DIGITALHOUSE

Enrutamiento: Web tradicional vs. React

Certified Tech Developer
The Ultimate Degree

Índice

01 Web tradicional

Enrutamiento tradicional

El enrutamiento es un mecanismo que abarca varias funciones, pero lo más importante que se logra con él es **leer las URL, interpretarlas y pasar esta información a la parte de la aplicación que se encarga de renderizar la vista correspondiente**.

Es muy importante entender que una URL no es en realidad una ruta, es más bien una indicación de cómo se deben armar y conseguir los recursos, y esta indicación es interpretada por el enrutador de la aplicación.

Desventaja

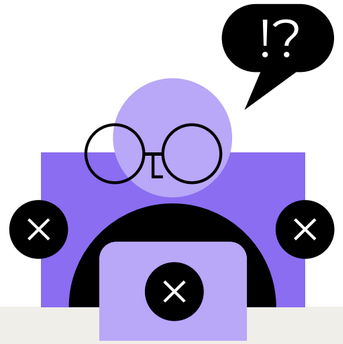
Este tipo de páginas con enrutamiento tradicional tiene una principal **desventaja**. Cada vez que “cambiamos” de ruta se realiza una llamada completamente nueva al back end y se recarga el contenido de toda la página.

Imaginemos que estamos navegando en **Facebook** y cada vez que cambiamos de página tenemos que esperar que se efectúe la llamada, se procese la respuesta y se cargue el contenido en pantalla. ¡Dejaríamos de usar Facebook definitivamente!

Por suerte, para nosotros usuarios (y desarrolladores) en la actualidad existe un nuevo enfoque para manejar la navegación en una aplicación web: las SPA.

02 React

Single Page Applications (SPA)



Hoy en día, gracias a la potencia de los navegadores, bibliotecas, frameworks de front end (como React, Angular o Vue) y a tecnologías propias de JavaScript, podemos conectar el navegador con el back end sin necesidad de recargar la página. **Esto dio paso a las Single Page Applications (SPA)**.

En una SPA no existe tal cosa como “varias rutas” dentro de una página, sino que **es una sola página con diferentes vistas**. Llevando este nuevo enfoque a Facebook, ahora, cada vez que “cambiemos” de ruta, lo que estamos haciendo es decirle al navegador la vista que debe mostrar (pero la URL es una sola). Solo cambiamos la porción en pantalla que queremos mostrar.

Navegación

Esta nueva forma de manejar dinámicamente el contenido de la web sin cambiar su URL trajo como consecuencia la pérdida de ciertos aspectos del comportamiento natural del navegador.

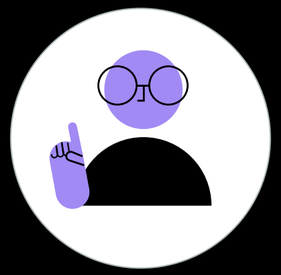
Recordemos algunas de las cosas que damos por sentado cuando navegamos en la web:

- Esperamos que dentro de un sitio web cada página tenga su propia URL, así podemos guardar bookmarks y visitar esa página en el futuro.
- En documentos muy largos, esperamos que un clic nos lleve exactamente al contenido dentro de la página. Esto lo hacen los navegadores con los identificadores de fragmento (hash). El identificador de fragmentos de la URL es la porción de caracteres de la URL desde el numeral en adelante.
- Queremos poder retroceder o adelantar usando las flechas de navegación del historial.
- Si algún link nos lleva fuera del website o a otra página del mismo website, esperamos ver una indicación de que la página está cargando los contenidos.

Problemas

Pero lo que sucede en una SPA es que todo el contenido se muestra en una sola página, por lo que veremos una única URL para todas las páginas. Con esto perdemos la capacidad de marcar nuestras páginas favoritas, y de que las flechas del historial nos faciliten la navegación.

Además, si la SPA simula la navegación entre páginas manipulando el identificador de fragmentos de la URL, esto puede entrar en conflicto con la función natural del navegador de poder usar el identificador de fragmentos para llevarnos a partes de la página identificadas con hashes.



Necesitamos **mantener la concordancia** entre una SPA y las funciones tradicionales del navegador.

01 El usuario aún debe poder navegar con URLs usando la barra de direcciones.

02 Hay que mantener un historial de cambios de las URL.

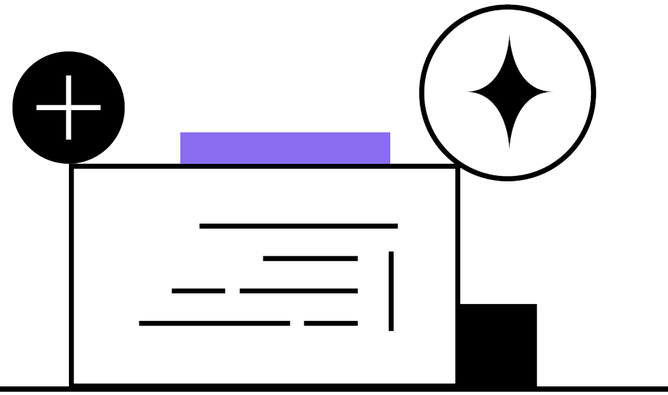
03 Hay que manejar los hashes (#), que llevan a partes específicas dentro de una misma página.

04 Hay que manejar las transiciones entre estados. Lo que antes era la recarga de la página.

05 Hay que entretener e informar al usuario mientras se actualiza el estado.

Algo más...

Además de estos requisitos, hay dos aspectos más de mucha importancia desde el punto de vista del negocio: es importante que las URL sean **SEO-friendly**, y que sean **semánticas**. Lo que queremos es que las URL sean fáciles de leer para el usuario y fáciles de ser indexadas de manera correcta por los buscadores web.



React Router

Para solventar estas necesidades, los frameworks modernos reintrodujeron el concepto de enrutador solo que ahora en el lado del front end, surgiendo así librerías como **React Router**.

DIGITALHOUSE

Gracias