

Ionic Components

Desarrollo de Aplicaciones Híbridas

Tabla de contenidos

- 1. Resumen 1
- 2. Introducción 3
 - 2.1. Crear una página 3
 - 2.2. Crear una página anidada 5
 - 2.3. Añadir tarjetas 7
 - 2.4. Iterando sobre un conjunto de resultados 9
 - 2.5. Eliminación de las páginas innecesarias 10
 - 2.6. Modificación de la barra páginas 11
 - 2.7. Modificación de la página de contacto 11

1

Resumen

Veremos los componentes básicos para el desarrollo de una app Ionic.

Objetivos

- Conocer los componentes básicos de Ionic
- Aprender a usar la navegación entre páginas

2

Introducción

Comenzaremos creando una app basada en pestañas. Inicialmente no integraremos la app con Cordova y no le incluiremos el Ionic Pro SDK.

```
$ ionic start ionicComponents tabs
$ cd ionicComponents
$ ionic serve
```

2.1. Crear una página

Crearemos una nueva página del tipo ficha denominada Estudios de esta forma

```
$ ionic generate tabs

? What should the name be? Estudios

? How many tabs? 1

? Name of this tab: Estudios
[OK] Generated a tabs page named Estudios!
```

A continuación tenemos que añadir la página a `app.module.ts`. Hay que importarla y añadirla a los arrays `declarations` y `entryComponents`

```
...
import { EstudiosPage } from '../pages/estudios/estudios';
...
@NgModule({
  declarations: [
```

```

    ...
    EstudiosPage
  ],
  ...
  entryComponents: [
    ...,
    EstudiosPage
  ],
  ...
})
export class AppModule {}

```

A continuación, hay que actualizar el componente `src/pages/tabs/tabs.ts` importando la página e incluyéndola en la clase `TabsPage`.

```

...
import { EstudiosPage } from '../estudios/estudios';
...
export class TabsPage {

  ...
  tab4Root = EstudiosPage;
  ...

}

```

Por último, ya sólo hay que incluir una nueva pestaña en `src/pages/tabs/tabs.ts` con la ficha Estudios. El [conjunto de iconos disponibles de Ionic](https://ionicframework.com/docs/ionicons/)¹ proporcionará el icono adecuado.

```

<ion-tabs>
  <ion-tab [root]="tab1Root" tabTitle="Home" tabIcon="home"></ion-tab>
  <ion-tab [root]="tab2Root" tabTitle="About" tabIcon="information-
circle"></ion-tab>
  <ion-tab [root]="tab3Root" tabTitle="Contact" tabIcon="contacts"></
ion-tab>
  <ion-tab [root]="tab4Root" tabTitle="Estudios" tabIcon="book"></ion-
tab>
</ion-tabs>

```

¹ <https://ionicframework.com/docs/ionicons/>

2.2. Crear una página anidada

Desde la página Estudios daremos la opción de ir a estudios de Grado, Máster y Doctorado. Esto lo haremos creando tres botones que abrirán sus respectivas páginas.

Comenzamos creando las páginas

```
$ ionic generate page grados
$ ionic generate page masters
$ ionic generate page doctorado
```

Esto creará tres directorios `grados`, `masters` y `doctorado` en `src/app/pages`. Cada directorio contendrá su controlador, su vista y su hoja de estilos.

A continuación, tenemos que modificar `src/app/app.module.ts` para incluir las páginas creadas.

```
...
import { GradosPage } from '../pages/grados/grados';
import { MastersPage } from '../pages/masters/masters';
import { DoctoradoPage } from '../pages/doctorado/doctorado';
...
@NgModule({
  declarations: [
    ...
    GradosPage,
    MastersPage,
    DoctoradoPage
  ],
  ...
  entryComponents: [
    ...
    GradosPage,
    MastersPage,
    DoctoradoPage
  ],
  ...
})
export class AppModule {}
```

En la vista de Estudios `src/app/pages/estudios/estudios.html` tenemos que añadir tres métodos para abrir el formulario correspondiente a estudios de grado, máster o doctorado. Para ellos usaremos la instancia de `NavController` disponible en el constructor pasándole la página que queremos abrir al método `push`.

Como siempre, tendremos que importar los componentes que queramos usar.

```
...
import { GradosPage } from '../grados/grados';
import { MastersPage } from '../masters/masters';
import { DoctoradoPage } from '../doctorado/doctorado';
...
export class EstudiosPage {
  ...
  openGrades() {
    this.navCtrl.push(GradosPage);
  }

  openMasters() {
    this.navCtrl.push(MastersPage);
  }

  openPhD() {
    this.navCtrl.push(DoctoradoPage);
  }
  ...
}
```

Por último, en la vista de Estudios (`src/app/pages/estudios/estudios.html`) tenemos que crear un enlace para abrir cada página. En nuestro caso lo haremos creando tres tags `` HTML, uno para cada opción, añadiéndoles el evento (`click`) configurado al método `openGrades()`, `openMasters()` u `openPhD()` según proceda.

```
<ion-header>
  <ion-navbar>
    <ion-title>Estudios</ion-title>
  </ion-navbar>
</ion-header>

<ion-content>
```

```
<img src = "../../../assets/imgs/Grados.png" (click)="openGrades()" >
<img src = "../../../assets/imgs/Masters.png" (click)="openMasters()" >
<img src = "../../../assets/imgs/Doctorado.png" (click)="openPhD()" >
</ion-content>
```

Las imágenes las colocaremos en la carpeta `src/app/assets/img`. Las usadas en este tutorial han sido capturadas directamente de la web de la UAL y recortadas posteriormente.



Al convertir las imágenes en elementos *clickables* si vemos la aplicación en un navegador no veremos que el puntero del ratón cambie a una mano cuando esté sobre las imágenes. Esto se soluciona añadiendo el estilo siguiente al archivo de estilos de la página de Estudios (`src/app/pages/estudios/estudios.scss`).

```
img {
  cursor: pointer;
}
```

2.3. Añadir tarjetas

Las [tarjetas](#)² son una forma de destacar texto en una página.

Las tarjetas pueden contener cabecera y contenido. En nuestra aplicación añadiremos a la vista de estudios (`src/app/pages/estudios/estudios.html`) crearemos una tarjeta con un enlace para mostrar más información sobre los estudios de la UAL en la web oficial.

```
<ion-card>
  <ion-card-header>
    Información complementaria
  </ion-card-header>
  <ion-card-content>
    Consulta la <a href = "https://www.ual.es/estudios">web de
    estudios en la UAL</a>
  </ion-card-content>
</ion-card>
```

² <https://ionicframework.com/docs/components/#cards>

Podemos mejorar nuestra app convirtiendo en tarjetas las imágenes que nos dan acceso a los grados, másters y doctorado creadas anteriormente y enriqueciéndolas con algunos estilos.

Ejemplo 2.1. El archivo `src/app/pages/estudios/estudios.html`

```
<ion-header>
  <ion-navbar>
    <ion-title>Estudios</ion-title>
  </ion-navbar>
</ion-header>

<ion-content class="card-background-page">
  <ion-card>
    <img src = "../assets/imgs/
Grados.png" (click)="openGrades()" >
    <div class="card-title">Grados</div>
  </ion-card>
  <ion-card>
    <img src = "../assets/imgs/
Masters.png" (click)="openMasters()" >
    <div class="card-title">Másters</div>
  </ion-card>
  <ion-card>
    <img src = "../assets/imgs/
Doctorado.png" (click)="openPhD()" >
    <div class="card-title">Doctorado</div>

  </ion-card>

  <ion-card>
    <ion-card-header>
      Información complementaria
    </ion-card-header>
    <ion-card-content>
      Consulta la <a href = "https://www.ual.es/estudios">web de
      estudios en la UAL</a>
    </ion-card-content>
  </ion-card>
</ion-content>
```

Los estilos a aplicar serían estos

Ejemplo 2.2. El archivo `src/app/pages/estudios/estudios.scss`

```
page-estudios {  
  
}  
  
.card-background-page {  
  
  ion-card {  
    position: relative;  
    text-align: right;  
  }  
  
  .card-title {  
    position: absolute;  
    top: 36%;  
    font-size: 2.0em;  
    width: 100%;  
    font-weight: bold;  
    color: #fff;  
  }  
}
```

2.4. Iterando sobre un conjunto de resultados

Tanto los grados, como los masters y doctorado mostrarán una lista de valores. Inicialmente, esos valores los tendremos almacenados en un JSON en el controlador de cada componente. Posteriormente veremos cómo crear un servicio que recupere los datos de una base de datos.

Mostraremos los datos usando [listas](#)³ iterando sobre ellas con la directiva `*ngFor`. En nuestro caso usaremos dos bucles anidados. El externo iterará sobre las ramas y el interno sobre los estudios de cada rama.

En la vista `src/app/pages/grados/grados.scss` cambiaremos el contenido de la página por el siguiente. En la iteración externa nos quedamos con la rama. Sobre cada rama iteraremos sobre los distintos grados de la rama.

³ <https://ionicframework.com/docs/components/#lists>

```
<ion-content>
  <ion-item-group>
    <ion-list *ngFor="let rama of estudios">
      <ion-item-divider>{{rama.rama}}</ion-item-divider>
      <ion-item *ngFor="let gradoRama of
rama.estudios" (click)="openURL(gradoRama.url)">
        <h2>{{gradoRama.estudios}}</h2>
        <p>{{gradoRama.plan}}</p>
      </ion-item>
    </ion-list>
  </ion-item-group>
</ion-content>
```

Al pulsar sobre unos estudios abriremos la URL de la web oficial. Para ello definiremos un método `openURL()` en el controlador. Este sería el código del método

```
openURL(url: string) {
  window.open(url, '_self');
}
```

2.5. Eliminación de las páginas innecesarias

Al crear el proyecto se crearon una serie de páginas que ahora resultan innecesarias. Vamos a eliminar la página About.

1. Modificar el archivo `src/app/app.module.ts`
 - a. Eliminar la importación de `AboutPage`.
 - b. Eliminar `AboutPage` de los arrays `declarations` y `entryComponents`.
2. Modificar el archivo `src/app/pages/tabs/tabs.ts`
 - a. Eliminar la importación de `AboutPage`.
 - b. Eliminar en la clase el `tabRoot` de `AboutPage`.
3. Modificar el archivo `src/app/pages/tabs/tabs.html`
 - a. Eliminar el `ion-tab` de `About`.

2.6. Modificación de la barra páginas

Queremos cambiar el orden en que aparecen las páginas en la barra inferior y el icono de Contacto. Ahora Contacto será la última.

En el archivo `src/app/pages/tabs/tabs.ts`:

1. Modificar el atributo `tabTitle` al valor `Contacto`
2. Modificar el atributo `tabIcon` al valor `information-circle`

2.7. Modificación de la página de contacto

