



# Varias Instancias

Vue permite llegar a tener todas las instancias que nos hagan falta para tener controlados distintos bloques del DOM. De hecho, para tener varias instancias es tan fácil como indicárselo en el JS.

```
1 <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
2
3 <div id="app1">
4   <h1>{{text}}</h1>
5 </div>
6
7 <div id="app2">
8   <h1>{{text}}</h1>
9 </div>

1 new Vue({
2   el: '#app1',
3   data: {
4     text: 'hola instancia 1'
5   },
6 });
7
8 new Vue({
9   el: '#app2',
10  data: {
11    text: 'hola instancia 2'
12  },
13 });
14
```

**hola instancia 1**

**hola instancia 2**

Para comunicar instancias las tengo que encapsular en variables y ya desde ellas podré acceder, como, por ejemplo:

```
const vm1 = new Vue({
  el: '#app1',
  data: {
    text: 'hola instancia 1'
  },
});

setTimeout(function () {
  vm1.text = ' he sido modificada desde el timeout'
}, 2000);

const vm2 = new Vue({
```

Ó

```

t> 1 const vm1 = new Vue({
    2   el: '#app1',
    3   data: {
    4     text: 'hola instancia 1'
    5   },
    6 });
    7
    8
    9
   10 const vm2 = new Vue({
   11   el: '#app2',
   12   data: {
   13     text: vm1.text
   14   },
   15 },
   16 });
   17

```

Si quiero hacer referencia a propiedades del DOM o un objeto en concreto puede utilizar ref.

```

HTML
1 <script
  src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
2
3
4 <div id="app2">
5   <h1 ref="tituloPrincipal">{{text}}</h1>
6   <button @click="showRefs">Mostrar</button>
7 </div>

```

```

JS
1 const vm1 = new Vue({
2   el: '#app2',
3   data: {
4     text: 'hola instancia 1'
5   },
6   methods: {
7     showRefs: function(){
8       console.log(this.$refs);
9     },
10  },
11 });
12
13
14

```

**ola instancia 1**

Mostrar

Y si despliego la consola

```

{tituloPrincipal: h1}
  tituloPrincipal: h1
  __proto__: Object
iframeConsoleRunner-...0b17617061ce0f.js:1

```

Ahí puedo encontrar la referencia en la etiqueta h1, si desplegase tendría visibles todas las propiedades de ese elemento y poder modificarla.

```
3- <div id="app">
4-   <h1 ref="tituloPrincipal">{{text}}</h1>
5-   <button @click="showRefs">Mostrar</button>
6- </div>
7
8
1- const vm1 = new Vue({
2   el: '#app',
3   data: {
4     text: 'hola',
5   },
6   methods: {
7     showRefs: function () {
8       console.log(this.$refs);
9     },
10  },
11 });
12
13
14 vm1.$refs.tituloPrincipal.style.backgroundColor = 'orange';
```

ola

Mostrar

# Vuecli y Webpack

Estas son herramientas que nos van a ayudar con nuestra fase de desarrollo. Lo primero será instalar Vuecli, para esto deberemos tener instalado Node JS (por ahora solo lo utilizaremos por su gestor de paquetes), cosa que podremos realizar desde su página web.

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

Una vez instalado iremos a la terminal e instalaremos vuecli, para ello ejecutaremos el comando desde la terminal: **sudo npm install -g vue-cli**

```
(base) iMac-de-JORGE:~ jorge$ npm install -g vue-cli
npm WARN deprecated vue-cli@2.9.6: This package has been deprecated in favour of @vue/cli
npm WARN deprecated coffee-script@1.12.7: CoffeeScript on NPM has moved to "coffeescript" (no hyphen)
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
i fetchMetadata: sill resolveWithNewModule yaml-js@0.0.8 checking installable status
```

Para comprobar su instalación teclearemos **vue**.

```
((base) iMac-de-JORGE:~ jorge$ vue
Usage: vue <command> [options]

Options:
  -V, --version  output the version number
  -h, --help     output usage information

Commands:
  init          generate a new project from a template
  list          list available official templates
  build         prototype a new project
  create        (for v3 warning only)
  help [cmd]    display help for [cmd]
(base) iMac-de-JORGE:~ jorge$
```

iiiiYa tenemos instalado vue-cli!!!!

Ahora ya podemos utilizar templates para montar el esqueleto de nuestra aplicación.

