# Desarrollo

Proyecto Final

Angular

Paso a Paso

# Tabla de contenido

OE	3JETIVO4	
	Descripción de la Aplicación4	
1.	ACTUACIONES PREVIAS5	)
2.	DESARROLLAR SERVICIO KITCHENS	,
3.	GENERAR ROUTING8	,
4.	COMPONENTE APP9	)
	Generar app.component.html9	
5.	COMPONENTE HOME10	)
	Generar home.component.html10	
	Generar home.component.ts11	
	Vista Componente Home12	
6.	COMPONENTE PORTOFOLIO13	)
	Generar portfolio.component.html13	
	Generar portfolio.component.ts14	
	Vista Componente Portfolio15	
7.	COMPONENTE PROJECT16	)
	Generar project.component.html16	
	Generar project.component.ts17	
	Vista Componente Project18	
9.	COMPONENTE PRICE-MODIFY19	,
	Generar price-modify.component.html19	
	Generar price-modify.component.ts20	
	Vista Componente Price-Modify21	
10	COMPONENTE ABOUT22	
	Generar about.component.html22	
	Vista Componente About23	

11	1. COMPONENTE CONTACT	24
	Generar contact.component.html24	
	Vista Componente Contact25	
12	2. ESTILOS CSS APLICADOS	26
	Estilos Generales en style.css26	
	Estilos app.component.css27	
	Estilos home.component.css28	
	Estilos portfolio.component.css29	
	Estilos project.component.css30	
	Estilos price-modify.component.css30	
	Estilos about.component.css31	
	Estilos contact.component.css31	

### **OBJETIVO**

El objetivo de ese documento es detallar el proceso de desarrollo, paso a paso, de una aplicación web usando Angular.

# Descripción de la Aplicación

La aplicación representa el portfolio de un interiorista, orientado al diseño de cocinas.

La aplicación permite conocer tanto el trabajo del interiorista como información sobre él, así mismo, hay un apartado para contactar a través de la aplicación. Para ello, se disponen de **6 vistas** manejadas por **7 componentes**.

Existe una **barra de navegación común** a todas las vistas, que mediante routing, permite la navegación entre componentes. Diche barra tiene los enlaces a: **home**, **portfolio**, **about** y **contact**.

La barra de navegación y el logo del interiorista se renderizan en el componente principal de la aplicación (app.component).

A continuación, se describe cada vista y sus funcionalidades integradas:

- 1. Home: Es la vista que se renderiza cuando entramos a la raíz de la página. En ella nos encontramos con un título, un párrafo que nos da la bienvenida y un botón que nos dirige a la vista de "About". Después, se muestran imágenes de los proyectos realizados.
- 2. Portfolio: En esta vista se renderiza el título de esta vista y se despliega una lista de los proyectos realizados junto a una breve descripción. Al hacer doble click sobre la foto de algún proyecto se abre la vista de "Project" con la información de dicho proyecto.
- 3. Project: En esta vista se renderiza la foto y la información del proyecto seleccionado. También se mostrará un botón para volver al home, un botón para volver atrás y un botón para modificar el precio que abrirá la vista de modificación de precio.
- 4. Modificación de precio: Esta vista renderiza un formulario para modificar el precio del producto seleccionado y un botón cerrar que nos dirige a la vista de portfolio.
- **5. About:** En esta vista se renderiza el título de esta y una descripción personal.
- **6. Contact:** En esta vista se renderiza el título de esta y un formulario de contacto.

### 1. ACTUACIONES PREVIAS

Antes de empezar a desarrollar los diferentes componentes debemos organizar la estructura del proyecto en base a la descripción y los requerimientos del mismo.

- 1. Creación del proyecto: Creamos el proyecto con la opción de routing y css. Eliminaremos el contenido por defecto que incluye angular cli en el componente app.component.html.
  - ng new Proyecto\_Final

Se creará el módulo de app-routing donde indicaremos las rutas más adelante.

2. Creación de los componentes:

```
> ng g c home
> ng g c portfolio
> ng g c Project
> ng g c priceModify
> ng g c about
> ng g c contact
```

3. Creación de ficheros modelo: Para este proyecto nos hará falta crear una clase que abstraiga la información de cada proyecto de cocina, tendrá el nombre de "kitchen".

La clase contendrá una **referencia numérica** que se **autoincrementa** con la creación de nuevas instancias y como propiedades públicas tendrá dicha **referencia**, la ruta de la **foto**, el **precio** y la **descripción**.

Este fichero se creará dentro de la carpeta /src/models que habrá que crear. (Se ha decidido crear las propiedades como públicas por simplicidad del código).

```
export class Kitchen {
    private static id = 0;
    public reference: number;

    constructor(public photo: string, public price: number, public description: string)
{
        this.reference = Kitchen.id;
        Kitchen.id++;
    }
    setPrice(price: number) {
            this.price = price;
    }
}
```

- **4. Creación de servicios:** Para esta aplicación necesitamos de un servicio que proporcione los datos de todos los proyectos de cocinas.
  - ng g s services/kitchens

Importamos el servicio en app.module.ts y lo añadimos al array de "providers"

- **5.** Creación de carpeta de imágenes: Creamos la carpeta "img" dentro de la carpeta de "assets" y añadimos las imágenes que vayamos a usar.
- 6. Vista de la estructura de la aplicación y del fichero app.module.ts:

```
EXPLORER: FINALPROJECTANG... [] 🛱 🖰 🖰 🙃 …
                                                 TS app.module.ts X
🗦 📹 .angular
                                                          You, 3 minutes ago | 2 authors (MoleiroClase and others)
import { NgModule } from '@angular/core';
> 戍 .vscode
> 👼 node_modules
                                                          import { BrowserModule } from '@angular/platform-browser';
✓ 륣 src
🗸 🍺 app
                                                          import { AppRoutingModule } from './app-routing.module';
                                                         import { AppComponent } from './app.component';
import { HomeComponent } from './home/home.component';
 > ii about
 > 📹 contact
                                                         import { PortfolioComponent } from './portfolio/portfolio.component';
  > 📹 home
                                                         import { ProjectComponent } from './project/project.component';
  > 🔳 portfolio
                                                        import { PriceModifyComponent } from './price-modify/price-modify.component';
import { AboutComponent } from './about/about.component';
 > ii price-modify
 > 🔳 project
                                                         import { ContactComponent } from './contact/contact.component';
import { KitchensService } from './services/kitchens.service';
 services
     kitchens.service.spec.ts
     TS kitchens.service.ts
                                                          @NgModule({
    TS app-routing.module.ts
                                                          declarations: [
    app.component.css
                                                             AppComponent,
HomeComponent,
    app.component.html
                                                              PortfolioComponent,
    app.component.spec.ts
                                                             ProjectComponent,
PriceModifyComponent,
    TS app.component.ts
    TS app.module.ts
                                                              AboutComponent,
∨ 🔞 assets
                                                               ContactComponent
  🗸 📹 img
     cocina1.jpg
                                                           imports: [
     cocina2.jpg
     cocina3.jpg
                                                               AppRoutingModule
     cocina4.jpg
                                                             providers: [KitchensService],
     아 .gitkeep
                                                             bootstrap: [AppComponent]
  environments
models
                                                          export class AppModule { }
    TS Kitchen.ts
   favicon.ico
   😈 index.html
   TS main.ts
   TS polyfills.ts
   styles.css
   TS test.ts
```

