

RxJs & NgRx

- ¿Qué es y para qué sirve NgRx?
- Programación funcional y programación reactiva
- RxJs y conceptos generales
- Cómo funciona NgRx y sus elementos

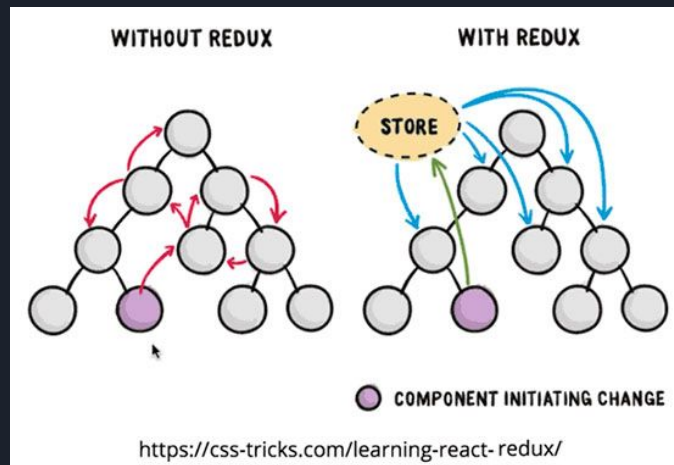
Fabian A. Zamudio

¿Qué es NgRx? ¿Para qué sirve?

Es un framework para hacer aplicaciones *reactivas* en Angular basadas en el patrón REDUX el cual implementa varias librerías para facilitar la administración del estado de la aplicación.

Esto es gracias a conceptos como:

- ÚNICA FUENTE DE LA VERDAD
- MANEJO DE EVENTOS Y EFECTOS
- ESTADO GLOBAL E INMUTABLE
- TRAZABILIDAD
- ADMINISTRACIÓN DE SUSCRIPCIONES



“Redux no hace rápido lo simple, sino mantenible lo complejo”



Programación reactiva

Programación orientada al manejo de streams de datos asíncronos y la propagación del cambio

Stream: Flujo de datos provenientes de alguna fuente

Observable: Es el productos del stream de datos. Eventos de teclado, promesas, llamadas http, arrays, etc

Observer: En la función de escucha del stream que reacciona a ciertos cambios

Iterador: Convierte a cualquier stream en un contenedor de información a recorrer



Programación funcional

Funciones puras: Son funciones que con los mismos argumentos siempre devolverán el mismo resultado. No tienen efectos colaterales (side effects) ya que no mutan datos, ni tienen acceso a ningún elemento externo.

Funciones de orden superior: Son funciones puras, que reciben otras funciones como argumentos para la realización de cálculos

Inmutabilidad: Un objeto no puede ser modificado una vez creado, se prefiere crear uno desde cero y destruir el anterior.

Let's code



RxJs y conceptos generales

Librería basada en programación funcional y reactiva con diferentes utilidades y mejoras de performance.

Observable: Colección de eventos y datos futuros

Observer: Colección de funciones que saben cómo reaccionar a los cambios del observable.

Subscription: Es la conexión entre un observer y un observable.

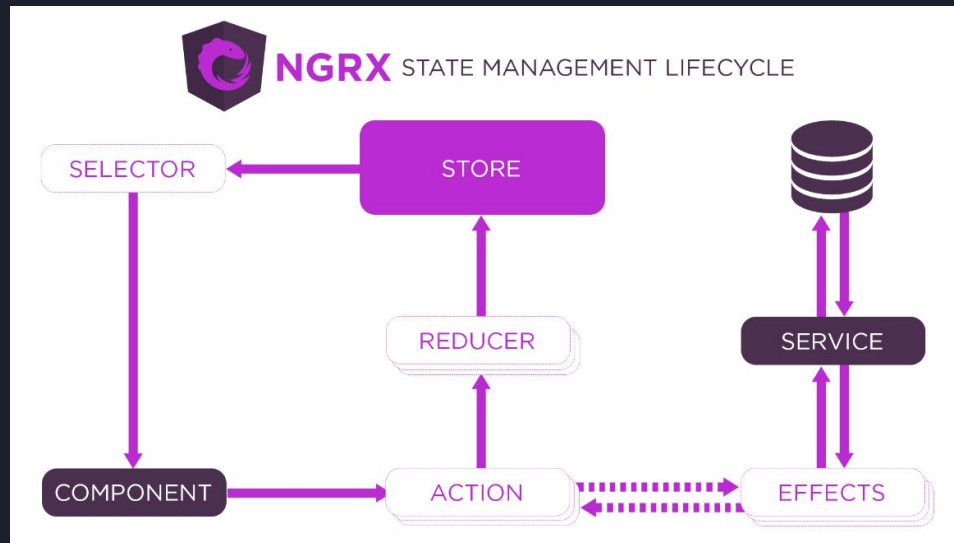
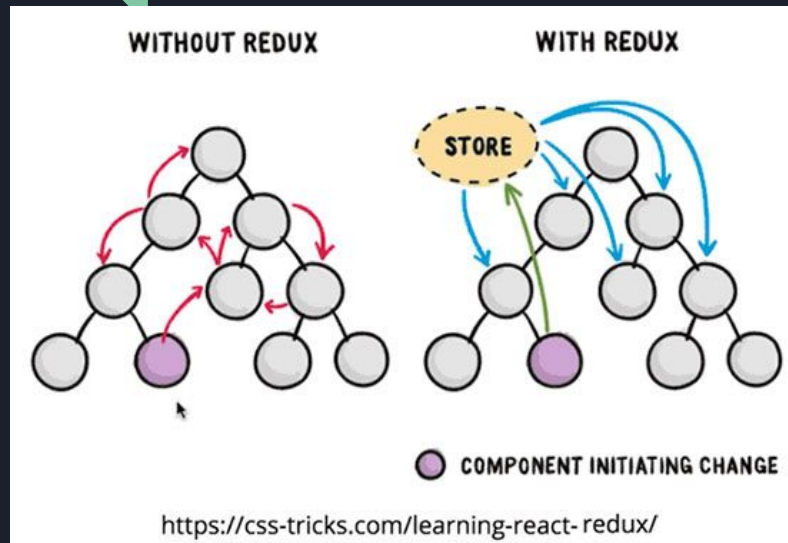
Operators: Funciones puras

Subject: Equivalen a los EventEmitter de Angular.

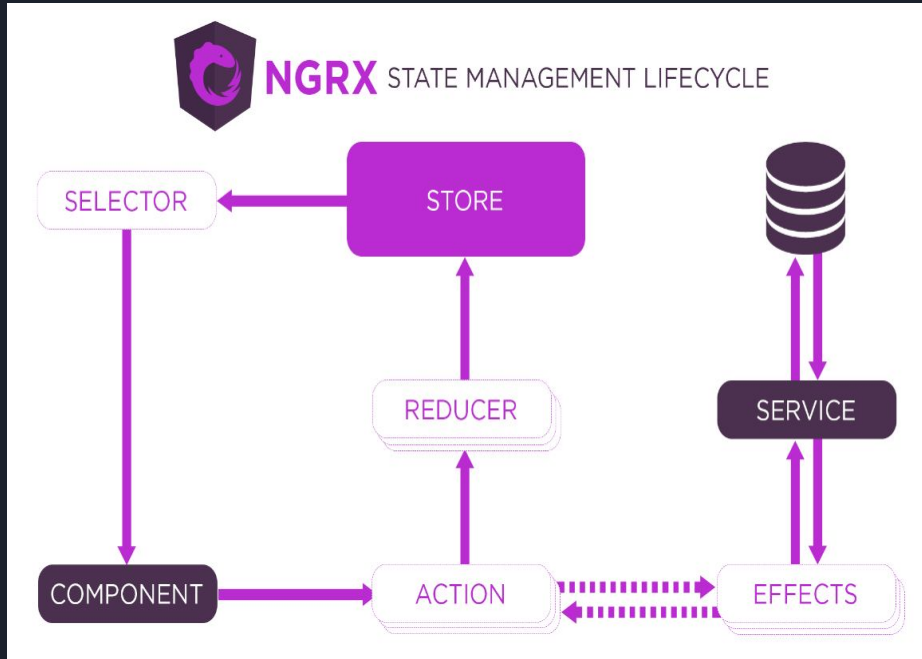
Schedulers: Permiten coordinar la concurrencia



Volviendo al problema



Elementos de NgRx



Store: Es el estado de la aplicación, es inmutable y sólo se modifica mediante las funciones “Reducers”.

Reducers: Son funciones puras que no modifican directamente partes de la interfaz, no almacenan en base de datos, no producen solicitudes Ajax a servicios web, no llaman a otras funciones para realizar su trabajo, etc. Solo modifican el estado del STORE.

Actions: Son los nombres/interfaces que los Reducers saben atender.

Selectors: Funciones que toman valores del STORE y aplican funciones puras para devolver datos procesados.

Efectos: Son funciones encargadas de aislar las llamadas a servicios externos para que los componentes sean puros.



Ahora **Beto** nos va a hacer la
demostración de código en vivo

