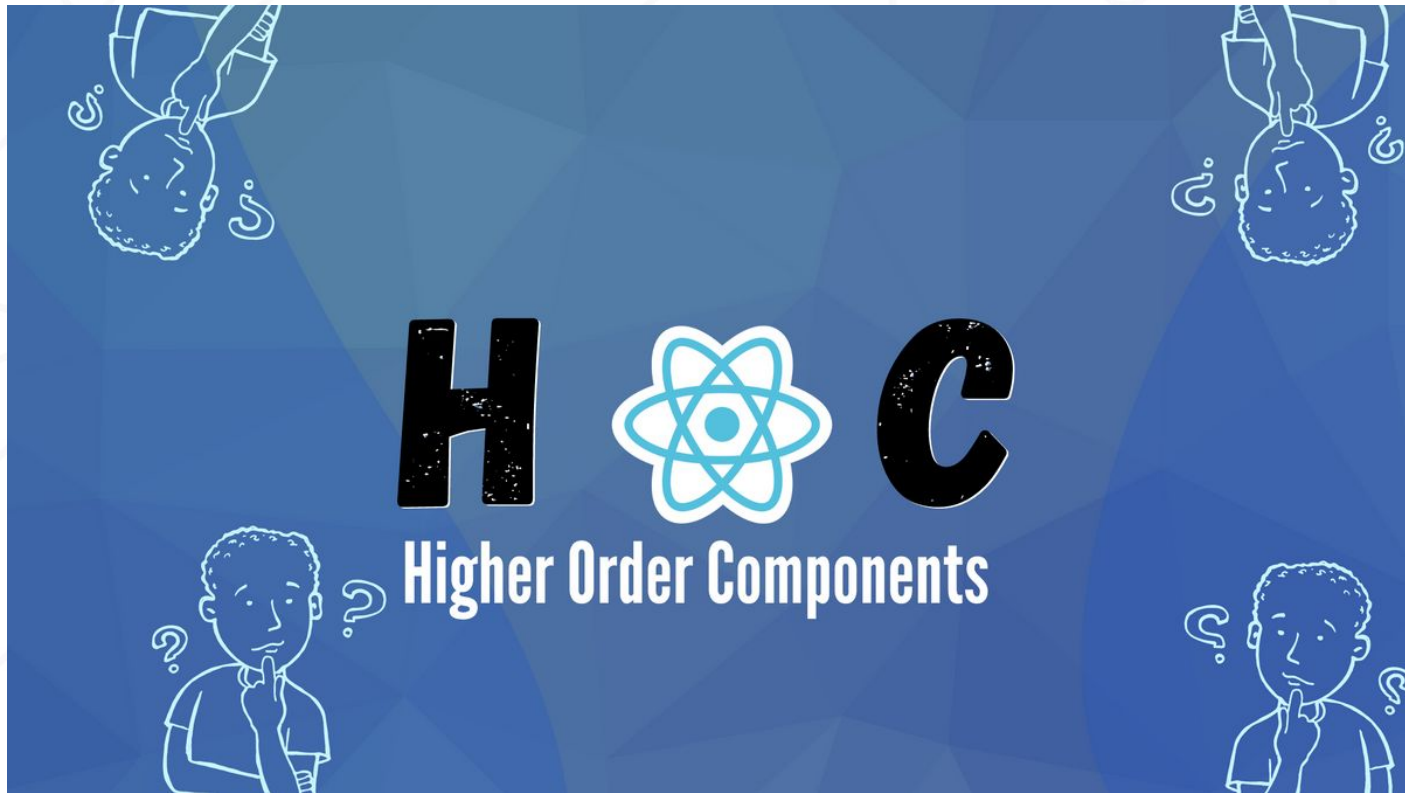
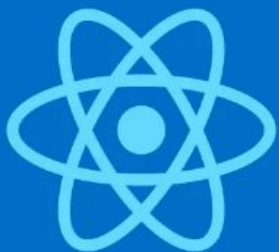


