

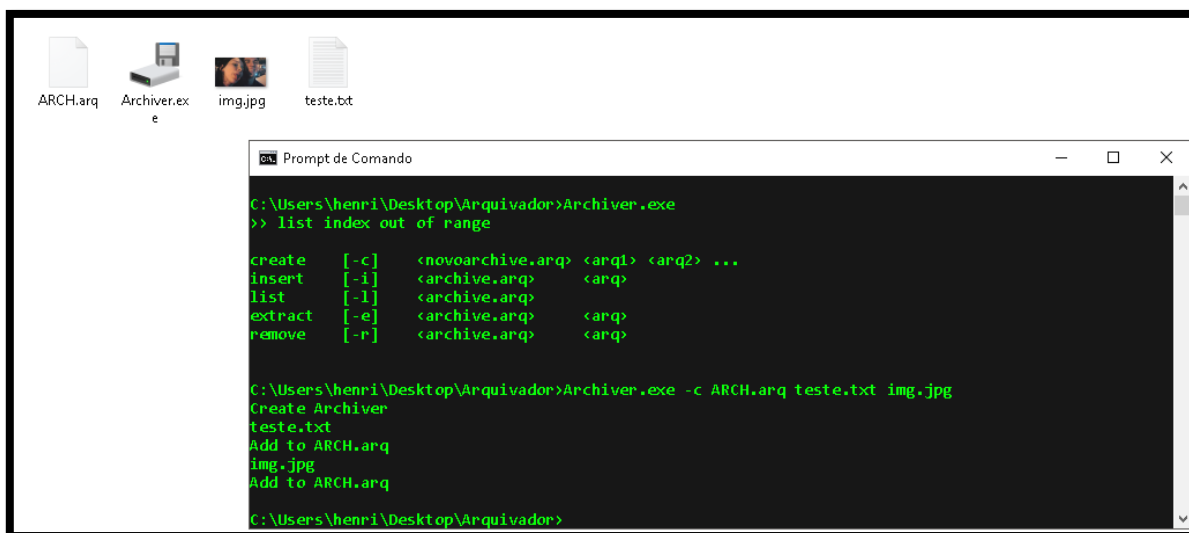
# Funções

1. Criar um archive com base em uma lista de arquivos informados.

**arquivador -c <novoarchiver> <arq1> <arq2> ...**

Para começar a nossa apresentação do código, temos a criação do caso de uso que cria um archive em que se coloca os arquivos dentro, sendo, nesse caso, os parâmetros passados.

