



ISWD813– Aplicaciones Web Avanzadas
FACULTAD DE INGENIERÍA DE SISTEMAS

Desarrollo SPA con VUE



DOCENTE: Ing. Christian Suárez

Grupo: 2

Estudiante: Nicolás Omar Barragán Hidalgo

Fecha de entrega: 12-09/2023

Conceptos básicos

Las aplicaciones de React se construyen a partir de componentes, los cuales son elementos de la interfaz de usuario (UI) que poseen su propia funcionalidad y aspecto. Un componente puede tener un tamaño tan reducido como el de un botón o ser tan extenso como una página completa. Entre las características principales de este marco de trabajo están las siguientes herramientas [1]:

Marco de trabajo progresivo

Vue está diseñado para ser flexible y adaptable de manera incremental, lo que significa que se puede utilizar de diversas formas según el caso de uso. Puede mejorar el HTML estático sin necesidad de un paso de construcción, integrarse como Componentes Web en cualquier página, utilizarse para crear Aplicaciones de Una Sola Página (SPA)

Componentes de un solo archivo

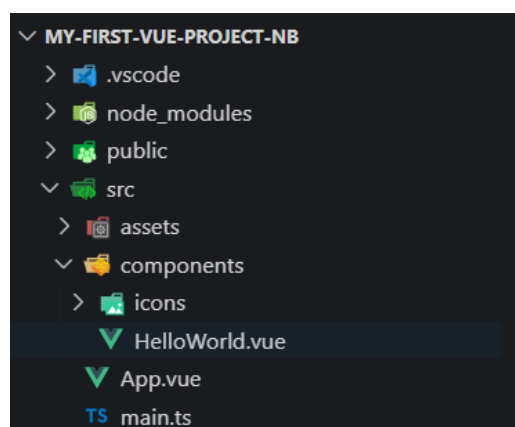
Vue utiliza un formato de archivo similar a HTML llamado "Single-File Component" o SFC. Como su nombre indica, un SFC de Vue encapsula la lógica del componente (JavaScript), la plantilla (HTML) y los estilos (CSS) en un solo archivo. Los SFC son una característica fundamental de Vue y son la forma recomendada de crear componentes de Vue.

Características principales

Ciclo de vida del componente

Cuando se crea una instancia de componente en Vue, esta instancia pasa por una secuencia de pasos esenciales para prepararse y funcionar correctamente. Estos pasos incluyen la configuración inicial de la observación de datos para detectar cambios, la compilación de la plantilla para generar la interfaz de usuario, la conexión de la instancia al DOM (Document Object Model) para que los usuarios puedan interactuar con ella y la capacidad de actualizar automáticamente el DOM cuando los datos cambian [1].

Estructura del proyecto



Según la estructura de la ilustración anterior lo más importante de destacar es en donde se crean los componentes. En la carpeta "src/components" es en donde se crean los archivos SFC y en "App.vue" es la página principal en donde se hace el llamado de los componentes.

SFC

Como se mencionó un SFC es una forma de estructurar componentes de Vue en un solo archivo que encapsula la lógica JavaScript, la plantilla HTML y los estilos CSS en un único archivo. Estos archivos tienen la extensión .vue. En la siguiente ilustración se podrá ver un ejemplo de creación de este tipo de archivo para el ejemplo que se presentará más adelante

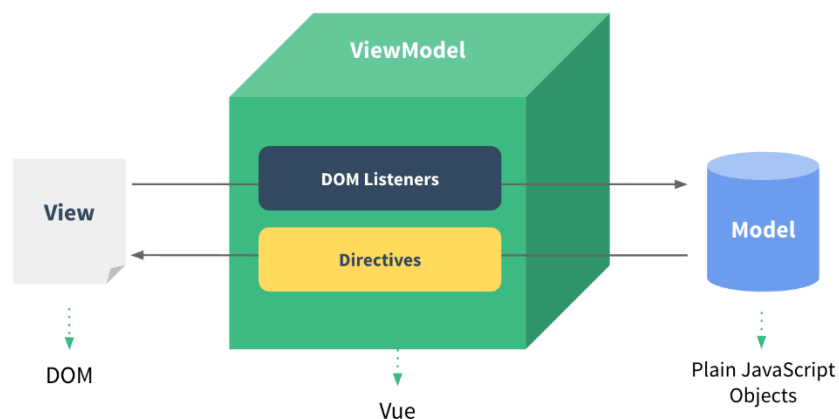
```
<script setup lang="ts">
  defineProps<{
    msg: string
 }>()
</script>

<template>
  <div class="greetings">
    <h1 class="green">{{ msg }}</h1>
    <h3>
      My name is Nicolás Barragán
    </h3>
  </div>
</template>
```

El código anterior comienza con la llamada a un script que recibirá un mensaje; este mensaje se ingresa en el componente principal o en el que va a ser llamado. En el template se encuentra toda la estructura del componente o el HTML común. Los estilos, si los hubiera, se añaden debajo del template, todo dentro del mismo archivo.

