Práctica Modern Visualization y D3.js

Guillermo Barrio

Nota explicativa sobre la práctica. En esta nota se explica muy brevemente el contenido de la práctica de *Modern Visualization* D3.js, que incluye el fichero principal de código JavaScript.js, (*practica-modernVisualization-Guillermo_Barrio.js*), el *main.html*, de estilos *main.css*, y el de los propios datos, este último situado en el mismo directorio que el resto. El fichero principal incorpora comentarios bajo '//' que complementan esta nota.

Dos partes en la práctica: mapa y gráficos. La práctica se puede dividir en dos partes relacionadas entre sí: **1)** coloración del mapa; y **2)** gráfico de cada barrio. A continuación, se explica el proceso de programación de cada una de ellas.

Coloración del mapa

Se hace un pequeño tratamiento de datos en el caso de que no figure el alquiler medio. Se ha tomado como base el fichero seguido en el curso 7_map.js, modificando el path para la toma de datos. A continuación, he recogido los datos del precio de alquiler de cada barrio. Lo que ocurre es que hay barrios donde no figura este dato, concretamente cinco, lo que comprobamos mediante el método count() con la condición que el dato de alquileres no sea *null*. Estos cinco barrios tendrán un tratamiento especial a la hora de la coloración. Mediante el método extent() obtenemos el intervalo de los alquileres (15-280 euros).

Por sencillez y claridad se ha tomado una gama de colores naranjas para los datos del alquiler, y rojo para cuando estos no existen. Para la coloración se ha tomado la escala de naranjas que se utilizó en el heatmap. Lo cierto es no encontré en el *palette* de D3 ninguno que me convenciese, y pensé que había que utilizar un solo color con una gama, en este caso naranja, y no usar varios. Esta escala de color se aplica a los barrios que tienen datos; a los cinco en los que no figura el precio del alquiler medio se les ha asignado el color rojo.

Solución quizás pedestre, pero efectiva, para la leyenda. Para la leyenda del mapa he procedido de una forma algo pedestre pero que no creo que suponga una pérdida de generalidad. He fijado inicialmente el número de rectángulos en diez, pero que es posible modificar. A continuación, se crea el array *listaPreciosRect* mediante código común de JS que almacena los límites numéricos de los alquileres que definirán el color de cada rectángulo, y que nos servirá también para identificarlos. Finalmente, incorporo al lienzo los rectángulos ya coloreados introduciendo en data() el propio array. Podría quizás haber utilizado para calcular el array el método tick(), pero al tenerlo ya calculado he decidido dejarlo así. Para los barrios sin datos de alquiler he añadido un rectángulo de color rojo y un texto que lo explica.

Tooltip con nombre y alquiler de cada barrio para conectar más cómodamente el mapa con los gráficos. Dentro del mapa se ha incorporado un tooltip, activado con el evento mouseover, con una pequeña animación, que identifica al barrio y el precio del alquiler, ya que pienso que es imprescindible para conectar de una manera comprensible el mapa coloreado con las gráficas de apartamentos/dormitorios.

Gráfico de cada barrio

El gráfico de apartamentos y dormitorios se sitúa en un lienzo independiente al del mapa coloreado. Mi intención inicial con el gráfico de cada barrio era integrarlo dentro del mapa coloreado; de hecho, fui capaz de dibujar los ejes. Sin embargo, por alguna razón, el sistema no reconocía el código donde definía y coloreaba los rectángulos. Tras infructuosos intentos, decidí crear el gráfico en otro lienzo, identificado como 'mapid'.

Los gráficos se activan mediante un evento click sobre el mapa coloreado. El gráfico se activa como un evento tipo click en el barrio correspondiente en el mapa coloreado. La función handleClick es la que crea el propio gráfico en su totalidad. Comienzo para ello recogiendo los datos principales de cada barrio, tales como su nombre y su distribución de apartamentos y dormitorios, entre otros, que se incorporan en la parte inferior del gráfico.

Gráfico de barras con código de colores, tooltip y datos adicionales. El gráfico es uno típico de barras, para el que se definen de forma habitual los ejes y escalas. Los rectángulos se colorean según el criterio siguiente: rojo, para el número de dormitorios con mayor número de apartamentos; azul, para el de menor número, que en muchas ocasiones es cero; y verde, para el resto de casos. En vez de los datos en la parte superior de los rectángulos pienso que queda mejor otro tooltip, también activado con mouseover, donde informa del número de dormitorios y apartamentos. En la parte inferior del gráfico se añaden el nombre del barrio, el número de apartamentos y el número de dormitorios medio de éstos.

No se borra el gráfico inicial tras hacer un segundo click. Se ha producido un fenómeno curioso: cada vez que hago click por segunda vez en un barrio del mapa coloreado se me crea correctamente el gráfico correspondiente, pero no me borra el anterior, sino que se muestra a la derecha. He intentado utilizar el método remove() y, efectivamente, borra el gráfico original, pero sigue creando el nuevo a la derecha. En los metadatos se muestra, de todas formas, que las coordenadas x del eje y las de los rectángulos son las que corresponderían a una posición correcta. Es posible que todo ello tenga que ver con el propio proceso de click, o que pueda solucionarse a través de algún método de html, pero no lo he encontrado. Con todo, este fenómeno puede ser útil para comparar los datos de dos o más barrios.