

# Aplicación Laravel con Landing Page y Gestión de Alumnos

## 1. Descripción general

Vamos a desarrollar una aplicación web utilizando **Laravel** que simula un pequeño sistema de gestión para alumnos de un sistema educativo. Este proyecto luego podrá crecer. La aplicación debe incluir:

- Una **Landing Page** pública.
  - Un sistema básico de **autenticación** (login/logout) utilizando **Blade**.
  - Una zona privada que muestre una **tarjeta de acceso a la gestión de alumnos**. Es posible que aquí incorporemos más tarjetas según avancemos
  - Un **CRUD completo de alumnos**.
- 

## 2. Objetivos del proyecto

El estudiante deberá demostrar que domina:

- La estructura MVC de Laravel.
  - El uso de **rutas, controladores, modelos y migraciones y seeder**.
  - El uso de **Blade** como sistema de plantillas.
  - La creación y utilización de **componentes Blade**.
  - La implementación de un sistema de autenticación básico.
  - La construcción de un CRUD completo (Create, Read, Update, Delete), usando sesiones de un solo uso para su información
  - Uso de traducción a otros idiomas de la aplicación
- 

## 3. Requisitos funcionales

### 3.1 Landing Page (pública)

Debe incluir:

- **Header**, con logotipo, título de la aplicación y botones de Login/Register.
- **Menú de navegación** (About, Contacta, Noticias, Referencias).
- **Sección principal (main)** con un mensaje de bienvenida y un botón de acceso al login.
- **Footer** con copyright y enlaces sociales (iconos ficticios).

Debe estar diseñada **exclusivamente con Blade** y debe utilizar **componentes Blade** (por ejemplo, para el header, footer o tarjetas).

Se propone usar tailwindcss como framework del css, y librerías como daisy que facilitarán la parte gráfica

---

### 3.2 Autenticación (login/logout)

- Debe implementarse usando el sistema de autenticación de Laravel .
  - Login y Logout deben utilizar **vistas Blade**, sin frameworks externos.
  - Solo los usuarios autenticados podrán acceder a las pantallas internas.
  - Igualmente del menú enviar mensaje será una opción solo disponible para usuari@s autenticados
  - Puede haber un usuario creado desde el seeder
- 

### 3.3 Dashboard privado

Una vez el usuario inicia sesión:

- Debe mostrarse una pantalla principal con **tarjetas de navegación**.
  - De momento solo será necesaria **una tarjeta**:
    - “*Gestión de alumnos*” con un botón “Ver alumnos”.
  - Esta tarjeta puede incluir una imagen o ícono libre, y debe construirse preferiblemente con un **componente Blade reutilizable**.
  - Igualmente la opción del menú **enviar mensaje** solo se visualizará si estamos autenticados
- 

### 3.4 CRUD de alumnos

Al entrar en la sección de alumnos, el usuario debe poder:

- **Listar** todos los alumnos.
- **Crear** un alumno (formulario).
- **Editar** un alumno.
- **Eliminar** un alumno (con confirmación en formulario o modal simple).
- **Ver detalles** del alumno (opcional).

#### Campos mínimos del alumno:

- nombre
- apellidos

- email
- dni
- fecha de nacimiento (*f\_nac*)

La tabla debe incluir acciones visuales, como iconos: para editar y para borrar (o los equivalentes FontAwesome). Se puede incluir el uso de esta librería

---

## 4. Requisitos técnicos

- Proyecto Laravel creado correctamente.
  - Migración modelo y seeders de **Alumno**.
  - Controladores separados para el CRUD.
  - Rutas organizadas (públicas y protegidas con middleware).
  - Vistas Blade ordenadas en carpetas (`layouts`, `components`, `alumnos`, `auth`, etc.).
  - Uso de **Blade** (ejemplos: layout general, cabecera, tarjeta, tabla, formulario...).
  - Código limpio y organizado.
- 

## 5. Mockups del proyecto

El documento debe incluir los mockups low-fi proporcionados:

### 1. Landing Page



### 2. Pantalla de Login

The screenshot shows a login interface with the following fields:

- Email: a@a.com
- Password: (redacted)
- Remember me
- [Forgot your password?](#)
- [LOG IN](#)

### 3. Dashboard privado con tarjetas

The dashboard features a central card with the following information:

- CRUD Alumnos!**
- Gestionamos Altas bajas actualizaciones y borrado de una tabla de alumnos
- [Ver alumnos](#)

### 4. Tabla CRUD de alumnos

The table displays the following student information:

id	nombre	f_nac	dni	email	Acciones
1	D. Alex Granado	2009-11-30	75761425-T	patricia71@example.net	Ver
2	Jesús Baca	2020-02-04	14946918-T	marcos53@example.com	Ver
3	Vera Sáez	1981-07-02	25180305-C	martos.biel@example.com	Ver
4	Iker Sepúlveda	2005-01-12	74860851-Z	rocio02@example.org	Ver
5	Dr. Nicolás Negrón	1978-06-02	34775437-N	salvador.eduardo@example.org	Ver

## 6. Entrega

El alumno debe entregar:

- El proyecto Laravel completo. (la url del git donde esté ubicado)
- Un archivo **README.md** indicando:
  - Cómo instalar dependencias.
  - Cómo ejecutar migraciones.
  - Usuario y contraseña de prueba.
  - Todos los comandos php artisan utilizados
- Los mockups entregados diseñados