COMANDOS

# angular-bootcamp - clase 5

1. angular material
2. Forms (dialog)
3. service fake-api

### Aplicar Angular material

1. install angular material

```
`bash
ng add @angular/material
`
```

2. imports app component

```
`typescript
import { MatButtonModule } from '@angular/material/button';;

imports: [
MatButtonModule,
],
`
```

3. add the styles.css

```
`css
@import '@angular/material/prebuilt-themes/rose-red.css';
`
```

4. add in the html

```
`html
<button mat-flat-button>Home</button>
`
```

5. probar con otros componentes (radio,input)

### Mat-tab-group

1. agregar el html en app.component.html

```
`html
<mat-tab-group [selectedIndex]="selectedTab" (selectedTabChange)="navigateToTab($event)">
<mat-tab label="Home">
</mat-tab>
<mat-tab label="About">
</mat-tab>
<mat-tab label="Productos">
</mat-tab>
</mat-tab-group>
<div class="content">
<router-outlet></router-outlet>
</div>
`
```

2. modificar el app.component.ts (quitar componentes de prueba)

```
`typescript
import { MatTabsModule } from '@angular/material/tabs';

imports: [
MatTabsModule,
],

selectedTab = 0;

constructor(private router: Router) {}

ngOnInit() {
this.router.events.subscribe((event) => {
if (event instanceof NavigationEnd) {
this.updateSelectedTab(event.urlAfterRedirects);
}
});
}

navigateToTab(event: any) {
switch (event.index) {
case 0:
this.router.navigate(['/home']);
break;
case 1:
this.router.navigate(['/about']);
break;
case 2:
this.router.navigate(['/productos']);
break;
}
}

private updateSelectedTab(url: string) {
if (url.includes('/home')) {
this.selectedTab = 0;
} else if (url.includes('/about')) {
this.selectedTab = 1;
} else if (url.includes('/productos')) {
this.selectedTab = 2;
}
}

`
```

3. modificar el css

```
`css
.content {
padding: 20px;
}
`
```

4. En el product-detail.component.html

```
`html
<button mat-flat-button (click)="goBack()">Back to Products</button>
`
```

5. En el product-detail.component.ts

```
`typescript
// providers: [ProductoService],
imports: [MatButton],

goBack(): void {
this.router.navigate(['/productos']);
}
`
```

### Dialog

1. Crear componente dialog

```bash
ng g c productos/product-dialog --standalone
```

2. En products.component.html

```html
<button mat-flat-button (click)="openDialog()">Nuevo</button>
<div>Total products: {{productos?.length}}</div>
```

3. En el products.component.ts

```typescript
// inject en el import @angular/core
import { MatDialog } from '@angular/material/dialog';
import { ProductDialogComponent } from './product-dialog/product-dialog.component';
import { MatButtonModule } from '@angular/material/button';

import: […,MatButtonModule]

// quitar el providers: [ProductoService],

dialog = inject(MatDialog);

openDialog() {
this.dialog.open(ProductDialogComponent, {
data: {
animal: 'panda',
},
});
}

```

4. en el product-dialog.component.html

```html
<h2 mat-dialog-title>Nuevo producto</h2>
<mat-dialog-content>
My favorite animal is:
<ul>
<li>
@if (data.animal === 'panda') {
<span>&#10003;</span>
} Panda
</li>
<li>
@if (data.animal === 'unicorn') {
<span>&#10003;</span>
} Unicorn
</li>
<li>
@if (data.animal === 'lion') {
<span>&#10003;</span>
} Lion
</li>
</ul>
</mat-dialog-content>
```

5. en el product-dialog.component.ts

```typescript
import { Component, inject } from '@angular/core';
import {
MAT_DIALOG_DATA,
MatDialogTitle,
MatDialogContent,
} from '@angular/material/dialog';

imports: [MatDialogTitle, MatDialogContent],

data = inject(MAT_DIALOG_DATA);
```

Form

1. Modificar el product-dialog.component.html - crear los campos

```html
<h2 mat-dialog-title>Nuevo producto</h2>
<mat-dialog-content>
<mat-form-field style="padding: 10px;">
<mat-label>Nombre</mat-label>
<input matInput>
</mat-form-field>
<mat-form-field style="padding: 10px;">
<mat-label>Precio</mat-label>
<input matInput>
</mat-form-field>
<mat-form-field style="padding: 10px;">
<mat-label>Descripción</mat-label>
<input matInput>
</mat-form-field>
<mat-form-field style="padding: 10px;">
<mat-label>Imagen Url</mat-label>
<input matInput>
</mat-form-field>
</mat-dialog-content>
```

