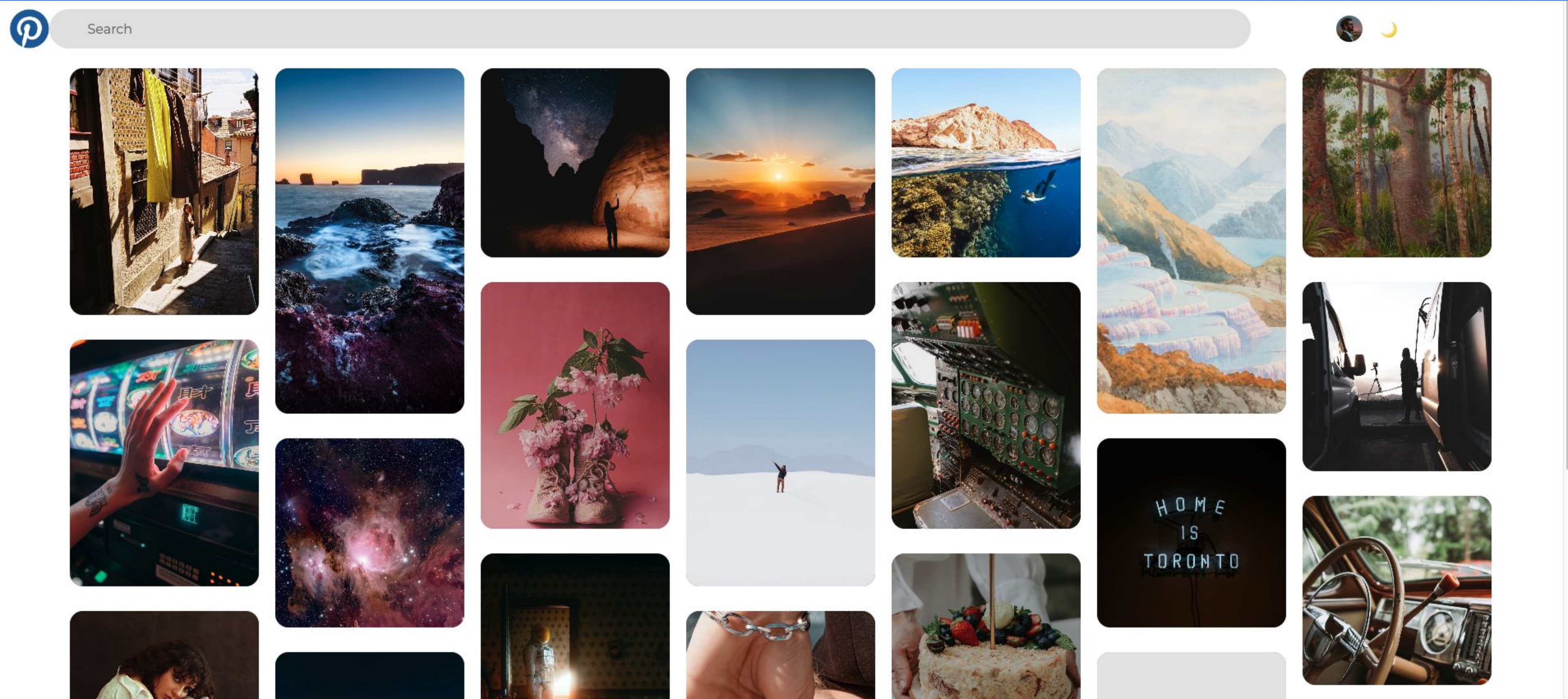
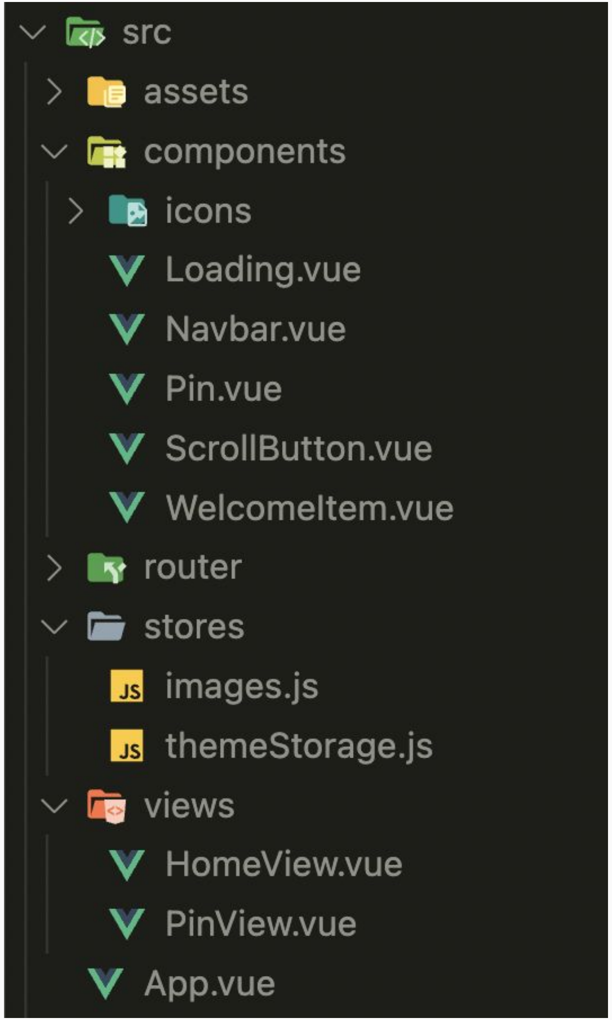
PUCterest

Grupo 12

Vue + The Unsplash API



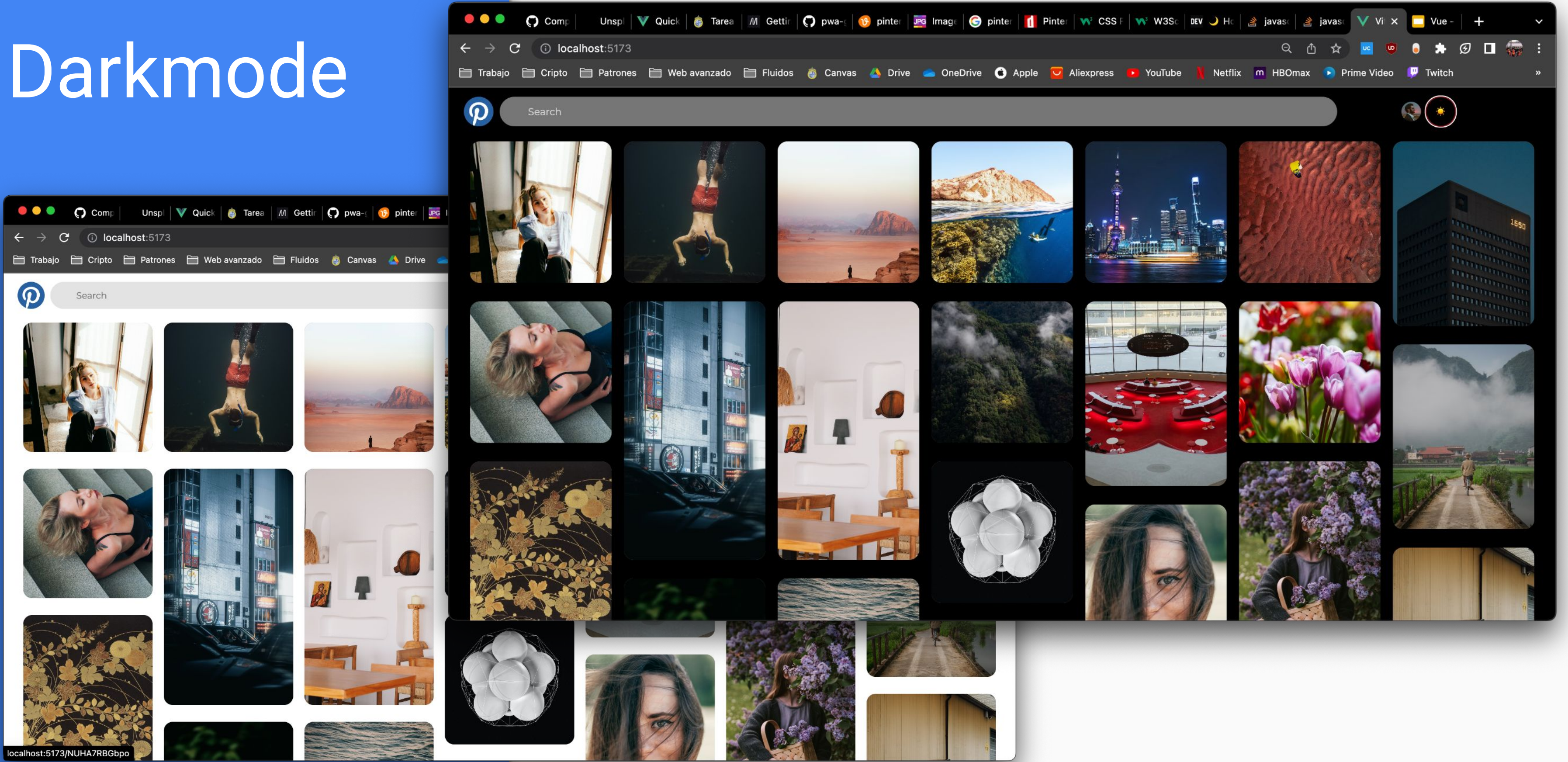
Estructura



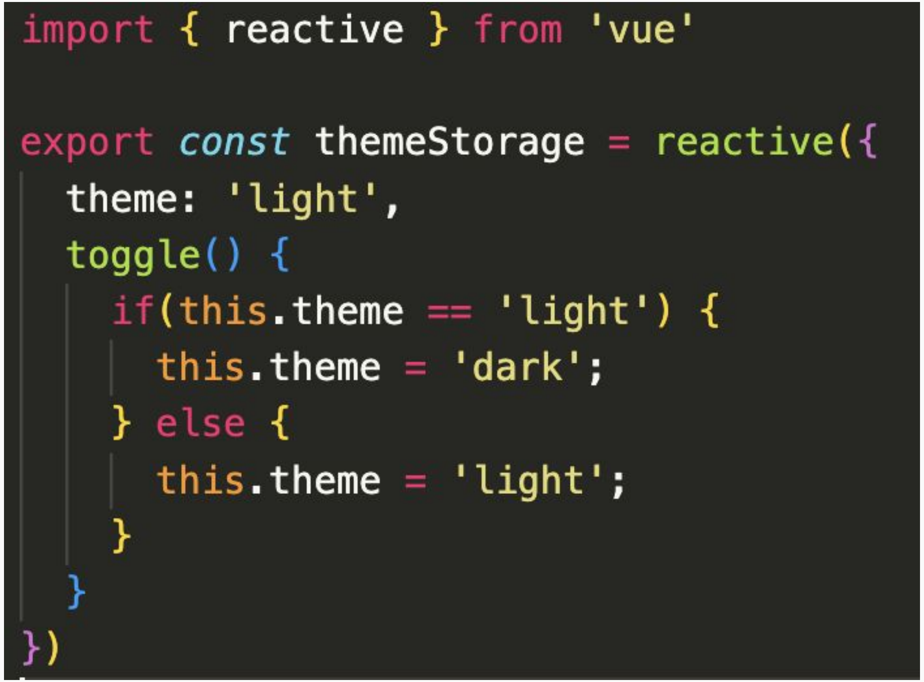
```
src
  assets
  components
    icons
    Loading.vue
    Navbar.vue
    Pin.vue
    ScrollButton.vue
    WelcomeItem.vue
  router
  stores
    images.js
    themeStorage.js
  views
    HomeView.vue
    PinView.vue
  App.vue
```

Características

Darkmode

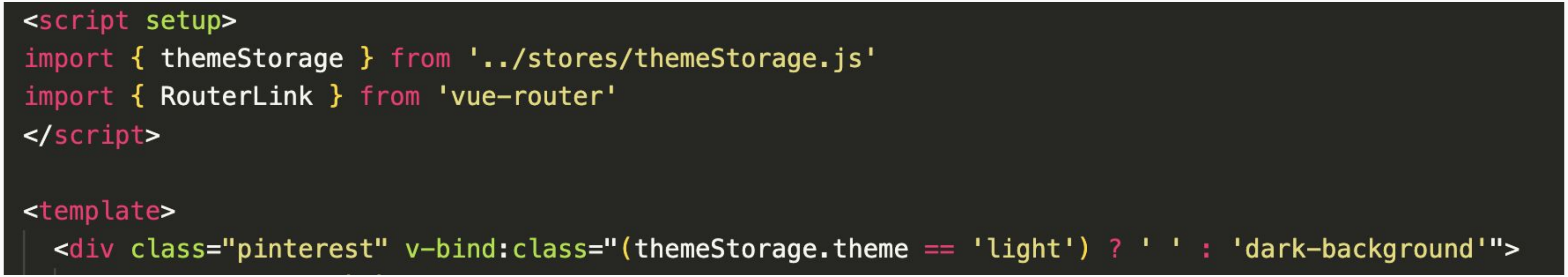

localhost:5173
Trabajo
Cripto
Patrones
Web avanzado
Fluidos
Canvas
Drive
OneDrive
Apple
Aliexpress
YouTube
Netflix
HBOmax
Prime Video
Twitch
Search
localhost:5173/NUHA7RBGbpo

themeStorage.js



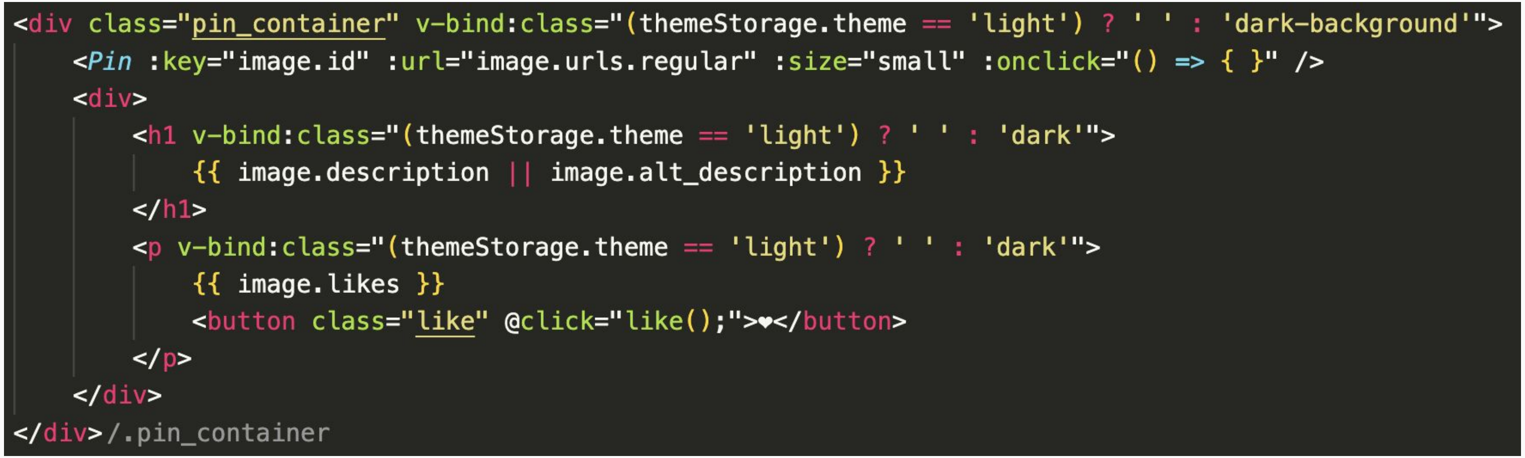
```
import { reactive } from 'vue'

export const themeStorage = reactive({
  theme: 'light',
  toggle() {
    if(this.theme == 'light') {
      this.theme = 'dark';
    } else {
      this.theme = 'light';
    }
  }
})
```

