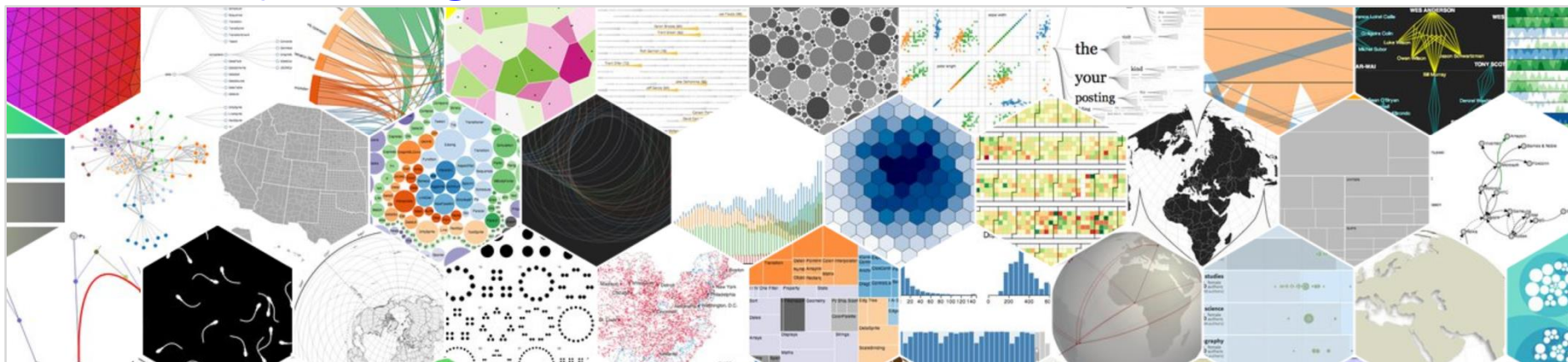


# Taller de Programación

## Clase 17

**Cristian Vallejos** ([cristian.vallejos@uach.cl](mailto:cristian.vallejos@uach.cl))

**Joel Torres** ([joel.torres@uach.cl](mailto:joel.torres@uach.cl))



- Introducción a la Visualización de Datos
- Introducción a D3.js
- Creación de Gráficos Básicos
- Actividad 1

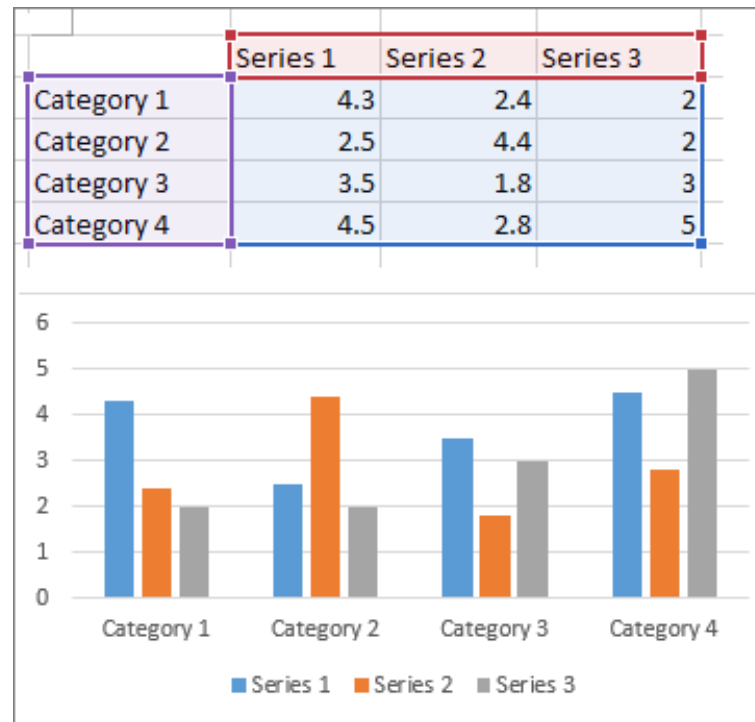
# Introducción a la Visualización de Datos



# Introducción a la Visualización de Datos

¿Qué es la visualización de datos?

- La visualización de datos es el proceso de representar información y datos de forma gráfica y visual, utilizando herramientas como gráficos, mapas, infografías y paneles interactivos.
- Su objetivo principal es hacer que los conjuntos de datos complejos sean más fáciles de entender, identificar patrones, tendencias, anomalías y comunicar los hallazgos de manera más accesible.



