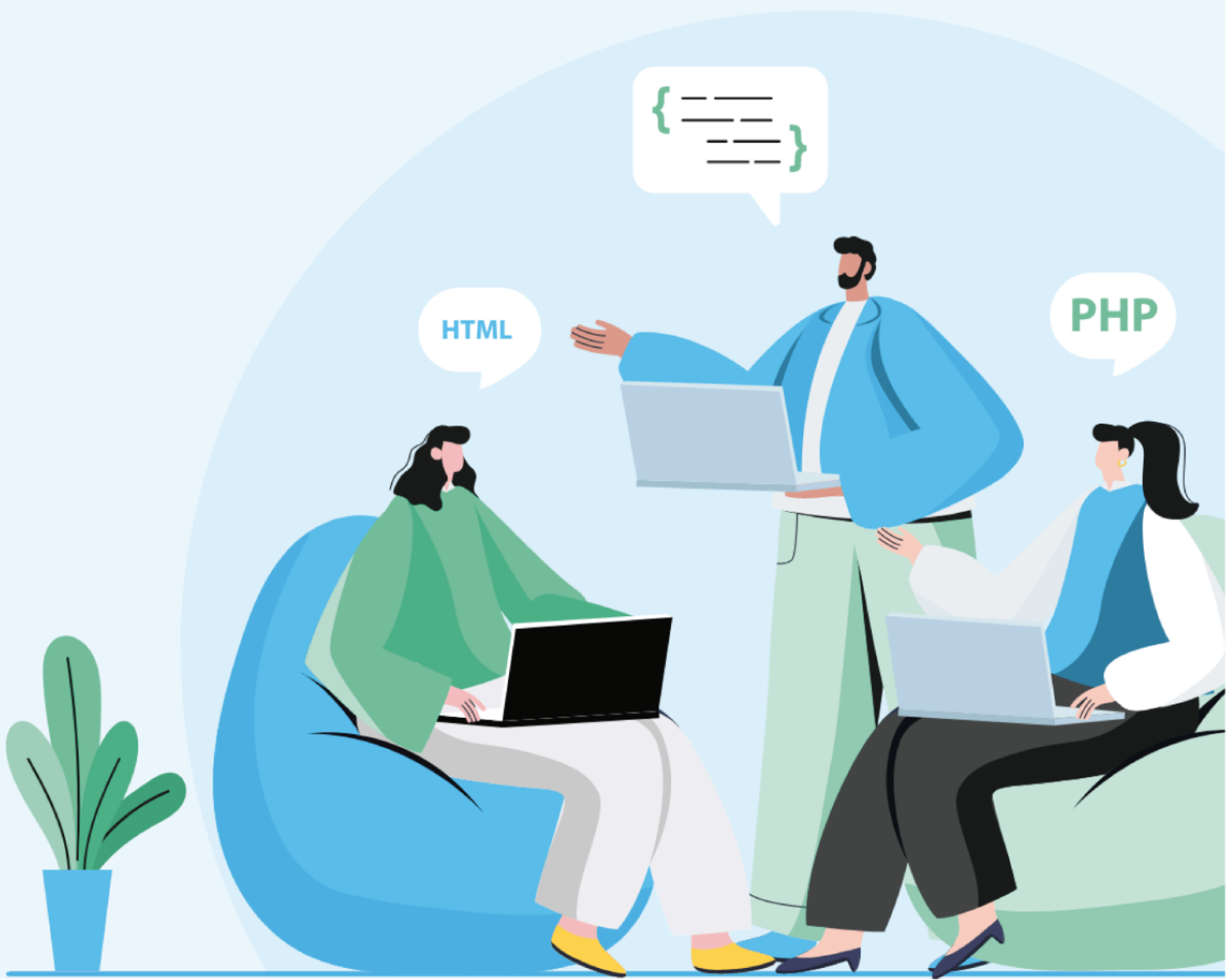


Skill Up Lab - React

Unidad 7





Unidad 7: Introducción a Redux

Antes de adentrarse en esta unidad

Para que puedas aprovechar mucho más el contenido presente en esta unidad, te recomendamos que antes de avanzar tengas muy en claro lo siguiente:

- cómo generar un sistema de búsqueda que impacte directamente a un endpoint de la API
- cómo generar mensajes de alerta en el escenario que la API no genere resultados de búsqueda
- cuál es el proceso más lógico para generar una sección dinámica de "*Favoritos*"
- dónde se debe guardar la información de la sección "*Favoritos*"
- la manera de compartir la información de la sección "*Favoritos*" con otros componentes de la aplicación

