



# Introducción a Componentes Web y Vue Js

Instalación y características principales de Vue JS

***Describir los aspectos fundamentales de un framework orientado a componentes para el desarrollo de una aplicación Front-End.***

- Unidad 1: Introducción a Componentes Web y Vue Js
- Unidad 2: Binding de formularios
- Unidad 3: Templates y rendering en Vue
- Unidad 4: Manejo de eventos y reutilización de componentes
- Unidad 5: Consumo de datos desde una API REST



Te encuentras aquí



## ¿Qué aprenderás en esta sesión?

- *Reconoce las características y beneficios de la utilización de componentes en una aplicación Front-End.*

¿Qué ventaja tiene el uso  
de componentes?



¿Cómo crearías un  
componente web con  
JavaScript puro?



# **/\* Formas de instalar Vue en una aplicación web \*/**

# CDN

## Instalación

Es posible agregar Vue Js a través de su CDN:



The screenshot shows the Vue.js official website. The left sidebar contains a navigation menu with the following items: 'Patrocinador Especial' (with Standard Library logo), 'Guía' (with a dropdown menu showing '2.x'), 'Conocimientos Esenciales', 'Instalación', 'Introducción', '¿Qué es Vue.js?', 'Empezando' (highlighted in green), 'Renderización Declarativa', 'Condicionales y Bucles', 'Manejando Interacciones del Usuario', 'Composición con Componentes', and 'Relación con Elementos Personalizados'. The main content area is titled 'Empezando' and contains a warning icon and text: 'La guía oficial asume conocimientos de nivel intermedio de HTML, CSS y JavaScript. Si usted es totalmente nuevo en el desarrollo frontend, puede que no sea la mejor idea saltar directamente a un *framework* como su primer paso - ¡capte lo básico y luego vuelva! La experiencia previa con otros *frameworks* ayuda, pero no es necesaria.' Below this, it says: 'La forma más fácil de probar Vue.js es usando el ejemplo [Hola Mundo en JSFiddle](#). Siéntase libre de abrirlo en otra pestaña y seguirlo mientras analizamos algunos ejemplos básicos. O puede [crear un archivo index.html](#) e incluir Vue con:' followed by a code block showing the CDN script tag: 

```
<!-- versión de desarrollo, incluye advertencias de ayuda en la consola -->
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

De esta manera podemos sumar Vue a una página web sin necesidad de instalar todo su ambiente.

No obstante, para aprovechar al máximo el framework crearemos una aplicación a través del Vue CLI.

# NPM, Vue CLI

La manera más completa de trabajar con Vue es hacerlo a través de su propio **Command Line Interface (CLI)**.

Para ocuparlo, primero hay que instalarlo, por lo que debemos abrir una nueva terminal y ejecutar la siguiente línea de comando:

```
npm install -g @vue/cli  
  
# O También puedes hacerlo con yarn  
  
yarn global add @vue/cli
```



# Vue CLI

Lo siguiente será crear una nueva carpeta en donde crearemos una nueva aplicación.

Una vez ubicado en la nueva carpeta debemos dirigirnos a ella desde la terminal ejecutar el siguiente comando:

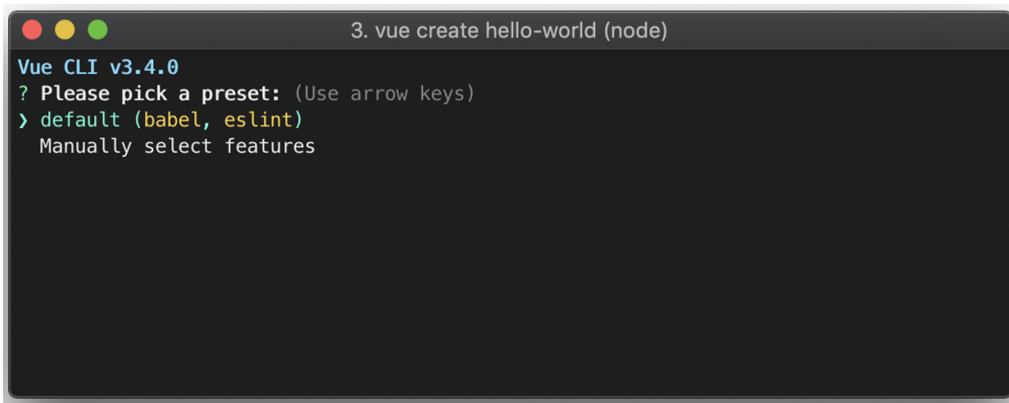
```
vue create hello-world
```

*hello-world* corresponderá al nombre de la carpeta y de la aplicación que se creará.

# Vue CLI

Al ejecutar el comando anterior, aparecerá este menú en donde podemos seleccionar la configuración por defecto o seleccionar las características manualmente.

Seleccionemos la opción **default**:

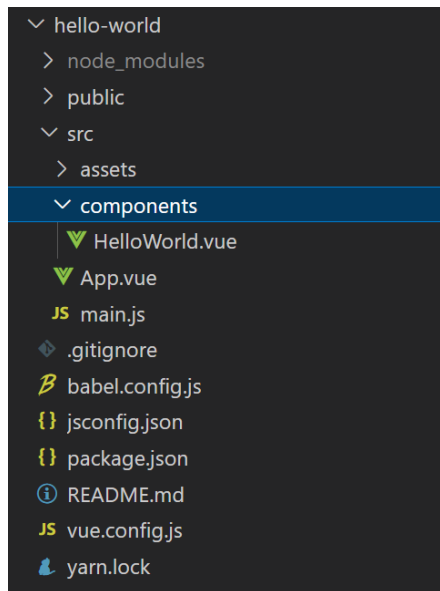


```
3. vue create hello-world (node)
Vue CLI v3.4.0
? Please pick a preset: (Use arrow keys)
> default (babel, eslint)
  Manually select features
```

Para cambiar de opción utiliza las flechas de tu teclado **↑↓** y presiona Enter para seleccionar

# Vue CLI

Una vez terminada la creación de la aplicación en la configuración **default**, veremos esta estructura de carpetas:



# Vue CLI

Entre los archivos de nuestra aplicación se incluye el **package.json**.

Este archivo es muy importante ya que representa el índice de las tecnologías que estamos ocupando en el proyecto.

No será necesario manipularlo ya que automáticamente se actualizará al momento de instalar varios plugins que conoceremos más adelante.

```
{
  "name": "hello-world",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint"
  },
  "dependencies": {
    "core-js": "^3.8.3",
    "vue": "^2.6.14"
  },
  "devDependencies": {
    "@babel/core": "^7.12.16",
    "@babel/eslint-parser": "^7.12.16",
    "@vue/cli-plugin-babel": "~5.0.0",
    "@vue/cli-plugin-eslint": "~5.0.0",
    "@vue/cli-service": "~5.0.0",
    "eslint": "^7.32.0",
    "eslint-plugin-vue": "^8.0.3",
    "vue-template-compiler": "^2.6.14"
  },
  "eslintConfig": {
    "root": true,
    "env": {
      "node": true
    },
    "extends": [
      "plugin:vue/essential",
      "eslint:recommended"
    ],
    "parserOptions": {
      "parser": "@babel/eslint-parser"
    },
    "rules": {}
  },
  "browserslist": [
    "> 1%",
    "last 2 versions",
    "not dead"
  ]
}
```

# Vue CLI

Ahora solo queda levantar la aplicación. Para esto debes ubicarte desde la terminal en la carpeta **hello-world** y ejecutar el siguiente comando:

```
npm run serve
```

Esto levantará un servidor local en el puerto :8080 y abrirá en tu navegador por defecto la aplicación.

La aplicación en el navegador se verá de esta manera:

¡Ahora ya estamos listos para empezar a crear aplicaciones con Vue Js!

**{desafío}**  
**latam\_**



## 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](#)

Creemos una aplicación con  
nombre el  
**vue-js-unidad-1**



**/\* Herramientas de desarrollo \*/**

# Herramientas de desarrollo

## *Webpack*

- Es un empaquetador de módulos.
- Su función principal es agrupar archivos y entregarnos una versión final lista para ser renderizada por los navegadores.
- A esta versión empaquetada se le conoce como la versión de producción.



Cuando trabajamos con aplicaciones creadas por el **Vue CLI**, Webpack ya está incluido y configurado, por lo que no tendremos que preocuparnos de manipular esta herramienta de desarrollo.



# Herramientas de desarrollo

## *Vue.js Devtools*

- Es una extensión para los navegadores que nos ayudará a conocer en tiempo real los valores de nuestras variables.
- Es muy útil para desarrollar con más claridad gracias a que sabremos en todo momento la información que manejan nuestros componentes.
- Se puede comparar este complemento con el inspector de elementos de los navegadores pero dedicado específicamente al desarrollo en Vue Js.



Vue.js devtools

 [vuejs.org](https://vuejs.org)

★★★★★ 1.827 ⓘ | [Herramientas para desarrolladores](#) | 1.000.000+ usuarios

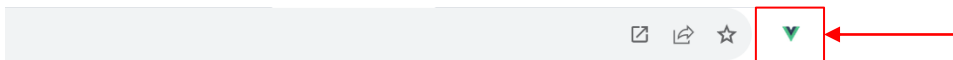
# Visualicemos los componentes que tenemos en el Vue Devtools



# Vue Devtools

Instala la extensión Vue Devtools y luego levanta la aplicación **vue-js-unidad-1** que creamos anteriormente.

Una vez en el navegador deberás notar que este icono se “enciende”, es decir se pinta del color verde de Vue.



Welcome to Your Vue.js App

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

Installed CLI Plugins

babel-eslint

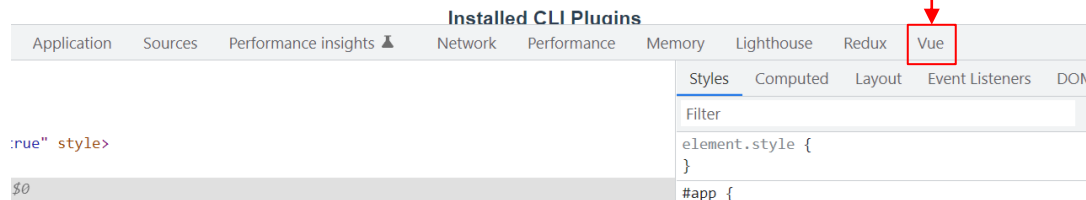
# Vue Devtools

Para abrir el DevTools abre la consola del navegador y encontrarás la pestaña **vue**



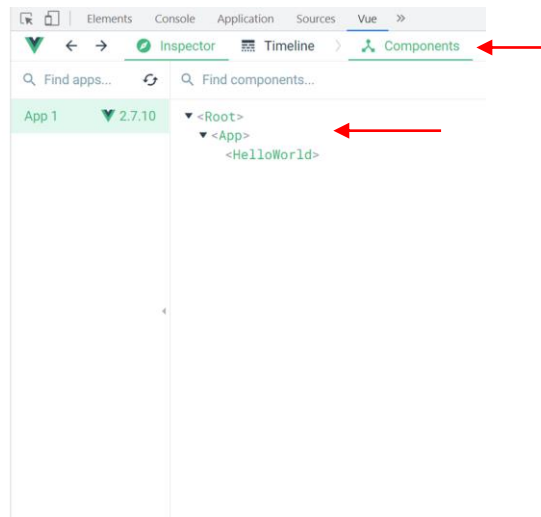
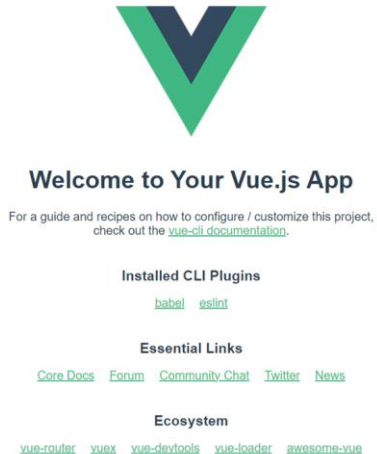
## Welcome to Your Vue.js App

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



# Vue Devtools

Al presionar en esa pestaña veremos el árbol de componentes de nuestra aplicación.



**/\* Morfología de un componente \*/**

# Morfología de un componente

Los componentes en Vue tienen extensión .vue y están basados en el formato Single File Component (SFC).

Esto quiere decir que en el mismo archivo tenemos código el **HTML**, **CSS** y **Js** correspondiente al componente.

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

```
<script>
import HelloWorld from './components/HelloWorld.vue'

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

```
<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

# Morfología de un componente

En el caso del código **HTML** es necesario declarar una etiqueta **template** que contenga un **único** envoltorio.

Típicamente optamos por un **div**, pero éste puede ser cualquier etiqueta.

```
<template> ←  
  <div id="app"> ←  
    <!-- HTML del componente -->  
  </div>  
</template>
```



# Modifiquemos el componente App



# El componente App

El componente **App** es el componente principal de la aplicación.

Modifica el código para que quede de la siguiente manera

```
<template>
  <div id="app">
    <h1>🌟 The best Framework of JavaScript 🌟 </h1>
  </div>
</template>

<script>

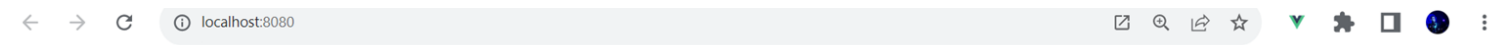
export default {
  name: 'App',
}
</script>

<style>
#app {
  text-align: center;
  margin-top: 60px;
}
</style>
```



# El componente App

Con solo guardar el archivo podrás ver que el navegador se actualiza automáticamente.



**The best Framework of JavaScript**



Esto es gracias a que el entorno que creamos con Vue CLI incluye el hot reloading.

¿Qué casos de uso  
imaginas que puedas  
darle al generador de  
templates de Vue?





## Próxima sesión...

- *Describe el rol de un framework orientado a componentes reconociendo sus beneficios para el desarrollo de una aplicación Front-End.*
- *Describe las características principales del framework Vue.js para el desarrollo de aplicaciones Front-End.*

**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

