



UNIVERSIDAD TECNOLÓGICA DEL VALLE DEL MEZQUITAL

MATERIA:

DESARROLLO WEB

NOMBRE DEL MAESTRO:

SANTIAGO LABRA HERNANDEZ

NOMBRE DEL ALUMNO:

ISAAC MORGADO ABREU

TRABAJO:

PRACTICA 3

CUATRIMESTRE: 9

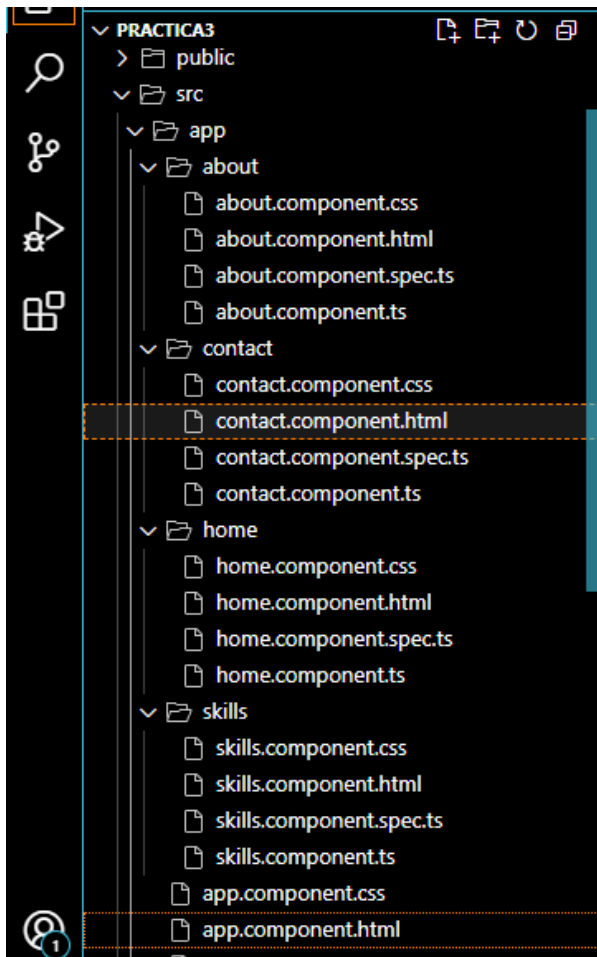
GRUPO: "B"

Estructura general:

- **Mínimo 3 vistas (páginas)** conectadas entre sí por medio de enlaces de navegación (routerLink).
- **Uso de componentes** separados para cada una de las secciones.
- **La página principal (app.component.html)** debe contener exclusivamente los llamados a los componentes, sin lógica ni contenido adicional.
- **Estilos personalizados** para cada componente, utilizando archivos .css o .scss por separado.
- **Uso de Angular Material** para mejorar la presentación de la interfaz.

Estructura de carpetas

- Todos los **componentes personalizados** creados deberán almacenarse dentro de una carpeta llamada components/.



Cada carpeta de componente debe incluir:

- Un archivo .ts para la lógica del componente.
- Un archivo .html para la plantilla.
- Un archivo .css o .scss para los estilos.

Importación de clases y módulos de Angular Material

Para usar elementos como mat-toolbar, mat-card, etc., se deben importar los módulos correspondientes al archivo .ts del componente que lo necesite, en la sección imports.

```
src > main.ts > ...
1 import { bootstrapApplication } from '@angular/platform-browser';
2 import { appConfig } from './app/app.config';
3 import { AppComponent } from './app/app.component';
4
5 bootstrapApplication(AppComponent, appConfig)
6   .catch((err) => console.error(err));
7
```

Una vez importado el módulo necesario, se puede usar la etiqueta directamente en el HTML del componente:

```
src > index.html > html > head > title
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>Practica3</title>
6   <base href="/">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="icon" type="image/x-icon" href="favicon.ico">
9 </head>
10 <body>
11   <app-root></app-root>
12 </body>
13 </html>
14
```

En el archivo `app-routing.module.ts`, se deben definir las rutas para cada componente:

```
src > app > app.routes.ts > ...
1 import { Routes } from '@angular/router';
2 import { HomeComponent } from '../home/home.component';
3 import { AboutComponent } from '../about/about.component';
4 import { SkillsComponent } from '../skills/skills.component';
5 import { ContactComponent } from '../contact/contact.component';
6
7 export const routes: Routes = [
8   { path: '', component: HomeComponent },
9   { path: 'about', component: AboutComponent },
10  { path: 'skills', component: SkillsComponent },
11  { path: 'contact', component: ContactComponent },
12  { path: '**', redirectTo: '' }
13 ];
14
```

Uso de enlaces (routerLink)

En el HTML de navegación, se deben usar enlaces con routerLink para cambiar de vista sin recargar la página:

```
src > app > app.component.html > ...
Go to component
1 <header>
2   <nav>
3     <a routerLink="/">Inicio</a>
4     <a routerLink="/about">Sobre mí</a>
5     <a routerLink="/skills">Mis gustos</a>
6     <a routerLink="/contact">Info para contactarme</a>
7   </nav>
8 </header>
9
10 <main>
11   <router-outlet></router-outlet>
12 </main>
13
14 <footer>
15   <p>&copy; 2025 Mi Portfolio Angular Isaac UTMV</p>
16 </footer>
17
18
```

Resumen de pasos para desarrollo

- Crear la aplicación en Angular (si no existe):
ng new mi-app
- Instalar Angular
ng add @angular/material
- Crear componentes dentro de src/app/components:
- Configurar rutas en app-routing.module.ts.
- Usar etiquetas routerLink para navegación entre vistas.
- Importar los módulos de Angular Material necesarios en cada componente.
- Definir estilos personalizados para cada sección.
- Verificar que app.component.html solo contenga llamados a componentes y no contenido adicional.

Resultado esperado

- Una aplicación en Angular que:
- Permita al usuario navegar entre tres vistas (por ejemplo, "Sobre mí", "Proyectos", "Contacto").
- Esté construida de forma modular con componentes bien organizados.
- Utilice Angular Material para un diseño atractivo y moderno.
- Tenga estilos propios para cada componente.
- Cumpla con las buenas prácticas de estructura y mantenimiento de proyectos Angular.