

# LARAVEL: BLADE, EL GESTOR DE PLANTILLAS



Separamos la vista del resto  
*Mantenmos relación. Ficheros  
diferentes pero pudiendose comunicar*

## La parte del html: Plantillas Blade

Vamos a aprender a utilizar el gestor de plantillas de Blade



- Una herramienta poderosa y flexible incluida en Laravel, que nos va a permitir escribir **html** e incluir **php** y **visualizar datos** del servidor de una *forma elegante y descriptiva*. (Esto facilita visualizar datos del servidor)
- Los ficheros blade, tienen extensión **.blade.php** y estarán ubicados en la carpeta **./resources/view**. Cuando hagamos referencia a los ficheros blade, esta información no hay que especificar, ni su ubicación, ni su extensión:

```
return view('welcome'); //Va a retornar el fichero  
// ./resources/view/welcome.blade.php
```

## La parte del html: Plantillas Blade

- Dentro de un fichero blade podemos encontrar el siguiente tipo de código o instrucciones:



- Código **html y js** (como cualquier página html)
- **{{}}** **Doble braquets** para mostrar el contenido de variables PHP.
  - Blade automáticamente escapa el HTML en las variables para evitar ataques XSS.
- **@** para utilizar directivas/estructuras de control propias de laravel, como como condicionales y bucles. Por ejemplo, **@if**, **@foreach**, **@switch**, entre otras.

# Las vistas

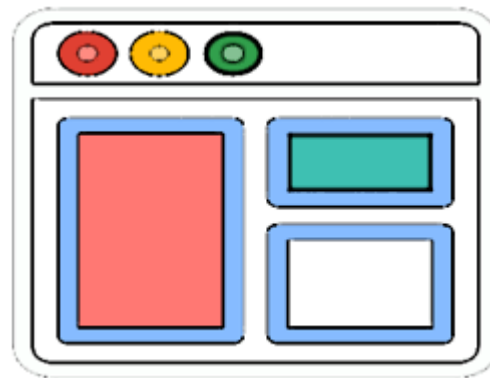
## herencia

**La herencia** en las plantillas Blade proporciona una forma eficiente de garantizar una estructura consistente en todas las páginas, fomentando un diseño corporativo uniforme.

En este enfoque, **se establece una página inicial o 'layout'** que define los aspectos generales de la interfaz, dejando espacios designados para personalizaciones específicas.

Luego, **cada página individual hereda de esta plantilla principal**, permitiendo la inclusión de su contenido único sin perder la coherencia visual.

Este método **facilita la gestión y mantenimiento del diseño**, ya que cualquier cambio realizado en la plantilla principal se refleja automáticamente en todas las páginas que heredan de ella.



Instalar **tailwindcss** node y entender vite



<https://nodejs.org/es/download>



# tailwindcss

<https://tailwindcss.com/docs/installation/framework-guides>



<https://vitejs.dev/>

***Ya viene instalado  
en el proyecto de  
laravel***

Toda esta instalación no es necesaria, si instalamos **breeze**

**Breeze** es un paquete para gestionar *la autenticación* que se explica en el siguiente tema

Breeze, entre otras herramientas, **instala tailwindcss** para los estilos de los formularios que crea para el login y el register

No obstante se explica cómo se instalaría tailwindcss en un proyecto si no instalara breeze

A la vez se explican aspectos del front, como es el uso de **npm**.

## node

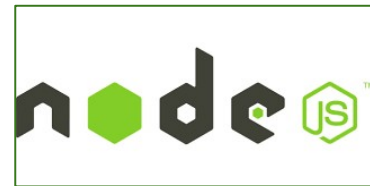
- **node y npm** son Gestores de paquetes para el front
- Van a crear un fichero llamado **package.json** (vs *composer.json*)
- Al ejecutar **npm install** se creará en el proyecto una carpeta llamada **node\_modules**, donde se instalarán los paquetes especificados en el fichero json. Este tampoco hay que subir a git lógicamente.
- **Su instalación es sencilla.** Una vez instalado cerramos el terminal y al volverlo a abrir, ya lo podemos utilizar.

