 Server-Side Rendering

El Server-side rendering (o renderizado en la parte servidor) se basa en la posibilidad de poder renderizar el HTML de nuestros componentes en cadenas de texto en la parte servidor, vez de la parte cliente. Estas cadenas serán las respuestas que nuestros servidores de NodeJS devolverán a las peticiones principales de nuestra Web. En vez de funciones que manipulan DOM en el navegador, delegamos este renderizado a una fase anterior en el servidor.

### **¿Que es Server Side Rendering?**

Es una técnica utilizada para representar una aplicación de página única (SPA) en el servidor. Si, el servidor envía una página totalmente representada al cliente y posteriormente la SPA tomará el control de la interacción y renderizado al terminar de descargar e interpretar su código JavaScript.

**¿Cómo es esto?**

El cliente realiza una petición al servidor y éste, consulta los datos necesarios y genera una página HTML enriquecida con los datos como respuesta, pero, cada vez que se navega o se cambia de ruta, es necesario realizar todo este proceso nuevamente.

Con la aparición de algunos frameworks y bibliotecas JavaScript modernas, conocimos las aplicaciones totalmente generadas por JavaScript y el concepto de SPA (Single Page Aplication = aplicación de página única), las cuales se renderizan del lado del cliente, son dinámicas y muy rápidas.

De este modo, el cliente realiza un petición al servidor y este devuelve un solo archivo HTML sin ningún contenido hasta que el navegador obtenga e interprete todo el JavaScript.

## **¿Qué impacto en desarrollo me puede suponer usar SSR?**

Tenemos que tener claro que hacer uso de SSR no va a ser útil para todos y que su uso tiene que estar muy justificado en nuestra aplicación, pues podemos tener 3 imprevistos con los que no habíamos contado:

1. Contar con SSR nos va a suponer tener **un aumento en gastos por infraestructura**. Nuestras aplicaciones SPA, cuando se encuentran empaquetadas, se comportan como estáticos que pueden ser servidos desde un CDN que tengamos contratado. Si necesitamos SSR, mínimo necesitamos una máquina que ejecute NodeJS para que se realice el sistema de renderizado en servidor.
2. Delegar este renderizado a servidor nos**va a suponer una mayor carga en recursos**. Lo que quizá provoque que nuestro sistema escale peor y que necesitemos más CPU o memoria en casos de un uso intensivo del sistema.
3. Algunas librerías de cliente que vayamos a usar, puede que no se lleven bien con el SSR y que**tengamos que adaptarlas para que funcionen con este sistema**. O incluso puede que ya tengamos una aplicación en vue servida como una SPA, que esté haciendo uso de hooks del ciclo de vida del componente, y que en SSR no vayan a funcionar por su propia naturaleza, teniendo que hacer adaptaciones.

Como funciona

1 imagen

Cómo sería la carga de una primera página (el servidor genera y sirve HTML / JS