



Curso de
React.js:
Patrones de Render
y Composición



JuanDC



Mejoremos
tu código React





**Mejoremos
tu código React**



Principios de Diseño en React

Filosofía React para el Team Facebook

Framework o librería

- Abstracciones comunes
- Interoperabilidad

Cambios al core

- Estabilidad
- Válvulas de escape
- Experiencia de desarrollo

Prioridades

- Implementación
- Optimizado para instrumentación
- Dogfooding



Filosofía React
para ti

El trabajo de React

- Planificación
- Configuración

Tu trabajo

- Depuración
- Compocisión



Composición de componentes

```
const App = () => (
  <TodoList todos={todos} />
);
```

```
const TodoList = ({ todos }) => (  
  <section>  
    {todos.map(todo => (  
      <TodoItem {...todo} />  
    ))}  
  </section>  
) ;
```

```
const TodoList = ({ children }) => (  
  <section>  
    {children}  
  </section>  
)
```

```
const App = () => (
  <TodoList>
    {todos.map(todo => (
      <TodoItem {...todo} />
    )));
  </TodoList>
);
```



Colocación del estado



¿Dónde va tu estado?

- Máxima cercanía
a la relevancia.
- Stateful vs. stateless.

```
const App = () => (  
  <>  
  <TodoHeader />  
  <TodoList />  
);
```



```
const App = () => (
```

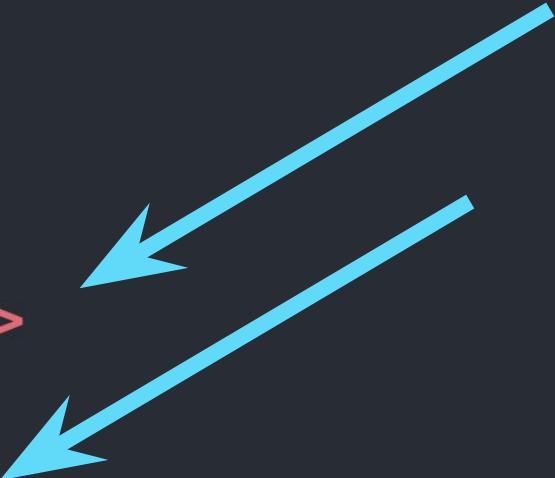
```
<>
```

```
<TodoHeader />
```

```
<TodoList />
```

```
</>
```

```
);
```



```
const App = () => (
```

```
<>
```

```
<TodoHeader />
```

```
<TodoList />
```

```
</>
```

```
);
```

¿Dónde va tu estado?

- Máxima cercanía
a la relevancia
- Stateful vs. stateless



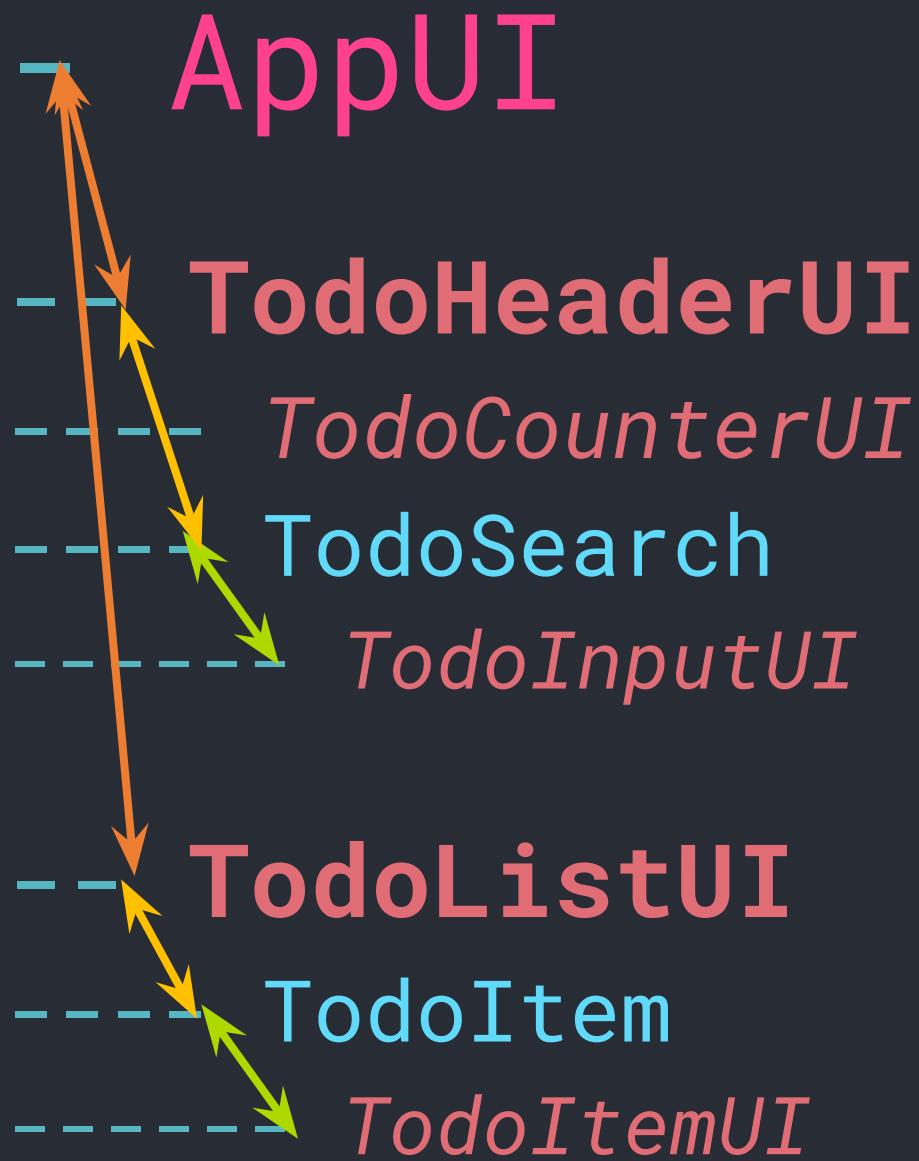
- App
- TodoHeader
- TodoList

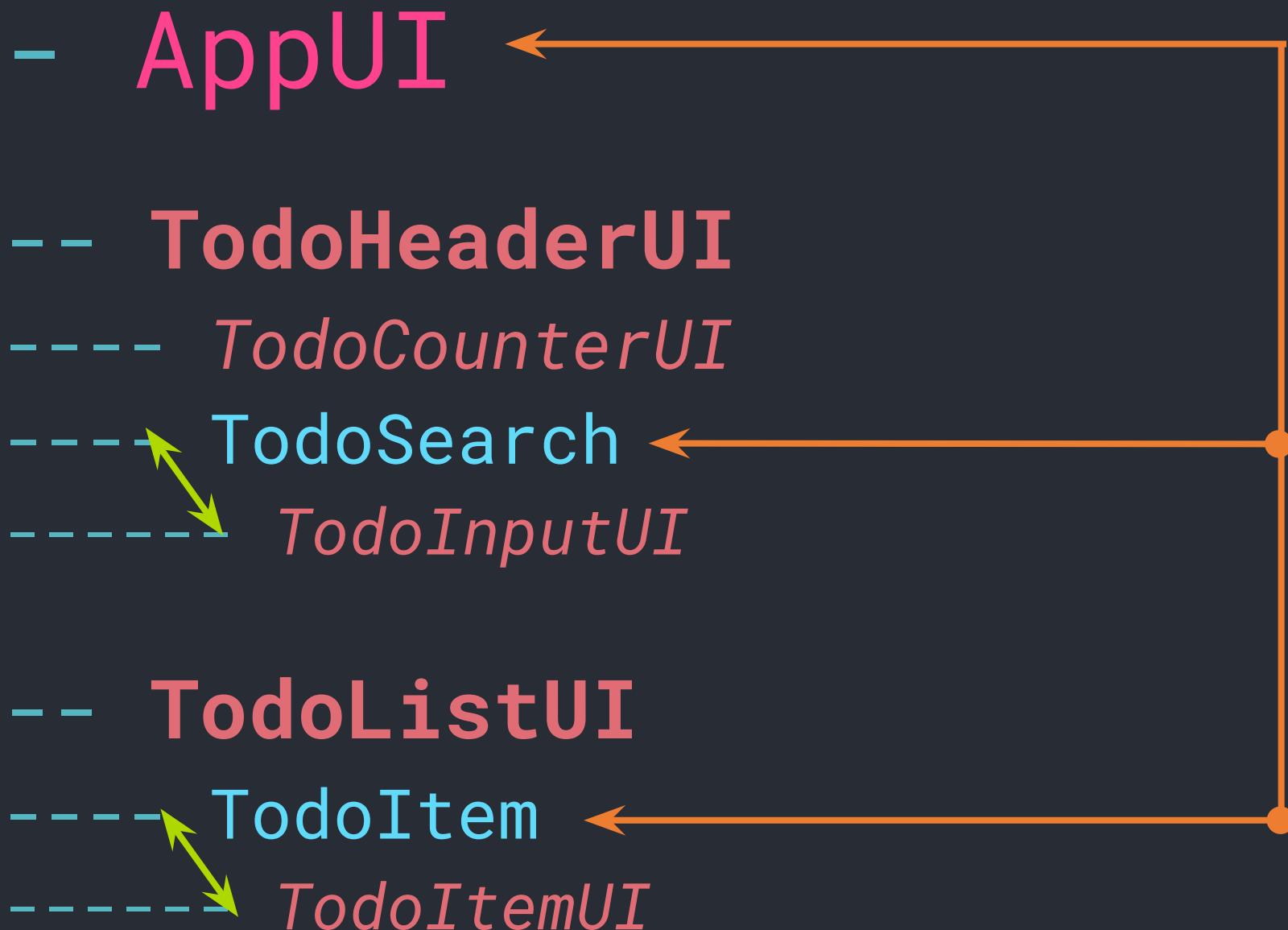
- App
- TodoHeader
 - TodoCounter
 - TodoSearch
- TodoList
 - TodoItem



- App
 - TodoHeaderUI
 - TodoCounterUI
 - TodoSearch
 - TodoInputUI
 - TodoListUI
 - TodoItem
 - TodoItemUI

- AppUI
 - TodoHeaderUI
 - TodoCounterUI
 - TodoSearch
 - TodoInputUI
- TodoListUI
 - TodoItem
 - TodoItemUI





- AppUI

-- TodoHeaderUI

----- TodoCounterUI

----- TodoSearch

----- TodoInputUI

-- TodoListUI

----- TodoItem

----- TodoItemUI

¿Necesitas

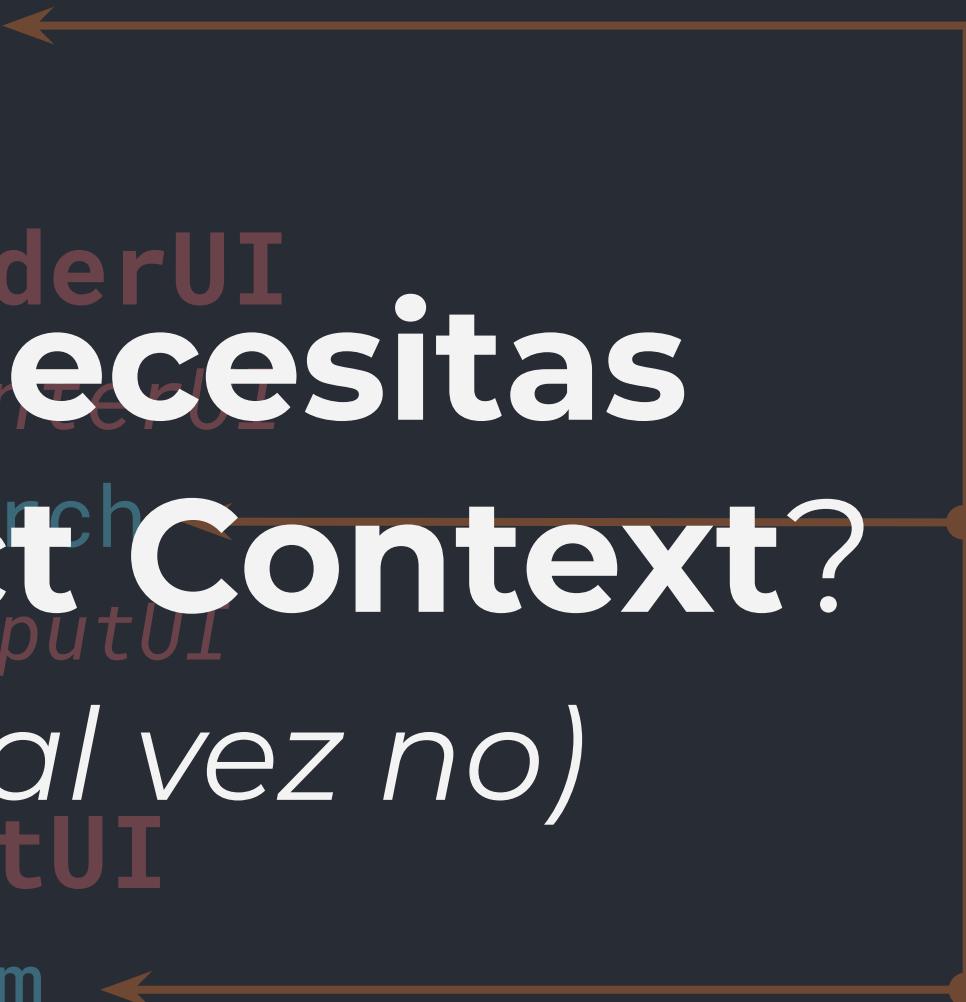
React Context?

(tal vez no)

TodoListUI

TodoItem

TodoItemUI



```
const App = () => (
  <TodoList todos={todos} />
);
```

```
const App = () => (
  <TodoList>
    {todos.map(todo => (
      <TodoItem {...todo} />
    )));
  </TodoList>
);
```

```
const App = () => {  
  const [state, setState] = React.useState();  
  
  return (  
    <>  
      <TodoHeader {...state} />  
      <TodoList {...state} />  
    </>  
  );  
}
```


- App



- App



-- TodoSomething

--- TodoSomethingUI

--- TodoWhatever

--- TodoWhateverUI

--- TodoAnythingComponent

----- TodoAnythingComponentUI

-- TodoSomething2

--- TodoSomethingUI2

--- TodoWhatever2

--- TodoWhateverUI2

----- TodoAnythingComponent2

----- TodoAnythingComponentUI2

-- TodoSomething3

--- TodoSomethingUI3

--- TodoWhatever3

--- TodoWhateverUI3

----- TodoAnythingComponent3

----- TodoAnythingComponentUI3



- AppUI

- TodoHeaderUI

- TodoCounterUI

- TodoSearch

- TodoInputUI

- TodoListUI

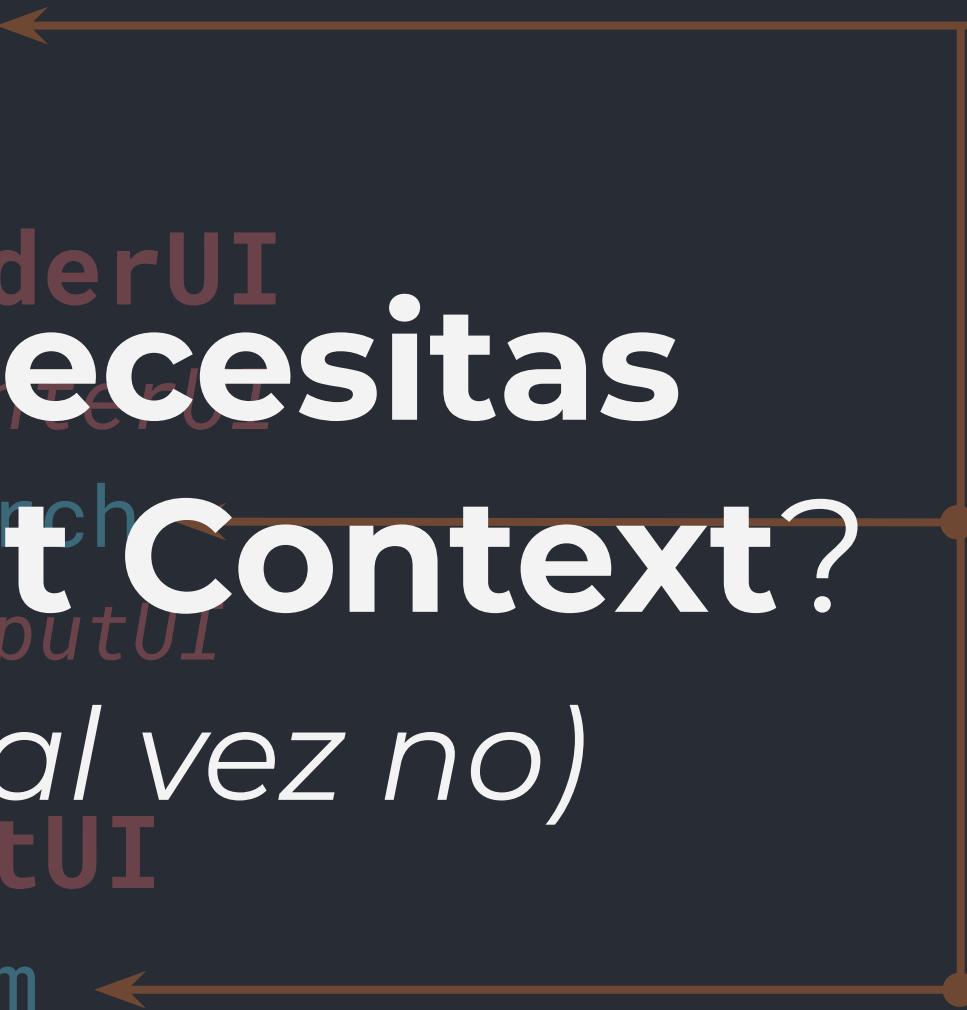
- TodoItem

- TodoItemUI

¿Necesitas

React Context?

(Tal vez no)





Composición con React Context



Composición sin React Context



Render props y render functions

```
<Provider value={...something}>  
  
  <Consumer>  
    {something => (  
      <TusComponentes  
        {...something.xyz}>  
      >  
    )}  
  </Consumer>  
</Provider>
```

```
<RenderFunction>
```

```
{info => (
```

```
  <OtroComponente {...info} />
```

```
)}
```

```
</RenderFunction>
```

```
<RenderProps
```

```
    renderProp={<0troCompo />}
```

/>

```
<RenderProps
```

```
    renderProp={info => <0troCompo {...info} />}
```

/>

```
<PlatziCourse  
    classes={platziClass => (  
        <PlatziClassPreview {...platziClass} />  
    )}  
  
    lastActivity={question => (  
        <PlatziComment q={true} {...question} />  
    )}  
  
/>
```

```
<PlatziClass
```

```
  video={videoUrl => (
```

```
    <PlatziVideoPlayer src={videoUrl} />
```

```
)}
```

```
  comments={comment => (
```

```
    <PlatziComment {...comment} />
```

```
)}
```

/>

```
<TodoList>

{error && <TodosError error={error} />}

{loading && <TodosLoading />}

{(!loading && !todos.length) && (
    <EmptyTodos />
) }

{todos.map(todo => <TodoItem {...todo} />)}

</TodoList>
```

```
<TodoList  
  onError={error => <TodosError error={error} />}  
  onLoading={() => <TodosLoading />}  
  onEmptyTodos={() => <EmptyTodos />}  
  render={todo => <TodoItem {...todo} />}  
/>
```



Continúa aprendiendo

¿Ya instalaste y configuraste Unreal Engine?

Curso de Introducción a Unreal Eng...

Etapas de un producto

Audiocurso Fundamentos de Product M...

Carrera como Product Manager

Curso de Product Management

Mis Rutas y Escuelas

Escuela de JavaScript

19 Cursos • 48% completada

Escuela de Platzi English Academy

40 Cursos • 10% completada

Escuela de Data Science

44 Cursos • 20% completada

Escuela de Matemáticas

19 Cursos • 24% completada

```
<PlatziHome
```

```
    lastClasses={classPreview => (
```

```
        <ClassPreview {...classPreview} />
```

```
)}
```

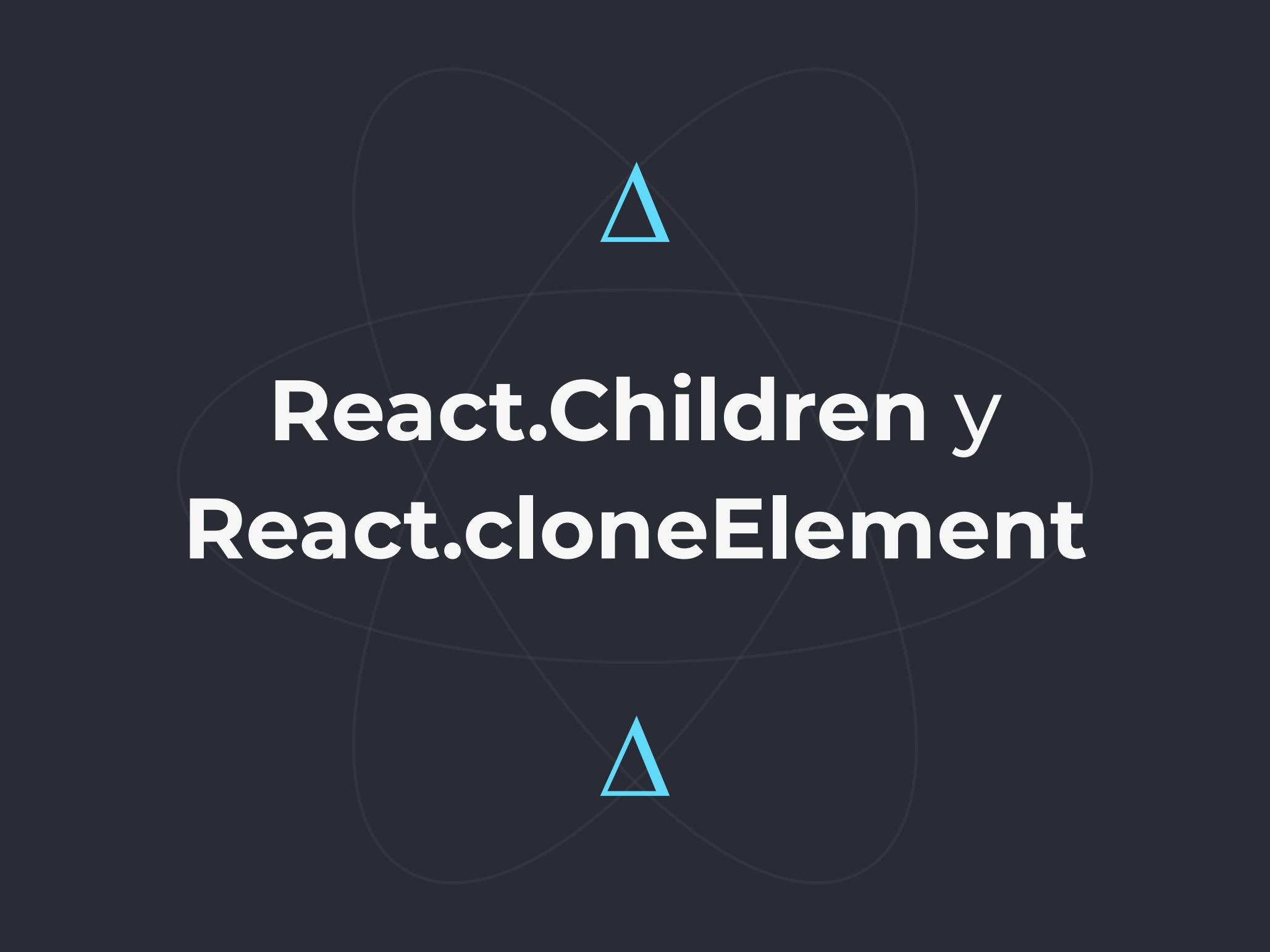
```
    learningPaths={lp => (
```

```
        <LearningPathPreview {...lp} />
```

```
)}
```

/>

```
<SuperFormMessage  
    onError={error => (  
        <FormError error={error} />  
    )}  
  
    onSuccess={newData => (  
        <FormSuccess data={newData} />  
    )}  
/>
```

**React.Children y
React.cloneElement**



High Order Components

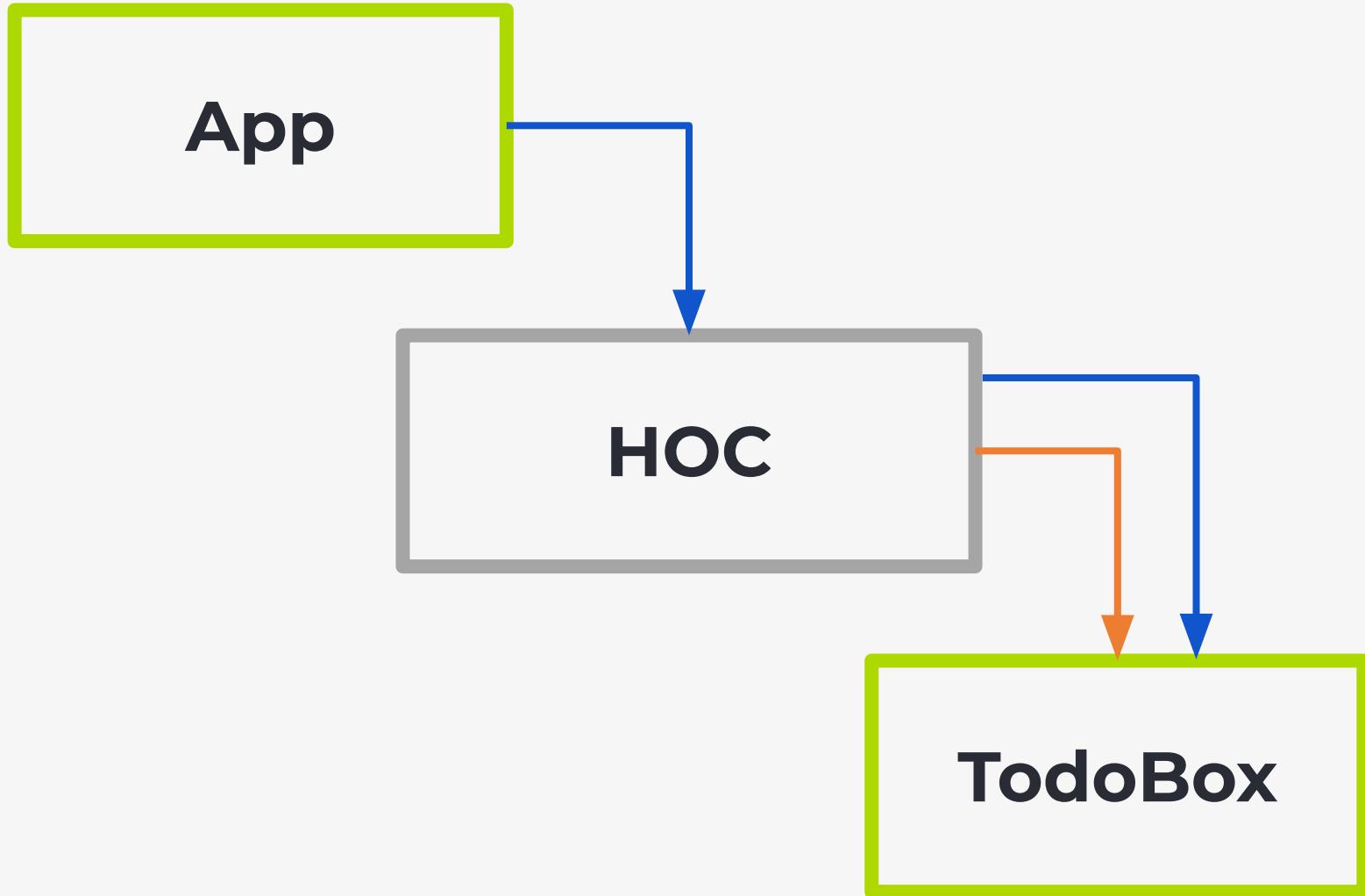
```
function highOrderFunction( ) {  
    return function returnFunction( ) {  
        return /* ... */  
    }  
}
```

```
function highOrderFunction(var1) {  
  return function returnFunction(var2) {  
    return var1 + var2;  
  }  
}  
  
const withSum1 = highOrderFunction(1);  
const sumTotal = withSum1(2);  
// 3
```

```
function Componente( props ) {  
  return <p>...</p>;  
}  
  
function highOrderComponent( ) {  
  return function Componente( props ) {  
    return <p>...</p>;  
  }  
}
```

```
function highOrderComponent(WrappedComponent) {  
  return function Componente(props) {  
    return (  
      <WrappedComponent  
        {...algoEspecial}  
        {...props}  
      />;  
    );  
  };  
}
```

```
function TodoBox(props) {  
  return (  
    <div prop={props.algoEspecial}>  
      {props.children}  
    </div>;  
  );  
  
const TodoBoxWithSpecialprops  
  = highOrderComponent(TodoBox);
```



```
function withApi(WrappedComponent) {  
  const apiData = fetchApi('https://api.com');  
  
  return function WrappedComponentWithApi(props) {  
    if (apiData.loading) return <p>Loading</p>;  
  
    return (  
      <WrappedComponent data={apiData.json} />  
    );  
  }  
}
```

```
function TodoBox(props) {  
  return (  
    <p>  
      Tu nombre es {props.data.name}  
    </p>;  
  );  
}  
  
const TodoBoxWithApi = withApi(TodoBox);
```

```
function withApi(apiUrl) {  
  return function withApiUrl(WrappedComponent) {  
    const apiData = fetchApi('https://api.com');  
  
    return function WrappedComponentWithApi(props) {  
      if (apiData.loading) return <p>Loading</p>;  
      return (  
        <WrappedComponent data={apiData.json} />  
      );  
    }  
  };  
}
```

```
function TodoBox( props ) {  
  return (  
    <p>  
      Tu nombre es {props.data.name}  
    </p>;  
  );  
  
}  
  
const TodoBoxWithApi  
  = withApi('https://api.com')(TodoBox);
```

```
const TodoBoxWithUser  
  = withRouter(TodoBox);
```

```
const TodoBoxWithFilters  
  = withFilters(TodoBoxWithUser);
```

```
const TodoBoxWithTodos  
  = withTodos(TodoBoxWithFilters);
```

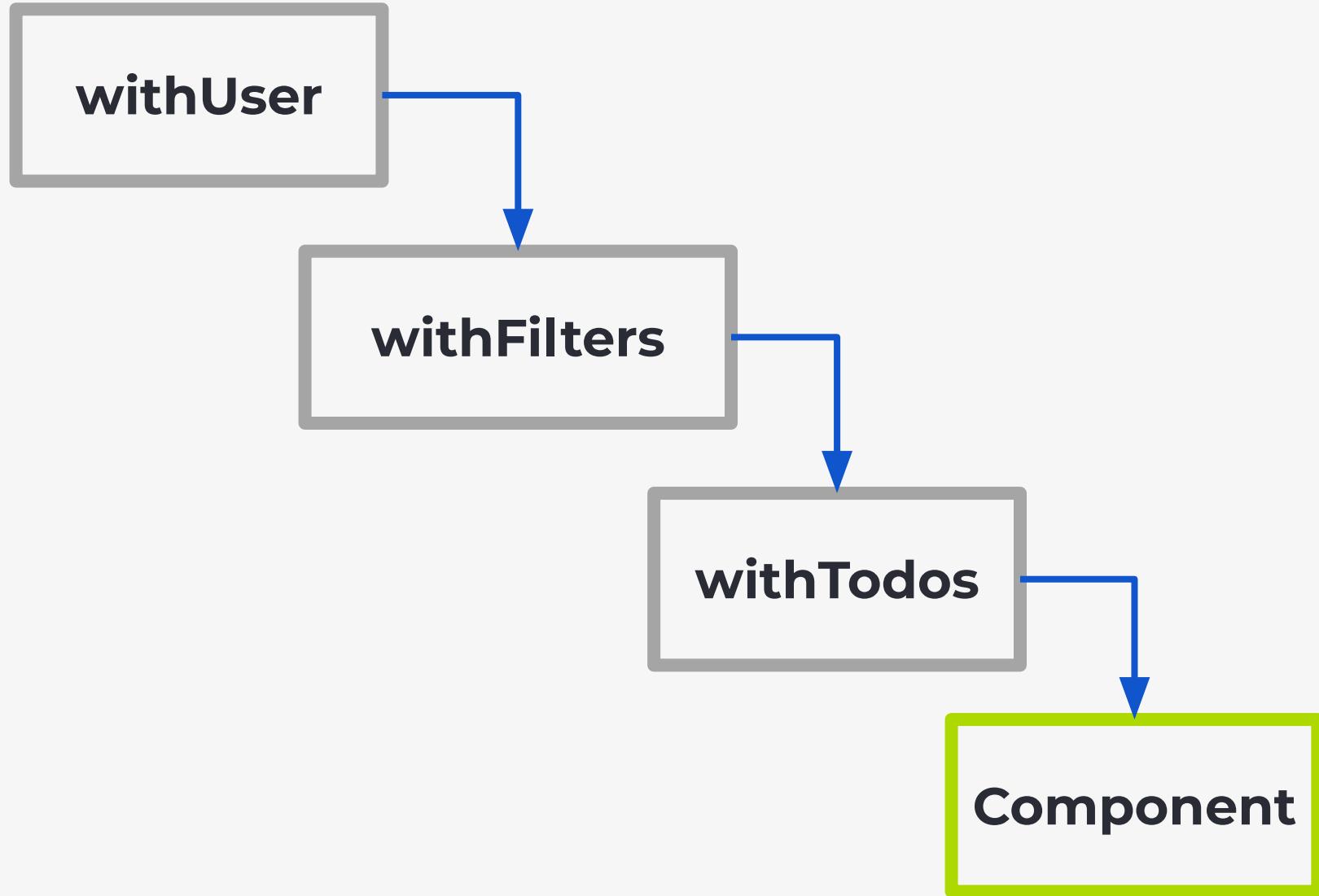
withUser



withTodos



Component



Ventajas y desventajas

- Render props 
- HOCs 
- Custom Hooks 

Maquetación



```
const { error, loading, todos } = customHook();

return (
  <TodoList>
    {error && <TodosError />}
    {loading && <TodosLoading />}
    {(!error && !loading && !todos.length)
      && <EmptyTodos />}
  )
  {todos.map(todo => <TodoItem {...todo} />)}
</TodoList>
);
```

```
<TodoList  
  error={error}  
  loading={loading}  
  todos={todos}  
  onError={() => <TodosError />}  
  onLoading={() => <TodosLoading />}  
  onEmptyTodos={() => <EmptyTodos />}  
  render={todo => <TodoItem {...todo} />}  
/>
```





Render props





Share data

```
<Consumer1>

{props1 => (
  <Componente {...props1} />
) }

</Consumer1>
```



```
function withApi(WrappedComponent) {  
  const apiData = fetchApi('https://api.com');  
  
  return function WrappedComponentWithApi(props) {  
    if (apiData.loading) return <p>Loading</p>;  
  
    return (  
      <WrappedComponent data={apiData.json} />  
    );  
  }  
}
```

```
function TodoBox(props) {  
  return (  
    <p>  
      Tu nombre es {props.data.name}  
    </p>;  
  );  
}  
  
const TodoBoxWithApi = withApi(TodoBox);
```

```
const TodoBoxWithEverything = withApi(  
  withSomething1(  
    withSomething2(  
      withSomething3(  
        withSomething4(  
          withSomething5(  
            TodoBox // iPOR FIN!  
          )  
        )  
      )  
    )  
  );
```

```
const [whatever1, setWhatever1] = React.useState( );  
  
return (  
  <Componente  
    whatever1={whatever1}  
    setWhatever1={setWhatever1}  
  />  
);
```

```
const [whatever1, setWhatever1] = React.useState();
const [whatever2, setWhatever2] = React.useState();
const [whatever3, setWhatever3] = React.useState();
const [whatever4, setWhatever4] = React.useState();
const [whatever5, setWhatever5] = React.useState();

return (
  <Componente {...todosLosWhatevers} />
);
```



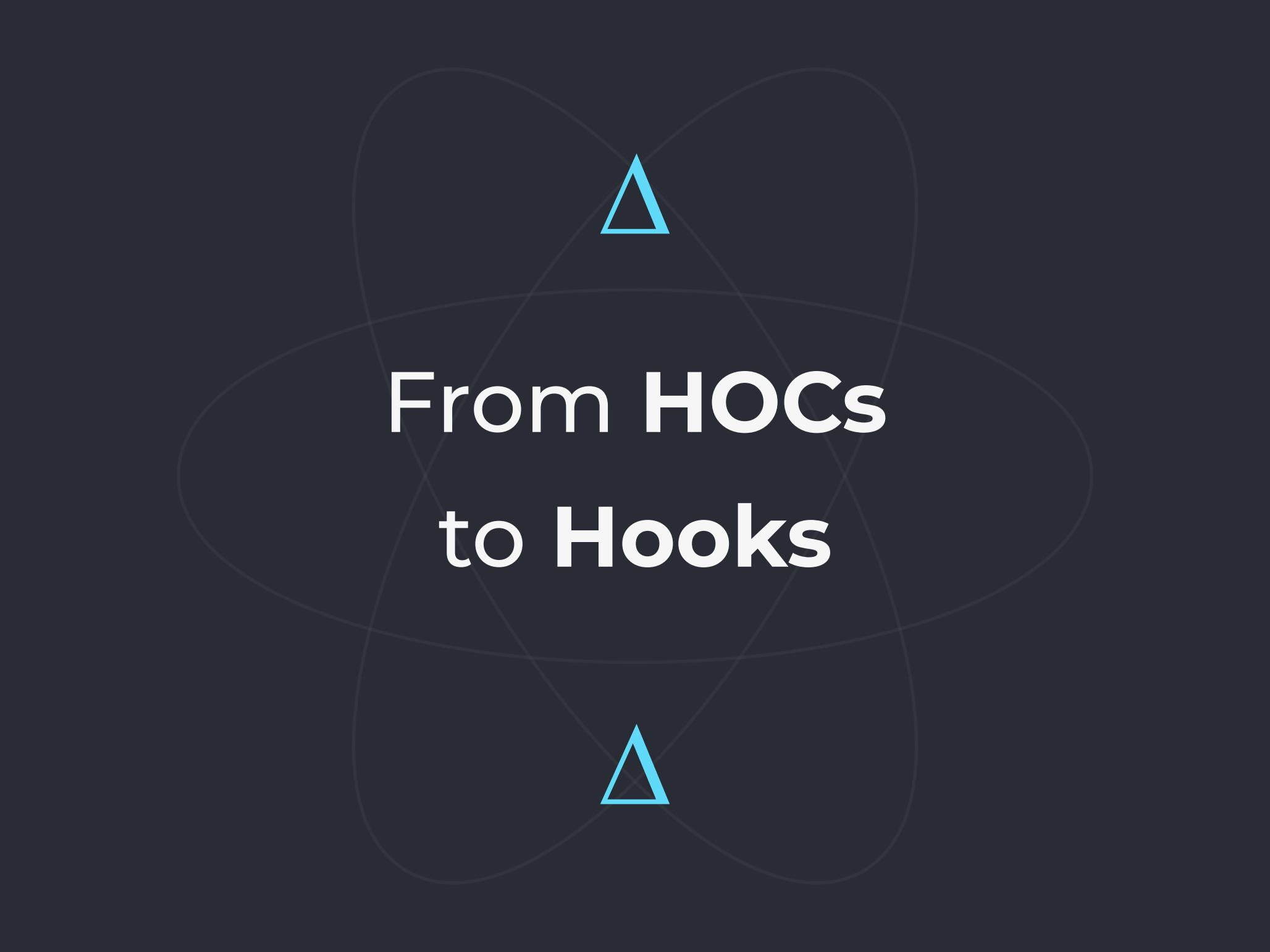


Custom Hooks 🔥



00 BONUS 00

```
const TodoBoxWithEverything = compose(  
  withApi,  
  withSomething1,  
  withSomething2,  
  withSomething3,  
  withSomething4,  
  withSomething5,  
  withSomething1,  
) (TodoBox);
```



From HOCs to Hooks

Próximos pasos