



VueJS

GRUPO 5

Nicolás Barría
Sebastián Carrasco
Juan Vargas



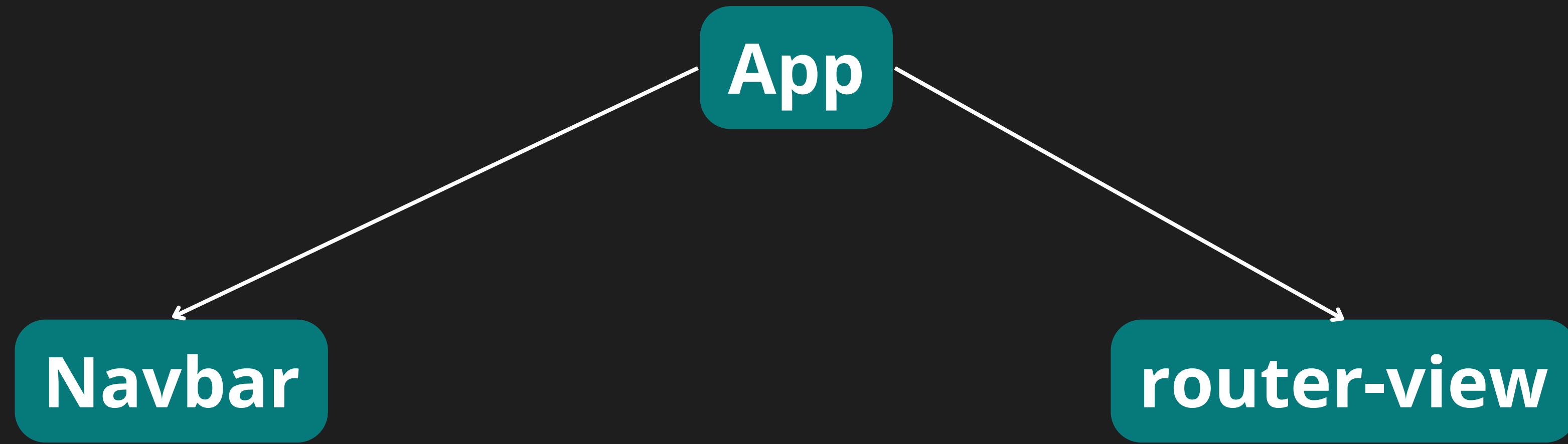
11:11 PM



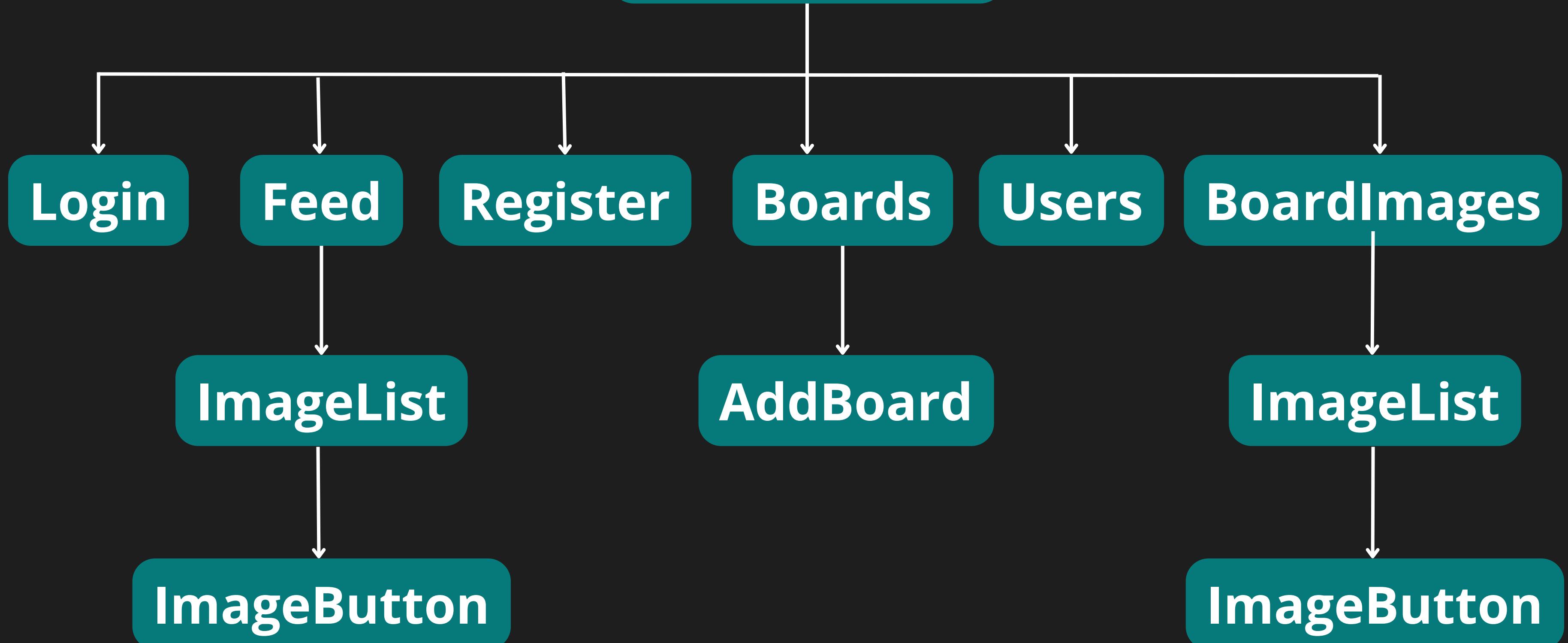
Demo

Herramientas utilizadas





router-view



Router



```
1 const routes = [
2   { path: "/", component: Feed },
3   { path: "/login", component: Login },
4   { path: "/register", component: Register },
5   { path: "/feed", component: Feed },
6   { path: "/users", component: Users },
7   { path: "/users/:userId/boards", component: Boards },
8   { path: "/boards/:boardId", component: BoardImages },
9   { path: "/:pathMatch(.*)*", redirect: "/" },
10 ];
11
12 const router = createRouter({
13   history: createWebHistory(),
14   routes,
15 });
16
17 export { router };
```

AuthStore



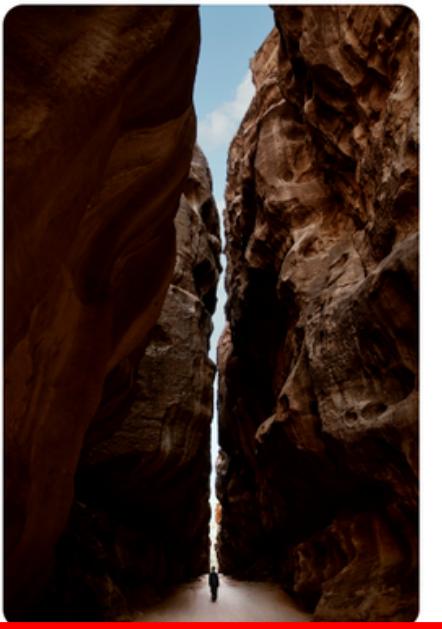
```
1  export const useAuthStore = defineStore("auth", () => {
2    const token = ref(localStorage.getItem("token"));
3    const id = ref(null);
4    const boards = ref([]);
5    if (localStorage.getItem("boards")) {
6      boards.value = JSON.parse(localStorage.getItem("boards"));
7    }
8    if (token.value) {
9      const { id: _id } = jwt_decode(token.value);
10     id.value = _id;
11   }
12
13   const login = async (email, password) => { ... };
14   const register = async (email, password) => { ... };
15   const logout = () => { ... };
16   const loadBoards = async () => { ... };
17
18   return { token, id, boards, login, register, logout, loadBoards };
19 });
```

Feed



```
1 <script setup>
2 const images = ref([]);
3 onMounted(async () => {
4   try {
5     const response = await fetch(
6       `${import.meta.env.VITE_API_URL}/photos?per_page=20&client_id=${import.meta.env.VITE_API_KEY}`,
7     {
8       method: "GET",
9       headers: {
10         "Content-Type": "application/json",
11       },
12     }
13   );
14   images.value = await response.json();
15 } catch (error) {
16   console.log(error);
17 }
18 });
19 </script>
20
21 <template>
22   <ImageList :images="images" />
23 </template>
```