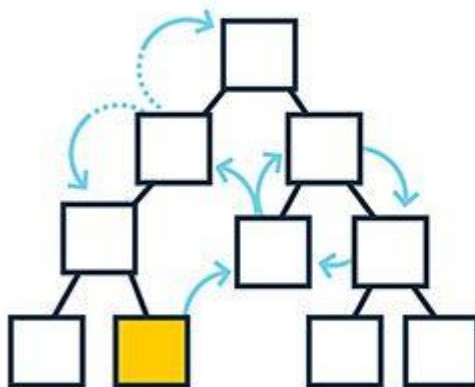
Objetivos de la unidad

Con el estudio de esta unidad buscamos que:

- Conozcas la librería Redux y entiendas el patrón de trabajo que ella propone
- Reconozcas la problemática presente en algunas aplicaciones de React y por qué Redux es una excelente solución a implementar
- Identifiques las ventajas de trabajar con Redux
- Asimiles los conceptos base y necesarios para usar esta librería
- Sepas analizar a conciencia en qué proyectos conviene usar Redux como solución



Clase 1: Entendiendo la problemática



¿Cual es el problema que Redux busca resolver?

Piensa que tienes varios componentes (cada uno hijo del otro) y que necesitas hacer llegar al último hijo de la cadena un state o una handler function que está presente en el componente principal. Aquí, seguramente lo más acertado sería pensar en pasar a través de las props aquello que necesitas.

Pues bien, esto es lo que vendría a resolver Redux. Evitar esa "escalera" de tránsito que puede llegar a ser conflictiva para su manejo.

Pero... ¿por qué no puede tener estas funcionalidades en el componente que las necesita y ya?

Cuando trabajamos con aplicaciones de React, es común que los states sean compartidos entre varios componentes. Y de tener dos componentes al mismo nivel (hermanos) hacer que el state presente en uno de ellos pueda ser compartido con el otro es una tarea algo compleja usando React "a secas". Por ello existen soluciones como esta librería.

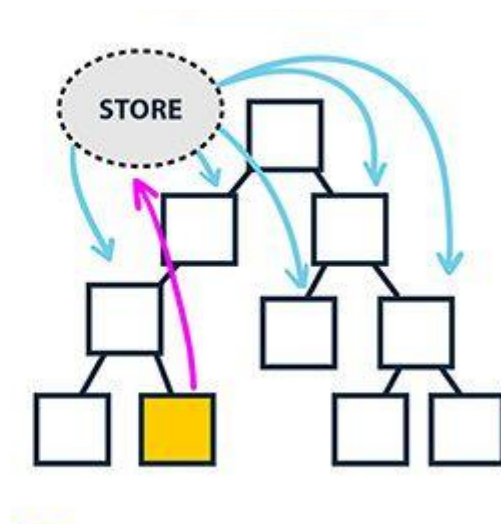


¿Es Redux usado realmente en la vida profesional o es algo que simplemente "vale la pena" conocer?

Redux es bastante usado en la industria de aplicaciones React. Así que saberlo implementar o por lo menos saber sus conceptos clave sumará mucho en tu camino como Dev.

Y aunque si bien existen algunas soluciones actuales creadas por la misma librería de React. Redux se ha consolidado como una solución óptima y casi un "must have" para cualquier persona que busque desenvolverse como React Developer.

Clase 2: Introducción y conceptos clave



¿Cuáles son los principios base de Redux?

Redux se basa en tres principios básicos:

1. Una sola fuente de la verdad

Redux propone que todo el estado de tu aplicación esté contenido en un único lugar llamado: "store", ya que esto facilitará usar la información allí guardada de una manera más fácil y sin mucho esfuerzo.

2. El estado es de solo lectura



Para Redux, la única forma de modificar el estado, es disparando "actions" que indiquen el cambio. Esto asegura que ninguna parte de la aplicación pueda alterar directamente el estado por error. Las "acciones" son objetos literales que describen cuál será el cambio a ejecutar.

3. Los cambios del estado se hacen mediante funciones puras

Para controlar cómo el "store" es modificado por las acciones, se usan "reducers", los cuales son funciones puras que reciben el estado actual de la aplicación y la acción que se debe realizar. Y retornan un nuevo estado, sin modificar directamente el estado actual.

¿Cuál es el flujo de una aplicación que usa Redux?

Todo parte desde el store, que es donde se almacena todo el state de la aplicación. El store tiene una línea de comunicación directa con los reducers, quienes se encargarán de realizar los cambios y setear el nuevo estado.

El store es quien se comunica con la vista o componente para mostrar estos cambios en pantalla. Y es aquí, en este último apartado donde la persona que visita la aplicación y a través de la interfaz gráfica, realiza interacciones (click, scroll, hover, etc) las cuales a su vez disparan (dispatch) acciones hacia el store para que sea éste a través de los reducers, quien pueda setear el nuevo state y así el flujo vuelva a comenzar de nuevo.

Como sabemos que esto así descrito es algo complejo de entender, aquí te dejamos una imagen que te permitirá visualizar el panorama de una manera más integral:

