

Sesión 02

Enrutamiento en angular

Instructor:

ERICK ARÓSTEGUI

earostegui@galaxy.edu.pe



17 ANGULAR

FULL-STACK
DEVELOPER

ÍNDICE

01 Introducción a enrutamiento

02 Enrutamiento con parámetros

03 Precarga de datos mediante Route Resolvers

04 Rutas hijas

05 Rutas secundarias

01

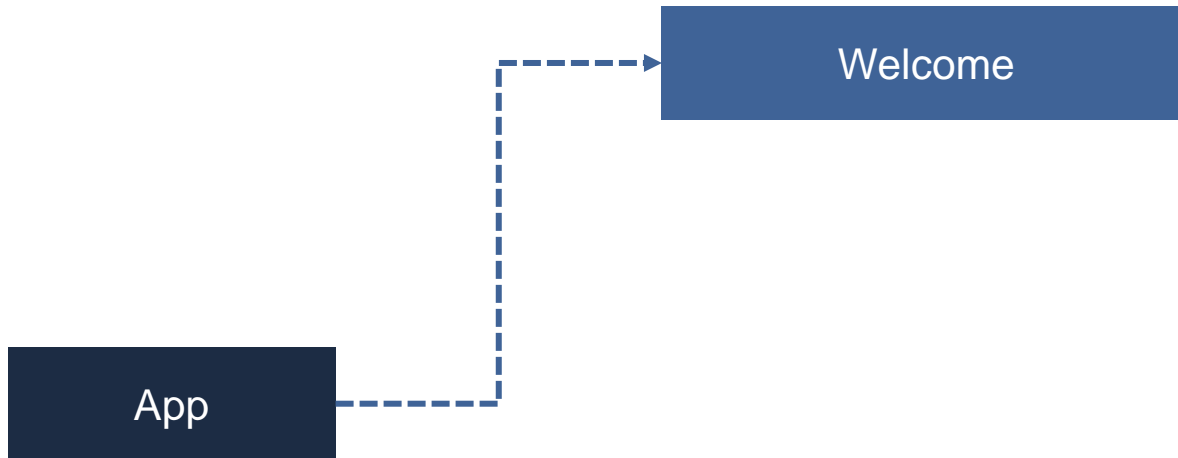


Introducción a enrutamiento

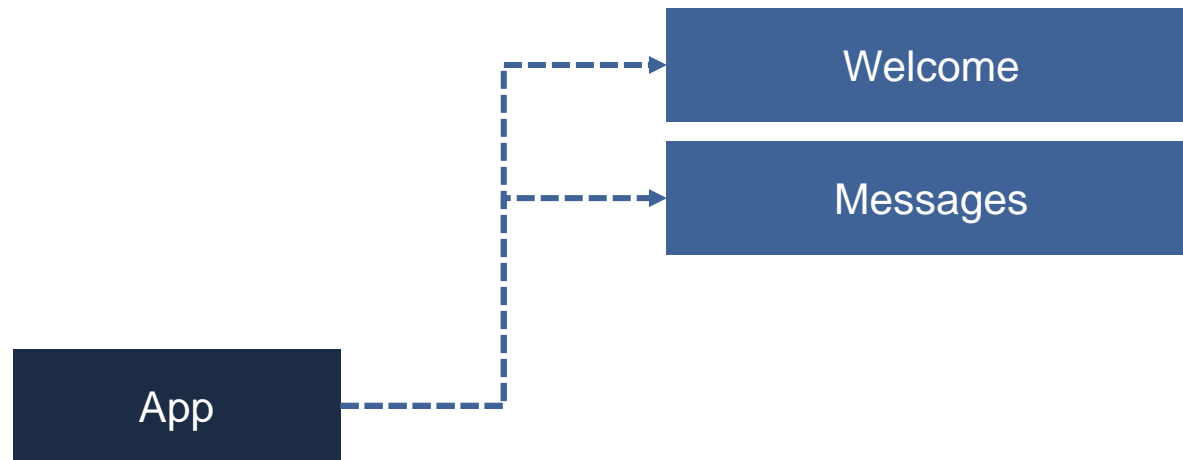
Componentes

App

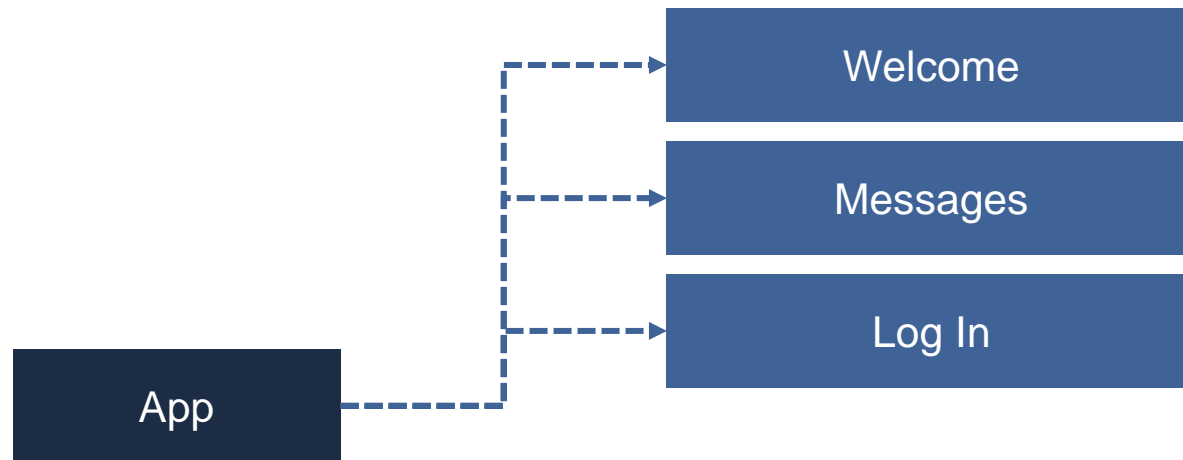
Componentes



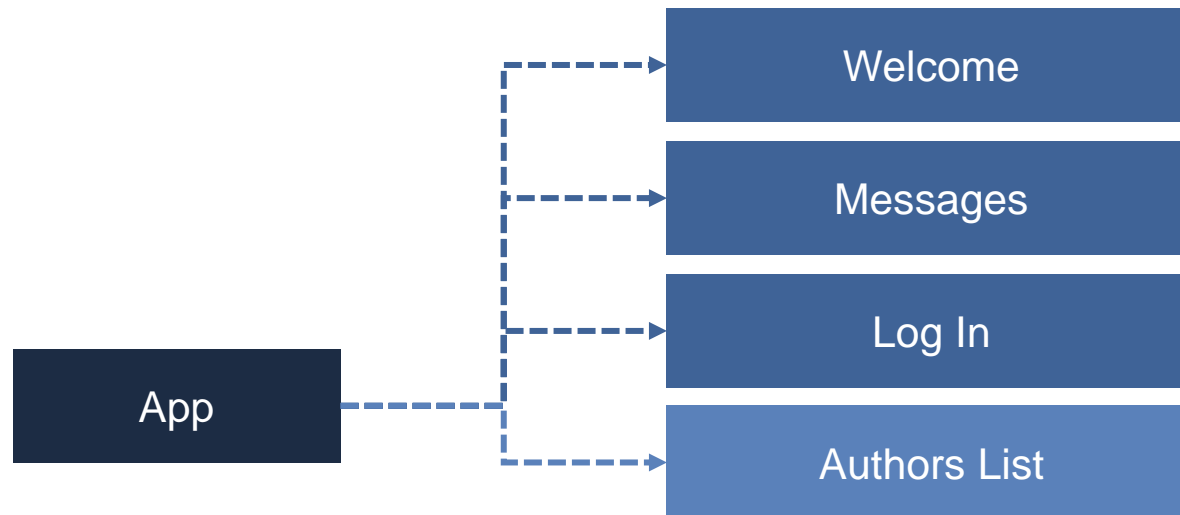
