Fundamentos de JavaScript

Uriel – CTO de Código Facilito

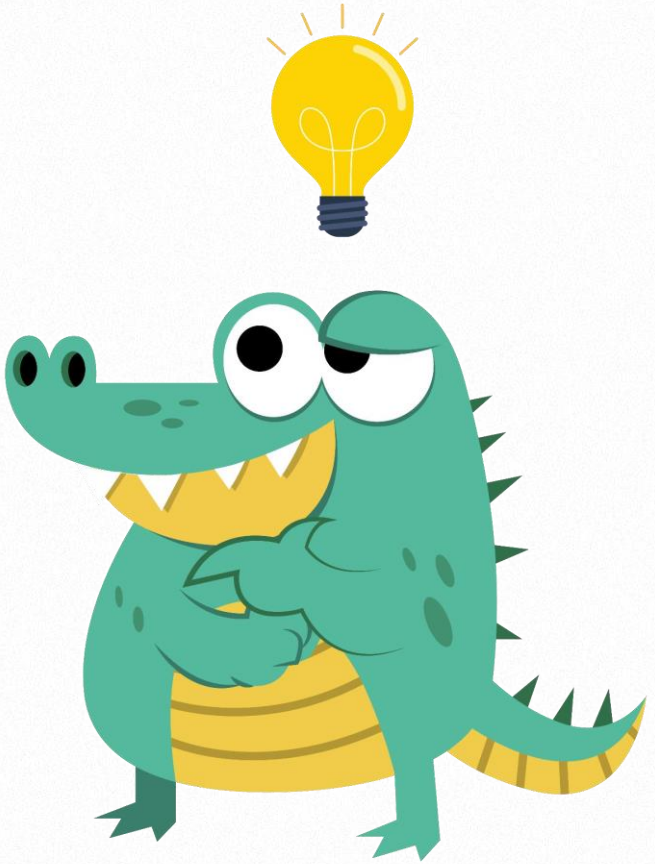
- Acerca de la clase
- React Context
- Context en práctica
- Taller práctico

Acerca de la clase

La clase de hoy sirve como un repaso de lo que hemos visto hasta ahora, principalmente:

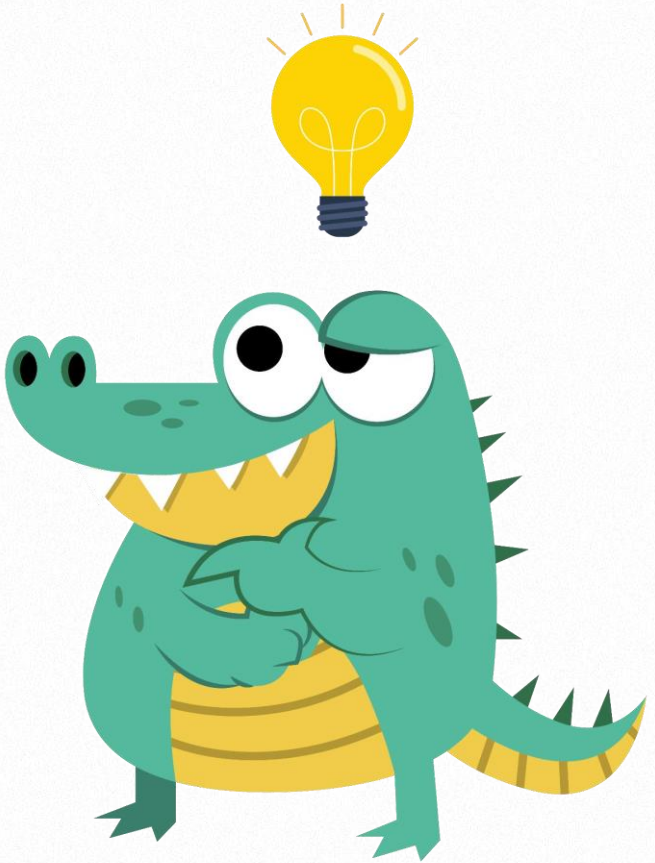
- **Uso de state y props**
- **styled components**
- **Despliegue de datos**
- **fetch**

También hablaremos de React Context: Definición, casos de uso, y ejemplo práctico.

