





Autor: Tiago Motta

Título: Documentação dos ambientes de desenvolvimento

Recife, 28 de Abril de 2019

Orientador: Manoel Rodolfo

Descrição: Este documento tem por objetivo auxiliar os programadores a criar o ambiente de desenvolvimento com as versões e plugins corretos, bem como também montar a infraestrutura do mesmo.

Sumário

Documentação Front-end	4
Documentação Back-end.....	5
Documentação Aplicativo Mobile	7
Instalação MySQL Server Ubuntu 18.04	9
Documentação de requisições	13
Diagrama de banco de dados.....	16

Documentação Front-end

Angular 6.1

[The Angular Framework for Web](#)

Pré-requisitos

- Html5
- Css3
- Sass
- Angular 6.1
- NodeJS >= 12.2

Instalação

Ambiente Windows 10

1. [Instalar nodeJS](#)
2. Iniciar o cmd
3. npm install -g typescript
4. npm install -g @angular/cli
5. ng -v (para verificar se tudo deu certo)
6. Com o cmd aberto no diretório do projeto, use o comando 'ng serve '
7. Abra o navegador e digite '<http://localhost:4200>'

Inicializando projeto

1. Abra o cmd e aponte o caminho para a pasta do projeto
2. digite 'ng serve '
3. Abra o navegador e digite '<http://localhost:4200>'

Documentação Back-end

Laravel 5.5

[The PHP Framework for Web Artisans](#)

Pré-requisitos

Apache

- MySQL Server
- PHP >= 7.1
- Composer

Documentação

1. [Documentação de Requisições](#) – Página 12
2. [Diagrama do Banco](#) – Página 15

Instalação

Ambiente Ubuntu 18.04

Composer:

1. Instale utilizando o comando no bash `sudo apt install composer`
2. Para verificar se instalou corretamente digite no bash `composer`

Apache:

1. Instale o apache utilizando o comando no bash `sudo apt install apache2`

MySQL Server:

1. [Instalar MySQL Server](#) – Página 9
2. Crie um novo usuário para acessar o banco acessando o bash e digitando `sudo mysql`
3. Em seguida digite para criar um novo usuário com o nome admin `CREATE USER 'admin'@'localhost' IDENTIFIED BY 'informe_a_senha_aqui';`
4. Para conceder os todos privilégios para o usuário admin digite `grant all privileges on *.* to 'admin'@'localhost';`

Instalar dependências secundárias:

```
sudo apt install -y php php-mysql php-pgsql libapache2-mod-php php-curl php-json  
php-fpm php-xml php-mbstring php-gd php-intl php-imap
```

Ambiente Windows

Inicializando o projeto

1. Digite no terminal `composer install` para instalar as dependências do projeto.
2. Copie o arquivo **.env.example** e crie o um novo arquivo **.env**.
3. Rode o comando `php artisan key:generate` para criar a key do projeto no arquivo **.env**.
4. Após isso configure o arquivo **.env** com as informações do banco.
5. Rode o comando `php artisan migrate` para rodar as migrations e criar as tabelas do banco.
6. Rode `php artisan db:seed` para popular o banco com instâncias de teste

Instalação do Passport

1. Rode o comando `php artisan passport:install --force`
2. E verifique a tabela **OauthClients** caso exista dois usuários tudo foi instalado corretamente.

Documentação Aplicativo Mobile

Delphi Community Edition

Pré-requisitos

- Instalador Delphi Community Edition
- Android SDK

Instalação

1. Acessar o site do [Delphi](#).
2. Clicar em “Get Community Edition Free” e realizar o cadastro ou login.
3. Realizar o download do instalador.
4. Abrir o instalador aceitando os termos de adesão.
5. Selecionar a opção “I already have a product serial number” e clicar em “Install”.
6. Ao abrir o Delphi, inserir a chave que foi recebida por e-mail e os dados de acesso e clicar em “Register”.
7. Marcar a opção de desenvolvimento Android.

Preparando o ambiente de desenvolvimento Android

1. Após marcar a opção de desenvolvimento Android, selecionar a opção “Android SDK” e clicar em “Install”.
2. Navegar até a pasta que o Delphi foi instalado em sua máquina e abrir a pasta “Android SDK” e executar o arquivo “SDK Manager.exe”
3. Selecionar as opções “Android SDK Tools”, “Android SDK Platform-tools”, “Android SDK Build-Tools”, “SDK Platform” e “Google USB Driver” e pressionar “Install 5 packages”.
4. Na pasta “Android SDK”, dentro da pasta de instalação do Delphi Community, abrir a pasta “platform-tools” e copiar o seu caminho.
5. No menu iniciar do Windows, digitar “Variáveis de Ambiente”, clicar em “Editar” na opção “Path”.
6. Pressionar “Novo”, colar o caminho que foi copiado e pressionar “Ok”.

Importar Android SDK

1. Abrir o Delphi Community Edition.
2. Dentro do programa, realizar o caminho “Tools > Options > SDK Manager”.
3. Deletar a versão Android SDK.
4. Clicar em “Add”.
5. Na opção “Select a platform” selecionar “Android”.
6. Em “Select an SDK version”, selecionar “Add New”.
7. No primeiro e segundo campo, colocar o caminho da pasta “C:\Users\Public\Documents\Embarcadero\Studio\Android SDK” dentro da pasta de instalação do Delphi.
8. Pressionar “Next” e após isso em “Finish”.
9. Na última janela aberta pressionar “Ok”.

Habilitando USB Debugging no celular Android

1. Abra o aplicativo “Configurações”.
2. No Android 8.0 ou em versões superiores, selecione “Sistema”.
3. Role a tela e selecione “Sobre o telefone”.
4. Role a tela e toque 7 vezes em “Versão do Android”.
5. Retorne à tela anterior “Configurações” e role até o final.
6. Você deve encontrar agora as “Opções do Desenvolvedor”.
7. Nas “Opções do Desenvolvedor” ativar a opção “USB Debbing”
8. Na pasta
C:\Users\Public\Documents\Embarcadero\AndroidSDK\extras\google\usb_driver”, clicar com mouse do lado direito em “android_winusb” e selecionar “Install”.
9. Agora é possível conectar o dispositivo Android ao computador pelo cabo USB.

Instalação MySQL Server Ubuntu 18.04

A versão curta da instalação é simples: atualize seu índice de pacotes, instale o pacote mysql-server, e então execute o script de segurança que vem incluído.

```
$sudo apt update
```

```
$sudo apt install mysql-server
```

```
$mysql_secure_installation
```

Este tutorial irá explicar como instalar o MySQL versão 5.7 em um servidor Ubuntu 18.04. Contudo, se você estiver querendo atualizar uma instalação MySQL existente, você pode ler esse guia de atualização do MySQL 5.7 em vez disso.

Pré-requisitos:

Para seguir esse tutorial, você vai precisar de:

- Um servidor Ubuntu 18.04 configurado seguindo esse guia de configuração inicial de servidor, incluindo um usuário com sudo que não seja root e um firewall.

Passo 1 — Instalando o MySQL

No Ubuntu 18.04, somente a última versão do MySQL está incluída no repositório de pacotes APT por padrão. No momento em que escrevo, ela é a MySQL 5.7.

Para instalá-la, atualize o índice de pacotes em seu servidor e instale o pacote padrão com apt:

```
$sudo apt update
```

```
$sudo apt install mysql-server
```

Isso irá instalar o MySQL, mas não solicitará que você configure uma senha ou faça quaisquer outras alterações de configuração. Como isso deixa a sua instalação do MySQL insegura, vamos abordar isso a seguir.

Passo 2 — Configurando o MySQL

Para novas instalações, você vai querer executar o script de segurança que está incluído. Isso altera algumas das opções padrão menos seguras para coisas como logins de root e usuários de exemplo. Em versões mais antigas do MySQL, você precisava inicializar o diretório de dados manualmente também, mas isso é feito automaticamente agora.

Execute o script de segurança:

```
$sudo mysql_secure_installation
```

Isto irá levá-lo através de uma série de prompts onde você poderá realizar algumas alterações nas opções de segurança da sua instalação do MySQL. O primeiro prompt irá perguntar se você quer configurar o Plugin Validate Password, que pode ser utilizado para testar a força de sua senha do MySQL. Independentemente de sua escolha, o próximo prompt será para configurar a senha do usuário root do MySQL. Entre e então confirme uma senha segura de sua escolha.

A partir daí, você pode pressionar Y e então ENTER para aceitar as respostas padrão para todas as questões subsequentes. Isso irá remover alguns usuários anônimos e o banco de dados de teste, desativar login remoto para o root, e carregar todas essas novas regras para que o MySQL respeite imediatamente as alterações que você fez.

Para inicializar o diretório de dados do MySQL, você usaria `mysql_install_db` para versões anteriores à versão 5.7.6, e `mysqld --initialize` para versão 5.7.6 e posteriores. Contudo, se você instalou o MySQL da distribuição Debian, como descrito no Passo 1, o diretório de dados foi iniciado automaticamente; você não tem que fazer nada. Se você tentar executar o comando de qualquer maneira, você verá o seguinte erro:

Output

```
2018-04-23T20:11:15.998193Z 0 [ERROR] --initialize specified but the data
directory has files in it. Aborting.
```

Finalmente, vamos testar a instalação do MySQL.

