

Informe de Laboratorio 4

Tema: AJAX

Nota

Estudiante	Escuela	Asignatura
Sergio Hanco Mullisaca Noa Camino Yenaro Joel shanco@unsa.edu.pe ynoa@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: II Código:

Laboratorio	Tema	Duración
4	AJAX	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 16 Mayo 2024	Al 18 Mayo 2024

1. Tarea

- Informe de laboratorio
- Video en Flip
- Ejercicios Propuestos
- Ejercicios

2. Equipos, materiales y temas utilizados

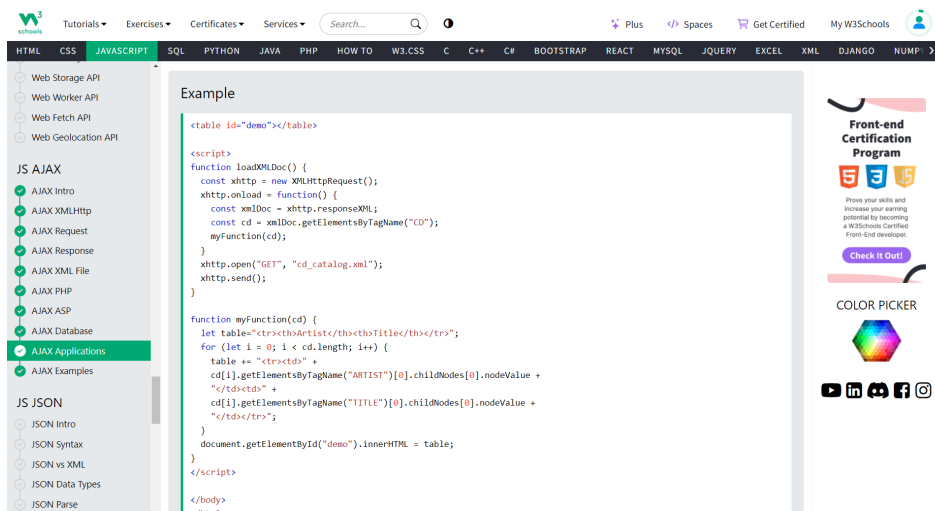
- VS
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.

3. URL de Repositorio Github

- URL del video en yt.
- <https://youtu.be/nc4tQ6ytBUw>
- URL del video en flip.
- https://flip.com/s/VeHz2Bb6FT_c
- URL del GITHUB.
- https://github.com/shanccom/Programacion_Web_2.git
- <https://github.com/ynoacamino/pweb2>

4. Actividades

4.1. W3school



4.2. Ejercicios Propuestos

- Listar los archivos Markdown disponibles
- Ver el contenido de un archivo Markdown traducido a HTML
- Crear nuevos archivos MarkDown y almacenarlos en el servidor

Listing 1: EJERCICIOS PROPUESTOS

```
const express = require('express');
const Markdown = require('markdown-it');
const fs = require('node:fs');
const { request } = require('node:http');
const path = require('node:path');
const bodyParser = require('body-parser');

const md = new Markdown();
```

```
const app = express();

app.use(express.static('public'));
app.use(bodyParser.urlencoded({extended: false}));

app.get('/', (request, response) => {
  response.send('Hola mundo');
})

app.get('/list', (request, response) => {
  response.send(`
    <ul>
      <li><a href="/primero.md">Markdown 1</a></li>
      <li><a href="/segundo.md">Markdown 2</a></li>
      <li><a href="/tercero.md">Markdown 3</a></li>
    </ul>
  `);
})

app.get('/html', (request, response) => {
  fs.readFile(path.join(__dirname, "public", "primero.md"), {encoding: "utf-8"}, (err,
    data) => {
    if (err) {
      response.send("Error al leer el archivo");
      return;
    }

    const htmlCode = md.render(data);

    response.send(htmlCode);
  })
})

app.get('/upload', (request, response) => {
  response.sendFile(path.join(__dirname, "public", "upload.html"));
});

app.post("/upload", async (request, response) => {
  const body = request.body;

  const { title, content } = body;

  fs.writeFile(path.join(__dirname, "public", title + ".md"), content, (err) => {
    if (err) {
      response.send("Error al guardar el archivo");
      return;
    }

    response.send(`Archivo guardado correctamente visítelo <a
      href="/${title}.md">aquí</a>`);
  })
});

app.listen(3000, () => {
```

```
console.log('Servidor escuchando en http://localhost:3000/')
})
```

- HTML usado para la ejecución del código.

