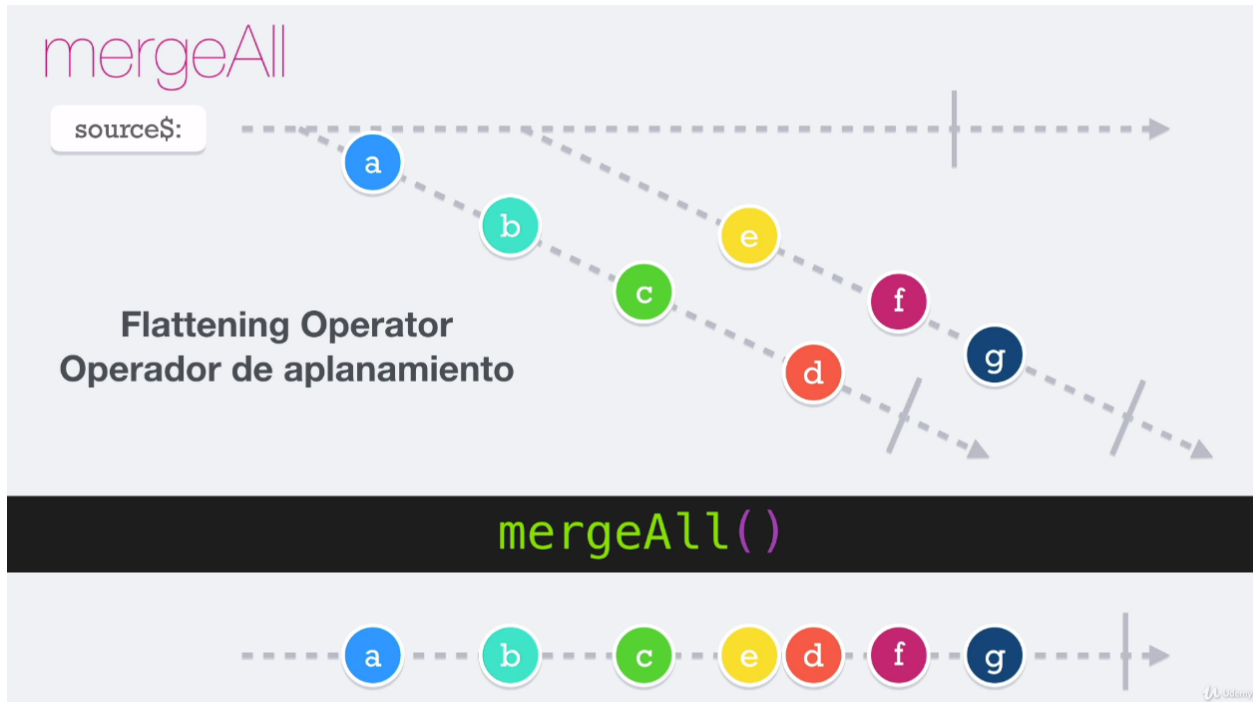


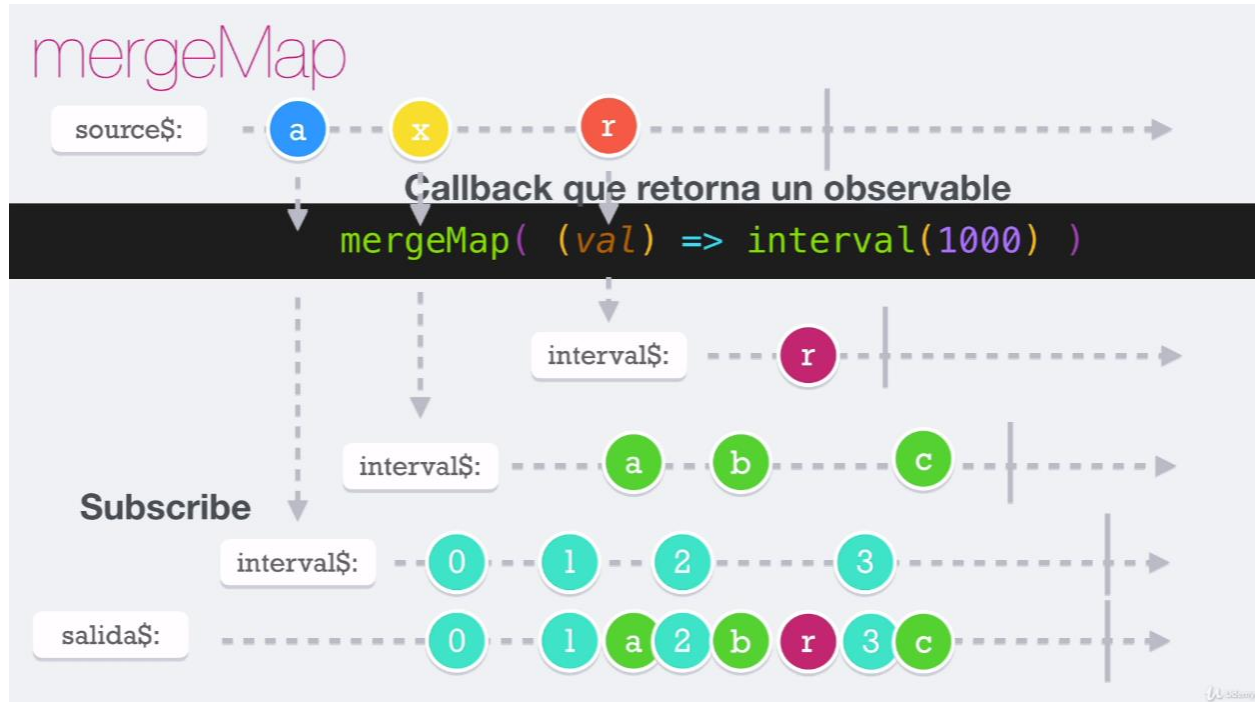
## 76. mergeAll

El operador `mergeAll` es un operador que sirve para trabajar con observables que internamente retornan observables y este no se completa hasta que todos sus observables hijos se completen. Este procedimiento de unificar observables en una sola