## ReactiveX

ReactiveX is a combination of the best ideas from the Observer pattern, the Iterator pattern, and functional programming

La programación reactiva es un paradigma que junto con el patrón Rx te ayudará a manejar mejor el flujo de información de una aplicación.

Se centra en flujos de información como secuencias ordenadas de datos. Estos datos pueden proceder de eventos en la interfaz de usuario, un arreglo, o una petición HTTP, etc.

ReactiveX es una plataforma de múltiples herramientas para operar secuencias de datos (basada en programación reactiva). Tiene implementaciones en varios lenguajes. Su implementación en JavaScript es RxJS.

### Patrón RxJS

El Observable se puede comunicar a través de los métodos: next, complete, y error. Y el Observador puede interactuar con el Observable a través de los métodos: subscribe y unsubscribe.

El Observable emite información con next, completa la información con complete, e informa de un error a través de error. El Observador puede suscribirse a un observable a través de subscribe y desuscribirse a través de unsubscribe.

Una vez un Observador se ha suscrito a un Observable, el Observador puede recibir valores de él.

Por definición un Observable no puede ser completado y luego generar un error, o generar un error y luego ser completado. También, por definición, un Observable no puede emitir más valores después de ser completado o después de generar un error.

Esta interacción es definida a través el Observable Contract (<https://reactivex.io/documentation/contract.html>).

## La idea de la reactividad

Patrones Observer e Iterator de comportamiento.

Observable son colecciones de múltiples valores

### **An Observable communicates with its observers with the following notifications:**

**OnNext**

conveys an item that is emitted by the Observable to the observer

**OnCompleted**

indicates that the Observable has completed successfully and that it will be emitting no further items

**OnError**

indicates that the Observable has terminated with a specified error condition and that it will be emitting no further items

**OnSubscribe (optional)**

indicates that the Observable is ready to accept Request notifications from the observer (see Backpressure below)

### An **observer** communicates with its Observable by means of the following notifications:

**Subscribe**

indicates that the observer is ready to receive notifications from the Observable

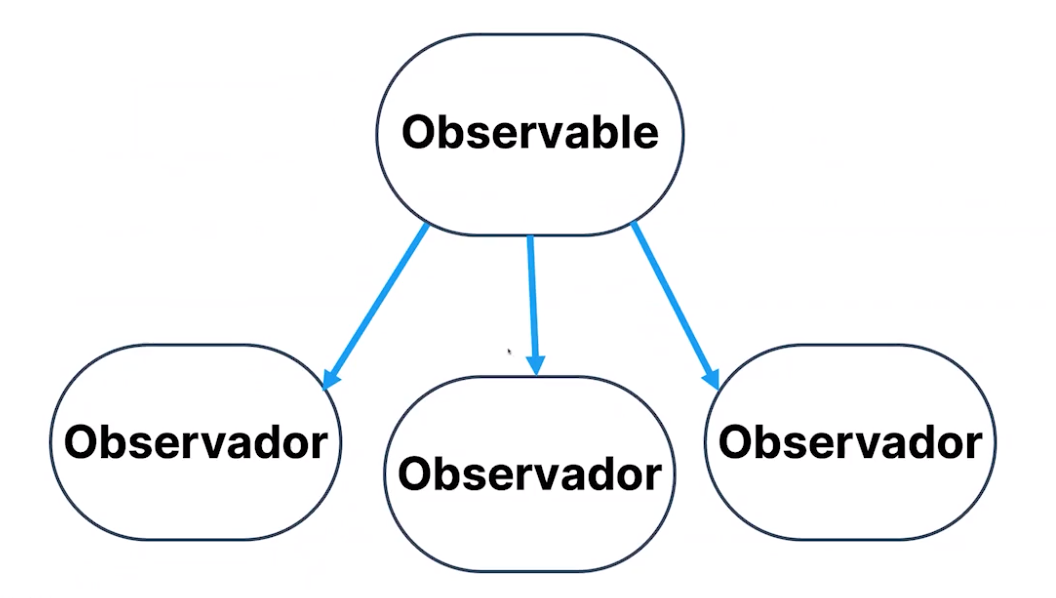
**Unsubscribe**

indicates that the observer no longer wants to receive notifications from the Observable

**Request (optional)**

indicates that the observer wants no more than a particular number of additional OnNext notifications from the Observable (see Backpressure below)

<https://reactivex.io/documentation/contract.html>



## Extensiones Reactivas

**ReactiveX != RxJS**

**REACTIVE EXTENSIONS LIBRARY FOR JAVASCRIPT**

RxJS is a library for reactive programming using Observables, to make it easier to compose asynchronous or callback-based code. This project is a rewrite of Reactive-Extensions/RxJS with better performance, better modularity, better debuggable call stacks, while staying mostly backwards compatible, with some breaking changes that reduce the API surface

## ¿Por qué deberías aprender programación reactiva con RxJS?

● Base de código mucho más legible.

● Simplifica los procesos asíncronos.

