



FACULTAD DE
POSTGRADO
SANTA CRUZ

PRACTICA 3

DESARROLLO FRONTEND y FRAMEWORKS JAVASCRIPT

-- ROMAN FRANCO VALERO CALLE --

PROYECTO 1

Gestor de Tareas

Clase 3: Servicios para Manejo de Datos

- Crear un servicio **TaskService** para manejar tareas en memoria.
- Implementar métodos para agregar, editar y eliminar tareas.
- Conectar el servicio con los componentes.

Uso de arquitectura de N-Capas o Cebolla

```
src/
  app/
    core/
      models/
        product.model.ts
      services/
        cart-repository.interface.ts
      use-cases/
        add-product.usecase.ts
        sales/
          create-sales-order.usecase.ts
          approve-sales-order.usecase.ts
          generate-sales-report.usecase.ts
        billing/
          generate-invoice.usecase.ts
          void-invoice.usecase.ts
          issue-credit-note.usecase.ts
        payments/
          register-payment.usecase.ts
          refund-payment.usecase.ts
          generate-cashflow-report.usecase.ts
      data/
        repositories/
          local-cart.repository.ts
      presentation/
        components/
          cart/
            cart.component.ts
            cart.component.html
        pages/
          home/
            home.component.ts
        state/
          cart.store.ts
          ---sino---
          cart.actions.ts
          cart.reducer.ts
          cart.selectors.ts
          cart.effects.ts
        injection-tokens.ts
        app.component.ts
        app.module.ts
    main.ts
```

Dominio puro (sin Angular)

Interfaz de producto

Abstracción del repositorio

Caso de uso

Implementaciones concretas

Implementación usando localStorage

Todo lo visible y que usa Angular

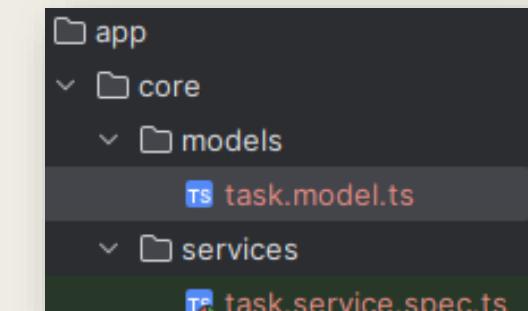
Creación de servicio

```
Sistemas1@Sistemas1 MINGW64 ~/OneDrive/Documentos/terminacion/practicas-ng-dev/to-do-list ((7fe86a5...))
$ ng generate service services/TaskService
CREATE src/app/services/task-service.service.spec.ts (399 bytes)
CREATE src/app/services/task-service.service.ts (149 bytes)
```

Creación de modelo

```
task.model.ts ×

1  export class TaskModel{ Show usages
2      id: number;
3      todo: string;
4      date: Date;
5      time: string;
6      completed: boolean;
7
8      constructor(id: number, todo: string, date: Date, time: string, completed: boolean){
9          this.id = id;
10         this.todo = todo;
11         this.date = date;
12         this.time = time;
13         this.completed = completed;
14     }
15 }
```



PROYECTO 1 ➔ Componente padre - lista

README.md x ts task-list.component.ts x

```
32  }
33  export class TaskListComponent implements OnInit {
34
35  /* para el modal de dialog */
36  readonly todo_m = signal(initialValue: '');
37  readonly taskId_m = signal(initialValue: 0);
38  readonly userId_m = model(initialValue: 0);
39  readonly dialog = inject(MatDialog);
40
41  taskRepository = inject(TaskRepository);
42  taskService = inject(TaskService);
43
44  tasksTodayOk: TaskModel[] = new Array<TaskModel>();
45  tasksTodayNo: TaskModel[] = new Array<TaskModel>();
46  taskYesterdayOk: TaskModel[] = new Array<TaskModel>();
47  taskYesterdayNo: TaskModel[] = new Array<TaskModel>();
48
49  refreshData() { Show usages & Roman Franco Valero Calle
50    this.tasksTodayOk = this.taskRepository.getTasks()?.filter((it: TaskModel) => it.completed);
51    this.tasksTodayNo = this.taskRepository.getTasks()?.filter((it: TaskModel) => !it.completed);
52    this.taskYesterdayOk = this.taskRepository.getTasksYes()?.filter((it: TaskModel) => it.completed);
53    this.taskYesterdayNo = this.taskRepository.getTasksYes()?.filter((it: TaskModel) => !it.completed);
54  }
55
56  ngOnInit() { no usages & Roman Franco Valero Calle
57    this.taskRepository.chargeInitialization();
58    this.refreshData();
59  }
60
61  onCheckedTask(colera: any) { Show usages & Roman Franco Valero Calle
62    let task = colera.task;
63    task.completed = colera.checked;
64    // enviar a modificar tarea completada de hoy en local storage
65    this.taskService.editCheckTask(task);
66    console.log(task);
67    this.refreshData();
68  }
69
70  openDialog(): void { Show usages & Roman Franco Valero Calle
71    const dialogRef = this.dialog.open(DialogOverviewExampleDialog, {
72      data: {taskId_m: this.taskId_m(), todo_m: this.todo_m()},
73    });
74  }
```

README.md < task-list.component.html x

```
1  <h1>LISTA DE TAREAS</h1>
2  <div>
3    <div class="titulo">
4      <h2>HOY</h2>
5      <button mat-mini-fab aria-label="" (click)="openDialog()">
6        <mat-icon fontIcon="add"></mat-icon>
7      </button>
8    </div>
9    <h5>pendiente</h5>
10   @for (item of tasksTodayNo; track item.id) {
11     <app-task-item [task]="item" (editor)="onEditTaskDialog($event)" (delete)="onDeleteTask($event)"
12                   (checked)="onCheckedTask($event)"></app-task-item>
13   } @empty {
14     <small><i>no hay pendientes</i></small>
15   }
16   <h5>Completado</h5>
17   @for (item of tasksTodayOk; track item.id) {
18     <app-task-item [task]="item" (editor)="onEditTaskDialog($event)" (delete)="onDeleteTask($event)"
19                   (checked)="onCheckedTask($event)"></app-task-item>
20   } @empty {
21     <small><i>no hay completados</i></small>
22   }
23 </div>
24 <div>
25   <div class="titulo">
26     <h2>AYER</h2>
27   </div>
28   @if (taskYesterdayNo.length > 0) {
29     <h5>pendiente</h5>
30     @for (item of taskYesterdayNo; track item.id) {
31       <app-task-item [task]="item"
32                     (checked)="onCheckedTask($event)"></app-task-item>
33     }
34   }
35   <h5>Completado</h5>
36   @for (item of taskYesterdayOk; track item.id) {
37     <app-task-item [task]="item"
38                     (checked)="onCheckedTask($event)"></app-task-item>
39   } @empty {
40     <small><i>no hay completados</i></small>
41   }
42 </div>
```

```
ts task-item.component.ts <> task-list.component.html <> task-item.component.html

1 import {ChangeDetectionStrategy, Component, EventEmitter, Input, Output} from '@angular/core';
2 import {MatSlideToggleModule} from '@angular/material/slide-toggle';
3 import {MatCard, MatCardActions, MatCardContent, MatCardHeader} from '@angular/material/card';
4 import {MatIcon} from '@angular/material/icon';
5 import {MatButtonModule} from '@angular/material/button';
6 import {MatCheckboxModule} from '@angular/material/checkbox';
7 import {TaskModel} from '../../../../../core/models/task.model';
8
9 @Component({
  selector: 'app-task-item',
10  imports: [MatSlideToggleModule, MatCard, MatCardHeader, MatCardContent,
11    MatCardActions, MatIcon, MatButtonModule, MatCheckboxModule],
12  templateUrl: './task-item.component.html',
13  styleUrls: ['./task-item.component.css'],
14  changeDetection: ChangeDetectionStrategy.OnPush
15 })
16 export class TaskItemComponent {
17   @Input() task!: TaskModel;
18
19   @Output() checked = new EventEmitter<any>();
20   @Output() editor = new EventEmitter<any>();
21   @Output() delete = new EventEmitter<any>();
22
23   checkedTask(checked: boolean, task: TaskModel) { Show usages & Roman Franco Valero Calle
24     console.log(checked)
25     this.checked.emit({task: task, checked: checked});
26   }
27
28   editTask(task: TaskModel) { Show usages & Roman Franco Valero Calle
29     this.editor.emit({task: task});
30   }
31
32   deleteTask(task: TaskModel) { Show usages & Roman Franco Valero Calle
33     this.delete.emit({task: task});
34   }
35 }
```

```
sk-item.component.ts <> task-list.component.html <> task-item.component.html ×
<mat-card class="tarjeta">
  <mat-card-header></mat-card-header>
  <mat-card-content class="tarjeta-contenido">
    <div class="tarjeta-letras">
      <p>{{ task.todo }}</p>
      <small>{{ task.time }}</small>
    </div>
    <mat-card-actions class="tarjeta-acciones">
      <button mat-mini-fab aria-label="" (click)="editTask(task)">
        <mat-icon aria-hidden="false" aria-label="" fontIcon="edit"></mat-icon>
      </button>
      <button mat-mini-fab aria-label="" (click)="deleteTask(task)">
        <mat-icon>delete</mat-icon>
      </button>
      <mat-checkbox [checked]="task.completed" (change)="checkedTask($event.checked, task)"></mat-checkbox>
    </mat-card-actions>
  </mat-card-content>
</mat-card>
```

