

Cápsula 4: *Join* de datos en D3.js II

Hola, bienvenidxs a una cápsula del curso Visualización de Información. Continuaré hablando del *join* de datos de D3, y de más formas de usarlo.

Hasta el momento hemos visto ejemplos de código que se ejecutan una única vez, por lo que las ideas de *enter*, *update* y *exit* pueden no tener tanto sentido considerando que siempre pasaría lo mismo con un documento inicial.

El diseño de estas selecciones es viendo hacia adelante en la eventualidad de que el **contenido del documento cambie durante la vida de una visualización**. Un usuario puede filtrar datos, por lo que los elementos desaparecen. Nuevos datos llegan, por lo que hay que mostrarlos en pantalla. O los datos cambian, por lo que hay que actualizar las marcas mostradas.

El código en pantalla muestra una función que realiza un *join* de datos y afecta a las tres selecciones resultantes. Genera el *update* mediante el *join*, luego extrae *exit* y remueve los elementos sobrantes, luego extrae *enter* y agrega rectángulos para los datos sin pareja.

Finalmente se llama el método “merge”, que permite unir dos selecciones. En este caso una *update* y *enter* de manera que a todos los elementos que sí están en pantalla se les ajusten sus atributos visuales bajo las mismas reglas.

La función depende de un argumento “datos”, que comienza como un arreglo con cuatro valores. La función se llama una primera vez, y se llamará cada vez que se haga clic sobre un párrafo en el documento. Si ejecutamos esto tal como está, no se percibe efecto porque una vez agregados los rectángulos en la primera llamada, estos no cambian.

Pero, si ahora agregamos números al arreglo de datos con cada clic, podremos ver que en cada llamada se hace una actualización y aparecen más rectángulos. Esto es porque en cada llamada, se encuentra un nuevo dato sin rectángulo asociado que necesita emparejar en la selección *enter*.

Ahora en cambio, si en cada llamada eliminamos un elemento de datos, vemos lo siguiente. En cada clic desaparece un rectángulo, ya que caen dentro de la selección *exit* al no contar con datos al cual vincularse y se remueven.

Si eliminamos el último elemento y lo agregamos al comienzo del arreglo de datos, veremos que se actualizan las barras, desplazándose en largo. **Esto es porque como el orden de datos cambia, y en cada llamada los rectángulos se asocian con un valor distinto, por lo que se actualizan sus alturas en la selección *update***. Esta forma de emparejar mediante el orden es el predeterminado, y veremos más adelante como alterarlo.

El llamar a funciones como estas, que manejan tanto *update*, *enter* como *exit*, son un patrón muy frecuente de programas en D3. Tanto así, que en la **versión 5** se introdujo un método que permitía reducir este código que tan frecuentemente ocurría, el método "join".

En pantalla se muestra el código mediante este método que hace lo mismo que el código antes mostrado. Al llamarlo después de "data", es decir sobre la selección *update*, **automáticamente** remueve elementos de *exit*, agrega elementos especificados en el argumento, y **retorna la unión de *enter* y *update*** para cambiar sus atributos visuales.

De forma alternativa, es capaz de recibir hasta tres funciones que especifiquen qué hacer con cada selección del *join*. Primero *enter*, luego *update* y finalmente *exit*. En pantalla se muestran las funciones predeterminadas equivalentes al ejemplo anterior.

Nuevamente quiero mencionar que probablemente estas cápsulas sean mucho para entender. Te sugiero revisar los ejemplos de código subidos y los ejercicios propuestos para ponerlos en práctica.

Con eso termina el contenido de esta cápsula. Recuerda que si tienes preguntas, puedes dejarlas en los comentarios del video para responderlas en la sesión en vivo de esta temática. ¡Chao!