

Ir al sitio oficial de ionic:
Hacer registro o iniciar sesión

Asignarle nombre y demás características al proyecto
Instalar IONIC:
`npm install -g @ionic/cli`

Se requiere previamente de NodeJS

Escribimos el commando:
`ionic start`

Nos sale la siguiente opción :
Use the app creation wizard? Yes/No
Yes para que creamos el proyecto mediante un ayudante web de Ionic o no para crear el proyecto por consola

Seleccionamos el framework

Ya creado el proyecto lo lanzamos con `ionic serve`

Podemos crear paginas nuevas con `ionic generate page`
Le asignamos nombre y ruta según preferencias

clonar el repositorio creado por ejemplo `git clone`
`https://github.com/DanielFelipeAgMo/app-crud-ionic.git` `app-crud-ionic`
`cd app-crud-ionic && npm install && ionic serve`

`ionic start ionic-crud-tdea blank`

Seleccionamos el framework de nuestro gusto -> Angular

Esperar a que procese y cree todo lo necesario luego ubicarnos en la ruta de nuestro proyecto: `cd ionic-crud-tdea`

Con el comando "code ." accedemos al editor de visual studio.

Lanzamos el proyecto con `ionic serve -lab`

```

PS G:\Docente\Desarrollo Movil TdeA\Proyectos desarrollo movil\ionic-crud-tdea> ionic serve -lab
[WARN] The --lab option has been deprecated and will be removed in an upcoming major release of the Ionic CLI. Please
visit https://ionicframework.com/docs/developing/previewing for alternatives to Ionic Lab.
> ng.cmd run app:serve --host=localhost --port=8100
[ng] One or more browsers which are configured in the project's Browserslist configuration will be ignored as ES5 output is not supported b
y the Angular CLI.
[ng] Ignored browsers: chrome 60
[ng] - Generating browser application bundles (phase: setup)...
[ng] TypeScript compiler options "target" and "useDefineForClassFields" are set to "ES2022" and "false" respectively by the Angular CLI
. To control ECMA version and features use the Browserslist configuration. For more information, see https://angular.io/guide/build#configur
ing-browser-compatibility
[ng] NOTE: You can set the "target" to "ES2022" in the project's tsconfig to remove this warning.
[ng] ✓ Browser application bundle generation complete.
[ng] Initial Chunk Files
Raw Size | Names |
[ng] vendor.js | vendor |
2.93 MB |
[ng] polyfills.js | polyfills |
316.97 kB |
[ng] styles.css, styles.js | styles |
256.08 kB |
[ng] main.js | main |
14.78 kB |
[ng] runtime.js | runtime |

```

luego instalamos con yes y finalmente esperamos a que nos genere las rutas para probarlo

```

found 0 vulnerabilities
> ionic-lab.cmd http://localhost:8100 --host localhost --port 8200 --project-type angular --app-name ionic-crud-tdea --app-version 0.0.1

[INFO] Development server running!

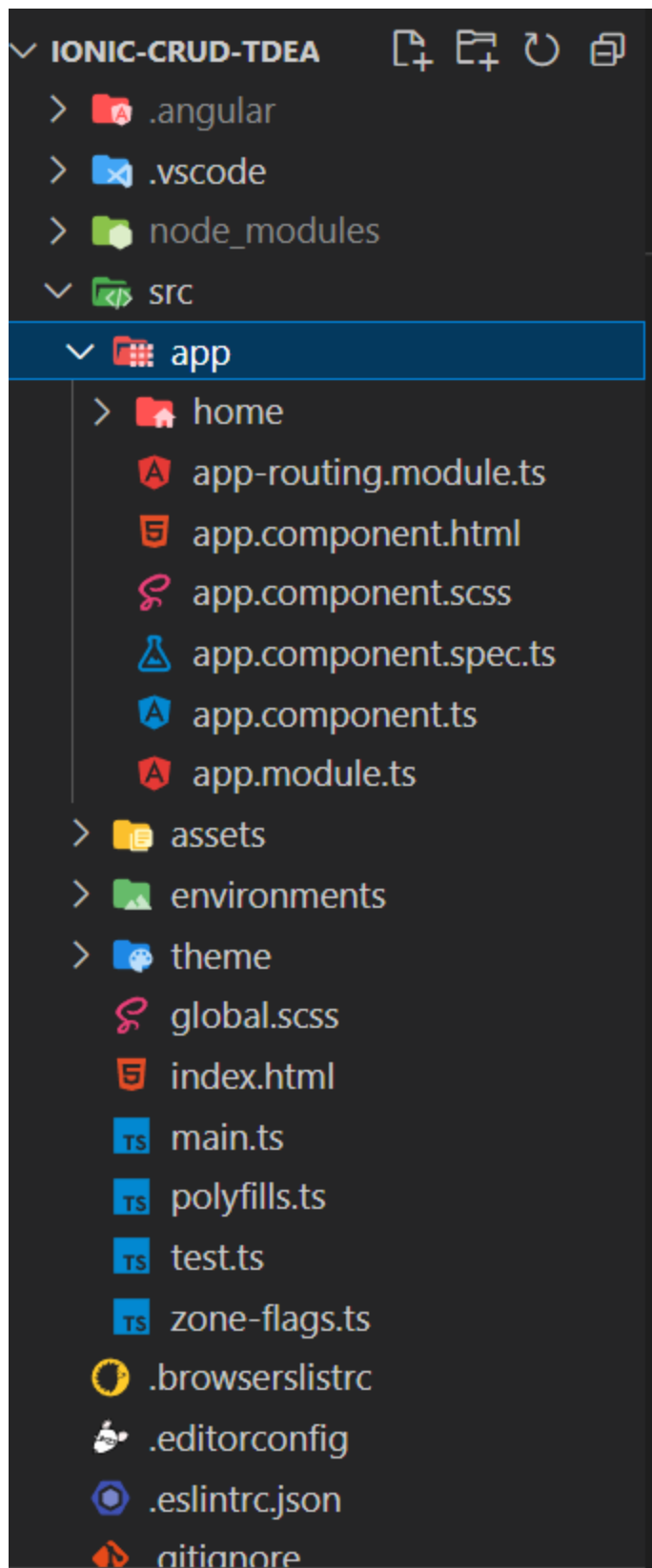
Lab: http://localhost:8200
Local: http://localhost:8100

Use Ctrl+C to quit this process

[ng] ✓ Browser application bundle generation complete.
[ng] Initial Chunk Files | Names | Raw Size
[ng] main.js | main | 14.78 kB |
[ng] runtime.js | runtime | 14.10 kB |
[ng]
[ng] Lazy Chunk Files | Names | Raw Size
[ng] src_app_home_home_module_ts.js | home-home-module | 9.13 kB |
[ng]
[ng] 59 unchanged chunks
[ng]
[ng] Build at: 2023-02-25T16:05:42.964Z - Hash: 7aeff719d46ae083 - Time: 2613ms
[ng] ✓ Compiled successfully.

