

# Unidad 3 - Frontend con Vue

# Manipulación del DOM con JavaScript

# The Progressive JavaScript Framework

An approachable, performant and versatile framework for building web user interfaces.

[Why Vue](#)[Get Started →](#)[Install](#)

Special Sponsor



Open Source Firebase Alternative

## Approachable

Builds on top of standard HTML, CSS and JavaScript with intuitive API and world-class documentation.

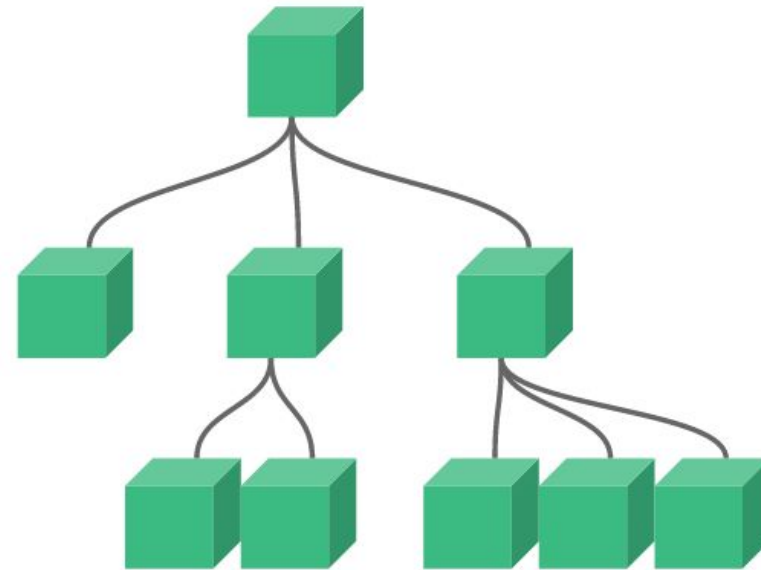
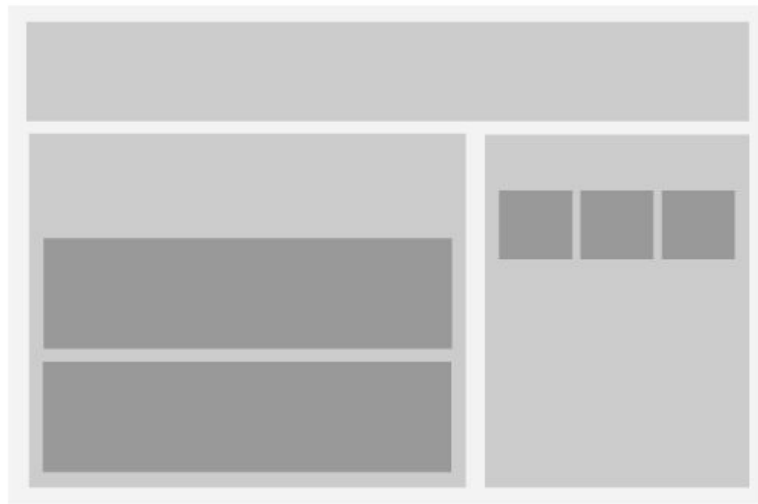
## Performant

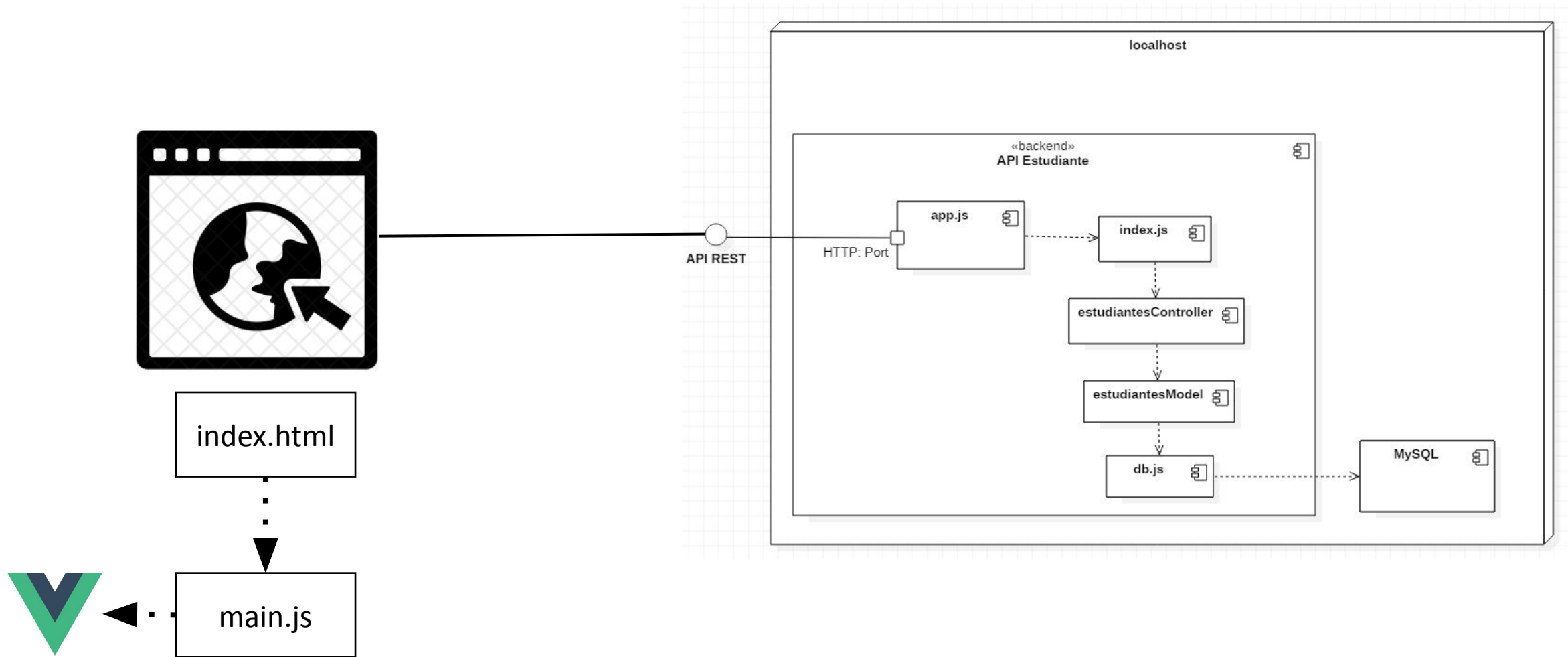
Truly reactive, compiler-optimized rendering system that rarely requires manual optimization.

## Versatile

A rich, incrementally adoptable ecosystem that scales between a library and a full-featured framework.

# Componentes





# Componentes

- Sirve para especializar y reusar elementos en la aplicación
- La organización de los componentes se vuelve complicada con aplicaciones grandes, por lo que se requiere SFC (single file components)

# Single File Components

- Se utilizan proyectos de Node.js con Webpack (o algún otro empaquetador de componentes - module bundler e.g. Browserify)
- Instalar
  - **`npm install -g @vue/cli`**

Fuente: <https://cli.vuejs.org/guide/installation.html>

Creación de proyectos



# Crear proyecto usando vue-cli

> `vue create mi-proyecto`

```
Vue CLI v3.12.0
? Please pick a preset:
  default (babel, eslint)
> Manually select features

Vue CLI v3.12.0
? Please pick a preset: Manually select features
? Check the features needed for your project:
  (*) Babel
  ( ) TypeScript
  ( ) Progressive Web App (PWA) Support
  (*) Router
  ( ) Vuex
> ( ) CSS Pre-processors
  (*) Linter / Formatter
  ( ) Unit Testing
  ( ) E2E Testing
```

```
Vue CLI v3.12.0
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, Router, Linter
? Use history mode for router? (Requires proper server setup for index fallback in production) No
? Pick a linter / formatter config: Prettier
? Pick additional lint features: (Press <space> to select, <a> to toggle all, <i> to invert selection) Lint on save
? Where do you prefer placing config for Babel, PostCSS, ESLint, etc.? In dedicated config files
? Save this as a preset for future projects? (y/N) n
```

> `cd mi-proyecto`

> `npm run serve`

# IU para crear proyectos

## vue ui

```
molguin@laptop1:~/javascript/vue$ vue ui
🚀 Starting GUI...
🔧 Ready on http://localhost:8000
```

# Administrador de Proyectos de Vue



 Proyectos

 Crear

 Importar




No hay proyectos

## Crear un proyecto nuevo

 Detalles

 Presets

 Características

 Configuración

Directorio del proyecto

 mi-proyecto

/home/molguin/javascript/vue/mi-proyecto



Gestor de paquetes

npm



Opciones adicionales

Sobreescribir directorio si ya existe



Esqueleto del proyecto sin instrucciones de principiante



Repositorio git

Inicializar repositorio git (recomendado)



 Cancelar

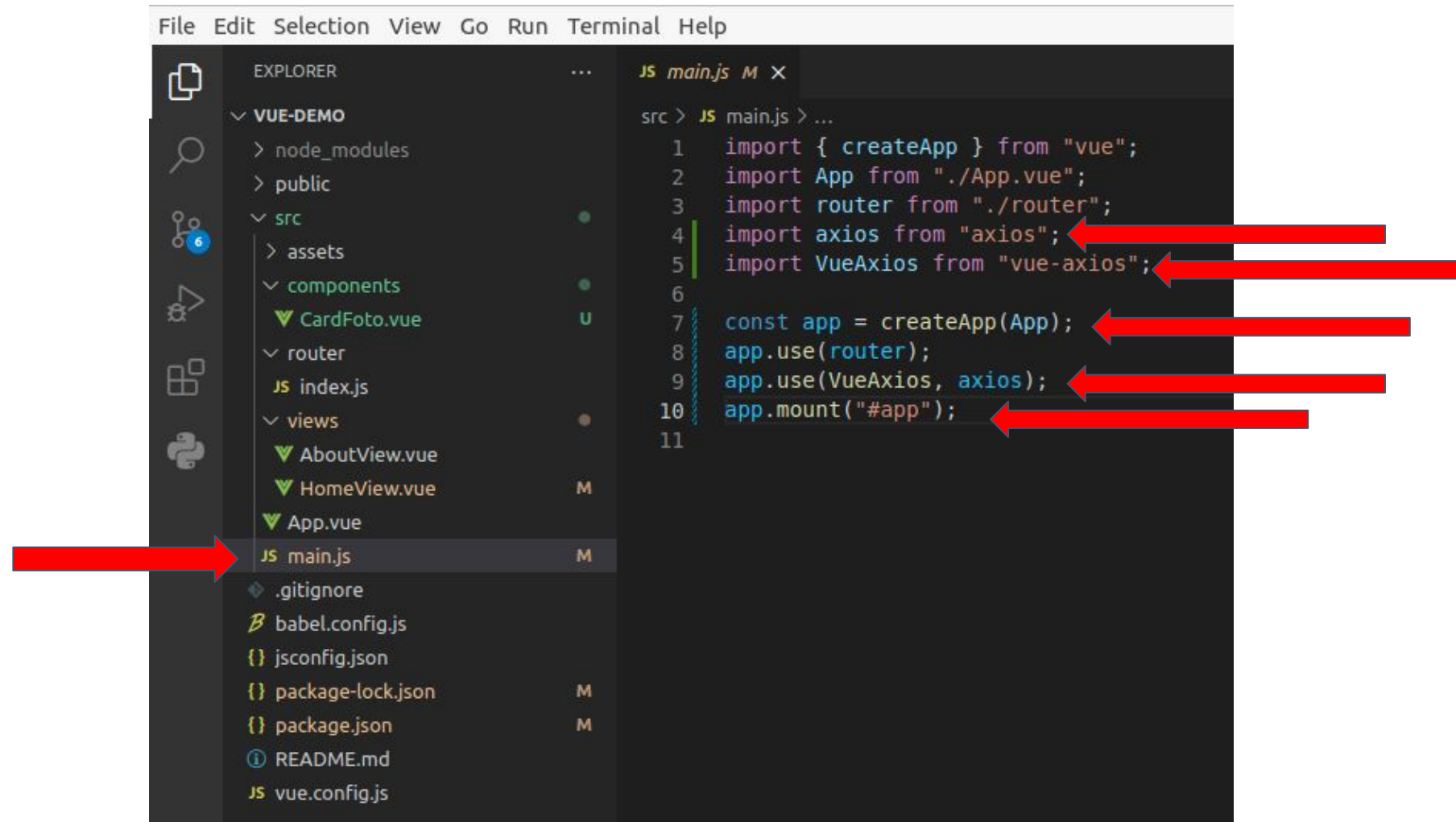
Siguiente 

# Instalar axios y vue-axios

npm i axios

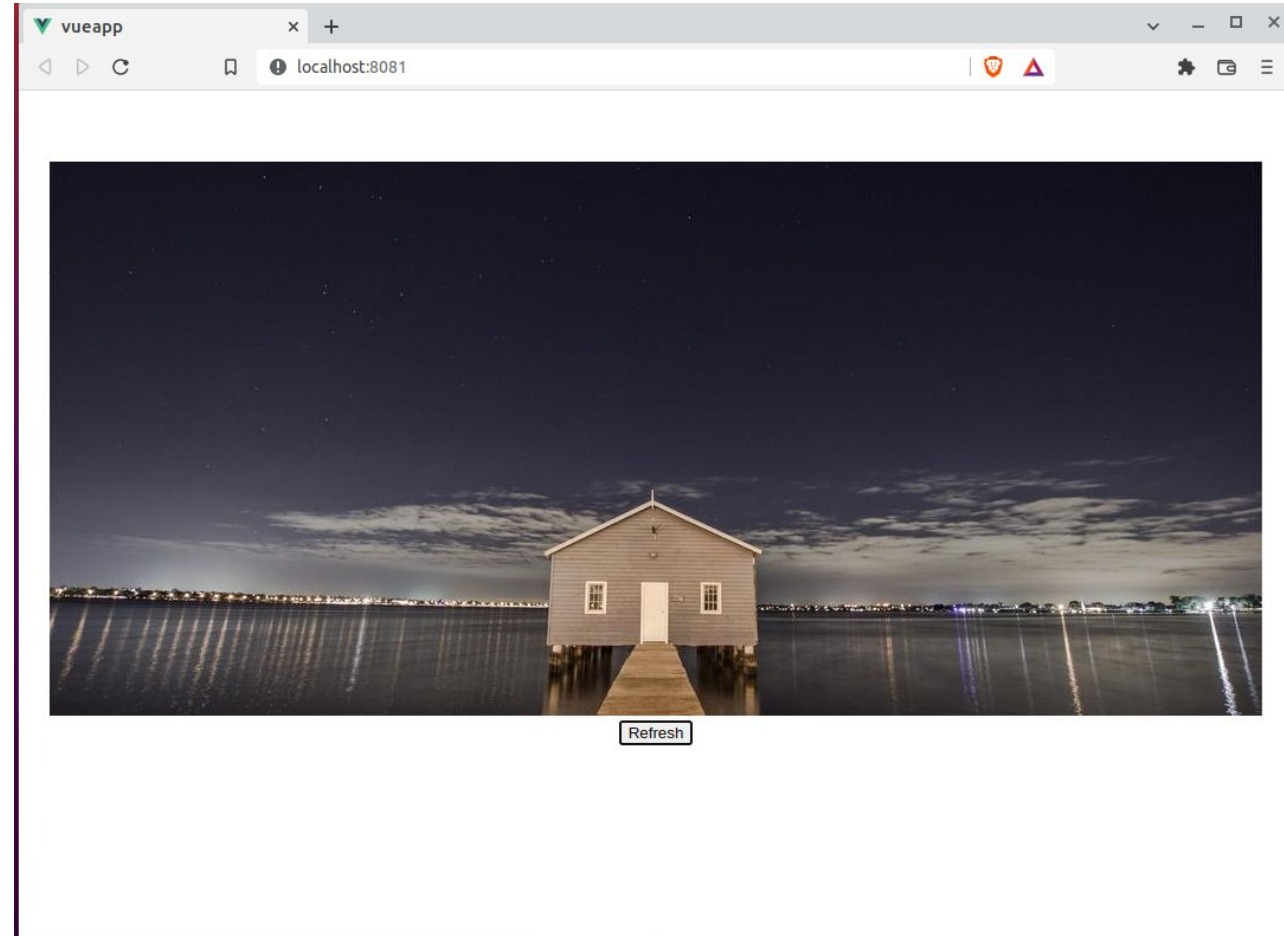
npm i vue-axios

# Configurar axios en Vue3

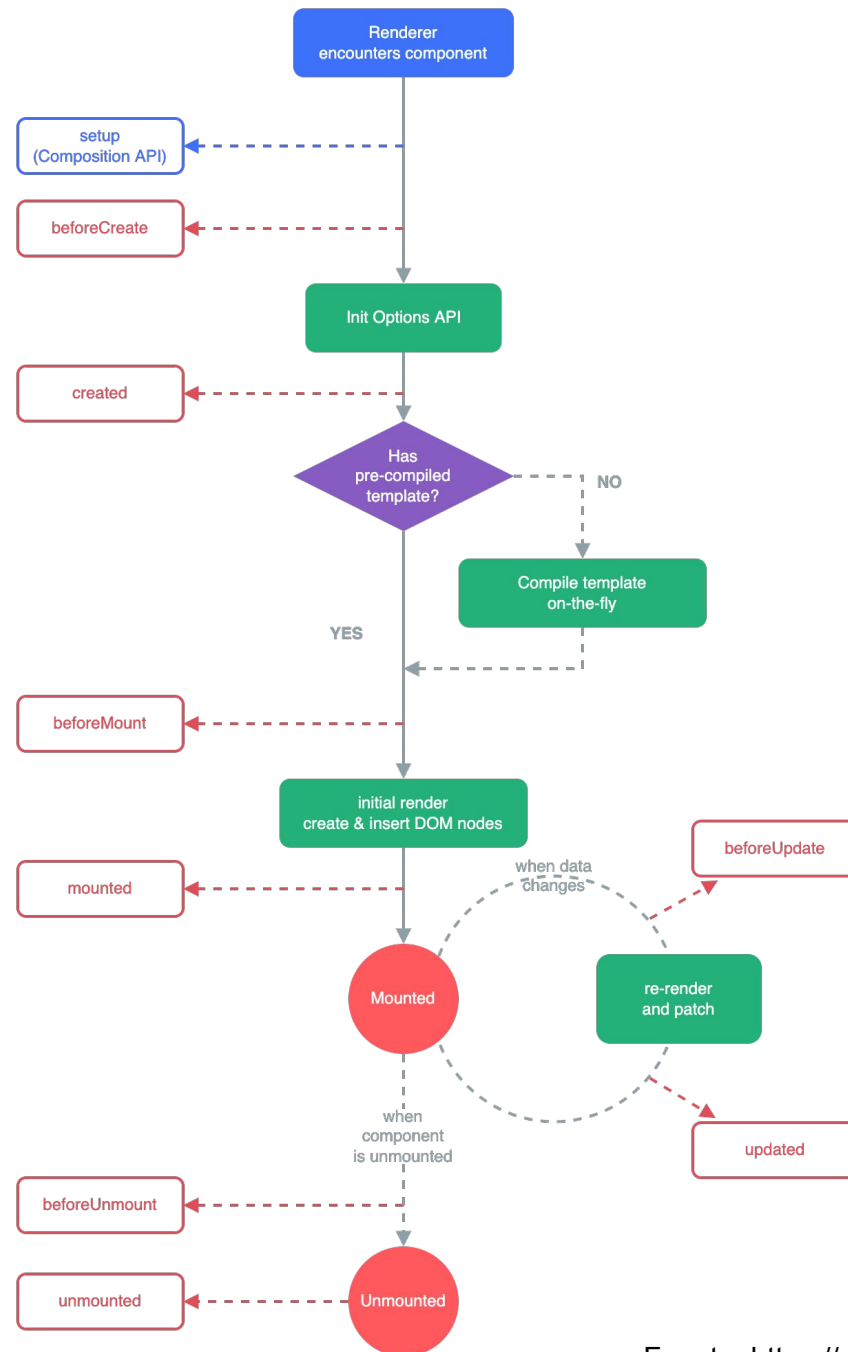


# Crear una Vue app

- Que acceda al servicio de imágenes random:  
<https://picsum.photos>
- Que despliegue la imagen en una página
- Que tenga un botón de “Refresh” para que actualice la imagen cada que sea presionado.



# Components Lifecycle





# Biblioteca de componentes gráficos

[Home](#)[Docs](#)[Components](#)[中文](#)[Dark](#)[GitHub](#)[2.33.5](#)

## Naive UI

A Vue 3 Component Library

Fairly Complete, Theme Customizable, Uses TypeScript, Fast  
Kinda Interesting

[Change Theme](#)[Get Started](#)

<https://www.naiveui.com>



# Instalar Naive UI

En el mismo directorio del proyecto creado con Vue

***npm i -D naive-ui***

***npm i -D vfonts***

# Uso de componentes

```
<template>
  <n-button>naive-ui</n-button>
</template>

<script>

  import { NButton } from 'naive-ui'

  export default {
    components: {
      NButton
    }
  }
</script>
```

```
<template>
  <div v-if="mostrar">
    <h1> Hola </h1>
  </div>
  <mi-boton @click="cambia">Switch</mi-boton>
</template>
<script>
import {NButton} from 'naive-ui';
export default {
  components: {
    MiBoton: NButton,
  },
  data() {
    return {
      mostrar: false
    }
  },
  methods: {
    cambia() {
      console.log(this.mostrar);
      this.mostrar = !this.mostrar;
    }
  }
}
</script>
<style>
</style>
```

```
<template>
  <div >
    <NDataTable :columns="columns" :data="data"></NDataTable>
  </div>
</template>
```

```
<script>
import { NDataTable } from 'naive-ui'

export default {
  components: {
    NDataTable
  },
  data() {
    return {
      columns: [
        {
          title: "No.",
          key: "no"
        },
        {
          title: "Descripcion",
          key: "descripcion"
        }
      ],
      data: [
        {
          no: "1",
          descripcion: "Hola Mundo"
        }
      ]
    }
  }
}
</script>
```

# Consultar la documentación

<https://www.naiveui.com/en-US/os-theme/components/data-table>

Simple Editable Table

Simple, but practical.

Name	Age	Address
<input type="text" value="John Brown"/>	<input type="text" value="32"/>	<input type="text" value="New York No. 1 Lake Park"/>
<input type="text" value="Jim Green"/>	<input type="text" value="42"/>	<input type="text" value="London No. 1 Lake Park"/>
<input type="text" value="Joe Black"/>	<input type="text" value="32"/>	<input type="text" value="Sidney No. 1 Lake Park"/>

<

1

>