



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

- Denominación del Programa de Formación: Tecnico Programación de Software..
- Código del Programa de Formación:
- Nombre del Proyecto:
- Fase del Proyecto: Desarrollo
- Actividad de Proyecto:
- Competencia: CONSTRUIR EL SISTEMA QUE CUMPLA CON LOS REQUISITOS DE LA SOLUCIÓN INFORMÁTICA.
- Resultados de Aprendizaje Alcanzar: Construir la interfaz de usuario, apoyado en la evaluación del prototipo, determinando las entradas y salidas requeridas en el diseño y definiendo los lineamientos para la navegación, de acuerdo con las necesidades del usuario.
- Duración de la Guía: 8 horas

2. PRESENTACIÓN

Vue.JS Framework



Vue JS, es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas Single-Page Applications cuando se utiliza en combinación con herramientas modernas y librerías de apoyo.

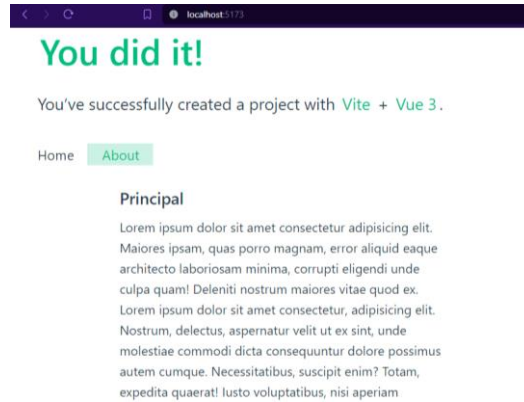
3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

- Descripción de la(s) Actividad(es):
 - Cree un proyecto Vue desde la terminal

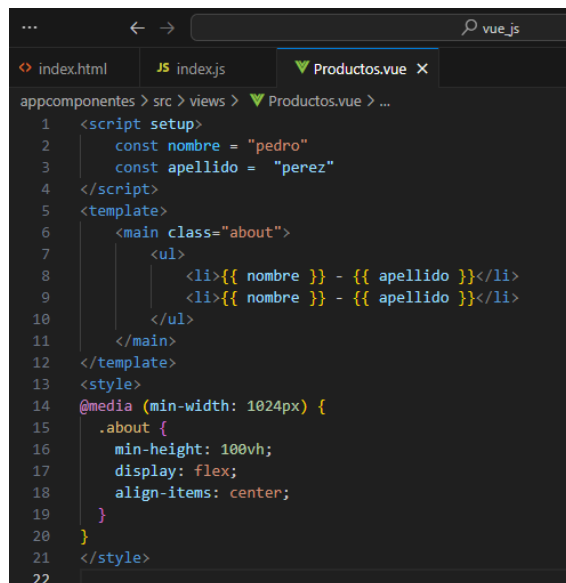




- El proyecto debe tener el ruteo habilitado.



Podemos pasar valores al template usando las {{ }} para mostrar los datos.



Así mismo cuando creamos propiedades las podemos crear en el template.

```
<script setup>
defineProps({
  msg: { type: String, required: true },
  likes : Number })
</script>
<template>
  <div class="greetings">
    <h1 class="green">{{ msg }}</h1>
    <h3>    {{ likes }}    </h3>
  </div>
</template>
```



Cuando tenemos una lista o un objeto por recorrer debemos realizarlo a través de un ciclo for, v-for

Con una lista:

```
index.html JS index.js Productos.vue X
appcomponentes > src > views > Productos.vue > ...
1 <script setup>
2   const datos = [
3     { nombre : "pedro", apellido : "perez"},
4     { nombre : "pedro", apellido : "perez"},
5     { nombre : "pedro", apellido : "perez"},
6   ]
7 </script>
8 <template>
9   <main class="about">
10     <ul>
11       <li v-for = "item in datos" >
12         {{ item.nombre }} - {{ item.apellido }}</li>
13     </ul>
14   </main>
15 </template>
16 <style>
17 @media (min-width: 1024px) {
18   .about {
19     min-height: 100vh;
20     display: flex;
21     align-items: center;
22   }
23 }
24 </style>
25
```

Con un objeto:

```
index.html JS index.js Productos.vue
appcomponentes > src > views > Productos.vue > ...
1 <script setup>
2   const datos =
3     { nombre : "pedro", apellido : "perez" }
4 </script>
5 <template>
6   <main class="about">
7     <ul>
8       <li v-for = "item in datos" >
9         {{ item }} </li>
10    </ul>
11  </main>
12 </template>
13 <style>
14 @media (min-width: 1024px) {
15   .about {
16     min-height: 100vh;
17     display: flex;
18     align-items: center;
19   }
20 }
21 </style>
22
```

