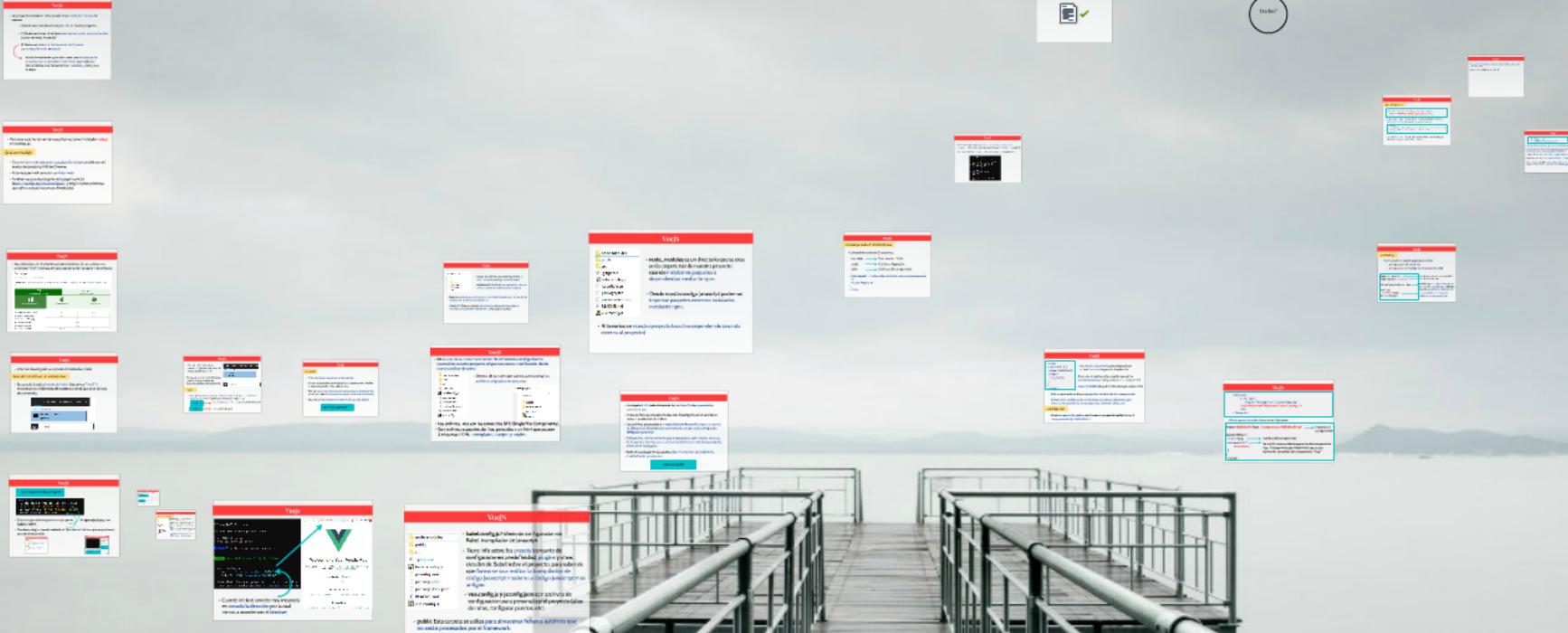
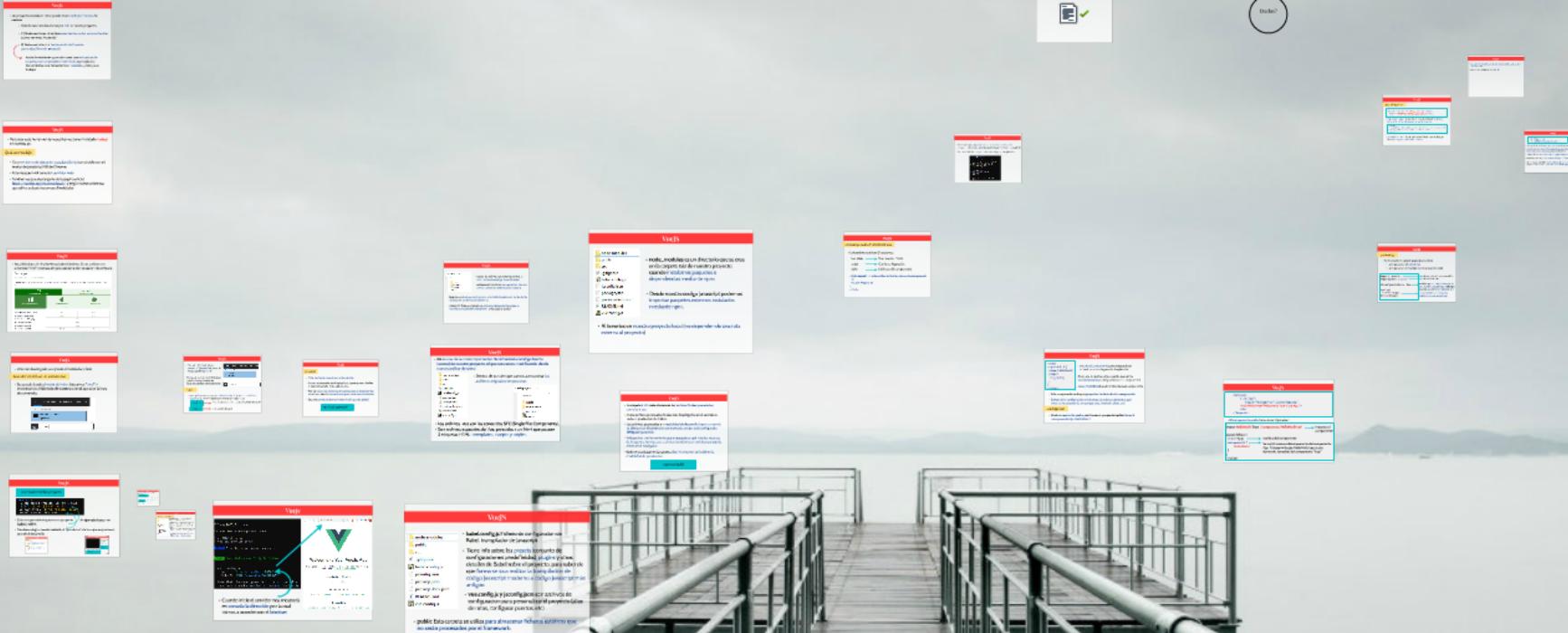


# ADM



Prezi

# ADM



# VueJS

- Un proyecto creado con Vue puede tener **múltiples formas de crearse.**
  - 1) Podemos vincular el core por **cdn** a nuestro proyecto.
  - 2 ) Podemos tener el archivo **core** dentro de los recursos locales (como venimos haciendo)
  - 3) Podemos usar una **herramienta de línea de comandos llamada @vue/cli**

Actúa de asistente y permite crear una **estructura de carpetas con un proyecto Vue inicial**, con todas las herramientas que necesitemos **instaladas** y listo para trabajar

# VueJS

- Para usar esta herramienta necesitamos tener instalado **nodejs** en nuestra pc.

## Qué es nodejs

- Es un **entorno de ejecución para JavaScript** construido con el motor de JavaScript V8 de Chrome.
- Esto va a permitir tener un **servidor web**.
- Tendremos que descargarlo de la página oficial <https://nodejs.org/es/download/>, y según nuestro sistema operativo, seleccionaremos el instalador.

# VueJS

- Para Windows, click sobre Instalador Windows. Es un archivo con extensión "msi" mostrará el típico asistente de instalación de software.

## Descargas

Versión actual: 14.16.1 (includes npm 6.14.12)

Descargue el código fuente de Node.js o un instalador pre-compilado para su plataforma, y comience a desarrollar hoy.

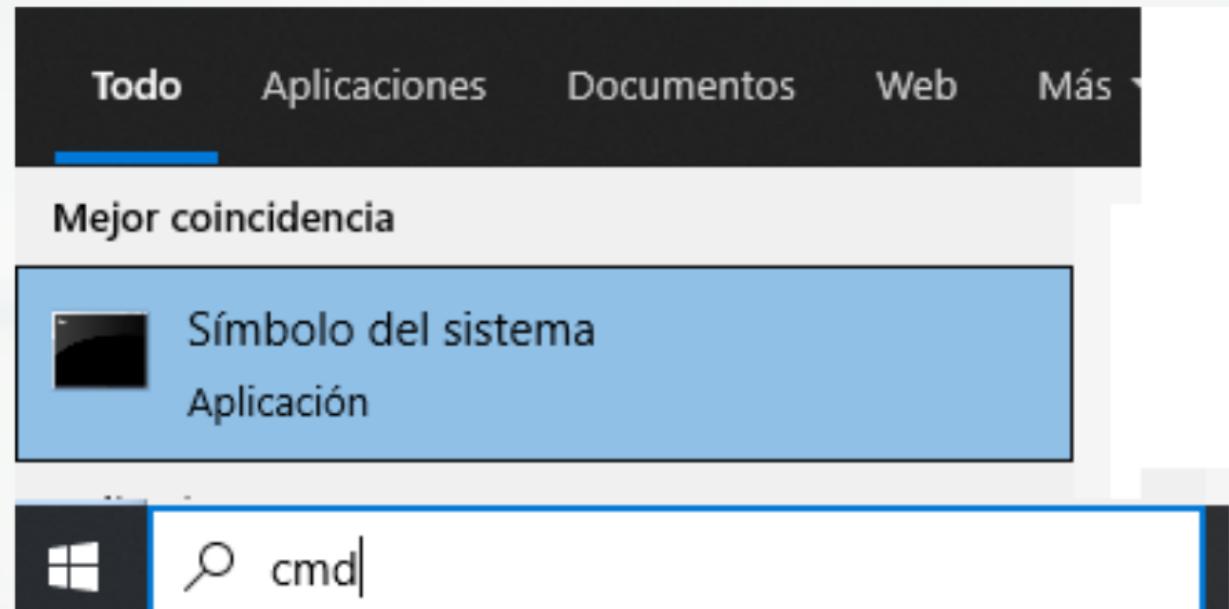
LTS	Actual
Recomendado para la mayoría	Últimas características
 Instalador Windows node-v14.16.1-x64.msi	 Instalador macOS node-v14.16.1.pkg
	 Código Fuente node-v14.16.1.tar.gz
Instalador Windows (.msi)	32-bit      64-bit
Binario Windows (.zip)	32-bit      64-bit
Instalador macOS (.pkg)	64-bit
Binario macOS (.tar.gz)	64-bit
Binario Linux (x64)	64-bit
Binario Linux (ARM)	ARMv7      ARMv8

# VueJS

- Una vez descargado, se ejecuta el instalador y listo

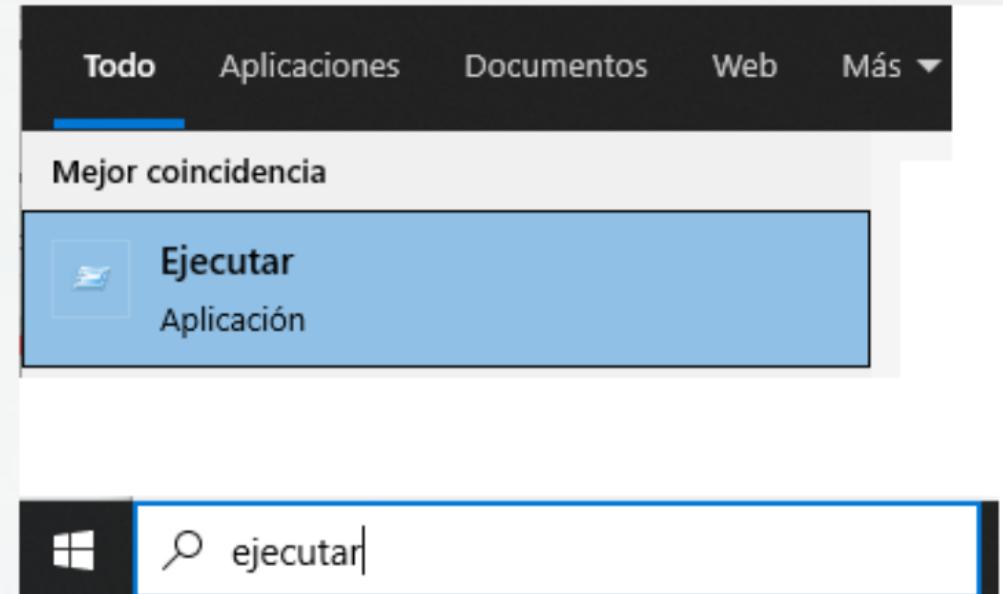
## Acceder a la línea de comandos

- Se accede desde el menú de inicio, buscamos "cmd" y encontramos el **Símbolo de sistema** o cmd, que abre la línea de comandos.



# VueJS

- O desde el menú de inicio, buscamos "Ejecutar" del menú de inicio y escribimos "cmd".
- El atajo de teclado, **tecla Windows + tecla r**, lanza el cuadro de Ejecutar, escriben cmd y dan enter



npm

- node package manager, es un **administrador de paquetes**, o módulos **javascript**, ya viene instalado por defecto al instalar node.

node -v



En consola nos permite saber la **versión de node**

npm -v



y la **versión de npm**

# VueJS

## @vue/cli

- Es la **interfaz de línea de comandos de Vue**.
- Herramienta creada por el equipo Vue.js para ayudar a facilitar el rápido desarrollo de las aplicaciones.
- Permite **crear una estructura de carpetas con un proyecto Vue inicial**, con todas las **herramientas que necesitemos instaladas**
- Desde la **consola debemos instalar la cli (-g, es de global)**

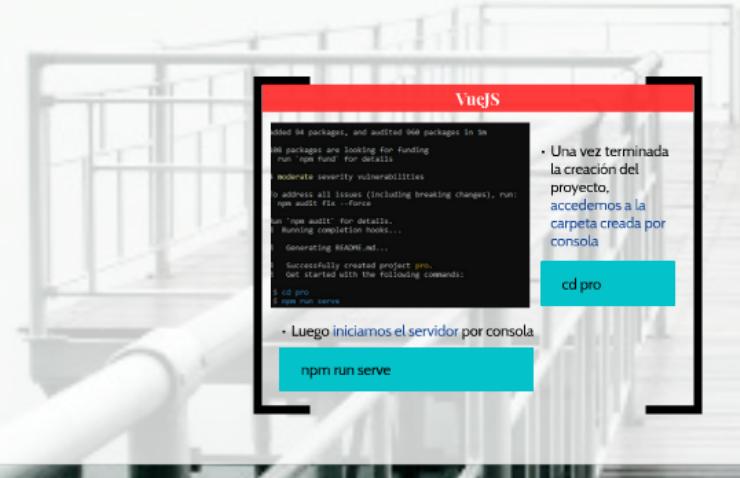
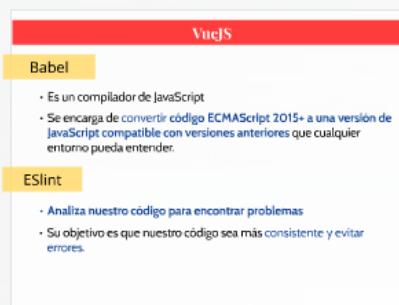
```
npm install -g @vue/cli
```

# VueJS

vue create nombre-proyecto

```
Vue CLI v4.5.13
? Please pick a preset: (Use arrow keys)
> Default ([Vue 2] babel, eslint)
Default (Vue 3) ([Vue 3] babel, eslint)
Manually select features
```

- Esto nos permite empezar con proyecto **base, con babel y eslint**
- También elegir a **modo manual**, el tipo de instalación que requerimos, acorde al desarrollo.



## Babel

- Es un compilador de JavaScript
- Se encarga de **convertir código ECMAScript 2015+ a una versión de JavaScript compatible con versiones anteriores** que cualquier entorno pueda entender.

## ESlint

- **Analiza nuestro código para encontrar problemas**
- Su objetivo es que nuestro código sea más **consistente y evitar errores**.

# VueJS

```
added 94 packages, and audited 960 packages in 1m  
108 packages are looking for funding  
  run `npm fund` for details  
+ moderate severity vulnerabilities  
To address all issues (including breaking changes), run:  
  npm audit fix --force  
  
Run `npm audit` for details.  
[?] Running completion hooks...  
[?] Generating README.md...  
[?] Successfully created project pro.  
[?] Get started with the following commands:  
  
$ cd pro  
$ npm run serve
```

- Una vez terminada la creación del proyecto, accedemos a la carpeta creada por consola

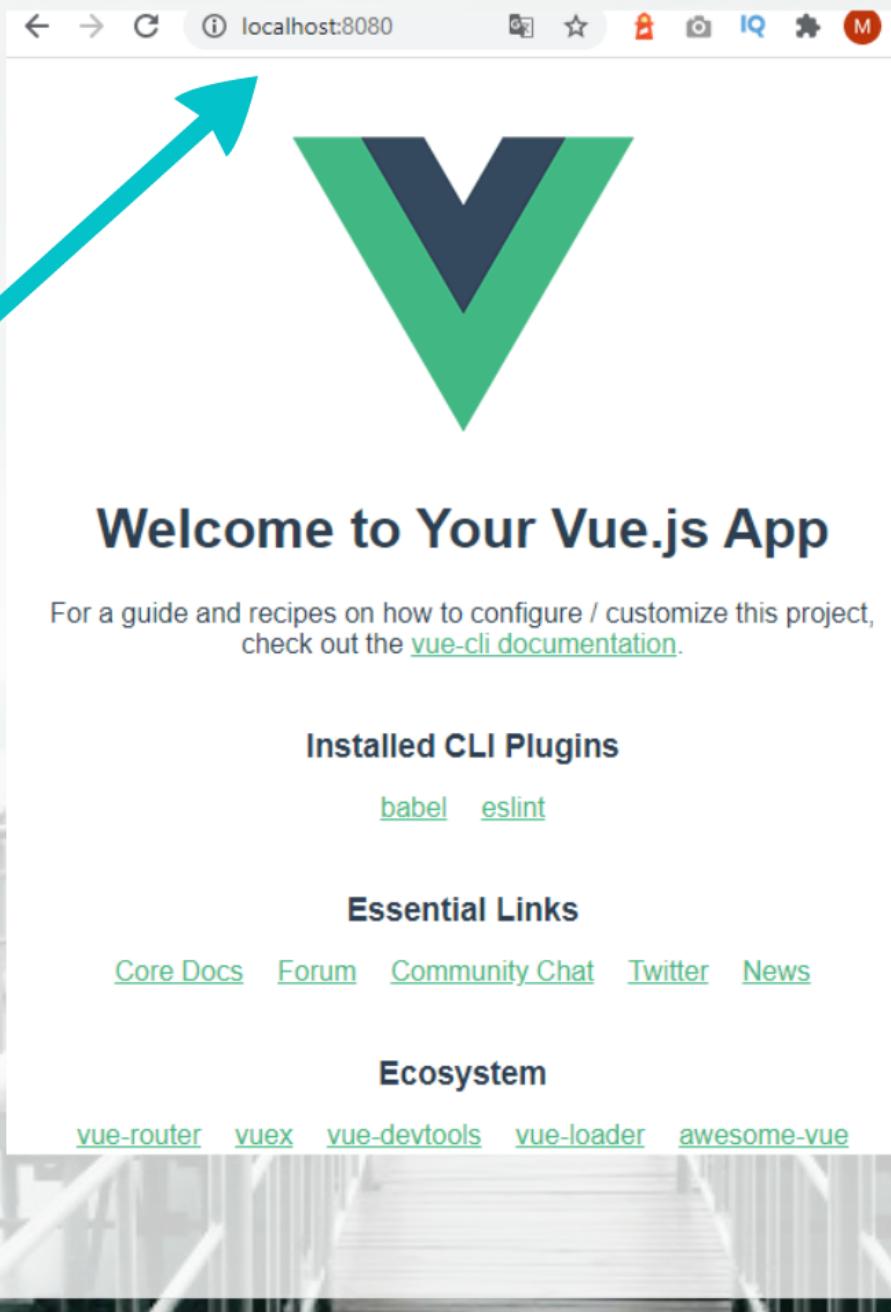
cd pro

- Luego iniciamos el servidor por consola

npm run serve

# Vuejs

```
C:\adm_2023_2>cd pro  
C:\adm_2023_2\pro>npm run serve  
> pro@0.1.0 serve  
> vue-cli-service serve  
  
INFO Starting development server...  
  
DONE Compiled successfully in 5938ms  
  
App running at:  
- Local: http://localhost:8080/  
- Network: http://192.168.0.98:8080/  
  
Note that the development build is not optimized.  
To create a production build, run npm run build.
```



A screenshot of a web browser window displaying the "Welcome to Your Vue.js App" page. The address bar shows "localhost:8080". The page features the Vue.js logo (a green triangle with a white 'V') and the text "Welcome to Your Vue.js App". Below it, there's a note about the development build being unoptimized and a link to the documentation. On the right, sections for "Installed CLI Plugins" (babel, eslint), "Essential Links" (Core Docs, Forum, Community Chat, Twitter, News), and "Ecosystem" (vue-router, vuex, vue-devtools, vue-loader, awesome-vue) are visible. A large blue arrow points from the "App running at:" text in the terminal to the "localhost:8080" address in the browser's address bar.

localhost:8080

Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project, check out the [vue-cli documentation](#).

Installed CLI Plugins

[babel](#) [eslint](#)

Essential Links

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

Ecosystem

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

- Cuando inicie el servidor nos mostrará en **consola la dirección** por la cual vamos a acceder con el **browser**

# VueJS

- Si queremos **cambiar el puerto por defecto :8080**, en la consola debemos poner:

```
npm run serve -- --port 3000
```

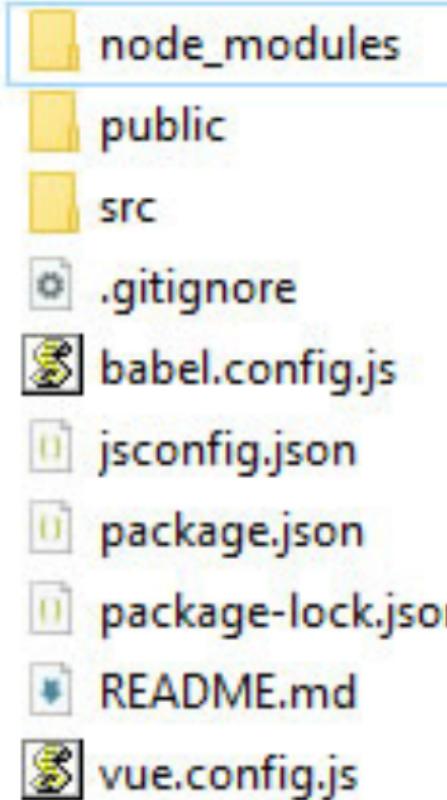
- Para **detener el servidor**, desde la consola presionamos la tecla **ctrl + c**
- Para saber los **comandos soportados por npm**, desde la consola:

```
npm help
```

- Para mayor detalle la documentacion oficial:

<https://docs.npmjs.com/>

## Estructura del proyecto



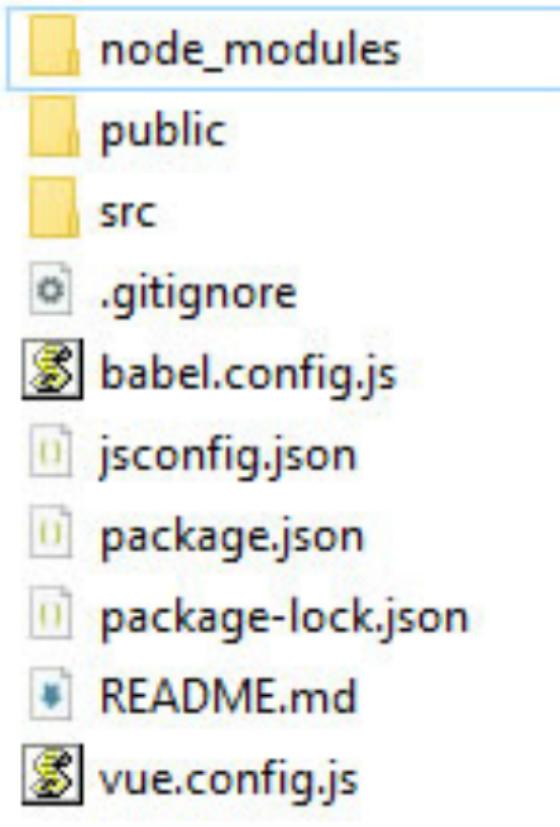
**README.md** es un fichero que sirve para dar información sobre el repositorio.

