 UNIVERSIDAD DON BOSCO	<p style="text-align: center;">UNIVERSIDAD DON BOSCO FACULTAD DE ESTUDIOS TECNOLÓGICOS COMPUTACIÓN</p>
CICLO: 01	<p style="text-align: center;"><i>GUIA DE LABORATORIO #12</i></p> <p>Nombre de la Practica: Utilización de AJAX en aplicaciones web</p> <p>Lugar de Ejecución: Centro de cómputo</p> <p>Tiempo Estimado: 2 horas con 30 minutos</p> <p>MATERIA: Desarrollo de Aplicaciones Web con Software Interpretado en el Servidor</p>

I. OBJETIVOS

Que el estudiante:

- Tenga claridad sobre el uso de las técnicas de AJAX en el desarrollo de aplicaciones web.
- Haga uso del objeto XMLHttpRequest en cualquier navegador sobre el que se estén visualizando las páginas web solicitadas al servidor.
- Adquiera habilidad en el proceso de aplicar AJAX dentro de una aplicación web.
- Haga uso de diferentes frameworks para trabajar en sus aplicaciones con AJAX.
- Realice aplicaciones avanzadas en las que use AJAX y acceso a bases de datos.

II. INTRODUCCION TEORICA

Origen y definición de AJAX

En primer lugar, hay que decir que AJAX no es en sí mismo, una tecnología, y mucho menos, se trata de algo nuevo. AJAX solamente es un término que se ha adoptado y aceptado a nivel de desarrolladores web para hacer referencia a la utilización de un objeto de JavaScript, denominado XMLHttpRequest, para obtener información de un servidor web de modo dinámico o asíncrono. La posibilidad de hacer uso de este objeto es posible desde 1998. Microsoft incorporó este concepto desde Internet Explorer 4.0, con muchas dificultades de programación para los desarrolladores, y en Internet Explorer 5.0, ya con el objeto XMLHttpRequest, pero adoptando otro nombre. En ese entonces se le denominó Remote Scripting. Lastimosamente nadie le vio mucho potencial, hasta que Microsoft mostró su utilización con el Outlook Web Access incluido con Exchange. Ahora es mucho más popular con el desarrollo que ha tenido, principalmente en los servicios ofrecidos por Google en sus interfaces de usuario.

Inicialmente AJAX utilizó técnicas como marcos invisibles con elementos IFRAME para hacer uso de esta técnica, hoy en día se prefiere utilizar el objeto XMLHttpRequest.

La popularización de su utilización se debe principalmente a empresas como Google, Yahoo, Flickr, etc. Que han desarrollado diversas aplicaciones web en las que han utilizado el concepto AJAX para realizar aplicaciones web con excelentes servicios de interfaz de usuario.

Hoy en día la utilización de AJAX tiene todo el soporte necesario en los diversos navegadores web, como Firefox, Opera, Safari, Chrome, etc. Todos ellos ofrecen una versión clon del objeto XMLHttpRequest para que las aplicaciones AJAX funcionen sin inconvenientes.

Elementos de la definición de AJAX

Los elementos de la definición de AJAX son tres principalmente; sin embargo, en la práctica son varios más. Veamos cada uno de estos:

Asíncrono: Significa que el usuario puede seguir interactuando con la página web mientras el navegador espera obtener la información solicitada al servidor desde la aplicación, sin que esto signifique que se tenga que recargar la página web.

JavaScript: A pesar de que AJAX descansa sobre la utilización de diversas tecnologías, como: XHTML, CSS y DOM, además de JavaScript, se ha preferido tomar la letra inicial de esta última para que el nombre resulte más atractivo. La principal utilización de JavaScript en una aplicación AJAX consiste en el uso del objeto XMLHttpRequest y generación de contenido de forma dinámica.

XML: Es uno de los tantos formatos de datos que pueden utilizarse para transferir contenido entre el servidor y la página web. Además, pueden utilizarse lenguajes estáticos como (X)HTML o lenguajes dinámicos como PHP, ASP o JSP. Además, de archivos de texto.

Tecnologías involucradas en la utilización de AJAX

En la práctica las tecnologías involucradas en el desarrollo de una aplicación web son:

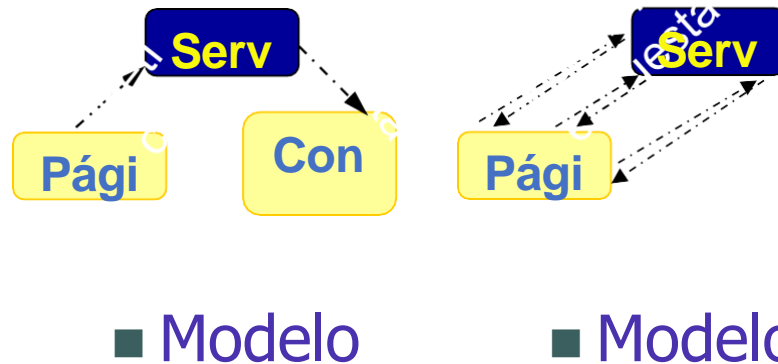
- XHTML y CSS para la estructura de la página web y su presentación, respectivamente.
- El Modelo de Objetos de Documento (DOM) para la visualización y manipulación de las páginas.
- El objeto XMLHttpRequest de JavaScript para transferir datos entre el cliente y el servidor web.
- XML como formato para los datos que fluyen entre el cliente y el servidor. No obstante, también puede hacerse uso de texto normal para este propósito.
- JavaScript para mostrar e interactuar dinámicamente con todo lo anterior.

Una aplicación AJAX actúa como intermediario entre el usuario y el servidor. De modo que si la acción del usuario no requiere una llamada al servidor, el motor de AJAX se ejecuta asincrónicamente, para que el usuario siga interactuando con la aplicación. El motor de AJAX actualizará el sector de la página cuando los datos estén disponibles, muy probablemente, sin que el usuario lo note.

DHTML + XMLHttpRequest + Server Files = AJAX

Modelo petición respuesta del protocolo HTTP

Una aplicación web tiene que someterse al esquema de petición/respuesta que rige al protocolo HTTP. De acuerdo a este modelo, para que ocurra algo en una página web debe solicitarse al servidor otra página web (que posiblemente sea la misma). Adicionalmente, para obtener la respuesta debe recargarse la página por completo para que se muestren los datos actualizados en el navegador del cliente.



Aplicaciones web con AJAX y aplicaciones web sin AJAX

Una aplicación web que no utilice AJAX estará confinada a respetar 100% el modelo petición/respuesta del protocolo HTTP. En cambio, en una aplicación web con AJAX, la diferencia fundamental es que el usuario puede seguir interactuando con la página web, mientras se procesa en el servidor de forma asíncrona. Las siguientes figuras ilustran este hecho:

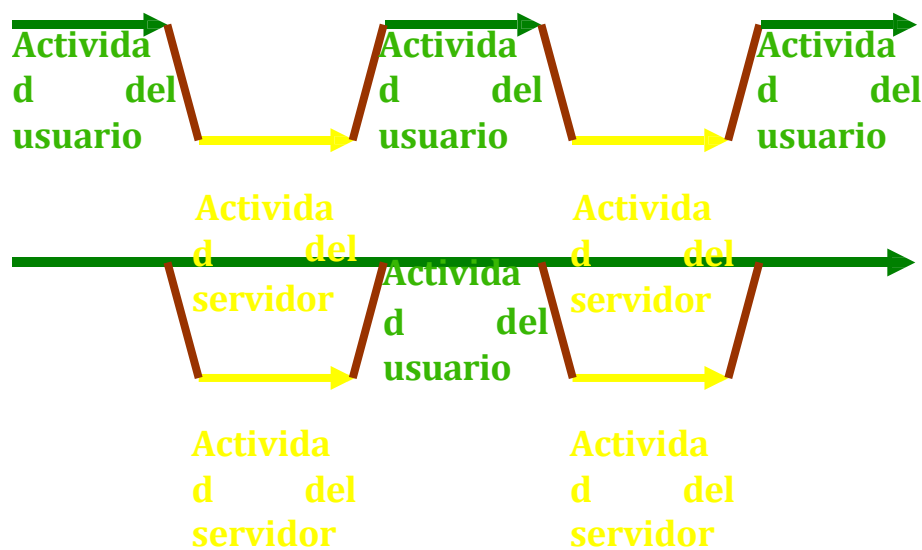


Figura: Esquemas de funcionamiento de aplicaciones web que no utilizan AJAX y que si utilizan AJAX. Note que cuando se usa AJAX la línea verde permanece continua, queriendo indicar que la acción del usuario con la página web no se ve interrumpida.

El objeto XMLHttpRequest

Se puede decir que AJAX es un concepto que se utiliza para describir cómo interacciona el objeto del lado del cliente XMLHttpRequest con los programas del lado del servidor. Este objeto permite realizar peticiones asíncronas al servidor. Los objetos XMLHttpRequest se crean y gestionan de forma diferente en cada navegador. En Internet Explorer 6.0 y anteriores el objeto es creado como control ActiveX. En tanto que en Firefox, Safari, Opera y Chrome se utiliza un objeto básico de JavaScript. Esto implica un cierto esfuerzo adicional para lograr compatibilidad.

Una vez que se ha creado el objeto XMLHttpRequest se puede disponer de una gran cantidad de propiedades y métodos cuya utilización dependerá de lo que queramos realizar.

Algunas de estas propiedades y métodos se muestran en la siguiente tabla:

Método	Descripción
abort()	Detiene o cancela prematuramente la petición actual.
getAllResponseHeaders()	Devuelve todos los encabezados HTTP como una variable de tipo String.
getResponseHeader()	Devuelve el valor del encabezado HTTP especificado en el argumento.
open()	Define los atributos necesarios para establecer una conexión con el servidor, permitiendo elegir el método de la petición (GET o POST), establecer si la conexión es asíncrona e indicar el URL con el que queremos conectar.
setRequestHeader()	Permite incluir en el encabezado de la petición un par etiqueta/valor al momento de enviarlo.
send()	Envía la petición actual al servidor. Si la petición es asíncrona, la respuesta se obtendrá de forma inmediata.
Propiedad	Descripción
onreadystatechange	Se utiliza como gestor de eventos para detectar cambios de estado.
readyState	Contiene el estado actual del objeto. En donde, 0: indica sin inicializar, 1: indicación de que el objeto se está cargando, 2: indica que el objeto está cargado y 3: indica que el objeto continúa en interacción, y 4: completado.
responseText	Contiene la respuesta de la petición hecha al servidor en formato cadena.
responseXML	Contiene la respuesta de la petición hecha al servidor pero en formato XML.
status	Contiene el estado de la petición en formato numérico. Devuelve 200 para indicar que se ha localizado el documento.
statusText	Contiene el estado de la petición, pero en formato de cadena. En vez de devolver 404, devolvería su cadena equivalente "Not Found".

Creación del objeto XMLHttpRequest

En vista de que es necesario saber qué navegador se está utilizando para poder crear el objeto XMLHttpRequest, es necesario hacer ciertas comprobaciones en el código antes de intentar hacer uso de las propiedades y/o métodos de este objeto. El código que se puede utilizar es el siguiente:

```
//Variable para comprobar el navegador
var xmlhttp = false;
//Comprobar si es IE
try {
    //Verificar si la version de JS es superior a la 5
    xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
    alert("Estas usando MS Internet Explorer");
} catch(e) {
    //Si no, utilizar el antiguo objeto ActiveX
    try {
        //Si se está usando Internet Explorer
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        alert("Estas usando MS Internet Explorer");
    } catch(E) {
        //Caso contrario se está usando un navegador diferente
        xmlhttp = false;
    }
}
//Si no se usa IE entonces crear una instancia JS del objeto
if(!xmlhttp && typeof XMLHttpRequest != 'undefined'){
    xmlhttp = new XMLHttpRequest();
    alert("Tú navegador no es Internet Explorer");
}
```

Enviar petición al servidor web

Una vez creado el objeto XMLHttpRequest ya se puede enviar la petición asíncrona al servidor, validando que la respuesta recibida sea correcta. Se sugiere utilizar el método GET para obtener esta información del servidor. En caso de querer enviar información hacia el servidor, es recomendable utilizar el método POST.

La técnica para enviar la petición al servidor requiere dos cosas importantes:

1. Indicar con qué secuencia de comandos en el servidor se desea conectar, y
2. Establecer en qué área de la página (elemento HTML debidamente identificado) se desea cargar el resultado de la ejecución de la secuencia de comandos.

Esto implementado en JavaScript, puede lucir como el código que se muestra a continuación:

```
function makerequest(serverPage, objID){
    var obj = document.getElementById(objID);
    xmlhttp.open("GET",serverPage);
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4 && xmlhttp.status == 200){
            obj.innerHTML = xmlhttp.responseText;
        }
    }
}
```

Los argumentos en la función anterior son dos: el nombre junto con su ubicación en el servidor y el ID del elemento HTML en el que se mostrará el resultado. A continuación, se utiliza el método open del objeto XMLHttpRequest para establecer la conexión. Si la propiedad readyState toma el valor de 4 al intentar la conexión, significa que la conexión se completó satisfactoriamente; si la propiedad status toma el valor de 200 significa que el documento solicitado al servidor fue encontrado y que se envió al cliente de forma exitosa.

Si se dan las dos condiciones anteriores la función makerequest() carga la respuesta del script PHP en la propiedad innerHTML del elemento que recibió inicialmente como argumento.

Framework de JavaScript

Un *framework* puede definirse como un conjunto de utilidades y herramientas necesarias para la construcción de un determinado tipo de aplicaciones. Como lo habrá podido observar, AJAX es una metodología de desarrollo que depende en gran medida de la programación con JavaScript.

En vista de que las aplicaciones web se vuelven día con día más complejas al incorporar nuevos y mejorados efectos visuales e interacciones dinámicas con los usuarios y, aparte de esto, es necesario hacer que funcionen del mismo modo en distintos navegadores (al menos en Internet Explorer, Firefox, Safari, Chrome y Opera). Esto hace que el tiempo invertido en el desarrollo de una aplicación se vuelva un inconveniente, sobre todo porque las empresas y organizaciones exigen tiempos cortos para presentar avances significativos en el desarrollo de sus aplicaciones web.

Lo anterior ha motivado el desarrollo de diversas librerías de JavaScript que realizan muchas de las tareas más comunes que son necesarias en una aplicación que utilice AJAX. Algunos de estos *frameworks* de JavaScript están disponibles para su descarga en un solo archivo .js o, como en el caso del Google Web Toolkit y Yahoo UI! Library, están disponibles como un conjunto de archivos y APIs (Interfaces de Programación de Aplicaciones) para el desarrollo web.

Algunos de los frameworks de JavaScript más populares son:

- Prototype.
- JQuery.
- Script.aculo.us.
- XAJAX.
- Google Web Toolkit.
- Yahoo UI! Library.

- Microsoft AJAX Control Toolkit.
- WebFX.

Prototype

Prototype es un framework bastante extendido y fácil de usar que facilita el desarrollo de aplicaciones web en las que se utiliza JavaScript y AJAX. Esta librería fue desarrollada por Sam Stephenson, pero las últimas versiones incluyen código aportado por muchos desarrolladores de forma altruista.

Prototype se ha convertido en una referencia básica de AJAX y es base de muchos otros frameworks y librerías relacionadas como JQuery, Script.aculo.us, etc.

Funciones elementales de Prototype

La función dólar (\$)

Esta función es una forma mejorada para llamar al método `getElementById()` del objeto `document`. La forma en que se utiliza es la siguiente:

```
//Con JavaScript
var parrafo = document.getElementById("texto");
//Con Prototype
var parrafo = $("texto");

//Con JavaScript
var titulo = document.getElementById('title');
var contenido = document.getElementById('content');
//Con Prototype
var tema = $('title', 'content');
```

La función \$F()

Funciona de forma similar a la función dólar (\$), con la diferencia que es utilizada para obtener de manera directa el valor de los campos de formulario.

Por ejemplo, si tenemos:

```
<input type="text" name="nombre" id="nombre" size="25" />
//Con JavaScript
var nombre = document.getElementById("nombre").value;
//Con Prototype
var nombre = $F("nombre");
```

Si se trata de un valor seleccionado de una lista desplegable, se tendría lo siguiente:

```
<select id="lista" name="lista">
  <option value="1">1</option>
</select>

//Con JavaScript
var lista =
document.getElementById("lista").options[document.getElementById("lista").selectedIndex].value;
//Con Prototype
var lista = $F("lista");
```

La función \$\$()

No posee un equivalente en JavaScript, la función recibe como parámetro un selector CSS, devolviendo una matriz o array con cada elemento que cumpla con el criterio del selector dado.

Por ejemplo:

```
<div id="ciudades">
  <p>San Salvador</p>
  <p>Santa Ana</p>
  <p>San Miguel</p>
</div>
```

```
<p>Guadalajara</p>
```

```
//Dentro de todosLosParrafos estarían todos los valores de los elementos P
var todosLosParrafos = $$('p');
//Dentro de parrafosCiudades estarían todos los valores de los elementos P que están dentro
//del DIV con id ciudades
var parrafosCiudades = $$('# ciudades p');
```

La función \$A()

Su funcionalidad es la de convertir en matriz o array todo lo que puede asemejarse a un array. Esto aplica para algunas funciones de JavaScript como `getElementsByTagName()` que devuelve un objeto de tipo `NodeList`, que no es un array exactamente; sin embargo, puede ser recorrido como tal.

La función \$R()

Sirve para crear rangos de valores. El rango de valores se crea desde el valor del primer argumento hasta el valor del segundo argumento, inclusive. El tercer parámetro de la función indica si se excluye o no el último valor. Por defecto, el tercer argumento tiene valor `false`, que indica que si se debe incluir el valor del segundo argumento en el rango.

III. MATERIALES Y EQUIPO

Para la realización de la guía de práctica se requerirá lo siguiente:

No.	Requerimiento	Cantidad
1	Guía de práctica #11: Utilización de AJAX en aplicaciones web	1
2	Computadora con Wamp Server y Sublime Text 3 instalado	1
3	Memoria USB o disco flexible	1

IV. PROCEDIMIENTO

Digite los *scripts* que se enumeran a continuación y guárdelos en el formato apropiado como páginas `.html` o como scripts `.php` según se indique.

Ejemplo #1: El siguiente ejemplo muestra cómo utilizar AJAX para llenar las opciones de un elemento `SELECT` que depende de la selección hecha en otro elemento `SELECT`.

Script 1: dynamicform.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <meta name="author" content="Ricardo Elías">
  <title>Control dinámico de los datos de un formulario</title>
  <link rel="stylesheet" href="css/dynamicForms.css" />
  <!--[if lt IE 9]>
    <script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
  <script src="js/ajaxBasics.js"></script>
  <script src="js/dynamicForms.js"></script>
</head>
<body>
<section>
<article>
```

```
<form action="javascript:void(0);" method="GET" id="myForm" name="myForm">
  <fieldset>
    <legend>Información personal</legend>
    <label for="firstName">Nombre:</label>
    <input class="formInputText" type="text" name="firstName" id="firstName"
placeholder="Nombre" maxlength="16" tabindex="1" /><br />
    <label for="lastName">Apellido:</label>
    <input class="formInputText" type="text" name="lastName" id="lastName"
placeholder="Apellido" maxlength="16" tabindex="2" /><br />
    <label for="address1">Dirección:</label>
    <input class="formInputText" type="text" name="address1" id="address1"
placeholder="Dirección 1" maxlength="24" tabindex="3" /><br />
    <label>&nbsp;  </label>
    <input class="formInputText" type="text" name="address2" id="address2"
placeholder="Dirección 2" maxlength="24" tabindex="4" /><br/>
    <label for="estado">País: </label>
    <select class="formSelect" name="country" id="country" tabindex="5">
      <option value="(Elige un país)" selected="selected">(Elige un país)</option>
      <option>Estados Unidos</option>
      <option>Canadá</option>
      <option>México</option>
    </select>
    <label for="state">Estado/Provincia:</label>
    <select class="formSelect" name="state" id="state" size="1" tabindex="5">
      <option selected="selected">(Elige un país primero)</option>
    </select>
    <div style="clear:both;">
      <label>&nbsp;  </label>
      <input class="formInputButton" type="submit" name="submitButtonName"
id="submitButtonName" value="Enviar" tabindex="9" />
    </div>
  </fieldset>
</form>
</article>
</section>
</body>
</html>
```

Script 2: ajaxBasics.js

```
function fetchData(url,dataRequested,objectID) {
  if (dataRequested) {
    var dataRequest = "dataRequest=" + dataRequested;
  }
  var pageRequest = false;
  if (window.XMLHttpRequest) {
    pageRequest = new XMLHttpRequest();
  }
  else if (window.ActiveXObject){
    try {
      pageRequest = new ActiveXObject("Msxml2.XMLHTTP");
    }
    catch (e) {
      try{
        pageRequest = new ActiveXObject("Microsoft.XMLHTTP");
      }
      catch (e){}
    }
  }
  else{
    return false;
  }

  pageRequest.onreadystatechange = function() {
```



```

        filterData(pageRequest,objectID);
    }
    if (dataRequested) {
        pageRequest.open('POST', url, true);
        pageRequest.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded');
        pageRequest.send(dataRequest);
    }
    else {
        pageRequest.open('GET', url, true);
        pageRequest.send(null);
    }
}

```

Script 3: dynamicForms.js

```

//Archivo: dynamicForms.js
function init(){
    var pais;
    var i = 0;
    var paisField = document.myForm.country;
    var textFields = document.getElementsByTagName("formInputText");
    if(paisField.addEventListener){
        //Al cambiar la opción del campo select name="country"
        //Detectar el nuevo país seleccionado y llamar a la función chooseCountry()
        paisField.addEventListener("change", function(){
            pais
document.myForm.country.options[document.myForm.country.selectedIndex].text;
            chooseCountry(pais,
document.getElementsByTagName("select")[1].getAttribute("name"));
        }, false);
    }
    else if(paisField.attachEvent){
        paisField.addEventListener("onchange", function(){
            pais
document.myForm.country.options[document.myForm.country.selectedIndex].text;
            chooseCountry(pais,
document.getElementsByTagName("select")[1].getAttribute("name"));
        });
    }
    /* Al existir el atributo placeholder, esto ya no es necesario
    for(i=0; i<textFields.length; i++){
        if(textFields[i].addEventListener){
            textFields[i].addEventListener("focus", function(){
                clearField(this);
            }, false);
        }
        else if(textFields[i].attachEvent){
            textFields[i].addEventListener("onfocus", function(){
                clearField(this);
            });
        }
    }
    */
}

function clearField(obj){
    if(obj.defaultValue == obj.value) obj.value = '';
}

function chooseCountry(requestedData, objectID){
    fetchData('dataPage.php', requestedData, objectID);
}

function filterData(pageRequest, objectID){

```

```

        if(pageRequest.readyState == 4 && (pageRequest.status == 200 ||
window.location.href.indexOf("http") == -1)){
            var object = document.getElementById(objectID);
            object.options.length = 0;
            if(pageRequest.responseText == '(Elige un país primero)'){
                alert("No elegiste un país");
            }
            if(pageRequest.responseText != ''){
                var arrSecondaryData = pageRequest.responseText.split(',');
                for(i=0; i<arrSecondaryData.length; i++){
                    if(arrSecondaryData[i] != ''){
                        object.options[object.options.length] = new Option(arrSecondaryData[i],
arrSecondaryData[i]);
                        //alert("Agregado " + arrSecondaryData[i]);
                    }
                }
            }
        }
    }
}

if(window.addEventListener){
    window.addEventListener("load", init, false);
}
else if(window.attachEvent){
    window.attachEvent("onload", init);
}

```

Script 4: dataPage.php

```

<?php
    $dataOptions = array();
    $dataOptions["Estados Unidos"] = array(
        "Alabama (AL)",
        "New York (NY)",
        "California (CA)",
        "Arkansas (AR)",
        "Florida (FL)",
        "Arizona (AZ)",
        "Colorado (CO)"
    );

    $dataOptions["Canadá"] = array(
        "Alberta (AB)",
        "British Columbia (BC)",
        "Ontario (ON)",
        "Quebec (QC)",
        "Toronto (TO)"
    );

    $dataOptions["México"] = array(
        "Guadalajara (GU)",
        "Monterrey (MO)",
        "Veracruz (VE)",
        "Queretaro (QU)",
        "Morelia (MO)",
        "Toluca (TO)"
    );

    if(isset($_POST["dataRequest"]) && isset($dataOptions[$_POST["dataRequest"]])){
        if(in_array($_POST["dataRequest"], array("Estados Unidos", "Canadá", "México"))){
            foreach($dataOptions[$_POST["dataRequest"]] as $secondaryOptions){
                printf("%s,", $secondaryOptions);
            }
        }
        else {
            $secondaryOptions = "(Elige un país primero)";
            printf("%s", $secondaryOptions);
        }
    }

```

```
}
}
?>
```

Al visualizarlo en el navegador de su preferencia podrá observar lo siguiente:

INFORMACIÓN PERSONAL

Nombre:

Apellido:

Dirección:

País:

Estado/Provincia:

Al seleccionar un país en el campo SELECT del formulario el campo SELECT dependiente con los Estados/Provincias se actualizará después de hacer una petición con AJAX al servidor web.

Resultado personal:

localhost/DSS_Guia12_LM242664/ejemplo1/dynamicform.html

CONTACDB intersys-dev · GitLab

INFORMACIÓN PERSONAL

Nombre:

Apellido:

Dirección:

País:

Estado/Provincia:

Jose Adrian Lopez Medina / LM242664 / Tecnico en Ingeniera en Computacion Ciclo I - 2025

Ejemplo #2: El siguiente ejemplo permite previsualizar las páginas web a las que vinculan una serie de enlaces apuntando con el ratón encima de uno de estos enlaces.

Script 1: prevlinks.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Previsualizar enlaces con AJAX</title>
  <meta charset="utf-8" />
  <link type="text/css" rel="stylesheet" href="css/prevlinks.css" />
  <script type="text/javascript" src="js/prevlinks.js"></script>
  <style type="text/css">
    ul
      {
        margin:0;
        padding:0;
        list-style-type:none;
      }
    ul li
      {
        background:url(..img/bullet.gif) no-repeat 0 50%;
        padding-left:30px;
      }
    li a:link
      {
        font:Bold 14pt "Arial Black";
        color:rgb(15,30,120);
        text-decoration:none;
      }
    li a:visited
      {
        font:Bold 14pt "Arial Black";
        color:rgb(60,30,10);
        text-decoration:none;
      }
    li a:focus, li a:active
      {
        font:Bold 14pt "Arial Black";
        color:rgb(10,75,15);
        text-decoration:none;
      }
  </style>
</head>
<body>
<header>
  <h1>Países de centroamérica</h1>
</header>
<section>
<article>
```

```
<ul>
  <li><a href="prevlinks.php?id=es">El Salvador</a></li>
  <li><a href="prevlinks.php?id=ho">Honduras</a></li>
  <li><a href="prevlinks.php?id=gu">Guatemala</a></li>
  <li><a href="prevlinks.php?id=ni">Nicaragua</a></li>
  <li><a href="prevlinks.php?id=cr">Costa Rica</a></li>
  <li><a href="prevlinks.php?id=be">Belice</a></li>
</ul>
<div id="previewWin"></div>
</article>
</section>
</body>
</html>
```

Script 2: prevlinks.js

```
window.onload = initAll;
var xhr = false;
var xPos, yPos;

function initAll(){
  var allLinks = document.getElementsByTagName("a");
  for(var i=0; i<allLinks.length; i++){
    allLinks[i].onmouseover = showPreview;
  }
}

function showPreview(evt){
  getPreview(evt);
  return false;
}

function hidePreview(){
  document.getElementById("previewWin").style.visibility = "hidden";
}

function getPreview(evt){
  if(evt){
    var url = evt.target;
  }
  else{
    evt = window.event;
    var url = evt.srcElement;
  }
  xPos = evt.clientX;
  yPos = evt.clientY;
  if(window.XMLHttpRequest){
    xhr = new XMLHttpRequest();
  }
  else{
    if(window.ActiveXObject){
      try{
        xhr = new ActiveXObject("Microsoft.XMLHTTP");
      }
      catch(e){}
    }
  }

  if(xhr){
    xhr.onreadystatechange = showContents;
    xhr.open("GET", url, true);
    xhr.send(null);
  }
  else{
```

```

        alert("ERROR: No se ha podido crear objeto XMLHttpRequest");
    }
}

function showContents(){
    var prevWin = document.getElementById("previewWin");
    if(xhr.readyState == 4){
        prevWin.innerHTML = (xhr.status == 200) ? xhr.responseText : "Hay un problema con la
solicitud: " + xhr.status + " - " + xhr.statusText;
        prevWin.style.top = parseInt(yPos) + 2 + "px";
        prevWin.style.left = parseInt(xPos) + 2 + "px";
        prevWin.style.visibility = "visible";
        prevWin.onmouseout = hidePreview;
    }
}
}

```

Script 3: prevlinks.php

```

<?php
    $host = "localhost";
    $user = "root";
    $pass = "";
    $db = "paises";
    $cn = new mysqli($host, $user, $pass, $db);
    $cn->set_charset("utf8");
    $qr = "SELECT pais, capital, idioma, moneda, bandera, escudo FROM pais ";
    $qr .= "WHERE idpais='" . $_GET['id'] . "'";
    $rs = $cn->query($qr);
    $pais = $rs->fetch_object();
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Información del país</title>
    <meta charset="utf-8" />
    <style type="text/css">
        .infoPais
        {
            width:64%;
            margin:0 auto;
            border:4px outset rgb(60,20,10);
            border-collapse:collapse;
        }
        .infoPais th
        {
            background-color:rgb(180,200,90);
            color:rgb(60,20,20);
            font:11pt bold "Lucida Sans Unicode";
        }
    </style>
</head>
<body>
<section>
<article>
<?php
    $tabla = "<table class=\"infoPais\">\n";
    $tabla .= "<tr>\n<th colspan=\"2\">\n" . html_entity_decode($pais->pais) .
"</th>\n</tr>\n";
    $tabla .= "<tr>\n<td>\nCapital</td>\n";
    $tabla .= "<td>\n" . html_entity_decode($pais->capital) . "</td>\n</tr>\n";
    $tabla .= "<tr>\n<td>\nIdioma</td>\n";
    $tabla .= "<td>\n" . html_entity_decode($pais->idioma) . "</td>\n</tr>\n";
    $tabla .= "<tr>\n<td>\nMoneda</td>\n";
    $tabla .= "<td>\n" . html_entity_decode($pais->moneda) . "</td>\n</tr>\n";
    $tabla .= "<tr>\n<td>\nBandera</td>\n";
    $tabla .= "<td>\n<img src=\"\" . $pais->bandera . \"\" /></td>\n</tr>\n";

```

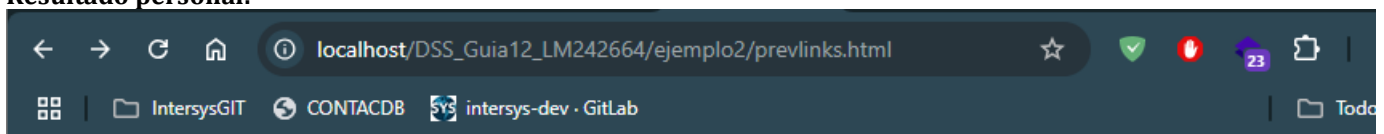
```
$tabla .= "<tr>\n<td>\nEscudo</td>\n";
$tabla .= "<td>\n<img src=\"\" . $pais->escudo . \"\" /></td>\n</tr>\n";
$tabla .= "<tr>\n<td colspan=\"2\">\n";
$tabla .= "<a href=\"prevlinks.html\">Regresar</a>\n";
$tabla .= "</td>\n</tr>\n";
$tabla .= "</table>\n";
echo $tabla;
?>
</article>
</section>
</body>
</html>
```

Al cargarlo en el navegador de su preferencia observará que al posicionarse con el ratón encima de cualquiera de los enlaces aparecerán prevvisualizaciones de las páginas a las que llevan los enlaces:

Países de centroamérica



Resultado personal:



Países de centroamérica



ico en Ingenieri a en Computacion Ciclo I - 2025

Ejemplo #3: El siguiente ejemplo muestra una aplicación que utiliza pestañas para mostrar tres tipos de contenido, uno en cada pestaña. La información que se muestra en cada pestaña se carga directamente de una base de datos usando AJAX.

Script 1: tabs.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <title>Ventana con pestañas usando AJAX</title>
  <script type="text/javascript" src="js/ahahLib.js"></script>
  <script type="text/javascript" src="js/activetab.js"></script>
  <link rel="stylesheet" type="text/css" href="css/tabs.css" />
</head>
<body>
<section>
<article>
<ul id="tabmenu">
  <li onclick="makeactive(1)">
    <a class="" id="tab1">Noticias</a>
  </li>
  <li onclick="makeactive(2)">
    <a class="" id="tab2">Sociales</a>
  </li>
  <li onclick="makeactive(3)">
    <a class="" id="tab3">Deportes</a>
  </li>
</ul>
<div id="content">
  (Aquí se mostrará el contenido de cada pestaña)
```



```
</div>
</article>
</section>
</body>
</html>
```

Script 2: ahahLib.js

```

/*****
/*  AHAAH functions by Phil Ballard
/*  Este código ha sido desarrollado únicamente con propósitos de estudio.
/*  Puedes hacer uso de este código, ya sea con propósito comercial o no.
/*  sin embargo; note que el autor no garantiza su utilidad, idoneidad
/*  o precisión y se exime de cualquier responsabilidad por la pérdida
/*  de datos en el uso del mismo.
*****/

function callAHAH(url, pageElement, callMessage, errorMessage) {
    document.getElementById(pageElement).innerHTML = callMessage;
    try {
        /* Par navegadores como Firefox, Safari, Chrome, Opera, etc */
        req = new XMLHttpRequest();
    }
    catch(e) {
        try {
            req = new ActiveXObject("Msxml2.XMLHTTP"); /* some versions IE */
        }
        catch (e) {
            try {
                req = new ActiveXObject("Microsoft.XMLHTTP"); /* some versions IE */
            }
            catch (E) {
                req = false;
            }
        }
    }
    req.onreadystatechange = function() {
        responseAHAH(pageElement, errorMessage);
    };
    req.open("GET",url,true);
    req.send(null);
}

function responseAHAH(pageElement, errorMessage) {
    var output = '';
    if(req.readyState == 4) {
        if(req.status == 200) {
            output = req.responseText;
            document.getElementById(pageElement).innerHTML = output;
        }
        else {
            document.getElementById(pageElement).innerHTML = errorMessage+"\n"+output;
        }
    }
}

```

Script 3: activetab.js

```

function makeactive(tab) {
    document.getElementById("tab1").className = "";
    document.getElementById("tab2").className = "";
    document.getElementById("tab3").className = "";
    document.getElementById("tab" + tab).className = "active";
}

```

```
callAAAH('content.php?content=' + tab, 'content', 'Esperando para cargar el contenido de
la pestaña' + tab + '. Espere, por favor...', 'Error!');
}
```

Script 4: content.php

```
<?php
$host = "localhost";
$user = "root"; //Cambiar por el usuario de la base de datos en su servidor
$pass = ""; //Cambiar por la contraseña de la base de datos en su servidor
$db = "prensa";
//Establecer conexión con el servidor MySQL
$cn = new mysqli($host, $user, $pass, $db);
if($cn->connect_errno){
    printf("Falló la conexión: %s\n", $cn->connect_error);
    exit(0);
}
$tipo = isset($_GET['content']) ? $_GET['content'] : 0;
if($tipo != 0){
    $qr = "SELECT titulonoticia, textonoticia, imgnoticia FROM noticia ";
    $qr .= "WHERE idnota = $tipo";
    $rs = $cn->query($qr);
    while($row = $rs->fetch_object()){
        $info = "<div id=\"titulo\">\n<h2>{$row->titulonoticia}</h2>\n</div>\n";
        $info .= "<div id=\"texto\">\n<p>\n{$row->textonoticia}\n";
        $info .= "<img src=\"{$row->imgnoticia}\n\" />\n";
        $info .= "</p>\n</div>\n";
    }
    echo utf8_encode($info);
}
?>
```

Al visualizar el ejemplo en el navegador de su preferencia, obtendremos lo siguiente:



Resultado personal:

localhost/DSS_Guia12_LM242664/ejemplo3/tabs.html
IntersysGIT
CONTACDB
intersys-dev · GitLab

Noticias
Sociales
Deportes


Obispos salvadoreños dicen que el Papa Francisco ora por canonización de Monseñor Romero

Los obispos salvadoreños se declararon regocijados por la visita Ad Limina Apostolorum (visita a las tumbas de los Apóstoles) en el Vaticano y su encuentro el papa Francisco, lo que muchos ven como una "cosa providencial que estemos en Roma (Italia) en estos días", refiriéndose a la víspera de los 37 años del asesinato del beato Óscar Romero.

"Ha sido una emoción estar casi dos horas con el Papa Francisco comentando sobre El Salvador", monseñor Fabio Colindres, uno de los obispos que viajó a Roma.

Monseñor Gregorio Rosa Chávez, también mostró su emoción, por la coincidencia de la visita y aseguró "que nunca en la vida había pasado algo parecido", además que destacó que la reunión era sin agenda y parte de ello significaba la discusión del tema monseñor Romero.

Colindres aseguró que también hablaron de la realidad que vive el país y que el pontífice los animó en la fe, para orar por la violencia. Agregó que fue importante y "feliz discutir el tema de monseñor Romero" y aseguró que su Santidad "también esta orando, unido a nosotros, por una pronta canonización".



MPG MULTIMEDIA

©EPA

Jose Adrian Lopez Medina / LM242664 / Tecnico en Ingenieria en Computacion Ciclo I - 2025

Ejercicio #4: El siguiente ejemplo muestra un formulario que permite ir insertando registros y a la vez mostrándolos insertando el registro en la última fila de la tabla con los datos de los usuarios que se van añadiendo.

Script 1: registroempleado.php

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Registro de empleados</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Roboto:400" />
  <link rel="stylesheet" href="css/styles.css" />
  <link rel="stylesheet" href="css/material-style.css">
</head>
<body>
<header>
  <h1>Nuevo empleado</h1>
</header>
<section>
<article>
<div class="form-box">
  <div class="head">Ingreso de datos</div>
  <form name="nuevo_empleado" action="" id="login-form">
    <div class="form-group">
      <label for="nombre" class="label-control">
        <span class="label-text">Nombre:</span>
      </label>
      <input type="text" name="nombre" id="nombre" maxlength="25" accesskey="n"
tabindex="1" class="form-control" />
    </div>
    <div class="form-group">
      <label for="apellido" class="label-control">
        <span class="label-text">Apellido</span>
      </label>
      <input type="text" name="apellido" id="apellido" maxlength="25" accesskey="a"
tabindex="2" class="form-control" />
    </div>
    <div class="form-group">
      <label for="correo" class="label-control">
        <span class="label-text">Correo</span>
      </label>
      <input type="text" name="correo" id="correo" maxlength="25" accesskey="c"
tabindex="3" class="form-control" />
    </div>
    <input type="submit" name="guardar" id="guardar" value="Guardar" accesskey="g"
tabindex="4" class="btn" />
  </form>
</div>
</article>
</section>
<div id="resultado"><?php include('consulta.php');?></div>
</body>
<script type="text/javascript" src="js/ajax.js"></script>
<script type="text/javascript" src="js/validateform.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
<script>
  $(window).load(function(){
    $('.form-group input').on('focus blur', function (e) {
      $(this).parents('.form-group').toggleClass('active', (e.type === 'focus' ||
this.value.length > 0));
    }).trigger('blur');
```

```
});
</script>
</html>
```

Script 2: registro.php

```
<?php
//Creando la instancia del manejador de base de datos
include_once("class/database.class.php");

//variables POST
$nom = isset($_POST['nombre']) ? trim(($con->real_escape_string($_POST['nombre']))) : "";
$aape = isset($_POST['apellido']) ? trim(($con->real_escape_string($_POST['apellido']))) : "";
$cor = isset($_POST['correo']) ? trim($con->real_escape_string($_POST['correo'])) : "";

//registra los datos del empleados
$sql="INSERT INTO empleados (nombre, apellido, correo) VALUES ('$nom', '$aape', '$cor')";

$res = $con->query($sql);

if(!$res){
    echo "Error: El registro no se ha podido ingresar.";
    exit(0);
}

include('consulta.php');
?>
```

Script 3: consulta.php

```
<?php
//Creando la instancia del manejador de base de datos
include_once("class/database.class.php");

//consulta todos los empleados
$sql = "SELECT nombre, apellido, correo FROM empleados";
$res = $con->query($sql);
?>
<table id="datos">
    <thead>
        <tr>
            <th>No.</th>
            <th>Nombre</th>
            <th>Apellido</th>
            <th>Correo</th>
        </tr>
    </thead>
    <tbody>
<?php
    $i = 0;
    //Con método fetch_array()
    /* while($row = $res->fetch_array()){
        echo "<tr>";
        echo "<td>" . ($i+1) . "</td>";
        echo "<td>" . $row['nombre'] . "</td>";
        echo "<td>" . $row['apellido'] . "</td>";
        echo "<td>" . $row['correo'] . "</td>";
        echo "</tr>";
        $i++;
    } */
    //Con método fetch_object()
    while($row = $res->fetch_object()){
        echo "\t<tr>\n";
        echo "\t\t<td>" . ($i+1) . "</td>\n";
```

```

        echo "\t\t<td>" . $row->nombre . "</td>\n";
        echo "\t\t<td>" . $row->apellido . "</td>\n";
        echo "\t\t<td>" . $row->correo . "</td>\n";
        echo "\t</tr>\n";
        $i++;
    }
    ?>
    </tbody>
</table>

```

Script 4: database.class.php

```

<?php
//Configuracion de la conexion a base de datos
$bd_host = "localhost";
$bd_usuario = "root";
$bd_password = "";
$bd_base = "empleadosajax";

$con = new mysqli($bd_host, $bd_usuario, $bd_password, $bd_base);
$con->set_charset("utf8");

if($con->connect_error){
    echo "<p>Se ha producido el error número " . $con->connect_errno . "Con la siguiente
    descripción " . $con->connect_error . "</p>";
    exit(0);
}
?>

```

Script 5: ajax.js

```

// Función para recoger los datos de PHP según el navegador, se usa siempre.
function objetoAjax(){
    var xmlhttp=false;
    try {
        xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try {
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (E) {
            xmlhttp = false;
        }
    }

    if (!xmlhttp && typeof XMLHttpRequest!='undefined') {
        xmlhttp = new XMLHttpRequest();
    }
    return xmlhttp;
}

//Función para recoger los datos del formulario y enviarlos por post
function enviarDatosEmpleado(){
    //div donde se mostrará lo resultados
    divResultado = document.getElementById('resultado');
    //recogemos los valores de los inputs
    nom = document.nuevo_empleado.nombre.value;
    ape = document.nuevo_empleado.apellido.value;
    cor = document.nuevo_empleado.correo.value;

    //alert(nom + " | " + ape + " | " + cor);

    //instanciamos el objetoAjax
    var ajax = objetoAjax();

```

```
//Uso del metodo POST
//archivo que realizará la operacion
//registro.php
ajax.open("POST", "registro.php", true);

//cuando el objeto XMLHttpRequest cambia de estado, la función se inicia
ajax.onreadystatechange = function() {
    //la función.responseText tiene todos los datos pedidos al servidor
    if(ajax.readyState == 4 && ajax.status == 200) {
        //mostrar resultados en esta capa
        divResultado.innerHTML = ajax.responseText;
        //llamar a funcion para limpiar los inputs
        LimpiarCampos();
    }
}

ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
//enviando los valores a registro.php para que inserte los datos
ajax.send("nombre=" + nom + "&apellido=" + ape + "&correo=" + cor);
}

//función para limpiar los campos
function LimpiarCampos(){
    document.nuevo_empleado.nombre.value = "";
    document.nuevo_empleado.apellido.value = "";
    document.nuevo_empleado.correo.value = "";
    document.nuevo_empleado.nombre.focus();
}
}
```

Script 6: validateform.js

```
/*
*****
* Validar Formulario Accesible sin onSubmit *
* (c) 2009 Alejandro Arco *
* Por Alejandro Arco - http://www.alejandroarco.es *
*****
*/
validateForm = function() {
    var nombre = new Array(document.getElementById('nombre'),
document.nuevo_empleado['nombre'].value);
    var apellido = new Array(document.getElementById('apellido'),
document.nuevo_empleado['apellido'].value);
    var correo = new Array(document.getElementById('correo'),
document.nuevo_empleado['correo'].value);
    var submit = document.getElementById('guardar');

    /* Eventos */
    nombre[0].onfocus = onFocus;
    nombre[0].onblur = onBlur;
    apellido[0].onfocus = onFocus;
    apellido[0].onblur = onBlur;
    correo[0].onfocus = onFocus;
    correo[0].onblur = onBlur;
    submit.onclick = validateFields;

    /* Funciones */
    function onFocus() {
        if(document.nuevo_empleado[this.name].value == eval(this.name)[1])
            document.nuevo_empleado[this.name].value='';
    }

    function onBlur() {
        var value = document.nuevo_empleado[this.name].value;
        value = value.replace(/^\s*/, '');
        value = value.replace(/\s*$/, '');
        if(!value) document.nuevo_empleado[this.name].value = eval(this.name)[1];
    }
}
```

```
}

function validateFields() {
    if(document.nuevo_empleado['nombre'].value == nombre[1]) {
        document.nuevo_empleado['nombre'].focus();
        alert('El campo "nombre" es obligatorio.');
```

```
    }
    else if(document.nuevo_empleado['apellido'].value == apellido[1]) {
        document.nuevo_empleado['apellido'].focus();
        alert('El campo "apellido" es obligatorio.');
```

```
    }
    else if(document.nuevo_empleado['correo'].value == correo[1]) {
        document.nuevo_empleado['correo'].focus();
        alert('El campo "correo" es obligatorio.');
```

```
    }
    else {
        enviarDatosEmpleado();
        //return true;
    }
    return false;
}

}

if(document.all && window.attachEvent) {
    window.attachEvent("onload",validateForm); // IE-Win
}
else if(window.addEventListener) {
    window.addEventListener("load",validateForm,false); // Otros
}
```

El resto de archivos necesarios para completar el ejercicio serán proporcionados en los recursos de la guía de práctica.

Al cargar el script registroempleado.php en el navegador de su preferencia podrá visualizar lo siguiente:



INICIO DE SESIÓN

Nombre:
María Isabel

Apellido
Benítez Umaña

Correo
mari_beni09@outlook.com

GUARDAR

Guía #12: Utilización de AJAX en aplicaciones web

No.	Nombre	Apellido	Correo
1	Julio	Rivera	juliox@gmail.com
2	Lorena	Cortéz	lorencortez@gmail.com
3	Karla	Ramírez	karlaram@hotmail.com
4	Rebeca	Benítez	beckys@hotmail.com
5	Victor	García	vikfra@yahoo.com
6	Armando	Monterrosa	mandy@gmail.com
7	Ricardo	Elías	ricardo.ellas@udb.edu.sv
8	Gricelda	Mejía	gricel@gmail.com
9	Carlos	Castro	filicastro@hotmail.com
10	Mauricio	Bolaños	mauribol@gmail.com
11	Maria	Gutiérrez	margutierrez@hotmail.com
12	Daniel	Ochoa	dani.ochoa@yahoo.es
13	Miguel Alejandro	Murcia Bolaños	migmurcia@yahoo.es
14	Yesenia	Olivares Cañas	yolivares@gmail.com
15	Julio César	Rodríguez	julio_rodrig@hotmail.com
16	Luis	Hernández	luisudb@gmail.com
17	Beatriz	Martínez	bea@hotmail.com
18	Marcos Vinicio	Alegre	marcosalegre@hotmail.com
19	Mario	Sánchez	mario_sanchez@gmail.com
20	José Allrio	Cornejo	allrio_cornejo12@aol.com
21	Maria Isabel	Benítez Umaña	mari_beni09@outlook.com

Resultado personal:

←

→

↺

🏠

🔍 localhost/DSS_Guia12_LM242664/ejemplo4/registroempleado.php

☆

🟢

🔴

📅 23

🔖

🗄️

IntersysGIT

🔄 CONTACDB

🔗 intersys-dev · GitLab

📁 Todos

Nuevo empleado

INGRESO DE DATOS

Nombre:

Apellido

Correo

GUARDAR

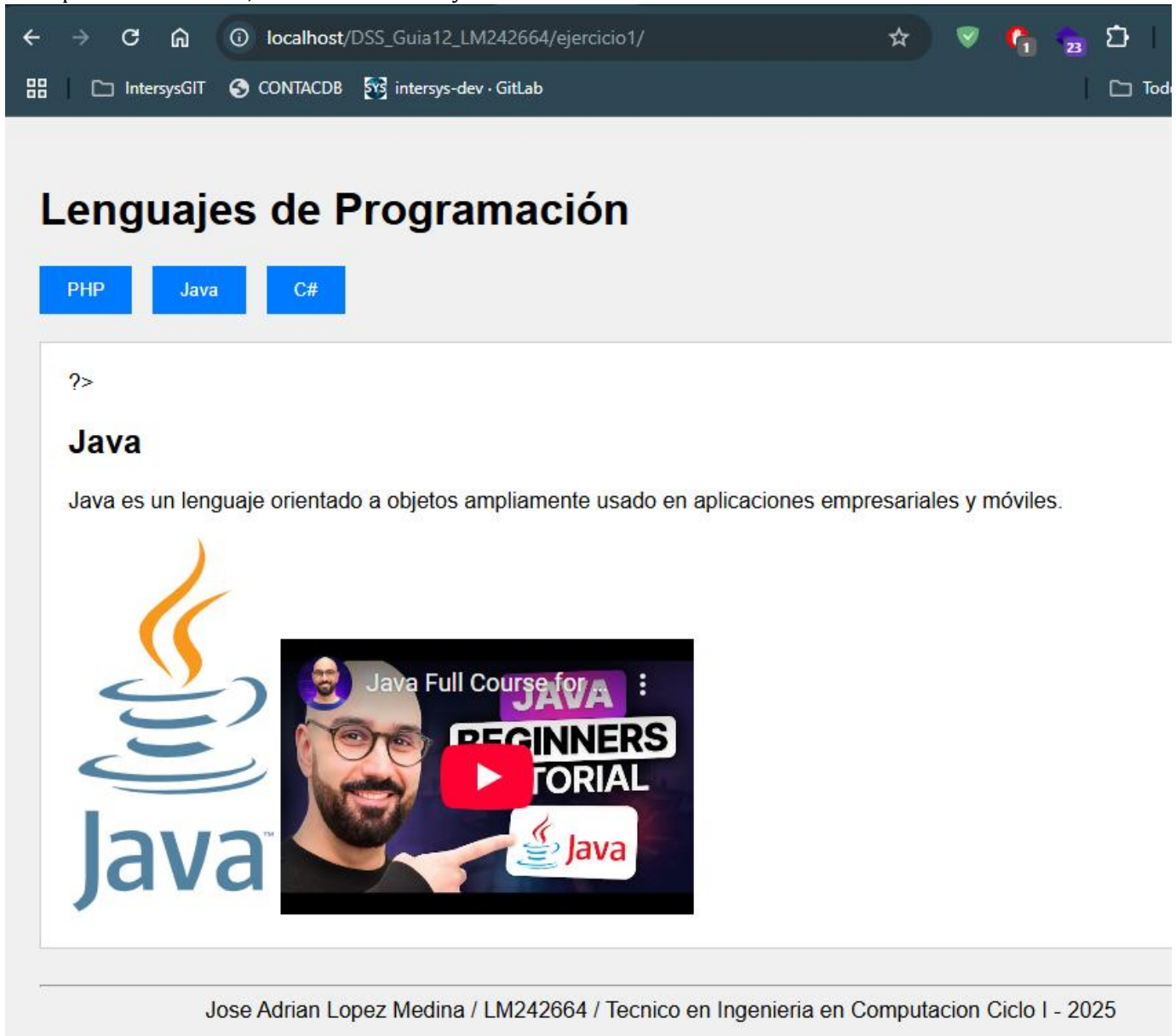
No.	Nombre	Apellido	Correo
1	Victor	García	vikfra@yahoo.com
2	Gricelda	Mejía	gricel@gmail.com
3	María	Gutiérrez	margutierrez@hotmail.com
4	Daniel	Ochoa	dani.ochoa@yahoo.es
5	Miguel Alejandro	Murcia Bolaños	migmurcia@yahoo.es
6	Yesenia	Olivares Cañas	yolivares@gmail.com
7	Julio César	Rodríguez	julio_rodrig@hotmail.com
8	María Eugenia	Calderón Flores	mar_calderon@yahoo.com
9	Yolanda Aida	Díaz Ordoñez	yolandiazord@gmail.com
10	Fausto Emilio	Guzmán Valladares	fausto_guz@hotmail.com
11	test	test	test@gmail.com

Jose Adrian Lopez Medina / LM242664 / Tecnico en Ingenieria en Computacion Ciclo I - 2025

r

V. DISCUSIÓN DE RESULTADOS


1. Realice una aplicación web con fichas tabulares para mostrar información sobre tópicos de su interés. Por ejemplo, lenguajes de programación: PHP, Java y C#, o sobre información bibliográfica: libros, tesis y revistas. Y muestre en áreas creadas con DIV información que incluya texto e imágenes u otros recursos que se puedan publicar en la web, como animaciones y vídeos.



?>

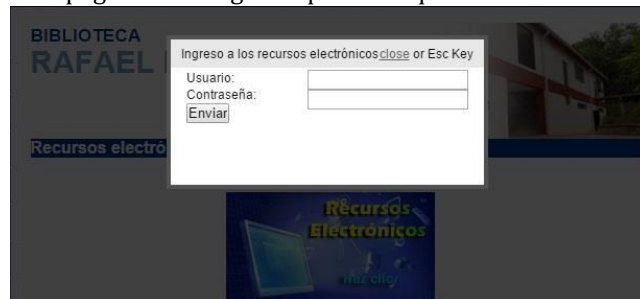
Java

Java es un lenguaje orientado a objetos ampliamente usado en aplicaciones empresariales y móviles.



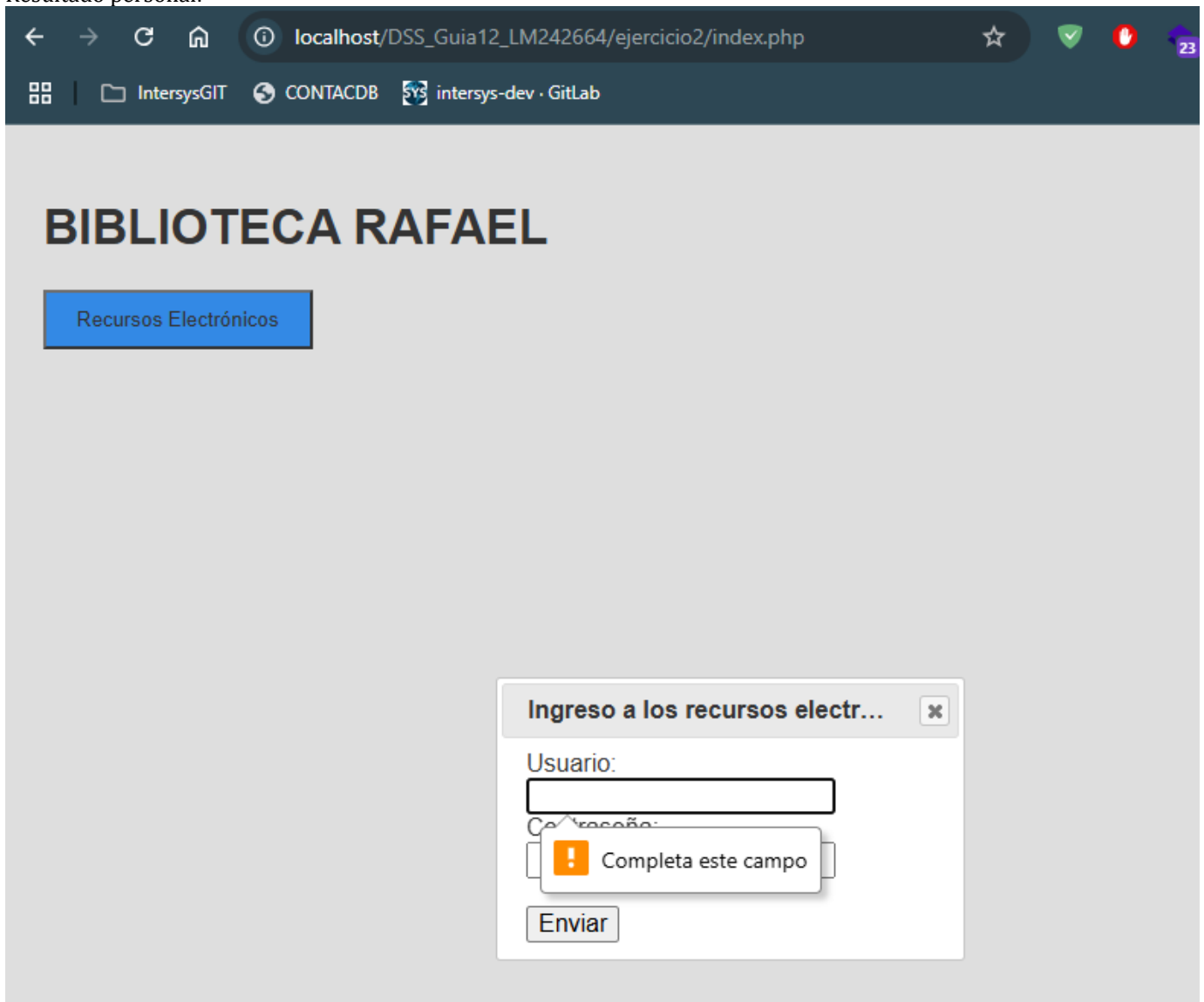
Jose Adrian Lopez Medina / LM242664 / Tecnico en Ingenieria en Computacion Ciclo I - 2025

2. Cree un formulario de inicio de sesión que usando algún plug-in de jQuery para enviar con AJAX los datos de autenticación que deben ser contrastados con una base de datos permita ingresar a una página web restringida al usuario que se autentique correctamente, en cuyo caso deberá mostrar el nombre completo del usuario en alguna parte de esa página restringida. Tome como ejemplo el siguiente formulario de autenticación de usuario y una página restringida hipotética que deberá mostrar:





Resultado personal:





VI. BIBLIOGRAFIA

- Lee Babin. Introducción a AJAX con PHP. 1a. Edición. Editorial Anaya Multimedia. Madrid, España 2007.
- Maximiliano Firtman. AJAX Web 2.0 para profesionales. 1ra. Edición. Editorial Alfaomega. México, 2007.
- Phil Ballard / Michael Moncur. AJAX, JavaScript y PHP. 1ª. Edición en español. SAMS / Anaya Multimedia, Madrid, España. 2009.
- Tom Negrino / Dori Smith. JavaScript & AJAX para diseño web. 6ta Edición. Editorial Pearson, Prentice Hall. Madrid, España, 2007.