PROYECTO 1

Repository

Servicio que usa Repository

```
task.repository.ts x task-item.component.ts <> task-list.component.html <> task-item.component.html

3
4 @Injectable({ Show usages & Roman Franco Valero Calle *
5   providedIn: 'root'
6 })
7 export class TaskRepository{
8   localKey: string = "task-list";
9   localKeyYest: string = "task-list-yesterday";
10  /*https://dummyjson.com/todos*/
11  > fen: any = [ 4 elements... ];
12
13  tasksTodays: any = [
14    {id: 7, todo: 'Abrir Power Point', date: new Date(), time: "02:30 PM", completed: true},
15    {id: 6, todo: 'Instalar Angular', date: new Date(), time: "02:35 PM", completed: true},
16    {id: 5, todo: 'Instalar Angular Material', date: new Date(), time: "03:05 PM", completed: false},
17    {id: 4, todo: 'Sacar Captura de instalacion', date: new Date(), time: "04:30 PM", completed: false},
18  ];
19
20  > taskYesterdays: any = [ 3 elements... ]
21
22  chargeInitialization(): void{ Show usages & Roman Franco Valero Calle
23    this.storeTask(this.tasksTodays);
24    this.saveTasksYes(this.taskYesterdays);
25  }
26
27  getTasks(): TaskModel[] { Show usages & Roman Franco Valero Calle
28    const data = localStorage.getItem(this.localKey);
29    return data ? JSON.parse(data) : [];
30  }
31
32  storeTask(cart: TaskModel[]): Show usages & Roman Franco Valero Calle
33    localStorage.setItem(this.localKey, JSON.stringify(cart));
34  }
35
36  getTasksYes(): TaskModel[] { Show usages & Roman Franco Valero Calle
37    const data = localStorage.getItem(this.localKeyYest);
38    return data ? JSON.parse(data) : [];
39  }
40
41  saveTasksYes(cart: TaskModel[]): Show usages & Roman Franco Valero Calle
42    localStorage.setItem(this.localKeyYest, JSON.stringify(cart));
43  }
44
45  >
```

```
task.repository.ts x task.service.ts <> task-item.component.ts <> task-list.component.html

1 import {inject, Injectable} from '@angular/core';
2 import {TaskRepository} from '../../../../../data/repository/task.repository';
3 import {TaskModel} from '../../../../../models/task.model';
4
5 @Injectable({ Show usages & Roman Franco Valero Calle
6   providedIn: 'root'
7 })
8 export class TaskService {
9   taskRepository = inject(TaskRepository);
10
11  editCheckTask(task: TaskModel): void { Show usages & Roman Franco Valero Calle
12    let lista = this.taskRepository.getTasks();
13    let renewLista = lista.filter(function (item: TaskModel) {
14      if (item.id === task.id) {
15        item.completed = task.completed;
16      }
17      return item;
18    });
19    this.taskRepository.storeTask(renewLista);
20  }
21
22  addTaskToList(task: TaskModel): void { Show usages & Roman Franco Valero Calle
23    let lista = this.taskRepository.getTasks();
24    let maz = lista.reduce((max, p) => p.id > max ? p.id : max, initialValue: 0);
25    task.id = maz + 1;
26    let renewLista = [...lista, task];
27    this.taskRepository.storeTask(renewLista);
28  }
29
30  getTask(id: number): TaskModel { Show usages & Roman Franco Valero Calle
31    let list = this.taskRepository.getTasks();
32    console.log(list);
33    console.log(id);
34    var s = list.filter((item, index) => item.id === id)[0];
35    console.log(s);
36    return list.filter((item) => item.id === id)[0];
37  }
38
39  updateTask(id: number, todo: string): void { Show usages & Roman Franco Valero Calle
40    let lista = this.taskRepository.getTasks();
41    let renewLista = lista.filter(function (item: TaskModel) {
42      if (item.id === id) {
```

Resultado

Marcar como tarea completada

The diagram illustrates the state transition of a task in a to-do list application. It consists of two side-by-side screenshots of the 'LISTA DE TAREAS' (Task List) interface.

Left Screenshot (Initial State):

- HOY (Today):**
 - Pendiente (Pending):** "Instalar Angular Material" (03:05 PM) with edit, delete, and pending icons.
 - Pendiente (Pending):** "Sacar Captura de instalacion" (04:30 PM) with edit, delete, and pending icons.
- Completado (Completed):** "Abrir Power Point" (02:30 PM) with edit, delete, and checked icons.
- Completado (Completed):** "Instalar Angular" (02:35 PM) with edit, delete, and checked icons.