### 2. DESARROLLAR SERVICIO KITCHENS

- 1. Abrir archivo kitchens.service.ts
- 2. Importar el modelo **Kitchen** que hemos creado previamente.
- 3. Declarar un **array** de objetos de tipo **Kitchen**.
- 4. Rellenar el array con los datos de los proyectos de cocina.
- 5. Crear el **método** que **devuelva** los elementos del array.
- 6. Crear el **método** que **modifique** el precio de un elemento del array. El método recibe el **nuevo precio** y la **referencia** de la cocina, después busca la cocina y haciendo uso del método **setPrice** de la clase **kitchen** cambia el valor del atributo price.

```
import { Injectable } from '@angular/core';
import { Kitchen } from 'src/models/Kitchen';
@Injectable({
   providedIn: 'root',
})
export class KitchensService {
   private kitchens: Kitchen[] = [
        new Kitchen('../../assets/img/cocina1.jpg', 200, 'Cocina 1'),
        new Kitchen('../../assets/img/cocina2.jpg', 300, 'Cocina 2'),
        new Kitchen('../../assets/img/cocina3.jpg', 400, 'Cocina 3'),
        new Kitchen('../../assets/img/cocina4.jpg', 500, 'Cocina 4'),
   ];
    constructor() {}
   getKitchens() {
        return this.kitchens;
    setPrice(price: number, index: number) {
        this.kitchens.find((kitchen) => kitchen.reference ===
index)?.setPrice(price);
```

### 3. GENERAR ROUTING

- 1. Abrir archivo app-routing.module.ts.
- 2. Realizar los **imports** de los **componentes** que queremos enrutar.
- 3. **Mapear** las **rutas** con los componentes de la aplicación en el **array** de **routes**.
- 4. Especificar las rutas que aceptan parámetros indicando el nombre del parámetro.

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { PortfolioComponent } from './portfolio/portfolio.component';
import { AboutComponent } from './about/about.component';
import { ContactComponent } from './contact/contact.component';
import { ProjectComponent } from './project/project.component';
import { PriceModifyComponent } from './price-modify/price-modify.component';
const routes: Routes = [
    { path: '', component: HomeComponent },
    { path: 'portfolio', component: PortfolioComponent },
    { path: 'about', component: AboutComponent },
    { path: 'contact', component: ContactComponent },
    { path: 'project/:id', component: ProjectComponent },
    { path: 'price-modify/:id', component: PriceModifyComponent }
];
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

### 4. COMPONENTE APP

### Generar app.component.html

- 1. Crear un **header** que contendrá el **logo** y la **navegación**.
- 2. Crear la directiva router-outlet.
- 3. La navegación constará de **enlaces** que haciendo uso del **routing** mostrarán los componentes en la etiqueta de **router-outlet**.
- 4. Para indicar la ruta a la que queremos navegar usamos la funcionalidad routerLink.
- 5. Para especificar los estilos que queramos en función de la ruta activa usamos routerLinkActive.
- 6. **routerLinkActiveOptions** nos permite configurar opciones adicionales. Por ejemplo, en este caso lo usamos para que solo se aplique el estilo si la ruta es exacta.

### 5. COMPONENTE HOME

### **Generar home.component.html**

- 1. Crear el **título**, el **párrafo** de presentación, el **botón** que nos dirige al componente **about** y la galería de **imágenes**.
- 2. Añadir al botón el **evento** *click* que llama al método goToAbou().

```
<section class="container">
    <h1 class="home__title">Hello, Welcome to my Portfolio</h1>
    Lorem ipsum dolor, sit amet consectetur adipisicing elit.
Quam voluptate ipsum officiis magni,
        sint ullam fugiat reprehenderit minima atque minus nisi quo, nihil commodi
dicta dolore. Inventore deleniti quia
       reprehenderit!
    <button class="btn" (click)="goToAbout()">More About Me</button>
    <div class="home__galery">
       <div class="home galery--item">
            <img src="../../assets/img/cocina1.jpg" alt="Cocina 1">
       </div>
       <div class="home galery--item">
            <img src="../../assets/img/cocina2.jpg" alt="Cocina 2">
       </div>
       <div class="home__galery--item">
            <img src="../../assets/img/cocina3.jpg" alt="Cocina 3">
       </div>
       <div class="home__galery--item">
            <img src="../../assets/img/cocina4.jpg" alt="Cocina 4">
       </div>
    </div>
  section>
```

# **Generar home.component.ts**

- 1. Importar Router.
- 2. Instanciar objeto **router** de tipo **Router**.
- 3. Crear método goToAbout() para ir al componente **about** a través del método **navigate** del objeto de tipo Router.

