

## Guia da Aula Prática da Semana 2

### Parte 2

### Instalação e Configuração do Ambiente de Desenvolvimento Web

Este material deve servir de guia para que você consiga ter um ambiente completo para desenvolver uma aplicação Web. Neste guia são consideradas as ferramentas necessárias para o front-end e para o back-end. É importante seguir os passos indicados para que, ao final, você tenha sucesso.

#### Pré-Requisitos

Sistema Operacional: Linux

Terminal de Linha de Comando: Terminal

#### Softwares que serão instalados e configurados

- **Para o Front-end**
  - Bootstrap 4, Visual Studio Code, JWT Token, Angular 12, Node.js e NPM.
- **Para o Back-end**
  - Composer, Apache PHP 7 ou superior, Laravel 8, Node.js e MySQL.

#### 1) Instalação do Apache

Vamos instalar o Apache a partir dos repositórios-padrão de software do Ubuntu, utilizando a ferramenta de gerenciamento de pacotes APT. Antes, precisamos atualizar o índice de pacotes com o comando:

```
sudo apt-get update
```

A seguir, utilizaremos o comando para a instalação do Apache e de todas as dependências necessárias para o seu funcionamento:

```
sudo apt-get install apache2
```

#### Verificação da Instalação do Apache

```
sudo systemctl status apache2
```

A saída desse comando é mostrada a seguir:

- *apache2.service - The Apache HTTP Server*  
*Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)*  
*Active: active (running) since Thu 2021-08-12 16:38:03 -03; 22s ago*  
*Docs: <https://httpd.apache.org/docs/2.4/>*  
*Main PID: 113537 (apache2)*  
*Tasks: 55 (limit: 19047)*  
*Memory: 6.1M*  
*CGroup: /system.slice/apache2.service*  
*└─113537 /usr/sbin/apache2 -k start*

```
└─113538 /usr/sbin/apache2 -k start
└─113539 /usr/sbin/apache2 -k start
```

ago 12 16:38:03 note systemd[1]: Starting The Apache HTTP Server...

ago 12 16:38:03 note apachectl[113531]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message

ago 12 16:38:03 note systemd[1]: Started The Apache HTTP Server.

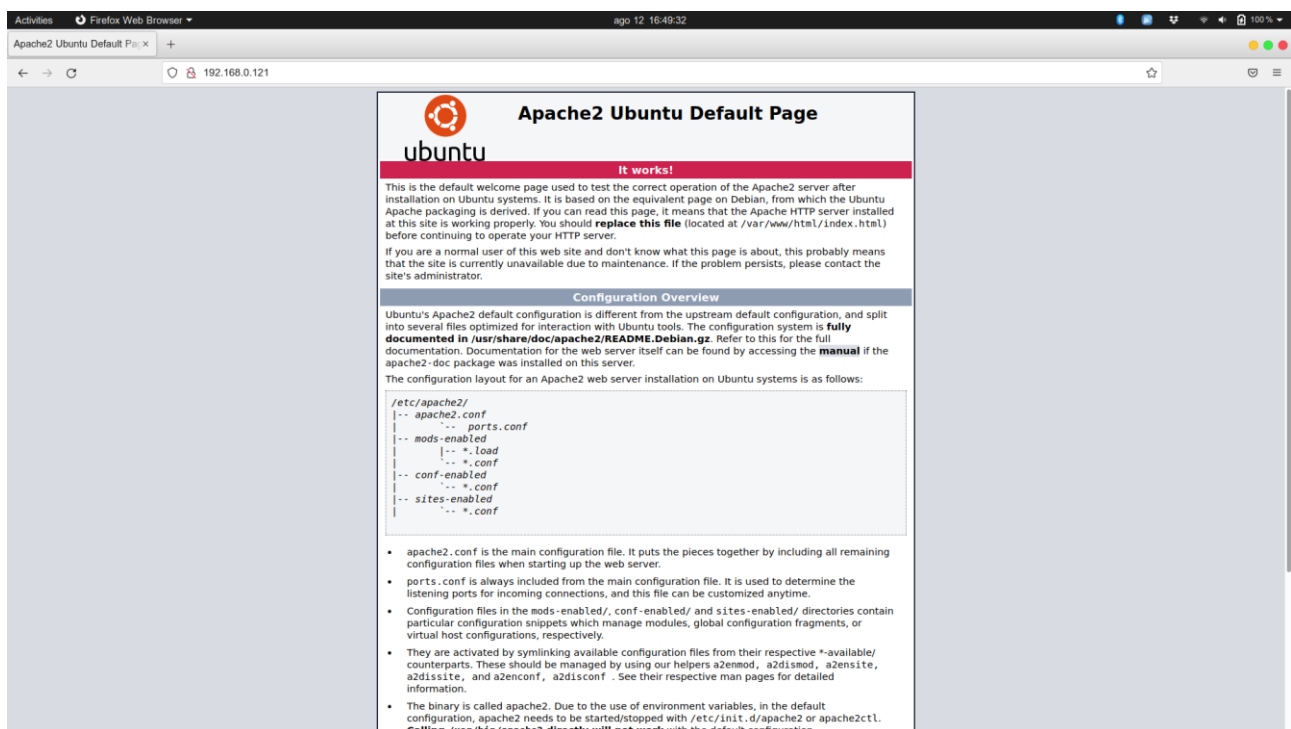
Vamos, então, solicitar uma página do Apache para verificar o seu pleno funcionamento. Para isso, precisamos saber o endereço IP do servidor. Digite o comando abaixo no prompt de comando do seu servidor:

**hostname -I**

Com o endereço IP do seu servidor, digite-o no seu navegador:

**http://endereco-ip-host**

A saída que você deve ver é a página-padrão do Apache no Ubuntu 20.04:



## Gerenciamento do Apache

Com o servidor Apache em pleno funcionamento, vamos conhecer alguns comando básicos para lidar com o servidor.

### Parar o Servidor

**sudo systemctl stop apache2**

### Inicializar o Servidor

**sudo systemctl start apache2**

## Reiniciar o Servidor

*sudo systemctl restart apache2*

## Recarregar sem perder a conexão

Caso tenhamos que fazer alguma alteração nas configurações sem precisar parar o servidor, utilize o comando a seguir:

*sudo systemctl reload apache2*

## 2) Instalação do MySQL

A instalação de um sistema de banco de dados é fundamental para conseguir armazenar e gerenciar os dados do seu site. O MySQL é um sistema de gerenciamento de banco de dados principalmente utilizado em ambientes PHP.

Utilize o comando apt-get para instalar o banco de dados:

*sudo apt-get install mysql-server*

Confirme a instalação digitando **Y** e, depois, **Enter**.

Agora, um script de segurança que vem pré-instalado com o MySQL. Esse script removerá algumas configurações-padrão inseguras e irá bloquear o acesso ao seu sistema de banco de dados.

*sudo mysql\_secure\_installation*

Esse script irá perguntar se você deseja configurar o **Validate Password Plugin**.

Selecione **Y** e escolha, em seguida, um nível de validação por senha.

Depois disso, será pedido para você escolher a senha de root do servidor MySQL (não é a senha de root do Ubuntu). Escolha uma senha forte com letras e números pelo menos.

Para o resto das perguntas escolha **Y** e tecla **Enter**.

A mensagem que irá aparecer será:

*All done!*

Ao terminar, faça login no banco de dados digitando no prompt de comando:

**sudo mysql**

Você verá algo como mostrado a seguir:



Para sair, digite:

**exit**

### 3) Instalação do PHP

Para armazenar conteúdos acerca das páginas da Web, instalamos o Apache, e para armazenar e gerenciar seus dados, instalamos o servidor de banco de dados MySQL. O PHP é responsável por processar códigos para a exibição de conteúdo dinâmico. O Laravel 8 requer o PHP 7 ou superior.

Precisamos instalar dois pacotes:

- 1) O pacote PHP,
- 2) O pacote PHP-MySQL, que é um módulo PHP que permite que o PHP comunique-se com os bancos de dados baseados em MySQL.
- 3) O pacote libapache2-mod-php para habilitar o Apache a lidar com arquivos PHP.

Para instalar esses pacotes e suas dependências, execute:

```
sudo apt install php libapache2-mod-php php-mbstring php-cli php-bcmath php-json  
php-xml php-zip php-pdo php-common php-tokenizer php-mysql
```

Assim que a instalação terminar, você pode executar o seguinte comando para confirmar sua versão PHP:

**php -v**

```
PHP 7.4.3 (cli) (built: Jul 5 2021 15:13:35) ( NTS )  
Copyright (c) The PHP Group  
Zend Engine v3.4.0, Copyright (c) Zend Technologies  
with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
```

Com isso, Apache, PHP e MySQL estão prontos para serem utilizados em conjunto.

Como na primeira parte do guia já configuramos um Apache Virtual Host para armazenar os arquivos e as pastas do nosso site, vamos apenas fazer alguns ajustes, como destacado na próxima etapa.

