



ANGULAR

DOCUMENTACIÓN
‘COCINA PARA TODOS’

VÍCTOR SÁNCHEZ

Índice

Introducción a la aplicación “Cocina para todos”	2
Primeros pasos – Creación del proyecto	3
Fichero Modelo.....	4
Servicio ‘CocinasService’	5
Componente ‘Navbar’ (app.component)	6
Componente ‘Home’	8
Componente ‘Portfolio’	10
Componente ‘About’	12
Componente ‘Contact’	14
Componente ‘Display’	16
Componente ‘Precio’	19

Introducción a la aplicación “Cocina para todos”

Bienvenido a la documentación de "**Cocina para todos**", una aplicación desarrollada en Angular que brinda a sus usuarios una experiencia única y accesible para visualizar las distintas cocinas disponibles en el mercado.

Con "Cocina para todos", podrás navegar por un portafolio de imágenes y descripciones detalladas de cada cocina, lo que te permitirá tener una idea clara de lo que estás buscando antes de tomar una decisión de compra.

En esta aplicación también encontrarás una sección dedicada a hablar sobre la empresa detrás de "Cocina para todos", para que conozcas un poco más sobre su historia y sus valores.

También encontrarás un formulario de contacto que te permitirá comunicarte directamente con ellos si tienes alguna pregunta o sugerencia.

Además, "Cocina para todos" no solo es útil para los usuarios que buscan adquirir una cocina, sino también para los gestores de la aplicación.

Con esta aplicación, los gestores tienen la opción de modificar el precio de las cocinas en tiempo real, lo que les permite adaptarse rápidamente a los cambios en el mercado y ofrecer a sus clientes los mejores precios posibles.

En resumen, "Cocina para todos" es una herramienta que brinda a los usuarios una experiencia de compra fácil y accesible, y a los gestores una herramienta eficiente y flexible para manejar su catálogo de productos.

Primeros pasos – Creación del proyecto

Para crear nuestro proyecto en Angular utilizamos el siguiente comando -> *ng n Cocinas*

También vamos a modificar el archivo *style.css* global de nuestro proyecto para aplicar las siguientes propiedades de manera global:

```
DIW > 05_Cocinas > src > styles.css > ...
1  /* You can add global styles to this file, and also import other style files */
2  * {
3      margin: 0;
4      padding: 0;
5      box-sizing: border-box;
6      background-color: #000000;
7      color: white;
8  }
```

Fichero Modelo

Creación -> Creamos el fichero `cocina.model.ts` en la ruta `src/app/Models`

El archivo '`cocina.model.ts`' es el modelo que representa a la clase '`Cocina`'.

La clase '`Cocina`' es una clase de datos que define las propiedades y los datos de cada cocina que se visualizará en la aplicación.

Esta clase incluye una serie de propiedades como '`ref`' (referencia), '`name`' (nombre), '`desc`' (descripción), '`color`' (color), '`price`' (precio) e '`img`' (imagen).

También incluye un constructor que se utiliza para inicializar una nueva instancia de la clase '`Cocina`' con valores específicos para cada una de estas propiedades.

```
export class Cocina {  
  ref: number;  
  name: string;  
  desc: string;  
  color: string;  
  price: number;  
  img: string;  
  
  constructor(ref: number, name: string, desc: string, color: string, price: number, img: string) {  
    this.ref = ref;  
    this.name = name;  
    this.desc = desc;  
    this.color = color;  
    this.price = price;  
    this.img = img;  
  }  
}
```

Servicio 'CocinasService'

Creación servicio -> `ng g s services/CocinasService`

Este archivo define un servicio en Angular que se utiliza para gestionar la lista de cocinas.

La clase **'CocinasService'** está decorada con **@Injectable** para indicar que es un servicio que se puede inyectar en otros componentes.

La clase **'CocinasService'** define una propiedad privada **'cocinas'** que es un array de objetos de tipo **'Cocina'**. Cada objeto **'Cocina'** representa una cocina específica y contiene información sobre su referencia, nombre, descripción, color, precio y ruta de la imagen.

La información sobre las cocinas se almacena en un array en este servicio, lo que significa que los componentes que lo utilicen tienen acceso a la misma información sin tener que cargarla cada vez que se necesita. Esto puede mejorar la eficiencia de la aplicación y asegurar la consistencia de la información.

```
import { Injectable } from '@angular/core';
import { Cocina } from '../Models/cocina.model';

@Injectable({
  providedIn: 'root'
})
export class CocinasService {

  private cocinas: Cocina[] = [
    new Cocina(1, 'Metod', 'Lorem', 'red', 100, 'assets/img/cocina1.jpg'),
    new Cocina(2, 'Enhet', 'Lorem', 'blue', 200, 'assets/img/cocina2.jpg'),
    new Cocina(3, 'Knoxhult', 'Lorem', 'green', 300, 'assets/img/cocina3.jpg'),
    new Cocina(4, 'Haxom', 'Lorem', 'yellow', 400, 'assets/img/cocina4.jpg')
  ];

  getCocinas(): Cocina[] {
    return this.cocinas;
  }

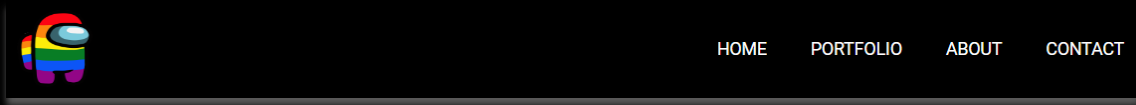
  constructor() { }
}
```

En resumen, este servicio se encarga de proporcionar los datos de las cocinas en la aplicación y puede ser utilizado por diferentes componentes o clases que lo necesiten.

Componente 'Navbar' (app.component)

Creación componente -> Este componente se crea modificando los archivos app originales del proyecto.

Este componente sería el principal de la aplicación y que, por ende, está visible en todas las vistas.



Consiste en un `<nav>` que permite la navegación a los componentes **'Home'**, **'Portfolio'**, **'About'** y **'Contact'** haciendo uso de la propiedad `routerLink` en cada una de las etiquetas `<a>`

En su **archivo HTML** se define una estructura de varios `<div>` que contienen el logo de la aplicación y el menú de navegación `<nav>` con los enlaces a los distintos componentes mencionados previamente.

```
<div id="header--container">
  <div id="header--logo">
    
  </div>
  <nav id="header--nav">
    <ul>
      <li><a routerLink="/">HOME</a></li>
      <li><a routerLink="/portfolio">PORTFOLIO</a></li>
      <li><a routerLink="/about">ABOUT</a></li>
      <li><a routerLink="/contact">CONTACT</a></li>
    </ul>
  </nav>
</div>
<router-outlet></router-outlet>
```

En su fichero **modules.ts** se define el Routing para poder hacer la navegación de la página posible, aquí se importan todos los componentes que componen la aplicación, y se define una variable llamada **'rutas'** que contiene un array con objetos, los cuales son las diferentes rutas a los componentes.

```
import { AppComponent } from './app.component';
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 { DisplayComponent } from './display/display.component';
import { PrecioComponent } from './precio/precio.component';
import { Routes, RouterModule } from '@angular/router';

const rutas: Routes = [
  { path: '', component: HomeComponent },
  { path: 'portfolio', component: PortfolioComponent },
  { path: 'about', component: AboutComponent },
  { path: 'contact', component: ContactComponent },
  { path: 'display/:ref', component: DisplayComponent },
  { path: 'precio/:ref', component: PrecioComponent }
];
```

También se importan funcionalidades de Angular tales como:

- `FormsModule`, para hacer uso de la directiva `[(ngModel)]` en toda la aplicación en caso de ser necesario.
- `Routes & RouterModule` para poder activar las distintas funcionalidades que permiten moverse entre las diferentes vistas de la aplicación.

```
import { FormsModule } from '@angular/forms';
```

```
import { Routes, RouterModule } from '@angular/router';
```

En el apartado imports de la directiva `@NgModule` se importan `FormsModule`, y `RouterModule.forRoot(rutas)`

```
imports: [  
  BrowserModule, FormsModule, RouterModule.forRoot(rutas)  
],
```

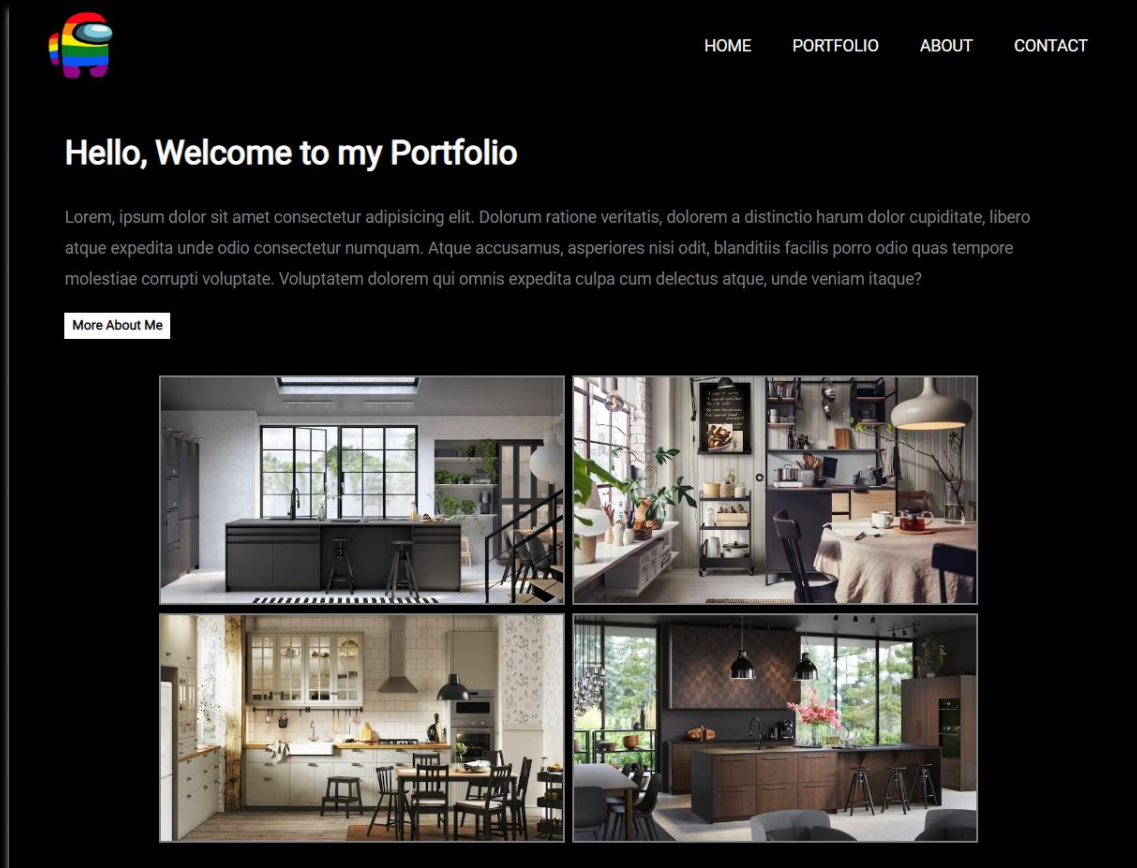
Fichero CSS

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');  
  
