

¡Hola 🙌! Espera mientras comienza la sesión.

Antes que todo, ¿cómo están?

Visualización de Información

IIC2026 2021-2

Selecciones y *join* de datos en D3.js

Visualización de Información

IIC2026 2021-2

Contenido

Contenido

1. **Selecciones en D3.js I**
2. **Selecciones en D3.js II**
3. ***Join* de datos en D3.js I**
4. ***Join* de datos en D3.js II**

D3.js

- Su intención es utilizar HTML, CSS y SVG para crear visualizaciones.
- Apareció en un momento en que esto no era común para herramientas de visualización.
- Escrita por Mike Bostock.



D3.js

NO es una librería de visualización de alto nivel.

```
const grafico_de_barra = d3.crear(); // ✘  
grafico_de_barra.graficar(); // ✘
```


D3.js

Utilizaremos la versión 6 o 7 en el curso.

👁️ con encontrar recursos y ejemplos escritos en la versiones anteriores.

Podrán encontrar muchos ejemplos en la web.

Le dedicaremos tiempo a aprender D3.js en el curso. ¡Aprovechenlo!

Utilizar D3

```
1 <!DOCTYPE html>
2 <html>
3
4   <head>
5     <meta charset="utf-8">
6     <title>Ejemplo con D3</title>
7     <script src="https://d3js.org/d3.v7.min.js"></script>
8   </head>
9
10  <body>
11
12    <script src='programa.js' charset='utf-8'></script>
13  </body>
14 </html>
```

Data Driven Documents

Librería escrita en JavaScript para manipular documentos basándose en datos.

Selecciones

Objeto de D3.js que se comporta como una colección de elementos HTML.

```
d3.select()
```

```
d3.selectAll()
```

```
seleccion.select()
```

```
seleccion.selectAll()
```

Selecciones

```
d3.selectAll("rect")  
  .attr("y", 50)  
  .style("fill", "red")  
  .attr("x", (d, i, all) => 100 * i);
```

```
1 <svg>  
2   <rect y="50" style="fill: red" x="0"></rect>  
3   <rect y="50" style="fill: red" x="100"></rect>  
4   <rect y="50" style="fill: red" x="200"></rect>  
5 </svg>
```

Antes:

```
1 <body>
2   <ul></ul>
3   <ul></ul>
4   <ul></ul>
5 </body>
```

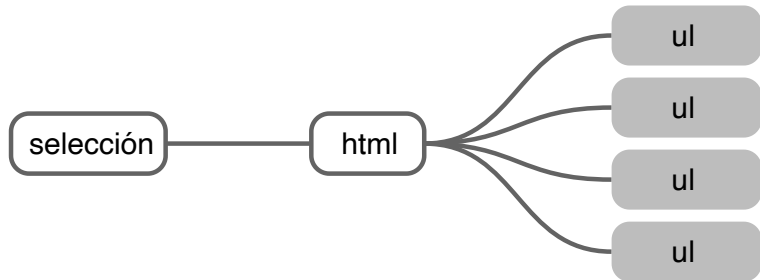
```
d3.selectAll("ul")
  .append("li");
```

Después:

```
1 <body>
2   <ul>
3     <li></li>
4   </ul>
5   <ul>
6     <li></li>
7   </ul>
8   <ul>
9     <li></li>
10  </ul>
11 </body>
```

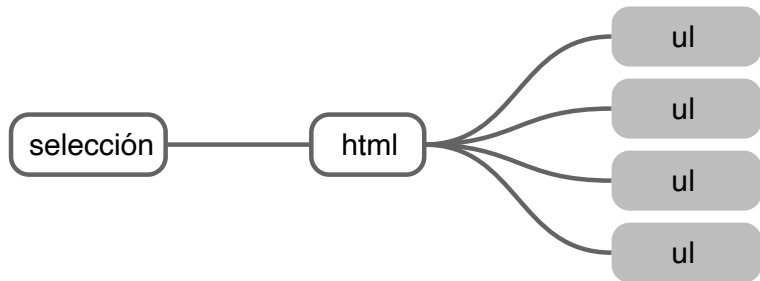
Múltiples grupos

```
d3.selectAll("ul");
```