Cuando queremos validar los valores podemos usar la sentencia v-if

```
<script setup>
  const valor = false
</script>
<template>
  <div v-if="valor">
    <h1> Se muestra el dato.....</h1>
  </div>
</template>
```

Si el valor no es verdadero podemos usar el v-else

```
<script setup>
  const valor = false
</script>
<template>
  <div v-if="valor">
    <h1> Se muestra el dato.....</h1>
  </div>
  <div v-else>
    <h1> No Se muestra el dato.....</h1>
  </div>
</template>
```

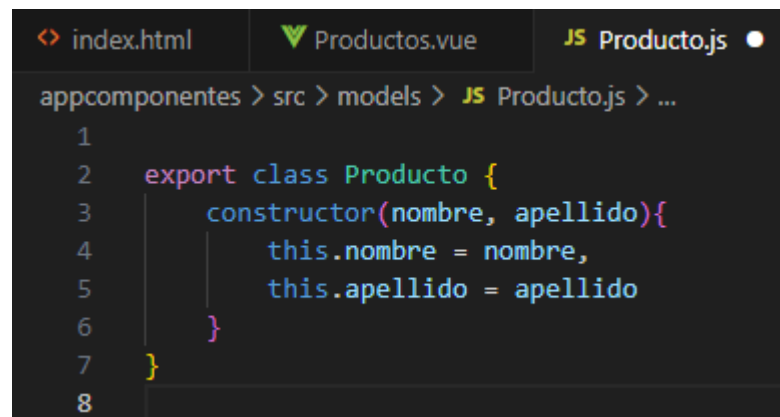
Cuando se evalúan varios valores usamos el condicional v-else-if

```
<script setup>
  const valor = false
</script>
<template>
  <div v-if="type === 'A'">
    A
  </div>
```



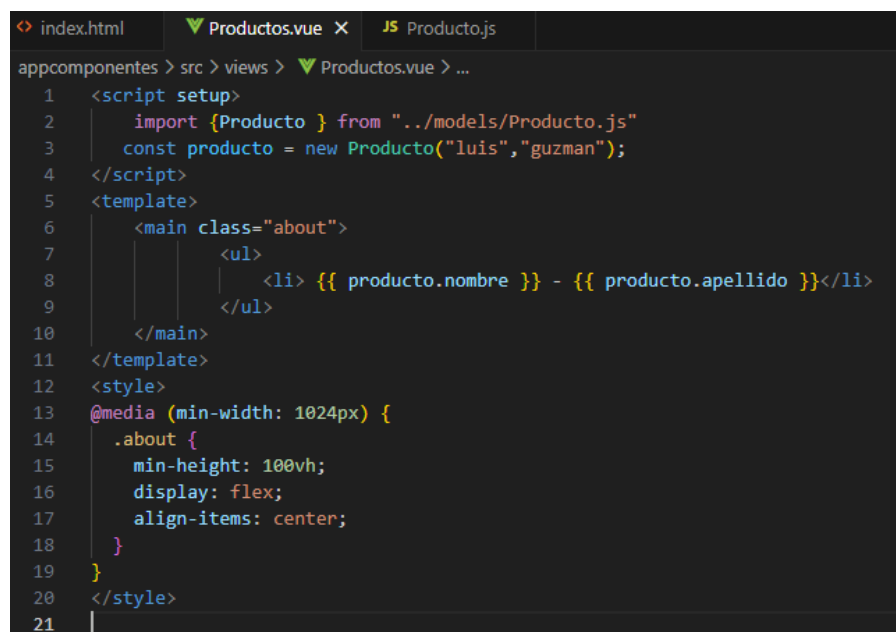
```
</div>
<div v-else-if="type === 'B'">
  B
</div>
<div v-else-if="type === 'C'">
  C
</div>
<div v-else>
  Not A/B/C
</div>
</template>
```

Para utilizar clases debemos realizarlas en una carpeta independiente y generarlas tipo export.



```
index.html  Productos.vue  JS Producto.js
appcomponentes > src > models > JS Producto.js > ...
1
2  export class Producto {
3      constructor(nombre, apellido){
4          this.nombre = nombre,
5          this.apellido = apellido
6      }
7  }
8
```

Desde el componente las importamos y las mostramos por pantalla.



```
index.html  Productos.vue X  JS Producto.js
appcomponentes > src > views > Productos.vue > ...
1  <script setup>
2      import {Producto } from "../models/Producto.js"
3      const producto = new Producto("luis","guzman");
4  </script>
5  <template>
6      <main class="about">
7          <ul>
8              <li> {{ producto.nombre }} - {{ producto.apellido }}</li>
9          </ul>
10     </main>
11 </template>
12 <style>
13 @media (min-width: 1024px) {
14     .about {
15         min-height: 100vh;
16         display: flex;
17         align-items: center;
18     }
19 }
20 </style>
21
```



Para consumir una Api a través del método REST debemos hacer lo siguiente:

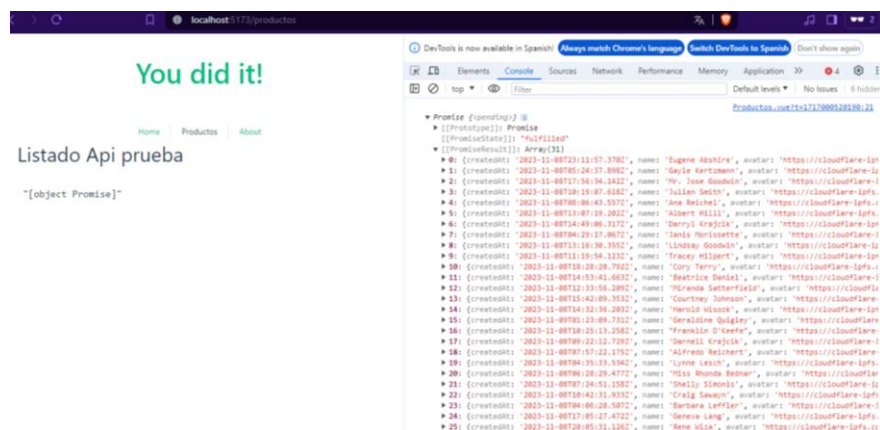
Crear la clase que se conectara a la Api.

```
index.html | Productos.vue | JS ConsultaApijs X
appcomponentes > src > models > JS ConsultaApijs > ...
1
2 export class ConsultaApi{
3
4   URL = "https://654c35fa7720d6ba8589ea2.mockapi.io/cliente/cliente"
5   async consultar(){
6     const respuesta = await fetch(this.URL)
7     const resultado = await respuesta.json()
8     return resultado
9   }
10 }
11
```

Importar la clase al componente y validar la llegada de los datos.

```
index.html | Productos.vue X | JS ConsultaApijs
appcomponentes > src > views > Productos.vue > ...
1 <script setup>
2 //import {Producto } from '../models/Producto.js' ...
3
13 import {ConsultaApi} from '../models/ConsultaApi.js'
14 const api = new ConsultaApi()
15 const datosApi = api.consultar()
16 console.log(datosApi);
17 </script>
18 <template>
19 <main class="about">
20 <!-- <ul> ...
21 <h1>
22 Listado Api prueba
23 </h1>
24 <br/>
25 <pre> {{ datosApi }}</pre>
26 </main>
27 </template>
28
```

Observamos el resultado en la consola:





Ahora procedemos a pasarle los datos al template para renderizar la Api en HTML

```
index.html | Productos.vue | JS ConsultaApi.js
appcomponentes > src > views > Productos.vue > ...
1  <script>
2    import {ConsultaApi} from '../models/ConsultaApi.js'
3    const api = new ConsultaApi()
4    export default {
5      data() {
6        return {
7          products: []
8        }
9      },
10     async mounted() {
11       this.products = await this.fetchData();
12     },
13     methods: {
14       async fetchData(){
15         return api.consultar()
16       }
17     }
18   }
19 > // const listadoApi = api.consultar()...
21 </script>
22 <template>
23   <main class="about">
24     <h1>
25       Listado Api prueba
26     </h1>
27     <br/>
28     <ul>
29       <li v-for = "item in products" > {{ item.name }} - {{ item.name }}</li>
30     </ul>
31   </main>
32 </template>
33
```

Observamos el resultado.

localhost:5173/productos

You did it!

Home | Productos | About

Listado Api prueba

- Eugene Abshire - Eugene Abshire
- Gayle Kertzmnn - Gayle Kertzmnn
- Mr. Jose Goodwin - Mr. Jose Goodwin
- Julian Smith - Julian Smith
- Ana Reichel - Ana Reichel
- Albert Hilll - Albert Hilll
- Darryl Krajcik - Darryl Krajcik
- Janis Morissette - Janis Morissette
- Lindsay Goodwin - Lindsay Goodwin
- Tracey Hilpert - Tracey Hilpert
- Cory Terry - Cory Terry
- Beatrice Daniel - Beatrice Daniel
- Miranda Satterfield - Miranda Satterfield
- Courtney Johnson - Courtney Johnson
- Harold Wisozk - Harold Wisozk
- Geraldine Quigley - Geraldine Quigley

DevTools is now available in Spanish!

