

Universidad Don Bosco.



Docente:

Jonathan Baltazar Escalante Guardado.

Materia:

Desarrollo de Aplic. Web con Soft. Interpret. En el Servidor.

Grupo:

05L.

Alumno:

Eduardo Andrés Cortez Ramírez.

Carnet:

CR210957.

Carrera:

Técnico en Ingeniería de la Computación.

Tema:

Desafío Práctico 3.

Fecha de entrega:

Domingo 12 de Mayo del 2024.

Nos desplazamos hasta la ubicación donde queremos instalar laravel, en este caso es la carpeta que aparece la ruta, ahí ejecuté el comando de instalación.

```
C:\xampp\htdocs\DSS>cd C:/xampp/htdocs/DSS/Laravel  
C:\xampp\htdocs\DSS\Laravel>composer require laravel/installer --dev
```

Mandamos el comando para decirle que queremos crear un nuevo proyecto de laravel, se comenzará a ejecutar y pedirá unas configuraciones como el kit de inicio, nuestras preferencias de framework y si queremos copiar un repositorio de github para empezar a trabajar.

```
C:\xampp\htdocs\DSS\Laravel>php vendor/bin/laravel new PracticoDSS3  
  
Laravel  
  
Would you like to install a starter kit? [No starter kit]:  
[none    ] No starter kit  
[breeze  ] Laravel Breeze  
[jetstream] Laravel Jetstream  
> none  
  
Which testing framework do you prefer? [Pest]:  
[0] Pest  
[1] PHPUnit  
> 1  
  
Would you like to initialize a Git repository? (yes/no) [no]:  
> no
```

Nos movilizamos a la carpeta ya creada, haremos un “composer install” para instalar paquetes y dependencias en la carpeta y asegurarnos de que todos los archivos necesarios hayan sido instalados correctamente.

```
C:\xampp\htdocs\DSS\Laravel>cd PracticoDSS3  
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>composer install
```

Instalaremos paquetes de yarn que nos ayudarán a mantener las dependencias del proyecto en orden, para ello es necesario tener Node js en el sistema.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>npm install-global yarn
```

Con este comando se van a compilar los archivos js que se tienen en el proyecto.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>yarn && yarn dev
```

Se harán las migraciones a la base de datos con el comando indicado, se debe tener creada la base de datos antes de hacerlo así como indicarle los datos en el archivo .dev, en este caso por usarse una plantilla para trabajar, ya existían archivos que debían ser necesarios migrar.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>php artisan migrate
```

Ahora vamos a hacer uso de un paquete de permisos para poder manejar los roles al momento de hacer el inicio de sesión, con este paquete podremos mostrarle la información al usuario según lo tenga permitido.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>composer require spatie/laravel-permission
```

Se tiene que hacer público los permisos de roles, por lo que la función del comando es copiar los archivos que permiten utilizar sus servicios, como las migraciones y el permission.php.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"
```

Acá solo se le indica que borre la caché para que no tenga problemas a la hora de cargar la página.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>php artisan optimize:clear
```

Dentro del código se deben hacer ciertas configuraciones que permiten la creación de las tablas y los roles, cuando esas configuraciones estén realizadas, se procede con la migración de las nuevas tablas.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>php artisan migrate
```

Se creará un archivo para la migración, dentro de ese archivo se especifican los roles que queremos implementar así como los usuarios que tendrán el rol, de momento solo será un rol para administrador, por lo que se deberá programar.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>php artisan make:migration crear_rol
```

Al crear este archivo de migración, nos ayudará para especificar que el primer usuario (o el que escojamos mediante programación) será administrador, configurado ese código, se procede con la migración.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>php artisan migrate
```

Se hará un controlador para redireccionar al usuario según las peticiones que realice, así se le mostrarán las vistas o los procesos a realizar.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>php artisan make:controller UserController
```

También se hará un controlador para ingresar los datos a la tabla que contiene los roles para cada usuario, esta tabla indica quien es un administrador y quien un trabajador, por lo que será necesario ingresar datos con la interfaz del proyecto.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>php artisan make:controller ModelHasRolController
```

Un modelo para ingresar los datos a la base también será necesario.

```
C:\xampp\htdocs\DSS\Laravel\PracticoDSS3>php artisan make:model ModelHasRol -f
```