NavBar.vue



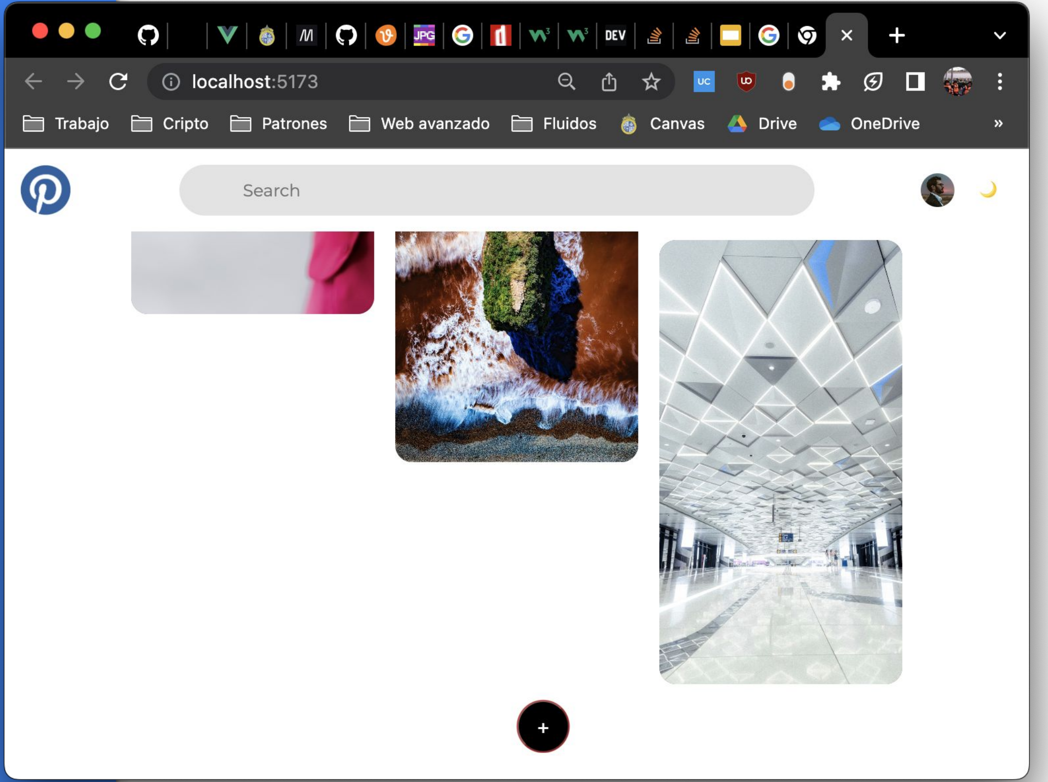
```
<script setup>
import { themeStorage } from '../stores/themeStorage.js'
import { RouterLink } from 'vue-router'
</script>

<template>
  <div class="pinterest" v-bind:class="(themeStorage.theme == 'light') ? ' ' : 'dark-background'">
```

PinView.vue

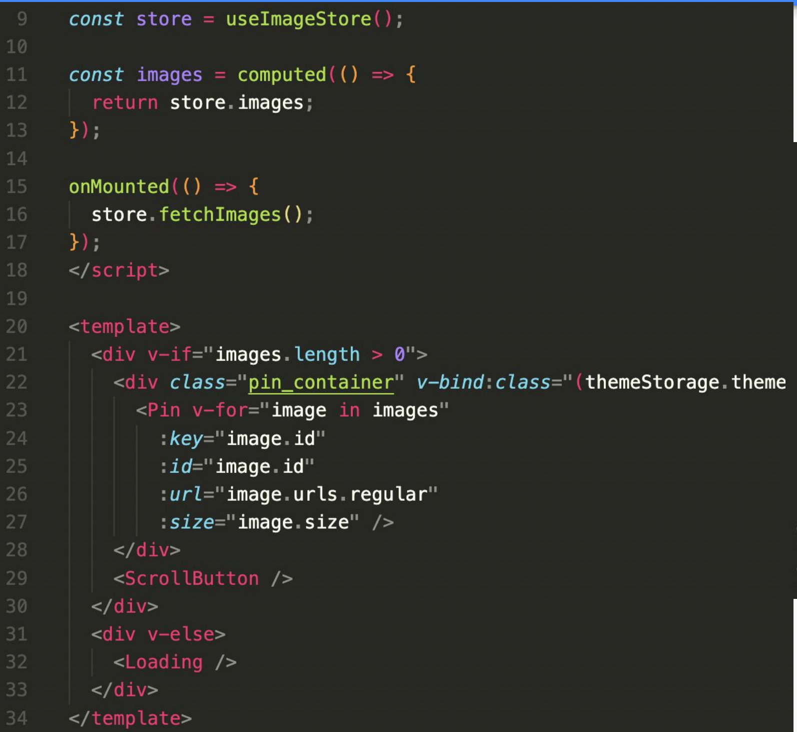


```
<div class="pin_container" v-bind:class="(themeStorage.theme == 'light') ? ' ' : 'dark-background'">
    <Pin :key="image.id" :url="image.urls.regular" :size="small" :onclick="() => { }" />
    <div>
        <h1 v-bind:class="(themeStorage.theme == 'light') ? ' ' : 'dark'">
            {{ image.description || image.alt_description }}
        </h1>
        <p v-bind:class="(themeStorage.theme == 'light') ? ' ' : 'dark'">
            {{ image.likes }}
            <button class="like" @click="like();">♥</button>
        </p>
    </div>
</div>/.pin_container
```

Load More

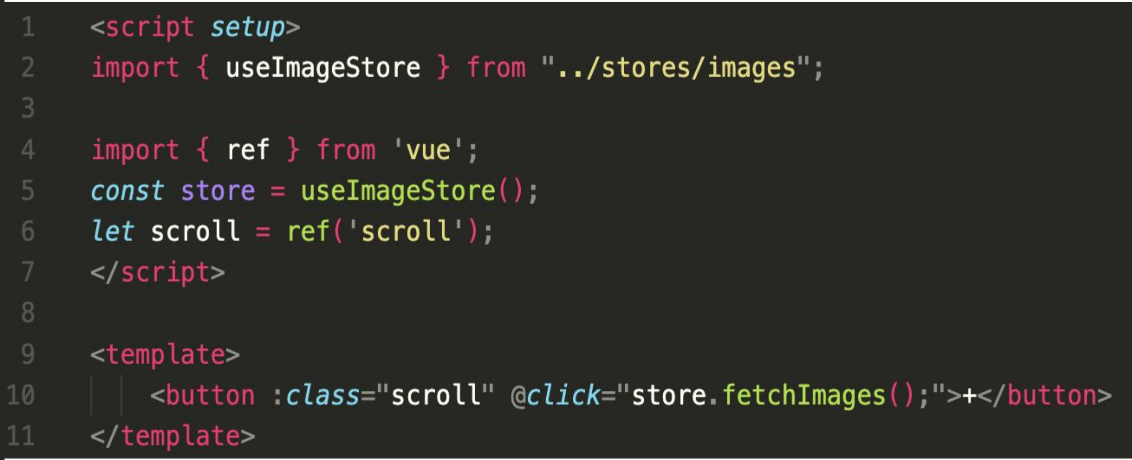

localhost:5173
Trabajo
Cripto
Patrones
Web avanzado
Fluidos
Canvas
Drive
OneDrive
Search
+

HomeView.vue



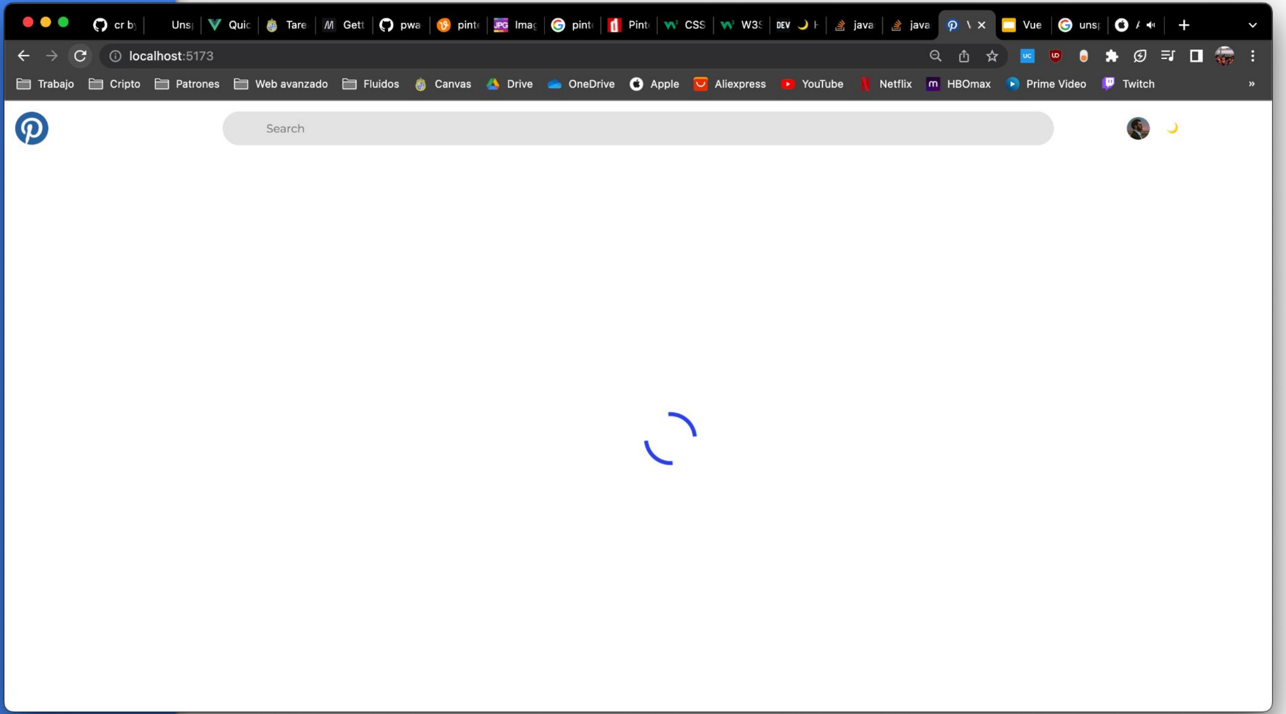
```
const store = useImageStore();

const images = computed(() => {
  return store.images;
});

onMounted(() => {
  store.fetchImages();
});
</script>

<template>
  <div v-if="images.length > 0">
    <div class="pin_container" v-bind:class="(themeStorage.theme
      <Pin v-for="image in images"
        :key="image.id"
        :id="image.id"
        :url="image.urls.regular"
        :size="image.size" />
    </div>
    <ScrollButton />
  </div>
  <div v-else>
    <Loading />
  </div>
</template>
```

ScrollButton.vue



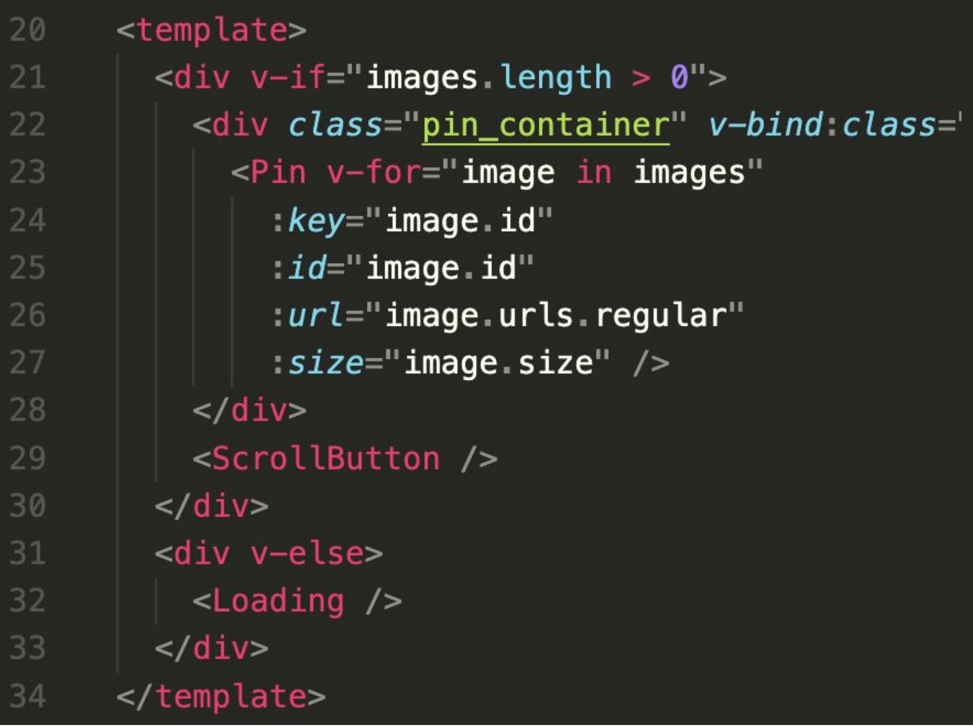
```
<script setup>
import { useImageStore } from "../stores/images";

import { ref } from 'vue';
const store = useImageStore();
let scroll = ref('scroll');
</script>

<template>
    <button :class="scroll" @click="store.fetchImages();">+</button>
</template>
```

Loading

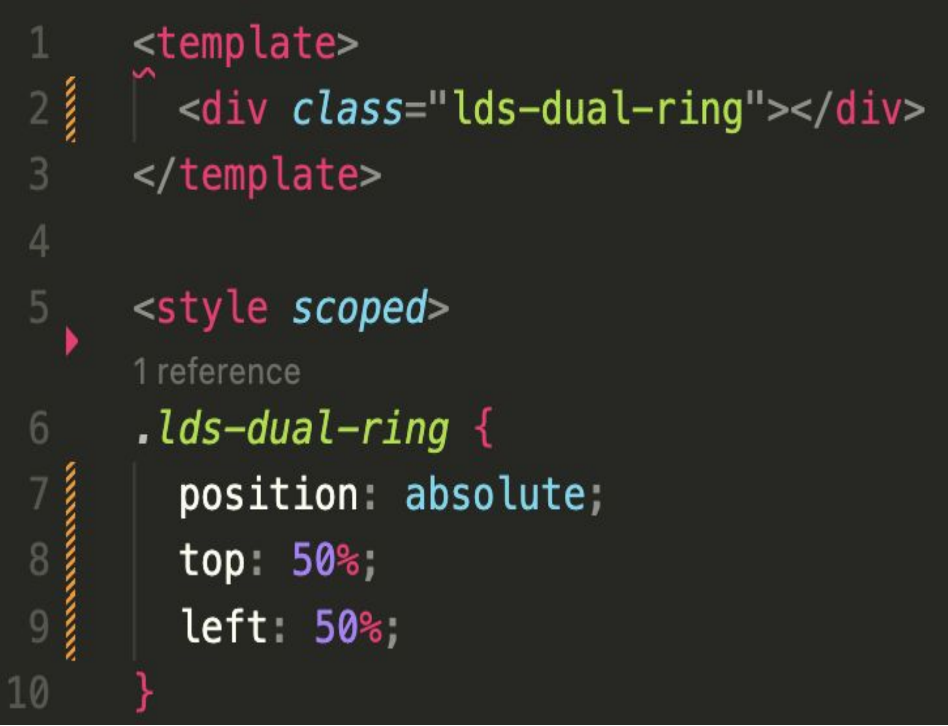

localhost:5173
Trabajo
Cripto
Patrones
Web avanzado
Fluidos
Canvas
Drive
OneDrive
Apple
Aliexpress
YouTube
Netflix
HBOmax
Prime Video
Twitch
Search

HomeView.vue

Loading.vue

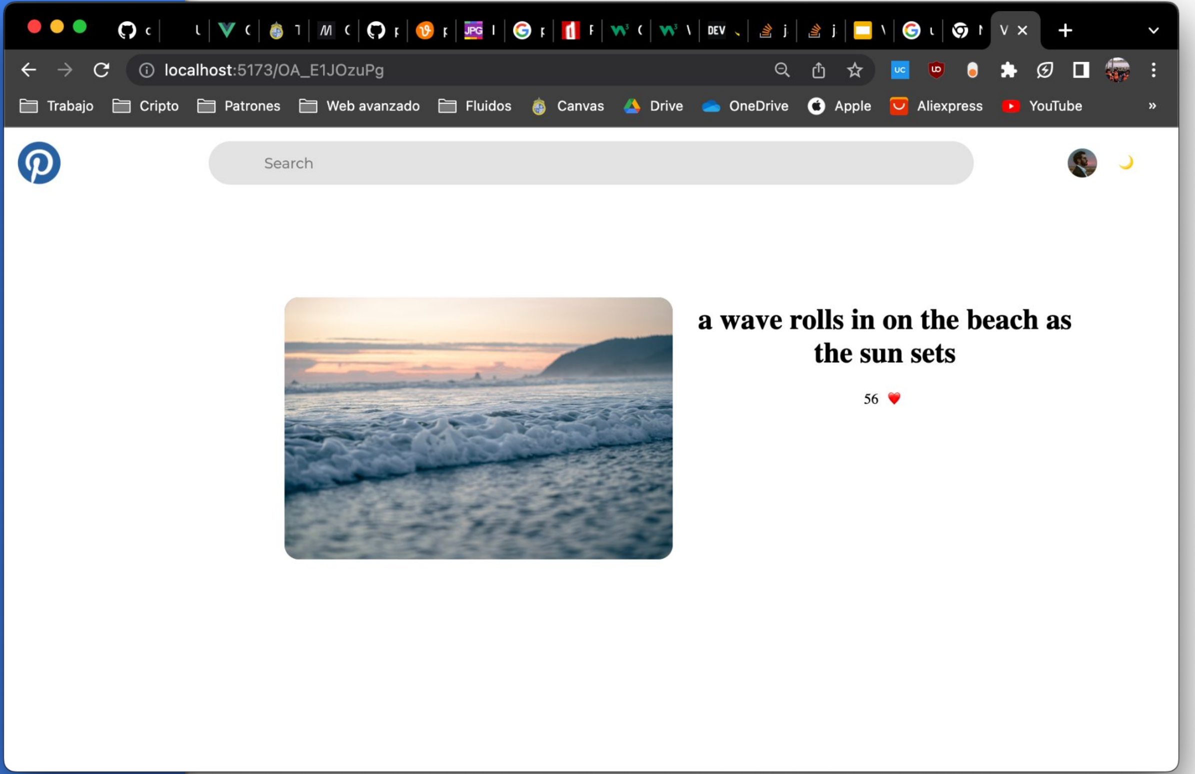


```
<template>
  <div v-if="images.length > 0">
    <div class="pin_container" v-bind:class='
      <Pin v-for="image in images"
        :key="image.id"
        :id="image.id"
        :url="image.urls.regular"
        :size="image.size" />
    </div>
    <ScrollButton />
  </div>
  <div v-else>
    <Loading />
  </div>
</template>
```



```
<template>
  <div class="lds-dual-ring"></div>
</template>

<style scoped>
1 reference
.lds-dual-ring {
  position: absolute;
  top: 50%;
  left: 50%;
}
```

