



INFORME DE GUÍA PRÁCTICA

I. PORTADA

Tema:	Componentes de angular
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	Sexto - A
Alumnos participantes:	Pazmiño Cordova Bryan Javier Fiallos Barros Edison Javier Mazabanda Pullutaxi Lenin Giovanni Poaquiza Minigauno Jefferson Stalin
Asignatura:	Aplicaciones Web y Móviles
Docente:	Ing. Dennis Chicaiza

II. INFORME DE GUÍA PRÁCTICA

2.1 Objetivos

General:

Comprender los fundamentos y requerimientos del desarrollo de aplicaciones web

Específicos:

- Implementar formularios usando las librerías que angular ofrece para su manejo.
- Implementar estilos usando una estructura minimalista y actual.
- Implementar servicios en angular para iconos y el menu

2.2 Modalidad

Presencial

2.3 Tiempo de duración

Presenciales: 8

No presenciales: 0

2.4 Instrucciones

- Lea el material y la bibliografía proporcionada en clases respecto al tema de la guía
- Revisar la documentación oficial del framework utilizado en clases

2.5 Listado de equipos, materiales y recursos

- **Inteligencia artificial, TAC**
- **Computador**
- **Internet**

TAC (Tecnologías para el Aprendizaje y Conocimiento) empleados en la guía práctica:

- ☒ Plataformas educativas
- ☒ Simuladores y laboratorios virtuales
- ☐ Aplicaciones educativas
- ☐ Recursos audiovisuales
- ☐ Gamificación
- ☒ Inteligencia Artificial

Otros (Especifique): _____

2.6 Actividades por desarrollar

- Desarrollar un prototipo de aplicación web en la cual se pueda evidenciar el uso de módulos y componentes y rutas en angular.



- Desarrollar un prototipo de aplicación web en la cual se pueda evidenciar el uso de formularios, servicios y guards en angular.

2.7 Resultados obtenidos

Se obtiene el conocimiento de cómo manejar formularios en angular mediante servicios. También se tiene el resultado de una buena UI, intuitiva y fácil de usar por parte de los usuarios.

2.8 Habilidades blandas empleadas en la práctica

- ☒ Liderazgo
- ☒ Trabajo en equipo
- ☒ Comunicación asertiva
- ☐ La empatía
- ☒ Pensamiento crítico
- ☒ Flexibilidad
- ☒ La resolución de conflictos
- ☐ Adaptabilidad
- ☐ Responsabilidad

2.9 Conclusiones

En el desarrollo del prototipo se implementaron seis módulos. Se trabajó en la estructura visual de los formularios, integrando distintos atributos y permitiendo una visualización dinámica dentro de un contenedor, lo que facilitó la organización y comprensión del contenido. Esta actividad permitió aplicar conceptos clave como módulos, formularios y servicios en Angular.

2.10 Recomendaciones

Se recomienda completar la lógica funcional de los formularios e integrar los servicios y guards para asegurar una aplicación más robusta, interactiva y segura.

2.11 Anexos

Menú vendedores

Vendedores						Sistema de Ventas	
Agregar Vendedor							
Nombre						Acciones	
Juan Pérez							
María Gómez							
Carlos López							



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE Elige un elemento.
CICLO ACADÉMICO: MARZO – JULIO 2025



```
app > sellers > components > list-seller > list-seller.component.ts > ListSellerComponent

import { Component, signal } from '@angular/core';
import { CommonModule } from '@angular/common';
import { SellerService } from '../../services/seller.service';
import { SellerFormComponent } from '../seller-form/seller-form.component';
import { ConfirmationModalComponent } from '../../shared/components/confirmation-modal/confirmation-modal.component';
import { MatTableModule } from '@angular/material/table';
import { MatButtonModule } from '@angular/material/button';
import { MatIconModule } from '@angular/material/icon';

@Component({
  selector: 'app-list-seller',
  imports: [
    CommonModule,
    SellerFormComponent,
    ConfirmationModalComponent,
    MatTableModule,
    MatButtonModule,
    MatIconModule
  ],
  templateUrl: './list-seller.component.html',
  styleUrls: ['./list-seller.component.css']
})
export class ListSellerComponent {
  sellers = signal<any[]>([]);
  selectedSeller = signal<any | null>(null);
  isEditModalOpen = signal(false);
  isConfirmModalOpen = signal(false);

  constructor(private sellerService: SellerService) {
    this.loadSellers();
  }

  loadSellers() {
    this.sellerService.getSellers().subscribe(sellers => {
      this.sellers.set(sellers);
    });
  }
}
```

