

Ejercicio sobre reactividad y rendering

Serhii Vidernikov Y4459773H

Nikita Polyanskiy Y4441167L

En 1969 se crearon las primeras hojas de cálculo (LAMPAR) que reaccionan dinámicamente a los cambios lo cual es la idea principal de la programación reactiva. Es decir, mapear, controlar y reaccionar a los cambios de las variables en cuanto se ha producido un cambio.

REACT fue creado en 2013, fue la primer librería que utilizó un DOM virtual, es una estrategia para optimizar el rendimiento de una aplicación, proporciona un mecanismo que compara 2 árboles de renderizado para ver qué ha cambiado exactamente, y solo actualiza los datos que hayan cambiado, pero no es lo suficientemente rápido, ya que tiene que leer de los 2 árboles constantemente para ver si hay algún cambio.

Svelte fue creado en 2016, es parecido a REACT, pero ofrece algunas funcionalidades nuevas con el objetivo de hacer el código más fácil de leer y escribir, además de optimizar el rendimiento de la aplicación. Svelte coge los componentes declarados y los convierte en código eficiente que manipula el DOM directamente. Antiguamente en Svelte, al igual que React, las variables tenían un estado, y al cambiarla obligaba a volver a renderizar la página. Esto suponía mucho trabajo y unas restricciones difíciles de anticipar. Svelte ha conseguido llevar esta tarea a nivel de lenguaje y compilador, es decir, el compilador automáticamente ha indicado que algunas variables han cambiado y hay que tenerlo en cuenta a la hora de actualizar la página. Aparte, también permite utilizar renderizado SSR (server side rendering), devolviendo una cadena con el contenido de la página, lo cual es más eficiente que utilizar DOM.

Además Svelte permite escribir funciones inline dentro del código html, haciendo el código más fácil de escribir y leer. Por otro lado, es posible utilizar declaraciones reactivas como por ejemplo etiquetar funciones o variables añadiendo "\$:" antes de dicha función o variable.

La forma en la que Svelte acelera el código es mirándolo de otra forma. Es decir, no intenta acelerar el framework que ya teníamos, consiguiendo muy pocos resultados, y trabajando mucho, sino que intentan hacer nuevos cambios radicales.

La mejor forma de acelerar código es tener menos código, de forma que Svelte gana un +-40% de aceleración con respecto a REACT, el cual requiere un +-40% más de líneas de código para hacer la misma función.

Por último se muestra como Svelte tiene mejor rendimiento que otros frameworks en dispositivos de bajo rendimiento. Como las televisiones o las terminales, consiguiendo funcionamiento muchísimo mejor, comparado con otros frameworks

Es interesante que Svelte logre cumplir las mismas funcionalidades que REACT, de hecho tiene más y encima es más eficiente a la hora de rendimiento. En la práctica anterior teníamos más interés en aprender React al tener mayor antigüedad y buenas opiniones parecía más fiable, pero ahora nos interesa más aprender Svelte. Respecto a la presentación ha sido entretenida, interactiva y sabían cómo coger la atención de las personas que estaban presentes, es decir, había muchas imágenes, código representativo, gráficas e interacción con la audiencia. Por último, la presentación y los ejemplos que se ponían eran bastante simples por lo que no produjo confusión y ha sido interesante.