

Integración de Vue.js 3 con Laravel 11/12.

::: By Freddy Alcarazo :::

@alcarazolabs

Detalle:

En esta guía veremos como integrar vue.js a un proyecto laravel. No se va a utilizar otras tecnologías como inertia.js o livewire solo (laravel, laravel breeze, vue.js y tailwind css). Lo de usar tailwind css es opcional ya que viene con laravel breeze, lo puntual de esta guía es implementar vue.js

Crear proyecto laravel

```
$ composer create-project laravel/laravel laravel12app "12.*"
```

Instalar laravel breeze

```
$ composer require laravel/breeze --dev
```

```
$ php artisan breeze:install
```

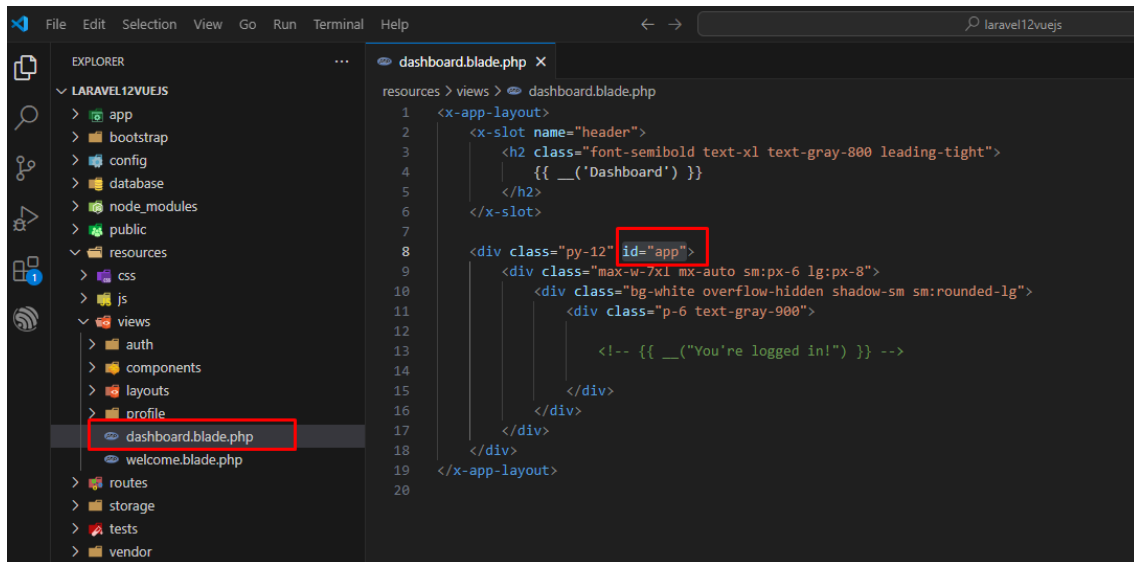
```
$ php artisan migrate
```

```
$ npm install
```

```
$ npm run dev
```

Nota: Luego de correr "npm run dev" probar la url que funcione luego detenerlo con ctrl+C. Luego ejecutar "php artisan serve" para probar.

Establecer id app a la vista dashboard.blade.php



Aquí es donde se mostrarán los componentes vue.js solo falta agregar un router-view o poner directamente el nombre de algún componente para que se muestre. ¡Esto lo vamos a agregar en donde esta comentado “you are logged in!”.

Configuración Vue.js

Por defecto, Breeze agrega Alpine.js, que no usaremos. Por lo tanto, modificamos package.json, presente en la raíz de Laravel, y eliminamos Alpine.js.

En devDependencies de package.json, eliminamos la línea.

```
{
  "private": true,
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build"
  },
  "devDependencies": {
    "@tailwindcss/forms": "^0.5.2",
    "alpinejs": "^3.4.2",
    "autoprefixer": "^10.4.2",
    "axios": "^1.6.4",
    "laravel-vite-plugin": "^1.0",
    "postcss": "^8.4.31",
    "tailwindcss": "^3.1.0",
    "vite": "^5.0"
  }
}
```

Ahora podemos instalar Vue y Vue Loader y configurar Vue:

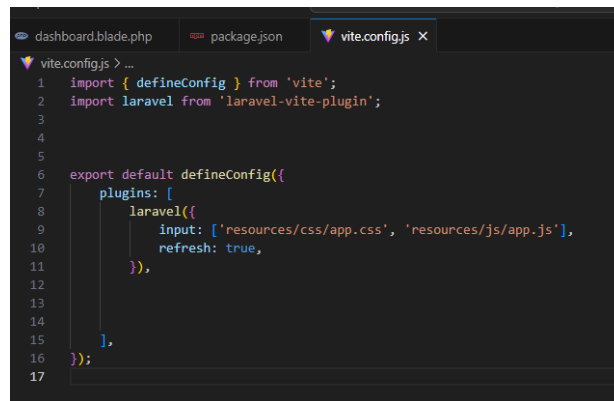
```
$ npm install vue vue-loader
```

A continuación, necesitamos instalar el complemento Vue Vite:

\$ npm install --save-dev @vitejs/plugin-vue

Y ahora editemos el archivo **vite.config.js** presente en la raíz de laravel, de la siguiente manera:

Por defecto tiene el siguiente contenido:



```
1 import { defineConfig } from 'vite';
2 import laravel from 'laravel-vite-plugin';
3
4
5
6 export default defineConfig({
7   plugins: [
8     laravel({
9       input: ['resources/css/app.css', 'resources/js/app.js'],
10      refresh: true,
11    }),
12  ],
13
14
15
16 });
17
```

Agregar lo siguiente:

```
import { defineConfig } from 'vite';
import laravel from 'laravel-vite-plugin';

import vue from '@vitejs/plugin-vue'; // <-- agregar esto

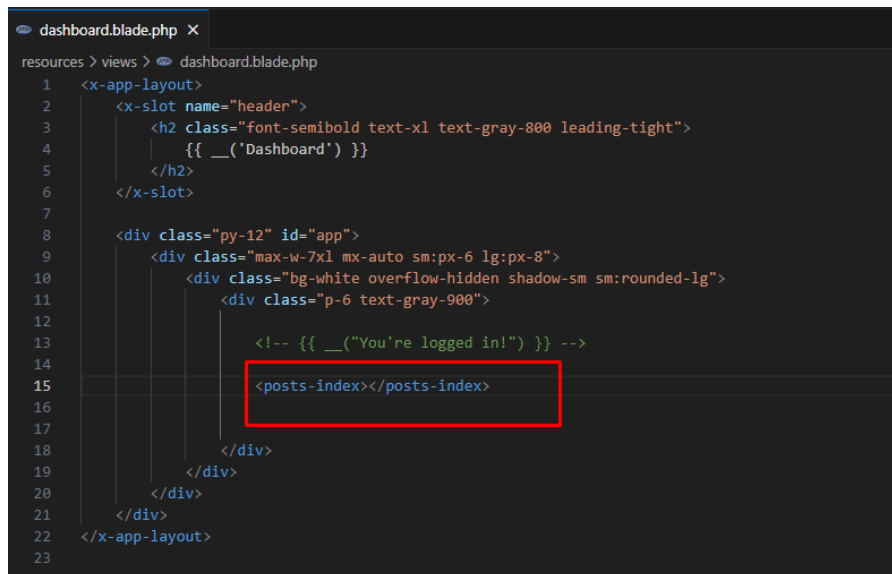
export default defineConfig({
  plugins: [
    laravel({
      input: ['resources/css/app.css', 'resources/js/app.js'],
      refresh: true,
    }),
    vue({
      template: {
        transformAssetUrls: {
          base: null,
          includeAbsolute: false,
        },
      },
    }),
  ],
  resolve: {
    alias: {
      vue: 'vue/dist/vue.esm-bundler.js',
    },
  },
});
```

Nota: Este código de **vite.config.js** lo pueden encontrar en mi repositorio:

<https://github.com/alcazozlabs/surflaweb-scripts/blob/main/larave-12-vite.config.js>

Creación de componentes Vue

Agreguemos un componente Vue en la página Dashboard. Editemos el archivo `resources/views/dashboard.blade.php` de la siguiente manera:



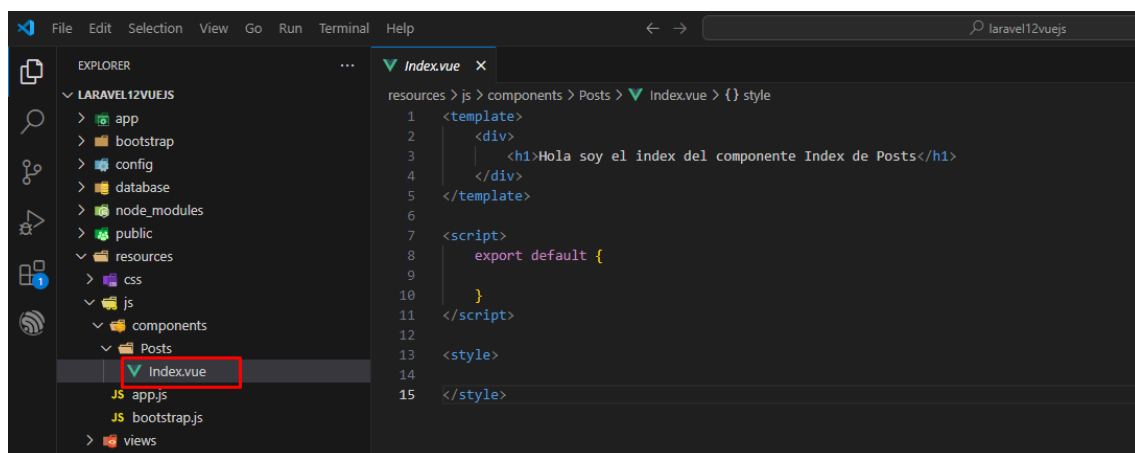
```
resources > views > dashboard.blade.php
1 <x-app-layout>
2   <x-slot name="header">
3     <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4       {{ __('Dashboard') }}
5     </h2>
6   </x-slot>
7
8   <div class="py-12 id="app">
9     <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10      <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
11        <div class="p-6 text-gray-900">
12          <!-- {{ __("You're logged in!") }} -->
13          <posts-index></posts-index>
14        </div>
15      </div>
16    </div>
17  </div>
18</x-app-layout>
```

En este código estamos indicando que se mostrará el componente llamado “post-index”.

Ahora procedamos a crear el componente Vue en:

`resources/js/components/Posts/Index.vue`

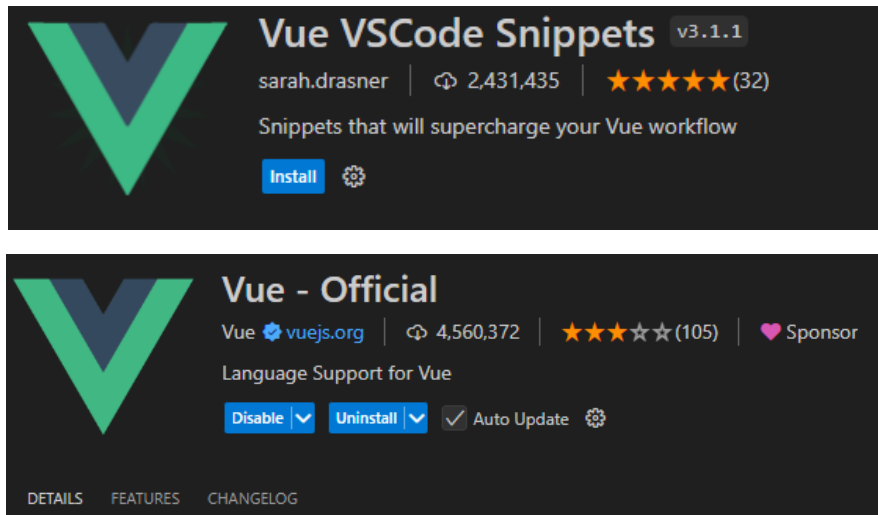
Por ahora, solo insertaremos texto estático ficticio dentro de él. Editemos el archivo `resources/js/components/Posts/Index.vue`



```
resources > js > components > Posts > Index.vue > {} style
1 <template>
2   <div>
3     <h1>Hola soy el index del componente Index de Posts</h1>
4   </div>
5 </template>
6
7 <script>
8   export default {
9
10  }
11 </script>
12
13 <style>
14
15 </style>
```