### DirectoryIndex no Apache

É importante mencionar que toda vez que abrimos uma página via browser sempre temos uma página-padrão que tem precedência sobre as outras. Em geral, esse arquivo é o index.html nas configurações-padrão de qualquer instalação do Apache. O problema é que para aplicações PHP, um arquivo index.php precisa vir antes do index.html nas configurações do Apache. Vamos ajustar isso.

Para alterar esse comportamento, vamos editar o arquivo `/etc/apache2/mods-enabled/dir.conf` e modificar a ordem na qual o arquivo index.php está listado na diretiva DirectoryIndex:

```
sudo pico /etc/apache2/mods-enabled/dir.conf
```

```
<IfModule mod_dir.c>  
DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm  
</IfModule>
```

Salve e feche o arquivo. Para salvar o arquivo tecle: CTRL+X e, depois, Y, em seguida, Enter. Em seguida, recarregue o Apache para que as alterações entrem em vigor:

```
sudo systemctl reload apache2
```

### Teste do PHP

No próximo passo, criaremos um script PHP para testar se o PHP está corretamente instalado e configurado em seu servidor.

Crie um novo arquivo chamado info.php dentro de sua pasta WebROOT:

```
pico /var/www/info.php
```

No arquivo em branco que foi aberto, adicione o seguinte texto, que é um código válido do PHP:

```
<?php  
phpinfo();
```

Salve e feche o arquivo. Para salvar o arquivo tecle: CTRL+X e, depois, Y, em seguida, Enter.

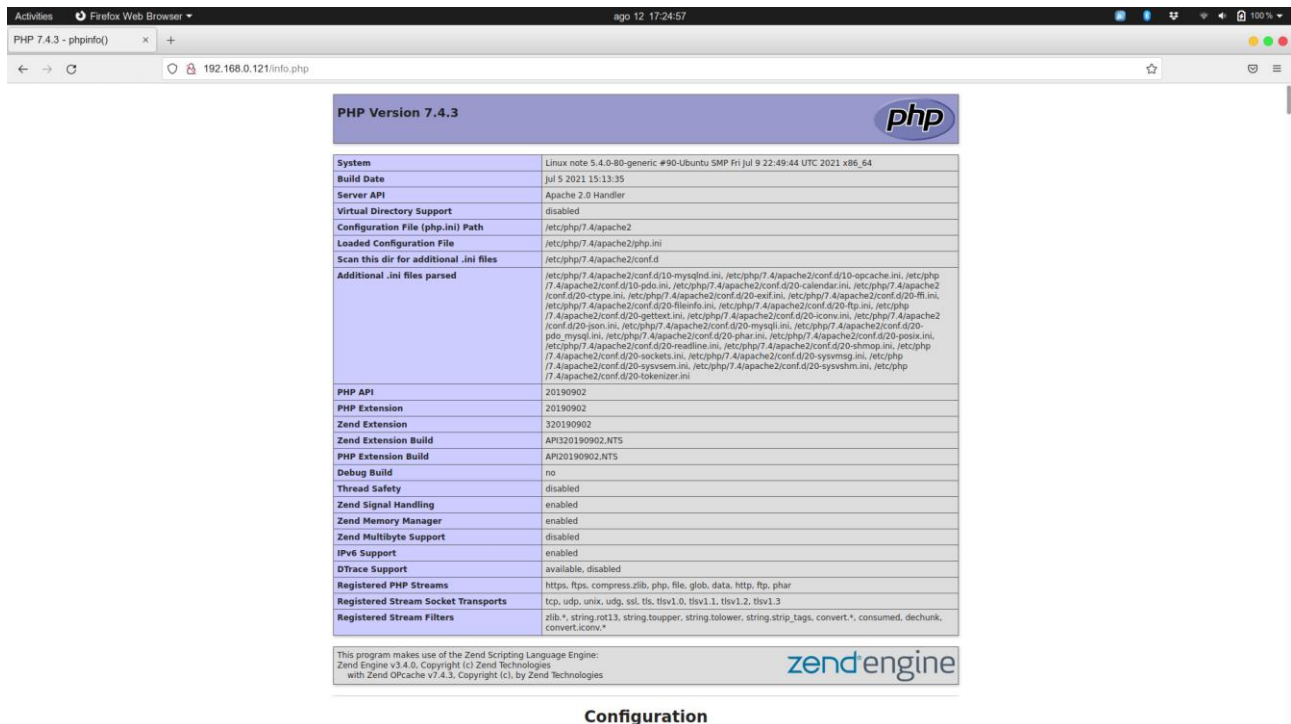
Abra seu navegador Web e acesse o nome de domínio ou o endereço IP do seu servidor seguido pelo nome do script, que neste caso é info.php:

```
http://localhost/info.php
```

OU

[http://SEU\\_IP/info.php](http://SEU_IP/info.php)

A saída é mostrada na imagem abaixo.