ImageList

[INICIAR SESIÓN](#)[REGISTRARSE](#)

ImageList



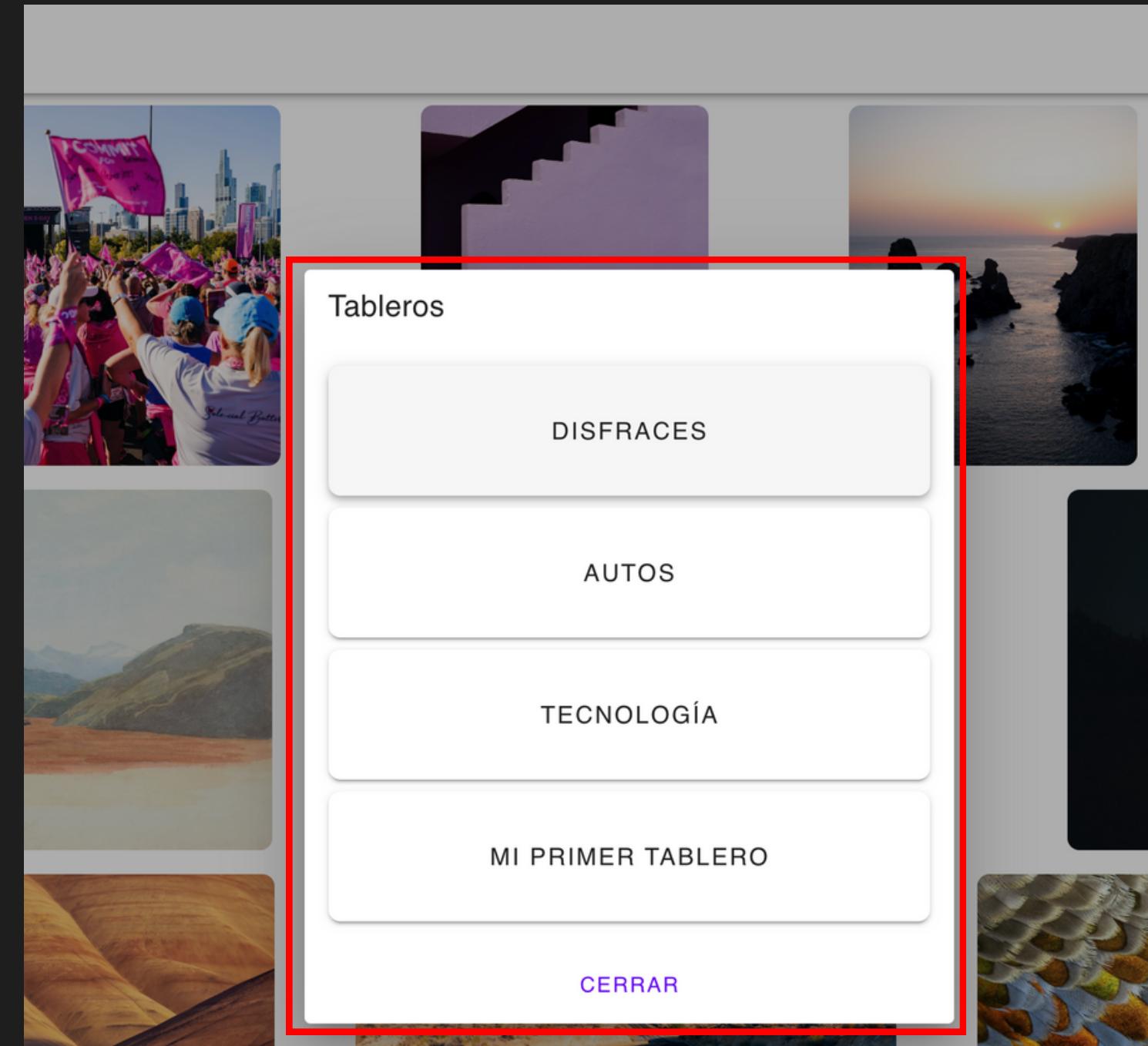
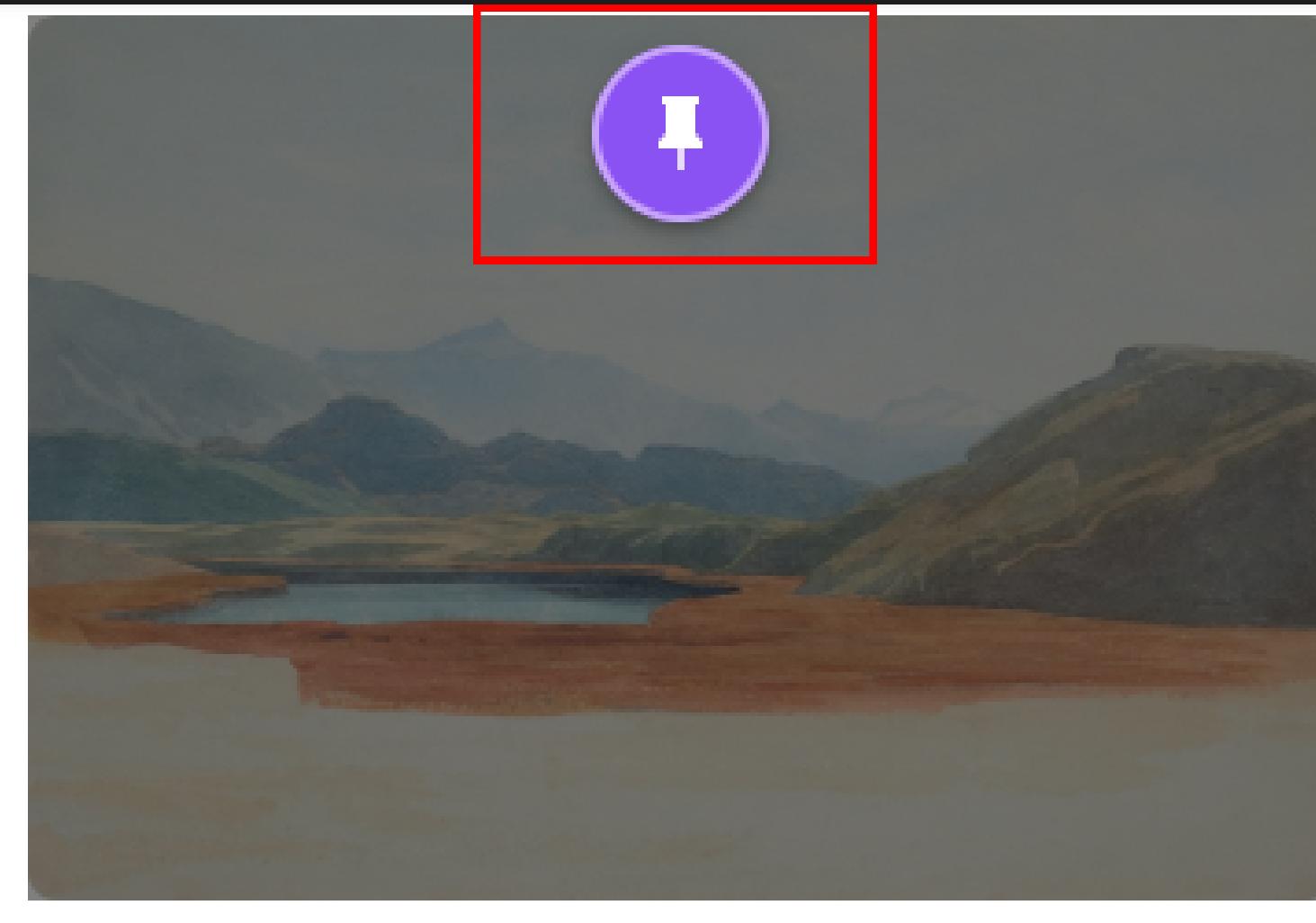
```
1 <script setup>
2 const props = defineProps({
3   images: {
4     type: Array,
5     required: true,
6   },
7 });
8 const authStore = useAuthStore();
9 const isLoggedIn = computed(() => !!authStore.token);
10 </script>
```