2. Agregar el form

```html
<form [formGroup]="productForm" (ngSubmit)="onSubmit()">
<mat-dialog-content>
<mat-form-field style="padding: 10px">
<input matInput formControlName="name">
<mat-error *ngIf="productForm.get('name')?.hasError('required')">
El nombre es requerido
</mat-error>
</mat-form-field>
<mat-form-field style="padding: 10px">
<input matInput type="number" formControlName="price">
<mat-error *ngIf="productForm.get('price')?.hasError('required')">
El precio es requerido
</mat-error>
</mat-form-field>
<mat-form-field style="padding: 10px">
<mat-label>Descripción</mat-label>
<input matInput formControlName="description">
</mat-form-field>
<mat-form-field style="padding: 10px">
<mat-label>Imagen Url</mat-label>
```

```
<input matInput formControlName="imgUrl">
</mat-form-field>
</mat-dialog-content>
<mat-dialog-actions align="end">
<button mat-button (click)="onCancel()">Cancelar</button>
<button mat-raised-button color="primary" type="submit" [disabled]="!productForm.valid">
Guardar
</button>
</mat-dialog-actions>
</form>
`
```

3. Agregar imports

```
`typescript
import { MatInputModule } from '@angular/material/input';
import { MatFormFieldModule } from '@angular/material/form-field';
import {
FormBuilder,
FormGroup,
Validators,
ReactiveFormsModule,
} from '@angular/forms';

import { NgIf } from '@angular/common';
import { MatButton } from '@angular/material/button';

constructor(
private fb: FormBuilder,
private dialogRef: MatDialogRef<ProductDialogComponent>,
private productoService: ProductoService
) {
this.productForm = this.fb.group({
name: ['', Validators.required],
price: ['', Validators.required],
description: [''],
imgUrl: [''],
isOferta: [false],
porcentajeOferta: [0],
});
}

onSubmit() {
if (this.productForm.valid) {
const newProduct = {
id: Date.now(),
...this.productForm.value,
};
this.productoService.addProduct(newProduct);
this.dialogRef.close(true);
}
}

onCancel() {
this.dialogRef.close();
}
`
```

4. En products.component.ts

```
`typescript
openDialog() {
const dialogRef = this.dialog.open(ProductDialogComponent, {
data: {
```

```
animal: 'unicorn',
},
});

dialogRef.afterClosed().subscribe((result) => {
if (result) {
this.productos = this.productoService.getProductos();
console.log('Products after dialog:', this.productos);
}
});
}
```

5. en el producto.service.ts

```typescript
addProduct(product: Product) {
this.productos.push(product);
console.log('Products in service:', this.productos);
}
```

### API

```bash
ng g s api-products
```

```typescript
import { HttpClient } from '@angular/common/http';
import { firstValueFrom } from 'rxjs';

export type ProductApi = {
id: number;
title: string;
price: number;
description: string;
category: string;
image: string;
rating: {
rate: number;
count: number;
};
};

private apiUrl = 'https://fakestoreapi.com';

constructor(private readonly http: HttpClient) {}

getAllProducts() {
return firstValueFrom(
this.http.get<ProductApi[]>(${this.apiUrl}/products`)
);
}
```

En el productos.component.ts

```typescript
// en la clase
productosApi: ProductApi[] = [];
// en el constructor
private readonly apiProductsService: ApiProductsService
// en el ngOnInit
this.productosApi = await this.apiProductsService.getAllProducts();
```

`

En el productos.component.html

```html
<hr>
<div class="product-container">
<div class="product-card" *ngFor="let producto of productosApi; let index = index;">
<div class="product-image-container">
<img src="{{producto.image}}" alt="{{producto.title}}" class="product-image">
</div>
<div class="product-details">
<h4 class="product-title">{{producto.title}}</h4>
<div class="price-container">
<p class="product-price">{{producto.price | currency: 'USD'}}</p>
</div>
<p class="product-description">{{producto.description | shorttext}}</p>
</div>
<a [routerLink]="['/productos', producto.id]" class="view-product-button">Ver</a>
</div>
</div>
```