Laravel es una herramienta de **back**.

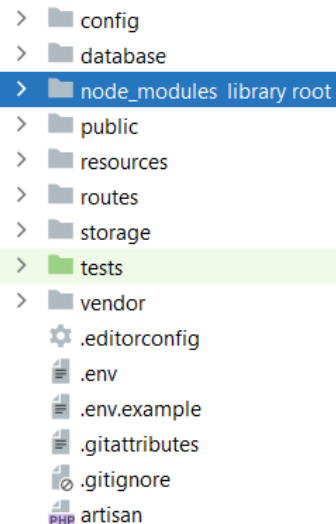
Con laravel podremos realizar un **full-stack**

Permite integrar de forma **agnóstica** cualquier herramienta para el **front**.

Muchos paquetes traen integrado los framework **tailwind** y **vue**



```
C:\Users\Profesor\laravel\estudios>npm install
up to date, audited 108 packages in 957ms
20 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
C:\Users\Profesor\laravel\estudios>
```





El framework

```
npm install -D tailwindcss postcss autoprefixer
```

Instalación en **Desarrollo**, en producción no necesitaremos estos paquetes, ya que tendremos el fichero css transpilado en la carpeta **public**

Herramienta de **transpilación** que a partir del fichero del framework (resources) **generará el css final**

Agregaré prefijos necesarios para **garantizar la compatibilidad** en distintos navegadores

```
npx tailwindcss init -p
```

```
C:\Users\Profesor\laravel\estudios>npx tailwindcss init -p
Created Tailwind CSS config file: tailwind.config.js
Created PostCSS config file: postcss.config.js
C:\Users\Profesor\laravel\estudios>
```





## Instalar tailwindcss

Incluimos las **directivas de preprocesador** para utilizar el framework en `./resources/css/app.css`

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

Estilo base de tailwindcss (h1, p,..)

Estilo para **componentes creados en tailwindcss base**

Clases de **utilidades** (flex,bg-XXX,..)

Especificar dónde van a estar los ficheros que contengan tailwind en **tailwind.config.css**

```
content: [  
  "./resources/**/*.blade.php",  
  "./resources/**/*.js",  
  "./resources/**/*.vue",  
],
```

**\*\*** especifica cualquier subdirectorio que hubiera

Transpilar el css de resources/css/app.css y ponerlos disponibles en la carpeta public para los ficheros

Tenemos un fichero **css** donde se especifican clases de tailwindcss **resources/css/app.css**

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

En nuestras vistas usamos clases de tailwind

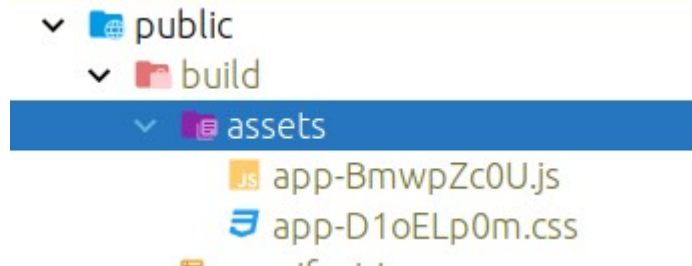
```
<header class="h-15v bg-header flex flex-row  
justify-between p-5 items-center">
```

Hay dos opciones: Opción 1 Para usar en producción, cuando ya esté el proyecto terminado: Generar físicamente los ficheros en la carpeta **public**

**npm run build**

Esta opción tiene el inconveniente que cada vez que incorporemos una nueva clase, hay que realizar la transpilación Y tarda unos segundos.

Esto es porque tailwind solo descarga las clases que usa. Para el desarrollo no es una buena opción, se debe de realizar cuando tengamos nuestro proyecto terminado.



## T Opcion 2 (para usar en el desarrollo)

Crear un servidor en memoria que de forma instantánea actualice el fichero de estilo y lo deje disponible para utilizar

**npm run dev**

Esto nos permite de forma prácticamente instantánea tener actualizado y disponibles las clases que necesitamos, realizando a través de vite una transpilación muy rápida.