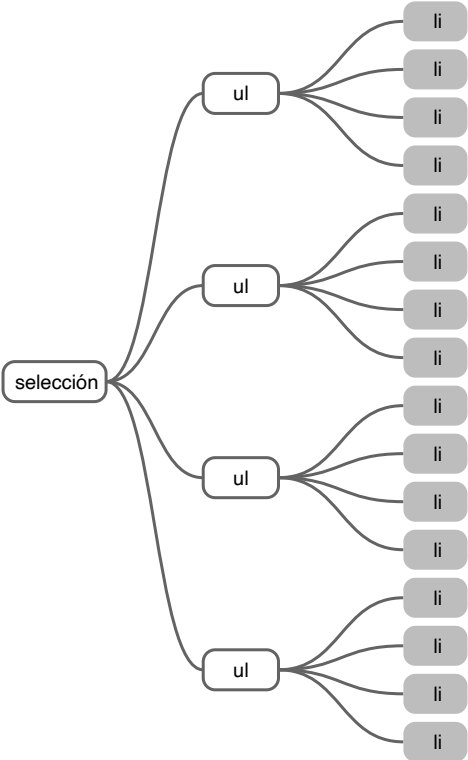
Múltiples grupos

```
d3.selectAll("ul")  
  .selectAll("li");
```



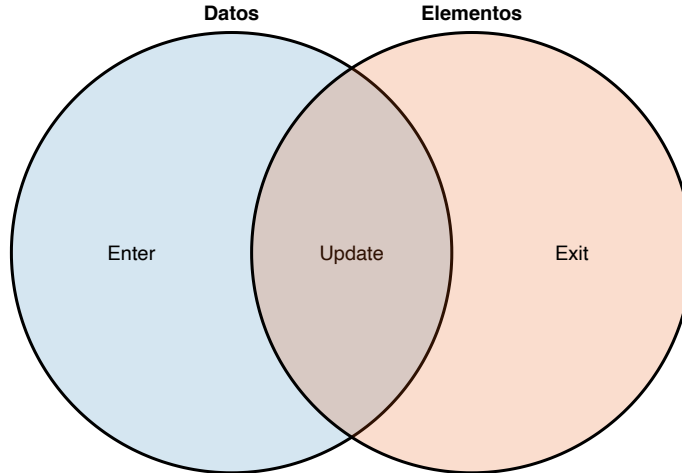
Múltiples grupos

```
d3.selectAll("ul")  
  .selectAll("li");
```



seleccion.data

- Hay datos que no se le asocian elementos ➡ *enter*
- Hay elementos y datos que se asocian entre ellos ➡ *update*
- Hay elementos que no se le asocian datos ➡ *exit*



Update

```
1 const datos = [23, 45, 120, 64];
2
3 d3.select("#svg")
4   .selectAll("rect")
5   .data(datos);
```

```
<svg id="svg" width="400" height="250">
  <rect></rect> <!-- 23 -->
  <rect></rect> <!-- 45 -->
  <rect></rect> <!-- 120 -->
  <rect></rect> <!-- 64 -->
</svg>
```

Update

```
1 const datos = [23, 45, 120, 64];  
2  
3 const update = d3.select("#svg")  
4   .selectAll("rect")  
5   .data(datos);
```

```
<svg id="svg" width="400" height="250">  
  <rect></rect> <!-- 23 -->  
  <rect></rect> <!-- 45 -->  
  <rect></rect> <!-- 120 -->  
  <rect></rect> <!-- 64 -->  
</svg>
```

Update

```
1 const datos = [23, 45, 120, 64];
2
3 const update = d3.select("#svg")
4   .selectAll("rect")
5   .data(datos);
6
7 update.attr("width", 50)
8   .attr("y", 0)
9   .attr("x", (d, i, all) => i * 100);
```

```
<svg id="svg" width="400" height="250">
  <rect></rect> <!-- 23 -->
  <rect></rect> <!-- 45 -->
  <rect></rect> <!-- 120 -->
  <rect></rect> <!-- 64 -->
</svg>
```

Update

```
1 const datos = [23, 45, 120, 64];
2
3 const update = d3.select("#svg")
4   .selectAll("rect")
5   .data(datos);
6
7 update.attr("width", 50)
8   .attr("y", 0)
9   .attr("x", (d, i, all) => i * 100);
```

```
<svg id="svg" width="400" height="250">
  <rect width="50" y="0" x="0"></rect> <!-- 23 -->
  <rect width="50" y="0" x="100"></rect> <!-- 45 -->
  <rect width="50" y="0" x="200"></rect> <!-- 120 -->
  <rect width="50" y="0" x="300"></rect> <!-- 64 -->
</svg>
```

Update

```
1 const datos = [23, 45, 120, 64];
2
3 const update = d3.select("#svg")
4   .selectAll("rect")
5   .data(datos);
6
7 update.attr("width", 50)
8   .attr("y", 0)
9   .attr("x", (d, i, all) => i * 100)
10  .attr("height", (d, i, all) => 2 * d);
```

```
<svg id="svg" width="400" height="250">
  <rect width="50" y="0" x="0"></rect> <!-- 23 -->
  <rect width="50" y="0" x="100"></rect> <!-- 45 -->
  <rect width="50" y="0" x="200"></rect> <!-- 120 -->
  <rect width="50" y="0" x="300"></rect> <!-- 64 -->
</svg>
```

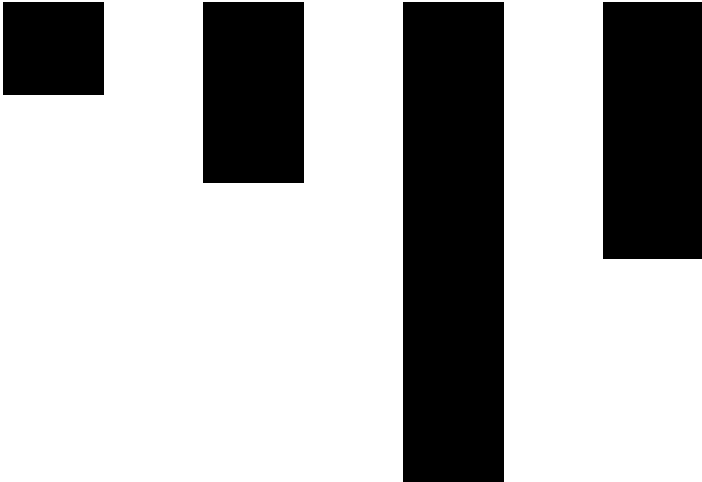
Update

```
1 const datos = [23, 45, 120, 64];
2
3 const update = d3.select("#svg")
4   .selectAll("rect")
5   .data(datos);
6
7 update.attr("width", 50)
8   .attr("y", 0)
9   .attr("x", (d, i, all) => i * 100)
10  .attr("height", (d, i, all) => 2 * d);
```

```
<svg id="svg" width="400" height="250">
  <rect width="50" y="0" x="0" height="46"></rect> <!-- 23 -->
  <rect width="50" y="0" x="100" height="90"></rect> <!-- 45 -->
  <rect width="50" y="0" x="200" height="240"></rect> <!-- 120 -->
  <rect width="50" y="0" x="300" height="128"></rect> <!-- 64 -->
</svg>
```


Update

```
1 const datos = [23, 45, 120, 64];  
2  
3 const update = d3.select("#svg")  
4   .selectAll("rect")  
5   .data(datos);  
6  
7 update.attr("width", 50)  
8   .attr("y", 0)  
9   .attr("x", (d, i, all) => i * 100)  
10  .attr("height", (d, i, all) => 2 * d);
```



Duda publicada

- Respecto al atributo "d" en (d, i, all); el parametro lo reconoce como dato por ser "d" o por la posicion dentro de la funcion?

Exit

```
<svg id="svg" width="400" height="250">
  <rect></rect>
  <rect></rect>
  <rect></rect>
  <rect></rect>
</svg>
```

```
1 const datos = [23, 45];
2
3 d3.select("#svg")
4   .selectAll("rect")
5   .data(datos);
```

```
<svg id="svg" width="400" height="250">
  <rect></rect> <!-- 23 -->
  <rect></rect> <!-- 45 -->
  <rect></rect> <!-- ? -->
  <rect></rect> <!-- ? -->
</svg>
```

Exit

```
<svg id="svg" width="400" height="250">
  <rect></rect>
  <rect></rect>
  <rect></rect>
  <rect></rect>
</svg>
```

```
1 const datos = [23, 45];
2
3 const update = d3.select("#svg")
4   .selectAll("rect")
5   .data(datos);
6
7 update.exit().remove();
```

```
<svg id="svg" width="400" height="250">
  <rect></rect> <!-- 23 -->
  <rect></rect> <!-- 45 -->
  <rect></rect> <!-- ? -->
  <rect></rect> <!-- ? -->
</svg>
```

Exit

```
<svg id="svg" width="400" height="250">
  <rect></rect>
  <rect></rect>
  <rect></rect>
  <rect></rect>
</svg>
```

```
1 const datos = [23, 45];
2
3 const update = d3.select("#svg")
4   .selectAll("rect")
5   .data(datos);
6
7 update.exit().remove();
```

```
<svg id="svg" width="400" height="250">
  <rect></rect> <!-- 23 -->
  <rect></rect> <!-- 45 -->
</svg>
```

Enter

```
<svg id="svg" width="400" height="250">  
</svg>
```

```
1 const datos = [23, 45, 120, 64];  
2  
3 d3.select("#svg")  
4   .selectAll("rect")  
5   .data(datos);
```

```
<svg id="svg" width="400" height="250">  
  <!-- ? -->  
</svg>
```

Enter

```
<svg id="svg" width="400" height="250">  
</svg>
```

```
1 const datos = [23, 45, 120, 64];  
2  
3 d3.select("#svg")  
4   .selectAll("rect")  
5   .data(datos)  
6   .enter();
```

```
<svg id="svg" width="400" height="250">  
  <!-- ? -->  
</svg>
```

Enter

```
<svg id="svg" width="400" height="250">
</svg>
```

```
1 const datos = [23, 45, 120, 64];
2
3 d3.select("#svg")
4   .selectAll("rect")
5   .data(datos)
6   .enter()
7   .append("rect");
```

```
<svg id="svg" width="400" height="250">
  <!-- ? -->
</svg>
```


Enter

```
<svg id="svg" width="400" height="250">  
</svg>
```

```
1 const datos = [23, 45, 120, 64];  
2  
3 d3.select("#svg")  
4   .selectAll("rect")  
5   .data(datos)  
6   .enter()  
7   .append("rect");
```

```
<svg id="svg" width="400" height="250">  
  <rect></rect> <!-- 23 -->  
  <rect></rect> <!-- 45 -->  
  <rect></rect> <!-- 120 -->  
  <rect></rect> <!-- 64 -->  
</svg>
```

Enter

```
1  const datos = [23, 45, 120, 64];
2
3  d3.select("#svg")
4    .selectAll("rect")
5    .data(datos)
6    .enter()
7    .append("rect")
8    .attr("width", 50)
9    .attr("y", 0)
10   .attr("x", (d, i, all) => i * 100)
11   .attr("height", (d, i, all) => 2 * d);
```

```
<svg id="svg" width="400" height="250">
  <rect></rect> <!-- 23 -->
  <rect></rect> <!-- 45 -->
  <rect></rect> <!-- 120 -->
  <rect></rect> <!-- 64 -->
</svg>
```

Enter

```
1 const datos = [23, 45, 120, 64];
2
3 d3.select("#svg")
4   .selectAll("rect")
5   .data(datos)
6   .enter()
7   .append("rect")
8   .attr("width", 50)
9   .attr("y", 0)
10  .attr("x", (d, i, all) => i * 100)
11  .attr("height", (d, i, all) => 2 * d);
```

```
<svg id="svg" width="400" height="250">
  <rect width="50" y="0" x="0" height="46"></rect> <!-- 23 -->
  <rect width="50" y="0" x="100" height="90"></rect> <!-- 45 -->
  <rect width="50" y="0" x="200" height="240"></rect> <!-- 120 -->
  <rect width="50" y="0" x="300" height="128"></rect> <!-- 64 -->
</svg>
```

Algunas aclaraciones

Algunas aclaraciones

- La vinculación entre elementos y datos mediante `data` queda en los elementos, no en la selección misma.
- La vinculación de un elemento a un dato potencialmente se sobre escribe con más llamadas a `data`.
- La vinculación entre arreglo de datos se hace por grupo, en vez de a nivel de selección.

¡Visualización del día!



Propuesto por estudiante Claudia González.

(Fuente: [Wealth shown to scale](#))

Próximos eventos:

Próximo martes revisaremos el material de **Utilidades de D3 I**.

Próximo jueves revisaremos el material de **Utilidades de D3 II**.

Domingo 26 de septiembre (20:00:00) termina plazo de **Hito 1**.

Selecciones y *join* de datos en D3.js

Visualización de Información

IIC2026 2021-2

¡Deja tus preguntas en los comentarios!