Passo 3 — Testando o MySQL

Independentemente de como você o instalou, o MySQL deve ter iniciado executando automaticamente. Para testar isso, verifique seu status.

```
$systemctl status mysql.service
```

Você verá uma saída similar à seguinte:

Output

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:
   en
   Active: active (running) since Wed 2018-04-23 21:21:25 UTC; 30min ago
   Main PID: 3754 (mysqld)
   Tasks: 28
   Memory: 142.3M
   CPU: 1.994s
   CGroup: /system.slice/mysql.service
           └─3754 /usr/sbin/mysqld
```

Se o MySQL não está executando, você pode iniciá-lo com `sudo systemctl start mysql`.

Para uma verificação adicional, você pode tentar se conectar ao banco de dados utilizando a ferramenta `mysqladmin`, que é um cliente que lhe permite executar comandos administrativos. Por exemplo, este comando diz para conectar como root (-u root), solicitar uma senha (-p), e retornar a versão.

```
$sudo mysqladmin -p -u root version
```

Você deverá ver uma saída similar a essa:

Output

```
mysqladmin Ver 8.42 Distrib 5.7.21, for Linux on x86_64
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Server version      5.7.21-1ubuntu1
Protocol version    10
Connection         Localhost via UNIX socket
UNIX socket         /var/run/mysqld/mysqld.sock
```

Uptime: 30 min 54 sec

Threads: 1 Questions: 12 Slow queries: 0 Opens: 115 Flush tables: 1 Open
tables: 34 Queries per second avg: 0.006

Documentação de requisições

Auth

Autenticar

url: http://104.131.99.239:5050/oauth/token

method: POST

body:

```
{
  "grant_type": "password",
  "client_id": 2,
  "client_secret": "g422Ugg1VaW9UcXaqrUKe6hJNb7tETtViB9AtY4X",
  "username": user_email,
  "password": user_password,
  "scope": ""
}
```

User

Create User

url: http://104.131.99.239:5050/api/mobile/users

method: POST

header:

bearer: Content-Type:application/json

Accept:application/json

body:

```
{
  "name": *string,
  "cpf": *string,
  "birthdate": *Y-m-d",
  "gender": *string",
  "skin_color": *string",
  "cellphone": *string",
  "phone": "string",
  "status": *string",
  "email": *string",
  "password": *string",
}
```

Validações:

'name' => 'required|max:250',

'cpf' => 'required|max:14|unique:users,cpf',

'birthdate' => 'required|date',

'gender' =>

'required|in:MASCULINO,FEMININO,TRANS_MASC,TRANS_FEM,NAO_DECLARADO',

```
'skin_color' =>
'required|in:BRANCO,PARD0,NEGRO,INDIGENA,AMARELO,NAO_DECLARADO',
'cellphone' => 'required|string',
'phone'      => 'string',
'email'      => 'required|email|max:150|unique:users,email',
'password'   => 'required|max:32|string',
'status'     => 'required|in:ATIVO,BLOQUEADO,INATIVO',
```

List Users

url: http://104.131.99.239:5050/api/mobile/users

method: GET

header:

bearer: Content-Type:application/json

Accept:application/json

Authorization:Bearer {{token}}

Update User

url: http://104.131.99.239:5050/api/users/{id}

method: PUT

header:

bearer: Content-Type:application/json

Accept:application/json

Authorization:Bearer {{token}}

body:

```
{
  "id": "integer"
  "name": *"string",
  "cpf": *"string",
  "birthdate": *"Y-m-d",
  "gender": *"string",
  "skin_color": *"string",
  "cellphone": *"string",
  "phone": "string",
  "status": *"string",
  "email": *"string",
  "password": *"string",
}
```

Validações:

'id' => 'required|integer',

'name' => 'max:250',

'cpf' => 'max:14|unique:users,cpf',

'birthdate' => 'date',

'gender' =>

'in:MASCULINO,FEMININO,TRANS_MASC,TRANS_FEM,NAO_DECLARADO',

'skin_color' =>

'in:BRANCO,PARD0,NEGRO,INDIGENA,AMARELO,NAO_DECLARADO',

'cellphone' => 'string',

'phone' => 'string',

```
'email'    => 'email|max:150|unique:users,email',  
'password' => 'max:32|string',  
'status'   => 'in:ATIVO,BLOQUEADO,INATIVO',
```

Diagrama de banco de dados



