¡Hola 👋! Espera mientras comienza la sesión.

Antes que todo, ¿cómo están?

Visualización de Información

IIC2026 2020-2

Utilidades en D3.js I

Visualización de Información

IIC2026 2020-2

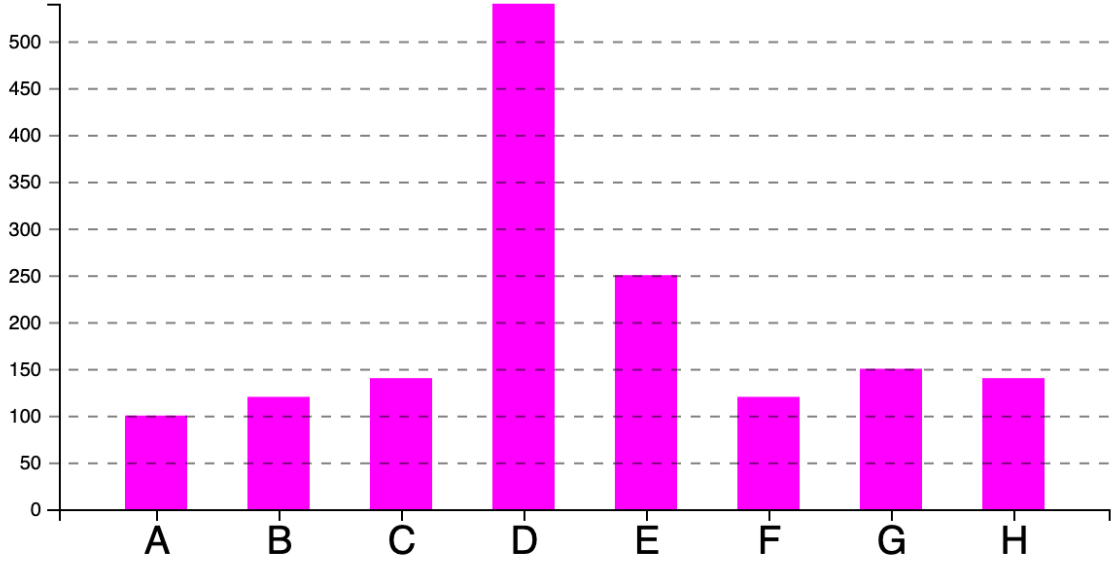
Repaso

Repaso

1. Cargado de datos en D3.js
2. Escalas de D3.js
3. Ejes de D3.js

```
const svg = d3.select("body").append("svg");

function joinDeDatos(datos) {
  svg.attr("width", 50 + datos.length * 100).attr("height", 500);

  svg
    .selectAll("rect")
    .data(datos)
    .join("rect")
    .attr("width", 50)
    .attr("fill", "magenta")
    .attr("height", (d) => d)
    .attr("x", (_, i) => 50 + i * 100);
}

const datos = [10, 20, 30, 40];
```

```
const width = 600;
const height = 400;
const margin = {
  top: 70,
  bottom: 70,
  right: 30,
  left: 30,
};

const svg = d3
  .select("body")
  .append("svg")
  .attr("width", width)
  .attr("height", height);

const contenedor = svg
```


