Informe de Laboratorio 04

Tema: Ajax

Nota

Estudiante	Escuela	Asignatura
Victor Gonzalo Maldonado Vilca vmaldonadov@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 1702122

Tarea	Tema	Duración
04	Ajax	2 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 9 de abril de 2024	Al 18 de mayo de 2024

1. Tarea

Ejercicios Propuestos:

- Revisar Ajax en w3schools y Presentar un informe con un pantallazo por cada ejemplo de Ajax realizado en su propio servidor web.
- Usar un servidor web como apache en su computadora (Xampp), descargar el archivo de datos data.json y dejarlo en el mismo lugar donde lanzó el servidor. **Para cada uno de estos problemas implemente un programa en ajax y página que:**
  - Liste todas las “regiones”.
  - Muestre el número total de confirmados por región.
  - Encuentre las 10 regiones cuya suma total sea la mayor.
  - Visualice un gráfico en el tiempo de los valores para la región de Arequipa.
  - Haga gráficos comparativos entre regiones usando líneas.
  - Visualice un gráfico comparativo del crecimiento en regiones excepto Lima y Callao.
  - Haga gráficos comparativos entre regiones elegidas por el usuario.
  - Visualice un gráfico comparativo del crecimiento en regiones excepto Lima y Callao, mostrando el número de confirmados por cada día.
- En grupos de 3 a 5 personas (los grupos se definirán en clase) implemente una aplicación web que navegue sobre archivos Markdown y permita:
  - Listar los archivos Markdown disponibles.

- Ver el contenido de un archivo markdown traducido a HTML.
- Crear nuevos archivos MarkDown y almacenarlos en el servidor

Tener en cuenta:

- La comunicación entre el cliente y el servidor tiene que ser usando JSON sólamente.
- El cliente debe usar AJAX para sus peticiones.
- El servidor debe usar NodeJS.
- Su aplicación debe ser de página única, es decir que sólo habrá un archivo index.html y nada más.

2. Entregables

- Informe de trabajo
- URL: Repositorio de GitHub

3. Equipos, materiales y temas utilizados

- HTML
- CSS
- JavaScript
- w3schools
- Xampp
- Node.js
- Ajax
- GitHub

4. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/Victor-Gonzalo-Maldonado-Vilca/TareaAjax.git`

5. Link de Video

- URL video en Youtube.
- `https://www.youtube.com/watch?v=tPlgxQkUKaY`

6. Desarrollo del trabajo

6.1. AJAX w3schools - EXAMPLES

Capturas de pantalla de los ejemplos de Ajax, usando su propio servidor.