Listing 2: HTML PARA LOS EJERCICIOS PROPUESTOS

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Subida de markdowns</title>
</head>
<body>
  <form action="/upload" method="POST">
    <label for="title">Titulo</label>
    <input type="text" name="title" id="title">
    <label for="content">Contenido</label>
    <textarea name="content" id="content" cols="30" rows="10"></textarea>
    <button type="submit">Enviar</button>
  </form>
</body>
</html>
```

4.3. Ejercicios

- Liste todas las “regiones”.

Listing 3: PROBLEMA1-INDEX

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Problema 1</title>
</head>
<body>
  <h1>Problema 1</h1>
  <div id="container">

  </div>
  <script src="./script.js"></script>
</body>
</html>
```

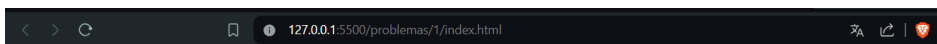
Listing 4: PROBLEMA1-SCRIPT

```
const $container = document.getElementById('container');

const xhr = new XMLHttpRequest();

xhr.open('GET', 'http://localhost:8000/data', true);
```

```
xhr.onreadystatechange = function () {  
  if(xhr.readyState === 4 && xhr.status === 200) {  
    const json = JSON.parse(xhr.responseText);  
    json.forEach((region) => {  
      const $region = document.createElement('div');  
      $region.textContent = region.region;  
      $container.appendChild($region);  
    })  
  }  
}  
  
xhr.send();
```



Problema 1

Amazonas
Ancash
Apurimac
Arequipa
Ayacucho
Cajamarca
Callao
Cusco
Huancaavelica
Huamuco
Ica
Junin
La Libertad
Lambayeque
Lima
Loreto
Madre de Dios
Moquegua
Pasco
Piura
Puno
San Martin
Tacna
Tumbes
Ucayali

- Muestre el número total de confirmados por región.

Listing 5: PROBLEMA2-INDEX

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Problema 1</title>  
</head>  
<body>  
  <h1>Nmero total de confirmados por regin</h1>  
  <div id="container">  
  
  </div>  
  <script src="./script.js"></script>  
</body>  
</html>
```

Listing 6: PROBLEMA2-SCRIPT

```
const $container = document.getElementById('container');

const xhr = new XMLHttpRequest();

xhr.open('GET', 'http://localhost:8000/data', true);
xhr.onreadystatechange = function () {
    if(xhr.readyState === 4 && xhr.status === 200) {
        const data = JSON.parse(xhr.responseText);

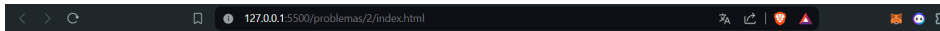
        for (const region of data) {
            const $region = document.createElement('div');

            let totalConfirmados = 0;
            for (const confirmado of region.confirmed) {
                totalConfirmados += parseInt(confirmado.value);
            }

            $region.textContent = `${region.region}: ${totalConfirmados} confirmados`;