Arquitectura

Vue.js se centra en la capa ViewModel del patrón MVVM (Model-View-ViewModel). Conecta la Vista y el Modelo a través de enlaces de datos bidireccionales. Las manipulaciones del DOM real y el formato de salida se abstraen en Directivas y Filtros. La API de Vue.js está fuertemente influenciada por AngularJS. A pesar de las similitudes, Vue.js ofrece una alternativa valiosa a estas bibliotecas existentes al encontrar un equilibrio entre simplicidad y funcionalidad [2].



Ejemplo

En este primer proyecto se mostrará un Hola mundo que represente la estructura más básica de Vue. Para crear un proyecto React únicamente se necesita tener Node.js y utilizar el comando “npm create vue “. Después de seguir las configuraciones se tendrá el proyecto funcional.

Retomando de la explicación de SFC en el mismo componente definiremos los estilos de la siguiente manera:

```
</template>

<style scoped>
h1 {
  font-weight: 500;
  font-size: 2.6rem;
  position: relative;
  top: -10px;
}

h3 {
  font-size: 1.2rem;
}

1 reference
.greetings h1,
1 reference
.greetings h3 {
  text-align: center;
}
```

Ahora simplemente se hace llamado al componente en el archivo “App.vue” siempre tomando en cuenta las importaciones, pero esto generalmente va en las etiquetas de script tal y como se muestra a continuación:

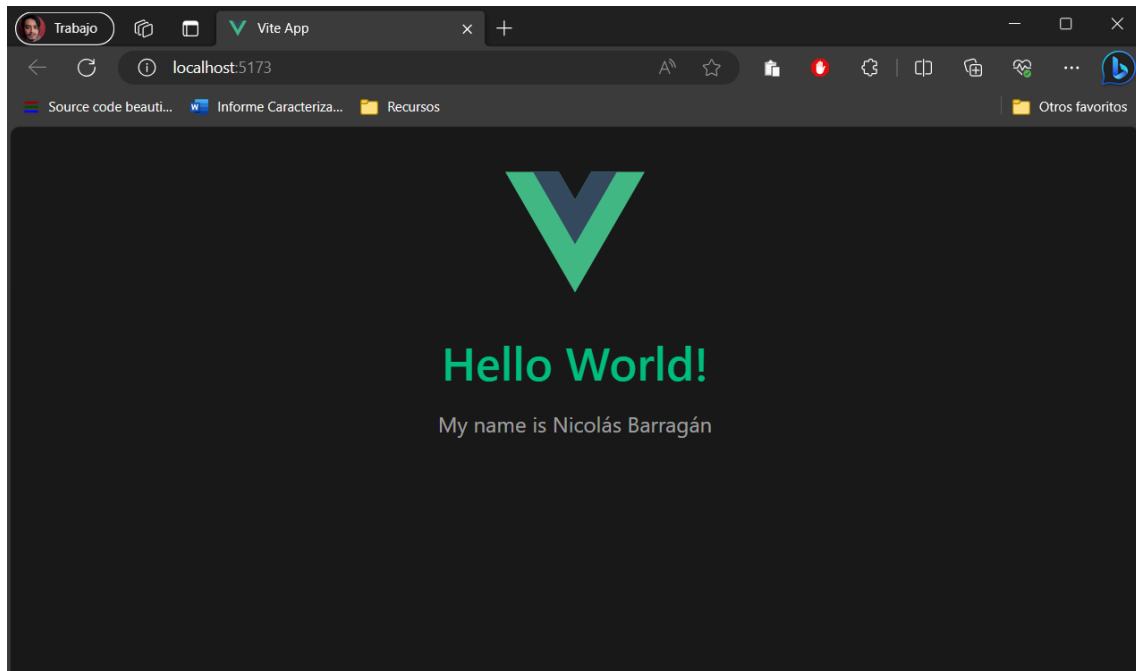
```
<script setup lang="ts">
import HelloWorld from './components/HelloWorld.vue'
</script>

<template>
  <header>
    

    <div class="wrapper">
      <HelloWorld msg="Hello World!" />
    </div>
  </header>
</template>

<style scoped>
header {
  line-height: 1.5;
}
```

Según el código anterior el llamado a nuestro componente se lo representa como una etiqueta y aparte se le manda el mensaje esperado. Con esto último el resultado final de correr el proyecto es el siguiente



Referencias

- [1] «Vue,» [En línea]. Available: <https://vuejs.org/guide/introduction.html#what-is-vue>. [Último acceso: 12 Septiembre 2023].
- [2] «Vue guide,» [En línea]. Available: <https://012.vuejs.org/guide/>. [Último acceso: 12 Septiembre 2023].

Anexos

Repositorio en Github

[NicolasBH9/my-first-vue-project-nb \(github.com\)](https://github.com/NicolasBH9/my-first-vue-project-nb)