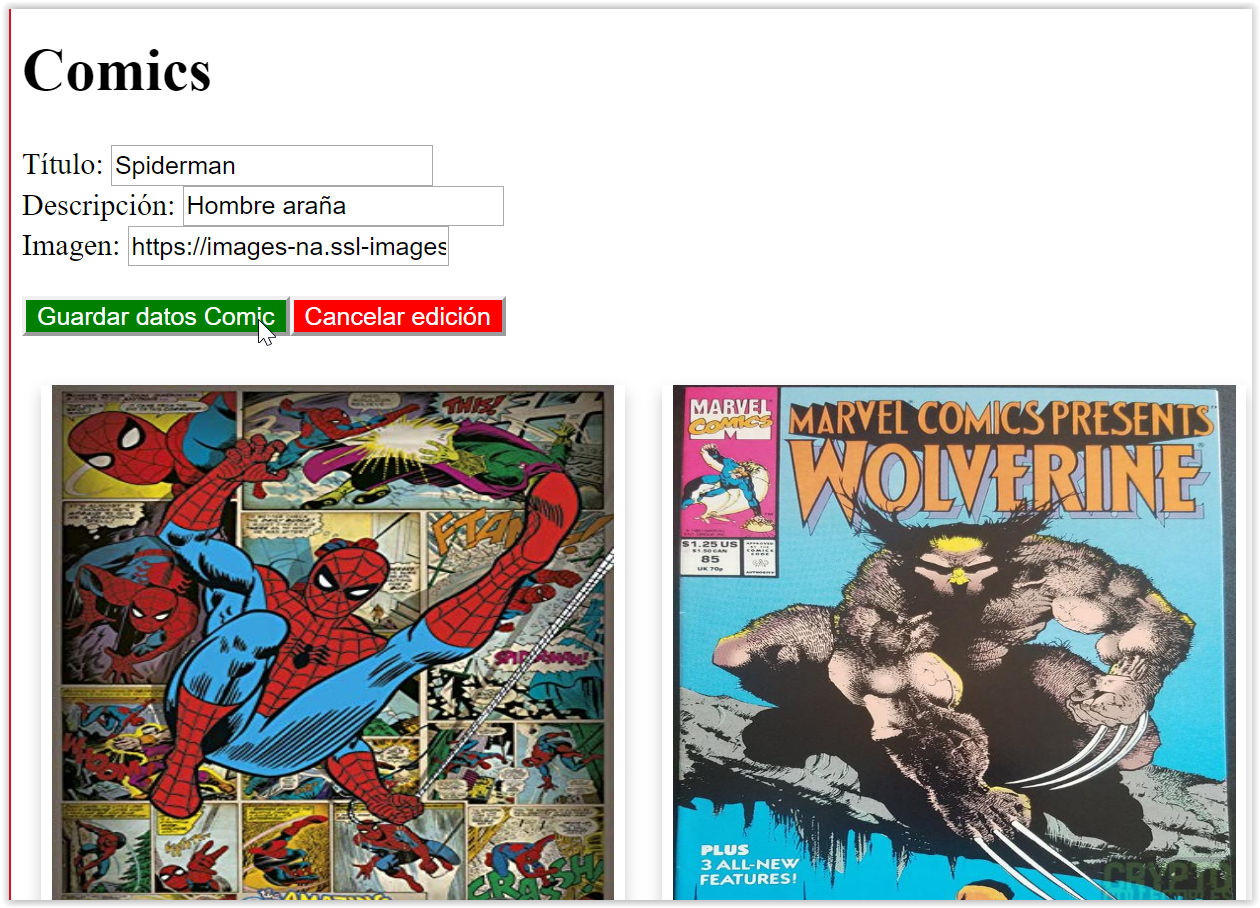
**TRABAJAR CON COLECCIONES EVENTOS COMPARTIDOS Y FORMULARIOS**

* Vamos a realizar una app dónde recorreremos objetos del State de un Component que tendrá un Array.
* Listaremos un Array de elementos Comic con dos clases.
* Posteriormente, utilizaremos la opción de eliminar un Comic o seleccionar un comic como favorito.



**VISTA EDICION**



Agregamos el módulo de formularios a **app.module.ts**

**APP.MODULE.TS**

Agregamos un nuevo modelo llamado **Comic** dentro de la carpeta **models** con nombre de archivo **comic.ts**

**COMIC.TS**

export class Comic {

  constructor(

    public titulo: string,

    public imagen: string,

    public descripcion: string

  ) {}

}

Creamos un nuevo componente llamado **Comic**

**COMIC.COMPONENT.CSS**

.flip-card {

  background-color: transparent;

  width: 300px;

  height: 300px;

  perspective: 1000px;

}

.flip-card-inner {

  position: relative;

  width: 100%;

  height: 100%;

  text-align: center;

  transition: transform 0.6s;

  transform-style: preserve-3d;

  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);

}

/\* Create three equal columns that floats next to each other \*/

.column {

  float: left;

  width: 33.33%;

  padding: 10px;

  height: 300px; /\* Should be removed. Only for demonstration \*/

}

body {

  font-family: Arial, Helvetica, sans-serif;

}

\* {

  box-sizing: border-box;

}

/\* Clear floats after the columns \*/

.row:after {

  content: "";

  display: table;

  clear: both;

}

.flip-card:hover .flip-card-inner {

  transform: rotateY(180deg);

}

.flip-card-front,

.flip-card-back {

  position: absolute;

  width: 100%;

  height: 100%;

  -webkit-backface-visibility: hidden;

  backface-visibility: hidden;

}

.flip-card-front {

  background-color: white;

  color: black;

}

.flip-card-back {

  background-color: #2980b9;

  color: white;

  transform: rotateY(180deg);

}

img {

  width: 300px;

  height: 300px;

}

p button {

  width: 100%;

}

p button.delete {

  background-color: red;

  color: white;

}

p button.edit {

  background-color: blue;

  color: white;

}

p button.save {

  background-color: green;

  color: white;

}

p button.cancel {

  background-color: red;

  color: white;

}

A continuación, hacemos el diseño de un comic.

**COMIC.COMPONENT.TS**

import { Component, OnInit, Input, Output, EventEmitter } from "@angular/core";

import { Comic } from "./../../models/comic";

@Component({

  selector: "app-comic",

  templateUrl: "./comic.component.html",

  styleUrls: ["./comic.component.css"]

})

export class ComicComponent implements OnInit {

  @Input() comic: Comic;

  @Input() index: number;

  @Output() seleccionarFavorito = new EventEmitter();

  @Output() eliminarComic = new EventEmitter();

  @Output() editarComic = new EventEmitter();

  //CUANDO EL USUARIO HAGA CLICK SOBRE UN COMIC

  //CREAMOS UNA FUNCION PARA ENVIAR LA INFO AL PARENT

  marcarFavorito() {

    //EMITIMOS LA RESPUESTA AL PADRE ENVIANDO

    //EL COMIC DE ESTE COMPONENTE

    this.seleccionarFavorito.emit({ comic: this.comic });

  }

  eliminar() {

    //ENVIAMOS EL INDICE PARA ELIMINAR EL COMIC

    this.eliminarComic.emit({ indice: this.index });

  }

  editar() {

    //ENVIAMOS EL INDICE PARA ELIMINAR EL COMIC

    this.editarComic.emit({ indice: this.index });

  }

  //METODO PARA ELIMINAR

  //¿QUE NECESITAMOS PARA ELIMINAR?

  //EL INDICE DEL PROPIO COMIC A ELIMINAR

  constructor() {}

  ngOnInit(): void {}

}

**COMIC.COMPONENT.HTML**

<div class="flip-card column">

  <div class="flip-card-inner">

    <div class="flip-card-front">

      <img src="{{ comic.imagen }}" alt="{{ comic.titulo }}" />

    </div>

    <div class="flip-card-back">

      <h1>{{ comic.titulo }}</h1>

      <p>{{ comic.descripcion }}</p>

      <p>

        <button (click)="marcarFavorito()">

          Seleccionar favorito

        </button>

      </p>

      <p>

        <button (click)="eliminar()" class="delete">

          Eliminar Comic

        </button>

      </p>

      <p>

        <button (click)="editar()" class="edit">

          Editar Comic

        </button>

      </p>

    </div>

  </div>

</div>

Creamos el component **Comics**

**COMICS.COMPONENT.CSS**

body {

  font-family: Arial, Helvetica, sans-serif;

}

\* {

  box-sizing: border-box;

}

/\* Clear floats after the columns \*/

.row:after {

  content: "";

  display: table;

  clear: both;

}

p button.delete {

  background-color: red;

  color: white;

}

p button.edit {

  background-color: blue;

  color: white;

}

p button.save {

  background-color: green;

  color: white;

}

p button.cancel {

  background-color: red;

  color: white;

}

**COMICS.COMPONENT.TS**

import { Component, OnInit, ViewChild, ElementRef } from "@angular/core";

import { Comic } from "./../../models/comic";

@Component({

  selector: "app-comics",

  templateUrl: "./comics.component.html",

  styleUrls: ["./comics.component.css"]

})

export class ComicsComponent implements OnInit {

  public comics: Array<Comic>;

  public favorito: Comic;

  public edicion: number;

  @ViewChild("cajatitulo") cajatitulo: ElementRef;

  @ViewChild("cajaimagen") cajaimagen: ElementRef;

  @ViewChild("cajadescripcion") cajadescripcion: ElementRef;

  guardarComic() {

    var titulo = this.cajatitulo.nativeElement.value;

    var imagen = this.cajaimagen.nativeElement.value;

    var descripcion = this.cajadescripcion.nativeElement.value;

    var comic = new Comic(titulo, imagen, descripcion);

    this.comics[this.edicion] = comic;

    this.edicion = -1;

  }

  cancelarEdicion() {

    this.edicion = -1;

  }

  crearComic() {

    var titulo = this.cajatitulo.nativeElement.value;

    var imagen = this.cajaimagen.nativeElement.value;

    var descripcion = this.cajadescripcion.nativeElement.value;

    var comic = new Comic(titulo, imagen, descripcion);

    this.comics.push(comic);

  }

  //METODO PARA CAPTURAR CUANDO EL HIJO NOS

  //MANDE UN COMIC FAVORITO

  seleccionarFavorito(event) {

    //EL HIJO NOS MANDARA UN EVENTO CON UN COMIC

    // event.comic

    this.favorito = event.comic;

  }

  eliminarComic(event) {

    var index = event.indice;

    this.comics.splice(index, 1);

  }

  editarComic(event) {

    this.edicion = event.indice;

    this.cajatitulo.nativeElement.value = this.comics[event.indice].titulo;

    this.cajaimagen.nativeElement.value = this.comics[event.indice].imagen;

    this.cajadescripcion.nativeElement.value = this.comics[

      event.indice

    ].descripcion;

  }

  constructor() {

    this.edicion = -1;

    this.comics = [

      new Comic(

        "Spiderman",

        "https://images-na.ssl-images-amazon.com/images/I/61AYfL5069L.jpg",

        "Hombre araña"

      ),

      new Comic(

        "Wolverine",

        "https://i.etsystatic.com/9340224/r/il/42f0e1/1667448004/il\_570xN.1667448004\_sqy0.jpg",

        "Lobezno"

      ),

      new Comic(

        "Guardianes de la Galaxia",

        "https://cdn.normacomics.com/media/catalog/product/cache/1/thumbnail/9df78eab33525d08d6e5fb8d27136e95/g/u/guardianes\_galaxia\_guadianes\_infinito.jpg",

        "Yo soy Groot"

      )

    ];

  }

  ngOnInit(): void {}

}

**COMICS.COMPONENT.HTML**

<h1>Comics</h1>

<form #comicForm="ngForm">

  <div>

    <label>Título: </label>

    <input type="text" id="titulo" name="titulo" #cajatitulo />

  </div>

  <div>

    <label>Descripción: </label>

    <input type="text" id="descripcion" name="descripcion" #cajadescripcion />

  </div>

  <div>

    <label>Imagen: </label>

    <input type="text" id="imagen" name="imagen" #cajaimagen />

  </div>

  <div \*ngIf="edicion != -1">

    <p>

      <button class="save" (click)="guardarComic()">

        Guardar datos Comic

      </button>

      <button class="cancel" (click)="cancelarEdicion()">

        Cancelar edición

      </button>

    </p>

  </div>

  <div \*ngIf="edicion == -1">

    <button (click)="crearComic()">Nuevo Comic</button>

  </div>

</form>

<h2 \*ngIf="favorito != null">

  Comic favorito:

  <span style="color:blueviolet">

    {{ favorito.titulo }}

  </span>

</h2>

<div class="row">

  <ng-container \*ngFor="let comic of comics; let i = index">

    <app-comic

      [comic]="comic"

      [index]="i"

      (eliminarComic)="eliminarComic($event)"

      (seleccionarFavorito)="seleccionarFavorito($event)"

      (editarComic)="editarComic($event)"

    ></app-comic>

  </ng-container>

</div>