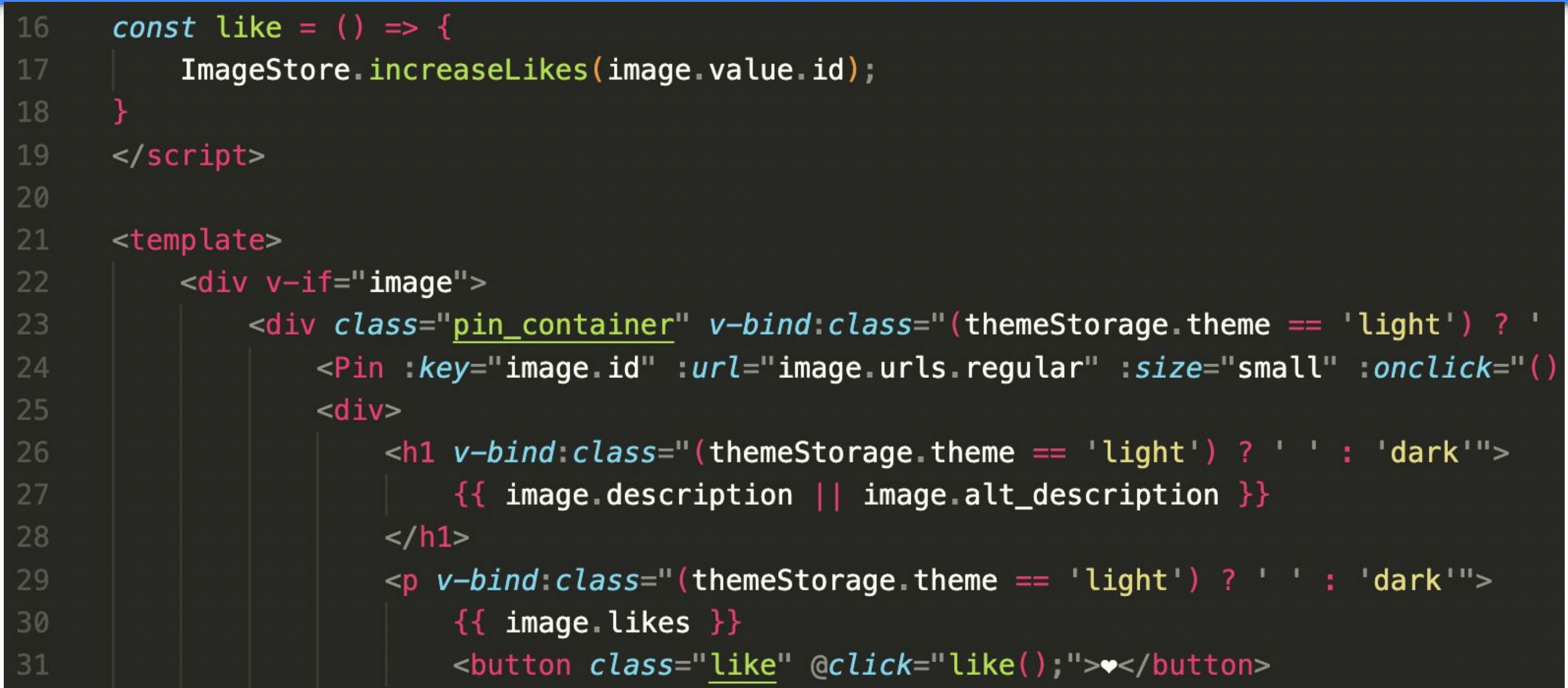
Likes



a wave rolls in on the beach as the sun sets

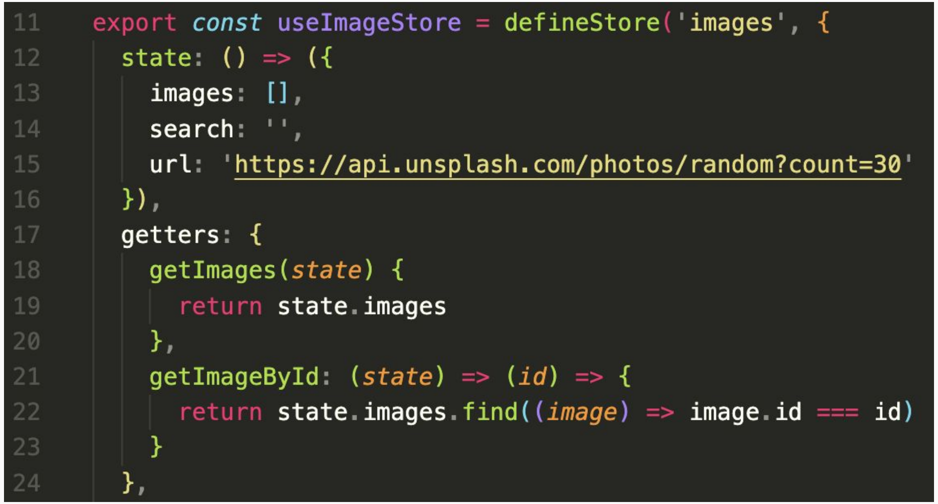
56 ❤

PinView.vue

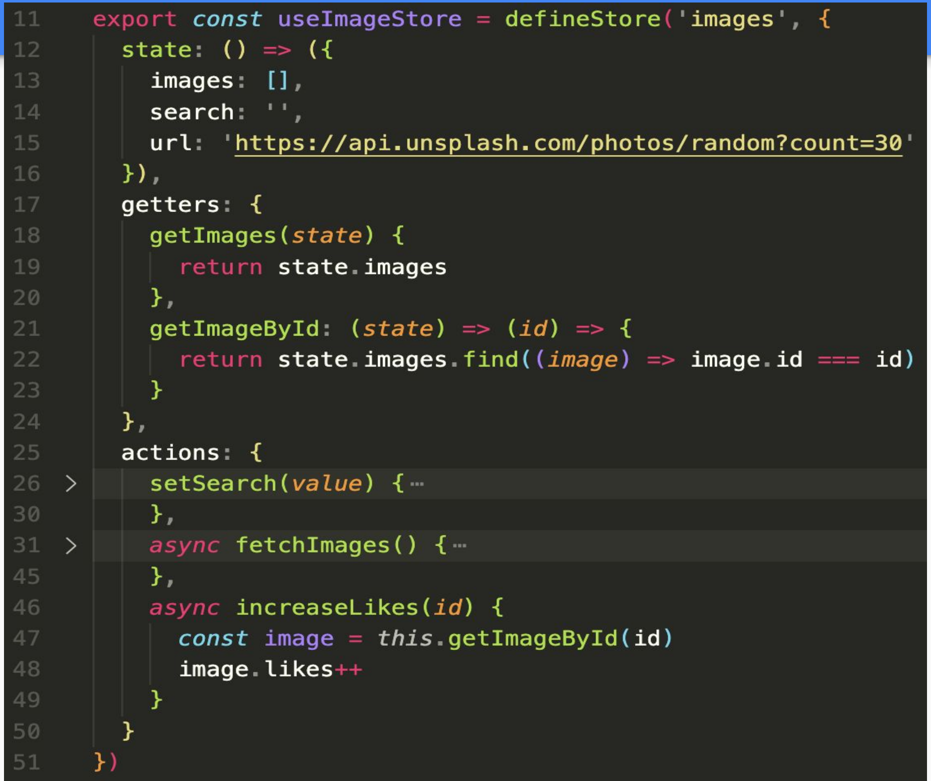


```
    const like = () => {
        ImageStore.increaseLikes(image.value.id);
    }
    </script>

    <template>
        <div v-if="image">
            <div class="pin_container" v-bind:class="(themeStorage.theme == 'light') ? '
                <Pin :key="image.id" :url="image.urls.regular" :size="small" :onclick="()
                <div>
                    <h1 v-bind:class="(themeStorage.theme == 'light') ? ' ' : 'dark'">
                        {{ image.description || image.alt_description }}
                    </h1>
                    <p v-bind:class="(themeStorage.theme == 'light') ? ' ' : 'dark'">
                        {{ image.likes }}
                        <button class="like" @click="like();">❤</button>
```

