



Livre Camp apresenta:

Semana da Calourada Livre

Interface de linha de comando

- Interação por meio de texto!
- É uma linguagem. Linguagem nos dá expressão
- Curva de aprendizado mais alta

Terminal em sistemas Unix-like

- Zilhões de opções
- Existe um padrão (POSIX)

```
dad@emachineubuntu: ~  
drwxr-xr-x 2 dad dad 4096 2011-07-30 10:31 dwhelper  
-rw-r--r-- 1 dad dad 179 2011-03-08 21:48 examples.desktop  
drwxr-xr-x 2 dad dad 4096 2011-05-17 21:26 Music  
-rw-r--r-- 1 dad dad 1262898 2011-07-22 19:39 newfilename.wmv  
-rw----- 1 dad dad 0 2011-07-09 20:06 nohup.out  
-rw-r--r-- 1 dad dad 434463 2011-08-03 21:54 out.ogv  
-rw-r--r-- 1 dad dad 39902309 2011-07-25 21:31 output.ps  
drwxr-xr-x 5 dad dad 4096 2011-05-19 11:26 Pictures  
drwxr-xr-x 2 dad dad 4096 2011-04-22 15:12 Podcasts  
drwxr-xr-x 2 dad dad 4096 2011-03-08 22:31 Public  
drwxr-xr-x 2 dad dad 4096 2011-07-10 18:48 roms  
-rw-r--r-- 1 dad dad 100000 2011-07-27 07:44 scripttest.sh  
-rw-r-xr-x 1 dad dad 112 2011-06-22 20:02 setauto.sh  
-rw-r--r-- 1 dad dad 6543 2011-04-26 23:36 shared-wallpapers.xml  
-r----- 1 dad dad 2805969 2011-05-14 17:08 Sources  
drwxr-xr-x 2 dad dad 4096 2011-03-08 22:31 Templates  
-rw-r--r-- 1 dad dad 204 2011-07-30 05:35 test.mp4  
drwxr-xr-x 2 dad dad 4096 2011-05-19 19:58 Ubuntu One  
-rw----- 1 dad dad 1440 2011-07-08 22:35 unison.log  
drwxr-xr-x 2 dad dad 4096 2011-03-08 22:31 Videos  
drwxrwxrwx 8 dad dad 4096 2011-08-01 20:27 VirtualBox VMs  
drwxr-xr-x 2 dad dad 4096 2011-07-30 10:16 xVideoServiceThief_downloads  
dad@emachineubuntu: ~$
```

```
1. ernes87lp@R2D2: ~/Sites/rubbers (zsh)  
[12:10:53] Finished 'default' after 965 ms  
[12:10:53] Starting 'default'...  
[12:10:53] Starting 'scripts'...  
  
Fetching Scripts Source Files...  
- resources/assets/js/admin/jquery.min.js  
- resources/assets/js/admin/bootstrap.min.js  
- resources/assets/js/admin/nicescroll/jquery.nicescroll.min.js  
- resources/assets/js/admin/icheck/icheck.min.js  
- resources/assets/js/admin/sweetalert.js  
- resources/assets/js/admin/lity.js  
- resources/assets/js/admin/custom.js  
  
Saving To...  
- ./public/admin/js/main.js  
  
[12:10:53] Finished 'default' after 543 ms  
[12:10:56] gulp-notify: [Laravel Elixir] Scripts Merged!  
[12:10:56] Finished 'scripts' after 2.85 s  
[12:10:56] Starting 'styles'...  
  
Fetching Styles Source Files...  
- resources/assets/css/admin/bootstrap.min.css  
- resources/assets/css/admin/fonts/css/font-awesome.min.css  
- resources/assets/css/admin/animate.min.css  
- resources/assets/css/admin/custom.css  
- resources/assets/css/admin/icheck/flat/green.css  
- resources/assets/css/admin/sweetalert.css  
- resources/assets/css/admin/lity.css
```

Comandos: ls

- Lista arquivos e diretórios

```
ls -l
```

mostra o conteúdo detalhado da pasta atual

```
ls /bin
```

mostra o conteúdo da pasta /bin

Comandos: ls

```
shum@sol:~$ ls -l
total 20
drwx----- 2 shum      staff    4096 Jan 16 22:04 Mail
drwx----- 3 shum      staff    4096 Jan 16 14:15 csc128
drwxr-xr-x  2 shum      staff    4096 Jan 13 16:42 public
drwxr-xr-x  2 shum      staff    4096 Jan 16 14:07 public_html
-rw-r--r--  1 shum      staff    628 Jan 15 20:04 verse
```

The diagram illustrates the components of the `ls -l` command output:

- file type**: Indicated by the first character of the permissions (e.g., `d` for directory, `-` for regular file).
- number of hard links**: The number following the file type (e.g., `2`).
- user (owner) name**: The user name following the number of hard links (e.g., `shum`).
- group name**: The group name following the user name (e.g., `staff`).
- size**: The file size in bytes (e.g., `4096`).
- date/time last modified**: The date and time the file was last modified (e.g., `Jan 16 22:04`).
- filename**: The name of the file (e.g., `Mail`).
- permissions**: The permission string (e.g., `drwxr-xr-x`), which is further detailed as:
 - user permissions**: The first three characters (e.g., `drwx`).
 - group permissions**: The next three characters (e.g., `r-x`).
 - other (everyone) permissions**: The last three characters (e.g., `r-x`).

Legend for permissions:

- r**: readable
- w**: writeable
- x**: executable

Comandos: ls

- Lista arquivos e diretórios

```
ls -a
```

mostra todos os arquivos da pasta atual
(incluindo ocultos)

arquivos ocultos em *nix tem um carácter
ponto no início. ex.: .config .etc

Comandos: ls

- Lista arquivos e diretórios

```
ls *.txt
```

o símbolo * é o coringa. nesse caso,
qualquer arquivo que terminar com .txt será
mostrado

coringas funcionam com vários comandos!

Comandos: ls

Exercício

Liste todos os arquivos do **/bin** que comecem com **get**

Comandos: ls

Exercício

Liste todos os arquivos do **/bin** que comecem com **get**

```
ls /bin/get*
```

Comandos: ls

- Como aprender mais opções?

```
ls --help
```

```
man ls
```

Comandos: cd

- Muda de diretório

```
cd /home
```

vai para a pasta /home

```
cd ~
```

vai para a pasta /home/usuário

Comandos: cd

- Muda de diretório

`cd -`

volta para a pasta anterior

`cd ..`

vai para a pasta superior

Comandos: cd

Exercício

Vá até a pasta **/bin**, liste todos os arquivos e volte para a pasta original

Comandos: cd

Exercício

Vá até a pasta **/bin**, liste todos os arquivos e volte para a pasta original

```
cd /bin; ls; cd -
```

Comandos: touch

- Inicialmente foi criado pra alterar data de arquivos, mas descobrimos que ele é mais útil criando eles.

```
touch oie.txt
```

cria o arquivo oie.txt

```
touch /tmp/oie.txt
```

cria o arquivo oie.txt na pasta /tmp

Comandos: mkdir

- Cria diretórios (make dir)

```
mkdir ~/pasta
```

cria a **pasta** na pasta do usuário

```
mkdir -p eu/ainda/nao/existo
```

cria todas as pastas recursivamente

Comandos: mkdir

Exercício

Crie uma pasta chamada **teste** na pasta do usuário e um arquivo dentro dela chamado **testearquivo**

Comandos: mkdir

Exercício

Crie uma pasta chamada **teste** na pasta do usuário e um arquivo dentro dela chamado **testearquivo**

```
mkdir ~/teste
```

```
touch ~/teste/testearquivo
```

Comandos: cp

- Copia arquivos

```
cp arquivo arquivo_copia
```

copia o **arquivo** para o **arquivo_copia**

```
cp -r diretorio diretorio_copia
```

copia recursivamente todo o conteúdo de
diretorio para **diretorio_copia**

Comandos: cp

- Cópia arquivos

```
cp arquivo /pasta/
```

copia o **arquivo** para a **pasta**, gerando uma cópia de **arquivo** em **/pasta/arquivo**

```
cp arquivo* /pasta/
```

copia todos os arquivos que começam com **arquivo** para **pasta**

Comandos: cp

Exercício

Crie um **arquivo** e uma **pasta**, depois copie o arquivo para a pasta

Comandos: cp

Exercício

Crie um **arquivo** e uma **pasta**, depois copie o arquivo para a pasta

```
touch arquivo  
mkdir pasta  
cp arquivo pasta
```