# Introducción a la Visualización de Datos

## Beneficios para la toma de decisiones

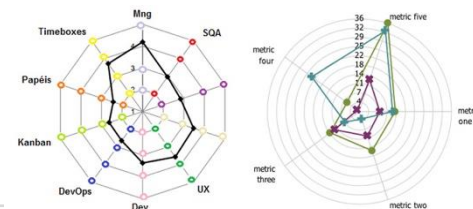
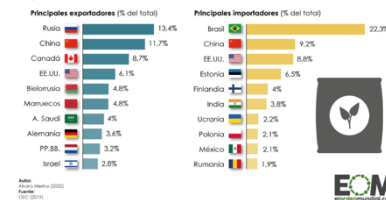
- **Acelera la comprensión:** El cerebro humano procesa la información visual de manera más rápida y eficiente que el texto o los números sin procesar, lo que permite una identificación de ideas clave más veloz.
- **Mejora la precisión:** Al basar las decisiones en información precisa y procesable derivada del análisis visual de datos, las empresas pueden mejorar su precisión y reducir el riesgo de errores asociados con la intuición sola.
- **Fomenta la colaboración:** Los elementos visuales proporcionan un lenguaje común para que los equipos discutan y se alineen en torno a ideas compartidas, facilitando la toma de decisiones colaborativa en diversos departamentos.
- **Optimiza el rendimiento:** Permite a las organizaciones generar información y predicciones en tiempo real, lo que lleva a la optimización del rendimiento y la capacidad de probar nuevas estrategias de manera eficiente.
- **Identifica nuevas oportunidades:** Ayuda a descubrir nuevas oportunidades de mercado, a comprender mejor a los clientes y a comparar el rendimiento de la empresa con el de sus competidores.

# Introducción a la Visualización de Datos

## Tipos de Gráficos:

- Gráficos Categóricos: Muestran y comparan la frecuencia o proporciones de diferentes categorías.
  - Ejemplo: gráficos de barras/columnas, gráfico de torta/anillo, diagrama de puntos, etc.
- Gráficos Temporales: Ilustran cambios y tendencias a lo largo de un periodo de tiempo.
  - Ejemplo: Gráficos de líneas, gráficos de áreas, gráficos de series temporales, etc.
- Gráficos Comparativos: Comparar directamente a dos conjuntos de datos o para analizar como una métrica varía entre diferentes grupos o categorías.
  - Ejemplo: Gráficos de barras múltiples, gráficos de dispersión, gráficos de radar, etc.

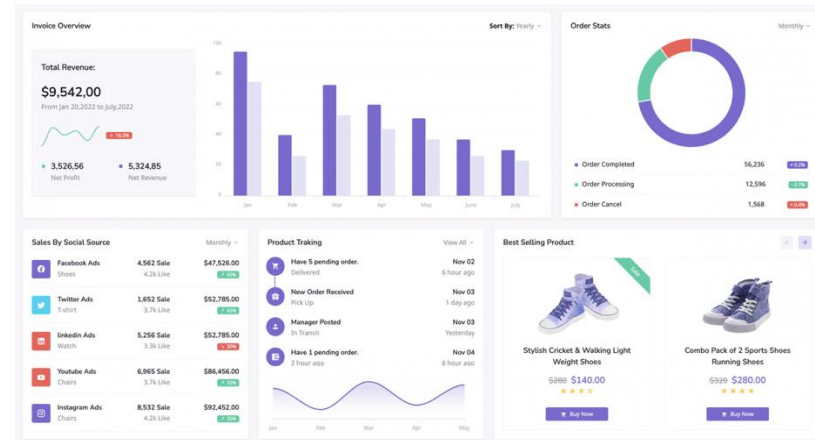
Los líderes del comercio de fertilizantes  
Flujos en 2019



# Introducción a la Visualización de Datos

## ¿Qué es un Dashboard?

- Un dashboard, o cuadro de mando, es una herramienta de visualización de datos que presenta métricas clave y KPIs (indicadores clave de rendimiento) de manera gráfica y centralizada.
- Su objetivo es proporcionar una visión clara y rápida del estado de un negocio, proceso o campaña, lo que facilita el monitoreo y la toma de decisiones informadas.





# Introducción a la Visualización de Datos

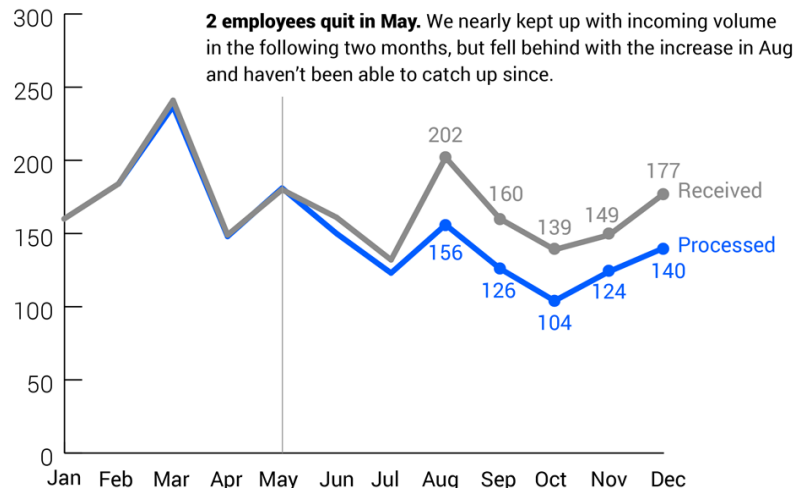
## ¿Qué es Storytelling?

- El storytelling de datos es la práctica de combinar datos y análisis complejos con una narrativa atractiva para comunicar un mensaje de manera clara, persuasiva y memorable.
- Su objetivo es transformar datos técnicos en una historia comprensible que capte la atención del público y motive una acción, integrando visualizaciones, análisis rigurosos y elementos emocionales para hacer la información más humana y fácil de entender.