**.gitignore** es un archivo donde podemos escribir, línea por línea, los nombres de archivos (o patrones o comodines) que queremos que Git, ignore

Es obligatorio utilizar en nuestros proyectos, como mínimo para ignorar la carpeta node\_modules.

- **package.json y package-lock.json son los que utiliza NPM para crear y gestionar una aplicación o proyecto y sus dependencias.**

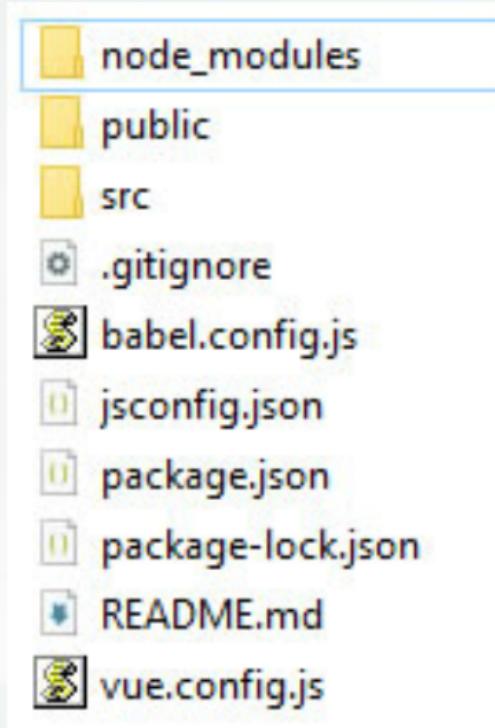
# VueJS



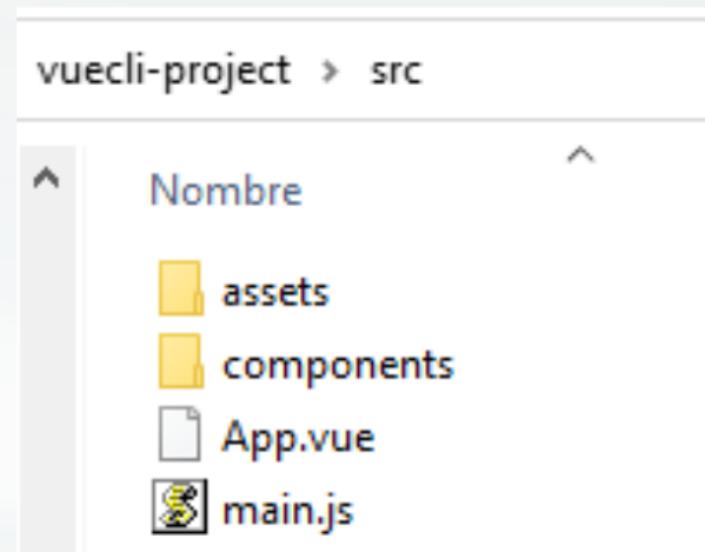
- **babel.config.js** Fichero de configuración de Babel, transpilador de Javascript.
- Tiene info sobre los **presets** (conjunto de configuraciones predefinidas), **plugins** y otros detalles de Babel sobre el proyecto, para saber de que forma se va a realizar la transpilación de código Javascript moderno a código Javascript más antiguo.
- **vue.config.js** y **jsconfig.json** son archivos de configuración para personalizar el proyecto (alias de rutas, configurar puertos, etc)
- **public** Esta carpeta se utiliza **para almacenar ficheros estáticos que no serán procesados por el framework**:

# VueJS

- **src** es una de las más importantes. **Se almacena el código fuente (source) de nuestro proyecto, el que estaremos modificando desde nuestro editor de texto**

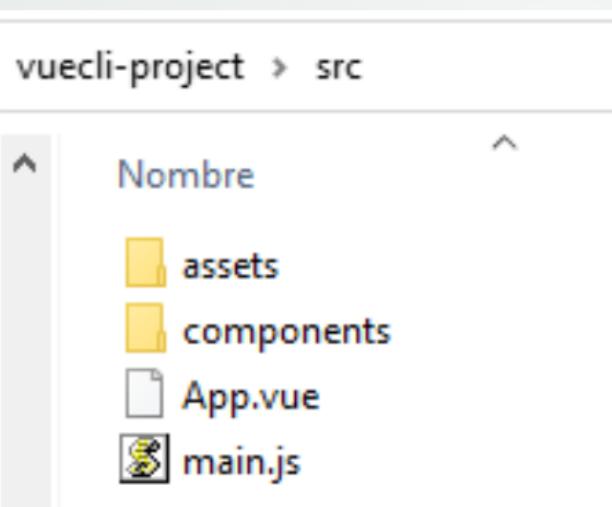


- Dentro de **src** siempre vamos a encontrar **los archivos originales sin procesar**:



- **Los archivos .vue son los conocidos SFC (Single File Components),**
- Son archivos especiales de Vue, parecidos a un .html que poseen 3 etiquetas HTML: **<template>**, **<script>** y **<style>**.

