

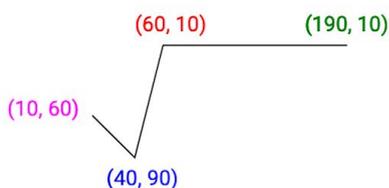
## Cápsula 2: Marcas en D3.js I

Hola, bienvenidxs a una cápsula del curso Visualización de Información. Esta corresponde a la primera parte sobre la generación de marcas en D3.js.

Hemos revisado ya como organizar información espacialmente, y hemos visto algunos ejemplos simples de uso de marcas. Específicamente, hemos usado rectángulos para generar barras, pero no todo se puede armar en base a rectángulos.

Por eso D3 provee un montón de facilidades para generar marcas de forma fácil y efectiva a partir de datos. Muchas de estas se basan en aprovechar el elemento SVG **“path”**. Mencioné brevemente este elemento en la cápsula de SVG, pero no fui en mucho detalle.

```
<path d="M10,60L40,90L60,10L190,10" fill="none" fill="black"></path>
```



En términos simples, este elemento se basa en definir un **camino**. Se define en un atributo **“d”**, y se especifica mediante una notación muy particular **de coordenadas y líneas**. Puede usar líneas rectas entre coordenadas específicas, líneas curvas, puede formar una figura cerrada, o figuras abiertas. ¡De todo!

Partiremos por un ejemplo simple de funciones de utilidad de D3 para generar marcas más complicadas. Primero está **“symbol”** que provee una forma de generar caminos para marcas pequeñas. Al llamar a **“symbol”** se retorna una función que genera caminos, y llamando a **“size”** podemos especificar el tamaño aproximado de área que tendrá la figura generada.

El retorno **es una función que genera caminos**, por lo que es necesario llamarlo. Si imprimimos lo que retorna al llamarse, podemos ver una cadena de texto del formato para declarar caminos de elemento **“path”**.

Eso por sí solo no dice mucho, así que lo asignaremos a un elemento **“path”**, y definiremos su atributo camino **“d”**. Por defecto, el generador dibuja figuras alrededor del origen coordenado, por lo que aprovecharé de trasladarlo a la coordenada (100, 100) para que veamos la figura resultante. Si ejecutamos el código, ¡vemos un círculo!

**Medio fome**, un círculo es la figura predeterminada que retorna el generador "symbol". Si llamamos al método "type" y le entregamos alguno de los tipos provistos por D3, podemos ver otras figuras. Aquí mostraré varios ejemplos de figuras más complicadas: una cruz, una estrella y un cuadrado. ¡Con esto ya podemos generar un gráfico de dispersión con otras marcas además de puntos!

Otra función para generar caminos es "line", cuyo propósito es generar líneas, como para gráficos de línea. Similar al caso de "symbol", esto retorna una función que al ser llamada retorna un camino que define una línea. Pero a diferencia de "symbol", **esta recibe un argumento**. Específicamente, recibe un arreglo puntos, que especifican las coordenadas X e Y que guiarán a la línea generada.

En el ejemplo en pantalla, tengo un arreglo de puntos para usar. Similar al caso de "symbol", imprimiré el resultado del camino y lo usaré para definir un elemento "path" SVG.

Es importante notar que por defecto el elemento "path" rellena con color el área ocupada por la línea. Como ahora solo queremos el contorno del camino, dejaré el relleno transparente. También de forma separada agregaré un círculo por cada punto entregado, para que veamos sus posiciones.

Si lo probamos, podemos ver la línea resultante. Si inspeccionas punto por punto, notarás que las coordenadas coinciden entre los puntos entregados y el resultado. **Nota que los puntos que le entregamos son ya puntos en el sistema coordenado, no son valores cualquiera.**

También, nota que se conectan puntos consecutivos en el **orden de puntos entregados**. Queda de forma agradable gracias a que los puntos estaban ordenados por su coordenada X.

Por defecto las líneas que conectan puntos consecutivos con una línea recta, pero esta no es la única forma de generar líneas. Mediante el método "curve" es posible definir el tipo de curva que usará para conectar puntos. Mostraré algunos ejemplos.

"curveBasis" por ejemplo genera una curva cúbica en base a los puntos entregados. Si bien el resultado es más agradable, notamos que no cruza exactamente cada punto entregado.

"curveCardinal" también es una curva cúbica que D3 provee, que si pasa por los puntos entregados y es suave. Pero en algunas vecindades de puntos llega a lugares que no son tan naturales. Por ejemplo, hay un puntos máximos donde la curva los exagera, y hay partes donde la altura de puntos consecutivos es idéntica, pero la curva genera una subida de todas formas.

**Hay detalles así a notar cuando pruebas con distintas curvas, ya que una mala elección puede alejar el *lie factor* de uno en una visualización.** "curveMonotoneX" es una curva que mantiene monotonía en la dirección Y, y no aumenta o disminuye cuando no ocurre en los datos.

También hay variaciones de líneas rectilíneas, como "curveStep" que genera escalones discretos entre valores de datos.

Ahora, un detalle importante es que los puntos entregados hasta ahora son coordenadas donde estamos dibujando y no datos cualquiera. Ya sabemos ocupar escalas que nos permiten transformar datos cualquiera a coordenadas, pero si quisiéramos ocuparlas con generadores de líneas, significa que debemos pre-procesar nuestros datos.

**Los generadores de líneas proveen formas de generar este comportamiento incluido**, en métodos "x" e "y". Cada uno recibe una función de acceso a los elementos que se le entreguen al ser llamada, **de forma que pueda extraer el valor de las coordenadas de cada punto**. Con esto, es posible entregar directamente un arreglo de datos no procesado, y que genere una línea con sentido en donde estemos dibujando.

En el ejemplo en pantalla, contamos con un arreglo de objetos con dos atributos cada uno. Se generan escalas en base a esos valores. Las coordenadas X en base al atributo "paso" y las coordenadas Y en base a "valor". Luego en la definición de generador de líneas con los atributos mencionados se especifica que se calcularán X e Y en base a los valores de los atributos y a las escalas recién definidas.

Si lo probamos, ¡vemos que funciona! Te invito a aprender más de líneas, que está en el paquete "d3-shape" de D3. En este también se encuentra "symbol", más definiciones de curvas como las que revisamos, y otros generadores que revisaremos en las siguientes cápsulas.

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!