

2- Configuración básica para este proyecto

Luego de la introducción presentada en el [artículo anterior](#) acerca de como trabajaremos en el proyecto, y de instalar nuestra versión de Laravel en blanco, es hora de codear.

Por ahora, solo presentaremos los archivos que se deben configurar y como para rápidamente poder seguir en el siguiente post con nuestro layout y la vista que usaremos para el proyecto.

Miremos primero que nada como queda el **package.json**:

```
"devDependencies": {
  "axios": "^0.16.2",
  "cross-env": "^5.0.1",
  "laravel-mix": "0.*",
  "lodash": "^4.17.4",
  "vue": "^2.1.10"
},
"dependencies": {
  "bulma": "^0.4.2"
}
```

Los cambios con respecto al **package.json** original que viene con Laravel es la eliminación de **jQuery** y **Bootstrap**. Y hemos agregado **Bulma**. Para agregar **Bulma** se puede hacer manualmente pero aconsejamos hacer a través de:

```
npm i --save bulma
```

Lo que asegura que se instale la última versión disponible.

Para incluir **Bulma** en el **Webpack** cambiaremos el **resource/assets/sass/app.scss**

```
@import "~bulma/sass/utilities/_all";
@import "~bulma/sass/base/_all";
@import "~bulma/sass/elements/_all";
@import "~bulma/sass/components/_all";
@import "~bulma/sass/grid/_all";
@import "~bulma/sass/layout/_all";
```

Cambiaremos también el **resource/assets/js/app.js**

```
require('./bootstrap');
window.Vue = require('vue');
```

y el **resource/assets/js/bootstrap.js**

```
window._ = require('lodash');
window.axios = require('axios');
window.axios.defaults.headers.common['X-Requested-With'] = 'XMLHttpRequest';
let token = document.head.querySelector('meta[name="csrf-token"]');
if (token) {
  window.axios.defaults.headers.common['X-CSRF-TOKEN'] = token.content;
} else {
  console.error('CSRF token not found: https://laravel.com/docs/csrf#csrf-x-csrf-token');
}
```

Casi todo el código expuesto aquí es el que viene originalmente con **Laravel**. Es más lo que se ha sacado que lo que se ha incluido.

Como novedad he creado un archivo CSS para ajustar algunos estilos. Lo he colocado en **resource/assets/css/style.css**

```

.personal-menu{
  background-color: #000;
  color:#fff;
}
.personal-content {
  background-color:#fff;
  color:#000;
}
.text-center {
  text-align: center;
  justify-content: center;
}
.text-danger {
  color:#800000;
  font-size:12px;
}
.vertical-center {
  display: flex;
  align-items: center;
  justify-content: center;
}
.vertical-top {
  display: flex;
  align-items: flex-start;
  justify-content: center;
}
.vertical-bottom{
  display: flex;
  align-items: flex-end;
  justify-content: center;
}
.hand-option{
  cursor: pointer;
}
.margin0 {
  margin:0;
}
.padding0 {
  padding:0;
}
.line-der {
  border-right:1px solid #000;
}
.modal-content {
  background-color: #fff;
  padding:20px;
}
.button-content {
  margin-top: 15px;
  text-align: center;
}
div.cov-vue-date {
  width:100%;
}
div.cov-vue-date div {
  width:100%;
}
div.cov-vue-date div input {
  width:100%;
}
.select {
  min-width: 100%;
  width:100%;
}

```

Para no dar vueltas alrededor del css no lo explicaremos ya que solo son detalles estéticos pero si es necesario se puede explicar más adelante.

Con esto ya casi terminamos de modificar archivos. Solo nos queda el **webpack.mix.js**

```

const { mix } = require('laravel-mix');
mix.js('resources/assets/js/app.js', 'public/js')
  .sass('resources/assets/sass/app.scss', 'public/css')
  .combine(['resources/assets/css/style.css'], 'public/css/style.css');

```

Bien. Ahora el trabajo es del terminal. Una vez configurado todo esto ya podemos hacer un

```
npm install
```

y posteriormente

```
npm run dev
```

Tenemos todo preparado para poder crear el layout y la única vista que usaremos en este proyecto.

13/6/2019

Creando un sistema CRUD con Vue y Laravel 2/23 - Laraveles

Código disponible en [Github](#)

En el siguiente post. [Creando el layout](#)