MEDAC
FUNDACIÓN

Práctica Evaluable 6

Mateo González Núñez

Flux

El patrón de diseño Flux es un patrón de arquitectura de software que se utiliza para construir aplicaciones web de una sola página. Este patrón se enfoca en el manejo del flujo de datos a través de la aplicación, separando la lógica de negocio de la interfaz de usuario.

El objetivo principal de Flux es proporcionar una estructura clara y unidireccional para manejar el flujo de datos en aplicaciones web. Busca separar la lógica de negocio de la interfaz de usuario y garantizar que los cambios en el estado de la aplicación sean manejados de manera predecible y escalable.

El patrón Flux consta de cuatro elementos principales:

- Acciones: son objetos que describen algo que sucede en la aplicación, como la entrada del usuario o la respuesta del servidor.
- Dispatcher: es un objeto que recibe las acciones y las envía a los Stores correspondientes.
- Stores: son objetos que contienen el estado de la aplicación y la lógica de negocio.
- Vista : es la interfaz de usuario de la aplicación.

Flux es especialmente útil en aplicaciones web de una sola página que manejan grandes cantidades de datos y estados complejos. Es adecuado para proyectos que requieren una estructura clara y predecible para manejar el flujo de datos. Muchos frameworks y bibliotecas, como Redux y Fluxible, se basan en el patrón Flux para proporcionar una solución más completa y fácil de usar.

Mobx

La librería MobX es una herramienta utilizada en el desarrollo de aplicaciones web, especialmente en el contexto de React, para gestionar el estado de la aplicación de manera eficiente. Se basa en el principio de observables y reacciones para actualizar automáticamente las partes relevantes de la interfaz de usuario cuando cambia el estado de la aplicación.

El objetivo principal de MobX es proporcionar una forma sencilla y eficiente de manejar el estado de la aplicación en aplicaciones web. Se basa en el principio de observables y reacciones para actualizar automáticamente la interfaz de usuario en función de los cambios en el estado.

Los principales conceptos de MobX son:

- **Observables**: Son objetos que pueden ser observados para detectar cambios en sus propiedades.
- **Acciones**: Son funciones que modifican el estado de los observables de manera controlada.
- **Reacciones**: Son funciones que se ejecutan automáticamente cuando cambia un observable que están observando.
- **Computed Values**: Son valores derivados de los observables que se recalculan automáticamente cuando alguno de los observables que dependen de ellos cambia.

MobX es adecuado para proyectos de todos los tamaños que requieren un manejo eficiente del estado de la aplicación. Es especialmente útil en aplicaciones web que necesitan una gestión flexible del estado y una actualización reactiva de la interfaz de usuario. Muchos desarrolladores prefieren MobX por su simplicidad y su enfoque más flexible en comparación con Redux.

Redux

Redux es un patrón orientado a la arquitectura de datos con el que podemos manejar el estado de nuestra aplicación de una manera sencilla y muy predecible.

Nace en la comunidad de React como una mejora a las ideas desarrolladas por Flux, actualmente es un patrón transversal adaptable a todo tipo de librería o Framework JavaScript (Jsfx). Esto le permite ser ejecutado en el lado del servidor o incluso en aplicaciones para dispositivos móviles.

El objetivo principal de Redux es proporcionar una forma estructurada y predecible de manejar el estado de la aplicación en aplicaciones de una sola página. Busca centralizar el estado de la aplicación en un store inmutable y facilitar la actualización de la interfaz de usuario en respuesta a cambios en el estado.

Diagrama con el flujo que sigue Redux desde una interacción, hasta que la aplicación actualiza la UI:

1. El **componente** recibe un evento (click, por ejemplo) y emite una acción.
2. Esta **acción**, se pasa a la *store*, que es donde se guarda el estado único.
3. La ***store*** comunica la acción junto con el estado actual a los reducers.
4. Los **reducers**, devuelven un nuevo estado, probablemente modificado en base a la acción.
5. Los componentes reciben el **nuevo estado** de la *store*.

Redux es ampliamente utilizado en el desarrollo de aplicaciones web, especialmente en combinación con React, debido a su simplicidad y eficiencia. Es adecuado para proyectos de todos los tamaños que requieren un manejo estructurado y centralizado del estado de la aplicación. Muchas empresas y proyectos de código abierto utilizan Redux para desarrollar aplicaciones web complejas y escalables.

Bibliografía

- Flux
- Redux
- Mobx