

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Programacion Web 2				
TÍTULO DE LA PRÁCTICA:	Tarea Ajax				
NÚMERO DE PRÁCTICA:	4	AÑO LECTIVO:	2025	NRO. SEMESTRE:	1
FECHA DE PRESENTACIÓN	03/05/2024	HORA DE PRESENTACIÓN			
INTEGRANTE (s): Silva Pino Jesus Francisco				NOTA:	
DOCENTE(s): CARLO JOSE LUIS CORRALES DELGADO					

SOLUCIÓN Y RESULTADOS
I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS problema1. Haga gráficos comparativos entre regiones elegidas por el usuario. Implementacion del HTML
<pre> Tarea Ajax > problema1.html > html 1 <!DOCTYPE html> 2 <html lang="en"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <meta http-equiv="X-UA-Compatible" content="IE=edge"> 7 <meta name="viewport" content="width=device-width, initial-scale=1.0"> 8 <title>Problema1</title> "Problema": Unknown word. 9 </head> 10 11 <body> 12 <h1>Comparar Regiones</h1> "Comparar": Unknown word. 13 14 <div> 15 <label>Region 1:</label> 16 <select id="region1"></select> 17 18 <label>Region 2:</label> 19 <select id="region2"></select> 20 21 <button onclick="compararRegiones()">Comparar</button> "comparar": Unknown word. 22 </div> 23 24 <div id="chartContainer"> 25 <canvas id="myChart" width="400" height="200"></canvas> 26 </div> 27 <script src="problem1.js"></script> 28 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script> 29 </body> 30 31 </html> </pre>

Creacion de las funciones en javascript: cargarDatos(), llenarSelectores(), compararRegiones()

```
let datosRegiones = [];    "datos": Unknown word.
let miGrafico = null;      "Grafico": Unknown word.

window.onload = function () {
    cargarDatos();          "cargar": Unknown word.
};

function cargarDatos() {    "cargar": Unknown word.
    const xhr = new XMLHttpRequest();
    xhr.open('Get', 'data.json', 'true');

    xhr.onload = function () {
        if (this.status === 200) {
            datosRegiones = JSON.parse(this.responseText);    "datos": Unknown word.
            llenarSelectores();    "llenar": Unknown word.
        }
        else {
            alert('Error al cargar los datos');    "cargar": Unknown word.
        }
    }
    xhr.send();
}

function llenarSelectores() {    "llenar": Unknown word.
    const select1 = document.getElementById('region1')
    const select2 = document.getElementById('region2')

    datosRegiones.forEach(region => {    "datos": Unknown word.
        const option1 = document.createElement('option');
        option1.value = region.region;
        option1.textContent = region.region;
        select1.appendChild(option1);

        const option2 = document.createElement('option');
        option2.value = region.region;
        option2.textContent = region.region;
        select2.appendChild(option2);
    });
}

function compararRegiones() {    "comparar": Unknown word.
    const region1 = document.getElementById('region1').value;
    const region2 = document.getElementById('region2').value;

    if (!region1 || !region2) {
        alert('Selecciona dos regiones para comparar');    "Selecciona": Unknown word.
        return;
    }
}
```

crear grafico para comparar las regiones

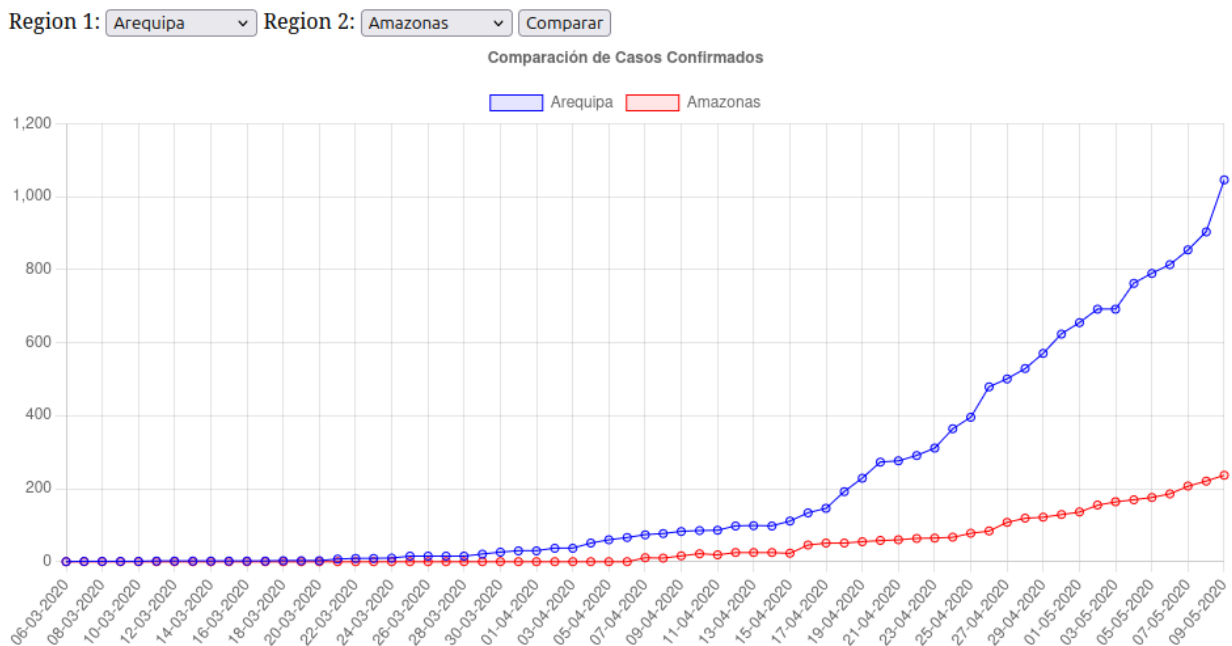
```
function crearGrafico(fechas, casos1, casos2, nombre1, nombre2) {    "crear": Unknown word.
    const ctx = document.getElementById('myChart').getContext('2d');

    // Si ya existe un gráfico, lo destruimos    "existe": Unknown word.
    if (miGrafico) {    "Grafico": Unknown word.
        miGrafico.destroy();    "Grafico": Unknown word.
    }

    miGrafico = new Chart(ctx, {
        type: 'line',
        data: {
            labels: fechas,    "fechas": Unknown word.
            datasets: [
                {
                    label: nombre1,    "nombre": Unknown word.
                    data: casos1,
                    borderColor: 'blue',
                    backgroundColor: 'rgba(0, 0, 255, 0.1)',
                    borderWidth: 1
                },
                {
                    label: nombre2,    "nombre": Unknown word.
                    data: casos2,
                    borderColor: 'red',
                    backgroundColor: 'rgba(255, 0, 0, 0.1)',
                    borderWidth: 1
                }
            ]
        },
        options: {
            responsive: true,
            plugins: {
                title: {
                    display: true,
                    text: 'Comparación de Casos Confirmados'    "Comparación": Unknown word.
                }
            },
            scales: {
                y: {
                    beginAtZero: true
                }
            }
        }
    });
}
```

resultado

Comparar Regiones



Implementacion del JS con las funciones necesarias cargarDatos()

```
Tarea Ajax > problema2.js > mostrarGrafico > datasets > regionesMostrar.map() callback
1 let datosRegiones = []; "datos": Unknown word.
2 let miGrafico = null; "Grafico": Unknown word.
3
4 window.onload = function () {
5   cargarDatos(); "cargar": Unknown word.
6 };
7
8 function cargarDatos() { "cargar": Unknown word.
9   const xhr = new XMLHttpRequest();
10  xhr.open('GET', 'data.json', true);
11
12  xhr.onload = function () {
13    if (this.status === 200) {
14      datosRegiones = JSON.parse(this.responseText); "datos": Unknown word.
15    } else {
16      alert('Error al cargar los datos'); "cargar": Unknown word.
17    }
18  };
19
20  xhr.send();
21 }
22
```

funcion mostrarGrafico() filtrando las regiones de Lima y Callao

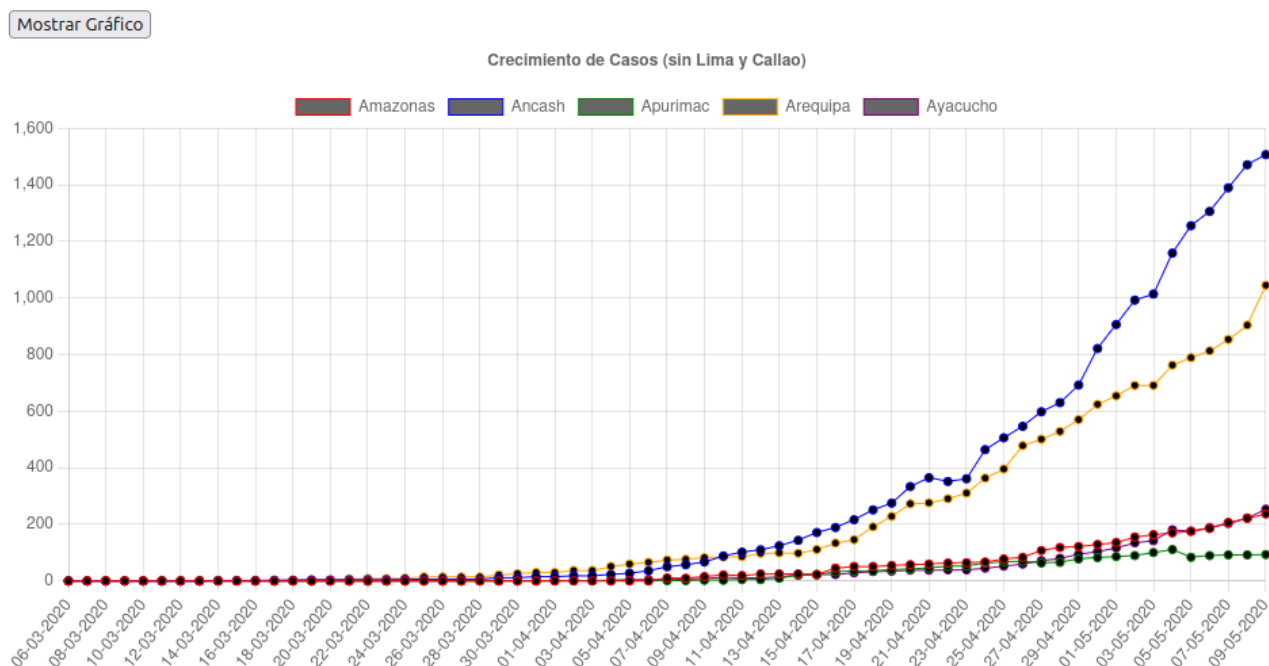
```
function mostrarGrafico() { "mostrar": Unknown word.
  const regionesFiltradas = datosRegiones.filter(region => "regiones": Unknown word.
    region.region !== "Lima" && region.region !== "Callao"
  );
  const regionesMostrar = regionesFiltradas.slice(0, 5); "regiones": Unknown word.
  const fechas = regionesMostrar[0].confirmed.map(item => item.date); "fechas": Unknown word.
  const datasets = regionesMostrar.map((region, index) => { "regiones": Unknown word.
    // colores básicos para cada region "colores": Unknown word.
    const colores = ['red', 'blue', 'green', 'orange', 'purple']; "colores": Unknown word.
    return {
      label: region.region,
      data: region.confirmed.map(item => parseInt(item.value)),
      borderColor: colores[index], "colores": Unknown word.
      backgroundColor: colores[index] + '20', "colores": Unknown word.
      borderWidth: 1
    };
  });
  crearGrafico(fechas, datasets); "crear": Unknown word.
}
```

crearGrafico()

```
48 function crearGrafico(fechas, datasets) { "crear": Unknown word.
49   const ctx = document.getElementById('myChart').getContext('2d');
50
51   if (miGrafico) { "Grafico": Unknown word.
52     miGrafico.destroy();
53   }
54
55   miGrafico = new Chart(ctx, {
56     type: 'line',
57     data: {
58       labels: fechas, "fechas": Unknown word.
59       datasets: datasets
60     },
61     options: {
62       responsive: true,
63       plugins: {
64         title: {
65           display: true,
66           text: 'Crecimiento de Casos (sin Lima y Callao)' "Crecimiento": Unknown word.
67         }
68       },
69       scales: {
70         y: {
71           beginAtZero: true
72         }
73       }
74     }
75   });
76 }
```

resultado

Crecimiento en Regiones (sin Lima y Callao)



II. SOLUCIÓN DEL CUESTIONARIO

III. CONCLUSIONES

El uso de la librería **Chart.js** facilita la representación gráfica de datos usados en este proyecto

RETROALIMENTACIÓN GENERAL

REFERENCIAS Y BIBLIOGRAFÍA

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p align="center">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p align="center">Código: GUIA-PRLE-001</p>	<p align="right">Página: 6</p>

--