# VueJS



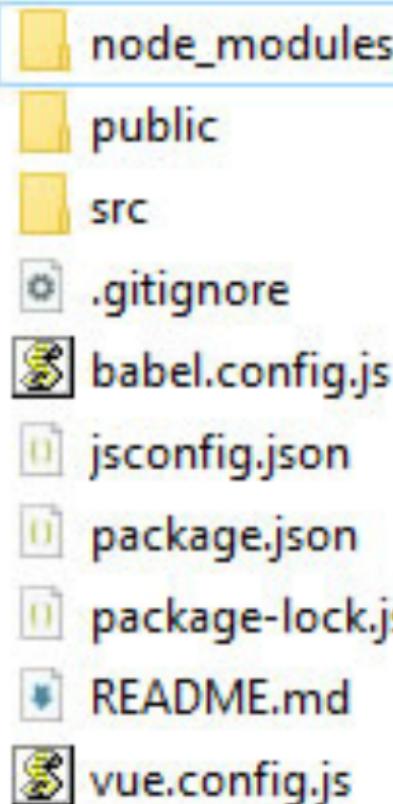
- **assets** los archivos que tenemos dentro **se usan en nuestro código importándolos**,
- **components** tendrá los **componentes .vue** que vamos creando durante nuestro proyecto.
- **App.vue** es el **componente general e inicial de la aplicación**, desde donde se cargarán los demás componentes.
- **main.js** Es fichero principal que **arranca el proyecto Vue y que se insertará en la plantilla index.html** ( en la carpeta public/)

# VueJS

- La carpeta `dist/` suele almacenar los **archivos finales procesados, como .js y .css,**
- Estos archivos procesados finales irán desplegados en el servidor o web en producción definitiva.
- Los archivos procesados en **modalidad de desarrollo** (`npm run serve`) **se almacenan directamente en memoria, ya que está configurado Webpack para esto.**
- Webpack es una herramienta que empaqueta y optimiza los recursos (js, imágenes, fuentes, css, etc) y los transforma en archivos que pueda entender el navegador.
- Sólo se guardarán en la carpeta `dist/` si creamos un build en la **modalidad de producción**

`npm run build`

# VueJS



- **node\_modules** es un directorio que se crea en la carpeta raíz de nuestro proyecto cuando instalamos paquetes o dependencias mediante npm.
- Desde nuestro código Javascript podemos importar paquetes externos instalados mediante npm,
- Al tenerlos en **nuestro proyecto local** (no dependen de una ruta externa al proyecto)



# VueJS

- Para saber que dependencias tenemos instaladas en un proyecto utilizar el comando **npm list** y el parámetro **--depth 0**
- Mostrará solo los paquetes directos del proyecto principal:

```
C:\>cd vuecli-project

C:\vuecli-project>npm list --depth 0
vuecli-project@0.1.0 C:\vuecli-project
+-- @vue/cli-plugin-babel@4.5.13
+-- @vue/cli-plugin-eslint@4.5.13
+-- @vue/cli-service@4.5.13
+-- babel-eslint@10.1.0
+-- bindings@1.5.0 extraneous
+-- core-js@3.12.1
+-- eslint-plugin-vue@6.2.2
+-- eslint@6.8.0
+-- file-uri-to-path@1.0.0 extraneous
+-- nan@2.14.2 extraneous
+-- vue-template-compiler@2.6.12
`-- vue@2.6.12
```

# VueJS

## src/components/HelloWorld.vue

- Este archivo contiene 3 secciones:

template → Con nuestro HTML

script → Con la configuración

style → Estilo css del componente

<style scoped> --> estos estilos se limitan solo a este componente

```
h3 {  
  margin: 40px 0 0;  
}  
</style>
```

# VueJS

```
<script>
export default {
  name: 'HelloWorld',
  props: {
    msg: String
  }
}
</script>
```

- exporta el componente para que pueda ser utilizado en otros lugares de la aplicación  
(Es igual a lo que hacíamos cuando usabamos **Vue.component** y configurábamos el componente)
  - **name: HelloWorld** es el nombre de este componente
- 
- Este componente recibe por **props una variable de otro componente**
  - *Dentro de la configuración, tendremos los mismos elementos que vimos en la creación de un componente (methods, data, etc)*

src/App.vue

- Es el **componente padre**, dentro de su propio template tiene al componente hijo **HelloWorld**

# VueJS

```
<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>
```

- El componente padre tiene su configuración:

*import HelloWorld from './components/HelloWorld.vue'*

→ Importar el componente

```
export default {
  name: 'App',
  components: {
    HelloWorld
  }
}
</script>
```

→ nombre del componente

→ Se registra como subcomponente del componente App. Esto permite que HelloWorld sea usado dentro del template del componente "App"