Componentes



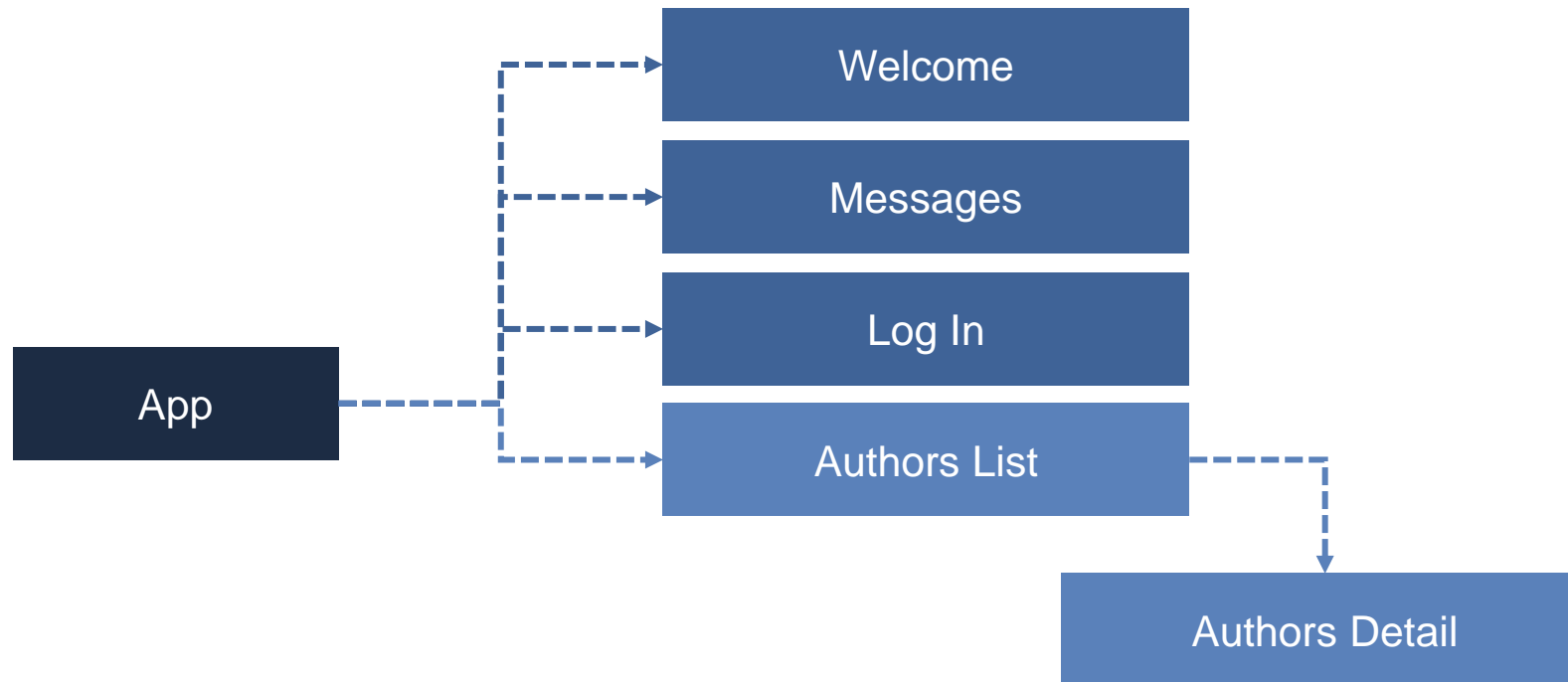
Componentes



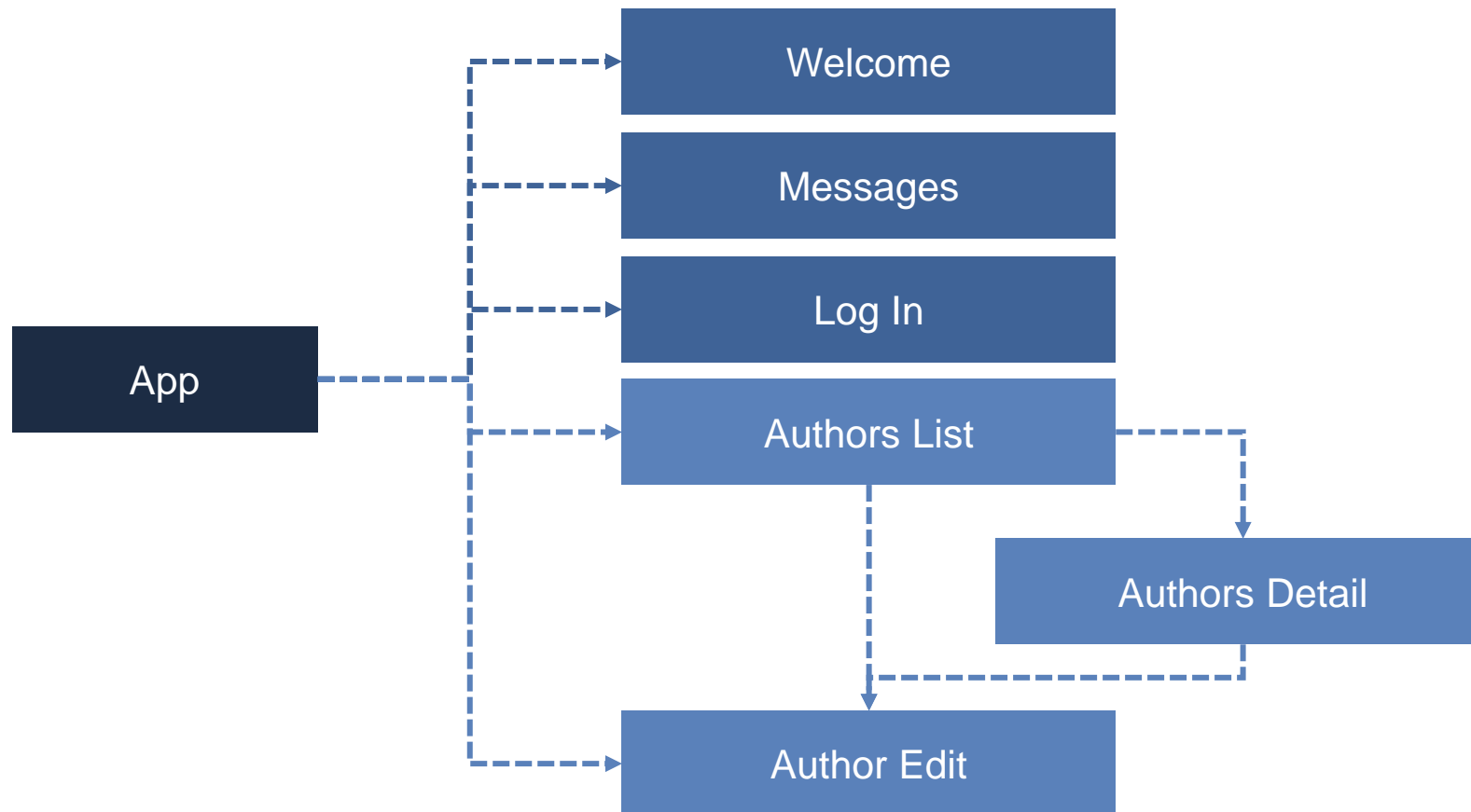
Componentes



Componentes



Componentes



Componentes



Componentes



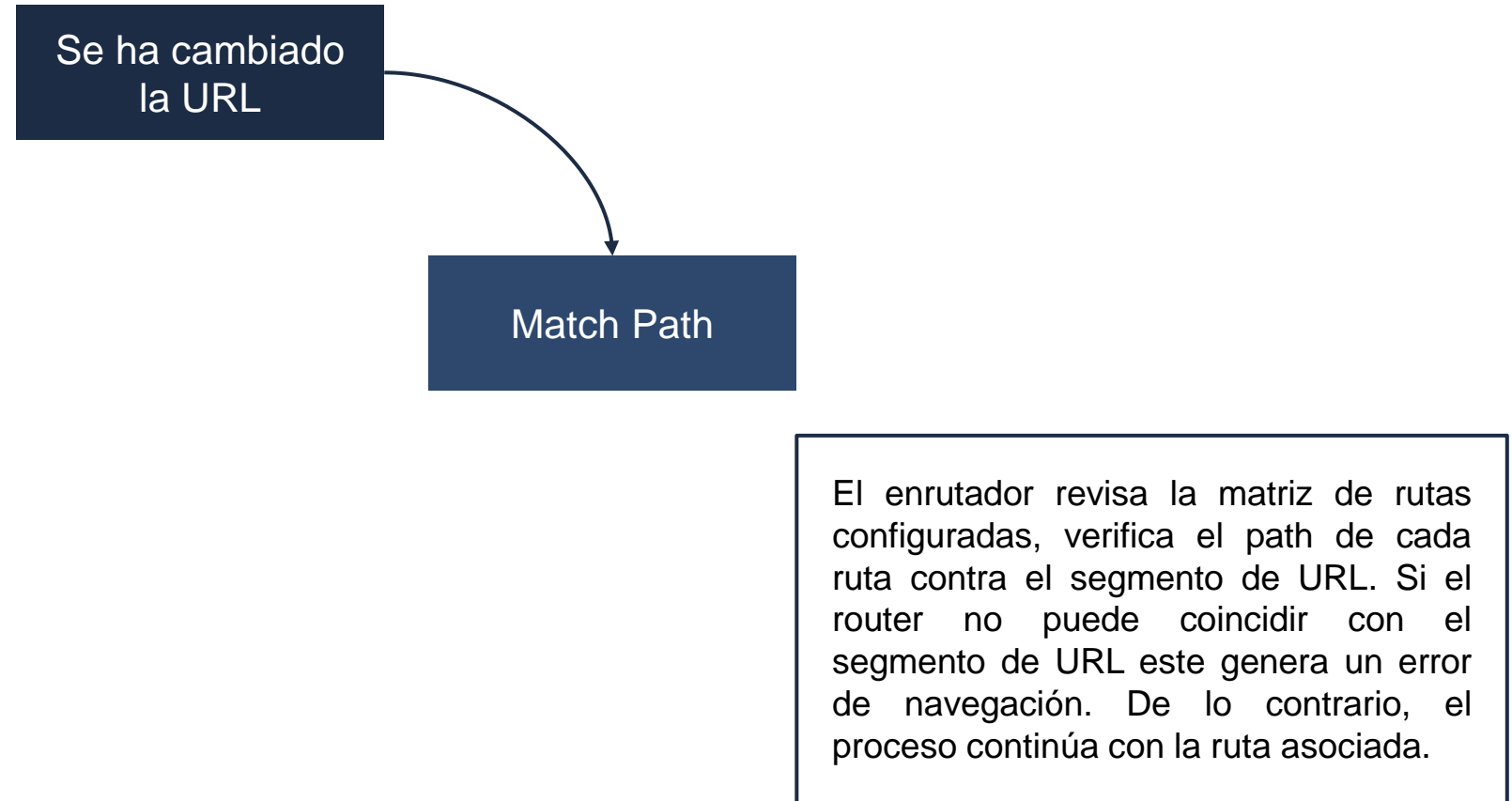
Ciclo de ejecución

Se ha cambiado
la URL

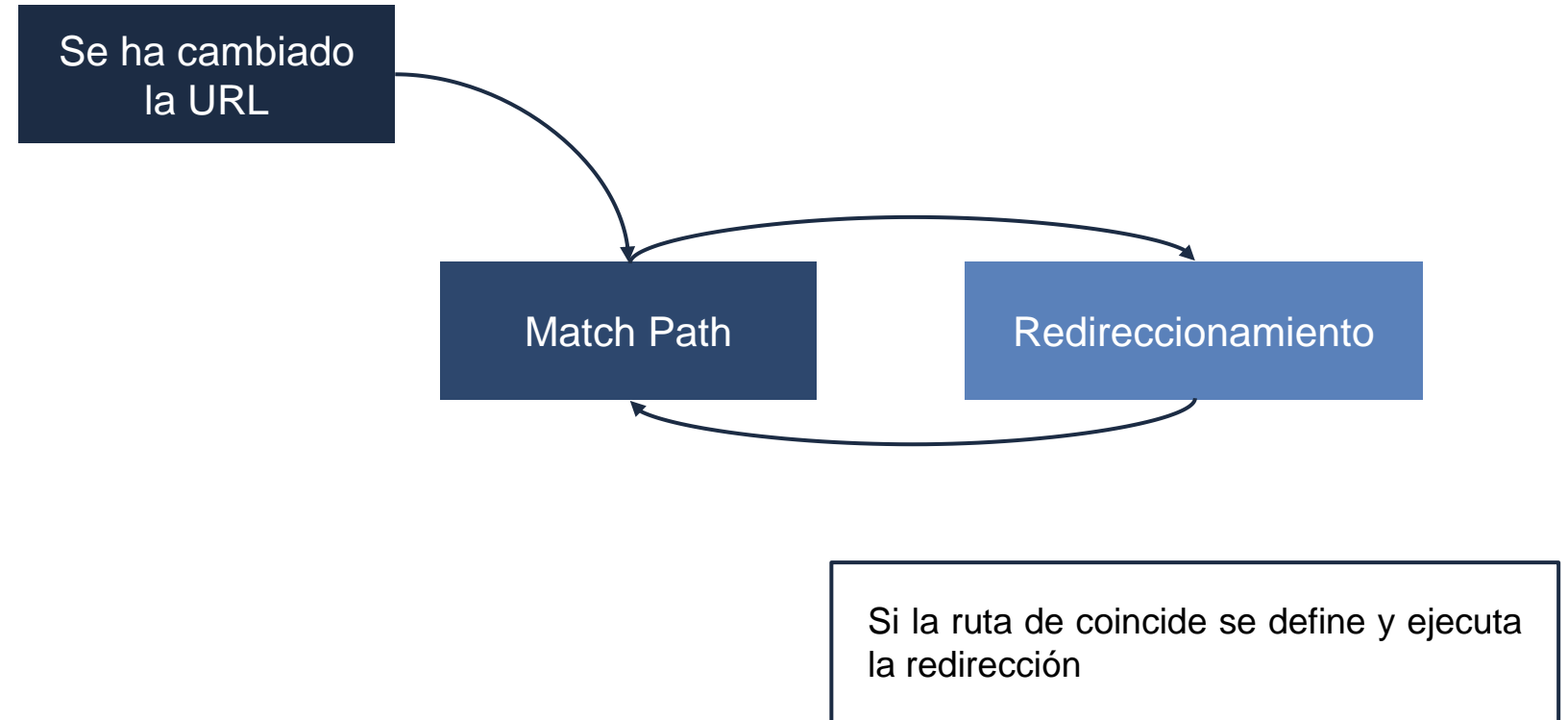
La URL cambia

- Cuando el usuario hace clic en un elemento que incluye la directiva routerLink
- Cuando el usuario cambia directamente la barra de direcciones
- Cuando navegamos por el código.

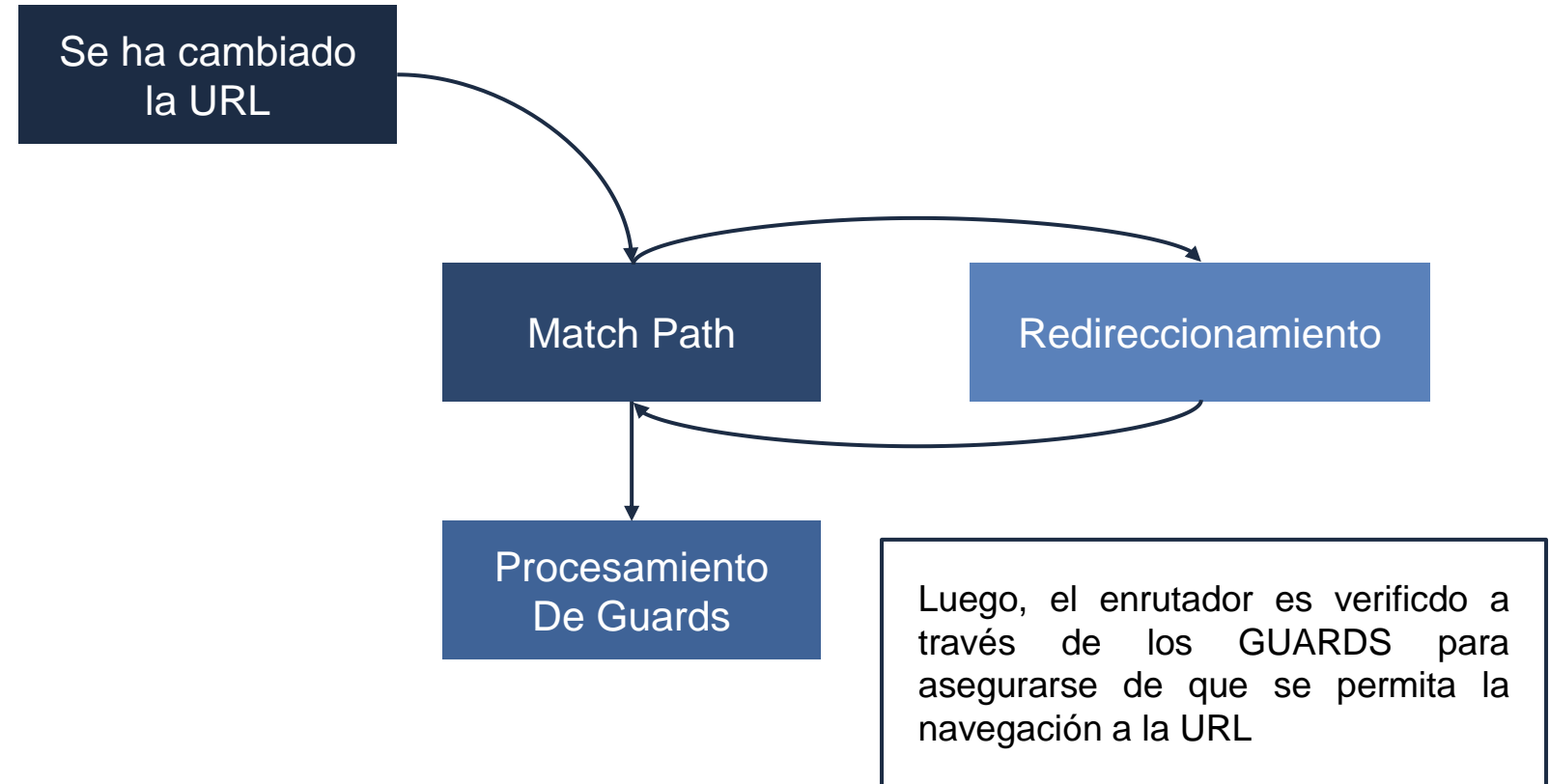
Ciclo de ejecución



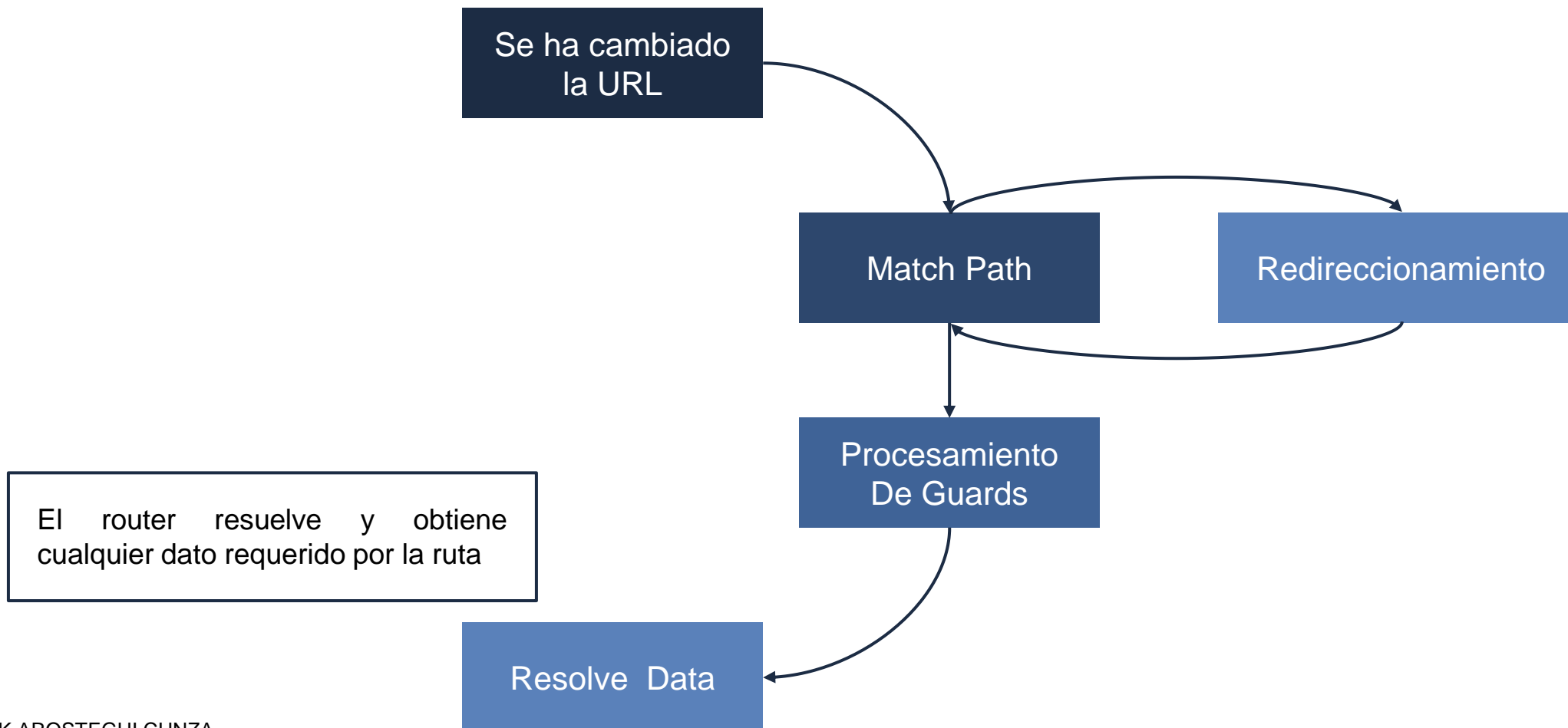
Ciclo de ejecución



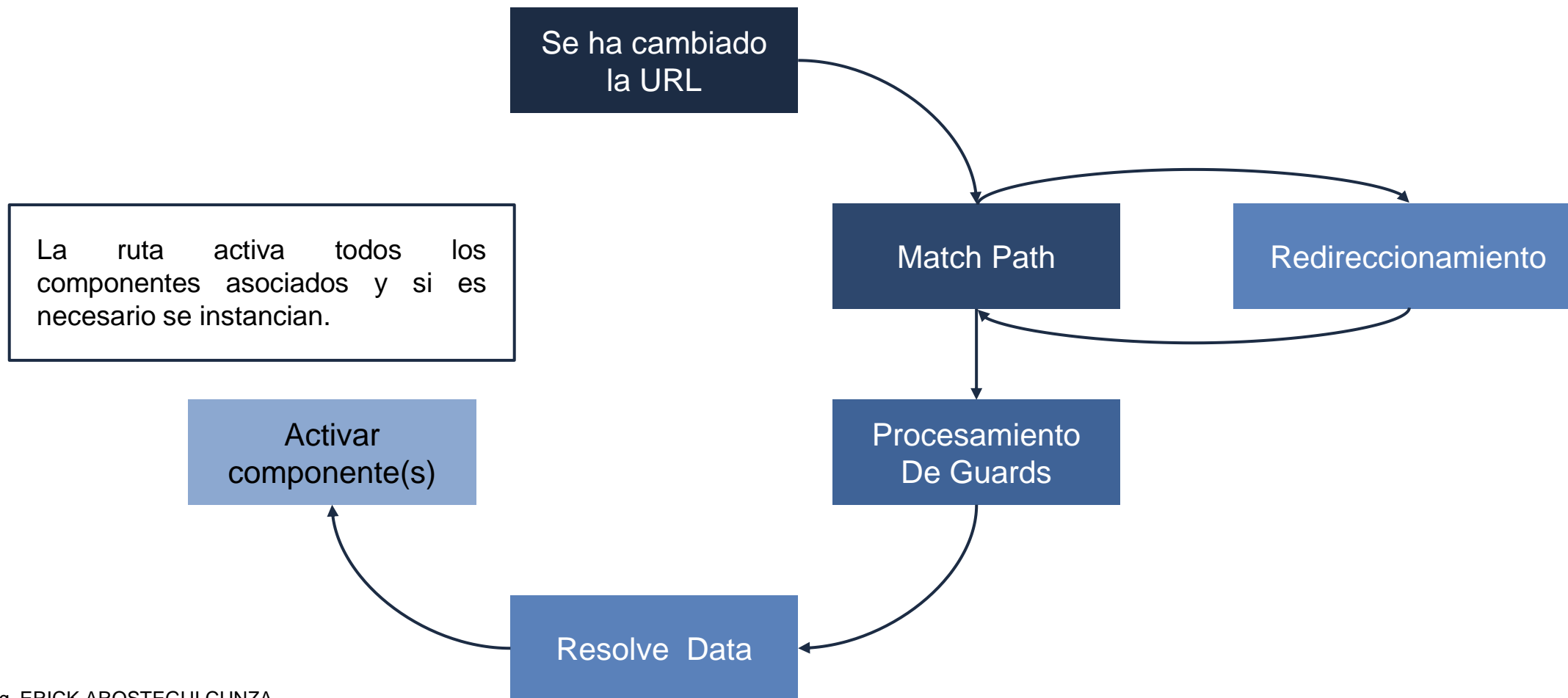
Ciclo de ejecución



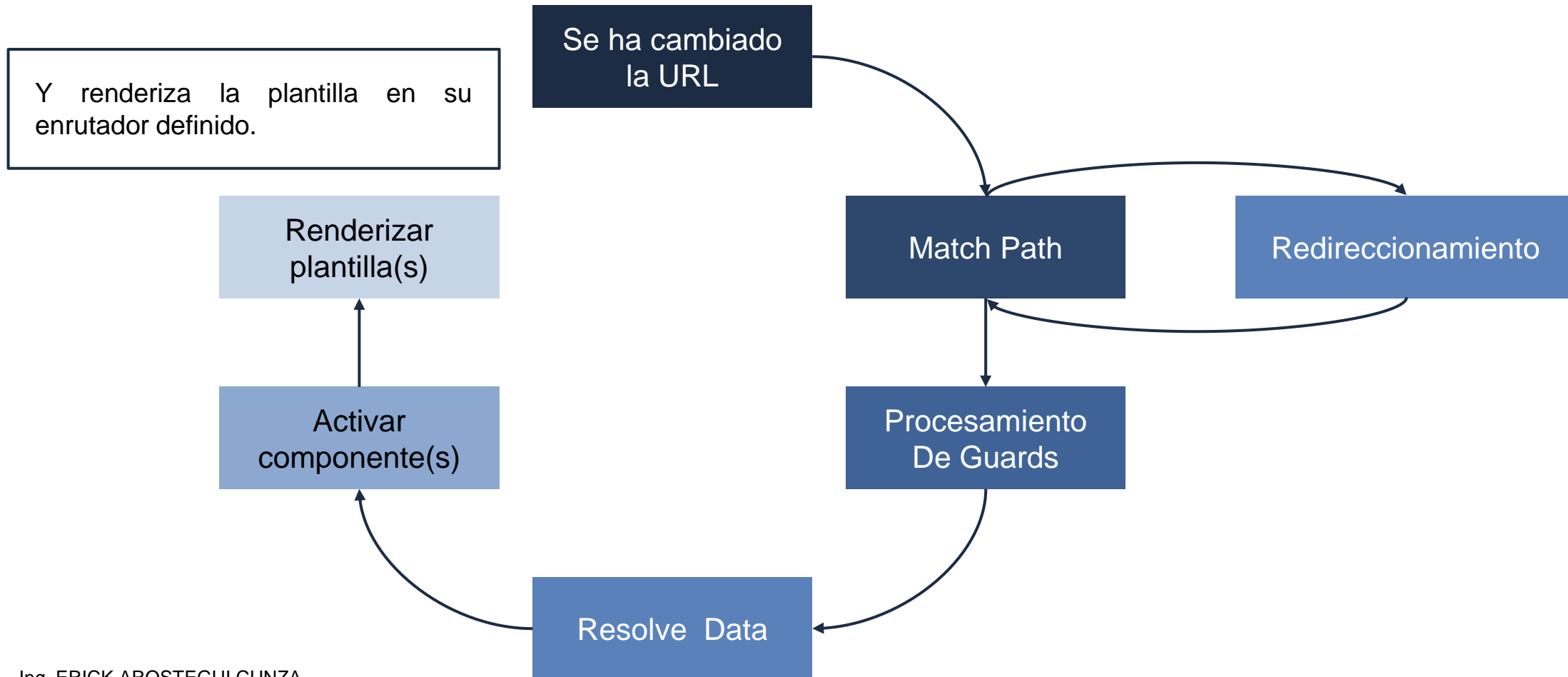
Ciclo de ejecución



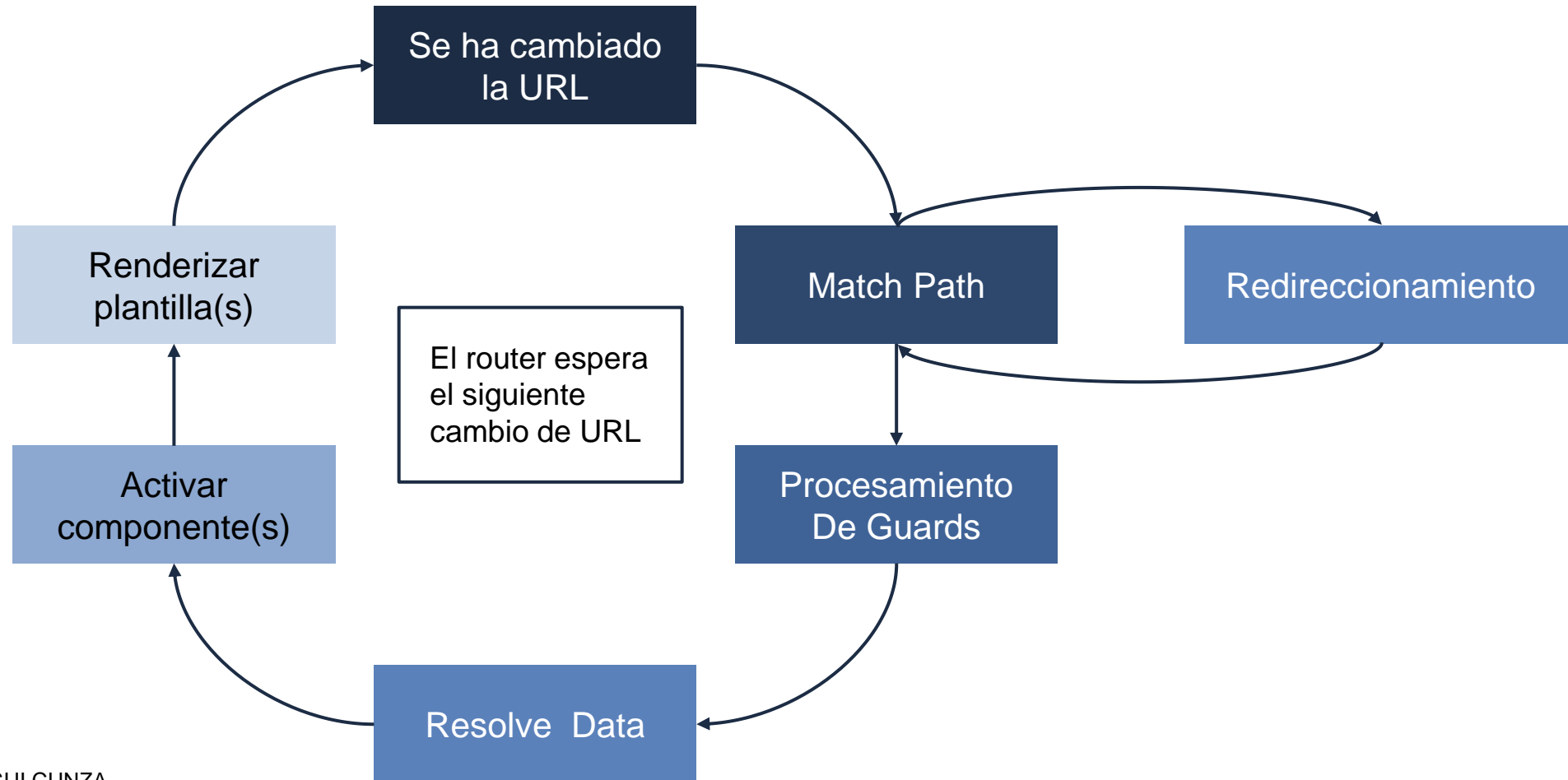
Ciclo de ejecución



Ciclo de ejecución



Ciclo de ejecución



Definición de la ruta base



- Utilizado por el router
 - Creación de la dirección URL para la navegación
- Utilizado por el navegador
 - Creación de la URL para descargar y vincular

Establer ruta base

`<base href="/">`

(localhost:4200)



`<base href="/library/">`

[\(www.myapp.com/library/\)](http://www.myapp.com/library/)

Establecer manualmente la ruta base

Usar la CLI angular

`ng build --base-href /library/`

Importación del enrutador angular

RouterModule.forRoot()

Declara las directivas del enrutador
Gestiona nuestra configuración de ruta
Registra el servicio de enrutador
Se usa una vez para la aplicación

RouterModule.forChild()

Declara las directivas del enrutador
Gestiona nuestra configuración de ruta
NO registra el servicio del router
Utilizado en módulos de feature

