

## Solución Prueba - My Store

- **Paso 1:** Crear el proyecto ocupando las siguientes opciones:  
(Requerimientos abordados: 1 )

```
vue create mammamia
```

```
? Please pick a preset: Manually select features
? Check the features needed for your project:
  ( ) Choose Vue version
  (*) Babel
  ( ) TypeScript
  ( ) Progressive Web App (PWA) Support
  (*) Router
  (*) Vuex
  ( ) CSS Pre-processors
  ( ) Linter / Formatter
>(*) Unit Testing
  (*) E2E Testing
```

- **Paso 2:** Crear los componentes Navbar y Header para la vista principal.  
(Requerimientos abordados: 2)

## Navbar.vue

```
<template>
  <div class="navbar d-flex justify-content-between text-white py-3">
    <div class="container">
      <router-link to="/">
        <div class="d-flex">
          <h4 class="mb-0">#127829;</h4>
          <h4 class="mx-1 mb-0">Pizzería Mamma Mia!</h4>
        </div>
      </router-link>

      <router-link to="/carrito">
        <div>
          <h4 class="mb-0">#128722; $ {{formatNumber(total)}}</h4>
        </div>
      </router-link>
    </div>
  </div>
</template>

<script>
import { mapGetters } from 'vuex';
import utils from "@/utils/functions";
const { formatNumber } = utils;
export default {
  name: "Navbar",
  props: {
    msg: String,
  },
  computed: {
    ...mapGetters([ "total" ]),
  },
  methods: {
    formatNumber
  }
};
</script>

<style scoped>
```

```
.navbar {  
  background: #17a2b8;  
}  
</style>
```

## Header.vue

```
<template>  
  <div  
    class="header d-flex flex-column justify-content-center text-white  
text-center"  
  >  
    <h1>¡Pizzería Mamma Mia!</h1>  
    <h6>¡Tenemos las mejores pizzas que podrás encontrar!</h6>  
    <hr class="w-75 mx-auto" />  
  </div>  
</template>  
  
<style scoped>  
.header {  
  background: linear-gradient(rgba(0, 0, 0, 0.7), rgba(0, 0, 0, 0.7)),  
    url("~/assets/Header.jpg");  
  background-size: cover;  
  height: 30vh;  
}  
  
h1 {  
  font-size: 50px;  
}  
  
h6 {  
  font-size: 20px;  
}  
</style>  
  
<script>  
export default {  
  name: "Header",  
};  
</script>
```

- **Paso 3:** Agregar el componente Navbar al App.vue para que esté disponible en donde la aplicación. (Requerimientos abordados: 2)

### App.vue

```
<template>
  <div id="app">
    <Navbar />
    <router-view />
  </div>
</template>

<script>
import Navbar from "@components/Navbar.vue";
import { mapActions } from "vuex";
export default {
  mounted() {
    this.get_Pizzas();
  },
  methods: {
    ...mapActions(["get_Pizzas"]),
  },
  components: {
    Navbar,
  },
};
</script>

<style>
a,
a:hover {
  text-decoration: none;
  color: white;
}

ul {
  list-style: none;
}
</style>
```

- **Paso 4:** Agregar el componente Header al Home. (Requerimientos abordados: 2)

## Home.vue

```
<template>
  <div class="home">
    <Header />
    <div class="container">
      <div class="row py-5">
        <div class="col-12 col-sm-3 mb-4" v-for="(pizza, i) in pizzas"
:key="i">
          <Card :pizza="pizza" />
        </div>
      </div>
    </div>
  </div>
</template>

<script>
import Header from "@components/Header.vue";
import Card from "@components/Card.vue";
import { mapState } from "vuex";
export default {
  name: "Home",
  components: {
    Header,
    Card,
  },

  computed: {
    ...mapState(["pizzas"]),
  },
};
</script>
```

- **Paso 5:** Crear la acción en el store para la obtención de las pizzas del archivo JSON y agregar el mounted en el app para ejecutarlo desde cualquier parte de la aplicación. (Requerimientos abordados: 5)

### Store.js

```
state: {
  pizzas: [],
},
mutations: {
  GET_PIZZAS(state, pizzas) {
    state.pizzas = pizzas;
  }
},
actions: {
  async get_Pizzas({ commit }) {
    const { data: pizzas } = await axios.get("/pizzas.json");
    commit("GET_PIZZAS", pizzas);
  },
}
}
```

### App.vue

```
mounted() {
  this.get_Pizzas();
},
methods: {
  ...mapActions(["get_Pizzas"]),
},
```

- **Paso 6:** Mostrar en la vista principal las vistas usando un componente Card y la directiva v-for. (Requerimientos abordados: 2, 3, 5)

### Home.vue

```
<template>
  <div class="home">
    <Header />
    <div class="container">
      <div class="row py-5">
        <div class="col-12 col-sm-3 mb-4" v-for="(pizza, i) in pizzas"
:key="i">
          <Card :pizza="pizza" />
        </div>
      </div>
    </div>
  </div>
</template>
```