Se você consegue ver essa página no seu navegador, então seu PHP está funcionando corretamente.

#### 4) Criar um banco de dados para o Laravel

Uma vez que já instalamos o MySQL, vamos então criar um banco de dados para o Laravel.

#### Logar no MySQL

```
sudo mysql -u root -p
```

#### Criar um banco de dados

```
CREATE DATABASE laravel_db;
```

#### Criar um usuário para a base de dados

```
CREATE USER 'laravel_user'@'localhost' IDENTIFIED BY 'secretpassword';
```

#### Fornecer privilégios para o usuário

```
GRANT ALL ON laravel_db.* TO 'laravel_user'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
QUIT;
```

#### 5) Instalação do Composer



Composer é um gerenciador de pacotes de dependência para PHP. Ele fornece uma estrutura para gerenciar bibliotecas e dependências necessárias. Para usar o Laravel, primeiro temos que instalar o composer.

```
curl -sS https://getcomposer.org/installer | php
```

Em seguida, mova o arquivo composer para o caminho `/usr/local/bin`

```
sudo mv composer.phar /usr/local/bin/composer
```

Dê permissão e execução:

```
sudo chmod +x /usr/local/bin/composer
```

Verifique se o composer foi instalado:

```
composer --version
```

## **6) Instalação do Laravel 8**

Uma vez que temos o composer instalado, o próximo passo é instalar o Laravel.

Mude as permissões do diretório `/var/www/html`

```
sudo chown -R seu-usuario.seu-usuario /var/www/html
```

Vá para o diretório:

```
cd /var/www/html
```

Instale o Laravel utilizando o composer. O comando cria um novo diretório chamado `laravelapp` e instala todos os arquivos e diretórios do Laravel.

```
composer create-project laravel/laravel laravelapp
```

Entre no diretório `laravelapp` e verifique a versão do Laravel

```
cd laravelapp
```

```
php artisan
```

## **7) Configurar o Apache para hospedar o site do Laravel 8**

Por último, precisamos configurar o servidor Web Apache para hospedar o site Laravel. Para que isso funcione, precisamos criar um arquivo `host virtual`.

```
sudo pico /etc/apache2/sites-available/laravel.conf
```

Troque a linha `ServerName example.com` por **`ServerName localhost`**

```
<VirtualHost *:80>  
ServerName example.com
```

```
ServerAdmin admin@example.com
DocumentRoot /var/www/html/laravelapp/public
<Directory /var/www/html/laravelapp>
AllowOverride All
</Directory>
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Salve e feche o arquivo. Para salvar o arquivo tecle: CTRL+X e, depois, Y, em seguida, Enter.

Em seguida, habilite o site do Laravel e o módulo de reescrita do Apache usando estes dois comandos.