● Simplifica la transformación de información.

● Uso creciente en la industria tech.

### Testimonios

Al equipo de GitHub Desktop les encantó tanto Rx, que el equipo de Mac creó su propia versión de Rx [...], y ahora la están usando para obtener beneficios similares.

Paul Betts, GitHub

Rx simplificó drásticamente nuestro flujo de trabajo [...]. Nos impresionó tanto su versatilidad y calidad que la utilizamos como base para nuestra nueva plataforma de acceso a datos.

Jafar Husain, Netflix

RxJS es una tecnología de aprendizaje obligatoria para los nuevos miembros del equipo [que se integran a Netflix]. Jafar Husain, Netflix

Si el código espagueti asíncrono fuese una enfermedad, Rx es la cura.

Dave Sexton, Microsoft

### \

### RxJS versión 7

Más rápido. ● Operadores actualizados. ● Mejor tipado: TypeScript 4.2. ● Menor tamaño: -57% \*.

Una librería para crear secuencias de

observables con soporte en múltiples

lenguajes de programación y frameworks.

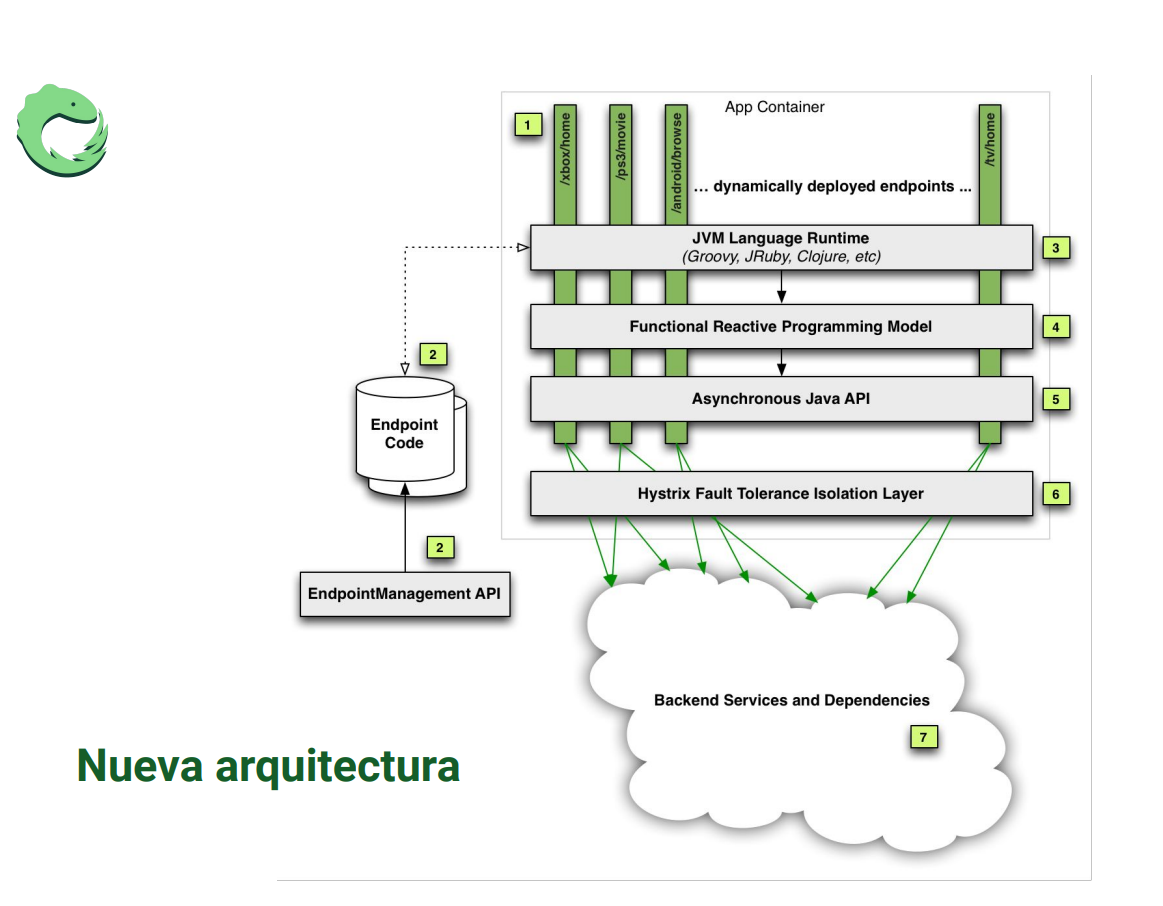
**Características**

● Basado en programación funcional.

● Reduce código con métodos.

● Manejo de errores para código asíncrono.

● Desarrollo sencillo de código asíncrono.



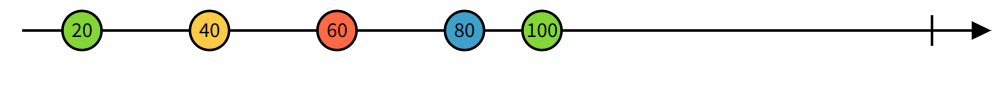
## Diagrama de canicas

Los diagramas de canicas están hechos para visualizar valores emitidos por un Observable en RxJS a través del tiempo.