Always match Chrome's language | Switch DevTools to Spanish | Don't show again

Elements | Console | Sources | Network | Performance | Memory

top | Filter | Default levels | No Issues | 2 hidden

Productos.vue:29

▼ Promise {<pending>}

[[Prototype]]: Promise

[[PromiseState]]: "fulfilled"

▼ [[PromiseResult]]: Array(31)

0: {createdAt: '2023-11-08T23:11:57.378Z', name: 'Eugene Abshire', avatar: 'https://cloudinary.com/avatar/...'}
1: {createdAt: '2023-11-08T05:24:37.898Z', name: 'Gayle Kertzmnn', avatar: 'https://cloudinary.com/avatar/...'}
2: {createdAt: '2023-11-08T17:56:34.142Z', name: 'Mr. Jose Goodwin', avatar: 'https://cloudinary.com/avatar/...'}
3: {createdAt: '2023-11-08T10:19:07.618Z', name: 'Julian Smith', avatar: 'https://cloudinary.com/avatar/...'}
4: {createdAt: '2023-11-08T08:06:43.557Z', name: 'Ana Reichel', avatar: 'https://cloudinary.com/avatar/...'}
5: {createdAt: '2023-11-08T13:07:19.202Z', name: 'Albert Hilll', avatar: 'https://cloudinary.com/avatar/...'}
6: {createdAt: '2023-11-08T14:49:06.317Z', name: 'Darryl Krajcik', avatar: 'https://cloudinary.com/avatar/...'}
7: {createdAt: '2023-11-08T04:29:17.067Z', name: 'Janis Morissette', avatar: 'https://cloudinary.com/avatar/...'}
8: {createdAt: '2023-11-08T13:16:30.355Z', name: 'Lindsay Goodwin', avatar: 'https://cloudinary.com/avatar/...'}
9: {createdAt: '2023-11-08T11:19:54.123Z', name: 'Tracey Hilpert', avatar: 'https://cloudinary.com/avatar/...'}
10: {createdAt: '2023-11-08T18:28:20.792Z', name: 'Cory Terry', avatar: 'https://cloudinary.com/avatar/...'}
11: {createdAt: '2023-11-08T14:53:41.663Z', name: 'Beatrice Daniel', avatar: 'https://cloudinary.com/avatar/...'}
12: {createdAt: '2023-11-08T12:33:56.209Z', name: 'Miranda Satterfield', avatar: 'https://cloudinary.com/avatar/...'}
13: {createdAt: '2023-11-08T15:42:09.353Z', name: 'Courtney Johnson', avatar: 'https://cloudinary.com/avatar/...'}
14: {createdAt: '2023-11-08T14:32:36.203Z', name: 'Harold Wisozk', avatar: 'https://cloudinary.com/avatar/...'}
15: {createdAt: '2023-11-09T01:23:09.731Z', name: 'Geraldine Quigley', avatar: 'https://cloudinary.com/avatar/...'}
16: {createdAt: '2023-11-08T10:25:13.258Z', name: 'Franklin O'Keefe', avatar: 'https://cloudinary.com/avatar/...'}
17: {createdAt: '2023-11-08T09:22:12.729Z', name: 'Darnell Krajcik', avatar: 'https://cloudinary.com/avatar/...'}
18: {createdAt: '2023-11-08T07:57:22.175Z', name: 'Alfredo Reichert', avatar: 'https://cloudinary.com/avatar/...'}
19: {createdAt: '2023-11-08T04:35:33.534Z', name: 'Lynne Lesch', avatar: 'https://cloudinary.com/avatar/...'}
20: {createdAt: '2023-11-08T06:28:29.477Z', name: 'Miss Rhonda Bednar', avatar: 'https://cloudinary.com/avatar/...'}
21: {createdAt: '2023-11-08T07:24:51.158Z', name: 'Shelly Simonis', avatar: 'https://cloudinary.com/avatar/...'}
22: {createdAt: '2023-11-08T10:42:31.933Z', name: 'Craig Sawayn', avatar: 'https://cloudinary.com/avatar/...'}
23: {createdAt: '2023-11-08T04:06:20.507Z', name: 'Barbara Leffler', avatar: 'https://cloudinary.com/avatar/...'}
24: {createdAt: '2023-11-08T17:05:27.472Z', name: 'Geneva Lang', avatar: 'https://cloudinary.com/avatar/...'}



Para importar imágenes podemos usar:

```
<template>
  <div>
    
  </div>
</template>
```

```
<template>
  <div>
    
  </div>
</template>
```

Validamos la funcionalidad de la App.

- Ambiente Requerido: Aula con equipos de computo e Internet
- Materiales: Tablero, Video vean o TV, programa IDE NetBeans

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento :	El aprendiz debe manejar los conceptos de básicos para la instalación de React.js	Lista de Chequeo
Evidencias de Desempeño:	Instalar y dejar funcional el paquete React y sus comandos básicos.	
Evidencias de Producto:	Realizar la compilación de un archivo React.js.	

5. GLOSARIO DE TÉRMINOS

6. REFERENTES BIBLIOGRÁFICOS



7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Oscar Iván Benavides C.	Instructor	CBA	Enero 26 de 2021

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					