De início é necessário criar o archive propriamente dito, assim, para isso é chamada uma função, na qual tem como prioridade abrir um arquivo em modo de escrita (“w”), e, visto que o arquivo não existe, ele é criado e logo após isso fechado, para a inserção dos parâmetros, passados em argumentos(“argv”), que é feito um laço para execução da função para adicionar, no qual ocorre a abertura do archive criado anteriormente em modo de acrescentar dados no arquivo(“ab+”) e, em seguida, é aberto o arquivo a ser adicionado em modo de leitura binária (“rb”) para obter seu conteúdo. Após a leitura do arquivo para inserção, é criado um cabeçalho de identificação do arquivo em que são passados seu nome e tamanho, em que após isso o mesmo é codificado para bytes (“header.encode()”) e adicionado no archive juntamente com o conteúdo propriamente dito.



```
C:\Users\henri\Desktop\Arquivador>Archiver.exe
>> list index out of range

create  [-c]    <novoarchive.arq> <arq1> <arq2> ...
insert  [-i]    <archive.arq>      <arq>
list    [-l]    <archive.arq>      <arq>
extract [-e]    <archive.arq>      <arq>
remove  [-r]    <archive.arq>      <arq>

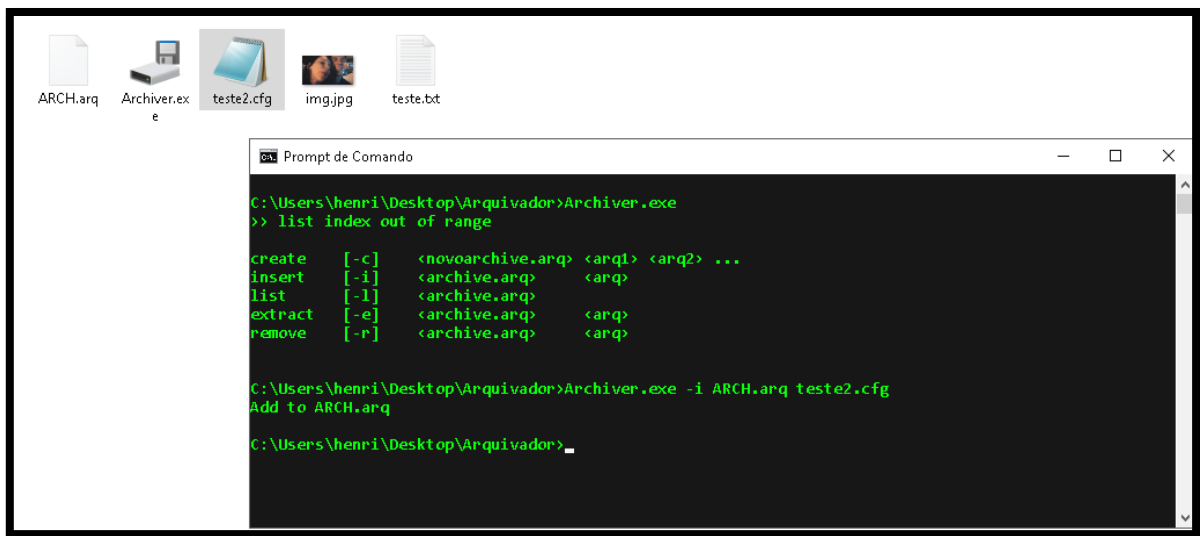
C:\Users\henri\Desktop\Arquivador>Archiver.exe -c ARCH.arq teste.txt img.jpg
Create Archiver
teste.txt
Add to ARCH.arq
img.jpg
Add to ARCH.arq

C:\Users\henri\Desktop\Arquivador>
```

## 2. Inserir um arquivo em um archive já criado.

**arquivador -i <archive.arq> <arq>**

Nessa função é realizada, basicamente, o mesmo princípio da anterior. Porém, como já explicado, é executado única e apenas uma vez, e não dentro do loop com quantidade de parâmetros. Assim, há a saída e o respectivo código apresentados abaixo.



```
C:\Users\henri\Desktop\Arquivador>Archiver.exe
>> list index out of range

create  [-c]    <novoarchive.arq> <arq1> <arq2> ...
insert  [-i]    <archive.arq>      <arq>
list    [-l]    <archive.arq>      <arq>
extract [-e]    <archive.arq>      <arq>
remove  [-r]    <archive.arq>      <arq>

C:\Users\henri\Desktop\Arquivador>Archiver.exe -i ARCH.arq teste2.cfg
Add to ARCH.arq

C:\Users\henri\Desktop\Arquivador>_
```

### 3. Listar os nomes dos arquivos armazenados em um archive.

**arquivador -l <archive.arq>**

Para fazer a listagem dos arquivos contidos no archive, de início é necessário abri-lo em modo de leitura binária (“rb”). Com isso, começamos a ler o cabeçalho do primeiro arquivo, onde está armazenado seu nome e tamanho (em bytes), e conteúdo, ambos separados pelo caractere “|”.

