

T1-Act2: Práctica IONIC

A.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\jarro\Documents\DAM\2º\MÓVILES\JesusArroyoAlvaroPec> ionic start
? Use the app creation wizard? No
Pick a framework!
```

```
? Framework: Angular

Every great app needs a name!

Please enter the full name of your app. You can char
the first argument to ionic start.

? Project name: UT1-Ionic
```

Iniciamos nuestra app.

```
Starter templates are ready-to-go Ionic apps that come packed with everything you need to
prompt next time, supply template, the second argument to ionic start.

? Starter template:
  tabs      | A starting project with a simple tabbed interface
  sidemenu  | A starting project with a side menu with navigation in the content area
> blank    | A blank starter project
  list      | A starting project with a list
  my-first-app | A template for the "Build Your First App" tutorial
```

Usamos la plantilla básica blank.

```
PS C:\Users\jarro\Documents\DAM\2º\MÓVILES\JesusArroyoAlvaroPec\UT1-Ionic> ionic generate
? What would you like to generate? (Use arrow keys)
> page
  component
  service
  module
  class
  directive
```

```
? Name/path of page: pages/inicio
```

Creamos la nueva pagina de inicio a través del comando que podemos ver en la imagen.

Dentro del módulo de rutas, indicamos que nos redirija a la página de inicio para que sea la que carga al iniciar la app.

A screenshot of a code editor with a dark theme. At the top, there's a commit message "You, 3 days ago • Initial commit ...". The code is in TypeScript and defines an Angular routing module. It imports NgModule from '@angular/core' and PreloadAllModules, RouterModule, Routes from '@angular/router'. A const routes array is defined with three entries: a root route that redirects to 'inicio', and two other routes for 'pagina2' and 'pagina3', each loading a corresponding page module. The @NgModule decorator is applied to the AppRoutingModule, importing RouterModule forRoot with the PreloadAllModules strategy and exporting RouterModule. The AppRoutingModule class is then exported.

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

const routes: Routes = [
  {
    path: '',
    redirectTo: 'inicio',
    pathMatch: 'full'
  },
  {
    path: 'inicio',
    loadChildren: () => import('./pages/inicio/inicio.module').then( m => m.InicioPageModule)
  },
  {
    path: 'pagina2',
    loadChildren: () => import('./pages/pagina2/pagina2.module').then( m => m.Pagina2PageModule)
  },
  {
    path: 'pagina3',
    loadChildren: () => import('./pages/pagina3/pagina3.module').then( m => m.Pagina3PageModule)
  },
];

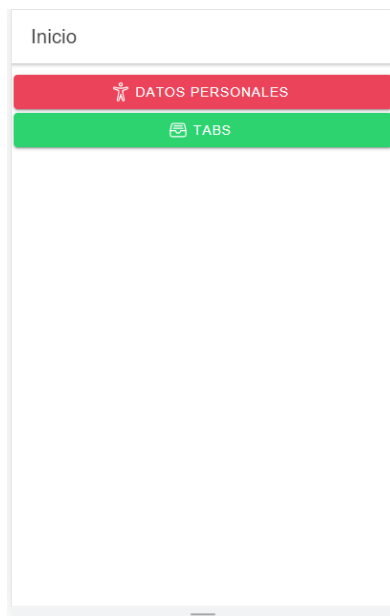
@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Creamos las páginas 2 y 3 de la misma forma que la de inicio. Aquí podemos ver las rutas de ambas páginas.

B.

```
<ion-header>
  You, 3 days ago • Realizo primer commit ...
  <ion-toolbar>
    <ion-title>Inicio</ion-title>
  </ion-toolbar>
</ion-header>

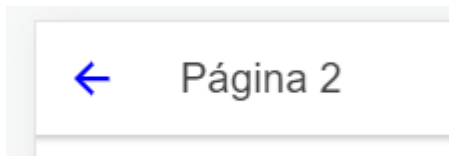
<ion-content>
  <ion-list style="text-align:center">
    <ion-button class="estilo-boton" routerLink="/pagina2">Datos personales</ion-button>
    <br>
    <ion-button class="estilo-boton2" routerLink="/pagina3">Tabs</ion-button>
  </ion-list>
</ion-content>
```



Aquí podemos ver la página de inicio, con sus botones para entrar en las otras 2 páginas.