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
    selector: 'app-home',
    templateUrl: './home.component.html',
    styleUrls: ['./home.component.css'],
})
export class HomeComponent {
    constructor(private router: Router) {}

    goToAbout() {
        this.router.navigate(['/about']);
    }
}
```

# Vista Componente Home



Home Portfolio About Contact

# Hello, Welcome to my Portfolio

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quam voluptate ipsum officiis magni, sint ullam fugiat reprehenderit minima atque minus nisi quo, nihil commodi dicta dolore. Inventore deleniti quia reprehenderit!

#### MORE ABOUT ME



### 6. COMPONENTE PORTOFOLIO

## Generar portfolio.component.html

- 1. Crear el título y la lista ul.
- 2. Crear elementos **li** con directiva **ngFor** para recorrer el array de **kitchens** y mostrar la **imagen** y la **descripción** de cada cocina.
- 3. Añadir evento *dblclick* a las imágenes que se muestran. Este evento llama al método goToProject(referencia) que recibe como parámetro la referencia de la cocina.

# **Generar portfolio.component.ts**

- 1. Importar Router e inyectar servicio KitchensService.
- 2. Crear array de kitchens.
- 3. Declarar **Router** y **KitchensService** en el constructor.
- 4. Dentro de **ngOnInit** inicializar array de **kitchens** llamando al método getKitchens() del servicio.
- 5. Crear **método** goToProject(referencia) que nos dirige al componente **project** pasando el **parámetro** index a través del **routing**.

```
import { KitchensService } from './../services/kitchens.service';
import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
    selector: 'app-portfolio',
    templateUrl: './portfolio.component.html',
    styleUrls: ['./portfolio.component.css']
})
export class PortfolioComponent {
    kitchens: any=[];
    constructor(private kitchenService:KitchensService, private router:Router) {}

