

# Modern Exploration & Visualization — D3.js

Javier Sánchez Gutiérrez



# ■ ¿Qué vamos a ver?

- D3.js ¿Por qué y para qué?
- Introducción SVG.
- Conceptos D3
  - Set up rápido
  - Selectores en D3.
  - Introduciendo datos
  - Escalas en D3.
  - Interacciones y Animaciones.
- Visualizaciones:
  - Nube de puntos
  - Gráfica de barras
  - Pie chart
  - Grafico de líneas
- Geo Visualización
  - Geo funciones en D3
  - D3 + Leaflet





- D3 (**D**ata **D**riven **D**ocuments)
- Librería Javascript para visualización de datos
- Éxito debido a su **FLEXIBILIDAD**.



# D3.js

Ejemplos: <https://d3js.org/> API: <https://github.com/d3/d3/blob/master/API.md>

Ejemplos sencillos: <https://bl.ocks.org/>

[Overview](#) [Examples](#) [Documentation](#) [API](#) [Source](#)

 Data-Driven Documents

Fork me on GitHub



## Algunos ejemplos:

<https://archive.nytimes.com/www.nytimes.com/interactive/2012/02/13/us/politics/2013-budget-proposal-graphic.html>

<https://www.jasondavies.com/wordtree/?source=steve-jobs-commencement.txt&prefix=l&reverse=0>

<https://observablehq.com/@mbostock/geodesic-rainbow>

<https://www.alhadaqa.com/wp-content/uploads/2020/04/horrified.html>





# Cuando usar D3



**No.** Son audiencias y usos diferentes.

Tableau: Herramienta BI

D3: Para Visualización Web

*Dos use cases:*

Diseño de productos web

Visualizaciones con mucha customización

...



# Diseño de productos web. Ejemplo I

Empresa medica lanza un SaaS que ayuda a doctores a monitorizar datos médicos (EHR).

The screenshot shows a web-based EHR system. On the left is a sidebar with navigation links like 'Patient List', 'Registry', 'Patient Data', 'Explore', 'Profile', 'Data Review', 'Summaries', 'Immunizations', and 'Care Team'. The main area displays 'General Preventative Care' and 'Vaccinations' for a patient. A tooltip is visible over the 'Breast Ca Risk' section, listing USPSTF guidelines for genetic testing for hereditary breast cancer, including criteria like '2 first-degree relatives with breast cancer' and 'Ashkenazi Jewish with any first/second-degree relatives with breast or ovarian cancer'.