* {  
  font-family: 'Roboto', sans-serif;  
}  
  
#header--container {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  height: 90px;  
  width: 100%;  
}  
  
#header--logo {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  margin-left: 20%;  
}  
  
#header--logo img {  
  height: 80px;  
  width: 100px;  
  padding: 5px;  
}  
  
#header--nav {  
  margin-right: 20%;  
}  
  
#header--nav ul {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  list-style: none;  
}  
  
#header--nav ul li {  
  padding: 20px;  
}  
  
#header--nav ul li a {  
  text-decoration: none;  
}
```


Componente 'Home'

Creación componente -> `ng g c Home`

Este componente es la página de inicio de la aplicación. Es responsable de mostrar un mensaje de bienvenida y un grid de imágenes de las diferentes cocinas.



En su **archivo HTML** se define una estructura de varios `<div>` que contienen una caja con un título, un párrafo de bienvenida, un grid de imágenes que muestra las fotos de las diferentes cocinas y un botón que lleva al usuario a la página/componente '**About**'.

El grid de imágenes está generado a través de la directiva `*ngFor` de Angular, que permite repetir un fragmento de HTML para cada elemento de un array.

```
<div id="home--wrapper">
  <div id="home--container">
    <div class="home--container--box">
      <h1>Hello, Welcome to my Portfolio</h1>
      <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dolorum ratione veritatis, dolorem a distinctio harum dolor cupiditate, libero
        harum dolor cupiditate, libero atque expedita unde odio consectetur numquam. Atque accusamus, asperiores nisi odit, blanditiis facilis porro odio quas tempore molestiae corrupti voluptate. Voluptatem dolorem
        qui omnis expedita culpa cum delectus atque, unde veniam itaque?</p>
      <button routerLink="/about" class="moreButton">More About Me</button>
    </div>
    <div class="home--container--grid">
      <ng-container *ngFor="let cocina of cocinas">
        
      </ng-container>
    </div>
  </div>
</div>
```

Su archivo **TypeScript** se encarga de obtener las imágenes a partir del servicio **'CocinasService'**.

En su constructor, se establece que la propiedad/variable **'cocinas'** del componente reciba el valor retornado por el método **'getCocinas()'** del servicio **'CocinasService'**.

```
import { Component, OnInit } from '@angular/core';
import { CocinasService } from '../services/cocinas.service';

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

  cocinas: any[] = [];

  constructor(private cocinasService: CocinasService) {
    this.cocinas = this.cocinasService.getCocinas();
  }

  ngOnInit(): void {
  }
}
```

En resumen, el componente 'Home' se encarga de mostrar una página de bienvenida que contiene una caja con un título y un párrafo de bienvenida, un botón que lleva al usuario a la página/componente **'About'** y un grid que muestra imágenes de cocinas, las cuales son obtenidas a través del servicio **'CocinasService'**.

Fichero CSS

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');

* {
  font-family: 'Roboto', sans-serif;
}

#home--wrapper {
  width: 80%;
  height: 100%;
  margin: 0 auto;
  padding: 10px;
}

#home--container {
  width: 100%;
  display: grid;
  grid-template-rows: 1fr;
  gap: 1rem;
  padding: 10px;
}

#home--container--box {
  width: 75%;
  margin: 0 auto;
  padding: 10px;
  display: grid;
  grid-template-rows: 1fr;
  gap: 0.5rem;
}

#home--container--box h1 {
  padding: 10px;
}

#home--container--box p {
  padding: 10px;
  line-height: 30px;
  color: gray;
}

#home--container--box button {
  width: 8em;
  padding: 5px;
  margin-left: 1%;
  border: none;
  background-color: white;
  color: black;
  font-weight: bolder;
  font-size: 0.8rem;
  cursor: pointer;
}

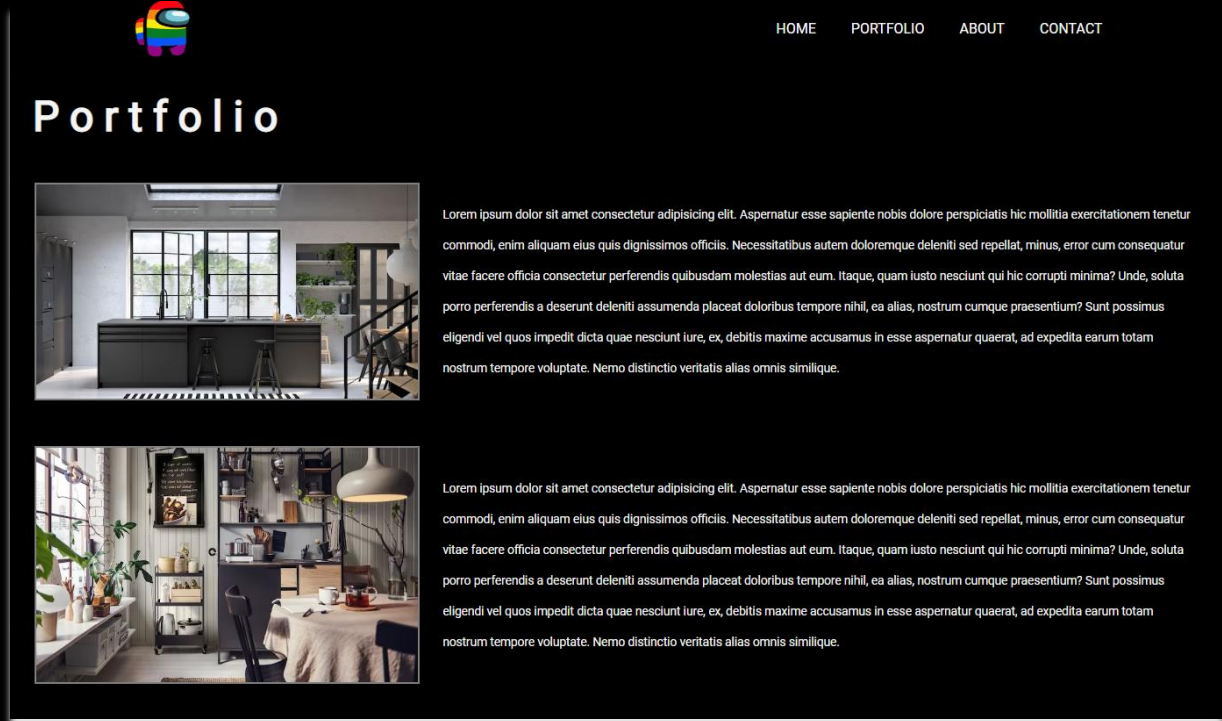
#home--container--grid {
  margin: 0 auto;
  display: grid;
  grid-template-columns: 1fr 1fr;
  place-items: center;
  width: 60%;
  padding: 10px;
  gap: 0.5rem;
}

#home--container--grid img {
  width: 100%;
  border: 2px solid gray;
}
```

Componente 'Portfolio'

Creación componente -> `ng g c Portfolio`

El componente 'Portfolio' es una sección en la aplicación que muestra las diferentes cocinas disponibles.



En su archivo HTML se define una estructura de varios `<div>` que contienen una caja con un título y un grid que muestra las distintas cocinas con su foto y descripción.

Utiliza una directiva `*ngFor` para iterar a través del array `'cocinas'` y mostrar cada una en una tarjeta con una imagen y una descripción.

También tiene un manejador de eventos `'doble click'` en las imágenes que navegará al componente `'Display'` pasando el atributo `"ref"` de la cocina como un parámetro.

```
<div id="portfolio--wrapper">
  <div id="portfolio--container">
    <h2>P o r t f o l i o</h2>
    <ng-container *ngFor="let cocina of cocinas">
      <div class="cocina--card">
        
        <p>{{cocina.desc}}</p>
      </div>
    </ng-container>
  </div>
</div>
```

Este componente importa el servicio **'CocinasService'** y el módulo **'Router'** de Angular, también utiliza la inyección de dependencias para acceder a la lista de cocinas y navegar a otras vistas/componentes.

```
import { Component, OnInit } from '@angular/core';
import { CocinasService } from '../services/cocinas.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-portfolio',
  templateUrl: './portfolio.component.html',
  styleUrls: ['./portfolio.component.css']
})
export class PortfolioComponent implements OnInit {

  cocinas: any[] = [];

  constructor(private cocinasService: CocinasService, private router: Router) {
    this.cocinas = this.cocinasService.getCocinas();
  }

  display(cocina: number) {
    this.router.navigate(['/display', cocina]);
  }

  ngOnInit(): void {
  }
}
```

En general, el componente **'Portfolio'** es responsable de mostrar las imágenes y descripciones de las cocinas y navegar al componente **'Display'** para ver más detalles sobre una cocina en particular cuando se hace un doble clic en la imagen.

Fichero CSS

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');

* {
  font-family: 'Roboto', sans-serif;
}

#portfolio--wrapper {
  margin: 0 auto;
  padding: 10px;
  width: 80%;
  height: 100%;
  margin-top: 0.6%;
}

#portfolio--container {
  display: grid;
  grid-template-rows: 0.5fr 1fr;
  place-items: center;
  height: 100%;
  width: 100%;
  gap: 2rem;
}

h2 {
  font-size: 3rem;
  color: #whitesmoke;
  width: 85%;
  margin: 0 auto;
  margin-right: 9.5%;
}

.cocina--card {
  margin: 0 auto;
  width: 90%;
  height: 100%;
  display: grid;
  grid-template-columns: 1fr 2fr;
  place-items: center;
  gap: 1rem;
  padding: 10px;
}