imagesStore.js

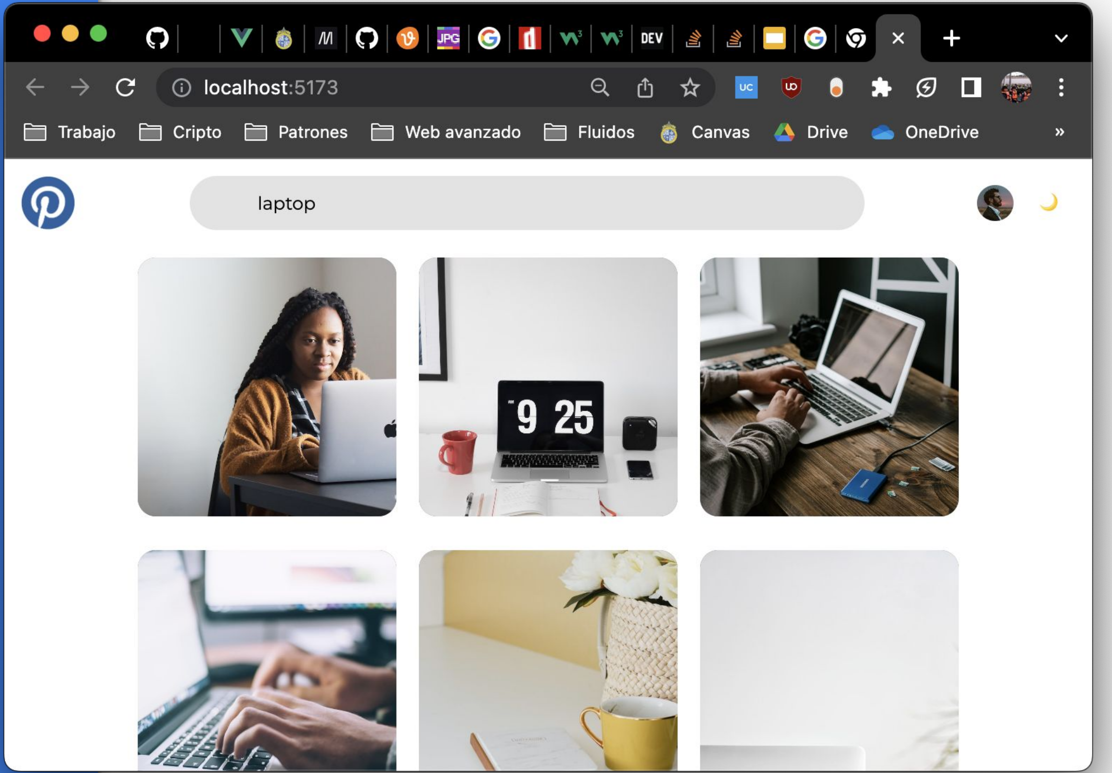


```javascript
export const useImageStore = defineStore('images', {
  state: () => ({
    images: [],
    search: '',
    url: 'https://api.unsplash.com/photos/random?count=30'
  }),
  getters: {
    getImages(state) {
      return state.images
    },
    getImageById: (state) => (id) => {
      return state.images.find((image) => image.id === id)
    }
  },
```



```javascript
export const useImageStore = defineStore('images', {
  state: () => ({
    images: [],
    search: '',
    url: 'https://api.unsplash.com/photos/random?count=30'
  }),
  getters: {
    getImages(state) {
      return state.images
    },
    getImageById: (state) => (id) => {
      return state.images.find((image) => image.id === id)
    }
  },
  actions: {
>   setSearch(value) {⋯
    },
>   async fetchImages() {⋯
    },
    async increaseLikes(id) {
      const image = this.getImageById(id)
      image.likes++
    }
  }
})
```