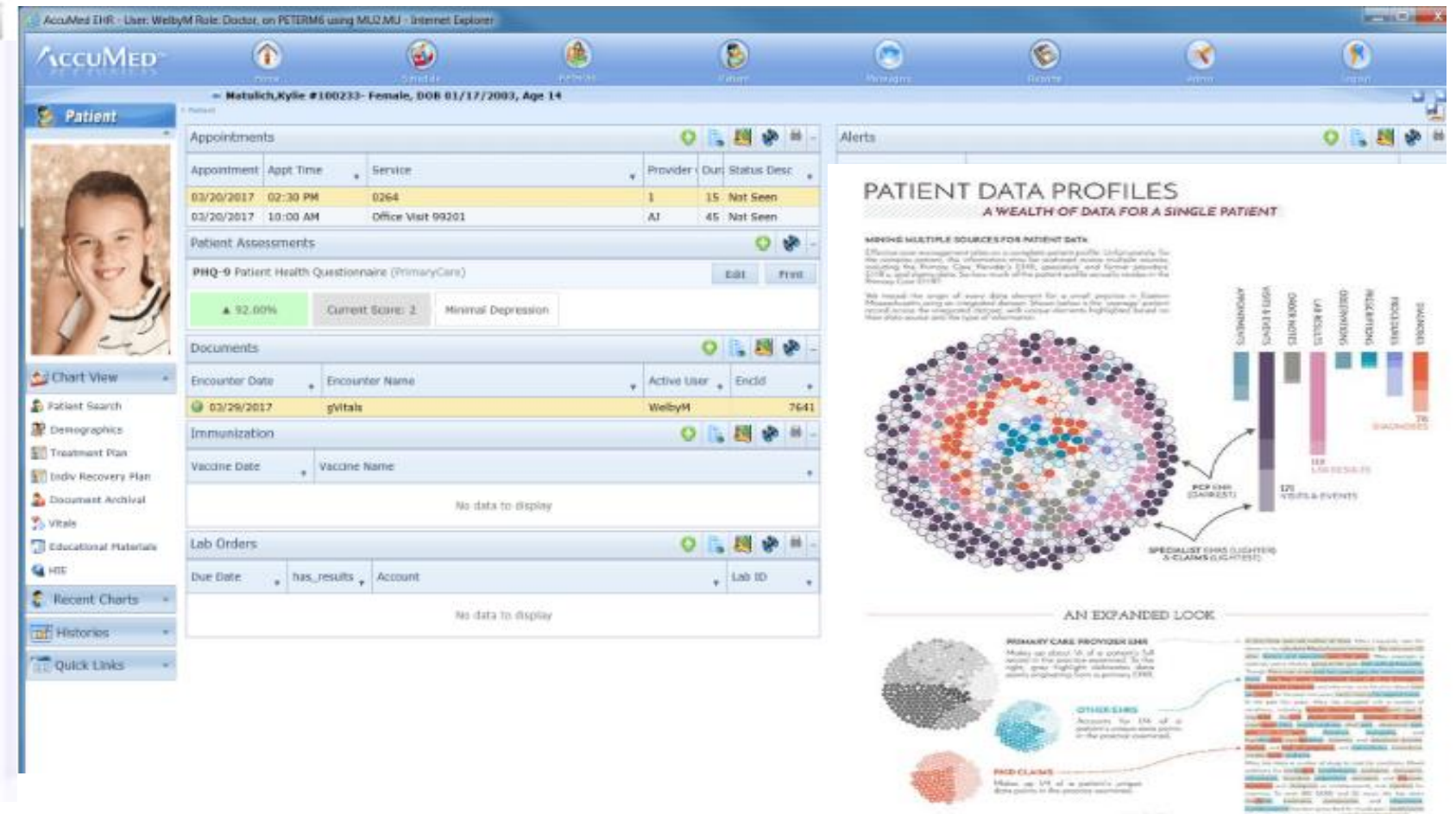
The screenshot shows the 'AccuMed EHR' interface for a user named 'WelbyM'. It displays a patient profile for 'Metulich, Kyle #100233- Female, DOB 01/17/2003, Age 14'. The interface includes sections for 'Appointments', 'Patient Assessments' (showing a PHQ-9 score of 2), 'Documents', 'Immunization', 'Lab Orders', 'Alerts', 'Problem List', 'Medication', 'Vitals', and 'Allergies'. The 'Vitals' section shows a table with columns for Date, Height, Weight, BMI, BP Systolic, BP Diastolic, Temperature, and Enc ID, with data for two dates: 03/29/2017 and 10/18/2018.

Novedad: Decide incorporar  
visualización de datos.  
Posible uso de D3.js



# Diseño de productos web. Ejemplo I

Empresa medica lanza un SaaS que ayuda a doctores a monitorizar datos médicos (EHR).





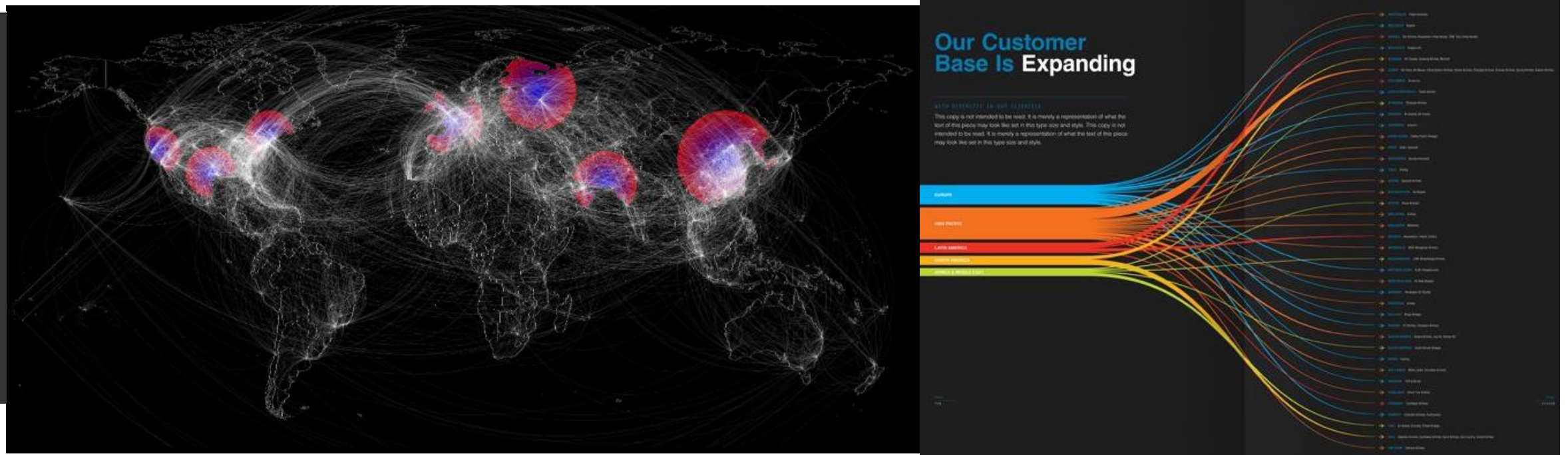
# Visualizaciones con mucha customización. Ejemplo II

Como parte de su rebranding una aerolínea decide publicar un informe Web con sus resultados en años anteriores

Aircraft List 80 of 92 Aircraft

Airbus A380-800	Volume: 22	Utilization: 83%
	#Flights: 8,827	Airtime: 59%
Airbus A340-600	Volume: 17	Utilization: 59%
	#Flights: 6,243	Airtime: 76%
Boeing 747-8	Volume: 14	Utilization: 72%
	#Flights: 6,826	Airtime: 51%
Airbus A340-300	Volume: 13	Utilization: 88%
	#Flights: 1,861	Airtime: 48%
Airbus A330-200	Volume: 11	Utilization: 90%
	#Flights: 9,431	Airtime: 65%
Boeing 747-400	Volume: 5	Utilization: 78%
	#Flights: 2,983	Airtime: 74%

Below target Above target Total Ordered

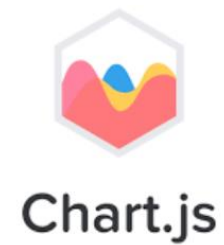


¿Por qué es importante la visualización de datos para una marca?

# (Otras) Librerías JS para Visualización de datos

¿Cuáles son las alternativas a D3.js para visualización de datos JS?

- Pros:
- Layouts o diseños ya contruidos
  - Ahorras tiempo
  - Facilidad para integrar Vzs en frameworks



`<Recharts />`

... [Comparativa](#)

- Pros:
- Control total de la Visualización.
  - Importación de módulos
  - Funcionalidades para tratamiento de datos
  - Animaciones complejas



# ¿Qué es SVG?

Scalable **V**ector **G**raphics ( Gráficos escalables en formato vectorial)

- Es un formato que usa **texto** para definir imágenes



**Etiquetas** similares a las que usamos en lenguaje HTML

- Compatible con HTML (IE >= 9)
- Conceptualmente es un **lienzo** sobre el que incluir elementos gráficos

`<svg width='800' height='500'></svg>`



# SVG

```
<svg width='800' height='500'>
```

...

```
</svg>
```

Dentro de nuestro lienzo  
podemos incluir elementos  
gráficos





# SVG ¿Cómo es nuestro lienzo?

```
<svg width='800' height='500' > ...</svg>
```

Por defecto los elementos se dibujan a partir del pixel 0,0



¿Cuáles son los elementos que podemos incluir en nuestro lienzo svg? [Wiki](#)



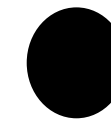
# SVG

- Elementos gráficos básicos

Rectángulo: **rect**



```
<rect x="100" y="100" width="100" height="40" />
```



Círculo: **circle**

```
<circle cx="250" cy="25" r="25" />
```



# SVG

- Elementos gráficos básicos

Elipse: **ellipse** 

```
<ellipse cx="250" cy="25" rx="100" ry="25">
```

Línea: **line** 

```
<line x1="0" y1="0" x2="500" y2="50" stroke="black"/>
```



# SVG

- Path: elemento grafico mas poderoso en SVG

Nos permite crear formas complejas combinando multiples lineas y curvaturas.

	Move to		Line to		Line to		Close path
<path	d="M	100	100	L	300	100	L 200 300 z"
	fill="blue"/>						

[Wiki](#)

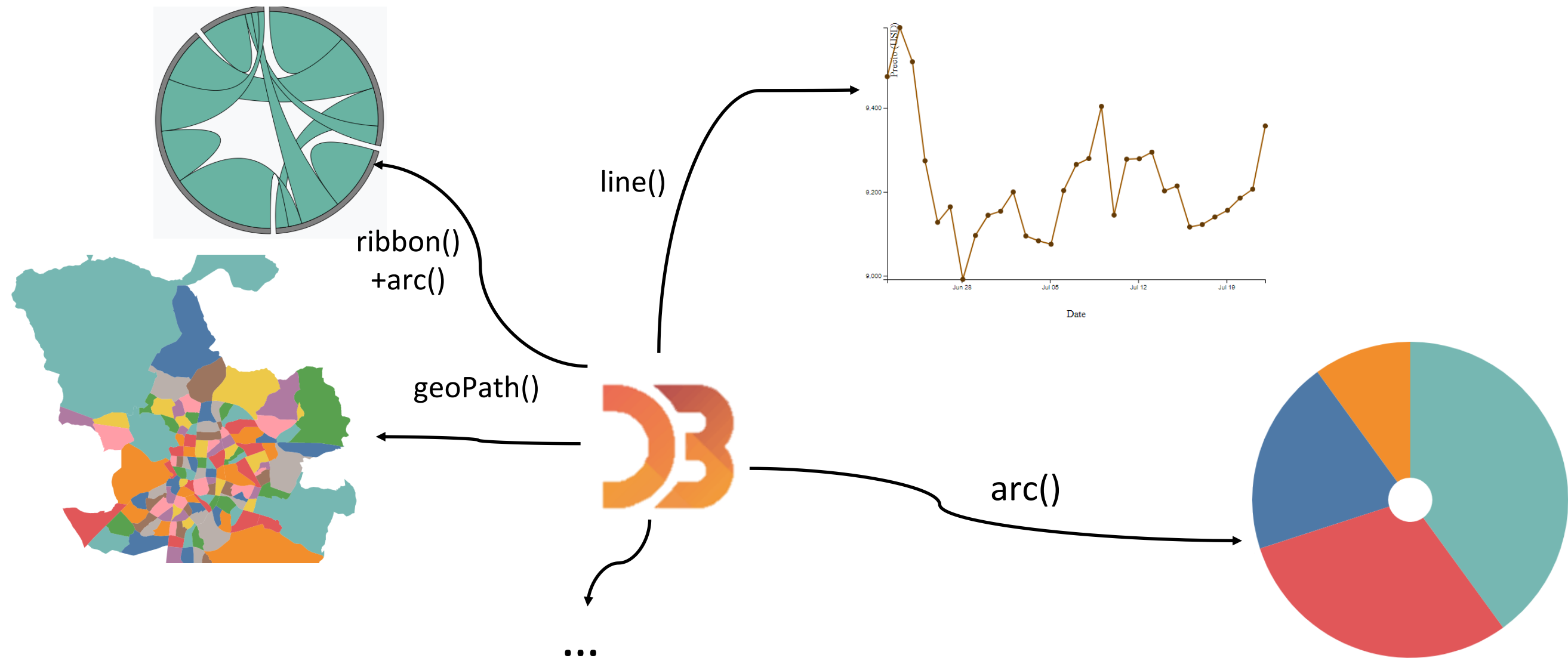




# SVG Paths : ¿Como D3 crea paths?



Mediante varios métodos:



# SVG

Otros elementos

- Elementos textuales

Texto: **text**

```
<text x="250" y="50" font-family="sans-serif" font-size="25" fill="gray">Prueba</text>
```

**Elemento g:** Elemento contenedor usado para agrupar objetos. Las transformaciones aplicadas al elemento **g** son realizadas sobre todos los elementos hijos del mismo.

- `<g></g>`



# SVG – Estilos y transformaciones

## Fill

- Por nombre – Orange
- Valor HEX – #3388aa o #38<sup>a</sup>
- Valor RGB – rgb(10, 150, 20)
- RGB con transparencia alfa – rgba(10, 150, 20, 0.5)

**stroke:** Un valor para el color

**stroke-width:** Una medida numérica (pixels)

**opacity:** Un valor numérico entre 0 y 1

**transform:** Aplicar una transformación en el eje x o y (rotación, traslación...)



# D3 controles básicos

- **select** (...) - Devuelve el primer elemento que contenga un **selector** CSS específico
- **selectAll** (...) – Devuelve todos los elementos que contengan un **selector** CSS específico



## Selectores

Selector de ID (#id)

Selector de elemento (g, circle...)

