La programación reactiva es un nuevo paradigma orientado a programar basado en flujos de datos​ que son los encargados de transmitir los cambios a nuestra aplicación-

Este ejemplo de programa se trata de las veces que se ha hecho clic en un botón.

Enfoque no reactivo:

**<!DOCTYPE>**

**<html>**

**<head>**

**<title></title>**

**<script src="rx.all.js" type="text/javascript">**

**</script>**

**</head>**

**<body>**

**<input type="button" id="boton" value="pulsar" />**

**</body>**

**<script type="text/javascript">**

**var total=0;**

**var boton = document.getElementById("boton");**

**boton.addEventListener("click", function(){**

**console.log(++total);**

**});**

**</script>**

**</html>**

Enfoque reactivo ​usando RxJS:

**<!DOCTYPE>**

**<html>**

**<head>**

**<title></title>**

**<script src="rx.all.js" type="text/javascript">**

**</script>**

**</head>**

**<body>**

**<input type="button" id="boton" value="pulsar" />**

**</body>**

**<script type="text/javascript">**

**var boton = document.getElementById("boton");**

**var botonStream = Rx.Observable.fromEvent(boton, 'click');**

**var suma=botonStream.map(function(x) {**

**return 1;**

**}).scan(function(x,y) {**

**return x+y;**

**}, 0).subscribe(function(resultado) {**

**console.log(resultado);**

**});**

**</script>**

**</html>**

Usar RxJS no es excesivamente complejo. El primer paso es generar un flujo de datos a partir de un conjunto de eventos (en este caso el click). Una vez hecho esto usamos dos operaciones comunes de programación funcional map y scan para convertir los eventos en números y finalmente sumarlos.

El último paso es subscribirse al flujo de datos e imprimir su resultado por consola, para

ello usamos el método subscribe.

Como podemos comprobar el código es mucho más reutilizable que el anterior y el

resultado idéntico.