500
450
400
350
300
250
200
150
100
50
0
A
B
C
D
E
F
G
H

```
const width = 600;
const height = 400;
const margin = {
  top: 70,
  bottom: 70,
  right: 30,
  left: 30,
};

const svg = d3
  .select("body")
  .append("svg")
  .attr("width", width)
  .attr("height", height);

const contenedor = svg
```

```
14    .attr("height", height);
15
16  const contenedor = svg
17    .append("g")
18    .attr("transform", `translate(${margin.left} ${margin.top})`);
19
20  function joinDeDatos(datos) {
21    const maximaFrecuencia = d3.max(datos, (d) => d.frecuencia);
22
23    const escalaAltura = d3
24      .scaleLinear()
25      .domain([0, maximaFrecuencia])
26      .range([0, height - margin.top - margin.bottom]);
27
28    const escalaY = d3
```

```
17    .append("g")
18    .attr("transform", `translate(${margin.left} ${margin.top})`);
19
20  function joinDeDatos(datos) {
21    const maximaFrecuencia = d3.max(datos, (d) => d.frecuencia);
22
23    const escalaAltura = d3
24      .scaleLinear()
25      .domain([0, maximaFrecuencia])
26      .range([0, height - margin.top - margin.bottom]);
27
28    const escalaY = d3
29      .scaleLinear()
30      .domain([0, maximaFrecuencia])
31      .range([height - margin.top - margin.bottom, 0]);
```

```
22
23    const escalaAltura = d3
24      .scaleLinear()
25      .domain([0, maximaFrecuencia])
26      .range([0, height - margin.top - margin.bottom]);
27
28    const escalaY = d3
29      .scaleLinear()
30      .domain([0, maximaFrecuencia])
31      .range([height - margin.top - margin.bottom, 0]);
32
33    const ejeY = d3.axisLeft(escalaY);
34
35    svg
36      .append("g")
```

```
38      .call(ejeY)
39      .selectAll("line")
40      .attr("x1", width - margin.right - margin.left)
41      .attr("stroke-dasharray", "5")
42      .attr("opacity", 0.5);
43
44    const escalaX = d3
45      .scaleBand()
46      .domain(datos.map((d) => d.categoria))
47      .rangeRound([0, width - margin.right - margin.left])
48      .padding(0.5);

49
50    const ejeX = d3.axisBottom(escalaX);
51
52    svg
```

```
56      .selectAll("text")
57      .attr("font-size", 20);
58
59    contenedor
60      .selectAll("rect")
61      .data(datos)
62      .join("rect")
63      .attr("width", escalaX.bandwidth())
64      .attr("fill", "magenta")
65      .attr("height", (d) => escalaAltura(d.frecuencia))
66      .attr("x", (d) => escalaX(d.categoria))
67      .attr("y", (d) => escalaY(d.frecuencia));
68  }
69
70  d3.json("datos.json")
71    .then((datos) => {
```

```
const width = 600;
const height = 400;
const margin = {
  top: 70,
  bottom: 70,
  right: 30,
  left: 30,
};

const svg = d3
  .select("body")
  .append("svg")
  .attr("width", width)
  .attr("height", height);

const contenedor = svg
```

```
10 const svg = d3
11   .select("body")
12   .append("svg")
13   .attr("width", width)
14   .attr("height", height);
15
16 const contenedor = svg
17   .append("g")
18   .attr("transform", `translate(${margin.left} ${margin.top})`);
19
20 function joinDeDatos(datos) {
21   const maximaFrecuencia = d3.max(datos, (d) => d.frecuencia);
22
23   const escalaAltura = d3
24     .scaleLinear()
```

```
27
28    const escalaY = d3
29      .scaleLinear()
30      .domain([0, maximaFrecuencia])
31      .range([height - margin.top - margin.bottom, 0]);
32
33    const ejeY = d3.axisLeft(escalaY);
34
35    svg
36      .append("g")
37      .attr("transform", `translate(${margin.left}, ${margin.top})`)
38      .call(ejeY)
39      .selectAll("line")
40      .attr("x1", width - margin.right - margin.left)
41      .attr("stroke-dasharray", "5")
42      .attr("opacity", 0.5);
```

```
44    const escalaX = d3
45      .scaleBand()
46      .domain(datos.map((d) => d.categoria))
47      .rangeRound([0, width - margin.right - margin.left])
48      .padding(0.5);
49
50    const ejeX = d3.axisBottom(escalaX);
51
52    svg
53      .append("g")
54      .attr("transform", `translate(${margin.left}, ${height - margin.
55      .call(ejeX)
56      .selectAll("text")
57      .attr("font-size", 20);
58
```

```
33    const ejeY = d3.axisLeft(escalaY);
34
35    svg
36      .append("g")
37      .attr("transform", `translate(${margin.left}, ${margin.top})`)
38      .call(ejeY)
39      .selectAll("line")
40      .attr("x1", width - margin.right - margin.left)
41      .attr("stroke-dasharray", "5")
42      .attr("opacity", 0.5);
43
44    const escalaX = d3
45      .scaleBand()
46      .domain(datos.map((d) => d.categoria))
47      .rangeRound([0, width - margin.right - margin.left])
```

```
48      .padding(0.5);
49
50    const ejeX = d3.axisBottom(escalaX);
51
52    svg
53      .append("g")
54      .attr("transform", `translate(${margin.left}, ${height - margin.
55      .call(ejeX)
56      .selectAll("text")
57      .attr("font-size", 20);
58
59    contenedor
60      .selectAll("rect")
61      .data(datos)
62      .join("rect")
```

`d3-fetch`

- `d3.csv(archivo)`: Para leer archivo CSV.
- `d3.json(archivo)`: Para leer archivo JSON.
- `d3.tsv(archivo)`: Para leer archivo TSV.
- `d3.dsv(delimitador, archivo)`: Para leer archivo separado por un delimitdor común.
- ...

¡Revisa la documentación!

d3-scale

- `d3.scaleLinear()`: Escalas continuas lineales.
- `d3.scalePow()`: Escalas continuas de potencia.
- `d3.scaleLog()`: Escalas continuas logarítmicas.
- `d3.scaleTime()`: Escalas que reciben fechas.
- `d3.scaleOrdinal()`: Escalas de dominio y rango discreto.
- `d3.bandScale()`: Escalas de dominio y rango discreto, pero donde el rango tiene valores numéricos y continuos. Pensado para generar barras en un espacio continuo.
- `d3.scalePoint()`: Escalas de dominio y rango discreto, pero donde el rango tiene valores numéricos y continuos. Pensado para generar puntos en un espacio continuo.
- ...

¡Revisa la documentación!