### Card.vue

```
<template>
  <div class="card w-100">
    
    <div class="card-body">
      <h4 class="card-title text-capitalize">{{ pizza.name }}</h4>
      <hr />
      <p class="card-text ">
        <b>Ingredientes:</b>
      </p>
      <ul class="m-0">
        <li
          class="text-capitalize w-75"
          v-for="(ingrediente, i) in pizza.ingredients"
          :key="i"
        >
          &#127829; {{ ingrediente }}
        </li>
      </ul>
    </div>
    <hr />
    <div class="card-body">
      <h2 class="text-center text-dark pb-3">
```

```
    $ {{ formatNumber(pizza.price) }}
  </h2>
  <div class="d-flex justify-content-around">
    <router-link :to="`/pizza/${pizza.id}`">
      <button class="btn btn-info text-white">
        Ver Más &#128064;
      </button></router-link>
    >
    <button class="btn btn-danger" @click="addToCart(pizza.id)">
      Añadir &#128722;
    </button>
  </div>
</div>
</div>
</template>
<script>
import utils from "@/utils/functions";
const { formatNumber } = utils;
import { mapActions } from "vuex";

export default {
  name: "Card",
  props: ["pizza"],
  methods: {
    formatNumber,
    ...mapActions(["addToCart"]),
  },
};
</script>
```

### functions.js

```
const formatNumber = (price) => {
  return price.toLocaleString("de-DE");
};

export default { formatNumber };
```



- **Paso 7:** Crear la vista de detalle. (Requerimientos abordados: 2, 3, 4 y 5)

#### router.js

```
{
  path: "/pizza/:id",
  name: "Detalle",
  props: true,
  component: () =>
    import(/* webpackChunkName: "Detalle" */ "../views/Detalle.vue"),
},
```

#### Detalle.vue

```
<template>
  <div class="detalle py-5 d-flex justify-content-center container">
    <div class="card mb-3">
      <div class="row g-0">
        <div
          class="col-md-5 img"
          :style="{ backgroundImage: `url(${pizza.img})` }"
        ></div>
        <div class="col-md-7">
          <div class="card-body">
            <h3 class="card-title text-capitalize">{{ pizza.name }}</h3>
            <hr />
            <p class="card-text">
              {{ pizza.desc }}
            </p>
            <p class="card-text ">
              <b>Ingredientes:</b>
            </p>
            <ul>
              <li
                class="text-capitalize w-75"
                v-for="(ingrediente, i) in pizza.ingredients"
                :key="i"
              >
                &#127829; {{ ingrediente }}
              </li>
            </ul>

            <div class="d-flex justify-content-between">
```

```

    <h2 class=" text-dark">
      Precio: ${{ formatNumber(pizza.price) }}
    </h2>
    <button class="btn btn-danger"
@click="addToCart(pizza.id)">
      Añadir &#128722;
    </button>
  </div>
</div>
</div>
</div>
</div>
</template>

<script>
import { mapGetters, mapActions } from "vuex";
import utils from "@/utils/functions";
const { formatNumber } = utils;

export default {
  name: "Detalle",
  props: ["id"],
  computed: {
    ...mapGetters(["pizzaById"]),

    pizza() {
      const { id } = this;
      return this.pizzaById(id);
    },
  },
  methods: { formatNumber, ...mapActions(["addToCart"]) },
};
</script>

<style scoped>
.img {
  background-size: cover;
  background-repeat: no-repeat;
  background-position: center;
}
</style>

```

- **Paso 8:** Crear la vista del carrito. (Requerimientos abordados: 2, 3, 4 y 5)

#### router.js

```
{
  path: "/carrito",
  name: "Carrito",
  props: true,
  component: () =>
    import(/* webpackChunkName: "Carrito" */ "../views/Carrito.vue"),
},
```

#### Carrito.vue

```
<template>
  <div class="carrito p-5">
    <div class="detalles bg-light w-75 m-auto p-5">
      <h5>Detalles del pedido:</h5>
      <div class="p-3 list">
        <div v-for="(pizza, i) in carrito" :key="i">
          <div class="d-flex justify-content-between align-items-center">
            <div class="d-flex align-items-center">
              
              <h6 class="mb-0 text-capitalize p-2">{{ pizza.name }}</h6>
            </div>
            <div class="d-flex justify-content-between align-items-center">
              <h6 class="mb-0 p-2 text-success">
                ${{ totalByPizza(pizza.cant, pizza.price) }}
              </h6>
              <button class="btn btn-danger"
                @click="minus(pizza.id)">-</button>
              <b class="mx-2">{{ pizza.cant }}</b>
              <button class="btn btn-primary"
                @click="plus(pizza.id)">+</button>
            </div>
          </div>
          <hr class="mt-2" />
        </div>
      </div>
      <h2>Total: ${{ formatNumber(total) }}</h2>
      <button class="btn btn-success">Ir a Pagar</button>
    </div>
  </div>
```

```
    </div>
  </div>
</div>
</template>

<script>
import { mapActions, mapGetters } from "vuex";
import utils from "@utils/functions";
const { formatNumber } = utils;

export default {
  name: "Carrito",
  computed: {
    ...mapGetters(["carrito", "total"]),
  },
  methods: {
    ...mapActions(["plus", "minus"]),
    totalByPizza(cant, price) {
      return formatNumber(cant * price);
    },
    formatNumber,
  },
};
</script>

<style scoped>
.list {
  background: white;
}
</style>
```

