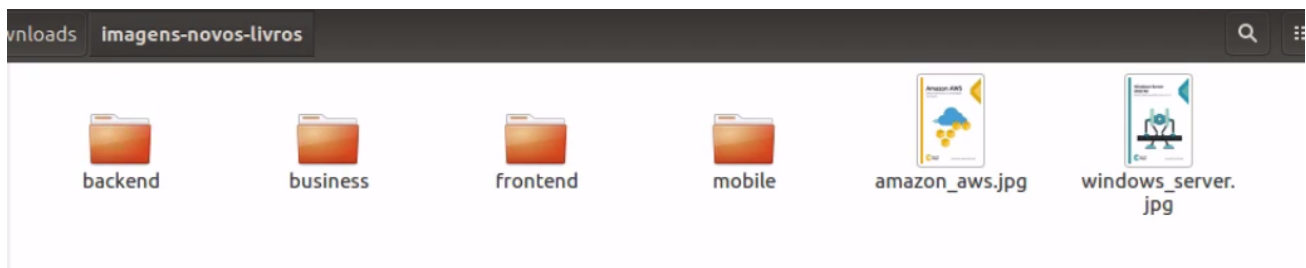
Transcrição

Na etapa anterior, os diretores da *Multillidae* haviam nos passado o diretório `imagens-livros` para realizarmos a conversão dos arquivos da extensão `.jpg` para a extensão `.png`.

Com o passar dos meses, a *Multillidae* cresceu muito, e eles repararam que a divisão de colocar todos os arquivos `.jpg` dentro de somente um diretório, não estava ajudando muito, pois por causa do volume de dados, acabava dificultando o manuseio com as imagens.

Então, eles decidiram dividir os arquivos `.jpg` em diretórios respectivos a sua área de conhecimento. Recebemos um novo diretório chamado de `imagens-novos-livros`, onde devemos também fazer a conversão dos arquivos `.jpg` para `.png`.

Esse diretório está estruturado da seguinte maneira:



Como podemos ver, ela está dividida por áreas de conhecimento. Quando clicamos em um diretório, encontramos outros diretórios e outras imagens, e assim por diante.

Acontece que o script que criamos nas aulas anteriores, não pode nos ajudar nessa tarefa, pois o script anterior considerava que todas as imagens estariam dentro do mesmo diretório.

Agora temos vários diretórios, que contêm vários diretórios, e que estes contêm vários outros diretórios, e também, várias imagens espalhadas nesses diretórios.

Criaremos um **novo script** para atender as requisições dos diretores da *Multillidae*.

Vamos ao terminal, e atualmente estamos no diretório `/Scripts`, onde criaremos esse novo script para atender essa requisição.

```
$ nano conversao-novos-livros.sh
```

Como já foi comentado, a primeira linha do script deve conter o **interpretador**.

```
#!/bin/bash
```

O primeiro passo a partir de agora, é sair do diretório `/Scripts`, e chegar no diretório `/imagens-novos-livros`, onde estão os arquivos que sofrerão a conversão:

```
#!/bin/bash

cd ~/Downloads/imagens-novos-livros
```

Uma vez que essa linha for executada, estaremos dentro do diretório `/imagens-novos-livros`. Uma vez estando nessa pasta, é necessário realizar uma varredura de todo o conteúdo dela. Para isso, utilizaremos a estrutura de repetição **for**!!

Junto com o `for`, colocaremos o nome de uma variável, e depois um `*`, dizendo que será feito uma varredura de todo o conteúdo dentro de `/imagens-novos-livros`.

```
#!/bin/bash

cd ~/Downloads/imagens-novos-livros
for arquivo in *
```

Após a linha do `for`, implementaremos o comando de **execução** do `do`. Dentro de `/imagens-novos-livros`, podemos encontrar arquivos que sejam **diretórios**, ou até mesmo **imagens** que estão no formato `.jpg`. Com isso, temos que analisar esses dois casos. (*Se é um diretório, ou se é uma imagem*).

Dentro do `if`, usaremos o comando `-d` para verificar se o conteúdo da variável `arquivo` é um diretório. Se for um *diretório*, temos que acessá-lo e fazer uma varredura do conteúdo.

Se o conteúdo dessa variável não é um diretório, então é uma imagem. Portanto, faremos a conversão do arquivo `.jpg` para `.png`.

```
for arquivo in *
do
    if [ -d $arquivo ]
    then
        #Entrar no diretório e fazer varredura do conteúdo
    else
        #Conversao jpg para png
    fi
done
```

Imagine que o conteúdo de `arquivo` é um diretório, por exemplo o `/backend`. Como foi falado, temos que entrar nesse diretório, e realizar uma varredura do conteúdo. E é isso que faremos agora no código:

```
for arquivo in *
do
    if [ -d $arquivo ]
    then
        cd $arquivo
        for conteudo_arquivo in *
        do
            if [ -d $conteudo_arquivo ]
            then
                #Entrar no diretorio e fazer a varredura
            else
                #Conversao jpg para png
            fi
        done
    fi
done
```

```
done
else
    #Conversao jpg para png
fi
done
```

Após acessar o diretório `cd $arquivo`, fizemos a varredura utilizando o laço de repetição `for`.

Perceba que esse código é exatamente igual ao código externo. E vimos que ficar repetindo código não é a solução mais elegante que podemos ter. Com isso, na sequência veremos como podemos solucionar esse problema.