    ngOnInit() {
        this.kitchens = this.kitchenService.getKitchens();
    }
    goToProject(index: number) {
        this.router.navigate(['/project', index]);
    }
}
```

### Vista Componente Portfolio



Home Portfolio About Contact

#### **Portfolio**



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras aliquam ac mauris id auctor. Curabitur a leo imperdiet, iaculis orci id, euismod felis. Suspendisse elementum molestie risus efficitur tempus. Proin laoreet diam quis nunc sollicitudin, in ullamcorper justo placerat. Nam id porta metus. Morbi fermentum, justo sed volutpat scelerisque, magna felis gravida arcu, a egestas velit felis at tellus. Curabitur ut sem a enim vestibulum pellentesque. Sed placerat nisi vel pellentesque ornare. In consequat aliquet est a feugiat. Curabitur gravida accumsan tortor non lobortis. Integer nec risus lacinia, pharetra nulla in ornare lacus.



Nunc vel faucibus orci, in aliquam neque. Ut mattis magna nec erat iaculis rhoncus. Nulla tortor libero, pretium eu mollis a, viverra eget massa. Nulla lectus eros, tincidunt vitae varius sit amet, sollicitudin ac nibh. Maecenas id fermentum justo, ut sodales nunc. Aliquam feugiat mi sit amet ligula pretium, tristique ornare nunc tincidunt. Ut eros lorem, commodo vitae nibh eu, facilisis tristique eros. Pellentesque iaculis tristique molestie. Praesent aliquet euismod enim sed finibus. Sed portitior, lectus vitae faucibus rutrum, lorem felis pulvinar tortor, lacinia sodales sapien justo at purus. Sed pharetra in enim sed fringilla. Nam tempus semper nulla nec portititor. In imperdiet lectus vitae ex malesuada tempus. Donec interdum venenatis tempor. Donec sed orci quis magna viverra molestie.



Donec sed nisl leo. Praesent congue, arcu vel lobortis accumsan, libero est aliquet libero, id porta diam tellus vitae nisi. Fusce sollicitudin felis id consequat hendrerit. In ultricies nibh sit amet feugiat vulputate. Aenean et aliquet purus. Maecenas consequat libero varius nisi consectetur finibus sit amet id nulla. Curabitur maximus ligula ac magna tempus, at porta velit ultrices. Praesent volutpat cursus nisi a portitior. Aliquam ac nulla euismod mauris vulputate malesuada.



Nulla tincidunt ipsum sed luctus eleifend. Quisque interdum dictum tempus. Proin porta sapien ac lorem pellentesque ultricies. Proin tincidunt feugiat mauris in rhoncus. Phasellus at mi ut elit gravida accumsan a eget magna. Sed pharetra libero mollis turpis finibus, egestas efficitur metus sodales. Mauris a velit sit amet erat condimentum imperdiet. Vestibulum vestibulum fermentum nisi, in luctus eros ultricies dignissim. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam erat volutpat. Interdum et malesuada fames ac ante ipsum primis in faucibus. Sed mollis, ligula sed interdum condimentum, metus nisi vehicula risus, id pharetra quam lacus a ante. In viverra velit ac libero porta pharetra pulvinar placerat purus. In hac habitasse platea dictumst.

# 7. COMPONENTE PROJECT

## Generar project.component.html

- 1. Crear el elemento **imagen** e **interpolar** la **url** sacada del **objeto kitchen** en el atributo **src**.
- 2. Crear la **descripción** y el **precio** interpolando los atributos del **objeto kitchen**.
- 3. Crear el **botón** para **modificar el precio** haciendo que llame al método goToModify() ante el evento *click.*
- 4. Crear el **botón** para volver al **home** que llama al método goToHome() ante el evento *click*.
- 5. Crear **botón** para volver **atrás** que llame al método goBack() ante el evento *click*.

### **Generar project.component.ts**

- 1. Importar Router y ActivatedRoute, Location para la navegación.
- 2. Inyectar el **servicio** de **KitchensServices**.
- 3. Importar la clase Kitchen.
- 4. Declarar array de donde guardaremos el array devuelto por el **servicio** y el **objeto kitchen** donde guardaremos el **objeto rescatado** del **array kitchens** que **coincida** con el **index** enviado a través del routing.
- 5. Declarar en el constructor el **servicio**, los **objetos** del **routing** y la navegación.
- 6. En el método **ngOnInit** inicializar el **index** con el **parámetro** enviado por **routing** que está almacenado en el **snapshot** del objeto **ActivatedRoute**.
- 7. Recoger el **array guardado** en el **servicio** y utilizando el **método find** para encontrar el objeto cuya **referencia coincida** con la enviada.
- 8. Crear los **métodos** de **navegación** goToHome(), goToModify() y goBack() haciendo uso de las instancias de **Router** y **Location**.

