

## ■ Criar uma nova aplicação web

1. **Aceder à pasta** onde pretendem que a nova aplicação seja criada. No ambiente de desenvolvimento, depois do processo de autenticação com as credenciais “<nome\_utilizador>/“cm”, deve ser executado o comando “**cd myApps**”, seguido de “**enter**”. Note-se que os **caracteres “<” e “>” não devem ser utilizados** no comando e pretendem, apenas, indicar a **necessidade de substituir** aquela expressão pelo **nome da nova aplicação** (preferencialmente sem acentos, caracteres especiais e espaços em branco).
2. Irá ser utilizado o **composer** (gestor de dependências para PHP) para criar uma nova aplicação, através do comando “**composer create-project laravel/laravel <nome\_da\_nova\_aplicação>**”, seguido de “**enter**”.
3. Deverá, **depois da execução completa** do comando (cerca de 5 minutos, dependendo das características físicas do computador de cada um e dos recursos alocados à máquina virtual), passar a estar **disponível uma pasta** com o nome **<nome\_da\_nova\_aplicação>** (pode ser utilizado o comando “**ls -al**” para verificar).
4. É necessário, agora, alterar as **permissões** de algumas das **pastas** (e sub-pastas) **criadas**. Assim, depois de **entrar** na pasta da **aplicação** (comando “**cd <nome\_da\_nova\_aplicação>**”), ter-se-á de aceder à pasta “**storage**”, de forma a alterar as **permissões** das suas sub-pastas (comando “**cd storage**”). O **comando** a usar para alterar as permissões das pastas “**app**”, “**framework**” e “**logs**” é “**chmod 775 \* -R**”. Este comando atribuirá **permissões de escrita** a utilizadores pertencentes ao **grupo do servidor Apache (www-data)** a todas estas pastas e sub-pastas que, eventualmente, contenham. Outra pasta que terá de ter as suas **permissões alteradas** é a “**cache**”: sub-pasta da pasta “**bootstrap**”. Para isso, será necessário **sair** da **pasta atual** - “**storage**” - através do comando “**cd ..**”. De seguida, aceder à pasta “**bootstrap**” (comando “**cd bootstrap**” e executar o comando “**chmod 775 cache -R**”).
5. No que respeita às permissões, falta, apenas, estabelecer que a aplicação criada **pertence** ao **utilizador** “<nome\_utilizador>” e ao **grupo** do **utilizador** do **servidor Apache (www-data)**. Assim, é necessário sair da pasta atual para a que alberga todas as aplicações Laravel existentes no ambiente de desenvolvimento, através do comando “**cd ..**”, seguido de outro “**cd ..**”. A **pasta atual** deve ser a “**myApps**”. Então, deverá ser assumido o papel de **super-utilizador**, através do comando “**su -**”, seguido da password “**cm**”. Para **assegurar a localização correta**, executar o comando “**cd /home/<nome\_utilizador>/myApps**”. De seguida, o comando a executar é “**chown <nome\_utilizador>:www-data <nome\_da\_nova\_aplicação> -R**”, seguido de “**enter**”. Deve-se voltar, agora, ao utilizador “<nome\_utilizador>”, executando “**CTRL+D**”.
6. É necessário verificar se a **aplicação** está **acessível** via **browser**. Contudo, antes disso, terá de ser **apagado o conteúdo atual** da pasta “**public\_html**” do utilizador e **colocar** lá o **conteúdo** da pasta “**public**” da **nova instalação**. Para o efeito, deve ser executado o comando “**rm /home/<nome\_utilizador>/public\_html/laravel/\* -R**”, seguido de “**enter**”. É, ainda, necessário, apagar o **ficheiro** “.htaccess” que se encontra nessa pasta, através do comando “**rm /home/<nome\_utilizador>/public\_html/laravel/.htaccess**”, seguido de “**enter**”. Deve-se, agora, **aceder** à pasta “**public**” dentro da **pasta da nova aplicação**. Assumindo que **não houve alteração de posição** desde o último comando, deve ser necessário, apenas, executar o comando “**cd <nome\_da\_nova\_aplicação>**”, seguido de “**cd public**”. Basta, então, **copiar** o **conteúdo** desta pasta para a “**public\_html/laravel/**”, através do comando “**cp \* /home/<nome\_utilizador>/public\_html/laravel/ -R**”, seguido de “**enter**”. É necessário, ainda, copiar o ficheiro “.htaccess”. O comando é similar ao anterior: “**cp .htaccess /home/<nome\_utilizador>/public\_html/laravel**”, seguido de “**enter**”.
7. De forma a garantir que a aplicação é inicializada corretamente e funciona como esperado, torna-se necessário alterar o ficheiro “**index.php**”, da pasta “**public\_html/laravel**”. Esta alteração pode (e deve) ser feita com o **editor** de **texto**, modificando a linha “**require \_\_DIR\_\_.'../../myApps/<nome\_da\_nova\_aplicação>/vendor/autoload.php';**” e “**\$app = require\_once \_\_DIR\_\_.'../../myApps/<nome\_da\_nova\_aplicação>/bootstrap/app.php';**”.
8. Num **browser**, escrevendo o endereço IP da máquina virtual seguido de “**/laravel**”, deve ser **apresentada** a página por **omissão** da **framework Laravel**, o que significa que a criação da nova aplicação foi **bem-sucedida**.

**Nota:** Existem diversas formas de realizar esta tarefa, bem como alguns dos pontos enumerados. É de referir, sobretudo, a hipótese alvitrada como mais correta no contexto de um servidor partilhado (alojamento contratado), de alterar a execução da aplicação da pasta “**public**” para a pasta “**public\_html**”.

### Súmula de comandos do sistema operativo Linux:

ls	Listar o conteúdo da pasta. O modificador “a” exhibe, também, os ficheiros escondidos – cujo nome começa com “.” - e o modificador “l”, apresenta o resultado sob a forma de uma lista, com detalhes como o tamanho e as permissões de cada entrada.
cd	Altera a pasta de trabalho.
chmod	Permite a alteração das permissões de um ficheiro ou pasta. Existem três tipos de permissões – leitura (R), escrita (W) e execução (X) - para três atores do sistema operativo: o utilizador que é

proprietário do recurso, o grupo a que o recurso pertence e todos os restantes utilizadores, respetivamente. Assim, “0” indica que não existe permissão para uma dada operação e “1” o oposto. Desta forma, “101” significa que dado ator tem permissão de leitura e de execução. Outro exemplo completo é “111 101 101”. Neste, o utilizador tem permissões completas – leitura, escrita e execução -, enquanto o grupo tem, apenas, de leitura e de execução, bem como os restantes utilizadores.

Deve-se ter em atenção que, tratando-se de numeração binária, “111” pode ser convertido para o decimal 7. Assim, permissões “777” significam “111 111 111”, ou seja, permissões completas para todos os utilizadores. Este tipo de permissão é considerada insegura, só deve ser utilizada em situações muito excecionais e nunca em servidores em produção.

chown	Por último, o modificador “-R” no final deste e de outros comandos significa, normalmente, um processo recursivo. Ou seja, pretende-se aplicar o comando a uma pasta e a todo o seu conteúdo. Este comando altera a propriedade de um ficheiro ou de uma pasta. De facto, em Linux, qualquer ficheiro ou pasta são propriedade de um utilizador e de um grupo (e.g. utilizador:grupo). Por exemplo, o grupo no qual é executado o servidor web Apache é, por omissão, o “www-data”. Isto significa que se um dado ficheiro pode pertencer ao utilizador “utilizador” e ao grupo “www-data”, o que, mediante as permissões concedidas, poderá permitir um conjunto de operações necessárias à boa execução de uma dada aplicação web.
rm	Permite apagar um ficheiro ou uma pasta (e todo o seu conteúdo, caso se utilize o modificador “-R”).
vim, joe ou nano	Trata-se de editores de texto, em consola, que possibilitam a edição de ficheiros como, por exemplo, os de configuração da <i>framework</i> Laravel. Serão objeto de explicação mais detalhada nas aulas práticas e laboratoriais da unidade curricular.