### Please approve the hire of 2 FTEs

to backfill those who quit in the past year

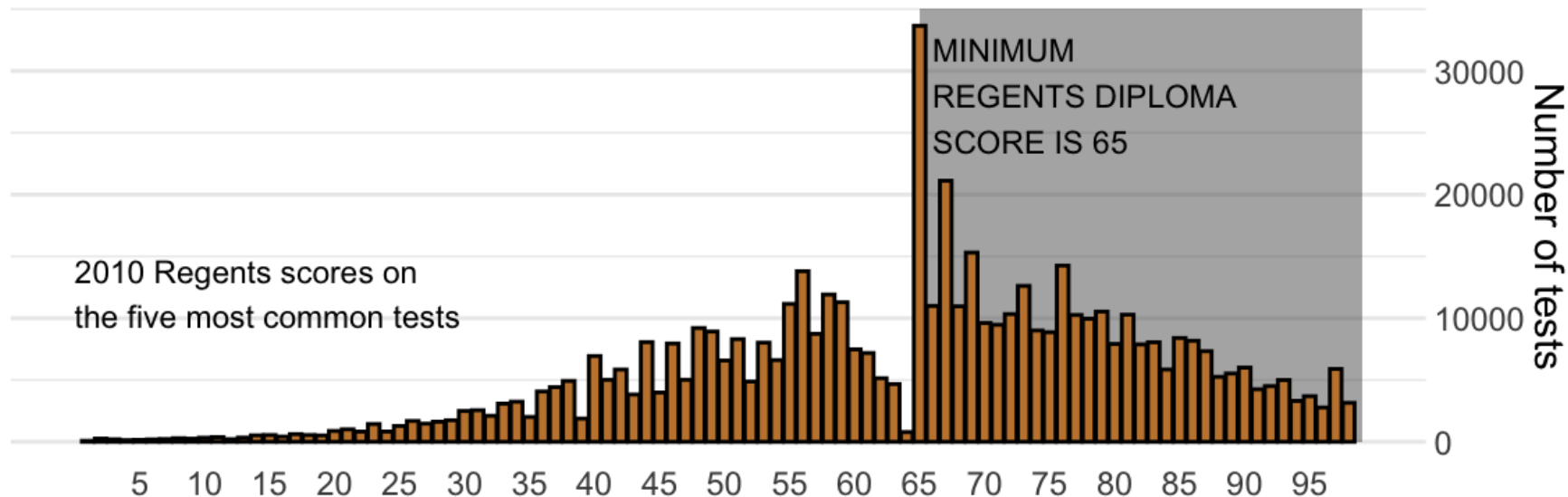
Ticket volume over time





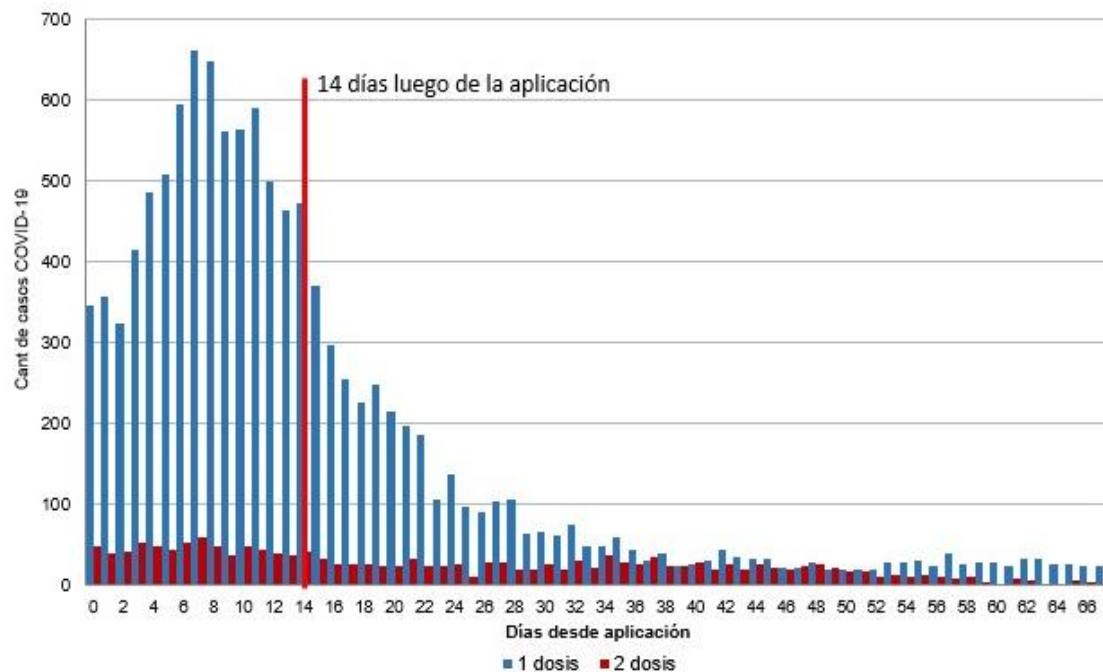
# Introducción a la Visualización de Datos

## Scraping by

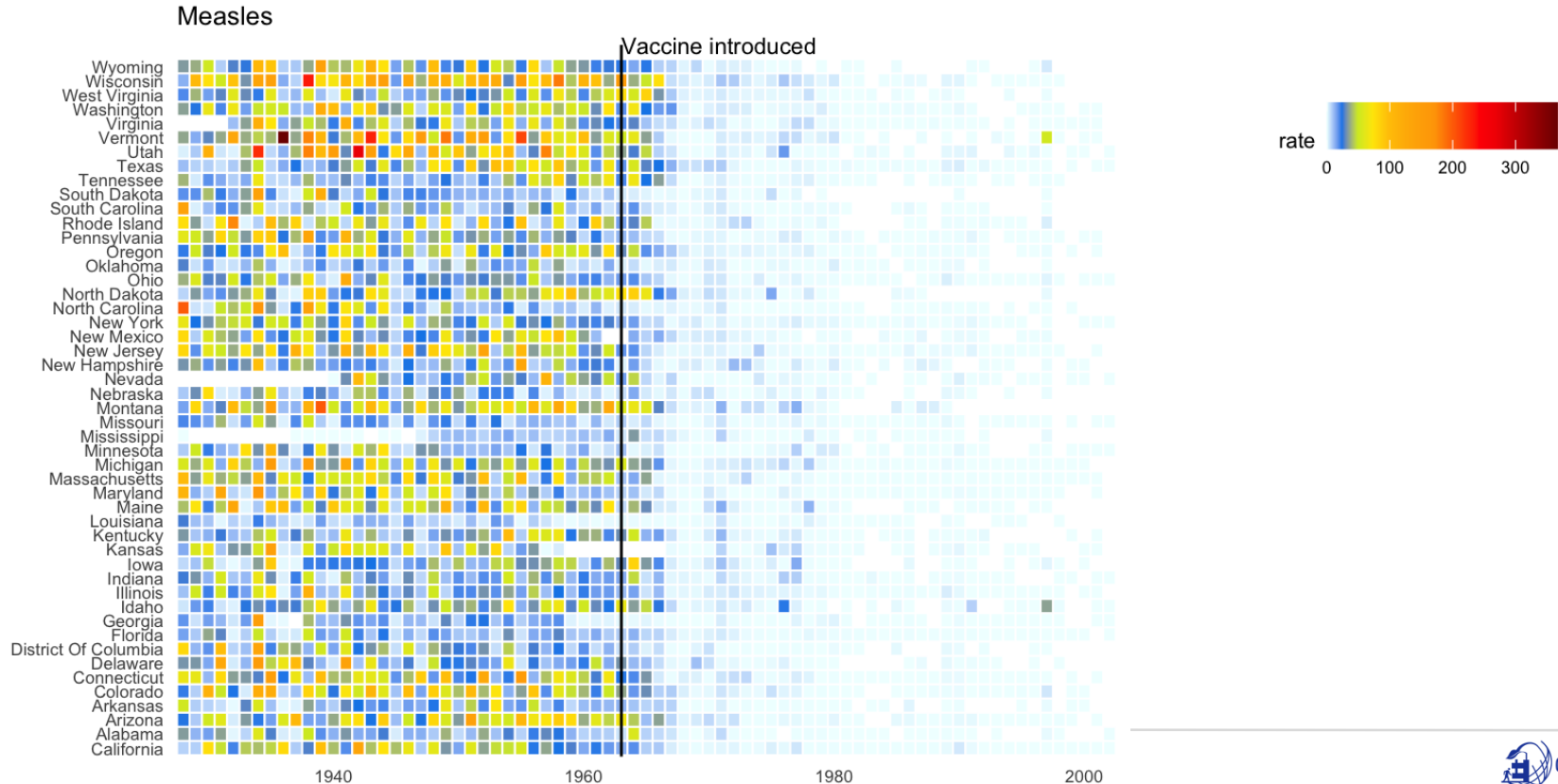


# Introducción a la Visualización de Datos

Casos confirmados de COVID-19 en personas vacunadas según días post aplicación y número de dosis



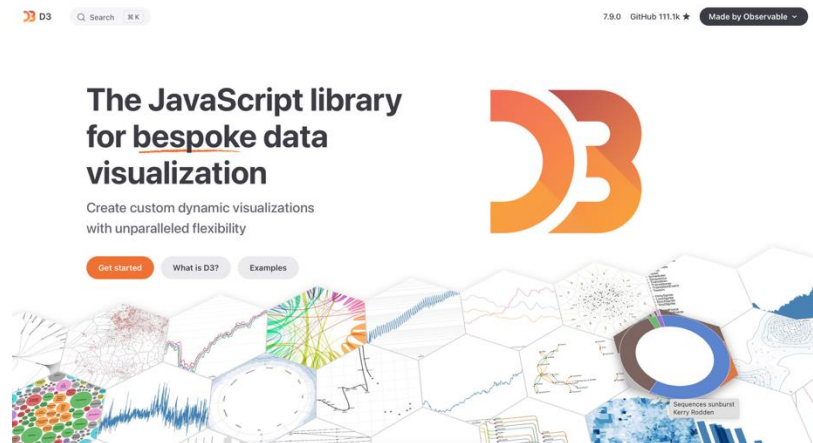
# Introducción a la Visualización de Datos



# Introducción a D3.js

## ¿Qué es D3.js?

- Es una librería de JavaScript gratuita y de código abierto para la visualización de datos.
- Permite creación de gráficos para el desarrollo web de forma dinámica y flexible.



<https://d3js.org/what-is-d3>

# Introducción a D3.js

---

## ¿Cómo usar D3.js?

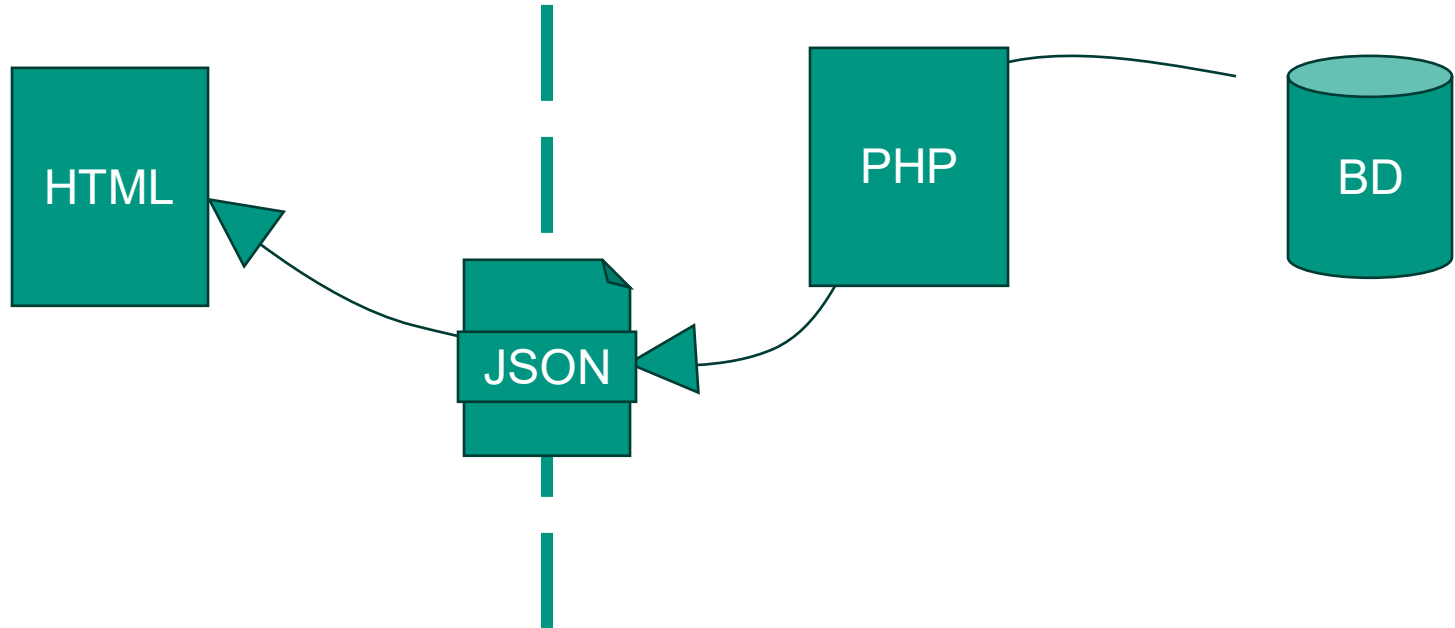
- Se debe indicar el CDN en el HEAD de HTML

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```

- Con esto, se puede tener acceso a todo el contenido de funcionalidades.

# Introducción a D3.js

- Se requiere utilizar el servidor web para simular el envío de datos



# Introducción a D3.js

## Sección de código en html Archivo de datos JSON

```
[  
  { "nombre": "A", "valor": 10 },  
  { "nombre": "B", "valor": 25 },  
  { "nombre": "C", "valor": 15 },  
  { "nombre": "D", "valor": 30 }  
]
```

Ver ejemplo en:

<https://github.com/profeJoel/taller-de-programaci-n/d3-ejemplo1>

```
<svg width="600" height="400"></svg>  
  
<script>  
  // Cargar datos desde un archivo JSON externo  
  d3.json("datos.json").then(function(datos) {  
    const svg = d3.select("svg");  
    const margin = { top: 20, right: 20, bottom: 50, left: 60 };  
    const width = +svg.attr("width") - margin.left - margin.right;  
    const height = +svg.attr("height") - margin.top - margin.bottom;  
  
    const chart = svg.append("g")  
      .attr("transform", `translate(${margin.left},${margin.top})`);  
  
    const x = d3.scaleBand()  
      .domain(datos.map(d => d.nombre))  
      .range([0, width])  
      .padding(0.2);  
  
    const y = d3.scaleLinear()  
      .domain([0, d3.max(datos, d => d.valor)])  
      .nice()  
      .range([height, 0]);  
  
    chart.append("g")  
      .attr("transform", `translate(0,${height})`)  
      .call(d3.axisBottom(x));  
  
    chart.append("g")  
      .call(d3.axisLeft(y));  
  
    chart.selectAll(".bar")  
      .data(datos)  
      .enter()  
      .append("rect")  
      .attr("class", "bar")  
      .attr("x", d => x(d.nombre))  
      .attr("y", d => y(d.valor))  
      .attr("width", x.bandwidth())  
      .attr("height", d => height - y(d.valor));  
  });  
</script>
```



# Introducción a D3.js

## Sección de código en html Datos en PHP

```
<?php
include 'db.php';

$query = "SELECT estado, COUNT(*) AS cantidad FROM Consulta GROUP BY estado";
$result = $conn->query($query);

$data = [];

while ($row = $result->fetch_assoc()) {
    $data[] = $row;
}

header('Content-Type: application/json');
echo json_encode($data);
?>
```

Ver ejemplo en:

<https://github.com/profeJoel/taller-de-programaci-n/d3-ejemplo2>

```
<svg width="600" height="400"></svg>

<script>
    fetch('data.php')
    .then(response => response.json())
    .then(data => {
        const svg = d3.select("svg"),
            width = +svg.attr("width"),
            height = +svg.attr("height"),
            margin = {top: 20, right: 30, bottom: 40, left: 40};

        const x = d3.scaleBand()
            .domain(data.map(d => d.estado))
            .range([margin.left, width - margin.right])
            .padding(0.2);

        const y = d3.scaleLinear()
            .domain([0, d3.max(data, d => +d.cantidad)]).nice()
            .range([height - margin.bottom, margin.top]);

        svg.append("g")
            .selectAll("rect")
            .data(data)
            .enter().append("rect")
            .attr("class", "bar")
            .attr("x", d => x(d.estado))
            .attr("y", d => y(d.cantidad))
            .attr("height", d => y(0) - y(d.cantidad))
            .attr("width", x.bandwidth());

        svg.append("g")
            .attr("transform", `translate(0,${height - margin.bottom})`)
            .call(d3.axisBottom(x))
            .attr("font-size", "12px");

        svg.append("g")
            .attr("transform", `translate(${margin.left},0)`)
            .call(d3.axisLeft(y))
            .attr("font-size", "12px");

    });
</script>
```

# Creación de Gráficos Básicos

## ► Revisaremos los gráficos mediante un ejemplo

```
CREATE DATABASE IF NOT EXISTS datos_visualizacion;
USE datos_visualizacion;

CREATE TABLE registros (
  id INT AUTO_INCREMENT PRIMARY KEY,
  categoria VARCHAR(50),
  valor INT,
  fecha DATE
);

INSERT INTO registros (categoria, valor, fecha) VALUES
("A", 10, "2023-01-01"),
("B", 20, "2023-02-01"),
("C", 30, "2023-03-01"),
("A", 25, "2023-04-01"),
("B", 15, "2023-05-01"),
("C", 35, "2023-06-01"),
("A", 40, "2023-07-01"),
("B", 10, "2023-08-01"),
("C", 50, "2023-09-01");
```



```
<?php
header('Content-Type: application/json');

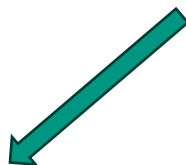
$host = "localhost";
$user = "root";
$pass = "";
$db = "datos_visualizacion";

$conn = new mysqli($host, $user, $pass, $db);
if ($conn->connect_error) {
  die("Conexion fallida: " . $conn->connect_error);
}

$sql = "SELECT categoria, SUM(valor) AS total FROM registros GROUP BY categoria";
$result = $conn->query($sql);

$data = array();
while ($row = $result->fetch_assoc()) {
  $data[] = $row;
}

echo json_encode($data);
$conn->close();
?>
```



```
fetch("data.php")
  .then(res => res.json())
  .then(data => {
    // Usar D3 para dibujar barras con data
  });
```

# Creación de Gráficos Básicos

## ► Gráfico de Líneas

**SELECT** fecha, SUM(valor) **AS** total **FROM** registros **GROUP BY** fecha **ORDER BY** fecha;

```
<svg width="700" height="400"></svg>
<script>
  fetch("data.php")
    .then(res => res.json())
    .then(data => {
      data = data.map((d, i) => ({ ...d, fecha: new Date(2023, i, 1), total: +d.total }));

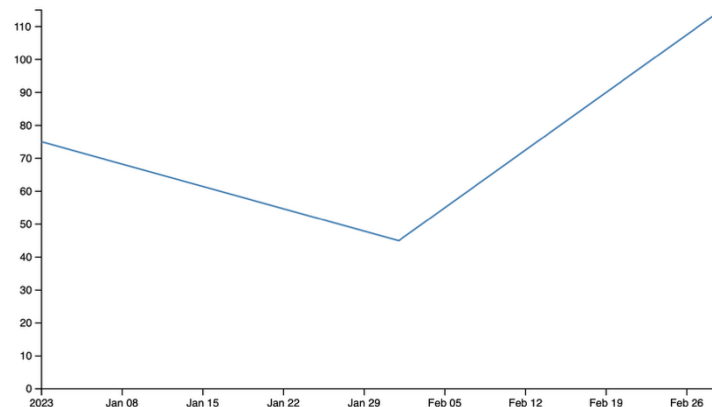
      const svg = d3.select("svg"),
            margin = {top: 20, right: 30, bottom: 30, left: 40},
            width = +svg.attr("width") - margin.left - margin.right,
            height = +svg.attr("height") - margin.top - margin.bottom,
            g = svg.append("g").attr("transform", `translate(${margin.left},${margin.top})`);

      const x = d3.scaleTime()
        .domain(d3.extent(data, d => d.fecha))
        .range([0, width]);

      const y = d3.scaleLinear()
        .domain([0, d3.max(data, d => d.total)])
        .range([height, 0]);

      const line = d3.line()
        .x(d => x(d.fecha))
        .y(d => y(d.total));
```

Gráfico de Línea desde PHP + MySQL



# Creación de Gráficos Básicos

## ► Gráfico de Torta

```
<svg width="500" height="500"></svg>
<script>
  fetch("data.php")
    .then(res => res.json())
    .then(data => {
      const width = 500, height = 500, radius = Math.min(width, height) / 2;

      const svg = d3.select("svg")
        .append("g")
        .attr("transform", `translate(${width / 2},${height / 2})`);

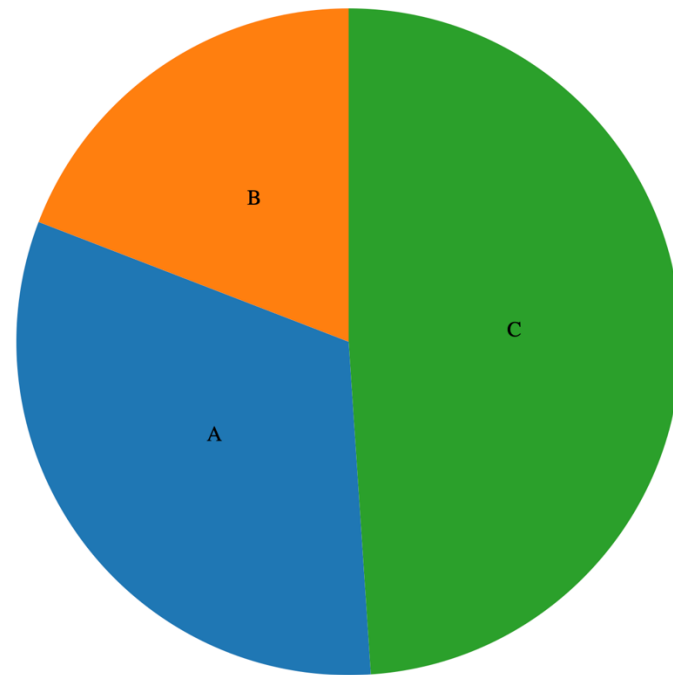
      const color = d3.scaleOrdinal(d3.schemeCategory10);
      const pie = d3.pie().value(d => d.total);
      const arc = d3.arc().innerRadius(0).outerRadius(radius);

      const arcs = svg.selectAll("arc")
        .data(pie(data))
        .enter()
        .append("g");

      arcs.append("path")
        .attr("d", arc)
        .attr("fill", d => color(d.data.categoria));

      arcs.append("text")
        .attr("transform", d => `translate(${arc.centroid(d)})`)
        .attr("text-anchor", "middle")
        .text(d => d.data.categoria);
    });
</script>
```

Gráfico de Torta desde PHP + MySQL



# Creación de Gráficos Básicos

## ► Gráfico de Barras

```
<svg width="600" height="400"></svg>
```

```
<script>
  fetch("data.php")
    .then(res => res.json())
    .then(data => {
      const svg = d3.select("svg");
      const margin = { top: 20, right: 20, bottom: 40, left: 50 };
      const width = +svg.attr("width") - margin.left - margin.right;
      const height = +svg.attr("height") - margin.top - margin.bottom;

      const g = svg.append("g").attr("transform", `translate(${margin.left},${margin.top})`);

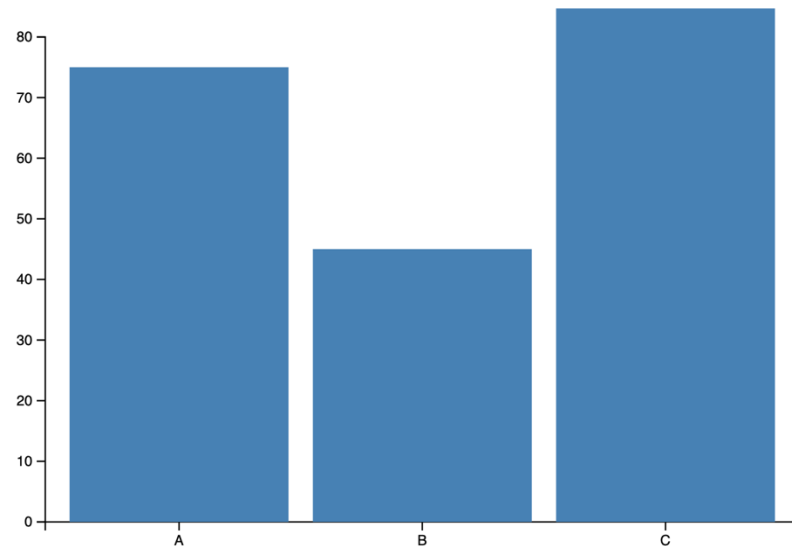
      const x = d3.scaleBand()
        .domain(data.map(d => d.categoria))
        .range([0, width])
        .padding(0.1);

      const y = d3.scaleLinear()
        .domain([0, d3.max(data, d => d.total)])
        .nice()
        .range([height, 0]);

      g.append("g")
        .attr("transform", `translate(0,${height})`)
        .call(d3.axisBottom(x));

      g.append("g").call(d3.axisLeft(y));
    });
</script>
```

