**COMANDOS DE LARAVEL 11**

# **Fase 0 - Arranque del proyecto**

## **0. Requisitos previos (verifica)**

php -v # >= 8.2

composer -V # Composer 2.x

node -v # >= 18

npm -v # >= 9

mysql --version

## **1️⃣ Crear el proyecto en Laravel 11**

Ubícate en la carpeta htdocs de tu **XAMPP**:

Carpeta: htdocs

cd C:\xampp\htdocs

composer create-project laravel/laravel proyectos\_investigacion

Crear la base

proyectos\_investigacion

## **2️⃣ Configurar .env con MySQL**

Abre el archivo .env y ajusta la conexión a tu base de datos:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=proyectos\_investigacion

DB\_USERNAME=root

DB\_PASSWORD=

## **3️⃣ Instalar Breeze (Blade)**

Entra al proyecto:

cd proyectos\_investigacion

composer require laravel/breeze --dev

php artisan breeze:install blade

## **4️⃣ Instalar dependencias y compilar con Vite**

Ejecuta:

npm install

npm run dev

## **5️⃣ Migrar la base de datos**

Ejecuta las migraciones iniciales:

php artisan migrate

## **6️⃣ Levantar el servidor**

Ejecuta:

php artisan serve

Y entra en el navegador a:

http://127.0.0.1:8000

## **7️⃣ Probar el login**

1. Haz clic en **Register** para crear un usuario.
2. Ingresa usuario y contraseña.
3. Verifica que puedas cerrar sesión y volver a entrar con **Login**.

✅ **Checklist de Validación (DoD):**

* El proyecto **proyectos\_investigacion** está creado en htdocs.
* Archivo .env configurado con base de datos MySQL **proyectos\_investigacion**
* Breeze (Blade) instalado y compilado con Vite sin errores.
* Migraciones ejecutadas correctamente.
* Se puede registrar un nuevo usuario.
* El login y logout funcionan sin fallos.

## **✅ Checklist de Validación (DoD — Fase 0)**

* Proyecto **proyectos\_investigacion** creado dentro de htdocs con Laravel 11.
* Archivo .env configurado correctamente con la base de datos MySQL **proyectos\_investigacion**
* Paquete **Laravel Breeze (Blade)** instalado y compilado con **Vite** sin errores.
* Migraciones ejecutadas (php artisan migrate) y tablas creadas en MySQL.
* Rutas de autenticación (/login, /register, /forgot-password) disponibles y funcionando.
* Registro de usuario probado exitosamente.
* Inicio de sesión y cierre de sesión funcionando correctamente.
* (Opcional) La ruta raíz (/) redirige a /login o al /dashboard según configuración.

## **📝 Resumen (Fase 0)**

En esta fase inicial se construyó la base del sistema con Laravel 11, instalando y configurando correctamente el proyecto en la carpeta htdocs. Se estableció la conexión con MySQL a través del archivo .env y se ejecutaron las migraciones iniciales. Posteriormente se instaló **Laravel Breeze con Blade**, que provee las rutas y vistas de autenticación. Tras compilar los assets con Vite, se validó el registro de nuevos usuarios, así como el inicio y cierre de sesión. Con ello, el sistema cuenta con un módulo de login plenamente funcional, constituyendo el punto de partida para la implementación de **roles y permisos** en la siguiente fase.

