

Observador

¿Qué es un Observador? Un Observador es un consumidor de los valores emitidos por un Observable. Los Observadores son simplemente conjuntos de *callbacks*, una para cada tipo de notificación entregada por el Observable: `next`, `error` y `complete`. A continuación vemos un ejemplo de un objeto Observador clásico:

```
const observer = {  
  next: (x) => console.log("El Observador ha recibido un valor next: " + x),  
  error: (err) => console.error("El Observador ha recibido un error: " + err),  
  complete: () =>  
    console.log("El Observador ha recibido una notificación de compleción"),  
};
```

Para utilizar el Observador, se le provee al `subscribe` de un Observable:

```
observable.subscribe(observer);
```

Los Observadores son objetos con tres callbacks, una para cada tipo de notificación que un Observable puede emitir.

Los Observadores en RxJS pueden ser parciales. Aunque no se proporcione una de las *callbacks*, la ejecución del Observable ocurrirá de forma normal, excepto por el hecho de que algunos tipos de notificaciones se ignorarán, ya que carecen de la *callback* correspondiente en el Observador.

El ejemplo visto a continuación muestra un Observador sin la *callback* `complete`:

```
const observer = {  
  next: (x) => console.log("El Observador ha recibido un valor next: " + x),  
  error: (err) => console.error("El Observador ha recibido un error: " + err),  
};
```

Al realizar la suscripción a un Observable, también se pueden proporcionar las *callbacks* como argumentos, sin tener que estar vinculadas a un objeto Observador. Por ejemplo:

```
observable.subscribe((x) =>
  console.log("El Observador ha recibido un valor next: " + x)
);
```

[Copy](#)

Internamente en `observable.subscribe`, se creará un objeto `Observador` utilizando el primer argumento *callback* como el manejador `next`. Los tres tipos de *callbacks* pueden proporcionarse como argumentos: