

# Conceptos básicos Laravel

## 1. Crear un nuevo proyecto Laravel

Crea un proyecto llamado BlogBasico usando el instalador o bien composer. Indica el comando de consola que utilizas y el comando que utilizarías para ejecutar el proyecto en localhost.

**laravel new Blogbasico**

**php artisan serve**



```
C:\Users\quimpueeyo>laravel new BlogBasico
```

M

PS C:\Users\quimpueeyo\BlogBasico> php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

## 2. Explorar estructura del proyecto

Identifica en el proyecto, los siguientes apartados. ¿Qué función cumple cada directorio?

**/routes/web.php:** Define las rutas "web" de la aplicación. Aquí se registran las URL que los usuarios visitarán en el navegador y se asocian con vistas o controladores.

**/app/Http/Controllers:** Contiene los controladores de la aplicación. Los controladores manejan la lógica de las peticiones HTTP, procesando datos y devolviendo respuestas (vistas, JSON, etc.).

**/resources/views:** Almacena las plantillas de visualización (views), generalmente archivos .blade.php, que definen la interfaz de usuario (HTML) que se envía al navegador

**/app/Models:** Contiene los modelos Eloquent ORM. Cada modelo representa una tabla en la base de datos y permite interactuar con ella de forma orientada a objetos.

### 3. Crear rutas simples

Edita el archivo routes/web.php y crear las siguientes rutas:

Ruta /hola que devuelva un texto.

Ruta /usuario/{nombre} que devuelva un saludo personalizado.

Ruta /suma/{a}/{b} que devuelva la suma de dos números.

```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  use App\Http\Controllers\PaginaController;
6
7  Route::get('/', [PaginaController::class, 'inicio']);
8  Route::get('/contacto', [PaginaController::class, 'contacto']);
9  Route::get('/hola', function () {
10     return 'Hola mundo';
11 });
12
13 Route::get('/usuario/{nombre}', function ($nombre) {
14     return 'Hola ' . $nombre;
15 });
16
17 Route::get('/suma/{a}/{b}', function ($a, $b) {
18     return $a + $b;
19 });
20 |
```

#### 4. Crear un controlador básico

##### php artisan make:controller PaginaController

Crear un controlador llamado PaginaController. Indica el comando que se utiliza para ello.

Añade al controlador, los siguientes métodos:

inicio() → devuelve “Bienvenido a la página de inicio”.

contacto() → devuelve “Página de contacto”.

Modifica el archivo web.php para incluir las rutas para llamar a las funciones creadas.

```
app > Http > Controllers > PaginaController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class PaginaController extends Controller
8  {
9      public function inicio()
10     {
11         return 'Bienvenido a la página de inicio';
12     }
13
14     public function contacto()
15     {
16         return 'Página de contacto';
17     }
18 }
```

## 5. Crear un modelo y migración

Crear un modelo Post con migración incluida. Indica el comando que utilizas.

Agrega los siguientes campos a la migración: título, contenido.

Ejecuta la migración. Indica el comando que utilizas.

**php artisan make:model Post -m**

```
database > migrations > 2026_02_05_163508_create_posts_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration {
8      /**
9       * Run the migrations.
10      */
11     public function up(): void
12     {
13         Schema::create('posts', function (Blueprint $table) {
14             $table->id();
15             $table->string('titulo');
16             $table->text('contenido');
17             $table->timestamps();
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      */
24     public function down(): void
25     {
26         Schema::dropIfExists('posts');
27     }
28 };
29 | Ctrl+I for Command, Ctrl+L for Agent
```