



PROGRAMACION DE APLICACIONES MÓVILES

PGY4121

Profesor:

DuocUC 

ESCUELA DE
INFORMÁTICA Y
TELECOMUNICACIONES





Actividad N°1.2: Modifiquemos el Proyecto

Navegación Angular

Objetivos

Lo que se espera que aprendas en esta Actividad es:

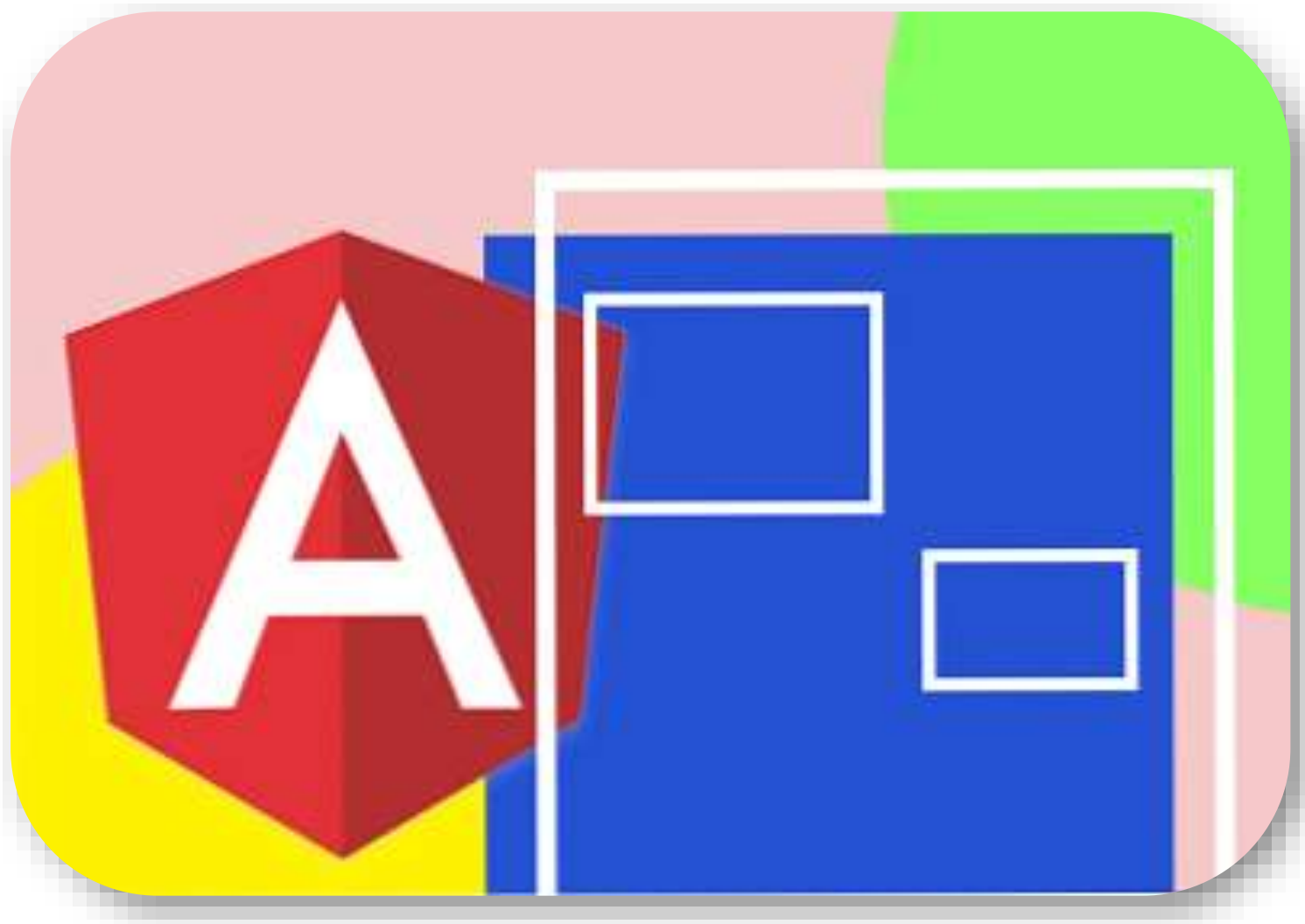
- » Conocer el funcionamiento del enrutamiento en una aplicación creada en Ionic y Angular
- » Identificar y crear rutas simples
- » Manejo de redireccionamiento
- » Conocer cómo navegar entre rutas
- » Cargas diferidas de ruta
- » Y navegación mediante Tabs



¿Que es el
enrutador?

Enrutador

- » El enrutador es una de las bibliotecas más importantes de una aplicación Angular ya que se pueden crear aplicaciones ricas en vinculaciones de Page y animaciones y no generar una aplicación de vista única o contexto único





Enrutadores

Enrutador Simple

» El enrutador más simple que se puede implementar es:

```
import { RouterModule } from '@angular/router';

@NgModule({
  imports: [
    ...
    RouterModule.forRoot([
      { path: '', component: LoginComponent },
      { path: 'detail', component: DetailComponent },
    ])
  ],
})
```

Ruta del componente

Cargamos el componente

Simple no? ahora, ¿Y si quisiéramos cargar una ruta diferente en nuestra carga inicial?

Enrutador Simple

» El enrutador más simple que se puede implementar es:

```
[  
  { path: '', redirectTo: 'login', pathMatch: 'full' },  
  { path: 'login', component: LoginComponent },  
  { path: 'detail', component: DetailComponent },  
]
```

Se busca la ruta
índice de la
aplicación

Se redirige al path login

Le indicamos que tiene
que comprar la ruta
completa

¿Cómo Navegar?

- » Desde HTML navegamos mediante [routerLink]

```
<ion-button [routerLink]="['/detail']">  
  Go to detail  
</ion-button>
```

El routerLink debe estar en el componente donde se encuentra el enrutamiento

¿Cómo Navegar?

- » Desde la programación navegamos mediante un API Enrutador

```
import { Component } from '@angular/core';  
import { Router } from '@angular/router';  
@Component({  
  ...  
})  
export class LoginComponent {  
  constructor(private router: Router) {}  
  navigate() {  
    this.router.navigate(['/detail'])  
  }  
}
```


Lazy loading routes

- » Rutas de carga diferidas es lo más óptimo para el enrutamiento de nuestra aplicación para evitar colocar todas las rutas en el módulo raíz (app) `TS app-routing.module.ts`
- » Lo ideal es aislar los componentes con sus propias rutas o fragmentos.

Lazy loading routes

- » En el módulo raíz (app) se indican los Page con sus módulos mediante el loadChildren una vez solicitado, cargando los enrutadores de esos Pages de forma progresiva

```
import { RouterModule } from '@angular/router';

@NgModule({
  imports: [
    ...
    RouterModule.forRoot([
      { path: '', redirectTo: 'login', pathMatch: 'full' },
      { path: 'login', loadChildren: './login/login.module#LoginModule' },
      { path: 'detail', loadChildren: './detail/detail.module#DetailModule' }
    ])
  ],
})
```

Trabajando con Tabs

- » Cuando trabajamos con Tabs, Angular le proporciona a Ionic un mecanismo para saber qué componentes deben cargar.
- » El componente o Page Tabs es que se lleva el trabajo del enrutamiento de los Tabs

Trabajando con Tabs

```
const routes: Routes = [
  {
    path: 'tabs',
    component: TabsPage,
    children: [
      {
        path: 'tab1',
        children: [
          {
            path: '',
            loadChildren: '../tab1/tab1.module#Tab1PageModule'
          }
        ]
      },
      {
        path: '',
        redirectTo: '/tabs/tab1',
        pathMatch: 'full'
      }
    ]
  },
  {
    path: '',
    redirectTo: '/tabs/tab1',
    pathMatch: 'full'
  }
];
```




Datos entre Pages

Usando Extras State(Envío)

» ¿Que pasa si queremos enviar información de un Page a otro?

Usamos NavigationExtras

```
import { Router, NavigationExtras } from '@angular/router';
```

```
let navigationExtras: NavigationExtras = {  
  state: {user: this.user}  
};  
this.router.navigate(['/home'], navigationExtras);
```

Usando Extras State(Recepcionista)

» ¿Y si queremos recibir?

```
import { Router, NavigationExtras } from '@angular/router';
...
constructor(private activeroute: ActivatedRoute, private router: Router)
{
  this.activeroute.queryParams.subscribe(params => {
    if (this.router.getCurrentNavigation().extras.state) {
      this.data = this.router.getCurrentNavigation().extras.state.user;
      console.log(this.data)
    }
  });
}
```


Usando Extras State (Consideraciones)

» Consideraciones

```
constructor(private router: Router) { } // Se debe instanciar  
...
```

```
constructor(private activeroute: ActivatedRoute) { } // Se debe instanciar  
...
```



En resumen...

Resumen

Lo que se espera que aprendas en esta Actividad es:

- » **Vimos lo que es enrutamiento en Angular**
- » **Identificar y crear rutas simples**
- » **Manejo de redireccionamiento**
- » **Navegar entre rutas**
- » **Cargas diferidas de ruta**
- » **Y navegación mediante Tabs**



Documentación Oficial

- » Para esta asignatura se les recomienda consultar la documentación oficial
- » <https://ionicframework.com/docs>