



# Vue.js

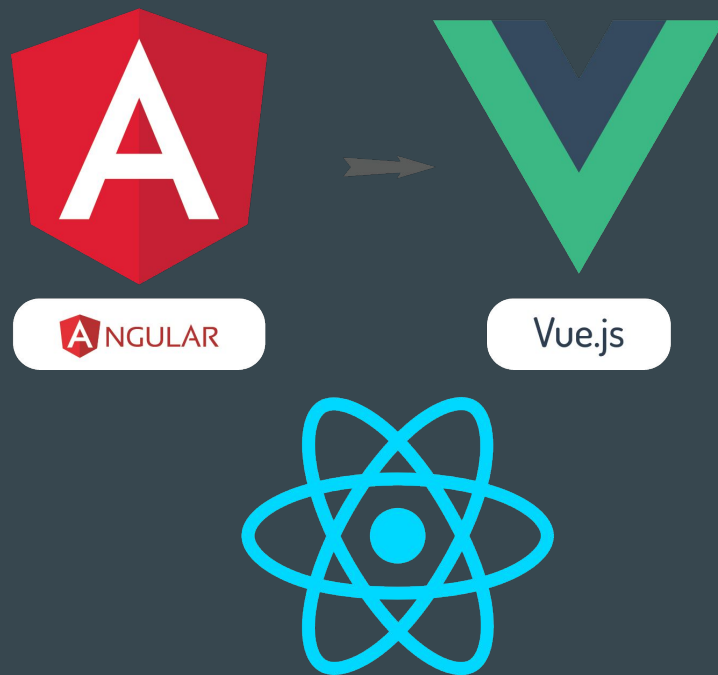
Silvia Rull

# Index

- Que és Vue.js?
- Com funciona?
- Components
- Directives
- Comunicació entre components
- Avantatges/Inconvenients

# Que és Vue.js?

- Framework de Javascript progressiu
- Interfícies d'usuari
- Codi obert
- Aplicacions d'una sola pàgina
- Inspirat en Angular



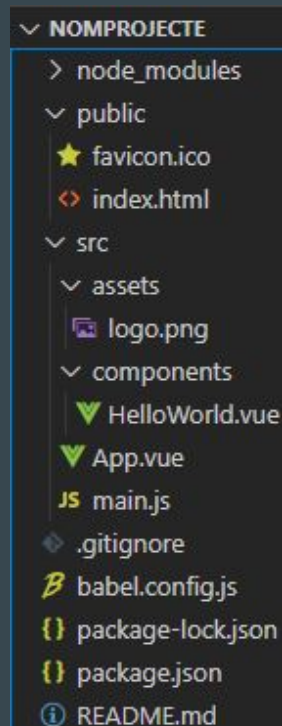
# Instal·lació

1. `npm install -g @vue/cli`
2. `vue --version` → Comprovem que s'hagi instal·lat bé i quina versió tenim

-Crear un projecte:

- a. `vue create nomprojecte`

```
Vue CLI v4.5.9
? Please pick a preset: (Use arrow keys)
> Default ([Vue 2] babel, eslint)
  Default (Vue 3 Preview) ([Vue 3] babel, eslint)
  Manually select features
```



▼ NOMPROJECTE

- > node\_modules
- ▼ public
  - ★ favicon.ico
  - <> index.html
- ▼ src
  - ▼ assets
    - 🖼 logo.png
  - ▼ components
    - ▼ HelloWorld.vue
    - ▼ App.vue
  - JS main.js
  - 🔒 .gitignore
  - 📄 babel.config.js
  - {} package-lock.json
  - {} package.json
  - 📖 README.md

# Instal·lació

-Iniciar el projecte:

- `npm install`
- `npm run serve`

```
DONE Compiled successfully in 2830ms
```

```
App running at:
```

```
- Local:   http://localhost:8080/  
- Network: http://192.168.86.22:8080/
```

```
Note that the development build is not optimized.  
To create a production build, run npm run build.
```

`npm update -g @vue/cli` → En cas de voler-ho  
actualitzar



## 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

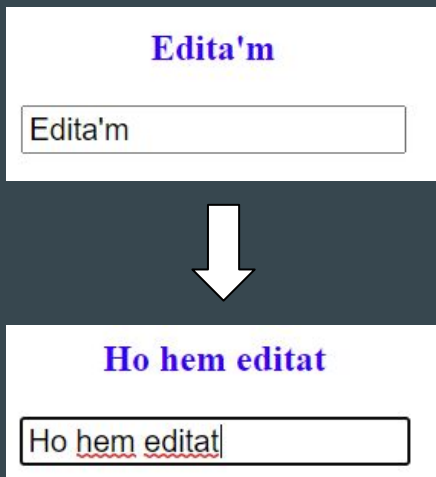
[Core Docs](#) [Forum](#) [Community Chat](#) [Twitter](#) [News](#)

### Ecosystem

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

# Com funciona?

- Processar dades de forma declarativa al DOM
- v- = atributs especials
- v-model = unió de dades bidireccionals (binding)



```
<template>
  <div id="app">
    <p>{{message}}</p>
    <input v-model="message">
  </div>
</template>
```

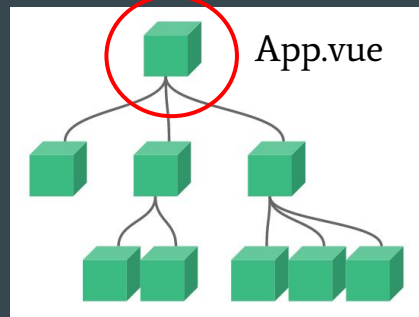
```
<script>
  export default {
    name: 'App',
    data: () => {
      return {
        message: 'Edita\'m'
      }
    },
  }
</script>
```

```
<style>
  #app {
    text-align: center;
  }
  p{
    color: blue;
    font-weight: bold;
  }
</style>
```

# Components

- Extensió dels elements bàsics d'HTML
- Codi reutilitzable
- Permet crear aplicacions a gran escala a partir de components petits

- Característiques
  - Independents
  - Reutilitzables



# Crear un component

- Fitxer .vue
- Templates → Plantilles
- Props = atributs
- Methods = funcions en javascript

```
<template>
  <div class="estudiant">
    <p>
      L'estudiant {{nom}} ha obtés un {{nota}} en l'assignatura {{assignatura}}
    </p>
  </div>
</template>

<script>
export default {
  name: 'Estudiant',
  props: {
    nom: String,
    nota: Number,
    assignatura: String,
  },
  methods: {
  }
}
</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped>
p{
  color: blue;
}
</style>
```



# Afegir un component

- Component pare
- :prop=x  
x és un variable guardada a data → prop tindrà el valor de la variable x
- prop=x  
prop tindrà el valor x

```
<template>
  <div id="app">
    <p>
      Aquest missatge sortirà de color vermell, ja que utilitza el style general, en <br>
      canvi la resta sortiran blaus pel fet que així ho especifica el component Estudiant
    </p>
    <Estudiant :nom='nom' :assignatura='assignatura' :nota='nota'></Estudiant>
    <Estudiant nom='Pedro' assignatura='PMUD' nota=8></Estudiant>
    <Estudiant nom='Maria' assignatura='REIN' nota=7></Estudiant>
  </div>
</template>

<script>
  import Estudiant from './components/Estudiant.vue'

  export default {
    name: 'App',
    data: () => {
      return{
        nom:"Joan",
        assignatura:"PROP",
        nota:6
      }
    },
    components:{
      Estudiant
    }
  }
</script>

<style>
  #app {
    text-align: center;
  }
  p{
    color: red;
  }
</style>
```

# Resultat

Aquest missatge sortirà de color vermell, ja que utilitza el style general, en canvi la resta sortiran blaus pel fet que així ho especifica el component Estudiant

L'estudiant Joan ha obtés un 6 en l'assignatura PROP

L'estudiant Pedro ha obtés un 8 en l'assignatura PMUD

L'estudiant Maria ha obtés un 7 en l'assignatura REIN

# Directives

- Atributs especials amb el prefix v-
  - v-if → inseriria/eliminarà un element

```
<div v-if="page === 'products'" class="other">  
  <ListProducts v-on:addItemToCart="addItemToCart"/>  
</div>
```

- v-for

```
<div v-for="(product, index) in products" :key="index" class="cart">  
  <Product :name="product.name" :price="product.price" :image="product.image"  
    v-on:addItemToCart="addItemToCart" />  
</div>
```

- v-on → escolta els esdeveniments DOM

```
<button v-on:click="addItemToCart()"> add to cart </button>
```

# Comunicació entre components

- De component pare a fill:
  - v-bind: (es pot abreviar en : ) indica que la variable s'actualitzarà de forma reactiva
  - v-on: quan el pare rep x funció del fill, aquest executa y

```
<Cart v-on:removeItemFromCart="removeItemFromCart" :cart="cart" />
```

- De component fill a pare:
  - \$emit

```
methods: {  
  addItemToCart(product){  
    this.$emit("addItemToCart", product);  
  }  
}
```

# Ventatges i inconvenients

- És progressiu
  - Bona documentació
  - Organitzat
  - Reactiu
  - Corba d'aprenentatge més senzilla
- 
- Poc utilitzat en comparació amb Angular i React



# Moltes gràcies per la vostra atenció



Link Github: [https://github.com/Whyvern99/VUE\\_Pmud](https://github.com/Whyvern99/VUE_Pmud)