```
sudo a2ensite laravel.conf
```

```
sudo a2enmod rewrite
```

**Prossiga com os ajustes abaixo:**

```
cd /var/www/html/laravelapp
chmod -R 775 storage bootstrap/cache
chmod 777 -R storage bootstrap/cache
```

**Reinicie o Apache**

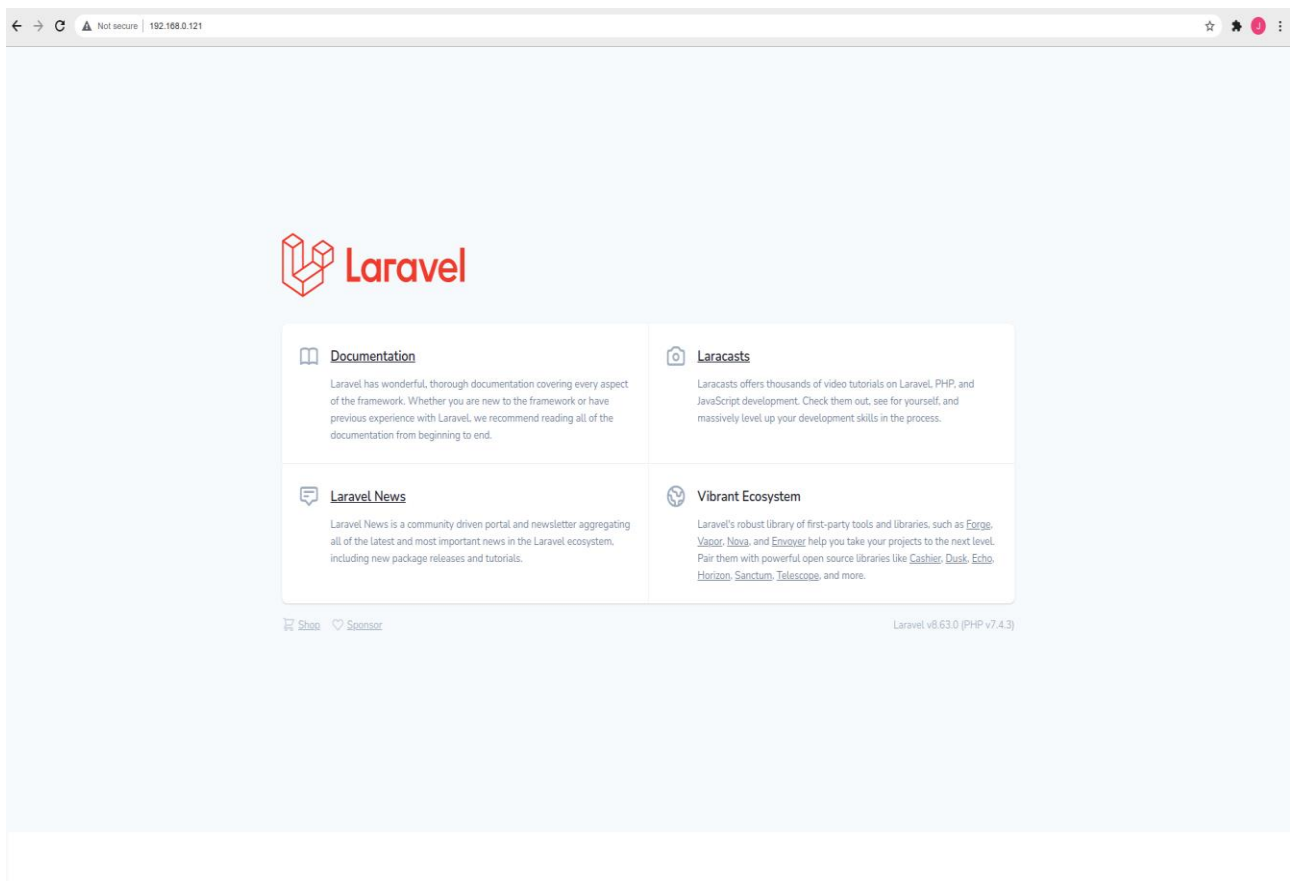
```
sudo systemctl restart apache2
```

**Acesse o Laravel**

Abra o browser de sua preferência e digite:

**http://localhost**





## 8) Instalação do NodeJS e NPM

O `Node.js` é um ambiente de execução do JavaScript criado para a programação do lado do servidor e que permite aos desenvolvedores criarem funcionalidades de back-ends utilizando o JavaScript. O JavaScript é uma linguagem que muitos já estão familiarizados com o desenvolvimento Web.

Para instalar o Node.js no terminal do Linux, execute o comando:

**apt-get update**

**curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh**

**curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh | bash**

**source ~/.bashrc**

**nvm list-remote**

**nvm install v14.18.0**

**nvm list**

Verifique se a instalação foi feita adequadamente.

**node -v**

O NPM é o gerenciador de pacotes do Node.js e precisa ser instalado como segue:

**sudo apt-get install npm**

## 9) Instalação da CLI do Angular

O Angular CLI é uma ferramenta de interface de linha de comando que você usa para inicializar, desenvolver, criar e manter aplicativos Angular diretamente de um shell de comando. Para instalar, basta digitar o comando no terminal:

```
npm install -g @angular/cli
```

## 10) Instalação do Visual Studio Code

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Apresenta suporte para depuração, controle de versionamento GIT incorporado, refatoração de código etc. Pela sua funcionalidade ele foi escolhido para desenvolvermos a aplicação Web. Você pode utilizar o editor de sua preferência também.

Para instalar o Visual Code Studio, acesse o site oficial:

<https://code.visualstudio.com/>

Escolha a versão para o Linux Ubuntu. Baixe o arquivo .deb e prossiga com a instalação.

Para instalar no Ubuntu, basta você clicar duas vezes no arquivo **code\_1.60.2-1632313585\_amd64.deb** que um instalador será aberto para que você finalize a instalação.

Após instalado, digite no terminal:

```
code
```

A janela do Visual Code Studio aparecerá como a seguir:

