Clase 5 :

Creamos un nuevo proyecto de cero pero le ponemos "ruteo"cuando nos pregunte al momento de instalar le decimos que SI:

" would you like to add Angular routing?"

El ruteo nos va a permitir ir navegando a través de mi aplicacion .

Si vamos a "app.module.ts" hay importado un "AppRoutingModule"

El app-routing.module.ts se encarga de donde vamos a hacer las rutas.

\* los metodos "forRoot" se encarga de pasarle knformacion al "route"

Dentro de la “const routes” :

Una ruta va a tener el "path" que va a ser lo que va a estar en la url y el "componente" que va a ser uno que yo quiera cargar

Estos se van a en donde este el componente "router-outlet" que lo podemos poner en cualquier lado , podemos crear varios , podemos anidarlos

\*Generamos module:

ng g module pages/pages

dentro de "page.module.ts" voy a incluir los módulos aca en vez de en la ruta principal

se agregan solos :

import { HomeComponent } from './home/home.component';

import { ClientesComponent } from './clientes/clientes.component';

import { ErrorComponent } from './error/error.component';

import { DetallesComponent } from './detalles/detalles.component';

Después este module lo agregamos en la ruta principal “app.module.ts” y

también en imports :

import { PagesModule } from './pages/pages.module';

Generamos componente en page:

ng g component pages/home

Dentro de "page.module.ts" importo "RouterModule" y tambien lo agrego dentro de "imports" :

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

Lo que vemos es que me modifico el "pages.module.ts" y me importa el "home"

Importo dentro de app.module.ts el "PagesModule" y lo agrego a "imports"

Importo dentro de app-routing.module el "componente" "HomeComponent" y ahí dentro dentro del const routes agregamos ese componente :

import { HomeComponent } from './pages/home/home.component';

const routes: Routes = [

{

path: '',

redirectTo : 'home',

pathMatch: 'full'

},

Creamos otro componente:

ng g component pages/clientes

vamos al “app-routing.module” y creamos la ruta clientes y lo importamos ahí mismo "ClientesComponent"

\*por conveniencia crear una carpeta "core" que dentro tengan "servicios,clases,pipes,comp"

path "\*\*" los asteriscos indica que si no se cargo alguna ruta se carga esa \*creamos componente de error\*

path"/" cuando no hay nada que redirija a home

Creamos 3 botones en los cuales apretemos el botón y naveguemos hacia otro lado con "routerLink":

Si entramos a clientes vamos a poder enviar un id a detalles "path: 'detalle/:id'" es decir el cliente que elegimos nos va a mostrar sus detalles en otra página.

Dentro de "index.html" hay una etiqueta base y hay una "/" puede llegar a causar algún error

<base href= "/">

Voy a tener "rutas anidadas" :

“ClientesComponent” va a tener su propio< routerOutlet> con la propiedad "children" que recibe un array y ahí dentro voy a cargar mis rutas

Desde app-routing.module.ts lo que veriamos en el path de clientes sera a su “childre” tambien :

{

path: 'clientes',

component: ClientesComponent,

children: [

{

path: 'detalles/:id',

component: DetallesComponent,

}

]

},

Para obtener el valor que pasamos desde “clientes” a su “children” en este caso a “detalles” , le pasaremos un “:id” , entonces desde “detalles.module.ts” hacemos lo siguiente :

// obtengo el valor que paso desde clientes a su "children"

this.valorChildren = this.route.snapshot.paramMap.get('id');

console.log(this.route.snapshot.paramMap.get('id'));

FIREBASE:

Servicio de authentication podemos crear un usuario

Storage es el almacenamiento de archivos

El hosting nos permite hostear nuestra página

El database hay dos tipos , utilizar "cloud Firestore"

1-Creamos una colección que seria como una tabla

, dentro un documento "users"

y dentro de users le agrego un campo y también una colección en donde vamos a meter todo

\*una colección no puede tener un hijo directo que sea un colección , a su vez un objeto no puede tener un hijo directo de objeto

Si tengo muchas subcolecciones no puede realizar una función directa

También tiene reglas esta base de datos para indicar quién puede leer o escribir

También tiene índices para cada búsqueda que queramos hacer

Angular Fire

valueChanges() se ejecuta cada vez que hago un cambio en la base de datos , se va a refrescar en mi aplicación (ideal para chats)

doc() le paso el documento para acceder a ese

Crear servicio que maneje peticiones a firebase ,

1-Seteamos los datos de firebase en config que lo pondremos dentro de "environment.ts"

2-AngularFireModule.initializeApp(environment.firebase)