# **Fase 1 - Roles y Permisos**

## **📌 Paso 1: Agregar campo role a la tabla users**

**Ejecuta en consola:**

php artisan make:migration add\_role\_to\_users\_table --table=users

**En el archivo generado** database/migrations/xxxx\_xx\_xx\_add\_role\_to\_users\_table.php,

**agrega:**

public function up(): void

{

Schema::table('users', function (Blueprint $table) {

$table->tinyInteger('role')->default(1)->after('password');

// 1 = Usuario, 2 = Coordinador, 3 = Administrador

});

}

public function down(): void

{

Schema::table('users', function (Blueprint $table) {

$table->dropColumn('role');

});

}

**Luego corre la migración:**

php artisan migrate

## **📌 Paso 2: Crear Seeder de usuarios iniciales**

**Genera un seeder:**

php artisan make:seeder UserSeeder

**Edita database/seeders/UserSeeder.php:**

use Illuminate\Support\Facades\Hash;

use App\Models\User;

public function run(): void

{

User::create([

'name' => 'Administrador',

'email' => 'admin@udg.mx',

'password' => Hash::make('admin123'),

'role' => 3,

]);

User::create([

'name' => 'Coordinador',

'email' => 'coordinador@udg.mx',

'password' => Hash::make('coord123'),

'role' => 2,

]);

}

**Ejecuta el seeder:**

php artisan db:seed --class=UserSeeder

## **📌 Paso 3: Crear Middleware Role**

**Genera un middleware:**

php artisan make:middleware RoleMiddleware

**Edita app/Http/Middleware/RoleMiddleware.php**:

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

class RoleMiddleware

{

public function handle(Request $request, Closure $next, ...$roles)

{

if (!in\_array($request->user()->role, $roles)) {

abort(403, 'No tienes permiso para acceder a esta ruta.');

}

return $next($request);

}

}

### **Cómo registrar el middleware role en Laravel 11**

1. Abre el archivo bootstrap/app.php.
2. Dentro de la configuración de rutas, agrega tu middleware así:

<?php

use Illuminate\Foundation\Application;

use Illuminate\Foundation\Configuration\Exceptions;

use Illuminate\Foundation\Configuration\Middleware;

return Application::configure(basePath: dirname(\_\_DIR\_\_))

->withRouting(

web: \_\_DIR\_\_.'/../routes/web.php',

commands: \_\_DIR\_\_.'/../routes/console.php',

health: '/up',

)

->withMiddleware(function (Middleware $middleware) {

// Aquí registramos middlewares de ruta personalizados

$middleware->alias([

'role' => \App\Http\Middleware\RoleMiddleware::class,

]);

})

->withExceptions(function (Exceptions $exceptions) {

//

})->create();

## **📌 Paso 4: Proteger rutas según rol**

**En routes/web.php:**

use Illuminate\Support\Facades\Route;

Route::middleware(['auth', 'role:1'])->group(function () {

Route::get('/usuario', function () {

return view('usuario.dashboard');

});

});

Route::middleware(['auth', 'role:2'])->group(function () {

Route::get('/coordinador', function () {

return view('coordinador.dashboard');

});

});

Route::middleware(['auth', 'role:3'])->group(function () {

Route::get('/admin', function () {

return view('admin.dashboard');

});

});

## **Paso 1: Crear controlador para redirección**

**Crea un controlador llamado RedirectController:**

php artisan make:controller RedirectController

📄 **app/Http/Controllers/RedirectController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class RedirectController extends Controller

{

public function \_\_invoke(Request $request)

{

$role = $request->user()->role;

return match ($role) {

1 => redirect()->route('usuario.dashboard'),

2 => redirect()->route('coordinador.dashboard'),

3 => redirect()->route('admin.dashboard'),

default => redirect()->route('dashboard'),

};

}

}

**routes/web.php**

<?php

use App\Http\Controllers\ProfileController;

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\RedirectController;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

\*/

// Página de inicio

Route::get('/', function () {

return view('welcome');

});

// Dashboard dinámico (redirige según rol)

Route::get('/dashboard', RedirectController::class)

->middleware(['auth', 'verified'])

->name('dashboard');

// Perfil de usuario (editar, actualizar, eliminar)

Route::middleware('auth')->group(function () {

Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');

Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');

Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');

});

// Rutas protegidas por rol

Route::middleware(['auth', 'role:1'])->group(function () {

Route::get('/usuario', fn() => view('usuario.dashboard'))

