

## Angular NgRx: Introducción - ¿Qué es NgRx?

NgRx es un framework para construir aplicaciones reactivas en Angular. NgRx proporciona gestión de estado, aislamiento de “side effects” y mucho más. Hay una lista de librerías que componen NgRx:

- @ngrx/store
- @ngrx/store-devtools
- @ngrx/effects
- @ngrx/router-store
- @ngrx/entity
- @ngrx/data
- @ngrx/component
- @ngrx/schematics

Aprovechamos NgRx en cualquier aplicación Angular nueva o existente importando estas librerías.

Para tener un modelo mental de NgRx, veámoslo de otra manera. NgRx es el popular patrón de gestión de estados llamado Redux, adaptado a Angular utilizando conceptos y técnicas estándar de Angular.



=



+



NgRx

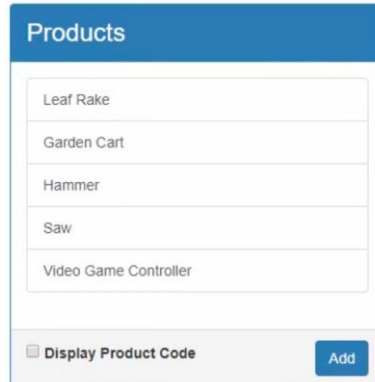
redux  
pattern

Angular

Centrémonos en revisar el patrón Redux para entender mejor qué es NgRx.

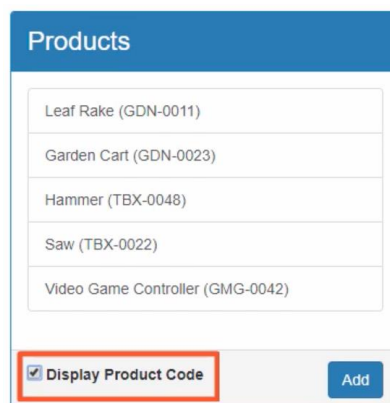
El patrón Redux nos ayuda a gestionar el estado de nuestra aplicación proporcionando un flujo de datos de tipo one-way a través de la aplicación.

Empecemos con la vista:



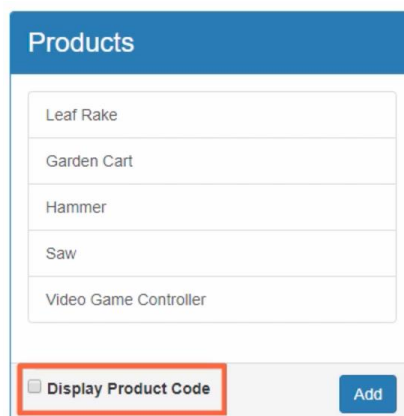
The screenshot shows a web form titled 'Products'. It contains a list of five items: 'Leaf Rake', 'Garden Cart', 'Hammer', 'Saw', and 'Video Game Controller'. At the bottom of the form, there is a checkbox labeled 'Display Product Code' which is currently unchecked, and a blue 'Add' button to its right.

En este ejemplo tenemos un checkbox en la parte inferior de la vista que el usuario puede seleccionar para incluir el código del producto en la pantalla:



This screenshot shows the same 'Products' form, but the 'Display Product Code' checkbox is now checked. A red rectangular box highlights the checkbox and its label. The product names in the list are now followed by their respective codes in parentheses: 'Leaf Rake (GDN-0011)', 'Garden Cart (GDN-0023)', 'Hammer (TBX-0048)', 'Saw (TBX-0022)', and 'Video Game Controller (GMG-0042)'. The 'Add' button remains visible.