## Gráfico de Barras desde PHP + MySQL



# Creación de Gráficos Básicos

## ► Gráfico de Barras

```
<svg width="600" height="400"></svg>
<script>
  const data = Array.from({length: 30}, () => ({
    x: Math.random() * 100,
    y: Math.random() * 100
  }));

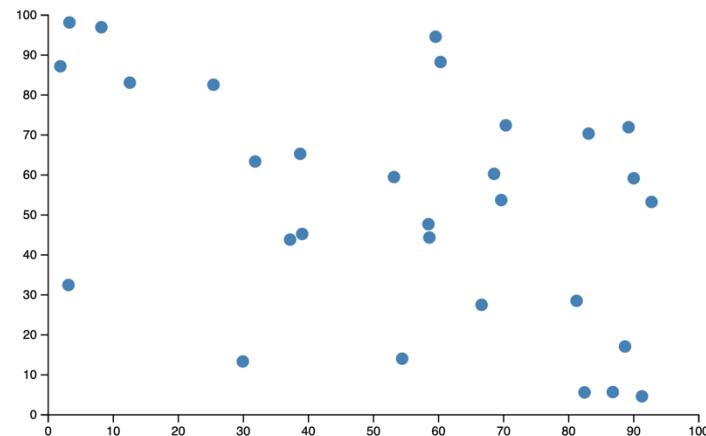
  const svg = d3.select("svg"),
        width = +svg.attr("width"),
        height = +svg.attr("height"),
        margin = 40;

  const x = d3.scaleLinear().domain([0, 100]).range([margin, width - margin]);
  const y = d3.scaleLinear().domain([0, 100]).range([height - margin, margin]);

  svg.append("g").attr("transform", `translate(0,${height - margin})`).call(d3.axisBottom(x));
  svg.append("g").attr("transform", `translate(${margin},0)`).call(d3.axisLeft(y));

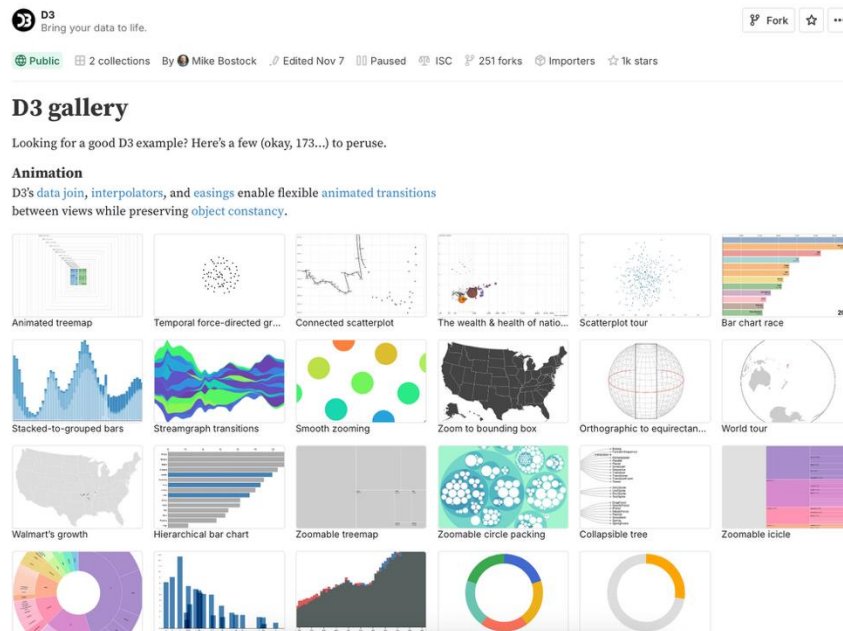
  svg.selectAll("circle")
    .data(data)
    .enter()
    .append("circle")
    .attr("cx", d => x(d.x))
    .attr("cy", d => y(d.y))
    .attr("r", 5)
    .attr("fill", "steelblue");
</script>
```

Gráfico de Dispersión (Datos simulados)



# Creación de Gráficos Básicos

[https://observablehq.com/@d3/gallery?utm\\_source=d3js-org&utm\\_medium=page-nav&utm\\_campaign=try-observable](https://observablehq.com/@d3/gallery?utm_source=d3js-org&utm_medium=page-nav&utm_campaign=try-observable)





# Actividad Final:

---

- Implemente el valor de métrica de secado de su Proyecto
  - **Identifique un gráfico que sea pertinente**
  - **Implemente el gráfico para presentar el datos requerido.**

# Taller de Programación

## Clase 17

**Cristian Vallejos** ([cristian.vallejos@uach.cl](mailto:cristian.vallejos@uach.cl))

**Joel Torres** ([joel.torres@uach.cl](mailto:joel.torres@uach.cl))

