PRÁCTICA 6 DISEÑO DE INTERFACES

Miguel Ángel Salcedo Guijarro 2ºDAW

ÍNDICE

1.- Patrón de software Flux.

- Flux es un patrón de arquitectura software, utilizado normalmente en aplicaciones basadas en **React**, usado para **facilitar el mantenimiento y escalabilidad de aplicaciones web y móviles.**

1.1.- Origen.

- La primera idea de crear Flux surgió a principios del 2010. **Facebook**, por ese entonces, **estaba adaptando su web a React** como su biblioteca principal para construir interfaces de usuario. Cada vez, **las aplicaciones que tenían que desarrollar se volvían más complejas**, lo que supuso un problema de estados en la aplicación. Para solucionar este problema, crearon un patrón, Flux, que se acabó por desarrollar alrededor de 2014.

1.2.- Objetivo.

- El principal objetivo fue **facilitar la escalabilidad de las aplicaciones que tuvieran gran cantidad de componentes interactivos**. Es decir, tener una arquitectura clara para manejar el estado de las aplicaciones basadas en React, consiguiendo así un desarrollo **más eficiente, mantenible y escalable.**

1.3.- Funcionamiento.

- Flux consta de cuatro partes principales: **la vista, las acciones, el despachador y los despachadores.**

- **Vistas (View)** → La interfaz del usuario, donde ocurren las interacciones. La vista es la encargada de envíar las acciones del usuario al despachador.
- **Acciones (Actions)** → Objetos que describen un cambio en la aplicación. Son creados, normalmente, por la **"Vista"**. Contiene los datos de la propia acción y datos para poder realizarla.
- **Despachador** → Actúa como un centro de distribución de acciones, es decir, recibe las acciones para posteriormente enviarlas a los **"Almacenes"**. Garantiza que las acciones se produzcan de manera ordenada y secuencial.
- **Almacenes** → Contienen y manejan el estado de las aplicaciones. Responden a ciertos tipos de acciones y actualiza su estado en consecuencia, enviando un evento para modificar las **"Vistas"** que lo requieran.

- En resumen, las acciones son enviadas al despachador, que posteriormente son enviadas a los almacenes, encargados de actualizar la vista de la aplicación.

2.- Redux

- Librería de estado de aplicaciones JavaScript. Comúnmente **usado en el desarrollo de aplicaciones basadas en React.**

2.1.- Origen.

- Fue **creado por Dan Abramov y Andrew Clark** en 2015. Surgió para dar una solución a los estados de las aplicaciones basadas en React, para que fueran **más predecibles y eficientes.** Cuando Dan y Andrew estaban trabajando en proyectos de React, **observaron que se podía mejorar la claridad y la gestión de las aplicaciones.**

2.2.- Objetivo.

- El objetivo principal era **crear aplicaciones más predecibles y centralizadas** usando JavaScript en React.

- **Redux facilita el seguimiento de los cambios en el código**, facilitando también la depuración del mismo. Además, al tener un flujo de datos unidireccional, **el código es más fácil de entender y mantener.**

- En resumen, Redux proporciona una solución para manejar las aplicaciones de JavaScript, haciéndolas **más claras y más fáciles de escalar** en entornos basados en React.

2.3.- Funcionamiento.

- Redux se basa, como ya se ha mencionado en el punto anterior, en un **flujo de datos unidireccional**. Consta de tres partes principales: **Store, Reducers y Actions.**

- **Store** → Objeto que contiene el estado de toda la aplicación. Al crearse, se específica el Reducer principal que define cómo se actualiza el estado, en función de las acciones.
- **Reducers** → Funciones que especifican cómo cambiará el estado de la aplicación en respuesta a las acciones enviadas a través de la aplicación.
- **Actions** → Objetos que describen un cambio en la aplicación. Son creadas por las vistas, utilizando el método “dispatch()”.

- En conclusión, **las acciones** de la aplicación **se envían al “Store”**, donde se pasan al Reducer principal. Este último determina **cómo actualizar el estado de la aplicación en función de la acción enviada.** Por último, el **“Store”** reemplaza el estado anterior por el nuevo, **notificando a las partes necesarias de la aplicación, y desencadenando,** generalmente, **la actualización de las vistas.**

3.- Mobx

- **Biblioteca para gestionar los estados de las aplicaciones creadas con JavaScript.** Utilizado en el desarrollo de aplicaciones basadas en React. Es una solución para manejar el estado de las aplicaciones.

3.1.- Origen.

- **Creado por Michel Weststrate** y lanzado en 2015. Lo desarrolló como **respuesta a la complejidad de algunos desafíos que mostraba trabajar con JavaScript.** Quería simplificar la gestión de las aplicaciones, reduciendo su código.

- Se empezó a usar en numerosas empresas, debido a que es **fácil de usar pero eficiente**, permitiendo crear aplicaciones en JavaScript de manera más sencilla.

3.2.- Objetivo.

- Su principal objetivo, visto ya en el punto anterior, era c**rear aplicaciones JavaScript que fueran más sencillas de implementar con un código reducido**, haciendo uso de React.

- Dentro de los objetivos que tenía Michel al crear Mobx, era que fuera **simple, reactivo, escalable y flexible**, permitiendo a los desarrolladores crear aplicaciones más accesibles.

3.3.- Funcionamiento

- Basado en tres conceptos: **observables, acciones y reacciones.**
    - **Observables** → Objetos que representan el estado de la aplicación. Observados por las partes de la aplicación que necesiten reaccionar a cambios. Cuando un observable cambia, todas las reacciones asociadas se activan.
    - **Acciones** → Funciones que modifican el estado de los observables. Pueden ser síncronas o asíncronas.
    - **Reacciones** → Funciones que se ejecutan automáticamente cuando los observables asociados cambian. Actualizan la interfaz de usuario.

- Resumidamente, **cuando un observable cambia**, todas las reacciones asociadas se activan automáticamente, garantizando que **toda la interfaz de usuario refleje el estado de la aplicación sin gestionar manualmente las actualizaciones.**

Bibliografía

- Flux →
https://medium.com/wix-engineering/the-rise-of-flux-how-facebooks-shift-away-from-mvc-led-to-a-new-era-of-ui-architecture-61d78b4377b0

- Redux →
https://es.wikipedia.org/wiki/Redux_(JavaScript)
https://dev.to/fosteman/the-little-history-of-redux-3p1m

- Mobx →
https://mobx.js.org/README.html