

# COMPONENTES Y PROPS

Las propiedades de un elemento HTML están predefinidas y nos permiten modificar el aspecto y comportamiento de ese elemento. Cada etiqueta tiene las suyas, por ejemplo:

```

```

En React, en nuestros componentes .jsx podemos acceder a estas propiedades predefinidas de cada tag, pero además podemos pasar al propio componente como propiedad cualquier dato con este mismo objetivo de personalización del aspecto y el comportamiento. A estas propiedades las llamamos props.

```
export default function Profile() {  
  
  return (  
  
    <Avatar  
  
      person={{ name: 'Lin Lanying', imageId: '1bX5QH6' }}  
  
      size={100}  
  
    />  
  
  );  
  
}
```

Las props, son un concepto fundamental en el desarrollo de aplicaciones con React. Son utilizadas para pasar datos desde un componente padre a un componente hijo, permitiendo así la comunicación entre ellos. Las props pueden contener una variedad de datos, como strings, números, objetos, funciones y más, y se utilizan para personalizar y configurar el comportamiento de los componentes.

Existen varios tipos de props en React. En general, están las props estáticas, que contienen datos que no cambian durante la vida útil del componente. Estas props se utilizan para configurar aspectos específicos de un componente, como su estilo o contenido. Por otro lado, están las props dinámicas, cuyos valores pueden cambiar durante la ejecución de la aplicación. Estas props son útiles para representar datos que pueden ser actualizados o modificados, como el estado de un usuario o el contenido de una lista.

Especificando un poco más, las props pueden ser

- Datos estáticos: no cambian
- Props de configuración: para cambiar la apariencia, comportamiento, estado
- Funciones: comunicar funciones de un componente a otro
- Datos dinámicos: van a cambiar varias veces a lo largo del ciclo de vida del componente.

Las props pueden depender tanto de un estado interno como de un estado externo. Las props que dependen de un estado interno son aquellas cuyos valores son gestionados por el propio componente y pueden cambiar en respuesta a eventos o acciones del usuario. Por otro lado, las props que dependen de un estado externo son aquellas cuyos valores son proporcionados por un componente padre y no pueden ser modificados por el componente hijo.

Sobre props que dependen de un estado interno ya vimos varios ejemplos en [este](#) repositorio. Ahora veremos unos ejemplos de props que dependen de un estado externo, es decir, que realmente reciben información de otro componente.

Las props juegan un papel importante en el diseño atómico de aplicaciones. El diseño atómico es un enfoque de diseño que descompone una interfaz de usuario en componentes más pequeños y reutilizables, que se combinan para crear interfaces más complejas. Las props permiten pasar datos entre estos componentes atómicos, lo que facilita la creación de interfaces cohesivas y flexibles.

En resumen, las props son un mecanismo fundamental en React que permite la comunicación entre componentes, la personalización de su comportamiento y la creación de interfaces de usuario modulares y reutilizables. Su versatilidad y flexibilidad las convierten en una herramienta esencial para el desarrollo de aplicaciones web modernas.

Visita la [documentación oficial](#) para ver otros ejemplos y practicar algunos retos