- **Tutorial JavaScript sección Ajax**

Figura 1: w3schools

6.1.1. AJAX - Intro

- **AJAX Example**

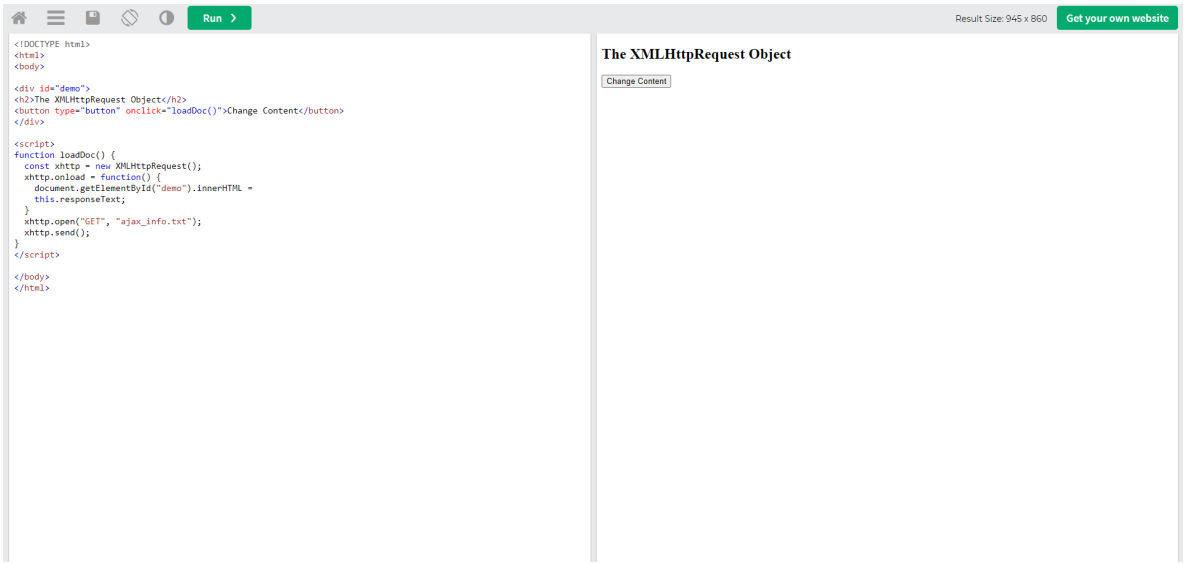

Figura 2: AJAX

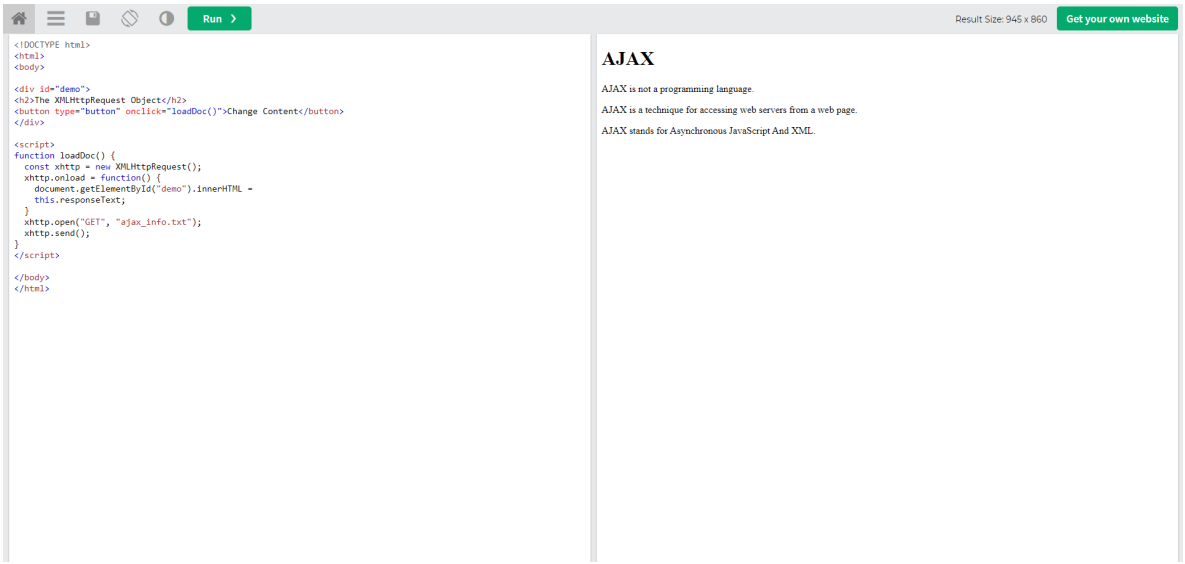

Figura 3: Presionando el botón

6.1.2. AJAX - XMLHttp

- **Ejemplo 1 - Enviar una Solicitud:**

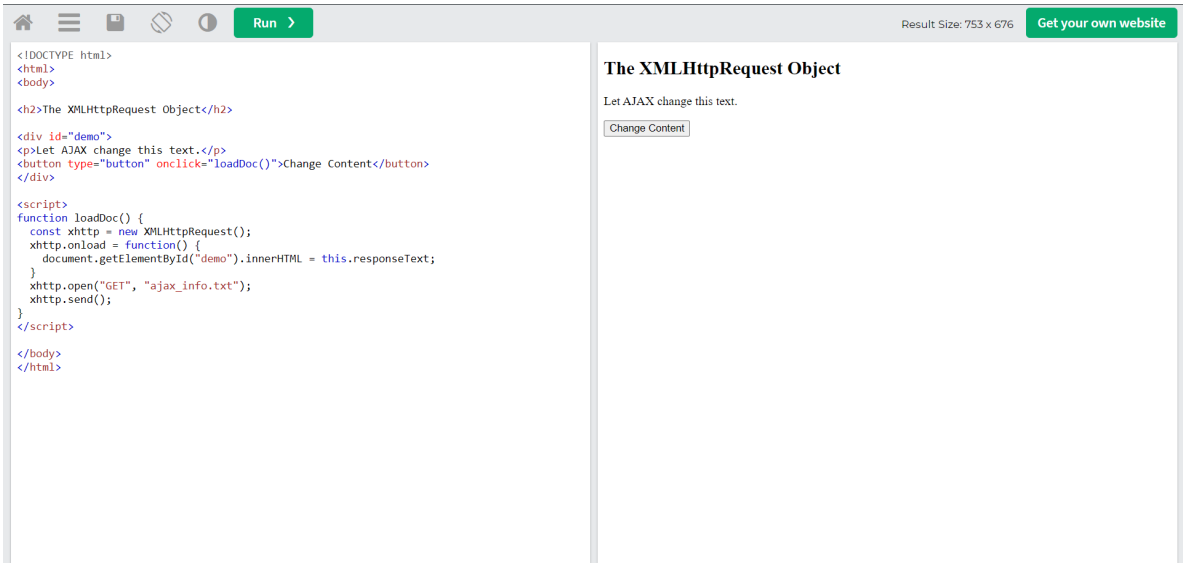

Figura 4: Enviar Solicitud

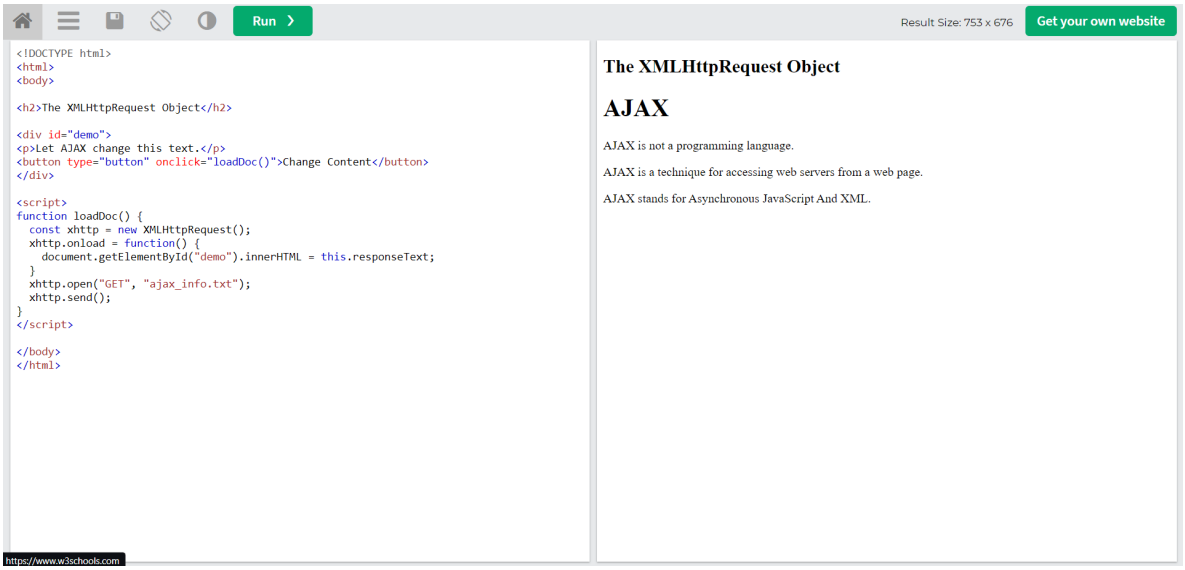

Figura 5: Presionando el botón

- **Ejemplo 2 - Propiedad de carga:**

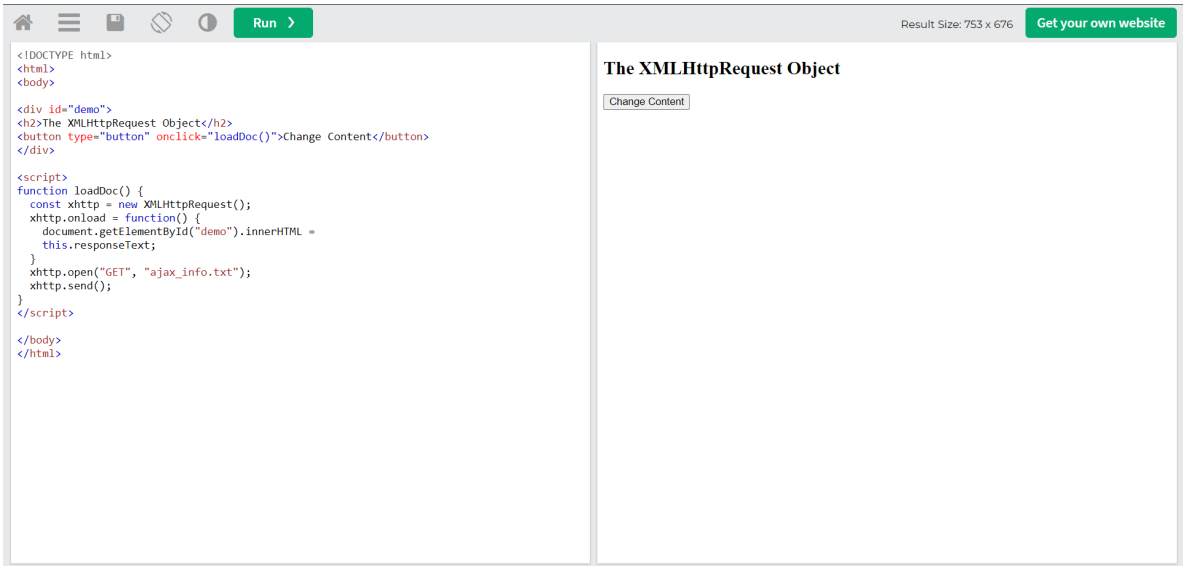

Figura 6: Propiedad de Carga

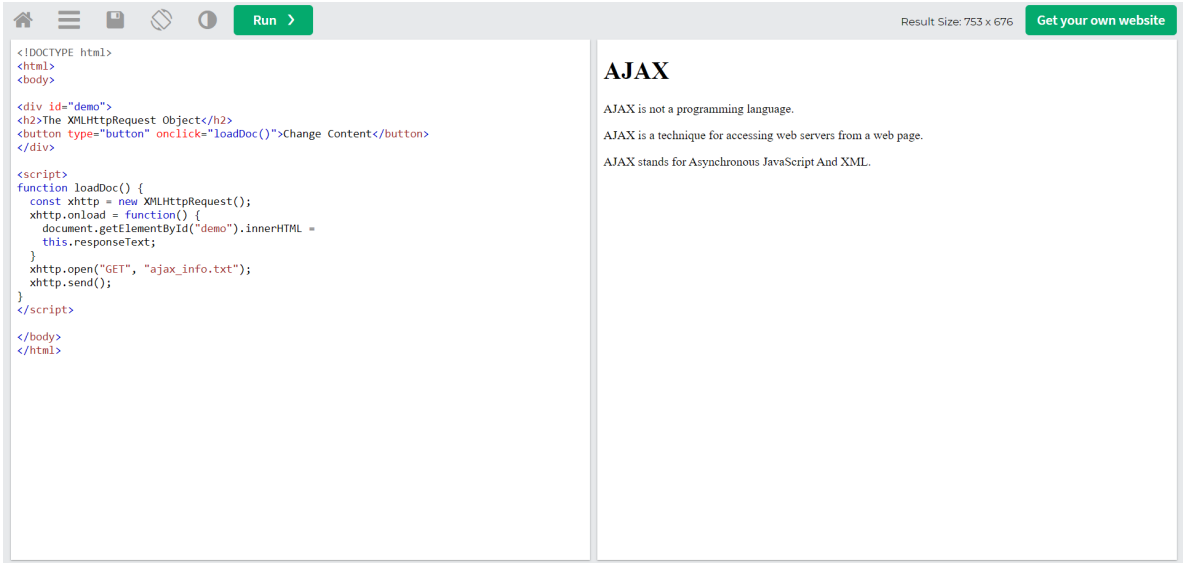

Figura 7: Presionando el botón

- **Ejemplo 3 - Propiedad Onreadystatechange:**

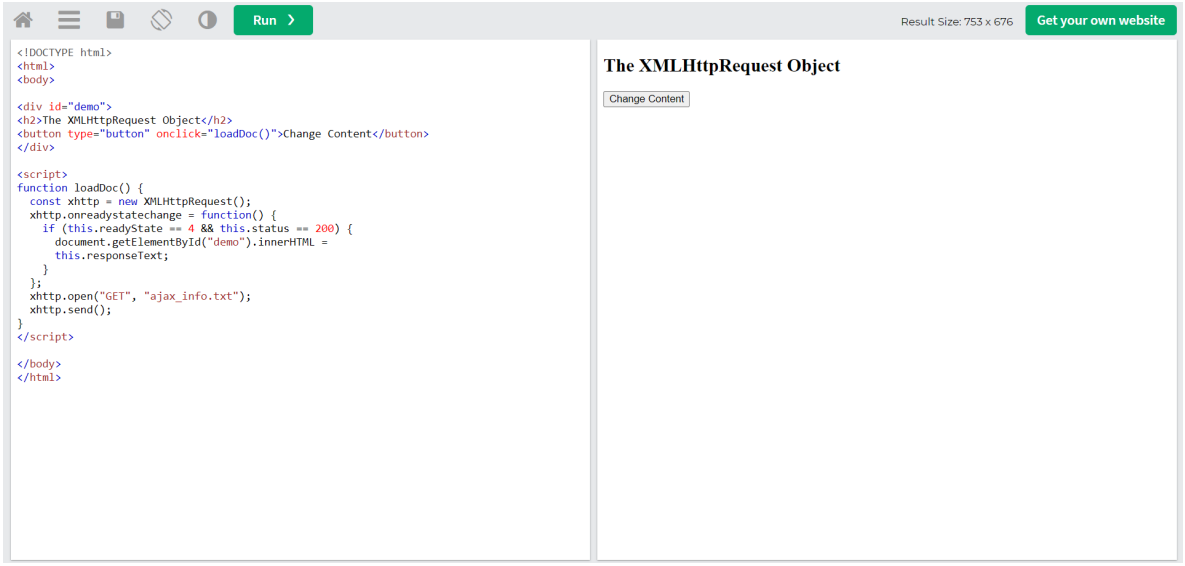

Figura 8: Propiedad Onreadystatechange

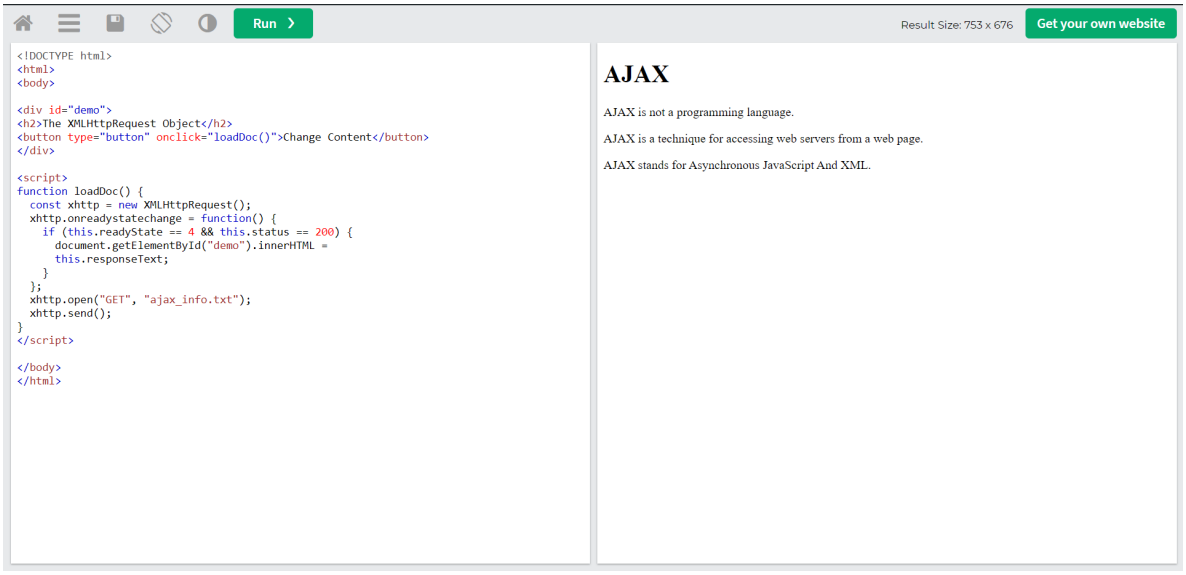

Figura 9: Presionando el botón

6.1.3. AJAX - Request

- **Get Requests:**

1. Primer Ejemplo:

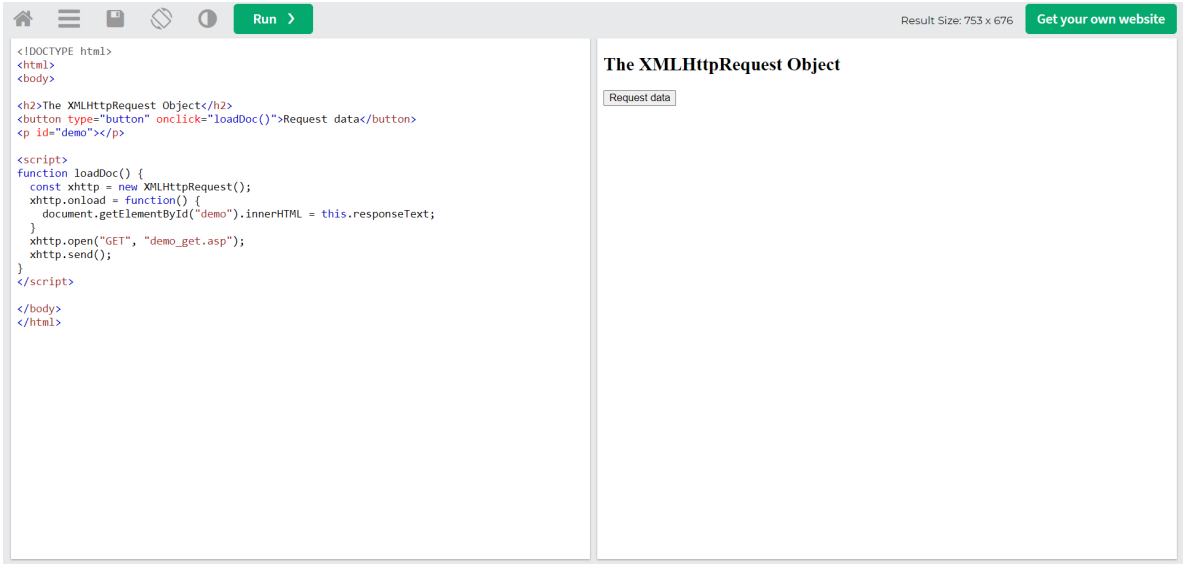

Figura 10: Ejemplo 1 - GET

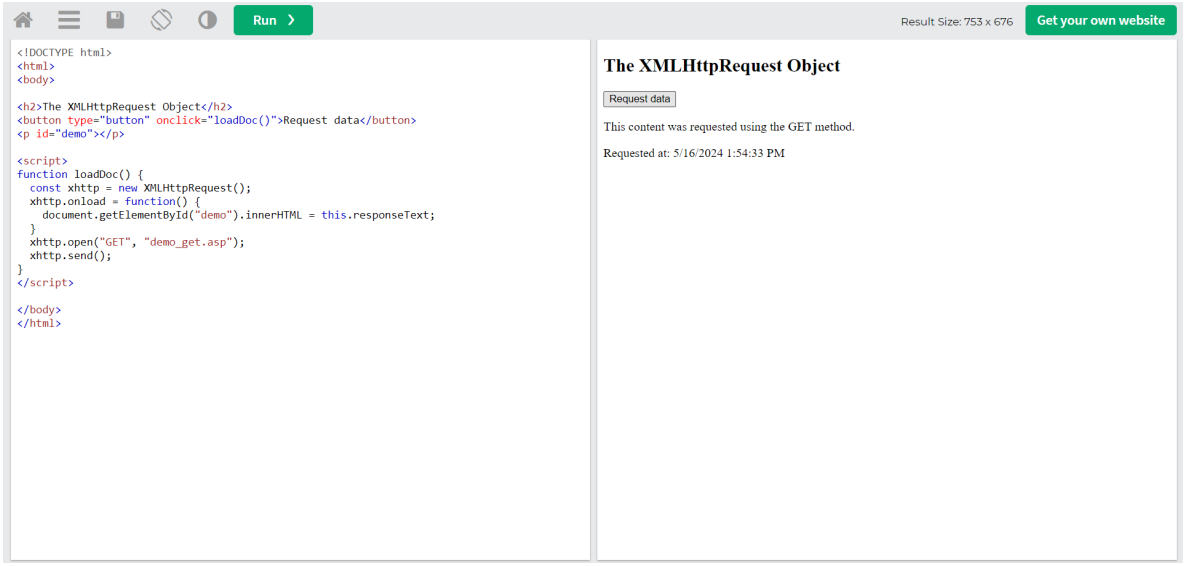

Figura 11: Presionando botón

2. Segundo Ejemplo:

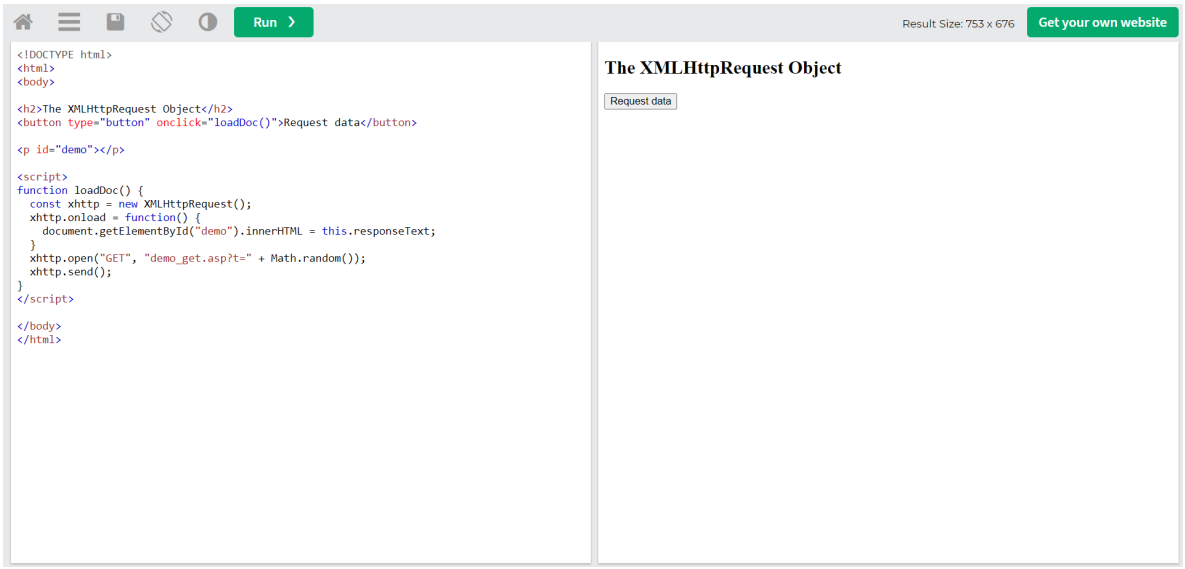

Figura 12: Ejemplo 2 - GET

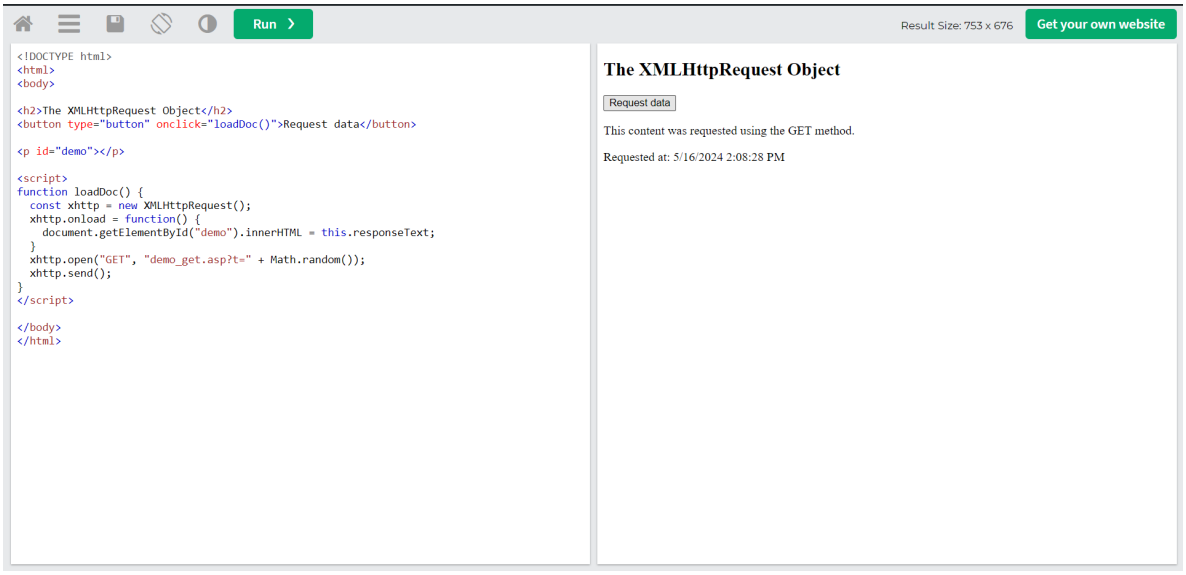

Figura 13: Presionando el botón

3. Tercer Ejemplo:

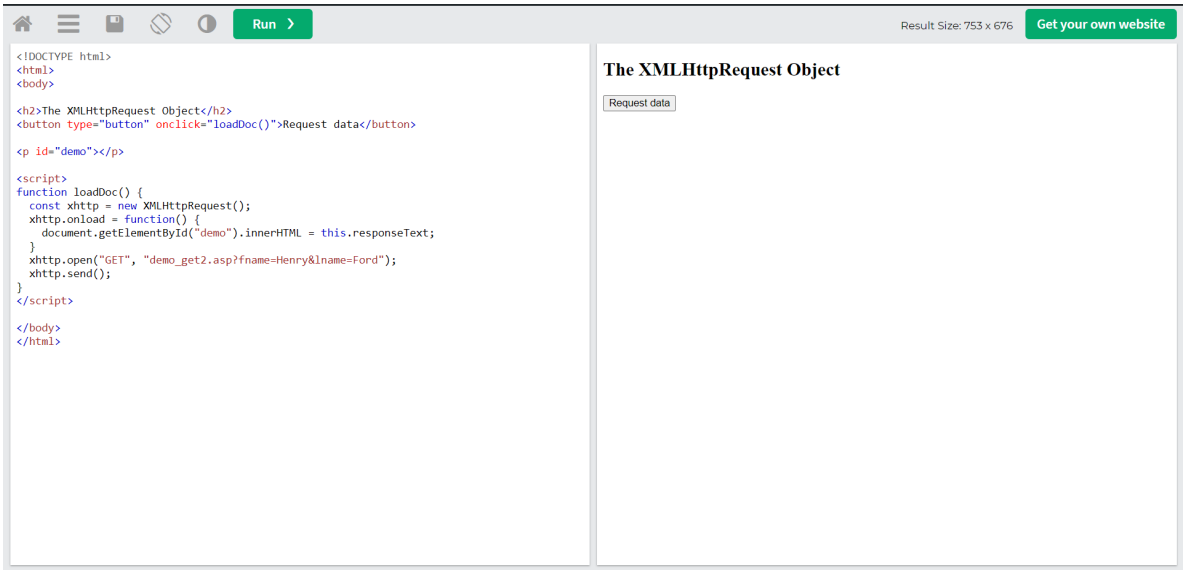

Figura 14: Ejemplo 3 - GET

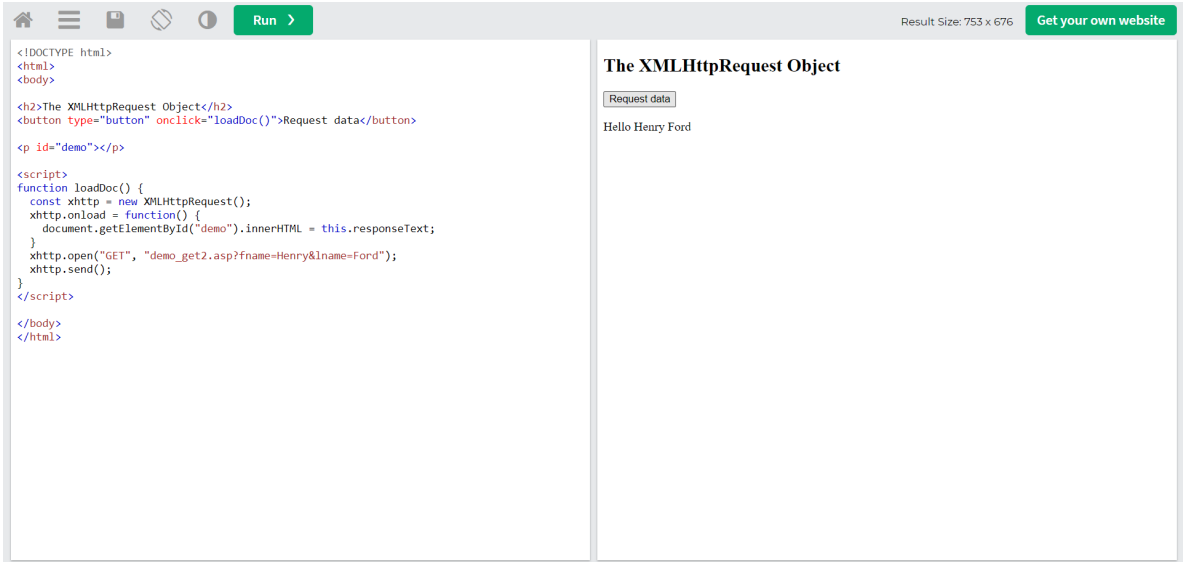

Figura 15: Presionando el botón

- **Post Request:**

1. Primer Ejemplo:

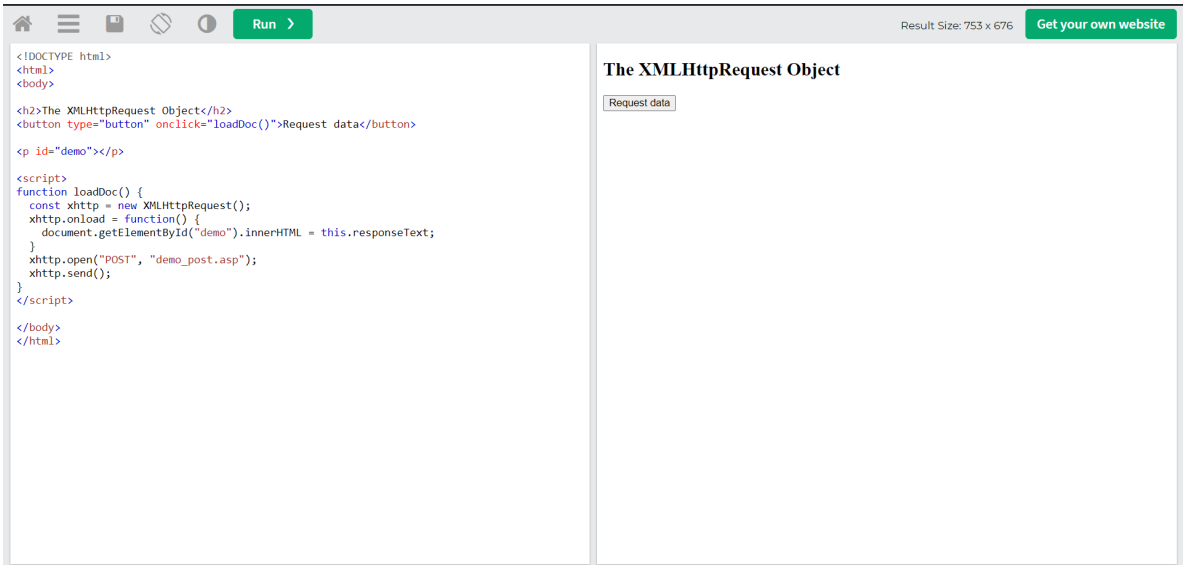

Figura 16: Ejemplo 1 - POST

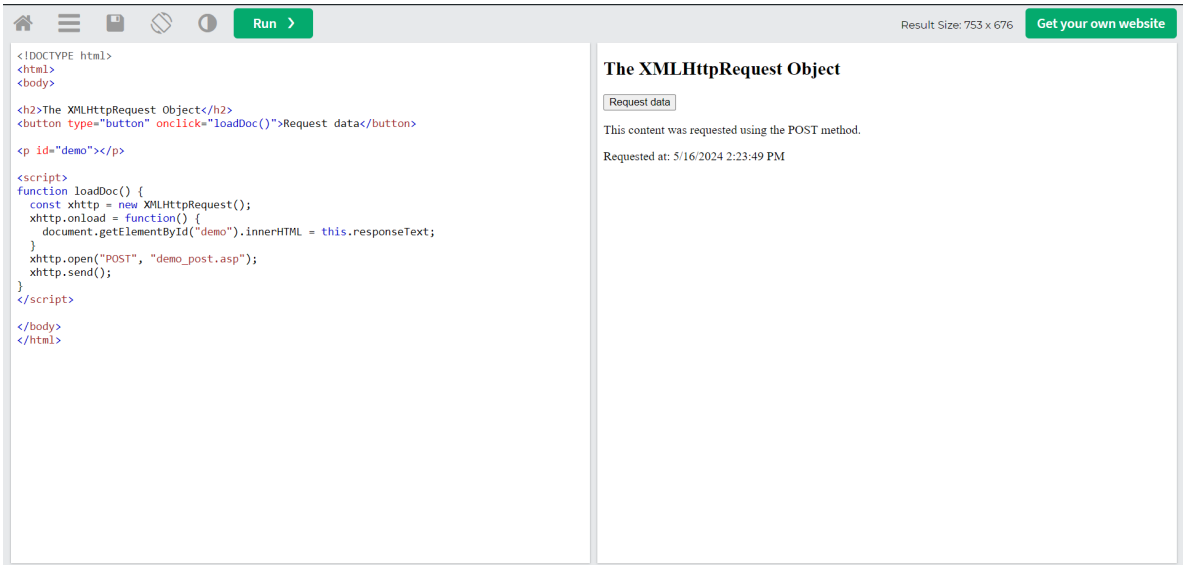

Figura 17: Presionando el botón

2. Segundo Ejemplo:

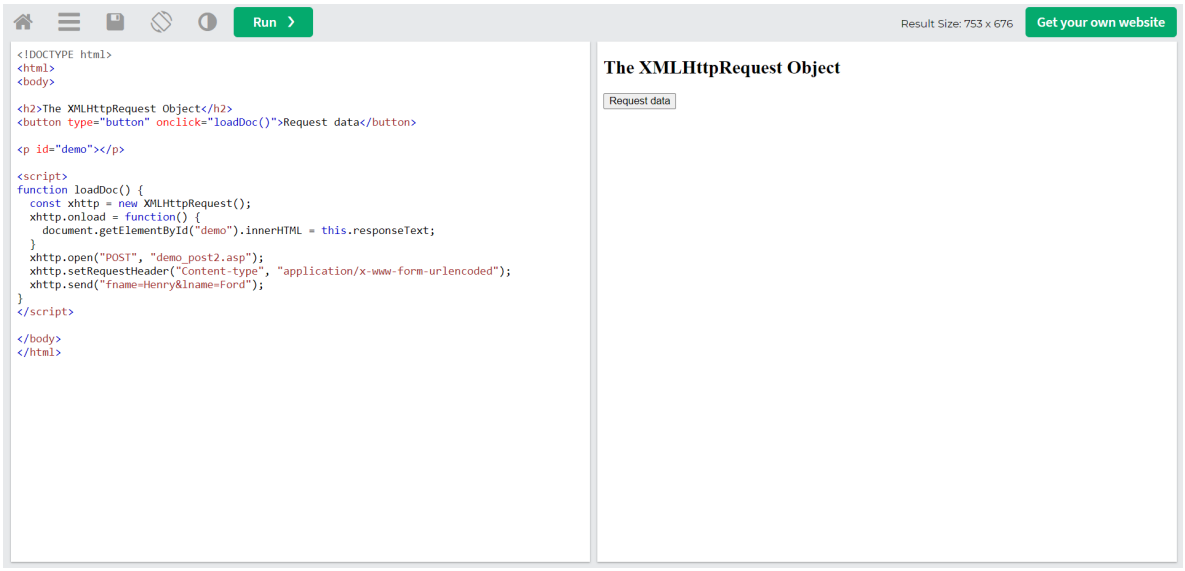

Figura 18: Ejemplo 2 - POST

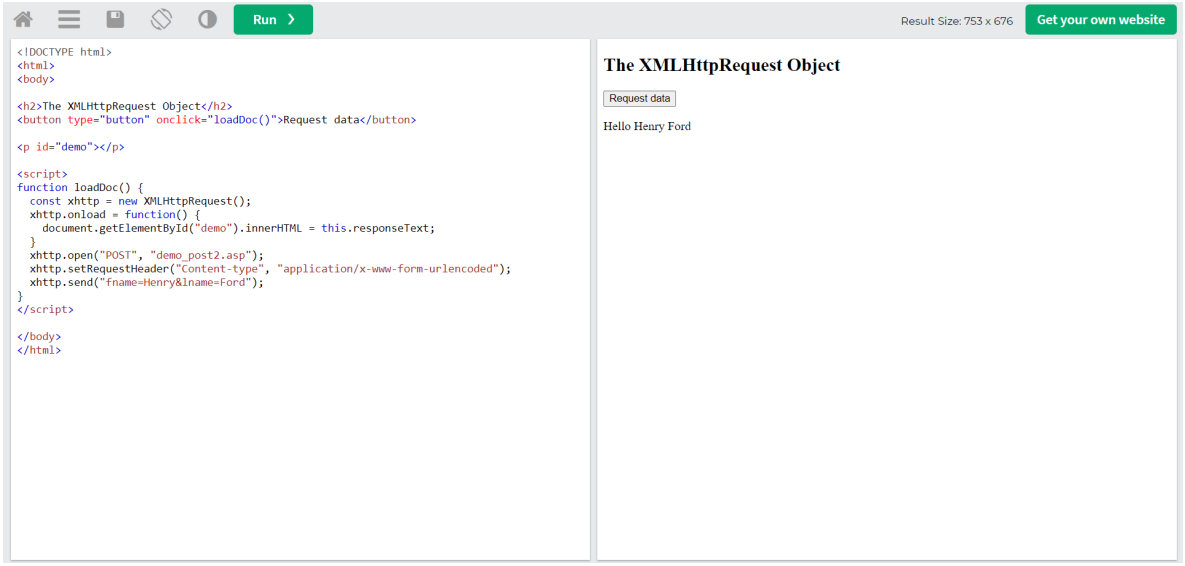

Figura 19: Presionando el botón

- **Synchronous Request**

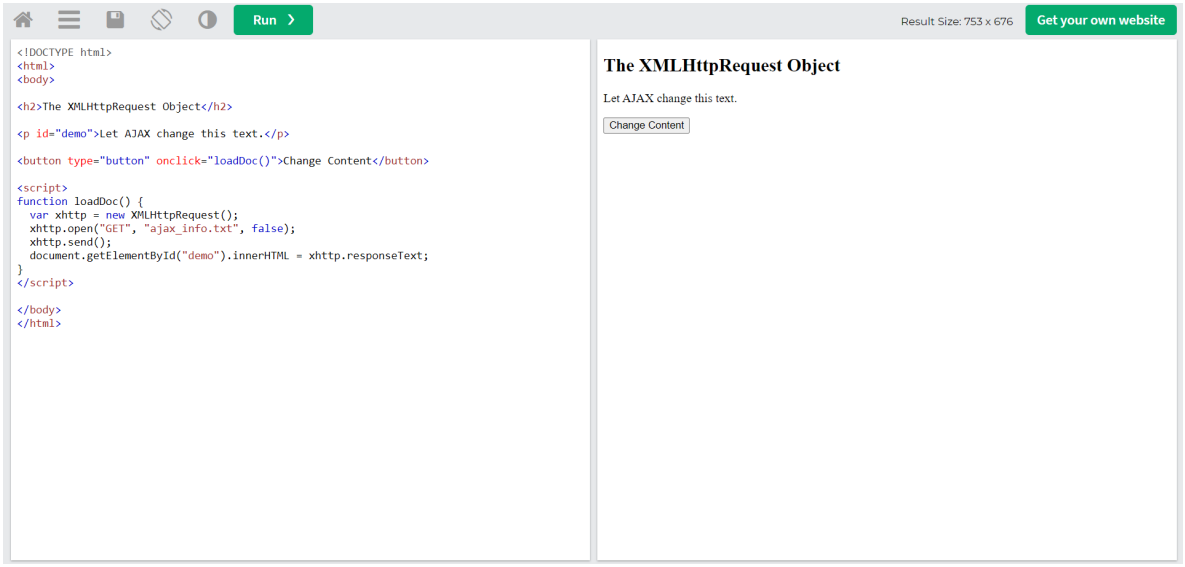

Figura 20: Ejemplo - Synchronous Request

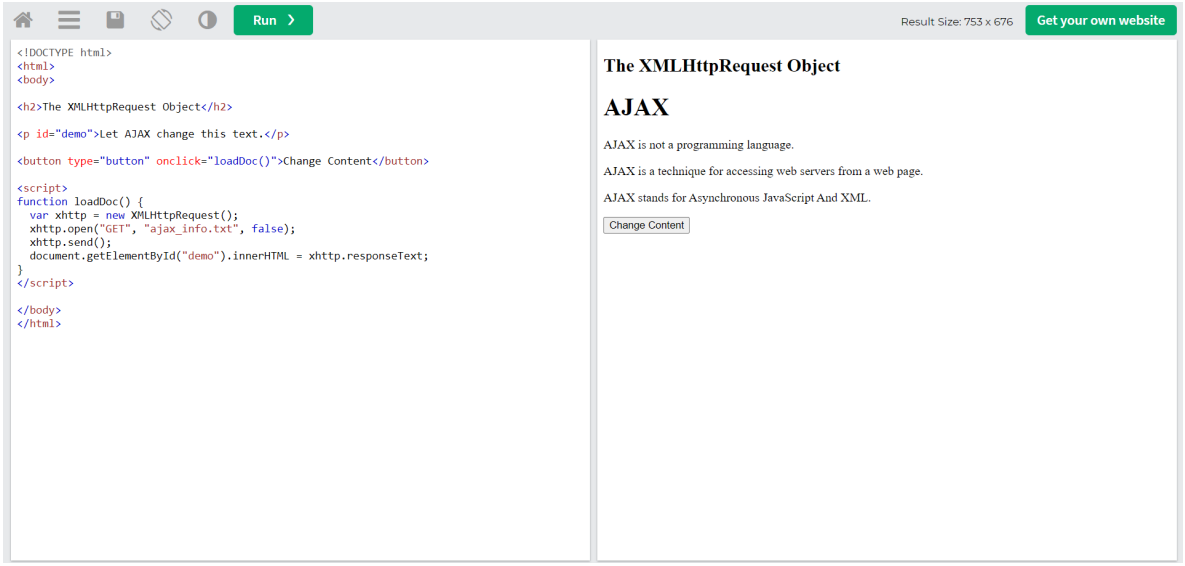

Figura 21: Presionando el botón

6.1.4. AJAX - Response

- **Propiedad de texto de respuesta**

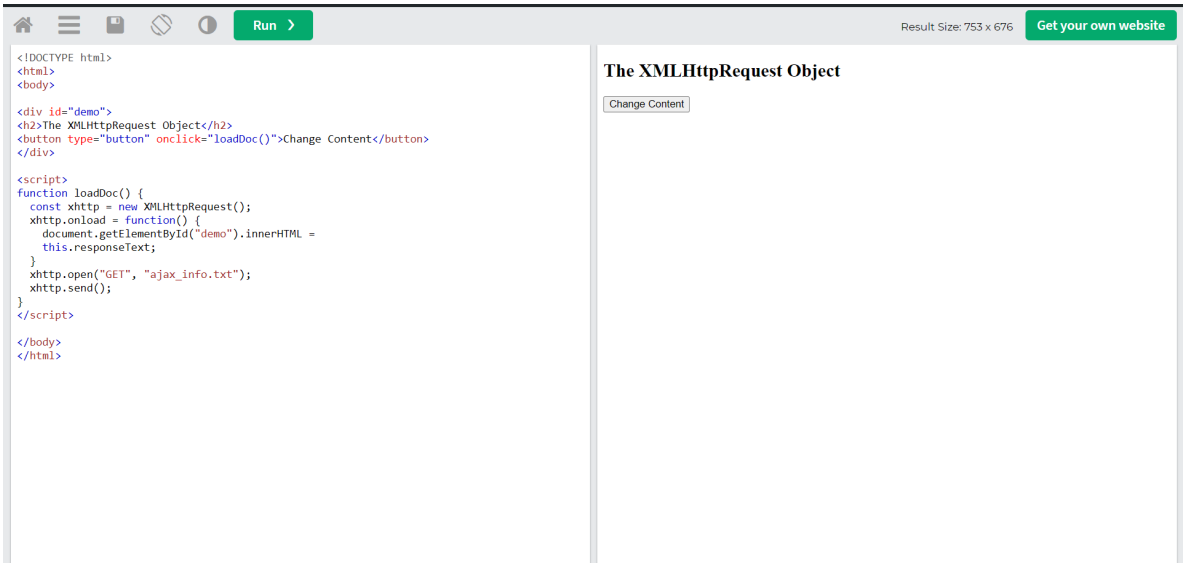

Figura 22: Propiedad de Texto de Respuesta

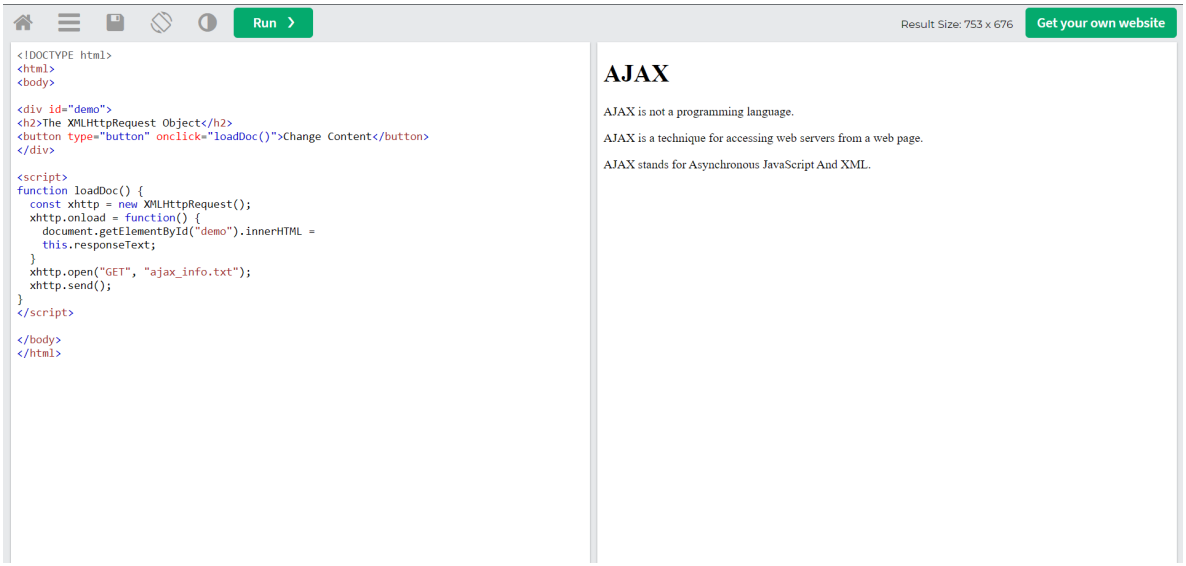

Figura 23: Presionando el botón

- **La propiedad ResponseXML**

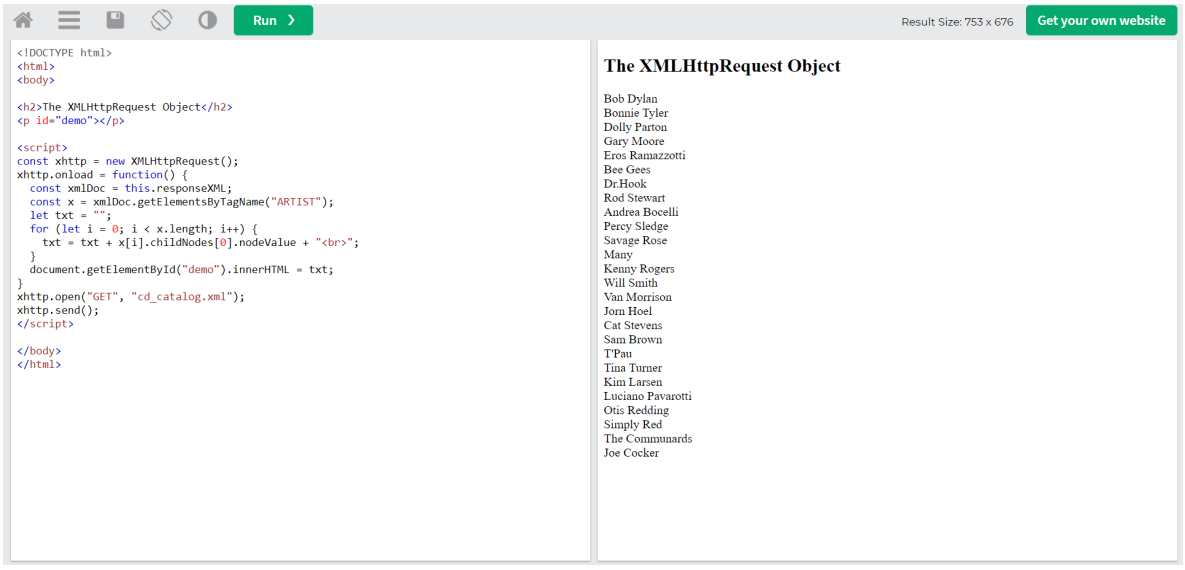

Figura 24: Propiedad de Texto de Respuesta

- **Método getAllResponseHeaders()**

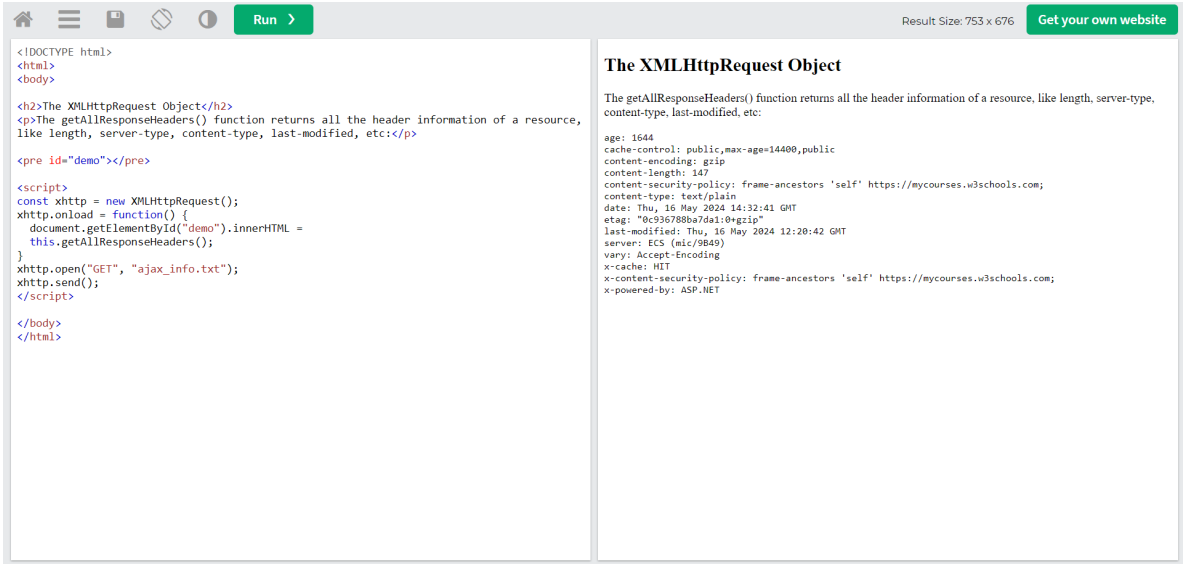

Figura 25: Método getAllResponseHeaders

- **Método getResponseHeader()**

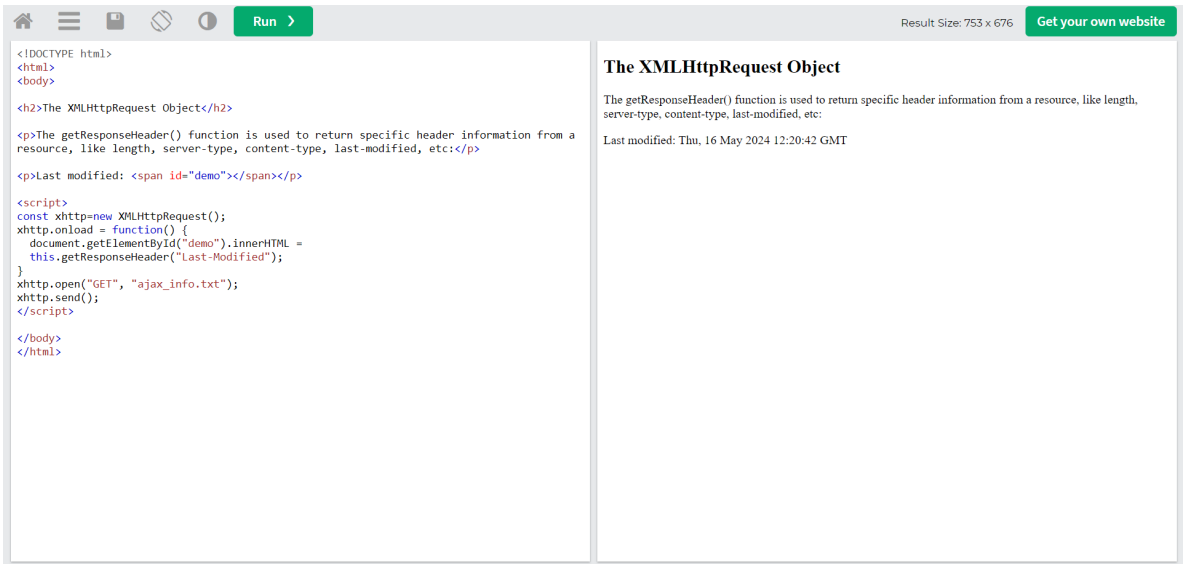

Figura 26: Método getResponseHeader

6.1.5. AJAX XML - File

- **AJAX XML EXAMPLE**

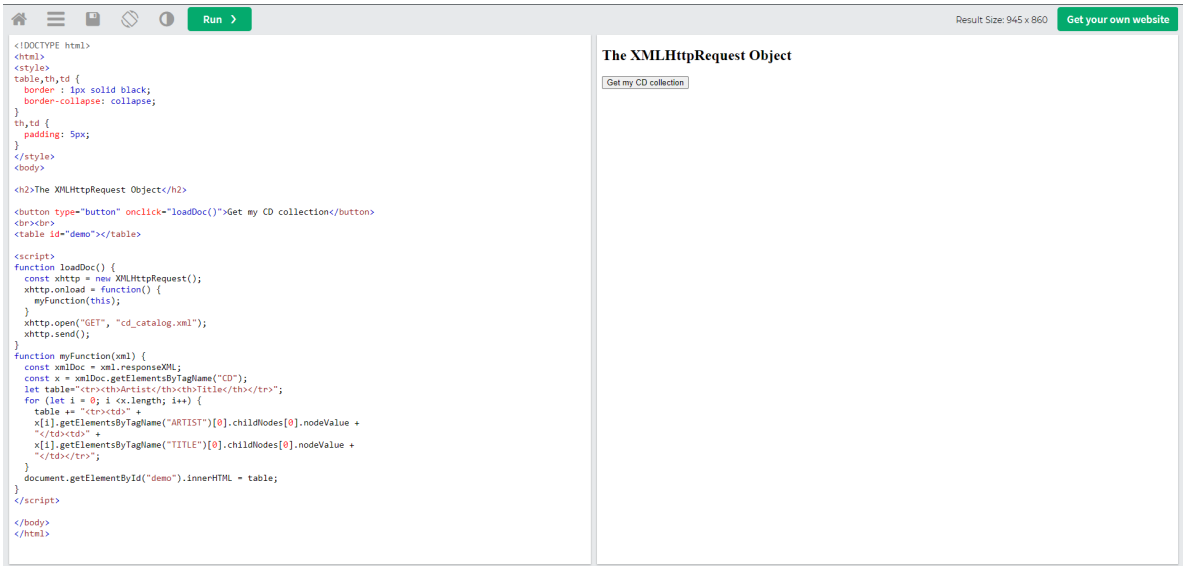

Figura 27: Ajax XML

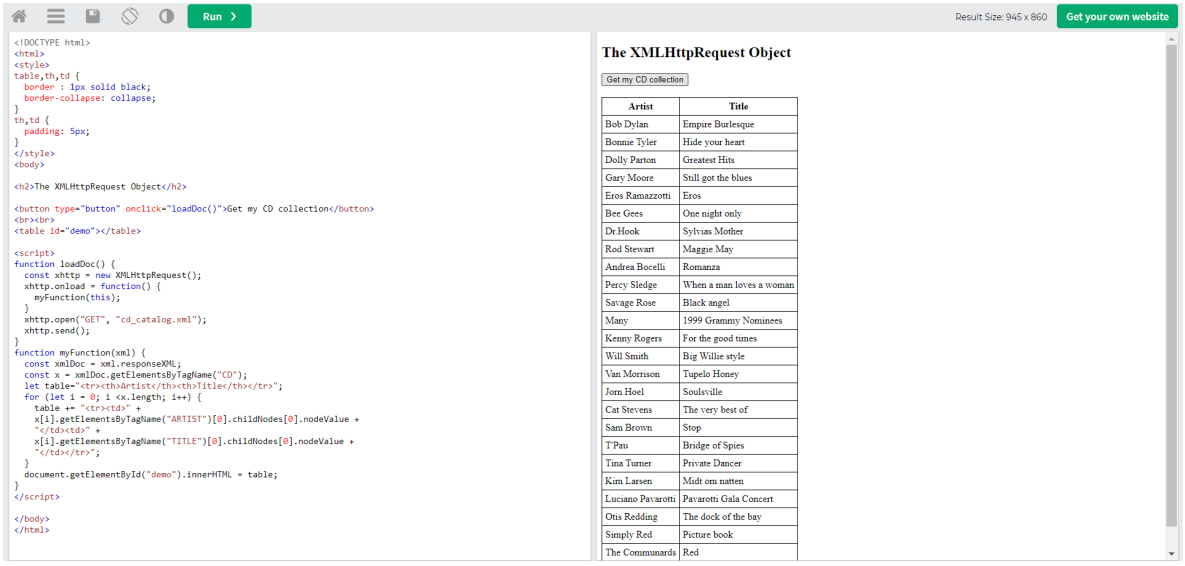

Figura 28: Presionando el botón

6.1.6. AJAX PHP

- **AJAX PHP EXAMPLE**

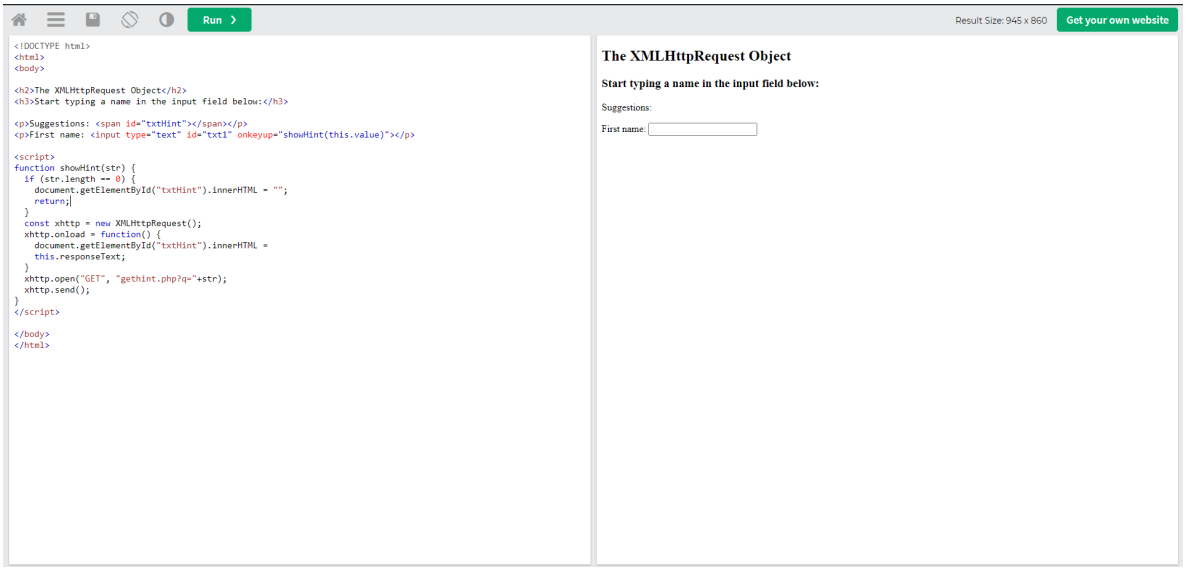

Figura 29: Ajax PHP

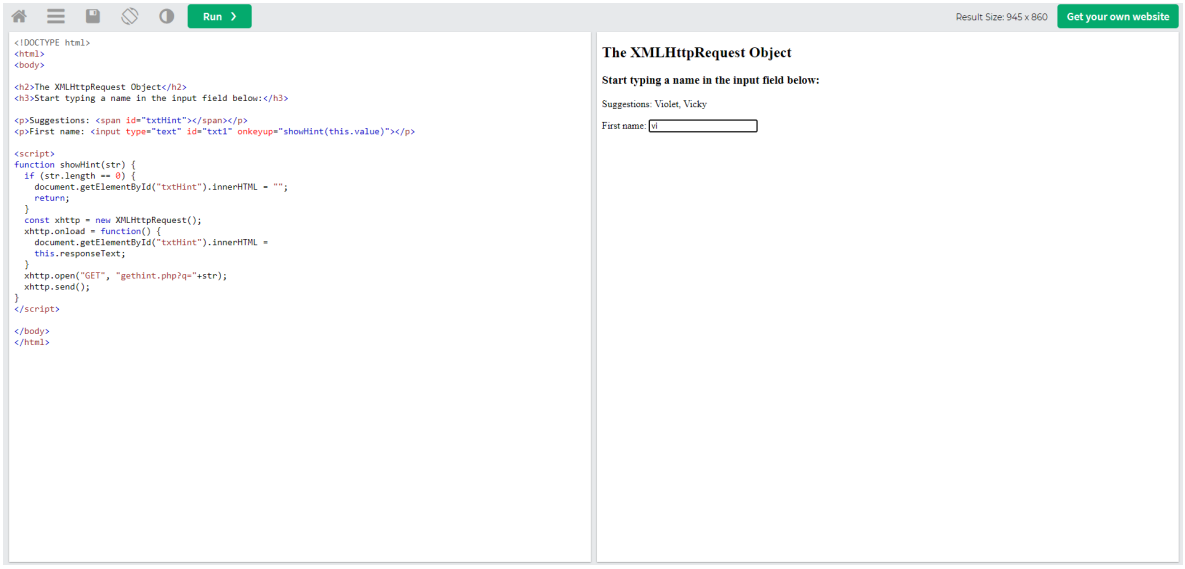

Figura 30: Escribiendo en el cuadro de texto

6.1.7. AJAX ASP

- **AJAX ASP EXAMPLE**

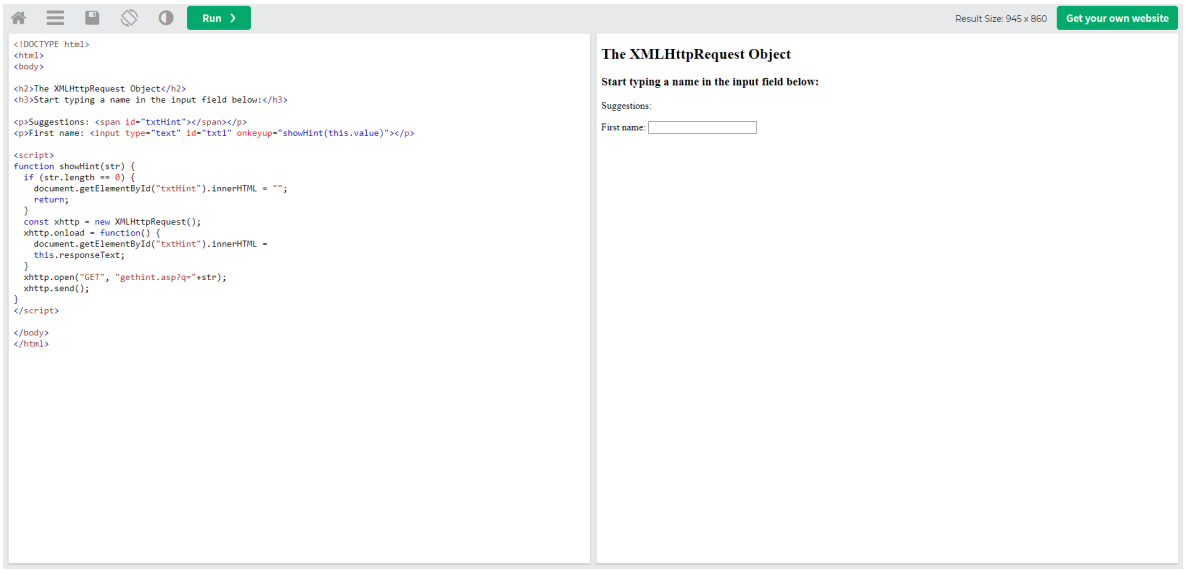

Figura 31: Ajax ASP Example

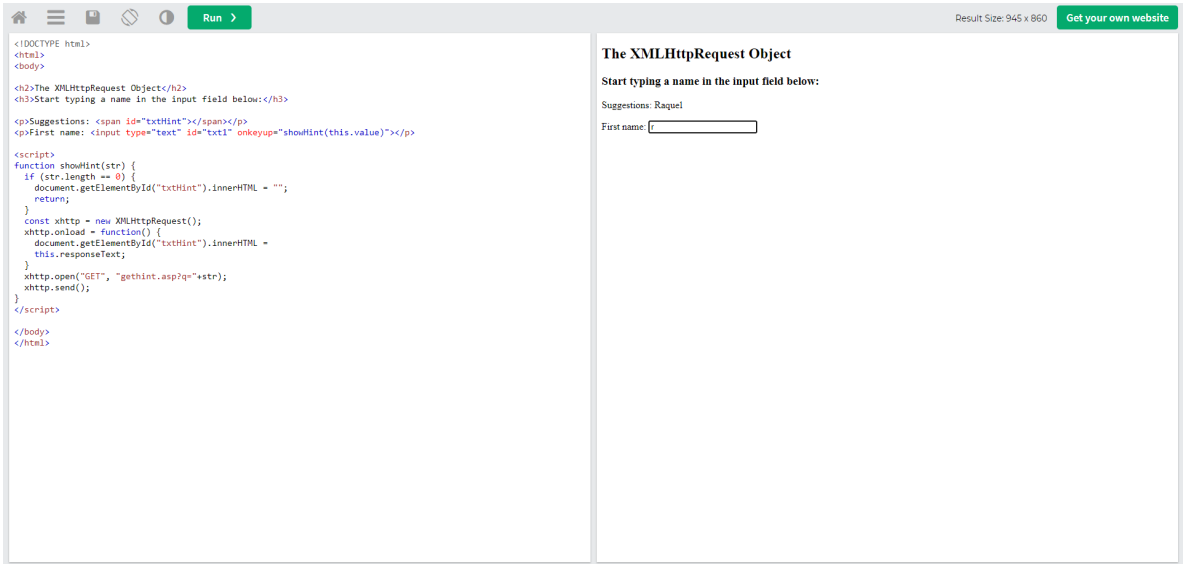

Figura 32: Escribiendo en el cuadro de texto

6.1.8. AJAX DATABASE

- **AJAX ASP EXAMPLE**

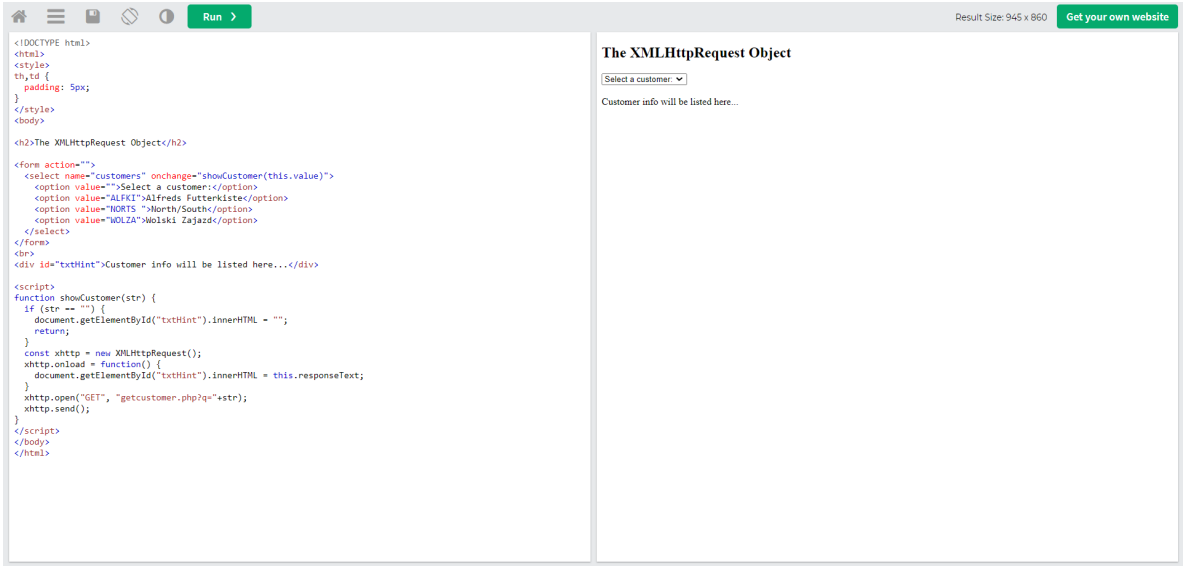

Figura 33: Ajax Database Example

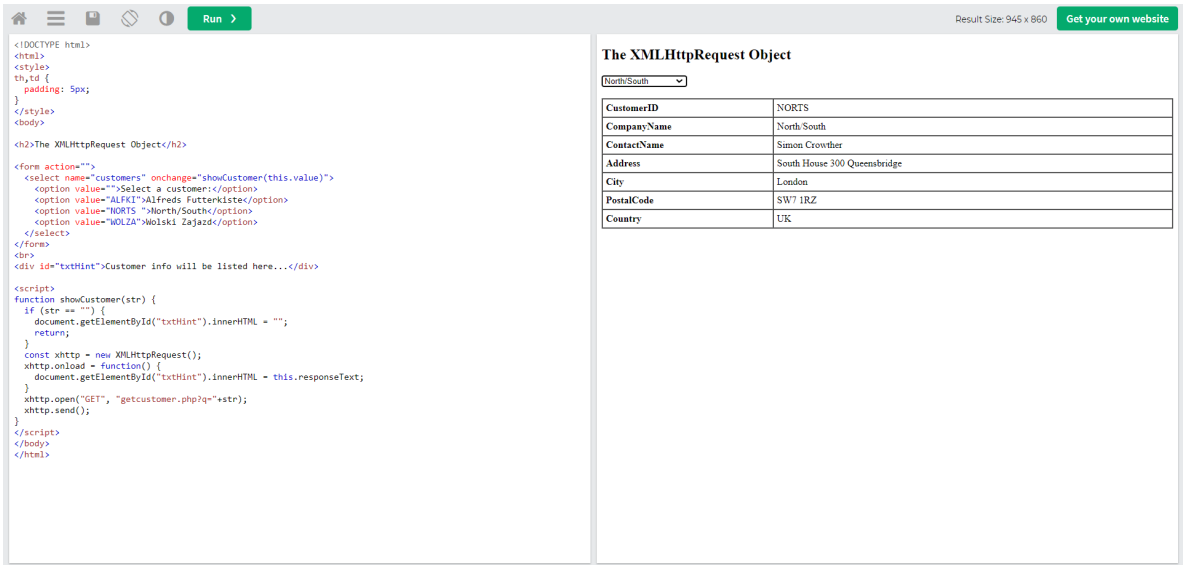

Figura 34: Eligiendo una opción

6.1.9. AJAX - Applications

- **Mostrar datos XML en una tabla HTML**

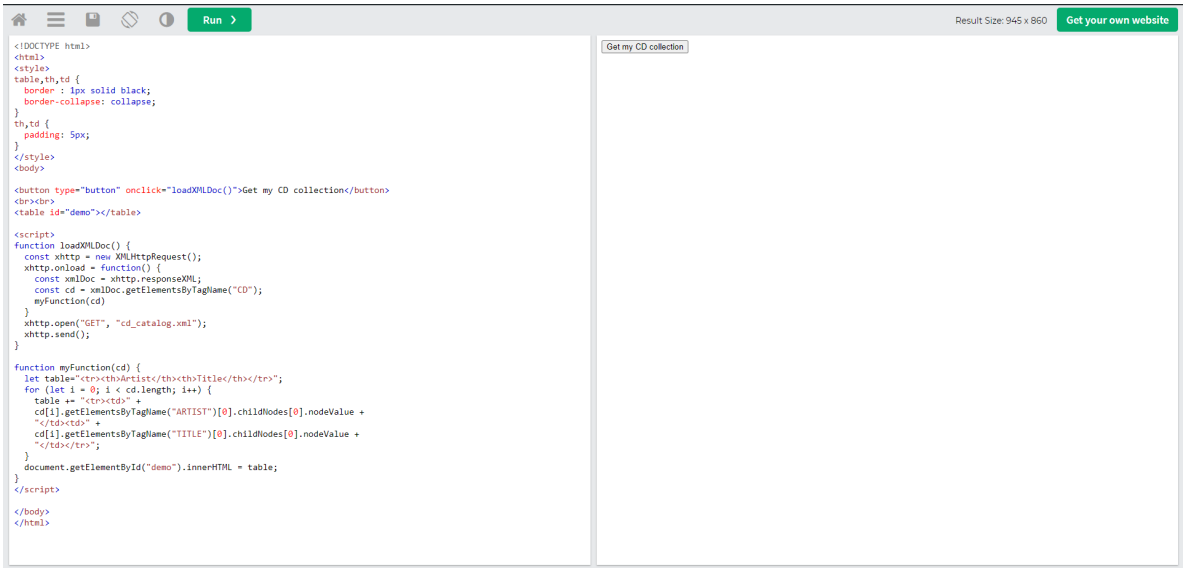

Figura 35: Estructura

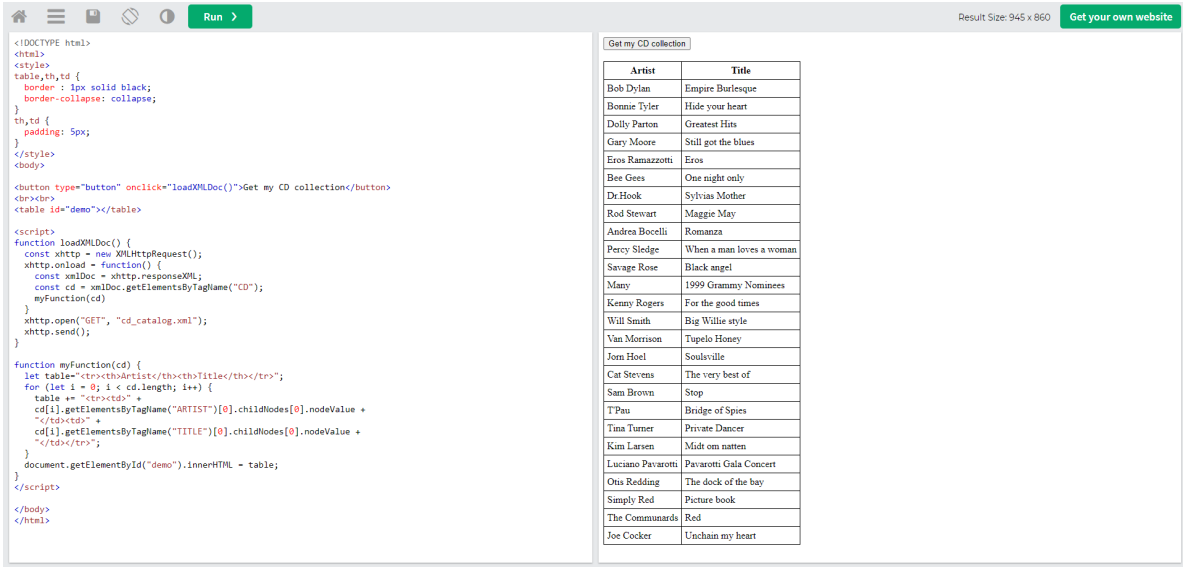

Figura 36: Mostrar Datos

- **Mostrar el primer CD en un elemento div HTML**

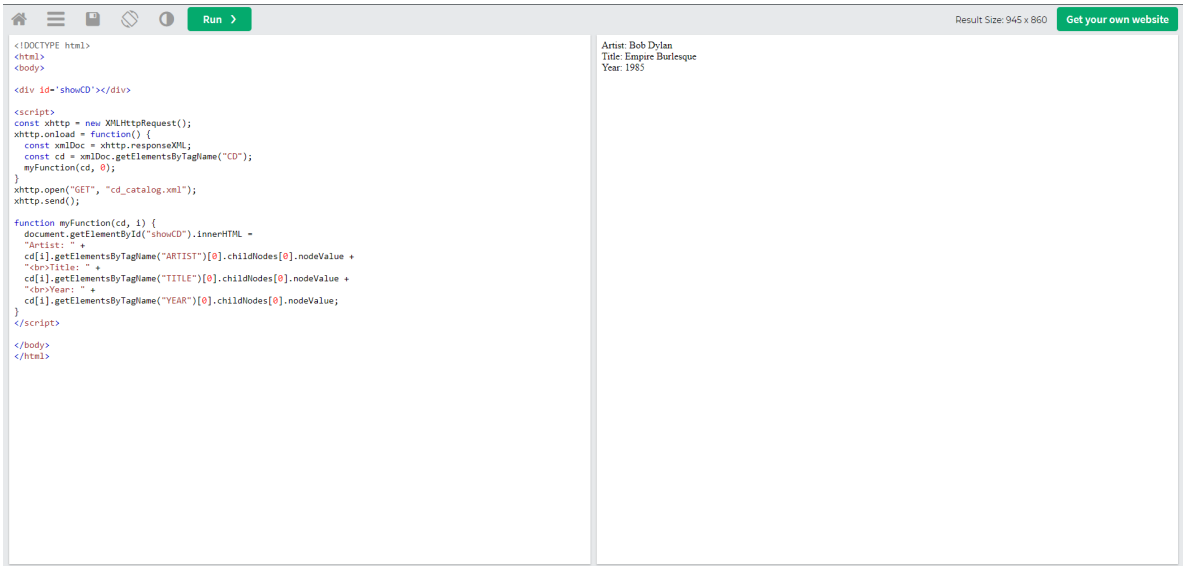

Figura 37: Mostrar primer elemento

- **Navegar entre los CD**

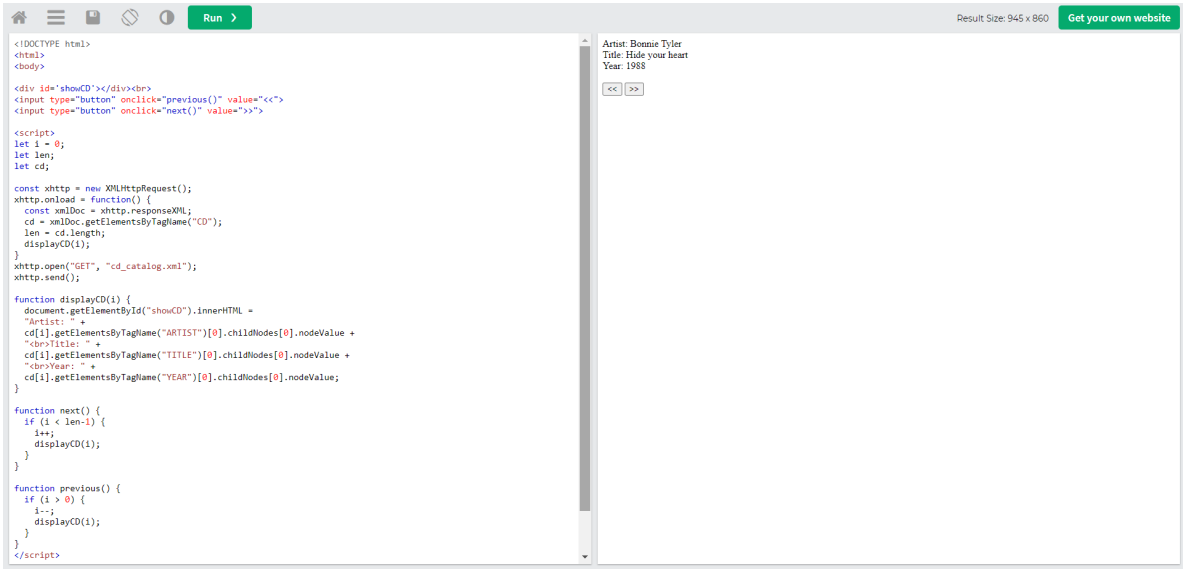

Figura 38: Navegar entre CD

- **Mostrar información del álbum al hacer clic en un CD**

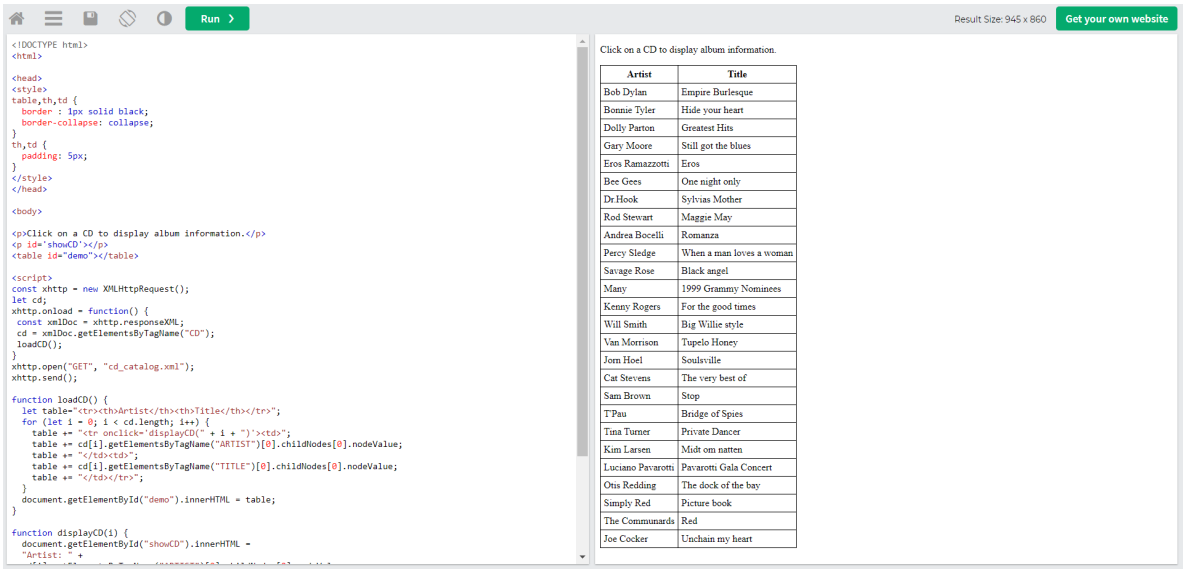

Figura 39: Mostrar tabla de CD

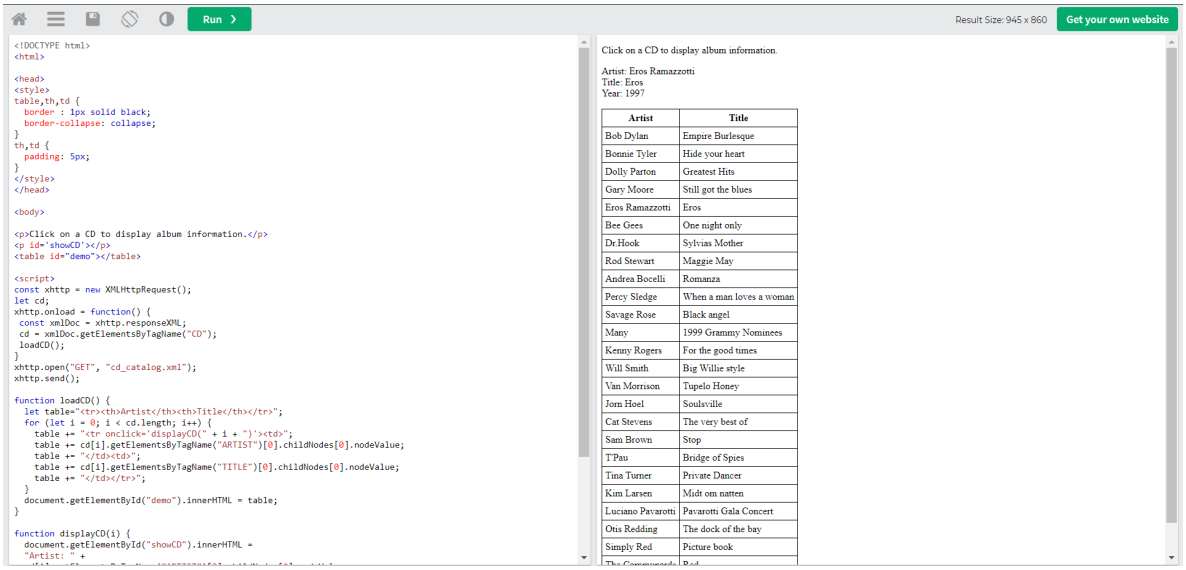

Figura 40: Haciendo Click en un CD

6.1.10. AJAX - Examples

- **SIMPLES EXAMPLES**

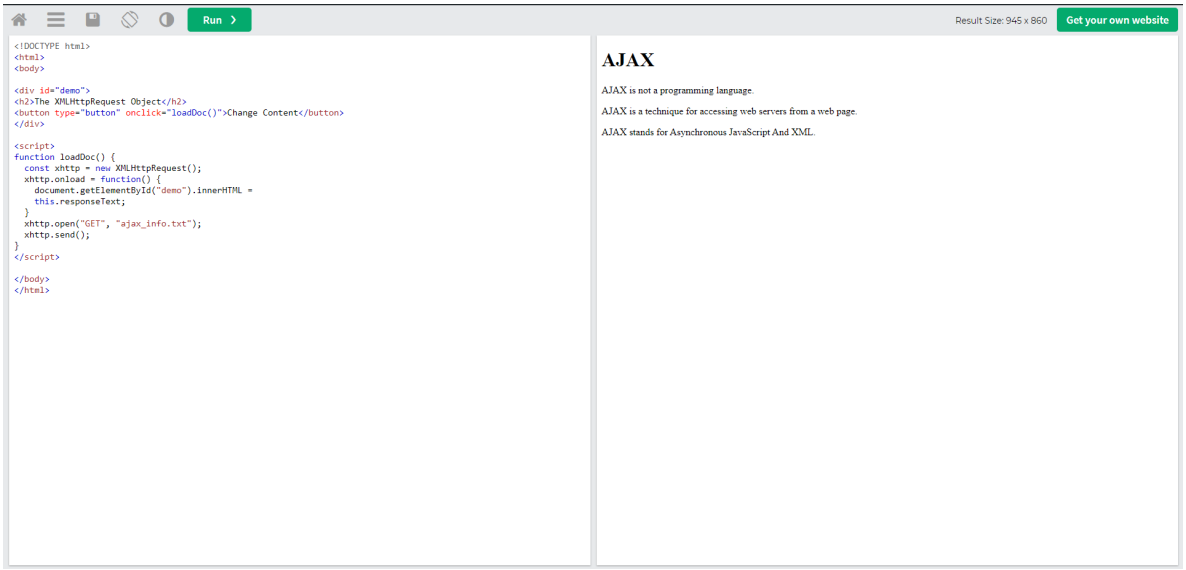

Figura 41: Ejemplo 1

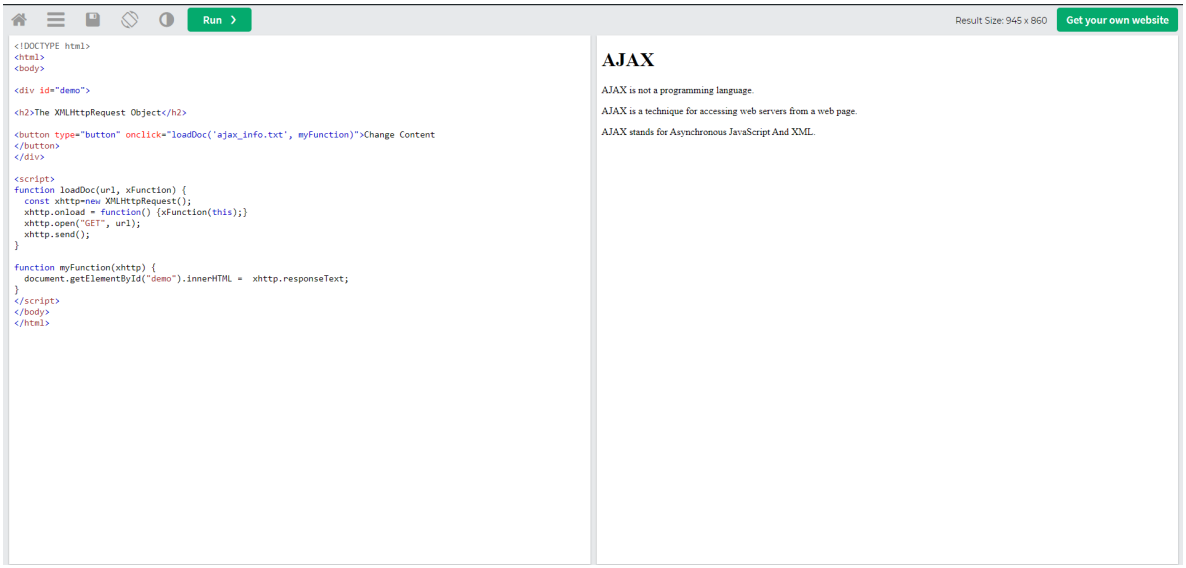

Figura 42: Ejemplo 2

- **Request Header Information**

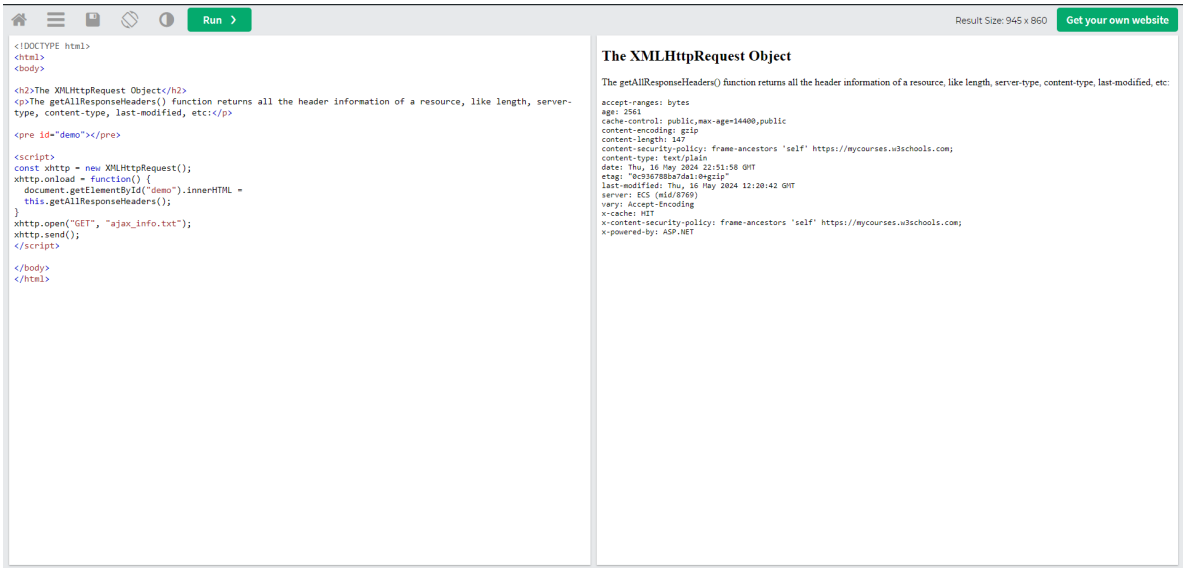

Figura 43: Ejemplo 1

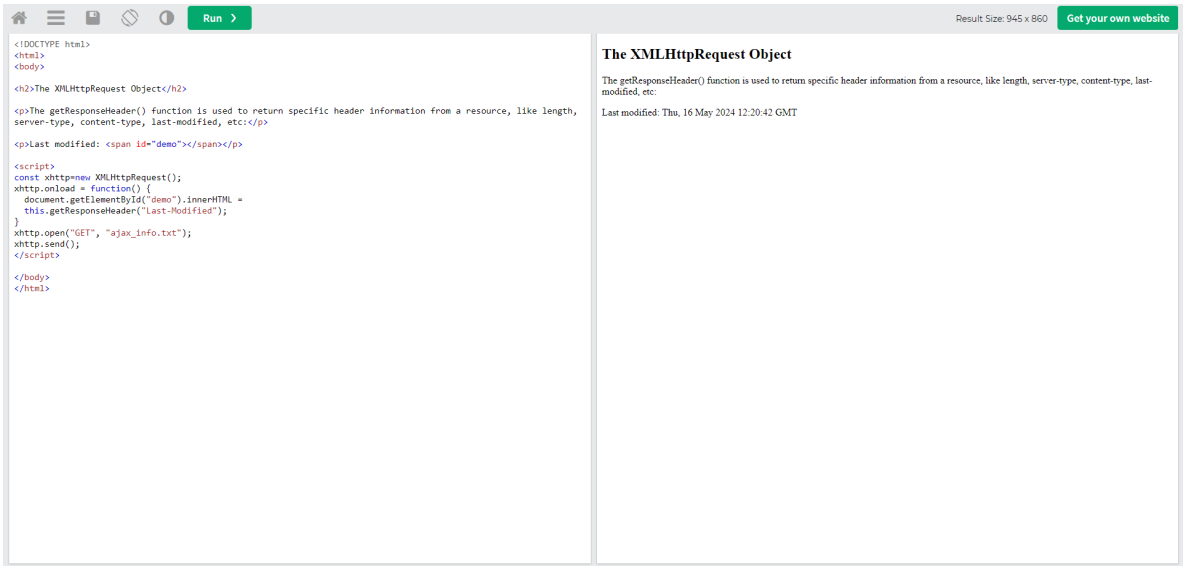

Figura 44: Ejemplo 2

- **Request XML Files**

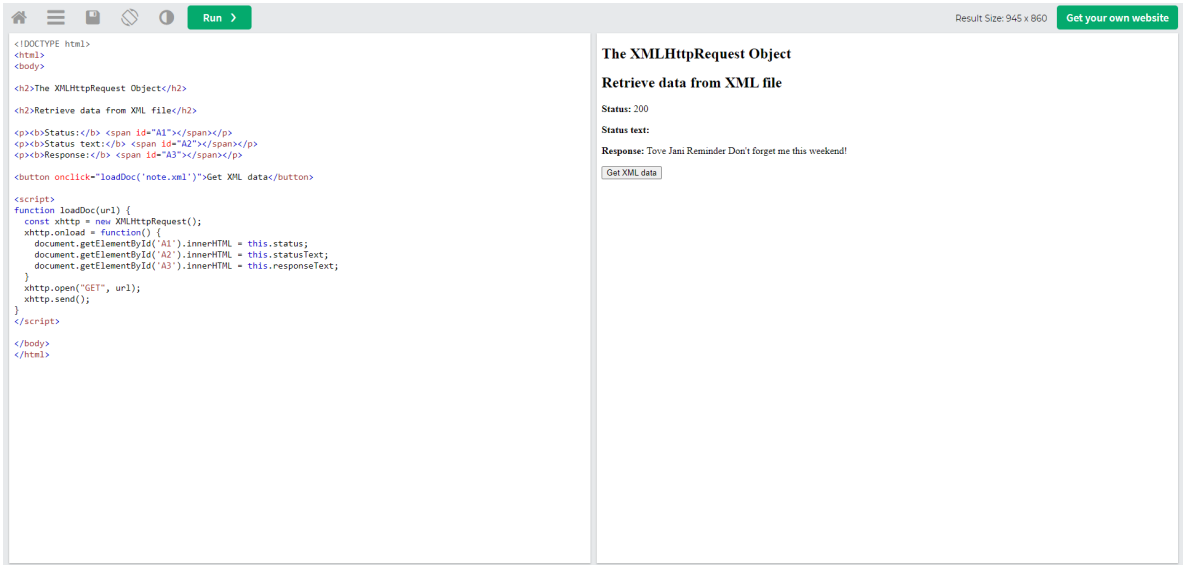

Figura 45: Ejemplo 1

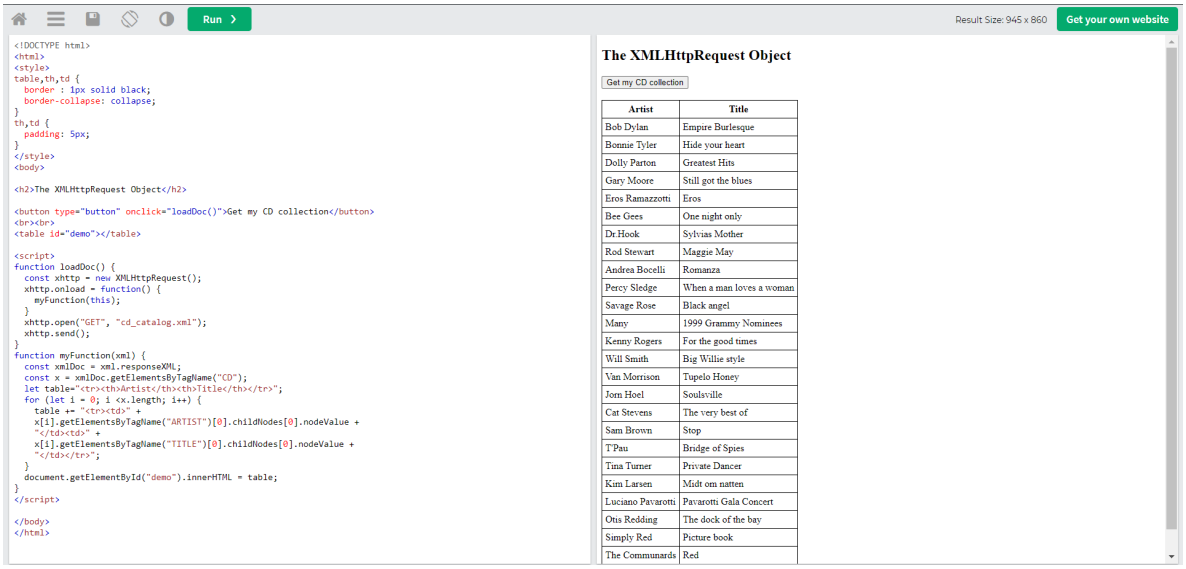

Figura 46: Ejemplo 2

- **Retrieve Server Data with PHP and ASP**

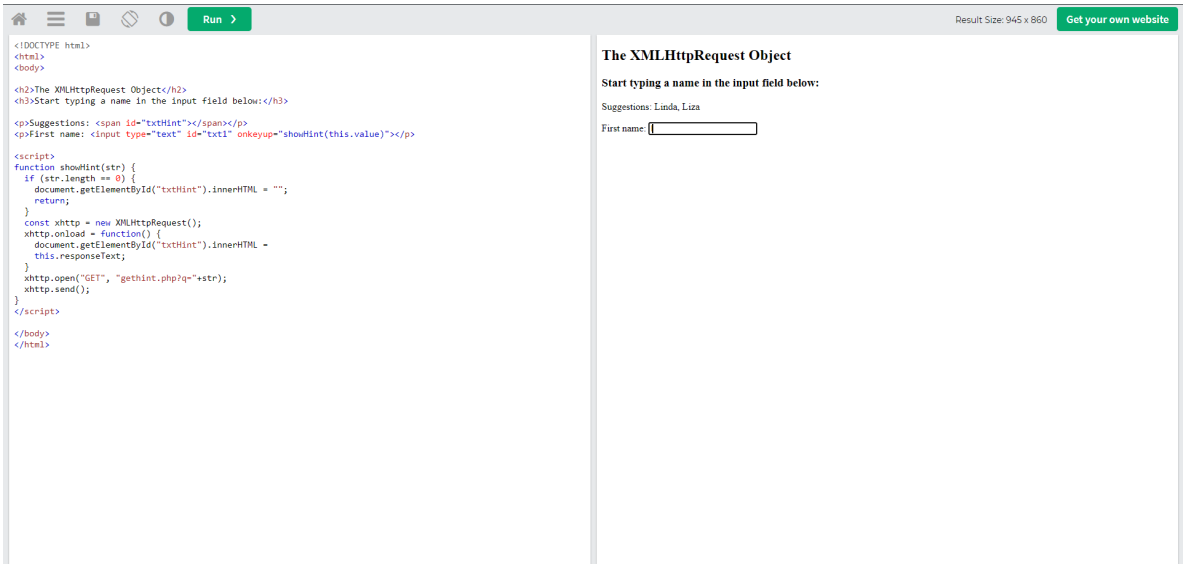

Figura 47: Ejemplo 1

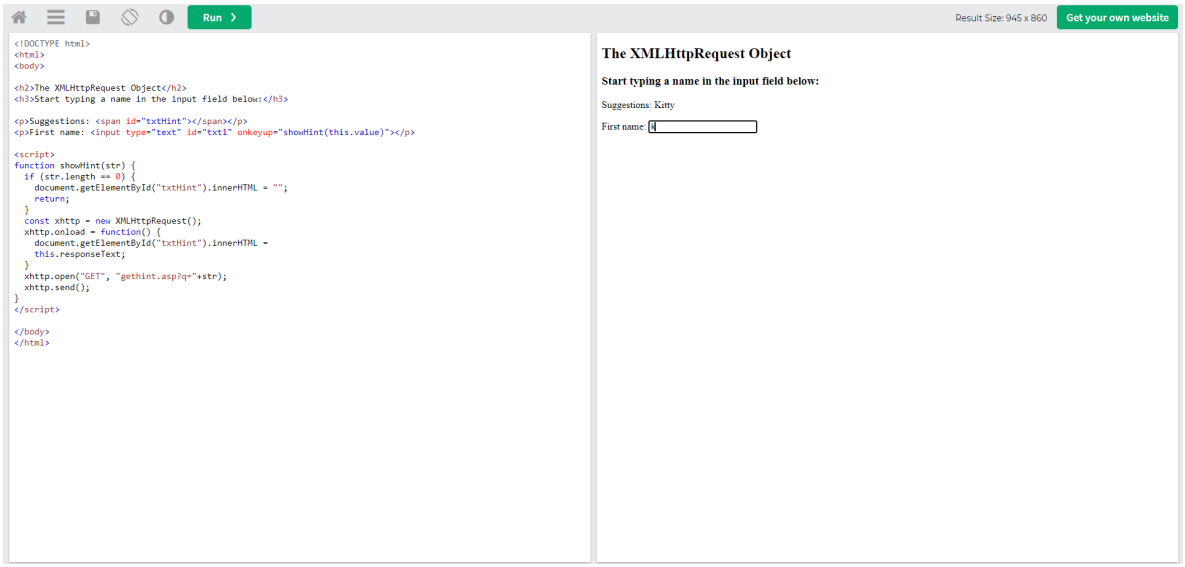

Figura 48: Ejemplo 2

- **Retrieve Database Information**

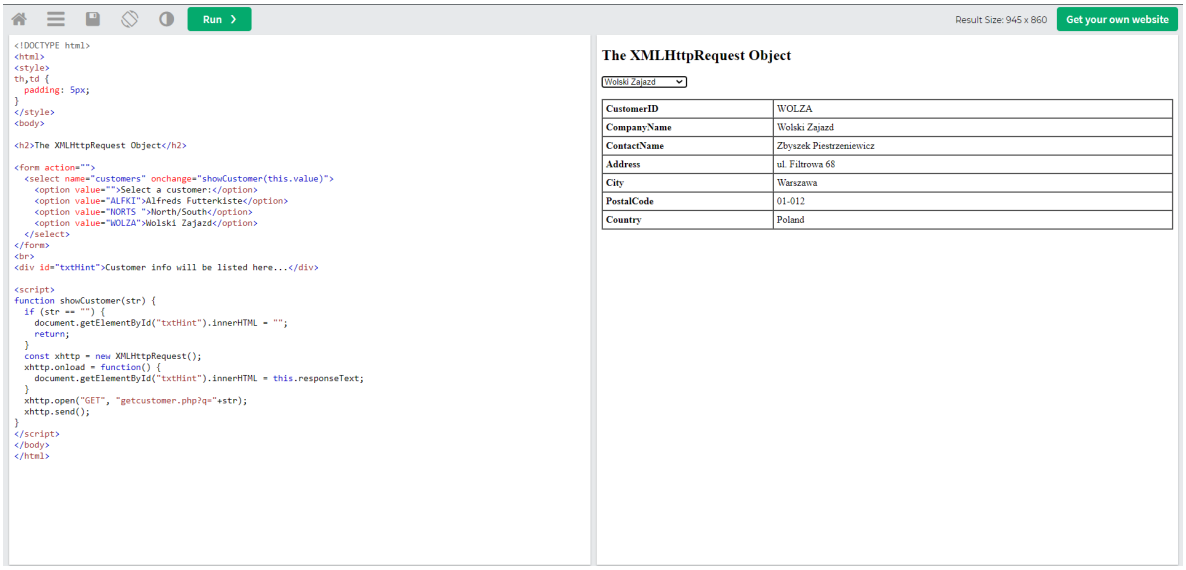

Figura 49: Ejemplo 1

- **Ajax Applications**

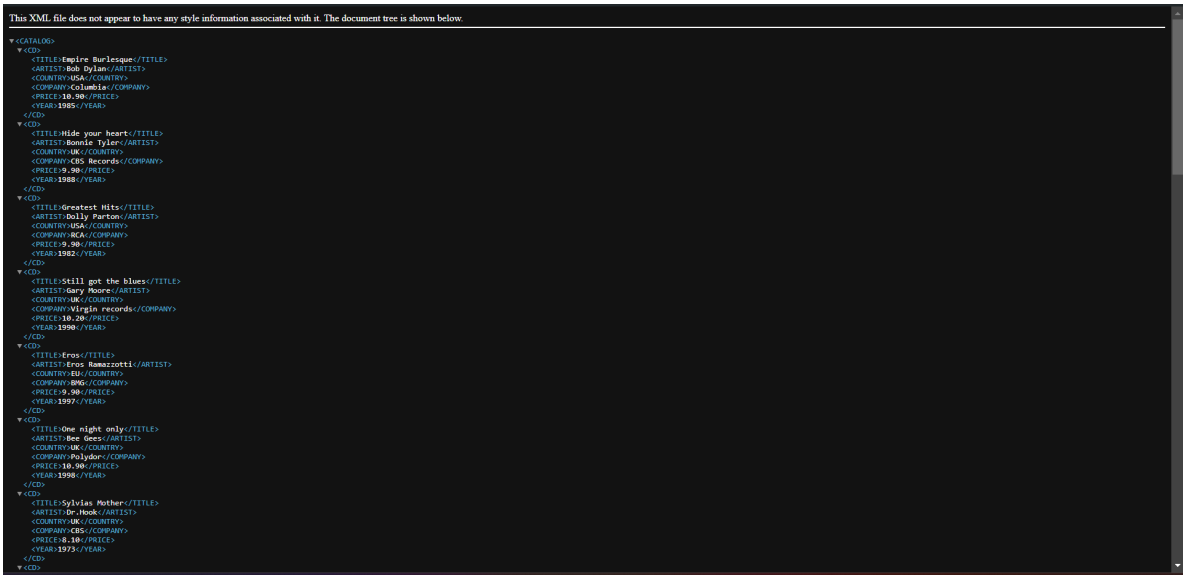

Figura 50: XML

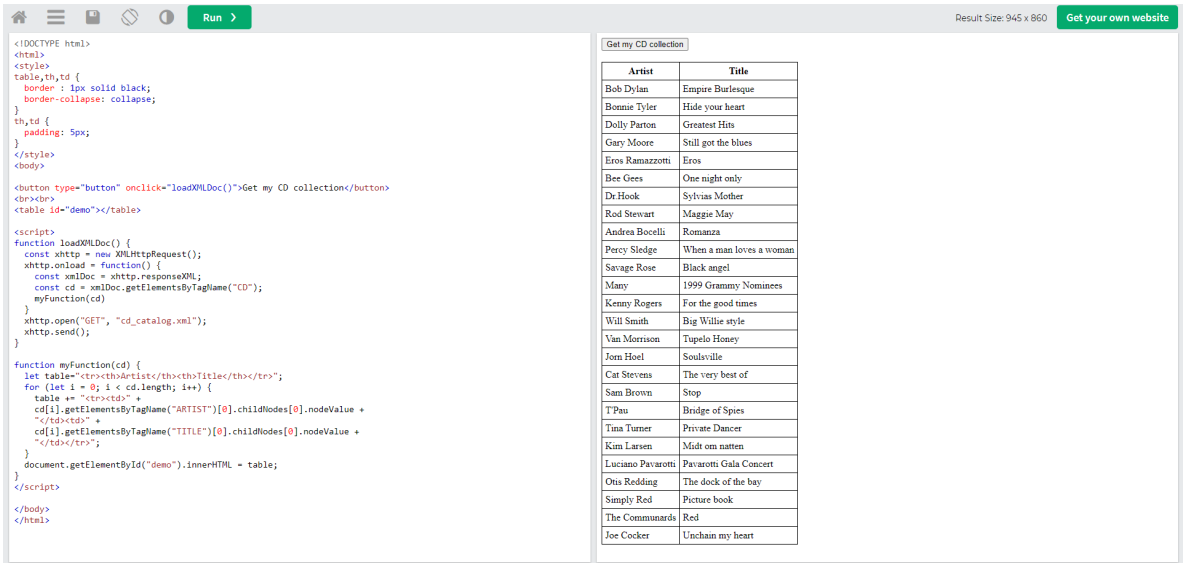

Figura 51: Ejemplo 1

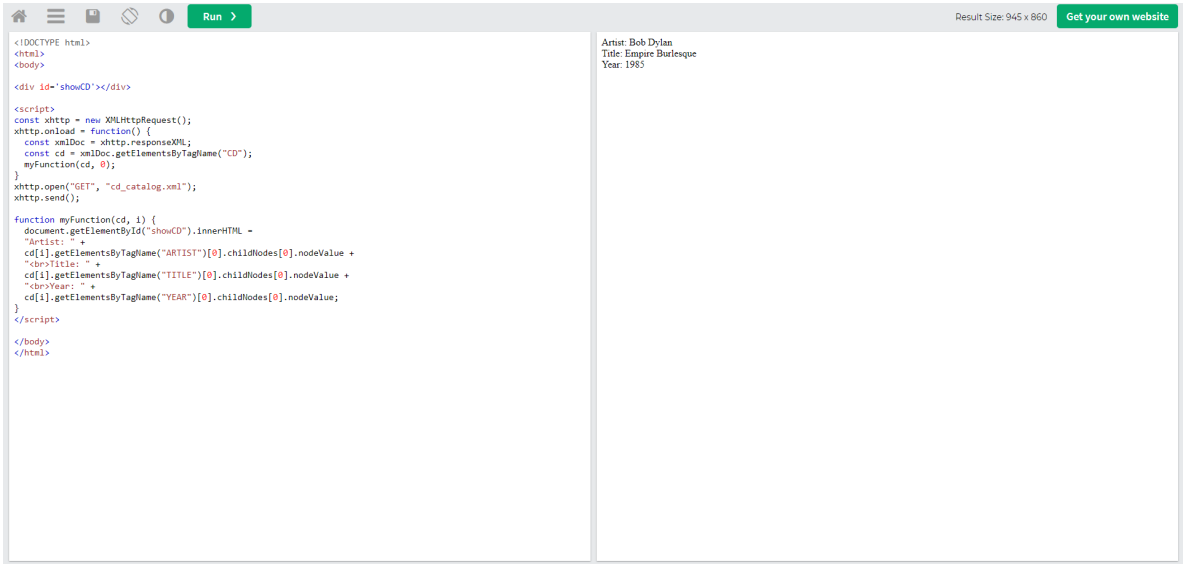

Figura 52: Ejemplo 2

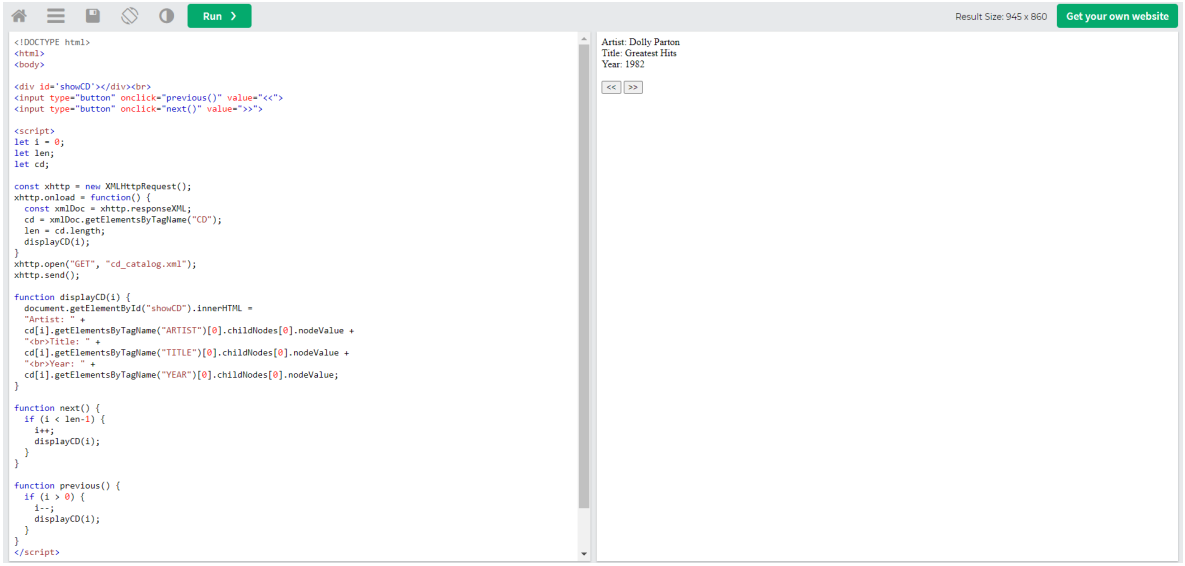

Figura 53: Ejemplo 3

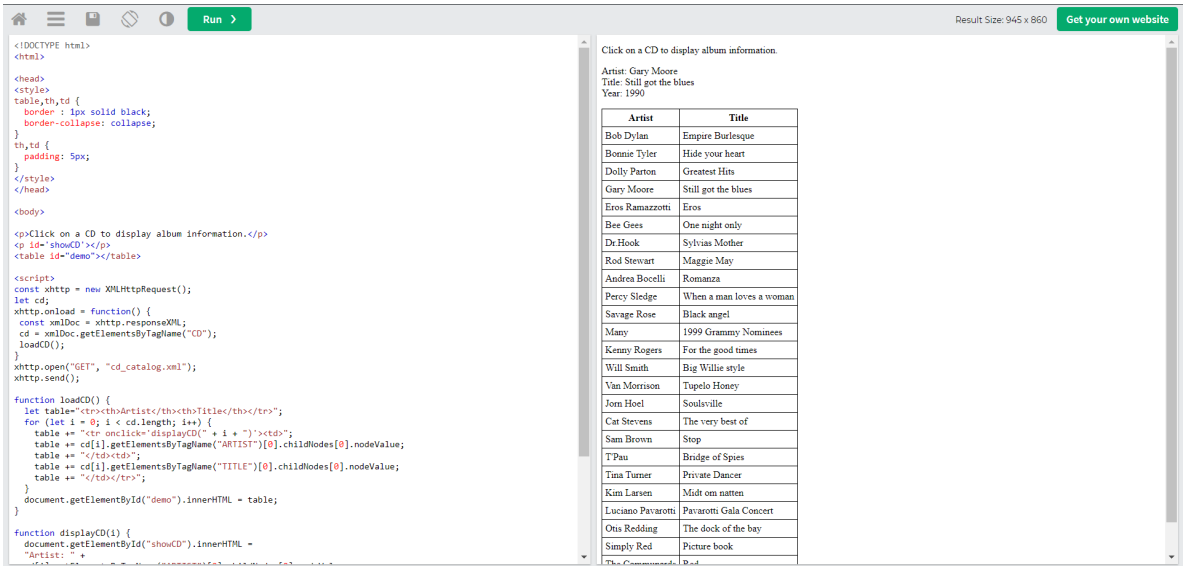

Figura 54: Ejemplo 4

6.2. Uso de Xampp

- Proporciona un entorno de servidor local que te permite desarrollar y probar aplicaciones web antes de publicarlas en un servidor web en producción. Usado para los ejercicios 4 - 8.

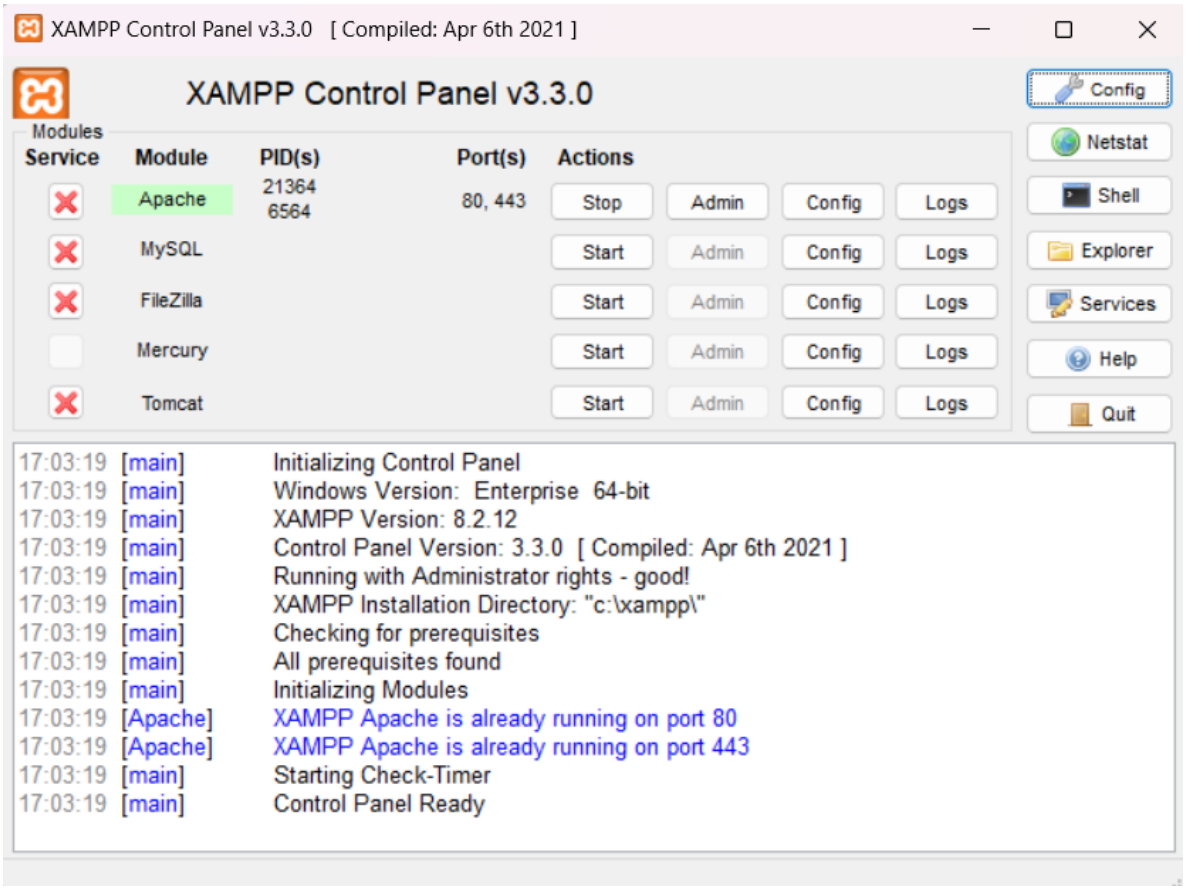

Figura 55: Xampp

6.3. Uso de Node.js

- Es una plataforma que permite ejecutar código JavaScript del lado del servidor, ofreciendo un entorno altamente escalable y eficiente para el desarrollo de aplicaciones web modernas. Usado para la página web Markdown y del ejercicio 1 - 3

6.3.1. Instalación

Al descargar el installer, solo seguir los pasos correspondientes.

