

Solución Prueba Ejemplo - Otto Klaus

Requerimiento 1

- **Paso 1:** Crear el entorno de desarrollo e instalar element-ui

```
vue create ottoklaus  
cd ottoklaus  
npm i -S element-ui
```

- **Paso 2:** Crear la vista Juguetes.vue

Juguetes.vue

```
<template>  
  <div class="juguetes">  
    <el-row>  
      <el-col><el-link @click="cerrarSesion">Cerrar  
sesión</el-link></el-col>  
      <el-col><h1>Bienvenido a Otto Klaus</h1></el-col>  
    </el-row>  
  
    <el-row>  
      <el-button  
        type="primary"  
        icon="el-icon-plus"  
        circle  
        title="Agregar Juguete"  
        @click="agregar"  
      ></el-button>  
    </el-row>  
    <div>  
      <el-table :data="juguetesConStock" lazy>  
        <el-table-column prop="data.codigo"  
label="Código"></el-table-column>  
        <el-table-column prop="data.nombre"  
label="Nombre"></el-table-column>  
        <el-table-column prop="data.stock"  
label="Stock"></el-table-column>
```

```
      <el-table-column prop="data.precio"
label="Precio"></el-table-column>
      <el-table-column label="Acciones">
        <template slot-scope="scope">
          <el-button
            @click="editar(juguetesConStock[scope.$index])"
            type="text"
            size="small"
          >Editar</el-button>
          <br>
          <el-button
            @click="dialogEliminar(juguetesConStock[scope.$index].id)"
            type="text"
            size="small"
          >Eliminar
          </el-button>
        </template>
      </el-table-column>
    </el-table>
  </div>
</div>
</template>

<script>
import { mapGetters, mapActions } from "vuex";
import Firebase from "firebase";

export default {
  data() {
    return {};
  },
  mounted() {
    this.getJuguetes();
  },
  computed: {
    ...mapGetters(["juguetesConStock"]),
  },
  methods: {
    ...mapActions(["getJuguetes", "eliminarJuguete"]),
    agregar() {
      this.$router.push({ name: "agregar" });
    },
    editar(juguete) {
      this.$router.push({ name: "editar", params: { juguete: juguete }

```

```
});
},
dialogEliminar(jugueteID) {
  this.$confirm(
    "Borrará el juguete seleccionado. ¿Desea continuar?",
    "Warning",
    {
      confirmButtonText: "Eliminar",
      cancelButtonText: "Cancelar",
      type: "warning",
    }
  )
  .then(async () => {
    await this.eliminarJuguete(jugueteID);
    let options = { type: "success", message: "Juguete eliminado"
};

    this.$message({
      ...options,
    });
  })
  .catch(() => {
    this.$message({
      type: "info",
      message: "Acción cancelada",
    });
  });
},
cerrarSesion() {
  Firebase.auth()
    .signOut()
    .then(
      (accept) => {
        this.$router.push({ name: "login" });
      },
      (reject) => {
        var errorCode = reject.code;
        this.loading = false;
        this.$message({
          showClose: true,
          message: reject.message,
          type: "error",
        });
      }
    )
  }
}
```

```
    );  
  },  
},  
};  
</script>  
  
<style scoped>  
.juguetes {  
  padding: 60px;  
}  
</style>
```

- **Paso 3:** Crear la vista Agregar.vue

Agregar.vue

```
<template>  
  <div>  
    <h1>Agregar Juguete</h1>  
    <el-form ref="form" :model="form" label-width="120px">  
      <el-form-item label="Código">  
        <el-input v-model="form.codigo"></el-input>  
      </el-form-item>  
      <el-form-item label="Nombre">  
        <el-input v-model="form.nombre"></el-input>  
      </el-form-item>  
      <el-form-item label="Stock">  
        <el-input v-model="form.stock"></el-input>  
      </el-form-item>  
      <el-form-item label="Precio">  
        <el-input v-model="form.precio"></el-input>  
      </el-form-item>  
      <el-form-item>  
        <el-button type="info" @click="volver">Volver</el-button>  
        <el-button type="primary" @click="Agregar(form)">  
          Agregar Juguete</el-button>  
      </el-form-item>  
    </el-form>  
  </div>  
</template>
```

```
<script>
import { mapActions } from "vuex";
export default {
  data() {
    return {
      form: {
        codigo: "",
        nombre: "",
        stock: "",
        precio: "",
      },
    };
  },
  methods: {
    ...mapActions(["agregarJuguete"]),
    Agregar(juguete) {
      this.agregarJuguete(juguete);
      const options = { type: "success", message: "Juguete agregado" };

      this.$message({
        ...options,
      });
      this.$router.push({ name: "juguetes" });
    },
    volver() {
      this.$router.push({ name: "juguetes" });
    },
  },
};
</script>
```

