VueJS

Nancy Cruz

Ing. en Tecnologías Computacionales

Agenda

1. Intro
2. Ventajas
3. Conceptos clave
4. Estructura
5. Cómo empezar
6. Instalación
7. Proyecto CLI

¿Qué es?

Open Source progressive JavaScript framework used to develop interactive web interfaces.

Se usa generalmente para crear **single-pages** apps que corren en el cliente pero pueden ser usadas para crear apps fullstack mediante HTTP requests a un servidor en el backend.

Ventajas

- Websites dinámicos
- Se integra fácilmente con otros proyectos
- Es rápido y ligero
- Virtual DOM
- OpenSource (no depende de compañías grandes como Google o Facebook)
- Gran soporte por la comunidad

Conceptos clave

- Virtual DOM
- Data Binding
- Components
- Event Handling
- Animation/Transition
- Computed Properties
- Templates
- Directives
- Watchers
- Routing
- Lightweight
- Vue-CLI

Estructura

- Template
- Script
- Style

Afecta al componente especificado

Afecta a todos los componentes en la página

C: > Users > nancy > Desktop > ex.js

```
<template>
<header>
    <h2>ToDo List</h2>
</header>
    <ul>
        <li v-for="(todo, index) in todos":key="index">
            <span :class="{ done: todo.done }" @click="doneTodo(todo)">{{ todo.conten
            <button @click="removeTodo(index)">Remove</button>
        </li>
    </ul>
    <h4 v-if="todos.length === 0">Empty list.</h4>
```

```
<style scoped>
header{
    margin: 10px;
    display: flex;
}
</style>

<style lang="scss">
body {
    margin: 0;
    padding: 0;
    font-family: Avenir, Helvetica, Arial, sans-serif;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
    background-color: $backgroundColor;
    color: $textColor;
```

```
orage.getItem('todos')) || defaultDat
```

```
newTodo.value = '';
}
saveData();
}
```

¿Cómo iniciamos?

Do you have NodeJS installed?

```
If (a="no"){
   goTo(https://nodejs.org/en/download/ );
   downloadLTS();
   nodeJsTerminal("node -v");
}
else {
   npmInstall("npm -v", "npm install npm@latest -g");
   goTo( https://vuejs.org/ );
   clickOn("Get Started");
}
```

https://github.com/ruz404/vuejsMC

Instalación 1

1. CDN

For prototyping or learning purposes, you can use the latest version with:

```html
<script src="https://unpkg.com/vue@next"></script>
```

For production, we recommend linking to a specific version number and build to avoid unexpected breakage from newer versions.

```
<script src="https://unpkg.com/vue@3.0.7"></script>
```

CDN

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>User Card</title>
</head>
<body>
    <div id="app">
        <h1>Hola, {{firstName}}!</h1>
        <img :class="gender" v-
bind:src="picture" :alt="`${firstName} ${lastName}`">
        <h1>{{firstName}} {{lastName}}</h1>
        <h3>{{gender}}</h3>
        <h3>{{email}} | {{phone}}</h3>
        <button v-
on:click="getUser()" :class="gender">Random User</button>
    </div>
    <!-- <h1>Hola {{firstName}}</h1> -->
    <script src="https://unpkg.com/vue@3.0.7"></script>
    <script src="app.js"></script>
</body>
</html>
```

```javascript
const app = Vue.createApp({
    data(){ //function that returns an object
        return {
            firstName: 'John',
            lastName: 'Smith',
            gender: 'male',
            email: 'john.smith@mail.com',
            phone: '812 365 7896',
            picture: 'https://randomuser.me/api/portraits/men/75.j
pg'
        }
    },
    methods:{
        async getUser(){
            const res = await fetch ('https://randomuser.me/api')
            const { results } = await res.json();
            this.firstName = results[0].name.first;
            this.lastName =  results[0].name.last;
            this.gender = results[0].gender;
            this.email = results[0].email;
            this.phone = results[0].phone;
            this.picture = results[0].picture.large;
        }
    }
})

app.mount('#app');
```

Instalación 2

2. NPM

NPM

NPM is the recommended installation method when building large scale applications with Vue. It pairs nicely with module bundlers such as Webpack or Browserify. Vue also provides accompanying tools for authoring Single File Components.

Shell

```
# latest stable
$ npm install vue
```

Instalación 3

3. Vue CLI

- Comando para crear proyectos
- Dev server
- Vue manager GUI
- Testing, typescript, otros

https://cli.vuejs.org/

Vue CLI

1. Get Stared
2. Installation

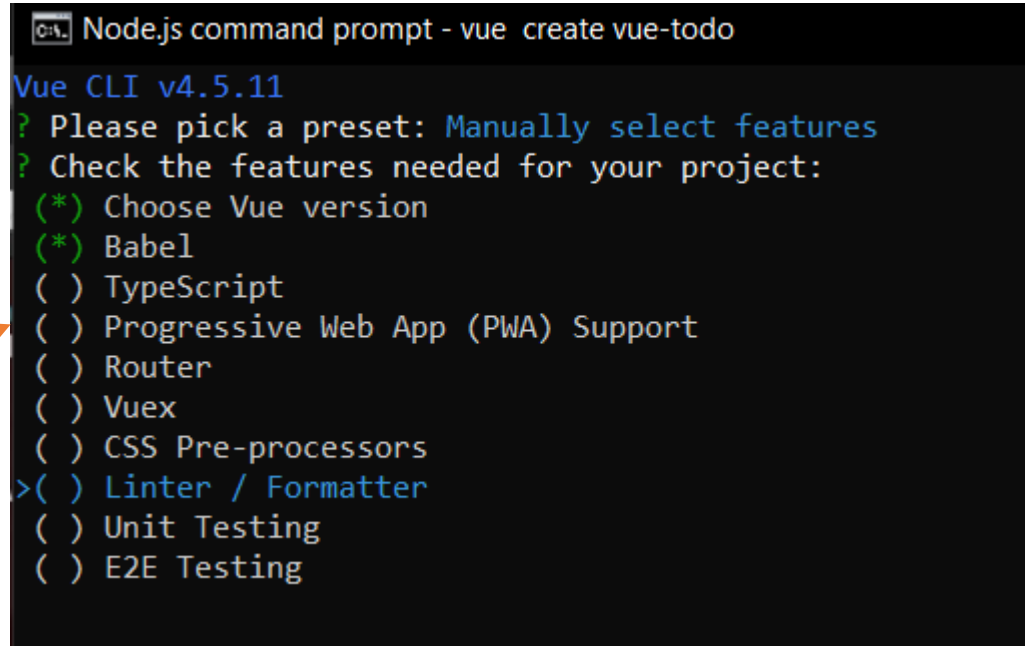
```
//Sin vue - PC
> npm install -g @vue/cli
> vue –version

//mac:
> Sudo npm i –g @vue/cli
> vue –version
```

3. vue UI (abrir GUI)
4. vue create
5. select features

```
//Instalaciones previas de vue
> vue --version

//es ¿1.x or 2.x?
> Sí→ npm uninstall vue-cli -g
```



//Crear proyecto

```
> vue create vue-todo
```

```
npm
C:\Users\nancy\Documents\MasterClass\vueMC\vuejsMC>vue create vue-todo


Vue CLI v4.5.11
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel
? Choose a version of Vue.js that you want to start the project with 3.x (Preview)
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config files
? Save this as a preset for future projects? No
? Pick the package manager to use when installing dependencies: NPM


Vue CLI v4.5.11
  Creating project in C:\Users\nancy\Documents\MasterClass\vueMC\vuejsMC\vue-todo.
  Installing CLI plugins. This might take a while...

[..................] / fetchMetadata: sill pacote range manifest for vue-style-loader@^4.1.2 fetched in 128ms
```

//Package

vuejsMC > vue-todo > {} package.json > {} scripts

```
1   {
2     "name": "vue-todo",
3     "version": "0.1.0",
4     "private": true,
      ▷ Debug
5     "scripts": {
6       "serve": "vue-cli-service serve",
7       "build": "vue-cli-service build"
8     },
9     "dependencies": {
10      "core-js": "^3.6.5",
11      "vue": "^3.0.0"
12    },
13    "devDependencies": {
14      "@vue/cli-plugin-babel": "~4.5.0",
15      "@vue/cli-service": "~4.5.0",
16      "@vue/compiler-sfc": "^3.0.0"
17    }
18  }
```

//index.html

vuejsMC > vue-todo > public > index.html > ...

```
1   <!DOCTYPE html>
2   <html lang="">
3     <head>
4       <meta charset="utf-8">
5       <meta http-equiv="X-UA-Compatible" content="IE=edge">
6       <meta name="viewport" content="width=device-width,initial-scale=1.0">
7       <link rel="icon" href="<%= BASE_URL %>favicon.ico">
8       <title><%= htmlWebpackPlugin.options.title %></title>
9     </head>
10    <body>
11      <noscript>
12        <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> does
13      </noscript>
14      <div id="app"></div>
15      <!-- built files will be auto injected -->
16    </body>
17  </html>
18
```

¿Dónde sucede la magia?

vuejsMC > vue-todo > src > JS main.js

```js
import { createApp } from 'vue'
import App from './App.vue'

createApp(App).mount('#app')

```

vuejsMC > vue-todo > src > App.vue

```vue
<template>
  <img alt="Vue logo" src="./assets/logo.png">
  <HelloWorld msg="Welcome to Your Vue.js App"/>
</template>

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

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

<style>
```

Settings (cmd+shift+p)

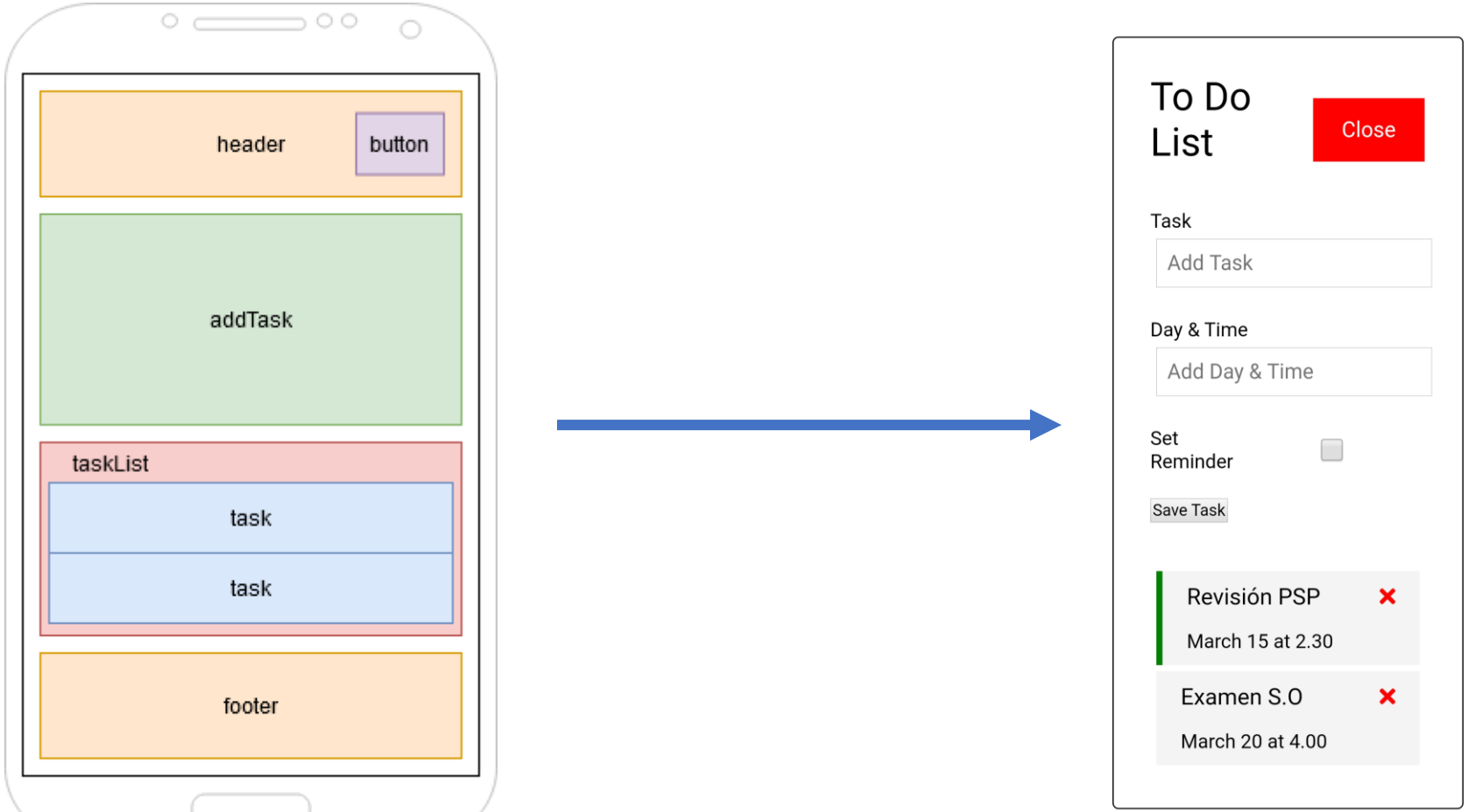
```
"emmet.includeLanguages":{
        "javascript": "javascriptreact",
        "vue-html": "html",
        "vue": "html"
    }
```

→ Install Vetur (highlighter)

Empecemos

1. Eliminar referencias del template HelloWorld

UI Components



Desarrollo

1. Crear componente HEADER
2. Agregar componente en App.vue
3. Props (header title) → `<Header title="To Do List" />`
4. Crear componente de BUTTON
5. Agregar componente Button en Header
6. Propiedades del Button
7. Agregar método de click

App.vue | Button.vue | Header.vue | package.json | style.css | app.js | index.html

vuejsMC > vue-todo > src > components > Button.vue > {} "Button.vue" > script

```vue
<template>
    <!-- <button>Add</button> -->
    <button @click="onClick()" :style="{background: color}">{{ text }}</button>
</template>

<script>
    export default{
        name: 'Button',
        props:{
            text: String,
            color: String
        },
        methods: {
            onClick(){
                console.log('click');
            }
        }
    }
</script>

<style scoped>
```

Desarrollo (2)

1. Generación de data (App)
2. Lifecycle
3. Método created

```
export default {
  name: 'App',
  components: {
    //1. HelloWorld
    Header
  },
  data(){
    return {
      tasks: []
    }
  },
  created(){
    this.tasks = [
      {id : 1, text: 'Revisión PSP', day: 'March 15 at 2.30', reminder: true,}, {id : 2, text: 'Examen S.O',
    ]
  }
}
```

Desarrollo (2)

5. Crear componente TASKS
6. Agregar componente Tasks en App.vue
7. Crear Loop de tasks en Tasks.vue

vuejsMC > vue-todo > src > components > Tasks.vue

```
1  <template>
2  <div v-for="task in tasks">
3      <h3>{{task.text}}</h3>
4  </div>
5  </template>
```

Dado que es un arreglo hay que especificar una llave única mediante un v-bind

vuejsMC > vue-todo > src > components > Tasks.vue > {} "Tasks.vue" > template > div

```
1  <template>
2      <div>
3          <div :key="task.id" v-for="task in tasks">
4              <h3>{{ task.text }}</h3>
5          </div>
6      </div>
7
8  </template>
```

Desarrollo (3)

1. Crear componente TASK
2. Importar Task en Tasks

vuejsMC > vue-todo > src > components > Tasks.vue > {} "Tasks.vue" >

```vue
<template>
    <div>
        <div :key="task.id" v-for="task in tasks">
            <!-- <h3>{{ task.text }}</h3> -->
            <Task :task="task" />
        </div>
    </div>

</template>

<script>
import Tasks from './Task'
    export default{
        name: 'Tasks',
        props:{
            tasks: Array,
        },
        components:{
            Task
        },
```

vuejsMC > vue-todo > src > components > Task.vue > {} "Task.vue" > style

```vue
<template>
    <h3>{{ task.text }}</h3>
</template>

<script>
    export default{
        name: 'Task',
        props:{
            task: Object,
        },
        methods: {

        }
    }
</script>

<style scoped>

</style>
```

Desarrollo (3)

1. Agregar fontawesome en index: https://cdnjs.com/
2. Agregar delete icon
3. Delete method
4. Config reminder

```
},
methods:{
    onDelete(id){
        // console.log(id);
        this.$emit('delete-task', id)
    }
```

```
toggleReminder(id){
this.tasks = this.tasks.map((task) =>
task.id === id ? {...task, reminder:
!task.reminder} : task)
}
```

Este método escala task>tasks>app

```
Tasks.vue
 <Task @delete-task="$emit('delete-
task', task.id)" :task="task" />

emits: ['delete-task'] //- array of
events

App.vue (pues aquí se tiene acceso a
data)
<Tasks @delete-task="deleteTask"
:tasks="tasks" />
```

Desarrollo 4

1. Crear componente ADDTASK
2. Uso de v-model: text, day, reminder
3. Función AddTask
4. Agregar v-show de toggle-add-task

Button>Header>App

Building for prod

- Let’s make requests!

```
npm run build
```

- **Dist** es la carpeta que pondremos en el server de producción
- Server para deploy dist folder

```
npm i –g serve
serve –s dist
```

*Nos permite acceder desde el teléfono para ver el ambiente

¡Gracias!

nancy.pcrz@gmail.com

ruz404