salida se conoce como operador de aplanamiento (flattening operator).



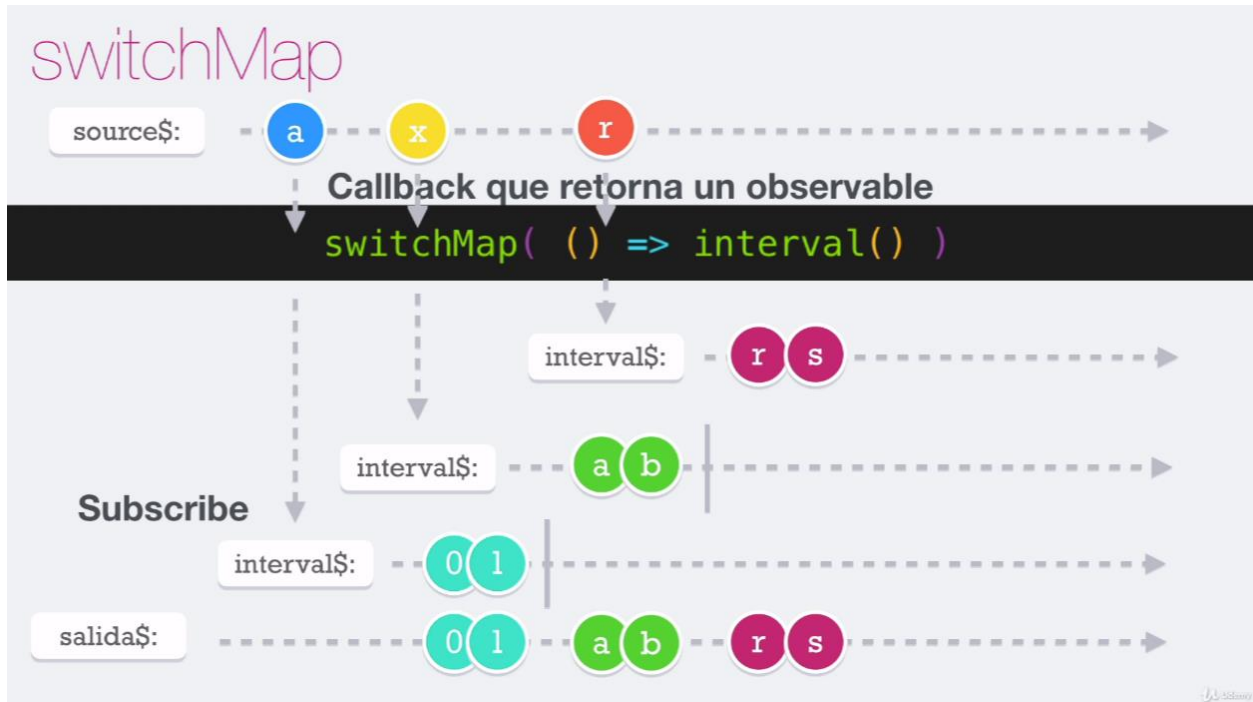
## 78. mergeMap

El operador mergeMap es otro operador de aplanamiento, es decir, que aunque estamos emitiendo un nuevo observable no tenemos un observable como salido sino que tenemos el producto de la suscripción del mismo.



