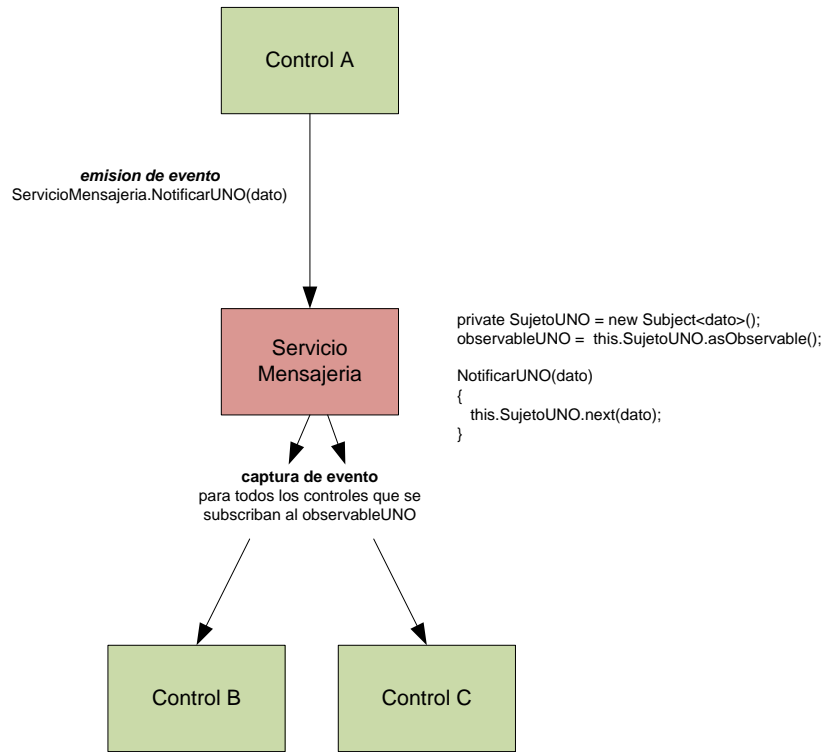


Multinotificación asíncrona de eventos entre controles sin relación padre/hijo



```

subs: Subscription;

constructor(private ServicioMensajeria: ServicioMensajeria)
{
  this.subs = ServicioMensajeria.observableUNO
    subscribe( objeto => { this.objeto = objeto; } );
}

ngOnDestroy()
{
  this.subs.unsubscribe();
}
  
```

```

TS mission.service.ts • TS astronaut.component.ts • TS mission-control.component.ts ✕
1 import { Component } from '@angular/core';
2
3 import { MissionService } from '../mission.service';
4
5 @Component({
6   selector: 'mission-control',
7   template: `
8     <h2>Mission Control</h2>
9     <button (click)="announce()">Announce mission</button>
10    <my-astronaut *ngFor="let astronaut of astronauts"
11      [astronaut]="astronaut">
12    </my-astronaut>
13    <h3>History</h3>
14    <ul>
15      <li *ngFor="let event of history">{{event}}</li>
16    </ul>
17  `
18 })
19 export class MissionControlComponent {
20   astronauts = ['Lovell', 'Swigert', 'Haise'];
21   history: string[] = [];
22   missions = ['Fly to the moon!',
23     'Fly to mars!',
24     'Fly to Vegas!'];
25   nextMission = 0;
26
27   constructor(private missionService: MissionService) {
28     missionService.missionConfirmed$.subscribe(
29       astronaut => {
30         console.log("MISSIONCOMPONENT captura evento_confirmed");
31         this.history.push(`${astronaut} confirmed the mission`);
32       });
33   }
34
35   announce() {
36     let mission = this.missions[this.nextMission++];
37     console.log("MISSIONCONTROL emite evento_announce");
38     this.missionService.announceMission(mission);
39     this.history.push('Mission "${mission}" announced');
40     if (this.nextMission >= this.missions.length)
41       this.nextMission = 0;
42   }
43 }
  
```

```

TS app.component.ts ✕
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   template: `
6     <div>
7       <h2>Hello {{name}}</h2>
8       <mission-control></mission-control>
9       <my-astronaut></my-astronaut>
10     </div>
11   `,
12   styleUrls: ['./app.component.css']
13 })
14 export class AppComponent {
15   title = 'app';
16 }
  
```

```

TS app.module.ts ✕
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5 import { AstronautComponent } from './astronaut/astronaut.component';
6 import { MissionControlComponent } from './mission-control/mission-control.component';
7 import { MissionService } from './mission.service';
8
9 @NgModule({
10   declarations: [
11     AppComponent,
12     MissionControlComponent,
13     AstronautComponent
14   ],
15   imports: [
16     BrowserModule
17   ],
18   providers: [MissionService],
19   bootstrap: [AppComponent]
20 })
21 export class AppModule {}
  
```

```

TS mission.service.ts • TS astronaut.component.ts • TS mission-control.component.ts
1 import { Injectable } from '@angular/core';
2 import { Subject } from 'rxjs/Subject';
3
4 @Injectable()
5 export class MissionService {
6
7   // Observable string sources
8   private missionAnnouncedSource = new Subject<string>();
9   private missionConfirmedSource = new Subject<string>();
10
11   // Observable string streams
12   missionAnnounced$ = this.missionAnnouncedSource.asObservable();
13   missionConfirmed$ = this.missionConfirmedSource.asObservable();
14
15   // Service message commands
16   announceMission(mission: string) {
17     this.missionAnnouncedSource.next(mission);
18   }
19
20   confirmMission(astronaut: string) {
21     this.missionConfirmedSource.next(astronaut);
22   }
23 }
  
```

```

TS mission.service.ts • TS astronaut.component.ts ✕ • TS mission-control.component.ts
1 import { Component, Input, OnDestroy } from '@angular/core';
2
3 import { MissionService } from '../mission.service';
4 import { Subscription } from 'rxjs/Subscription';
5
6 @Component({
7   selector: 'my-astronaut',
8   template: `
9     <p>
10       {{astronaut}}: <strong>{{mission}}</strong>
11       <button
12         (click)="confirm()">
13         Confirm
14       </button>
15     </p>
16   `
17 })
18 export class AstronautComponent implements OnDestroy {
19   @Input() astronaut: string;
20   mission = 'no mission announced';
21   confirmed = false;
22   announced = false;
23   subscription: Subscription;
24
25   constructor(private missionService: MissionService) {
26     this.subscription = missionService.missionAnnounced$.subscribe(
27       mission => {
28         this.mission = mission;
29         this.announced = true;
30         this.confirmed = false;
31         console.log("ASTRONAUTCOMPONENT captura evento_announced")
32       });
33   }
34
35   confirm() {
36     this.confirmed = true;
37     this.missionService.confirmMission(this.astronaut);
38     console.log("ASTRONAUTCOMPONENT emite evento_confirmed")
39   }
40
41   ngOnDestroy() {
42     // prevent memory leak when component destroyed
43     this.subscription.unsubscribe();
44   }
45 }
  
```