

Frontend Basico

Para el manejo de el website y panel de administracion se usara el Framework de Vuejs en conjunto con su librería VueRouter, con la que se manejaran las direcciones y la mayoría de las acciones de el sitio web.

Mas informacion de Vuejs:

<https://es-vuejs.github.io/vuejs.org/v2/guide/>

Mas informacion de VueRouter:

<https://router.vuejs.org/es/>

Template:

Este proyecto trabaja bajo un motor de plantillas hechas con blade de Laravel. A partir del template se utilizaran los componentes de VueRouter para generar todas las rutas del website.

Por recomendación, se propone usar el siguiente archivo como template del website:

resources\views\page.blade.php

Componentes

Con el uso de VueRouter usaremos componentes para generar cada una de las vistas, por lo que cada vista sera un componente de Vue. Los componentes estan en la ruta:

resources\assets\js\components\

Los componentes estan separados por dos carpetas; “*page*” y “*admin*”. Donde “*page*” seran los componentes del website y “*admin*” los componentes del panel de administracion.

Estilos

Los estilos del website pueden ser manejados ya sea con *less* o *sass*, estos podemos encontrarlos en sus rutas respectivas:

Para LESS:

resources\assets\less\

Para SASS:

resources\assets\sass\

Uso de webpack y laravel-mix

Para que el sitio web funcione con lo programado en Vuejs y los estilos hechos (en less o sass). Es necesario convertir el codigo de Vuejs a JS y los estilos a CSS.

Para eso el proyecto hace uso de la herramienta que Laravel propone que es laravel-mix. Laravel-mix convertira nuestro codigo de Vuejs y lo recopilara en un solo archivo (que debe subirse al servidor en caso de estar trabajando en uno). El archivo convertido puedes encontrarlo en:

Si es JS:	public/js
Si es CSS:	public/css

Para hacer uso de el se tiene que abrir el proyecto desde la terminal y ejecutar uno de los siguientes comandos:

Conversion a desarrollo:

```
npm run dev
```

Este comando convertira el codigo a modo de desarrollo. Con el podemos debuggear los errores y ver y modificar variables de Vue en tiempo de ejecucion.

Conversion a produccion:

```
npm run prod
```

Con este comando se convertira el codigo a modo de produccion, por lo que nos arrojará una conversion comprimida y que no puede ser debuggeable (es mucho mas liviano y seguro).

Conversion a desarrollo automatica:

```
npm run watch
```

Mientras este corriendo esta funcion, laravel-mix convertira nuestros archivos automaticamente a desarrollo cada vez que ejecutemos un cambio en nuestros archivos.

Configuracion de webpack

Por defecto laravel-mix convertira todos los archivos (archivos Vue y estilos del website y del panel de administracion). Por lo que puede ser muy lenta la conversion.

Para configurar que webpack convierta solo los archivos que necesitamos en ese momento, se debe modificar el siguiente archivo:

```
/webpack.mix.js
```

Aqui podemos comentar las lineas que no necesitamos, para que webpack solo convierta el codigo que necesitamos en ese momento.

Al hacer uso de npm (gestor de paquetes JS), tenemos la posibilidad de tener acceso a muchos plugins y librerias extras. Puedes ver todas las que existen en su pagina oficial:

<http://npmjs.com/>

Plugins y librerias ya incluidas

Para el funcionamiento basico del sitio, estan ya incluidas las librerias mas utiles y usadas.

- [jQuery \(librería JS\)](#)
- [Bootstrap \(estilos CSS\)](#)
- [BootstrapTable \(tablas dinamicas\)](#)
- [Axios \(peticiones AJAX\)](#)
- [PushJS \(notificaciones de escritorio\)](#)
- [AlertifyJS \(notificaciones y mensajes de alerta\)](#)
- [VueValidate \(validaciones en el cliente de vue\)](#)
- [VueDatetime \(dateTime al estilo android\)](#)
- [VueSelect \(selects html dinamicos\)](#)
- [VueCarousel \(Carrouseles con Vue\).](#)

Cada una esta pensada para su facil uso y convivencia con VueJS. Y se pueden añadir muchas mas librerias si es necesario.