

ReactJS Developer

Proyecto integrador - Etapa 5

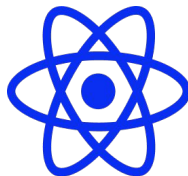


Etapa 5

El objetivo de esta etapa del proyecto es realizar el árbol de componentes.

Hemos llegado a un punto muy importante en nuestra aplicación. Como planteamos, vamos a rediseñar el árbol de componentes. Como pueden ver, nuestra aplicación funciona. Sin embargo, no es escalable y no tiene reactividad.

Ahora bien, no podemos lanzarnos así como así a ir dividiendo en componentes. Es importante parar de codificar y tomar un poco de perspectiva, ya que la decisión sobre cómo estructuraremos nuestro árbol de componentes conllevará resultados y consecuencias en el medio y largo plazo.



A grandes rasgos hay dos tipos de componentes:

- Aquellos que **renderizan directamente** una porción de la interfaz (**generan HTML**).
- Aquellos que **no renderizan directamente** una porción de la interfaz (**abstraen lógica**).

Vamos a analizar, página a página, qué componentes podemos separar:



1. Index

- NavBar (el componente de la barra de navegación)
- SearchBox (su evento input cambiará el parámetro 'q' del query de la URL para actualizar la búsqueda)
- ItemList (necesita el listado de items)
- Item

2. Details

- NavBar (el componente de la barra de navegación)
- BackButton
- ItemDetails (necesita un item)
- AddToCartButton

3. Cart

- NavBar (el componente de la barra de navegación)
- CartList (solo haremos el HTML, sin pensar en la lógica del carrito por ahora -CartItem
- ContinueButton
- CheckoutButton

Nota: Hay componentes que se repiten o repiten la mayor parte de su código de vista en vista (como *NavBar*), así como subcategorías de componentes (hay botones) y componentes que envuelven a otros (las listas envuelven a los ítems).

Entonces, vamos a dividir la interfaz en este árbol de componentes trabajando con otro tipo de componentes que nos permitan abstraer la lógica para, por ejemplo, crear un *hook* llamado *useSearchFilter* que obtenga el parámetro 'q' del query de la url y devuelva su valor, mejorando así el filtro por el campo de búsqueda.

Realizar los siguientes pasos:

1. La estructura de datos para cada ítem es la siguiente:

```
Item: {  
  id: number,  
  image: string,  
  title: string,  
  description: string,  
  sizes: string[],  
  colors: string[],  
  price: number  
}
```

2. La estructura de datos para cada item del carrito es la siguiente:

```
CartItem: {  
  productId: number,  
  size: string,  
  color: string,  
  count: number  
}
```

Nota: Recomendamos tener *mocks* tanto del listado de ítems como de los ítems actuales del carrito.

En clases futuras usaremos **Redux** y **Context** para gestionar el origen de los datos y la persistencia de estas estructuras de datos.

Recordá que en la sección de **Descargas** encontrarás los recursos necesarios para realizar los ejercicios y su resolución para que verifiques cómo te fue.



**¡Sigamos
trabajando!**