            $container.appendChild($region);
        }
    }
}

xhr.send();
```

**Número total de confirmados por región**

Amazonas: 2985 confirmados
Ancash: 18753 confirmados
Apurímac: 1740 confirmados
Arequipa: 13817 confirmados
Ayacucho: 2540 confirmados
Cajamarca: 4506 confirmados
Callao: 78099 confirmados
Cusco: 6187 confirmados
Huancaavelica: 2332 confirmados
Huancayo: 4716 confirmados
Ica: 11925 confirmados
Junín: 10171 confirmados
La Libertad: 18175 confirmados
Lambayeque: 51206 confirmados
Lima: 626744 confirmados
Loreto: 30242 confirmados
Madre de Dios: 2438 confirmados
Moquegua: 2443 confirmados
Pasco: 2449 confirmados
Piura: 29966 confirmados
Puno: 2078 confirmados
San Martín: 5445 confirmados
Tacna: 2457 confirmados
Tumbes: 7796 confirmados
Ucayali: 14488 confirmados

- Encuentre las 10 regiones cuya suma total sea la mayor.

Listing 7: PROBLEMA3-INDEX

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Top 10 Regiones con Mayor Nmero de Confirmados</title>
</head>
<body>
  <h1>Top 10 Regiones con Mayor Nmero de Confirmados</h1>
  <div id="container">

  </div>
  <script src="./script.js"></script>
</body>
</html>
```

Listing 8: PROBLEMA3-SCRIPT

```
const $container = document.getElementById('container');

const xhr = new XMLHttpRequest();

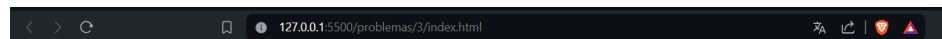
xhr.open('GET', 'http://localhost:8000/data', true);
xhr.onreadystatechange = function () {
  if(xhr.readyState === 4 && xhr.status === 200) {
    const data = JSON.parse(xhr.responseText);

    const regionSumas = [];
    for (const region of data) {
      let totalConfirmados = 0;
      for (const confirmado of region.confirmed) {
        totalConfirmados += parseInt(confirmado.value);
      }
      regionSumas.push({ region: region.region, total: totalConfirmados });
    }

    regionSumas.sort((a, b) => b.total - a.total);

    for (let i = 0; i < 10; i++) {
      const $region = document.createElement('div');
      $region.textContent = `${i + 1}. ${regionSumas[i].region}: ${regionSumas[i].total}
        confirmados`;
      $container.appendChild($region);
    }
  }
}

xhr.send();
```



Top 10 Regiones con Mayor Número de Confirmados

1. Lima: 626744 confirmados
2. Callao: 78099 confirmados
3. Lambayeque: 51206 confirmados
4. Loreto: 30242 confirmados
5. Piura: 29966 confirmados
6. Ancash: 18753 confirmados
7. La Libertad: 18175 confirmados
8. Ucayali: 14488 confirmados
9. Arequipa: 13817 confirmados
10. Ica: 11925 confirmados

- Visualice un gráfico en el tiempo de los valores para la región de Arequipa

Listing 9: PROBLEMA4-INDEX

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Problema 4</title>
  <script type="text/javascript"
    src="https://www.gstatic.com/charts/loader.js"></script>
</head>
<body>
  <h1>
    Problema 4
  </h1>
  <div id="chart_div"></div>
  <script src="./script.js"></script>
</body>
</html>
```

Listing 10: PROBLEMA4-SCRIPT

```
const $container = document.getElementById('container');

const xhr = new XMLHttpRequest();

xhr.open('GET', 'http://localhost:8000/data', true);
xhr.onreadystatechange = function () {
  if(xhr.readyState === 4 && xhr.status === 200) {
    const json = JSON.parse(xhr.responseText);

    const arequipa = json.find((region) => region.region === 'Arequipa');

    google.charts.load('current', {packages: ['corechart', 'line']});
    google.charts.setOnLoadCallback(drawBasic);

    function drawBasic() {
      var data = new google.visualization.DataTable();
      data.addColumn('number', 'X');
      data.addColumn('number', 'Arequipa');

      data.addRow(arequipa.confirmed.map((confirmed, index) => [index,
        Number(confirmed.value)]));