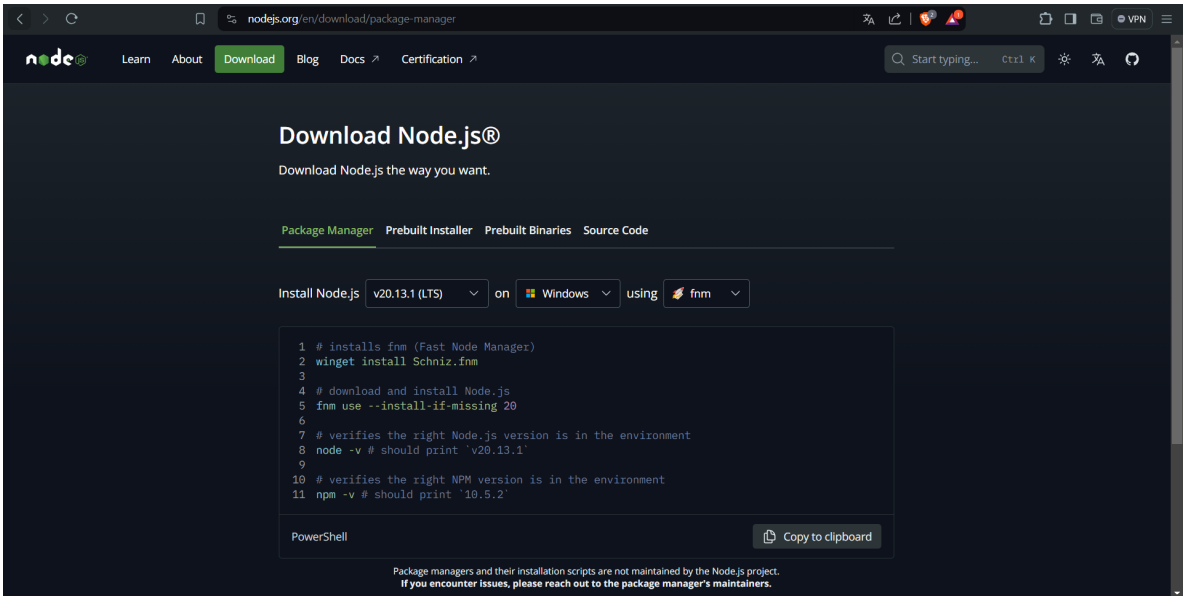

Figura 56: Instalación Node.js

6.4. Investigación sobre Chart.js

6.4.1. Página Oficial

Figura 57: Chart.js

6.4.2. Documentación

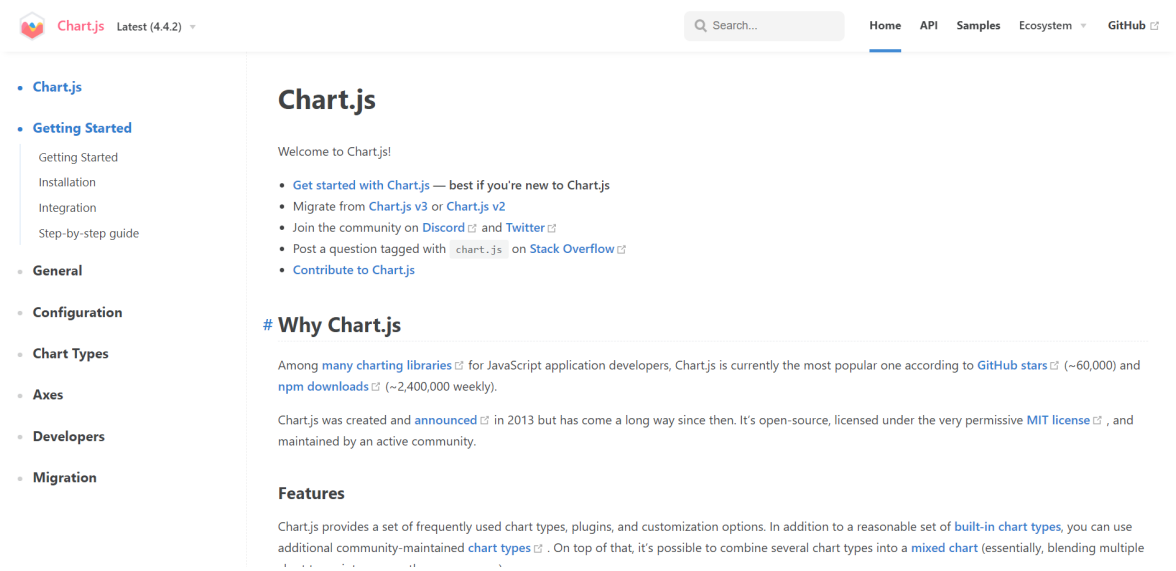

Figura 58: Chart.js DOC

6.5. EJERCICIOS

6.5.1. Ejercicio 1

- textbfScript

```
import('node-fetch').then(async (module) => {
  const fetch = module.default;
    const response = await fetch('http://localhost/nuevo/JASON/data.json');
    // Verifica si la respuesta fue exitosa
    if (response.ok) {
        const data = await response.json();
        console.log("\t Lista de Regiones");
        //Itera sobre los datos obtenidos e imprime las regiones
        for (let i = 0; i < data.length; i++) {
            console.log(data[i].region);
        }
    } else {
        console.error('Error en la solicitud:', response.status);
    }
//Captura de error
}).catch(error => {
    console.error('Error:', error);
});
```

Figura 59: Script 1

- **Ejecución**

```
NppExec Console
Current directory: C:\xampp\htdocs\Nuevo\Ejercicio1
node Ejercicio1.js
Process started (PID=21780) >>>
            Lista de Regiones
Amazonas
Ancash
Apurimac
Arequipa
Ayacucho
Cajamarca
Callao
Cusco
Huancavelica
Huanuco
Ica
Junin
La Libertad
Lambayeque
Lima
Loreto
Madre de Dios
Moquegua
Pasco
Piura
Puno
San Martin
Tacna
Tumbes
Ucayali
<<< Process finished (PID=21780). (Exit code 0)
================ READY ================
```

Figura 60: Ejercicio 1

6.5.2. Ejercicio 2

- **Script**

```
1  import('node-fetch').then(async (module) => {
2    const fetch = module.default;
3    const response = await fetch('http://localhost/nuevo/JASON/data.json');
4    if (response.ok) {
5      const data = await response.json();
6      console.log("\t Lista de Regiones");
7      //Itera sobre los datos obtenidos e imprime la ultima confirmación dada osea la mayor
8      for (let i = 0; i < data.length; i++) {
9        let confirmados = data[i].confirmed;
10       let region = data[i].region;
11       let totalConfirmados = confirmados[confirmados.length - 1].value;
12       console.log(region + ": " + totalConfirmados);
13     }
14   } else {
15     console.error('Error en la solicitud:', response.status);
16   }
17 //Captura de error
18 }).catch(error => {
19   console.error('Error:', error);
20 });
```

Figura 61: Script 2

- **Ejecución**

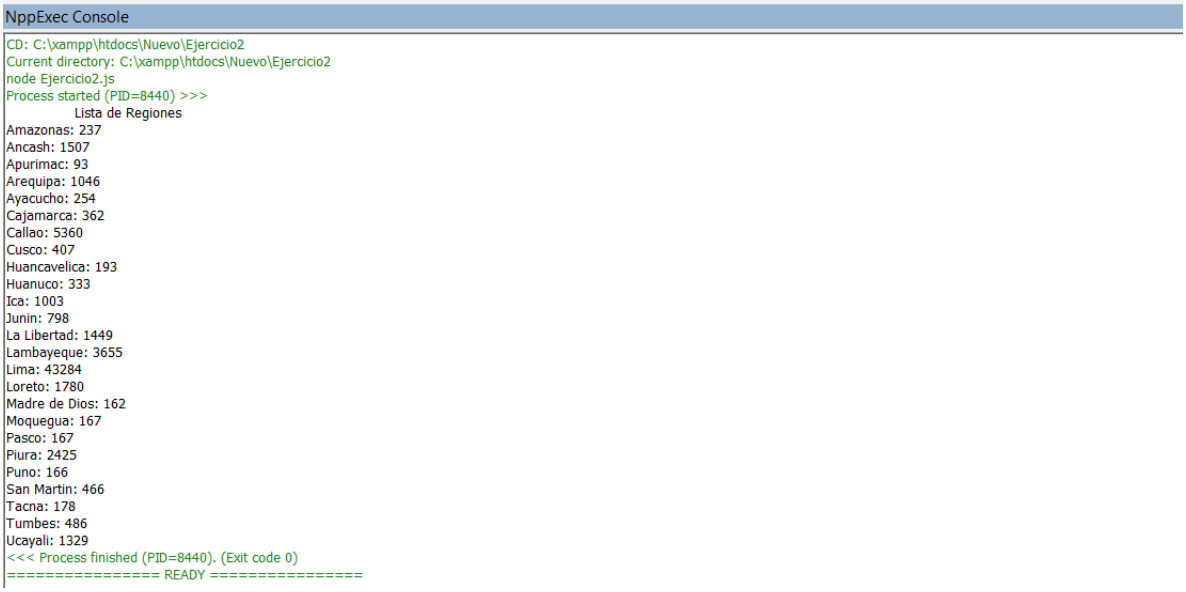

Figura 62: Ejercicio 2

6.5.3. Ejercicio 3

- **Script**

```
import('node-fetch').then(async (module) => {
  const fetch = module.default;
  const response = await fetch('http://localhost/nuevo/JASON/data.json');
  let mapa = new Map();
  if (response.ok) {
    const data = await response.json();
    console.log("\t Lista de Regiones");
    //Agregar datos tanto suma como region a un mapa
    for (let i = 0; i < data.length; i++) {
      let confirmados = data[i].confirmed;
      let region = data[i].region;
      let suma = 0;
      for(let c = 0; c < confirmados.length; c++){
        suma += parseInt(confirmados[c].value);
      }
      mapa.set(region,suma);
    }
    //convertir mapa en arreglo
    let arreglo = Array.from(mapa);
    //Ordenar arreglo
    arreglo.sort((a, b) => b[1] - a[1]);
    let ordenamiento = new Map(arreglo);
    let i = 0;
    //Imprimir datos del mapa
    for (let [clave, valor] of ordenamiento) {
      console.log((i + 1) + " " + clave + ": " + valor);
      i++;
      if(i >= 10){
        break;
      }
    }
  } else {
    console.error('Error en la solicitud:', response.status);
  }
}).catch(error => {
  console.error('Error:', error);
});
```

Figura 63: Script 3

- **Ejecución**

```
NppExec Console
CD: C:\xampp\htdocs\Nuevo\Ejercicio3
Current directory: C:\xampp\htdocs\Nuevo\Ejercicio3
node Ejercicio3.js
Process started (PID=6192) >>>
            Lista de Regiones
1 Lima: 626744
2 Callao: 78099
3 Lambayeque: 51206
4 Loreto: 30242
5 Piura: 29966
6 Ancash: 18753
7 La Libertad: 18175
8 Ucayali: 14488
9 Arequipa: 13817
10 Ica: 11925
<<< Process finished (PID=6192). (Exit code 0)
================ READY ================
```

Figura 64: Ejercicio 3

6.5.4. Ejercicio 4

- **HTML**

```
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title>Ejercicio 4</title>
    <meta charset="UTF-8"/>
    <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
    <meta name="description" content="Usando ajax para la comparación de gráficas"/>
    <meta name="keywords" content="comparación, ajax, gráfica"/>
    <link rel="stylesheet" href="http://localhost/nuevo/css/estilos.css">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body class="cuerpo">
    <div>
      <form id="comparasion">
      <center>
        <button class="Generar" type="submit">Generar</button>
      </center>
    </div>
    <div>
        <canvas id="grafica" width="800" height="650"></canvas>
    </div>
    <script src="Script.js"></script>
  </body>
</html>
```

Figura 65: HTML

- **Script**

```
document.getElementById("comparasion").addEventListener("submit", function (evento) {
  evento.preventDefault();

  const region1 = 'Apurimac';
  const region2 = 'Lima';

  fetch("http://localhost/nuevo/JASON/data.json")
    .then(response => response.json())
    .then(data => {
      //Solicitando informacion del archivo data.json
      const datosRegion1 = data.find(region => region.region === region1);
      const datosRegion2 = data.find(region => region.region === region2);

      const fechas = datosRegion1.confirmed.map(entry => entry.date);
      const valoresRegion1 = datosRegion1.confirmed.map(entry => parseInt(entry.value));
      const valoresRegion2 = datosRegion2.confirmed.map(entry => parseInt(entry.value));
      //caso contrario la crea
      const ctx = document.getElementById('grafica').getContext('2d');
      myChart = new Chart(ctx, {
        type: 'line',
        data: {
          labels: fechas,
          datasets: [
            {
              label: region1,
              data: valoresRegion1,
              borderColor: 'blue',
              backgroundColor: 'rgba(0, 0, 255, 0.1)'
            },
            {
              label: region2,
              data: valoresRegion2,
              borderColor: 'green',
              backgroundColor: 'rgba(0, 255, 0, 0.1)'
            }
          ]
        },
        options: {
          responsive: true,
          maintainAspectRatio: false
        }
      });
    })
    .catch(error => console.error('Error:', error));
});
```

Figura 66: Script 4

- **Ejecución**

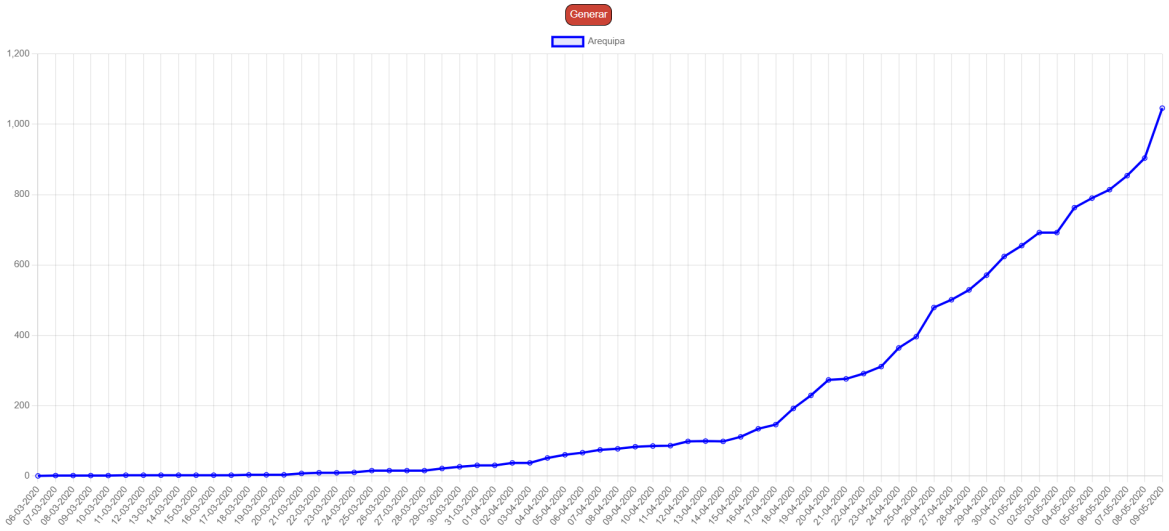

Figura 67: Ejercicio 4

6.5.5. Ejercicio 5

- **comparasion 1**
  - **HTML**

```html
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title>Ejercicio 5</title>
    <meta charset="UTF-8"/>
    <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
    <meta name="description" content="Usando ajax para la comparación de gráficas"/>
    <meta name="keywords" content="comparación, ajax, gráfica"/>
    <link rel="stylesheet" href="http://localhost/nuevo/css/estilos.css">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body class="cuerpo">
    <div>
      <form id="comparasion">
      <center>
        <button class="Generar" type="submit">Generar</button>
      </center>
    </div>
    <div>
        <canvas id="grafica" width="800" height="650"></canvas>
    </div>
    <script src="Script.js"></script>
  </body>
</html>
```

Figura 68: HTML 5

- **Script**

```
document.getElementById("comparasion").addEventListener("submit", function (evento) {
  evento.preventDefault();

  const region1 = 'Apurimac';
  const region2 = 'Lima';

  fetch("http://localhost/nuevo/JASON/data.json")
    .then(response => response.json())
    .then(data => {
      //Solicitando informacion del archivo data.json
      const datosRegion1 = data.find(region => region.region === region1);
      const datosRegion2 = data.find(region => region.region === region2);

      const fechas = datosRegion1.confirmed.map(entry => entry.date);
      const valoresRegion1 = datosRegion1.confirmed.map(entry => parseInt(entry.value));
      const valoresRegion2 = datosRegion2.confirmed.map(entry => parseInt(entry.value));
      //caso contrario la crea
      const ctx = document.getElementById('grafica').getContext('2d');
      myChart = new Chart(ctx, {
        type: 'line',
        data: {
          labels: fechas,
          datasets: [
            {
              label: region1,
              data: valoresRegion1,
              borderColor: 'blue',
              backgroundColor: 'rgba(0, 0, 255, 0.1)'
            },
            {
              label: region2,
              data: valoresRegion2,
              borderColor: 'green',
              backgroundColor: 'rgba(0, 255, 0, 0.1)'
            }
          ]
        },
        options: {
          responsive: true,
          maintainAspectRatio: false
        }
      });
    })
    .catch(error => console.error('Error:', error));
});
```

Figura 69: Script 5 - 1

- **Ejecución**

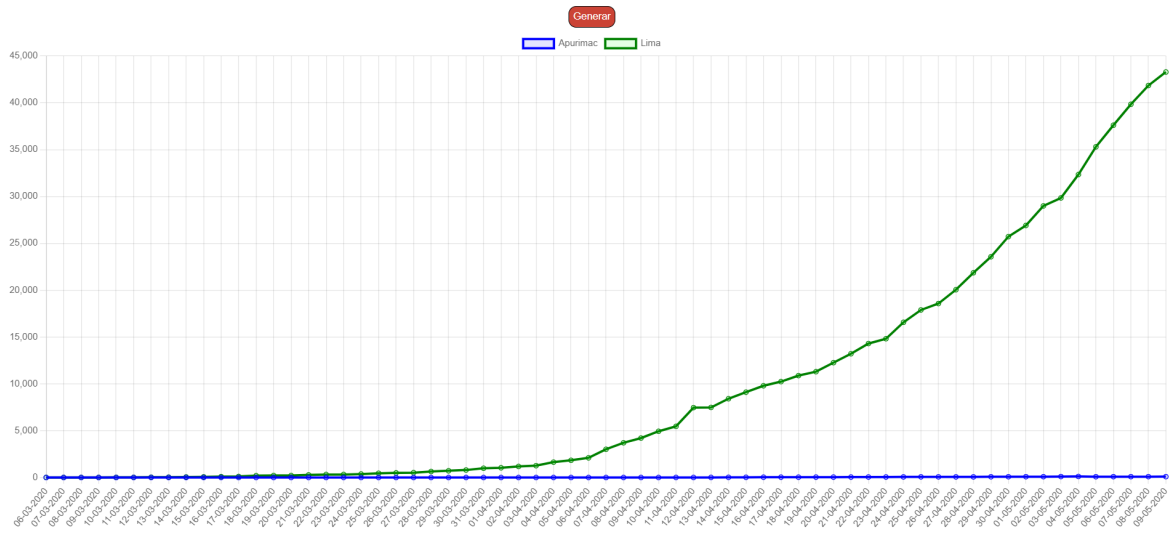

Figura 70: Ejercicio 5 - 1

- **comparasion 2**
  - **HTML**

```
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title>Ejercicio 5</title>
    <meta charset="UTF-8"/>
    <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
    <meta name="description" content="Usando ajax para la comparación de gráficas"/>
    <meta name="keywords" content="comparación, ajax, gráfica"/>
    <link rel="stylesheet" href="http://localhost/nuevo/css/estilos.css">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body class="cuerpo">
    <div>
      <form id="comparasion">
      <center>
        <button class="Generar" type="submit">Generar</button>
      </center>
    </div>
    <div>
        <canvas id="grafica" width="800" height="650"></canvas>
    </div>
    <script src="Script.js"></script>
  </body>
</html>
```

Figura 71: HTML 5

- **Script**

```
document.getElementById("comparasion").addEventListener("submit", function (evento) {
  evento.preventDefault();

  const region1 = 'Ancash';
  const region2 = 'Callao';

  fetch("http://localhost/nuevo/JASON/data.json")
    .then(response => response.json())
    .then(data => {
      //Solicitando informacion del archivo data.json
      const datosRegion1 = data.find(region => region.region === region1);
      const datosRegion2 = data.find(region => region.region === region2);

      const fechas = datosRegion1.confirmed.map(entry => entry.date);
      const valoresRegion1 = datosRegion1.confirmed.map(entry => parseInt(entry.value));
      const valoresRegion2 = datosRegion2.confirmed.map(entry => parseInt(entry.value));
      //caso contrario la crea
      const ctx = document.getElementById('grafica').getContext('2d');
      myChart = new Chart(ctx, {
        type: 'line',
        data: {
          labels: fechas,
          datasets: [
            {
              label: region1,
              data: valoresRegion1,
              borderColor: 'blue',
              backgroundColor: 'rgba(0, 0, 255, 0.1)'
            },
            {
              label: region2,
              data: valoresRegion2,
              borderColor: 'green',
              backgroundColor: 'rgba(0, 255, 0, 0.1)'
            }
          ]
        },
        options: {
          responsive: true,
          maintainAspectRatio: false
        }
      });
    })
    .catch(error => console.error('Error:', error));
});
```

Figura 72: Script 5 - 2

- **Ejecución**

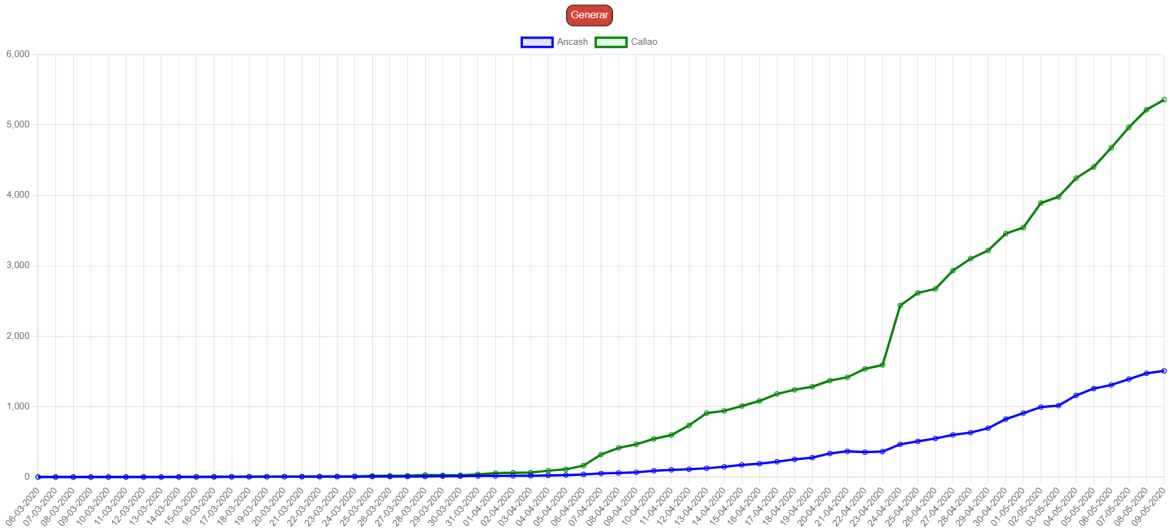

Figura 73: Ejercicio 5 - 2

- **comparasion 3**
  - **HTML**

```html
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title>Ejercicio 5</title>
    <meta charset="UTF-8"/>
    <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
    <meta name="description" content="Usando ajax para la comparación de gráficas"/>
    <meta name="keywords" content="comparación, ajax, gráfica"/>
    <link rel="stylesheet" href="http://localhost/nuevo/css/estilos.css">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body class="cuerpo">
    <div>
      <form id="comparasion">
      <center>
        <button class="Generar" type="submit">Generar</button>
      </center>
    </div>
    <div>
        <canvas id="grafica" width="800" height="650"></canvas>
    </div>
    <script src="Script.js"></script>
  </body>
</html>
```

Figura 74: HTML 5

- **Script**

```
document.getElementById("comparasion").addEventListener("submit", function (evento) {
  evento.preventDefault();

  const region1 = 'Tacna';
  const region2 = 'Cusco';

  fetch("http://localhost/nuevo/JASON/data.json")
    .then(response => response.json())
    .then(data => {
      //Solicitando informacion del archivo data.json
      const datosRegion1 = data.find(region => region.region === region1);
      const datosRegion2 = data.find(region => region.region === region2);

      const fechas = datosRegion1.confirmed.map(entry => entry.date);
      const valoresRegion1 = datosRegion1.confirmed.map(entry => parseInt(entry.value));
      const valoresRegion2 = datosRegion2.confirmed.map(entry => parseInt(entry.value));
      //caso contrario la crea
      const ctx = document.getElementById('grafica').getContext('2d');
      myChart = new Chart(ctx, {
        type: 'line',
        data: {
          labels: fechas,
          datasets: [
            {
              label: region1,
              data: valoresRegion1,
              borderColor: 'blue',
              backgroundColor: 'rgba(0, 0, 255, 0.1)'
            },
            {
              label: region2,
              data: valoresRegion2,
              borderColor: 'green',
              backgroundColor: 'rgba(0, 255, 0, 0.1)'
            }
          ]
        },
        options: {
          responsive: true,
          maintainAspectRatio: false
        }
      });
    })
    .catch(error => console.error('Error:', error));
});
```

Figura 75: Script 5 - 3

- **Ejecución**

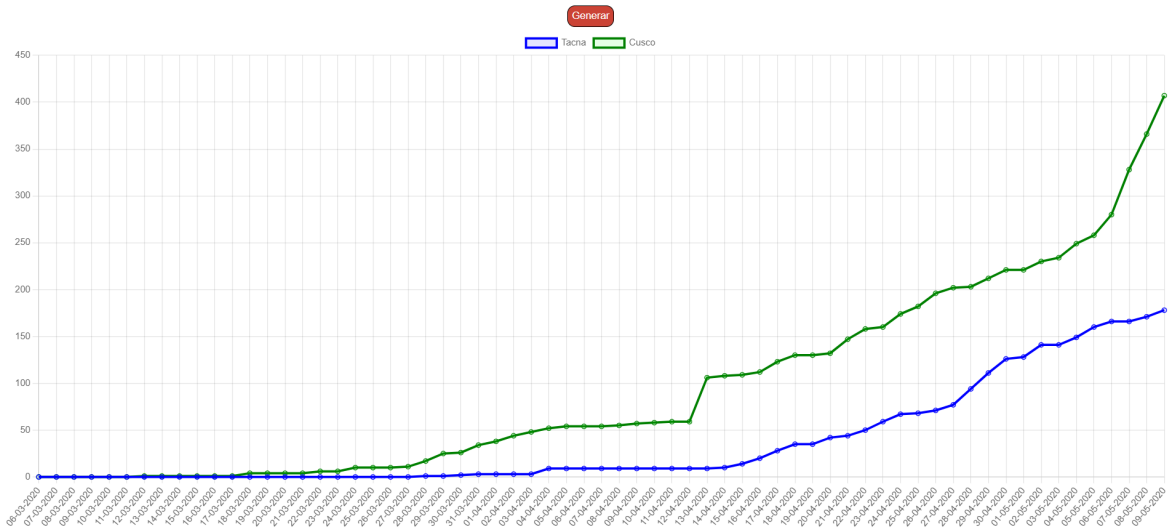

Figura 76: Ejercicio 5 - 3

- **comparasion 4**
  - **HTML**

```html
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title>Ejercicio 5</title>
    <meta charset="UTF-8"/>
    <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
    <meta name="description" content="Usando ajax para la comparación de gráficas"/>
    <meta name="keywords" content="comparación, ajax, gráfica"/>
    <link rel="stylesheet" href="http://localhost/nuevo/css/estilos.css">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body class="cuerpo">
    <div>
      <form id="comparasion">
      <center>
        <button class="Generar" type="submit">Generar</button>
      </center>
    </div>
    <div>
        <canvas id="grafica" width="800" height="650"></canvas>
    </div>
    <script src="Script.js"></script>
  </body>
</html>
```

Figura 77: HTML 5

- **Script**

```
document.getElementById("comparasion").addEventListener("submit", function (evento) {
  evento.preventDefault();

  const region1 = 'Pasco';
  const region2 = 'Piura';

  fetch("http://localhost/nuevo/JASON/data.json")
    .then(response => response.json())
    .then(data => {
      //Solicitando informacion del archivo data.json
      const datosRegion1 = data.find(region => region.region === region1);
      const datosRegion2 = data.find(region => region.region === region2);

      const fechas = datosRegion1.confirmed.map(entry => entry.date);
      const valoresRegion1 = datosRegion1.confirmed.map(entry => parseInt(entry.value));
      const valoresRegion2 = datosRegion2.confirmed.map(entry => parseInt(entry.value));
      //caso contrario la crea
      const ctx = document.getElementById('grafica').getContext('2d');
      myChart = new Chart(ctx, {
        type: 'line',
        data: {
          labels: fechas,
          datasets: [
            {
              label: region1,
              data: valoresRegion1,
              borderColor: 'blue',
              backgroundColor: 'rgba(0, 0, 255, 0.1)'
            },
            {
              label: region2,
              data: valoresRegion2,
              borderColor: 'green',
              backgroundColor: 'rgba(0, 255, 0, 0.1)'
            }
          ]
        },
        options: {
          responsive: true,
          maintainAspectRatio: false
        }
      });
    })
    .catch(error => console.error('Error:', error));
});
```

Figura 78: Script 5 - 4

- **Ejecución**

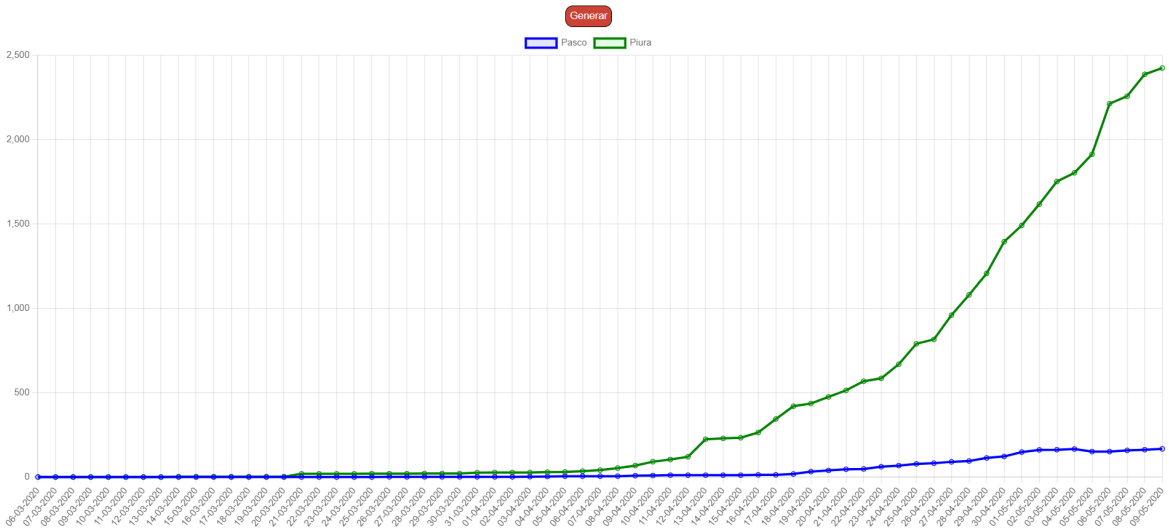

Figura 79: Ejercicio 5 - 4

6.5.6. Ejercicio 6

- **HTML**

```
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title>Ejercicio 6</title>
    <meta charset="UTF-8"/>
    <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
    <meta name="description" content="Usando ajax para la comparación de gráficas"/>
    <meta name="keywords" content="comparación, ajax, gráfica"/>
    <link rel="stylesheet" href="http://localhost/nuevo/css/estilos.css">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body class="cuerpo">
    <div>
      <form id="comparasion">
      <center>
        <button class="Generar" type="submit">Generar</button>
      </center>
    </div>
    <div>
        <canvas id="grafica" width="800" height="650"></canvas>
    </div>
    <script src="Script.js"></script>
  </body>
</html>
```

Figura 80: HTML 6

- **Script**

```
document.getElementById("comparasion").addEventListener("submit", function (evento) {
  evento.preventDefault();

  fetch("http://localhost/nuevo/JASON/data.json")
    .then(response => response.json())
    .then(data => {
      //extrayendo primera región
      const primeraRegion = data.find(region => region.region !== 'Lima' && region.region !== 'Callao');

      // Verificar que se encontró la primera región
      if (!primeraRegion) {
        console.error('No se encontró la primera región con las fechas necesarias para el gráfico.');
        return;
      }

      // Obtener las fechas de la primera región
      const fechas = primeraRegion.confirmed.map(entry => entry.date);

      // Extraer datos  de las regiones menos Lima y Callao
      const datasets = data.filter(region => region.region !== 'Lima' && region.region !== 'Callao').map(region => {
        const valores = [];
        // Llenar los valores con los datos correspondientes a las fechas de la primera región
        fechas.forEach(fecha => {
          const dato = region.confirmed.find(entry => entry.date === fecha);
          if (dato) {
            valores.push(parseInt(dato.value));
          } else {
            valores.push(null);
          }
        });
        return {
          label: region.region,
          data: valores,
          borderColor: getRandomColor(),
          backgroundColor: 'rgba(0, 0, 255, 0.1)',
          pointStyle: 'none',
          pointRadius: 0
        };
      });
```

Figura 81: Script 6 - 1

```
40
41          // Crear el gráfico con los datos de todas las regiones
42          const ctx = document.getElementById('grafica').getContext('2d');
43          myChart = new Chart(ctx, {
44            type: 'line',
45            data: {
46              labels: fechas,
47              datasets: datasets
48            },
49            options: {
50              responsive: true,
51              maintainAspectRatio: false,
52              scales: {
53                y: {
54                  beginAtZero: true
55                }
56              }
57            }
58          });
59        })
60        .catch(error => console.error('Error:', error));
61    });
62
63    // Función para obtener un color aleatorio
64    function getRandomColor() {
65      const letters = '0123456789ABCDEF';
66      let color = '#';
67      for (let i = 0; i < 6; i++) {
68        color += letters[Math.floor(Math.random() * 16)];
69      }
70      return color;
71    }
```

Figura 82: Script 6 - 2

- **Ejecución**

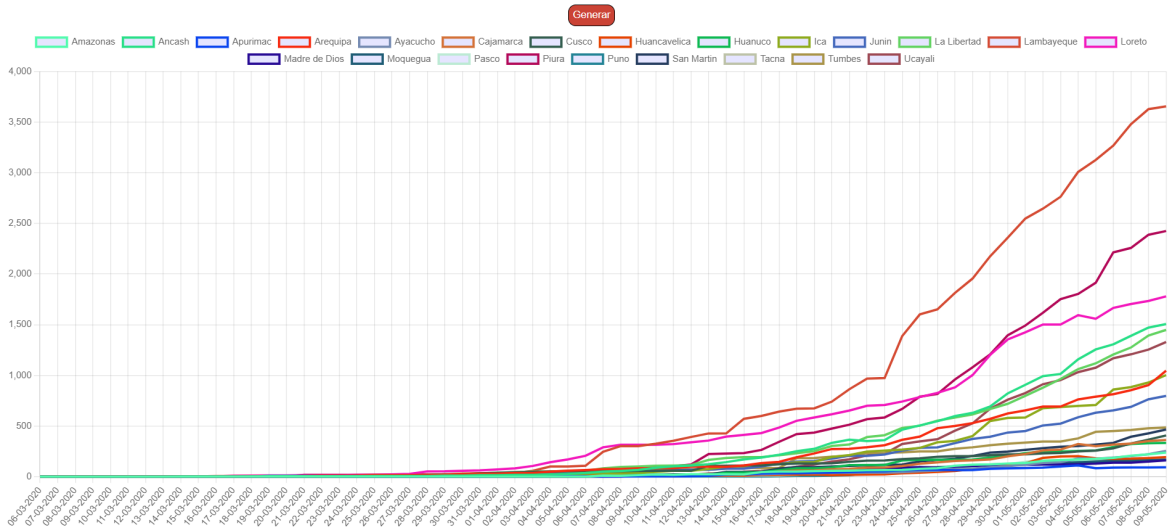

Figura 83: Ejercicio 6

6.5.7. Ejercicio 7

- **HTML**

```html
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title>Ejercicio 7</title>
    <meta charset="UTF-8"/>
    <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
    <meta name="description" content="Usando ajax para la comparación de gráficas"/>
    <meta name="keywords" content="comparación, ajax, gráfica"/>
    <link rel="stylesheet" href="http://localhost/nuevo/css/estilos.css">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body class="cuerpo">
    <div>
      <form id="comparasion">
        <center>
          <label for="region1">Región 1:</label>
          <select id="region1" name="region1">
            <option value="Amazonas">Amazonas</option>
            <option value="Ancash">Ancash</option>
            <option value="Apurimac">Apurímac</option>
            <option value="Arequipa">Arequipa</option>
            <option value="Ayacucho">Ayacucho</option>
            <option value="Cajamarca">Cajamarca</option>
            <option value="Callao">Callao</option>
            <option value="Cusco">Cusco</option>
            <option value="Huancavelica">Huancavelica</option>
            <option value="Huanuco">Huánuco</option>
            <option value="Ica">Ica</option>
            <option value="Junin">Junín</option>
            <option value="La Libertad">La Libertad</option>
            <option value="Lambayeque">Lambayeque</option>
            <option value="Lima">Lima</option>
            <option value="Loreto">Loreto</option>
            <option value="Madre de Dios">Madre de Dios</option>
            <option value="Moquegua">Moquegua</option>
            <option value="Pasco">Pasco</option>
            <option value="Piura">Piura</option>
            <option value="Puno">Puno</option>
            <option value="San Martin">San Martín</option>
            <option value="Tacna">Tacna</option>
            <option value="Tumbes">Tumbes</option>
            <option value="Ucayali">Ucayali</option>
          </select>
```

Figura 84: HTML 7 - 1

```
44            <label for="region2"> Región 2:</label>
45            <select id="region2" name="region2">
46              <option value="Amazonas">Amazonas</option>
47              <option value="Ancash">Ancash</option>
48              <option value="Apurimac">Apurímac</option>
49              <option value="Arequipa">Arequipa</option>
50              <option value="Ayacucho">Ayacucho</option>
51              <option value="Cajamarca">Cajamarca</option>
52              <option value="Callao">Callao</option>
53              <option value="Cusco">Cusco</option>
54              <option value="Huancavelica">Huancavelica</option>
55              <option value="Huanuco">Huánuco</option>
56              <option value="Ica">Ica</option>
57              <option value="Junin">Junín</option>
58              <option value="La Libertad">La Libertad</option>
59              <option value="Lambayeque">Lambayeque</option>
60              <option value="Lima">Lima</option>
61              <option value="Loreto">Loreto</option>
62              <option value="Madre de Dios">Madre de Dios</option>
63              <option value="Moquegua">Moquegua</option>
64              <option value="Pasco">Pasco</option>
65              <option value="Piura">Piura</option>
66              <option value="Puno">Puno</option>
67              <option value="San Martin">San Martín</option>
68              <option value="Tacna">Tacna</option>
69              <option value="Tumbes">Tumbes</option>
70              <option value="Ucayali">Ucayali</option>
71            </select>
72            <button class="Generar" type="submit">Generar</button>
73          </center>
74        </form>
75      </div>
76      <div>
77          <canvas id="grafica" width="800" height="650"></canvas>
78      </div>
79      <script src="Script.js"></script>
80    </body>
81  </html>
```

Figura 85: HTML 7 - 2

- **Script**

```
let myChart;
document.getElementById("comparasion").addEventListener("submit", function (evento) {
  evento.preventDefault();
  //Extraccion de datos del formulario
  const formData = new FormData(this);
  const region1 = formData.get('region1');
  const region2 = formData.get('region2');

  fetch("http://localhost/nuevo/JASON/data.json")
    .then(response => response.json())
    .then(data => {
      //Solicitando informacion del archivo data.json
      const datosRegion1 = data.find(region => region.region === region1);
      const datosRegion2 = data.find(region => region.region === region2);

      const fechas = datosRegion1.confirmed.map(entry => entry.date);
      const valoresRegion1 = datosRegion1.confirmed.map(entry => parseInt(entry.value));
      const valoresRegion2 = datosRegion2.confirmed.map(entry => parseInt(entry.value));
      if (myChart) {
        //Si el archivo ya fue creado.
        myChart.data.labels = fechas;
        myChart.data.datasets[0].label = region1;
        myChart.data.datasets[0].data = valoresRegion1;
        myChart.data.datasets[1].label = region2;
        myChart.data.datasets[1].data = valoresRegion2;
        myChart.update();
      } else {
        //caso contrario la crea
        const ctx = document.getElementById('grafica').getContext('2d');
        myChart = new Chart(ctx, {
          type: 'line',
          data: {
            labels: fechas,
```

Figura 86: Script 7-1

```
34              datasets: [
35                {
36                  label: region1,
37                  data: valoresRegion1,
38                  borderColor: 'blue',
39                  backgroundColor: 'rgba(0, 0, 255, 0.1)'
40                },
41                {
42                  label: region2,
43                  data: valoresRegion2,
44                  borderColor: 'green',
45                  backgroundColor: 'rgba(0, 255, 0, 0.1)'
46                }
47              ]
48            },
49            options: {
50              responsive: true,
51              maintainAspectRatio: false
52            }
53          });
54        }
55      })
56      .catch(error => console.error('Error:', error));
57  });
```

Figura 87: Script 7 - 2

- **Ejecución**

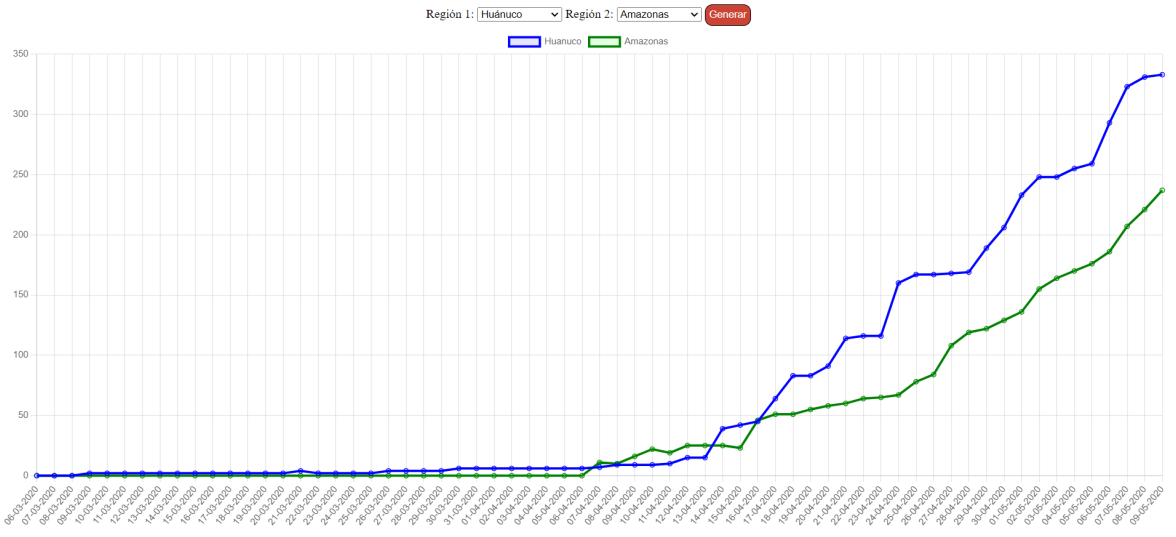

Figura 88: Ejercicio 7

6.5.8. Ejercicio 8

- **HTML**

```html
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title>Ejercicio 8</title>
    <meta charset="UTF-8"/>
    <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
    <meta name="description" content="Usando ajax para la comparación de gráficas"/>
    <meta name="keywords" content="comparación, ajax, gráfica"/>
    <link rel="stylesheet" href="http://localhost/nuevo/css/estilos.css">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body class="cuerpo">
    <div>
      <form id="comparasion">
      <center>
        <button class="Generar" type="submit">Generar</button>
      </center>
    </div>
    <div>
        <canvas id="grafica" width="800" height="650"></canvas>
    </div>
    <script src="Script.js"></script>
  </body>
</html>
```

Figura 89: HTML 8

- **Script**

```javascript
document.getElementById("comparasion").addEventListener("submit", function (evento) {
  evento.preventDefault();

  fetch("http://localhost/nuevo/JASON/data.json")
    .then(response => response.json())
    .then(data => {
      //extrayendo primera región
      const primeraRegion = data.find(region => region.region !== 'Lima' && region.region !== 'Callao');

      // Verificar que se encontró la primera región
      if (!primeraRegion) {
        console.error('No se encontró la primera región con las fechas necesarias para el gráfico.');
        return;
      }

      // Obtener las fechas de la primera región
      const fechas = primeraRegion.confirmed.map(entry => entry.date);

      // Extraer datos  de las regiones menos Lima y Callao
      const datasets = data.filter(region => region.region !== 'Lima' && region.region !== 'Callao').map(region => {
        const valores = [];
        // Llenar los valores con los datos correspondientes a las fechas de la primera región
        fechas.forEach(fecha => {
          const dato = region.confirmed.find(entry => entry.date === fecha);
          if (dato) {
            valores.push(parseInt(dato.value));
          } else {
            valores.push(null);
          }
        });
        return {
          label: region.region,
          data: valores,
          borderColor: getRandomColor(),
          backgroundColor: 'rgba(0, 0, 255, 0.1)'
        };
      });
```

Figura 90: Script 8 - 1

```
39        // Crear el gráfico con los datos de todas las regiones
40        const ctx = document.getElementById('grafica').getContext('2d');
41        myChart = new Chart(ctx, {
42          type: 'line',
43          data: {
44            labels: fechas,
45            datasets: datasets
46          },
47          options: {
48            responsive: true,
49            maintainAspectRatio: false,
50            scales: {
51              y: {
52                beginAtZero: true
53              }
54            }
55          }
56        });
57      })
58      .catch(error => console.error('Error:', error));
59  });
60
61  // Función para obtener un color aleatorio
62  function getRandomColor() {
63    const letters = '0123456789ABCDEF';
64    let color = '#';
65    for (let i = 0; i < 6; i++) {
66      color += letters[Math.floor(Math.random() * 16)];
67    }
68    return color;
69  }
```

Figura 91: Script 8 - 2

- **Ejecución**

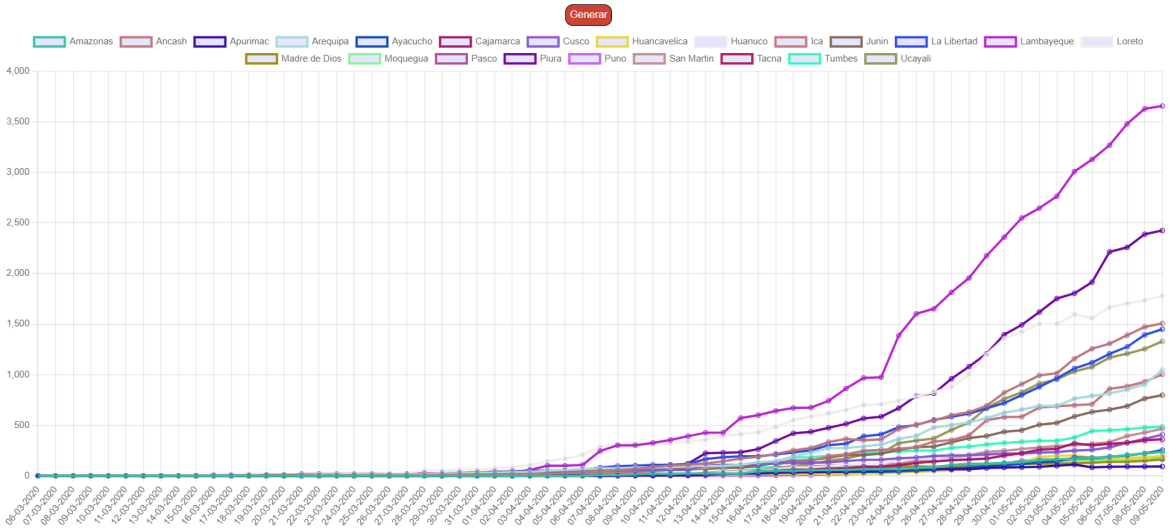

Figura 92: Ejercicio 8

6.5.9. CSS

- **CSS**

```css
.Generar{
    border-radius: 10px;
    background: #CB4335;
    padding: 5px;
    color: white;
    border: 1px solid black;
    margin-bottom: 5px;
}
.Generar:hover{
    background: #EC7063;
}
```

Figura 93: Estilos ejercicio 4 - 8

6.6. Ejercicio Propuesto: Página Web - Markdown

- **HTML**

```html
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/estilos.css">
    <title>Markdown</title>
  </head>
  <body>
    <h1>MARKDOWN</h1>
    <form id="markupForm">
      <label for="markupTitle">Ingrese titulo</label><br>
      <input type="text" id="markupTitle" name="markupTitle"/><br>
      <label for="markupText">Ingrese el texto</label><br>
      <textarea id="markupText" name="markupText" rows="10" cols="50"></textarea><br>
      <input class="botones" type="submit" value="Enviar">
    </form>
    <form id="markupList">
      <input class="botones" type="submit" value="Listar">
    </form>
    <h2>Resultado:</h2>
    <div id="htmlCode"></div>
    <script src="app.js"></script>
  </body>
</html>
```

Figura 94: HTML

- **Script 1**

```
const fs = require('fs');
const path = require('path');
const express = require('express');
const bp = require('body-parser');
const MarkdownIt = require('markdown-it');
const md = new MarkdownIt();
const app = express();

let archivosGuardados = [];
app.use(express.static('pub'));
app.use(bp.json());
app.use(bp.urlencoded({ extended: true }));

app.listen(3000, () => {
    console.log("Escuchando en: http://localhost:3000");
});

app.get('/', (request, response) => {
    response.sendFile(path.resolve(__dirname, 'index.html'));
});

app.post('/crear', (request, response) => {
  console.log(request.body);
  const markDownText = request.body.text;
  const titulo = request.body.title;
  let htmlCode = md.render(markDownText);
  response.setHeader('Content-Type', 'application/json');
  response.end(JSON.stringify({ text: htmlCode, title: "TITULO: " + titulo}));
  let archivo = {
      nombre: titulo,
      text: markDownText,
      html: htmlCode,
      fecha: new Date().toISOString()
  };
  archivosGuardados.push(archivo);
  fs.writeFile('pub/archivos/Markdown.json',JSON.stringify(archivosGuardados), (err) => {
    if (err) {
      console.error('Error al guardar el archivo JSON:', err);
    } else {
      console.log('Información de archivos guardados actualizada en JSON.');
    }
  });
});
app.get('/listar', (request, response) => {
    response.setHeader('Content-Type', 'application/json');
    response.end(JSON.stringify(archivosGuardados.map(archivo => archivo.nombre)));
});
```

Figura 95: Script index.js 1

```javascript
function crear(markupText, markupTitle) {
  const url = 'http://localhost:3000/crear'
  const data = {
    title: markupTitle,
    text: markupText
  }
  console.log(data);
  const request = {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify(data),
  }
  fetch(url, request)
    .then(response => response.json())
    .then(data => {
      document.querySelector("#htmlCode").innerHTML = "<h1>" + data.title + "</h1><br>" + data.text;
    })
    .catch(error => {
      console.error('Error al enviar la solicitud:', error);
    });
}
//listar todas los nombres markdown en un boton
function listar(){
  const url = 'http://localhost:3000/listar';
  fetch(url)
    .then(response => response.json())
    .then(data => {
      const archivosList = document.querySelector("#htmlCode");
      archivosList.innerHTML = '';
      data.forEach(nombre => {
        const button = document.createElement('button');
        button.textContent = nombre;
        button.classList.add('botones');
        button.addEventListener('click', () => {
          mostrarHtml(nombre);
        });
        archivosList.appendChild(button);
        const saltoLinea = document.createElement('br');
        archivosList.appendChild(saltoLinea);
      });
    })
    .catch(error => {
    console.error('Error al obtener la lista de archivos:', error);
    });
}
```

Figura 96: Script app.js 2 - 1

```
48      //funcion para colocar el contenido html al presionar el boton generado en la funcion listar()
49      function mostrarHtml(nombreArchivo) {
50        const url = 'http://localhost:3000/archivos/Markdown.json'; // Ruta del archivo JSON
51        fetch(url)
52          .then(response => response.json())
53          .then(data => {
54            const archivo = data.find(archivo => archivo.nombre === nombreArchivo);
55            if (archivo) {
56              const htmlCode = document.querySelector("#htmlCode");
57              htmlCode.innerHTML = `<h1>${archivo.nombre}</h1><br>${archivo.html}`;
58            } else {
59              console.error('Archivo no encontrado en el JSON.');
60            }
61          })
62          .catch(error => {
63          console.error('Error al obtener el JSON de archivos:', error);
64          });
65      }
66      //Espera a que la página se cargue por completa
67      document.addEventListener('DOMContentLoaded', function () {
68        const title = document.querySelector('#markupTitle');
69        const text = document.querySelector('#markupText');
70        //boton para colocar en el archivo data.json, los datos respectivos
71        document.querySelector('#markupForm').onsubmit = () => {
72          event.preventDefault();
73          crear(text.value, title.value);
74          return false;
75        }
76        //boton para generar la lista
77        document.querySelector('#markupList').onsubmit = () => {
78          event.preventDefault();
79          listar();
80          return false;
81        }
82      })
```

Figura 97: Script app.js 2 - 2

- **Ejecución**

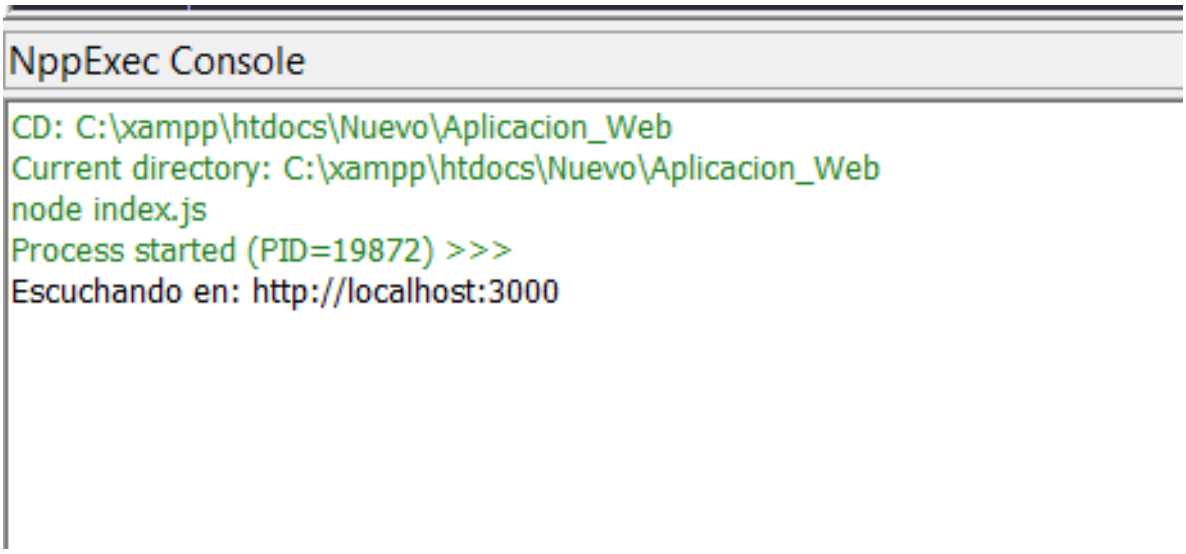

Figura 98: Servidor Activado

Figura 99: Botón Enviar

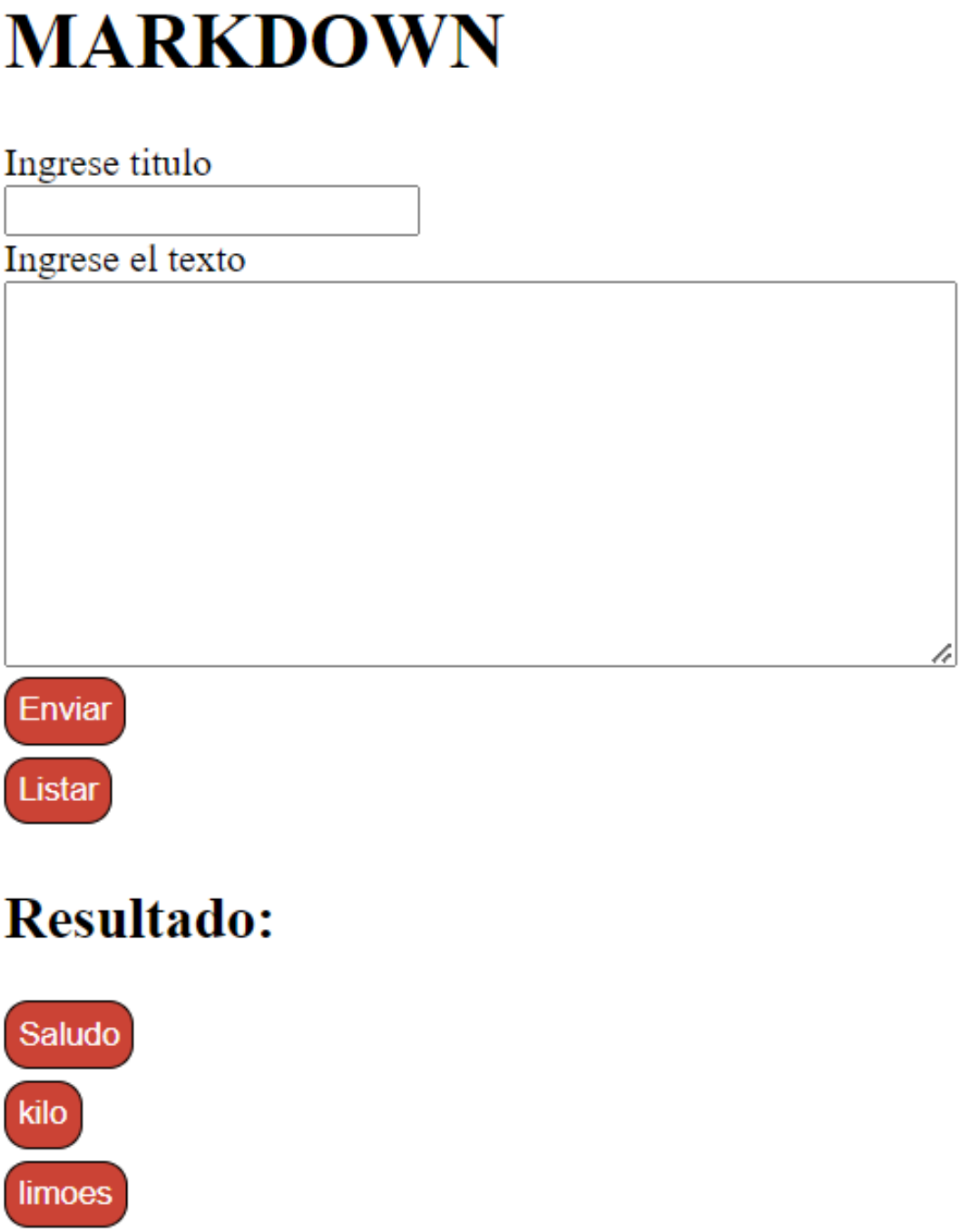

Figura 100: Botón Listar

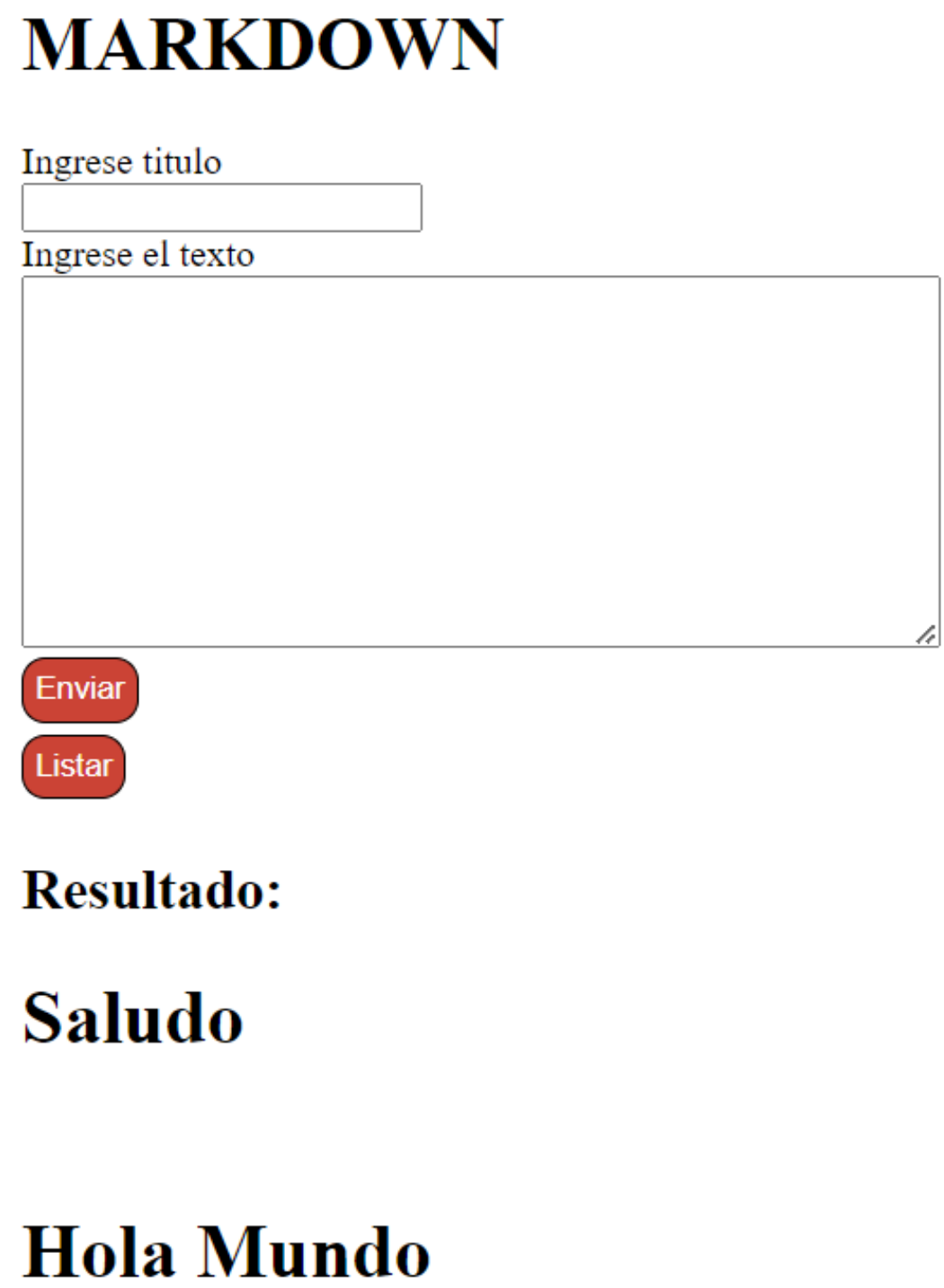

Figura 101: Mostrar página de la Lista

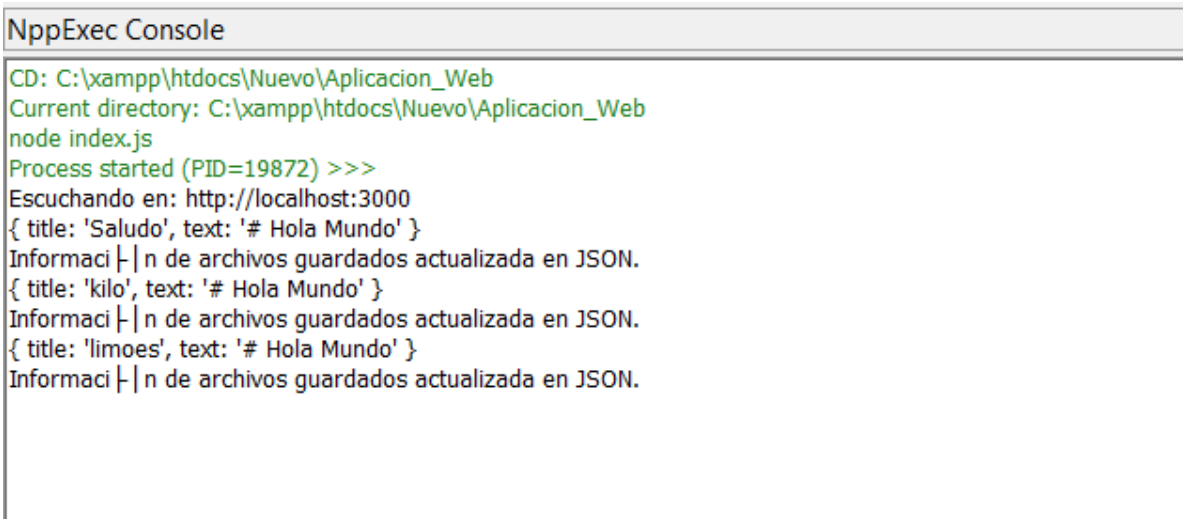

Figura 102: console.log

- **CSS**

```css
.botones{
    border-radius: 10px;
    background: #CB4335;
    padding: 5px;
    color: white;
    border: 1px solid black;
    margin-bottom: 5px;
}
.botones:hover{
    background: #EC7063;
}
```

Figura 103: Estilos de Página Markdown

6.7. Uso de GitHub

6.7.1. Cuenta de GitHub

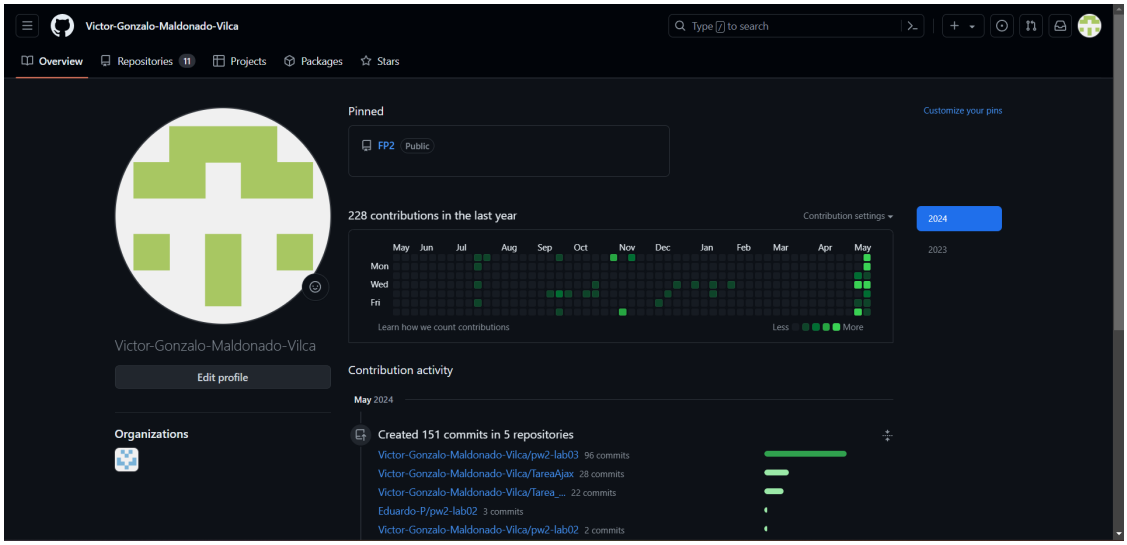

Figura 104: Usuario

6.8. Creación de un Nuevo Repositorio

6.8.1. Implementación de Readme.md

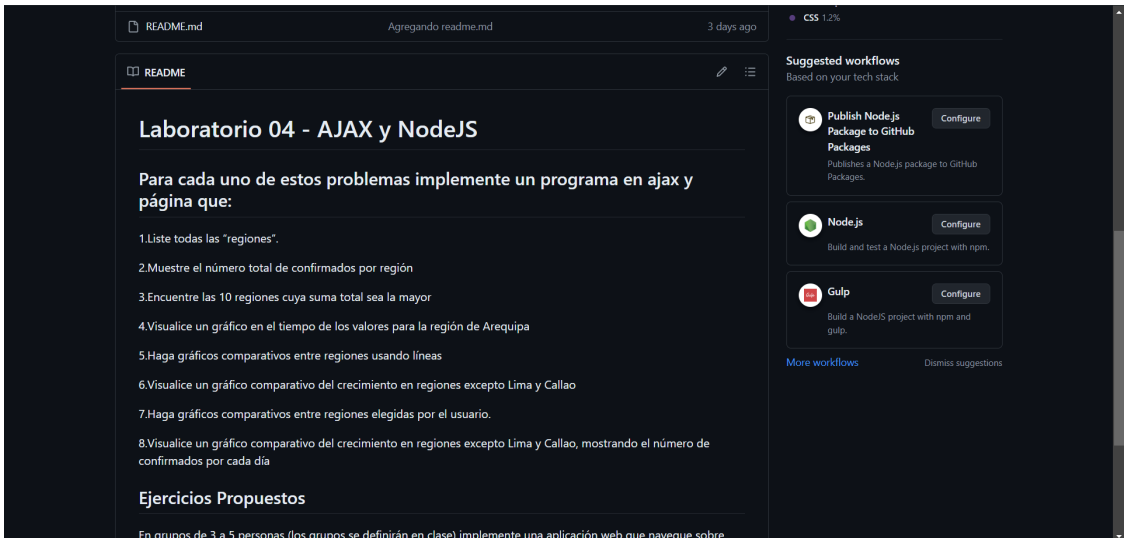

Figura 105: Readme.md

6.8.2. Registro de cambios en mi código

- **Commits**

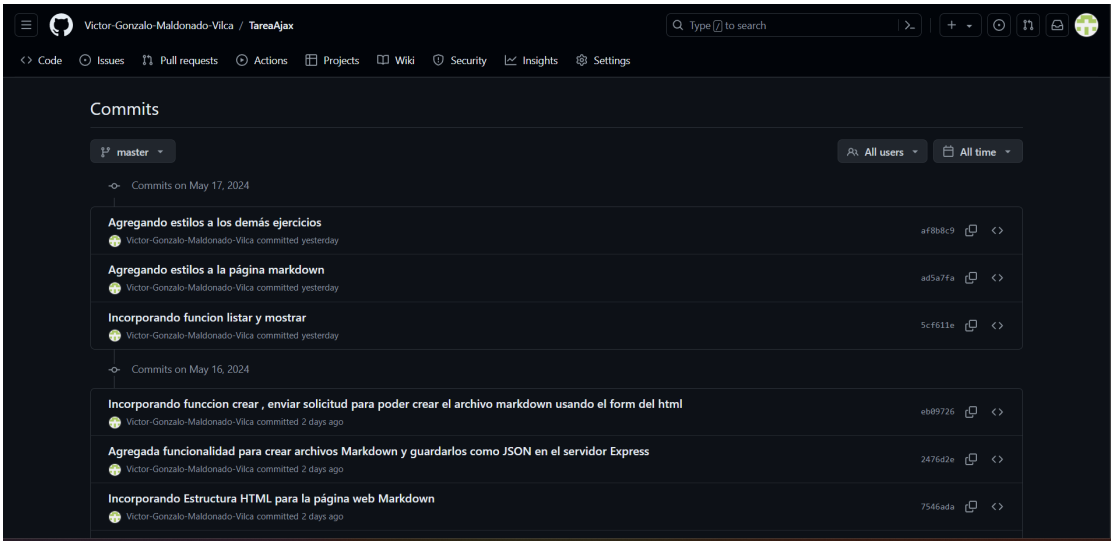

Figura 106: Commits

6.8.3. Repositorio

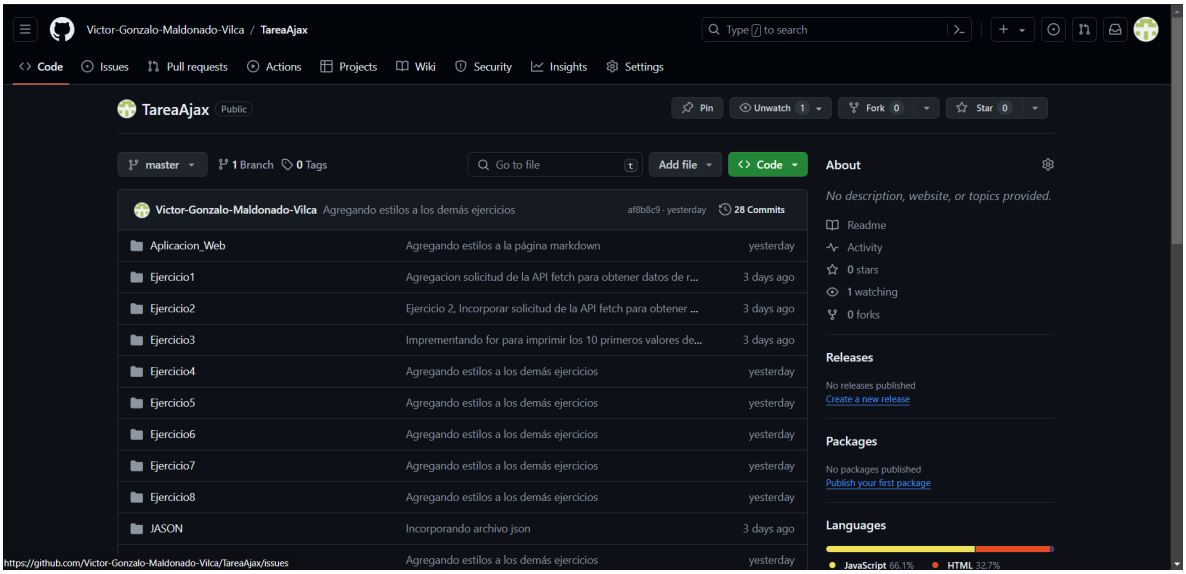

Figura 107: Repositorio

6.8.4. Proyecto compartido con el profesor de github

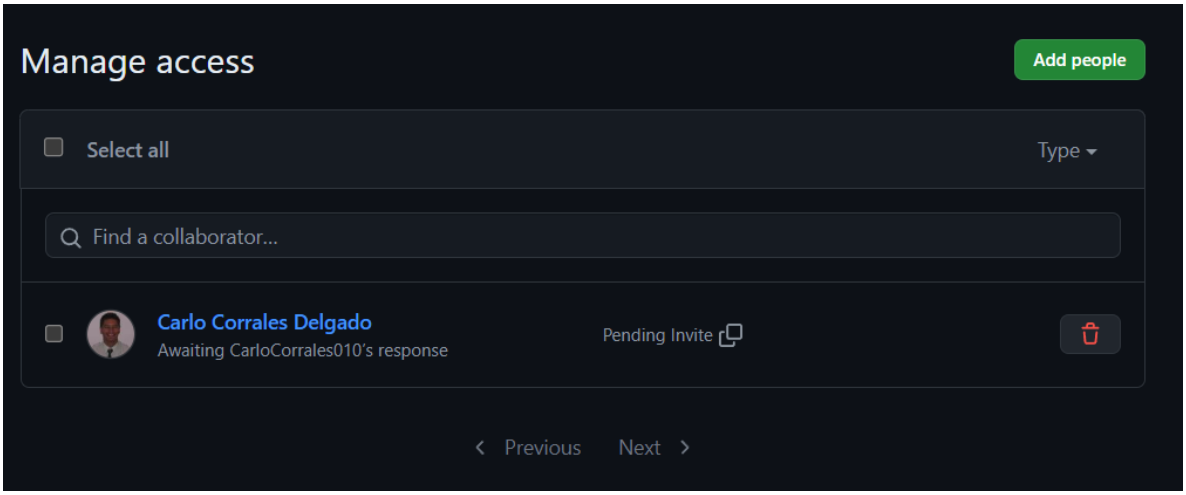

Figura 108: Compartir con el Profesor

6.9. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumplio 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 lo items.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

	Nivel			
Puntos	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 2: 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	4	
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	2	
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 estan 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		20	

7. Referencias

- https://www.chartjs.org/
- https://www.chartjs.org/docs/latest/
- https://git-scm.com/doc
- https://www.w3schools.com/js/default.asp