- **Paso 4:** Crear la vista Editar.vue

Editar.vue

```
<template>
<div>
  <h1>Editar Juguete</h1>
  <el-form ref="form" :model="form" label-width="120px">
    <el-form-item label="Código">
      <el-input v-model="form.codigo" :disabled="true"></el-input>
    </el-form-item>
  </el-form>
</div>
```

```
<el-form-item label="Nombre">
  <el-input v-model="form.nombre"></el-input>
</el-form-item>
<el-form-item label="Stock">
  <el-input v-model="form.stock"></el-input>
</el-form-item>
<el-form-item label="Precio">
  <el-input v-model="form.precio"></el-input>
</el-form-item>
<el-form-item>
  <el-button type="info" @click="volver">Volver</el-button>
  <el-button type="primary" @click="dialogActualizar"
    >Actualizar Juguete</el-button
  >
</el-form-item>
</el-form>
</div>
</template>
<script>
import { mapActions } from "vuex";
export default {
  props: ["juguete"],
  data() {
    return {
      form: {
        codigo: "",
        nombre: "",
        stock: "",
        precio: "",
      },
    };
  },
  mounted() {
    this.form = this.juguete.data;
  },
  methods: {
    ...mapActions(["actualizarJuguete"]),
    dialogActualizar(juguete) {
      this.$confirm(
        "Actualizará el juguete seleccionado. ¿Desea continuar?",
        "Warning",
        {
          confirmButtonText: "Actualizar",
          cancelButtonText: "Cancelar",
        }
      );
    }
  }
};
```

```
        type: "warning",
      }
    )
    .then(async () => {
      const juguete = this.form;
      const id = this.juguete.id;
      await this.actualizarJuguete({ juguete, id });
      let options = { type: "success", message: "Juguete
actualizado" };

      this.$message({
        ...options,
      });
      this.$router.push({ name: "juguetes" });
    })
    .catch(() => {
      this.$message({
        type: "info",
        message: "Acción cancelada",
      });
    });
  },
  volver() {
    this.$router.push({ name: "juguetes" });
  },
},
};
</script>
```

- **Paso 5:** Crear la vista Login.vue

Login.vue

```
<template>
  <div id="login">
    <h1>Acceso al sistema - Otto Klaus</h1>
    <el-form label-width="100px" class="form-login"
:label="formLabelAlign">
      <el-input
        placeholder="Correo electrónico"
        v-model="formLabelAlign.user"
      ></el-input>
```

```
<el-input
  type="password"
  placeholder="Contraseña"
  v-model="formLabelAlign.password"
  autocomplete="off"
></el-input>
<el-button
  type="primary"
  :loading="loading"
  class="ingresar-login"
  @click="login"
>Ingresar</el-button>
</el-form>
</div>
</template>

<script>
import Firebase from "firebase";

export default {
  data() {
    return {
      formLabelAlign: {
        user: "ottoklaus@desafiolatam.com",
        password: "12345678",
      },
      loading: false,
    };
  },

  methods: {
    login() {
      this.loading = true;
      Firebase.auth()
        .signInWithEmailAndPassword(
          this.formLabelAlign.user,
          this.formLabelAlign.password
        )
        .then(
          (accept) => {
            this.$router.push("juguetes");
            this.loading = false;
          }
        );
    }
  }
};
```



```
    },  
    (reject) => {  
      var errorCode = reject.code;  
      this.loading = false;  
      this.$message({  
        showClose: true,  
        message: reject.message,  
        type: "error",  
      });  
    }  
  );  
},  
},  
};  
</script>
```

```
<style lang="scss" scoped="">  
#login {  
  padding: 60px;  
  .ingresar-login {  
    width: 100%;  
  }  
  .img-inicial,  
  .form-login {  
    display: block;  
    margin: auto;  
    margin-bottom: 15px;  
  }  
  .form-login {  
    width: 300px;  
    .el-input {  
      margin-bottom: 10px;  
    }  
  }  
}  
</style>
```