->name('usuario.dashboard');

});

Route::middleware(['auth', 'role:2'])->group(function () {

Route::get('/coordinador', fn() => view('coordinador.dashboard'))

->name('coordinador.dashboard');

});

Route::middleware(['auth', 'role:3'])->group(function () {

Route::get('/admin', fn() => view('admin.dashboard'))

->name('admin.dashboard');

});

// Autenticación (Breeze/Jetstream)

require \_\_DIR\_\_.'/auth.php';

### **1) Usuario (rol = 1)**

**resources/views/usuario/dashboard.blade.php**

{{-- resources/views/usuario/dashboard.blade.php --}}

@extends('layouts.app')

@section('content')

<div class="max-w-7xl mx-auto p-6">

<div class="mb-6">

<h1 class="text-2xl font-bold">Panel del Usuario</h1>

<p class="text-gray-600">Bienvenido, {{ auth()->user()->name }}.</p>

</div>

<div class="grid sm:grid-cols-2 lg:grid-cols-3 gap-6">

<a href="{{ route('usuario.dashboard') }}"

class="block rounded-2xl border p-5 hover:shadow">

<h2 class="font-semibold text-lg">Mis Proyectos</h2>

<p class="text-sm text-gray-600">Crear y dar seguimiento a proyectos.</p>

</a>

<a href="#" class="block rounded-2xl border p-5 hover:shadow">

<h2 class="font-semibold text-lg">Documentos Adjuntos</h2>

<p class="text-sm text-gray-600">Sube, visualiza y gestiona PDFs.</p>

</a>

<a href="#" class="block rounded-2xl border p-5 hover:shadow">

<h2 class="font-semibold text-lg">Reportes</h2>

<p class="text-sm text-gray-600">Descarga en PDF/Excel.</p>

</a>

</div>

</div>

@endsection

### **2) Coordinador (rol = 2)**

**resources/views/coordinador/dashboard.blade.php**

{{-- resources/views/coordinador/dashboard.blade.php --}}

@extends('layouts.app')

@section('content')

<div class="max-w-7xl mx-auto p-6">

<div class="mb-6">

<h1 class="text-2xl font-bold">Panel del Coordinador</h1>

<p class="text-gray-600">Centro Universitario: (configurar CU del coordinador)</p>

</div>

<div class="grid sm:grid-cols-2 lg:grid-cols-3 gap-6">

<a href="{{ route('coordinador.dashboard') }}"

class="block rounded-2xl border p-5 hover:shadow">

<h2 class="font-semibold text-lg">Proyectos por revisar</h2>

<p class="text-sm text-gray-600">Listar, filtrar y validar.</p>

</a>

<a href="#" class="block rounded-2xl border p-5 hover:shadow">

<h2 class="font-semibold text-lg">Observaciones</h2>

<p class="text-sm text-gray-600">Emite comentarios y devuelve proyectos.</p>

</a>

<a href="#" class="block rounded-2xl border p-5 hover:shadow">

<h2 class="font-semibold text-lg">Oficios preliminares</h2>

<p class="text-sm text-gray-600">Generación PDF (DomPDF).</p>

</a>

</div>

</div>

@endsection

### **3) Administrador (rol = 3)**

**resources/views/administrador/dashboard.blade.php**

{{-- resources/views/administrador/dashboard.blade.php --}}

@extends('layouts.app')

@section('content')

