



INDEX:

Pàgina principal o landing page	3
Template:.....	3
Script:.....	3
Style:.....	4
Pagina secundaria (API)	4
Template:.....	4
Script:.....	5
Style:.....	6
Router	7

VUETIFY:

Primerament he creat un projecte vue amb les següents comandes.

```
→ alumne npm init vue@latest

Vue.js - The Progressive JavaScript Framework

✓ Project name: ... VueAmineProj
✓ Package name: ... vueamineproj
✓ Add TypeScript? ... No / Yes
✓ Add JSX Support? ... No / Yes
✓ Add Vue Router for Single Page Application development? ... No / Yes
✓ Add Pinia for state management? ... No / Yes
✓ Add Vitest for Unit Testing? ... No / Yes
✓ Add an End-to-End Testing Solution? > No
✓ Add ESLint for code quality? ... No / Yes
✓ Add Vue DevTools 7 extension for debugging? (experimental) ... No / Yes

Scaffolding project in /home/alumne/VueAmineProj...

Done. Now run:

  cd VueAmineProj
  npm install
  npm run dev

→ VueAmineProj npm install
( ) : idealTree:VueAmineProj: timing idealTree:#root Completed in 1406ms
```

Per al meu projecte de vue, he triat el framework de VUETIFY per la seva simplesa.

Amb la comanda següent he instal·lat el framework.

```
→ VueAmineProj npm i vuetify@next @mdi/font

added 2 packages, and audited 31 packages in 3s

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
VueAmineProj
```

Pàgina principal o landing page

Template:

Dins del template de la pagina principal, basicament he afegit un div amb una imatge de fons a un “v-img” i bastants de gifs per a mostrar pokemons, també tinc un “if” per mostrar o no la imatge de fons segons si estem a la ruta indicada o no.

Al final tinc un “router-view” que apunta a “/pokemon” que es on tinc la api, més endavant mostraré les rutes.

```
<template> Show component usages
  <v-app>
    <div v-if="$route.path !== '/pokemon'" class="main-container">
      <v-img src="/public/poke.png" class="background-image"></v-img>
      
      
      
      
      
      
      <v-layout class="button-container" fill-height align-center justify-center>
        <v-btn class="explore-button" @click="buttonClicked">
          <v-icon left>mdi-pokeball</v-icon> Explore Pokémon
        </v-btn>
      </v-layout>
    </div>
    <router-view v-if="$route.path === '/pokemon'"></router-view>
  </v-app>
</template>
```

Script:

El script de la pàgina principal es bastant basic, ja que solament afegeix la imatge de fons i redirigeix a la pàgina secundaria.

```
<script>
import backgroundImage from '/public/poke.png';
Show usages
export default {
  data() {
    return {
      backgroundImage: backgroundImage, // Imatge fons
      showMainContent: true
    };
  },
  methods: {
    buttonClicked() {
      this.showMainContent = false;
      // Redirect a pagina secundaria
      this.$router.push('/pokemon');
    }
  }
};
</script>
```

Style:

Finalment tinc el style, que bàsicament afegeixo css als components que ho necessiten, mostrare alguns exemples ja que es un poc llarg.

```
.background-image {
  width: 100%;
  height: 100%;
  object-fit: cover;
  position: absolute;
  top: 0;
  left: 0;
}

.pika {
  position: absolute;
  top: 55%;
  left: 35%;
  transform: translate(-50%, -50%);
  z-index: 1; /* Coloco el gif sobre la imatge de fons */
}
```

Pagina secundaria (API)

Template:

Aquí la template és més llarga i complexa, ja que tinc que carregar les dades de la api i mostrar-les, tambe a tindre en compte que una vegada els pokemons s'han carregat, utilitzo un “modal” per desplegar detalls de cada pokemon al fer click sobre el mateix.

```
<template> Show component usages
<!-- Llista de Pokémon... -->
<div class="background">
  <h1 class="titul">Llista de Pokémon</h1>
  <div v-if="loading">Cargant...</div>
  <div v-else class="pokemon-list">
    <div v-for="pokemon in pokemons" :key="pokemon.name" class="pokemon-item" @click="openModal(pokemon)">
      <div class="pokemon-image">
        
      </div>
      <div class="pokemon-name">{{ pokemon.name }}</div>
    </div>
  </div>
<!-- Modal para els detalls del Pokémon -->
<div v-if="selectedPokemon">
  <v-dialog v-model="modalOpen">
    <div class="modal" :style="{ backgroundColor: getModalHeaderColor(selectedPokemon.name) }">
      <div class="modal-header">
        <v-card-title class="modal-title">{{ selectedPokemon.name }}</v-card-title>
      </div>
      <div class="modal-body">
        <v-img class="modal-image" :src="selectedPokemon.image" :alt="selectedPokemon.name" style="width: 700px; height: 300px;"></v-img>
        <div>Altura: {{ selectedPokemon.height }}</div>
        <div>Pes: {{ selectedPokemon.weight }}</div>
        <div>Habilitats:</div>
        <ul class="modal-abilities">
          <li v-for="(ability, index) in selectedPokemon.abilities" :key="index">{{ ability }}</li>
        </ul>
      </div>
      <div class="modal-footer">
        <v-btn @click="closeModal">Tancar</v-btn>
      </div>
    </v-dialog>
  </div>
</div>
</template>
```

Script:

El script també és més complex que el de la landing page, ja que amb axios realitza un fetch a la api de pokemon i almacena detalls del mateix i la seva imatge per més endavant al fer click sobre un es puguin mostrar aquests detalls amb el “modal”.

```
<script>
import axios from 'axios';

Show usages
export default {
  data() {
    return {
      loading: true,
      pokemons: [],
      selectedPokemon: null,
      modalOpen: false
    };
  },
  async created() {
    try {
      const response = await axios.get( url: 'https://pokeapi.co/api/v2/pokemon?limit=27');
      this.pokemons = response.data.results.map(pokemon => ({
        name: pokemon.name,
        url: pokemon.url,
        image: `https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/${this.getPokemonIdFromUrl(pokemon.url)}.png`
      }));
      this.loading = false;
    } catch (error) {
      console.error('Error al obtener los Pokémon:', error);
      this.loading = false;
    }
  },

```

```
  methods: {
    async openModal(pokemon) {
      try {
        const response = await axios.get(pokemon.url);
        this.selectedPokemon = {
          name: response.data.name,
          height: response.data.height,
          weight: response.data.weight,
          abilities: response.data.abilities.map(ability => ability.ability.name),
          image: `https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/${this.getPokemonIdFromUrl(pokemon.url)}.png`
          // Agrega más detalles del Pokémon según lo necesites
        };
        this.modalOpen = true; // Open the modal
      } catch (error) {
        console.error('Error al obtener los detalles del Pokémon:', error);
      }
    },
    closeModal() {
      this.modalOpen = false; // Close the modal
    },
    getPokemonIdFromUrl(url) {
      const parts = url.split('/');
      return parts[parts.length - 2];
    },
    getModalHeaderColor(pokemonName) {
      // Example function to calculate color based on Pokémon name
      const hash = pokemonName.split('').reduce((acc, char) => char.charCodeAt(0) + acc, 0);
      const hue = hash % 360;
      return `hsl(${hue}, 70%, 50%)`;
    }
  }
};
</script>
```

Style:

Aquí el style també és bastant llarg, aquí mostro uns exemples.

```
.modal {  
  max-width: 400px;  
  margin: 0 auto;  
  border-radius: 10px;  
  overflow: hidden;  
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}
```

```
.pokemon-list {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));  
  grid-gap: 20px;  
  justify-items: center;  
  margin-top: 120px;  
}
```

Router

Al router bàsicament, importem els components que formen part del nostre projecte i al apartat “routes” indiquem un “path”, “name” i el component al que apunta la ruta en qüestió, per a que més endavant al components amb “router.push” poder-se desplaçar entre components.

```
import { createRouter, createWebHistory } from 'vue-router';  
import APP from '../App.vue';  
import list from "../components/list.vue";  
  
const router : Router = createRouter({ options: {  
  history: createWebHistory(import.meta.env.BASE_URL),  
  routes: [  
    {  
      path: '/',  
      name: 'app',  
      component: APP  
    },  
    {  
      path: '/pokemon',  
      name: 'PokemonList',  
      component: list,  
      props: true  
    },  
  ],  
});  
  
Show usages  
export default router;
```