Informe de la aplicación realizada para el segundo parcial.

Alumno: Tomasetig Lautaro

Esta aplicación utiliza Vue para crear una aplicación de un cine, la cual le permite al usuario ver que películas hay en cartelera, opinar sobre que películas le gustaría ver próximamente y valorar la película destacada de la semana.

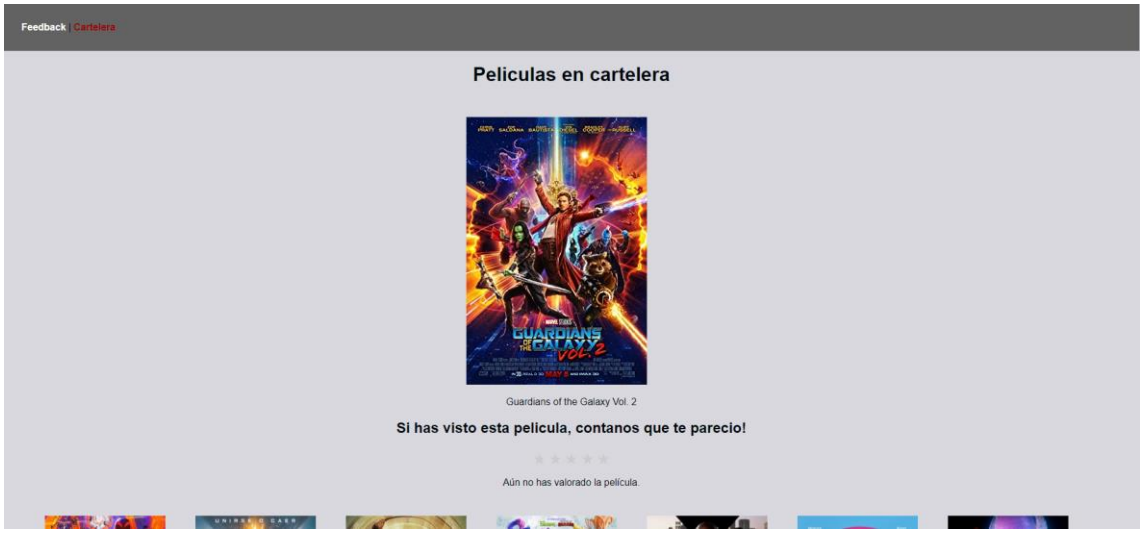
Primer pantalla:

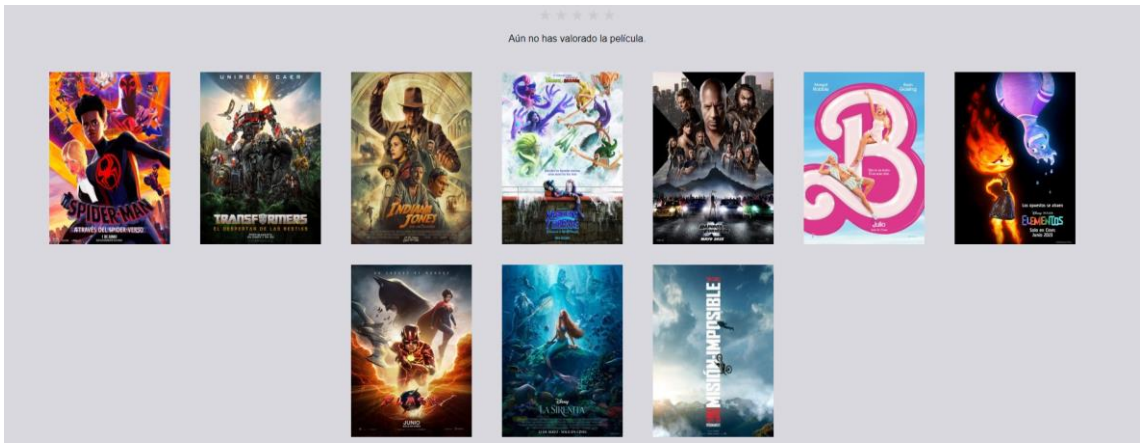
En esta pantalla se muestra el apartado llamado “Feedback”, el cual le pide al usuario que ingrese la película que le gustaría ver, y si la quiere en 2D o 3D. Al ingresar los datos y clickear en el botón “cargar”, los datos se cargan y se muestran debajo, además se le permite al usuario eliminar una película ya cargada.

Debajo se muestra la cantidad de películas cargadas.

Ademas, la lista contiene localStorage, el cual, permite guardar los datos cargados si el usuario recarga la pagina o quiere ver la cartelera.

Segunda pantalla:





En esta pantalla muestra en que apartado de la aplicación se encuentra, en este caso, en el apartado “Cartelera”. Luego, muestra la película destacada, la cual se le pide al usuario que si

la vió, la califique mediante estrellas. Debajo de esto, se encuentra el catalogo de las películas que están en cartelera actualmente.

Funcionalidades de la aplicación:

Como primer punto, se encuentra el enrutamiento de la pagina. En el archivo App.vue se define la estructura y los estilos que muestra barra de navegación. Luego en el archivo de enrutamiento, se definen las rutas.

```vue
<template>
  <div id="app">
    <nav>

      <router-link to="/">Feedback</router-link> |
      <router-link to="/cartelera">Cartelera</router-link>


    </nav>
    <router-view/>
  </div>
</template>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;

  color: #00060c;

}

nav {
  padding: 30px;
  background-color: rgb(98, 98, 98);
}

nav a {
  font-weight: bold;
  color: #ffffff;
  text-decoration: none;
}

nav a.router-link-exact-active {
  color: rgb(156, 3, 3);
}
</style>
```

```vue
import Vue from 'vue'
import VueRouter from 'vue-router'
import HomeView from '../views/HomeView.vue'


Vue.use(VueRouter)

const routes = [
  {
    path: '/',
    name: 'home',
    component: HomeView
  },
  {
    path: '/cartelera',
    name: 'cartelera',

    component: () => import( '../views/CarteleraView.vue')
  },


]

const router = new VueRouter({
  routes
})

export default router
```

Luego están los archivos de vista:

```vue
<template>
  <div class="home">

    <Peliculas />
  </div>
</template>

<script>

import Peliculas from '@/components/PeliculasComponente.vue'



export default {
  name: 'HomeView',
  components: {
    Peliculas
  }
}
</script>

```

```vue
<template>

  <div class="pelis">
    <h1>Peliculas en cartelera</h1>

    <img v-if="pelicula" :src="pelicula.Poster" :alt="pelicula.Title" class="poster" />
    <p>{{ pelicula.Title }}</p>


    <valoracion-pelicula />
    <img v-bind:src="spiderman" alt="Poster promocional de Spiderman"  />

    <img v-bind:src="transformers" alt="Poster promocional de Transformers" />
    <img v-bind:src="indiana" alt="Poster promocional de Indiana Jones" />
    <img v-bind:src="krakens" alt="Poster promocional de Krakens y Sirenas" />
    <img v-bind:src="rapidos" alt="Poster promocional de Rapidos y furiosos" />
    <img v-bind:src="barbie" alt="Poster promocional de Barbie" />
    <img v-bind:src="elementos" alt="Poster promocional de Elementos" />
    <img v-bind:src="flash" alt="Poster promocional de Flash" />
    <img v-bind:src="sirenita" alt="Poster promocional de La Sirenita" />
    <img v-bind:src="mision" alt="Poster promocional de Mision Imposible" />

  </div>

</template>

<script>
import ValoracionPelicula from '@/components/ValoracionPelicula.vue';
export default {
  components: {
    ValoracionPelicula
  },
  mounted(){
      fetch("https://www.omdbapi.com/?i=tt3896198&apikey=d1355d4a")
          .then(response => response.json())
          .then(response => {
            this.pelicula = response;
            console.log(response)
          localStorage.setItem("dato",JSON.stringify(this.pelicula))
          })
          .catch( err => console.error(err));



  },
  data() {
    return {
      pelicula: [],
      spiderman: require('@/assets/spiderman.jpg'),
      transformers: require('@/assets/transformers.jpg'),
      indiana: require('@/assets/indiana.jpg'),
      krakens: require('@/assets/krakens.jpg'),
      rapidos: require('@/assets/rapidos.jpg'),
      barbie: require('@/assets/barbie.jpg'),
      elementos: require('@/assets/elementos.jpg'),
      flash: require('@/assets/flash.jpg'),
      sirenita: require('@/assets/sirenita.jpg'),
      mision: require('@/assets/mision.jpg')
    };
  }
};
</script>

<style>
  img{
    margin-right: 50px;
    margin-top: 30px;

  }

  .pelis{
    text-align: center;
  }

  p{
    text-align: center;
  }

  .poster{
```

Estos importan sus propios componentes, los cuales contienen las funcionalidades de las vistas. Ademas el archivo “CarteleraView.vue” contiene los estilos para las imágenes y las propias imagenes. En este mismo también se importa la api sobre una película, la cual se utiliza como película destacada a valorar.

Luego están los componentes:

```vue
<template>
    <div>
      <h2>Películas cargadas: {{ contador() }}</h2>
    </div>
  </template>

  <script>
  export default {
    props: {
      peliculas: {
        type: Array,
        required: true
      }
    },
    methods: {
      contador() {
        return this.peliculas.length;
      }
    }
  };
  </script>
```

Este es el componente ContadorComponente.vue, el cual permite que se cuente la cantidad de películas cargadas y muestre la cantidad en pantalla.

Luego esta el componente “PeliculasComponente.vue”:

```vue
<template>
  <div>
    <h1>Contanos qué películas te gustaría ver</h1>
    <form v-bind:class="mostrarError ? 'error-form' : 'my-form'" @submit.prevent="agregarPelicula">
      <input type="text" v-model="nuevaPelicula" placeholder="Inserte su película" />
      <select v-model="tipoPelicula">
        <option value="2D">2D</option>
        <option value="3D">3D</option>
      </select>
      <button v-bind:class="mostrarError ? 'error-button' : 'my-button'" type="submit">Cargar</button>
    </form>

    <ul v-if="Array.isArray(peliculas) && peliculas.length > 0">
  <li v-for="(pelicula, index) in peliculas" :key="index">
    {{ pelicula.titulo }} - {{ pelicula.tipo }}
    <button @click="borrarPelicula(index)">Borrar</button>
  </li>
</ul>

    <p v-bind:class="mostrarError ? 'error-paragraph' : 'my-paragraph'" v-if="mostrarError">Por favor, complete los campos.</p>



    <contador :peliculas="peliculas" />
  </div>
</template>

<script>
import Contador from '@/components/ContadorComponente.vue';

export default {
  components: {
    Contador
  },
  data() {
    return {
      nuevaPelicula: '',
      tipoPelicula: '2D',
      peliculas: [],
      mostrarError: false
```

```
      mostrarError: false
    };
  },
  mounted() {
    this.recuperarPeliculas();
  },
  methods: {
    agregarPelicula() {
      if (this.nuevaPelicula.trim() !== '' && this.tipoPelicula.trim() !== '') {
        const pelicula = {
          titulo: this.nuevaPelicula,
          tipo: this.tipoPelicula
        };
        this.peliculas.push(pelicula);
        this.nuevaPelicula = '';
        this.tipoPelicula = '';
        this.guardarPeliculas();
        this.mostrarError = false;
      } else {
        this.mostrarError = true;
      }
    },
    borrarPelicula(index) {
      if (confirm("Estas seguro desea eliminar: "  +this.peliculas[index].titulo + "?")) {
        this.peliculas.splice(index, 1);
      this.guardarPeliculas();
      }

    },
    guardarPeliculas() {
      localStorage.setItem('peliculas', JSON.stringify(this.peliculas));
    },
    recuperarPeliculas() {
      const peliculas = localStorage.getItem('peliculas');
      console.log('Peliculas recuperadas:', peliculas);
      if (peliculas) {
        this.peliculas = JSON.parse(peliculas);
      }
    }
  }
```


```
      mostrarError: false
    };
  },
  mounted() {
    this.recuperarPeliculas();
  },
  methods: {
    agregarPelicula() {
      if (this.nuevaPelicula.trim() !== '' && this.tipoPelicula.trim() !== '') {
        const pelicula = {
          titulo: this.nuevaPelicula,
          tipo: this.tipoPelicula
        };
        this.peliculas.push(pelicula);
        this.nuevaPelicula = '';
        this.tipoPelicula = '';
        this.guardarPeliculas();
        this.mostrarError = false;
      } else {
        this.mostrarError = true;
      }
    },
    borrarPelicula(index) {
      if (confirm("Estas seguro desea eliminar: "  +this.peliculas[index].titulo + "?")) {
        this.peliculas.splice(index, 1);
      this.guardarPeliculas();
      }

    },
    guardarPeliculas() {
      localStorage.setItem('peliculas', JSON.stringify(this.peliculas));
    },
    recuperarPeliculas() {
      const peliculas = localStorage.getItem('peliculas');
      console.log('Peliculas recuperadas:', peliculas);
      if (peliculas) {
        this.peliculas = JSON.parse(peliculas);
      }
    }
  }
```


Y luego están los estilos…

En este componente se muestra el formulario con su funcionalidad, como la de cargar los datos, eliminarlos, su validación (si el usuario ingreso datos en los 2 campos). Luego esta la carga de ‘peliculas’ en el localStorage. Tambien se agrega a la función eliminar, un confirm, el cual le pregunta al usuario si desea eliminar definitivamente la película cargada.

Por ultimo, esta el componente “ValoracionPelicula.vue”:

```vue
<template>
    <div class="valoracion-pelicula">
      <h3>Si has visto esta pelicula, contanos que te parecio!</h3>
      <div class="stars">
        <span
          v-for="(star, index) in 5"
          :key="index"
          :class="{ filled: index < rating }"
          @mouseover="hoverRating(index + 1)"
          @mouseout="hoverRating(0)"
          @click="setRating(index + 1)"
        >
          ★
        </span>
      </div>
      <p v-if="rating > 0">Has valorado la película con {{ rating }} estrellas.</p>
      <p v-else>Aún no has valorado la película.</p>

    </div>
  </template>

  <script>
  export default {
    data() {
      return {
        rating: 0,
        hoveredRating: 0
      };
    },
    methods: {
      setRating(rating) {
        this.rating = rating;

      },
      hoverRating(rating) {
        this.hoveredRating = rating;
      }
    }
  };
  </script>
```

Este le permite al usuario valorar la película destacada mediante estrellas. Dependiendo de la cantidad de estrellas seleccionadas por el usuario, debajo se muestra un mensaje dependiendo de la cantidad, y también, si el usuario no ingreso ninguna puntación, se muestra el mensaje “Aún no has valorado la pelicula”.

Flujo de la aplicación:

1-Al cargar la aplicación, carga el apartado Feedback, el cual le permite al usuario ingresar que películas quiere ver próximamente.

2-El usuario ingresa la película y en que formato verla, si en 2D o 3D.

3-El usuario hace click en el botón cargar.

4-Se valida si al usuario le falto llenar algún campo

5-Si se completaron los campos, se cargan los datos y los muestra debajo. En lo contrario, se le pide al usuario que ingrese el dato faltante.

6-Se actualiza el contador.

7-Si el usuario lo desea, puede eliminar los campos ingresados.

8-Luego, el usuario puede dirigirse al apartado de Cartelera para ver que películas hay actualmente para ver.

9-En el apartado Cartelera, el usuario puede decidir si valorar la película destacada o no.

10-El usuario puede decidir si irse o seguir recomendando películas en el apartado anterior.