ImageList



```
1 <template>
2   <v-container
3     class="container"
4     d-flex
5     justify-space-between
6     align-self-center
7     flex-wrap overflow-auto"
8   >
9   <div v-for="image in props.images" :key="image.id">
10    <v-hover>
11      <template v-slot:default="{ props }">
12        <div
13          class="image-container rounded-lg"
14          v-bind="props"
15          ref="imageContainer"
16        >
17          <v-img :src="image.urls.regular" class="img rounded-lg"></v-img>
18          <div v-if="isLoggedIn" class="overlay">
19            <ImageButton :image="image" />
20          </div>
21        </div>
22      </template>
23    </v-hover>
24  </div>
25 </v-container>
26 </template>
```

ImageButton



ImageButton



```
1 <script setup>
2 const props = defineProps({
3   image: Object,
4 });
5 const dialog = ref(false);
6 const authStore = useAuthStore();
7 const boards = computed(() => authStore.boards);
8 const pb = new PocketBase('https://pocketbase-docker-production.up.railway.app');
9 const saveImage = async (boardId, image) => {
10   await pb.collection('boardspictures').create({
11     board: boardId,
12     image,
13   });
14   dialog.value = false;
15 };
16 </script>
```



ImageButton

```
1 <template>
2   <div class="text-center">
3     <v-dialog v-model="dialog">
4       <template v-slot:activator="{ props }">
5         <v-btn color="primary" v-bind="props" icon="mdi-pin" class="mt-2">
6           </v-btn>
7       </template>
8
9     <v-card>
10      <v-card-title class="headline">Tableros</v-card-title>
11      <v-container class="d-flex flex-wrap overflow-auto container">
12        <v-btn
13          v-for="board in boards"
14          :key="board.id"
15          size="large"
16          class="boardBtn rounded-lg m-auto"
17          @click="saveImage(board.id, image)"
18        >
19          {{ board.name }}
20        </v-btn>
21      </v-container>
22      <v-card-actions class="mt-auto">
23        <v-btn color="primary" block @click="dialog = false">Cerrar</v-btn>
24      </v-card-actions>
25    </v-card>
26  </v-dialog>
27 </div>
28 </template>
```

Users

The screenshot shows a user management interface with a dark background. At the top right are three icons: a house, a heart, and a person. Below them is a section titled "Usuarios" containing ten user profiles arranged in two rows of five. Each profile consists of a name in bold and a "VER TABLEROS" button below it. A red rectangular box highlights the first five profiles.

Usuarios									
Magdalena	Lucas	Catalina	Jose	Daniel					
VER TABLEROS									
Ignacio	Felipe	Antonia	Carolina	Fernanda					
VER TABLEROS									

Users

```
1 <script setup>
2 const pb = new PocketBase(import.meta.env.VITE_PB_URL);
3 const users = ref([]);
4 onMounted(async () => {
5   users.value = await pb.collection("users").getFullList({ sort: "-created" });
6 });
7 </script>
8
9 <template>
10   <h2 class="text-center">Usuarios</h2>
11   <v-container
12     class="container d-flex justify-space-between align-self-center flex-wrap overflow-auto"
13   >
14     <div v-for="user in users" class="d-flex justify-center align-center">
15       <v-card
16         class="user"
17         :key="user.id"
18       >
19         <v-card-title class="">{{ user.name }}</v-card-title>
20         <v-btn class="btn btn-primary w-100" :to=`/users/${user.id}/boards` :elevation="3">
21           Ver tableros
22         </v-btn>
23       </v-card>
24     </div>
25   </v-container>
26 </template>
```

Aspectos a mejorar

- Protección de vistas
- Encapsulamiento de componentes
- Limpieza de código



VueJS

GRUPO 5

Nicolás Barría
Sebastián Carrasco
Juan Vargas



11:11 PM