```
import { Component } from '@angular/core';
import { Router,ActivatedRoute } from '@angular/router';
import { KitchensService } from '../services/kitchens.service';
import { Location } from '@angular/common';
import { Kitchen } from '.../.../models/Kitchen';
@Component({
  selector: 'app-project',
  templateUrl: './project.component.html',
  styleUrls: ['./project.component.css']
})
export class ProjectComponent {
    kitchens: any=[];
    kitchen: any;
    index: number=0;
    constructor(private kitchenService:KitchensService, private router:Router,
private ruta:ActivatedRoute,private location:Location) {}
    ngOnInit() {
        this.index= this.ruta.snapshot.params['id'];
        this.kitchens = this.kitchenService.getKitchens();
        this.kitchen = this.kitchens.find((kitchen:Kitchen) => kitchen.reference
== this.index);
    goToHome() {
        this.router.navigate(['/home']);
    goToModify() {
        this.router.navigate(['/price-modify', this.index]);
    goBack() {
        this.location.back();
```

# **Vista Componente Project**



#### **Home Portfolio About Contact**



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras aliquam ac mauris id auctor. Curabitur a leo imperdiet, iaculis orci id, euismod felis. Suspendisse elementum molestie risus efficitur tempus. Proin laoreet diam quis nunc sollicitudin, in ullamcorper justo placerat. Nam id porta metus. Morbi fermentum, justo sed volutpat scelerisque, magna felis gravida arcu, a egestas velit felis at tellus. Curabitur ut sem a enim vestibulum pellentesque. Sed placerat nisi vel pellentesque ornare. In consequat aliquet est a feugiat. Curabitur gravida accumsan tortor non lobortis. Integer nec risus lacinia, pharetra nulla in, ornare lacus.

#### Precio 200 €

MODIFICAR EL PRECIO

VOLVER AL HOME

VOLVER ATRÁS

### 9. COMPONENTE PRICE-MODIFY

# **Generar price-modify.component.html**

- Crear el título h2, la etiqueta del input donde ingresamos el precio. El valor del input lo recogeremos con un ngModel de la variable que contiene el precio de la cocina. Para hacer uso de la directiva ngModel debemos ir al app.module.ts importar FormsModule y añadir al array de imports.
- 2. Crear el botón para **modificar** el precio que llama al método modifyPrice() ante el evento *click*.
- 3. Crear el botón de cerra que llama al método goToPortfolio() ante el evento *click*.

### **Generar price-modify.component.ts**

- 1. Inyectar el **servicio KitchensService** e importar la clase **Kitchen**, **ActivatedRoute** y **Router**.
- 2. Declarar la variable que va a guardar la **cocina** que rescatemos del array y la variable que va a guardar el **precio** de la cocina.
- 3. **Instanciar** en el constructor los **servicios** y **objetos** que vamos a necesitar.
- 4. En el método **ngOnInit** inicializar la variable **kitchen** buscando en el array del servicio, después guardar el **precio** en la variable price.
- Crear el método modifyPrice() que llama al método del servicio setPrice y le pasa el precio y la referencia de la cocina como parámetros, después de pedir confirmación.
- 6. Crear el método goToPortfolio() que utiliza el método **navigate** del **router** para ir a la vista del componente *portfolio*.

```
import { Component } from '@angular/core';
import { KitchensService } from '../services/kitchens.service';
import { Kitchen } from '../../models/Kitchen';
import { ActivatedRoute, Router } from '@angular/router';
@Component({
  selector: 'app-price-modify',
 templateUrl: './price-modify.component.html',
  styleUrls: ['./price-modify.component.css']
})
export class PriceModifyComponent {
 kitchen: any;
 price: number=0;
 constructor(private kitchenService: KitchenService, private route: ActivatedRoute,
private router: Router) { }
 ngOnInit() {
    this.kitchen=this.kitchenService.getKitchens().find((kitchen: Kitchen) =>
kitchen.reference == this.route.snapshot.params['id']);
    this.price=this.kitchen.price;
 modifyPrice() {
    if(confirm("¿Estás seguro de que quieres modificar el precio?"))
    this.kitchenService.setPrice(this.price??=0,this.kitchen.reference);
 goToPortfolio() {
    this.router.navigate(['/portfolio']);
```

# **Vista Componente Price-Modify**



### **10. COMPONENTE ABOUT**

## **Generar about.component.html**

1. Crear **título**, **párrafos** con la información personal y la imagen de la sección.

