

Seguimos con el proyecto y continuamos con la creación de vistas

1.- crear vista

Vamos a crear una primera vista de prueba porque posteriormente la convertiremos en una vista de Blade.

Hasta ahora los mensajes que estamos mostrando: primero lo mostraba la propia ruta luego realizamos el cambio para que fuese el controlador para que finalmente sea una página web la que tiene que realizar la acción.

Vamos a crear la página de inicio de la web con extensión blade.php, nada más crear el documento si ponemos ! nos escribe el esqueleto de html. Simplemente crearemos un h1 con el mensaje

2.- Modificar el controlador

Modificaremos el controlador InicioController para que dentro del método se llame a la página creada en el punto anterior.

3.- Prueba que funciona la ruta

4.- Instalamos Tailwindcss en laravel

En visual studio code abrimos el proyecto y una terminal

Recordar que instalamos Node.js porque contiene npm responde a las siglas de **Node Package Manager** o manejador de paquetes de node, es la herramienta por defecto de JavaScript para la tarea de compartir e instalar paquetes. El archivo **package.json**, este archivo indica a npm que el directorio en el que se encuentra es un proyecto o paquete npm.

Adjunta las pantallas

1.- Instalamos npm en el proyecto ejecutar el comando **npm install**

2. Vamos a desinstalar Bootstrap

Realizamos los siguientes pasos: ir a la carpeta resources/js/ abrimos el archivo app.js y comentar el import de Bootstrap

Comprobamos también que el archivo app.css que figura en resources/css no tenga información tampoco.

Desinstalamos Bootstrap, ejecutamos el comando **npm uninstall bootstrap**

3. Instalamos Tailwindcss: en la página oficial de tailwindcss encontramos los pasos a seguir:

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

una vez en la guía de Laravel, lo primero que nos dice es crear un proyecto de Laravel pero, en nuestro caso ya lo tenemos creado omitimos este paso.

- a) Instalamos Tailwind CSS, los comandos a ejecutar son:

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

```
npx tailwindcss init -p
```

- b) Configuramos las plantillas que utilizaran Tailwind CSS en el fichero **tailwind.config.js**
Proponen los ficheros que se encuentren en estas direcciones y con esas extensiones

```
"/resources/**/*.blade.php",  
"/resources/**/*.js",  
"/resources/**/*.vue",
```

- c) Añadir las directivas de Tailwind CSS en el fichero **./resources/css/app.css**

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

Me indicará un fallo en Tailwind y es porque me falta instalar extensiones en visual studio code.

Para que reconozca las directivas instalar la extensión



PostCSS Language Support v1.0.9

csstools | 404.342 | ★★★★★ (19)

Syntax highlighting for modern and experimental CSS in VSCode

[Deshabilitar](#) [Desinstalar](#) ⚙️

Esta extensión está habilitada globalmente.

Para las ayudas en Tailwind CSS instalar



Tailwind CSS IntelliSense v0.9.1 Vista Previa

Tailwind Labs | 2.179.026 | ★★★★★ (71)

Intelligent Tailwind CSS tooling for VS Code

[Deshabilitar](#) [Desinstalar](#) ⚙️

Esta extensión está habilitada globalmente.

- d) Para poder ir viendo los cambios que realizamos en el css tendremos una terminal abierta donde ejecutaremos el comando **npm run dev**

y cuando nuestros cambios queremos que pasen a definitivo ejecutamos el comando **npm run build**, que generara un archivo *.css , un archivo *.js y un archivo manifest.json , solo con las clases de Tailwind CSS que hay definidas en el proyecto. En el directorio público del proyecto.

5. – Realizamos una prueba de aplicación de estilos

Vamos a modificar la vista creada y poner la directiva para que recoja los estilos dentro de la etiqueta head **@vite('resources/css/app.css')**

Y vamos a poner unos estilos por ejemplo modificamos el h1 con **<h1 class="text-3xl font-bold underline">**

Para poder ver el css, tenemos que arrancar vite con npm run dev y para ver los cambios podemos seguir utilizando el servidor de Laravel o bien el servidor de Apache. Adjunta ambas pruebas.

6.- realiza la compilación para genere los archivos finales como si tuviéramos que subir ya el proyecto a producción y observa los ficheros que ha generado. Explica los pasos realizados donde se encuentra y que contienen.