

# **Práctica evaluable 6: Flux, Mobx y Redux.**

Daniel José Tercero Montero



<b>Práctica evaluable 6: Flux, Mobx y Redux.</b>	<b>1</b>
<b>1.- Patrón de software Flux.</b>	<b>2</b>
Origen.	2
Objetivo.	2
Funcionamiento.	3
Aplicabilidad.	3
<b>2.- Redux.</b>	<b>3</b>
Origen.	3
Objetivo.	4
Funcionamiento.	4
Aplicabilidad.	4
<b>3.- Mobx.</b>	<b>4</b>
Origen.	4
Objetivo.	4
Funcionamiento.	5
Aplicabilidad.	5
<b>Bibliografía</b>	<b>5</b>
<b>Enlaces</b>	<b>5</b>

## 1.- Patrón de software Flux.

### Origen.

Flux nació desde Facebook por un problema que se les presentaba al tener una comunicación bidireccional entre los modelos y los controladores, haciéndoles muy difícil poder depurar y rastrear errores.

### Objetivo.

El patrón de diseño Flux es un patrón de arquitectura de software que se utiliza para construir aplicaciones web de una sola página (Single-Page Applications). 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.

## Funcionamiento.

El patrón Flux consta de cuatro elementos principales:

- **Acciones (Actions):** 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 (View):** es la interfaz de usuario de la aplicación.

El flujo de datos en el patrón Flux se realiza en una sola dirección, lo que significa que los datos fluyen desde las acciones hacia los Stores y luego a la Vista. Cuando un usuario realiza una acción en la Vista, esta acción se envía al Dispatcher, que la envía a los Stores correspondientes. Los Stores actualizan su estado y emiten un evento para indicar que su estado ha cambiado. La Vista escucha estos eventos y actualiza su interfaz de usuario en consecuencia.

## Aplicabilidad.

Con el patrón de diseño Flux podremos construir aplicaciones web más escalables y mantenibles al proporcionar una estructura clara para el flujo de datos en la aplicación.

## 2.- Redux.

### Origen.

El origen de Redux se basa en las ideas de la arquitectura Flux, que también fue desarrollada por Facebook para abordar los desafíos de manejar el estado en aplicaciones de React. Redux simplifica y unifica aún más este enfoque, proporcionando un contenedor



de estado predecible y unidireccional que facilita la gestión del estado de la aplicación y su flujo de datos.

## **Objetivo.**

Redux es en esencia una librería de JavaScript que aplica y se basa en el patrón de diseño flux, cuya función principal es la de ayudarnos a estructurar el manejo del estado de nuestra aplicación

## **Funcionamiento.**

El funcionamiento final de cara al usuario de este patrón, se podría resumir en la aplicación de acciones, que desencadenan cambios en el estado de la aplicación.

## **Aplicabilidad.**

Redux sirve para controlar el estado de las aplicaciones web fácilmente de una forma consistente entre el cliente y el servidor.

# **3.- Mobx.**

## **Origen.**

Inventada por Michel Weststrate en 2015. Se implementa en los principios de la programación reactiva funcional (FRP) y permite envolver los estados en objetos "observables". Esto facilita la reacción ante cualquier cambio de estado de la aplicación.

## **Objetivo.**

MobX es una biblioteca que logra un renderizado óptimo sin esfuerzo. Todos los cambios y usos de sus datos se rastrean en tiempo de ejecución, creando un árbol de dependencia que captura todas las relaciones entre el estado y la salida. Esto garantiza que los cálculos que dependen de su estado, como los componentes de React, se ejecuten solo cuando sea estrictamente necesario.



## Funcionamiento.

MobX se centra en derivar cualquier cosa del estado de la aplicación, incluidas, entre otras, las interfaces de usuario, la comunicación del servidor y la serialización de datos. Al igual que otras tecnologías, MobX funciona muy bien cuando se combina con React.

## Aplicabilidad.

Al igual que Redux, Mobx sirve para controlar el estado de las aplicaciones web fácilmente.

# Bibliografía

### Patrón de Software Flux:

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

[Enlace2](#)

### Redux:

[Origen](#)

[Objetivo y funcionamiento](#)

### Mobx

[Origen](#)

[Objetivo y funcionamiento](#)

# Enlaces

[Repositorio de GitHub.](#)

<https://practica6daniel.onrender.com>