



Vue Router

Manejo de rutas

***Implementar la navegación
de una interfaz web
utilizando Vue Router
para dar solución
a un requerimiento.***

- Unidad 1:
Vue Router
- Unidad 2:
Vuex
- Unidad 3:
Firebase
- Unidad 4:
Pruebas Unitarias y end-to-end
en un entorno Vue



Te encuentras
aquí



¿Qué aprenderás en esta sesión?

- *Construye una aplicación web usando Vue y Vue router para manejar rutas estáticas, dinámicas o anidadas y así resolver el problema planteado.*
- *Construye una aplicación web usando Vue y Vue Router para redireccionar a “página 404” en caso de ruta no encontrada.*

¿Cuál es el nombre de la carpeta y archivo en el que se controlan y definen las rutas de una aplicación vue js?



¿Es la propiedad del
objeto json en la cual se
define la ruta de
redirección?



Las páginas en una
aplicación vue js son
manejadas por vue
router como...



Contexto antes de iniciar


En esta sesión veremos más elementos asociados a vue router. Para comprender su funcionamiento es necesario comprender elementos que se integran en el HTML como `<router-link>` y `<router-view>`. Además, revisaremos qué son las rutas estáticas, dinámicas, paso de props a un componente según la ruta, redirecciones, entre otros elementos.



/* Rutas estáticas */

¿Cómo se definen las rutas estáticas?

Las rutas estáticas son representaciones de los componentes de Vue que queremos presentarle al usuario. Recordemos que estas se definen a través de objetos JSON indicando en la propiedad path la dirección.



```
const routes = [  
  {  
    path: '/',  
    name: 'home',  
    component: HomeView  
  },  
]
```

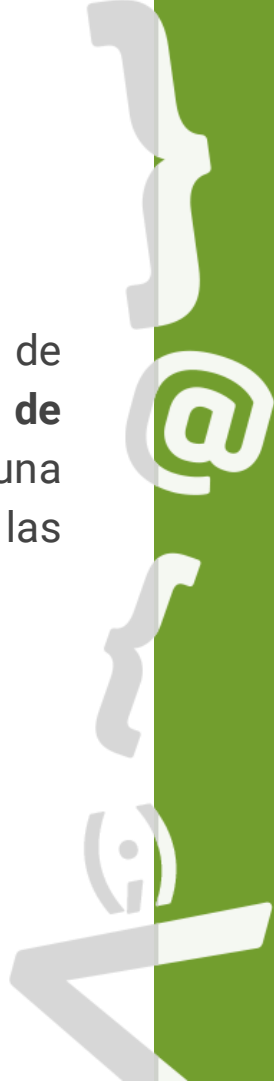
Ejercicio guiado: "Aplicando rutas estáticas"



Aplicando rutas estáticas

A continuación, realizaremos un ejercicio en el cual practicaremos la creación de rutas estática. Para lograrlo, deberán descargar de la plataforma el “**Material de apoyo - Ejercicio aplicando rutas estáticas**”. Este contiene la base de una aplicación vue js con vue router integrado. Luego de descargar, realiza las siguientes acciones:

- **Acción 1:** Descomprimir en una carpeta el material de apoyo.
- **Acción 2:** Abrir la carpeta descomprimida en el editor.
- **Acción 3:** Ejecutar el comando npm install para instalar las dependencias.



Sigue los pasos...

- **Paso 1:** Abrimos el archivo `router/index.js` y agregamos una nueva ruta llamada `services`.

```
{
  path: '/services',
  name: 'services',
  component: () => import(/* webpackChunkName: "about" */ '../views/ServicesView.vue')
}
```


- **Paso 2:** En la carpeta `/views` agregamos un nuevo componente llamado `ServicesView.vue`. El código de esta vista se muestra como en la imagen.

```
<template lang="">
  <div class="services">
    <h1>Servicios</h1>
  </div>
</template>
```



Sigue los pasos

- **Paso 3:** En el componente principal `App.vue`, agregamos el nuevo link a esta nueva ruta generada.



```
<template>
  <nav>
    <router-link to="/">Home</router-link> |
    <router-link to="/about">About</router-link> |
    <router-link to="/services">Services</router-link>
  </nav>
  <router-view/>
</template>
```

Resultado parcial de ejecución

- Con este resultado parcial, evidenciamos cómo agregar una ruta estática (*services*).
- Además, vimos qué es la etiqueta `<router-link>` y su comportamiento para redirigir y generar la navegación entre las vistas.

[Home](#) | [About](#) | [Services](#)



/* Rutas dinámicas */

¿Qué son las rutas dinámicas?



- Las rutas dinámicas permiten mostrar contenido en función de las circunstancias que se estén demandando.
- Estas circunstancias están determinadas por las acciones que realizan los/as visitantes del sitio, por ejemplo, al dar clic en un botón o un enlace.
- Supongamos que tenemos un blog, este está compuesto por entradas o publicaciones. Dichas entradas pueden ser mostradas en específico a partir de la captura dinámica de parámetros en la url.

Veamos cómo implementarlo en la aplicación.

