Contenido

[1. Crear el proyecto de angular 1](#_Toc141547967)

[a. Creamos la carpeta de angular 1](#_Toc141547968)

[b. Instalamos angular materials 1](#_Toc141547969)

[2. Usamos el schematic del menu 1](#_Toc141547970)

[a. Creamos el schematic del menú 1](#_Toc141547971)

[3. Generamos el componente 2](#_Toc141547972)

[a. Login 2](#_Toc141547973)

[HTMl: 2](#_Toc141547974)

[CSS 3](#_Toc141547975)

[4. Configuracion de rutas 7](#_Toc141547976)

[a. Ruta login 7](#_Toc141547977)

[5. Formularios 7](#_Toc141547978)

[a. Formulario Login 8](#_Toc141547979)

[Ts 8](#_Toc141547980)

# Crear el proyecto de angular

## Creamos la carpeta de angular

ng new FrontEnd

## Instalamos angular materials

ng add @angular/material

ng add @angular/cdk

# Usamos el schematic del menu

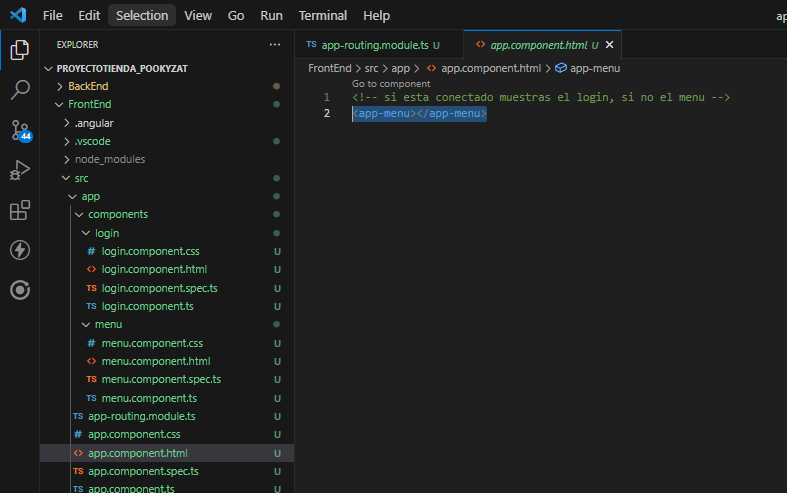
## Creamos el schematic del menú

Creamos una carpeta Components y dentro de esta carpeta ejecutamos el siguiente comando en consola

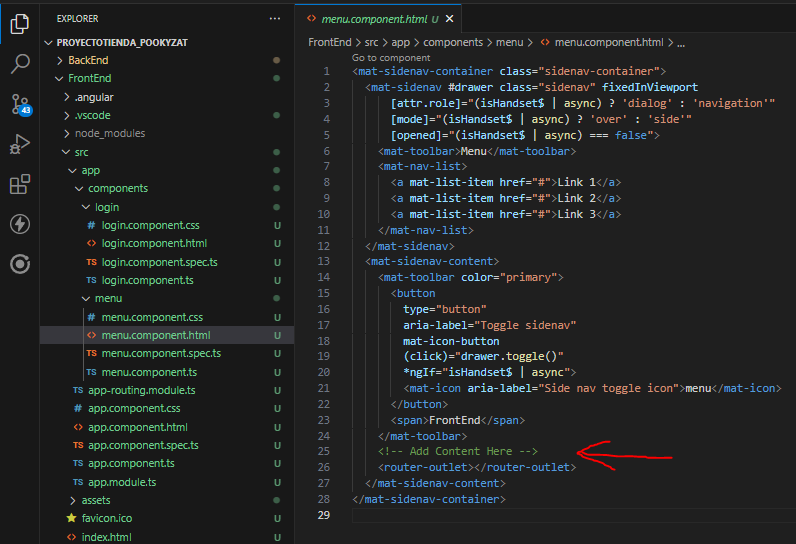
ng generate @angular/material:navigation <component-name> este es el ejemplo que encontré en angular material schematics cuando busqué en Google

* ng generate @angular/material:navigation menu

y en el app.component.html vamos a borrar todo y poner el selector del menú



en el html de este componente vamos a buscar el comentario que dice Add Content Here y agregaremos el <router-outlet></router-outlet>