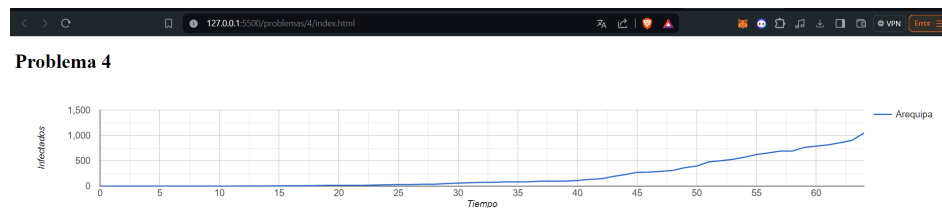
      var options = {
        hAxis: {
          title: 'Tiempo'
        },
        vAxis: {
          title: 'Infectados'
        }
      };
    };
  }
};
```



```
var chart = new
    google.visualization.LineChart(document.getElementById('chart_div'));

chart.draw(data, options);
}
}
}

xhr.send();
```



- Haga gráficos comparativos entre regiones usando líneas.

Listing 11: PROBLEMA5-INDEX

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Problema 5</title>
  <script type="text/javascript"
    src="https://www.gstatic.com/charts/loader.js"></script>
</head>
<body>
  <h1>Problema 5</h1>
  <div id="chart_div"></div>
  <script src="./script.js"></script>
</body>
</html>
```

Listing 12: PROBLEMA5-SCRIPT

```
const $container = document.getElementById('container');

const xhr = new XMLHttpRequest();

xhr.open('GET', 'http://localhost:8000/data', true);
xhr.onreadystatechange = function () {
  if(xhr.readyState === 4 && xhr.status === 200) {
    const json = JSON.parse(xhr.responseText);

    const arequipa = json.find((region) => region.region === 'Arequipa');
```

```

google.charts.load('current', {packages: ['corechart', 'line']});
google.charts.setOnLoadCallback(drawBasic);

function drawBasic() {
  var data = new google.visualization.DataTable();
  data.addColumn('number', 'X');

  json.forEach((region) => {
    data.addColumn('number', region.region);
  })

  const rows = [];
  for(let i = 0; i < arequipa.confirmed.length; i += 1) {
    const row = [i];
    json.forEach((region) => {
      row.push(Number(region.confirmed[i].value));
    })
    rows.push(row);
  }

  data.addRow(rows);

  var options = {
    hAxis: {
      title: 'Tiempo'
    },
    vAxis: {
      title: 'Infectados'
    }
  };

  var chart = new
    google.visualization.LineChart(document.getElementById('chart_div'));

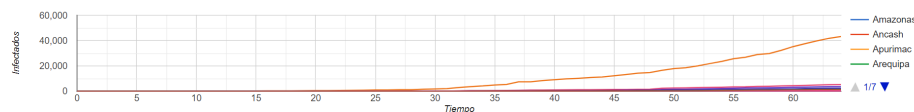
  chart.draw(data, options);
}
}

xhr.send();

```



Problema 5



- Visualice un gráfico comparativo del crecimiento en regiones excepto Lima y Callao

Listing 13: PROBLEMA6-INDEX

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Problema 6</title>
  <script type="text/javascript"
    src="https://www.gstatic.com/charts/loader.js"></script>
</head>
<body>
  <h1>Problema 6</h1>
  <div id="chart_div"></div>
  <script src="./script.js"></script>
</body>
</html>
```

Listing 14: PROBLEMA6-SCRIPT

```
const $container = document.getElementById('container');

const xhr = new XMLHttpRequest();

xhr.open('GET', 'http://localhost:8000/data', true);
xhr.onreadystatechange = function () {
  if(xhr.readyState === 4 && xhr.status === 200) {
    const json = JSON.parse(xhr.responseText);

    const regionesExcluidas = ['Lima', 'Callao'];
    const regionesFiltradas = json.filter(region =>
      !regionesExcluidas.includes(region.region));

    const datos = regionesFiltradas.map(region => {
      const confirmados = region.confirmed.map(entry => Number(entry.value));
      return { nombre: region.region, confirmados: confirmados };
    });

    google.charts.load('current', {packages: ['corechart', 'line']});
    google.charts.setOnLoadCallback(() => drawChart(datos));

    function drawChart(datos) {
      var data = new google.visualization.DataTable();
      data.addColumn('number', 'Das');

      datos.forEach(region => {
        data.addColumn('number', region.nombre);
      });

      const maxLength = datos.reduce((max, region) => Math.max(max,
        region.confirmados.length), 0);

      const rows = [];
      for (let i = 0; i < maxLength; i++) {
        const row = [i];
```