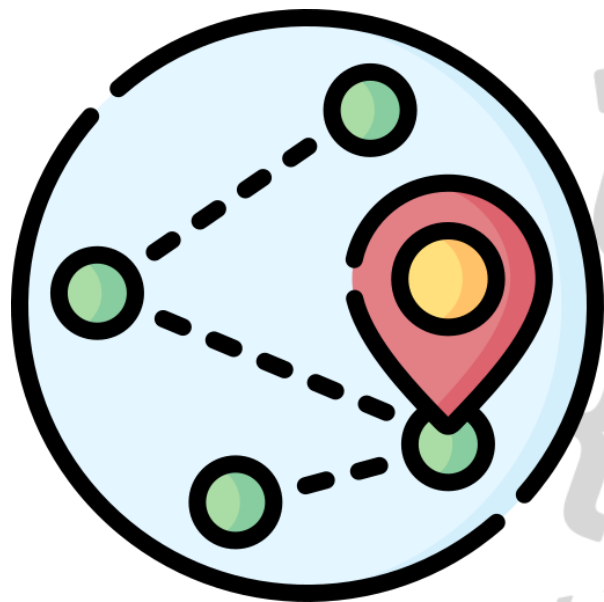
Ejercicio guiado: "Agregando rutas dinámicas"



Contexto

Sigamos trabajando en la misma aplicación anterior, en esta ocasión vamos a intencionar el uso de rutas dinámicas para mostrar las entradas que posea una sección llamada blog.

¡Sigue los pasos a continuación!



Sigue los pasos...

- **Paso 4:** Generamos una nueva vista para el blog que mostrará las entradas o publicaciones del mismo, lo llamaremos `BlogView.vue`.

```
<template>
  <div>
    <h1> {{ tituloEntrada }} </h1>
    <router-link :to="paginar(false)">Anterior</router-link> |
    <router-link :to="paginar(true)">Siguiete</router-link>
  </div>
</template>
```

Sigue los pasos...

- **Paso 5:** Creamos un método y una propiedad computada para la lógica de nuestra vista.

```
export default {  
  methods: {  
    paginar(siguiete) {  
      let entrada = parseInt(this.$route.params.entrada)  
      if (siguiete)  
        return entrada < 10 ? `/blog/${entrada + 1}` : `/blog/${entrada}`  
      else  
        return entrada > 1 ? `/blog/${entrada - 1}` : `/blog/${entrada}`  
    }  
  },  
  computed: {  
    tituloEntrada() {  
      return `Entrada ${this.$route.params.entrada} del Blog`  
    }  
  },  
}
```



Sigue los pasos...

- **Paso 6:** acto seguido, como lo hicimos con la ruta estática que agregamos, incluimos el componente recién creado en el archivo `index.js`, realizando la importación del mismo junto con la implementación de la ruta dentro del objeto JSON.

```
{  
  path: '/blog:entrada',  
  component: () => import(/* webpackChunkName: "BlogView" */ '../views/BlogView.vue')  
}
```

Nótese la sintaxis de dos puntos antes del nombre de la ruta.

Resultado de ejecución



{desafío}
latam_

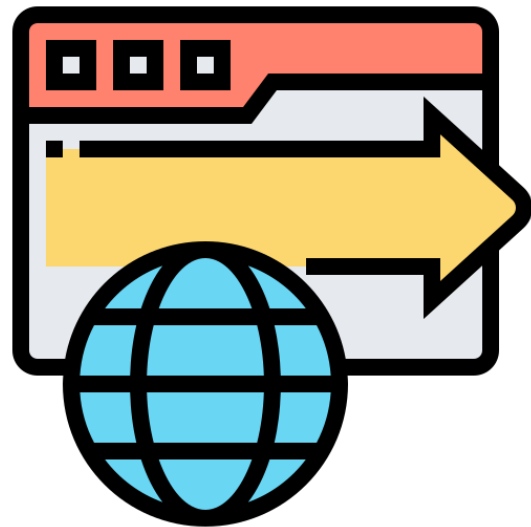


/* Redirecciones con el router */

¿Para qué sirven las redirecciones?

- Las redirecciones son comunes en los sitios web.
- Se utilizan para redirigir al visitante a partir de una acción determinada.
- Supongamos que tenemos un botón que al dar click sobre él, lleva al visitante a la página de inicio.

Sigamos con la aplicación anterior e implementemos el sistema de redirección.



Sigue los pasos...

- **Paso 7:** Dentro del template de la vista BlogView agregamos el código HTML del botón y los estilos CSS.

```
<button  
  class="button-index"  
  @click="volverInicio">  
  Volver al inicio  
</button>
```

```
<style scoped>  
  1 reference  
  .button-index {  
    margin-top: 10px;  
    padding: 5px;  
    color: ■ ■ white;  
    background-color: ■ ■ #42b983;  
  }  
</style>
```

Sigue los pasos...

- **Paso 8:** Definimos el método `volverInicio()` para realizar la redirección.

```
volverInicio() {  
  return this.$router.push('/');  
}
```



Tarea:

**Implementa el proceso
de redirección a través
de un botón en las vistas
about y services**



Resumen

- En esta sesión revisamos la etiqueta `<router-link>` y su comportamiento similar a los anclas de HTML.
- Generamos rutas estáticas y dinámicas, esta última capturando el título de una entrada en una vista de blogs.
- Realizamos el proceso de redirección a través del método `.push()` que nos provee el router.

¿Qué fue lo que más les
dificultó de la sesión?





Próxima sesión...

- *Guía de ejercicios.*

{desafío}
latam_

*Academia de
talentos digitales*

