



Enfoques de Desarrollo de Aplicaciones Móviles Multiplataforma

Enfoque Híbrido con Ionic





Directivas de Angular: ngFor

Permite **iterar** sobre una **lista de elementos**

```
<ion-content>
  <ion-list>
    <ion-list-header>
      Superheroes de Marvel
    </ion-list-header>
    <ion-item *ngFor="let hero of marvelHeroes">
      <ion-label>{{hero.nombre}}</ion-label>
    </ion-item>
  </ion-list>
</ion-content>
```

```
export class HomePage {
  marvelHeroes = [
    {nombre: 'Spider man'},
    {nombre: 'Black Panther'},
    {nombre: 'Thor'},
    {nombre: 'Capitán América'}
  ];

  constructor() {
  }
}
```



Directivas de Angular: ngFor

Super heroes List

Superheroes de Marvel

Spider man

Black Panther

Thor

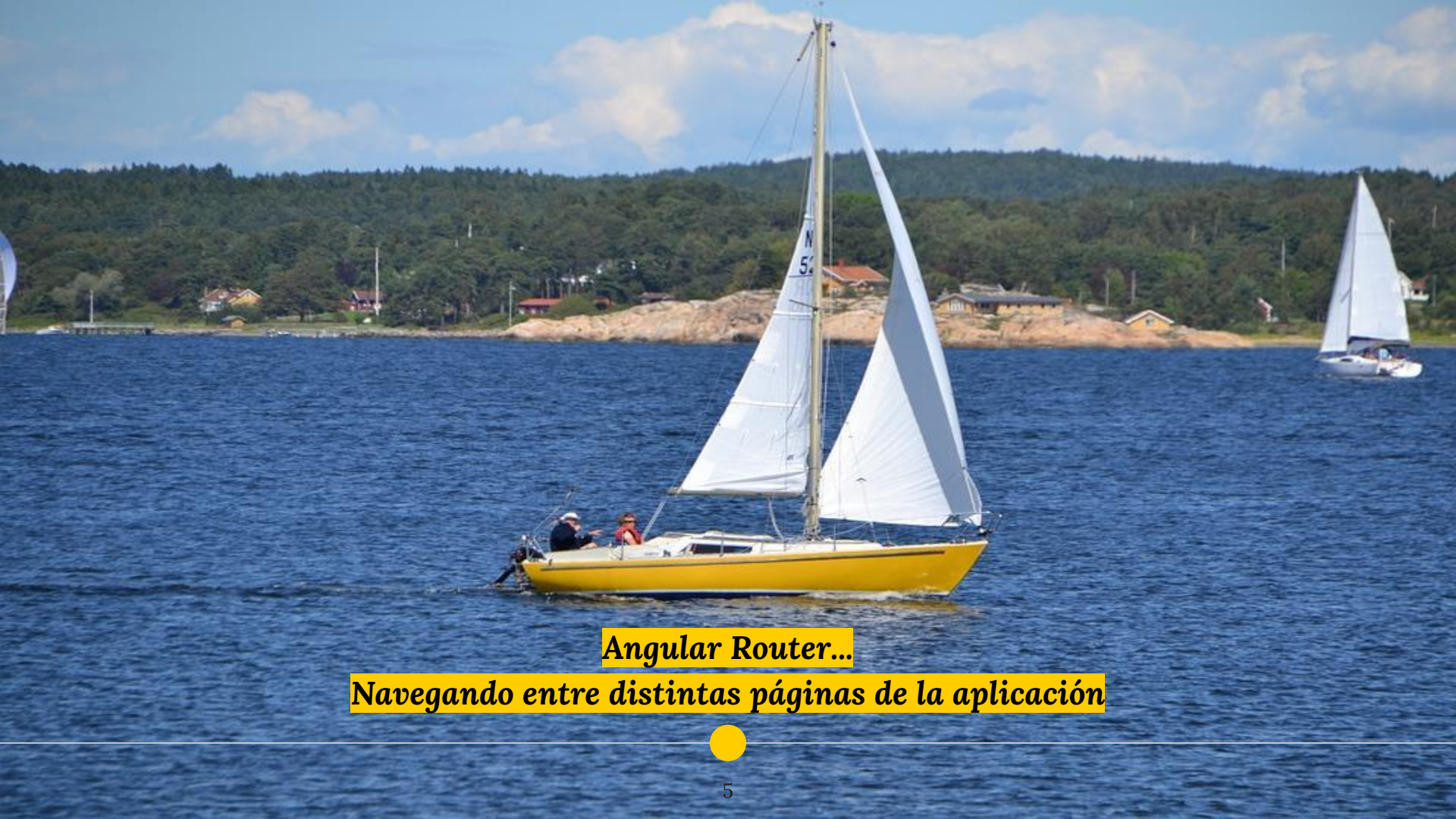
Capitán América



Actividad 1.

- ◉ Ingresar a <http://bit.ly/enfoques-A1> copiar y pegar la definición de “items” en el archivo `home.ts`
- ◉ Modificar el template del proyecto utilizando la directiva *ngFor*, obteniendo los datos necesarios desde el componente.





Angular Router...

Navegando entre distintas páginas de la aplicación





app-routing.module.ts

```
const routes: Routes = [  
  { path: '', redirectTo: 'home', pathMatch: 'full' },  
  { path: 'home', loadChildren: () => import('./home/home.module').then( m =>  
m.HomePageModule)},  
  { path: 'first-page', loadChildren:  
    './pages/first-page/first-page.module#FirstPagePageModule' },  
];
```

¿Cómo hago para que la
primer página de mi app sea
“first-page” ?



Navegación desde el template

- **routerLink** → Ruta o path al que se debe redirigir → `routerLink="/login"`
- **routerDirection** → Animación cuando se cambia de página →
 - Root
 - Forward
 - Back

```
<ion-button routerLink="/home" routerDirection="root">Login</ion-button>
```

La **inyección de dependencias** es un patrón de diseño que permite **suministrar funcionalidades, services, providers, etc, a un componente en lugar de ser el propio componente el encargado de crearlas.**

“



Inyectando NavController

```
import { NavController } from '@ionic/angular';  
  
class MyComponent {  
  constructor(public navCtrl: NavController) {  
  
  }  
}
```



Nos va a dar una instancia de NavController



Navegando a otra página

../pages/home/home.page.html

```
<ion-content padding>
  <ion-button (click)="goToPage()">Next</ion-button>
</ion-content>
```

../pages/home/home.ts

```
export class HomePage {
  constructor(public navCtrl: NavController) {

    navigateBack("/home");
    navigateRoot("/home");

    goToPage(){
      back()
      this.navCtrl.navigateForward("/home");
    }
  }
}
```



Actividad 2.

- Crear una nueva página en nuestro proyecto.
- Agregar un botón de Ionic en la página de origen, de tal manera que al presionarlo abra la nueva página.





Pasando parámetros entre las páginas app-routing.module.ts

```
const routes: Routes = [  
  { path: '', redirectTo: 'home', pathMatch: 'full' },  
  { path: 'home', loadChildren: () => import('./home/home.module').then( m =>  
    m.HomePageModule)},  
  { path: 'first-page', loadChildren:  
    './pages/first-page/first-page.module#FirstPAGePageModule' },  
  { path: 'superheroes-detail/:name', loadChildren:  
    './pages/superheroes-detail/superheroes-detail.module#SuperheroesDetailPa  
geModule' },  
];
```



Pasando parámetros entre las páginas

../pages/home/home.page.ts

Página origen

```
export class HomePage {  
  constructor(public navCtrl:  
    NavController) {  
  
  }  
  
  goToPage(name: string){  
  
    this.navCtrl.navigateForward("/superheroes  
-detail/" + name);  
  }  
}
```

../pages/detail/detail.page.ts

Página nueva

```
import {ActivatedRoute} from  
"@angular/router";  
export class HomePage {  
  name: string;  
  constructor(private activatedRoute:  
    ActivatedRoute) {  
  }  
  ngOnInit() {  
    this.activatedRoute.params.subscribe((data)  
=> {  
      console.log(data);  
      this.name = data.name  
    })  
  }  
}
```



Actividad 3.

- Enviar un dato por parámetro a la página nueva y mostrarlo en la vista.





Ionic native

Acceso a funcionalidades *nativas* del dispositivo

Capacitor, creado por el equipo de Ionic, te permite agregar **funciones nativas** a tu app **de forma sencilla**. Permite acceder a la cámara, archivos del dispositivo y ubicación, usando una **API en TypeScript**.

“



Plugins oficiales Ionic + Capacitor

<https://ionicframework.com/docs/native>

- Action Sheet
- App Launcher
- App
- Browser
- Camera
- Clipboard
- Device
- Dialog
- Filesystem
- Geolocation
- Google Maps
- Haptics
- Keyboard
- Local Notifications
- Motion
- Network
- Preferences
- Push Notifications
- Screen Reader
- Share
- Splash Screen
- Status Bar
- Text Zoom
- Toast

<https://capacitorjs.com/docs/plugins/community>



Plugins creados por la comunidad