→ Local: <http://localhost:5173/>

→ Network: use `--host` to expose

→ press `h + enter` to show help

LARAVEL v10.45.1 plugin v1.0.1

→ APP\_URL: <http://localhost>

# 02 Blade y tailwind

## Tailwindcss: componentes

Hay muchos componentes ya creados que podemos utilizar: <https://tailwindcomponents.com/>

**Hyper UI** <https://www.hyperui.dev/>

Wicked Blocks <https://wickedblocks.dev/>

Flowbite <https://flowbite.com/>

Kutty <https://kutty.netlify.app/>

Tailblocks <https://kutty.netlify.app/>

DaisyUI <https://daisyui.com/>



Es una librería que nos va a permitir utilizar clases que tienen estilos ya predefinidos para nuestro componentes

Vamos a la página oficial y seguimos los pasos de instalación:

<https://daisyui.com/docs/install/>

```
npm i -D daisyui@latest
```

Después añadimos en nuestro fichero **tailwind.config.js**, donde estás establecidos los plugins, que tendremos forms, añadimos la línea **require("daisyui")**

```
plugins: [forms, require("daisyui")],
```

Ahora simplemente usamos sus clases que podemos ver en la web, por ejemplo en botones

```
<a href="about" class=" btn btn-primary"> About</a>
```

En la sección de components, nos muestra los elementos, cómo se ven y qué clases hay que usar, por ejemplo en los botones:

<https://daisyui.com/components/button/>

## # Buttons with brand colors

Preview

HTML

JSX

Button

Neutral

Primary

Secondary

Accent

Ghost

[Link](#)

## # Buttons with brand colors

Preview

HTML

JSX

```
<button class="btn">Button</button>
<button class="btn btn-neutral">Neutral</button>
<button class="btn btn-primary">Primary</button>
<button class="btn btn-secondary">Secondary</button>
<button class="btn btn-accent">Accent</button>
<button class="btn btn-ghost">Ghost</button>
<button class="btn btn-link">Link</button>
```

# Las vistas

## herencia

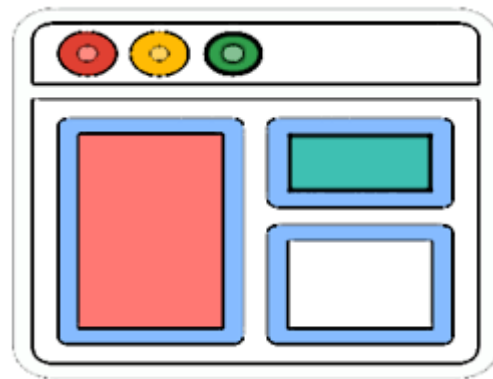
La idea de la herencia en diseño es muy sencilla

1.- Creamos una plantilla con la parte general.

2.- Al crear una **página nueva**, **heredamos** de ella y **añadimos** lo particular

Existen dos formas de conseguir la herencia.

Veremos las dos, pero usaremos los componentes.



# 02 Blade y layout

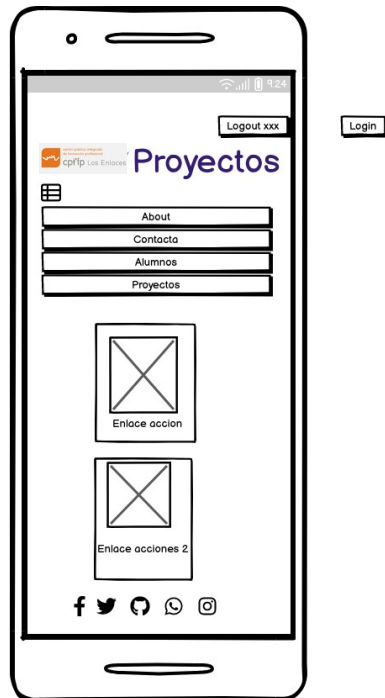
## Diseñando un layout

Vamos a crear un layout general para nuestra aplicación.  
Vamos a desarrollar esta, pero sentiros libres para mejorarla



