

Universidad Carlos III de Madrid

Grado en Ingeniería Informática

Diseño de Sistemas Interactivos

Uso de servicios

Introducción

La presente micro-práctica consiste en un ejercicio guiado que el alumno deberá resolver de manera individual.

Objetivos

Adquirir conocimientos básicos para el desarrollo de aplicaciones móviles utilizando ionic.

Duración

La resolución de la micro-práctica comenzará durante la clase práctica y deberá ser actualizada en el repositorio de Github como máximo antes del inicio de la próxima clase práctica.

Preparación para la sesión

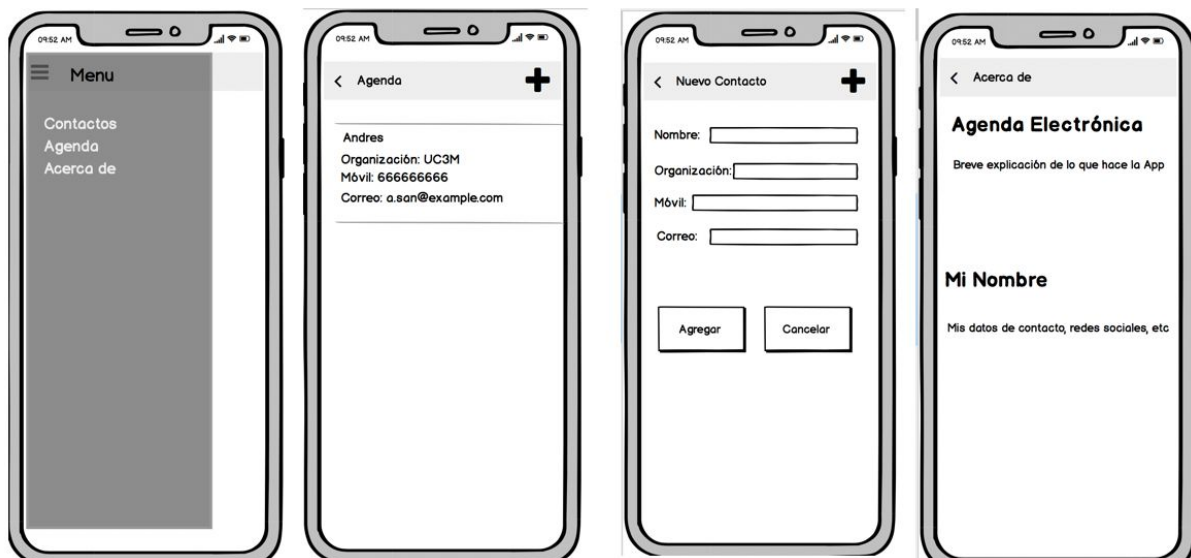
Para realizar la micro-práctica se recomienda que el alumno haya revisado la documentación de ionic(ionic v3: <https://ionicframework.com/docs/v3/> ionic v4: <https://ionicframework.com/docs/>).

Escenario

Se quiere realizar una aplicación móvil para manejar contactos y una agenda con las actividades del usuario. En esta primera práctica de ionic realizaremos la estructura de navegación de la app. En concreto la app tendrá las siguientes páginas:

- Libreta de Contactos
- Agenda electrónica
- Acerca de.

Diagrama de navegación de las páginas (se desarrollará en varias clases).



Requisitos funcionales:

1. Lista de Contactos que permita visualizar, agregar, modificar y eliminar contactos. Los contactos deben tener los siguientes datos: Nombre, Organización, Número de Teléfono, Correo Electrónico.
2. Agenda que permita visualizar, agregar, modificar y eliminar los eventos del usuario. Los eventos deben tener los siguientes datos: Nombre del evento, fecha, lugar, prioridad.
3. Incluir una página con los datos del desarrollador.

Tareas

1. Crear el modelo de datos para contactos.
2. Crear el servicio para poder agregar, recuperar, modificar y eliminar contacto.
3. Crear una lista en la página **libreta-contactos**.
4. Inyectar servicios.

Instrucciones de realización

Abrir proyecto

Para esta micro-práctica se continuará trabajando con el proyecto realizado en la micro-práctica 4.

1. Localiza y abre el proyecto elaborado en la micro-practica 4.
2. Comprueba el correcto funcionamiento del proyecto creado.

```
$ionic serve
```

Crear el modelo de datos para contactos

1. Añadir en el directorio de trabajo **src** la carpeta models
2. Añadir en el directorio de trabajo el archivo **src/models/contact.model.ts**.
3. Definir el modelo de datos de acuerdo a los requerimientos establecidos.

```
export interface Contact {  
  
  key?: string; // manejará el id cuando se utiliza firebase  
  nombre: string;  
  organizacion: string;  
  movil: string;  
  correo: string;  
}
```

Crear servicio **ContactService**

1. Añadir en el directorio de trabajo **src** la carpeta **services**.
2. Dentro de la carpeta crear el archivo **contact.service.ts**
3. Definir un array **contacts: Contact []** que contenga la lista de contactos de la aplicación.
4. Definir el método **getContact ()** que retorne la lista de contactos.
5. Definir el método **addContact()** que mediante el método **push** de la lista añade un nuevo contacto.
6. Definir el método **updateContact()**
7. Definir el método **removeContact()**
8. Declarar el servicio **ContactService** dentro del apartado providers del módulo principal de la aplicación.

contact.service.ts

```
import { Contact } from "../models/contact.model";

export class ContactService{
    private contacts: Contact []=
    [{"nombre":"Andres","organizacion":"UC3M","movil":"666666666","correo":"andres@example.com"}];

    constructor() {

    }

    addContact(value: Contact){
        this.contacts.push(value);
    }

    getContacts(){
        return this.contacts;
    }

    updateContact(value: Contact){

    }

    removeContact(value: Contact){

    }
}
```

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';

import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';
import { ListPage } from '../pages/list/list';
import { LibretaContactosPage } from '../pages/libreta-contactos/libreta-contactos';
import { AcercaDePage } from '../pages/acerca-de/acerca-de';
import { NuevoContactoPage } from '../pages/nuevo-contacto/nuevo-contacto';

import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';
import { ContactService } from '../services/contact.service';
import { Services } from '@angular/core/src/view';

@NgModule({
  declarations: [
    MyApp,
    HomePage,
    ListPage,
    LibretaContactosPage,
    AcercaDePage,
    NuevoContactoPage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    HomePage,
    ListPage,
```

```
LibretaContactosPage,  
AcercaDePage,  
NuevoContactoPage  
],  
providers: [  
  StatusBar,  
  SplashScreen,  
  {provide: ErrorHandler, useClass: IonicErrorHandler},  
  ContactService  
]  
})  
export class AppModule {}
```

Crear lista de contactos

1. Crear la lista de contacto mediante el UI-component **list** para ello utiliza las etiquetas **<ion-list>** e **<ion-item>**.
2. Utiliza la directiva ***ngfor** para poblar la lista con los contactos obtenidos del servicio **ContactService**. Utiliza interpolación para mostrar los datos del contacto (Nombre, Organización, Móvil, Correo)

libreta-contactos.html (dentro de <ion-content padding>)

```
<ion-list>  
  <button ion-item *ngFor="let contact of contacts">  
  
    <p><strong>Nombre: </strong>{{contact.nombre}}</p>  
  
    <p><strong>Organización: </strong>{{contact.organizacion}}</p>  
  
    <p><strong>Móvil: </strong>{{contact.movil}}</p>  
  
    <p><strong>Correo: </strong>{{contact.correo}}</p>  
  </button>  
  
</ion-list>
```

Inyectar servicios **ContactoService**

Servicio para obtener contactos.

1. Sobre-escribir el método **ionViewWillEnter()** (Este método se ejecuta cada vez que la página es vista por el usuario).
2. Crear la variable **contacts** que sea un array de **Contact**.
3. En el método **ionViewWillEnter()** utilizar el método **getContacts()** del servicio **ContactService** para poblar el array **contacts**.

libreta-contactos.ts

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { NuevoContactoPage } from '../nuevo-contacto/nuevo-contacto';
import { Contact } from '../../models/contact.model';
import { ContactService } from '../../services/contact.service';

@IonicPage()
@Component({
  selector: 'page-libreta-contactos',
  templateUrl: 'libreta-contactos.html',
})
export class LibretaContactosPage {

  contacts: Contact[] = [];

  constructor(public navCtrl: NavController, public navParams: NavParams, private
ContactService:ContactService) {
  }

  ionViewWillEnter(){

    this.contacts = this.ContactService.getContacts();

  }

  onLoadContactosPage() {
    this.navCtrl.push(NuevoContactoPage);
  }
}
```

Inyectar servicio onAddContact()

1. Definir el método **onAddContact(value: {nombre:string,organizacion:string,movil:string,correo:string})**
2. Este método realiza dos acciones. En primer lugar mediante el método **addContact** del servicio **contactService** añadir los datos del formulario **nuevo-contacto**. En segundo lugar, destruye la página para regresar a la página anterior con el método **pop()**.

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { ContactService } from '../../../services/contact.service'

@IonicPage()
@Component({
  selector: 'page-nuevo-contacto',
  templateUrl: 'nuevo-contacto.html',
})
export class NuevoContactoPage {

  constructor(public navCtrl: NavController, public navParams: NavParams, private
contactService: ContactService) {

  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad NuevoContactoPage');
  }

  onAddContact(value:
{nombre:string,organizacion:string,movil:string,correo:string}) {

    this.contactService.addContact(value);
    this.navCtrl.pop();
  }
}
```


Formulario en la página nuevo-contacto

Utilizar las directivas (**ngSubmit**), **ngModel** para llenar el formulario con los datos de un nuevo contacto.

1. El código se coloca dentro de la etiqueta **<ion-content padding>**

```
<form (ngSubmit)="onAddContact(f.value)" #f="ngForm">
  <ion-list>
    <ion-item>
      <ion-label>Nombre</ion-label>
      <ion-input type="text" placeholder="Texto" name="nombre" ngModel
required></ion-input>
    </ion-item>

    <ion-item>
      <ion-label>Organización</ion-label>
      <ion-input type="text" placeholder="Texto" name="organizacion" ngModel
required></ion-input>
    </ion-item>

    <ion-item>
      <ion-label>Numero</ion-label>
      <ion-input type="text" placeholder="666666666" name="movil"
ngModel></ion-input>
    </ion-item>

    <ion-item>
      <ion-label>Correo</ion-label>
      <ion-input type="text" placeholder="name@example.com" name="correo"
ngModel></ion-input>
    </ion-item>

  </ion-list>

  <button ion-button block type="submit"
[disabled]="!f.valid">Aceptar</button>

</form>
```

Solución

Nota: Se recomienda ir a la solución al final de la resolución de esta micro-práctica solo para comparar tu solución con la propuesta por los profesores y si no te queda claro algún apartado de la misma.

Solución ionic v3

https://github.com/andresSantos9/micro_05.git