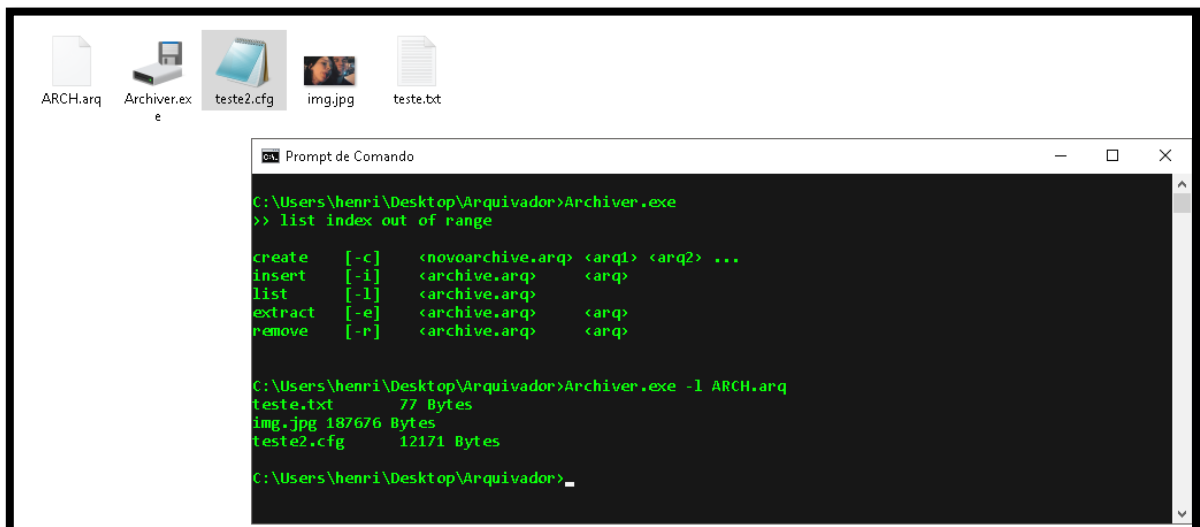
Exemplo: “teste1.txt|48|...”

Com isso precisamos tratá-lo de forma separada do restante do archive e não como um bloco só. Para isso, separamos os conteúdos do cabeçalho pelo caractere separado (header = data.split(b'|')), e assim o header se transforma em um vetor, onde header[0] corresponde ao nome header[1] corresponde ao tamanho, e header[2] ao conteúdo do arquivo que será listado. Dessa forma, é printado na tela o header[0] e header[1].

Exemplo: “teste1.txt 48”

Em seguida para avançar o próximo arquivo dentro do archive posicionamos o cursor de dentro do archive (“archive.seek()”) adicionando o conteúdo do header[1] para pular o conteúdo e partir para o cabeçalho do próximo arquivo em que a operação se repete.

Exemplo: pos += len(header[0]) + len(header[1]) + 2 + int(header[1])  
archive.seek(pos, 0)



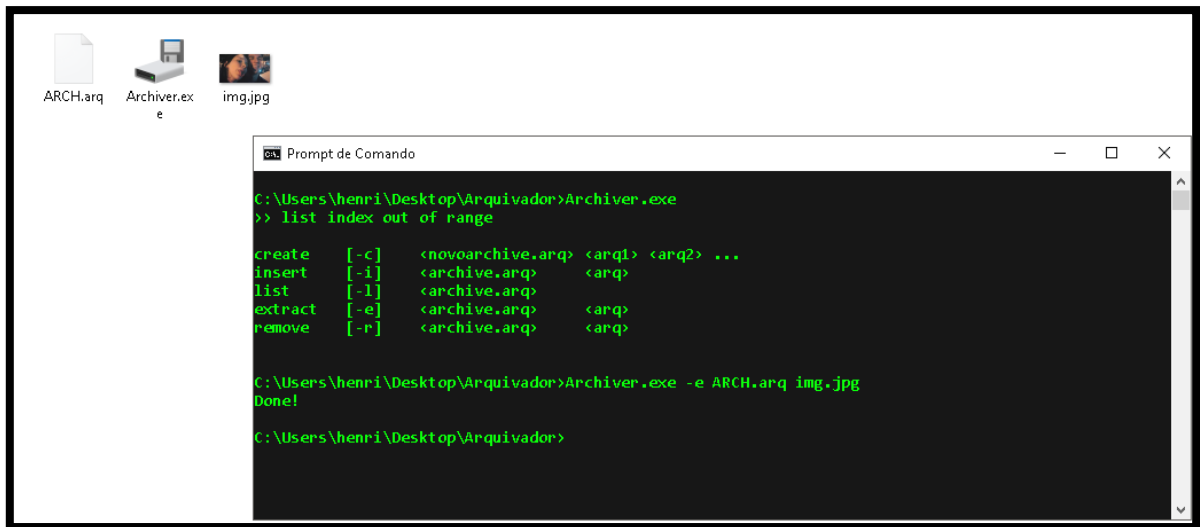
#### 4. Extrair um arquivo de um archive.

