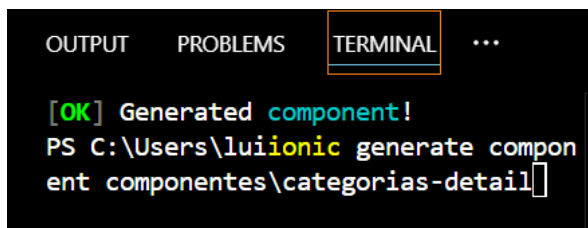


# Crear rutas y enlazar componentes

Para la creación la creación de una ruta la cual vamos a agregar ah nuestra página de Udemty, la cual al momento de darle Clic a cualquiera de una de nuestras categorías nos debe de mandar a uno de nuestros componentes, el cual vamos a enlazar a una ruta y al momento de entrar esto nos debe de proporcionar la ruta en la que estamos navegando.

## Categorías Ruta

El primer componente al cual le agregamos una ruta es al de categorías categorías.



```
OUTPUT  PROBLEMS  TERMINAL  ...  
[OK] Generated component!  
PS C:\Users\luis\nic generate compon  
ent componentes\categorias-detail
```

Una vez que agregamos el componente donde vamos a alojar nuestra ruta debemos agregar una nueva ruta ah nuestro archivo de app routing module.

```
import { NgModule } from '@angular/core';  
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';  
import { TopicDetailComponent } from './componentes/topic-detail/topic-detail.component';  
import { PieDetailComponent } from './componentes/pie-detail/pie-detail.component';  
import { CategoriasDetailComponent } from './componentes/categorias-detail/categorias-detail.component';  
  
const routes: Routes = [  
  {  
    path: 'home',  
    loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)  
  },  
  {  
    path: '',  
    redirectTo: 'home',  
    pathMatch: 'full'  
  },  
  {  
    path: 'centro-comercial',  
    loadChildren: () => import('./centro-comercial/centro-comercial.module').then( m =>  
m.CentroComercialPageModule)  
  },  
  {  
    path: 'topic/:name', // Ruta TOPIC-DETAIL  
    component: TopicDetailComponent,  
  }  
];
```

```

    {
      path: 'pie-detail/:name', // Ruta PIE-DETAIL
      component: PieDetailComponent
    },
    {
      path: 'categorias-detail/:name', // Ruta CATEGORIA-DETAIL
      component: CategoriasDetailComponent
    }
  ];

  @NgModule({
    imports: [
      RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
    ],
    exports: [RouterModule]
  })
  export class AppRoutingModule { }

```

Una vez que pusimos la ruta en el pat y también debemos importar nuestro componente hasta arriba.

```

<ion-content>
  <div class="categories-container">
    <h2>Categorías principales</h2>
    <div class="categories-grid">
      <ion-card *ngFor="let category of categories" class="category-card">
        <img [src]="category.img" alt="{{ category.name }}" />
        <ion-card-content>
          <h3>
            <a [routerLink]="['/categorias-detail', category.name]">{{ category.name }}</a>
          </h3>
        </ion-card-content>
      </ion-card>
    </div>
  </div>
</ion-content>

```

Y por último lo que vamos a agregar o a modificar en el html es el router link para así pasar el nombre de cada categoría a nuestro nuevo componente de html.

```

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

```

```

@Component({
  selector: 'app-categorias-detail',

```

```

    templateUrl: './categorias-detail.component.html',
    styleUrls: ['./categorias-detail.component.scss'],
  })
  export class CategoriasDetailComponent implements OnInit {
    categoryName: string | null = "";

    constructor(private route: ActivatedRoute) {}

    ngOnInit() {
      // Obtener el parámetro 'name' de la ruta
      this.categoryName = this.route.snapshot.paramMap.get('name');
    }
  }

```

En nuestro nuevo componente que creamos el cual se llama categorías-details vamos a convertir categoría name en una cadena y hacemos privada la ruta para así permitir al componente acceda a un parámetro de ruta en este caso categoría name y le agregamos contenido a nuestra nueva ruta que es donde va a parar.

