Practica Evaluable 6

Lucia Martinez Tomas

MEDAC
Instituto Oficial de Formación Profesional

Índice

Enlaces:

https://github.com/lmt0004/mobx.git

https://mobx-lucia.onrender.com

1. Tarea 1

- Realiza un breve estudio del patrón de software Flux. Analiza su origen, objetivo, funcionamiento, aplicabilidad en el mundo real, etc.

Flux es una arquitectura que sirve para el manejo de datos en una aplicación web y sobre todo en el apartado de Front-End.

Fue ideada por Facebook y servirá para sustituir el patrón MVC o MVVM. El patrón sigue una arquitectura modelo-vista-controlador. Haciendo que sea más sencillo el manejo de datos en aplicaciones web con algo de complejidad.

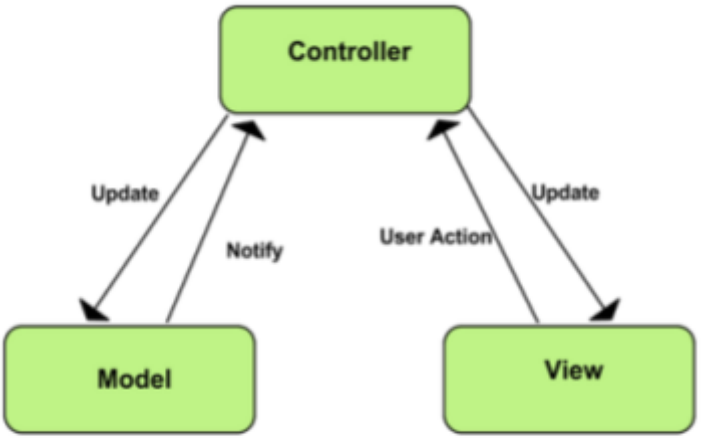

Fig.0- Modelo Vista Controlador

El origen de flux nació ya que a Facebook se le presentó un problema de comunicación entre modelos y controladores bidireccionales , haciendo difícil la depuración y resetear errores.

Flux presenta una arquitectura de datos unidireccional. Y los datos viajan desde la vista por medio de acciones y llegan a un store desde el cual se actualiza la vista de nuevo.

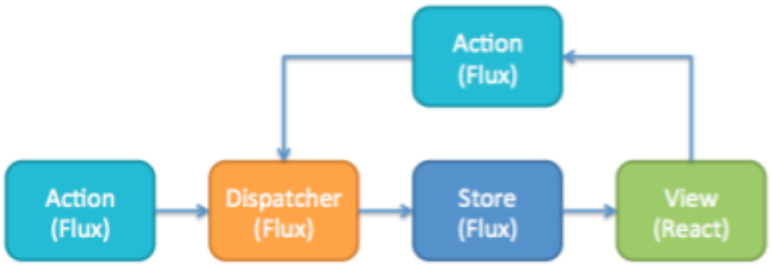

Fig.1- Funcionamiento Flux

Al igual que MVC tenemos distintos actores:

- Vista: la vista es los componentes web
- Store: sería lo más parecido al modelo de la app. Guarda los datos y los estados de la app. No se puede modificar datos.
- Acciones: es un objeto de JS que indica una intención de realizar algo y que lleva datos asociados si es necesario.
- Dispatcher: es un mediador entre la Store y las acciones. Sirve para desacoplar la Store de la vista.

Aplicación para la vida real:

Flux, especialmente cuando se utiliza con React, se ha aplicado en una variedad de situaciones para gestionar de manera eficiente el estado de la aplicación y mejorar la estructura de código.

2. Tarea 2

Redux

- Realiza un breve estudio de la librería Redux. Analiza su origen, objetivo, funcionamiento, aplicabilidad en el mundo real, etc.

Redux es una biblioteca de gestión de estado para aplicaciones JavaScript, principalmente relacionada con el desarrollo de aplicaciones React. Creada por Dan Abramov y Andrew

Clarke, esta solución se basa en el patrón Flux para proporcionar una forma fácil y predecible de gestionar el estado en aplicaciones complejas.

