

■ Configurar uma base de dados MySQL

1. Após a criação de uma nova aplicação web em Laravel é geralmente **necessário configurar** a ligação a uma **base de dados** para continuar o processo de desenvolvimento.
2. Assume-se a **criação** de uma **base de dados** no servidor MySQL instalado no sistema operativo do ambiente de desenvolvimento, recorrendo à ferramenta de gestão de bases de dados **Adminer**. Existe uma base de dado já criada, com o intuito de ser utilizada na fase de aprendizagem da Framework Laravel (**“workshop”**). Tem acesso **completo** garantido ao **utilizador** <nome_utilizador> (e.g. “ana”), com a palavra-passe “potter”.
3. Para esta receita, ir-se-á criar uma **nova** base de dados: **“laravel”**.
4. Assim, é necessário aceder à ferramenta de gestão de bases de dados Adminer através do **browser**, colocando o endereço IP da máquina virtual, seguido de “/adminer” (e.g. “192.168.56.108/adminer”).
5. Segue-se a **autenticação** como **super-utilizador**: username “root” e password “potter”.
6. De seguida, selecciona-se a opção **“Criar Base de dados”** (Figura 1).



Figura 1 - Página principal da ferramenta Adminer, após autenticação como super-utilizador.

7. O nome da nova base de dados será **“laravel”**. Segue-se a seleção da opção **“guardar”**.
8. Após a criação **bem-sucedida** da nova base de dados, torna-se necessário **garantir** que um utilizador regular possui não só o **acesso**, como as **permissões** necessárias para com ela **interagir**. Assim, selecciona-se a opção **“privilegios”**, apresentada na Figura 2.
9. Neste exemplo ir-se-á utilizar um **utilizador** já **existente** – “ana” – que já possui acesso e permissões na base de dados criada para o workshop. Após introduzir o nome do utilizador, seleccionar a opção **“Modificar”**.



Figura 2 - Seleção da opção “Privilegios” após a criação bem-sucedida de uma nova base de dados.

10. No interface que surge, deve-se escrever, na **coluna livre** (última), **“laravel.*”**, bem como **seleccionar** a **primeira checkbox**, indicando que se pretende que o utilizador tenha privilégios completos (Figura 3).

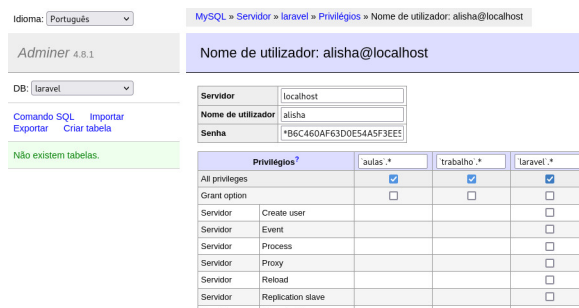
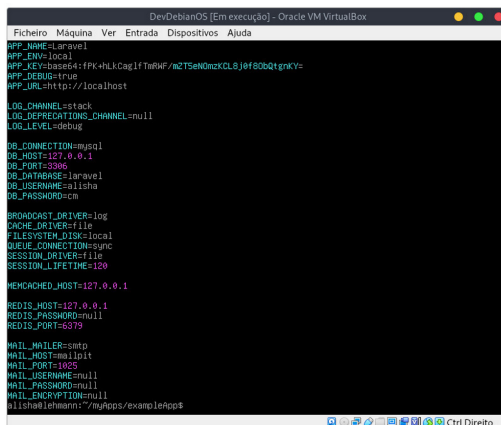


Figura 3 - Modificação dos privilégios do utilizador para a base de dados criada – “laravel”, atribuindo privilégios completos.

11. Por último, selecciona-se, no fundo do interface, a opção **“Guardar”**.

12. Existe, na pasta raiz da aplicação Laravel, um **ficheiro “.env”** que contém as configurações da aplicação web. Para lhe aceder, é necessário entrar na pasta da aplicação. Considerando uma autenticação bem-sucedida acabada de realizar com as credenciais do **utilizador regular** - <nome_utilizador> - acede-se à pasta pretendida através da execução do comando “**cd myApps/<nome_da_nova_aplicação>**” (e.g. “cd myApps/exampleApp”).
13. Recorrendo a um editor de texto da linha de comandos do sistema operativo (e.g. vim, joe, nano) edita-se o ficheiro, através do comando e.g. “**vim .env**”, seguido de “**enter**”.
14. Entre as diversas configurações disponíveis (Figura 4), chama-se a atenção para o nome da aplicação web, para o seu URL e para os elementos que configuram a ligação à base de dados – respetivamente, “APP_NAME”, “APP_URL”, “DB*”.



```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:FKxHkCag1fThRf/n2T5eN0mZKLBj0f80bqgKvY
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

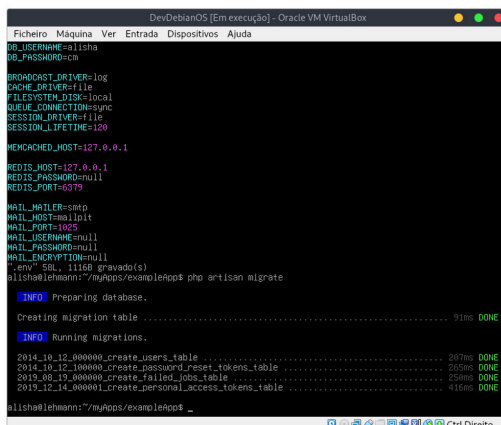
MEMCACHED_HOST=127.0.0.1

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
MAIL_ENCRYPTION=null
```

Figura 4 - Configurações da aplicação web, disponíveis no ficheiro “.env”.

15. Embora não sejam centrais a esta receita, é **boa prática** proceder à alteração do nome da aplicação e do seu URL assim que esta seja criada. Enquanto que o nome da aplicação poderá ser simplesmente substituído, no que respeita ao URL, terá de ser tida em conta a **organização** do ambiente de desenvolvimento. Assim, deve-se **modificar** o URL existente para “/laravel”. No editor “vim”, carregar na tecla “a” para começar a editar. Este passo terá de ser adaptado a cada **contexto**, fora do ambiente de desenvolvimento.
16. Procede-se, então, à configuração da ligação à base de dados, alterando “DB_DATABASE” para “**laravel**”, “DB_USERNAME” para “<nome_utilizador>” (e.g. “ana”) e “DB_PASSWORD” para “**potter**”. Finaliza-se a operação guardando as alterações (no editor “vim”, carregar em “:”, seguido de “wq!” e “enter”).
17. Para **testar** se a ligação à base de dados é **bem-sucedida**, basta executar o comando “**php artisan migrate**” na pasta raiz da nova aplicação - myApps/<nome_da_nova_aplicação> - (Figura 5), o que irá resultar na **criação de tabelas**, definidas por omissão, em cada nova aplicação Laravel (maioritariamente relacionadas com o registo e autenticação de utilizadores). Depois, basta aceder à base de dados “**laravel**”, via Adminer, e **visualizar** as **tabelas** criadas **automaticamente** (Figura 6).



```
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_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
MAIL_ENCRYPTION=null

.env 58L, 1168 gravado(s)
alishahlehmann~/myApps/exampleApp$ php artisan migrate
[INFO] Preparing database.
Creating migration table ..... 91ms DONE
[INFO] Running migrations.
2014-10-12-000000.create_users_table ..... 207ms DONE
2014-10-12-100000.create_password_reset_tokens_table ..... 265ms DONE
2015-08-19-000000.create_failed_jobs_table ..... 259ms DONE
2015-12-14-000001.create_personal_access_tokens_table ..... 416ms DONE
alishahlehmann~/myApps/exampleApp$
```

Figura 5 - Resultado da execução bem-sucedida do comando “php artisan migrate” para testar a ligação à nova base de dados.

Tabela	Motor	Colação	Tamanho de dados	Tamanho de índice	Espaço Livre	Incremento Automático
failed_jobs			?	?	?	?
migrations			?	?	?	?
password_reset_tokens			?	?	?	?
personal_access_tokens			?	?	?	?
users			?	?	?	?
5 no total	InnoDB	utf8mb4_general_ci				

Figura 6 - Tabelas criadas na nova base de dados, após a execução do comando “php artisan migrate”.

18. De forma a manter a base de dados igual ao que se encontrava antes do teste executado no número 17 desta receita, basta executar o comando “**php artisan migrate:rollback**”. Como resultado, as tabelas anteriormente criadas serão apagadas, mantendo-se, apenas, uma tabela “**migrations**”, onde é **registada** qualquer **operação realizada** na base de dados pela aplicação.
19. Porém, também existe um ficheiro de configuração da base de dados localizado na pasta “**config/database.php**” (Figura 7), considerando como ponto de partida a pasta raiz da aplicação Laravel. Como diferenciar? Em primeiro lugar, a Framework Laravel **prioriza variáveis de ambiente** (i.e. as definidas em ficheiros “.env”) sobre outras. Acresce, ainda, que o ficheiro “database.php” possui um conjunto de configurações adicionais da base de dados que **não estão presentes** noutra localização (e.g. **charset**). Estas configurações, embora importantes, são consideradas **não-críticas** e podem estar disponíveis no código-fonte da aplicação. Como boa prática, **informação crítica** (e.g. username, password) deve ser **definida** no **ficheiro de configuração da aplicação** (“.env”) e a restante pode ser modificada neste ficheiro.

```

'mysql' => [
    'driver' => 'mysql',
    'url' => env('DATABASE_URL'),
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '3306'),
    'database' => env('DB_DATABASE', 'forge'),
    'username' => env('DB_USERNAME', 'forge'),
    'password' => env('DB_PASSWORD', ''),
    'unix_socket' => env('DB_SOCKET', ''),
    'charset' => 'utf8mb4',
    'collation' => 'utf8mb4_unicode_ci',
    'prefix' => '',
    'prefix_indexes' => true,
    'strict' => true,
    'engine' => null,
    'options' => extension_loaded('pdo_mysql') ? array_filter([
        PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
    ]) : [],
],

'pgsql' => [
    'driver' => 'pgsql',
    'url' => env('DATABASE_URL'),
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '5432'),
    'database' => env('DB_DATABASE', 'forge'),
    'username' => env('DB_USERNAME', 'forge'),
    'password' => env('DB_PASSWORD', ''),
    'charset' => 'utf8',
    'prefix' => '',
    'prefix_indexes' => true,
    'schema' => 'public',
    'sslmode' => 'prefer',
],

```

Figura 7 - Ficheiro de configuração da base de dados localizado em “config/database.php”.

Nota: A ferramenta “artisan” possibilita a **interação** com a *framework* Laravel através da **linha de comandos**. Terá de ser executado na **pasta raiz** de cada aplicação (e.g. “myApps/exampleApp”) e **disponibiliza** uma série de **comandos** muito **úteis** para o **desenvolvimento** de uma aplicação.