



# ¿Es posible crear gráficas con solo CSS?

Azahara Fernández Guizán  
Juan Manuel Rodríguez Pérez  
[AtlanticaConf2023](#)

SNGULAR

# ¿Quiénes somos?



Azahara Fernández

FrontEnd

Sngular

@azahara\_fergui



Juan M. Rdguez.

FrontEnd

Sngular

@juan\_manuel\_rp



# ¿Por qué usar sólo CSS para hacer gráficas?

La explicación profesional (tras haber usado Chart.js, D3.js, ...)

- Las librerías actuales para crear gráficas en JavaScript suelen tener cierto grado de **complejidad**.
- Añaden peso a tu aplicación y el **rendimiento** se ve afectado.

Al ser de proveedores externos, puede que en un momento determinado tengas problemas con las versiones o que dejen de **mantenerlas**.





¿Has visto el nuevo framework que comparte @midudev? Es para hacer gráficas con solo CSS sin tener que usar librerías.

No, pero suena un poco ilógico.



¿Ilógico por qué?

Porque al final para quitar una librería tienes que usar otra, por mucho CSS que sea...



Pero el rendimiento y el peso no van a tener nada que ver. Sino ¿cómo lo haces?



Pues fácil, usas CSS grid y creas un algoritmo sencillo con un poco de matemáticas.

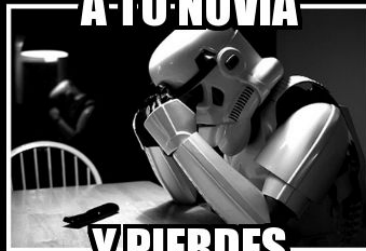


¡Se te va la pinza! Además aunque lo consigas será complejo y poco eficiente...



**APUESTA REALIZADA!!!**

**CUANDO APUESTAS  
A TU NOVIA**



**Y PIERDES  
LA APUESTA**

memegenerator.es

**CUANDO APUESTAS  
CON TU NOVIA Y**



**SABES QUE GANARAS**



# Gráficas con CSS Grid

SNGULAR

# Gráfico de barras

## PASO 1: Formatear los datos.

Obtener el **valor máximo de todos los datos**

`Math.max`

**Redondear** el valor máximo

`Math.ceil`

**Calcular cuantas celdas cada valor:**

`Math.ceil(value * maxCeils / maximumValueInGraph)`

## PASO 2: Pintar el grid.

```
currentGraphDiv.style.gridTemplateRows =  
'repeat(' + maxCeils + ', 1vh)';
```

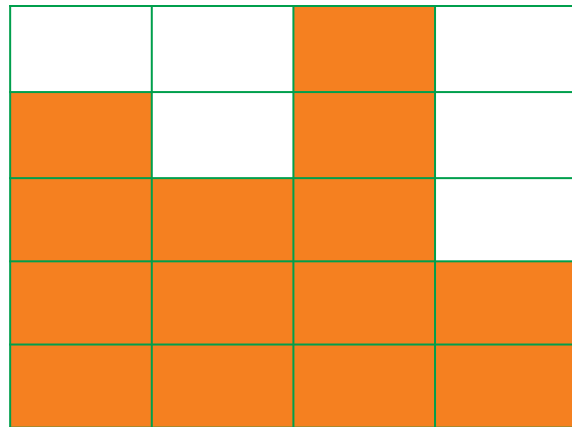
```
columnsWidth = calculatedData.length < 20 ?  
100/calculatedData.length  
: 5;
```

```
currentGraphDiv.style.gridTemplateColumns =  
'repeat(' + calculatedData.length + ',  
columnsWidth + '%)';
```

## PASO 3. Rellenar el grid.

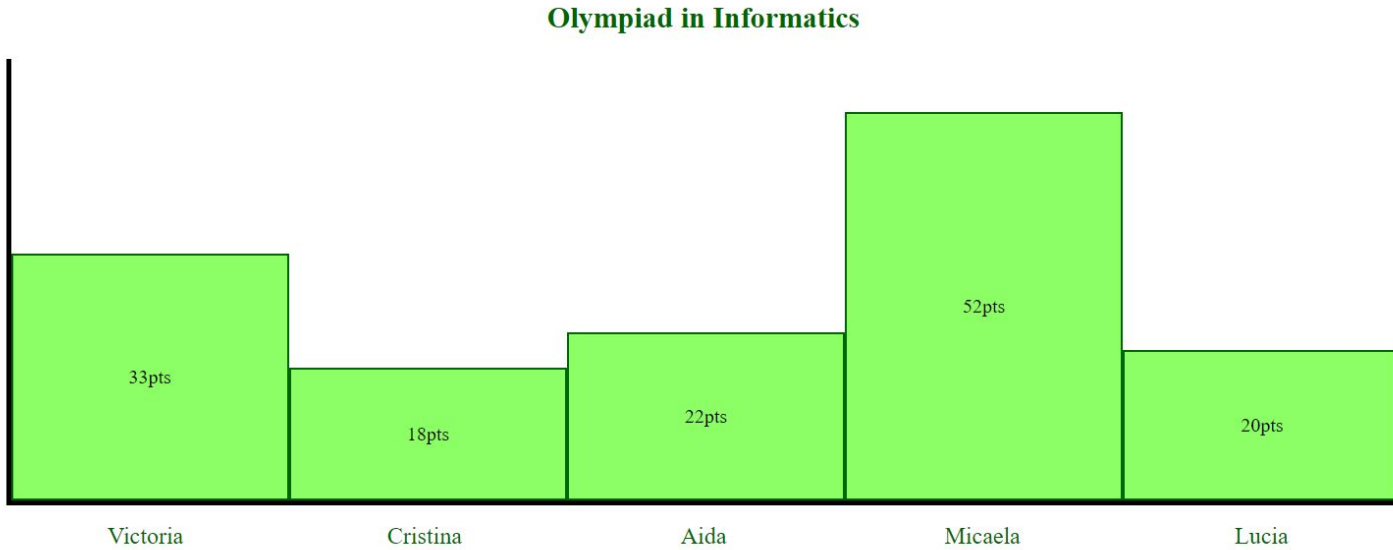
```
currentDiv.style.gridRowStart =  
maxCeils + 1;
```

```
currentDiv.style.gridRowEnd =  
maxCeils + 1 - data.value;
```



## Drawing graphs with CSS

Bar Bubble Area





# Gráfico de burbujas

## PASO 1: Formatear los datos.

Obtener el **valor máximo de todos los datos**

Math.max

**Calcular el porcentaje de cada dato.**

**Calcular tamaño cada punto:**

```
if(valueInPercet > 75){  
    return 3;  
} else if (valueInPercent > 50){  
    return 2;  
} else {  
    return 1;  
}
```

## PASO 2: Pintar el grid.

```
numberColumns = 3 * calculatedData.length;
```

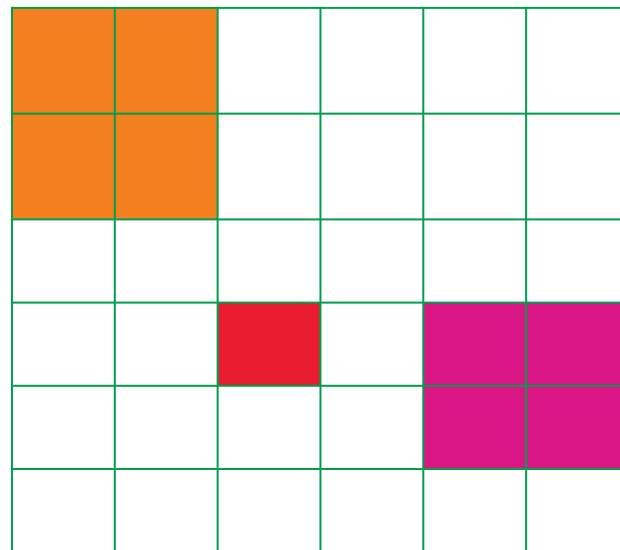
```
currentGraphDiv.style.gridTemplateColumns =  
'repeat(' + numberColumns + ', ' + columnWidth +  
'px)';
```

```
currentGraphDiv.style.gridTemplateRows =  
'repeat(' + numberColumns + ', ' + columnWidth +  
'px)';
```

## PASO 3. Rellenar el grid.

Posicionar en columna que toque  
(3 por cada valor).

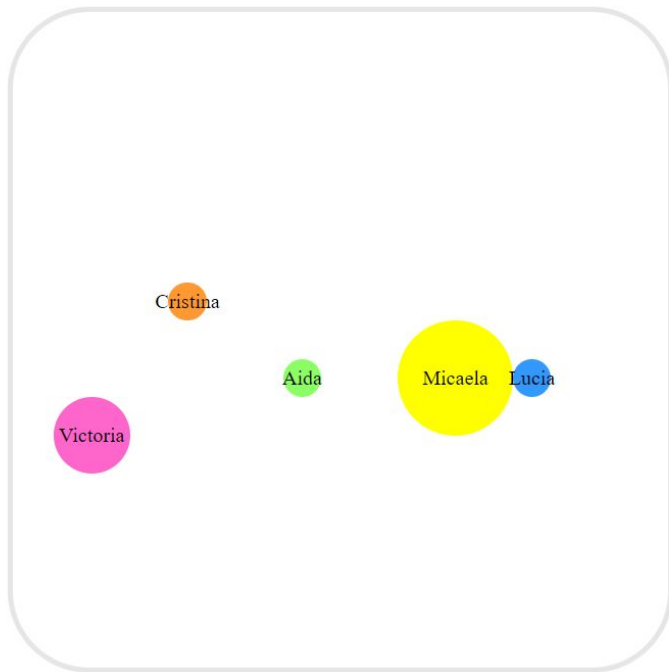
Tomar color al azar y pintar con  
valor.