# VueJS

## src/main.js

- Es el punto de entrada de Javascript y contiene:
  - La importación del **core de vue**
  - La importacion de **App** (que es un componente .vue)

```
import Vue from 'vue'  
import App from './App.vue'
```

desde la carpeta node\_modules  
desde el archivo

```
Vue.config.productionTip = false
```

variable que **desactiva el mensaje de consola del navegador** que indica que estamos usando **Vue en producción**.

```
new Vue({  
  render: h => h(App),  
}).$mount('#app')
```

**funcion render, crea elemento y devuelve el elemento (App)**

# VueJS

## public/index.html

```
<link rel="icon" href="<%= BASE_URL %>/favicon.ico">
<title><%= htmlWebpackPlugin.options.title %></title>
```

- Son **etiquetas especiales** que Vue (a través de Webpack) **modificará** para reemplazar por valores que conoce del proyecto.

```
<noscript>
  <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly
without JavaScript enabled. Please enable it to continue.</strong>
</noscript>
```

- **<noscript>** --> tiene texto que aparecerá si el usuario que ve la página tiene un **navegador sin Javascript activado**.

# VueJS

```
<div id="app"></div>
  <!-- built files will be auto injected -->
```

- Va a contener toda la aplicación web realizada con Vue.
- *Fuera de esta etiqueta podemos colocar lo que queramos en este fichero .html, pero no será manipulado mediante Vue*
- Luego del comentario se insertarán algunas etiquetas HTML <script>.
- Estas etiquetas son Javascript procesadas por el framework,
- Se encarga de determinar cuál es el Javascript relevante para la página y crear un archivo, generalmente llamado app.js.

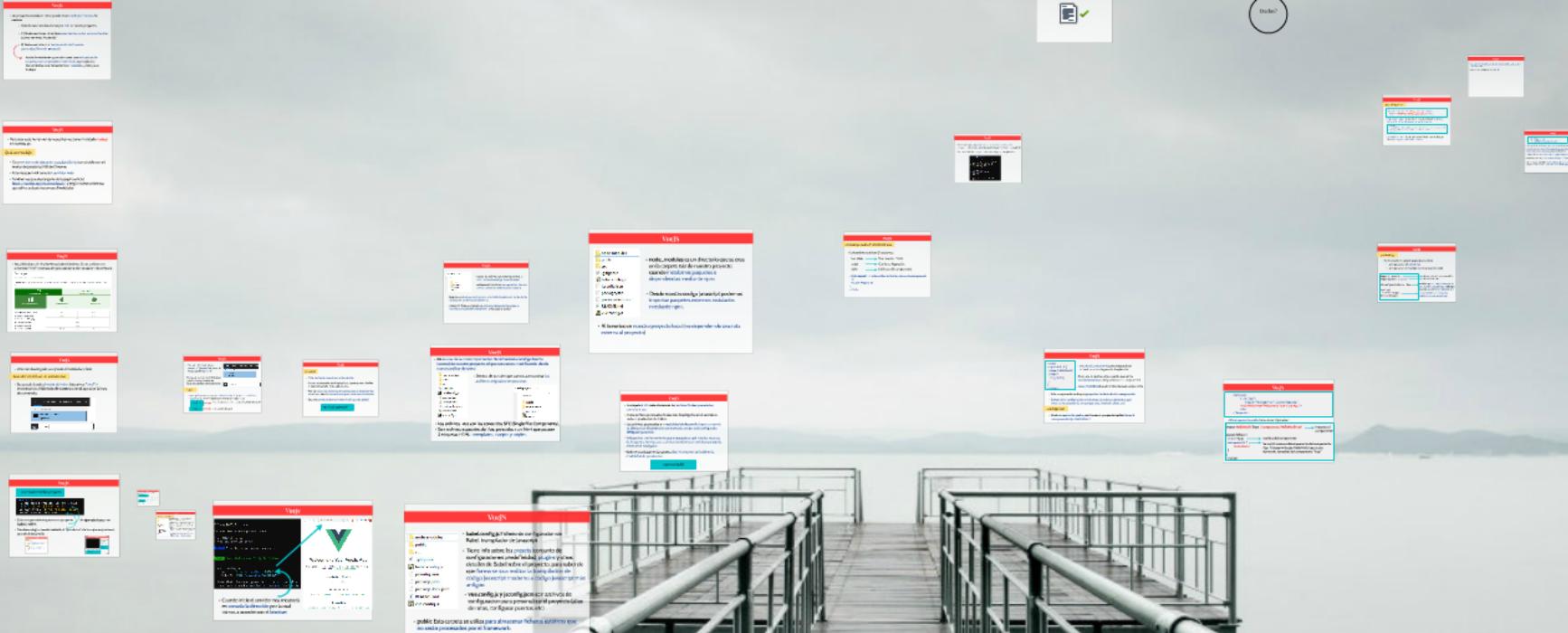
# VueJS

- Este archivo contendrá todo el código Javascript de la aplicación en un sólo archivo.
- Este proceso se llama crear un **bundle**

# Dudas?



# ADM



Prezi