



## Front End III

# ¿Por qué Hooks? Si con las clases estábamos bien

## Un breve contexto

Como sabemos, las tecnologías de Internet se han desarrollado exponencialmente en los últimos años a través del éxito irrefrenable de las FAANG<sup>1</sup>. Por ejemplo, el caso de Facebook que pasó de ser —en su etapa germinal (2003)— un sitio llamado FaceMatch (creado por Mark Zuckerberg para el ámbito de la Universidad de Harvard), en el que se calificaba rostros universitarios; hasta llegar a hoy —tras casi 20 años— a ser una de las más grandes corporaciones de alcance cuasi global: propietaria de Instagram, WhatsApp, Oculus, creadora de Internet.org, entre otras miles de iniciativas. El avance continuo de estas corporaciones se da en el marco de una **competencia permanente por crear una Internet a espejo de sus necesidades** y definir las formas de comunicación más usadas en occidente.

Este es el contexto en el que surge React, como así también Angular (Google). El de la búsqueda de controlar el rumbo de Internet y, con ello, debido a la necesidad implícita de **sofisticación y optimización permanente de las tecnologías de Internet**, la necesidad de mejora continua en términos de estos frameworks/bibliotecas.

---

<sup>1</sup> FAANG se refiere a las cinco empresas de tecnología estadounidenses prominentes: Facebook, Amazon, Apple, Netflix y Alphabet (GOOG). En: [https://es.wikipedia.org/wiki/Gigantes\\_tecnol%C3%B3gicos#FAANG](https://es.wikipedia.org/wiki/Gigantes_tecnol%C3%B3gicos#FAANG).

## ¿Pero cómo se llegó a Hooks?

La liberación de React de la mano de Facebook, allá por el año 2013, fue muy exitosa. Su enfoque basado en una arquitectura de datos fluyendo hacia abajo, composición en componentes y una superrápida actualización del DOM de forma declarativa –gracias al uso del Virtual Dom– se viralizó en el mundo front end. Fue una estrategia exitosa. Lo sigue siendo hoy.

## Problemas

A pesar de esto, la historia de React no estuvo exenta de problemas y adecuaciones. Principalmente respecto a la implementación de sus componentes, su pieza maestra. Cuando React fue lanzado a la comunidad, JavaScript no contaba aún con un sistema de clases, así que React optó por crear la API **React.createClassAPI** para la creación de componentes. Fue un camino muy eficiente.

No obstante, la felicidad duró poco. Con el lanzamiento de la versión 6 de ECMAScript, en el año 2015, JavaScript introdujo oficialmente la palabra reservada Class y, ante la disyuntiva de seguir su propio camino o adecuarse a ECMAScript para no perder seguidores, **React optó por adoptar las clases de JavaScript.**

Sin embargo, esto introdujo nuevos problemas generados por las particularidades del funcionamiento de las clases en JavaScript. Este nuevo camino exigió la necesidad de crear subclases extendidas de React.Component, declarar el estado dentro de un constructor, usar this, bind y super(props). Podríamos igualmente convenir que, más allá de ser un poco molesto, usar bind y super(props) no eran en sí problemas. Igualmente, a nadie le gusta programar años de su vida con cierta molestia.

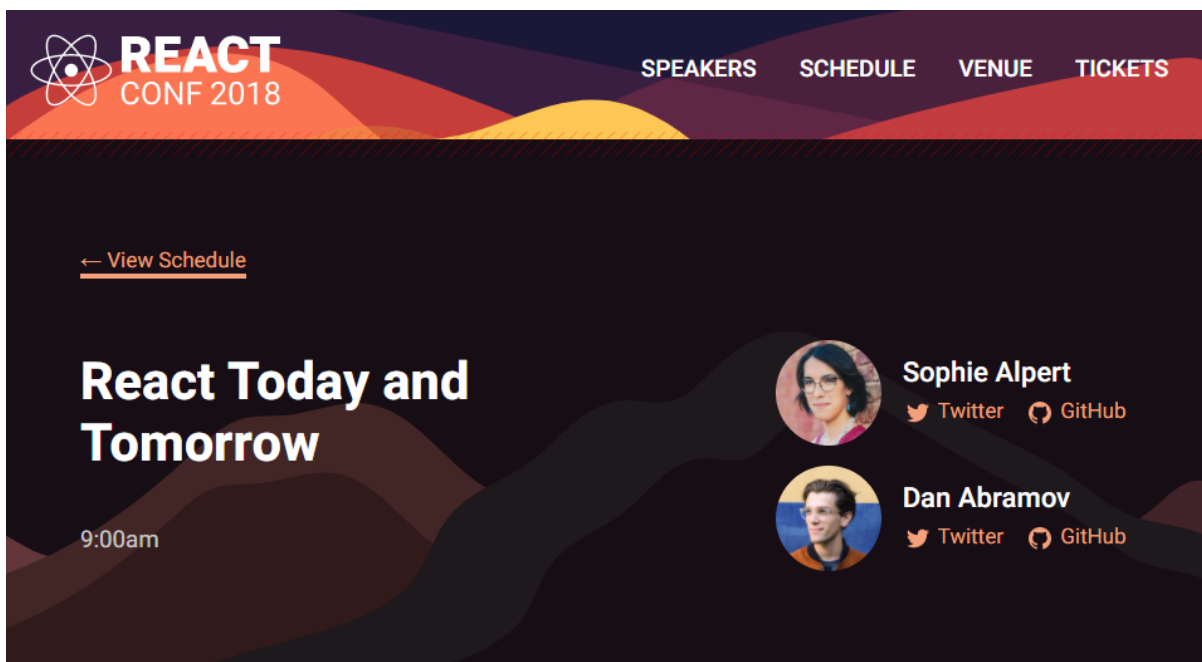
Al detectar el fastidio en la comunidad front end, React trabajó en encontrar soluciones y presentó la sintaxis Campos de Clase. Con esta nueva característica, accedimos a la posibilidad de declarar el estado inicial de un componente sin declarar el constructor y, además, evitamos bind mediante la implementación de funciones flecha en los handlers.

```
class Mejorando extends React.Component {  
  state = { goal: '' };  
  
  handleGoalChange = (goal) => this.setState({ goal });  
}
```

```
render() {  
  const { goal } = this.state;  
  return <Goal onGoalChange={this.handleGoalChange} goal={goal} />  
}  
  
export default Mejorando;
```

## Nacen los Hooks

En el mes de octubre del año 2018 –en la React Conf, celebrada en Henderson, Nevada, EEUU–, Sophie Alpert y Dan Abramov, miembros del equipo de Facebook encargado de desarrollar React (por ese entonces), presentaron una nueva característica revolucionaria para el mundo front end: React Hooks. Más tarde, en febrero de 2019, esta propuesta se hizo realidad y React liberó su versión 16.8 incluyendo la API de Hooks.



**¡Hasta la próxima!**