# Drawing graphs with CSS

Bar Bubble Area

## Olympiad in Informatics



# Gráfico de área

## PASO 1: Formatear los datos.

Obtener el **sumatorio de todos los datos:**

```
data.reduce(function (acc, obj){  
    return acc + obj.value;},  
0);
```

**Calcular tamaño cada dato:**

```
Math.floor(value * (maxCeils /  
valuesSumUp))
```

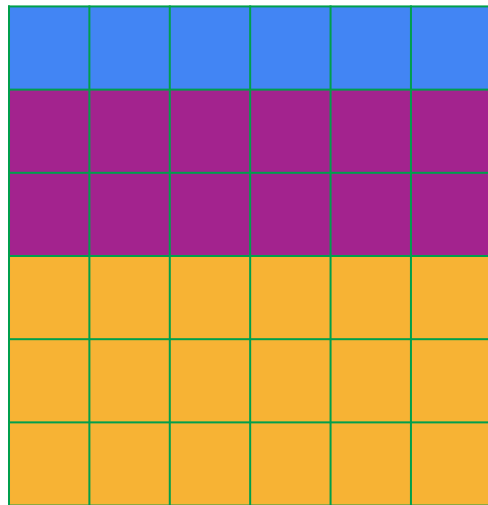
## PASO 2: Pintar el grid.

```
currentGraphDiv.style.gridTemplateColumns =  
'repeat(' + totalCells + ',' + preferredWidth + 'px)';
```

```
currentGraphDiv.style.gridTemplateRows =  
'repeat(' + totalCells + ',' + preferredWidth + 'px)';
```

## PASO 3. Rellenar el grid.

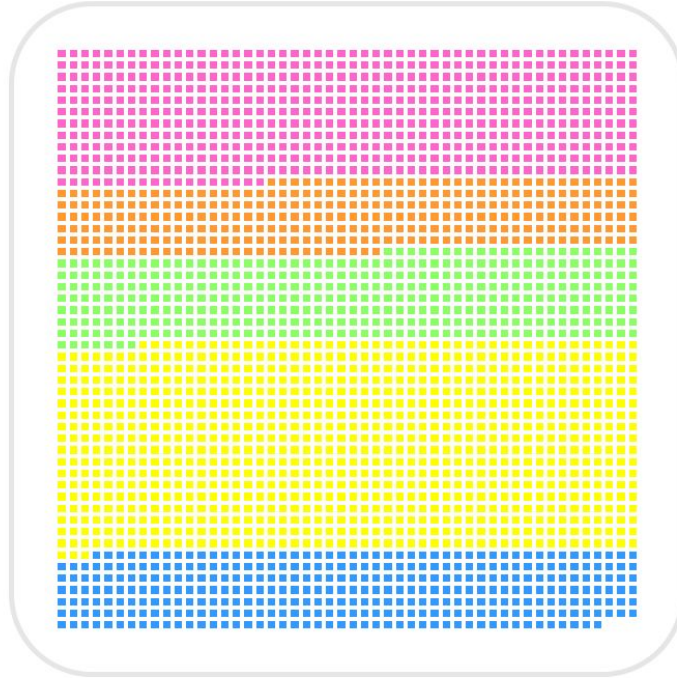
Para cada dato se crean tantos divs como el valor calculado y se añaden al grid.



# Drawing graphs with CSS

Bar Bubble Area

## Olympiad in Informatics



Victoria - 33 pts

Cristina - 18 pts

Aida - 22 pts

Micaela - 52 pts

Lucia - 20 pts



# Gráficas con framework

---

**SNGULAR**

# Charts.css

**¿Qué es?** → Framework usando solo estilos CSS

**Instalación** →

**npm:** ``npm install charts.css``

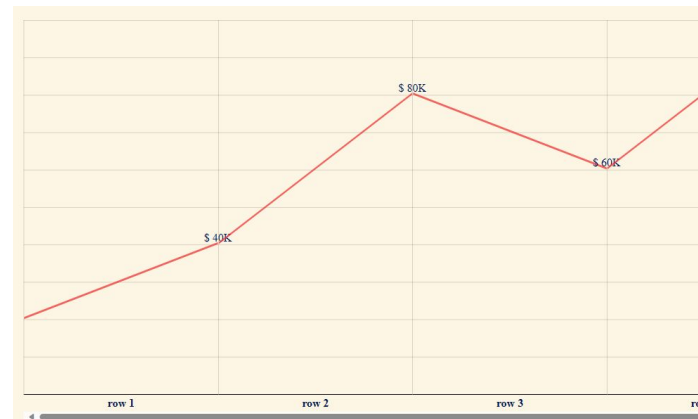
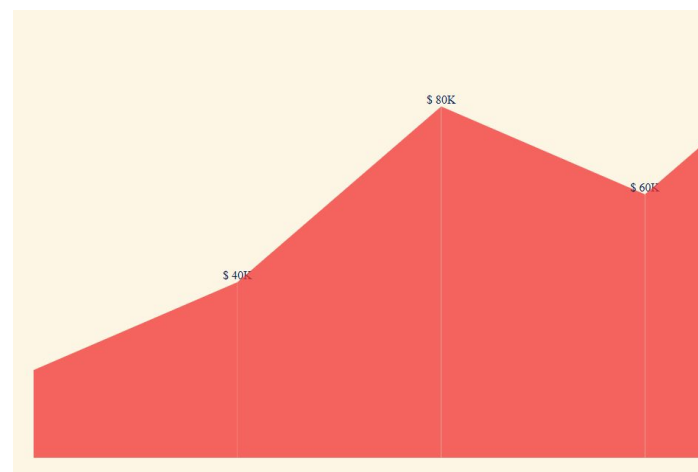
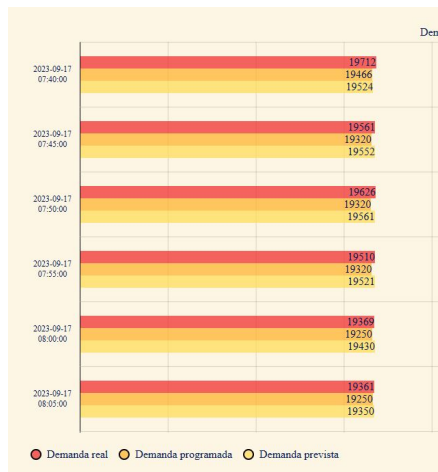
**yarn:** ``yarn add charts.css``

**Estructura base** → Se basa en la utilización de tablas html



# Gáficas disponibles

- **Bar**
- **Column**
- **Area**
- **Line**
- Radial (En desarrollo)
- Pie (En desarrollo)
- Polar (En desarrollo)
- Radar (En desarrollo)
- **Mixed**



# Usando Chart.css

```
<div id="tabBarGraph" class="tab">
  <table class="charts-css bar
show-heading
show-labels
show-primary-axis
show-10-secondary-axes
show-data-axes
data-spacing-20
multiple">

    <caption id="titleBarGraph"></caption>

    <thead>
      <tr>
        <th scope="col"></th>
      </tr>
    </thead>

    <tbody id="tBodyBarGraph">
    </tbody>
  </table>

  <ul id="labelsBarGraph" class="charts-css legend legend-inline legend-circle">
  </ul>
</div>
```

- **Propiedades**

- Definición estructura:  
**"charts-css"**
- Definición tipo gráfica:  
**"bar"** (column, area, line, ...)
- Definición formato:



# Usando Chart.css

```
<div id="tabLineGraph" class="tab">
  <table class="charts-css line
show-heading
show-labels
show-primary-axis
show-10-secondary-axes
show-data-axes">

    <caption id="titleLineGraph"></caption>

    <thead>
      <tr>
        <th scope="col"></th>
      </tr>
    </thead>

    <tbody id="tBodyLineGraph">
      <tr>
        <th scope="row">row 1</th>
        <td style="--start: 0.2; --size: 0.4">
          <span class="data"> $ 40K </span>
        </td>
      </tr>
      <tr>
        <th scope="row">row 2</th>
        <td style="--start: 0.4; --size: 0.8">
          <span class="data"> $ 80K </span>
        </td>
      </tr>
    </tbody>
  </table>

  <ul id="labelsLineGraph" class="charts-css legend legend-inline legend-circle">
  </ul>
</div>
```



# Conclusiones

---

SNGULAR

- ❖ Dibujarlos con CSS Grid solo si conoces muy bien datos y no se te complica la matemática y además no quieres un gráfico muy bonito.
- ❖ Usando el framework es muy fácil de emplear y customizar al ser solo con css pero aún está en desarrollo y los valores deben estar entre 0 y 1.



**🏠🏠 En Sngular siempre buscamos buena gente y gente buena**

Puedes encontrar el repositorio en: [github.com/Afaya/AtlanticaConf\\_2023](https://github.com/Afaya/AtlanticaConf_2023)