Logeado como XXXX

Logout





# 02 Blade y layout

## Diseñando un layout

Vamos a crear un layout general para nuestra aplicación.  
Vamos a desarrollar esta, pero sentiros libres para mejorarla



# 02 Blade y layout

Siempre diseño responsive

Podemos probar aquí nuestras páginas (hay muchos sitios para ello

<http://responsivetesttool.com/>

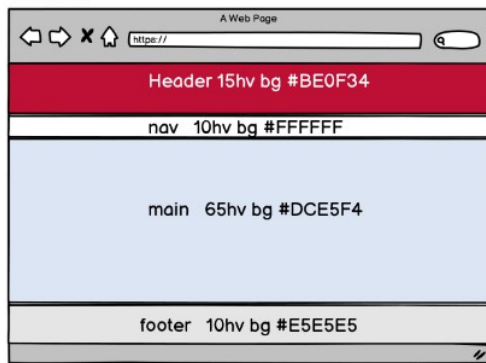
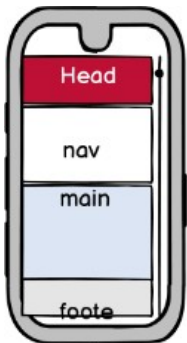
**tailwind**, al igual que bootstrap es **mobile first**

```
'sm': '640px',  
// ⇒ @media (min-width: 640px) { ... }  
  
'md': '768px',  
// ⇒ @media (min-width: 768px) { ... }  
  
'lg': '1024px',  
// ⇒ @media (min-width: 1024px) { ... }  
  
'xl': '1280px',  
// ⇒ @media (min-width: 1280px) { ... }  
  
'2xl': '1536px',  
// ⇒ @media (min-width: 1536px) { ... }
```

En nuestro diseño estableceremos un único punto de ruptura. **md**.

# 02 Blade y layout

## El layout general



```
<!doctype html>
<html lang="en">
  <head>
    @vite(['resources/css/app.css', 'resources/js/app.js'])
  </head>
  <body>
    <header class="lg:hidden bg-header flex flex-col justify-center items-center py-2 space-y-1">
      <img class="h-3/5"
        header para móviles
    </header>
    <header class="hidden lg:flex h-15v bg-header flex flex-row justify-center items-center">
      header para desktop
    </header>
    <nav class="hidden lg:flex h-10v bg-nav flex flex-row justify-start items-center space-x-2">
      Nav móvil
    </nav>
    <nav class="lg:hidden bg-nav flex flex-col justify-start space-y-1 px-2">
      Nav Desktop
    </nav>
    <main class="h-65v bg-main">
      @yield("contenido")
    </main>
    <footer class="h-10v bg-footer footer items-center p-4 bg-neutral text-neutral-content">
    </footer>
  </body>
</html>
```

# 02 Blade y layout

## Pasando datos al layout desde otra página: Creando yield

layout.blade.php

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport"
content="width=device-width, user-scalable=no,...">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<meta name="description" content="@yield('metaDescription')">
<title>@yield('title')</title>

@vite("resources/css/app.css")
</head>
...
<main class="h-65vh
bg-gradient-to-r from-main_left via-red-400 to-main_right
flex flex-row justify-center ">
@yield("main")
</main>
```

about.blade.php

```
@extends("layout")
@section('metaDescription')
Encuentra alumnos de
nettfornación con los proyectos
que realizan en su master de
desarrollo web full stack
utilizando tecnologías
estudiadas
@endsection
@section('title')
About alumnos
@endsection
@section("main")
<h1>Estoy en about</h1>
@endsection
```

Los componentes van a ser ficheros html (.blade.php), que van a residir en la carpeta

**/resources/view/components**

Si no existe la carpeta la creamos

Posteriormente los referenciaremos como elementos html anteponiéndole **<x-**

**nombre.blade.php>**

Es como si la **x-** le dijera a laravel ve a buscar este elemento a la carpeta **components**