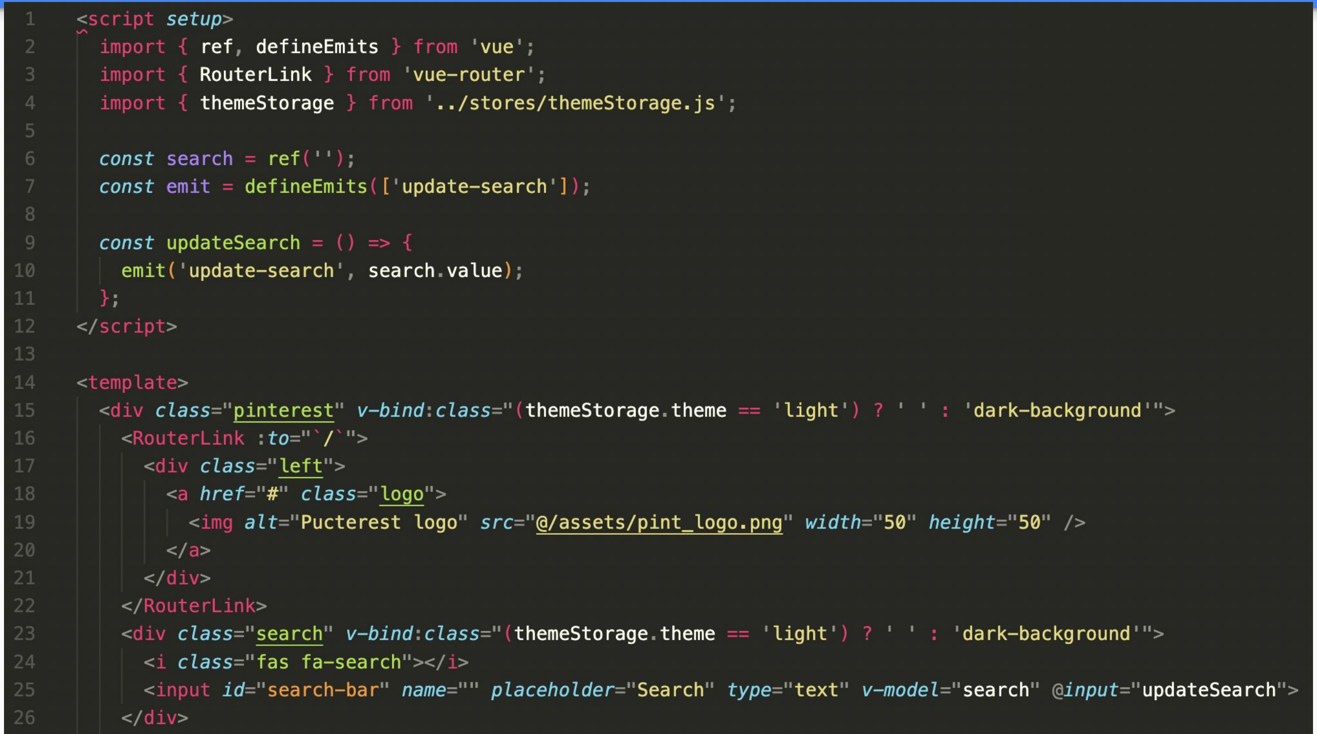
Búsqueda



localhost:5173

laptop

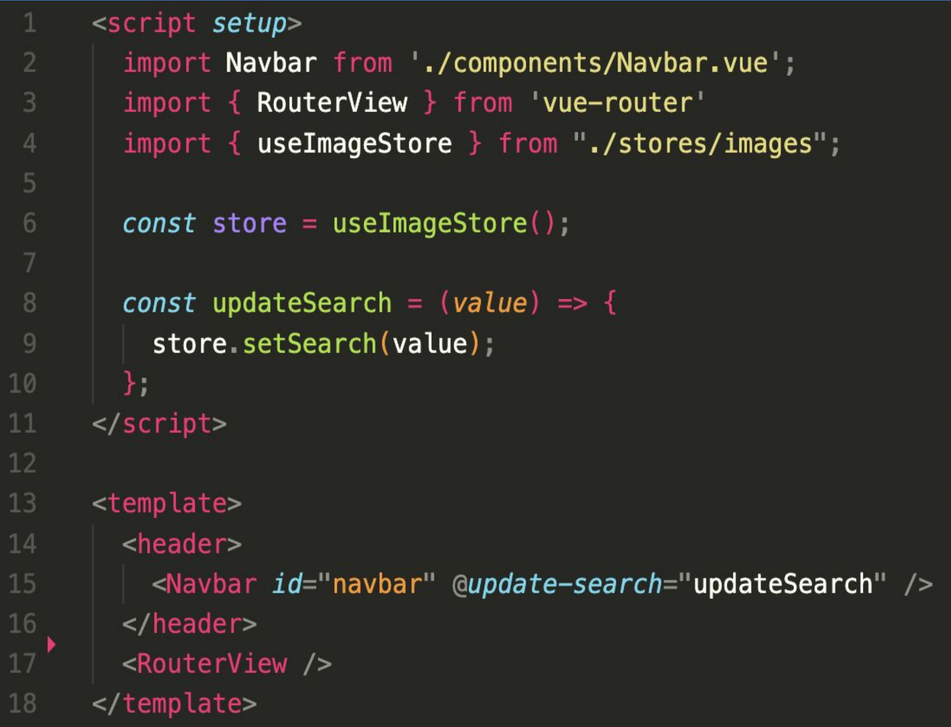
Navbar.vue



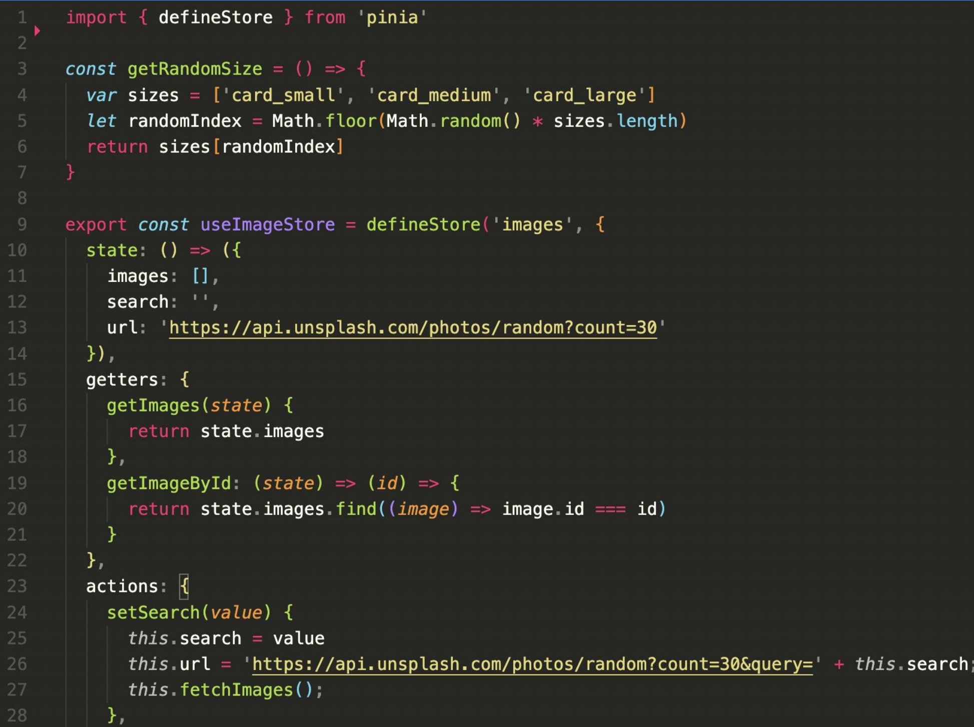
```
<script setup>
  import { ref, defineEmits } from 'vue';
  import { RouterLink } from 'vue-router';
  import { themeStorage } from '../stores/themeStorage.js';

  const search = ref('');
  const emit = defineEmits(['update-search']);

  const updateSearch = () => {
    emit('update-search', search.value);
  };
</script>

<template>
  <div class="pinterest" v-bind:class="(themeStorage.theme == 'light') ? ' ' : 'dark-background'">
    <RouterLink :to="`/`">
      <div class="left">
        <a href="#" class="logo">
          <img alt="Pucterest logo" src="@/assets/pint_logo.png" width="50" height="50" />
        </a>
      </div>
    </RouterLink>
    <div class="search" v-bind:class="(themeStorage.theme == 'light') ? ' ' : 'dark-background'">
      <i class="fas fa-search"></i>
      <input id="search-bar" name="" placeholder="Search" type="text" v-model="search" @input="updateSearch">
    </div>
```

App.vue

imagesStore.js

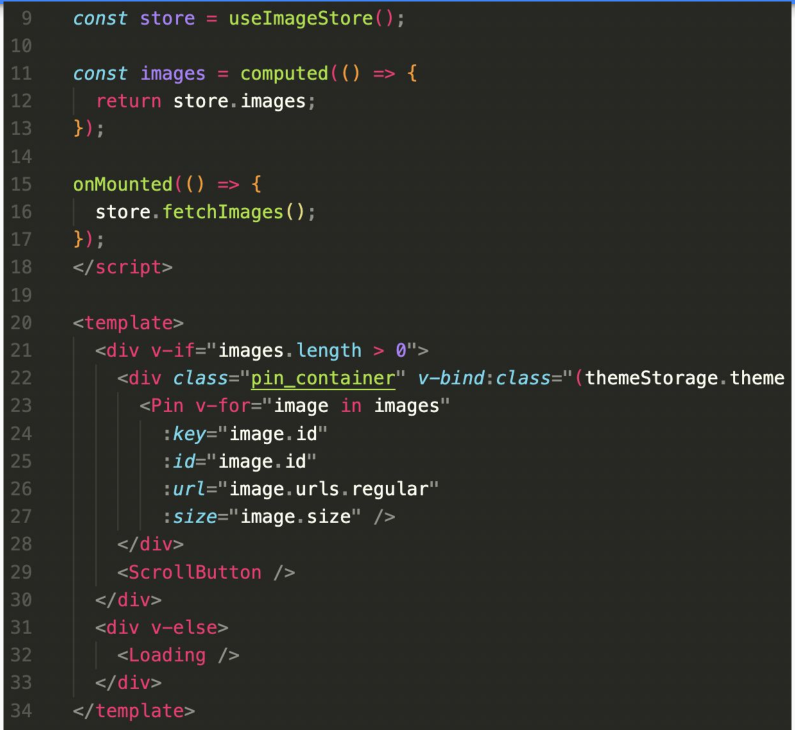


```
<script setup>
  import Navbar from './components/Navbar.vue';
  import { RouterView } from 'vue-router'
  import { useImageStore } from "./stores/images";

  const store = useImageStore();

  const updateSearch = (value) => {
    store.setSearch(value);
  };
</script>

<template>
  <header>
    <Navbar id="navbar" @update-search="updateSearch" />
  </header>
  <RouterView />
</template>
```



```
import { defineStore } from 'pinia'

const getRandomSize = () => {
  var sizes = ['card_small', 'card_medium', 'card_large']
  let randomIndex = Math.floor(Math.random() * sizes.length)
  return sizes[randomIndex]
}

export const useImageStore = defineStore('images', {
  state: () => ({
    images: [],
    search: '',
    url: 'https://api.unsplash.com/photos/random?count=30'
  }),
  getters: {
    getImages(state) {
      return state.images
    },
    getImageById: (state) => (id) => {
      return state.images.find((image) => image.id === id)
    }
  },
  actions: {
    setSearch(value) {
      this.search = value
      this.url = 'https://api.unsplash.com/photos/random?count=30&query=' + this.search;
      this.fetchImages();
    },
```

HomeView.vue



```
const store = useImageStore();

const images = computed(() => {
  return store.images;
});

onMounted(() => {
  store.fetchImages();
});
</script>

<template>
  <div v-if="images.length > 0">
    <div class="pin_container" v-bind:class="(themeStorage.theme
      <Pin v-for="image in images"
        :key="image.id"
        :id="image.id"
        :url="image.urls.regular"
        :size="image.size" />
    </div>
    <ScrollButton />
  </div>
  <div v-else>
    <Loading />
  </div>
</template>
```

PUCterest

Grupo 12