# React Higher Order Components

**DEV.F**  
DESARROLLAMOS(PERSONAS);

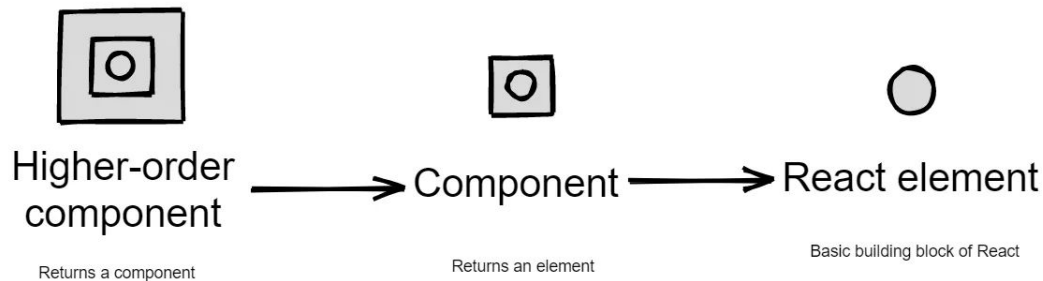
dev





## Higher-Order Components

Reutilizar código es una de las principales acciones que debemos realizar para mejorar la calidad de nuestro código.



Uno de los aspectos en los cuales más nos obsesionamos los desarrolladores al momento de escribir código, es construir código que pueda ser reutilizable a lo largo de nuestro proyecto, lo cual reduce la complejidad de nuestro proyecto y aumenta nuestra productividad como desarrolladores.



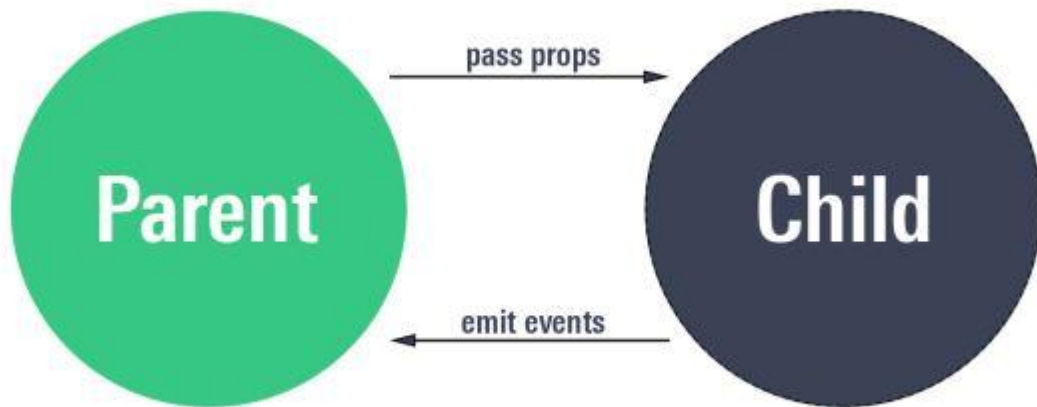
Los componentes de React están diseñados para ser reutilizables: deben tener un propósito y solo **contener la lógica necesaria para ese propósito**. Los HOC nos permiten cumplir con estas expectativas.

En concreto, un **componente de orden superior** es una función que recibe un componente y devuelve un nuevo componente.

```
const ComponenteMejorado = hoc(ComponenteOriginal);
```

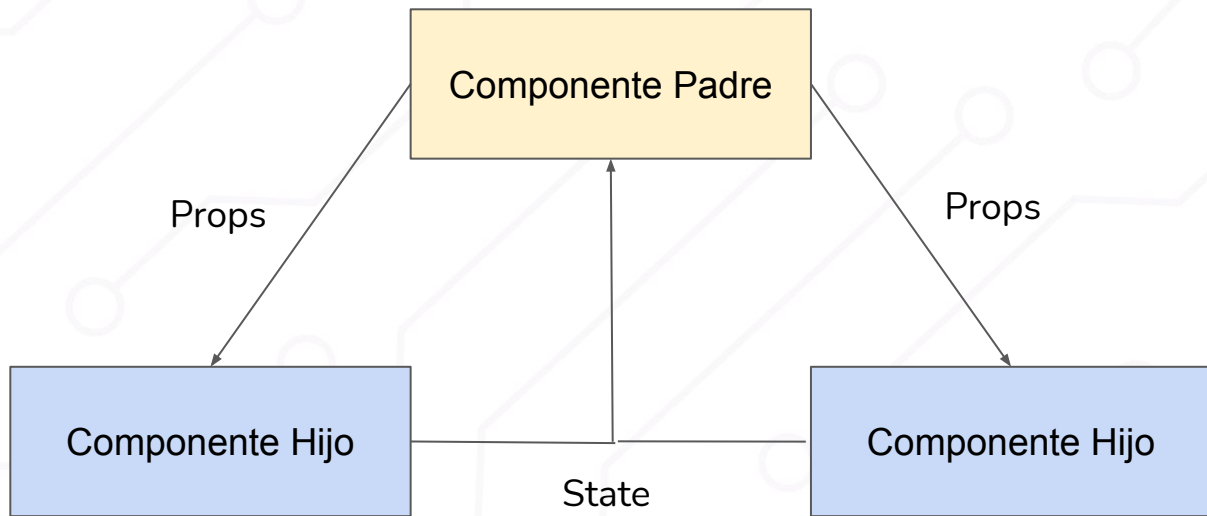
Los **HOC** suelen tener el prefijo (pero no de forma obligatoria) **with** o **get**. **withNombreComponente** se espera que inyecte funcionalidad mientras que se espera que **getNombreComponente** inyecte datos en el componente original.