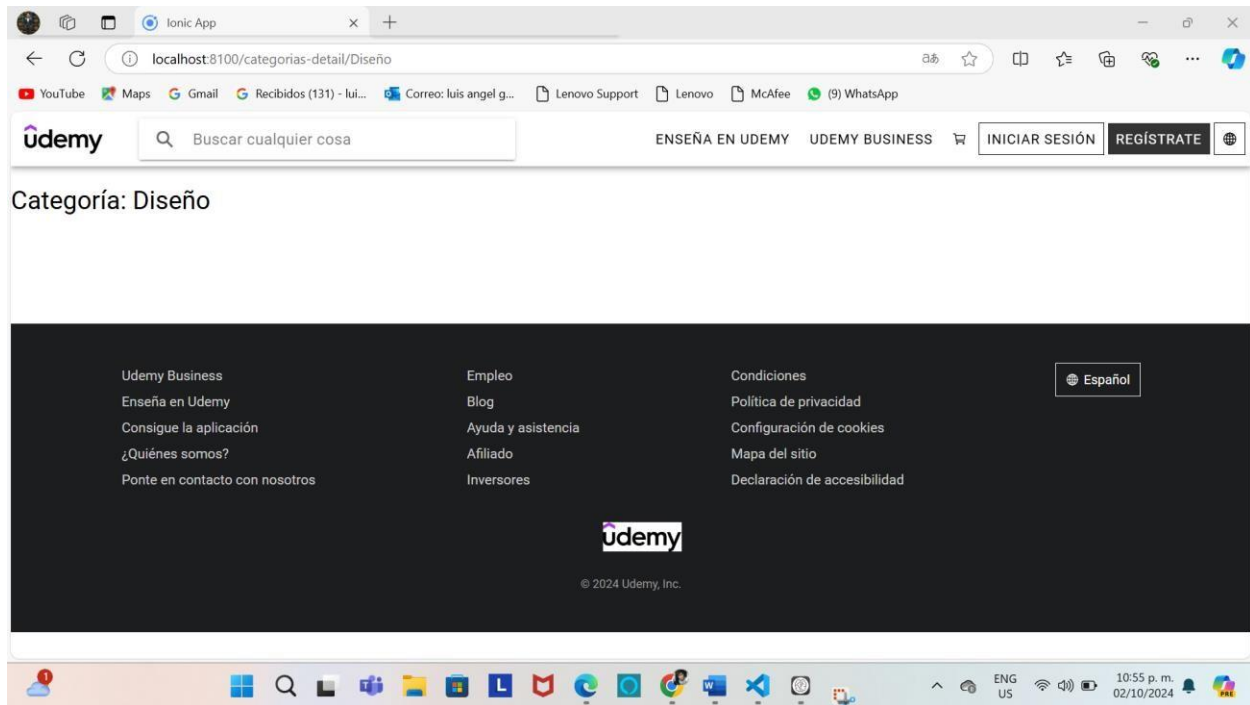
```

<ion-content>
  <app-header></app-header>
  <h1>Categoría: {{ categoryName }}</h1>

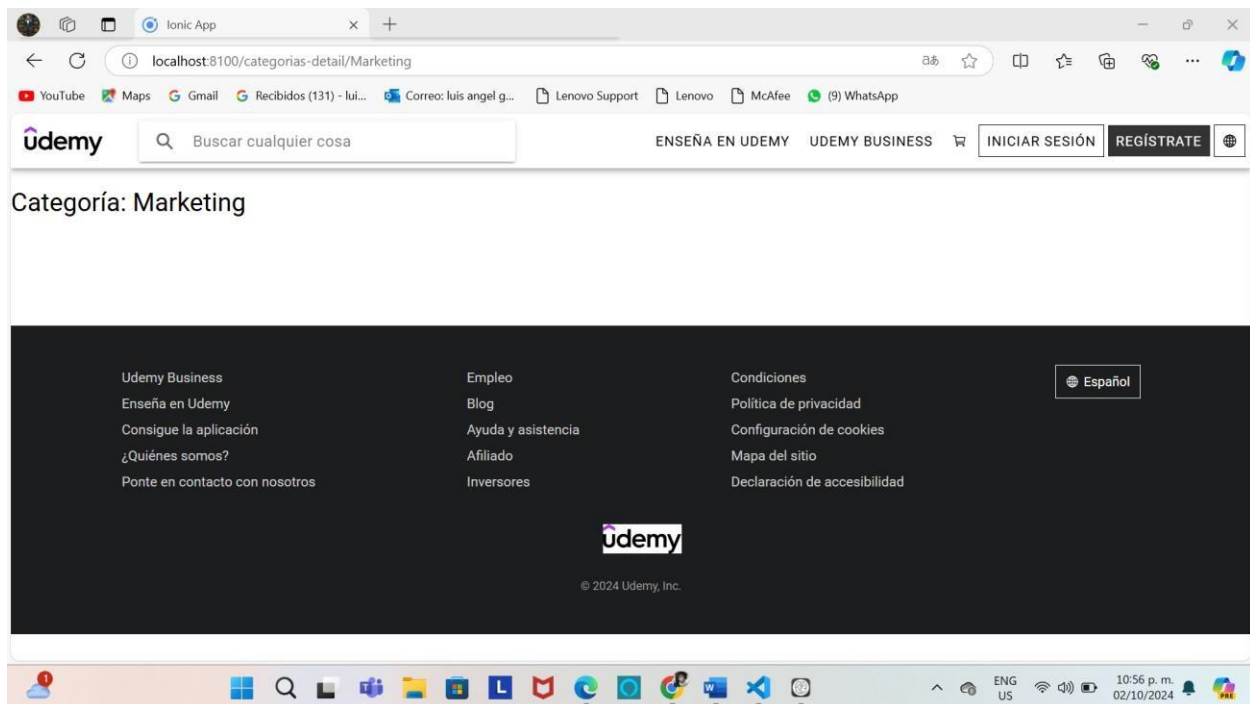
  <div class="spacer"></div>
  <app-pie> </app-pie>
</ion-content>

```

Y una cada vez que entremos o le damos Clic a nuestro botón no es arrojará una nueva ruta y un nuevo nombre del tema al botón que le demos.



Y si le damos en Clic en otra imagen que tengo otro nombre nos debe aparecer el mismo nombre en la página y con otra ruta



## Temas Ruta

Y para iniciar con la segunda ruta qué vamos a crear en nuestra página de udemy va a ser el componente de temas el cual vamos a agregar un nuevo componente para alojar nuestra nueva ruta ahí.

```
OUTPUT  PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
○ PS C:\Users\luian\OneDrive\Escritorio\Ionic\udemy> ionic generate component componentes\topic-detail
```

Una vez agregado nuestro nuevo componente de topic detail vamos a alojar una ruta en nuestras rutinas módulo

```
{
  path: 'topic/:name', // Ruta TOPIC-DETAIL
  component: TopicDetailComponent,
},
```

Una vez que agregamos la ruta en el pad y importamos nuestra categoría de de topic details podemos continuar

```
<ion-content>
  <div class="highlighted-themes-container">
    <h2>Temas destacados por categoría</h2>

    <div class="themes-grid">
      <div *ngFor="let category of categories" class="category-column">
        <h3>{{ category.name }}</h3>
        <ul>
          <li *ngFor="let topic of category.topics">
            <a [routerLink]="['/topic', topic.name]">{{ topic.name }}</a>
            <p>{{ topic.learners }} learners</p>
          </li>
        </ul>
      </div>
    </div>

    <ion-button expand="outline" class="discover-button">
      Descubrir más temas
    </ion-button>
  </div>
</ion-content>
```

En nuestro componente de temas en el HTML lo que le vamos a cambiar va a ser el router link que es un componente para redirigir lo que va a redirigir va a ser el nombre de cada uno de los temas.

Por y en la línea 10 vamos a agregar router link y cuál este parámetro nos va a ayudar a llamar el nombre de cada una de las categorías y ponerlo en el html the topic details

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

@Component({
  selector: 'app-topic-detail',
  templateUrl: './topic-detail.component.html',
  styleUrls: ['./topic-detail.component.scss'],
})
export class TopicDetailComponent implements OnInit {
  topicName: string | null = "";
  topicLearners: string | null = "";

  constructor(private route: ActivatedRoute) {}

  ngOnInit() {
    // Obtener el parámetro 'name' de la ruta
    this.topicName = this.route.snapshot.paramMap.get('name');
    this.topicLearners = this.route.snapshot.paramMap.get('learners');
  }
}
```

Y en el TS de nuestro nuevo componente que creamos de topic details vamos a agregar topic name y lo vamos a construir en una cadena así como vamos a declarar private y a obtener el parámetro de la ruta name donde también agregamos nuestras rutas.

Y con esto queda de la siguiente manera:

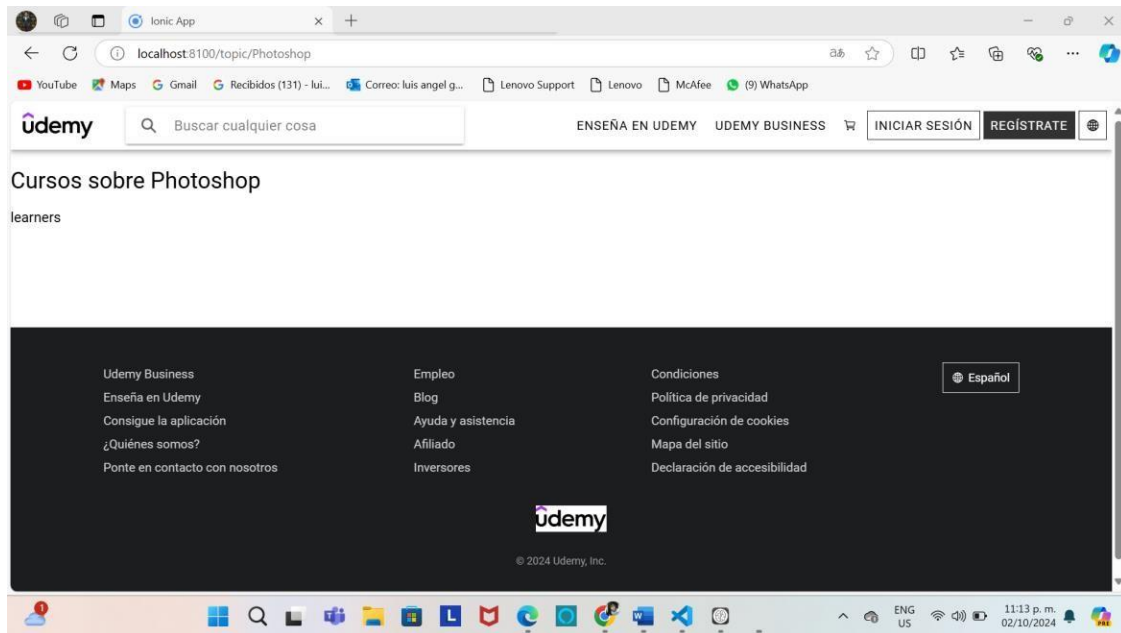
```
<ion-content>
  <app-header></app-header>
  <h1>Cursos sobre {{ topicName }}</h1>
  <p>{{ topicLearners }} learners</p>
```

<div class="spacer"></div>

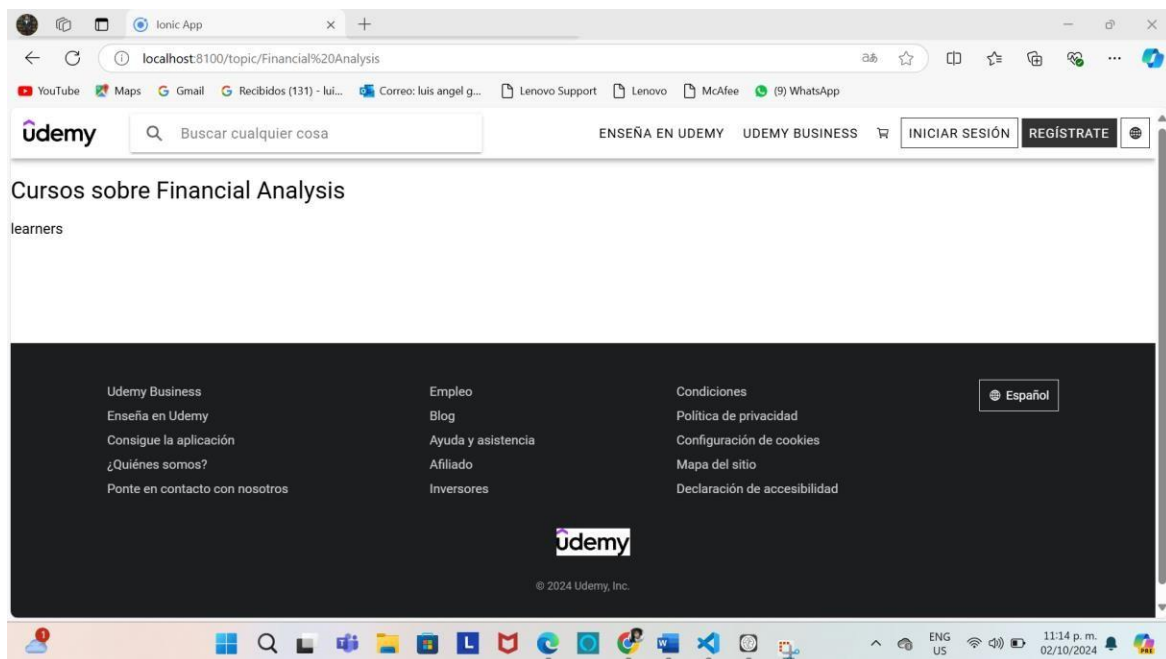
<app-pie></app-pie>

</ion-content>

Y se visualiza de la siguiente manera en donde le das sobre cada uno de los demás te voy a salir una nueva ruta y un diferente nombre por cada curso que te metas.



Otro tema:



## Pie Ruta

Y por último en el último componente el cual agregamos para el pie de página fue el de pie detail

```
OUTPUT  PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
PS C:\Users\luian\OneDrive\Escritorio\Ionic\udemy> ionic generate component componentes\pie-detail
```

Agregamos la ropa de pie detail

```
{
  path: 'pie-detail/:name', // Nueva ruta dinámica para 'PieDetailComponent'
  component: PieDetailComponent
},
```

y también la importamos como cada una de las rutas

```
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
import { TopicDetailComponent } from './componentes/topic-detail/topic-detail.component';
import { PieDetailComponent } from './componentes/pie-detail/pie-detail.component';
import { CategoriasDetailComponent } from './componentes/categorias-detail/categorias-detail.component';

const routes: Routes = [
  {
    path: 'home',
    loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)
  },
  {
    path: '',
    redirectTo: 'home',
    pathMatch: 'full'
  },
  {
    path: 'centro-comercial',
    loadChildren: () => import('./centro-comercial/centro-comercial.module').then( m =>
m.CentroComercialPageModule)
  },
  {
    path: 'topic/:name', // Ruta TOPIC-DETAIL
    component: TopicDetailComponent,
  },
  {
    path: 'pie-detail/:name', // Ruta PIE-DETAIL
```



```

    component: PieDetailComponent
  },
  {
    path: 'categorias-detail/:name', // Ruta CATEGORIA-DETAIL
    component: CategoriasDetailComponent
  }
];

@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModule { }

```

Y en nuestro HTML de pie agregamos el router link para así redirigir nuestros nombres de cada una de ellas y aparezca al momento de abrir o de dar Clic en cada uno de los botones de pie de página.

```

<footer class="footer">
  <div class="footer-links">
    <div *ngFor="let section of footerSections" class="column">
      <a *ngFor="let link of section.links" [routerLink]="['/pie-detail', link.name]">{{ link.name }}</a>
    </div>
    <div class="column">
      <button class="btn-language">
        <ion-icon name="globe-outline"></ion-icon> {{ language }}
      </button>
    </div>
  </div>

  <div class="footer-logo">
    <img [src]="footerLogo.src" [alt]="footerLogo.alt" />
    <p>© 2024 Udemy, Inc.</p>
  </div>
</footer>

```

En nuestro html de pie de página en la línea cuatro lo que agregamos fue el router link para así redirigir el nombre de cada uno de los detalles de pie de página.

```

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

```

```

@Component({
  selector: 'app-pie-detail',
  templateUrl: './pie-detail.component.html',

```

```

    styleUrls: ['./pie-detail.component.scss'],
  })
  export class PieDetailComponent implements OnInit {
    name: string | null = "";

    constructor(private route: ActivatedRoute) {}

    ngOnInit() {
      // Obtener el parámetro 'name' de la URL
      this.name = this.route.snapshot.paramMap.get('name');
    }
  }
}

```

Sí filtrará el name de cada uno de los parámetros de pie de página para así nos mande al nuevo componente creado con su nombre sólo manda el name del código de pie de página no el de detail pie

```

<ion-content>
  <app-header> </app-header>
  <h1>Detalles de {{ name }}</h1>
  <div class="spacer"></div>

  <app-pie></app-pie>
</ion-content>

```

Y a todos los html de los componentes creados para la ruta se les agregó el app-header y pie de página, para que en cada una de las rutas al abrir se visualice también estos componentes como lo es en la página de mi oficial.

Y así se logra visualizar en el ionic serve que al momento de entrar de alguna en cada uno de los name que se encuentran en el pie de página nos debe dar acceso a la página del nuevo componente y se debe visualizar de una manera donde no es filtre los nombres de cada uno de los detalles de pie de página y no desaparezca con otra ruta.

