

Ensayo – Almacenamiento en la web

Materia: Aplicaciones Web

Grupo: GR1COM

Estudiante: Wilman Armijos

Fecha: 06/08/2021

Dentro del desarrollo de aplicaciones web, existen algunas opciones para almacenar los datos que son utilizados sobre una aplicación. Muchas veces, las aplicaciones necesitan de un medio para almacenar datos los cuáles se espera perduren a lo largo del tiempo, sea por un periodo corto o largo de tiempo, según sea el caso (credenciales de sesiones, recordatorios de búsquedas, información generada dentro de una sesión pero que se elimina para las siguientes, entre otros).

De aquí, es que existe una variedad de opciones las cuáles permiten almacenar la información sin necesidad de ocupar espacio del servidor o del cliente, al menos no directamente, o al menos no utilizando una gran cantidad de espacio. A continuación, se van a revisar seis opciones para el almacenamiento de la información dentro de aplicaciones web.

La primera forma de almacenar datos se trata de variables de JavaScript. Como su nombre lo dice, en realidad lo único que hace es almacenar datos pequeños a través de variables de un lenguaje de programación como cualquier otro, solo que en este caso es un lenguaje orientado a aplicaciones web. La principal limitación es que la información es muy volátil, pues basta con que el usuario recargue la página para que estas variables se establezcan por defecto hasta que se les vuelva a asignar el valor que tenían antes, aunque siguen siendo opciones rápidas para almacenar datos, quizás podría ser útil para aplicaciones web que guardan y usan datos en tiempo real que no se vuelven a usar (Buckler, 2021).

La segunda forma es lo que se conoce como Web Storage, la cual puede implementarse a través de Local Storage o Session Storage. Este modo trabaja por medio de APIs que funcionan a través de JavaScript, los cuales permiten trabajar con archivos temporales que almacenan la información como si fuera una lista con un conjunto de clave-valor, donde cada para es conocido como “ítem”. En este caso, como se dijo anteriormente, existen las opciones de Almacenamiento Local o Almacenamiento de Sesión, donde el primero almacena toda la información que se requiera, mientras el segundo guarda información solo de las sesiones específicamente (Buckler, 2021).

La tercera forma es algo más completa y se conoce como IndexedDB. Se trata de una base de datos de tipo NoSQL, la cual permite que los datos sean almacenados de forma ordenada y, como su nombre lo indica, por medio de índices, que hacen más fácil buscar y extraer su información de nuevo a la aplicación cuando esta la necesite. Al ser un tipo de base de datos NoSQL, su principal característica es que las operaciones de lectura y escritura se vuelven muy rápidas, además que ofrece una capacidad de almacenamiento muy grande. Otra de las características, que es su desventaja más grande, es que puede llegar a ser un poco complejo de manejar, pues las APIs que nos permiten utilizar este método cuentan con muchas funciones algo complejas que son necesarias para poder conectarse, leer, escribir y manejar estas operaciones (Buckler, 2021).

Otra de las opciones es lo que se conoce como CacheAPI. Esta opción se centra principalmente en el almacenamiento de información que es enviada a través de las HTTP Request, es decir, a través de datos que son enviados desde el servidor hacia el cliente. Estos se caracterizan por almacenar información de red de una aplicación progresiva, esto se hace para que cuando un usuario haya utilizado y trabajando con algunas características en línea de la aplicación, esta pueda seguir siendo utilizada, o al menos en parte, cuando el usuario no se encuentre conectado a internet; por ejemplo, se tiene Google Maps o Facebook (Buckler, 2021).

La siguiente opción a ser descrita es File System Access API. Esta opción se basa principalmente utilizar el sistema de archivos del sistema operativo sobre el cual está siendo utilizada la aplicación

para leer, escribir, crear o borrar archivos y la información dentro de la misma. Estos trabajan a través de los conocidos “handlers”, los cuáles permiten efectuar operaciones directamente sobre el sistema operativo. Trabajar con este método otorga las mismas facilidades que trabajar desde dentro del computador, pues los archivos se quedan hasta que el usuario los limpie, al igual que la cantidad de datos que se pueden almacenar depende del espacio dentro del computador. Sin embargo, el principal problema, es que esta solo funciona para el navegador Google Chrome (Buckler, 2021).

La última opción de almacenamiento es quizás la más conocida, pues son las conocidas Cookies. Estas cookies son opciones existentes dentro de JavaScript, las cuales son principalmente utilizadas para mantener la información de sesiones, así como de elementos que hayan sido seleccionados o ingresados y posteriormente enviados al servidor. Estas trabajan a través de un par clave-valor, el cual permite luego poder trabajar con dicha información. De estas herramientas, quizás algo que se puede decir es que muchas veces vemos que las cookies suelen no ser aceptadas por el usuario, haciendo que su definición a veces sea innecesaria, aunque no se puede saber cuándo. Además, su capacidad es limitada, y solo puede almacenar cadenas de caracteres, haciendo que su posterior manipulación también lo sea (Buckler, 2021).

Bibliografía

Buckler, C. (2021). *10 Client-side Storage Options and When to Use Them*. Site Point. Retrieved 20 August 2021, from <https://www.sitepoint.com/client-side-storage-options-comparison/>.