- **Paso 9:** Agregar las funciones en vuex para el cálculo total por pizza, total a pagar, agregar y eliminar cantidades. (Requerimientos abordados: 2, 3 y 5)

### Store.js

```
import Vue from "vue";
import Vuex from "vuex";
import axios from "axios";

import createPersistedState from "vuex-persistedstate";

Vue.use(Vuex);

export default new Vuex.Store({
  state: {
    pizzas: [],
    carrito: [],
  },
  mutations: {
    GET_PIZZAS(state, pizzas) {
      state.pizzas = pizzas;
    },
    ADD_TO_CART(state, pizza) {
      state.carrito.push(pizza);
    },
    UPDATE_CART(state, carrito) {
      state.carrito = carrito;
    },
  },
  actions: {
    async get_Pizzas({ commit }) {
      const { data: pizzas } = await axios.get("/pizzas.json");
      commit("GET_PIZZAS", pizzas);
    },

    addToCart({ commit, state, dispatch }, pizzaId) {
      const pizzaExist = state.carrito.find((p) => p.id == pizzaId);
      if (pizzaExist) {
        dispatch("plus", pizzaId);
      } else {
        const pizza = { id: pizzaId, cant: 1 };
        commit("ADD_TO_CART", pizza);
      }
      alert("Pizza añadida al carrito!");
    },
  },
});
```

```
    },

    plus({ state, commit }, id) {
      const carrito = state.carrito.map((p) =>
        p.id == id ? (p.cant++, p) : p
      );
      commit("UPDATE_CART", carrito);
    },

    minus({ state, commit }, id) {
      const { cant } = state.carrito.find((p) => p.id == id);
      let carrito;
      if (cant == 1) {
        carrito = state.carrito.filter((p) => p.id !== id);
      } else {
        carrito = state.carrito.map((p) => (p.id == id ? (p.cant--, p) :
p)));
      }
      commit("UPDATE_CART", carrito);
    },
  },

  getters: {
    pizzaById: (state) => (id) => {
      const pizza = state.pizzas.find((p) => p.id == id);
      return pizza;
    },
    carrito: (state) => {
      const carrito = state.carrito.map((p) => {
        const pizza = state.pizzas.find((pizza) => pizza.id == p.id);
        return { ...pizza, cant: p.cant };
      });
      return carrito;
    },

    total: (state) => {
      return state.carrito.reduce((a, b) => {
        const pizza = state.pizzas.find((p) => p.id == b.id);
        const totalByPizza = b.cant * pizza.price;
        return a + totalByPizza;
      }, 0);
    },
  },
},
```

```
plugins: [createPersistedState()],  
});
```

- **Paso 10:** Agregar los tests unitarios para comprobar las acciones addToCart y plus. (Requerimientos abordados: 6)

#### tests.spec.js

```
import { createLocalVue } from '@vue/test-utils'  
import Vuex from 'vuex'  
import storeConfig from './mocks/store-config'  
  
const localVue = createLocalVue()  
localVue.use(Vuex)  
const store = new Vuex.Store(storeConfig)  
  
describe("Tests de vuex", () => {  
  
  const pizza = { cant: 1, id: "P003" }  
  
  it('Acción para agregar una pizza al carrito', () => {  
    expect(store.state.carrito).toHaveLength(0)  
    store.dispatch('addToCart', pizza.id)  
    expect(store.state.carrito).toEqual([pizza])  
  })  
  
  it('Acción para aumentar la cantidad de una pizza en el carrito', ()  
=> {  
    store.dispatch('plus', pizza.id)  
    expect(store.state.carrito[0].cant).toBe(2)  
  })  
  
})
```

- **Paso 11:** Agregar los tests e2e para probar el botón añadir al carrito y la eliminación de una pizza en el carrito. (Requerimientos abordados: 7)

### Carrito.js

```
describe('Tests del carrito', () => {
  it('Botón para agregar una pizza al carrito', () => {
    cy.visit('/')
    const btn = cy.get(':nth-child(1) > .card .btn-danger')
    btn.click({ force: true })
    cy.visit('/carrito')
    cy.contains("h6", "napolitana")
  })

  it('Botón para eliminar una pizza del carrito', () => {
    cy.get('.btn-danger').click()
    cy.get("h6").should('not.exist')
  })
})
```