```
<section class="container">
   <h2 class="about title">About</h2>
   <div class="about__content">
       <div class="about__img">
            <img src="../../assets/img/about.jpg" alt="Portrait">
       </div>
       <div class="about text">
           Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus,
praesentium? Exercitationem quibusdam
               impedit
               quasi, deserunt corporis suscipit, sit maxime alias fuga dicta velit,
consequuntur earum? Distinctio
               illum
               officia ipsum! Delectus?
           Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod quisquam
totam animi mollitia eum possimus
               quos
               vero, illo obcaecati. Animi exercitationem, enim itaque quisquam illo
cum sunt dolor minima obcaecati?
           </div>
   </div>
</section>
```

# **Vista Componente About**



#### Home Portfolio About Contact

### **About**



Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus, praesentium? Exercitationem quibusdam impedit quasi, deserunt corporis suscipit, sit maxime alias fuga dicta velit, consequuntur earum? Distinctio illum officia ipsum! Delectus?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod quisquam totam animi mollitia eum possimus quos vero, illo obcaecati. Animi exercitationem, enim itaque quisquam illo cum sunt dolor minima obcaecati?

### 11. COMPONENTE CONTACT

# **Generar contact.component.html**

- 1. Crear el **título** del componente en un elemento h2.
- 2. Crear el **formulario** de contacto con los campos de **nombre**, **email**, **cuerpo** del mensaje y el **botón** para enviar.
- 3. Crear la imagen de la sección.

```
<section class="container">
    <h2 class="contact__title">Contact</h2>
    <div class="contact__content">
        <form action="" class="contact__form">
            <div class="field">
                <label for="name" class="field label">Name</label>
                <input type="text" class="field__input" id="name" name="name"</pre>
placeholder="Your name">
            </div>
            <div class="field">
                <label for="email" class="field label">Email</label>
                <input type="email" class="field__input" id="email" name="email"</pre>
placeholder="Your email">
            </div>
            <div class="field">
                <label for="message" class="field__label">Message</label>
                <textarea name="message" id="message" class="field__input"</pre>
placeholder="Your message"></textarea>
            </div>
            <button class="btn">Send Message</button>
        </form>
        <div class="contact__image">
            <img src="../../assets/img/mail.jpg" alt="Mail">
        </div>
    </div>
  section>
```

# **Vista Componente Contact**



### 12. ESTILOS CSS APLICADOS

# Estilos Generales en style.css

```
:root{
    --color-bg: #c0c2a8;
    --color-text: #292418;
    --color-white: #fff;
html{
    font-size: 62.5%;
    box-sizing: border-box;
 ,*:before,*:after{
    margin: 0;
    padding: 0;
body{
    font-family: 'Poppins', sans-serif;
    line-height: 1.5;
    background-color: var(--color-bg);
    color: var(--color-text);
    font-size: 1.6rem;
a{
    text-decoration: none;
    color: var(--color-text);
li{
    list-style: none;
img{
    max-width: 100%;
    height: auto;
section{
    padding: 5rem 2rem;
h1{
    font-size: 3.6rem;
    font-weight: 700;
    margin-bottom: 2rem;
h2{
    font-size: 4.4rem;
    font-weight: 700;
    margin-bottom: 2rem;
```

```
.container{
   max-width: 1400px;
   margin: 0 auto;
.btn{
   width: fit-content;
   padding: 1rem 2rem;
   background-color: var(--color-text);
   color: var(--color-white);
   border-radius: 2rem;
   text-transform: uppercase;
   font-size: 1.6rem;
   font-weight: 700;
   cursor: pointer;
   transition: all .3s;
   border: unset;
.btn:hover{
   box-shadow: var(--color-text) 2px 2px 7px 0px;
   transform: scale(1.05);
```

# **Estilos app.component.css**

```
header{
    box-shadow: var(--color-text) 0px 2px 5px 0px;
}
.header__content {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 1rem 2rem;
}
@media (max-width: 500px){
        .header__content{
            flex-direction: column;
        }
}
.logo {
    width: 10rem;
}
.navigation{
        display: flex;
        gap: 1rem;
}
```

```
.navigation a{
    font-size: 2.4rem;
    font-weight: bold;
    transition: all .2s;
}
.navigation a:hover{
    color: var(--color-white);
    text-shadow: var(--color-text) 2px 2px 2px;
}
@media (max-width: 500px){
    .navigation a{
        font-size: 1.8rem;
    }
}
.active{
    color: var(--color-white);
    text-shadow: var(--color-text) 2px 2px 2px;
}
```

# **Estilos home.component.css**