<div class="max-w-7xl mx-auto p-6">

<div class="mb-6">

<h1 class="text-2xl font-bold">Panel del Administrador</h1>

<p class="text-gray-600">Gestión global del sistema.</p>

</div>

<div class="grid sm:grid-cols-2 lg:grid-cols-3 gap-6">

<a href="#" class="block rounded-2xl border p-5 hover:shadow">

<h2 class="font-semibold text-lg">Usuarios y Roles</h2>

<p class="text-sm text-gray-600">CRUD de usuarios y permisos.</p>

</a>

<a href="#" class="block rounded-2xl border p-5 hover:shadow">

<h2 class="font-semibold text-lg">Catálogos</h2>

<p class="text-sm text-gray-600">Centros, fondos, convocatorias, ODS, etc.</p>

</a>

<a href="#" class="block rounded-2xl border p-5 hover:shadow">

<h2 class="font-semibold text-lg">Reportes institucionales</h2>

<p class="text-sm text-gray-600">Excel y PDF, métricas con Chart.js.</p>

</a>

</div>

</div>

@endsection

## **✅ Checklist de lo realizado (DoD)**

* **Migración**: se agregó el campo role en la tabla users (1=Usuario, 2=Coordinador, 3=Administrador).
* **Seeder**: se creó UserSeeder.php con los usuarios iniciales (Admin y Coordinador).
* El seeder se ajustó con updateOrCreate/firstOrCreate para **evitar duplicados** al correr db:seed.
* Se actualizó DatabaseSeeder.php para llamar automáticamente a UserSeeder.
* **Middleware de roles**: se creó RoleMiddleware.php con validación de acceso por rol.
* Se registró el middleware role en **Laravel 11** dentro de bootstrap/app.php usando $middleware->alias().
* **Rutas protegidas**: se configuraron rutas para cada rol:  
  + /usuario → rol 1
  + /coordinador → rol 2
  + /admin → rol 3 (en carpeta administrador)
* **Redirección automática después del login**: se creó RedirectController para enviar al usuario a su dashboard según el rol.
* **Dashboards iniciales**: se implementaron vistas Blade con <x-app-layout> para Usuario, Coordinador y Administrador.
* **Validación**: al intentar acceder a una ruta que no corresponde al rol, se muestra **403 - Acceso denegado**.

## 

## 

## **📝 Resumen de la Fase 1**

En esta fase se implementó la **gestión de roles y permisos** en el sistema *Proyectos de Investigación 2025*.  
 Se añadió un campo role a la tabla users para diferenciar los tres perfiles principales: Usuario (1), Coordinador (2) y Administrador (3).  
 Se crearon usuarios base mediante seeders, asegurando que no se generen duplicados en cada ejecución.  
 Se desarrolló un middleware personalizado que restringe el acceso a rutas según rol, registrado en la nueva estructura de Laravel 11 (bootstrap/app.php).  
 Asimismo, se configuraron rutas seguras y un controlador de redirección para que cada usuario acceda directamente a su dashboard al iniciar sesión.  
 Se construyeron dashboards iniciales para cada perfil, adaptados a la estructura de Breeze con <x-app-layout>.  
 Con ello, el sistema ya cuenta con un **control de acceso robusto**, base para continuar con la captura de información del formulario de proyectos (Anexo A).

Comandos para crear, puedes crear el **Modelo, Migración, Controlador y Factory** en **Laravel 11** de una sola vez con el comando Artisan:

php artisan make:model Profesor -mcr

🔎 **Descripción:**

* -m → Crea también la **migración** de la tabla.
* -c → Genera el **controlador**.
* -r → Hace que el controlador sea **resource**, con los métodos CRUD listos (index, create, store, show, edit, update, destroy).

Esto generará:

* **Modelo** → app/Models/Profesor.php
* **Controlador** → app/Http/Controllers/ProfesorController.php
* **Migración** → en database/migrations/xxxx\_xx\_xx\_create\_profesors\_table.php

⚠️ Nota: Laravel por convención crea la tabla en plural (profesors), pero si prefieres **profesores**, edita la migración o indícalo en el modelo:

protected $table = 'profesores';

👉 Si también quieres que genere un **seeder y factory**, puedes usar:

php artisan make:model Profesor -mcrsf

**Crear tabla en la Base de datos**

php artisan migrate

**Poblar Seeders**

php artisan db:seed --class=PfpProfesoresSeeder

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned