

Educación Informativa UCC

Hecho en: Laravel 4.2

Instrucciones de Instalación

Para ejecutar el proyecto en un servidor nginx en Ubuntu GNU/Linux 14.04. Seguir las siguientes instrucciones:

1. Instalar los componentes del Backend

Necesitaremos correr las actualizaciones en nuestro sistemas operativo; además instalar nginx y sus respectivas librerías en PHP 5

```
sudo apt-get update
```

```
sudo apt-get install nginx php5-fpm php5-cli php5-mcrypt git
```

2. Modificar la configuración de PHP

Lo primera que necesitamos es abrir el archivo de configuración principal de PHP (`php.ini`), para el procesador `PHP-fpm` que utiliza `nginx` . Abrir el archivo con permisos de administrador con un editor de texto

```
sudo nano /etc/php5/fpm/php.ini
```

Una vez abierto el archivo, modifique la línea con el parámetro: `cgi.fix_pathinfo` (posiblemente está comentado o con un valor de 1) y la cambias a cero

```
cgi.fix_pathinfo=0
```

Esto le indica a PHP que no intente ejecutar scripts con nombres similares si el archivo que busca no es encontrado. Una vez finalizado, guardar y cerrar el archivo.

La última pieza de la administración de PHP a modificar es activar la extension `mcrypt` de la cual `Laravel` depende.

```
sudo php5enmod mcrypt
```

Ahora reiniciamos el servicio `php5-fpm` para que se implementen los cambios

realizados

```
sudo service php5-fpm restart
```

3. Configurar Nginx y la ruta web

Crearemos un directorio para alojar nuestro proyecto, para eso necesitamos hacerlo con permisos de administrador, en este caso será:

```
sudo mkdir -p /var/www/mooc_ucc
```

Ahora que tenemos una ubicación para los componentes de laravel, podemos mover y editar un servidor nginx, para eso abrimos el archivo de configuración por defecto con permisos de administrador:

```
sudo nano /etc/nginx/sites-available/default
```

Dentro del archivo agregamos nuestra configuración del servidor

```
server {
    listen 80;
    listen [::]:80 default_server ipv6only=on;

    root /var/www/mooc_ucc/public;
    index index.php index.html index.htm;

    server_name 127.0.0.1;
    ssl_certificate /etc/nginx/ssl/nginx.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx.key;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~ \.php$ {
        try_files $uri /index.php =404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

}
```

Guardar y cerrar el archivo cuando hayas finalizado. Después para que el servidor

nginx lea los cambios, lo reiniciamos ejecutando la siguiente línea:

```
sudo service nginx restart
```

Ahora, descomprimos el archivo `.zip` del proyecto, por ejemplo `mooc_ucc.zip` y copiamos la carpeta del proyecto a `/var/www`

```
sudo cp /var/www
```

Ahora nuestros archivos quedan en el directorio `/var/www/mooc_ucc` pero quedan con cuenta de administrador (grupo root), por lo cual vamos a cambiarlos a otro (grupo web):

```
sudo chown -R :www-data /var/www/mooc_ucc
```

Nuestro siguiente paso es cambiar los permisos de `/var/www/mooc_ucc/app/storage` los cuáles necesitan tener permisos de escritura para que la aplicación funcione

```
sudo chmod -R 775 /var/www/mooc_ucc/app/storage
```

EL último paso es reiniciar nuestro servidor nginx

```
sudo service nginx restart
```

Una vez hecho esto podremos ingresar a localhost y nos debería permitir ver nuestra página index

Entorno de desarrollo

La carpeta `mooc_ucc` es la raíz del proyecto, así que basta con abrir con el editor de texto favorito (por ejemplo [atom](#) o [sublime text](#)) la carpeta.

Árbol del proyecto

```
|— app
|— artisan
|— bootstrap
|— composer.json
|— composer.json~
|— composer.lock
|— CONTRIBUTING.md
|— phpunit.xml
|— Procfile
|— public
|— README.md
|— server.php
|— vendor
```

En la carpeta `app` podremos encontrar los archivos y código fuente del MOOC (modelos, vistas y controladores). El archivo `composer.json` contiene los paquetes, dependencias y librerías utilizadas en éste proyecto. En la carpeta `public` se encontrarán los assets como archivos `css` , `javascript` , imágenes entre otros archivos estáticos.

Ahora bien, si miramos la estructura de nuestra carpeta `app` :

```
|— commands
|— config
|— controllers
|— database
|— filters.php
|— lang
|— library
|— models
|— routes.php
|— start
|— storage
|— tests
|— views
```

Podremos observar nuestras tres carpetas principales: `models` (modelos/clases), `controllers` (controladores) y `views` (vistas).