

O primeiros comandos no prompt

Transcrição

Aqui no Alura usaremos várias ferramentas na linha de comando! Usar a linha de comando significa que não teremos uma interface gráfica bonita, e o mouse não vai funcionar! E pode acreditar, dá para fazer muita coisa sem usar o mouse ou toques na tela.

Mas aí vem a pergunta, por que eu preciso aprender isso? Bom, muitas ferramentas no setor de desenvolvimento não possuem uma interface gráfica, pois elas devem funcionar em qualquer computador, inclusive nos computadores que não possuem essa interface, como por exemplo os computadores na nuvem.

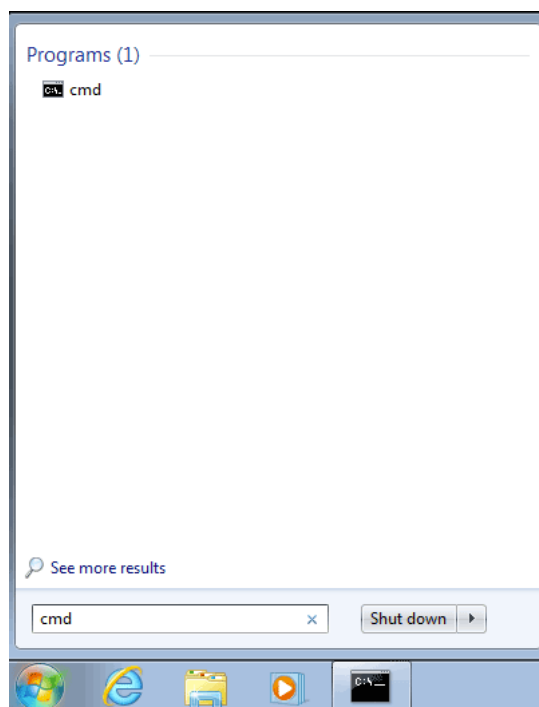
Outra razão, e talvez a mais importante, é que as ferramentas na linha de comando podem ser controlados facilmente através de scripts, ou seja outras ferramentas. Isso é relacionado com a automação, ensinando as máquinas a fazerem vários passos automaticamente, e muito mais rápido do que a gente consegue com o mouse em mãos.

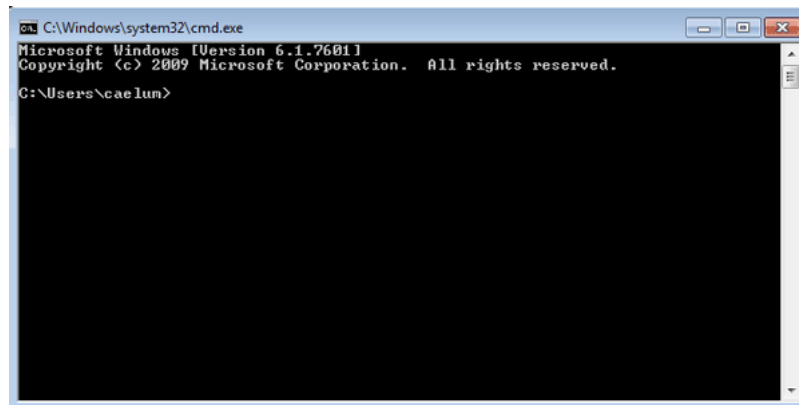
Enfim, o desenvolvedor de hoje em dia precisa dominar alguns comandos da linha de comando para se dar bem no desenvolvimento e também aqui na plataforma Alura! Vamos lá? Bora aprender?

Abrir o Prompt de comando

O primeiro passo é abrir um terminal (ou console). No mundo Windows esse terminal se chama **Prompt de comando** ou abreviando **cmd**. Para abrir um novo terminal devemos ir em *Todos os Programas*, *Acessórios* e depois no *Prompt de Comando*, aí abrirá uma tela preta linda :).