**arquivador -e <archive.arq> <arq>**

Para executar a extração é seguido o mesmo conceito da listagem, mas com uma verificação do nome do arquivo em cada cabeçalho percorrido até encontrar o nome de cabeçalho igual ao nome do arquivo a ser extraído.

Exemplo: “if (header[0] == nome)”

Quando o mesmo é encontrado se inicia o processo de extração. É aberto um arquivo de mesmo nome em modo de escrita binária (“file = open(nome, 'wb')”), e em seguida é lido o conteúdo do arquivo dentro do archive (“data = archive.read(int(header[1].decode())”), e este por sua vez é escrito no arquivo criado (“file.write(data)”).



## 5. Remover um arquivo de um archive.

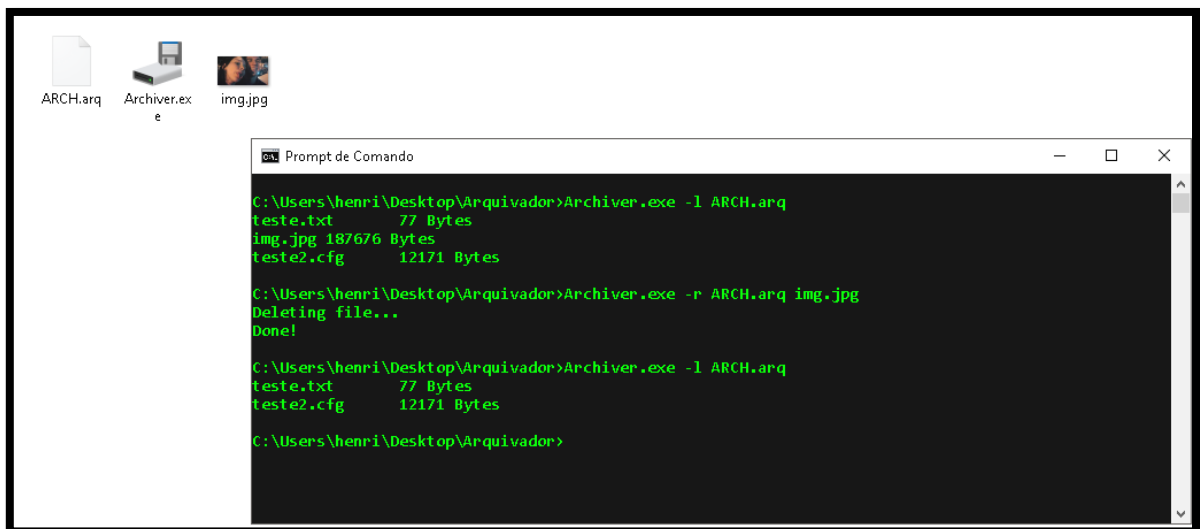
**arquivador -r <archive.arq> <arq>**

Para deletar um arquivo do archive segue o mesmo princípio da extração com algumas modificações. De início, o processo se comporta de maneira semelhante até encontrar o nome de cabeçalho correspondente ao nome do arquivo a ser excluído, a partir daí se inicia o processo de exclusão.

Esse processo se baseia no princípio de sobreposição, uma vez que o archive é reescrito com a modificação.

Após localizado o arquivo a ser excluído, armazenamos todos os arquivos que precedem o mesmo (“antesLixo”), o mesmo (“lixo”), e posterior a ele (“depoisLixo”).

Em seguida, abrimos o archive em modo de escrita binária (“wb”), o que possibilita a sobreposição dos dados. Dessa forma, o archive é reescrito com os dados “antesLixo” e “depoisLixo”, fazendo assim com que o arquivo em questão seja excluído.



```
C:\Users\henri\Desktop\Arquivador>Archiver.exe -l ARCH.arq
teste.txt      77 Bytes
img.jpg 187676 Bytes
teste2.cfg     12171 Bytes

C:\Users\henri\Desktop\Arquivador>Archiver.exe -r ARCH.arq img.jpg
Deleting file...
Done!

C:\Users\henri\Desktop\Arquivador>Archiver.exe -l ARCH.arq
teste.txt      77 Bytes
teste2.cfg     12171 Bytes

C:\Users\henri\Desktop\Arquivador>
```