

## Sesión práctica: nuestro primer proyecto Laravel

En esta sesión práctica desarrollaremos un pequeño proyecto en Laravel que nos permitirá conocer diferentes herramientas de este framework y al mismo tiempo llevar a la práctica conceptos como arquitectura MVC, modelo, controlador, vista, ruta, etc.

Lo primero que haremos será instalar el software necesario para poder trabajar con Laravel. Para ello debemos instalar Composer y el instalador de Laravel.

### Instalación de Composer en Linux

Para realizar la instalación accederemos a la terminal linux y ejecutaremos los siguientes comandos.

Para instalar Composer lo primero que haremos será descargar Composer desde su sitio web:

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
```

Podemos instalar Composer de forma local dentro de un directorio de un proyecto o globalmente en el sistema. Esta segunda forma de instalación nos permitirá que el gestor de dependencias se ejecute en cualquier parte del sistema. Para realizar una instalación global ejecutaremos el siguiente comando:

```
php composer-setup.php --install-dir=/usr/local/bin -filename=composer
```

Una vez realizada la instalación, si queremos eliminar el instalador ejecutamos el siguiente comando:

```
php -r "unlink('composer-setup.php');"
```

Una vez instalado Composer necesitamos renombrar el archivo /usr/local/bin/composer.phar como /usr/local/bin/composer. De no hacerlo, nos podría dar error a la hora de ejecutar Laravel/Installer.

Para comprobar si Composer se ha instalado correctamente escribiremos en la terminal el siguiente comando que mostrará la versión instalada:

```
composer --version
```

### Instalación de Laravel en Linux.

Para instalar Laravel, lo que hacemos es instalar el instalador de Laravel que ejecutaremos cada vez que queramos crear un nuevo proyecto Laravel. Para instalar este instalador haremos uso de composer y ejecutaremos el siguiente comando:

```
composer global require laravel/installer
```

Para verificar que todo ha ido bien ejecutaremos el comando que nos muestra la versión de Laravel/installer.

```
Laravel --version
```

Una vez instalado composer y el instalador de Laravel ya podremos crear nuestros proyectos en Laravel.

## **Creación de nuestro proyecto Laravel**

Para familiarizarnos con el framework Laravel crearemos una pequeña aplicación que muestre el contenido de la tabla “producto” de la base de datos “tienda” cuando el usuario acceda a la URL nuestra aplicación.

Empezaremos creando el proyecto que podemos llamar “GestorProductos”. Para crear dicho proyecto laravel ejecutaremos el siguiente comando:

```
laravel new GestorProductos
```

Al ejecutar este comando el instalador de Laravel descargará los diferentes componentes que conformarán nuestro proyecto. Se creará una carpeta con el nombre del proyecto que contendrá la estructura de carpetas de nuestro proyecto.

Una vez creada la carpeta del proyecto abriremos esta en VSCode y podremos ver la estructura de carpetas de nuestro proyecto.

Lo primero que haremos una vez creada la estructura de carpetas de nuestro proyecto será modificar el archivo de configuración **.env** para indicarle los parámetros de conexión a la base de datos “tienda”. Debemos indicar el nombre de la base de datos, ubicación de esta (localhost), el nombre de usuario y la contraseña.

Para desarrollar nuestro proyecto crearemos un controlador, un modelo y una vista. El controlador lo llamaremos “ControladorProductos”, el modelo ModeloProductos y la vista VistaProductos.

### Creación de nuestro controlador “ControladorProductos”

Para crear nuestro controlador ejecutaremos desde la terminal dentro de VSCode el siguiente comando:

```
php artisan make:controller ControladorProductos
```

Una vez creado nuestro controlador podremos verlo en la carpeta del proyecto app/Http/Controllers con el nombre ControladorProductos.php.

### Creación de nuestro modelo “ModeloProductos”

Para crear nuestro modelo ejecutaremos desde la terminal dentro de VSCode el siguiente comando:

```
php artisan make:model ModeloProductos
```

Una vez creado nuestro modelo podremos verlo en la carpeta del proyecto app/Models con el nombre ModeloProductos.php.

### Creación de nuestra vista “VistaProductos”

Para crear nuestro modelo ejecutaremos desde la terminal dentro de VSCode el siguiente comando:

```
php artisan make:view VistaProductos
```

Una vez creada nuestra vista podremos verla en la carpeta del proyecto resources/views con el nombre VistaProductos.php.

Ya hemos creado los tres ficheros php (controlador, modelo y vista) que contendrán el código de nuestra aplicación.

Hecho esto podemos crear en el “route dispatcher” (web.php) la ruta que asociará la URL de nuestra aplicación con el controlador que la gestionará:

Para asociar la URL de la aplicación con el controlador que la gestionará escribiremos en el archivo web.php el código para importar la clase que implementa el controlador y el código para crear la ruta:

```
use App\Http\Controllers\ControladorProductos;
Route::get('/', [ControladorProductos::class, 'MuestraProductos'])->name('Productos');
```

A continuación escribiremos el código del controlador, del modelo y de la vista.

En el modelo únicamente tenemos que indicar a qué tabla de la base de datos “tienda” estará asociado, y para ello escribiremos el siguiente código:

```
protected $table = 'producto';
```

De esta manera le indicamos al modelo que la tabla asociada es la tabla ‘producto’ de nuestra base de datos “tienda”.

En el controlador debemos añadir el código de la función “MuestraProductos” y el código para importar la clase que implementa el modelo. Esta función lo que hará será obtener los datos de la tabla “producto” haciendo uso del modelo que creamos previamente y pasárselos a la vista para que los muestre. A continuación vemos el código de esta función:

```
use App\Models\ModeloProductos;

public function MuestraProductos() {
    $productos = ModeloProductos::all();
    return view('VistaProductos', ['productos' => $productos]);
}
```

Por último en la vista tendremos que introducir el código que recorre el array que le hemos pasado como parámetro con los datos de los productos para mostrarlos por pantalla. En la vista tendremos código HTML y directivas Blade:

```
<html>
<ul>
    <p style="font-size: 30px;">Listado de productos</p>
    @if($productos->isEmpty())
        <p>No hay productos.</p>
    @else
        @foreach ($productos as $producto)
            Nombre: {{ $producto->nombre }}, Precio: {{ $producto->precio }}<p></p>
        @endforeach
    @endif
</ul>
</html>
```

Para probar nuestro código haremos uso del servidor incluído en Laravel que estará escuchando en localhost:8000. Para ejecutar este servidor escribiremos el siguiente comando en la terminal de VSCode:

```
php artisan serve
```

Para abrir el navegador haremos Control+Click sobre la dirección [http://127.0.0.1:8000].  
Para pararlo haremos Control+C desde la terminal.

Para probar nuestro proyecto iremos a la dirección <http://localhost:8000> en nuestro navegador.

### Possible ampliación de nuestro proyecto

Podemos ampliar nuestro proyecto permitiendo mostrar un único producto añadiendo su identificador de producto en la URL. Para ello tendremos que añadir una nueva ruta en el route dispatcher, una nueva función en el controlador y una nueva vista.