Selector de clase (.class)





# D3 controles básicos

- `append(...)` – Crear un elemento (y seleccionarlo)
- `attr(... , ...)` - Cambiar/añadir/seleccionar un atributo de un elemento
- `style(... , ... )` - Cambiar/añadir/seleccionar el estilo de un elemento



# D3 y los datos

```
d3.select("div")  
  .selectAll('p')  
  .data(dataset)  
  .enter()  
  .append('p')  
  .text((d) => {  
    return d;  
  })
```

Selección de elementos.      *Selecciona todos los elementos **p** dentro del elemento **div***

Asocia nuestros datos a la selección

Crea tantos elementos como datos tengamos

Añadimos los elementos p creados a la selección

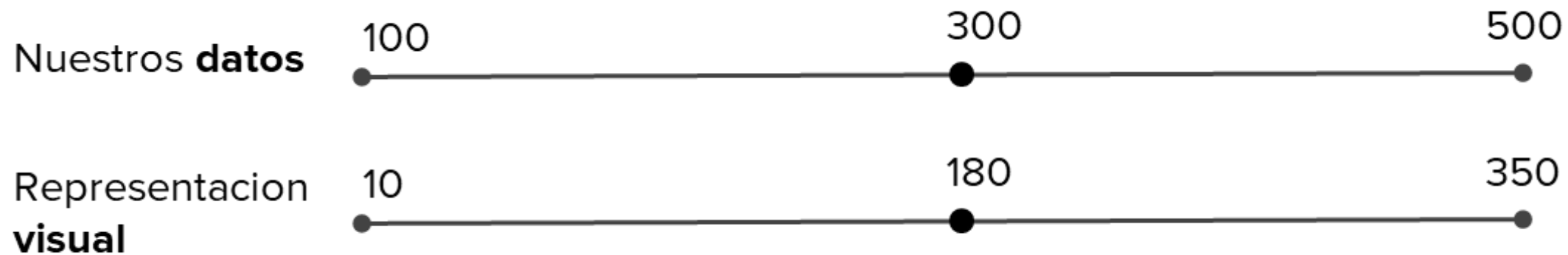
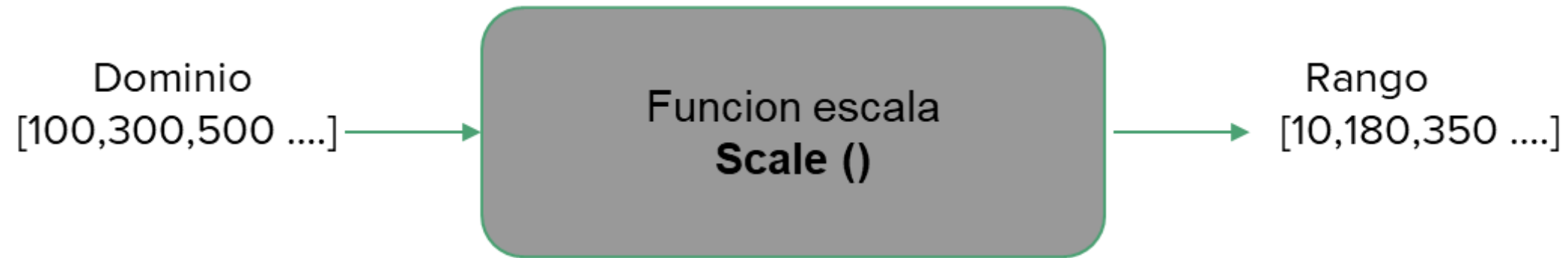
Añadimos funcionalidades y atributos a los elementos



# Escalas y ejes

## Escalas

Transformar **unos datos** de entrada en una **representación visual** de salida  
Transformar **unos datos** de entrada en **píxeles** de salida



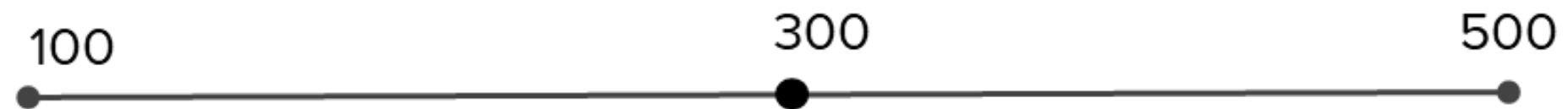
→ Número de píxeles de ancho disponibles?



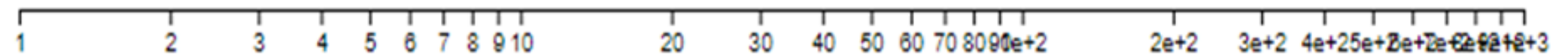
# Escalas.

Datos de entrada  
continuos

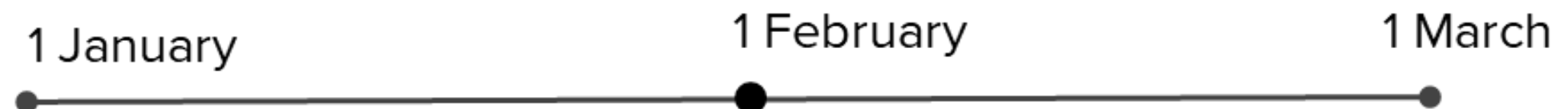
## LINEAL (scaleLinear)



## LOGARITMICA (scaleLog)



## TEMPORAL (scaleTime)



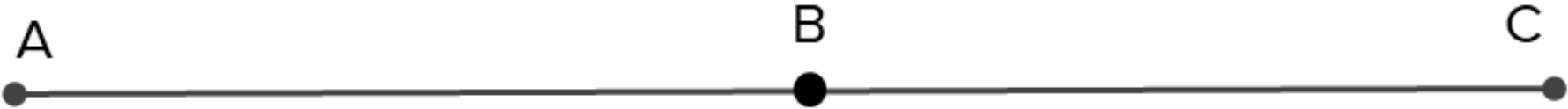




# Escalas.

Datos de entrada  
discretos

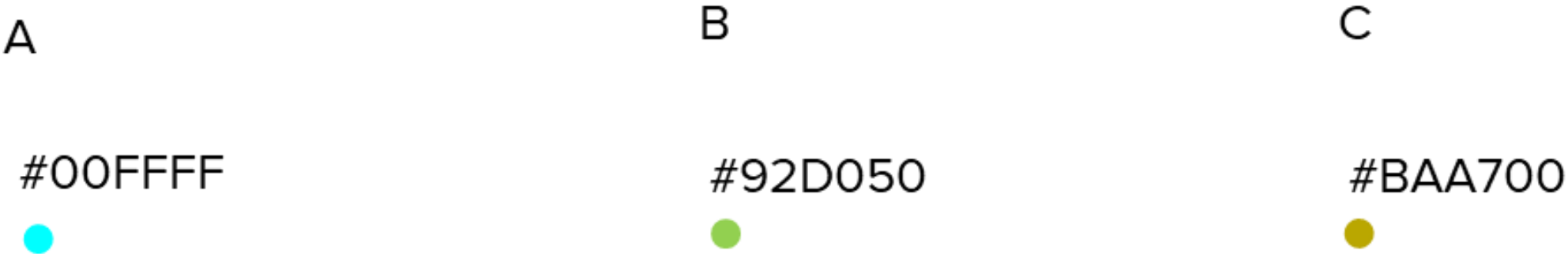
## Discreta (scalePoint o scaleBand)



## (scaleOrdinal)

Datos de entrada  
discretos y salida  
discreta.

Populares como escalas de color



Más escalas: [Wiki](#)



# ■ Interacciones

Método `on` : Para capturar eventos de ratón sobre elementos

Una selección  
`d3.select(...)`  
`d3.selectAll(...)`  
+

```
.on("mouseover", ... )  
.on("mouseout", ... )  
.on("click", ... )  
.on("zoom", ... )
```

```
function(...){  
  d3.select(this)  
    .attr(..., ...)  
}
```

Selecciona el elemento actual para aplicarle cambios

`event`: Capturamos el evento.

`d`: Datos asociados a un elemento.

Otros eventos: [Wiki](#)

`d3.select(this)`: Seleccionamos el elemento



# ■ Animaciones

`d3.select("#target")` -----> Seleccionamos elemento a cambiar

`.transition()` -----> Método que indica el comienzo de la transición.

`.ease(...)` -----> *Opcional*: Tipo de función de transición que se aplica. [Comparativa](#)

`.duration(1000)` -----> Duración de la transición. En milisegundos(ms).

