



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 1

INFORME DE ACTIVIDAD

INFORMACIÓN BÁSICA					
ASIGNATURA:	<i>PW2 - Laboratorio</i>				
TÍTULO DE LA PRÁCTICA:	<i>Ejercicios resueltos y ejercicios AJAX en wetschools</i>				
NÚMERO DE ACTIVIDAD:	<i>Laboratorio 4</i>	AÑO LECTIVO:	<i>2025-A</i>	NRO. SEMESTRE:	<i>III</i>
FECHA DE PRESENTACIÓN	<i>25/04/25</i>	HORA DE PRESENTACIÓN	<i>23:30</i>	NOTA (0-20)	
INTEGRANTE (s)	<i>Renzo Geomar Mamani Quispe</i>			DOCENTE(s):	
	<i>Ing. CORRALES DELGADO, CARLO JOSE LUIS</i>				

RESULTADOS Y PRUEBAS



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 2

I. PRUEBAS

Ejercicios resueltos:

Ejercicio 1:

index.html:

```
ejercicioResuelto1 > pub > < index.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1.0">
6          <title>Mi Aplicación Web</title>
7      </head>
8      <body>
9          <h1>¡Bienvenido a mi aplicación!</h1>
10         <p>Este es un ejemplo de página web servida por Node.js y Express.</p>
11         <p id="mensaje"></p>
12         <script>
13             document.getElementById('mensaje').innerText = "Este JavaScript se ejecuta en el cliente.";
14         </script>
15     </body>
16 </html>
```

index.js:

```
ejercicioResuelto1 > js index.js > ...
1  const path = require('path');
2  const express = require('express');
3  const app = express();
4  app.use(express.static('pub'));

5
6  app.listen(3000, () => {
7      console.log("Escuchando en: http://localhost:3000")
8
9 });
10
11 app.get('/', (request, response) => {
12     response.sendFile(path.resolve(__dirname, 'pub', 'index.html'));
13 });
14
```

Pruebas:

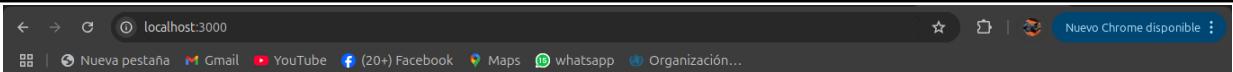


Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 3



¡Bienvenido a mi aplicación!

Este es un ejemplo de página web servida por Node.js y Express.

Este JavaScript se ejecuta en el cliente.



Ejercicio 2:

index.html:

```
ejercicioResuelto2 > ◊ index.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Mi Aplicación Web</title>
7  </head>
8  <body>
9      <h1>¡Bienvenido a mi aplicación!</h1>
10     <p>Probando el ejercicio 2</p>
11     <button onclick="recitar()">Recitar poema</button>
12     <div id="poema" style="margin-top: 20px;"></div>
13     <script>
14         // En el lado del cliente
15         function recitar() {
16             const url = 'http://localhost:3000/recitar'
17             fetch(url).then(
18                 response => response.json()
19             ).then(
20                 data => {
21                     document.querySelector("#poema").innerHTML = data.text
22                 }
23             )
24         }
25     </script>
26  </body>
27  </html>
```

node.js:

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 4

```
ejercicioResuelto2 > js node.js > ...
2  const fs = require('fs')
3  const path = require('path')
4  const express = require('express')
5  const app = express()
6  app.use(express.static('pub'))
7
8  app.listen(3000, () => {
9    console.log("Escuchando en: http://localhost:3000")
10 })
11
12 app.get('/', (request, response) => {
13   response.sendFile(path.resolve(__dirname, 'index.html'))
14 })
15
16 app.get('/recitar', (request, response) => {
17   fs.readFile(path.resolve(__dirname, 'priv/poema.txt'), 'utf8',
18   (err, data) => {
19     if (err) {
20       console.error(err)
21       response.status(500).json({
22         error: 'message'
23       })
24       return
25     }
26     response.json({
27       text: data.replace(/\n/g, '<br>')
28     })
29   })
30 })
```

poema.txt:

```
poema.txt x
ejercicioResuelto2 > priv > poema.txt
1 Me gustas cuando callas porque estás como ausente,
2 y me oyes desde lejos, y mi voz no te toca.
3 Parece que los ojos se te hubieran volado
4 y parece que un beso te cerrara la boca.
5
6 Como todas las cosas están llenas de mi alma
7 emerges de las cosas, llena del alma mía.
8 Mariposa de sueño, te pareces a mi alma,
9 y te pareces a la palabra melancolía.
10
11 Me gustas cuando callas y estás como distante.
12 Y estás como quejándote, mariposa en arrullo.
13 Y me oyes desde lejos, y mi voz no te alcanza:
14 déjame que me calle con el silencio tuyo.
15
16 Déjame que te hable también con tu silencio
17 claro como una lámpara, simple como un anillo.
18 Eres como la noche, callada y constelada.
19 Tu silencio es de estrella, tan lejano y sencillo.
20
21 Me gustas cuando callas porque estás como ausente.
22 Distante y dolorosa como si hubieras muerto.
23 Una palabra entonces, una sonrisa bastan.
24 Y estoy alegre, alegre de que no sea cierto.
```



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 5

Pruebas:

The screenshot shows a web browser window with the URL `localhost:3000`. The page title is "¡Bienvenido a mi aplicación!". Below it, there is a heading "Probando el ejercicio 2" and a button labeled "Recitar poema". The main content area contains two poems. The first poem is:

Me gustas cuando callas porque estás como ausente,
y me oyes desde lejos, y mi voz no te toca.
Parece que los ojos se te hubieran volado
y parece que un beso te cerrara la boca.

The second poem is:

Como todas las cosas están llenas de mi alma
emerges de las cosas, llena del alma mía.
Mariposa de sueño, te pareces a mi alma,
y te pareces a la palabra melancolía.

Below these poems, there is another section:

Me gustas cuando callas y estás como distante.
Y estás como quejándote, mariposa en arrullo.
Y me oyes desde lejos, y mi voz no te alcanza:
déjame que me calle con el silencio tuyo.

At the bottom of the page, there is a "Recitar" button. The browser's address bar shows `localhost:3000`, and the taskbar at the bottom of the screen shows various application icons.

Ejercicio 3:

index.html:

```
ejercicioResuelto3 > index.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Mi Aplicación Web</title>
7  </head>
8  <body>
9      <h1>¡Bienvenido a mi aplicación!</h1>
10     <p>Probando el ejercicio 3</p>
11     <form id="markupForm">
12         <input type="text" id="markupText" placeholder="Escribe tu texto aquí" required>
13         <button type="submit">Recitar</button>
14     </form>
15     <div id="htmlCode" style="margin-top: 20px; border: 1px solid #ccc; padding: 10px;"></div>
16     <script>
17         // En el lado del cliente
18         function recitar(markupText) {
19             const url = 'http://localhost:3000/';
20             const data = {
21                 text: markupText
22             }
23             console.log(data)
24             const request = {
25                 method: 'POST', // Podría ser GET
26                 headers: {
27                     'Content-Type': 'application/json',
28                 },
29                 body: JSON.stringify(data)
30             }
31             fetch(url, request)
32             .then(response => response.text())
33             .then(text => document.getElementById('htmlCode').innerHTML = text)
34         }
35     </script>
```

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 6

```
29         body: JSON.stringify(data),
30     }
31     http = fetch(url, request)
32     http.then(
33         response => response.json()
34     ).then(
35         data => {
36             document.querySelector("#htmlCode").innerHTML = data.text
37         }
38     )
39 }
40 document.addEventListener('DOMContentLoaded', function () {
41     const text = document.querySelector('#markupText')
42     document.querySelector('#markupForm').onsubmit = () => {
43         recitar(text.value)
44         return false;
45     }
46 })
47 </script>
48 </body>
49 </html>
```

node.js:

```
ejercicioResuelto3 > js node.js > ...
1 // En el lado del servidor
2 const fs = require('fs')
3 const path = require('path')
4 const express = require('express')
5 const bp = require('body-parser')
6 const MarkdownIt = require('markdown-it'),
7     md = new MarkdownIt();
8 const app = express()
9
10 app.use(express.static('pub'))
11 app.use(bp.json())
12 app.use(bp.urlencoded({
13     extended: true
14 }))
15
16 app.listen(3000, () => {
17     console.log("Escuchando en: http://localhost:3000")
18 })
19
20 app.get('/', (request, response) => {
21     response.sendFile(path.resolve(__dirname, 'index.html'))
22 })
23
24 app.post('/', (request, response) => {
25     console.log(request.body)
26     let markDownText = request.body.text
27     console.log(markDownText)
28     let htmlText = md.render(markDownText)
29     response.setHeader('Content-Type', 'application/json')
```



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

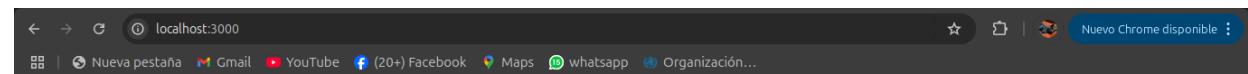
Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 7

```
26     let htmlText = `<h1>${data}</h1>`;
27     response.setHeader('Content-Type', 'application/json');
28     response.end(JSON.stringify({
29       text: htmlText
30     }));
31   });
32 }
33 })
34 }
```

Pruebas:



¡Bienvenido a mi aplicación!

Probando el ejercicio 3

hola mundo Recitar

hola mundo



Tarea Ajax en w3schools:

Cree una solicitud XMLHttpRequest simple y recupere datos de un archivo TXT:
`index.html`:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 8

Tarea Ajax en w3schools > ejercicio1 > index.html > ...

```
1  <!DOCTYPE html>
2  <html>
3  |   <body>
4  |
5  |       <div id="demo">
6  |           <h2>The XMLHttpRequest Object</h2>
7  |           <button type="button" onclick="loadDoc()">Change Content</button>
8  |       </div>
9  |
10    <script>
11        function loadDoc() {
12            const xhttp = new XMLHttpRequest();
13            xhttp.onload = function() {
14                document.getElementById("demo").innerHTML =
15                    this.responseText.replace(/\n/g, "<br>");
16            }
17            xhttp.open("GET", "ajax_info.txt");
18            xhttp.send();
19        }
20    </script>
21
22    </body>
23 </html>
```

node.js:

Tarea Ajax en w3schools > ejercicio1 > node.js > ...

```
1  const path = require('path');
2  const express = require('express');
3  const app = express();
4
5  // Servir archivos estáticos desde el directorio actual
6  app.use(express.static(__dirname));
7
8  app.listen(3000, () => {
9      console.log('Servidor escuchando en http://localhost:3000');
10 });
11
```

ajax_info.txt:

Tarea Ajax en w3schools > ejercicio1 > ajax_info.txt

```
1  AJAX
2  AJAX no es un lenguaje de programación.
3
4  AJAX es una técnica para acceder a servidores web desde una página web.
5
6  AJAX son las siglas de JavaScript asíncrono y XML.
```

Pruebas:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 9

AJAX
AJAX no es un lenguaje de programación.
AJAX es una técnica para acceder a servidores web desde una página web.
AJAX son las siglas de JavaScript asincrónicas y XML.



Cree una solicitud XMLHttpRequest con una función de devolución de llamada y recupere datos de un archivo TXT:

index.html:

node.js:

```
Tarea Ajax en w3schools > ejercicio1 > js node.js > ...
1 const path = require('path');
2 const express = require('express');
3 const app = express();
4
5 // Servir archivos estáticos desde el directorio actual
6 app.use(express.static(__dirname));
7
8 app.listen(3000, () => {
9   console.log('Servidor escuchando en http://localhost:3000');
10 });
11
```

ajax_info.txt:

```
Tarea Ajax en w3schools > ejercicio1 > ajax_info.txt
1 AJAX
2 AJAX no es un lenguaje de programación.
3
4 AJAX es una técnica para acceder a servidores web desde una página web.
5
6 AJAX son las siglas de JavaScript asincrónico y XML.
```



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 10

Pruebas:

AJAX
AJAX no es un lenguaje de programación.
AJAX es una técnica para acceder a servidores web desde una página web.
AJAX son las siglas de JavaScript asincrónicas y XML.