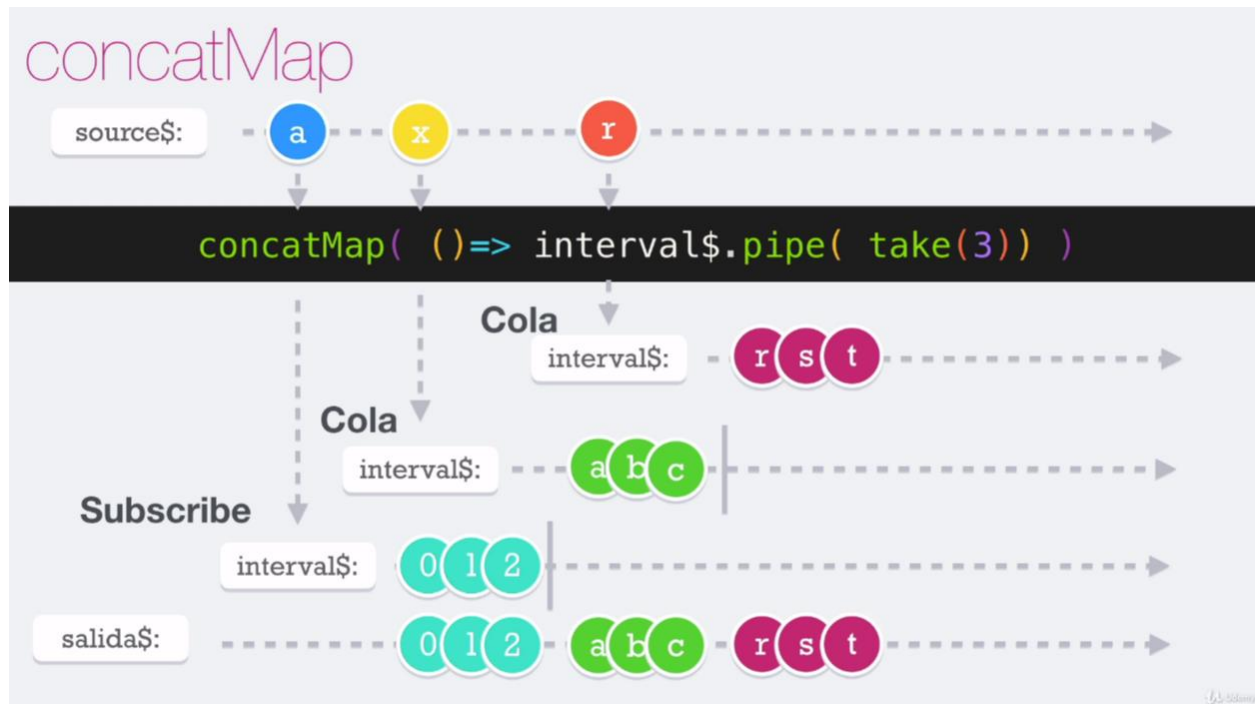
## 80. switchMap

El operador switchMap es muy parecido al mergeMap pero en lo que se diferencia es la forma en la que se maneja cuando el source emite un nuevo valor ya que cuando se genera una nueva línea de tiempo la anterior automáticamente se va a completar cosa que con el mergeMap no sucede. El switchMap es un operador que recibe un callback y retorna un observable, ese nuevo observable es el que se va a suscribir para hacer la emisión en la salida.



## 82. concatMap

El operador `concatMap` es otro operador de aplanamiento que nos sirve para concatenar los observables resultantes que pueden fluir a través de ese operador y hay que tener en cuenta que los va encolando y lo concatena al último valor emitido cuando el observable anterior se completa. Recordemos que al ser un operador de aplanamiento cuando se recibe un observable el operador automáticamente se va a suscribir a él y el resultado en la salida será el producto de dicho observable.



### 83. exhaustMap

El operador `exhaustMap` es otro de los operadores de aplanamiento, es decir, recibe un observable y maneja la suscripción internamente. Cuando `exhaustMap` recibe una entrada este se suscribe automáticamente y empieza a emitir valores, pero si recibe una nueva entrada y la anterior no se ha completado esta nueva entrada va a ser ignorada y la que estaba suscrita primer continua con su trabajo hasta que se complete, posteriormente si vuelve a recibir una suscripción este valida si tiene una suscripción activa y como no la tiene ahí si automáticamente va a empezar a emitir valores nuevamente. En pocas palabras el `exhaustMap` solo mantiene una suscripción activa a la vez antes de poder añadir otra suscripción.