```
<!doctype html>
<html lang="en">
<head>
<meta name="description"
content="{{ $metaDescription ?? "valor por defecto" }}">
<title>{{ $title ?? "Alumnos Master" }}</title>
@vite("resources/css/app.css")
</head>
<body>
<header class="h-15vh bg-header flex flex-row justify-between mx-5 p-2"> ... </header>
<nav class=" h-10vh bg-nav, flex flex-row justify-start mx-3 space-x-4"> ... </nav>
<main class="h-65vh ....">
{{ $slot }}
</main>
<footer class="text-white h-10vh bg-footer flex flex-row..."> ... </footer>
</body>
</html>
```

**resources/componets/layout/app.blade.php**

**main.blade.php**

```
<x-layout.app title="Proyectos
Nett"
meta-description="información
especial">
Página principal
</x-layout.app>
```

resources/components/layouts/app.blade.php = <x-layout.app>

```
<!doctype html>
<html lang="en">
<head>
```

```
    <meta name="description"
          content="{{ $metaDescription ?? "Información de proyectos" }}">
    <title>{{ $title ?? "Proyectos" }}</title>
    @vite(["resources/css/app.css"])
```

```
</head>
```

```
<body>
```

```
    <x-layout.header/>
```

```
    <x-layout.nav/>
```

```
    <main class="h-65vh
```

```
    bg-gradient-to-r from-main_left via-red-400 to-main_right
    flex flex-row justify-center ">
```