# Props

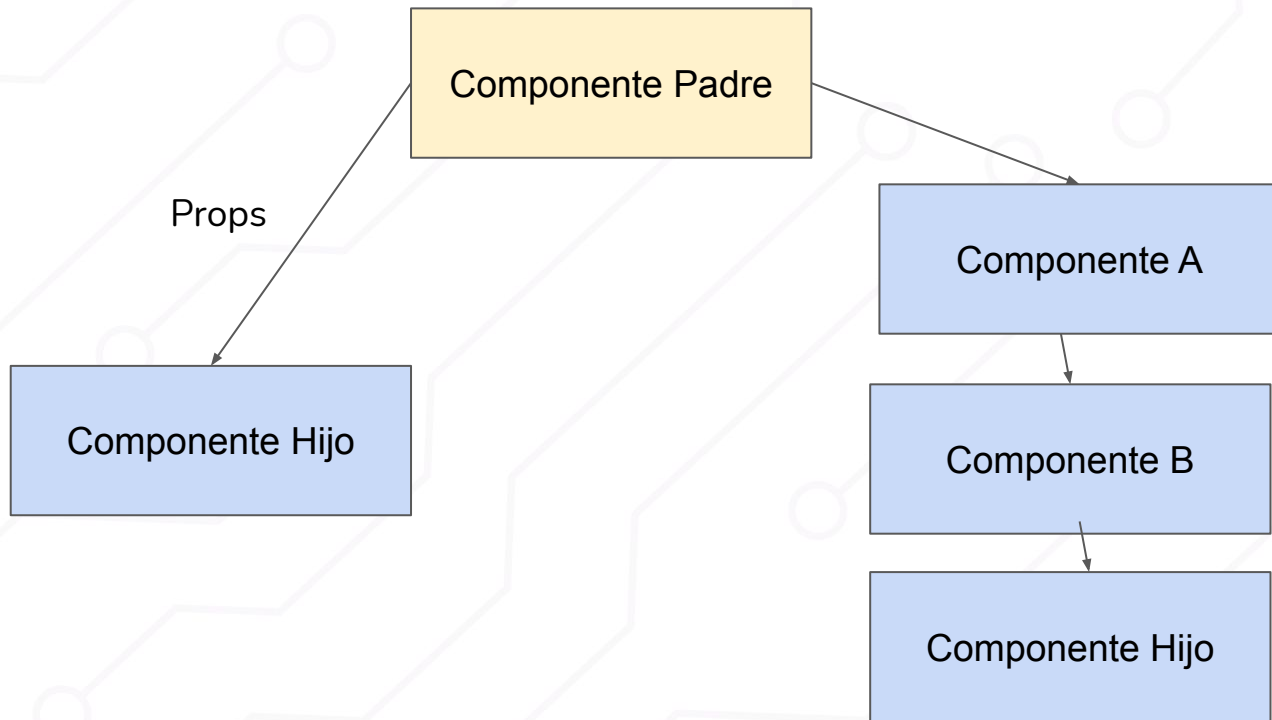




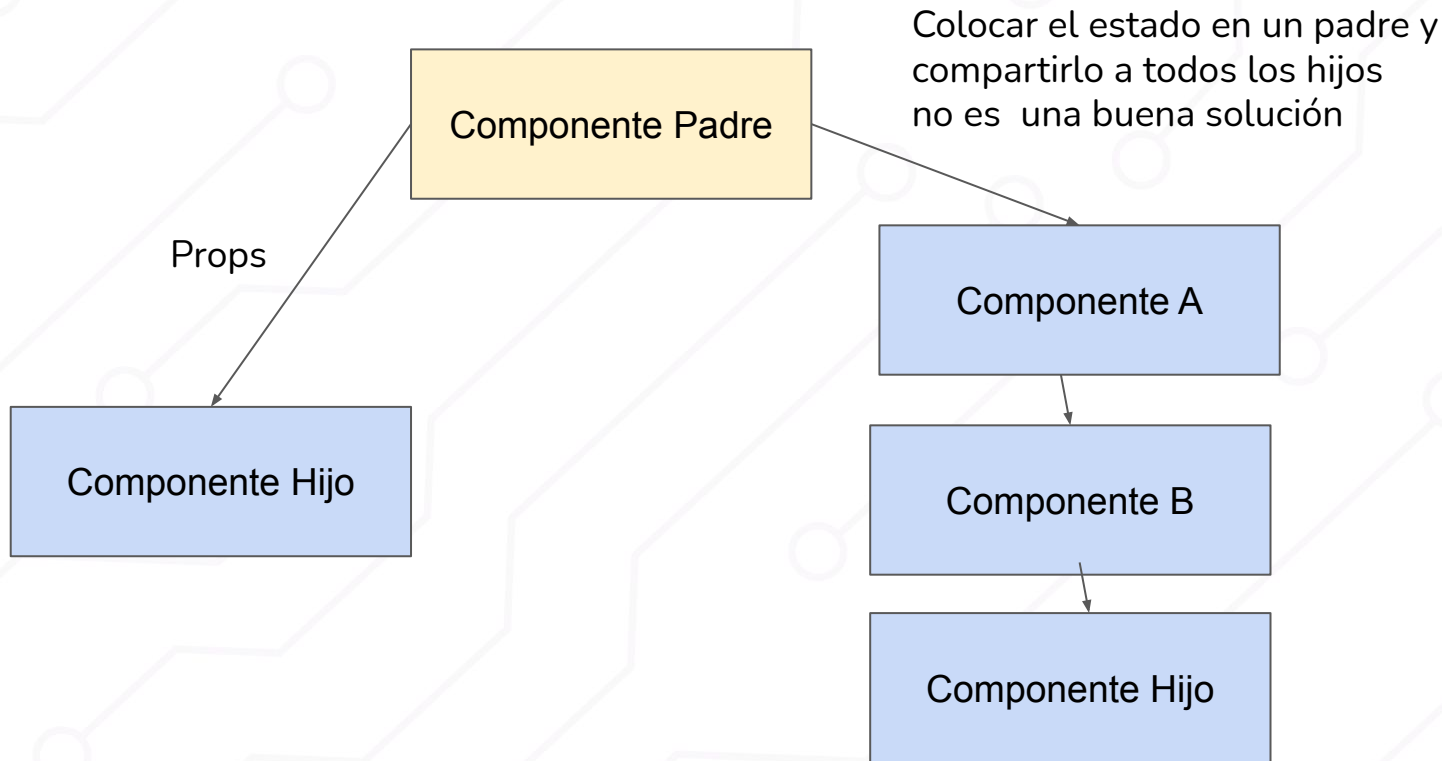
# Árbol de Componentes



# Árbol de Componentes



# Árbol de Componentes



# ¿Por qué utilizar Higher order Components?

Es un patrón donde **una función recibe un componente como argumento** y retorna un nuevo componente, o un componente *mejorado*.

```
// Tomemos un componente
const componenteOriginal = () => (<h1>Hola Mundo</h1>);

// Y pasémoslo como argumento de una función para obtener...
const componenteNuevo = higherOrderComponent(componenteOriginal); // Un nuevo componente
const componenteMejorado = higherOrderComponent(componenteOriginal); // Un componente mejorado

// Ejemplo
const IronMan = withSuit( TonyStark );
```