

Clase 4 SkelletionApp

1. Instalar Ionic CLI:

- Abre una terminal o línea de comandos y ejecuta el siguiente comando para instalar el CLI de Ionic globalmente:

```
npm install -g @ionic/cli
```

- Verifica la instalación con:

```
ionic --version
```

Paso 2: Crear un Nuevo Proyecto con Ionic y Angular

1. Crear un nuevo proyecto Ionic:

- Para crear un nuevo proyecto con Ionic y Angular, ejecuta el siguiente comando:

```
ionic start SkeletonAPP blank --type=angular
```

- Elige "Angular" como el framework durante la configuración interactiva.
- Navega al directorio del proyecto recién creado:

```
cd SkeletonAPP
```

2. Servir la aplicación inicial:

- Para iniciar el servidor de desarrollo y ver la aplicación básica en el navegador, utiliza:

```
ionic serve
```

- Esto abrirá la aplicación en tu navegador por defecto en

`http://localhost:8100` .

Paso 3: Crear las Páginas Login e Inicio

1. Generar la página Login:

- Usa el comando de Ionic para generar una nueva página llamada "login":

```
ionic generate page login
```

2. Generar la página Inicio:

- Ahora, genera una página llamada "inicio" (en lugar de Home):

```
ionic generate page inicio
```

Paso 4: Configurar el Enrutamiento

1. Actualizar el archivo de enrutamiento:

- Abre el archivo `src/app/app-routing.module.ts` y asegúrate de que el enrutamiento esté configurado para las páginas login e inicio:

```
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from
 '@angular/router';

const routes: Routes = [
  { path: '', redirectTo: 'login', pathMatch: 'full' },
  { path: 'login', loadChildren: () => import('./login/
login.module').then(m => m.LoginPageModule) },
  { path: 'inicio', loadChildren: () => import('./inici
o/inicio.module').then(m => m.InicioPageModule) },
];

@NgModule({
  imports: [
```

```

    RouterModule.forRoot(routes, { preloadingStrategy:
PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModuleModule { }

```

Paso 5: Implementar la Página Login

1. Estructura HTML de la Página Login:

- Modifica el archivo `src/app/login/login.page.html` con el siguiente código:

```

<ion-header>
  <ion-toolbar>
    <ion-title>Login</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-item>
    <ion-label position="floating">Usuario</ion-label>
    <ion-input [(ngModel)]="username" required minlength=
h="3" maxlength="8" type="text"></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Contraseña</ion-label>
    <ion-input [(ngModel)]="password" required minlength=
h="4" maxlength="4" type="password"></ion-input>
  </ion-item>

  <ion-button expand="full" (click)="login()">Ingresar
</ion-button>
</ion-content>

```

2. Lógica de la Página Login:

- Implementa la lógica de la página login en `src/app/login/login.page.ts` :

```
import { Component } from '@angular/core';
import { NavController } from '@ionic/angular';

@Component({
  selector: 'app-login',
  templateUrl: './login.page.html',
  styleUrls: ['./login.page.scss'],
})
export class LoginPage {
  username: string = '';
  password: string = '';

  constructor(private navCtrl: NavController) {}

  login() {
    if (this.username.length >= 3 && this.username.length <= 8 && this.password.length === 4) {
      this.navCtrl.navigateForward('/inicio', {
        queryParams: { username: this.username }
      });
    } else {
      alert('Ingresa datos válidos');
    }
  }
}
```

Paso 6: Implementar la Página Inicio

1. Estructura HTML de la Página Inicio:

- Modifica el archivo `src/app/inicio/inicio.page.html` con el siguiente código:

```

<ion-header>
  <ion-toolbar>
    <ion-title>Inicio</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-item>
    <ion-label position="floating">Nombre</ion-label>
    <ion-input [(ngModel)]="name" required></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Apellido</ion-label>
    <ion-input [(ngModel)]="surname" required></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Nivel de Educación</ion-label>
    <ion-input [(ngModel)]="educationLevel" required></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Fecha de Nacimiento</ion-label>
    <ion-datetime [(ngModel)]="birthdate"></ion-datetime>
  </ion-item>

  <ion-button expand="full" (click)="clearFields()">Limpiar</ion-button>
  <ion-button expand="full" (click)="showMessage()">Mensaje</ion-button>

```

```
trar</ion-button>
</ion-content>
```

2. Lógica de la Página Inicio:

- Implementa la lógica de la página inicio en `src/app/inicio/inicio.page.ts`:

```
import { Component } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { AlertController } from '@ionic/angular';

@Component({
  selector: 'app-inicio',
  templateUrl: './inicio.page.html',
  styleUrls: ['./inicio.page.scss'],
})
export class InicioPage {
  name: string = '';
  surname: string = '';
  educationLevel: string = '';
  birthdate: string = '';
  username: string;

  constructor(private route: ActivatedRoute, private alertController: AlertController) {
    this.route.queryParams.subscribe(params => {
      this.username = params['username'];
    });
  }

  clearFields() {
    this.name = '';
    this.surname = '';
    this.educationLevel = '';
    this.birthdate = '';
  }
}
```

```
async showMessage() {  
  const alert = await this.alertCtrl.create({  
    header: 'Información',  
    message: `Nombre: ${this.name} ${this.surname}`,  
    buttons: ['OK']  
  });  
  await alert.present();  
}
```

Paso 7: Probar la Aplicación

1. Iniciar el servidor de desarrollo:

- Ejecuta el siguiente comando para iniciar el servidor y probar la aplicación:

```
ionic serve
```