

## ■ Instalação do Breeze (gestão de utilizadores)

1. De acordo com a documentação do pacote Breeze ([aqui](#)), é necessário proceder à criação das tabelas necessárias, na base de dados, às valências associadas ao processo de autenticação. Para esse fim, na pasta raiz da aplicação web - “myApps/<nome\_da\_nova\_aplicação>” - deve ser executado o comando de consola “**php artisan migrate**”, seguido de “**enter**”. As tabelas por omissão de uma nova aplicação web e as relativas ao processo de autenticação serão **criadas** na **base de dados**. Caso não esteja, ainda, configurada a base de dados na aplicação web, deve ser seguida a Receita “**Configurar uma base de dados MySQL**”.
2. Na raiz pasta da aplicação (<nome\_da\_nova\_aplicação>), é necessário, enquanto **utilizador regular**, instalar o pacote **laravel/breeze**, através do comando “**composer require laravel/breeze --dev**”, seguido de “**enter**”. Os recursos alocados à máquina virtual são importantes nestes processos.
3. Para proceder à publicação dos **layouts**, **rotas**, **controladores** e outros **recursos**, é necessário executar o comando “**php artisan breeze:install**”, seguido de “**enter**”.
4. Será apresentado um menu para **selecionar** a **tecnologia** que será utilizada para apresentar o **frontend** ao utilizador. Por omissão, será a **Blade**. Contudo, poder-se-á optar, por exemplo, pelo React.JS ou Vue. Para esta receita, a opção por omissão – “**0**” – deve ser selecionada.
5. É ainda perguntado se é pretendido o **suporte** ao **dark mode**, hoje uma opção muito em voga por uma parte significativa dos utilizadores de aplicações web. Para esta receita a escolha é **não suportar** este **modo**, através da opção “**no**”.
6. Por último, deve ser selecionada a tecnologia que permite a aplicação de testes à aplicação. **Manter-se-á** a opção por **omissão**: PHPUnit.
7. Procede-se à compilação dos recursos gráficos, executando os comandos “**npm install**”, seguido de “**npm run build**”. Os recursos compilados (e.g. js, css) serão **armazenados** numa pasta “**build**”, dentro da pasta “**public**”, disponível na raiz da aplicação.
8. É necessário ter acesso à pasta “**build**” na pasta “**public\_html/laravel/**”, para que os recursos possam ser utilizados na apresentação da aplicação aos utilizadores. Pode, para o efeito, ser criada, na pasta “**public\_html/laravel/**”, uma **ligação simbólica** à pasta “**build**”, existente na pasta “**public**” da aplicação e que é **atualizada** a cada **compilação**. A ligação simbólica significa que, caso se lhe aceda dentro de “**public\_html/laravel**” está-se, efetivamente, a ter acesso à **pasta original**. Para o conseguir, é necessário executar, dentro da pasta “**public\_html/laravel**”, o comando “**ln -s ../../myApps/<nome\_da\_nova\_aplicação>/public/build build**”. Caso o processo descrito no número 7 desta receita também produza uma pasta “**vendor**”, deve ser criada, de forma semelhante, a respetiva ligação simbólica.
9. Antes de **confirmar** o **resultado** da **instalação** deste sistema de **gestão de utilizadores** através do endereço “http://<ip>/laravel/login” (Figura 1), terá de ser **modificada** a **configuração** do **servidor Apache**.

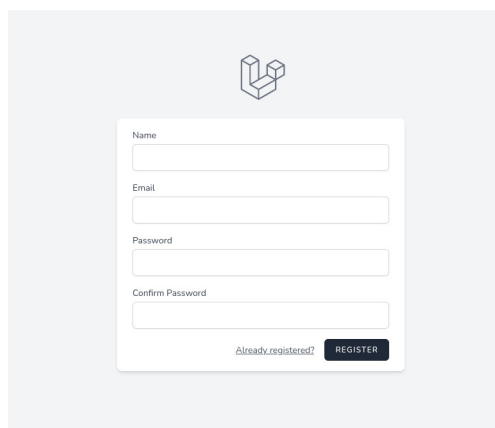


Figura 1 – Formulário de registo do gestor de utilizadores Breeze.