Obs: Outra forma de abrir o Prompt é clicar no *Botão Iniciar* e no campo de pesquisa digitar *Prompt* ou *cmd*:





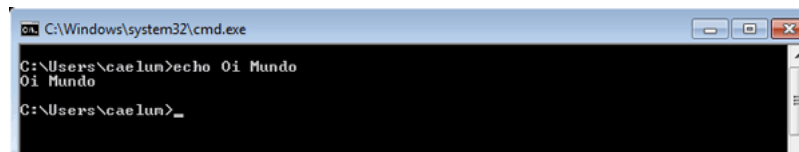
A partir dessa tela podemos interagir com o sistema operacional escrevendo comandos! Vamos ver como navegar entre pastas, criar novas pastas, ler, executar, copiar e mover arquivos e muito mais. Vamos tentar?

*Obs: O termo ***prompt** se refere ao sinal **>***

*Lembrando: O Prompt também é chamado de **cmd**, terminal ou console. Todos estes são sinônimos nesse contexto.*

No mundo do desenvolvimento qualquer nova tecnologia que você aprende começa com o *Oi Mundo* para te dar um boas vindas nessa tecnologia. Aqui não será diferente! Com um terminal aberto digite:

```
echo Oi Mundo
```



Você já deve saber o que o comando `echo` faz. Sim, imprime no terminal, mas ainda não é muito útil para gente, né?

Listando arquivos

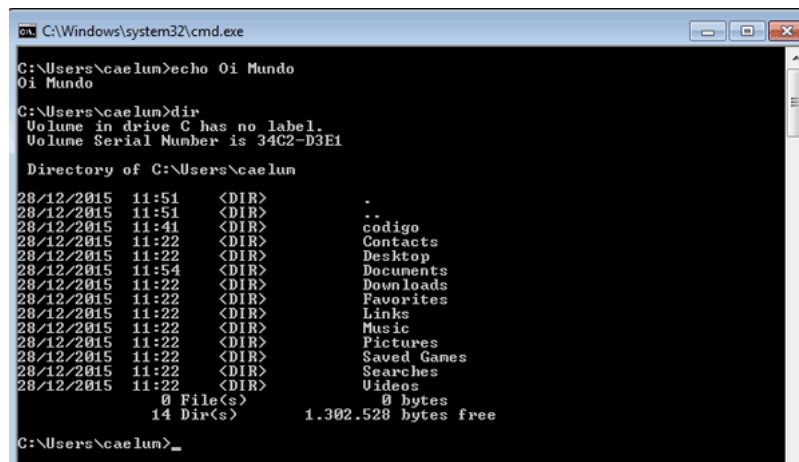
Vamos dar uma olhada com mais carinho no que aconteceu. Ao abrir o prompt, ele automaticamente inicia na sua pasta pessoal. Repare que antes do cursor aparece o caminho da pasta atual seguido pelo prompt (**>**), que aqui é:

```
C:\Users\caelum>
```

O `c:` é a partição principal nesse computador. Pode ser que você tenha outras instaladas como o `d:`, mas o *prompt* sempre começa no `c:`, seguido pelo a pasta `Users` e na sub-pasta `caelum`. A *contra barra* `\` é o sinal separador de diretórios na linha de comando, ok?

Já que o Prompt está pedindo que a gente execute algo vamos listar todos os arquivos através do comando **dir**. Escreva no prompt:

```
dir
```



```
C:\Windows\system32\cmd.exe

C:\Users\caelum>echo Oi Mundo
Oi Mundo

C:\Users\caelum>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelum

28/12/2015  11:51    <DIR>        .
28/12/2015  11:51    <DIR>        ..
28/12/2015  11:41    <DIR>        codigo
28/12/2015  11:22    <DIR>        Contacts
28/12/2015  11:22    <DIR>        Desktop
28/12/2015  11:54    <DIR>        Documents
28/12/2015  11:22    <DIR>        Downloads
28/12/2015  11:22    <DIR>        Favorites
28/12/2015  11:22    <DIR>        Links
28/12/2015  11:22    <DIR>        Music
28/12/2015  11:22    <DIR>        Pictures
28/12/2015  11:22    <DIR>        Saved Games
28/12/2015  11:22    <DIR>        Searches
28/12/2015  11:22    <DIR>        Videos
                0 File(s)                0 bytes
                14 Dir(s)            1.302.528 bytes free

C:\Users\caelum>
```

Recebemos uma lista dos arquivos e pastas existentes. Então através do `dir` sempre sabemos o que se encontra na pasta atual do prompt, ok?

