

Domos Patagonia Viva



Entrega final Desarrollo Front-end - Karina Ortiz.

Mi portafolio personal es un sitio web responsivo trabajado con Vue, se trata de un sitio de domos en la patagonia; el tema se eligió para unir mi experiencia y pasión por el turismo y la hotelería con mi nueva competencia en front-end.

Características:

- Maquetación con HTML y CSS
- Uso de Componentes Vue
- Programación con Javascript ES6
- Consumo, manejo de datos y estados
- Estructura de archivos y carpetas

1. Maquetación con HTML y CSS: Uso de Bootstrap y etiquetas.

Ejemplo de HTML y Bootstrap en src/views/HomeView.vue.

```
<section id="quienessomos" class="container">
  <div class="row">
    <h2 class="col">Quienes Somos</h2>
  </div>

  <div class="seccion_content row">
    <p class="col">Somos una empresa dedicada a ofrecer una experiencia única de alojamiento en domos sustentables en la hermosa tierra austral. Nos enorgullece proporcionar un ambiente ecológico y sostenible para nuestros huéspedes, mientras disfrutan de una estadía cómoda y tranquila en medio de la naturaleza. ¡Bienvenidos a nuestro paraíso patagónico!</p>
  </div>
```

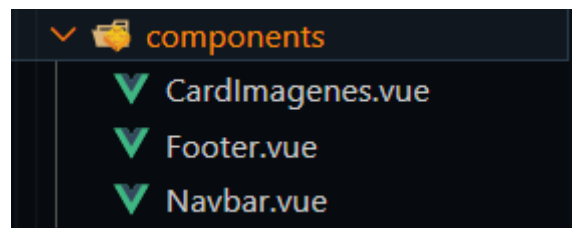
Este código representa la sección "Quienes Somos" del sitio web donde se ofrece alojamiento en domos sustentables, resaltando su enfoque ecológico y su ubicación en la región austral.

Responsividad por CSS usando media query en src/views/HomeView.vue.

```
@media screen and (min-width: 768px) {  
  1 reference  
  .hero_title {  
    font-size: 64px;  
  }  
}  
  
@media screen and (min-width: 992px) {  
  1 reference  
  .hero_title {  
    font-size: 72px;  
  }  
}
```

2. Uso de Componentes Vue

Uso de componentes en src/components y visualizado en carpetas:



Componentes:

NavBar: Este componente es la barra de navegación del sitio web.

CardImagenes: el componente representa un modal que muestra información sobre una ave, incluyendo su nombre en latín, una imagen y su hábitat. También emite un evento de cierre cuando el usuario hace clic en el modal.

Footer: Este componente es el pie de página del sitio.

Ejemplo de uso de ciclos de vida de un componente:

```

182     async created() {
183       try {
184         const promesa = await fetch("https://dog.ceo/api/breeds/image/random/" + this.cantDatos);
185         const promesaNombres = await fetch("https://randomuser.me/api/?results=" + this.cantDatos);
186         let data = await promesa.json();
187         let dataNombres = await promesaNombres.json();
188
189         console.log(dataNombres);
190         data.message.forEach((element, index) => {
191           let sexo = "";
192           let edad = Math.trunc(dataNombres.results[index].dob.age/12);
193           if(dataNombres.results[index].gender == "female"){
194             sexo = "Hembra";
195           }
196           else{
197             sexo = "Macho";
198           }
199           this.perros.push({"title": dataNombres.results[index].name.title, "nombre": dataNombres.results[index].name.first,
200             "img": element, "genero": sexo, "edad": edad});
201         });
202       } catch(error){
203         console.log(error)
204       }
205     }
206   }

```

Se utilizan estos métodos en un componente Vue.js para obtener y mostrar información detallada sobre aves mediante solicitudes a una API. El método "mostrarCard" se utiliza para mostrar la información de un ave específica, mientras que el método "cerrarCard" se utiliza para ocultar la tarjeta de información. El hook "created" se utiliza para obtener una lista de aves desde la API cuando el componente se crea.

Ejemplo de utilización de buenas prácticas en la definición de rutas:

```

const routes = [
  {
    path: '/',
    name: 'home',
    component: HomeView
  },
  {
    path: '/souvenirs',
    name: 'souvenirs',
    component: SouvenirsView
  },
  {
    path: '/domos',
    name: 'domos',
    component: DomosView
  },
  {
    path: '/aves',
    name: 'aves',
    component: FaunaView,
    children: [
      {
        path: ':id',
        name: 'cardimagenes',
        component: CardImagenes
      }
    ]
  }
]

```

Rutas del proyecto:

La **primera ruta** es para la página de **inicio** ("/") y se asocia al componente HomeView.

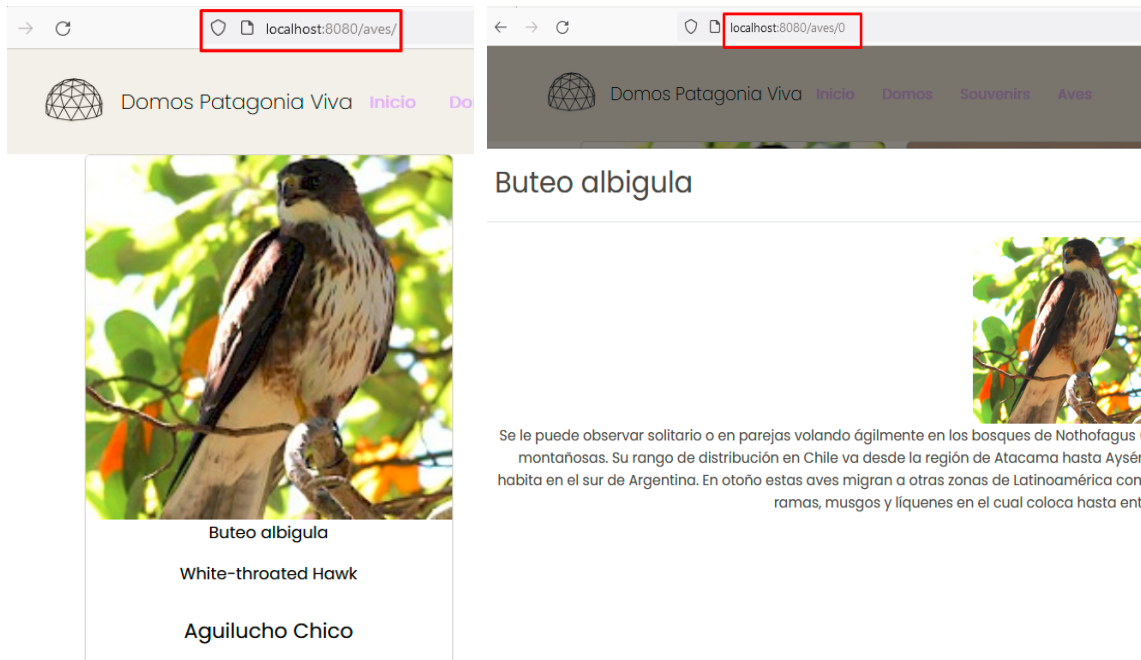
La **segunda ruta** es para la vista de **Souvenirs** ("/SouvenirsView") y se asocia al componente SouvenirsView.

La **tercera ruta** es para la vista de **domos** ("/DomosView") y se asocia al componente DomosView.

La **cuarta ruta** es para la vista de **aves** ("/FaunaView") y se asocia al componente FaunaView. También incluye una ruta secundaria para ver un ave específica.

La **última ruta** es una ruta de comodín ("/*") que se utiliza para manejar las **rutas no encontradas** y se asocia al componente NotFoundView.

Ejemplo de URL variable al hacer click en alguna ave:



3. Programación con JavaScript

El hook "created" se utiliza para simular la obtención de datos desde un archivo y asignarlos a una variable, mientras que el método "validarDatos" se utiliza para realizar validaciones en un formulario y mostrar alertas según corresponda.

```
created: async function () {
  try {
    this.cargando = true;
    // Simulación de obtención de datos desde el archivo Doms.js
    this.doms = Doms.getAllDoms();
    console.log(this.doms);
    this.cargando = false;
  } catch (error) {
    console.error(error);
  }
},
methods: {
  validarDatos() {
    if (!this.nombre){
      // console.log("faltatexto");
      alert("Falta el nombre");
    }
    else if (!this.mail){
      // console.log("faltatexto");
      alert("Falta el correo");
    }
  }
}
```

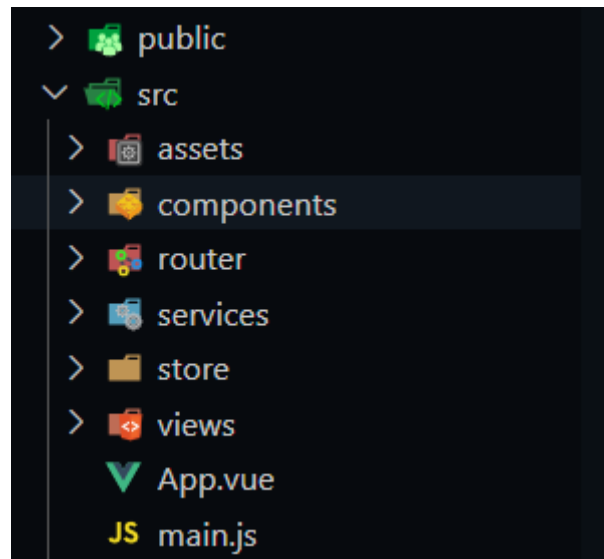
4. Consumo de datos

Consumo de **API** (<https://aves.ninjas.cl/api/birds>) para obtener la información de un ave.

```
methods: {
  async mostrarCard(ave) {
    try {
      const response = await axios.get(`https://aves.ninjas.cl/api/birds/${ave.uid}`);
      this.cardSelected = response.data;
    } catch (error) {
      console.error(error);
    }
  },
  cerrarCard() {
    this.cardSelected = null;
  }
}
```

Estos métodos y el hook se utilizan en un componente Vue.js para obtener y mostrar información detallada sobre aves utilizando solicitudes a una API. El método "mostrarCard" se utiliza para mostrar la información de un ave específica, el método "cerrarCard" se utiliza para ocultar la tarjeta de información, y el hook "created" se utiliza para obtener una lista de aves desde la API cuando el componente se crea.

5. Estructura de archivos y carpetas



Link a proyecto: https://github.com/krynortz91/proyecto_demos