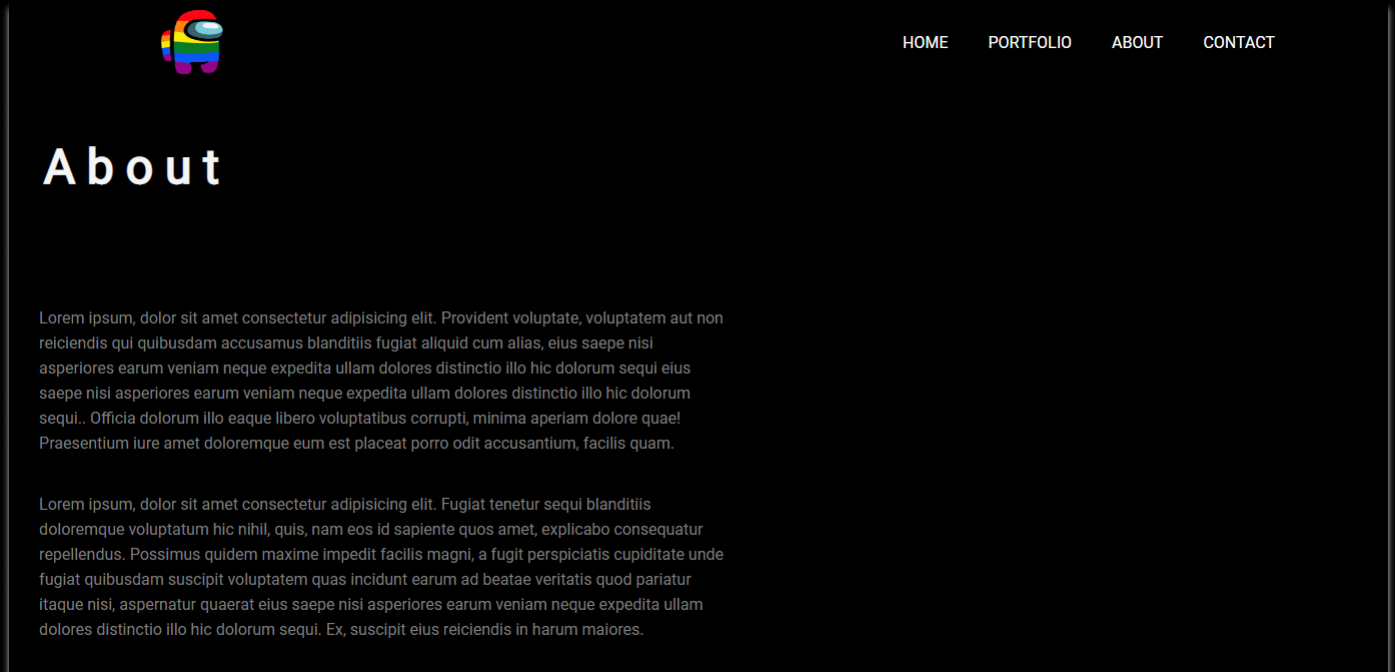
.cocina--img {
  border: 2px solid #gray;
  width: 100%;
  height: 100%;
  cursor: pointer;
}

.cocina--card p {
  color: #whitesmoke;
  line-height: 35px;
  font-size: 0.85rem;
  padding: 10px;
}
```

Componente 'About'

Creación componente -> `ng g c About`

El componente **'About'** se encarga de mostrar información acerca de lo que sería el negocio/empresa propietaria de las cocinas.



En este caso, se puede ver que, en el contenido HTML, se muestra un `<div>` con un título y un par de párrafos. El contenido de estos párrafos puede ser reemplazado por el texto deseado para describir la empresa y sus valores, por ejemplo.

```
<div id="about--wrapper">
  <div id="about--container">
    <div class="about--container--box">
      <h2>A b o u t</h2>
      <div class="about--container--text">
        <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Provident voluptate, voluptatem aut non reiciendis qui quibusdam accusamus blanditiis fugiat aliquid cum alias, eius saepe nisi asperiores earum veniam neque expedita ullam dolores distinctio illo hic dolorum sequi eius saepe nisi asperiores earum veniam neque expedita ullam dolores distinctio illo hic dolorum sequi.. Officia dolorum illo eaque libero voluptatibus corrupti, minima aperiam dolore quae! Praesentium iure amet doloremque eum est placeat porro odit accusantium, facilis quam.</p>
        <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Fugiat tenetur sequi blanditiis doloremque voluptatum hic nihil, quis, nam eos id sapiente quos amet, explicabo consequatur repellendus. Possimus quidem maxime impedit facilis magni, a fugit perspiciatis cupiditate unde fugiat quibusdam suscipit voluptatem quas incidunt earum ad beatae veritatis quod pariatur itaque nisi, aspernatur quaerat eius saepe nisi asperiores earum veniam neque expedita ullam dolores distinctio illo hic dolorum sequi. Ex, suscipit eius reiciendis in harum maiores.</p>
      </div>
    </div>
  </div>
</div>
```

El componente no tiene ninguna funcionalidad adicional en este momento, ya que solo muestra texto estático. Sin embargo, se pueden agregar funciones adicionales en el futuro si es necesario.

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-about',
  templateUrl: './about.component.html',
  styleUrls: ['./about.component.css']
})
export class AboutComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

Fichero CSS

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');

* {
  font-family: 'Roboto', sans-serif;
}

#about--wrapper {
  width: 80%;
  margin: 0 auto;
  padding: 10px;
}

#about--container {
  width: 100%;
  display: grid;
  grid-template-rows: 1fr;
  gap: 1rem;
  padding: 15px;
}

.about--container--box {
  margin: 0 auto;
  padding: 15px;
  display: grid;
  grid-template-rows: 1fr;
  gap: 5em;
}

#about--container h2 {
  color: ■whitesmoke;
  font-size: 3rem;
  padding: 10px;
  width: 80%;
  margin-left: 3%;
}

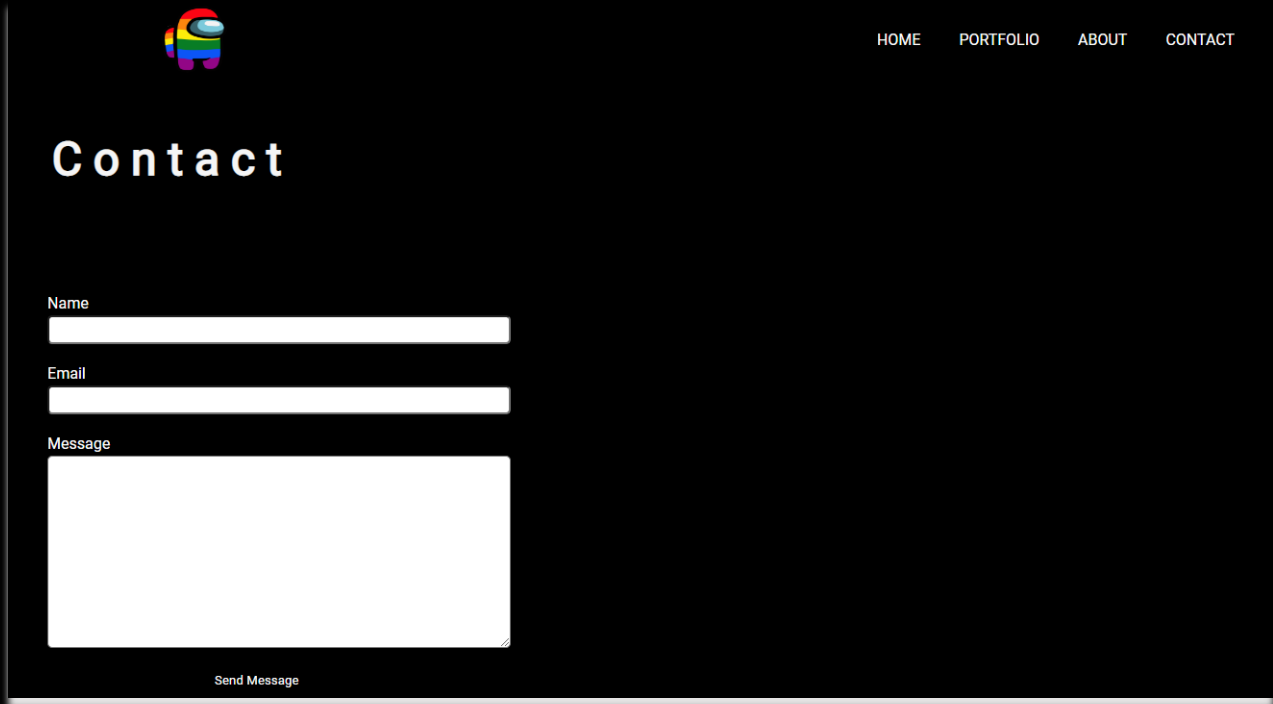
.about--container--text {
  padding: 10px;
  display: grid;
  grid-template-columns: 1fr;
  gap: 1rem;
  width: 50%;
  margin-left: 2%;
}

.about--container--text p {
  padding: 10px;
  line-height: 25px;
  color: ■gray;
}
```

Componente 'Contact'

Creación componente -> `ng g c Contact`

El componente 'Contact' se encarga de mostrar un formulario de contacto al usuario.



El archivo **HTML** contiene una estructura de varios `<div>` que define la estructura de la página de contacto, compuesta por un título y un formulario que contiene tres campos para que el usuario introduzca su nombre, correo electrónico y un mensaje.

También hay un botón de "Enviar mensaje" que, cuando se hace clic, permite enviar el formulario.

```
<div id="contact--wrapper">
  <div id="contact--container">
    <h2>C o n t a c t</h2>
    <div class="contact--container--box">
      <form class="contact--form">
        <label for="name">Name</label><br>
        <input type="text" id="name" name="name" /><br><br>

        <label for="email">Email</label><br>
        <input type="email" id="email" name="email" /><br><br>

        <label for="message">Message</label><br>
        <textarea id="message" name="message"></textarea><br><br>

        <button class="sendMessageButton" type="submit">Send Message</button>
      </form>
    </div>
  </div>
</div>
```

Este componente es un componente básico y no tiene funcionalidades complejas.

Se utiliza principalmente para mostrar un formulario de contacto al usuario y permitirle enviar un mensaje a través del formulario.

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-contact',
  templateUrl: './contact.component.html',
  styleUrls: ['./contact.component.css']
})
export class ContactComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

Este componente puede ser utilizado para proporcionar una funcionalidad para enviar un mensaje de contacto.

Fichero CSS

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');

* {
  font-family: 'Roboto', sans-serif;
}

#contact--wrapper {
  width: 80%;
  margin: 0 auto;
  padding: 10px;
}

#contact--container {
  width: 100%;
  display: grid;
  grid-template-rows: 1fr;
  gap: 5rem;
  padding: 30px;
}

#contact--container h2 {
  color: whitesmoke;
  font-size: 3rem;
  padding: 10px;
  width: 80%;
  margin-left: 3%;
}

#contact--container--box {
  width: 50%;
  padding: 10px;
  margin-left: 2%;
}

.contact--form {
  padding: 10px;
}

.contact--form label {
  line-height: 1.5em;
}

.contact--form input, textarea {
  background-color: white;
  border-radius: 5px;
  width: 70%;
  padding: 5px;
}

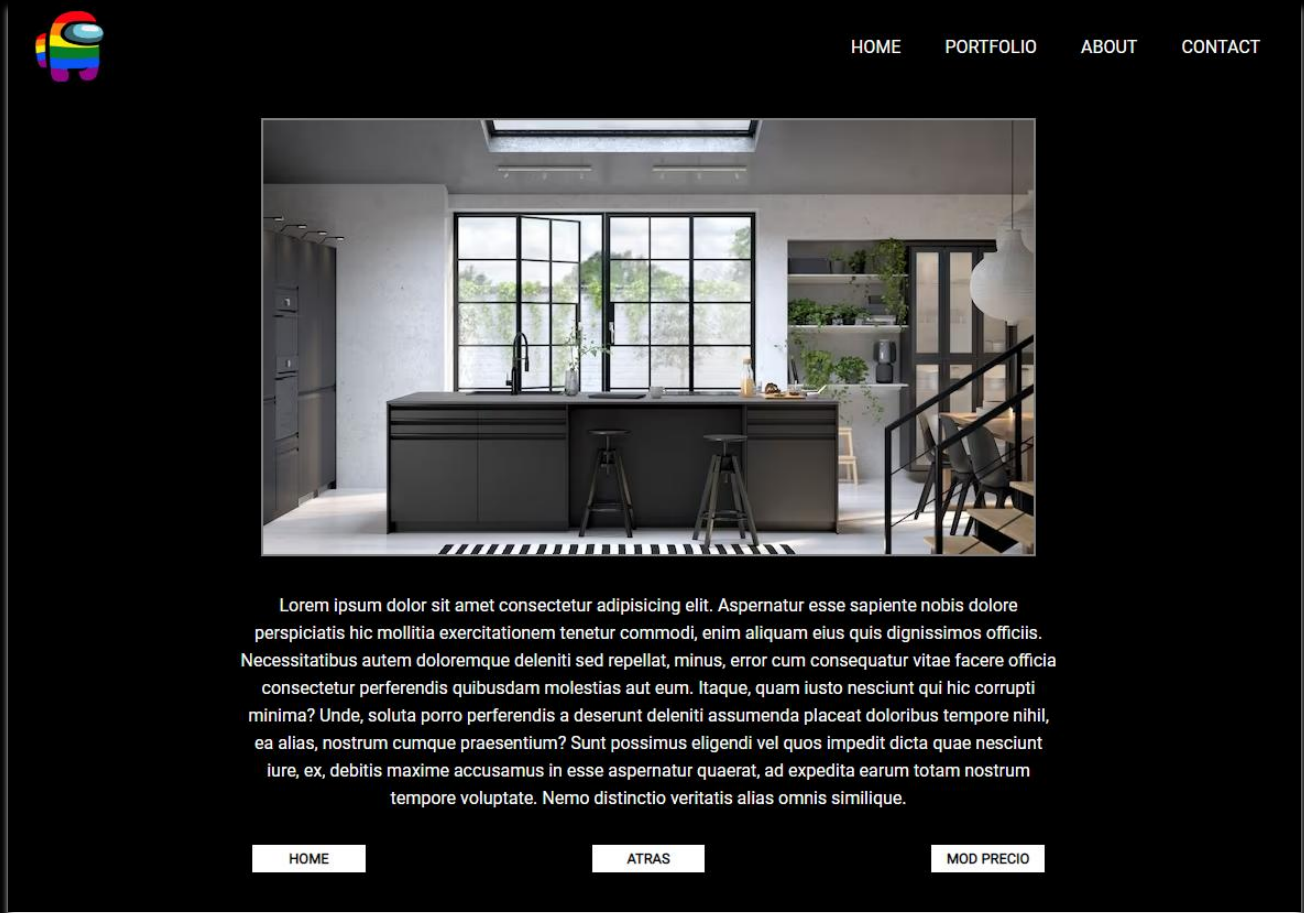
.contact--form textarea {
  height: 15em;
}

.sendMessageButton {
  cursor: pointer;
  margin-left: 25%;
}
```


Componente 'Display'

Creación componente -> `ng g c Display`

El componente **'Display'** se encarga de mostrar la información detallada de una cocina en particular.



Esta información es obtenida del servicio **'CocinasService'** que consiste en una imagen ampliada y una descripción de la cocina.

El **fichero HTML** define una estructura que se repite para cada elemento del array **'cocinas'**.

Se utiliza la directiva ***ngFor** para iterar sobre el array de **'cocinas'** y mostrar solo la información de la cocina que su atributo **"ref"** coincida con el valor de la variable **"refCocina"**, la cual utiliza la ruta activa (ActivatedRoute) para obtener el valor.

Además, el componente también tiene tres botones para navegar a diferentes secciones de la aplicación:

"HOME" para regresar a la página principal (**routerLink="/"**).

"ATRAS" para regresar a la página anterior (**goBack()**).

"MOD PRECIO" para navegar a la sección de modificación de precios de la cocina (**precio()**).

```
<ng-container *ngFor="let cocina of cocinas">
  <ng-container>
    <div *ngIf="refCocina == cocina.ref" id="display--container">
      
      <p>{{cocina.desc}}</p>
    </div>
  </ng-container>
</ng-container>
<ng-container *ngFor="let cocina of cocinas">
  <div class="box--button" *ngIf="refCocina == cocina.ref">
    <button routerLink="/">HOME</button>
    <button (click)="goBack()">ATRAS</button>
    <button (click)="precio(cocina.ref)">MOD PRECIO</button>
  </div>
</ng-container>
```

En su **fichero TypeScript**, la lógica de navegación se maneja mediante los servicios **'ActivatedRoute'**, **'Location'** y **'Router'** de Angular, los cuales permiten obtener información de la ruta actual, regresar a la página anterior y navegar a otras páginas de la aplicación, respectivamente.

```
import { Component, OnInit } from '@angular/core';
import { CocinasService } from '../services/cocinas.service';
import { ActivatedRoute } from '@angular/router';
import { Location } from '@angular/common';
import { Router } from '@angular/router';

@Component({
  selector: 'app-display',
  templateUrl: './display.component.html',
  styleUrls: ['./display.component.css']
})
export class DisplayComponent implements OnInit {

  refCocina: any;

  cocinas: any[] = [];

  goBack() {
    this.location.back();
  }

  precio(cocina: number) {
    this.router.navigate(['/precio', cocina]);
  }

  constructor(private cocinasService: CocinasService, private activateRoute: ActivatedRoute, private location: Location, private router: Router) {
    this.cocinas = this.cocinasService.getCocinas();
  }

  ngOnInit(): void {
    this.refCocina = this.activateRoute.snapshot.params['ref'];
  }
}
```

Fichero CSS

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');

* {
  font-family: 'Roboto', sans-serif;
}

#display--container {
  margin: 0 auto;
  width: 80%;
  display: grid;
  place-items: center;
  grid-template-rows: 1fr;
  gap: 2rem;
  padding: 20px;
}

.box--button button {
  cursor: pointer;
  width: 8em;
  padding: 5px;
  border: none;
  background-color: #white;
  color: #black;
  font-weight: bolder;
  font-size: 0.8rem;
}

#display--container img {
  border: 2px solid #gray;
}

#display--container p {
  text-align: center;
  line-height: 25px;
  width: 50%;
}

.box--button {
  display: flex;
  justify-content: space-evenly;
  align-items: center;
  padding: 10px;
  width: 60%;
  margin: 0 auto;
}
```

Componente 'Precio'

Creación componente -> `ng g c Precio`

El componente 'Precio' se encarga de mostrar el precio de una cocina específica y permitir su modificación.



En el **archivo HTML** se utiliza la directiva `*ngFor` para iterar sobre el array `'cocinas'` y mostrar la información de la cocina que coincide con el parámetro `"ref"` en la URL el cual guardamos en la variable `'refCocinas'`.

Se muestra el nombre y la referencia de la cocina, el precio actual, y un input para ingresar un nuevo precio. Además, se incluyen dos botones, uno para cambiar el precio y otro para cerrar la página.

```
<div id="precio--wrapper">
  <ng-container *ngFor="let cocina of cocinas">
    <div class="precio--container" *ngIf="refCocina == cocina.ref">
      <h2>Cocina '{{cocina.name}}' con referencia '{{cocina.ref}}'</h2><br><br>
      <p>Precio actual: {{cocina.price}}€</p><br>
      <input type="text" [(ngModel)]="newPrice" /><br><br>
      <div class="box--button">
        <button (click)="modPrice(cocina.ref)">CAMBIAR PRECIO</button>
        <button routerLink="/portfolio">CERRAR</button>
      </div>
    </div>
  </ng-container>
</div>
```

En el archivo **TypeScript** que contiene la lógica del componente encontramos lo siguiente:

Se importan las dependencias necesarias, como el servicio **'CocinasService'** y **'ActivatedRoute'**, y se inicializan en el constructor.

El método **'modPrice()'** se encarga de modificar el precio de la cocina.

Se utiliza un bucle **forEach** para encontrar la cocina que coincide con la referencia enviada en la URL, la cual es almacenada en la variable **'refCocinas'** y se actualiza su precio con el valor ingresado en el input.

Se muestran alertas para confirmar si se ha modificado o no el precio.

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { CocinasService } from '../services/cocinas.service';

@Component({
  selector: 'app-precio',
  templateUrl: './precio.component.html',
  styleUrls: ['./precio.component.css']
})
export class PrecioComponent implements OnInit {

  newPrice: string = "0";

  refCocina: any;

  cocinas: any[] = [];

  modPrice(ref: number) {
    if (confirm("¿Estás seguro de querer modificar el precio?")) {
      this.cocinas.forEach(valor => {
        if (valor.ref == ref) {
          valor.price = parseInt(this.newPrice);
        }
      });
      alert("Precio modificado");
    } else {
      alert("No se ha modificado el precio");
    }
    console.table(this.cocinas);
  }

  constructor(private cocinasService: CocinasService, private activateRoute: ActivatedRoute) {
  }

  ngOnInit(): void {
    this.cocinas = this.cocinasService.getCocinas();
    this.refCocina = this.activateRoute.snapshot.params['ref'];
  }
}
```

Fichero CSS

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');

* {
  font-family: 'Roboto', sans-serif;
}

#precio--wrapper {
  margin: 0 auto;
  margin-top: 3%;
  width: 80%;
  display: grid;
  place-items: center;
  grid-template-rows: 1fr;
  gap: 2rem;
  padding: 20px;
}

.box--button button {
  cursor: pointer;
  width: 9em;
  padding: 5px;
  border: none;
  background-color: white;
  color: black;
  font-weight: bolder;
  font-size: 0.8rem;
}

.precio--container {
  display: grid;
  grid-template-columns: 1fr;
  place-items: center;
  padding: 20px;
}

.box--button {
  width: 100%;
  display: grid;
  place-items: center;
  grid-template-columns: 1fr 1fr;
  padding: 10px;
}
```