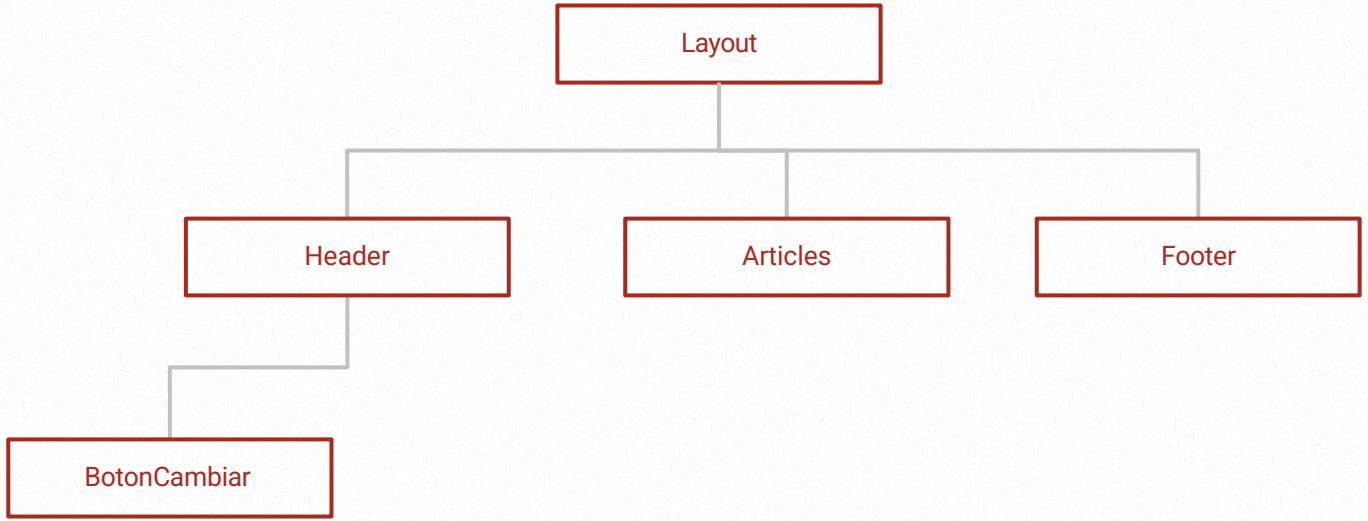


React Context

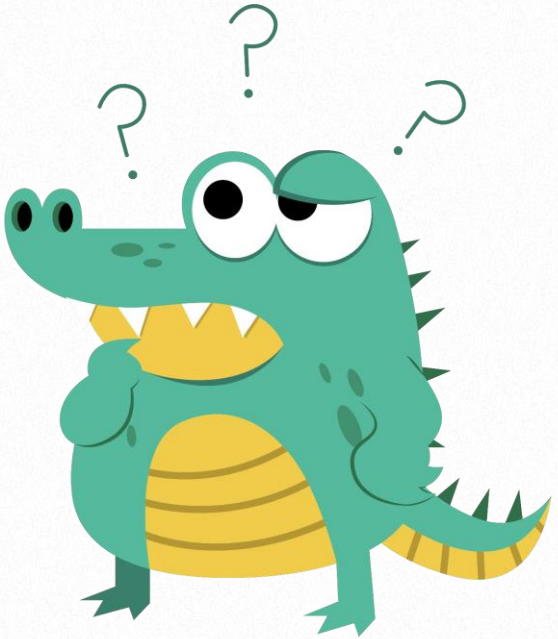
Context es una API de React que permite comunicar información entre componentes sin recorrer cada nivel del árbol de nuestros componentes.



```
<Layout>
  <Header>
    <BotonCambiar>Cambiar</BotonCambiar>
  </Header>
  <Articles></Articles>
  <Footer></Footer>
</Layout>
```

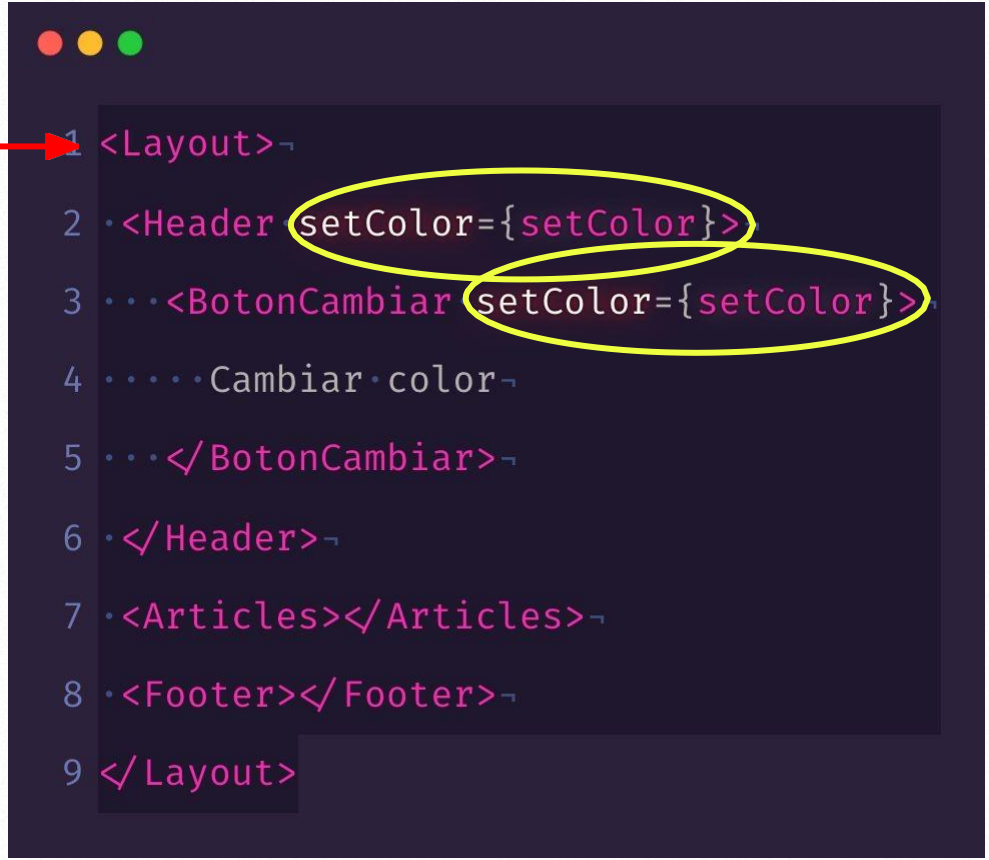

Layout
Header
Articles
Footer
BotonCambiar

Considera que para esta interfaz necesitas poder personalizar el color de fondo del Layout desde el BotonCambiar

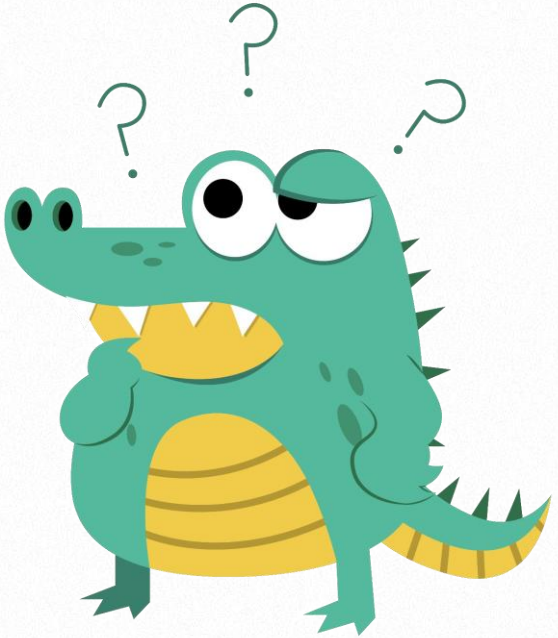


El método viene de Layout

Puedes solucionarlo compartiendo un método desde el padre:

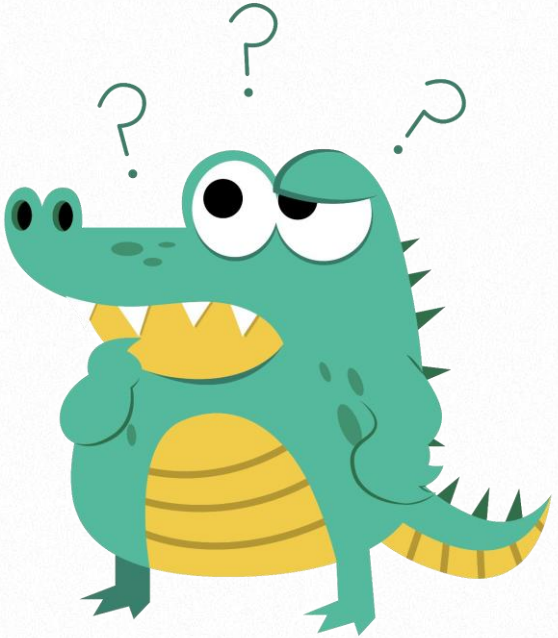


¿Qué pasa si también deseas que el color actual sea desplegado en el componente que cambia de color?



Puedes continuar enviando props desde un componente superior hasta uno inferior en el árbol:

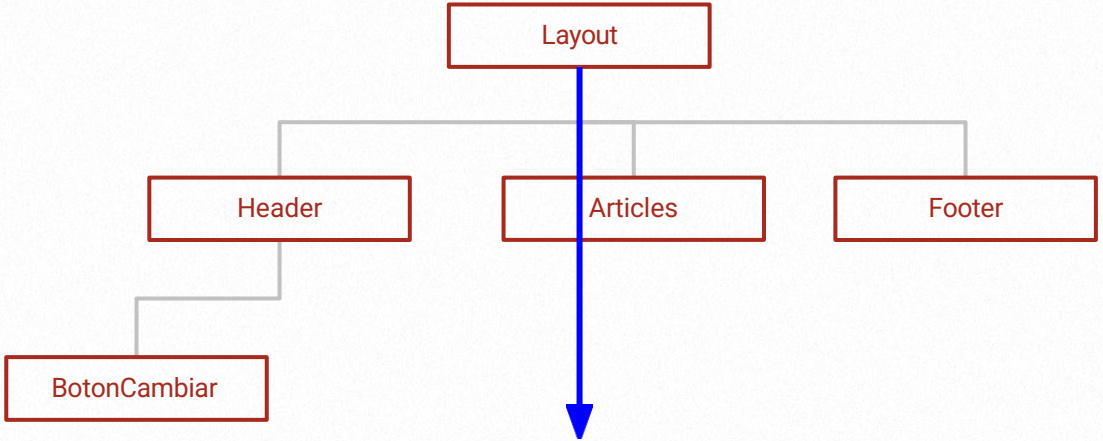
Esto puede traer más problemas en aplicaciones grandes, para entenderlo cubramos otros fundamentos.



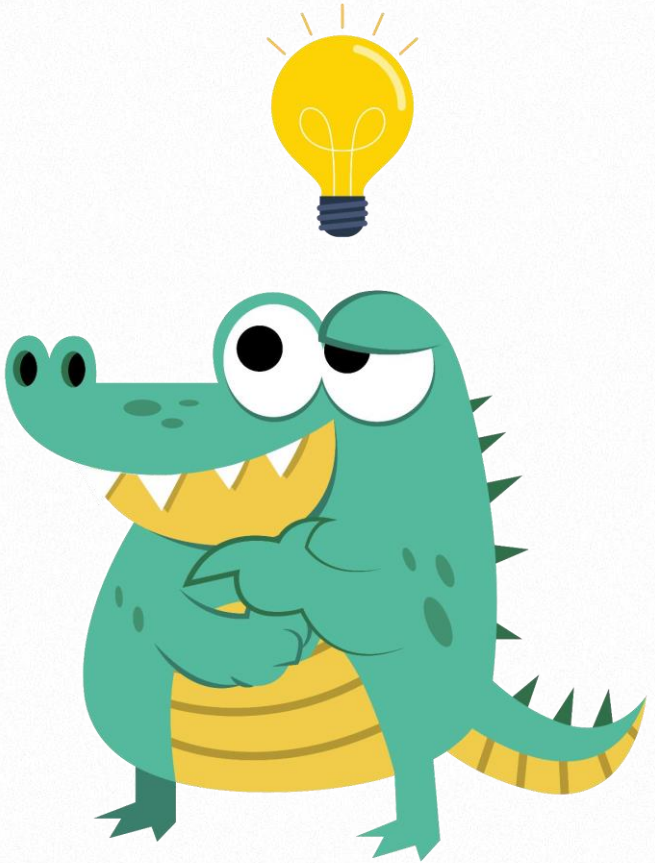
React Context

Data Flow

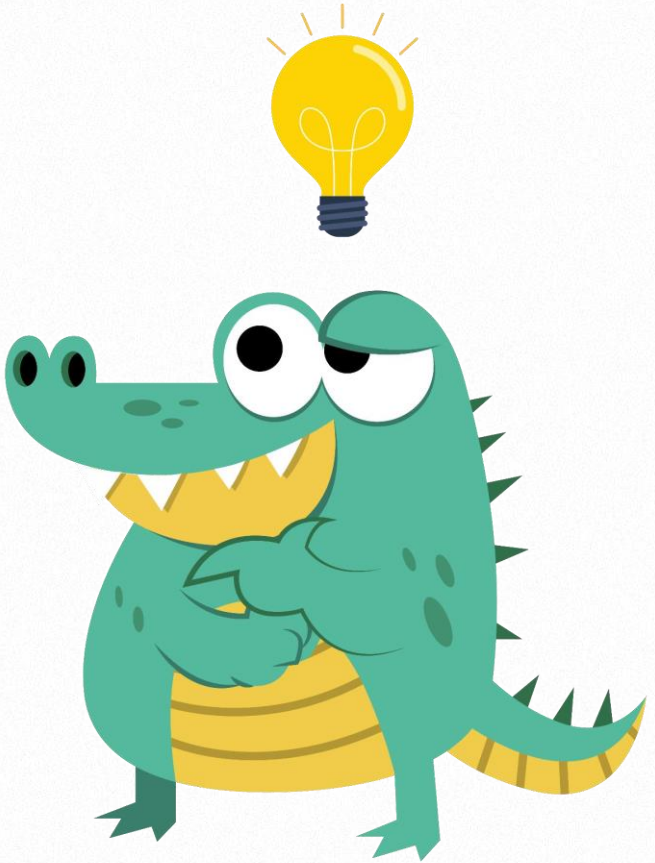
React está diseñado para que la información fluye en una sola dirección, desde los componentes superiores hacia abajo.



Esta arquitectura es una parte clave del diseño de la librería.



Al seguir esta regla, el flujo de información es explícito y más fácil de comprender.

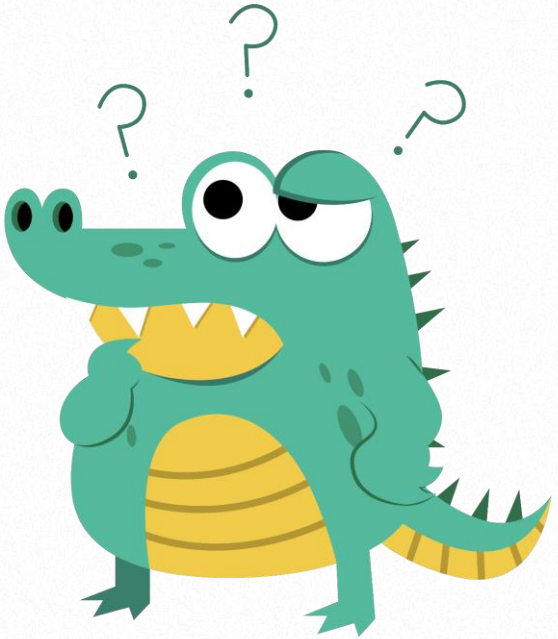


color y setColor van hacia abajo

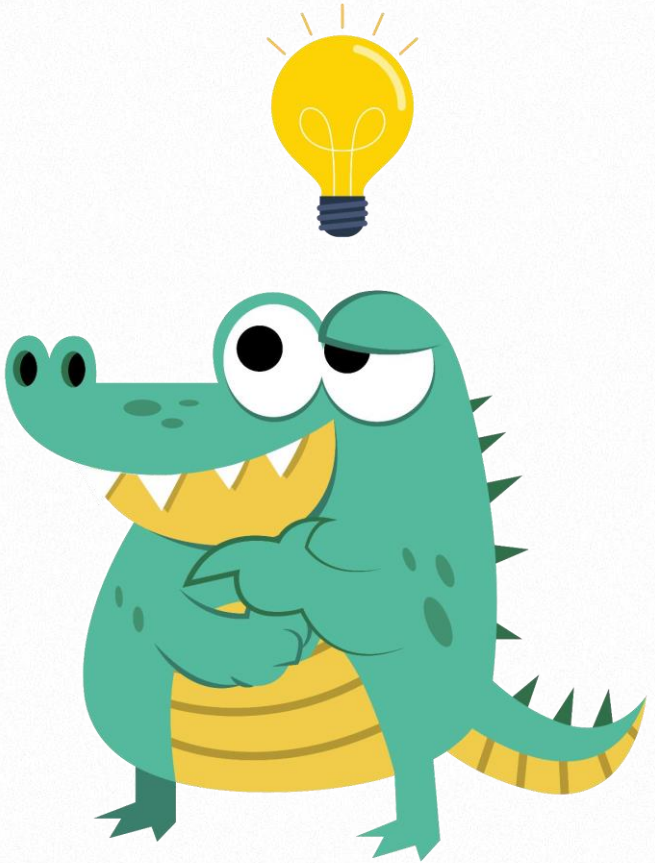
React Context

Prop drilling

Aunque React está diseñado para enviar información desde los componentes superiores hacia abajo, existen escenarios para los cuales esto puede ser problemático.

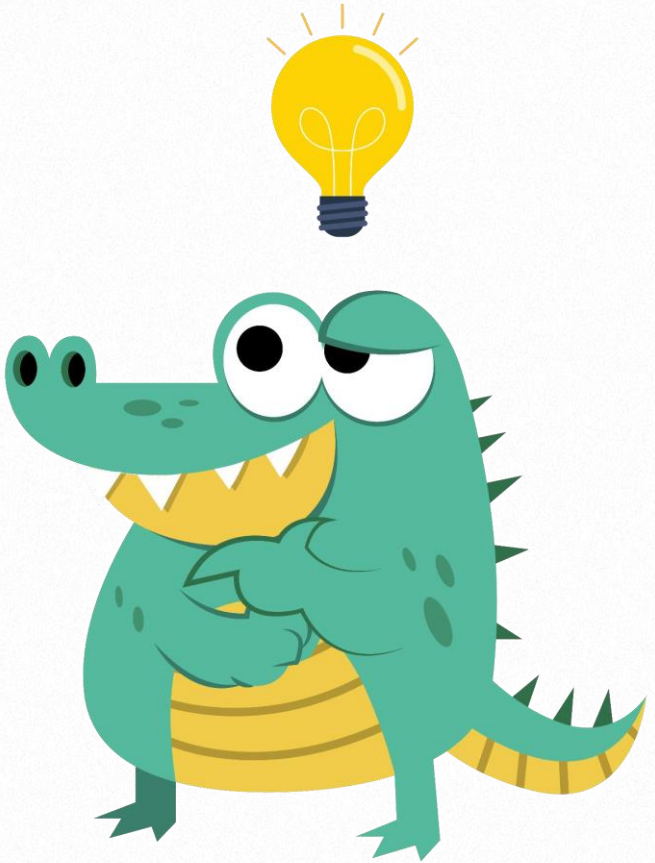


“Prop drilling” es el problema por el que los componentes solo para que esta información llegue a componentes hijos



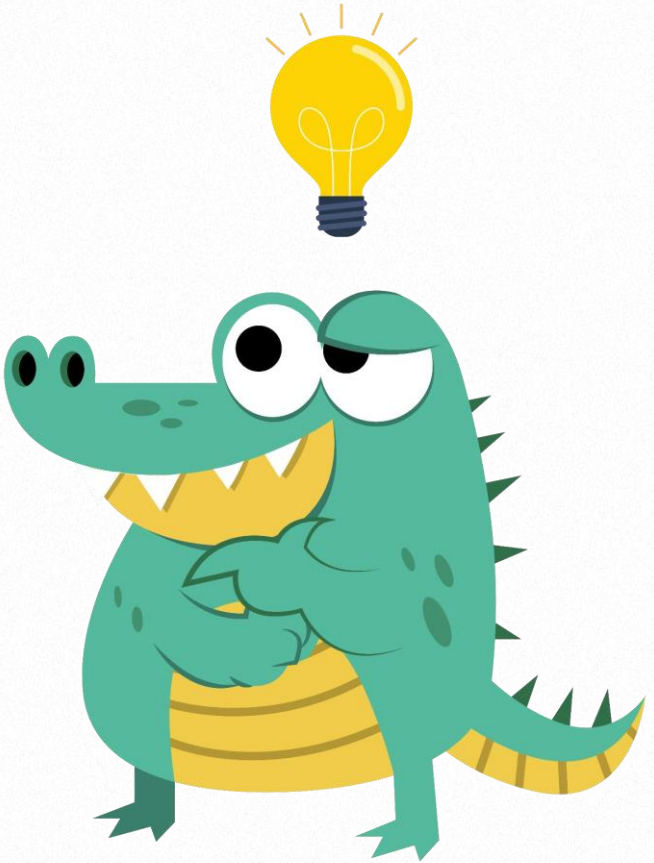
color y setColor no son necesarios para Header

El problema de prop drilling no está directamente relacionado con Context, ya que no todos los problemas de prop drilling deben ser resueltos con Context



React Context

Cuándo usar Context

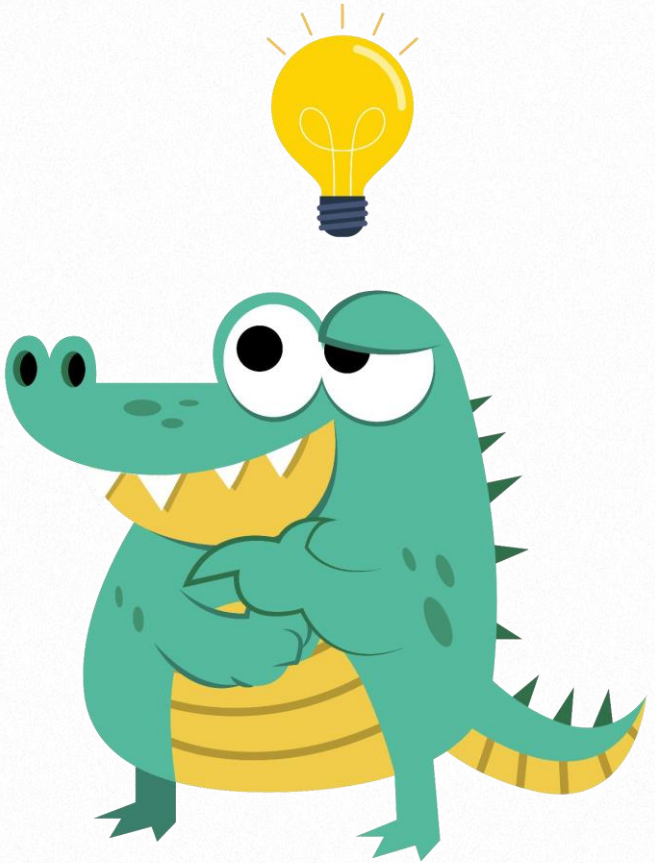


Context es un manejador de estado por lo que su uso debe asociarse con almacenar la información que debe ser compartida por muchos componentes.

Algunos ejemplos son:

- **Autenticación de usuario**
- **Theming**
- **Manejo de distintos idiomas**
- **Preferencias de moneda, zona horaria**
- **Entre otros**

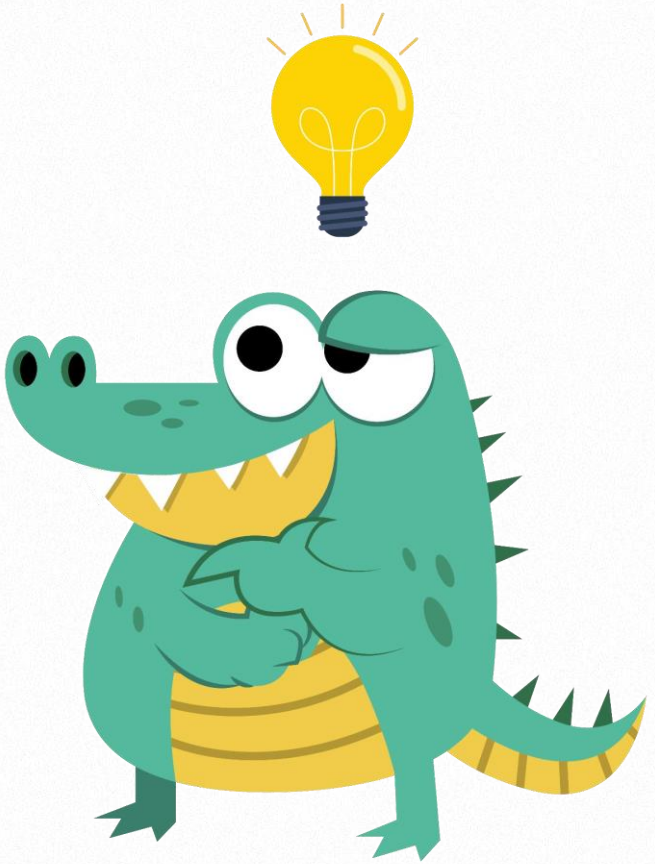
Si estás trabajando con información que afecta a distintos componentes, usar Context puede ayudarte a compartir la información entre todos los componentes.



También puedes usarlo en secciones del árbol

- Un formulario para el que los controles deben compartir Información
- Un componente separado en distintos pasos para mostrar

Una aplicación puede tener y generalmente requiere de más de un contexto.

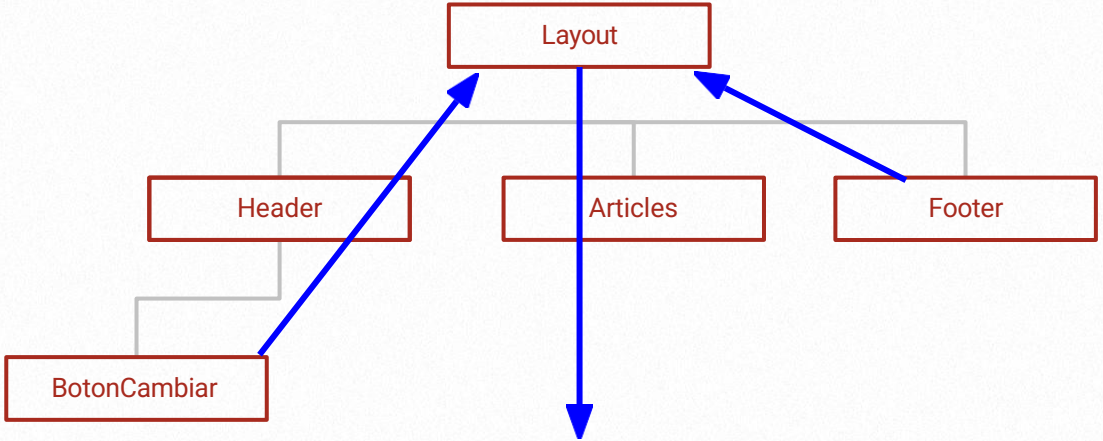


React Context

Cuándo NO usar Context

Las ventajas de Context viene con un precio a pagar, principalmente: Complejidad y Acoplamiento.

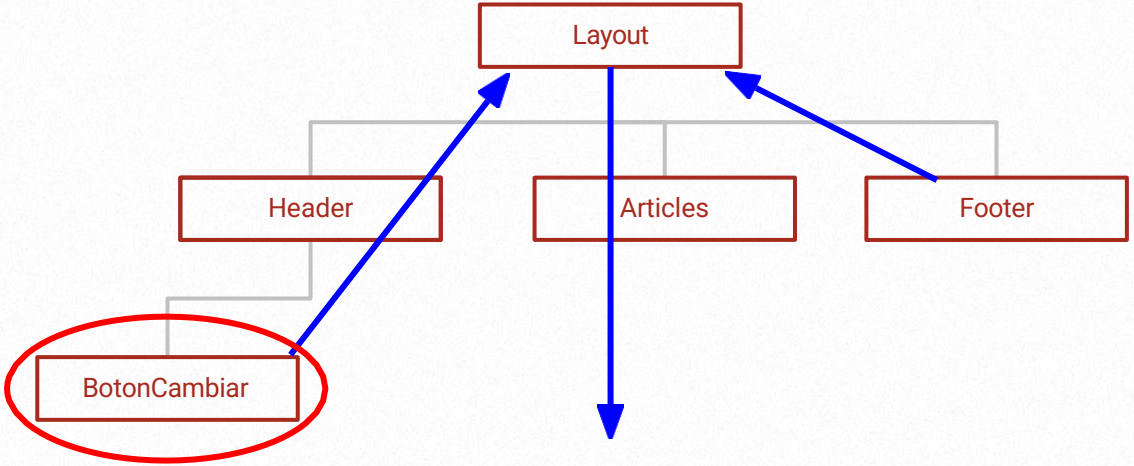
Esto significa que el código ahora es más complejo de entender.



Con Context la información ya no fluye en una sola vía, puede ir en ambos sentidos.

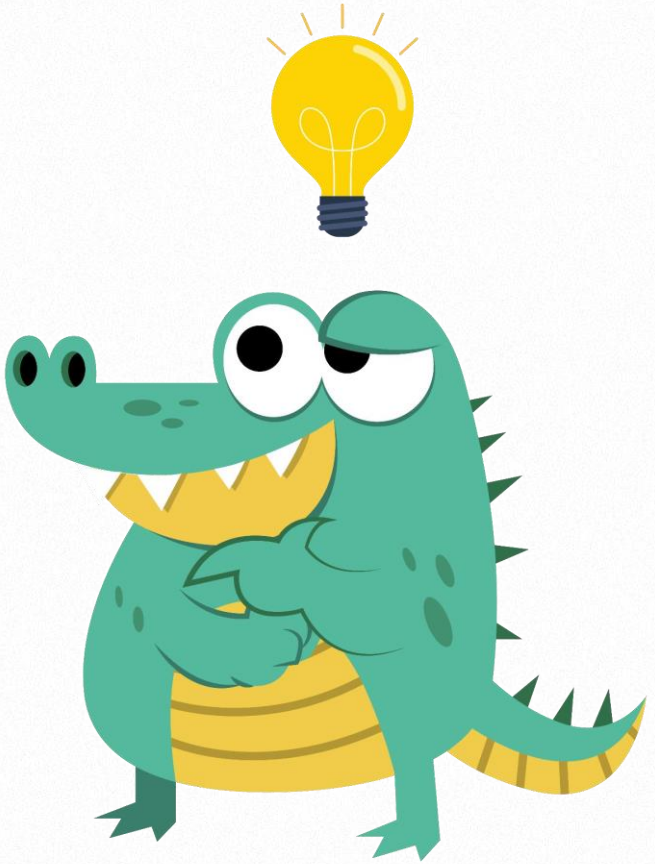
El uso de Context genera dependencia entre nuestros componentes

No puede ser reciclado en partes del árbol que no tengan el contexto



Si este componente usa Context.

En muchos casos prop drilling puede ser mejor que usar Context. En otros podemos usar composición.



Veámoslo en práctica:

- Cómo usar Context
- createContext
- Providers
- useContext
- useContext + useState

Ejercicio final

Bienvenido, soy Uriel Hernández desarrollador web

MIS PROYECTOS

DESCARGAR MI CV

O LEE MIS ARTÍCULOS

Veámoslo en práctica:

- Página personal +
- Dev.to posts

Tarea

- Escribe un post en
- Dev.to sobre alguno
- de estos temas: Qué
- es React, Qué es un
- componente en React,
- State vs Props

Tarea

- Qué son styled components
- Introducción a styled components
- Qué es un efecto en React

Tarea

- Qué es React Context

Tarea

- Finalmente este post debería aparecer en tu página del ejercicio de hoy. Envía screenshot al grupo.

Códigos utilizados en clase:

- Código de ejercicio #1: https://stackblitz.com/edit/react-hycjz8?file=src%2FApp.js
- Código del taller: https://github.com/codigofacilito/taller-practico-react

Fin.