Vamos a crear un proyecto de nombre test-vue-cli y para ello tecleamos en la consola:

### vue init webpack-simple test-vue-cli

```
(base) iMac-de-JORGE:~ jorge$ vue init webpack-simple test-vue-cli

? Project name test-vue-cli
? Project description A Vue.js project
? Author jorloque <jorloque@gmail.com>
? License MIT
? Use sass? No

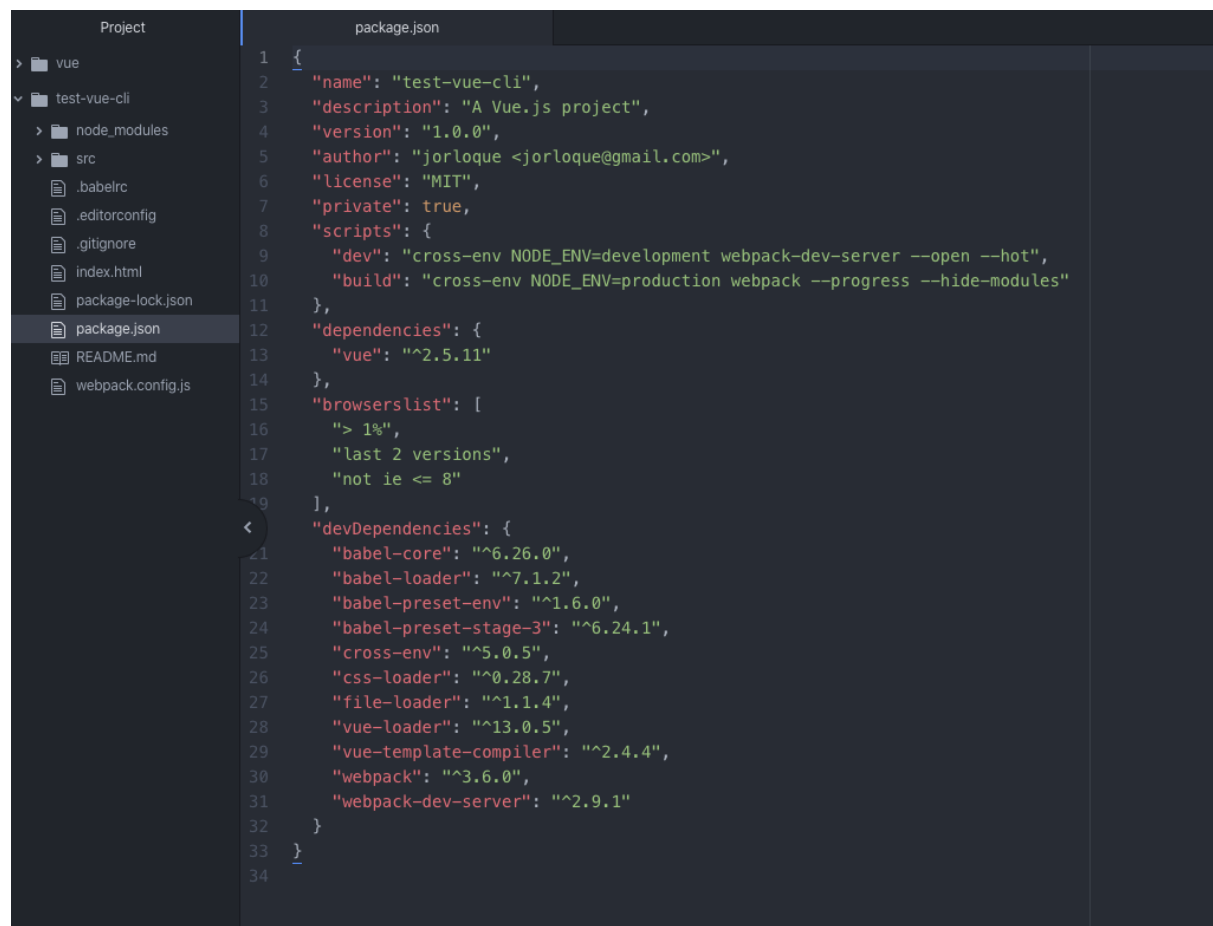
vue-cli · Generated "test-vue-cli".

To get started:

cd test-vue-cli
npm install
npm run dev
```

Y seguiremos los pasos que nos va indicando.

Si abrimos la carpeta con Atom podremos ver que ya ha ido creando su estructura de carpetas:



The screenshot shows the Atom editor interface. On the left, the 'Project' sidebar displays the file structure of the 'test-vue-cli' project, including folders like 'node\_modules' and 'src', and files like 'package.json', 'README.md', and 'webpack.config.js'. The main editor area shows the content of 'package.json', which is a JSON object containing project metadata and dependencies.

```
1 {
2   "name": "test-vue-cli",
3   "description": "A Vue.js project",
4   "version": "1.0.0",
5   "author": "jorloque <jorloque@gmail.com>",
6   "license": "MIT",
7   "private": true,
8   "scripts": {
9     "dev": "cross-env NODE_ENV=development webpack-dev-server --open --hot",
10    "build": "cross-env NODE_ENV=production webpack --progress --hide-modules"
11  },
12  "dependencies": {
13    "vue": "^2.5.11"
14  },
15  "browserslist": [
16    "> 1%",
17    "last 2 versions",
18    "not ie <= 8"
19  ],
20  "devDependencies": {
21    "babel-core": "^6.26.0",
22    "babel-loader": "^7.1.2",
23    "babel-preset-env": "^1.6.0",
24    "babel-preset-stage-3": "^6.24.1",
25    "cross-env": "^5.0.5",
26    "css-loader": "^0.28.7",
27    "file-loader": "^1.1.4",
28    "vue-loader": "^13.0.5",
29    "vue-template-compiler": "^2.4.4",
30    "webpack": "^3.6.0",
31    "webpack-dev-server": "^2.9.1"
32  }
33 }
```

Si ejecutamos run dev para levantar nuestro proyecto, entonces levantará la aplicación en el navegador: **localhost:8080**



## Welcome to Your Vue.js App

### Essential Links

[Core Docs](#) [Forum](#) [Community Chat](#) [Twitter](#)

### Ecosystem

[vue-router](#) [vuex](#) [vue-loader](#) [awesome-vue](#)

Para resaltar el código en Atom descarga la siguiente extensión:

<https://atom.io/packages/language-vue-component>

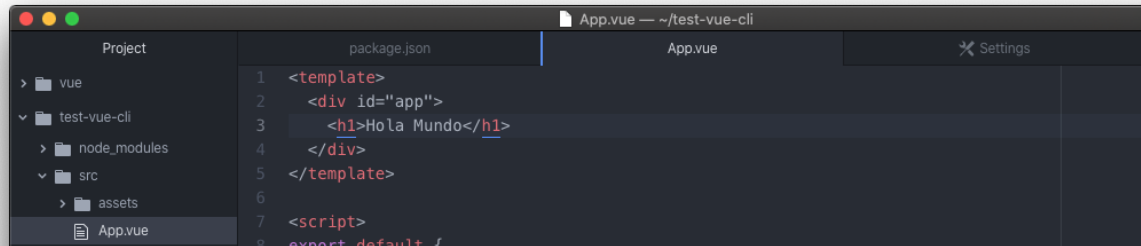
Por defecto en la plantilla tengo el siguiente código.

```
Project    package.json    App.vue    bk_App.vue    Settings
> vue
├─ test-vue-cli
│  └─ node_modules
│     └─ src
│        └─ assets
│           └─ App.vue
│              bk_App.vue
│              main.js
│              .babelrc
│              .editorconfig
│              .gitignore
│              index.html
│              package-lock.json
│              package.json
│              README.md
│              webpack.config.js
└─ package.json

1 <template>
2   <div id="app">
3     
4     <h1>{{ msg }}</h1>
5     <h2>Essential Links</h2>
6     <ul>
7       <li><a href="https://vuejs.org" target="_blank">Core Docs</a></li>
8       <li><a href="https://forum.vuejs.org" target="_blank">Forum</a></li>
9       <li><a href="https://chat.vuejs.org" target="_blank">Community Chat</a></li>
10      <li><a href="https://twitter.com/vuejs" target="_blank">Twitter</a></li>
11    </ul>
12    <h2>Ecosystem</h2>
13    <ul>
14      <li><a href="http://router.vuejs.org/" target="_blank">vue-router</a></li>
15      <li><a href="http://vuex.vuejs.org/" target="_blank">vuex</a></li>
16      <li><a href="http://vue-loader.vuejs.org/" target="_blank">vue-loader</a></li>
17      <li><a href="https://github.com/vuejs/awesome-vue" target="_blank">awesome-vue</a></li>
18    </ul>
19  </div>
20 </template>
21
22 <script>
23 export default {
24   name: 'app',
25   data () {
26     return {
27       msg: 'Welcome to Your Vue.js App'
28     }
29   }
}
```

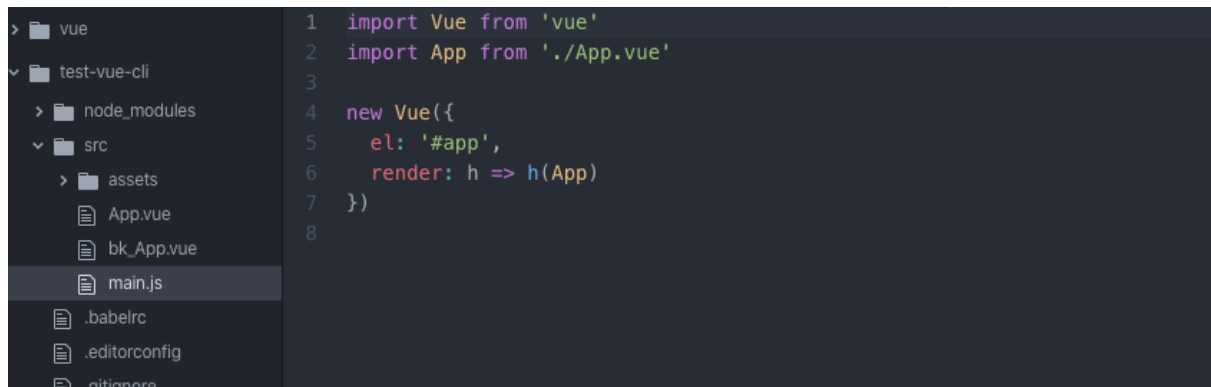
Vamos a poner el típico “Hola Mundo”.

# Hola Mundo



A la vez que modifico el código se actualiza automáticamente en la página.

Si miramos con detalle en el main.js, ahí aparece mi instancia de Vue.



La propiedad render le dice que coja todo el código que hay en los componentes en app y que lo renderice en la id que tienes designada en él.