`.delay(200)` -----> *Opcional*: "Tiempo offset de retraso": Cuanto hay que esperar hasta que empiece la transición. En ms.

`.attr(... , ...)`

↓

Atributos a cambiar



## 1. GeoJSON

Formato JSON-based usado para describir información geográfica

```
{ "type": "FeatureCollection",  
  "features": [  
    {  
      "type": "Feature",  
      "properties": {  
        "name": "Embajadores"  
      },  
      "geometry": {  
        "type": "Polygon",  
        "coordinates": [[  
          [-3, 40.1], ... , [-3, 40.12]]  
        ]  
      },  
      "type": "Feature",  
      ...  
    }  
  ]  
}
```

Array de Objetos ("features") a representar

**Properties:** Metadatos. Almacenamos nombre, id... u otra información sobre la "Feature" a representar

**Geometry:** Array de coordenadas [lon,lat] del objeto a representar (Polygon\*)

\* Puede haber varios tipos de geometría dependiendo del objeto: Polygon, MultiPolygon, Point, MultiPoint, LineString, MultiLineString...





# GeoVisualización Mapas y D3.js

## 2. Proyecciones

Proyectar puntos de una esfera sobre una superficie plana  
Transformar unas coordenadas [long,lat] en valores [X,Y] (píxeles)

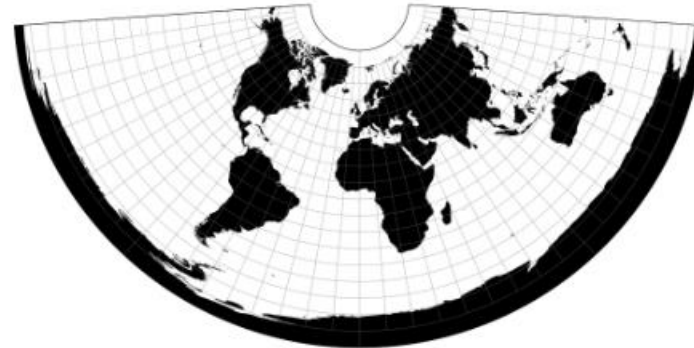
En D3 existen muchas funciones de proyección



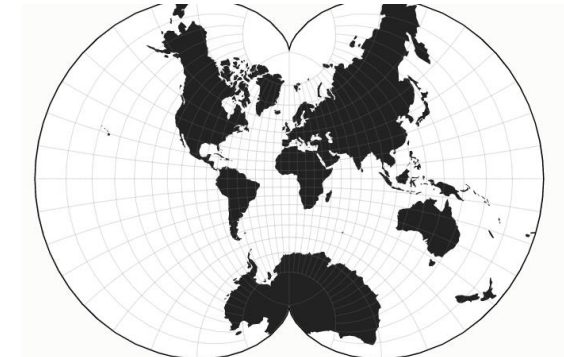
Ortographic



Mercator



Conic



August

...







# GeoVisualización Mapas y D3.js

## 3. GeoPath()

Método que convierte proyecciones en elementos gráficos path.

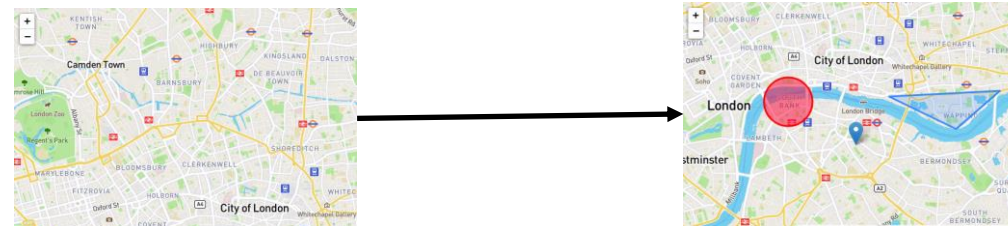
```
var pathProjection = d3.geoPath().projection(projection);
```



# GeoVisualización D3.js y Leaflet



Librería JS open-source que permite incluir mapas interactivos en Webs/apps



Usamos D3 para añadir capas sobre los mapas.



Todas las funcionalidades de d3 en webmapping

Para importar Leaflet, en nuestro header:

```
<!--Importamos los estilos CSS asociados a Leaflet-->
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css">
<!--Importamos Leaflet -->
<script src='https://unpkg.com/leaflet@1.4.0/dist/leaflet-src.js'></script>
```

[Quick set up.](#)

