

Manual del Sistema - Proyecto de Gestión de Proyectos de Aula

1. Requerimientos del Sistema

Framework: Laravel 11

Base de datos: PostgreSQL

Autenticación: Laravel Breeze (incluye Tailwind CSS)

Seguridad: Spatie Laravel-permission (gestión de roles y permisos)

Node.js y npm: Requeridos para compilar Tailwind CSS

Servidor local: Laravel Artisan

Editor sugerido: Visual Studio Code

2. Creación del Proyecto - Paso a Paso

a. Clonar el repositorio:

- i. `git clone https://github.com/juanjoramos/Proyecto_Pia.git`
- ii. `cd Proyecto_Pia`

b. Instalar dependencias PHP:

- i. `composer install`

c. Instalar dependencias [Node.js](#):

- i. `npm install`

d. Configurar archivo .env:

- i. `cp .env.example .env`


```
DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5432
DB_DATABASE=nombre_de_base_de_datos
DB_USERNAME=usuario
DB_PASSWORD=contraseña
```

e. Configurar php.ini para habilitar extensiones (pdo_pgsql, pgsql):

```
extension=pdo_pgsql
extension=pgsql
```

f. Crear la base de datos con el nombre Proyecto_Pia.

 Create - Database

General	Definition	Security	Parameters	Advanced	SQL
Database	proyecto_pia				
OID					
Owner	 postgres				

- g. Generar clave de aplicación:
 - i. `php artisan key:generate`
- h. Ejecutar migraciones y seeders:
 - i. `php artisan migrate --seed`
- i. Compilar estilos con Tailwind CSS
 - i. `npm run dev`
- j. Levantar servidor local
 - i. `php artisan serve`
- k. Accede al sistema desde: <http://localhost:8000>

3. Puesta en Marcha (Local)

- a. Levantar servidor de [Node.js](#) para Tailwind CSS:
 - i. `npm run dev`
- b. Levantar servidor PHP con laravel:
 - i. `php artisan serve`
- c. Acceder en el navegador a:
 - i. <http://localhost:8000>

4. Paquetes Instalados y Configurados

- Laravel Breeze
 - Proporciona autenticación básica: registro, login, recuperación de contraseña.
 - Incluye interfaz pre-diseñada con Tailwind CSS.
- Spatie Laravel-permission
 - Maneja asignación de roles y permisos.
 - Genera automáticamente las tablas necesarias:
 - roles
 - permissions
 - model_has_roles
 - model_has_permissions
 - role_has_permissions
- Tailwind CSS
 - Permite aplicar estilos responsivos y modernos sin escribir CSS desde cero.
 - Se compila con el comando
 - `npm run dev`

5. Instalación de los paquetes

- Laravel Breeze
 - `composer require laravel/breeze --dev`
 - `php artisan breeze:install`
 - `npm install`
 - `npm run dev`
- Spatie Laravel Permission
 - `composer require spatie/laravel-permission`
 - `php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"`
 - `php artisan migrate`

6. Estructura de Carpetas Importante (Proyecto_Pia)

/app/Models: Modelos del sistema

/app/Http/Controllers: Controladores (CRUDs, autenticación)

/database/migrations: Migraciones de la base de datos

/resources/views: Vistas Blade

/routes/web.php: Rutas del sistema

/public: Archivos públicos (Tailwind compilado, etc.)

7. Notas Finales

- Recordar correr `npm run dev` para compilar los estilos Tailwind CSS.
- Usar `php artisan migrate --seed` para configurar tablas y roles iniciales.
- La carpeta `Proyecto_Pia` es el núcleo del sistema, incluye todo el desarrollo final del PIA.