

Suscripción

¿Qué es una Suscripción? Una Suscripción es un objeto que representa un recurso desechable, normalmente la ejecución de un Observable. Las Suscripciones tienen un método importante, `unsubscribe`, que no recibe ningún argumento y se encarga de deshacerse del recurso mantenido por la Suscripción. En versiones anteriores de RxJS, la Suscripción llevaba el nombre de "Disposable" ("Desechable").

```
import { interval } from "rxjs";

const observable = interval(1000);
const subscription = observable.subscribe((x) => console.log(x));
/*
  Después:
  Esto cancela la ejecución del Observable que
  comienza al hacer una llamada subscribe con un Observador.
*/
subscription.unsubscribe();
```

[Copy](#)

Una Suscripción contiene una función `unsubscribe()` para desechar recursos o cancelar ejecuciones de Observables.

Las suscripciones también pueden unirse, de manera que una llamada a la función `unsubscribe()` de una suscripción pueda cancelar múltiples suscripciones. Se puede lograr esto al "añadir" una suscripción a otra:

```
import { interval } from "rxjs";

const observable1 = interval(400);
const observable2 = interval(300);

const subscription = observable1.subscribe((x) => console.log("first: " + x));
const childSubscription = observable2.subscribe((x) =>
  console.log("second: " + x)
);

subscription.add(childSubscription);
```

[Copy](#)

```
setTimeout(() => {  
  // Cancela AMBAS suscripciones: subscription y childSubscription  
  subscription.unsubscribe();  
}, 1000);
```

Al ejecutar el ejemplo anterior, se puede ver la siguiente salida en la consola:

```
second: 0  
first: 0  
second: 1  
first: 1  
second: 2
```

Las Suscripciones también tienen un método `remove(otherSubscription)`, para poder deshacernos de las Suscripciones hijas que se añadan.