



Front End III

¿Qué es useState?

En React, la interfaz de usuario representa el estado en el momento de renderización inicial. Si alguna pieza de ese estado actualiza su valor, React volverá a renderizar los componentes involucrados con dicha pieza de estado tras hacer una comparación entre el virtual DOM y el DOM guardado en memoria. El objetivo de esto es que la actualización sea eficiente y basada en la diferencia entre ambos DOM.

Desde el lanzamiento de Hooks, los componentes funcionales pueden declarar y actualizar su estado interno mediante el hook **useState**. Este es una función que crea una variable que permite almacenar una pieza de estado interno y, a la vez, actualizar dicho valor en un componente funcional.

```
const [state, setState] = useState("Learning");
```

Cada componente tiene asociado una lista interna de "celdas de memoria". Estas celdas de memoria se traducen en objetos literales de JavaScript en donde incluimos nuestros datos. Cuando invocamos useState, la función va leyendo las celdas una a una. De este modo, múltiples llamados a useState obtienen múltiples estados locales independientes.

En teoría, esto nos permitiría declarar infinitas const de useState.

```
const [count, setCount] = useState(0);  
const [phone, setPhone] = useState("");  
const [username, setUsername] = useState("");  
const [email, setEmail] = useState("");  
const [password, setPassword] = useState("");
```

```
const [confirmPassword, setConfirmPassword] = useState("");
const [otaku, setOtaku] = useState(false);
const [loveLevel, setLoveLevel] = useState({day: "orange", night:
"blue"});
```

Podemos incluir useState las veces que necesitemos, la única condición es que sea llamada desde un nivel superior de código, no en un bloque.

```
const Post = () => {
  const estoEstaMal = () => {
    const [likes, setLikes] = useState(0);      (incorrecto)
  }
  return <h3>{estoEstaMal}</h3>;
};
```

```
const Post = () => {
  const [likes, setLikes] = useState(0);      (Correcto)
  return <h3>{likes}</h3>;
};
```