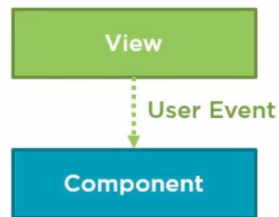
Podríamos definir una propiedad local en el componente para realizar un seguimiento de este checkbox. Pero si el usuario navega a otra vista y vuelve, perderemos su selección:



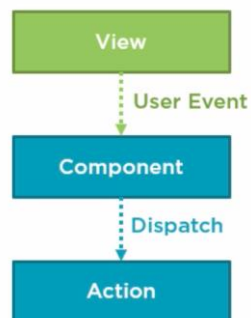
This screenshot shows the 'Products' form after the user has navigated away and returned. The 'Display Product Code' checkbox is now unchecked again, despite having been checked in the previous state. A red rectangular box highlights the checkbox and its label, illustrating that the local state was lost.

Veamos cómo puede ayudarnos el patrón Redux:

1. Cuando el usuario hace clic en el checkbox, la vista utiliza el event binding para notificar al componente del evento del usuario:

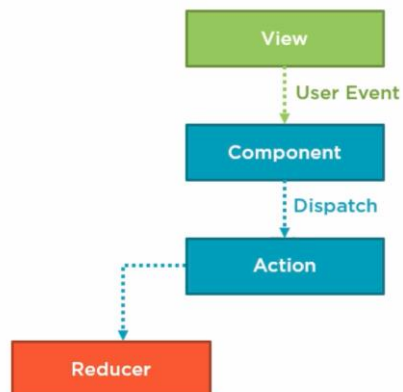


2. A continuación, el componente crea un Action que representa ese evento:

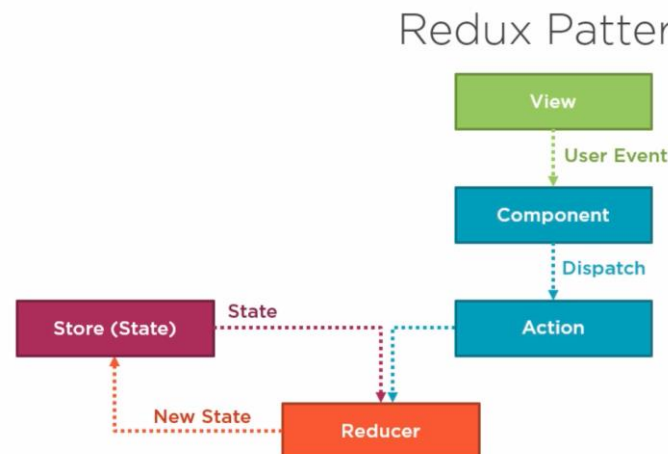


3. El componente envía esta acción a una función llamada Reducer:

## Redux Pattern

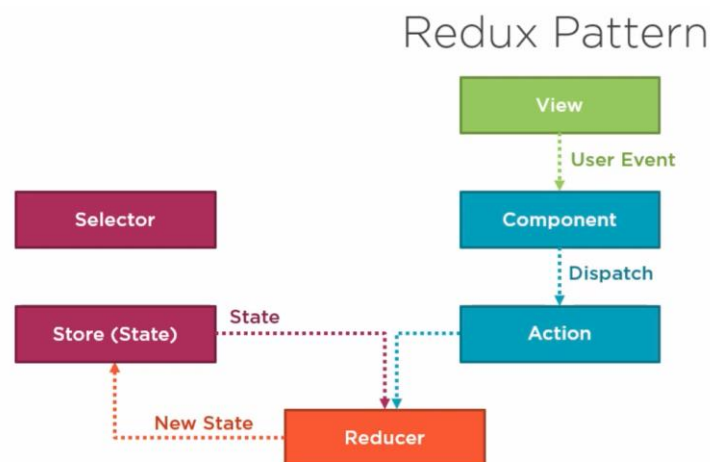


4. El Reducer utiliza el Action y el estado actual de la aplicación desde el Store de NgRx para definir un nuevo estado y actualizar el store con ese estado:

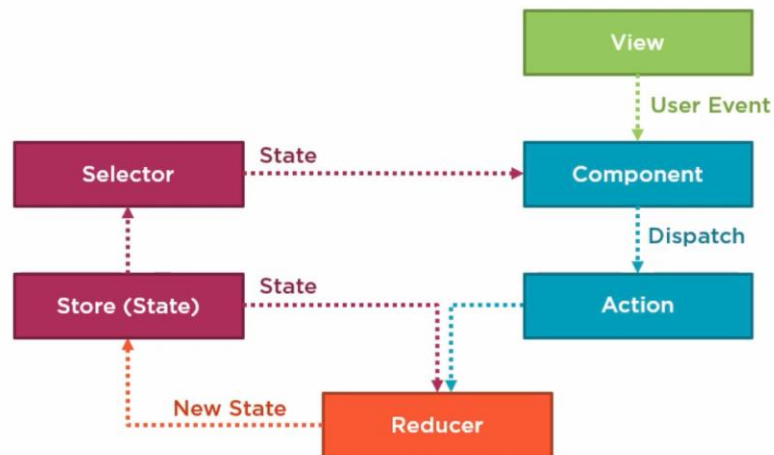


El Store es un contenedor de estado único, en memoria y del lado del cliente. Sólo almacena el estado de nuestra aplicación durante la ejecución de la aplicación misma. Usando ese patrón, nuestro estado es inmutable, lo que básicamente significa que nunca se modifica. En su lugar, el Reducer crea un nuevo estado a partir del estado existente y el Action definido, haciendo que los cambios de estado sean mucho más explícitos.

5. Cualquier componente se suscribe al Store utilizando un Selector para recibir notificaciones de cambios de estado específicos. Piense en un Selector como un Store Procedure que sabe cómo localizar y devolver los datos de nuestro Store:



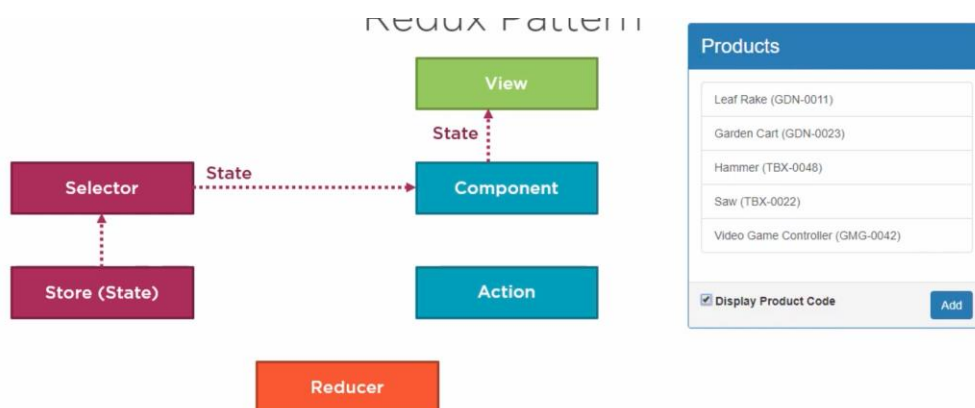
6. Cuando se sustituye un nuevo estado del Store, el componente recibe una notificación de ese nuevo estado y se actualizan los valores vinculados en la vista:



En este caso, la vista muestra los códigos de los productos:

A screenshot of a web form titled 'Products'. It contains a list of products with their codes: Leaf Rake (GDN-0011), Garden Cart (GDN-0023), Hammer (TBX-0048), Saw (TBX-0022), and Video Game Controller (GMG-0042). Below the list is a checkbox labeled 'Display Product Code' which is checked, and an 'Add' button.

7. Si el usuario navega fuera de la vista y regresa, el componente se vuelve a suscribir, obteniendo inmediatamente el estado del Store, y la vista se actualiza con la selección del usuario guardada:



Con NgRx, para cualquier bit de estado que retenemos más allá de la vida de un solo componente, o que proporcionamos a otros componentes, enviamos un Action para agregar esos datos a nuestro Store para que podamos gestionarlos mejor. Nuestro componente y otros componentes se suscriben a través de un Selector para vigilar cualquier cambio en los datos almacenados. Esto define nuestro flujo de datos unidireccional:

