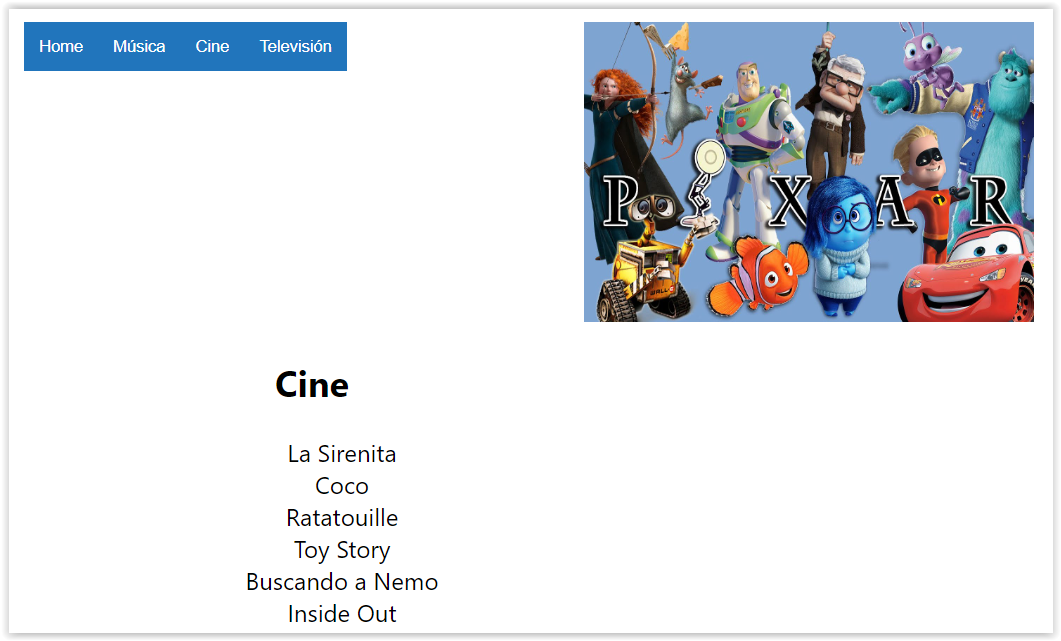
**RUTAS CON ANGULAR**

* Vamos a visualizar cómo podemos cambiar entre componentes con navegación URL en Angular.
* Esto nos permitirá poder cambiar la URL y que cada componente tenga una página por separado, pero sin recargar la página SPA.
* Angular ya tiene instalado Routing por defecto.







Vamos a visualizar una aplicación sin acciones, solamente con componentes visuales para poder navegar entre ellos.

Nos creamos una carpeta **components** dentro de **src/app** y cuatro componentes con diferente diseño para poder navegar entre ellos.

Creamos un componente de Música, otro de Televisión y otro de Cine e incluimos un simple listado en cada uno.

También vamos a crear un component **Home** para la página inicial.

**HOME.COMPONENT.HTML**

<div>

  <h1 style="color: blue;">Ejemplo de Routing, página Home</h1>

  <img

    src="../../../assets/images/home.jpg"

    style="width: 300px; height: 200px;"

  />

</div>

**MUSICA.COMPONENT.HTML**

<div>

  <img

    src="../../../assets/images/music.jpg"

    style="width: 200px; height: 200px;"

  />

  <ul>

    <h2>Lista de canciones</h2>

    <li>Ivan Ferreiro: <span>Turnedo</span></li>

    <li>Izal: <span>La mujer de verde</span></li>

    <li>Supersubmarina: <span>De las dudas infinitas</span></li>

    <li>Love of Lesbian: <span>Allí dónde solíamos gritar</span></li>

    <li>Carlos Sadness: <span>Hoy es el día</span></li>

  </ul>

</div>

**TELEVISION.COMPONENT.HTML**

<div>

  <img

    src="../../../assets/images/television.jpg"

    style="width: 300px; height: 200px;"

  />

  <ul>

    <h2>Televisión</h2>

    <li>

      <span>Como conocí a vuestra madre</span>

    </li>

    <li>

      <span>Stranger Things</span>

    </li>

    <li>

      <span>Chernobil</span>

    </li>

    <li>

      <span>Watchmen</span>

    </li>

  </ul>

</div>

**CINE.COMPONENT.HTML**

<div>

  <h2>Cine</h2>

  <img

    src="../../../assets/images/cine.jpg"

    style="width: 300px; height: 200px;"

  />

  <ul>

    <li>La Sirenita</li>

    <li>Coco</li>

    <li>Ratatouille</li>

    <li>Toy Story</li>

    <li>Buscando a Nemo</li>

    <li>Inside Out</li>

  </ul>

</div>

Para configurar el Router, debemos hacerlo desde el componente principal, por lo que iremos a la carpeta **app** y crearemos un nuevo archivo llamado **app.routing.ts**

**APP.ROUTING.TS**

//IMPORTAMOS LOS MODULOS DEL ROUTER DE ANGULAR

import { ModuleWithProviders } from "@angular/core";