```

Observe la estructura del proyecto, en la carpeta src/app



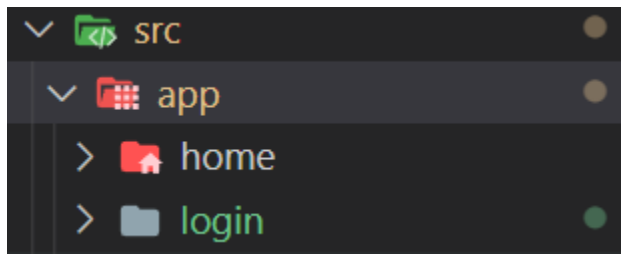
Allí crearemos una nueva pagina con el comando “ionic generate page nombre-pagina”

```

PS G:\Docente\Desarrollo Movil TdeA\Proyectos desarrollo movil\ionic-crud-tdea> ionic generate page login
> ng.cmd generate page login --project=app
? Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.io/analytics. No
Global setting: not set
Local setting: disabled
Effective status: disabled
CREATE src/app/login/login-routing.module.ts (343 bytes)
CREATE src/app/login/login.module.ts (465 bytes)
CREATE src/app/login/login.page.html (124 bytes)
CREATE src/app/login/login.page.spec.ts (654 bytes)
CREATE src/app/login/login.page.ts (252 bytes)
CREATE src/app/login/login.page.scss (0 bytes)
UPDATE src/app/app-routing.module.ts (607 bytes)
[OK] Generated page!
PS G:\Docente\Desarrollo Movil TdeA\Proyectos desarrollo movil\ionic-crud-tdea>

```

Ahora compruebe de nuevo la ruta app



Repetimos el proceso con las páginas que deseamos anexar...

```

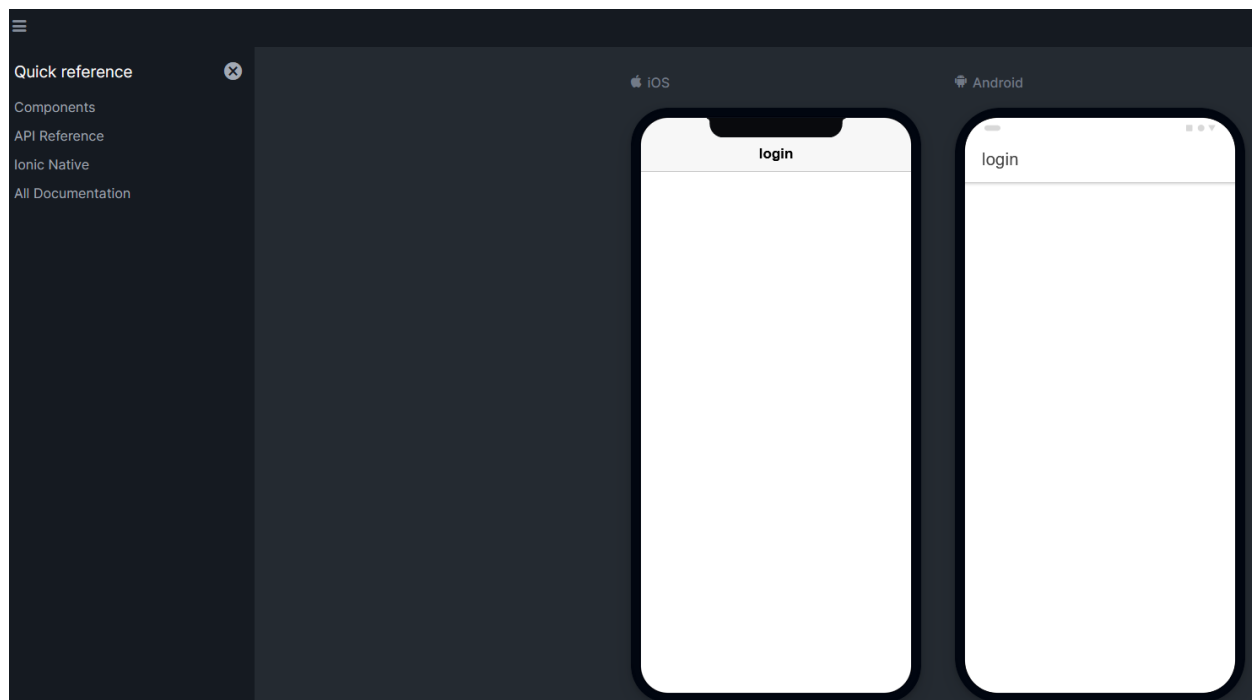
PS G:\Docente\Desarrollo Movil TdeA\Proyectos desarrollo movil\ionic-crud-tdea> ionic generate page register
> ng.cmd generate page register --project=app
CREATE src/app/register/register-routing.module.ts (355 bytes)
CREATE src/app/register/register.module.ts (486 bytes)
CREATE src/app/register/register.page.html (127 bytes)
CREATE src/app/register/register.page.spec.ts (675 bytes)
CREATE src/app/register/register.page.ts (264 bytes)
CREATE src/app/register/register.page.scss (0 bytes)
UPDATE src/app/app-routing.module.ts (732 bytes)
[OK] Generated page!

```

Ahora diríjase a la ruta app/app-routing.modules.ts y enrute a login como la vista inicial

```
app-routing.module.ts M X
src > app > app-routing.module.ts > routes > path
1 import { NgModule } from '@angular/core';
2 import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [
5   {
6     path: 'home',
7     loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)
8   },
9   {
10    path: '',
11    redirectTo: 'login',
12    pathMatch: 'full'
13  },
14 ];
```

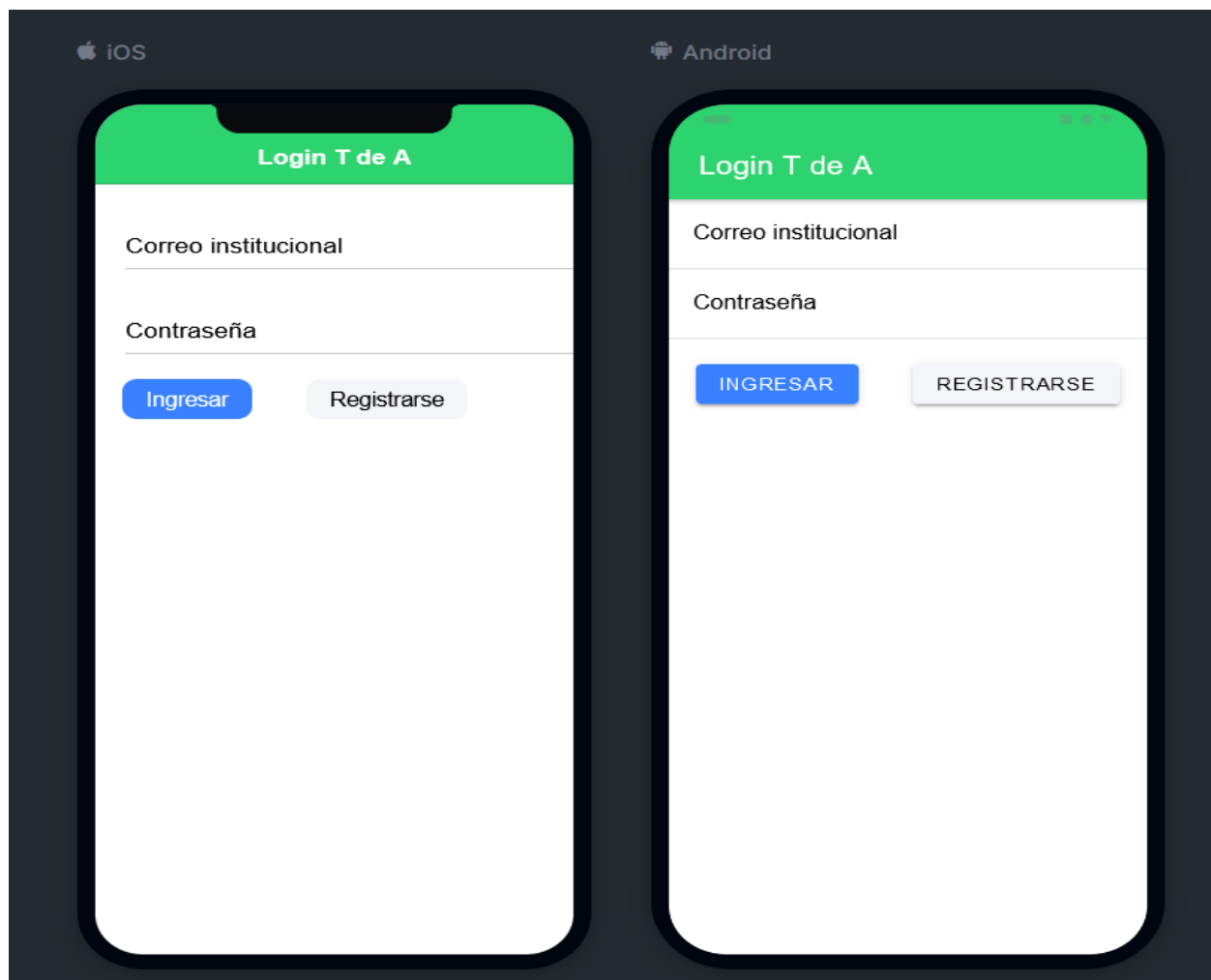
Compruebe en el servidor el cambio



Nos dirigimos a login.page.html y maquetamos el login

```
app-routing.module.ts M login.page.html U X
src > app > login > login.page.html > ...
6
7 <ion-content>
8   <ion-item>
9     <ion-label position="floating">Correo institucional</ion-label>
10    <ion-input type="email" required</ion-input>
11  </ion-item>
12
13  <ion-item>
14    <ion-label position="floating">Contraseña</ion-label>
15    <ion-input type="password" required</ion-input>
16  </ion-item>
17
18  <ion-button class="ion-padding" color="primary">Ingresar</ion-button>
19
20  <ion-button class="ion-padding" color="light" [routerLink]="['/register']">Registrarse</ion-button>
21 </ion-content>
22
```

Comprobamos los cambios en el servidor



Realice la misma acción en la vista register

```

<ion-header>
  <ion-toolbar color="success">
    <ion-title>Registrese aquí</ion-title>
  </ion-toolbar>
</ion-header>

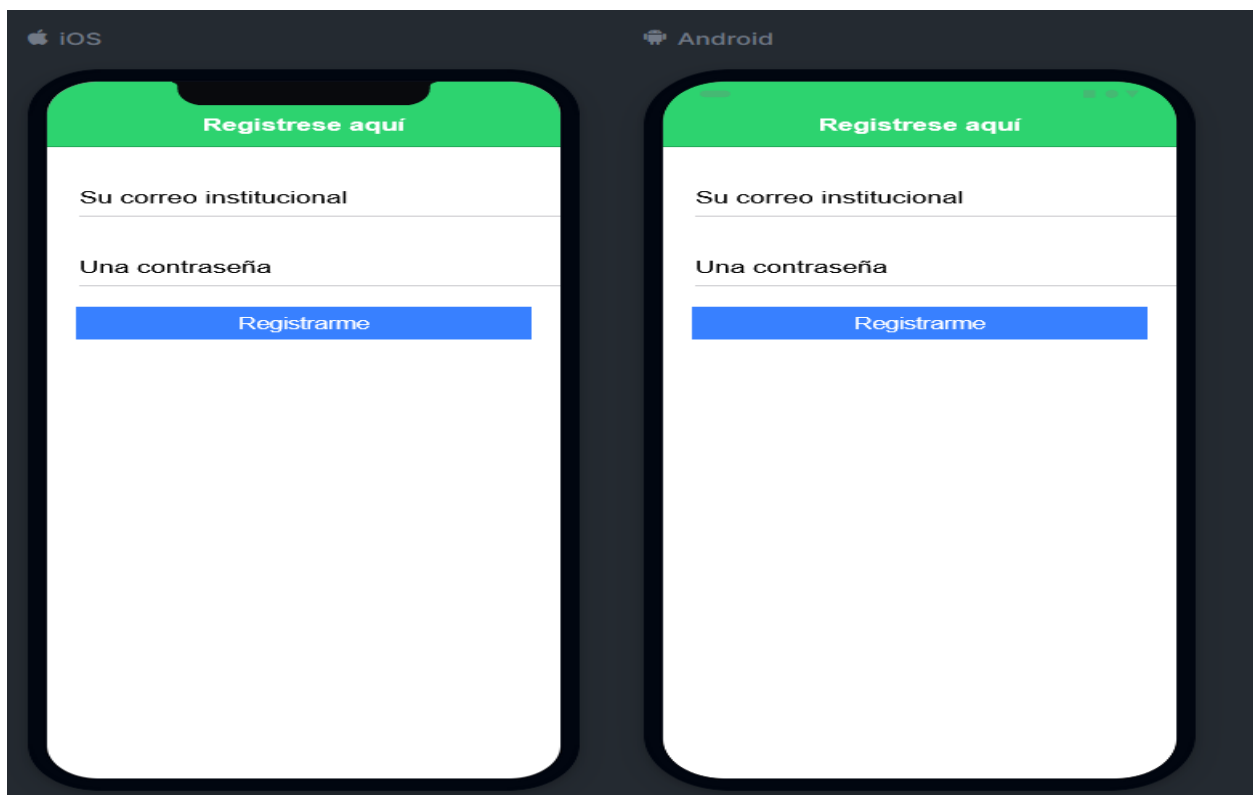
<ion-content>
  <ion-item>
    <ion-label position="floating">Su correo institucional</ion-label>
    <ion-input type="email" required></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Una contraseña</ion-label>
    <ion-input type="password" required></ion-input>
  </ion-item>

  <ion-button class="ion-padding" color="primary" expand="full">Registrarme</ion-button>
</ion-content>

```

Y siga comprobando los cambios



Lo siguiente es personalizar el home al que se accederá una vez logueado en nuestra app tdea

```

<ion-header [translucent]="true">
  <ion-toolbar color="success">
    <ion-title>
      App T de A
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-list lines="none">
    <ion-item-sliding>
      <ion-item class="no-padding">
        <ion-label>
          <h2>Titulo</h2>
          <p>Detalles</p>
        </ion-label>
      </ion-item>

      <ion-item-options side="start">
        <ion-item-option>
          <ion-icon slot="icon-only" name="pencil-outline"></ion-icon>
        </ion-item-option>
      </ion-item-options>

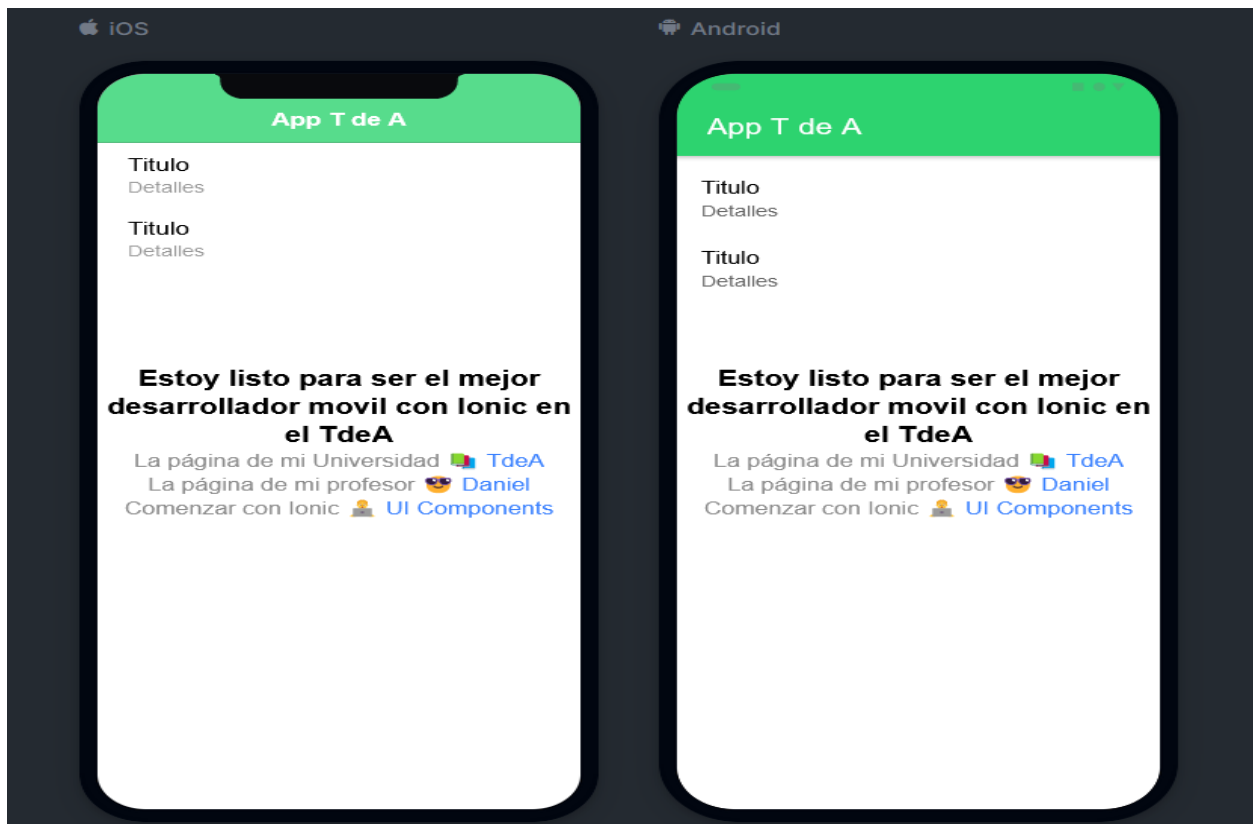
      <ion-item-options side="end">
        <ion-item-option color="danger">
          <ion-icon slot="icon-only" name="trash"></ion-icon>
        </ion-item-option>
      </ion-item-options>
    </ion-item-sliding>

    <ion-item-sliding>...
  </ion-item-sliding>
</ion-list>

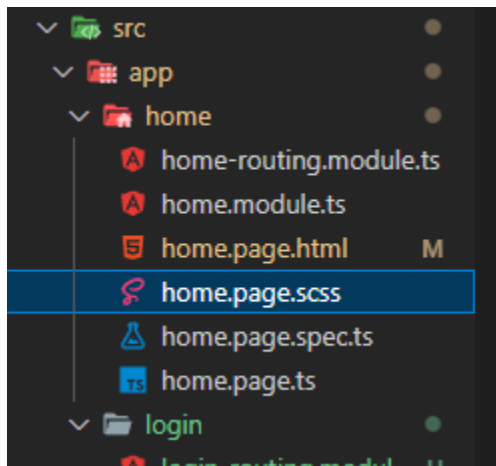
<div id="container">
  <strong>Estoy listo para ser el mejor desarrollador movil con Ionic en el TdeA </strong>
  <p>La página de mi Universidad <a target="_blank" rel="noopener noreferrer" href="https://www.tdea.edu.co/">TdeA</a></p>
  <p>La página de mi profesor <a target="_blank" rel="noopener noreferrer" href="https://dfamsoluciones.com/">Daniel</a></p>
  <p>Comenzar con Ionic <a target="_blank" rel="noopener noreferrer" href="https://ionicframework.com/docs/components">UI Components</a></p>
</div>

```

Compruebe cada cambio en el servidor



Ahora modificaremos los estilos que trae nuestra app ionic y le daremos nuestro toque.



```
ion-item-sliding{  
  border-bottom: 1px solid rgba(0, 0, 0, 0.13);  
}
```

Ya esto es a gusto de cada quien

Lo siguiente es instalar firebase como gbd por medio de npm, así: “npm install firebase @angular/fire --save”

Diríjase a su navegador y busque Firebase console y cree un nuevo proyecto



Aprovisionando recursos...
ionic-crud-tdea

Firestore

Descripción general ...

Categorías de producto

Compilación

Lanzamiento y supervisión

Analytics

Participación

Todos los productos

Personaliza tu navegación

Ahora puedes personalizar la navegación para enfocar tu experiencia en la consola

Más información

Entendido

Spark

Sin costo \$0 por mes

Actualizar

ionic-crud-tdea

Recibe notificaciones por correo electrónico sobre nuevas funciones de Firebase, investigaciones y eventos

Regístrate

ionic-crud-tdea

Plan Spark

Comienza por agregar Firebase a tu app

iOS

</>

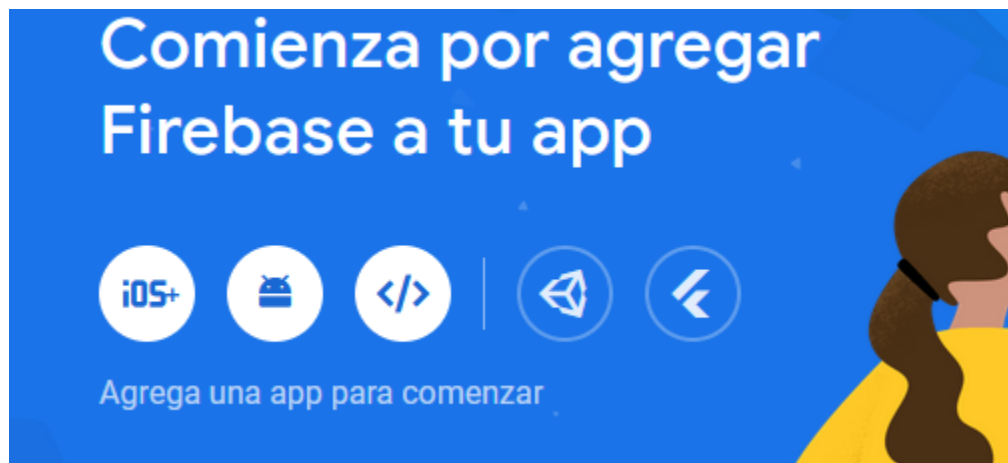
Agrega una app para comenzar

Almacena y sincroniza datos de app en milisegundos

Authentication

Cloud Firestore

Seleccione la opción deseada en este caso web



× Agrega Firebase a tu aplicación web

1 Registrar app

Sobrenombre de la app ?

ionic-crud-tdea

☐

Además, configura **Firebase Hosting** para esta app.

[Más información](#) 

Hosting también se puede configurar más adelante. Puedes comenzar cuando quieras sin costo alguno.

Registrar app

2 Agrega el SDK de Firebase

Se genera a siguiente vista, por lo tanto los valores como dominio, key, id etc varían a cada persona.

× Agrega Firebase a tu aplicación web

✓ Registrar app

2 Agrega el SDK de Firebase

☒ Usar npm ☐ Usar una etiqueta <script>

Si ya usas [npm](#) y un agrupador de módulos como [Webpack](#) o [Rollup](#), puedes ejecutar el siguiente comando para instalar la versión más reciente del SDK ([más información](#)):

```
$ npm install firebase
```



Luego, inicializa Firebase y comienza a usar los SDK de los productos que quieres utilizar.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyAjZYnpCvVgTWBPo1izIyfufPo2eWpcwDA",
  authDomain: "ionic-crud-tdea.firebaseio.com",
  projectId: "ionic-crud-tdea",
  storageBucket: "ionic-crud-tdea.appspot.com",
  messagingSenderId: "197905154648",
  appId: "1:197905154648:web:fcedb3843a2dd52d96002d",
  measurementId: "G-0FY5XHVQ18"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

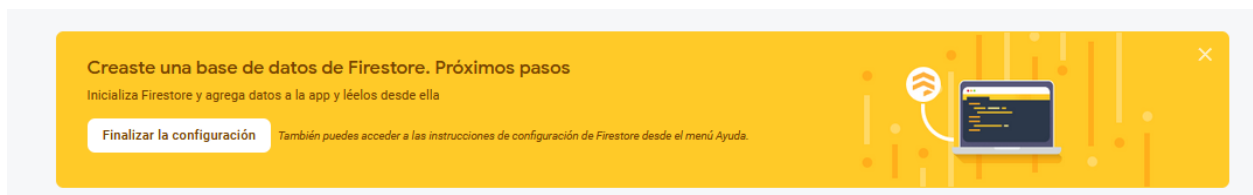
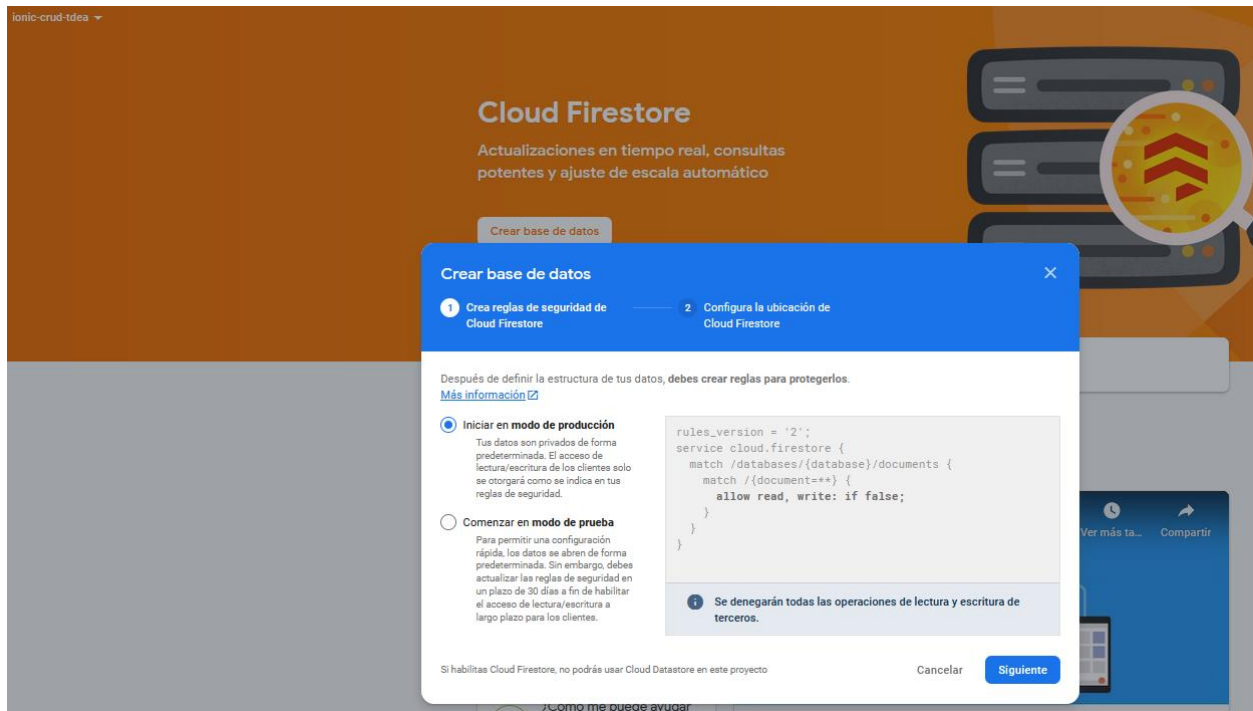


Nota: Esta opción utiliza el [SDK de JavaScript modular](#), que proporciona un tamaño reducido del SDK.

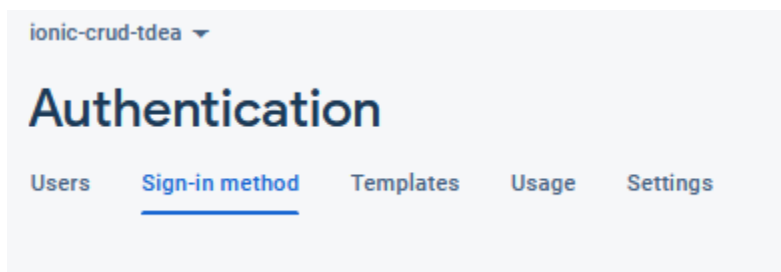
Obtén más información sobre Firebase para la Web: [primeros pasos](#), [referencia de la API del SDK web](#) y [muestras](#)

[Ir a la consola](#)

Copiamos los datos necesarios y luego haciendo clic en ir a la consola nos vamos a la opción Firestore Database y creamos nuestra base de datos seleccionando el modo prueba




Luego vaya a autenticación y seleccione Sign-in method



Ahora habilite la opción por correo

 Correo electrónico/contraseña

☒ Habilitar

Permite que los usuarios se registren con su dirección de correo electrónico y contraseña. Nuestros SDK también proporcionan verificación de la dirección de correo electrónico, recuperación de contraseñas y primitivas de cambio de dirección de correo. [Más información](#) 

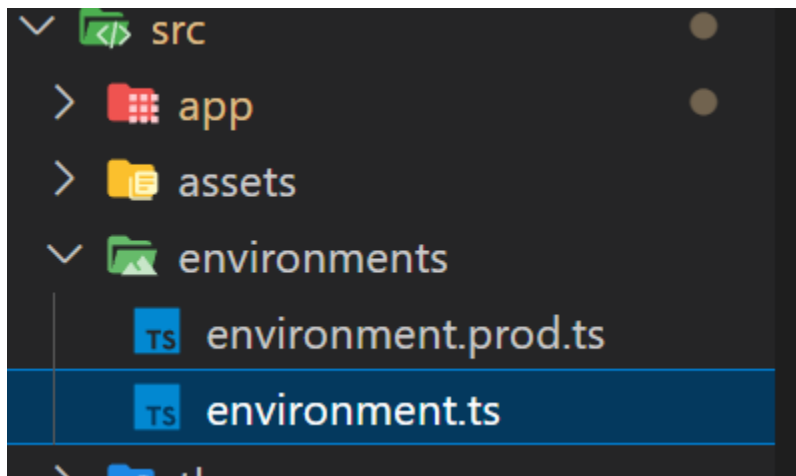
Vínculo del correo electrónico (acceso sin contraseña)

☐ Habilitar

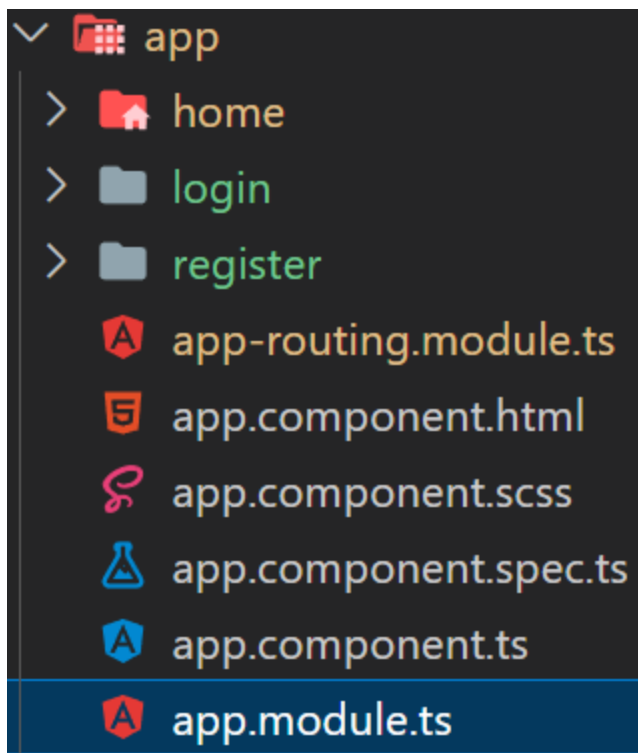
Cancelar

Guardar

Luego diríjase a la ruta `environments.ts` del proyecto ionic



Allí cree una constante llamada `FIREBASE_CONFIG` o como usted desee nombrarlo y pasele los valores de la conexión generados por firebase, luego en la ruta :



Procure comparar su script con este

```
import { AngularFireAuthModule } from "@angular/fire/compat/auth";
import { AngularFireModule } from '@angular/fire/compat/';
import { environment } from 'src/environments/environment';

@NgModule({
  declarations: [AppComponent],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    AppRoutingModule,
    AngularFireModule.initializeApp(environment.FIREBASE_CONFIG),
    AngularFireAuthModule,
  ],
  providers: [{ provide: RouteReuseStrategy, useClass: IonicRouteStrategy }],
  bootstrap: [AppComponent],
})
```

Ahora cree una carpeta dentro de app llamada modelos o models como prefiera y dentro de este un nuevo archivo llamado “user.mode.ts” enfocado en la lógica de usuario. Con el siguiente script:

```
export interface User {
  email: string;
  password: string;
}
```

Ahora en el archivo register.page.ts agregar lo siguiente

```
import {User} from '../models/user.model';
```

```
y esto user = {} as User;
```

dentro de la class RegisterPage

Posteriormente diríjase a la vista html de register y actualice el formulario con ngModel aplicando la regla a los dos input como por ejemplo: [(ngModel)]="user.email"

Y al botón (click)="register(user)"

Ahora actualice el register.ts así:

```
import { Component, OnInit } from '@angular/core';
import { User } from '../models/user.model';
import { LoadingController, ToastController, NavController } from
'@ionic/angular';
import { AngularFireAuth } from "@angular/fire/compat/auth";

@Component({
  selector: 'app-register',
  templateUrl: './register.page.html',
  styleUrls: ['./register.page.scss'],
})
export class RegisterPage implements OnInit {
  user = {} as User;

  constructor(
    private toastCtrl: ToastController,
    private loadingCtrl: LoadingController,
    private afAuth: AngularFireAuth,
    private navCtrl: NavController
  ) {}

  ngOnInit() {}

  async register(user: User) {
    if (this.formValidation()) {
      let loader = await this.loadingCtrl.create({
        message: "Espere un momento por favor..."
      });
      await loader.present();

      try {
```



```

        await this.afAuth.createUserWithEmailAndPassword(user.email,
user.password).then(data =>{
            console.log(data);

            this.navCtrl.navigateRoot("home");
        })

    } catch (e) {

    }

    await loader.dismiss();
}

formValidation() {
    if (!this.user.email) {
        this.showToast("Ingrese un correo");
        return false;
    }

    if (!this.user.password) {
        this.showToast("Ingrese una contraseña");
        return false;
    }

    return true;
}

showToast(message: string) {
    this.toastCtrl.create({
        message: message,
        duration: 5000 // Aumenta la duración del mensaje a 5 segundos
    }).then(toastData => toastData.present());
}
}

```

Y el register.html así:

```

<ion-header>
  <ion-toolbar color="success">
    <ion-title>Regístrese aquí</ion-title>
  </ion-toolbar>

```

```
</ion-header>

<ion-content>
  <ion-item>
    <ion-label position="floating">Su correo institucional</ion-label>
    <ion-input type="email" [(ngModel)]="user.email" required></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Una contraseña</ion-label>
    <ion-input type="password" [(ngModel)]="user.password" required></ion-input>
  </ion-item>

  <ion-button class="ion-padding" color="primary" expand="full"
(click)="register(user)">Registrarme</ion-button>
</ion-content>
```

Ya en este punto debe ser posible realizar un registro en el app

🍏 iOS

🤖 Android

Regístrese aquí

Su correo institucional

daniel123@correo.tdea.edu.co

Una contraseña

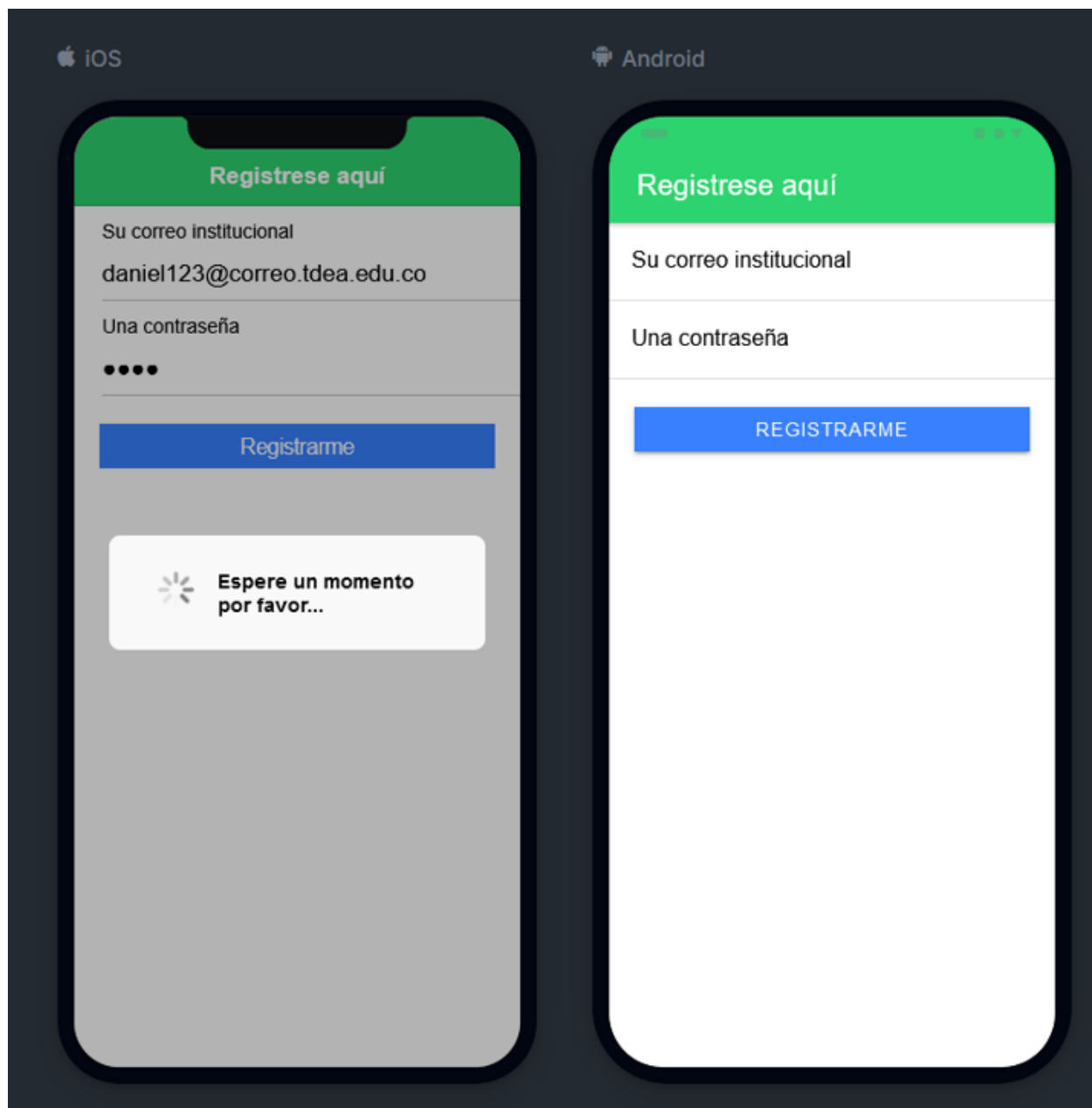
Regístrame

Regístrese aquí

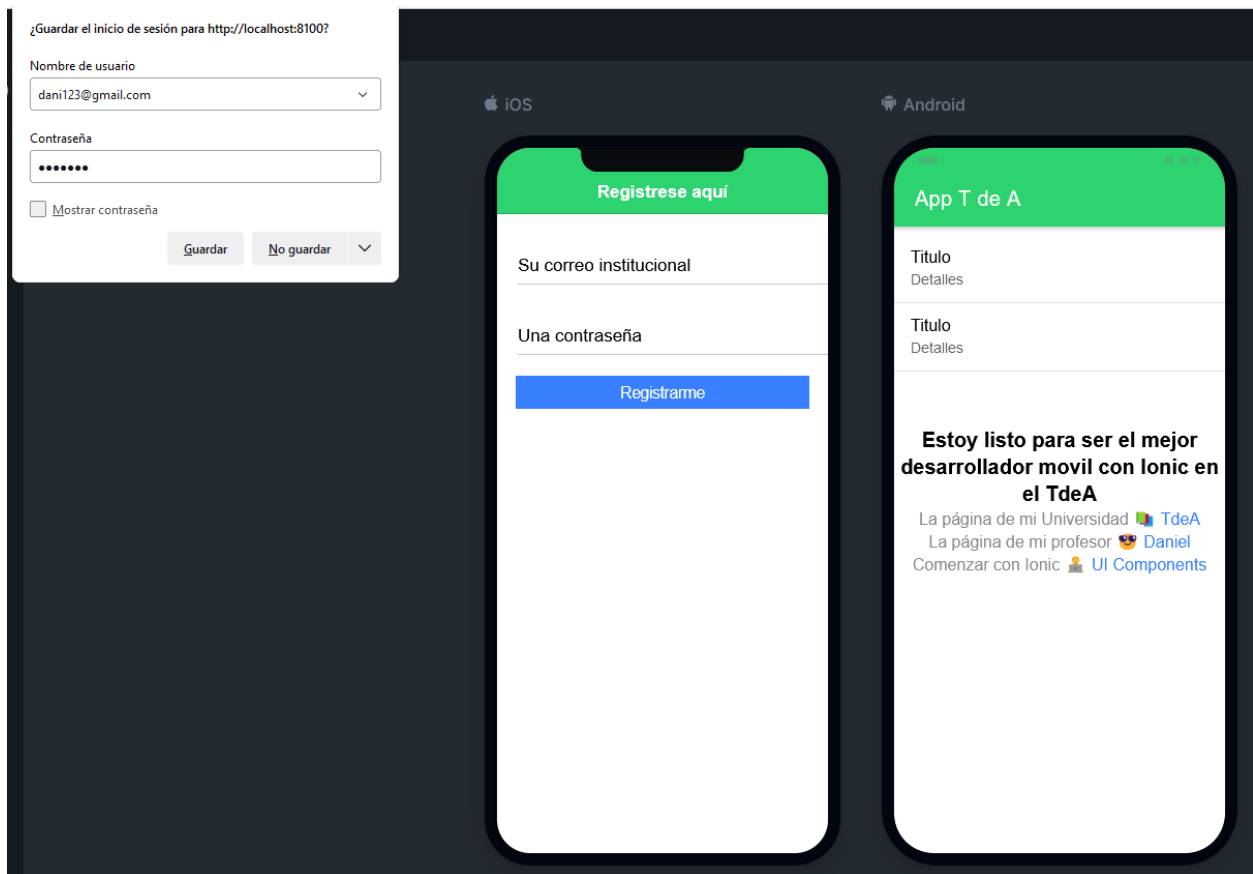
Su correo institucional

Una contraseña

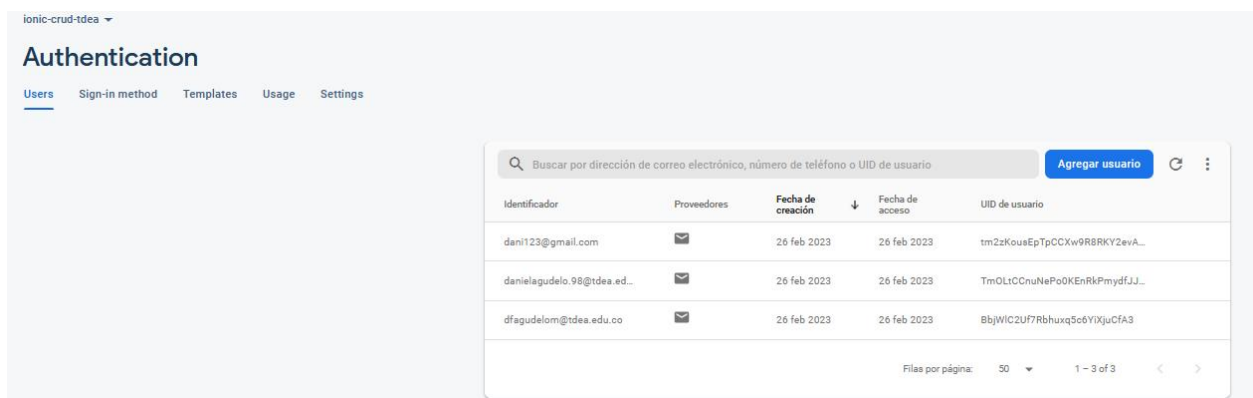
REGISTRARME



Note que la petición de registro se realiza correctamente



Compruebe ahora que su colección firebase almacene los nuevos usuarios



Es momento de configurar el login tanto en el html como en el typescript así:

```
<ion-header>
  <ion-toolbar color="success">
    <ion-title>Login T de A 🤖</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
```

```

<ion-item>
  <ion-label position="floating">Correo institucional</ion-label>
  <ion-input type="email" required [(ngModel)]="user.email" ></ion-input>
</ion-item>

<ion-item>
  <ion-label position="floating">Contraseña</ion-label>
  <ion-input type="password" required [(ngModel)]="user.password" ></ion-input>
</ion-item>

<ion-button class="ion-padding" color="primary"
(click)="login(user)">Ingresar</ion-button>

<ion-button class="ion-padding" color="light"
[routerLink]="['/register']">Registrarse</ion-button>
</ion-content>

```

Y el código ts:

```

import { Component, OnInit } from '@angular/core';
import { LoadingController, ToastController } from '@ionic/angular';
import { User } from '../models/user.model';
import { AngularFireAuth } from '@angular/fire/compat/auth';
import { NavController } from '@ionic/angular';

@Component({
  selector: 'app-login',
  templateUrl: './login.page.html',
  styleUrls: ['./login.page.scss'],
})

export class LoginPage implements OnInit {

  user = {} as User;

  constructor(
    private toastCtrl: ToastController,
    private loadingCtrl: LoadingController,
    private afAuth: AngularFireAuth,
    private navCtrl: NavController
  ) { }

```

```
ngOnInit() {}

async login(user: User) {

    if (this.formValidation()) {
        let loader = await this.loadingCtrl.create({
            message: "Espere un momento por favor..."
        });
        await loader.present();

        try {
            await this.afAuth.signInWithEmailAndPassword(user.email,
user.password).then(data =>{
                console.log(data);

                this.navCtrl.navigateRoot("home");
            })

        } catch (e:any) {

            let errorMessage = e.message || e.getLocalizedMessage();

            this.showToast(errorMessage);
        }

        await loader.dismiss();
    }
}

formValidation() {
    if (!this.user.email) {
        this.showToast("Ingrese un correo");
        return false;
    }

    if (!this.user.password) {
        this.showToast("Ingrese una contraseña");
        return false;
    }

    return true;
}

showToast(message: string) {
```

```

    this.toastCtrl.create({
      message: message,
      duration: 5000 // Aumenta la duración del mensaje a 5 segundos
    }).then(toastData => toastData.present());
  }
}

```

```

import { Component, OnInit } from '@angular/core';
import { User } from '../models/user.model';
import { ToastController } from '@ionic/angular';

@Component({
  selector: 'app-register',
  templateUrl: './register.page.html',
  styleUrls: ['./register.page.scss'],
})
export class RegisterPage implements OnInit {
  user = {} as User;

  constructor(private toastCtrl: ToastController) {}

  ngOnInit() {}

  async register(user: User){}

  showToast(message: string){
    this.toastCtrl.create({
      message:message,
      duration: 3000
    })
    .then(toastData => toastData.present());
  }
}

```

Actualice el home page.html así:


```

<ion-header [translucent]="true">
  <ion-toolbar color="success">
    <ion-title>
      App T de A 📄
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">

  <ion-fab vertical="bottom" horizontal="end" slot="fixed">
    <ion-fab-button >
      <ion-icon name="add"></ion-icon>
    </ion-fab-button>
  </ion-fab>

  <ion-list lines="none">
    <ion-item-sliding *ngFor="let post of posts">
      <ion-item class="no-padding">
        <ion-label>
          <h2>{{post.title}}</h2>
          <p>{{post.details}}</p>
        </ion-label>
      </ion-item>

      <ion-item-options side="start">
        <ion-item-option>
          <ion-icon slot="icon-only" name="pencil-outline"></ion-icon>
        </ion-item-option>
      </ion-item-options>

      <ion-item-options side="end">
        <ion-item-option color="danger">
          <ion-icon slot="icon-only" name="trash"></ion-icon>
        </ion-item-option>
      </ion-item-options>
    </ion-item-sliding>

  </ion-list>

  <div id="container">
    <strong>Estoy listo para ser el mejor desarrollador movil con Ionic en el
    TdeA </strong>
  </div>

```

```

    <p>La página de mi Universidad 📖 <a target="_blank" rel="noopener
noreferrer" href="https://www.tdea.edu.co/">TdeA</a></p>
    <p>La página de mi profesor 😊 <a target="_blank" rel="noopener noreferrer"
href="https://dfamsoluciones.com/">Daniel</a></p>
    <p>Comenzar con Ionic 📖 <a target="_blank" rel="noopener noreferrer"
href="https://ionicframework.com/docs/components">UI Components</a></p>
  </div>
</ion-content>

```

Y el homepage.ts así:

```

import { Component } from '@angular/core';
import { LoadingController, ToastController } from '@ionic/angular';
import { AngularFireStore } from '@angular/fire/compat/firestore';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})

export class HomePage {

  posts:any;

  constructor(
    private loadingCtrl: LoadingController,
    private toastCtrl: ToastController,
    private firestore : AngularFireStore
  ) {}

  ionViewWillEnter(){}

  async getPosts(){
    let loader = await this.loadingCtrl.create({
      message: "Espere un momento por favor..."
    });
    await loader.present();

    try {
      this.firestore.collection('posts')
        .snapshotChanges()
        .subscribe((data:any[]) =>{

          this.posts = data.map((e:any) =>{

```

```

        return{
            id: e.payload.doc.id,
            title: e.payload.doc.data()["title"],
            details: e.payload.doc.data()["details"]
        }
    });
});

await loader.dismiss();
} catch (e:any) {
    e.message = "mensaje de error del home";
    let errorMessage = e.message || e.getLocalizedMessage();

    this.showToast(errorMessage);
}
}

showToast(message: string) {
    this.toastCtrl.create({
        message: message,
        duration: 5000 // Aumenta la duración del mensaje a 5 segundos
    }).then(toastData => toastData.present());
}
}

```

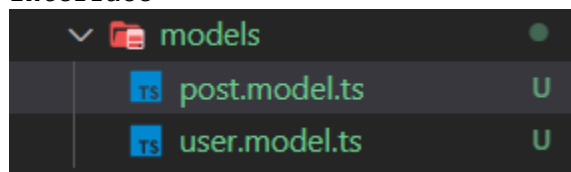
Ahora diríjase a su tsconfig.json y modifique el siguiente valor así:

CompilerOptions:

"skipLibCheck": true,

Para continuar con el crud crearemos la página para agregar. ionic generate page addPost , también para editPage

Cree un nuevo arch modelo para post en la carpeta models y asigne su debida interface



En la homepage.ts agregue lo siguiente:

```

ionViewWillEnter(){
    this.getPosts();
}

```

Actualice el app.modules.ts así:

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';


import { IonicModule, IonicRouteStrategy } from '@ionic/angular';

import { AppComponent } from './app.component';
import { AppRoutingModuleModule } from './app-routing.module';

import { AngularFireAuthModule } from "@angular/fire/compat/auth";
import { AngularFireModule } from '@angular/fire/compat/';
import { environment } from 'src/environments/environment';
import { AngularFireStoreModule } from '@angular/fire/compat/firestore';

@NgModule({
  declarations: [AppComponent],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    AppRoutingModuleModule,
    AngularFireModule.initializeApp(environment.FIREBASE_CONFIG),
    AngularFireAuthModule,
    AngularFireStoreModule
  ],
  providers: [{ provide: RouteReuseStrategy, useClass: IonicRouteStrategy }],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

Ahora agregue lo siguiente a addppostpage.html

```
<ion-header>
  <ion-toolbar color="success">
    <ion-title>Agregar  </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-item>
    <ion-label position="floating">Título</ion-label>
```

```

    <ion-input type="text" required [(ngModel)]="post.title" ></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Detalles</ion-label>
    <ion-input type="text" required [(ngModel)]="post.details" ></ion-input>
  </ion-item>

  <ion-button class="ion-padding" color="primary" expand="full"
(click)="createPost(post)">Crear</ion-button>

</ion-content>

```

Y a add-postpage.ts lo siguiente:

```

import { Component, OnInit } from '@angular/core';
import { ToastController, LoadingController } from '@ionic/angular';
import { Post } from '../models/post.model';
import { AngularFireAuth } from '@angular/fire/compat/auth';
import { NavController } from '@ionic/angular';
import { AngularFirestore } from '@angular/fire/compat/firestore';

@Component({
  selector: 'app-add-post',
  templateUrl: './add-post.page.html',
  styleUrls: ['./add-post.page.scss'],
})
export class AddPostPage implements OnInit {

  post = {} as Post;

  constructor(private toastCtrl: ToastController,
    private afAuth: AngularFireAuth,
    private loadingCtrl: LoadingController,
    private navCtrl: NavController,
    private firestore: AngularFirestore) { }

  ngOnInit() {
  }

  async createPost(post: Post){

    if (this.formValidation()) {
      let loader = await this.loadingCtrl.create({
        message: "Espere un momento por favor..."

```

```

    });
    await loader.present();

    try {
        await this.firestore.collection("posts").add(post);

    } catch (e:any) {
        e.message = "Mensaje de error en post";
        let errorMessage = e.message || e.getLocalizedMessage();

        this.showToast(errorMessage);
    }

    await loader.dismiss();

    this.navCtrl.navigateRoot("home");
}

}

formValidation() {
    if (!this.post.title) {
        this.showToast("Ingrese un titulo");
        return false;
    }

    if (!this.post.details) {
        this.showToast("Ingrese una descripción");
        return false;
    }

    return true;
}

showToast(message: string) {
    this.toastCtrl.create({
        message: message,
        duration: 5000 // Aumenta la duración del mensaje a 5 segundos
    }).then(toastData => toastData.present());
}


```

```
}
```

Actualice enrutador de la aplicación en la parte del edit-page así:

```
{
  path: 'edit-page/:id',
  loadChildren: () => import('./edit-page/edit-page.module').then( m =>
m.EditPagePageModule)
},
```

El código para el edit.page.html sería el siguiente:

```
<ion-header>
  <ion-toolbar color="success">
    <ion-title>Editar elemento 
```

Y para el archivo edit.page.ts así:

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { Post } from '../models/post.model';
import { LoadingController, ToastController } from '@ionic/angular';
import { AngularFireStore } from '@angular/fire/compat/firestore';
import { Router } from '@angular/router';

@Component({
```

```

selector: 'app-edit-page',
templateUrl: './edit-page.page.html',
styleUrls: ['./edit-page.page.scss'],
})
export class EditPagePage implements OnInit {

  post = {} as Post;
  id: any;

  constructor(
    private actRoute: ActivatedRoute,
    private loadingCtrl: LoadingController,
    private firestore: AngularFirestore,
    private router: Router,
    private toastCtrl: ToastController) {
    this.id = this.actRoute.snapshot.paramMap.get("id");
  }

  ngOnInit() {
    this.getPostById(this.id);
  }

  async getPostById(id:string){
    let loader = await this.loadingCtrl.create({
      message: "Espere un momento por favor..."
    });
    await loader.present();

    this.firestore
      .doc("posts/"+ id)
      .valueChanges()
      .subscribe((data: any) => {
        const { title, details } = data as { title: string, details: string };
        this.post.title = data.title;
        this.post.details = data.details;

        loader.dismiss();
      });

    await loader.dismiss();
  }

  async updatePost(post: Post){
    let loader = await this.loadingCtrl.create({
      message: "Actualizando..."
    });
  }

```



```

});
await loader.present();

this.firestore
.doc("posts/" + this.id)
.update(post)
.then(() => {
  console.log("Elemento actualizado correctamente!");
  this.router.navigate(['/home']);
  loader.dismiss();
})
.catch((error) => {
  console.error("Error al actualizar el elemento: ", error);
  loader.dismiss();
});
}

formValidation() {
  if (!this.post.title) {
    this.showToast("Ingresa un título");
    return false;
  }

  if (!this.post.details) {
    this.showToast("Ingresa una descripción");
    return false;
  }

  return true;
}

showToast(message: string) {
  this.toastCtrl.create({
    message: message,
    duration: 5000 // Aumenta la duración del mensaje a 5 segundos
  }).then(toastData => toastData.present());
}
}

```

Recuerda siempre actualizar el servidor y siempre comprobar los errores en consola.