10. A alteração na configuração do servidor Apache não está **diretamente relacionada** com esta **receita** e seria habitualmente pedida ao **gestor** do **alojamento** da aplicação web. Contudo, é **útil** perceberem o processo completo que **suporta** o **funcionamento** desde e de outros módulos já desenvolvidos. Esta modificação terá de ser feita, apenas, uma vez. Procede-se, então, à **ativação** do **módulo “rewrite”** do servidor Apache. Este módulo é o **responsável** por **reescrever** ou **redirecionar** pedidos apresentados ao servidor Apache.
11. Após a autenticação como super-utilizador no ambiente de desenvolvimento – username “root” e password “potter” – acede-se à pasta de configuração de módulos do servidor Apache, através do comando “**cd etc/apache2/mods-enabled**”, seguido de “**enter**”.

12. Esta pasta possui ligações simbólicas a todos os módulos ativos na configuração do servidor Apache, disponíveis na pasta “**/etc/apache2/mods-available**”. Para ativar o módulo “rewrite”, terá de ser executado o comando “**ln -s ../mods-available/rewrite.load rewrite.load**”, seguido de “enter”.
13. O servidor Apache terá de ser reiniciado através do comando “**/etc/init.d/apache2 restart**”, seguido de “enter”. Este passo **conclui** a **alteração** da **configuração** do servidor.
14. Agora, é necessário verificar se o ficheiro “.htaccess”, disponível na pasta “**public\_html/laravel**”, possui as linhas:

```
<IfModule mod_negotiation.c>
    Options -MultiViews -Indexes
</IfModule>
```

15. Isto pode ser feito, após uma autenticação bem-sucedida como utilizador regular (<nome\_utilizador>), acesso à pasta em causa através do comando “**cd public\_html/laravel**” caso a autenticação tenha sido realizada na pasta base do utilizador, e o recurso a um editor de texto (e.g. “**vim .htaccess**”). Se as linhas indicadas no número 14 **existem**, terão de ser **apagadas** (tecla “a” para início da edição do ficheiro, tecla “ESC”, seguida da escrita de “:wq!” para gravar e sair do editor). Esta alteração também só terá de ser efetuada uma vez.
16. O último passo consiste em **acrescentar** uma linha de **configuração** ao ficheiro “.env”, localizado na raiz da aplicação. Assim, após execução do comando “**cd myApps/exampleApp**”, procede-se à edição do ficheiro recorrendo, por exemplo, ao “**vim**”. Deve ser adicionada a variável de ambiente “**ASSET\_URL**”, com o valor “**/laravel**” (nome da pasta contida dentro da “public\_html”). Esta alteração é **essencial** para indicar à aplicação Laravel onde pode **encontrar** os **recursos** (e.g. css e js) necessários para as interfaces e é apresentada na Figura 2.

```
DevDebianOS [Em execução] - Oracle VM VirtualBox
Ficheiro Máquina Ver Entrada Dispositivos Ajuda
APP_ENV=local
APP_KEY=base64:FPK+hLjGag1fThRqF/nZTSnDmcrCL8j0f8DQdtgkY=
APP_DEBUG=true
APP_URL=/laravel
ASSET_URL=/laravel
LOG_CHANNEL=stderr
LOG_REPLICATION_CHANNEL=null
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=trabalho
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
MEMOACHED_HOST=127.0.0.1
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
MAIL_MAILER=smtp
MAIL_HOST=mailpit
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
ASSET_URL=""
```

Figura 2 – Modificação do ficheiro de configuração da aplicação Laravel (.env).

## Súmula de comandos do sistema operativo Linux:

composer	Trata-se de uma ferramenta que permite a gestão de dependências em PHP, nomeadamente através da indicação das bibliotecas necessárias a um dado projeto e à sua consequente instalação e/ou atualização.
npm	É o gestor de pacotes para a plataforma Node.js: permite-lhe encontrar os módulos necessários à sua operação numa dada aplicação e gere conflitos de dependências entre eles. A operação “install” instala módulos e as suas dependências. Quanto à operação “build”, constrói os recursos necessários ao funcionamento da aplicação web (e.g. css e js).
ln	Permite a criação de uma ligação para um ficheiro ou uma pasta. Por norma, utiliza-se com a opção “s”, que cria uma ligação simbólica (i.e. um atalho).