Son de gran ayuda cuando queremos enteneder cómo funcionan los métodos en RxJS y visualizar los observables

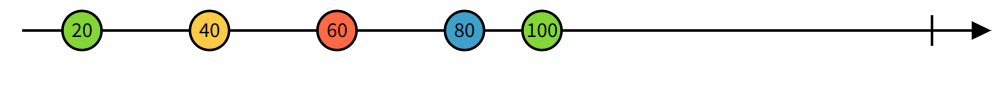
Representador por medio de una linea recta y una flecha para representar el tiempo, los valores se representan mediante circulos.

De la siguiente manera:



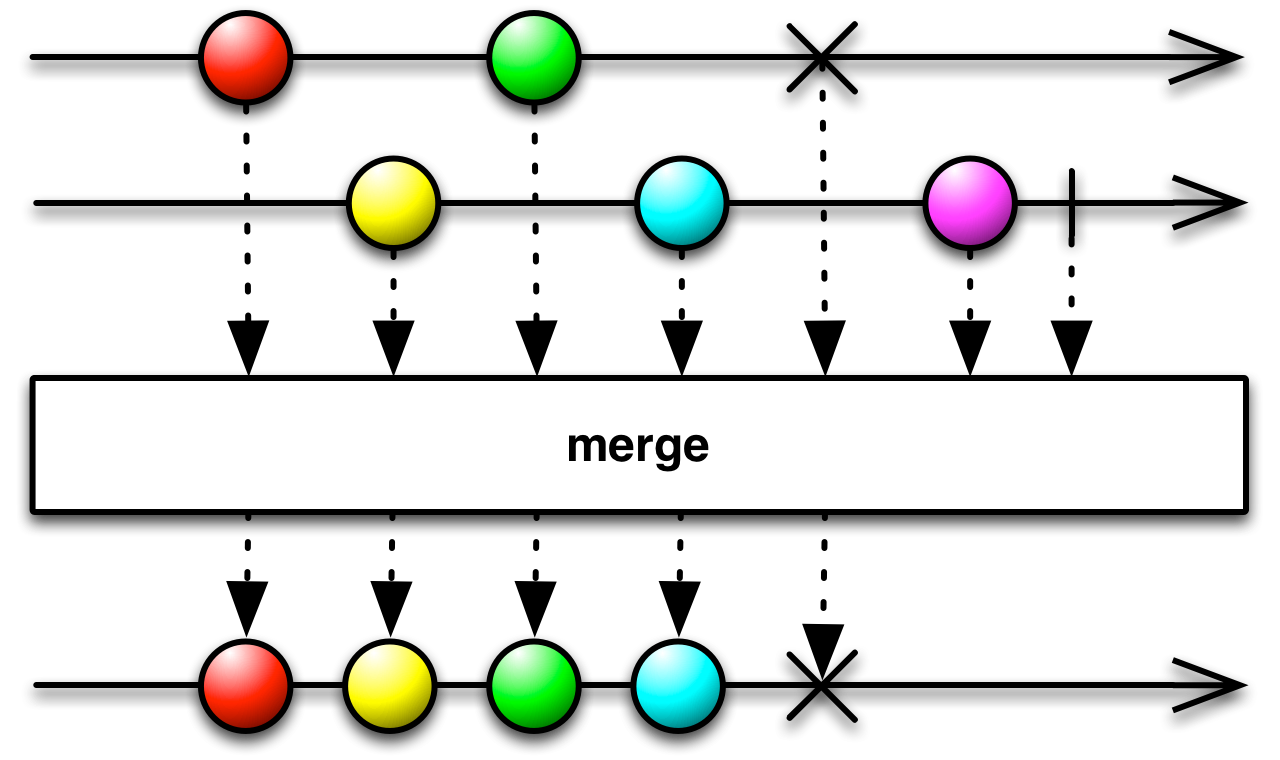
**Forma de terminar los observables**

**Complete**



Representado por una linea vertical que indica el final

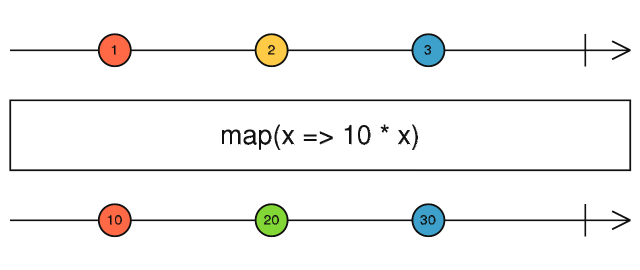
**Error**



Representado por una X vertical que indica el final

En ambos casos el observable dejara de emitir valores.

**Operator**



Representado por el rectangulo y la linea de abajo es el resultado del operador

<https://rxmarbles.com/>