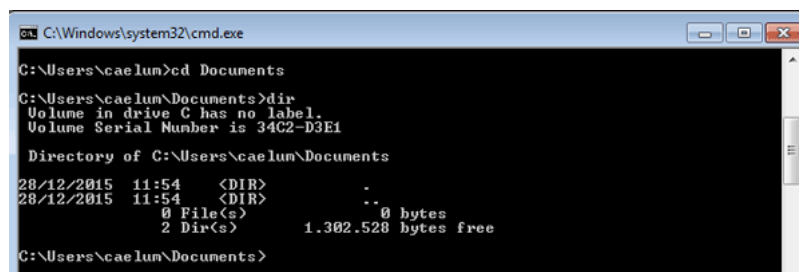
Navegando entre diretórios

Vimos o comando `dir`, que mostra todo o conteúdo da pasta atual. Vamos entrar em uma subpasta e depois re-executar o comando. Podemos mudar a pasta com o comando `cd` (*change directory*) seguido do nome da subpasta:

```
cd Documents
```

Depois liste novamente:

```
dir
```



```
C:\Windows\system32\cmd.exe

C:\Users\caelum>cd Documents
C:\Users\caelum\Documents>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelum\Documents

28/12/2015  11:54    <DIR>        .
28/12/2015  11:54    <DIR>        ..
                0 File(s)                0 bytes
                2 Dir(s)            1.302.528 bytes free

C:\Users\caelum\Documents>
```

Vamos voltar para o diretório anterior. Em outras palavras, vamos subir na hierarquia de diretórios, ir uma pasta para cima. Novamente o comando `cd` tem esse papel mas agora seguido por `..` :

```
cd ..
dir
```

```
C:\Windows\system32\cmd.exe

C:\Users\caelum>cd ..

C:\Users\caelum>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelum

28/12/2015  11:51    <DIR>          .
28/12/2015  11:51    <DIR>          ..
28/12/2015  11:41    <DIR>          codigo
28/12/2015  11:22    <DIR>          Contacts
28/12/2015  11:22    <DIR>          Desktop
28/12/2015  11:54    <DIR>          Documents
28/12/2015  11:22    <DIR>          Downloads
28/12/2015  11:22    <DIR>          Favorites
28/12/2015  11:22    <DIR>          Links
28/12/2015  11:22    <DIR>          Music
28/12/2015  11:22    <DIR>          Pictures
28/12/2015  11:22    <DIR>          Saved Games
28/12/2015  11:22    <DIR>          Searches
28/12/2015  11:22    <DIR>          Videos
               0 File(s)              0 bytes
               14 Dir(s)            1.302.528 bytes free

C:\Users\caelum>
```

Criando diretórios

Voltamos para a pasta do seu usuário, mas será que a gente também pode criar novos diretórios nela? A resposta é sim e o comando que faz a mágica se chama `mkdir` (*make directory*). Vamos criar um nova pasta com o nome de `codigo` :

```
mkdir codigo
dir
```

```
C:\Windows\system32\cmd.exe

C:\Users\caelum>mkdir codigo

C:\Users\caelum>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelum

28/12/2015  12:10    <DIR>          .
28/12/2015  12:10    <DIR>          ..
28/12/2015  12:10    <DIR>          codigo
28/12/2015  11:22    <DIR>          Contacts
28/12/2015  11:22    <DIR>          Desktop
28/12/2015  12:10    <DIR>          Documents
28/12/2015  11:22    <DIR>          Downloads
28/12/2015  11:22    <DIR>          Favorites
28/12/2015  11:22    <DIR>          Links
28/12/2015  11:22    <DIR>          Music
28/12/2015  11:22    <DIR>          Pictures
28/12/2015  11:22    <DIR>          Saved Games
28/12/2015  11:22    <DIR>          Searches
28/12/2015  11:22    <DIR>          Videos
               0 File(s)              0 bytes
               14 Dir(s)            1.359.872 bytes free

C:\Users\caelum>
```

Ao listar podemos ver a nossa pasta criada, ótimo!

Será que podemos entrar nessa pasta? Lógico! E você já sabe o comando: `cd codigo` .

Ao executar `dir` nós vemos duas *coisas* estranhas. Tem um diretório que se chama `.` e um outro `..` . O que é isto?

Bom o `.` se refere ao diretório atual, onde nos estamos. Por exemplo, podemos passar para o comando `dir` o diretório que queremos listar:

```
dir C:\Users\caelum\codigo\
```

Mas já que estamos nessa pasta podemos usar:

```
dir .
```

```
C:\Windows\system32\cmd.exe

C:\Users\caelun>cd codigo
C:\Users\caelun\codigo>dir .
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelun\codigo

28/12/2015  12:10    <DIR>        ..
28/12/2015  12:10    <DIR>        .
               0 File(s)              0 bytes
               2 Dir(s)              905.216 bytes free

C:\Users\caelun\codigo>_
```

ou mais fácil ainda:

`dir`

Tudo bem até aqui? Agora tente executar `dir ..`. O resultado deve ser familiar para você, o comando deve listar o conteúdo do diretório logo acima!

```
C:\Windows\system32\cmd.exe

C:\Users\caelun\codigo>dir ..
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelun

28/12/2015  12:10    <DIR>        .
28/12/2015  12:10    <DIR>        ..
28/12/2015  12:10    <DIR>        codigo
28/12/2015  11:22    <DIR>        Contacts
28/12/2015  11:22    <DIR>        Desktop
28/12/2015  12:10    <DIR>        Documents
28/12/2015  11:22    <DIR>        Downloads
28/12/2015  11:22    <DIR>        Favorites
28/12/2015  11:22    <DIR>        Links
28/12/2015  11:22    <DIR>        Music
28/12/2015  11:22    <DIR>        Pictures
28/12/2015  11:22    <DIR>        Saved Games
28/12/2015  11:22    <DIR>        Searches
28/12/2015  11:22    <DIR>        Videos
               0 File(s)              0 bytes
              14 Dir(s)              905.216 bytes free

C:\Users\caelun\codigo>_
```

Atenção: Já vou te dar uma dica, de desenvolvedor para desenvolvedor. **Evite os acentos e espaços nos nomes de arquivos e diretórios.** Muitas ferramentas de desenvolvimento não se dão bem com esses caracteres. Por exemplo o Django do mundo Python gera vários problemas e dificulta demais o trabalho. Evite isso o máximo possível!

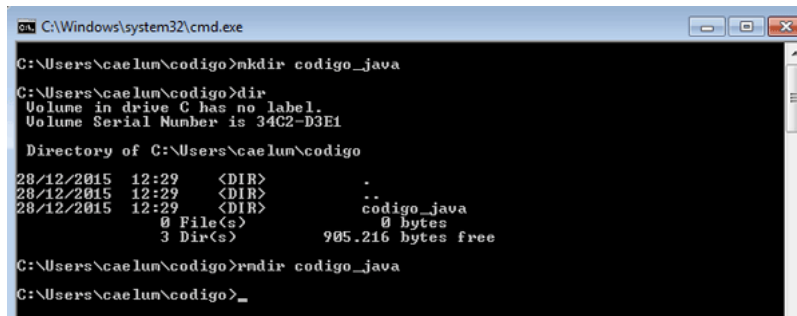
Muitos desenvolvedores na Alura usam o caractere `_` invés do espaço na hora de dar nomes para diretórios. Por exemplo:

`mkdir codigo_java`

Removendo diretórios

Vamos remover a pasta `codigo_java` e criar um sub-diretório que se chama `java` apenas. Para remover um diretório existe o comando `rmdir` :

`rmdir codigo_java`



```
C:\Windows\system32\cmd.exe

C:\Users\caelum\codigo>mkdir codigo_java

C:\Users\caelum\codigo>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

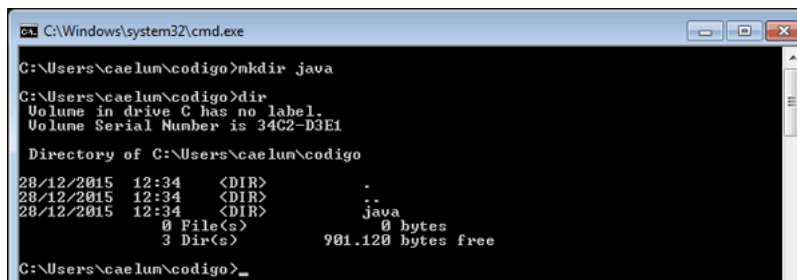
Directory of C:\Users\caelum\codigo

28/12/2015  12:29    <DIR>          .
28/12/2015  12:29    <DIR>          ..
28/12/2015  12:29    <DIR>          codigo_java
               0 File(s)                0 bytes
               3 Dir(s)              905.216 bytes free

C:\Users\caelum\codigo>rmdir codigo_java
C:\Users\caelum\codigo>
```

e para criar a pasta nova:

```
mkdir java
```



```
C:\Windows\system32\cmd.exe

C:\Users\caelum\codigo>mkdir java

C:\Users\caelum\codigo>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelum\codigo

28/12/2015  12:34    <DIR>          .
28/12/2015  12:34    <DIR>          ..
28/12/2015  12:34    <DIR>          java
               0 File(s)                0 bytes
               3 Dir(s)              901.120 bytes free

C:\Users\caelum\codigo>
```

##Limpando o terminal

Com o tempo, o terminal vai ficar poluído com comandos antigos, que tiram o foco e dificultam a legibilidade. Por isso existe um comando para limpar a tela que se chama `cls` (*clear screen*). Vamos testar:

```
cls
```

Ótimo, melhor continuar com tela a limpa!

##Trabalhando com arquivos

Já podemos criar, apagar diretórios e navegar entre eles na linha de comando. Isso já vai nos ajudar muito nos treinamentos do Alura, pode acreditar. No entanto também é preciso saber como lidar com arquivos. Claro que nada substitui um editor de texto de verdade mas em alguns casos é uma boa ajuda poder mexer com arquivos na linha de comando.

Vamos criar um arquivo rapidinho. Você se lembra do comando `echo` ? Vamos testá-lo:

```
echo Oi Mundo > saida.txt
```

O que isso faz?? Calma, o comando `echo` você já conhece, mas o resto? Talvez você reparou que não apareceu nenhuma saída no terminal! Toda a saída foi gravada no arquivo `saida.txt` . Então o caractere `>` pega a saída de um comando e a grava no arquivo indicado (ou repassa para outro comando). Digite `dir` para ver o arquivo recém criado:

```
dir
```



```
C:\Windows\system32\cmd.exe

C:\Users\caelum\codigo>echo Oi Mundo > saida.txt

C:\Users\caelum\codigo>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

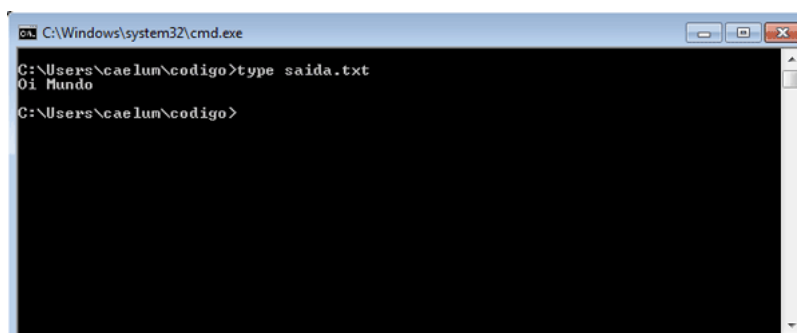
Directory of C:\Users\caelum\codigo

28/12/2015  13:15    <DIR>        .
28/12/2015  13:15    <DIR>        ..
28/12/2015  12:34    <DIR>        java
28/12/2015  13:15                11 saida.txt
                        11 bytes
                1 File(s)
                3 Dir(s)          286.720 bytes free

C:\Users\caelum\codigo>
```

Deve aparecer o arquivo `saida.txt` . Mas como posso ver o conteúdo desse arquivo? Fechar o terminal e abrir um editor de texto não vale :) Claro que há um comando para isso, e ele se chama `type` , então digite:

```
type saida.txt
```



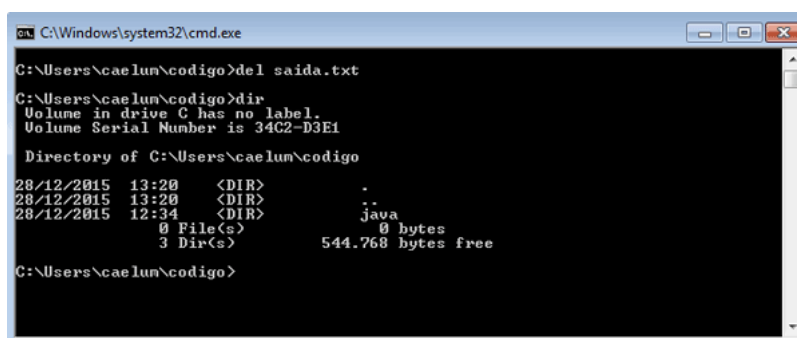
```
C:\Windows\system32\cmd.exe

C:\Users\caelum\codigo>type saida.txt
Oi Mundo

C:\Users\caelum\codigo>
```

Vamos testar novamente, mas antes disto vamos limpar o nosso terminal (`cls`). Primeiro apagaremos o arquivo antigo, o comando que faz isso se chama `del` :

```
del saida.txt
```



```
C:\Windows\system32\cmd.exe

C:\Users\caelum\codigo>del saida.txt

C:\Users\caelum\codigo>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelum\codigo

28/12/2015  13:20    <DIR>        .
28/12/2015  13:20    <DIR>        ..
28/12/2015  12:34    <DIR>        java
                0 File(s)
                0 bytes
                3 Dir(s)          544.768 bytes free

C:\Users\caelum\codigo>
```

E agora vamos criar um novo arquivo, novamente usando o comando `echo` para imprimir o nosso texto e o `>` para salvá-lo em um arquivo:

```
echo Bem-vindo a Alura > saida.txt
```

Usando o `type` para mostrar o conteúdo do arquivo...

```
type saida.txt
```

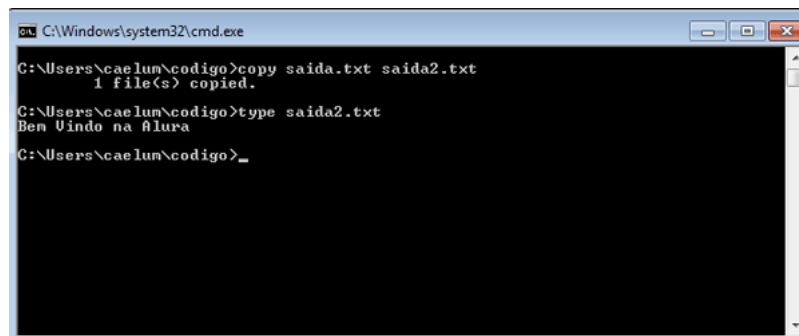
##Movendo e copiando arquivos

Gostamos tanto do nosso arquivo que queremos criar uma cópia. Isso é muito fácil de se fazer, e eu acho até mais fácil do que usando o mouse. Vamos lá:

```
copy saida.txt saida2.txt
```

Ótimo, o comando `copy` (*copy*) gera então uma cópia do arquivo. Vamos verificar o conteúdo desse arquivo:

```
type saida2.txt
```



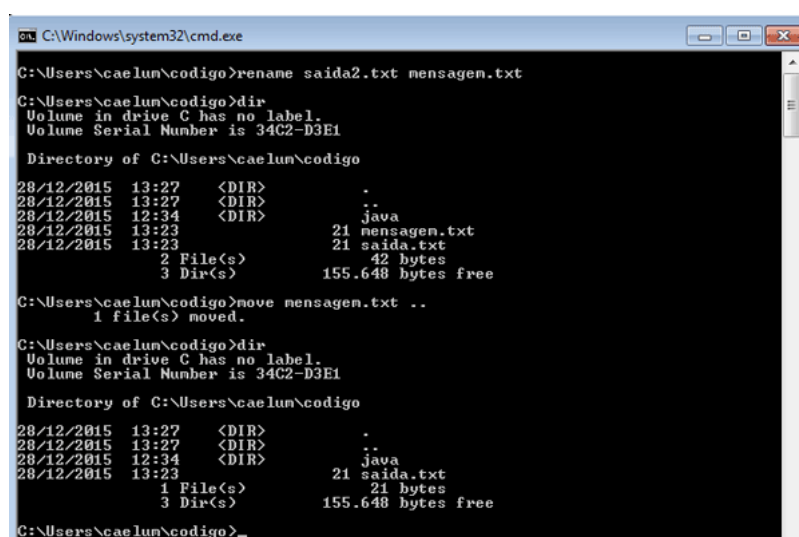
```
C:\Windows\system32\cmd.exe
C:\Users\caelum\codigo>copy saida.txt saida2.txt
1 file(s) copied.
C:\Users\caelum\codigo>type saida2.txt
Ben Vindo na Alura
C:\Users\caelum\codigo>_
```

Ok beleza, funcionou como esperado, mas eu não gostei do nome. Ainda bem que podemos renomear facilmente o arquivo:

```
rename saida2.txt mensagem.txt
```

O comando `rename` altera o nome dos arquivos! Aliás, o comando `rename` não só serve para arquivos como também para pastas. A mesma regra aplica para o comando `move` que move um arquivo ou pasta para outro lugar, por exemplo:

```
move mensagem.txt ..
```



```
C:\Windows\system32\cmd.exe
C:\Users\caelum\codigo>rename saida2.txt mensagem.txt
C:\Users\caelum\codigo>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelum\codigo
28/12/2015 13:27 <DIR>      .
28/12/2015 13:27 <DIR>      ..
28/12/2015 12:34 <DIR>      java
28/12/2015 13:23      21 mensagem.txt
28/12/2015 13:23      21 saida.txt
                2 File(s)      42 bytes
                3 Dir(s)      155.648 bytes free
C:\Users\caelum\codigo>move mensagem.txt ..
1 file(s) moved.
C:\Users\caelum\codigo>dir
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Users\caelum\codigo
28/12/2015 13:27 <DIR>      .
28/12/2015 13:27 <DIR>      ..
28/12/2015 12:34 <DIR>      java
28/12/2015 13:23      21 saida.txt
                1 File(s)      21 bytes
                3 Dir(s)      155.648 bytes free
C:\Users\caelum\codigo>_
```

Nesse comando movemos o arquivo `mensagem.txt` um diretório a cima, indicado pelo `..`.

Agora você já pode comemorar parcialmente. Esse conhecimento já vai ter ajudar muito. Melhor ainda se você fizer os exercícios.

No próximo capítulo veremos como mexer nas variáveis de ambiente, e aí sim você conseguirá configurar o Java, Ruby ou Python pela linha de comando!