# **Componente**s

## Login

En la carpeta components escribimos el siguiente comando en consola

* ng g c login

y acomodamos el html, css y ts

### HTMl:

<div class="login-box">

    <h2>Login</h2>

    <form>

      <div class="user-box">

        <input type="text" name="" required="">

        <label>Username</label>

      </div>

      <div class="user-box">

        <input type="password" name="" required="">

        <label>Password</label>

      </div>

      <a href="#">

        <span></span>

        <span></span>

        <span></span>

        <span></span>

        Submit

      </a>

    </form>

</div>

### CSS

html {

    height: 100%;

  }

  body {

    margin:0;

    padding:0;

    font-family: sans-serif;

    background: linear-gradient(#141e30, #243b55);

  }

  .login-box {

    position: absolute;

    top: 50%;

    left: 50%;

    width: 400px;

    padding: 40px;

    transform: translate(-50%, -50%);

    background: rgba(0,0,0,.5);

    box-sizing: border-box;

    box-shadow: 0 15px 25px rgba(0,0,0,.6);

    border-radius: 10px;

  }

  .login-box h2 {

    margin: 0 0 30px;

    padding: 0;

    color: #fff;

    text-align: center;

  }

  .login-box .user-box {

    position: relative;

  }

  .login-box .user-box input {

    width: 100%;

    padding: 10px 0;

    font-size: 16px;

    color: #fff;

    margin-bottom: 30px;

    border: none;

    border-bottom: 1px solid #fff;

    outline: none;

    background: transparent;

  }

  .login-box .user-box label {

    position: absolute;

    top:0;

    left: 0;

    padding: 10px 0;

    font-size: 16px;

    color: #fff;

    pointer-events: none;

    transition: .5s;

  }

  .login-box .user-box input:focus ~ label,

  .login-box .user-box input:valid ~ label {

    top: -20px;

    left: 0;

    color: #03e9f4;

    font-size: 12px;

  }

  .login-box form a {

    position: relative;

    display: inline-block;

    padding: 10px 20px;

    color: #03e9f4;

    font-size: 16px;

    text-decoration: none;

    text-transform: uppercase;

    overflow: hidden;

    transition: .5s;

    margin-top: 40px;

    letter-spacing: 4px

  }

  .login-box a:hover {

    background: #03e9f4;

    color: #fff;

    border-radius: 5px;

    box-shadow: 0 0 5px #03e9f4,

                0 0 25px #03e9f4,

                0 0 50px #03e9f4,

                0 0 100px #03e9f4;

  }

  .login-box a span {

    position: absolute;

    display: block;

  }

  .login-box a span:nth-child(1) {

    top: 0;

    left: -100%;

    width: 100%;

    height: 2px;

    background: linear-gradient(90deg, transparent, #03e9f4);

    animation: btn-anim1 1s linear infinite;

  }

  @keyframes btn-anim1 {

    0% {

      left: -100%;

    }

    50%,100% {

      left: 100%;

    }

  }

  .login-box a span:nth-child(2) {

    top: -100%;

    right: 0;

    width: 2px;

    height: 100%;

    background: linear-gradient(180deg, transparent, #03e9f4);

    animation: btn-anim2 1s linear infinite;

    animation-delay: .25s

  }

  @keyframes btn-anim2 {

    0% {

      top: -100%;

    }

    50%,100% {

      top: 100%;

    }

  }

  .login-box a span:nth-child(3) {

    bottom: 0;

    right: -100%;

    width: 100%;

    height: 2px;

    background: linear-gradient(270deg, transparent, #03e9f4);

    animation: btn-anim3 1s linear infinite;

    animation-delay: .5s

  }

  @keyframes btn-anim3 {

    0% {

      right: -100%;

    }

    50%,100% {

      right: 100%;

    }

  }

  .login-box a span:nth-child(4) {

    bottom: -100%;

    left: 0;

    width: 2px;

    height: 100%;

    background: linear-gradient(360deg, transparent, #03e9f4);

    animation: btn-anim4 1s linear infinite;

    animation-delay: .75s

  }

  @keyframes btn-anim4 {

    0% {

      bottom: -100%;

    }

    50%,100% {

      bottom: 100%;

    }

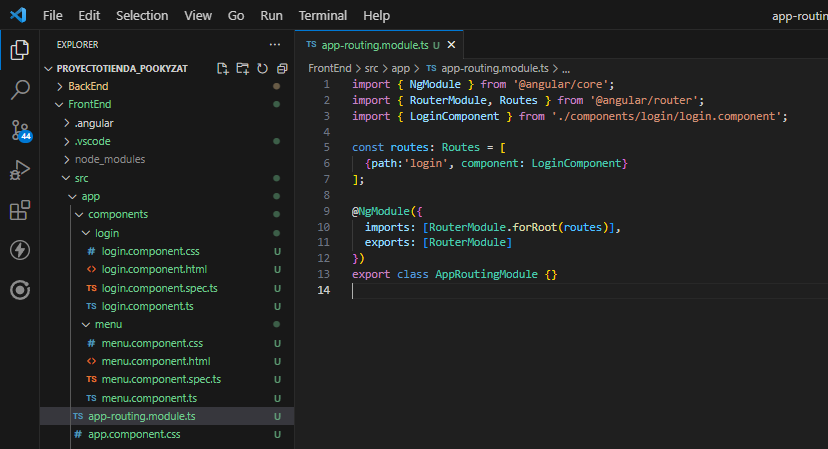
  }

# Routes

## Ruta login

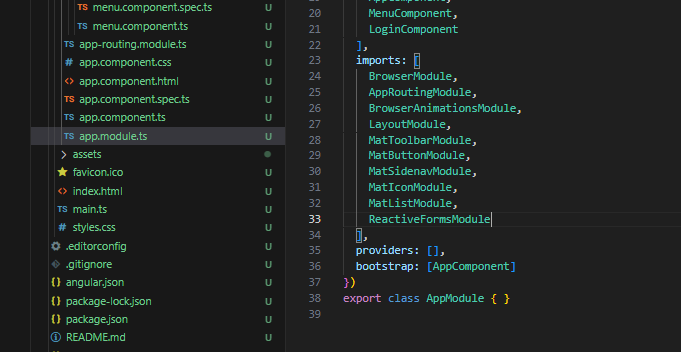
Nos ubicamos en app-routing.module y dentro del arreglo de routing vamos a crear un objeto asi:

{path:'login', component: LoginComponent}



# Form

En el archivo app.module.ts en la parte de los imports agregamos el ReactiveFormsModule



## Formulario Login

Configuramos validaciones y obtención de datos

### HTML

<div class="login-box">

    <h2>Bienvenido</h2>

    <form [formGroup]="loginForm" (ngSubmit)="onSubmit()">

      <!-- contenedor input -->

      <div class="user-box" [ngClass]="{'user-box-err':loginForm.get('email')?.touched && loginForm.get('email')?.invalid}">

        <input type="text" formControlName="email">

        <label >Username</label>

        <div \*ngIf="loginForm.get('email')?.touched && loginForm.get('email')?.invalid" class="form-err">

          <div \*ngIf="loginForm.get('email')?.hasError('required')" >Email es requerido.</div>

          <div \*ngIf="loginForm.get('email')?.hasError('email')">Requiere formato de email ex: ejemplo@pookyzat.com</div>

        </div>

      </div>

      <!-- contenedor input -->

      <div class="user-box" [ngClass]="{'user-box-err':loginForm.get('email')?.touched && loginForm.get('email')?.invalid}">

        <input type="password" id="password" formControlName="password">

        <label>Password</label>

        <div \*ngIf="loginForm.get('password')?.touched && loginForm.get('password')?.invalid" class="form-err">

          <div \*ngIf="loginForm.get('password')?.hasError('required')">Contraseña es requerida.</div>

        </div>

      </div>

      <button type="submit">

        <span></span>

        <span></span>

        <span></span>

        <span></span>

        login

      </button>

    </form>

</div>

### CSS

/\* :root{

    pricipal: rgba(3, 22, 67, 0.785);

    secundario: #73472c;

    blanco: #f4f4f4;

    rosa: #c9b2b2;

    otroblanco: #aaa39e;

    cor:#d6d5ca;

    colorin:#9a8768;

} \*/

html{

    height: 100%;

}

body {

    margin:0;

    padding:0;

    font-family: sans-serif;

}

/\* caja grande \*/

.login-box{

    position: absolute;

    top: 50%;

    left: 50%;

    left: 50%;

    width: 400px;

    padding: 40px;

    transform: translate(-50%, -50%);

    background: rgba(3, 22, 67, 0.785);

    box-sizing: border-box;

    box-shadow: 0 15px 25px rgba(0,0,0,.6);

    border-radius: 10px;

}

/\* titulo \*/

.login-box h2 {

    margin: 0 0 30px;

    padding: 0;

    color: #f4f4f4;

    text-align: center;

}

.login-box .user-box {

    position: relative;

}

/\* Inputs \*/

.login-box .user-box input {

    width: 100%;

    padding: 10px 0;

    font-size: 16px;

    color: #f4f4f4;

    margin-bottom: 30px;

    border: none;

    border-bottom: 1px solid #f4f4f4;

    outline: none;

    background: transparent;

}

/\* if error \*/

.login-box .user-box-err input {

    width: 100%;

    padding: 10px 0;

    font-size: 16px;

    color: #f4f4f4;

    margin-bottom: 10px;

    border: none;

    border-bottom: 1px solid #f4f4f4;

    outline: none;

    background: transparent;

}

/\* labels \*/

.login-box .user-box label {

    position: absolute;

    top:0;

    left: 0;

    padding: 10px 0;

    font-size: 16px;

    color: #f4f4f4;

    pointer-events: none;

    transition: .5s;

}

/\* animacion del label \*/

/\* esta animacion no aplica porque el formGroup me da un value lo cual lo hace valid\*/

.login-box .user-box input:focus ~ label,

.login-box .user-box input:valid ~ label {

  top: -20px;

  left: 0;

  color: #03e9f4;

  font-size: 12px;

}

/\*boton\*/

.login-box form button {

    position: relative;

    display: inline-block;

    padding: 10px 20px;

    color: #f4f4f4;

    font-size: 16px;

    text-decoration: none;

    text-transform: uppercase;

    overflow: hidden;

    transition: .5s;

    margin-top: 40px;

    letter-spacing: 4px;

    border: none;

    background: linear-gradient(90deg, transparent, #000202, transparent);

}

.login-box form button:hover {

    background: #03e9f4;

    color: #f4f4f4;

    border-radius: 5px;

    box-shadow: 0 0 5px #03e9f4,

                0 0 25px #03e9f4,

                0 0 50px #03e9f4,

                0 0 100px #03e9f4;

}

/\* bordes del boton \*/

.login-box button span {

    position: absolute;

    display: block;

}

/\* animacion top\*/

.login-box button span:nth-child(1) {

    top: 0;

    width: 100%;

    height: 2px;

    background: linear-gradient(90deg, transparent, #03e9f4);

    animation: btn-anim1 1s linear infinite;

}

@keyframes btn-anim1 {

    0% {

    left: -100%;

    }

    50%,100% {

    left: 100%;

    }

}

/\* animacion right\*/

.login-box button span:nth-child(2) {

    top: -100%;

    right: 0;

    width: 2px;

    height: 100%;

    background: linear-gradient(180deg, transparent, #03e9f4);

    animation: btn-anim2 1s linear infinite;

    animation-delay: .25s

}

@keyframes btn-anim2 {

    0% {

      top: -100%;

    }

    50%,100% {

      top: 100%;

    }

}

/\* animacion button\*/

.login-box button span:nth-child(3) {

    bottom: 0;

    right: -100%;

    width: 100%;

    height: 2px;

    background: linear-gradient(270deg, transparent, #03e9f4);

    animation: btn-anim3 1s linear infinite;

    animation-delay: .5s

}

@keyframes btn-anim3 {

    0% {

        right: -100%;

    }

    50%,100% {

        right: 100%;

    }

}

/\* animacion left\*/

.login-box button span:nth-child(4) {

    bottom: -100%;

    left: 0;

    width: 2px;

    height: 100%;

    background: linear-gradient(360deg, transparent, #03e9f4);

    animation: btn-anim4 1s linear infinite;

    animation-delay: .75s

}

@keyframes btn-anim4 {

    0% {

        bottom: -100%;

    }

    50%,100% {

        bottom: 100%;

    }

}

/\* Clase error \*/

.form-err{

    margin-top: 0;

    margin-bottom: 20px;

    font-size: 12px;

    color: #e5b7b7;

    text-align: center;

}

### TS

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

import {FormControl, FormGroup, Validators} from '@angular/forms';

@Component({

  selector: 'app-login',

  templateUrl: './login.component.html',

  styleUrls: ['./login.component.css']

})

export class LoginComponent implements OnInit{

  loginForm:FormGroup = new FormGroup({}) //inicializacion del formgroup

  constructor(){

  }

  ngOnInit(): void {

      this.loginForm = new FormGroup({

        email: new FormControl('',[Validators.required, Validators.email]),

        password: new FormControl('',Validators.required)

      })

  }

  onSubmit(){

        if(this.loginForm.valid){

          console.log(this.loginForm.value)

        }

  }

}

# Modules

“Ng g class clientes” en la consola dentro de la carpeta modules y cambiamos el class por interface

## Clientes

### Ts