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

//AHORA DEBEMOS IMPORTAR LOS COMPONENTES QUE VAYAN A FORMAR PARTE

//DE LA NAVEGACION, ES DECIR, LOS QUE VISUALIZAREMOS

//DENTRO DE LAS ZONAS DE NUESTRO SITE

//IMPORTAMOS EL COMPONENTE DE HOME

import { HomeComponent } from "./components/home/home.component";

//MUSICA

import { MusicaComponent } from "./components/musica/musica.component";

//CINE

import { CineComponent } from "./components/cine/cine.component";

//TELEVISION

import { TelevisionComponent } from "./components/television/television.component";

import { ErrorComponent } from "./components/error/error.component";

//DEFINIMOS EL ARRAY DE RUTAS, ES DECIR, LA CONFIGURACION DE TODAS

//LAS RUTAS PARA NUESTRA APP

const appRoutes: Routes = [

  { path: "", component: HomeComponent },

  { path: "home", component: HomeComponent },

  { path: "musica", component: MusicaComponent },

  { path: "cine", component: CineComponent },

  { path: "television", component: TelevisionComponent }

  { path: "\*\*", component: ErrorComponent }

];

//AHORA EXPORTAMOS EL MODULO DEL ROUTER PARA PODER

//UTILIZARLO DENTRO DE ANGULAR

export const appRoutingProviders: any[] = [];

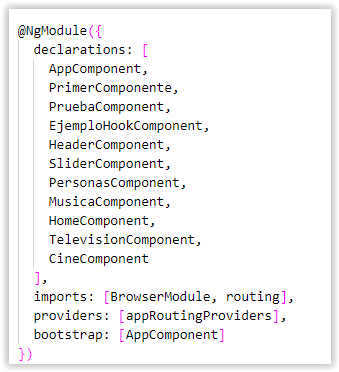
export const routing: ModuleWithProviders = RouterModule.forRoot(appRoutes);

A continuación, abrimos el módulo del componente principal de **app**, para declarar el routing que vayamos a utilizar. Abrimos el archivo **app.module.ts** y declaramos las dos constantes que hemos exportado en la clase de Routing, en la última línea.

**APP.MODULE.TS**

import { routing, appRoutingProviders } from "./app.routing";

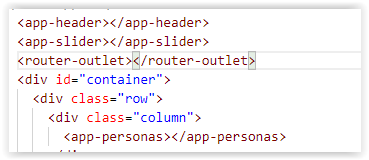
Cargamos **routing** como módulo y **appRoutingProviders** como servicio dentro de la misma clase.



El siguiente paso será utilizar las rutas dentro de la página principal **app.component.html**

Para poder generar rutas, debemos utilizar la etiqueta **<router-outlet>**, que será la etiqueta encargada de buscar las rutas creadas en nuestro Routing.

**APP.COMPONENT.HTML**



Y podremos comprobar las rutas incluyendo los Path en la URL:

<http://localhost:4200/musica>

El último paso es crear un Component de Menú para poder visualizar la navegación correctamente.

**MENU.COMPONENT.CSS**

#header {

  height: 120px;

  width: 100%;

  background: white;

  border-bottom: 1px solid #ccc;

  border-top: 5px solid red;

}

.center {

  width: 75%;

  margin: 0px auto;

}

#logo img {

  display: block;

  float: left;

  height: 80px;

  width: 250px;

}

#logo {

  width: 30%;

  float: left;

  margin-top: 20px;

}

#menu {

  width: 70%;

  float: right;

}

#menu ul {

  list-style-type: none;

}

#menu ul li {

  display: inline;

  margin-left: 15px;

  margin-right: 15px;

}

#menu a {

  text-decoration: none;

}

**MENU.COMPONENT.HTML**

<header id="header">

  <div class="center">

    <div id="logo">

      <img src="./../../../assets/images/angularlogo.png" alt="logo React" />

    </div>

    <nav id="menu">

      <ul>

        <li>

<a [routerLink]="['/home']">Home</a>

        </li>

        <li>

          <a href="musica">Musica</a>

        </li>

        <li>

          <a href="cine">Cine</a>

        </li>

        <li>

          <a href="television">Televisión</a>

        </li>

      </ul>

    </nav>

  </div>

</header>

Por último, incluimos el menú dentro de la página principal App.component.js

**APP.COMPONENT.HTML**

<app-header></app-header>

<app-slider></app-slider>

<div id="container">

  <div class="row">

    <div class="column">

<router-outlet></router-outlet>

    </div>

  </div>

</div>

El siguiente concepto es poder capturar errores de URL dentro de nuestra aplicación, tales como el error 404.

La solución es crear un componente de Error para 404 y llamarlo con Route.

Para indicar una página 404 para rutas, debemos crear un componente **error.component**, por ejemplo con un diseño de 404.

Posteriormente, dentro del Routing, se llama el último path y se utiliza \*\*.

  { path: "\*\*", component: ErrorComponent }

Otra opción también es crear componentes sin rutas. Para ello, podemos cargar directamente código HTML desde el Component Route con la siguiente sintaxis: