



ING. EN SISTEMAS COMPUTACIONALES
TECNOLOGIAS WEB

“investigación 2: generación de graficas en Angular”

ALUMNOS:

Duran Plascencia Alejandro
García Salazar Kenya Josselin
Nathan Iñaki
Noriega Villalobos Andrés
Pedroza Casillas Cecilia Lilliana
***Rosas Aguilar Ariadna Judith**

Semestre: 6° Grupo: “B”

PROFESOR:

Georgina Salazar Partida

Aguascalientes, Ags.

Lunes, 29 de mayo del 2023

Link a videoconferencia:

Librerías investigadas

Chart Js javascript: Charts.js es una librería de JavaScript que utiliza el canvas de HTML5 para mostrar impresionantes gráficos para tu web. Todo lo que se requiere hacer es añadir el script de Chart.js al documento y se podrá hacer uso de su funcionalidad. Es una de las bibliotecas de visualización más simples para JavaScript.

GOOGLE charts: Es una aplicación de Google para realizar estadísticas web, de fácil uso para desarrolladores de software web, usado en muchos campos como Google Analytics, se puede usar con diferentes formatos, Json, Javascript y plugins que se pueden integrar con varios lenguajes de programación.

Morris JS Chart: Morris.js es una biblioteca JS liviana que usa jQuery y Raphael para generar varios diagramas de temporización. Su ventaja es que es liviano y fácil de usar, y es relativamente fácil comenzar.

Plotly de javascript: Es una librería de declarativa de alto nivel construida encima de d3.js y stack.gl. Se pueden crear gráficos interactivos con facilidad usando Plotly.js. Cualquier gráfica que crees con la librería es equipada con características como acercamiento, alejamiento, paneo, auto escalado, etc. Estas características son muy útiles cuando quieres estudiar gráficos con un gran número de puntos graficados. Todos estos eventos son expuestos en la API, así que puedes escribir código personalizado para realizar tus propias acciones cuando cualquiera de estos eventos es disparado.

Librería seleccionada:

CHART JS JAVASCRIPT debido a que cuenta con una gran variedad de gráficos de los cuales podemos disponer de los siguientes tipos de gráficos integrados:

- Diagrama de dispersión
- Gráfico de líneas
- Gráfico de barras
- Gráfico circular
- Tabla de donas
- Gráfico de burbujas
- Gráfico de área
- Gráfico de radar
- Gráfico mixto

Los gráficos pueden ser personalizados y pueden ser configurados para que el diseño se adapte al diseño del sitio web.

Además de que es compatible con varios navegadores y pueden ser conectados los gráficos a un panel interactivo.

Instalación de la librería:

Dentro de la terminal del editor ejecutar el comando para la instalación.

```
npm install chart.js --save
```

También puede ser usado el CDN

```
<!--Chart.js JS CDN-->
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart
.min.js"></script>
```

Se tiene que importar la librería en el app.component.ts, además de agregar otras cosas como ViewChild para que pueda ser visualizada la gráfica.

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { Chart } from 'chart.js';
```

Dentro de la clase de app.component.ts se establece el código de la gráfica que se quiere usar. Indicando con el “type” el tipo de grafica que queremos y en “data”, los datos de la grafica.

```
export class AppComponent {
  title = 'graficas';

  constructor(){}

  ngOnInit(){}

  canvas: any;
  ctx: any;
  @ViewChild('mychart') mychart:any;

  ngAfterViewInit(){
    this.canvas = this.mychart.nativeElement;
    this.ctx = this.canvas.getContext('2d');

    new Chart(this.ctx,{
      type: ,
      data: {

      },
    });
  }
}
```

Por default se instala la última versión de chart.js por lo que es necesario a cambiar la versión al 2.9.4, la cual es la que nos servirá para este proyecto.

Usando el siguiente comando cambiaremos la versión del chart.js

```
npm i chart.js@2.9.4
```

Se agregan los derivados de la librería para hacer uso de los types.

```
npm i -save-dev @types/chart.js
```

Para que las gráficas puedan ser visualizadas dentro del html se integra la etiqueta canva al app.component.html

```
<div class="container">
  <div style="width: 100%;">
    <canvas id="myChart" width="700" height="250" #mychart></canvas>
  </div>
</div>
```

Ejemplo de un código con los datos de la grafica:

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { Chart } from 'chart.js';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  title = 'graficas';

  constructor(){}

  ngOnInit(){}

  canvas: any;
  ctx: any;
  @ViewChild('mychart') mychart: any;

  ngAfterViewInit(){
    this.canvas = this.mychart.nativeElement;
    this.ctx = this.canvas.getContext('2d');

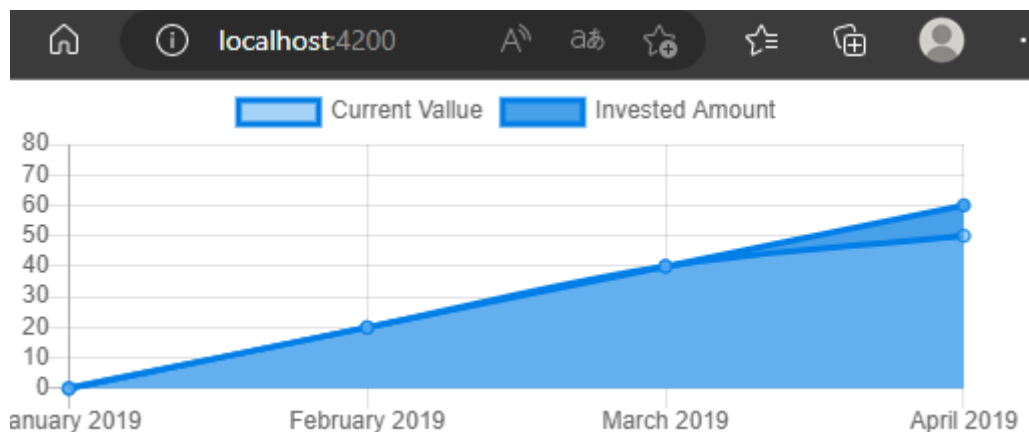
    new Chart(this.ctx,{
      type: 'line',
      data: {
        datasets: [{
          label: 'Current Vallue',
```

```

    data: [0,20,40,50],
    backgroundColor: "rgb(115 185 243 / 65%)",
    borderColor: "#007ee7",
    fill: true,
  },
  {
    label: 'Invested Amount',
    data: [0,20,40,60,80],
    backgroundColor: "#47a0e8",
    borderColor: "#007ee7",
    fill: true,
  }],
  labels: ['January 2019', "February 2019", "March 2019", "April
2019"]
},
});
}
}

```

Resultado del código:



BIBLIOGRAFIA:

[Gráfico.js \(w3schools.com\)](https://www.w3schools.com/chartjs/)

[How to use Chart.js | 11 Chart.js Examples \(ordinarycoders.com\)](https://ordinarycoders.com/chartjs-examples/)

[Charts | Google for Developers](https://developers.google.com/chart/)

[Usando morris.js para hacer un método de gráfico de líneas - programador clic \(programmerclick.com\)](https://programmerclick.com/morrisjs-line-chart/)

[Biblioteca gráfica de javascript plotly en JavaScript](https://d3js.org/)