```
.home__text{
    width: min(100%, 800px);
    margin: 2rem 0;
.home__galery {
    margin: 2rem auto;
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    gap: 1rem;
@media (max-width: 500px){
    .home__galery {
        grid-template-columns: 1fr;
.home__galery--item img {
   width: 100%;
    height: 100%;
    border-radius: 2rem;
    box-shadow: var(--color-text) 0px 0px 7px 0px;
```

### **Estilos portfolio.component.css**

```
.portfolio__list{
    display:flex;
    margin-top: 2rem;
    flex-direction: column;
    gap: 2rem;
.portfolio__item{
   display: flex;
   gap: 2rem;
    box-shadow: var(--color-text) 2px 2px 7px 0px;
    padding: 2rem;
    border-radius: 2rem;
.portfolio__item--img img{
    height: 100%;
   border-radius: 2rem;
   cursor: pointer;
    transition: all .3s;
    box-shadow: var(--color-text) 0px 0px 5px 0px;
.portfolio__item--img img:hover{
    filter: brightness(0.4);
.portfolio item--content{
   width: 50%;
    display: flex;
    align-items:center ;
@media (max-width: 900px){
  .portfolio__item{
   flex-direction: column;
 .portfolio__item--content{
   width: 100%;
```

### **Estilos project.component.css**

```
.project{
    display: flex;
    gap: 2rem;
.project img img{
    border-radius: 2rem;
    box-shadow: var(--color-text) 2px 2px 7px 0px;
.project__content{
   width: 50%;
   display: flex;
    flex-direction: column;
    justify-content: center;
    gap: 2rem;
.btn container{
   display: flex;
   gap: 2rem;
    margin-top: 2rem;
.project__price{
    font-size: 2.4rem;
    font-weight: 700;
@media (max-width: 768px){
    .project{
        flex-direction: column;
    .project__content{
        width: 100%;
        padding-bottom: 2rem;
        border-bottom: var(--color-text) 1px solid;
```

# **Estilos price-modify.component.css**

```
.modify__field{
    width: fit-content;
    display: flex;
    align-items: center;
    gap: 2rem;
    padding: 2rem;
    border-radius: 2rem;
    margin-bottom: 2rem;
    box-shadow: var(--color-text) 2px 2px 7px 0;
```

```
.modify__field label{
    font-size: 2.4rem;
    font-weight: bold;

}
.modify__field input{
    width: 10rem;
    padding: 1rem 2rem;
    border: var(--color-text) 1px solid;
    border-radius: 1rem;
    font-size: 2.2rem;
}
```

### **Estilos about.component.css**

```
.about__content{
   display:flex ;
   gap: 2rem;
   margin: 0 auto;
   padding: 2rem;
   border-radius: 2rem;
   box-shadow: var(--color-text) 2px 2px 7px 0;
.about__text{
   width: 50%;
   display: flex;
   flex-direction: column;
   justify-content: center;
   gap: 2rem;
.about__img img{
   width: 100%;
   height: 100%;
   border-radius: 2rem;
```

# **Estilos contact.component.css**

```
.contact__form{
   width: 50%;
   padding: 2rem;
   border-radius: 2rem;
   box-shadow: var(--color-text) 2px 2px 7px 0;
}
.contact__content{
   display: flex;
   gap: 2rem;
```

```
@media (max-width: 768px) {
    .contact__content{
        flex-direction: column;
    .contact__form{
        width: unset;
.field{
    display: flex;
    flex-direction: column;
    margin-bottom: 1rem;
field input{
    padding: 1rem;
    border: 1px solid var(--color-text);
    border-radius: 0.5rem;
.field label{
    margin-bottom: 0.5rem;
    font-weight: bold;
.field textarea{
   min-height:20rem;
    padding: 1rem;
    border: 1px solid var(--color-text);
    border-radius: 0.5rem;
.contact__image{
    width: 50%;
@media (max-width: 768px) {
    .contact__image{
        width: 100%;
.contact__image img{
    width: 100%;
    height: 100%;
    border-radius: 2rem;
    box-shadow: var(--color-text) 2px 2px 7px 0;
```