Recuperar toda la información del encabezado de un recurso(archivo):

index.html:

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <h2>The XMLHttpRequest Object</h2>
5          <p>The getAllResponseHeaders() function returns all the header information of a resource,
6              like length, server-type, content-type, last-modified, etc:</p>
7          <pre id="demo"></pre>
8          <script>
9              const xhttp = new XMLHttpRequest();
10             xhttp.onload = function() {
11                 document.getElementById("demo").innerHTML =
12                     this.getAllResponseHeaders();
13             }
14             xhttp.open("GET", "ajax_info.txt");
15             xhttp.send();
16         </script>
17
18     </body>
19 </html>
```

node.js:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 11

Tarea Ajax en w3schools > ejercicio1 > **js node.js** > ...

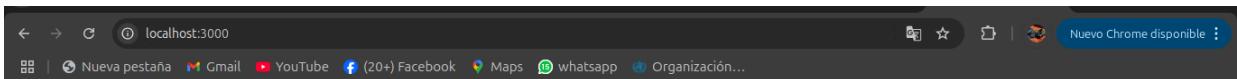
```
1 const path = require('path');
2 const express = require('express');
3 const app = express();
4
5 // Servir archivos estáticos desde el directorio actual
6 app.use(express.static(__dirname));
7
8 app.listen(3000, () => {
9   console.log('Servidor escuchando en http://localhost:3000');
10 });
11
```

ajax_info.txt:

Tarea Ajax en w3schools > ejercicio1 > ajax_info.txt

```
1 AJAX
2 AJAX no es un lenguaje de programación.
3
4 AJAX es una técnica para acceder a servidores web desde una página web.
5
6 AJAX son las siglas de JavaScript asíncrono y XML.
```

Pruebas:



The XMLHttpRequest Object

The getAllResponseHeaders() function returns all the header information of a resource, like length, server-type, content-type, last-modified, etc:

```
accept-ranges: bytes
cache-control: public, max-age=0
content-length: 175
content-type: text/plain; charset=utf-8
date: Tue, 06 May 2025 16:18:21 GMT
etag: W/"af-196a5bbbd31"
last-modified: Tue, 06 May 2025 13:14:26 GMT
x-powered-by: Express
```



Recuperar información específica del encabezado de un recurso(archivo):
index.html:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 12

Tarea Ajax en w3schools > ejercicio4 > index.html > ...

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>The XMLHttpRequest Object</h2>
5     <p>The getResponseHeader() function is used to return specific header information from a resource, like
6     <p>Last modified: <span id="demo"></span></p>
7     <script>
8       const xhttp=new XMLHttpRequest();
9       xhttp.onload = function() {
10         document.getElementById("demo").innerHTML =
11           // indica cuándo fue modificado por última vez ese archivo en el servidor
12           this.getResponseHeader("Last-Modified");
13       }
14       xhttp.open("GET", "ajax_info.txt");
15       xhttp.send();
16     </script>
17   </body>
18 </html>
```

node.js:

Tarea Ajax en w3schools > ejercicio1 > node.js > ...

```
1 const path = require('path');
2 const express = require('express');
3 const app = express();
4
5 // Servir archivos estáticos desde el directorio actual
6 app.use(express.static(__dirname));
7
8 app.listen(3000, () => {
9   console.log('Servidor escuchando en http://localhost:3000');
10 });
11
```

ajax_info.txt:

Tarea Ajax en w3schools > ejercicio1 > ajax_info.txt

```
1 AJAX
2 AJAX no es un lenguaje de programación.
3
4 AJAX es una técnica para acceder a servidores web desde una página web.
5
6 AJAX son las siglas de JavaScript asincrónico y XML.
```

Pruebas:

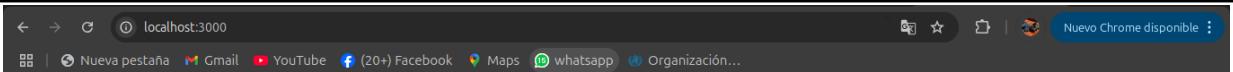


Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 13



El objeto XMLHttpRequest

La función `getResponseHeader()` se utiliza para devolver información de encabezado específica de un recurso, como longitud, tipo de servidor, tipo de contenido, última modificación, etc.:

Última modificación: martes, 6 de mayo de 2025, 13:14:26 GMT



Cargar un archivo XML con AJAX:

index.html:

```
Tarea Ajax en w3schools > ejercicio5 > index.html > ...
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <h2>The XMLHttpRequest Object</h2>
5          <h2>Retrieve data from XML file</h2>
6          <p><b>Status:</b> <span id="A1"></span></p>
7          <p><b>Status text:</b> <span id="A2"></span></p>
8          <p><b>Response:</b> <span id="A3"></span></p>
9          <button onclick="loadDoc('note.xml')">Get XML data</button>
10         <script>
11             function loadDoc(url) {
12                 const xhttp = new XMLHttpRequest();
13                 xhttp.onload = function() {
14                     document.getElementById('A1').innerHTML = this.status; //código de estado de la respuesta
15                     document.getElementById('A2').innerHTML = this.statusText; //El texto del estado
16                     document.getElementById('A3').innerHTML = this.responseText; //contenido completo del archivo
17                 }
18                 xhttp.open("GET", url);
19                 xhttp.send();
20             }
21         </script>
22     </body>
23 </html>
```

node.js:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 14

Tarea Ajax en w3schools > ejercicio1 > `js node.js` > ...

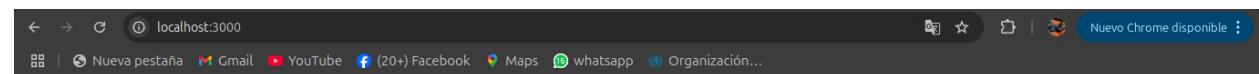
```
1 const path = require('path');
2 const express = require('express');
3 const app = express();
4
5 // Servir archivos estáticos desde el directorio actual
6 app.use(express.static(__dirname));
7
8 app.listen(3000, () => {
9   console.log('Servidor escuchando en http://localhost:3000');
10 });
11
```

note.xml:

Tarea Ajax en w3schools > ejercicio5 > `note.xml`

```
1 <note>
2   <to>Usuario</to>
3   <from>Servidor</from>
4   <heading>Recordatorio</heading>
5   <body>Este es un mensaje XML de prueba</body>
6 </note>
```

Pruebas:



The XMLHttpRequest Object

Retrieve data from XML file

Status: 200

Status text: OK

Response: Usuario Servidor Recordatorio Este es un mensaje XML de prueba



Recuperar el contenido de un archivo XML:

index.html:



```
Tarea Ajax en w3schools > ejercicio6 > index.html > ...
1  <!DOCTYPE html>
2  <html>
3      <style>
4          table,th,td {
5              border : 1px solid black;
6              border-collapse: collapse;
7          }
8          th,td {
9              padding: 5px;
10         }
11     </style>
12     <body>
13         <h2>The XMLHttpRequest Object</h2>
14         <button type="button" onclick="loadDoc()">Get my CD collection</button>
15         <br><br>
16         <table id="demo"></table>
17         <script>
18             // carga el archivo XML, cuando se recibe, llama a myFunction(this).
19             function loadDoc() {
20                 const xhttp = new XMLHttpRequest();
21                 xhttp.onload = function() {
22                     myFunction(this);
23                 }
24                 xhttp.open("GET", "cd_catalog.xml");
25                 xhttp.send();
26             }
27             //Accede al XML completo y busca los nodos <CD>
28             function myFunction(xml) {
29                 const xmlDoc = xml.responseXML :
```



```
30                 const x = xmlDoc.getElementsByTagName("CD");
31                 //crea la tabla
32                 let table=<tr><th>Artist</th><th>Title</th></tr>;
33                 for (let i = 0; i <x.length; i++) {
34                     table += "<tr><td>" +
35                     x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue +
36                     "</td><td>" +
37                     x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue +
38                     "</td></tr>";
39                 }
40                 //inserta la tabla en el elemento con id="demo"
41                 document.getElementById("demo").innerHTML = table;
42             }
43         </script>
44     </body>
45 </html>
```

node.js:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 16

Tarea Ajax en w3schools > ejercicio1 > **js** node.js > ...

```
1 const path = require('path');
2 const express = require('express');
3 const app = express();
4
5 // Servir archivos estáticos desde el directorio actual
6 app.use(express.static(__dirname));
7
8 app.listen(3000, () => {
9   console.log('Servidor escuchando en http://localhost:3000');
10 });
11
```

cd_catalog.xml:

Tarea Ajax en w3schools > ejercicio6 > [cd_catalog.xml](#)

```
1 <CATALOG>
2   <CD>
3     <TITLE>Empire Burlesque</TITLE>
4     <ARTIST>Bob Dylan</ARTIST>
5     <COUNTRY>USA</COUNTRY>
6     <COMPANY>Columbia</COMPANY>
7     <PRICE>10.90</PRICE>
8     <YEAR>1985</YEAR>
9   </CD>
10  <CD>
11    <TITLE>Hide your heart</TITLE>
12    <ARTIST>Bonnie Tyler</ARTIST>
13    <COUNTRY>UK</COUNTRY>
14    <COMPANY>CBS Records</COMPANY>
15    <PRICE>9.90</PRICE>
16    <YEAR>1988</YEAR>
17  </CD>
18  <CD>
19    <TITLE>Greatest Hits</TITLE>
20    <ARTIST>Dolly Parton</ARTIST>
21    <COUNTRY>USA</COUNTRY>
22    <COMPANY>RCA</COMPANY>
23    <PRICE>9.90</PRICE>
24    <YEAR>1982</YEAR>
25  </CD>
26  <CD>
27    <TITLE>Still got the blues</TITLE>
28    <ARTIST>Gary Moore</ARTIST>
29    <COUNTRY>UK</COUNTRY>
```

Pruebas:

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación**Aprobación:** 2022/03/01**Código:** GUIA-PRLE-001**Página:** 17

The screenshot shows a web browser window with the URL `localhost:3000` in the address bar. The page title is "The XMLHttpRequest Object". Below the title, there is a button labeled "Get my CD collection". A table is displayed with two columns: "Artist" and "Title". The data in the table is as follows:

Artist	Title
Bob Dylan	Empire Burlesque
Bonnie Tyler	Hide your heart
Dolly Parton	Greatest Hits
Gary Moore	Still got the blues
Eros Ramazzotti	Eros
Bee Gees	One night only
Dr.Hook	Sylvias Mother
Rod Stewart	Maggie May
Andrea Bocelli	Romanza
Percy Sledge	When a man loves a woman
Savage Rose	Black angel
Many	1999 Grammy Nominees
Kenny Rogers	For the good times
Will Smith	Big Willie style
Van Morrison	Tupelo Honey

Recuperar el contenido de un archivo PHP:

Para aclarar para este ejercicio fue necesario utilizar XAMPP

index.html:

```
Tarea Ajax en w3schools > ejercicio7 > index.html > html > body
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <h2>The XMLHttpRequest Object</h2>
5          <h3>Start typing a name in the input field below:</h3>
6          <p>Suggestions: <span id="txtHint"></span></p>
7          <p>First name: <input type="text" id="txt1" onkeyup="showHint(this.value)"></p>
8          <script>
9              function showHint(str) {
10                  if (str.length == 0) {
11                      document.getElementById("txtHint").innerHTML = "";
12                      return;
13                  }
14                  const xhttp = new XMLHttpRequest();
15                  xhttp.onload = function() {
16                      document.getElementById("txtHint").innerHTML =
17                          this.responseText;
18                  }
19                  xhttp.open("GET", "gethint.php?q="+str);
20                  xhttp.send();
21              }
22          </script>
23      </body>
24  </html>
```

gethint.php:

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 18

Tarea Ajax en w3schools > ejercicio7 >  gethint.php

```
1  <?php
2      // Array with names
3      $a[] = "Anna";
4      $a[] = "Brittany";
5      $a[] = "Cinderella";
6      $a[] = "Diana";
7      $a[] = "Eva";
8      $a[] = "Fiona";
9      $a[] = "Gunda";
10     $a[] = "Hege";
11     $a[] = "Inga";
12     $a[] = "Johanna";
13     $a[] = "Kitty";
14     $a[] = "Linda";
15     $a[] = "Nina";
16     $a[] = "Ophelia";
17     $a[] = "Petunia";
18     $a[] = "Amanda";
19     $a[] = "Raquel";
20     $a[] = "Cindy";
21     $a[] = "Doris";
22     $a[] = "Eve";
23     $a[] = "Evita";
24     $a[] = "Sunniva";
25     $a[] = "Tove";
26     $a[] = "Unni";
27     $a[] = "Violet";
28     $a[] = "Liza";
29     $a[] = "Elizabeth":
```



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

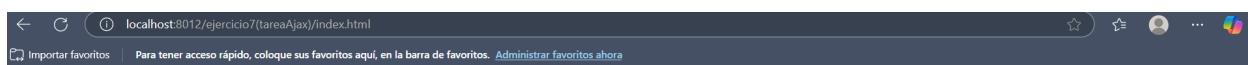
Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 19

```
30     $a[] = "Ellen";
31     $a[] = "Wenche";
32     $a[] = "Vicky";
33
34     // get the q parameter from URL
35     $q = $_REQUEST["q"];
36
37     $hint = "";
38
39     // lookup all hints from array if $q is different from ""
40     if ($q !== "") {
41         $q = strtolower($q);
42         $len=strlen($q);
43         foreach($a as $name) {
44             if (stristr($q, substr($name, 0, $len))) {
45                 if ($hint === "") {
46                     $hint = $name;
47                 } else {
48                     $hint .= ", $name";
49                 }
50             }
51         }
52     }
53
54     // Output "no suggestion" if no hint was found or output correct values
55     echo $hint === "" ? "no suggestion" : $hint;
56 ?>
```

Pruebas:



The XMLHttpRequest Object

Start typing a name in the input field below:

Suggestions: Raquel

First name:

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 20

The screenshot shows the XAMPP Control Panel interface. On the left, there's a sidebar with icons for Modules, Service, and a search bar. The main area has a table titled "XAMPP Control Panel v3.3.0" with columns: Service, Module, PID(s), Port(s), and Actions (Stop, Admin, Config, Logs). The table lists the following services:

Service	Module	PID(s)	Port(s)	Actions
	Apache	4180	1443, 8012	Stop Admin Config Logs
	MySQL	4432	3306	Stop Admin Config Logs
	FileZilla			Start Admin Config Logs
	Mercury			Start Admin Config Logs
	Tomcat			Start Admin Config Logs

Below the table is a log window displaying the following messages:

```
17:10:34 [main] The Mercury module is disabled
17:10:34 [main] The Tomcat module is disabled
17:10:34 [main] Enabling autostart for module "Apache"
17:10:34 [main] Enabling autostart for module "MySQL"
17:10:34 [main] Starting Check-Timer
17:10:34 [main] Control Panel Ready
17:10:34 [Apache] Autostart aborted: Apache is already running
17:10:34 [mysql] Autostart aborted: MySQL is already running
```

Recuperar el contenido de un archivo ASP:

Para esto fue necesario activar IIS y mover los archivos a la carpeta wwwroot.

index.html:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 21

```
Tarea Ajax en w3schools > ejercicio8 > index.html > ...
1  <!DOCTYPE html>
2  <html>
3      <body>
4
5          <h2>The XMLHttpRequest Object</h2>
6          <h3>Start typing a name in the input field below:</h3>
7
8          <p>Suggestions: <span id="txtHint"></span></p>
9          <p>First name: <input type="text" id="txt1" onkeyup="showHint(this.value)"></p>
10
11     <script>
12         function showHint(str) {
13             if (str.length == 0) {
14                 document.getElementById("txtHint").innerHTML = "";
15                 return;
16             }
17             const xhttp = new XMLHttpRequest();
18             xhttp.onload = function() { (property) Element.innerHTML: string
19                 document.getElementById("txtHint").innerHTML =
20                     this.responseText;
21             }
22             xhttp.open("GET", "gethint.asp?q="+str);
23             xhttp.send();
24         }
25     </script>
26
27     </body>
28 </html>
```

Ln 1 Col 1

gethint.asp:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 22

Tarea Ajax en w3schools > ejercicio8 > gethint

```
1 <%  
2 response.expires=-1  
3 dim a(30)  
4 'Fill up array with names  
5 a(1)="Anna"  
6 a(2)="Brittany"  
7 a(3)="Cinderella"  
8 a(4)="Diana"  
9 a(5)="Eva"  
10 a(6)="Fiona"  
11 a(7)="Gunda"  
12 a(8)="Hege"  
13 a(9)="Inga"  
14 a(10)="Johanna"  
15 a(11)="Kitty"  
16 a(12)="Linda"  
17 a(13)="Nina"  
18 a(14)="Ophelia"  
19 a(15)="Petunia"  
20 a(16)="Amanda"  
21 a(17)="Raquel"  
22 a(18)="Cindy"  
23 a(19)="Doris"  
24 a(20)="Eve"  
25 a(21)="Evita"  
26 a(22)="Sunniva"  
27 a(23)="Tove"  
28 a(24)="Unni"  
29 a(25)="Violet"
```

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 23

```
1  <%  
--  
34 a(30)="Vicky"  
35  
36 'get the q parameter from URL  
37 q=ucase(request.querystring("q"))  
38  
39 'lookup all hints from array if length of q>0  
40 if len(q)>0 then  
41   hint=""  
42   for i=1 to 30  
43     if q=ucase(mid(a(i),1,len(q))) then  
44       if hint="" then  
45         hint=a(i)  
46       else  
47         hint=hint & " , " & a(i)  
48       end if  
49     end if  
50   next  
51 end if  
52  
53 'Output "no suggestion" if no hint were found  
54 'or output the correct values  
55 if hint="" then  
56   response.write("no suggestion")  
57 else  
58   response.write(hint)  
59 end if  
60 %>
```

Pruebas:

Nombre	Fecha de modificación	Tipo	Tamaño
gethint.asp	3/05/2025 00:41	Archivo de origen ...	1 KB
iisstart.htm	3/05/2025 00:38	Microsoft Edge HT...	1 KB
iisstart.png	3/05/2025 00:38	Archivo PNG	97 KB
index.html	3/05/2025 00:42	Microsoft Edge HT...	1 KB



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 24



The XMLHttpRequest Object

Start typing a name in the input field below:

Suggestions: Raquel

First name:

Recuperar contenido de una base de datos:

Para esto fue necesario usar XAMPP:

index.html:

```
Tarea Ajax en w3schools > ejercicio9 > index.html > ...
1  <!DOCTYPE html>
2  <html>
3      <style>
4          th,td {
5              padding: 5px;
6          }
7      </style>
8      <body>
9          <h2>The XMLHttpRequest Object</h2>
10         <form action="">
11             <select name="customers" onchange="showCustomer(this.value)">
12                 <option value="">Select a customer:</option>
13                 <option value="ALFKI">Alfreds Futterkiste</option>
14                 <option value="NORTS ">North/South</option>
15                 <option value="WOLZA">Wolski Zajazd</option>
16             </select>
17         </form>
18         <br>
19         <div id="txtHint">Customer info will be listed here...</div>
20         <script>
21             function showCustomer(str) {
22                 if (str == "") {
23                     document.getElementById("txtHint").innerHTML = "";
24                     return;
25                 }
26                 const xhttp = new XMLHttpRequest();
27                 xhttp.onload = function() {
28                     document.getElementById("txtHint").innerHTML = this.responseText;
29                 }
30             }
31         </script>
```

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 25

```
29         }
30         xhttp.open("GET", "getcustomer.php?q="+str);
31         xhttp.send();
32     }
33     </script>
34   </body>
35 </html>
```

getCustomer.php:

```
Tarea Ajax en w3schools > ejercicio9 > getcustomer.php
1  <?php
2 // Conexión a la base de datos
3 $mysqli = new mysqli("localhost", "root", "", "empresa_demo");
4 if ($mysqli->connect_error) {
5 |   exit('No se pudo conectar a la base de datos.');
6 }
7
8 // Consulta SQL ajustada a tu tabla "clientes"
9 $sql = "SELECT id_cliente, nombre_empresa, nombre_contacto, direccion, ciudad, codigo_postal, pais
10 |   FROM clientes WHERE id_cliente = ?";
11
12 // Preparar y ejecutar
13 $stmt = $mysqli->prepare($sql);
14 $stmt->bind_param("s", $_GET['q']);
15 $stmt->execute();
16 $stmt->store_result();
17 $stmt->bind_result($id, $empresa, $contacto, $direccion, $ciudad, $postal, $pais);
18 $stmt->fetch();
19 $stmt->close();
20
21 // Mostrar resultados en tabla HTML
22 if ($id) {
23   echo "<table border='1'>";
24   echo "<tr><th>ID</th><td>$id</td></tr>";
25   echo "<tr><th>Empresa</th><td>$empresa</td></tr>";
26   echo "<tr><th>Contacto</th><td>$contacto</td></tr>";
27   echo "<tr><th>Dirección</th><td>$direccion</td></tr>";
28   echo "<tr><th>Ciudad</th><td>$ciudad</td></tr>";
29   echo "<tr><th>Código Postal</th><td>$postal</td></tr>":
30   echo "<tr><th>País</th><td>$pais</td></tr>";
31   echo "</table>";
32 } else {
33   echo "Cliente no encontrado.";
34 }
35 ?>
```

clientes.sql:

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 26

```
1 -- Crear base de datos
2 CREATE DATABASE IF NOT EXISTS empresa_demo;
3 USE empresa_demo;
4
5 -- Crear tabla clientes
6 CREATE TABLE IF NOT EXISTS clientes (
7     id_cliente VARCHAR(10) PRIMARY KEY,
8     nombre_empresa VARCHAR(100),
9     nombre_contacto VARCHAR(100),
10    direccion VARCHAR(100),
11    ciudad VARCHAR(50),
12    codigo_postal VARCHAR(20),
13    pais VARCHAR(50)
14 );
15
16 -- Insertar datos de prueba
17 INSERT INTO clientes (id_cliente, nombre_empresa, nombre_contacto, direccion, ciudad, codigo_postal, pais) VALUE
18 ('ALFKI', 'Alfreds Futterkiste', 'Maria Anders', 'Obere Str. 57', 'Berlin', '12209', 'Germany'),
19 ('NORTS', 'North/South', 'Simon Crowther', 'South House 300 Queensbridge', 'London', 'SW7 1RZ', 'UK'),
20 ('WOLZA', 'Wolski Zajazd', 'Zbyszek Piestrzeniewicz', 'ul. Filtrowa 68', 'Warszawa', '01-012', 'Poland');
```

Pruebas:

The XMLHttprequest Object

Alfreds Futterkiste

ID	ALFKI
Empresa	Alfreds Futterkiste
Contacto	Maria Anders
Dirección	Obere Str. 57
Ciudad	Berlin
Código Postal	12209
País	Germany

Mostrar datos XML dentro de un elemento div HTML:**index.html:**



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 27

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <div id='showCD'></div>
5          <script>
6              const xhttp = new XMLHttpRequest();
7              xhttp.onload = function() {
8                  const xmlDoc = xhttp.responseXML;
9                  const cd = xmlDoc.getElementsByTagName("CD");
10                 myFunction(cd, 0);
11             }
12             xhttp.open("GET", "cd_catalog.xml");
13             xhttp.send();
14             //muestra la información de un CD en el contenedor
15
16             function myFunction(cd, i) {
17                 document.getElementById("showCD").innerHTML =
18                     "Artist: " +
19                     cd[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue +
20                     "<br>Title: " +
21                     cd[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue +
22                     "<br>Year: " +
23                     cd[i].getElementsByTagName("YEAR")[0].childNodes[0].nodeValue;
24             }
25         </script>
26     </body>
27 </html>
```

node.js:

```
Tarea Ajax en w3schools > ejercicio1 > js node.js > ...
1  const path = require('path');
2  const express = require('express');
3  const app = express();
4
5  // Servir archivos estáticos desde el directorio actual
6  app.use(express.static(__dirname));
7
8  app.listen(3000, () => {
9      console.log('Servidor escuchando en http://localhost:3000');
10 });
11
```

cd_catalog.xml:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 28

Tarea Ajax en w3schools > ejercicio6 > cd_catalog.xml

```
1  <CATALOG>
2    <CD>
3      <TITLE>Empire Burlesque</TITLE>
4      <ARTIST>Bob Dylan</ARTIST>
5      <COUNTRY>USA</COUNTRY>
6      <COMPANY>Columbia</COMPANY>
7      <PRICE>10.90</PRICE>
8      <YEAR>1985</YEAR>
9    </CD>
10   <CD>
11     <TITLE>Hide your heart</TITLE>
12     <ARTIST>Bonnie Tyler</ARTIST>
13     <COUNTRY>UK</COUNTRY>
14     <COMPANY>CBS Records</COMPANY>
15     <PRICE>9.90</PRICE>
16     <YEAR>1988</YEAR>
17   </CD>
18   <CD>
19     <TITLE>Greatest Hits</TITLE>
20     <ARTIST>Dolly Parton</ARTIST>
21     <COUNTRY>USA</COUNTRY>
22     <COMPANY>RCA</COMPANY>
23     <PRICE>9.90</PRICE>
24     <YEAR>1982</YEAR>
25   </CD>
26   <CD>
27     <TITLE>Still got the blues</TITLE>
28     <ARTIST>Gary Moore</ARTIST>
29     <COUNTRY>UK</COUNTRY>
```

Pruebas:



Artista: Bob Dylan
Título: Empire Burlesque
Año: 1985



Navegar por los nodos XML:
[index.html](#):



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 29

```
Tarea Ajax en w3schools > ejercicio11 > index.html > ...
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <div id='showCD'></div><br>
5          <input type="button" onclick="previous()" value="<<">
6          <input type="button" onclick="next()" value=">>">
7              <script>
8                  let i = 0;
9                  let len;
10                 let cd;
11                 const xhttp = new XMLHttpRequest();
12                 xhttp.onload = function() {
13                     const xmlDoc = xhttp.responseXML;
14                     cd = xmlDoc.getElementsByTagName("CD");
15                     len = cd.length;
16                     displayCD(i);
17                 }
18                 xhttp.open("GET", "cd_catalog.xml");
19                 xhttp.send();
20                 function displayCD(i) {
21                     document.getElementById("showCD").innerHTML =
22                         "Artist: " +
23                         cd[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue +
24                         "<br>Title: " +
25                         cd[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue +
26                         "<br>Year: " +
27                         cd[i].getElementsByTagName("YEAR")[0].childNodes[0].nodeValue;
28                 }
29                 function next() {
30                     if (i < len-1) {
31                         i++;
32                         displayCD(i);
33                     }
34                 }
35                 function previous() {
36                     if (i > 0) {
37                         i--;
38                         displayCD(i);
39                     }
40                 }
41             </script>
42         </body>
43     </html>
```

node.js:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 30

Tarea Ajax en w3schools > ejercicio1 > **js** node.js > ...

```
1 const path = require('path');
2 const express = require('express');
3 const app = express();
4
5 // Servir archivos estáticos desde el directorio actual
6 app.use(express.static(__dirname));
7
8 app.listen(3000, () => {
9   console.log('Servidor escuchando en http://localhost:3000');
10 });
11
```

cd_catalog.xml:

Tarea Ajax en w3schools > ejercicio6 > [cd_catalog.xml](#)

```
1 <CATALOG>
2   <CD>
3     <TITLE>Empire Burlesque</TITLE>
4     <ARTIST>Bob Dylan</ARTIST>
5     <COUNTRY>USA</COUNTRY>
6     <COMPANY>Columbia</COMPANY>
7     <PRICE>10.90</PRICE>
8     <YEAR>1985</YEAR>
9   </CD>
10  <CD>
11    <TITLE>Hide your heart</TITLE>
12    <ARTIST>Bonnie Tyler</ARTIST>
13    <COUNTRY>UK</COUNTRY>
14    <COMPANY>CBS Records</COMPANY>
15    <PRICE>9.90</PRICE>
16    <YEAR>1988</YEAR>
17  </CD>
18  <CD>
19    <TITLE>Greatest Hits</TITLE>
20    <ARTIST>Dolly Parton</ARTIST>
21    <COUNTRY>USA</COUNTRY>
22    <COMPANY>RCA</COMPANY>
23    <PRICE>9.90</PRICE>
24    <YEAR>1982</YEAR>
25  </CD>
26  <CD>
27    <TITLE>Still got the blues</TITLE>
28    <ARTIST>Gary Moore</ARTIST>
29    <COUNTRY>UK</COUNTRY>
```

Pruebas:

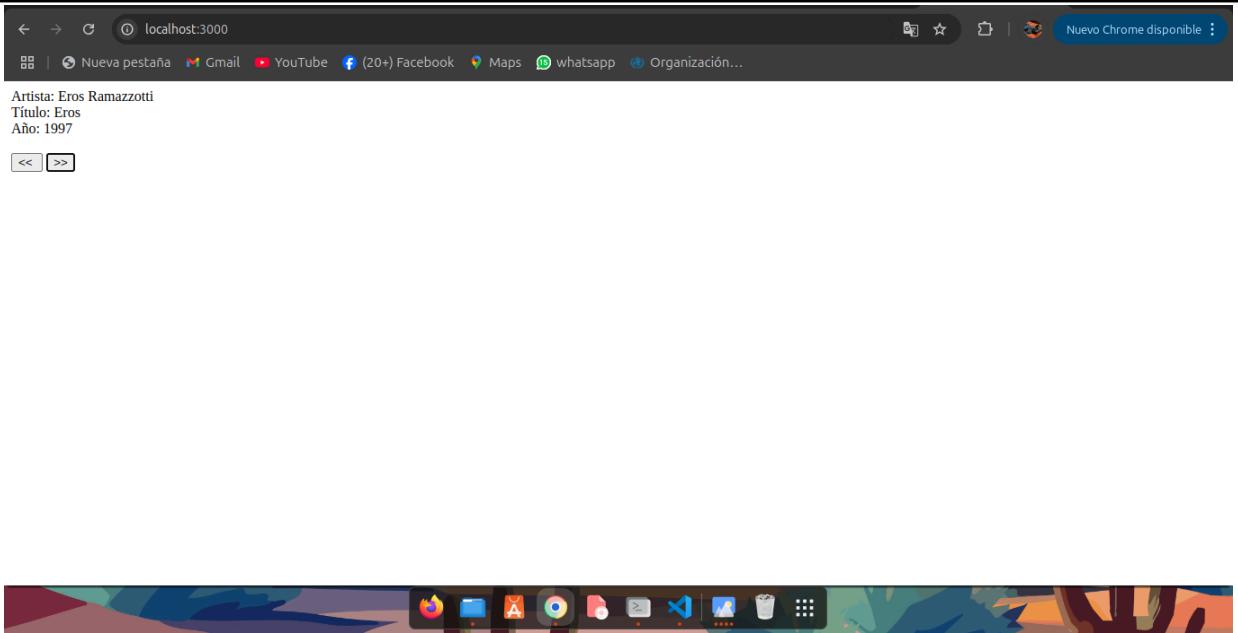


Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 31



Una sencilla aplicación de catálogo de CD:

index.html:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <style>
5              table,th,td {
6                  border : 1px solid black;
7                  border-collapse: collapse;
8              }
9              th,td {
10                  padding: 5px;
11              }
12          </style>
13      </head>
14      <body>
15          <p>Click on a CD to display album information.</p>
16          <p id='showCD'></p>
17          <table id="demo"></table>
18          <script>
19              const xhttp = new XMLHttpRequest();
20              let cd;
21              xhttp.onload = function() {
22                  const xmlDoc = xhttp.responseXML;
23                  cd = xmlDoc.getElementsByTagName("CD");
24                  loadCD();
25              }
26              xhttp.open("GET", "cd_catalog.xml");
27              xhttp.send();
28              function loadCD() {
29                  let table=<tr><th>Artist</th><th>Title</th></tr>;
30
31                  for (let i = 0; i < cd.length; i++) {
32                      table += "<tr onclick='displayCD(" + i + ")'><td>";
33                      table += cd[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue;
34                      table += "</td><td>";
35                      table += cd[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue;
36                      table += "</td></tr>";
37                  }
38                  document.getElementById("demo").innerHTML = table;
39              }
40              function displayCD(i) {
41                  document.getElementById("showCD").innerHTML =
42                      "Artist: " +
43                      cd[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue +
44                      "<br>Title: " +
45                      cd[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue +
46                      "<br>Year: " +
47                      cd[i].getElementsByTagName("YEAR")[0].childNodes[0].nodeValue;
48              }
49          </script>
50      </body>
51  </html>
```

node.js:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 33

Tarea Ajax en w3schools > ejercicio1 > **js** node.js > ...

```
1 const path = require('path');
2 const express = require('express');
3 const app = express();
4
5 // Servir archivos estáticos desde el directorio actual
6 app.use(express.static(__dirname));
7
8 app.listen(3000, () => {
9   console.log('Servidor escuchando en http://localhost:3000');
10 });
11
```

cd_catalog.xml:

Tarea Ajax en w3schools > ejercicio6 > [cd_catalog.xml](#)

```
1 <CATALOG>
2   <CD>
3     <TITLE>Empire Burlesque</TITLE>
4     <ARTIST>Bob Dylan</ARTIST>
5     <COUNTRY>USA</COUNTRY>
6     <COMPANY>Columbia</COMPANY>
7     <PRICE>10.90</PRICE>
8     <YEAR>1985</YEAR>
9   </CD>
10  <CD>
11    <TITLE>Hide your heart</TITLE>
12    <ARTIST>Bonnie Tyler</ARTIST>
13    <COUNTRY>UK</COUNTRY>
14    <COMPANY>CBS Records</COMPANY>
15    <PRICE>9.90</PRICE>
16    <YEAR>1988</YEAR>
17  </CD>
18  <CD>
19    <TITLE>Greatest Hits</TITLE>
20    <ARTIST>Dolly Parton</ARTIST>
21    <COUNTRY>USA</COUNTRY>
22    <COMPANY>RCA</COMPANY>
23    <PRICE>9.90</PRICE>
24    <YEAR>1982</YEAR>
25  </CD>
26  <CD>
27    <TITLE>Still got the blues</TITLE>
28    <ARTIST>Gary Moore</ARTIST>
29    <COUNTRY>UK</COUNTRY>
```

Pruebas:



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 34

Haga clic en un CD para mostrar la información del álbum.

Artista	Título
Bob Dylan	Imperio burlesco
Bonnie Tyler	Esconde tu corazón
Dolly Parton	Grandes éxitos
Gary Moore	Todavía tengo la tristeza
Eros Ramazzotti	Eros
Bee Gees	Sólo una noche
Dr. Hook	La madre de Silvia
Rod Stewart	Maggie May
Andrea Bocelli	Romanza
Percy Sledge	Cuando un hombre ama a una mujer
Rosa salvaje	Ángel negro
Muchos	Nominados al Grammy de 1999
Kenny Rogers	Para los buenos tiempos
Will Smith	Estilo Big Willie
Van Morrison	Miel de Tupelo
Elton John	Soulsville

Commits:

- Commits on May 3, 2025

hubo un error al cargar el commit

Renzo Geomar Mamani Quispe committed 3 days ago

d23dfa4 ↗ ↗

corrigiendo errores

Renzo Geomar Mamani Quispe committed 3 days ago

54b10bf ↗ ↗

copiando el ultimo, aclarando que algunos se repiten

Renzo Geomar Mamani Quispe committed 3 days ago

df0439c ↗ ↗

copiando el ejercicio

Renzo Geomar Mamani Quispe committed 3 days ago

97b77a9 ↗ ↗



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 35

<p>-o- Commits on May 3, 2025</p> <ul style="list-style-type: none">funciona correctamente Renzo Geomar Mamani Quispe committed 3 days ago 6dedb9acopiando el ejercicio 10, ademas de reutilizar el node.js y cd_catalog.xml Renzo Geomar Mamani Quispe committed 3 days ago 9b2a9a7ya funciona la base de datos, pero desde el servidor XAMPP Renzo Geomar Mamani Quispe committed 3 days ago 6f5c222incorporando base de datos Renzo Geomar Mamani Quispe committed 3 days ago 74b4c22copiando el ejemplo 9, además el archivo .php el cual no funciona con nodeJS Renzo Geomar Mamani Quispe committed 3 days ago 07f7887copiando el ejercicio 8 y el archivo .asp, para aclarar esto no funcionara con nodeJS para ejecutarlo se tiene que hacer con IIS que esta en otro repositorio Renzo Geomar Mamani Quispe committed 3 days ago 42e43d8el node.js no funciona para archivos php, por lo cual tuve que usar XAMPP, por lo cual el ejercicio esta ejecutandose desde otro repositorio Renzo Geomar Mamani Quispe committed 3 days ago 65b894c	
<p>-o- Commits on May 2, 2025</p> <ul style="list-style-type: none">copiando el ejercicio 7 de wetschools, ademas se reutiliza el codigo de node.js e incorpooro el archivo que me dan osea gethint.php que sera mi base de datos Renzo Geomar Mamani Quispe committed 3 days ago 2fc7de5	<p>Copy full SHA for 2fc7de5</p>
<p>-o- Commits on May 2, 2025</p> <ul style="list-style-type: none">probando el codigo Renzo Geomar Mamani Quispe committed 3 days ago 68f6789copiando el ejercicio 6 de wetschools, ademas se reutiliza el codigo de node.js y incorpooro el archivo que me dan osea cd_catalog que sera mi base de datos Renzo Geomar Mamani Quispe committed 4 days ago 641cbbbagregando contenido a note.xml Renzo Geomar Mamani Quispe committed 4 days ago d80be14copiando el ejercicio 5 de wetschools, ademas se reutiliza el codigo de node.js Renzo Geomar Mamani Quispe committed 4 days ago 2529dbcya se ha probado el codigo, funciona sin problemas Renzo Geomar Mamani Quispe committed 4 days ago 0964180copiando el ejercicio 4 de wetschools, ademas se reutiliza el codigo de node.js Renzo Geomar Mamani Quispe committed 4 days ago e582b0c	



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 36

corrigiendo errores, ya funciona el ejercicio propuesto 2	ce9c3e4		
Renzo Geomar Mamani Quispe committed 4 days ago			
agregando poema y el contenedor donde se mostrara	577956f		
Renzo Geomar Mamani Quispe committed 4 days ago			
colocando el codigo del documento con una pagina de prueba, aun no funciona	a7aaee5		
Renzo Geomar Mamani Quispe committed 4 days ago			
Iniciando laboratorio 4 con la instalacion de los paquetes necesarios y probando con el primer ejercicio propuesto	cc6c25a		
Renzo Geomar Mamani Quispe committed 4 days ago			

LINK DE A MI REPOSITORIO DE GITHUB:

<https://github.com/renzogeomar/Laboratorio-Programacion-Web-2.git>