`d3-axis`

- `d3.axisTop()`: Eje horizontal con leyenda sobre eje.
- `d3.axisBottom()`: Eje horizontal con leyenda bajo eje.
- `d3.axisRight()`: Eje vertical con leyenda a la izquierda de eje.
- `d3.axisLeft()`: Eje vertical con leyenda a la derecha de eje.
- ...

¡Revisa la documentación!

`d3-time` y `d3-time-format`

- `d3.timeDay.count(inicio, fin)`: Cuenta cuantos días han pasado entre dos fechas.
- `d3.timeFormat(formato)`: Para transformar fechas en strings en cierto formato.
- `d3.timeParse(formato)`: Para obtener fechas desde strings en cierto formato.
- ...

¡Revisa la documentación aquí y acá!

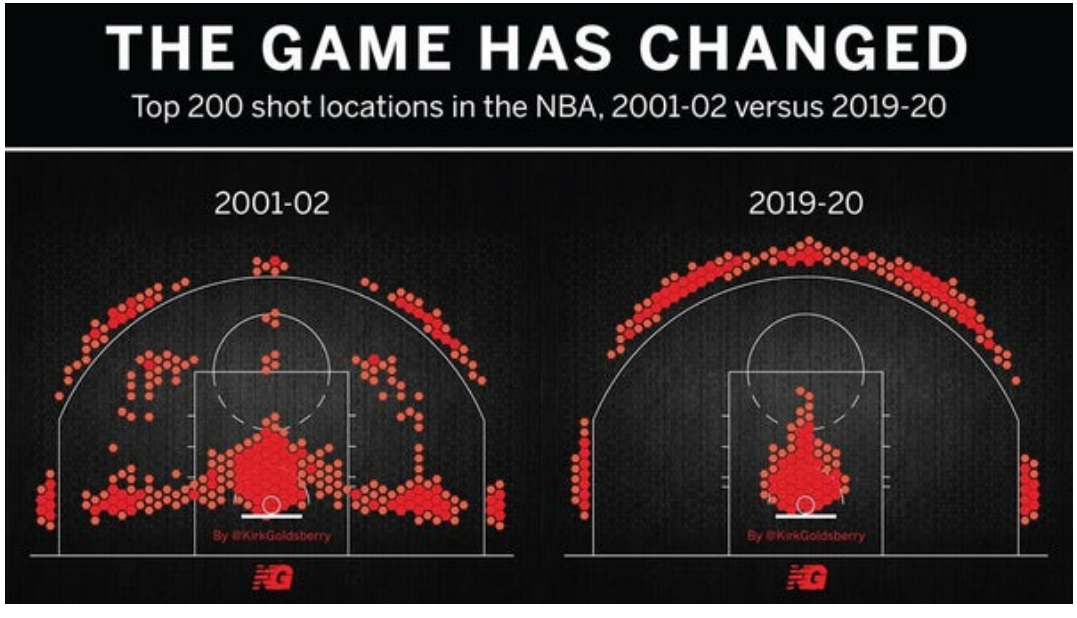
`d3-array`

- `d3.extent(arreglo)`: Para obtener valor mínimo y máximo de arreglo datos.
- `d3.median(arreglo)`: Para mediana de arreglo de datos.
- `d3.variance(arreglo)`: Para varianza de arreglo de datos.
- `d3.group(arreglo)`: Agrupa datos de un arreglo según sus valores.
- ...

¡Revisa la documentación!

¡Visualización del día!

¡Visualización del día!



Visualización de frecuencia de lanzamientos de basketball en la NBA.
Propuesta por estudiante Juan Dlugoszewski.

(Fuente: Shot chart dramatically shows the change in the NBA game)

¡Visualización del día!

d3.basketball-shot-chart

This visualization aims to become a generic means of generating charts on a basketball court. Currently it only supports hexbin shot charts, with lots of flexibility, but is alpha quality and will be refactored to support other binning mechanisms and other mark types on top of a basketball court.

Currently customizable:

- Court dimensions/lines
- Binning definition
- Hexagon size range and color range
- Integrating different shot chart datasets
- Titles and labels

¡Hay personas que han publicado código para generar este tipo de visualizaciones! Aquí algunas en uso

(Fuente: d3.basketball-shot-chart)

¿Más dudas?

Próximos eventos:

Viernes 11 de septiembre (20:00) termina plazo de Hito 1.

Próximos eventos:

Viernes 11 de septiembre (20:00) termina plazo de Hito 1.

Sesión de martes 15 de septiembre revisamos Utilidades en D3.js II.

Próximos eventos:

Viernes 11 de septiembre (20:00) termina plazo de Hito 1.

Sesión de martes 15 de septiembre revisamos Utilidades en D3.js II.

¡Sesión de jueves 17 de septiembre es libre! Me conectaré de todas formas por si tienen preguntas :)

Utilidades en D3.js I

Visualización de Información

IIC2026 2020-2

¡Deja tus preguntas en los comentarios!