IIC2026
Ayudantía 2 - D3 y JS

Camila Basulto Correa
camila.basulto@uc.cl

Temas de la ayudantía

1. Repaso de librería d3
    a. Selecciones
    b. Data Join
    c. Cargar/leer datos

2. Gráfico estático
    a. Cargar datos JSON
    b. .data(...).join(...)
    c. Uso de ejes de d3
    d. .enter()

D3.js

D3.js

- **NO** es una librería de visualización de alto nivel
- Es una herramienta para **crear visualizaciones desde 0**.
  - Definir cada elemento del SVG.
  - Personalizar los elementos del SVG según los datos.
  - Definir las interacciones posibles.
  - Agregar leyendas a mano.
  - entre otros...

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ejemplo con D3</title>
    <script
src="https://d3js.org/d3.v7.min.js"></s
cript>
  </head>
  <body>
    <script src='programa_4.js'
charset='utf-8'></script>
  </body>
</html>
```

Selecciones en D3

.d3.select() | d3.selectAll()

- `d3.select("p")` → Por tag del elemento HTML.
- `d3.select(".class")` → Por la clase del elemento HTML.
- `d3.select("#id")` → Por el ID del elemento HTML.

seleccion.attr(atributo, valor) | seleccion.style(propiedad, valor)

seleccion.append(tag)

```
d3.selectAll("rect").attr("x", 10).attr("y", 50).style("fill", "orange");
```

*podemos encadenar funciones

```
seleccion.attr("x", (_, i, _) => i*100);
```

*primer argumento:dato vinculado a la selección; segundo argumento:iterador; tercer argumento:todos los elementos visuales utilizados en la selección.

Data Join

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

```
const datos = [23, 45, 99, 64]
d3.select("#vis").selectAll("rect").data(datos);
```

```
<svg id="vis" width="400" height="250">
 ?? <!-- 23 -->
 ?? <!-- 45 -->
 ?? <!-- 99 -->
 ?? <!-- 64 -->
</svg>
```

*Visualmente no pasa nada. Por dentro, d3 entiende que hay 4 datos sin vincularse a nada.

```
d3.select("#vis").selectAll("rect").data(datos).join("rect");
```

```
<svg id="vis" width="400" height="250">
 <rect></rect> <!-- 23 -->
 <rect></rect> <!-- 45 -->
 <rect></rect> <!-- 99 -->
 <rect></rect> <!-- 64 -->
</svg>
```

1. Si hay datos pero no elementos, **crea un nuevo elemento para cada dato.**
2. Si ya existe algún elemento para vincular al dato, **no** creará un nuevo elemento, **asignará la data al elemento existente.**
3. Si hay algún elemento que **no quedó vinculado** a algún dato, **lo elimina (puede ser que hay más elementos que datos)**.

Cargar datos de un archivo o enlace - CSV y JSON

d3.csv(url_o_path, funcion_de_parseo)

```
index,rank,repo_name,stars
0,1,PayloadsAllTheThings,49975
1,2,face_recognition,49072
2,3,localstack,48113
3,4,gpt4free,44160
4,5,cheat.sh,35820
5,6,ChatGLM-6B,33475
```

d3.json(url_o_path)

```
[
    {
        "index": "0",
        "rank": "1",
        "repo_name": "PayloadsAllTheThings",
        "stars": "49975"
    },
    {...}
]
```

2 o más datasets

```
d3.csv("datos.csv").then(datosCSV => {
    d3.json("datos2.json").then(datosJSON => {
        funcion(datosCSV, datosJSON)
    })
})
```

```
Promise.all([
    d3.csv("datos.csv"),
    d3.json("datos2.json")
]).then(function(data) {
    console.log(data[0])  // es lo mismo que datosCSV
    console.log(data[1])  // es lo mismo que datosJSON
});
```

Práctica

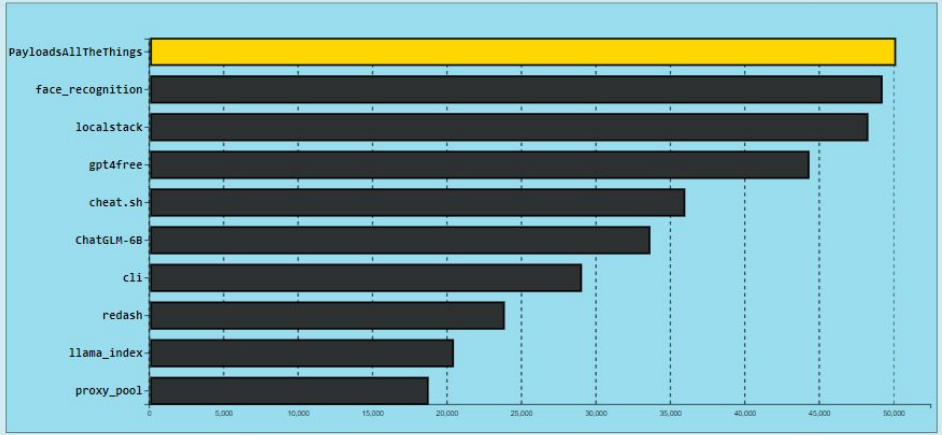
Gráfico estático

Ahora, vamos a poner en práctica las herramientas aprendidas para hacer un gráfico estático simple.

- Datos: Most Popular Python Projects on GitHub

Gráfico estático con D3

Vamos a utilizar datos de las 100 proyectos más populares de python en GitHub "Most Popular Python Projects on GitHub (2018-2023)". Estos datos fueron preprocesados para obtener los 100 proyectos más populares de python en GitHub a la última fecha registrada en el dataset (21-08-2023), para anlizar el **Top 10 de librerias/repositorios**.



IIC2026
Ayudantía 2 - D3 y JS

Camila Basulto Correa
camila.basulto@uc.cl