Fig.2- Funcionamiento Redux

El objetivo principal que tiene Redux es proporcionar un contenedor de estado centralizado y predecible para aplicaciones JS. Intenta resolver los problemas asociados con la gestión del estado en aplicaciones a gran escala proporcionando un flujo de datos unidireccional y una forma clara de actualizar y acceder al estado de la aplicación.

El funcionamiento de Redux se basaría en que Redux sigue el principio de un árbol de estado único. En el corazón de Redux hay una "tienda" que almacena todo el estado de la aplicación en objetos de estado inmutable. Una operación es un evento que provoca un cambio de estado y se envía a la tienda. Un reductor es una función pura que especifica cómo cambia el estado de la aplicación en respuesta a una operación particular. Los componentes de React se conectan a la tienda y reciben actualizaciones automáticamente cuando cambia su estado.

Para su aplicación al mundo real:

Redux se usa ampliamente en aplicaciones React y ha demostrado ser efectivo en escenarios donde la gestión del estado es compleja. Esto es especialmente útil para aplicaciones grandes donde múltiples componentes deben acceder y actualizar el estado

constantemente. Además, la naturaleza unidireccional de Redux facilita la depuración y el seguimiento del flujo de datos.

Mobx

- Realiza un breve estudio de la librería Mobx. Analiza su origen, objetivo, funcionamiento, aplicabilidad en el mundo real, etc.

El origen de Mobx es que MobX es una biblioteca de gestión de estado para aplicaciones JavaScript, utilizada principalmente para el desarrollo de aplicaciones React. Fue creado por Michel Weststrat para proporcionar una solución simple y eficiente para la gestión del estado en aplicaciones complejas.

El objetivo principal de MobX es que es una biblioteca de gestión de estado para aplicaciones JavaScript, utilizada principalmente para el desarrollo de aplicaciones React. Fue creado por Michel Weststrat para proporcionar una solución simple y eficiente para la gestión del estado en aplicaciones complejas.

El funcionamiento de MobX es el siguiente: utiliza un modelo reactivo donde el estado se define como un árbol de objetos monitoreados. MobX mantiene automáticamente estos observables, por lo que cuando un observable cambia, todos los componentes que dependen de él se actualizan automáticamente. Además de Observables, MobX utiliza "acciones" para cambiar de estado de forma controlada y "reacciones" para definir cómo responde un componente a los cambios de estado.

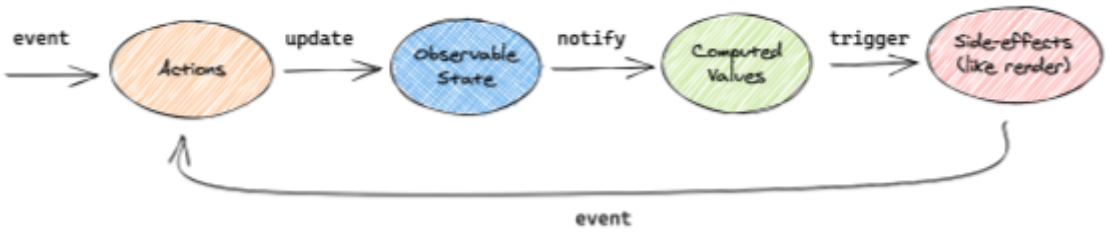

Fig.3- Funcionamiento de Mobx

Aplicación al mundo real:

MobX se utiliza en una variedad de escenarios específicos, especialmente aquellos que buscan una solución de gestión pública más completa y sencilla. Esto es especialmente útil para aplicaciones o equipos pequeños donde la simplicidad y el rendimiento son prioridades. La flexibilidad de MobX lo hace adecuado para proyectos de todos los tamaños.

BIBLIOGRAFIA:

https://carlosazaustre.es/como-funciona-flux

https://carlosazaustre.es/como-funciona-redux-conceptos-basicos

https://rootstack.com/es/technology/mobx