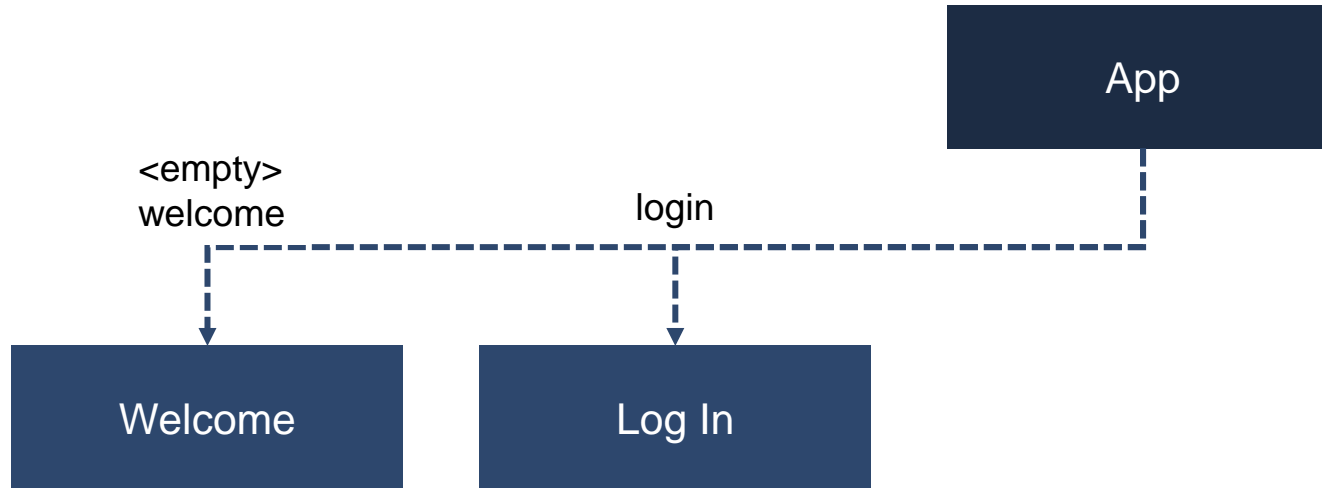
→ Introducción a enrutamiento

App

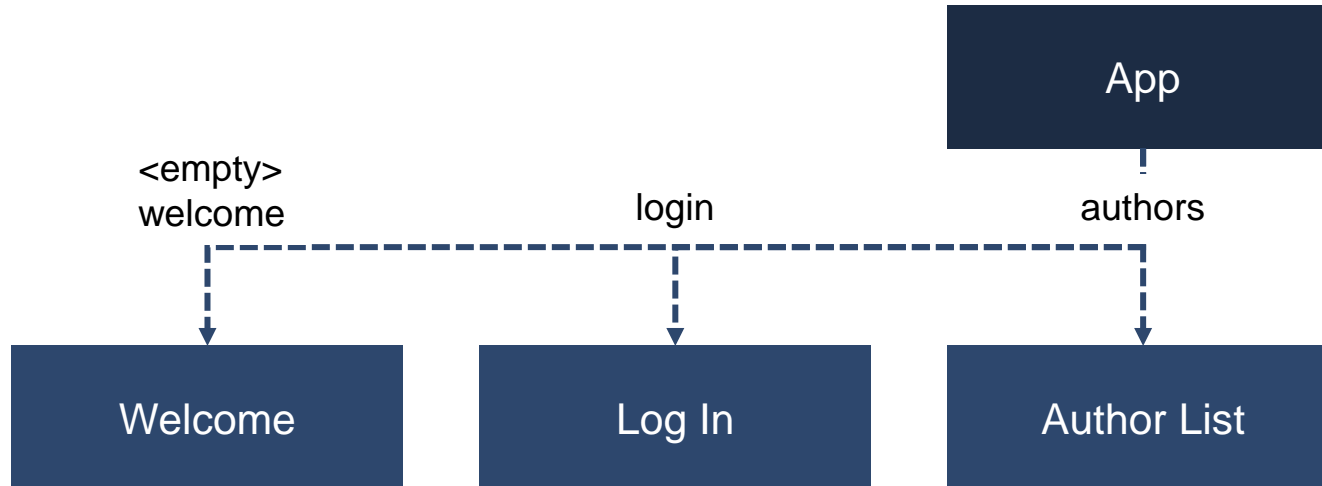
→ Introducción a enrutamiento



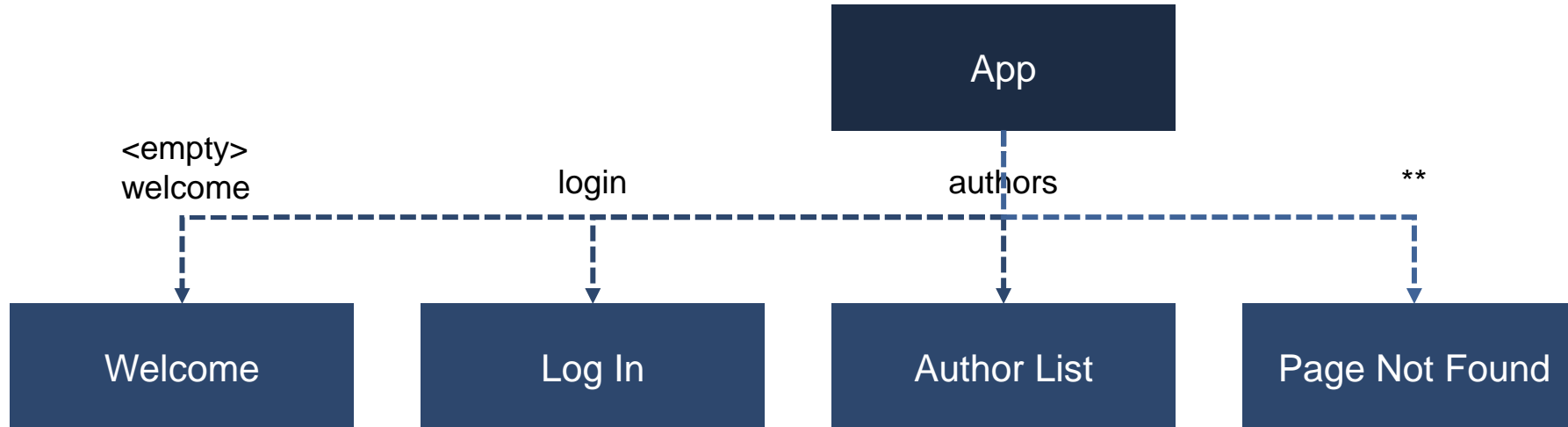
→ Introducción a enrutamiento



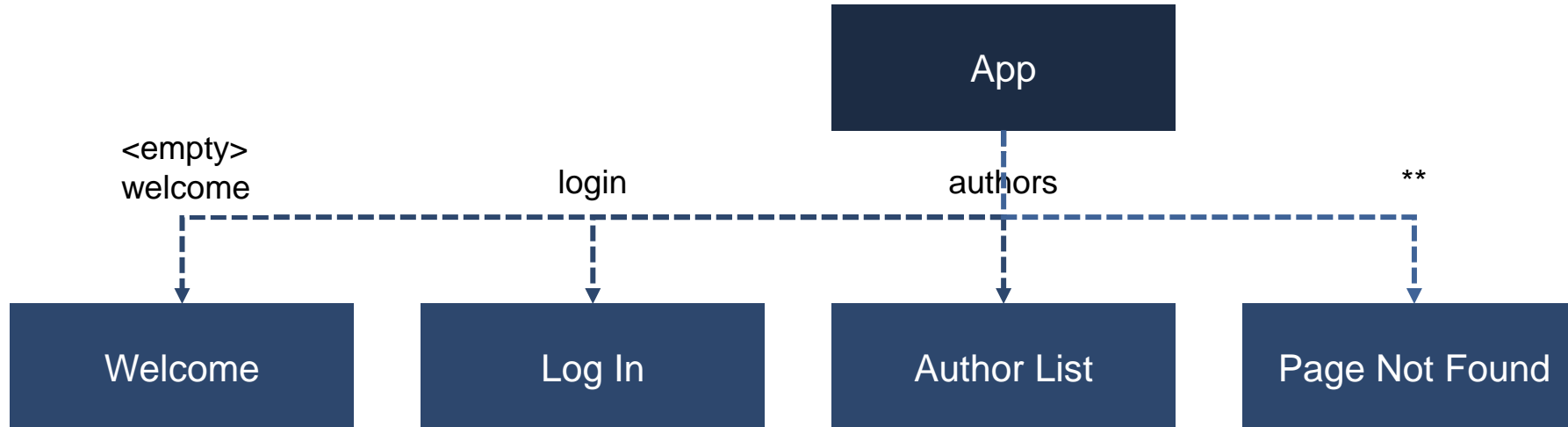
→ Introducción a enrutamiento



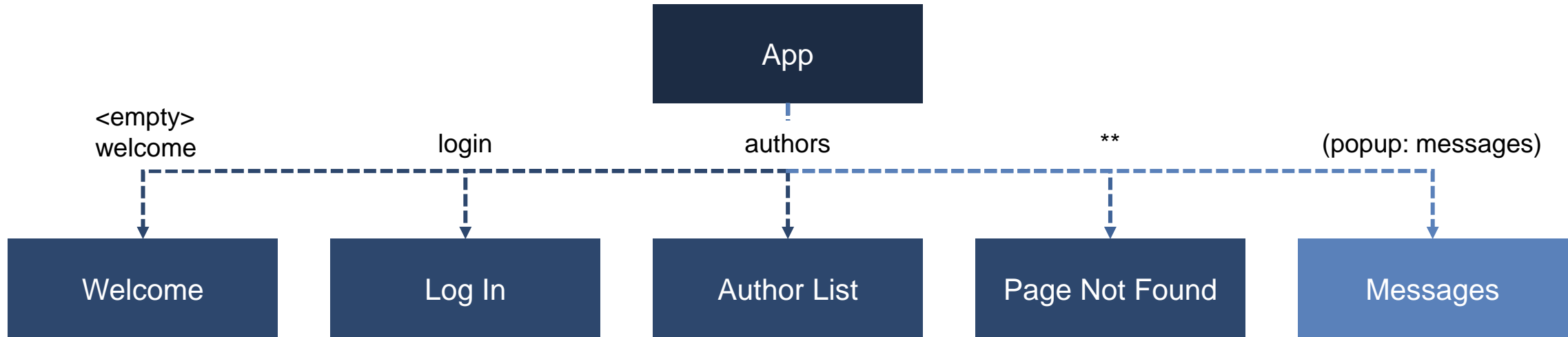
→ Introducción a enrutamiento



→ Introducción a enrutamiento



Introducción a enrutamiento



Configuración de routes

```
[  
  { path: 'home', component: WelcomeComponent },  
  { path: 'welcome', redirectTo: 'home', pathMatch: 'full' },  
  { path: '', redirectTo: 'home', pathMatch: 'full' },  
  { path: '**', component: PageNotFoundComponent }  
]
```

Configuración de routes

```
[  
  { path: 'home', component: WelcomeComponent },  
  { path: 'welcome', redirectTo: 'home', pathMatch: 'full' },  
  { path: '', redirectTo: 'welcome', pathMatch: 'full' },  
  { path: '**', component: PageNotFoundComponent }  
]
```


Configuración de routes

```
[  
  { path: 'home', component: WelcomeComponent },  
  { path: '', redirectTo: 'welcome', pathMatch: 'full' },  
  { path: '**', component: PageNotFoundComponent }  
]
```

Configuración de routes

```
[  
  { path: 'welcome', component: WelcomeComponent },  
  { path: '', redirectTo: 'welcome', pathMatch: 'full' },  
  { path: '**', component: PageNotFoundComponent }  
]
```

Colocación de las plantillas

app.component.html

```
...  
  
<ul>  
  <li><a [routerLink]="['/welcome']">Home</a></li>  
  <li><a [routerLink]="['/authors']">Authors List</a></li>  
</ul>  
<router-outlet></router-outlet>
```

Activación de una ruta

app.component.html

```
...

<ul>
  <li><a [routerLink]="['/welcome']">Home</a></li>
  <li><a [routerLink]="['/authors']">Author
    List</a></li>
</ul>
<router-outlet></router-outlet>
```

```
...

<ul>
  <li><a routerLink="/welcome">Home</a></li>
  <li><a routerLink="/authors">Author List</a></li>
</ul>
<router-outlet></router-outlet>
```

Estilos de URL del navegador

Estilo HTML 5 (.../welcome)

Aprovecha el historial de HTML 5 pushState

Default

URL más natural

Requiere reescritura de URL

Hash-based (.../#/welcome)

Aprovecha los fragmentos de URL

Debe establecerse con **{ useHash: true }**

No requiere reescritura de URL

Uso de URL basadas en hash - modules

app.module.ts

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

@NgModule({
  imports: [
    BrowserModule,
    RouterModule.forRoot([...], { useHash: true })
  ],
  declarations: [
    ...
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

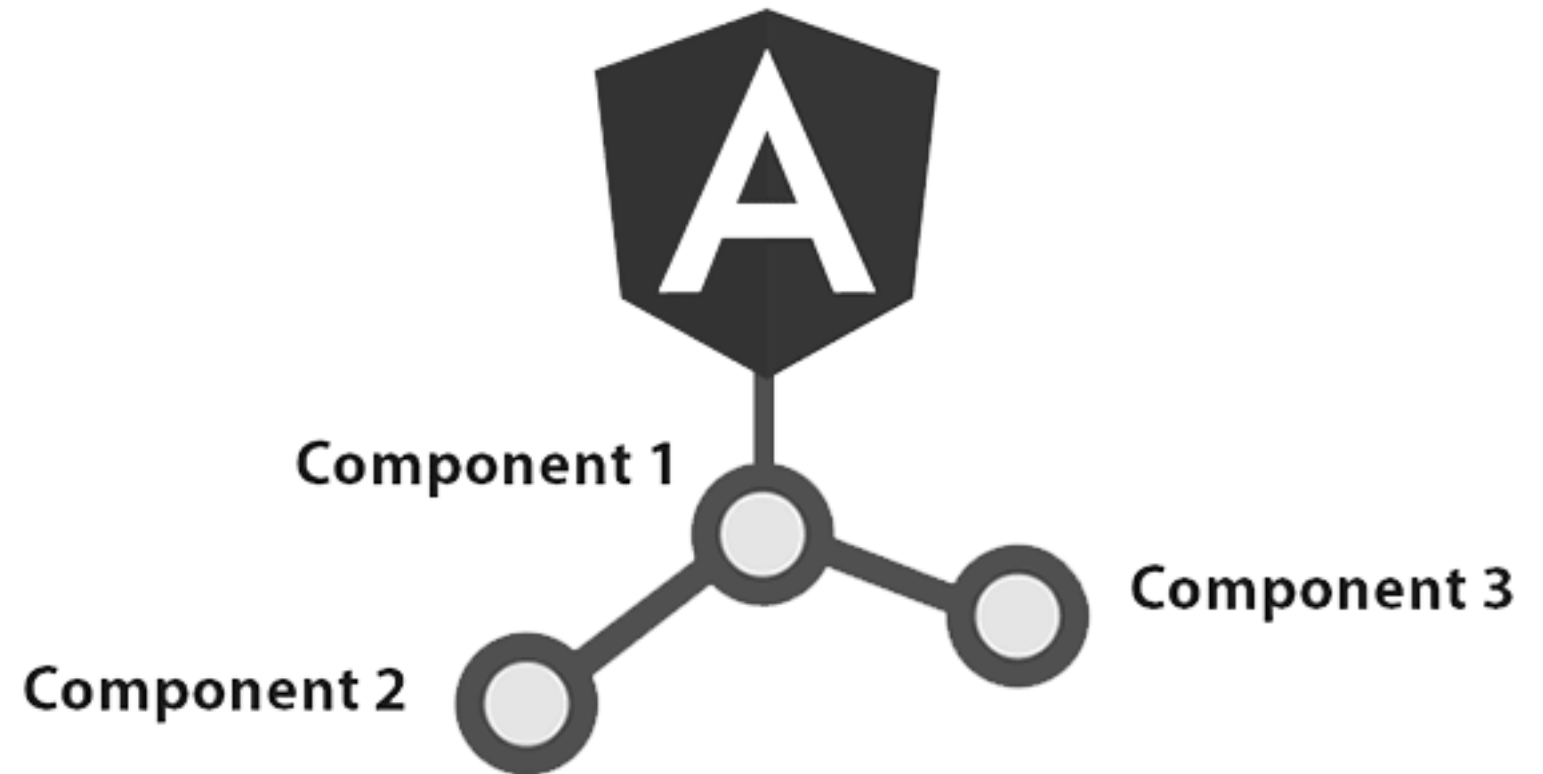
Uso de URL basadas en hash - standalone

app.config

```
import { provideRouter, withComponentInputBinding, withHashLocation } from '@angular/router';  
...  
  
export const appConfig: ApplicationConfig = {  
  providers: [  
    ...  
    provideRouter(APP_ROUTES, withHashLocation(), withComponentInputBinding()),  
    ...  
  ]  
};
```

Enrutamiento básico

DEMO

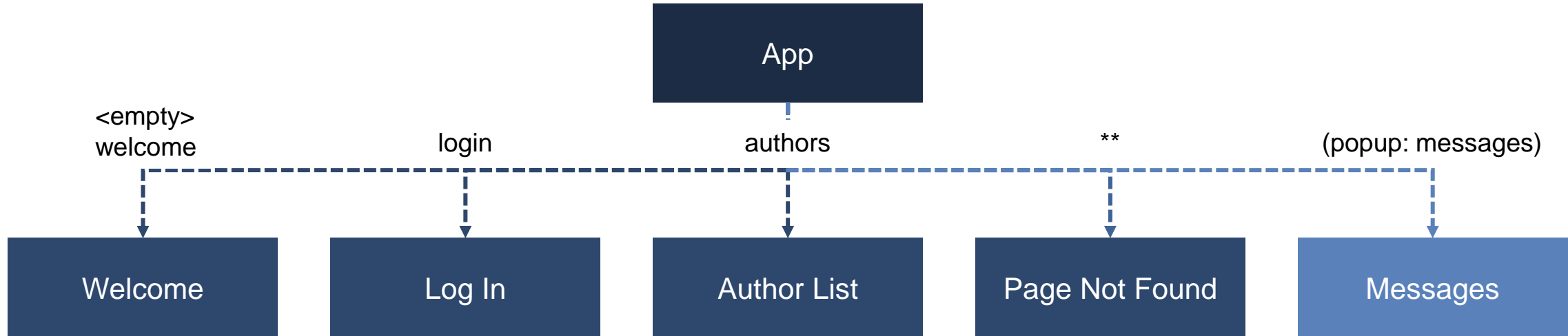


02

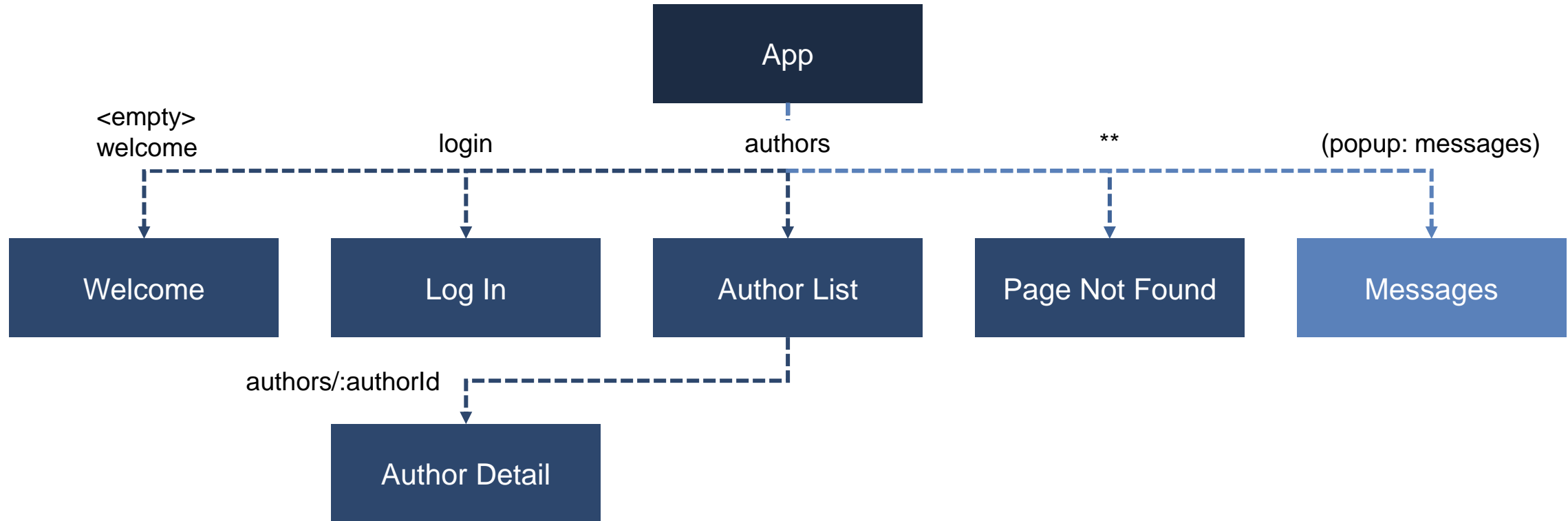


Enrutamiento con parámetros

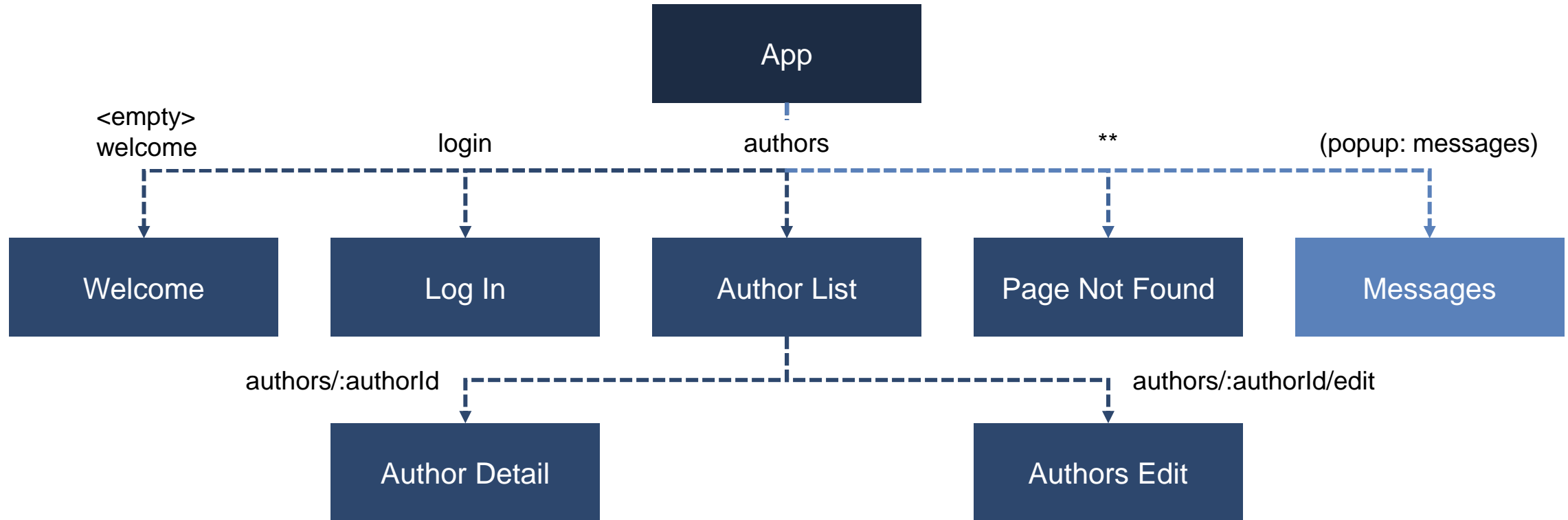
Enrutamiento con parámetros



Enrutamiento con parámetros



Enrutamiento con parámetros



Configuración de rutas con parámetros

Angular Routes

```
[  
  { path: 'authors', component: AuthorListPageComponent },  
  { path: ' authors/:authorId', component: AuthorDetailPageComponent },  
  { path: ' authors/:authorId/edit', component: AuthorEditPageComponent }  
]
```

Configuración de rutas con parámetros

Angular Routes

```
[  
  { path: 'books', component: BookListPageComponent },  
  { path: ' books/:bookId', component: BookDetailPageComponent },  
  { path: ' books/:bookId/edit', component: BookPageEditComponent }  
]
```

Rellenar parámetros de ruta

Template

```
<a [routerLink]="['/authors', author.authorId]">{{author.name}}</a>  
  
<a [routerLink]="['/authors', author.authorId, 'edit']">Edit</a>  
  
<a [routerLink]="['/authors', 0, 'edit']">Add Author</a>  
<a routerLink="/authors/0/edit">Add Author</a>
```

Component Class

```
this.router.navigate(['authors', this.author.authorId]);
```

Lectura de parámetros de ruta: Snapshot

Component Class

```
import { ActivatedRoute } from '@angular/router';  
...  
constructor(private route: ActivatedRoute) {  
    console.log(this.route.snapshot paramMap.get('authorId'))  
};  
}
```

Angular Route

```
{ path: 'authors/:authorId', component: AuthorEditPageComponent }
```

Template

```
<a [routerLink]="['/authors', author.authorId, 'edit']">{{author.authorName}}</a>
```


Lectura de parámetros de ruta: Observable

Component Class

```
import { ActivatedRoute } from '@angular/router';  
...  
constructor(private route: ActivatedRoute) {  
    this.route.paramMap.subscribe(  
        params => {  
            console.log(params.get('authorId'));  
        }  
    );  
}
```

Angular Route

```
{ path: 'authors/:authorId/edit', component: AuthorEditPageComponent }
```

Template

```
<a [routerLink]="['/authors', author.authorId, 'edit']">{{author.authorName}}</a>
```

Lectura de parámetros de ruta: Input

Component Class

```
import { Component, Input } from '@angular/core';  
...  
export class AuthorEditPageComponent {  
  @Input() authorId!: string;  
}
```

App.config

```
import { provideRouter, withComponentInputBinding, withHashLocation } from '@angular/router';  
...  
export const appConfig: ApplicationConfig = {  
  providers: [  
    ...  
    provideRouter(APP_ROUTES, withHashLocation(), withComponentInputBinding()),  
    ...  
  ]  
};
```

Angular Route

```
{ path: 'authors/:authorId/edit', component: AuthorEditPageComponent }
```

Template

```
<a [routerLink]="['/authors', author.authorId, 'edit']">{{author.authorName}}</a>
```

Lectura de parámetros de ruta

Snapshot

Para leer los parámetros sólo una vez
Código simple

Observable

Para observar los cambios de parámetros
Código más complejo

Definición de Query Parameters

Template

```
<a [routerLink] = "['/authors', author.authorId]"  
  [queryParams] = "{ filterBy: 'er', showImage: true }">  
  {{author.authorName}}  
</a>
```

Component Class

```
this.router.navigate(['authors'],  
  {  
    queryParams: { filterBy: 'er', showImage: true }  
  }  
);
```

Angular Route

```
{ path: 'authors', component: AuthorListPageComponent }
```

Lectura de Query Parameters - Snapshot

Component Class

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

constructor(private route: ActivatedRoute) {
  console.log(this.route.snapshot.queryParamMap.get('filterBy'));
  console.log(this.route.snapshot.queryParamMap.get('showImage'));
}
```

Template

```
<a [routerLink] = "[`/authors`, author.id]"
  [queryParams] = "{ filterBy: listFilter, showImage: showImage }">
  {{author.authorName}}
</a>
```

Lectura de Query Parameters - Input

Component Class

```
import { Component, Input } from '@angular/core';

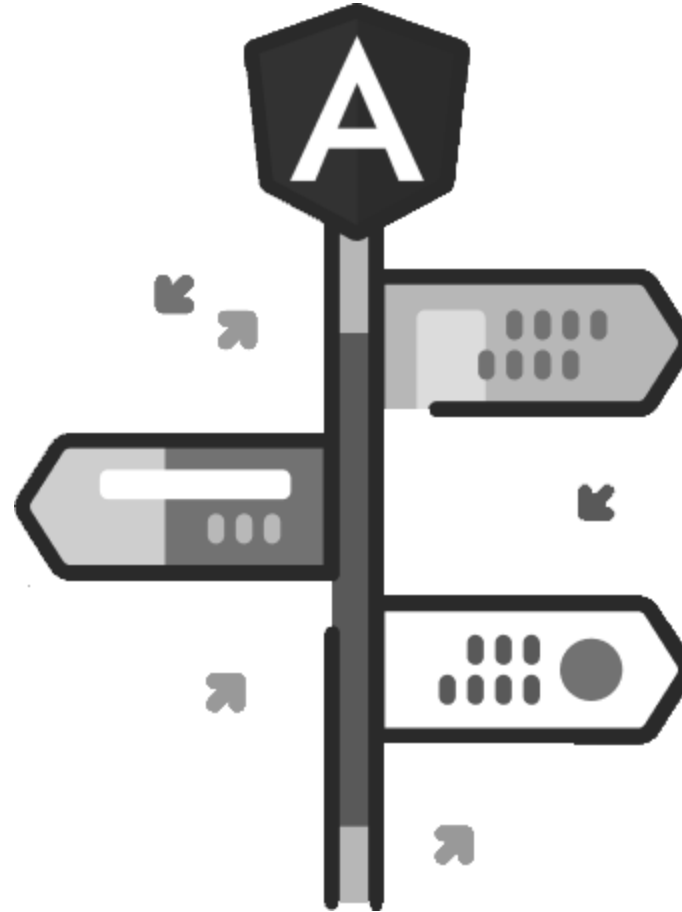
export class AuthorDetailPageComponent {
  @Input() authorId?: string;
  @Input() searchQuery?: string;
  @Input() orderBy?: string;
  @Input() author!: AuthorModel;
}
```

Template

```
<a
  [routerLink]="['./authors', '578359b7-1967-41d6-8b87-64ab7605587e']"
  [queryParams]="{searchQuery:'test',orderBy:'FirstName'}">Detail author
</a>
```

Enrutamiento con parámetros

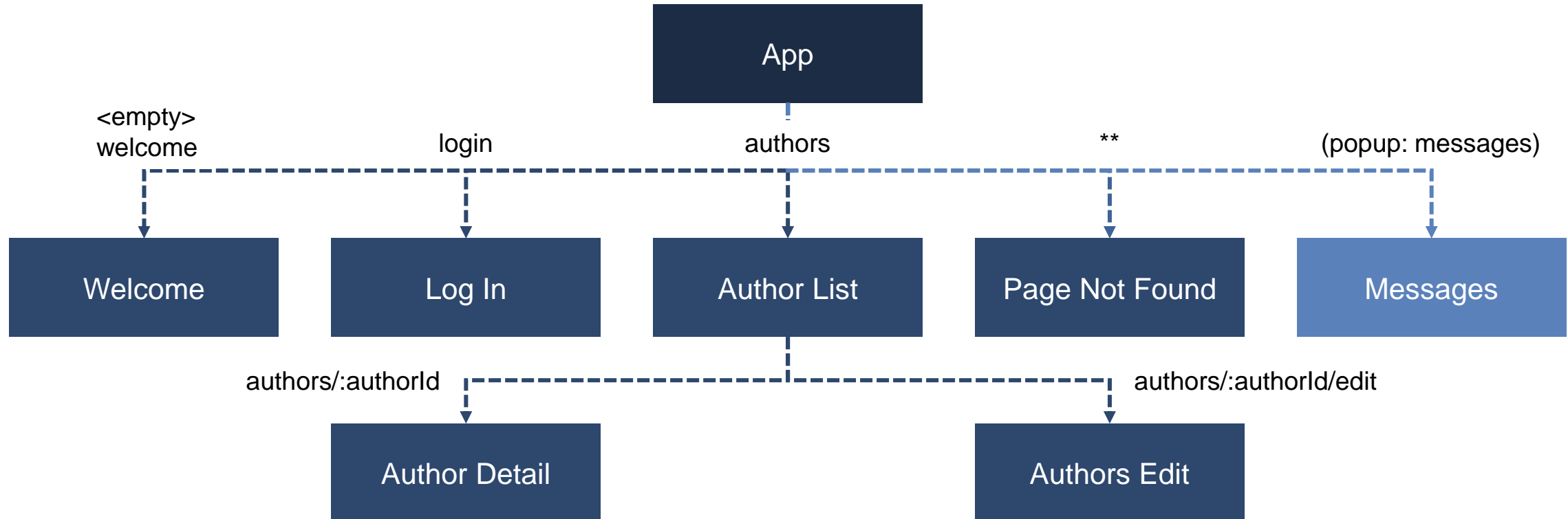
DEMO



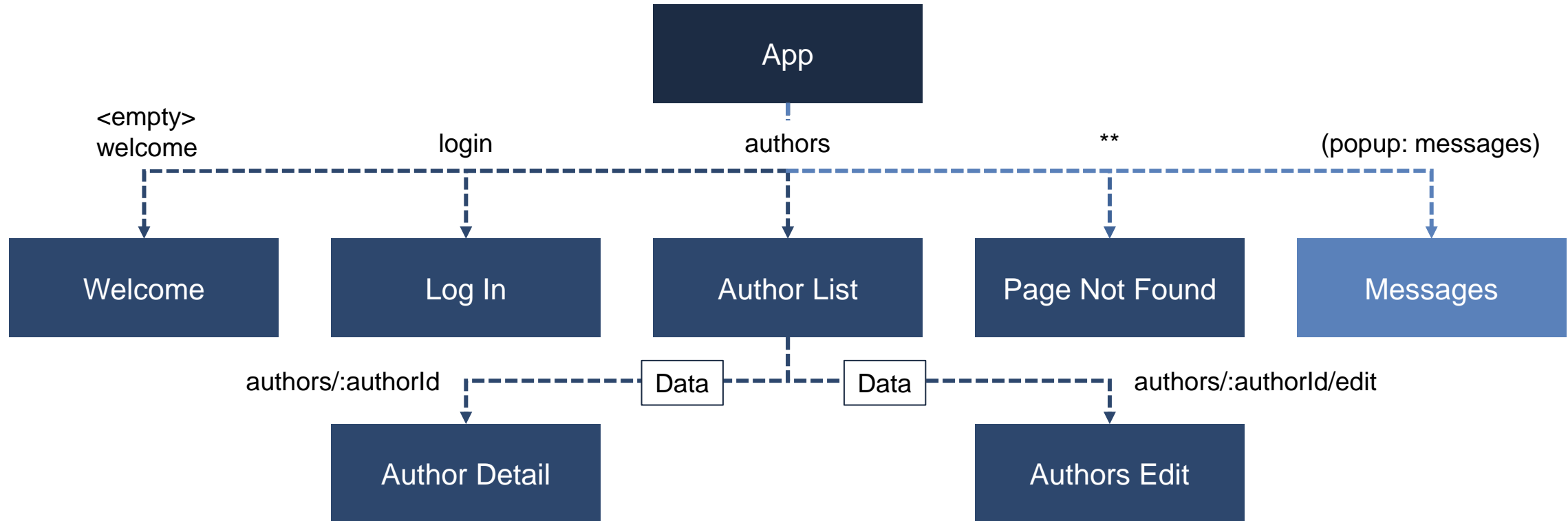
03

Precarga de datos mediante Route Resolvers

→ Route Resolvers



→ Route Resolvers



Propiedad Data de Route

Angular Route

```
export const routes: Routes = [  
  {  
    path: 'authors',  
    component: AuthorListPageComponent,  
    data: { pageTitle: 'Author List' }  
  },  
  { path: 'authors/:authorId', component: AuthorDetailPageComponent },  
  { path: 'authors/:authorId/edit', component: AuthorEditPageComponent }  
]
```

Component Class

```
this.pageTitle = this.route.snapshot.data['pageTitle'];
```

Component Class

```
@Input() pageTitle?: string;
```

Creación de un servicio de resolución de rutas

author-detail.resolver.ts

```
import { ResolveFn } from '@angular/router';
...

export const AuthorDetailResolver: ResolveFn<Promise<AuthorModel>> = async(route, state) => {
  const getAuthorByAuthorIdUseCase = inject(GetAuthorByAuthorIdUseCase);
  const authorId = route.paramMap.get('authorId');

  const author = getAuthorByAuthorIdUseCase.execute(authorId!);
  return author;
};
```

Agregar resolver a la configuración de ruta

Angular Route

```
[  
  { path: 'authors', component: AuthorListPageComponent },  
  { path: 'authors/:authorId', component: AuthorDetailPageComponent,  
    resolve: { author: AuthorDetailResolver }},  
  { path: 'authors/:authorId/edit', component: AuthorEditPageComponent,  
    resolve: { author: AuthorEditResolver }}  
]
```

Lectura de datos de Resolver: Snapshot

Component Class

```
import { ActivatedRoute } from '@angular/router';  
...  
author: Author;  
  
constructor(private route: ActivatedRoute) {  
    this.author = this.route.snapshot.data['author'];  
}
```

Angular Route

```
{ path: 'authors/:authorId', component: AuthorDetailPageComponent,  
  resolve: { author: AuthorDetailResolver } }
```

Resolver Service

```
export class AuthorDetailResolver implements Resolve<Author> { }
```

Lectura de datos de Resolver: Observables

Component Class

```
import { ActivatedRoute } from '@angular/router';  
...  
author: Author;  
constructor(private route: ActivatedRoute) {  
    this.route.data.subscribe(  
        data => this.author = data['author'];  
    );  
}
```

Angular Module

```
{ path: 'authors/:authorId/edit', component: AuthorEditPageComponent,  
  resolve: { author: AuthorEditResolver } }
```

Resolver Service

```
export class AuthorEditResolver implements Resolve<Author> { }
```

Lectura de datos de Resolver: Input

Component Class

```
import { Component, Input } from '@angular/core';  
...  
export class AuthorEditPageComponent {  
  ...  
  @Input() author!: AuthorModel;  
}
```

Angular Route

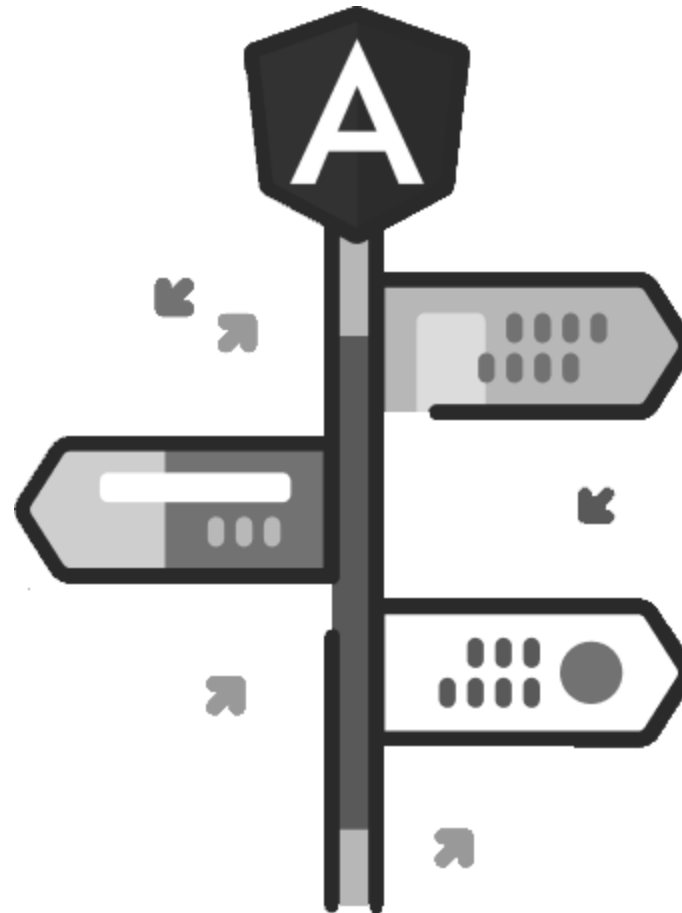
```
{ path: 'authors/:authorId/edit', component: AuthorEditPageComponent, resolve:  
  { author: AuthorEditResolver } }
```

Resolver Service

```
export const AuthorDetailResolver: ResolveFn<AuthorModel> = (route, state): AuthorModel => {}
```


Precarga de datos mediante Route Resolvers

DEMO

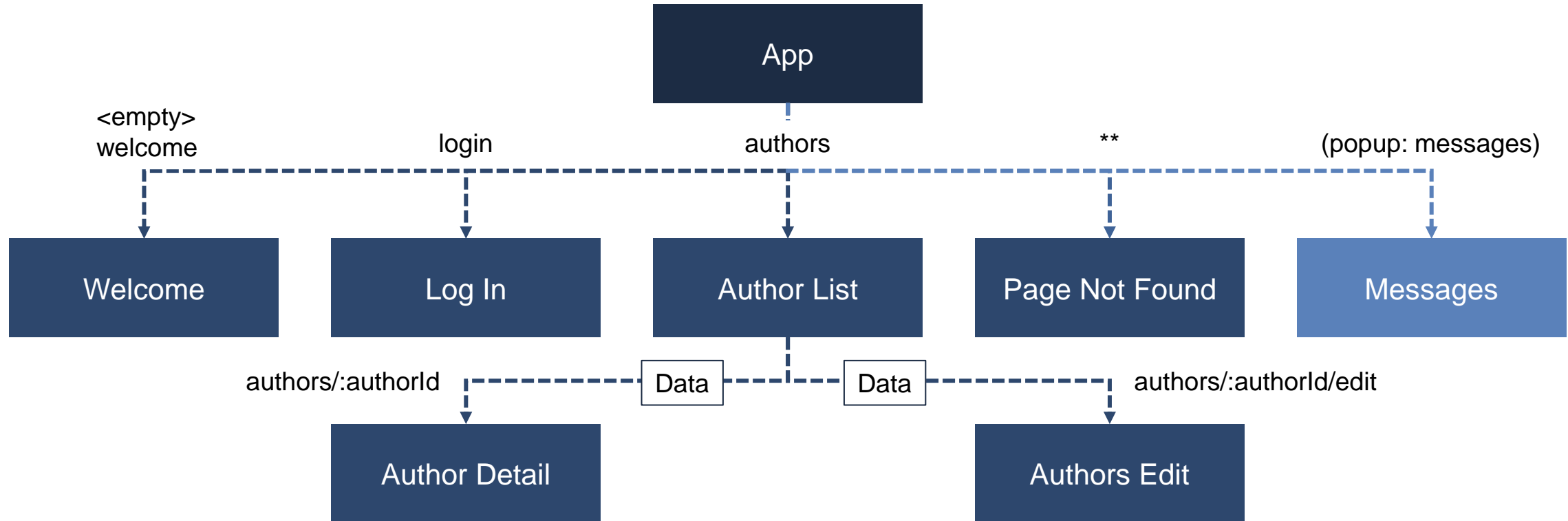


04

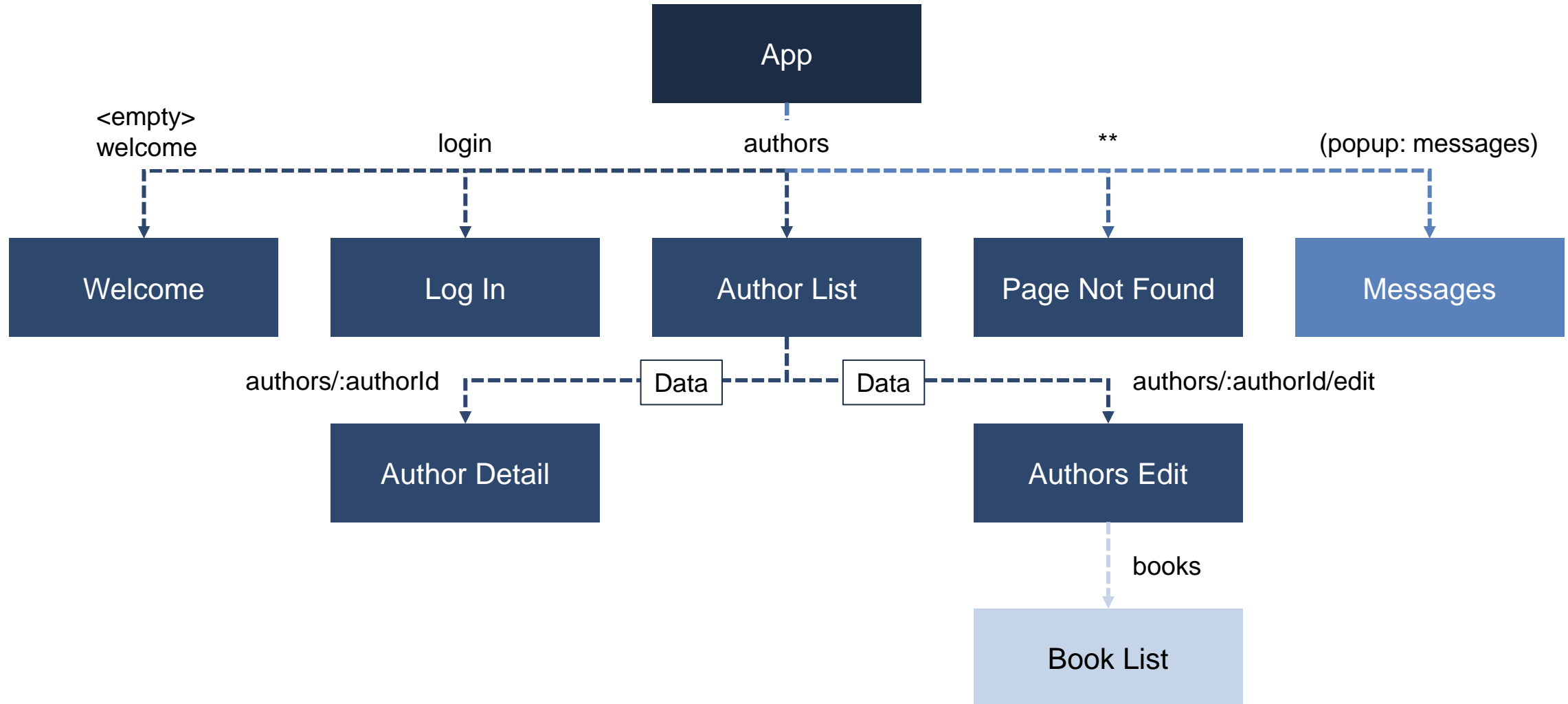


Rutas hijas

→ Rutas hijas



→ Rutas hijas



Configuración de rutas secundarias

Angular Route

```
{
  path: 'authors/:authorId/edit', component:
  AuthorEditPageComponent,
  resolve: { author: AuthorEditResolver },
  children: [
    {
      path: '', redirectTo: 'info', pathMatch: 'full'
    },
    {
      path: 'info', component: AuthorEditInfoComponent
    },
    {
      path: 'books', component: BookListComponent
    }
  ]
}
```

Activación de rutas hijas

Template

```
// Absolute path
<a [routerLink]="['/author', author.authorId, 'edit', 'info']">Info</a>

// Relative path
<a [routerLink]="['info']">Info</a>
```

Component Class

```
// Absolute path
this.router.navigate(['/authors', this.author.authorId, 'edit', 'info']);

// Relative path
this.router.navigate(['info'], { relativeTo: this.route });
```

Obtención de datos para una ruta hija

Author Data Service

```
this.authService.getAuthor(authorId)  
    .subscribe(author => this.author = author);
```

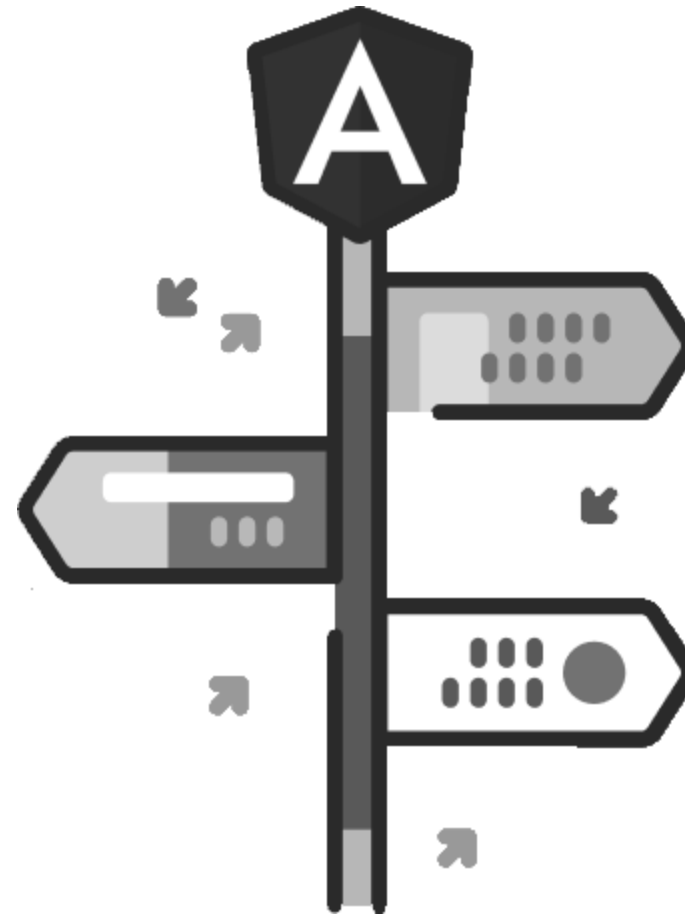
Child Route Resolver

```
this.author = this.route.snapshot.data['author'];
```

Parent Route Resolver

```
this.author = this.route.parent.snapshot.data['author'];
```

Rutas hijas



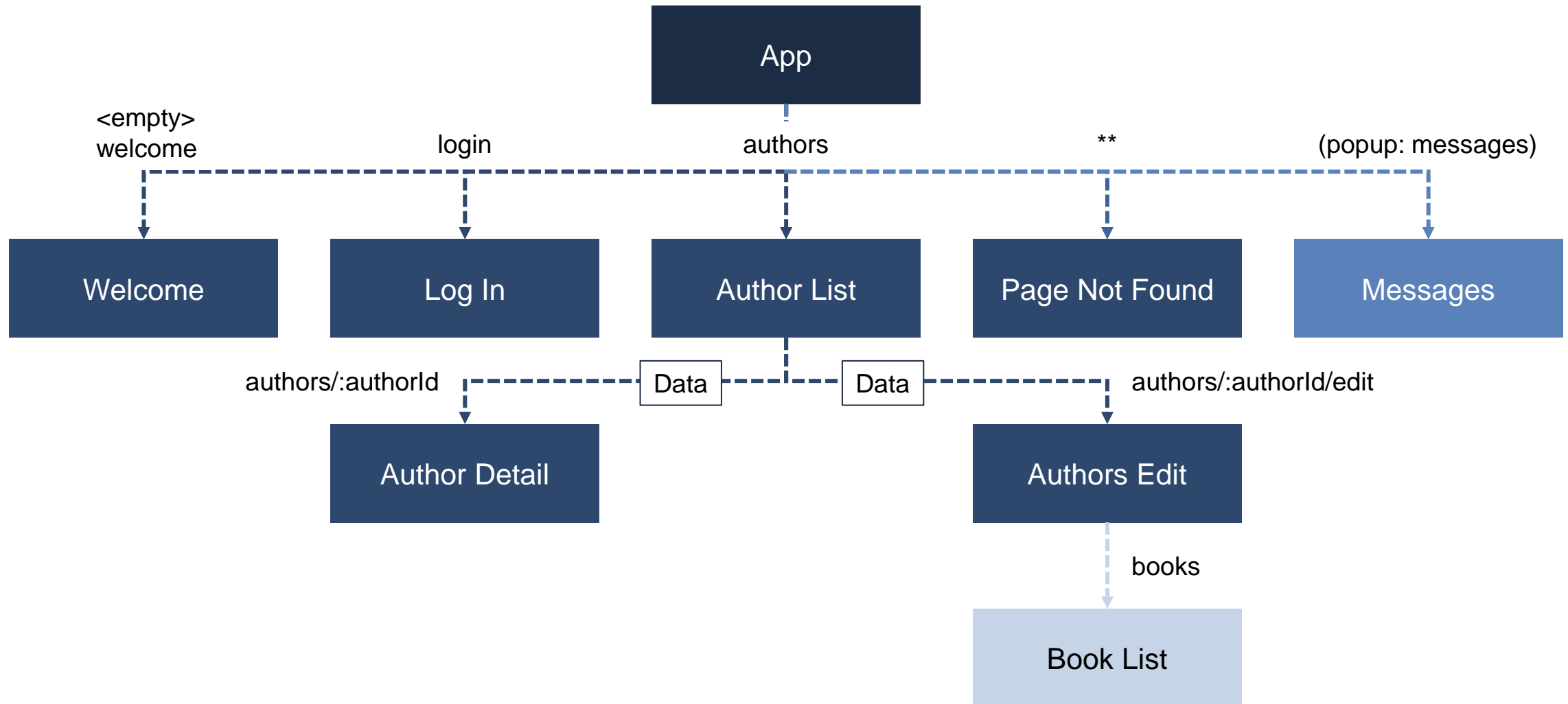
DEMO

05



Rutas secundarias

→ Rutas secundarias



Configuración de rutas secundarias

Angular Module

```
RouterModule.forChild([  
  {  
    path: 'messages',  
    component: MessageComponent,  
    outlet: 'popup'  
  },  
])
```

Activación de rutas secundarias: enlace de enrutador

Template

```
<a [routerLink]="[{ outlets: { popup: ['messages'] } }]">Messages</a>
```

```
<a [routerLink]("[{ outlets: { primary: ['authors', author.authorId, 'edit'], popup: ['summary'] } }]">Messages</a>
```

Activación de rutas secundarias: en código

Component Class

```
this.router.navigate([{ outlets: { popup: ['messages'] } }]);

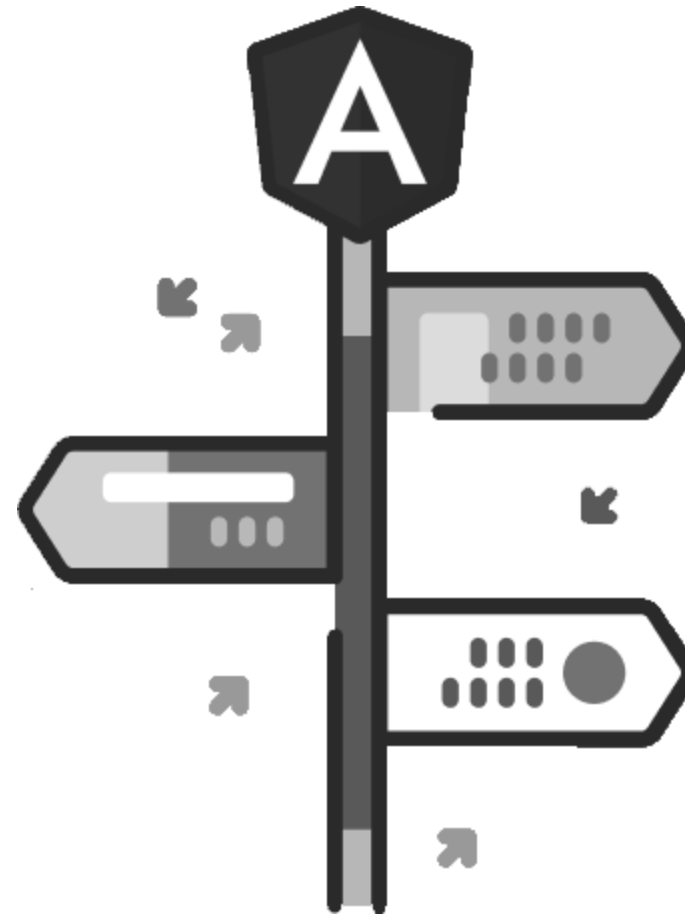
this.router.navigate(['/authors', authorId, 'edit',
    { outlets: { popup: ['summary', author.authorId] } }]);

this.router.navigate([{ outlets: {
    primary: ['/authors', author.authorId, 'edit'],
    popup: ['summary', author.authorId]
}}]);

this.router.navigateByUrl('/authors/5/edit(popup:summary/5)');
```

Rutas secundarias

DEMO





GRACIAS
POR SU PREFERENCIA