```
        {{ $slot }}
```


```
    </main>
```

```
    <x-layout.footer/>
```

```
</body>
```

```
</html>
```

Atributos de <x-layout.app>




```
<x-layout.app>
```


```
    Página principal
```

```
</x-layout.app>
```

## Diseño pantalla grande

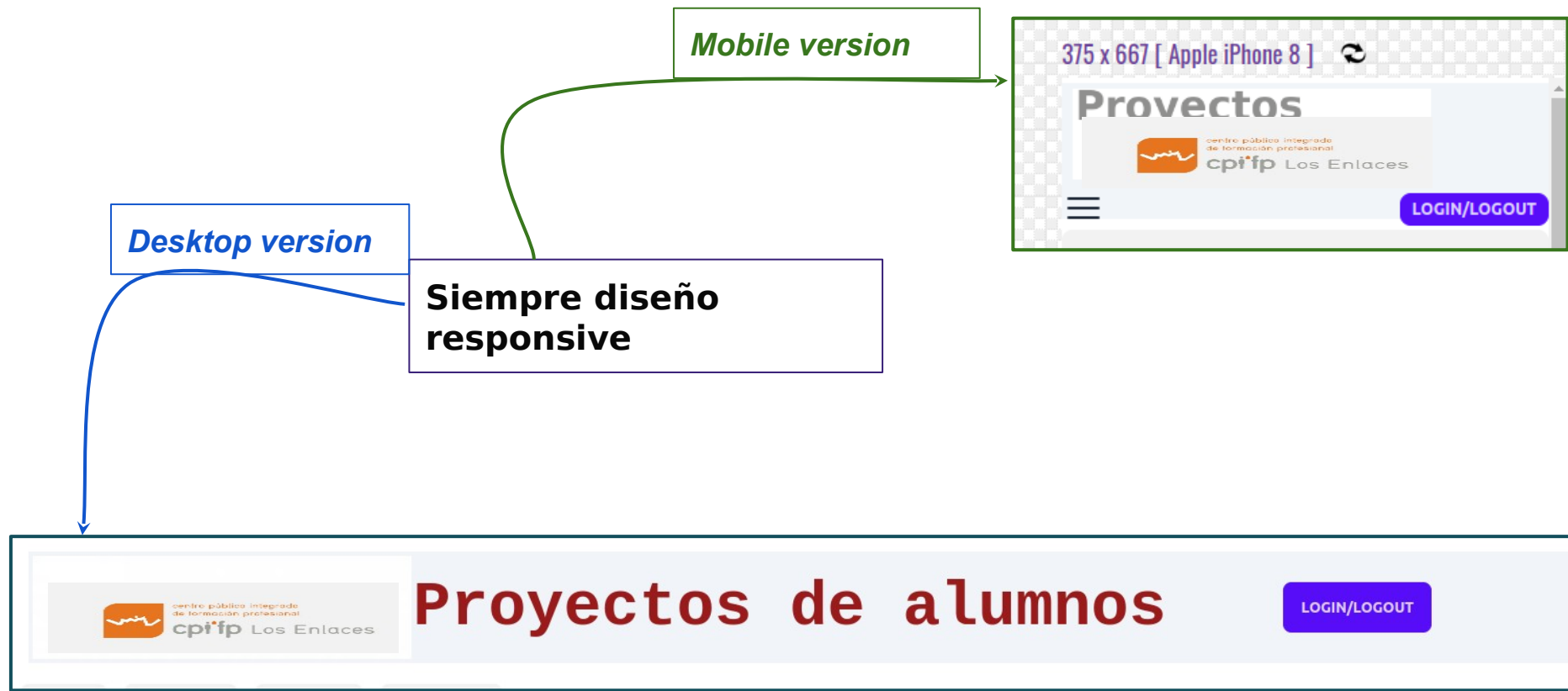
 centro público integrado de formación profesional <b>cpi*fp</b> Los Enlaces	<h1>Proyectos de alumnos</h1>	<input type="text" value="Username"/> <input type="text" value="Password"/> <input type="button" value="Register"/> <input type="button" value="Login"/>
--	-------------------------------	---

 centro público integrado de formación profesional <b>cpi*fp</b> Los Enlaces	<h1>Proyectos de alumnos</h1>	<input type="button" value="Logout username"/>
--	-------------------------------	--

## Diseño móvil

<h2>Proyectos</h2>  centro público integrado de formación profesional <b>cpi*fp</b> Los Enlaces	<h2>Proyectos</h2>  centro público integrado de formación profesional <b>cpi*fp</b> Los Enlaces
<input type="button" value="Login"/> <input type="button" value="Register"/>	<input type="button" value="Logout username"/>



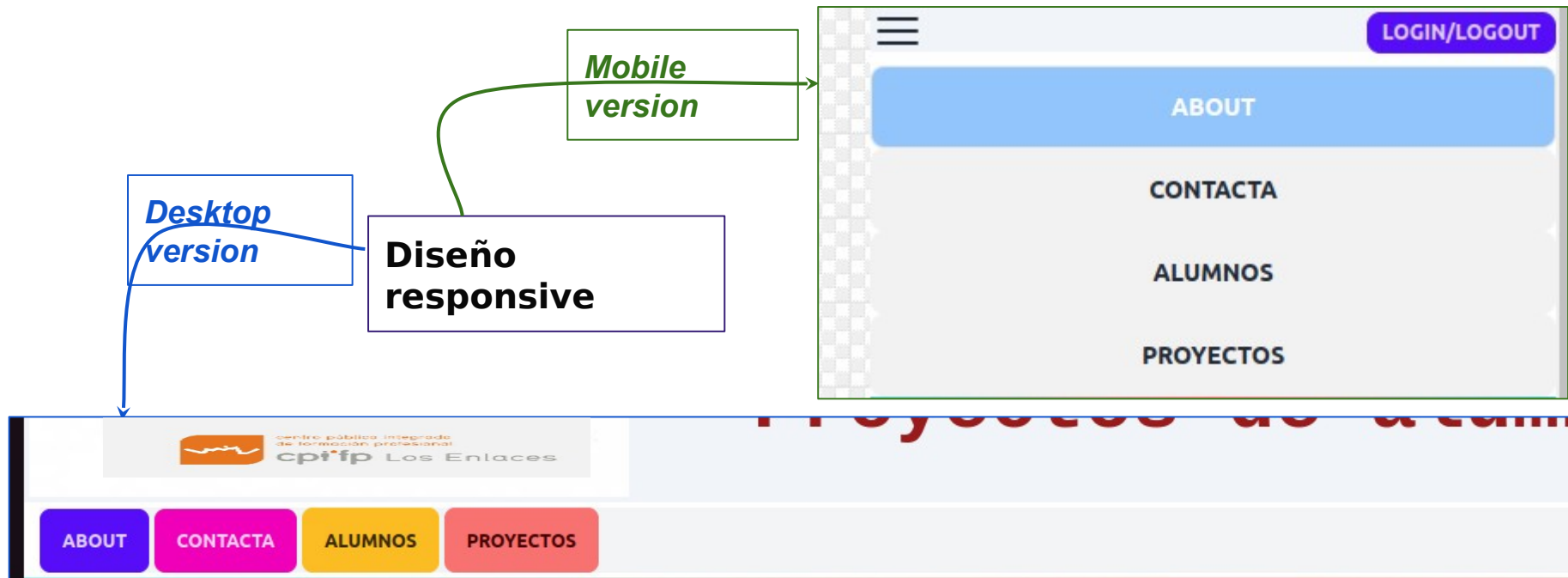


```

<header class="hidden md:flex h-15vh bg-slate-100 flex-row items-center justify-center md:justify-between p-1">
  
  <h1 class="w-2/4 font-mono text-center md:block text-6xl font-bold ext- text-red-800">Proyectos de alumnos</h1>
  <form class=" text-center hidden w-1/4 md:block mr-5" action="">
    <button class="btn btn-primary" type="submit">Login/Logout</button>
  </form>
</header>
{{--Para móbile md--}}
<header class="block md:hidden h-15vh bg-slate-100 flex-col md:flex-row items-center justify-center md:justify-between mb-2 ">
  
  <div class="h-1/4 flex flex-row justify-between items-center ">
    <button >
      <svg class="h-8 w-8 fill="none" stroke="currentColor" stroke-width="1.5" viewBox="0 0 24 24"
        xmlns="http://www.w3.org/2000/svg" aria-hidden="true">
        <path stroke-linecap="round" stroke-linejoin="round"
          d="M3.75 6.75h16.5M3.75 12h16.5m-16.5 5.25h16.5"></path>
      </svg>
    </button>
    <form class="" action="">
      <button class="btn btn-primary btn-xs text-center " type="submit">Login/Logout</button>
    </form>
  </div>
</header>

```

# <x-layout.nav>resources/components/layouts/nav.blade.php = <x-layout.nav>



```
<nav class="hidden md:block h-5vh bg-gray-100 flex flex-row flex-wrap overflow-auto
justify-start mx-3 my-1 ">
  <a class="btn btn-primary " href="{{route('about')}}">About</a>
  <a class="btn btn-secondary " href="contacta">Contacta</a>
  <a class="h-full btn btn-warning" href="alumnos">Alumnos</a>
  <a class="h-full btn btn-error" href="proyectos">Proyectos</a>
</nav>
```

```
<nav class="block md:hidden flex flex-col justify-center" >
  <a class="btn w-full hover:bg-blue-300 hover:text-white transition duration-300"
href="about">About</a>
  <a class="btn w-full hover:bg-blue-300 hover:text-white transition duration-300"
href="contacta">Contacta</a>
  <a class=" btn w-full hover:bg-blue-300 hover:text-white transition duration-300"
href="alumnos">Alumnos</a>
  <a class=" btn w-full hover:bg-blue-300 hover:text-white transition duration-300"
href="proyectos">Proyectos</a>
</nav>
```

# <x-layout.footer>

resources/components/layouts/footer.blade.php = <x-layout.footer>