Requerimiento 2

- **Paso 1:** Crear las acciones en Vuex para el CRUD

Store/index.js

```
actions: {
  getJuguetes({ commit }) {
    commit("loadingTable");
    firebase
      .firestore()
      .collection("juguetes")
      .onSnapshot((snapshot) => {
        const juguetes = [];
        snapshot.forEach((doc) => {
          juguetes.push({ id: doc.id, data: doc.data() });
        });

        commit("setJuguetes", juguetes);
      });
  },
  agregarJuguete({ commit }, juguete) {
    commit("loadingTable");
    firebase
      .firestore()
      .collection("juguetes")
      .add(juguete)
      .then(commit("loadingTable"));
  },
  eliminarJuguete({ commit }, jugueteID) {
    commit("loadingTable");
    firebase
      .firestore()
      .collection("juguetes")
      .doc(jugueteID)
      .delete()
      .then(commit("loadingTable"));
  },
  async actualizarJuguete({ commit }, { juguete, id }) {
    commit("loadingTable");
    firebase
      .firestore()
```

```
        .collection("juguetes")
        .doc(id)
        .set(juguete)
        .then(commit("loadingTable"));
    },

    getJuguetesByPart({ commit }) {
        commit("loadingTable");
        firebase
            .firestore()
            .collection("juguetes")
            .where("codigo", "==", "A0001")
            .where("precio", ">", "2000")
            .onSnapshot((snapshot) => {
                const juguetes = [];
                snapshot.forEach((doc) => {
                    juguetes.push({ id: doc.id, data: doc.data() });
                });

                console.log(juguetes)
            });
    },

    },

    },
```

- **Paso 2:** Crear las mutaciones en Vuex para el CRUD

Store/index.js

```
mutations: {
  setJuguetes(state, juguetes) {
    state.juguetes = juguetes;
    state.loading = false;
  },
  loadingTable(state) {
    state.loading = true;
  },
  eliminarJuguete(state, juguete) {
    state.juguetes = state.juguetes.filter((j) => {
      return j.codigo !== juguete.codigo;
    });
  },
  actualizarJuguete(state, juguete) {
    state.juguetes = state.juguetes.map((j) => {
      if (j.codigo === juguete.codigo) return juguete;
      else return j;
    });
  },
  agregarJuguete(state, juguete) {
    state.juguetes.push(juguete);
  },
},
```

- **Paso 3:** Crear el state en Vuex para el CRUD

Store/index.js

```
state: {
  juguete: {
    codigo: "",
    nombre: "",
    stock: 0,
    precio: 0,
  },
  edit: false,
  loading: false,
```

```
juguetes: [],  
},
```

Requerimiento 3

- **Paso 1:** Crear el getters en Vuex para filtrar los juguetes por stock

Store/index.js

```
getters: {  
  juguetesConStock({ juguetes }) {  
    return juguetes.filter((juguete) => juguete.data.stock > 0);  
  },  
},
```

- **Paso 2:** Incluir la computed property en la vista Juguetes.vue

Juguetes.vue

```
computed: {  
  ...mapGetters(["juguetesConStock"]),  
},
```

Requerimiento 4

- **Paso 1:** Incluir un método en Login.vue para la autenticación de usuario

Login.vue

```
methods: {  
  login() {  
    this.loading = true;  
    Firebase.auth()  
      .signInWithEmailAndPassword(  
        this.formLabelAlign.user,  
        this.formLabelAlign.password  
      )
```

```
.then(  
  (accept) => {  
    this.$router.push("juguetes");  
    this.loading = false;  
  },  
  (reject) => {  
    var errorCode = reject.code;  
    this.loading = false;  
    this.$message({  
      showClose: true,  
      message: reject.message,  
      type: "error",  
    });  
  }  
);  
},  
},
```

Requerimiento 5

- **Paso 1:** Agregar los guardianes (beforeEach) en Vue Router

```
router.beforeEach((to, from, next) => {  
  let user = firebase.auth().currentUser;  
  let authRequired = to.matched.some((route) => route.meta.login);  
  
  if (!user && authRequired) {  
    next("login");  
  } else if (user && !authRequired) {  
    next("lista");  
  } else {  
    next();  
  }  
});
```

Requerimiento 6

```
npm install -g firebase-tools  
firebase login  
firebase init  
npm run build  
firebase deploy
```