```
<ion-header>
  You, 3 days ago • Realizo segundo commit ...
  <ion-toolbar>
    <ion-title>{{ titulo }}</ion-title>
    <ion-buttons slot="start">
      <ion-back-button class="back-button" defaultHref="Inicio"></ion-back-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>
```

```
.back-button{
  color: blue;
}
```



Aquí podemos ver como hemos creado el botón que nos redirija a la página de inicio.

C.

```
export class Pagina2Page implements OnInit {
  nombre : String;
  apellidos : String;
  dni: String;
  localidad: String;
  email: String;
  contrasena: String;

  constructor() { }

  ngOnInit() {
  }

  submit() {
    if(this.nombre==null){
      alert("Rellene el campo nombre");
    }else if(this.apellidos==null){
      alert("Rellene el campo apellidos");
    }else if(this.dni==null){
      alert("Rellene el campo DNI");
    }else if(this.email==null){
      alert("Rellene el campo Email");
    }else if (this.localidad==null){
      alert ("rellene el campo localidad")
    }else if (this.contrasena==null){
      alert ("Rellene el campo contraseña, minimo 8 caracteres")
    }
    else{
      alert("Formulario enviado");
      this.nombre="";
      this.apellidos="";
      this.dni="";
      this.localidad="";
      this.email="";
    }
  }
}
```

Este es el ts de la página 2, donde hacemos un formulario con los datos personales. En este código nos declaramos las variables que vamos a utilizar para rellenar los campos, y también ponemos condiciones para que no se pueda quedar el campo vacío, o en el caso de las contraseñas, que mínimo tengan 8 caracteres.

```

✓ <ion-content>
✓   <ion-item>
      <ion-label>Nombre</ion-label>
      <ion-input type="text" [(ngModel)]="nombre"></ion-input>
    </ion-item>

    <ion-item>
      <ion-label>Apellidos</ion-label>
      <ion-input type="text" [(ngModel)]="apellidos"></ion-input>
    </ion-item>
    <ion-item>
      <ion-label>DNI</ion-label>
      <ion-input type="text" [(ngModel)]="dni"></ion-input>
    </ion-item>

    <ion-item>
      <ion-label>Localidad</ion-label>
      <ion-input type="text" [(ngModel)]="localidad"></ion-input>
    </ion-item>

    <ion-item>
      <ion-label>Email</ion-label>
      <ion-input type="text" [(ngModel)]="email"></ion-input>
    </ion-item>

    <ion-item>
      <ion-label>Contraseña</ion-label>
      <ion-input type="password" minlength="8"></ion-input>
    </ion-item>
    <ion-button fill="solid" (click)="submit()">Enviar</ion-button>
  </ion-content>

```

En el HTML de la página 2 donde creamos el formulario de los datos personales, como podemos ver llamamos con la función ngmodel a las variables que creamos en el ts para hacer uso del databinding.

D.

```
<ion-content>
  <ion-tabs>
    <ion-tab-bar>
      <ion-tab-button tab="1dam">
        <ion-icon name="hardware-chip-outline"></ion-icon>
        <ion-label>1º DAM</ion-label>
      </ion-tab-button>

      <ion-tab-button tab="2dam">
        <ion-icon name="logo-microsoft"></ion-icon>
        <ion-label>2º DAM</ion-label>
      </ion-tab-button>

      <ion-tab-button tab="fct">
        <ion-icon name="logo-snapchat"></ion-icon>
        <ion-label>FCT</ion-label>
      </ion-tab-button>
    </ion-tab-bar>
  </ion-tabs>
</ion-content>
```

Creamos las diferentes tabs o pestañas dentro del HTML.

```
PS C:\Users\jarro\Documents\DAM\2º MÓVILES\UT1_Act2_JesusArroyoAlvaroPeco\JesusArroyoAlvaroPeco> ionic g page
? Name/path of page: pages/dam1
```

```
▼ pages
  > dam1
  > dam2
  > fct
  > inicio
  > pagina2
  > pagina3
```

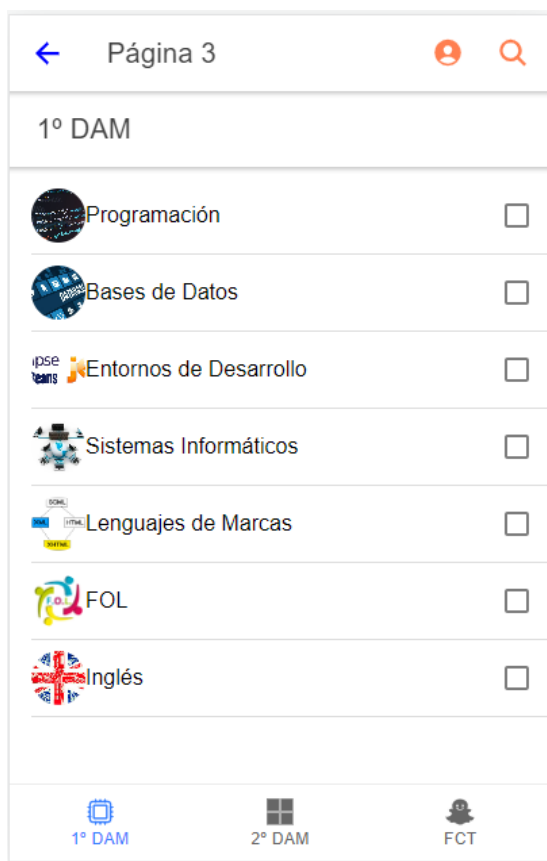
Nos creamos las 3 páginas nuevas (dam1, da2, fct).

```

path: '|',
component: Pagina3Page,
children: [
  {
    path: 'tab1',
    loadChildren: () => import('../dam1/dam1.module').then( m => m.Dam1PageModule)
  },
  {
    path: 'tab2',
    loadChildren: () => import('../dam2/dam2.module').then( m => m.Dam2PageModule)
  },
  {
    path: 'tab3',
    loadChildren: () => import('../fct/fct.module').then( m => m.FctPageModule)
  },
],

```

Configuramos las rutas dentro de nuestra pagina3-routing.module.ts para que, al pulsar en cada tab, se abra la página deseada.



Aquí podemos ver en la parte inferior las distintas tabs, y la pagina de 1º DAM que se abre dentro de su pestaña.

E.

```
<ion-content>
  <ion-list>
    <ion-item>
      <ion-avatar>
        
      </ion-avatar>
      <ion-label>Programación</ion-label>
    </ion-item>
    <ion-item>
      <ion-avatar>
        
      </ion-avatar>
      <ion-label>Bases de Datos</ion-label>
    </ion-item>
  </ion-list>
</ion-content>
```

Creamos un ion-list para hacer un listado de todas nuestras asignaturas en la página 1ºDAM (la hemos podido ver en la imagen del apartado anterior).

```
export class Dam1Page implements OnInit {

  checkPro: boolean = false;
  checkBd: boolean = false;
  checkEd: boolean = false;
  checkLm: boolean = false;
  checkSis: boolean = false;
  checkFol: boolean = false;
  checkEng: boolean = false;

  constructor(public toastController: ToastController) { }
}
```

Nos creamos unas variables que llamaremos dentro del checkbox y las inicializamos dándole valor false, para que nuestro checkbox no aparezca marcado de primeras.


```

constructor(public toastController: ToastController) { }

ngOnInit() {
}

async programacion() {
  const toast = await this.toastController.create({
    message: 'Ha elegido Programacion',
    duration: 2000,
  });
  toast.present();
}

async bbdd() {
  const toast = await this.toastController.create({
    message: 'Ha elegido Bases de Datos',
    duration: 2000,
  });
  toast.present();
}

```

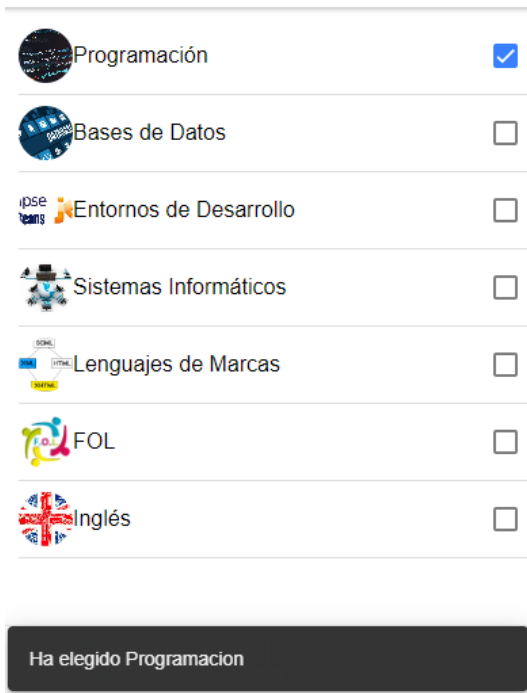
Para hacer el toast, importamos la clase ToastController y posteriormente, creamos dentro del constructor el método para que funciones nuestro toast.

```

<ion-content>
  <ion-list>
    <ion-item>
      <ion-avatar>
        
      </ion-avatar>
      <ion-label>Programación</ion-label>
      <ion-checkbox [(ngModel)]="checkPro" (click)="programacion()"></ion-checkbox>
    </ion-item>
    <ion-item>
      <ion-avatar>
        
      </ion-avatar>
      <ion-label>Bases de Datos</ion-label>
      <ion-checkbox [(ngModel)]="checkBd" (click)="bbdd()"></ion-checkbox>
    </ion-item>
  </ion-list>
</ion-content>

```

En el HTML llamamos a través de databinding, a las variables del checkbox con el ngModel, y a los métodos toast a través de click.



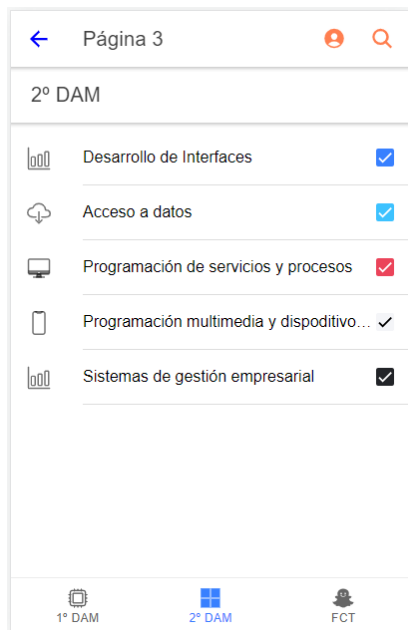
Aquí vemos como se marca nuestro checkbox y aparece el mensaje del toast.

F.

```
asignaturas=[
  {nombre:"Desarrollo de Interfaces",
    icono:"bar-chart-outline",
    checkbox: "checkbox1",
    color: "primary",
    mostrar: "false"},
  {nombre: "Acceso a datos",
    icono: "cloud-download-outline",
    checkbox: "checkbox2",
    color: "secondary",
    mostrar: "true"}],
```

En la página 2ª DAM vamos a crear una lista con el ngfor por lo que en el ts añadimos las variables que vamos a utilizar.

```
<ion-content>
  <ion-list>
    <ion-item *ngFor="let e of asignaturas">
      <ion-icon [name]="e.icono" slot="start"></ion-icon>
      <ion-label>{{e.nombre}}</ion-label>
      <ion-checkbox [color]="e.color" [(ngModel)]= "e.mostrar">{{e.checkbox}}</ion-checkbox>
    </ion-item>
  </ion-list>
</ion-content>
```



En el HTML creamos el ion list y lo recorremos con el ngfor para que así nos muestre las variables que hemos creado en el ts.

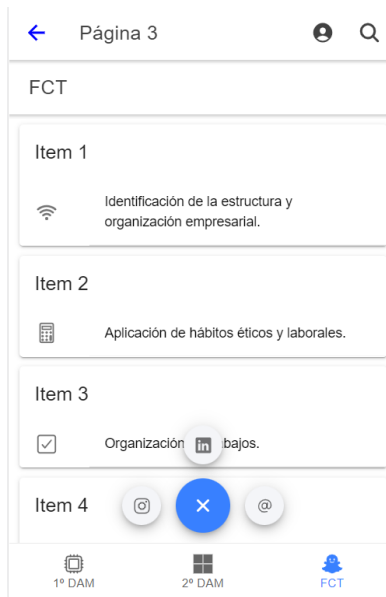
G.

```
<ion-content fullscreen>
  <ion-card>
    <ion-card-header>
      <ion-card-title>Item 1</ion-card-title>
    </ion-card-header>
    <ion-item>
      <ion-icon name="wifi" slot="start"></ion-icon>
      <ion-card-content>Identificación de la estructura y organización empresarial.</ion-card-content>
    </ion-item>
  </ion-card>

  <ion-card>
    <ion-card-header>
      <ion-card-title>Item 2</ion-card-title>
    </ion-card-header>
    <ion-item>
      <ion-icon name="calculator-outline" slot="start"></ion-icon>
      <ion-card-content>Aplicación de hábitos éticos y laborales.</ion-card-content>
    </ion-item>
  </ion-card>
</ion-content>
```

En el HTML de FCT creamos las ion-card, las ajustamos de la forma que creamos más optima, y le ponemos los iconos.

```
<ion-fab vertical="bottom" horizontal="center" slot="fixed">
  <ion-fab-button>
    <ion-icon name="add-outline"></ion-icon>
  </ion-fab-button>
  <ion-fab-list side="top">
    <ion-fab-button href="https://www.linkedin.com/in/alvaro-peco-garcia-3b2620156/"><ion-icon name="logo-linkedin"></ion-icon></ion-fab-button>
  </ion-fab-list>
  <ion-fab-list side="end">
    <ion-fab-button href="https://efa-centro.org/efa-moratalaz/"><ion-icon name="at-outline"></ion-icon></ion-fab-button>
  </ion-fab-list>
  <ion-fab-list side="start">
    <ion-fab-button href="https://www.instagram.com/arroyito92/?hl=es"><ion-icon name="logo-instagram"></ion-icon></ion-fab-button>
  </ion-fab-list>
</ion-fab>
</ion-content>
```



En la misma página también añadimos un ion-fab del que salen otros ion-fab, que nos redirecciona a otras páginas.

H.

```
PS C:\Users\jarro\Documents\DAM\2º MÓVILES\JesusArroyoAlvaroPec\UT1-Ionic> ionic generate
? What would you like to generate? module
? Name/path of module: componentes
```

Generamos desde la terminal el módulo componente.

```
PS C:\Users\jarro\Documents\DAM\2º MÓVILES\JesusArroyoAlvaroPec\UT1-Ionic> ionic generate
? What would you like to generate? component
? Name/path of component: componentes/cabecera
```

Después de generar el módulo, nos generamos el componente y le llamamos cabecera.

```

<ion-header>
  <ion-toolbar>
    <ion-title>{{ titulo }}</ion-title>
    <ion-buttons slot="start">
      <ion-back-button class="back-button" defaultHref="Inicio"></ion-back-button>
    </ion-buttons>
    <ion-buttons slot="secondary">
      <ion-button class="icon-button">
        <ion-icon slot="icon-only" name="person-circle"></ion-icon>
      </ion-button>
      <ion-button class="icon-button">
        <ion-icon slot="icon-only" name="search"></ion-icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

```

```

.back-button{
  color: blue;
}

.icon-button{
  color: coral;
}

```

Aquí nos creamos nuestra cabecera, con el backButton y el resto de botones que queremos incorporar.

También podemos ver cómo le damos estilos.

```

<app-cabecera titulo="Página 3"></app-cabecera>

```

Aquí llamamos a nuestra cabecera desde el HTML de la página que deseemos.

```

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    Pagina3PageRoutingModule,
    ComponentesModule
  ],
  declarations: [Pagina3Page]
})
export class Pagina3PageModule {}

```

Para que funcione la cabecera, debemos incluir el modulo dentro de la pagina en la que la vayamos a incorporar. En nuestro caso, pagina3.module.ts.