

¡Hola 🙌! Espera mientras comienza la sesión.

**Antes que todo, ¿cómo están?**

# Visualización de Información

**IIC2026 2021-2**

# Utilidades en D3.js II

**Visualización de Información**

**IIC2026 2021-2**

# Contenidos

1. Eventos en D3.js
2. Transiciones de D3.js
3. *Join* de datos personalizado

## ¡Más detalles para nuestro programa!

Comenzamos con una visualización estática.

Terminamos con una visualización dinámica e interactiva.

# Eventos

```
1  const width = 600;
2  const height = 400;
3  const margin = {
4    top: 30,
5    bottom: 30,
6    right: 30,
7    left: 30,
8  };
9
10 const svg = d3
11   .select("body")
12   .append("svg")
13   .attr("width", width)
14   .attr("height", height);
15
16 const boton = d3.select("body").append("button").text("Agregar elem
```

# Eventos

```
120     categoria: String.fromCharCode(
121         datos[datos.length - 1].categoria.charCodeAt(0) + 1
122     ),
123     frecuencia: Math.floor(Math.random() * 800),
124 });
125
126 let datos;
127
128 d3.json("datos.json")
129     .then((datosCargados) => {
130         console.log(datosCargados);
131         datos = datosCargados;
132         joinDeDatos(datos);
133         boton.on("click", () => {
134             datos.push(datoNuevoRandom(datos));
135         });
136     });
```



# Eventos

```
102     .duration(500)
103     .attr("y", height - margin.top - margin.bottom)
104     .attr("height", 0)
105     .remove()
106 )
107 .on("mouseenter", (_, d) => {
108     parrafo.text(`Categoría: ${d.categoria}, Frecuencia: ${d.frecu
109 })
110 .on("mouseleave", () => {
111     parrafo.text("");
112 })
113 .on("click", (_, d) => {
114     datos.splice(datos.indexOf(d), 1);
115     joinDeDatos(datos);
116 });
```

# Eventos

¡Hay muchos tipos de eventos!

- `click`
- `dblclick`
- `change`
- `dragstart`
- `dragover`
- ...

Lista de eventos: [MDN](#)

# Transiciones

```
1  const width = 600;
2  const height = 400;
3  const margin = {
4    top: 30,
5    bottom: 30,
6    right: 30,
7    left: 30,
8  };
9
10 const svg = d3
11   .select("body")
12   .append("svg")
13   .attr("width", width)
14   .attr("height", height);
15
16 const boton = d3.select("body").append("button").text("Agregar elem
```

# Transiciones

```
75     .data(datos, (d) => d.categoria)
76     .join(
77         (enter) =>
78             enter
79                 .append("rect")
80                 .attr("fill", "magenta")
81                 .attr("y", height - margin.top - margin.bottom)
82                 .attr("x", (d) => escalaX(d.categoria))
83                 .attr("width", escalaX.bandwidth())
84                 .attr("height", 0)
85                 .transition()
86                 .duration(1000)
87                 .attr("height", (d) => escalaAltura(d.frecuencia))
88                 .attr("y", (d) => escalaY(d.frecuencia))
89     )
```

# Transiciones

```
84     .attr("height", 0)
85     .transition()
86     .duration(1000)
87     .attr("height", (d) => escalaAltura(d.frecuencia))
88     .attr("y", (d) => escalaY(d.frecuencia))
89     .selection(),
90 (update) =>
91     update
92     .transition()
93     .duration(1000)
94     .attr("height", (d) => escalaAltura(d.frecuencia))
95     .attr("y", (d) => escalaY(d.frecuencia))
96     .attr("x", (d) => escalaX(d.categoria))
97     .attr("width", escalaX.bandwidth())
98     .selection(),
99 (exit) =>
100     exit
```

# Transiciones

```
93     .duration(1000)
94     .attr("height", (d) => escalaAltura(d.frecuencia))
95     .attr("y", (d) => escalaY(d.frecuencia))
96     .attr("x", (d) => escalaX(d.categoria))
97     .attr("width", escalaX.bandwidth())
98     .selection(),
99     (exit) =>
100     exit
101     .transition()

102     .duration(500)
103     .attr("y", height - margin.top - margin.bottom)
104     .attr("height", 0)
105     .remove()
106 )
107 .on("mouseenter", (_, d) => {
```

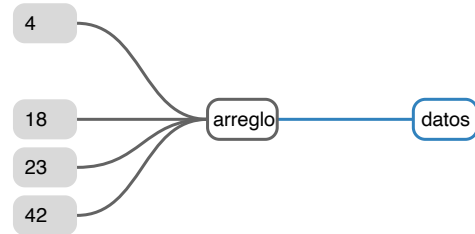
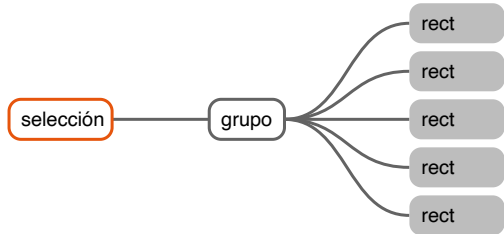
# Transiciones

Detalles de transiciones en submódulo `d3-transition`.

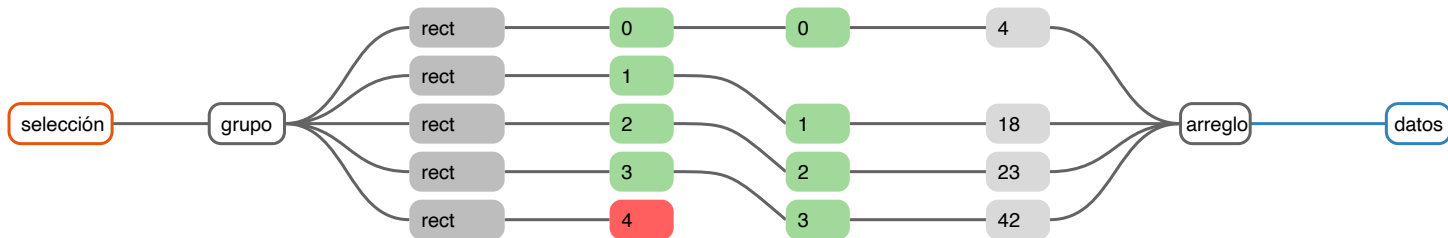
- Tienen un método `delay` para fijar un atraso.
- Tienen un método `ease` para alterar como es el cambio en el tiempo.

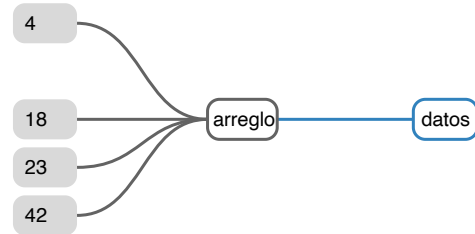
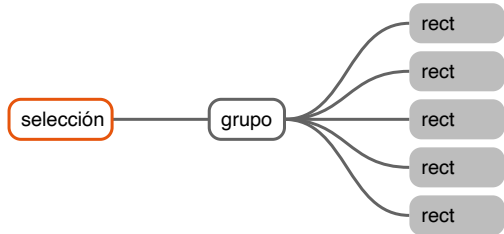
*Join* de datos personalizado



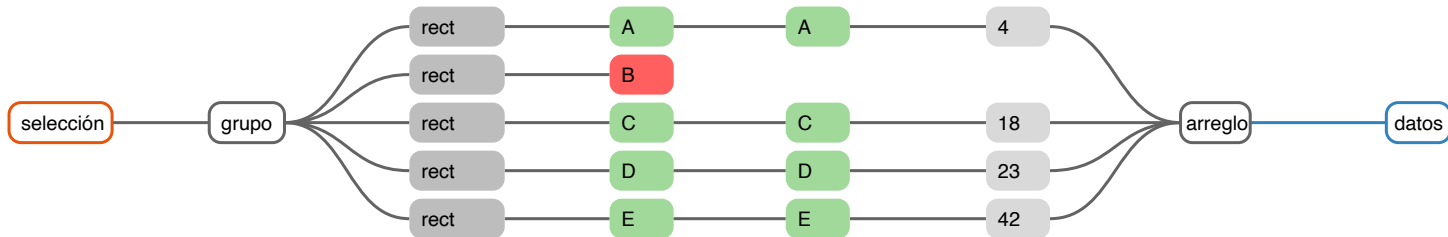


```
1 selection.data(data);
```





```
1 selection.data(data, (d) => d.llave);
```



**¡Practico entre todes!**

# ¡Visualización del día!



Página interactiva de información sobre armas nucleares.

Propuesta por estudiante "Java B".

(Fuente: [Outrider - Nuclear Weapons Interactive](#))

## Próximos eventos:

Próxima semana revisaremos el material de **Visualización de datos tabulares** y *Layouts* tabulares en D3.js.  
Domingo 26 de septiembre (20:00:00) termina plazo de **Hito 1**.

# Utilidades en D3.js II

Visualización de Información

IIC2026 2021-2

¡Deja tus preguntas en los comentarios!