

¡Hola 🙌! Espera mientras comienza la sesión.

Antes que todo, ¿cómo están?

Visualización de Información

IIC2026 2020-2

Utilidades en D3.js II

Visualización de Información

IIC2026 2020-2

Repaso

Repaso

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](#)

¡Alerta de versión!

V5:

```
1 seleccion.on("click", (d, i, elements) => {
2   console.log(d3.event());
3   console.log(d);
4   console.log(elements[i]);
5 })
```

V6:

```
1 seleccion.on("click", (event, d) => {
2   console.log(event);
3   console.log(d);
4   console.log(event.currentTarget);
5 })
```

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     )  
90     .selection()
```

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()

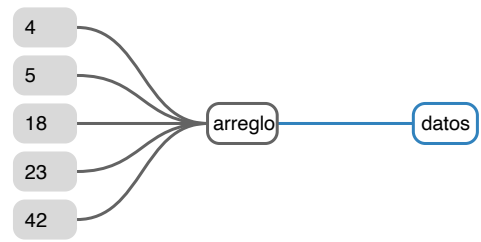
102     .duration(500)
103     .attr("y", height - margin.top - margin.bottom)
104     .attr("height", 0)
105     .remove()
106 )
107 .on("mouseenter", (_, d) => {
```

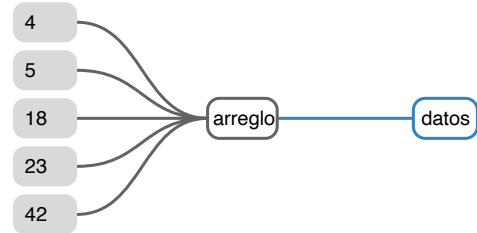

Transiciones

Detalles de transiciones en submódulo `d3-transition`.

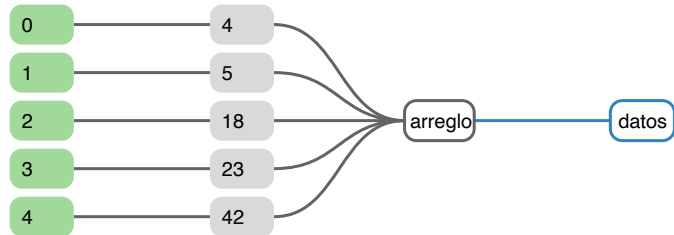
- Tienen un método `delay` para fijar un atraso.
- Tienen un método `ease` para alterar como es el cambio en el tiempo.

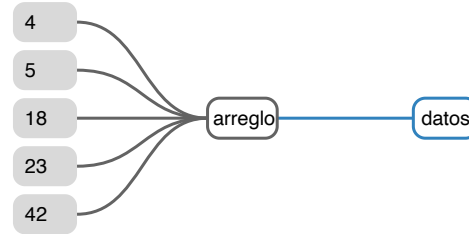
Join de datos personalizado





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

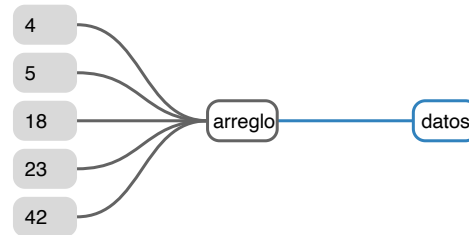
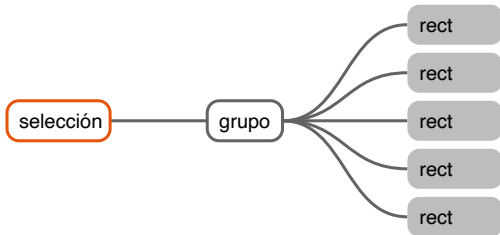


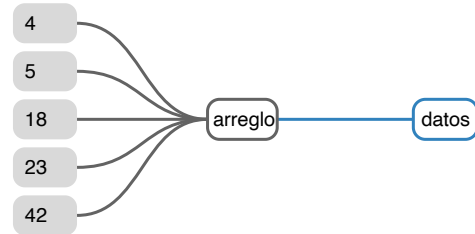
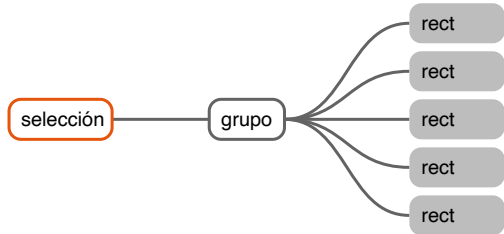


```
1 selection.data(data)
```

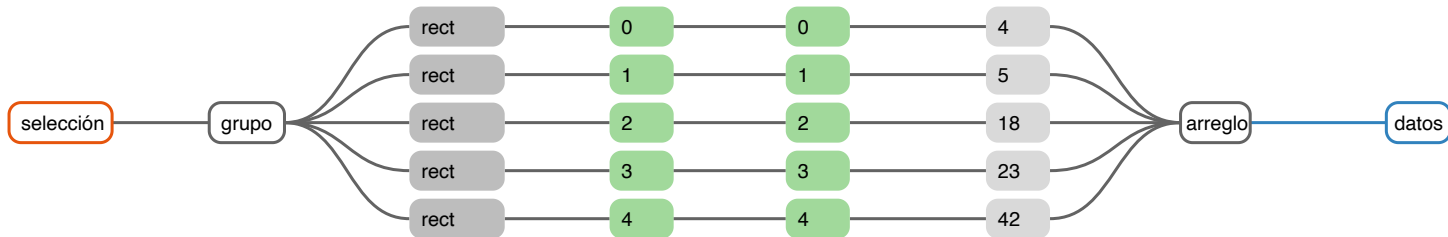


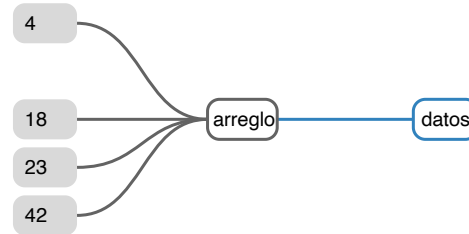
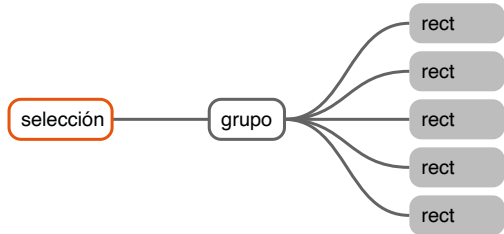
```
1 selection.data(data).enter().append("rect")
```



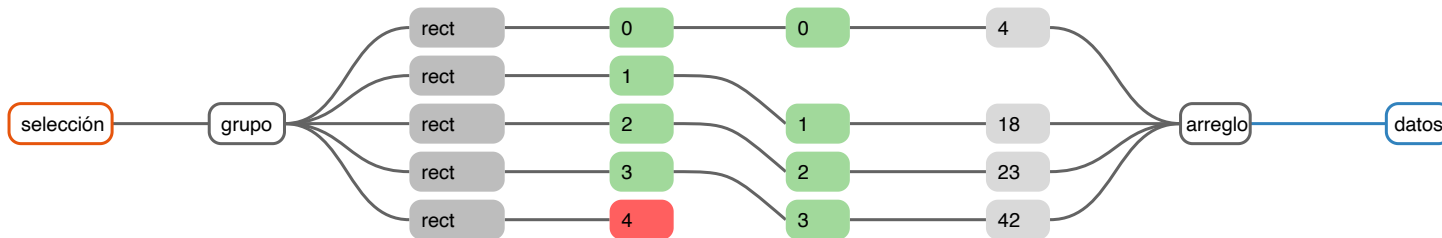


```
1 selection.data(data);
```





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

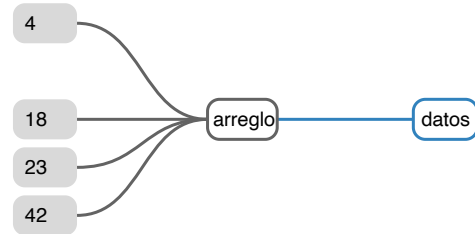
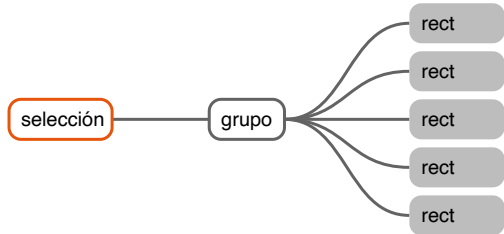


Función de llave en *data join*

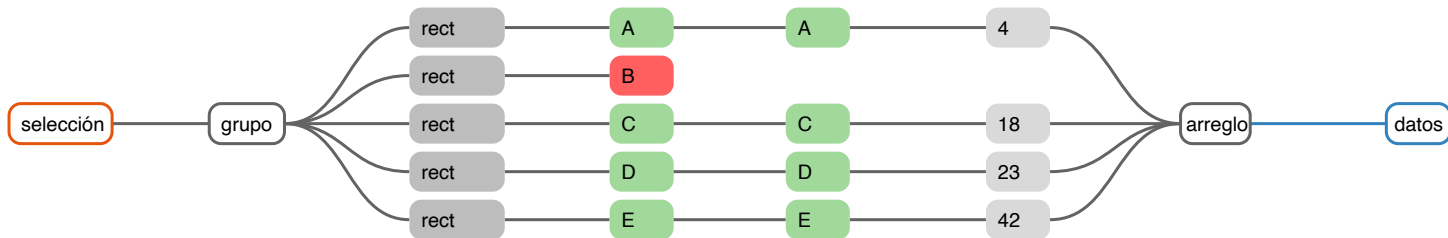
```
1  const width = 600;
2  const height = 400;
3  const margin = {
4    top: 30,
5    bottom: 30,
6    right: 30,
7    left: 30,
8  };
9
10 const svg = d3
11   .select("body")
12   .append("svg")
13   .attr("width", width)
14   .attr("height", height);
15
16 const boton = d3.select("body").append("button").text("Agregar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


Función de llave en *data join*

```
66     .transition()
67     .duration(1000)
68     .call(ejex)
69     .selection()
70     .selectAll("text")
71     .attr("font-size", 20);
72
73     contenedorBarras
74     .selectAll("rect")
75     .data(datos, (d) => d.categoria)
76     .join(
77     (enter) =>
78     enter
79     .append("rect")
80     .attr("fill", "magenta")
81     .attr("x", height - margin.top - margin.bottom)
```





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





¡Visualización del día!

¡Visualización del día!

On average, each ...

Infects 1  per 4 days
(at the start of the epidemic)



Takes 10 days to go from  to 

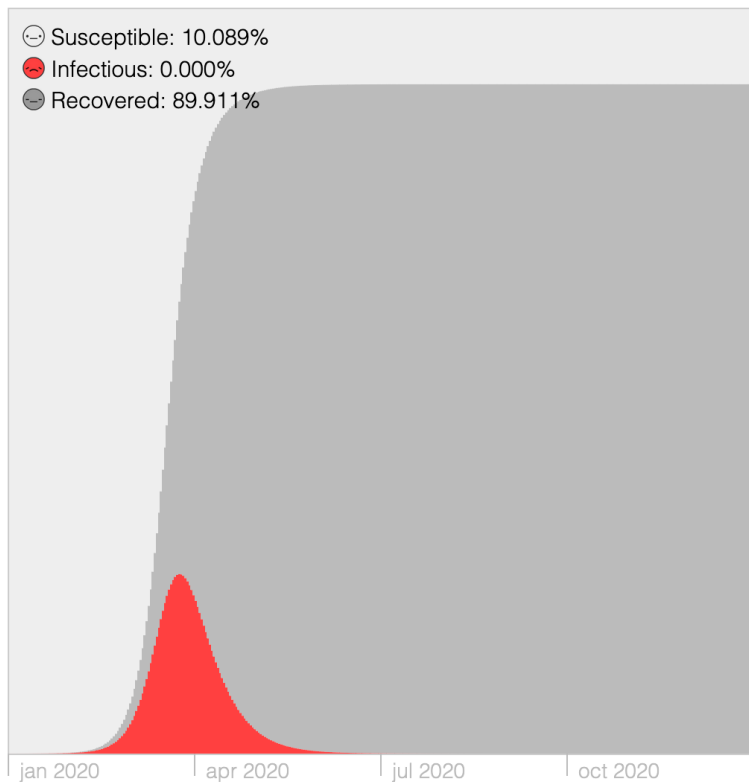


Simulate 1.0 years in 10 seconds



 Reset

Reset All



Artículo de explicación de pandemia COVID-19 mediante simulaciones.

Propuesta por ayudante Francisca Ibarra.

(Fuente: [What happens next?](#))

¿Más dudas?

Próximos eventos:

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

Próximos eventos:

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

Se publicaron los contenidos de la semana siguiente a la semana de receso. Los ejercicios propuestos se actualizarán a medida que estén disponibles.

Próximos eventos:

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

Se publicaron los contenidos de la semana siguiente a la semana de receso. Los ejercicios propuestos se actualizarán a medida que estén disponibles.

¡Disfruten la semana de receso!

Utilidades en D3.js II

Visualización de Información

IIC2026 2020-2

¡Deja tus preguntas en los comentarios!