



PROGRAMACIÓN DE APLICACIONES MÓVILES

PGY4121

Profesor:

DuocUC



ESCUELA DE
INFORMÁTICA Y
TELECOMUNICACIONES





Actividad N°2.1: Navegando de forma Optimizada

Route y Navegación

Objetivos

Lo que se espera que aprendas en esta Actividad es:

- » Manejo de redireccionamiento
- » Conocer cómo navegar entre rutas
- » Cargas diferidas de ruta
- » Conocer Route Guards
- » Qué hacer con un 404
- » Conocer los Children en las Rutas

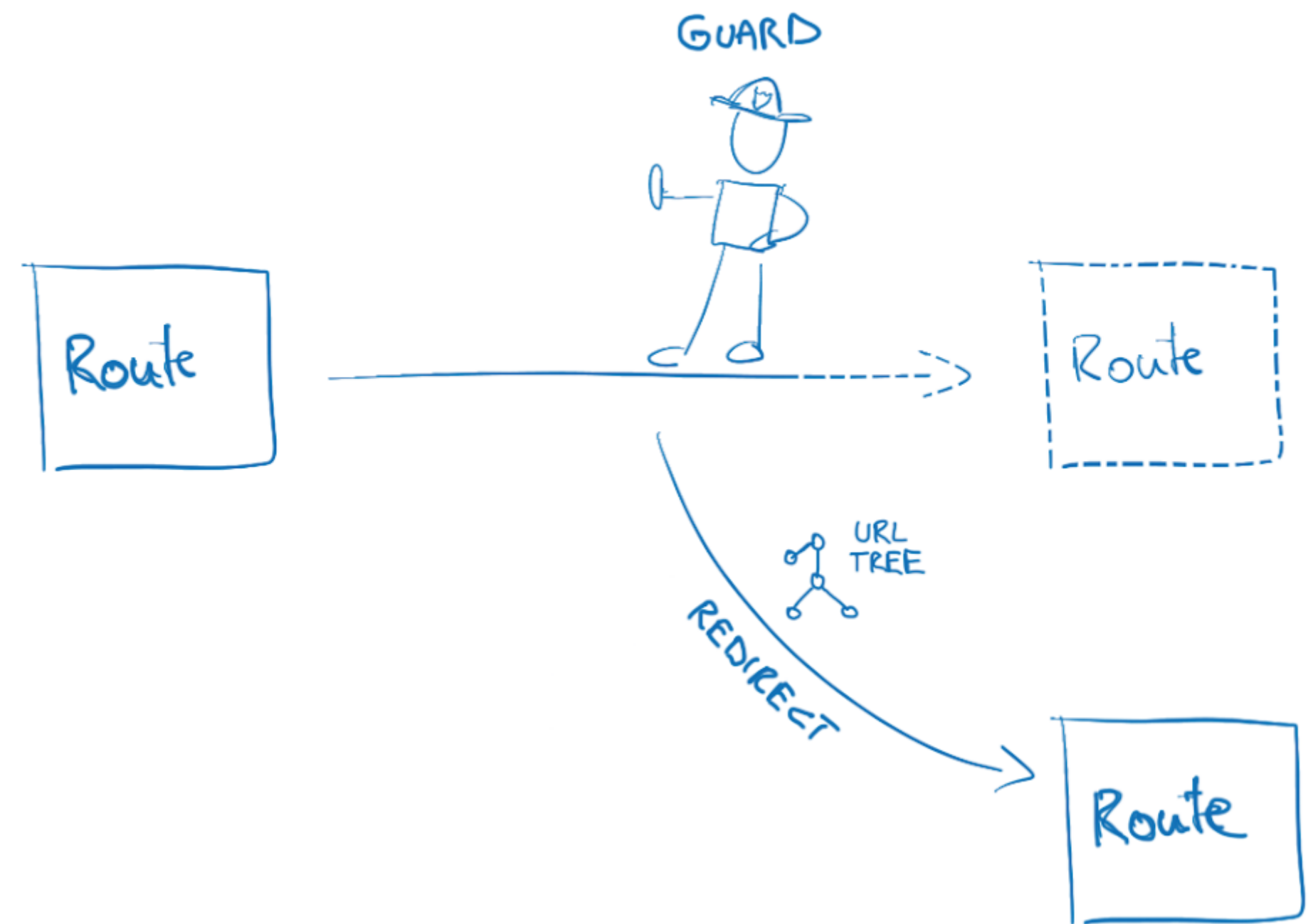


¿Qué es Route
Gards?

PGY4121 | Route Guards

- » Guards nos ayuda a determinar si se puede realizar una acción determinada contra una ruta.
- » Se implementan en una interfaz predeterminada (Page o Component)

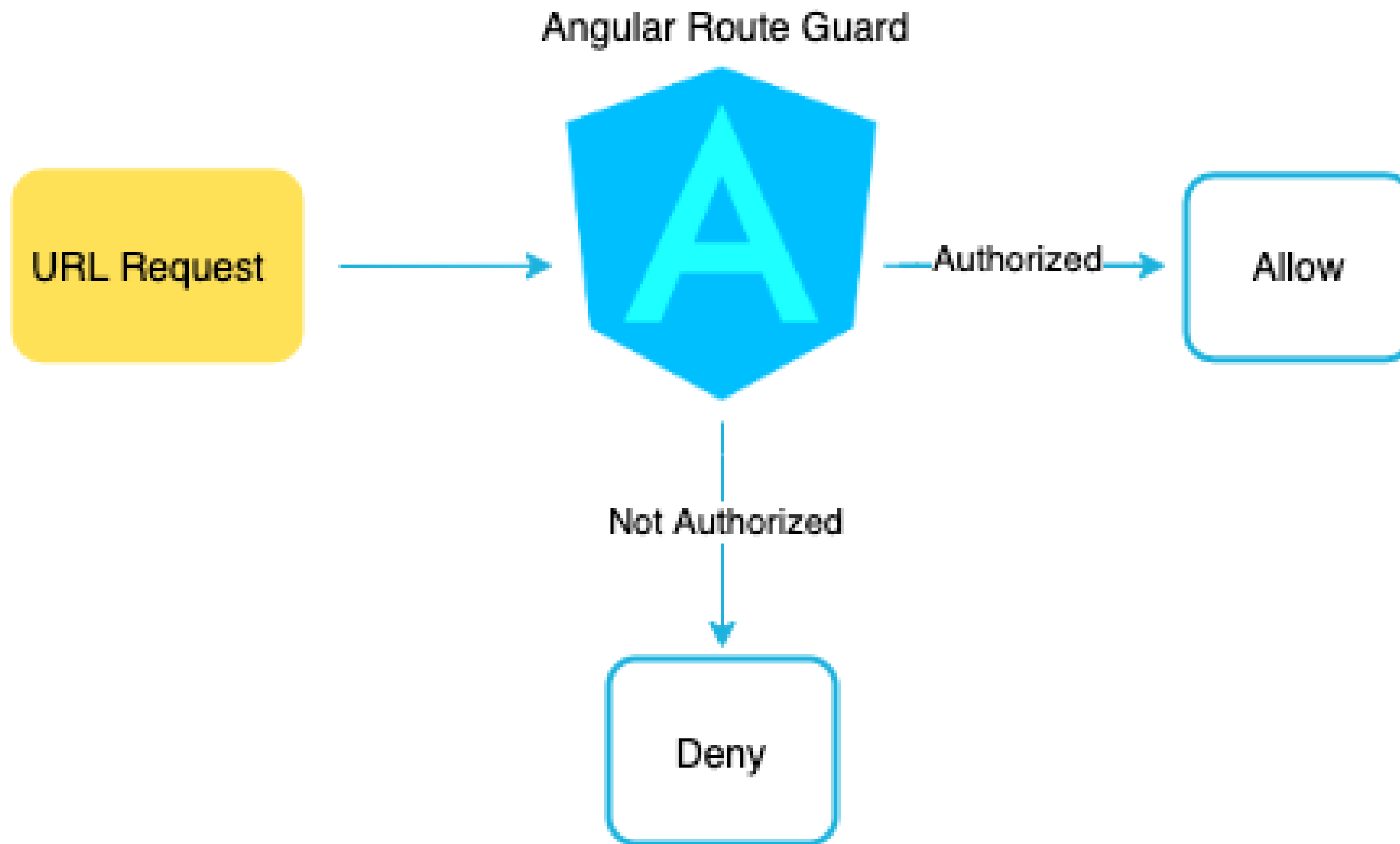
*Tips: con un Guards correcto puedo implementar “mantener sesión iniciada” (algo ideal en aplicaciones móviles)





Implementación Route Gards

PGY4121 | Route Guards



PGY4121 | Route Guards

» Extracto de un Guard

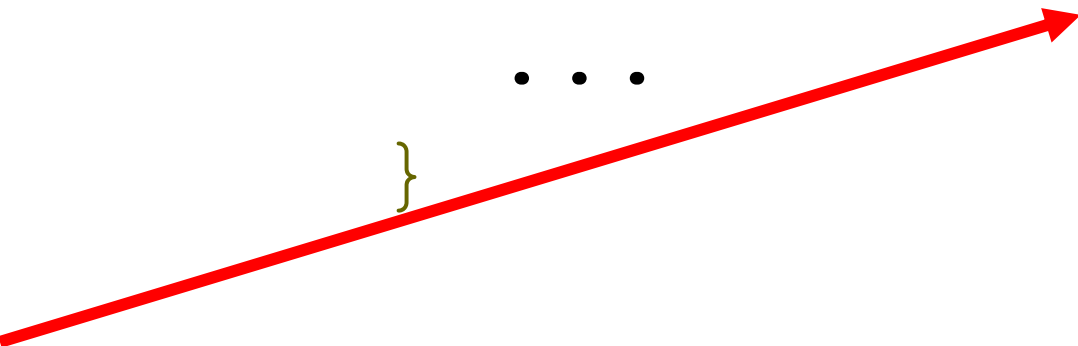
```
export class AuthGardService implements CanActivate {  
  constructor(  
    public authenticationService: AuthenticationService,  
    public router: Router  
  ) { }  
  canActivate(): boolean {  
    return this.authenticationService.isAuthenticated();  
  }  
}
```


PGY4121 | Routing Module

» Extracto de una ruta en Routing Module

```
{  
  path: '/your-path',  
  component: YourComponent,  
  canActivate: [YourGuard],  
}
```

`export class YourGuard
implements CanActivate {
 ...
}`



*Haga click en `CanActivate` [y ve al enlace de la documentación Angular]



Page 404

» Extracto de una ruta en Routing Module

```
const appRoutes: Routes = [  
  { path: 'heroes', component: HeroListComponent },  
  { path: '', redirectTo: '/heroes', pathMatch: 'full' },  
  { path: '**', component: PageNotFoundComponent }  
];
```

Nos permitirá redirigir a un Page si la ruta no coincide con las establecidas



Redireccionamientos

PGY4121 | Redireccionamiento

» Utilizamos Redireccionamiento

Se busca la ruta indice de la aplicación

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

Se redirige al path login

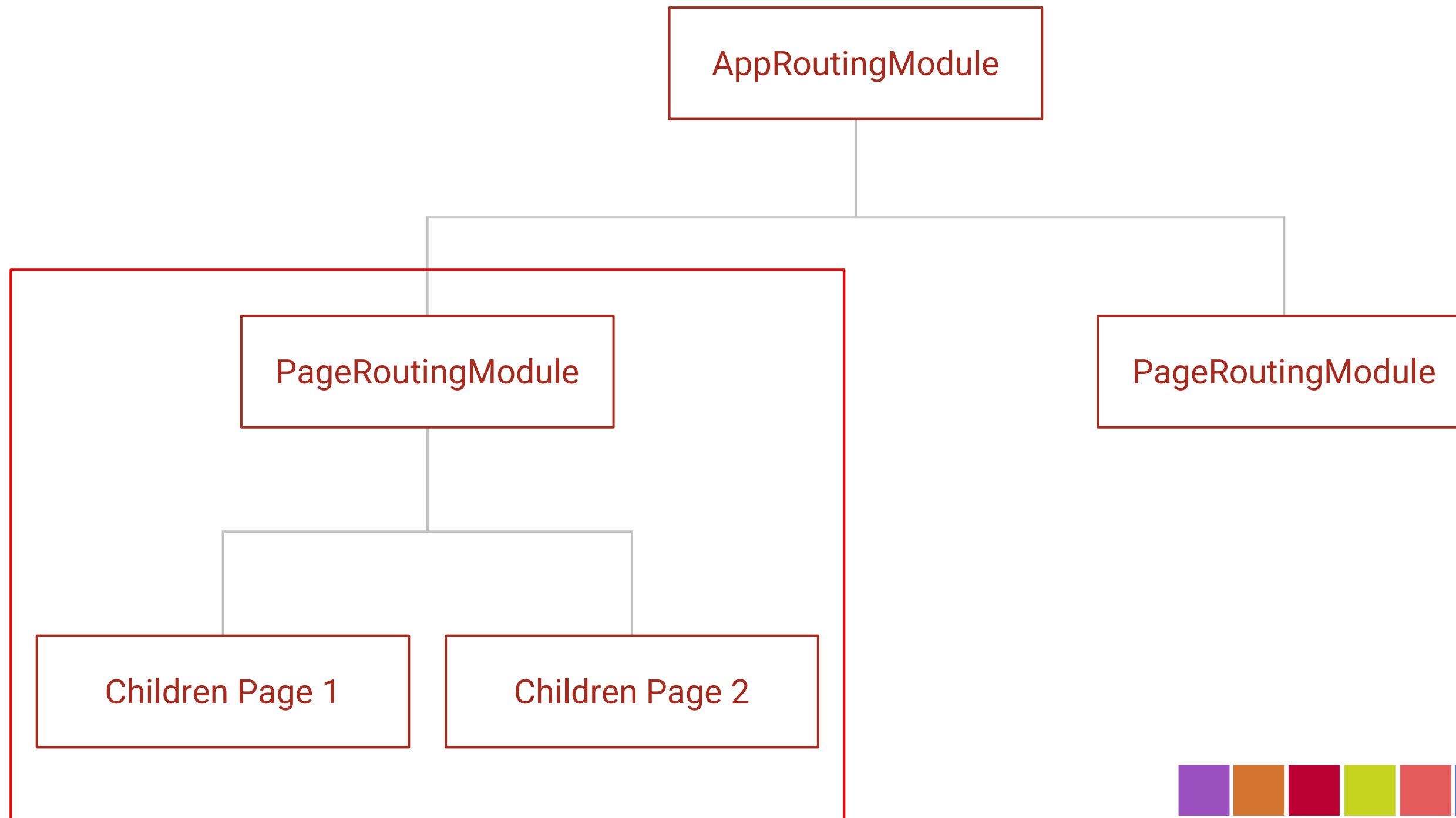
Le indicamos que tiene que comprar la ruta completa



Children

PGY4121 | Children

- » Cuando queremos trabajar con múltiples visuales manteniendo el mismo Page, podemos recurrir a children de una ruta y a ion-route-outlet



PGY4121 | Children

```
const routes: Routes = [  
  {  
    path: '',  
    component: HomePage,  
    children: [  
      {  
        path: 'perfil',  
        component: MiPerfilComponent  
      }  
    ]  
  },  
];
```


PGY4121 | Ion-router-outlet

```
<ion-content>  
  <!-- Aquí se desplegara todos los Component que se declararon como  
children en  
  HomePageModule -->  
  <ion-router-outlet></ion-router-outlet>  
</ion-content>
```




Lazy Loading Module

PGY4121 | Lazy loading routes

- » Rutas de carga diferidas es lo mas optimo 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

PGY4121 | 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' }
    ])
  ],
})
```





Profundización

- » En el recurso “2.1.2 In-app navigation: routing to views” podrán indagar sobre la ruta en Angular, por lo tanto muchos de los comandos no son aplicables, recuerden procurar usar comandos Ionic
- » En el recurso “2.1.3 Router Reference” Encontrarán todos los conceptos claves



Resumen

- » Redireccionar
- » Navegar entre rutas
- » Cargas diferidas de ruta (Lazy Loading Route)
- » Conocer Route Guards
- » Qué hacer con un 404
- » Children en las rutas