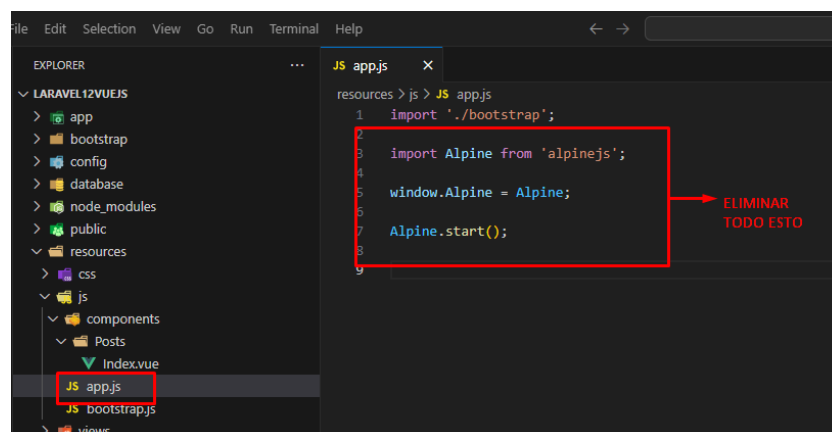
Nota: La carpeta components no existía, se ha te nido que crear y de la misma manera la carpeta Posts y el archivo Index.vue

Extensiones recomendadas para visual studio code para trabajar con vue.js

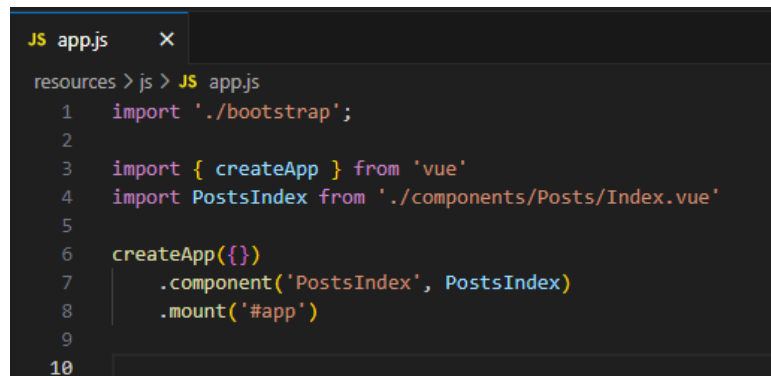


Continuemos...

Ahora, volviendo al elemento id="app", procedamos a inicializar una aplicación Vue, agreguemos el componente de arriba y móntarlo en ese elemento de aplicación. Todo esto debe hacerse en el archivo principal resources/js/app.js. De forma predeterminada, como se define con Breeze, **el archivo app.js contiene código que utiliza Alpine.js** y, por elección de implementación, **lo eliminamos**.



El archivo resources/js/app.js se convierte en lo siguiente:



```
JS app.js x
resources > js > JS app.js
1  import './bootstrap';
2
3  import { createApp } from 'vue'
4  import PostsIndex from './components/Posts/Index.vue'
5
6  createApp({})
7    .component('PostsIndex', PostsIndex)
8    .mount('#app')
9
10
```

Código:

```
import './bootstrap';

import { createApp } from 'vue'
import PostsIndex from './components/Posts/Index.vue'

createApp({})
  .component('PostsIndex', PostsIndex)
  .mount('#app')
```

En el archivo **app.js** definimos la aplicación Vue con `createApp()`, importándola de antemano. Adjuntamos un componente a la aplicación Vue, importándolo de antemano y dándole el nombre `PostIndex`. Montemos la aplicación Vue en el elemento `#app` del diseño principal. Así es como le damos el nombre `PostIndex` al componente para que sepas que es un componente para Posts y es un archivo `Index.vue`. Para llamar a los componentes en los archivos Vue, hay dos formas:

- Kebab case
- Pascal Case

Más información sobre estos: <https://www.freecodecamp.org/news/snake-case-vs-camel-case-vs-pascal-case-vs-kebab-case-whats-the-difference/>

Vue.js admite ambos casos, por lo que para nombrar los componentes usamos `PascalCase`.