```

<footer class="bg-gray-200 h-5vh bg-footer flex flex-row justify-center space-x-5 m-3">
<h2>
  @ copyright
</h2>
<a href="facebook">
  <svg class="h-6 w-6 fill-current hover:text-[#1877F2]" . . . </svg>
</a>
<a href="github">
  <svg class="h-6 w-6 fill-current hover:text-[#181717]" . . . </svg>
</a>
<a href="twitter">
  <svg class="h-6 w-6 fill-current hover:text-[#1DA1F2]" . . . </svg>
</a>
<a href="instagram">
  <svg class="h-6 w-6 fill-current hover:text-[#E4405F]" . . . </svg>
</a>
<a href="whatsapp">
  <svg class="h-6 w-6 fill-current hover:text-[#25D366]" . . . </svg>
</a>
</footer>

```

*Mismo diseño mobile Vs desktop con diferente ancho*

Observa otra forma de personalizar clases en tailwind

Tomar los svg de <https://simpleicons.org/>

@ copyright   f   t   r   w   i

Tomaremos un sistema de tarjetas de daisy.com <https://daisyui.com/components/card/>

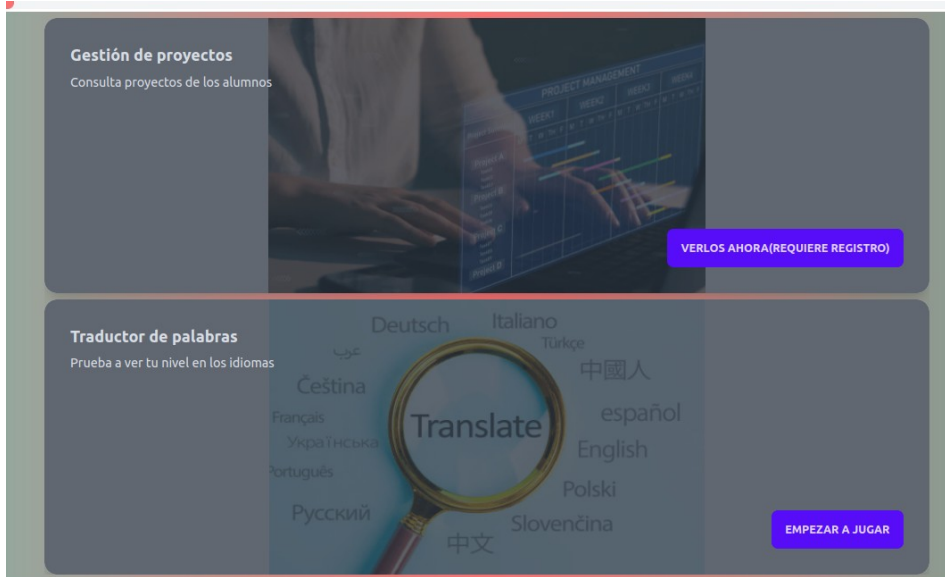
Imágenes gratuitas (cuida mucho el tema de propiedad), optimízalas antes de su uso (Optimizar imágenes para la web <https://tinypng.com/>).

En mobile designer, dejaremos el mismo diseño que se adaptará

Añadimos dos acciones planteadas para este curso:

**Gestión de proyectos** (Base de datos)

**Traductor de palabras** (imágenes y front)



```
<x-layout.app
    title="Proyectos Master" meta-description="Proyectos de alumnos del master desarrollo
        web fullstak en Nettformación promoción 2022">
    web fullstak en Nettformación promoción 2022">
<div class="flex flex-col justify-between m-2 space-y-2 overflow-auto">
    <div class="card w-full h-full bg-base-100 shadow-xl image-full">
        <figure></figure>
        <div class="card-body">
            <h2 class="card-title">Gestión de proyectos</h2>
            <p>Consulta proyectos de los alumnos</p>
            <div class="card-actions justify-end">
                <button class="btn btn-primary">Verlos ahora(requiere registro)</button>
            </div>
        </div>
    </div>
</div>
<div class="card w-full h-full bg-base-100 shadow-xl image-full h-full ">
    <figure></figure>
    <div class="card-body">
        <h2 class="card-title">Traductor de palabras</h2>
        <p>Prueba a ver tu nivel en los idiomas</p>
        <div class="card-actions justify-end">
            <button class="btn btn-primary">Empezar a jugar</button>
        </div>
    </div>
</div>
</div>
</x-layout.app>
```

# 02 Referencias Web

Diseño responsive <http://responsivetesttool.com/>  
Balsamiq diseño de pantallas <https://balsamiq.cloud/>  
Página oficial de laravel <https://laravel.com/>  
Página oficial de tailwindcss <https://tailwindcss.com/>  
Componentes de daisyui <https://daisyui.com/docs/install/>  
Iconos en general tipo awesome en formato svg <https://heroicons.dev/>  
Iconos de empresas/productos formato svg <https://simpleicons.org/>  
Api para avatar en letras <https://ui-avatars.com/api>  
Optimizar imágenes para la web <https://tinypng.com/>