```

        datos.forEach(region => {
            row.push(region.confirmados[i] || null);
        });
        rows.push(row);
    }
    data.addRows(rows);

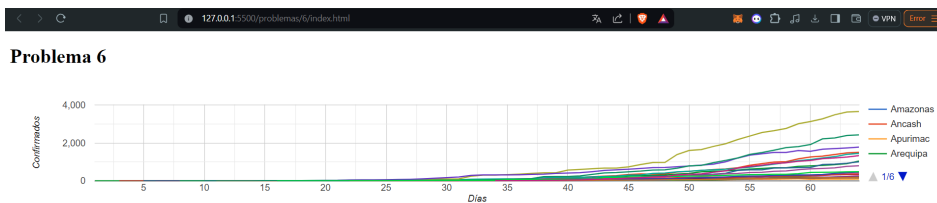
    var options = {
        hAxis: {
            title: 'Días',
        },
        vAxis: {
            title: 'Confirmados'
        }
    };

    var chart = new
        google.visualization.LineChart(document.getElementById('chart_div'));

    chart.draw(data, options);
}
}
}

xhr.send();

```



- Haga gráficos comparativos entre regiones elegidas por el usuario.

Listing 15: PROBLEMA7-INDEX

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Comparacin de Regiones</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript"
        src="https://www.gstatic.com/charts/loader.js"></script>
</head>
<body>
    <h1>Problema 7: Comparacin de Regiones</h1>
    <div>

```

```
<label for="region1">Seleccione la primera regin:</label>
<select id="region1">

</select>
</div>
<div>
  <label for="region2">Seleccione la segunda regin:</label>
  <select id="region2">

  </select>
</div>
<div id="chart_div" style="width: 100%; height: 500px;"></div>
<script src="./script.js"></script>
</body>
</html>
```

Listing 16: PROBLEMA7-SCRIPT

```
$(document).ready(function(){
  const xhr = new XMLHttpRequest();

  xhr.open('GET', 'http://localhost:8000/data', true);
  xhr.onreadystatechange = function () {
    if(xhr.readyState === 4 && xhr.status === 200) {
      const json = JSON.parse(xhr.responseText);
      const regiones = json.map(region => region.region);

      const $region1Select = $('#region1');
      const $region2Select = $('#region2');

      regiones.forEach(region => {
        $region1Select.append('<option value="'+region+'">'+region+'</option>');
        $region2Select.append('<option value="'+region+'">'+region+'</option>');
      });

      $region1Select.add($region2Select).change(function() {
        const region1 = $region1Select.val();
        const region2 = $region2Select.val();
        drawChart(json, region1, region2);
      }).change();
    }
  }
}

xhr.send();

function drawChart(json, region1, region2) {
  const lima = json.find(region => region.region === region1);
  const callao = json.find(region => region.region === region2);

  google.charts.load('current', {packages: ['corechart', 'line']});
  google.charts.setOnLoadCallback(function() {
    var data = new google.visualization.DataTable();
    data.addColumn('number', 'Das');
    data.addColumn('number', region1);
    data.addColumn('number', region2);
```