Servicio para cargar vendedores

The screenshot shows the Visual Studio Code interface. On the left, the 'EXPLORADOR' (File Explorer) pane shows the project structure. The 'src' folder is expanded, showing 'app', 'shared', and 'services'. The 'services' folder is expanded, showing 'seller.service.spec.ts' and 'seller.service.ts'. The 'seller.service.ts' file is selected. The main editor shows the code for 'SellerService'. The code includes imports for 'Injectable' and 'signal' from '@angular/core'. It defines a class 'SellerService' with a 'providedIn: 'root'' decorator. The class has a private 'sellers' signal initialized with an array of three seller objects. It has three methods: 'getSellers()' which returns the 'sellers' signal, 'addSeller()' which adds a new seller to the 'sellers' array, and 'deleteSeller()' which removes a seller from the 'sellers' array based on its ID. The 'generateId()' method is also defined, which returns the next available ID. The terminal at the bottom shows the message 'Page reload sent to client(s)'.

```
src > app > sellers > services > seller.service.ts > SellerService > sellers

1 import { Injectable, signal } from '@angular/core';
2
3 @Injectable({
4   providedIn: 'root'
5 })
6 export class SellerService {
7   private sellers = signal<any[]>([
8     { id: 1, name: 'Juan Pérez' },
9     { id: 2, name: 'María Gómez' },
10    { id: 3, name: 'Carlos López' }
11  ]);
12
13   getSellers(): any[] {
14     return this.sellers();
15   }
16
17   addSeller(seller: any) {
18     const newSeller = { ...seller, id: this.generateId() };
19     this.sellers.set([...this.sellers(), newSeller]);
20   }
21
22   updateSeller(updatedSeller: any) {
23     const updatedList = this.sellers().map(seller =>
24       seller.id === updatedSeller.id ? updatedSeller : seller
25     );
26     this.sellers.set(updatedList);
27   }
28
29   deleteSeller(id: number) {
30     const updatedList = this.sellers().filter(seller => seller.id !== id);
31     this.sellers.set(updatedList);
32   }
33
34   private generateId(): number {
35     return this.sellers().length > 0
36       ? Math.max(...this.sellers().map(s => s.id)) + 1
37       : 1;
38   }
39 }
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POSTMAN CONSOLE

Page reload sent to client(s).

Lector de pantalla optimizado Lin. 11, col. 6 Espacios: 2 UTF-

Menú Productos



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE Elige un elemento.
CICLO ACADÉMICO: MARZO – JULIO 2025



Productos

Agregar Producto

Productos

Nuevo Producto

Nombre del Producto

Descripción

Precio

Categoría

Stock

Guardar Producto

Código:

```
File Edit Selection View Go Run ... app-grupal
EXPLORER
  APP-GRUPAL
  src
  app
  core
  services
  favorites
  offers
  products
  sellers
  services
  shared
  components
  container
  form
    favorite-form
    product-form
      product-form.component.css
      product-form.component.html
      product-form.component.ts
    header
    services
  transactions
    components
      list-transaction.component.css
      list-transaction.component.html
      list-transaction.component.ts
    app.component.css
    app.component.html
    app.component.ts
  OUTLINE
  TIMELINE
  PROBLEMS
  OUTPUT
  DEBUG CONSOLE
  TERMINAL
  PORTS
  esbuild
  NOTE: Raw file sizes do not reflect development server per-request transformations.
  Local: http://localhost:4200/
  press h + enter to show help
Ln 5, Col 24 Spaces: 2 UTF-8 CRLF TypeScript
```

```
src > app > shared > components > form > product-form > TS product-form.component.ts > ProductFormComponent
1 import { Component, EventEmitter, Output } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
3 @Component({
4   selector: 'bpaz-product-form',
5   imports: [FormsModule],
6   templateUrl: './product-form.component.html',
7   styleUrls: ['./product-form.component.css']
8 })
9 export class ProductFormComponent {
10   @Output() onSubmit = new EventEmitter<Product>();
11
12   product: Product = {
13     id: '',
14     name: '',
15     description: '',
16     price: 0,
17     category: '',
18     stock: 0
19   };
20
21   submitForm() {
22     this.onSubmit.emit(this.product);
23     // Reset form after submit
24     this.product = {
25       id: '',
26       name: ''
27     };
28   }
29 }
```

Menú Favoritos



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE Elige un elemento.
CICLO ACADÉMICO: MARZO – JULIO 2025



Favoritos

Agregar Favorito

Favoritos

Nuevo Favorito

ID del Producto

ID del Usuario

Notas

Guardar Favorito

Volver a la lista

Código:

```
File Edit Selection View Go Run ... app-grupal
EXPLORER
  APP-GRUPAL
  src
  app
  core
  services
  favorites
  offers
  products
  sellers
  services
  shared
  components
  container
  form
  favorite-form
    # favorite-form.component.css
    favorite-form.component.h...
    TS favorite-form.component.ts
  product-form
    # product-form.component.css
    product-form.component.h...
    TS product-form.component.ts
  header
  services
  transactions \components\ list-tr...
    # list-transaction.component.css
    list-transaction.component.html
  OUTLINE
  TIMELINE

src > app > shared > components > form > favorite-form > TS favorite-form.component.ts > FavoriteFormComponent
1 import { Component, EventEmitter, Output } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
3
4 @Component({
5   selector: 'bpaz-favorite-form',
6   imports: [FormsModule],
7   templateUrl: './favorite-form.component.html',
8   styleUrls: ['./favorite-form.component.css']
9 })
10 export class FavoriteFormComponent {
11   @Output() onSubmit = new EventEmitter<Favorite>();
12
13   favorite: Favorite = {
14     id: '',
15     productId: '',
16     userId: '',
17     addedAt: new Date(),
18     notes: ''
19   };
20
21   submitForm() {
22     this.onSubmit.emit(this.favorite);
23     // Reset form after submit
24     this.favorite = {
25       id: '',
26       productId: ''
27     };
28   }
29 }
```

Menú Ofertas

Ofertas

Identificador	Oferta	Descripción
1	Oferta 1	Descripción
2	Oferta 2	Descripción
3	Oferta 3	Descripción
4	Oferta 4	Descripción
5	Oferta 5	Descripción
6	Oferta 6	Descripción

Nueva Oferta

Identificador de la oferta
Ej. Oferta 1

Descripción de la Oferta

Cancelar Guardar

Agregar

Código



```
<h2 mat-dialog-title class="text-lg font-semibold">Nueva Oferta</h2>

<mat-dialog-content class="py-4 px-1">
  <div style="display: flex; flex-direction: column; gap: 1rem">
    <mat-form-field appearance="outline">
      <mat-label>Nombre de la Oferta</mat-label>
      <input matInput placeholder="Ej. Oferta 1" />
    </mat-form-field>

    <mat-form-field appearance="outline">
      <mat-label>Descripción de la Oferta</mat-label>
      <input matInput placeholder="Ej. Descripción" />
    </mat-form-field>
  </div>
</mat-dialog-content>

<mat-dialog-actions align="end" class="px-3 pb-3">
  <button mat-button color="warn" (click)="close()">Cancelar</button>
  <button
    mat-raised-button
    color="primary"
    [disabled]="!transactionName.trim()"
    (click)="save()"
  >
    Guardar
  </button>
</mat-dialog-actions>
```

Menú

Transacciones

Opciones del Sistema

Vendedores

Cientes

Productos

Favoritos

Ofertas

Transacciones

Usuario

Transacciones

Identificador	Transaccion
1	Transaction 1
2	Transaction 2
3	Transaction 3
4	Transaction 4
5	Transaction 5

Nueva Transacción

Ingrese un nombre a la transacción

Ej. Compra de materiale

Cancelar

Guardar

Agregar

Código



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE Elige un elemento.
CICLO ACADÉMICO: MARZO – JULIO 2025



```
<h2 mat-dialog-title class="text-lg font-semibold">
  Nueva Transacción
</h2>

<mat-dialog-content class="py-4 px-1">
  <mat-form-field appearance="outline" class="w-full">
    <mat-label>Nombre de la Transacción</mat-label>
    <input matInput [(ngModel)]="transactionName" placeholder="Ej. Compra de materiales" />
  </mat-form-field>
</mat-dialog-content>

<mat-dialog-actions align="end" class="px-3 pb-3">
  <button mat-button color="warn" (click)="close()">Cancelar</button>
  <button
    mat-raised-button
    color="primary"
    [disabled]="!transactionName.trim()"
    (click)="save()"
  >
    Guardar
  </button>
</mat-dialog-actions>
```