Ahora vamos a compilar todo usando `npm run dev` o `npm run build`, cuando terminemos, abrimos la página en el navegador y deberíamos ver el texto del componente Index, de la siguiente manera:

Recompilación de JS y uso de Tailwind CSS

Al trabajar con cualquier archivo del front-end, necesitamos recompilar después de cada cambio para ver los cambios en el navegador. Por lo tanto, debemos ejecutar `npm run build` cada vez que hay a un pequeño cambio, **o mejor aún, ejecutar `npm run dev`** y mantenerlo ejecutándose en segundo plano.

Resultado:



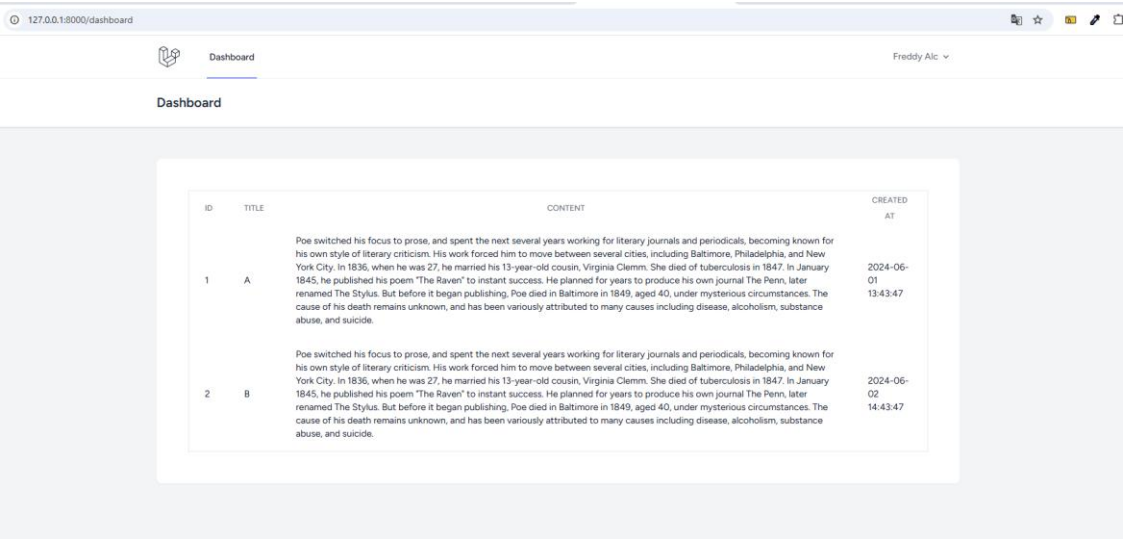
Cuando ejecutamos `npm run dev` con Vite, después de cada recompilación, el navegador se actualizará automáticamente.

Usando estilos de tailwind css

Ahora intentemos una modificación simple en el archivo Vue: agreguemos una tabla estática simple (por ahora) al componente PostIndex Vue.

```
JS app.js Indexvue X
resources > js > components > Posts > indexvue > {} script > @ default
1 <template>
2 <div class="overflow-hidden overflow-x-auto p-6 bg-white border-gray-200">
3 <div class="min-w-full align-middle">
4 <table class="min-w-full divide-y divide-gray-200 border">
5 <thead>
6 <tr>
7 <th class="px-6 py-3 bg-gray-50 text-left">
8 <span class="text-xs leading-4 font-medium text-gray-500 uppercase tracking-wider">ID</span>
9 </th>
10 <th class="px-6 py-3 bg-gray-50 text-left">
11 <span class="text-xs leading-4 font-medium text-gray-500 uppercase tracking-wider">Title</span>
12 </th>
13 <th class="px-6 py-3 bg-gray-50 text-left">
14 <span class="text-xs leading-4 font-medium text-gray-500 uppercase tracking-wider">Content</span>
15 </th>
16 <th class="px-6 py-3 bg-gray-50 text-left">
17 <span class="text-xs leading-4 font-medium text-gray-500 uppercase tracking-wider">Created at</span>
18 </th>
19 </tr>
20 </thead>
21 <tbody class="bg-white divide-y divide-gray-200 divide-solid">
22 <tr>
23 <td class="px-6 py-4 whitespace-no-wrap text-sm leading-5 text-gray-900">1</td>
24 <td class="px-6 py-4 whitespace-no-wrap text-sm leading-5 text-gray-900">A</td>
25 <td class="px-6 py-4 whitespace-no-wrap text-sm leading-5 text-gray-900">Poe switched his focus to prose, and spent the next several years working for literar
26 <td class="px-6 py-4 whitespace-no-wrap text-sm leading-5 text-gray-900">2024-06-01 13:43:47</td>
27 </tr>
28 <tr>
29 <td class="px-6 py-4 whitespace-no-wrap text-sm leading-5 text-gray-900">2</td>
30 <td class="px-6 py-4 whitespace-no-wrap text-sm leading-5 text-gray-900">B</td>
31 <td class="px-6 py-4 whitespace-no-wrap text-sm leading-5 text-gray-900">Poe switched his focus to prose, and spent the next several years working for literar
32 <td class="px-6 py-4 whitespace-no-wrap text-sm leading-5 text-gray-900">2024-06-02 14:43:47</td>
33 </tr>
34 </tbody>
35 </table>
36 </div>
37 </template>
38
39 <script>
40 export default {
41 }
42 </script>
43
44 <style>
45
46
47
48 </style>
```

Código disponible en: <https://github.com/alcarazolabs/surflaweb-scripts/blob/main/tabla-tailwind-example.vue>




Pero ahora tenemos un problema con Tailwind: no recoge los nuevos estilos CSS que hemos añadido. Necesitamos añadir la ubicación de los componentes de Vue al archivo de configuración de Tailwind, para que recopile y compile automáticamente las clases utilizadas en Vue. Editemos el archivo **tailwind.config.js**:

Agregar:

```
'./resources/js/components/**/*.vue',
```

```
JS app.js tailwind.config.js x
tailwind.config.js > ...
1 import defaultTheme from 'tailwindcss/defaultTheme';
2 import forms from '@tailwindcss/forms';
3
4 /** @type {import('tailwindcss').Config} */
5 export default {
6   content: [
7     './vendor/laravel/framework/src/Illuminate/Pagination/resources/views/*.blade.php',
8     './storage/framework/views/*.php',
9     './resources/views/**/*.blade.php',
10    './resources/js/components/**/*.vue',
11  ],
12
13  theme: {
14    extend: {
15      fontFamily: {
16        sans: ['Figtree', ...defaultTheme.fontFamily.sans],
17      },
18    },
19  },
20
21  plugins: [forms],
22 };
23
```

Ahora notar que los nombres de las columnas aparecen no centradas.

 Dashboard

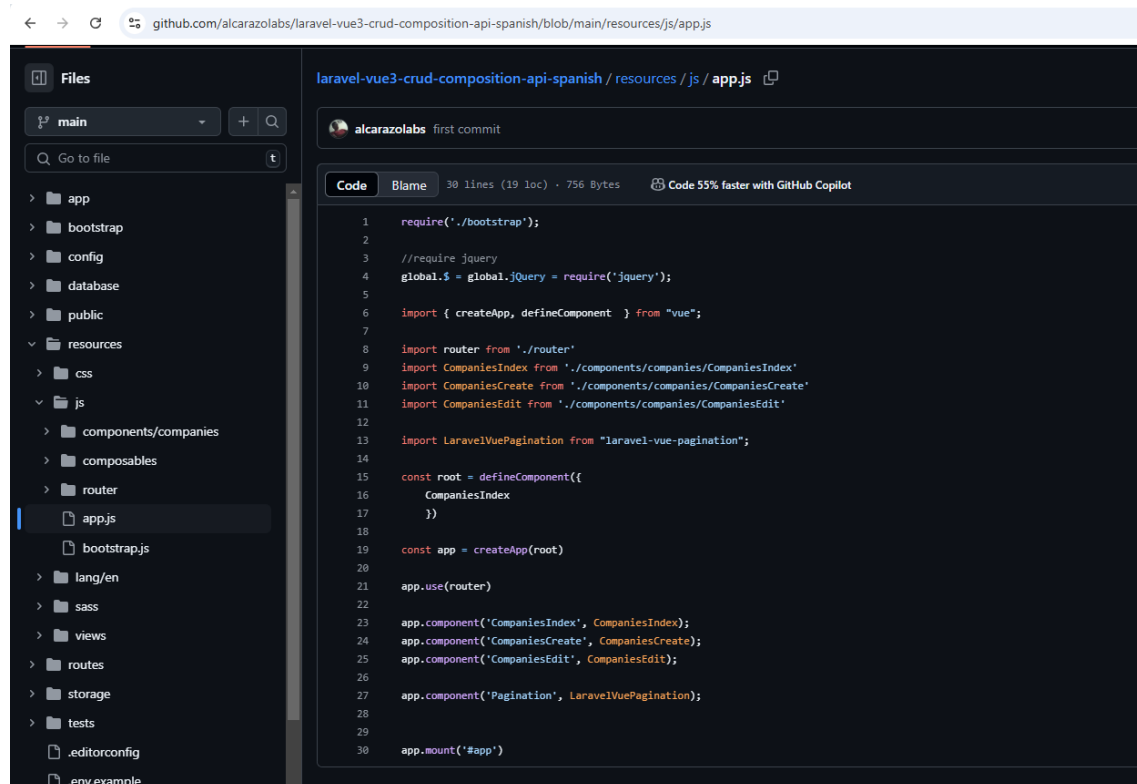
Freddy Alc ▾

Dashboard

ID	TITLE	CONTENT	CREATED AT
1	A	Poe switched his focus to prose, and spent the next several years working for literary journals and periodicals, becoming known for his own style of literary criticism. His work forced him to move between several cities, including Baltimore, Philadelphia, and New York City. In 1836, when he was 27, he married his 13-year-old cousin, Virginia Clemm. She died of tuberculosis in 1847. In January 1845, he published his poem "The Raven" to instant success. He planned for years to produce his own journal The Penn, later renamed The Stylus. But before it began publishing, Poe died in Baltimore in 1849, aged 40, under mysterious circumstances. The cause of his death remains unknown, and has been variously attributed to many causes including disease, alcoholism, substance abuse, and suicide.	2024-06-01 13:43:47
2	B	Poe switched his focus to prose, and spent the next several years working for literary journals and periodicals, becoming known for his own style of literary criticism. His work forced him to move between several cities, including Baltimore, Philadelphia, and New York City. In 1836, when he was 27, he married his 13-year-old cousin, Virginia Clemm. She died of tuberculosis in 1847. In January 1845, he published his poem "The Raven" to instant success. He planned for years to produce his own journal The Penn, later renamed The Stylus. But before it began publishing, Poe died in Baltimore in 1849, aged 40, under mysterious circumstances. The cause of his death remains unknown, and has been variously attributed to many causes including disease, alcoholism, substance abuse, and suicide.	2024-06-02 14:43:47

Mejoras

Si deseamos una mejor customización del archivo **app.js** como se muestra en la siguiente imagen:



The screenshot shows a GitHub web interface for the repository `alcarazolabs/laravel-vue3-crud-composition-api-spanish`. The file `resources/js/app.js` is selected, showing its first commit. The code is as follows:

```
1  require('./bootstrap');
2
3  //require jquery
4  global.$ = global.jQuery = require('jquery');
5
6  import { createApp, defineComponent } from "vue";
7
8  import router from './router'
9  import CompaniesIndex from './components/companies/CompaniesIndex'
10 import CompaniesCreate from './components/companies/CompaniesCreate'
11 import CompaniesEdit from './components/companies/CompaniesEdit'
12
13 import LaravelVuePagination from "laravel-vue-pagination";
14
15 const root = defineComponent({
16   CompaniesIndex
17 })
18
19 const app = createApp(root)
20
21 app.use(router)
22
23 app.component('CompaniesIndex', CompaniesIndex);
24 app.component('CompaniesCreate', CompaniesCreate);
25 app.component('CompaniesEdit', CompaniesEdit);
26
27 app.component('Pagination', LaravelVuePagination);
28
29
30 app.mount('#app')
```

Revisar el código del siguiente repositorio:

<https://github.com/alcarazolabs/laravel-vue3-crud-composition-api-spanish/blob/main/resources/js/app.js>

Nota: Si siguen el código de este repositorio, no se olviden de instalar “vue-router” ya que en esta guía no se instaló solo para mostrar cómo se configura vue, pero si ustedes deciden implementar el setup de este repositorio les hará falta instalar “vue-router”:

\$ npm install vue-router

Fuentes:

- <https://bloginnovazione.it/en/vue-laravel/48405/>
- <https://github.com/bloginnovazione/laravel11-vue>
- <https://polinwei.com/laravel-11-with-sanctum-vue3/>