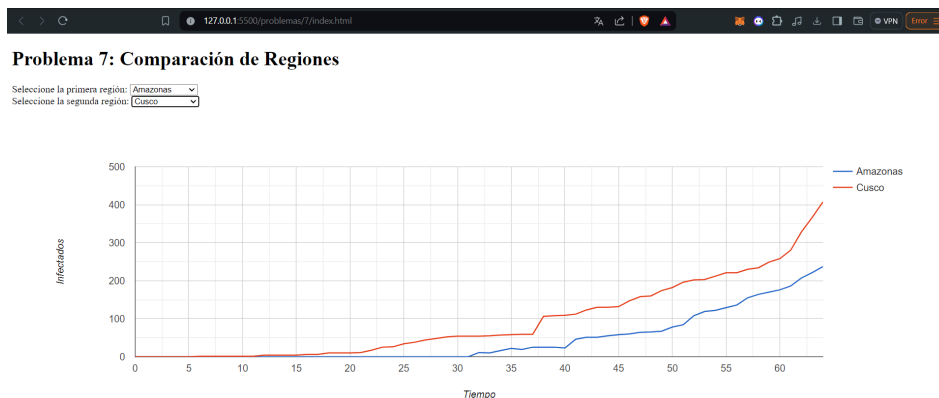
```
const rows = [];
const maxLength = Math.max(lima.confirmed.length, callao.confirmed.length);
for(let i = 0; i < maxLength; i++) {
    const row = [i];
    row.push(Number(lima.confirmed[i]?value || 0));
    row.push(Number(callao.confirmed[i]?value || 0));
    rows.push(row);
}

data.addRow(rows);

var options = {
    hAxis: {
        title: 'Tiempo'
    },
    vAxis: {
        title: 'Infectados'
    }
};

var chart = new
    google.visualization.LineChart(document.getElementById('chart_div'));

chart.draw(data, options);
});
}
});
```



- Visualice un gráfico comparativo del crecimiento en regiones excepto Lima y Callao, mostrando el número de confirmados por cada día.

Listing 17: PROBLEMA8-INDEX

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Problema 8</title>
<script type="text/javascript"
  src="https://www.gstatic.com/charts/loader.js"></script>
</head>
<body>
  <h1>Problema 8</h1>
  <div id="chart_div"> </div>
  <script src="./script.js"></script>
</body>
</html>
```

Listing 18: PROBLEMA8-SCRIPT

```
const $container = document.getElementById('container');

const xhr = new XMLHttpRequest();

xhr.open('GET', 'http://localhost:8000/data', true);
xhr.onreadystatechange = function () {
  if(xhr.readyState === 4 && xhr.status === 200) {
    const json = JSON.parse(xhr.responseText);

    const noLimaCallao = json.filter((region) => region.region !== 'Lima' &&
      region.region !== 'Callao');

    google.charts.load('current', {packages: ['corechart', 'line']});
    google.charts.setOnLoadCallback(drawBasic);

    function drawBasic() {
      var data = new google.visualization.DataTable();
      data.addColumn('number', 'X');

      noLimaCallao.forEach((region) => {
        data.addColumn('number', region.region);
      })

      const rows = [];
      for(let i = 0; i < noLimaCallao[0].confirmed.length; i += 1) {
        const row = [i];
        noLimaCallao.forEach((region) => {
          row.push(Number(region.confirmed[i].value));
        })
        rows.push(row);
      }

      data.addRow(rows);

      var options = {
        hAxis: {
          title: 'Tiempo'
        },
        vAxis: {
          title: 'Infectados'
        }
      };
    }
  }
};
```

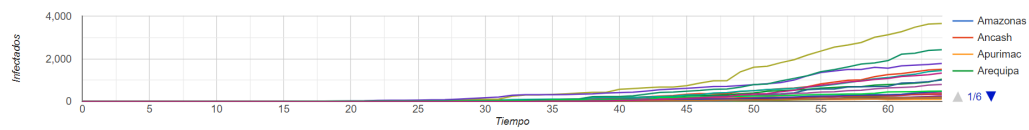
```
var chart = new
    google.visualization.LineChart(document.getElementById('chart_div'));

    chart.draw(data, options);
}
}
}

xhr.send();
```



Problema 8



5. Rúbricas

5.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

Puntos	Nivel			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
Total		20		17	