

## **Incorporando una interfaz de usuario a la aplicación de grafos**

Una de las limitantes que presentan los programas creados por línea de comandos, es su limitada experiencia de usuario y su accesibilidad. Por medio de interfaces gráficas web, se pueden crear programas más fáciles de utilizar y entender; además accesibles a todo el mundo con acceso a internet. React, creada por Facebook, se utiliza para el desarrollo de “single-page applications” y busca proveer simplicidad, velocidad y estabilidad. Al ser una librería y no un framework, React es utilizado conjuntamente con otras librerías fácilmente. En la presente investigación se exploran las ventajas que ofrece dicha librería a la hora de presentar una interfaz gráfica cómoda y accesible por medio web.

### **Componentes I**

Todo aquel que haya experimentado con JQuery o Javascript nativo sabrá que construir interfaces de usuario dinámicas no es tarea fácil. Para cambiar el estado de los elementos es importante saber el estado previo al que estaban. Además, conseguir reusabilidad es complicado, y termina siendo una tarea de copiar y pegar código. React ofrece un modelo orientado a componentes donde se encapsula el comportamiento y estilo en componentes específicos. Estos componentes a su vez pueden estar compuestos de otros componentes. Esto permite que el software sea mu-

cho mas fácil de razonar, modificar y reutilizar.

Para construir un grafo por ejemplo se tendría un componente Grafo compuesto a su vez por ejes y nodos. Cada uno con características visuales: ya sea color, tamaño etc. Y características de comportamiento: como mostrar la información cuando se le hace click etc.

Una gran ventaja de dichos componentes es que son funcionales, es decir, reciben una serie de atributos y producen un resultado definido en función de dichos componentes. Cambiar el componente de un estado a otro es tan fácil como cambiar sus atributos. No es necesario conocer el estado previo en el que se encontraba. Esto permite una programación mas declarativa y fácil de entender y darle mantenimiento.

### **Eficiencia II**

React se auto proclama como una librería muy eficiente. Esto es debido a que ha sido construido a partir de ciertas reglas de juego que permiten un buen rendimiento a la hora de renderizar elementos en el navegador. Utilizando javascript nativo, cada vez que se requiere hacer un cambio en el DOM, se hace directamente sobre él. Esto supone un desperdicio de eficiencia ya que cuando se requieren hacer multiples cambios a la vez, el navegador realiza todo el trabajo de hacer los

cálculos para dibujar en pantalla por cada cambio. React en cambio utiliza una estructura de datos llamada el Virtual DOM la cual es una representación del DOM en javascript. Esta representación esta en memoria y hacerle cambios es muy eficiente. Cuando muchos cambios en la interfaz gráfica ocurren seguidos, React modifica el shadow dom con todos los cambios y después utiliza la representación resultante para que el navegador dibuje en pantalla los cambios. De esta manera el navegador hace el proceso de cálculos y dibujo una sola vez por todos los cambios.

### **JSX III**

React utiliza JSX para representar los componentes.

JSX es un lenguaje de programación orientado a objetos y de tipo estático, diseñado para ejecutarse en navegadores web modernos.

Siendo desarrollado en DeNA como un proyecto de investigación, el lenguaje tiene las siguientes características. JSX realiza la optimización al compilar el código fuente en JavaScript. El código generado se ejecuta más rápido que un código equivalente escrito directamente en JavaScript. A diferencia de JavaScript, JSX tiene un tipo de escritura estática y es más seguro para los tipos. La calidad de las aplicaciones aumenta cuando se desarrolla con JSX, ya que se detectan muchos errores durante el proceso de compilación. También ofrece funciones de depuración a nivel del compilador. JSX ofrece un sistema de clases sólido como Java, que libera a los desarrolladores de trabajar con el sistema de herencia basado en prototipos demasiado primitivo proporcionado por Java-

Script. Las expresiones y declaraciones, sin embargo, son en su mayoría iguales a JavaScript, por lo que es fácil para los programadores de JavaScript comiencen a usar JSX.

### **Conclusión IV**

React es una opción viable para la construcción de una interfaz de usuario para la aplicación realizada. Provee eficiencia y facilidad de uso. Además es fácil de aprender y portable ya que funciona por componentes que pueden ser intercambiables. Al ser web permite que sea accesible cómodamente por medio de internet.

### **Bibliografía:**

1. 4 Razones del crecimiento de React. (2017). KeepCoding | Cursos y Bootcamp: programación Web, Mobile y Big Data.. Retrieved 23 April 2018, from <https://keepcoding.io/es/blog/4-razones-del-crecimiento-de-react/>
2. "React - A JavaScript library for building user interfaces". React. Retrieved 23 April 2018.
3. "Refs and the DOM". React Blog.
4. "Components and Props". React. Facebook. Retrieved 23 April 2018
5. "Draft: JSX Specification". JSX. Facebook. Retrieved 23 April 2018.