

Uno de los problemas más comunes a la hora de trabajar con React es que los componentes crecen sin control. Partimos de un componente pequeño, muy enfocado a una funcionalidad. El proyecto comienza a crecer y nuestro componente debe de recibir más props para adaptarse a los nuevos cambios. El método `render` pasa de unas 10 líneas de código (LOC) a tener 100. **Esta situación hace que los componentes sean difíciles de manejar, toleren peor los errores y cueste depurar sus problemas.**

En la mayoría de estos casos, **la mejor solución es crear otro componente nuevo.** Así crearemos dos componentes más específicos en lugar de uno que se utilice en distintas situaciones. Esto no quiere decir que no tengamos componentes más complejos en nuestra aplicación.

Una manera de detectar problemas de este tipo es cuando un el método `render` supera las 20 líneas de código. Como he comentado, algunos componentes serán más complejos por lo que esta situación estará justificada.

Otra manera de mejorar nuestro método `render` es no incluir excesivo código JavaScript en JSX. El objetivo de este método es representar información, por lo que el procesamiento de los datos debe de ser previo. Para ello podemos utilizar llamadas a métodos o el propio estado del componente para procesar y almacenar datos antes de representarlos.

```
class Article extends React.Component {
  static propTypes = {
    title: React.PropTypes.string.isRequired,
    text: React.PropTypes.string.isRequired,
    date: React.PropTypes.number.isRequired,
    tags: React.PropTypes.arrayOf(React.PropTypes.string).isRequired,
    author: React.PropTypes.object.isRequired
  }

  render() {
    return <article className="Article">
      <header className="Article__Header">
        <h1>{ this.props.title }</h1>
        <div className="Article__Author">
          <p>{ this.props.author.name }</p>
          <p>{ this.props.author.bio }</p>
          { /* ... */ }
        </div>
      </header>
      <div>
        { /* ... */ }
      </div>
    </article>
  }
}
```