Right Screenshot (Final State):

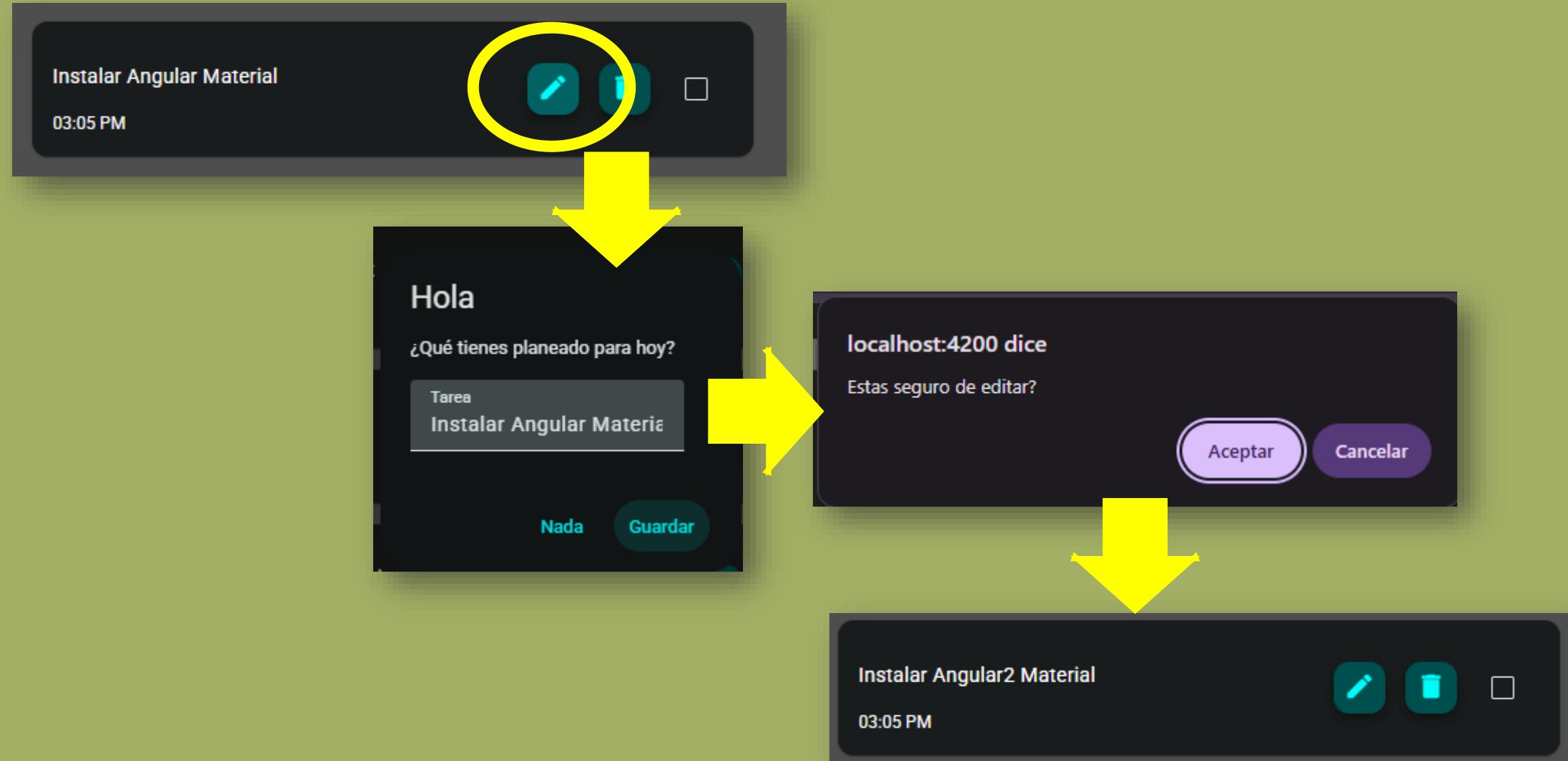
- HOY (Today):**
 - Pendiente (Pending):** "Instalar Angular Material" (03:05 PM) with edit, delete, and pending icons.
 - Completado (Completed):** "Abrir Power Point" (02:30 PM) with edit, delete, and checked icons.
- AYER (Yesterday):**
 - Completado (Completed):** "Sacar Captura de instalacion" (04:30 PM) with edit, delete, and checked icons.
- AYER (Yesterday):**
 - Completado (Completed):** "Desayunar" (06:30 AM) with edit, delete, and checked icons.

A large yellow arrow points from the pending task "Sacar Captura de instalacion" in the left screenshot to the completed task "Sacar Captura de instalacion" in the right screenshot, indicating the transition from pending to completed status.

Agregar una tarea



Editar una tarea



Eliminar una tarea



PROYECTO 2

Catálogo de productos

Clase 3: Servicios y Almacenamiento Local

- Crear un servicio `ProductService` para manejar datos de productos.
- Usar almacenamiento en memoria para gestionar el carrito.
- Mostrar la cantidad de productos en el carrito.

Clean Arquitecture

```
src/
  app/
    core/
      models/
        product.model.ts          # Dominio puro (sin Angular)
      services/
        cart-repository.interface.ts # Interfaz de producto
      use-cases/
        add-product.usecase.ts      # Abstracción del repositorio
        sales/
          create-sales-order.usecase.ts
          approve-sales-order.usecase.ts
          generate-sales-report.usecase.ts
        billing/
          generate-invoice.usecase.ts
          void-invoice.usecase.ts
          issue-credit-note.usecase.ts
        payments/
          register-payment.usecase.ts
          refund-payment.usecase.ts
          generate-cashflow-report.usecase.ts
      data/
        repositories/
          local-cart.repository.ts    # Implementaciones concretas
      presentation/
        components/
          cart/
            cart.component.ts
            cart.component.html
        pages/
          home/
            home.component.ts
        state/
          cart.store.ts
          ---sino---
          cart.actions.ts
          cart.reducer.ts
          cart.selectors.ts
          cart.effects.ts
        injection-tokens.ts          # Tokens de DI para repositorios
        app.component.ts
        app.module.ts
    main.ts
```

Creación de servicio carrito

```
RomSys@RomSys-PC MINGW64 /e/DIPLOMADO DESARROLLO WEB FULL STACK/2. FRAMEWORKS JAVASCRIPT/terminacion/practicas-ng-dev/product-catalog ((78cdcd3...))
$ ng generate service services/cart
CREATE src/app/services/cart.service.spec.ts (363 bytes)
CREATE src/app/services/cart.service.ts (142 bytes)
```

Instalando NgRx para el estado global

```
RomSys@RomSys-PC MINGW64 /e/DIPLOMADO DESARROLLO WEB FULL STACK/2. FRAMEWORKS JAVASCRIPT/terminacion/practicas-ng-dev/product-catalog ((78cdcd3...))
$ ng add @ngrx/store
✓ Determining Package Manager
  > Using package manager: npm
✓ Searching for compatible package version
  > Found compatible package version: @ngrx/store@19.2.0.
✓ Loading package information from registry
✓ Confirming installation
  ⌄ Installing package
```

```
RomSys@RomSys-PC MINGW64 /e/DIPLOMADO DESARROLLO WEB FULL STACK/2. FRAMEWORKS JAVASCRIPT/terminacion/practicas-ng-dev/product-catalog ((78cdcd3...))
$ ng add @ngrx/store-devtools
✓ Determining Package Manager
  > Using package manager: npm
```

```
RomSys@RomSys-PC MINGW64 /e/DIPLOMADO DESARROLLO WEB FULL STACK/2. FRAMEWORKS JAVASCRIPT/terminacion/practicas-ng-dev/product-catalog ((78cdcd3...))
$ npm install @ngrx/effects
added 1 package, and audited 100 packages in 0.29s
```

Creación de servicio carrito

```

  cart.store.ts ×
  1 const repository = new LocaCartRepository();
  2 const addUseCase = new AddProductUseCase();
  3 const removeUseCase = new RemoveProductUseCase();
  4
  5
  6 const initialState = {
  7   items: repository.getCart()
  8 }
  9
 10
 11
 12
 13
 14
 15
 16 export const CartStore = signalStore( Show usages  Roman Franco Valero Calle
 17   {providedIn: 'root'},
 18   withState(initialState),/*Inicializa el carrito con los productos almacenados en el local storage */
 19   withComputed(({items}) => ({
 20     /* variables globales */
 21     /* cantidad de productos en carrito */
 22     totalProdsInCart: computed(() => items().length),
 23     /* cantidad total de cantidades carrito */
 24     totalCartProdsInCart: computed(() => items().reduce((sum, p) => sum + p.quantity, initialValue: 0)),
 25     /* si el items se actualiza, esto crea una cadena de reacciones y afecta en los lugares que se usa items */
 26     totalCartUSD: computed(() => items().reduce((sum, p) => sum + p.price * p.quantity, initialValue: 0)),
 27   })),
 28   withMethods(store => ({
 29     /* Son las funciones del servicio para usarlo desde fuera */
 30     addProduct(product: ProductModel) {
 31       const updatedCart = addUseCase.execute(store.items(), product);
 32       /*patchState(store, (state) => ( { items: { ...state.items, product } } ) );*/
 33       patchState(store, (_ ) => ({items: updatedCart}));
 34       repository.saveCart(updatedCart);
 35     },
 36     removeProduct(productId: number) {
 37       //const updatedCart = store.items().filter(p => p.id !== productId);
 38       const updatedCart = removeUseCase.execute(store.items(), productId);
 39       // modifica el estado
 40       /*patchState(store, (state) => ({items: updatedCart}));*/
 41       patchState(store, (_ ) => ({items: updatedCart}));
 42       // modifica en el localstorage
 43       repository.saveCart(updatedCart);
 44     },
 45     clearCart() {
 46       patchState(store, (_ ) => ({items: []}));
 47       repository.saveCart([]);
 48     },
 49   }));

```

NgRx/Signal

Repository

```

  cart.store.ts ×  loca-cart.repository.ts ×
  1 import {ProductModel} from '../../../../../core/models/product.model';
  2
  3
  4 export class LocaCartRepository{ Show usages  Roman Franco Valero Calle
  5   localKey: string = "cart";
  6
  7   getCart(): ProductModel[] { Show usages  Roman Franco Valero Calle
  8     const data = localStorage.getItem(this.localKey);
  9     return data ? JSON.parse(data) : [];
 10   }
 11
 12   saveCart(cart: ProductModel[]) { Show usages  Roman Franco Valero Calle
 13     localStorage.setItem(this.localKey, JSON.stringify(cart));
 14   }
 15

```

```

product-list.component.ts x  cart.store.ts  loca-cart.repository.ts  product-list.component.html
1  import {Component, inject, OnInit} from '@angular/core';
2  import {ProductCardComponent} from './product-card/product-card.component';
3  import {MatSlideToggleModule} from '@angular/material/slide-toggle';
4  import {MatButtonModule} from '@angular/material/button';
5  import {ProductModel} from '../../../../../core/models/product.model';
6
7  import {CartStore} from '../../../../../state/cart.store';
8  import {MatIcon} from '@angular/material/icon';
9
10 @Component({
11   selector: 'app-product-list-component',
12   imports: [MatSlideToggleModule, MatButtonModule, ProductCardComponent],
13   templateUrl: './product-list.component.html',
14   styleUrls: ['./product-list.component.css']
15 })
16 export class ProductListComponent implements OnInit {
17   listProducts: any = [ 20 elements... ];
18   /*https://dummyjson.com/products*/
19   listProductsGlobal: any = {
20     "products": [ 30 elements... ], "total": 194, "skip": 0, "limit": 30
21   }
22   listProducts: Array<ProductModel> = this.listProductsGlobal.products;
23
24   /* En lugar del servicio usamos el store que es lo mismo */
25   /*cartService = inject(CartService);*/
26   cartStore = inject(CartStore);
27
28   ngOnInit() {
29     this.listProducts = this.listProducts.filter(function (item, index) {
30       item.quantity = 0;
31       return item;
32     })
33   }
34
35   cartItemAdd(item: ProductModel) {
36     /*alert("Se agregó el producto " + item.title + " al carrito de compras");*/
37     this.cartStore.addProduct(item);
38   }
39 }

```

Componente Padre

```

product-list.component.html x  product-list.component.ts
1  <div>
2    <h1>LISTA DE PRODUCTOS</h1>
3  </div>
4  <div class="lista-tarjetas">
5    @for (item of listProducts; track item.id) {
6      <app-product-card
7        [item]="item" class="tarjeta"
8        (addedToCart)="cartItemAdd($event)">
9      </app-product-card>
10    </div>
11  </div>

```

Componente Hijo

product-list.component.html **product-card.component.ts** product-list.component.ts

```

import {Component, EventEmitter, Input, output, Output} from '@angular/core';
import {MatSlideToggleModule} from '@angular/material/slide-toggle';
import {MatCard, MatCardContent, MatCardFooter} from '@angular/material/card';
import {MatIcon} from '@angular/material/icon';
import {MatButtonModule} from '@angular/material/button';
import {RouterModule} from '@angular/router';
import {ProductModel} from '../../../../../core/models/product.model';

@Component({
  selector: 'app-product-card',
  imports: [MatSlideToggleModule, MatCard, MatIcon, MatButtonModule, RouterModule],
  templateUrl: './product-card.component.html',
  styleUrls: ['./product-card.component.css']
})
export class ProductCardComponent {
  @Input() item!: ProductModel;
  @Output() addedToCart = new EventEmitter<ProductModel>();

  addToCart(item: ProductModel) {
    this.addedToCart.emit(item);
  }
}

```

product-list.component.html product-card.component.html product-list.component.ts

```

<mat-card class="tarjeta">
  <div class="tarjeta-contenido">
    <div class="tarjeta-imagen">
      <!---->
      
    </div>
    <div class="tarjeta-prod-title">
      <p>{{ item.title }}</p>
    </div>
  </div>
  <div class="tarjeta-pie">
    <div class="tarjeta-pie-letras">
      <div>
        {{ item.category }}
      </div>
      <div>
        $ {{ item.price }}
      </div>
    </div>
    <div class="tarjeta-pie-botones">
      <button mat-flat-button aria-label="" (click)="addToCart(item)">
        <mat-icon fontIcon="shopping_cart"></mat-icon>
        Agregar
      </button>
    </div>
    <!-- <a [routerLink]="/details", item.id]" routerLinkActive="route-->
  </div>
</mat-card>

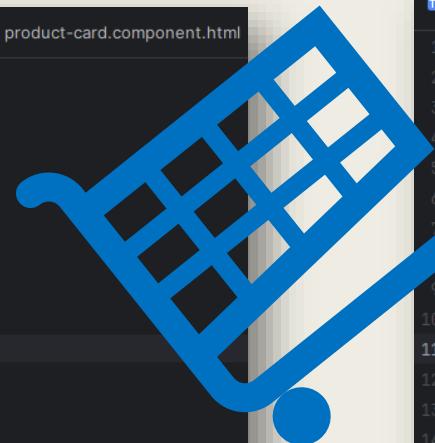
```

Componente Carrito de Compras

```

ts cart.component.ts x <> product-list.component.html <> product-card.component.html
1 @Component({
2   selector: 'app-cart',
3   imports: [
4     /*MatMiniFabButton,*/
5     MatIcon,
6     MatButton,
7     /*MatBadge*/
8   ],
9   templateUrl: './cart.component.html',
10  styleUrls: ['./cart.component.css'],
11 })
12 export class CartComponent{
13   cartStore = inject(CartStore);
14
15   /* Sincroniza o realiza un enlace con el estado inicial y global */
16   get items() { Show usages & Roman Franco Valero Calle
17     return this.cartStore.items();
18   }
19
20   get totalPrdsInCart(){ Show usages & Roman Franco Valero Calle
21     return this.cartStore.totalPrdsInCart();
22   }
23
24   /* Sincroniza o realiza un enlace con el total */
25   get totalUSD() { Show usages & Roman Franco Valero Calle
26     return this.cartStore.totalCartUSD();
27   }
28
29   addSampleProduct(product: ProductModel) { no usages & Roman Franco Valero Calle
30     //const sample: ProductModel = { id: 1, title: 'Producto Demo', price
31     this.cartStore.addProduct(product);
32   }
33
34   removeProduct(id: number) { Show usages & Roman Franco Valero Calle
35     this.cartStore.removeProduct(id);
36   }
37
38   clearCart() { Show usages & Roman Franco Valero Calle
39     this.cartStore.clearCart();
40   }
41
42 }

```



```

ts cart.component.ts x <> cart.component.html x <> product-list.component.html <> product-card.component.html
1 <h2>Carrito</h2>
2 @if (items.length) {
3   <div>
4     @for (item of items; track item.id) {
5       <div class="card-cart">
6         
7         <div class="card-cart-content">
8           <p>{{ item.title }}</p>
9           <div class="card-cart-details">
10             <div>
11               <p>{{ item.quantity }} x ${{ item.price }}<!-- = $ {{ item.price * item.quantity }}--&gt;
12             &lt;/div&gt;
13             &lt;div&gt;
14               &lt;button mat-button aria-label="Example icon button with a menu icon"
15                 (click)="removeProduct(item.id)"&gt;
16                 &lt;mat-icon&gt;delete&lt;/mat-icon&gt;
17               &lt;/button&gt;
18             &lt;/div&gt;
19           &lt;/div&gt;
20           &lt;!--&lt;span matBadge="{{ item.quantity }}" matBadgePosition="after" matBadgeColor="accent"&gt;
21             {{ item.quantity }}
22           &lt;/span&gt;--&gt;
23         &lt;/div&gt;
24       &lt;/div&gt;
25     }
26     &lt;p&gt;&lt;strong&gt;Total: ${{ totalUSD }}&lt;/strong&gt;&lt;/p&gt;
27     &lt;p&gt;&lt;strong&gt;Total Products: {{ totalPrdsInCart }}&lt;/strong&gt;&lt;/p&gt;
28
29     &lt;div class="tarjeta-pie-botones"&gt;
30       &lt;button mat-flat-button&gt;
31         &lt;mat-icon aria-hidden="false" aria-label="Example home icon" fontIcon="launch"&gt;&lt;/mat-icon&gt;
32         Ir carrito
33       &lt;/button&gt;
34       &lt;button mat-button (click)="clearCart()"&gt;Vaciar carrito&lt;/button&gt;
35     &lt;/div&gt;
36   &lt;/div&gt;
37 } @else {
38   &lt;p&gt;El carrito está vacío.&lt;/p&gt;
39 }
</pre>

```

localhost:4200

LISTA DE PRODUCTOS

0 (0)



Essence Mascara
Lash Princess
beauty \$ 9.99

 Agregar



Eyeshadow Palette
with Mirror
beauty \$ 19.99

 Agregar



Powder Canister
beauty \$ 14.99

 Agregar



Red Lipstick
beauty \$ 12.99

 Agregar



Red Nail Polish
beauty \$ 8.99

 Agregar



Calvin Klein CK One
fragrances \$ 49.99

 Agregar



Chanel Coco Noir Eau
De
fragrances \$ 129.99

 Agregar



Dior J'adore
fragrances \$ 89.99

 Agregar



Dolce Shine Eau de
fragrances \$ 69.99

 Agregar



Gucci Bloom Eau de
fragrances \$ 79.99

 Agregar



Annibale Colombo
Bed



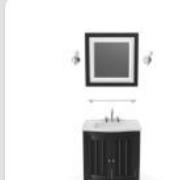
Annibale Colombo
Sofa



Bedside Table African
Cherry



Knoll Saarinen
Executive Conference
Chair



Wooden Bathroom
Sink With Mirror

Página con sección de carrito

localhost:4200

LISTA DE PRODUCTOS

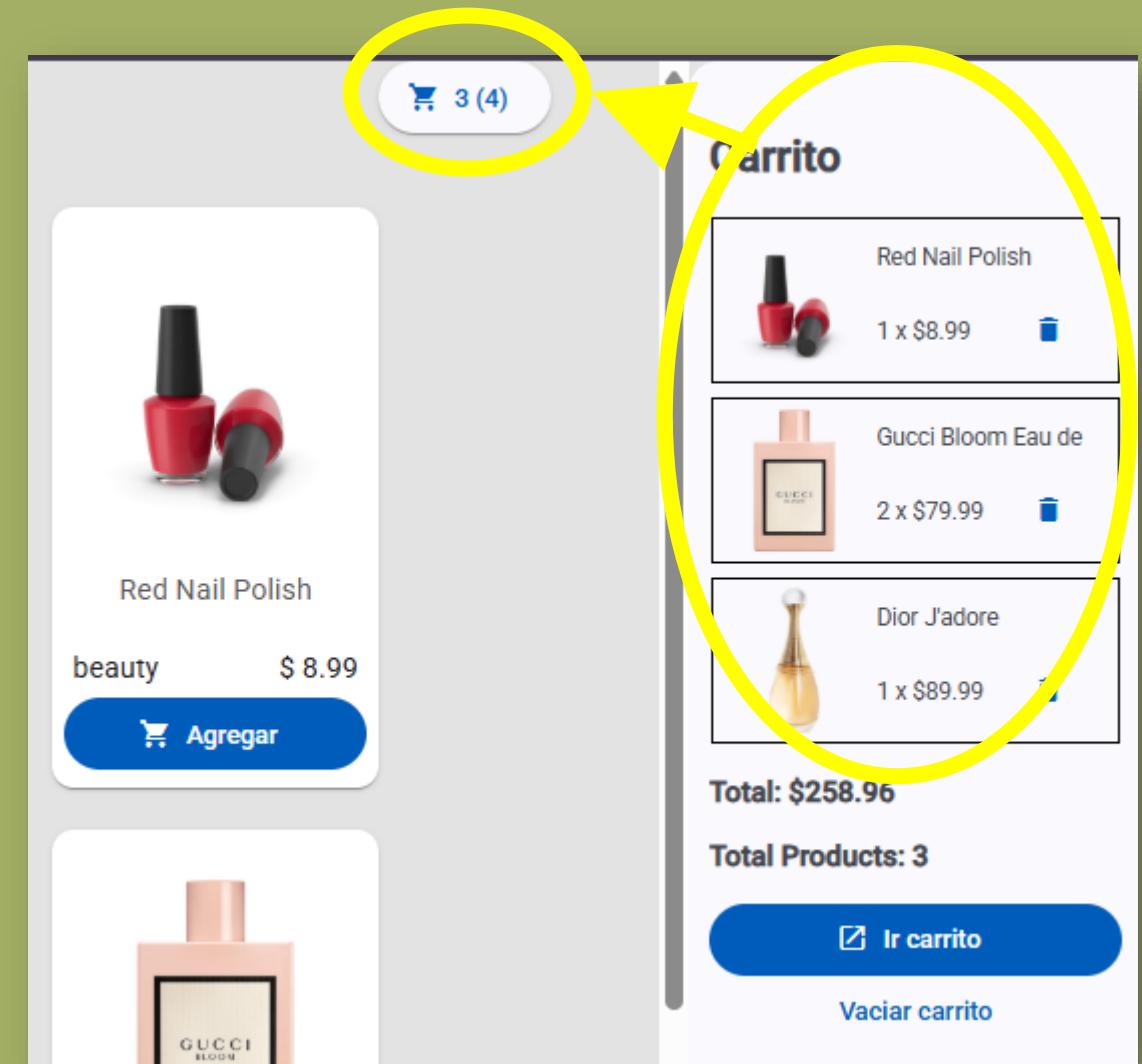
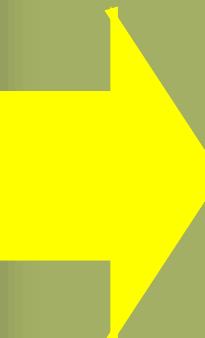
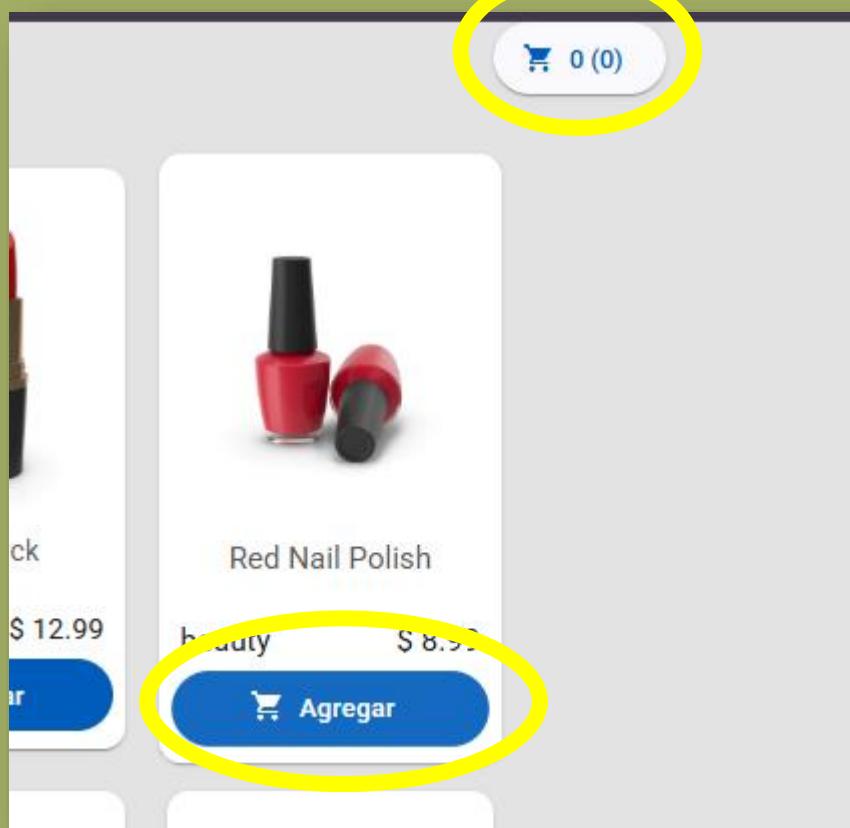
0 (0)

Imagen	Nombre	Categoría	Precio	Opciones
	Essence Mascara Lash Princess	beauty	\$ 9.99	Agregar
	Eyeshadow Palette with Mirror	beauty	\$ 19.99	Agregar
	Powder Canister	beauty	\$ 14.99	Agregar
	Red Lipstick	beauty	\$ 12.99	Agregar
	Red Nail Polish	beauty	\$ 8.99	Agregar
	Calvin Klein CK One	fragrances	\$ 49.99	Agregar
	Chanel Coco Noir Eau De	fragrances	\$ 129.99	Agregar
	Dior J'adore	fragrances	\$ 89.99	Agregar
	Dolce Shine Eau de	fragrances	\$ 69.99	Agregar
	Gucci Bloom Eau de	fragrances	\$ 79.99	Agregar
	Annibale Colombo Bed			
	Annibale Colombo Sofa			
	Bedside Table African Cherry			
	Knoll Saarinen Executive Conference Chair			
	Wooden Bathroom Sink With Mirror			

Carrito

El carrito está vacío.

Agregar productos a carrito



Vista de productos en carrito

localhost:4200

LISTA DE PRODUCTOS

3 (4)



Essence Mascara Lash Princess
beauty \$ 9.99

[Aregar](#)



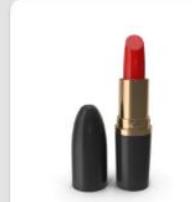
Eyeshadow Palette with Mirror
beauty \$ 19.99

[Aregar](#)



Powder Canister
beauty \$ 14.99

[Aregar](#)



Red Lipstick
beauty \$ 12.99

[Aregar](#)



Red Nail Polish
beauty \$ 8.99

[Aregar](#)



Calvin Klein CK One
fragrances \$ 49.99

[Aregar](#)



Chanel Coco Noir Eau De
fragrances \$ 129.99

[Aregar](#)



Dior J'adore
fragrances \$ 89.99

[Aregar](#)



Dolce Shine Eau de
fragrances \$ 69.99

[Aregar](#)



Gucci Bloom Eau de
fragrances \$ 79.99

[Aregar](#)



Annibale Colombo Bed
furniture \$ 1800.00



Annibale Colombo Sofa
furniture \$ 2400.00



Bedside Table African Cherry
furniture \$ 200.00



Knoll Saarinen Executive Conference Chair
furniture \$ 400.00



Wooden Bathroom Sink With Mirror
furniture \$ 700.00

Carrito



Red Nail Polish
1 x \$8.99



Gucci Bloom Eau de
2 x \$79.99



Dior J'adore
1 x \$89.99

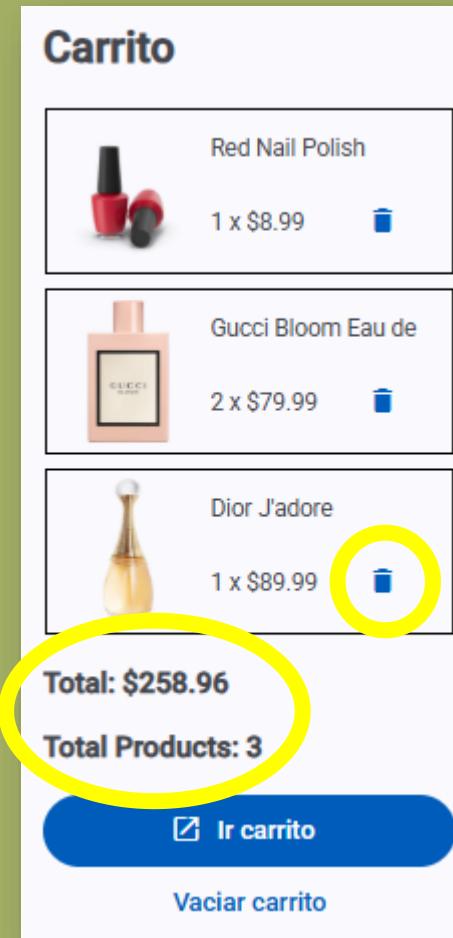
Total: \$258.96

Total Products: 3

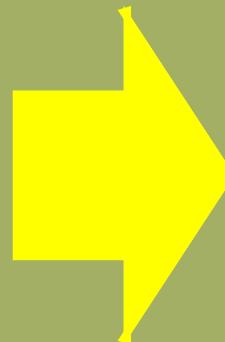
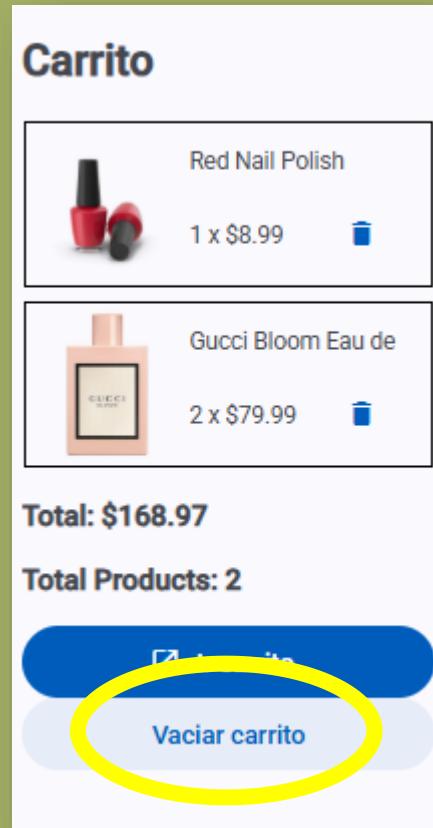
[Ir carrito](#)

[Vaciar carrito](#)

Quitar productos en carrito



Vaciar el carrito



PROYECTO 3

Blog Personal

Clase 3: Servicios para Manejo de Datos

- Crear un servicio **PostService** para manejar publicaciones.
- Implementar métodos para agregar, editar y eliminar publicaciones.
- Conectar el servicio con los componentes.

Clean Arquitecture

```
src/
  app/
    core/
      models/
        product.model.ts          # Dominio puro (sin Angular)
      services/
        cart-repository.interface.ts # Interfaz de producto
      use-cases/
        add-product.usecase.ts      # Abstracción del repositorio
        sales/
          create-sales-order.usecase.ts
          approve-sales-order.usecase.ts
          generate-sales-report.usecase.ts
        billing/
          generate-invoice.usecase.ts
          void-invoice.usecase.ts
          issue-credit-note.usecase.ts
        payments/
          register-payment.usecase.ts
          refund-payment.usecase.ts
          generate-cashflow-report.usecase.ts
      data/
        repositories/
          local-cart.repository.ts    # Implementaciones concretas
      presentation/
        components/
          cart/
            cart.component.ts
            cart.component.html
        pages/
          home/
            home.component.ts
        state/
          cart.store.ts
          ---sino---
          cart.actions.ts
          cart.reducer.ts
          cart.selectors.ts
          cart.effects.ts
    injection-tokens.ts          # Tokens de DI para repositorios
    app.component.ts
    app.module.ts
  main.ts
```

Repository

```
1  import {Injectable} from '@angular/core';
2  import {PostModel} from '../../../../../core/models/post.model';
3
4  @Injectable({
5      providedIn: 'root'
6  })
7  export class PostRepository {
8      localKey: string = "post-list";
9      response: any = [{"status": "ok"}];
10     postList: any = this.response.articles;
11
12     storePost(post: PostModel[]) {
13         localStorage.setItem(this.localKey, JSON.stringify(post));
14     }
15
16     chargeInitialization(): void {
17         this.storePost(this.postList);
18     }
19
20     getPost(): PostModel[] {
21         const data = localStorage.getItem(this.localKey);
22         return data ? JSON.parse(data) : [];
23     }
24 }
```

```
8  export class PostService {
9    postRepository = inject(PostRepository);
10
11    createRandomString(length: number) { Show usages  ↳ Roman Franco Valero Calle
12      const chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
13      let result = "";
14      for (let i = 0; i < length; i++) {
15        result += chars.charAt(Math.floor(Math.random() * chars.length));
16      }
17      return result;
18    }
19
20    addPostToList(task: PostModel): void { Show usages  ↳ Roman Franco Valero Calle
21      let lista = this.postRepository.getPost();
22      task.url = this.createRandomString( length: 10 );
23      let renewLista = [...lista, task];
24      this.postRepository.storePost(renewLista);
25    }
26
27    getPost(url: string): PostModel { Show usages  ↳ Roman Franco Valero Calle
28      let list = this.postRepository.getPost();
29      console.log(list);
30      var s = list.filter((item, index) => item.url === url)[0];
31      console.log(s);
32      return list.filter((item) => item.url === url)[0];
33    }
34
35    updatePost(post: PostModel): void { no usages  ↳ Roman Franco Valero Calle
36      let lista = this.postRepository.getPost();
37      let renewLista = lista.filter(function (item: PostModel) {
38        if (item.url === post.url) {
39          item = post;
40        }
41        return item;
42      });
43      this.postRepository.storePost(renewLista);
44    }
45
46    removePostFromList(url: string): void { Show usages  ↳ Roman Franco Valero Calle
47      let lista = this.postRepository.getPost();
48      let newList = lista.filter((item: PostModel) => item.url !== url);
49      this.postRepository.storePost(newList);
50    }
51
52    getPostList(): PostModel[] { Show usages  ↳ Roman Franco Valero Calle
53      return this.postRepository.getPost();
54    }
55  }
```

```
1  s post.model.ts  x
2
3  export interface PostModel{
4      source: {
5          id: string;
6          name: string;
7      };
8      author: string;
9      title: string;
10     description: string;
11     url: string;
12     urlToImage: string;
13     publishedAt: string;
14     content: string;
15 }
```

Componente padre

```

  post-list-component.component.ts ×  post.model.ts      post.service.ts
30  export class PostListComponentComponent implements OnInit {
32      postRepository = inject(PostRepository);
33
34      readonly todo_m = signal( initialValue: '' );
35      readonly taskId_m = signal( initialValue: 0 );
36      readonly showCreate = signal( initialValue: false );
37
38      postList: PostModel[] = [];
39      reloadData() { Show usages  ↳ Roman Franco Valero Calle
40          this.postList = this.postRepository.getPost();
41      }
42
43 ①↑  ngOnInit() {  no usages  ↳ Roman Franco Valero Calle
44      this.postRepository.chargeInitialization();
45      this.reloadData();
46  }
47
48      showCreateCom(){ Show usages  ↳ Roman Franco Valero Calle
49          this.showCreate.set( value: true );
50      }
51
52      cancelArticle(){ Show usages  ↳ Roman Franco Valero Calle
53          this.showCreate.set( value: false );
54      }
55
56      onArticleSubmitted(article: PostModel) { Show usages  ↳ Roman Fra
57          this.postService.addPostToList(article);
58          this.reloadData();
59      }
60
61      onEditPost(data: any){ Show usages  ↳ Roman Franco Valero Calle
62          let post = this.postService.getPost(data.url)
63
64      }
65
66      onDeletePost(data: any) { Show usages  ↳ Roman Franco Valero Calle
67          if(window.confirm( message: "Estas seguro?")){
68              console.log(data);
69              this.postService.removePostFromList(data.url);
70              this.reloadData();
71          }
72      }

```

```

  post.model.ts ×  post.service.ts      post-list-component.component.html ×  post.repository.ts
<div class="titulo">
    <h1>Blog de Noticias</h1>
</div>
<button mat-mini-fab aria-label="" (click)="showCreateCom()">
    <mat-icon fontIcon="add"></mat-icon>
</button>

@if (showCreate()) {
    <app-form-post (submitArticle)="onArticleSubmitted($event)" (cancelArticle)="cancelArticle()"></app-form-post>
}

<div class="lista-tarjetas">
    @for (item of postList; track item.url) {
        <app-post-card [item]="item" (onDeletePost)="onDeletePost($event)" (onEditPost)="onEditPost($event)"></app-post-card>
    }
    <h1> Comentarios </h1>
    <div class="comentarios">
        <div class="mi-foto">
            
        </div>
        <div class="comentario-x">
            <p>Un comentario de producto, también conocido como review o reseña, es una evaluación escrita por un consumidor sobre un producto o servicio. Estos comentarios ofrecen información valiosa sobre las experiencias de otros compradores, ayudando a otros potenciales compradores a tomar decisiones informadas. </p>
        </div>
        <div class="tu-comentario">
            <mat-form-field appearance="outline" class="caja-comentario">
                <mat-label>Publicar mi comentario</mat-label>
                <input matInput placeholder="Mi comentario..." class="caja-comentario">
                <!--<mat-icon matSuffix>sentiment_very_satisfied</mat-icon>-->
                <!--<mat-hint>Hint</mat-hint>-->
            </mat-form-field>
            <button mat-fab aria-label="Example icon button with a delete icon">
                <mat-icon>send</mat-icon>
            </button>
        </div>
    </div>

```

Componente hijo

```

 1  import {Component, EventEmitter, Input, Output} from '@angular/core';
 2  import {DatePipe} from "@angular/common";
 3  import {MatCard, MatCardContent, MatCardFooter} from "@angular/material/card";
 4  import {MatIcon} from '@angular/material/icon';
 5  import {MatMiniFabButton} from '@angular/material/button';
 6
 7  @Component({
 8    selector: 'app-post-card',
 9    imports: [
10      DatePipe,
11      MatCard,
12      MatCardContent,
13      MatCardFooter,
14      MatIcon,
15      MatMiniFabButton
16    ],
17    templateUrl: './post-card.component.html',
18    styleUrls: ['./post-card.component.css']
19  })
20  export class PostCardComponent {
21    @Input() item: any;
22
23    @Output() onDeletePost = new EventEmitter<any>();
24    @Output() onEditPost = new EventEmitter<any>();
25
26    editPost(item: any) { Show usages  Roman Franco Valero Calle
27      this.onEditPost.emit(item);
28    }
29
30    deletePost(item: any){ Show usages  Roman Franco Valero Calle
31      this.onDeletePost.emit(item);
32    }
33  }

```

```

 1  <mat-card class="tarjeta">
 2    <div class="tarjeta-cabeza">
 3      <div class="aside-izq-imagen">
 4        
 7        <div class="aside-titulo">
 8          <h1>{{ item.title }}</h1>
 9        </div>
10        <div class="aside-texto">
11          <h2>
12            {{ item.description }}
13          </h2>
14          <h1>
15            {{ item.publishedAt|date }}
16          </h1>
17          <div>
18            Autor: {{ item.author }}
19          </div>
20        </div>
21      </div>
22    </div>
23    <mat-card-content class="tarjeta-contenido">
24      <div class="tarjeta-descripcion-principal">
25        <div class="contenido-descripcion">
26        </div>
27        <p>{{ item.content }}</p>
28      </div>
29    </mat-card-content>
30    <mat-card-footer class="tarjeta-pie">
31      <div class="tarjeta-pie-botones">
32        <button mat-mini-fab aria-label="" >
33          <mat-icon aria-hidden="false" fontIcon="edit" (click)="editPost(item)"></mat-icon>
34        </button>
35        <button mat-mini-fab aria-label="" (click)="deletePost(item)">
36          <mat-icon>delete</mat-icon>
37        </button>
38      </div>
39    </mat-card-footer>
40  </mat-card>

```

Página principal

localhost:4200

Blog de Noticias

+



La frenética semana de Trump en negociaciones de paz da una pista de lo que realmente quiere

Después de siete días de intensos intercambios diplomáticos, tenemos una idea más clara de la ideología global del presidente de Estados Unidos y de los límites de su arte de la acuerdo.

May 16, 2025

Autor: BBC News

La centralidad de Trump en la política exterior estadounidense también se ha hecho evidente esta semana. Esto es más que una simple obviedad. Se puso de manifiesto la falta de implicación de otras partes del gobierno estadounidense que... [+1043 chars]

Edit



Moodys rebaja la calificación crediticia de EE. UU. por el aumento de la deuda

Moodys afirmó que las sucesivas administraciones estadounidenses no habían logrado revertir los crecientes déficits y los costos de los intereses.

May 16, 2025

Autor: BBC News

Creación de una publicación

Blog de Noticias

+

ID Fuente

Nombre Fuente

Autor

Título

Descripción

URL

Imagen (URL)

Publicado en
dd/mm/aaaa --:--

Contenido

Cancelar

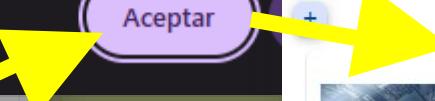
Publicar

Eliminación

Blog de Noticias



localhost:4200 dice
Estás seguro?

Aceptar

La frecuencia de las
negociaciones de paz da una pista de
lo que realmente quiere

Después de siete días de intensos intercambios
diplomáticos, tenemos una idea más clara de la
ideología global del presidente de Estados Unidos y
de los límites de su arte de la acuerdo.

May 16, 2025

Autor: BBC News

La centralidad de Trump en su política exterior estadounidense también se ha hecho evidente esta semana. Esto es más que una simple obviedad. Se puso de manifiesto la falta de implicación de otros miembros del gobierno estadounidense que... [+1043 chars]



Moodys rebaja la calificación crediticia de EE. UU. por el aumento de la deuda

Moodys afirmó que las sucesivas administraciones estadounidenses no habían logrado revertir los crecientes déficits y los costos de los intereses.

May 16, 2025

Autor: BBC News

EE. UU. ha perdido su última calificación crediticia triple A otorgada por una importante agencia de calificación crediticia tras ser rebajada por Moodys, que alegó el aumento de la deuda federal en los últimos años. década. Al rebajar la calificación de EE. UU. a 'Aa1',... [+442 caracteres]



De Aberdeen a Penzance: Fin de la ruta ferroviaria directa más larga del Reino Unido

Los pasajeros disfrutan del viaje de 1240
kilómetros de Aberdeen a Penzance por última vez.

Blog de Noticias



Moodys rebaja la calificación crediticia de EE. UU. por el aumento de la deuda

Moodys afirmó que las sucesivas administraciones estadounidenses no habían logrado revertir los crecientes déficits y los costos de los intereses.

May 16, 2025

Autor: BBC News

EE. UU. ha perdido su última calificación crediticia triple A otorgada por una importante agencia de calificación crediticia tras ser rebajada por Moodys, que alegó el aumento de la deuda federal en los últimos años. década. Al rebajar la calificación de EE. UU. a 'Aa1',... [+442 caracteres]



De Aberdeen a Penzance: Fin de la ruta ferroviaria directa más larga del Reino Unido

Los pasajeros disfrutan del viaje de 1240
kilómetros de Aberdeen a Penzance por última vez.

May 16, 2025

Autor: BBC News

Aimee, que vive en Derby, ama los trenes desde niña. "Cuando era pequeña, mi abuelo vivía en Wigginton, cerca de Tamworth. Soy uno de 21 primos y solíamos reunirnos y, obviamente, cuando... [+465 chars]



Sleep Token: Even in Arcadia es el primer álbum número uno de la banda en el Reino Unido

El misterioso grupo de rock alcanza el primer puesto con su último álbum, Even in Arcadia Arcadia.