

Introducción a Hooks ReactJS

Introducción

Los hooks son una colección de distintas funciones, cada una con un propósito muy claro y descriptivo y que están pensadas para ser utilizadas dentro de componentes funcionales, permitiéndonos no solo disponer de un manejo de state o control del ciclo de vida entre nuestros componentes funcionales si no que además, nos va a permitir reutilizar lógica entre nuestros componentes e incluso crear nuestros propios hooks.

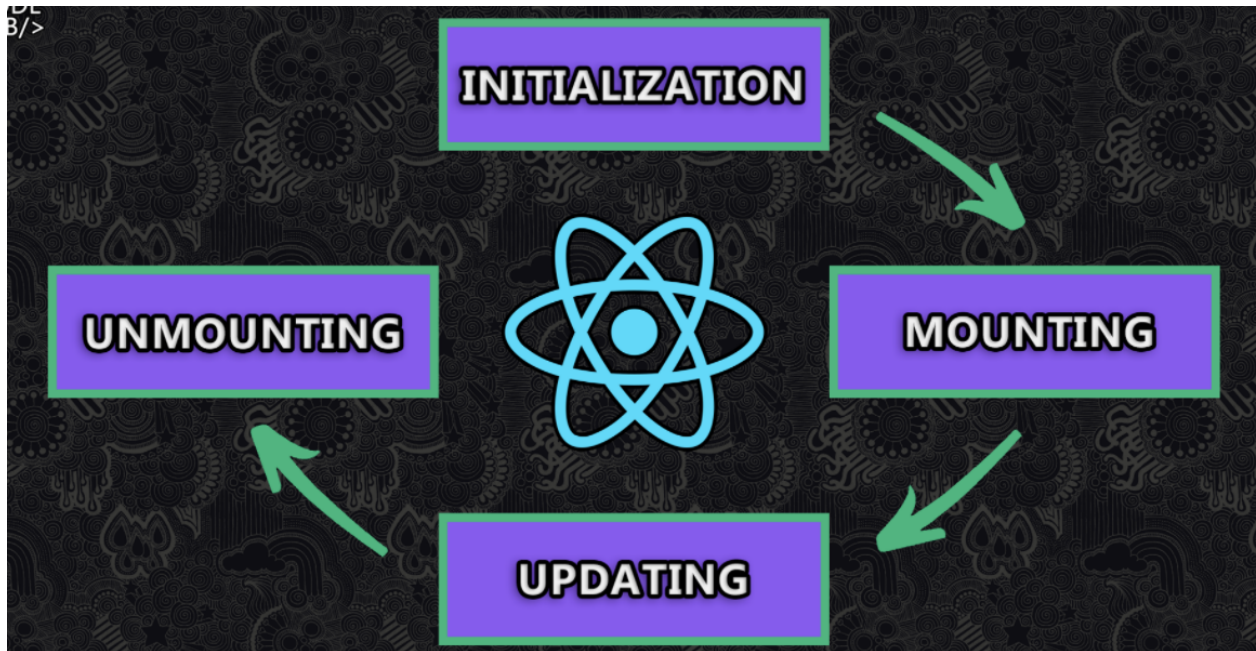
Los hooks siguen una convención de nombre que emplea **use** seguido de un descriptor para el hook, por lo que todos los hooks son useHookName, como por ejemplo useState que es probablemente el hook más importante.

Pero antes de continuar hablemos sobre el ciclo de vida de los componentes y así entender la necesidad de los Hooks de React.

Ciclo de vida de componentes

Todo suele seguir un ciclo, por ejemplo, los humanos nacemos, crecemos y morimos. Los componentes de React también se rigen por un ciclo parecido.

Los componentes se crean (se montan en el DOM), se actualizan ("crecen"), y mueren (se liberan del DOM). A esto lo denominamos **ciclo de vida de un componente**.



Concepto

Los Hooks nos ayudan a interactuar en el ciclo de vida del componente transformando la información e interactuando con el usuario. La propia librería nos provee de estas funcionalidades y aunque ya conocemos algunas como el `useState` o `useEffect` vamos a ir viendo sus características y uso poco a poco.