LENGUAJE DE PROGRAMACION II

Docente: Ing. Díaz Leva Teodoro

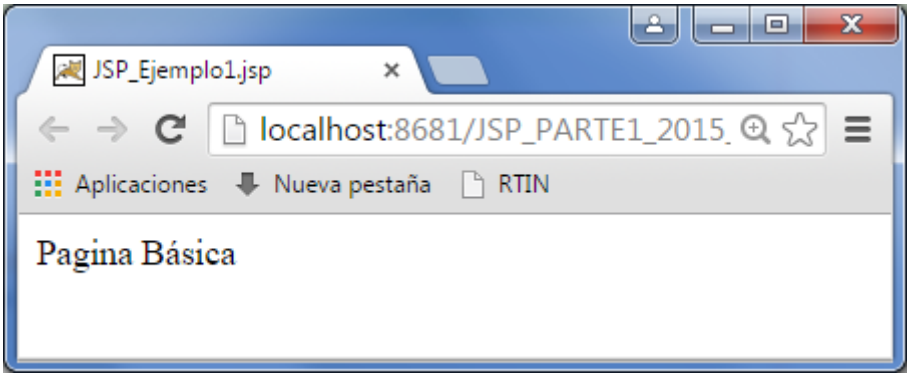
Tema: Introducción a Java Server Pages (JSP)

Un JSP ("Java Server Page") es uno de los componentes más básicos empleados para aplicaciones de Servidor en Java. Su composición consta de dos grandes partes: HTML y lenguaje Java.

Mediante HTML se especifica el contenido estático de despliegue y es mediante fragmentos del lenguaje Java que se genera contenido **dinámico** en efecto cumpliendo la definición de aplicación de Servidor; a continuación se ilustra una porción de un JSP:

```
<html>
  <head>
    <title>JSP_Ejemplo1.jsp</title>
  </head>
  <body>
    <%out.println("Pagina Básica");  %>
  </body>
</html>
```

Los elementos que se encuentran entre **<%** y **%>** son elementos Java , mientras el resto es considerado HTML puro. El método Java utilizado en la porción anterior (out.println) imprime el texto correspondiente; al ejecutarse este JSP da como resultado final:



El código fuente de la página:

```
<html>
  <head>
    <title>JSP_Ejemplo1.jsp</title>
  </head>
  <body>
            Pagina Básica
  </body>
</html>
```

La ejecución y/o transformación de elementos Java es trabajo de un "Servlet Engine" y para este curso es utilizado el "Servlet Engine" Tomcat.

El primer paso puede ser sorprendente, pero efectivamente todo JSP es convertido a un Servlet , la razón de esto es que el surgimiento de JSP's se debió a la necesidad de facilitar un formato programático sencillo para personal no familiarizado con Java. Si se modifica el código de la página JSP, entonces se regenera y recompila automáticamente el servlet y se recarga la próxima vez que sea solicitada.

En un JSP además de los Tags **<%** y **%>** existen otras variaciones que son las siguientes.

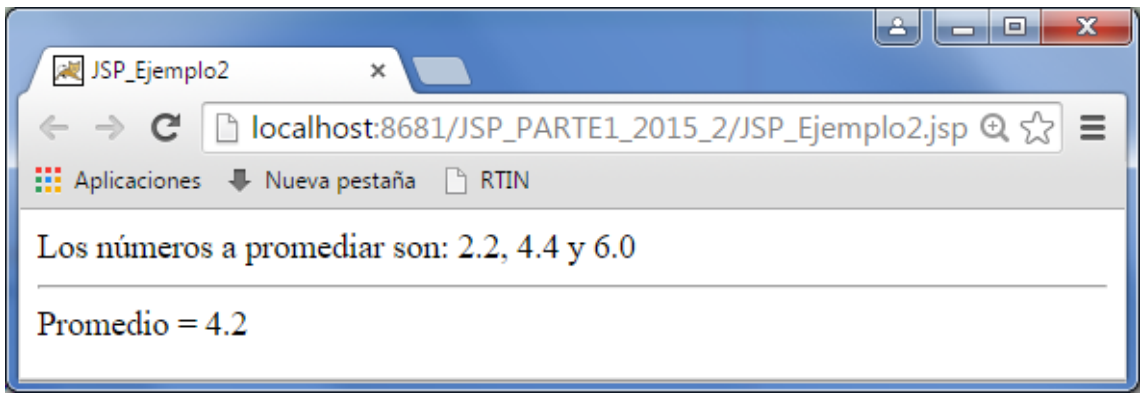
- ***<%= Expresión Java %>***
- ***<%! Declaración Java %>***
- ***<%-- Comentario Java --%>***

Un ejemplo para utilizar los tags:

```
<html>
  <head>
    <title>JSP_Ejemplo2.jsp</title>
  </head>
  <body>
    <%! double numero1=2.2, numero2=4.4, numero3=6.0;%>

    Los números a promediar son: <%=numero1%>, <%=numero2%> y <%=numero3%><br><hr>
    <%--funcion promedio--%>
    <%! public double promedio(double n1,double n2,double n3){
      return (n1+n2+n3)/3;
    } %>
    <%--llamar a la funcion promedio e imprimir--%>
    Promedio = <%=promedio(numero1,numero2,numero3)%>
  </body>
</html>
```

En el ejemplo anterior, puede verse la declaración de las variables, la declaración del método promedio, la utilización de las variables y finalmente la utilización del promedio. Para mostrar los valores tanto de las variables como el promedio se han utilizado expresiones JSP.

Ejecución de la página:



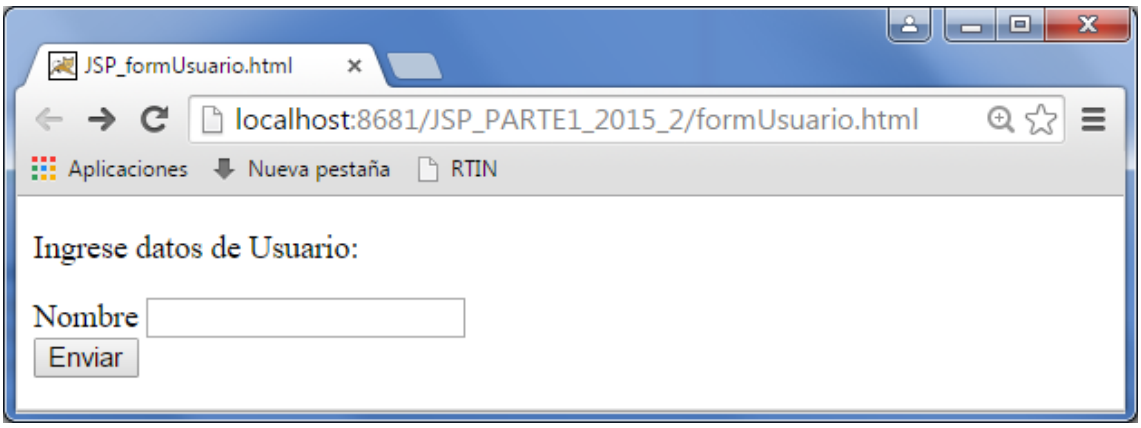
Debido a que un JSP ofrece una abstracción por arriba de un Servlet, y éste último es un **Objeto (Clase)** Java, también se tienen acceso a diversos Objetos implícitos dentro de un JSP sin la necesidad de realizar declaraciones complejas

OBJETO	USO
request	Representa el Objeto de Solicitud dentro de un JSP/Servlet, entre otras cosas los parámetros recibidos de los formularios por POST o GET
response	Representa el Objeto de Respuesta dentro de un JSP/Servlet.
pageContext	Representa el Contexto del JSP/Servlet.
session	Representa la sesión del Usuario en un JSP/Servlet.
application	Representa el Objeto de aplicación (Contexto) para un JSP/Servlet.
out	Representa el Objeto de Escritura (Para enviar a pantalla) en un JSP/Servlet.
config	Representa el Objeto de Configuración para un JSP/Servlet.
page	Representa el Objeto del JSP/Servlet en sí.
exception	Representa el Objeto de errores para un JSP/Servlet.

Ejemplo
Una aplicación que pide el nombre al usuario y le devuelve un saludo. Utiliza un archivo HTML como formulario que pide los datos al usuario y se los pasa a una página JSP que muestra el saludo con estos datos. El paso de los datos del formulario al JSP se realiza a través de un objeto implícito: objeto **request.**

Esquema del funcionamiento del formulario:

Formulario HTML: formUsuario.html



El código HTML de la página: formUsuario.html

```
<html>
  <head>
    <title>JSP_formUsuario.html</title>
  </head>
  <body>
    <p>Ingrese datos de Usuario:</p>

    <form action="mostrarSaludo.jsp" method="post">
        Nombre <input name="txtnombre" type="text"><br>
        <input value="Enviar" type="submit">
    </form>
  </body>
</html>
```

Etiquetas usadas en formUsuario.html:

- La etiqueta <form> encierra todos los contenidos del formulario (botones, cuadros de texto, casillas de verificación, listas desplegables).

  Propiedades principales del objeto form:

  action: es el lugar al cual se envía los datos del formulario para ser procesado. El action define la URL a la cual se envía dicho formulario. En nuestro ejemplo será el archivo **mostrarSaludo.jsp**

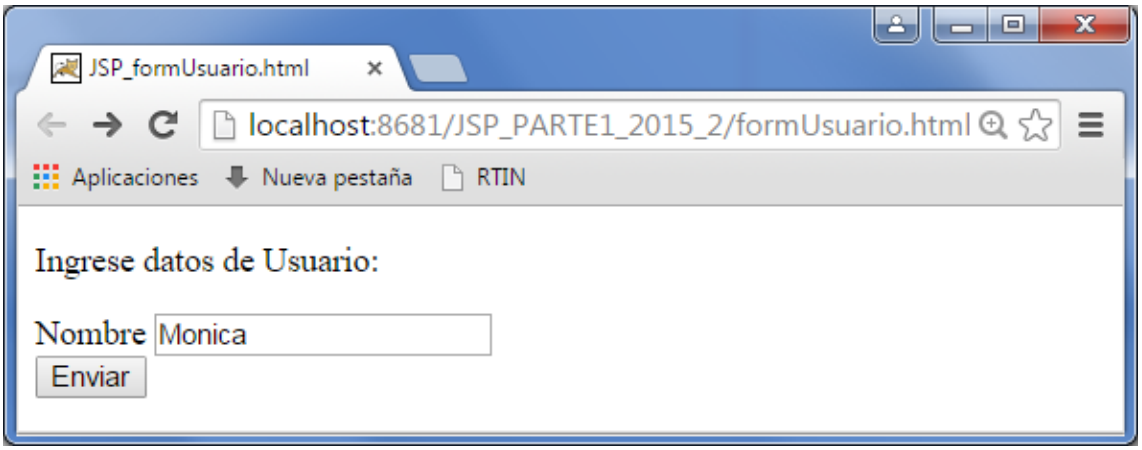
  method="post": envía los datos en forma "invisible". Conveniente para enviar una gran cantidad de datos .Existe otro valor **get** para method que envía los datos en una cadena "visible" por la URL.

- La etiqueta <input> permite definir varios tipos diferentes de elementos (botones y cuadros de texto). En este caso la etiqueta **"text"** para un cuadro de texto y **"submit"** para un botón de envio.
- La etiqueta
 es un salto de línea y <p> es un párrafo.

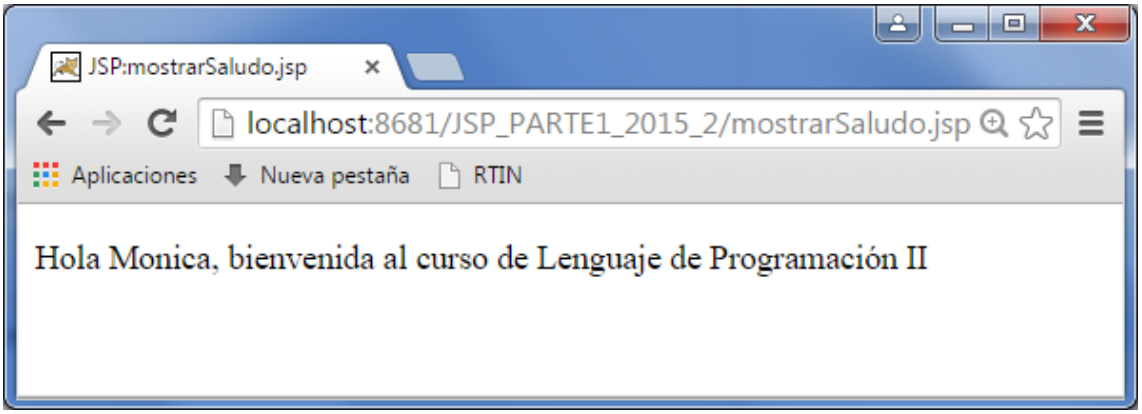
Archivo: **mostrarSaludo.jsp**, página que muestra información basada en los datos ingresado en el formulario:

```
<html>
  <head>
    <title>JSP Page</title>
  </head>
  <body>
    <!-- código java -->
    <%
   //obtener valor del cuadro de texto txtnombre del formUsuario.htm
       String nom=request.getParameter("txtnombre");
    %>

     <!-- impresión de  la variable nom-->

    <p> Hola <%=nom%> bienvenida al curso de Lenguaje de Programación II </p>
  </body>
</html>
```

Ejecución del aplicativo:



Presionar botón Enviar:



Laboratorio 1:
El objetivo es reconocer las etiquetas básicas del lenguaje HTML y escribir funciones en lenguaje JavaScript para la validación de formularios.

Procedimiento y Resultados:

HTML y JavaScript

Archivo: ingreso.html

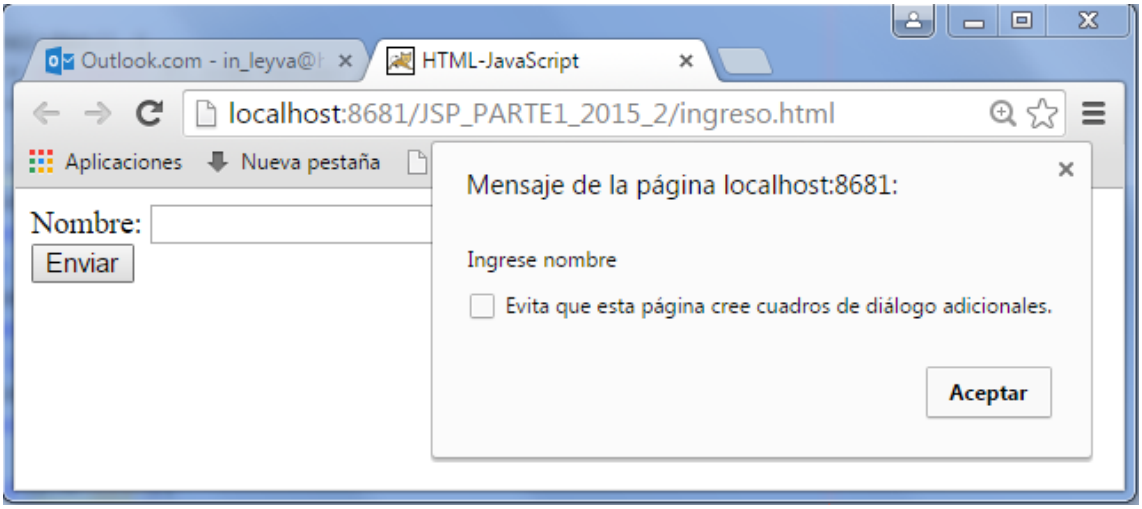
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Formulario</title>
</head>
<body>

<form name="f1" action="" method="post">
  Nombre: <input type="text" name="nom">
  Clave: <input type="password" name="clave"> <br/>
  <input type="submit" value="Enviar">
</form>

</body>
</html>
```

Modificar el ingreso.html, ahora tendrá validaciones en JavaScript:

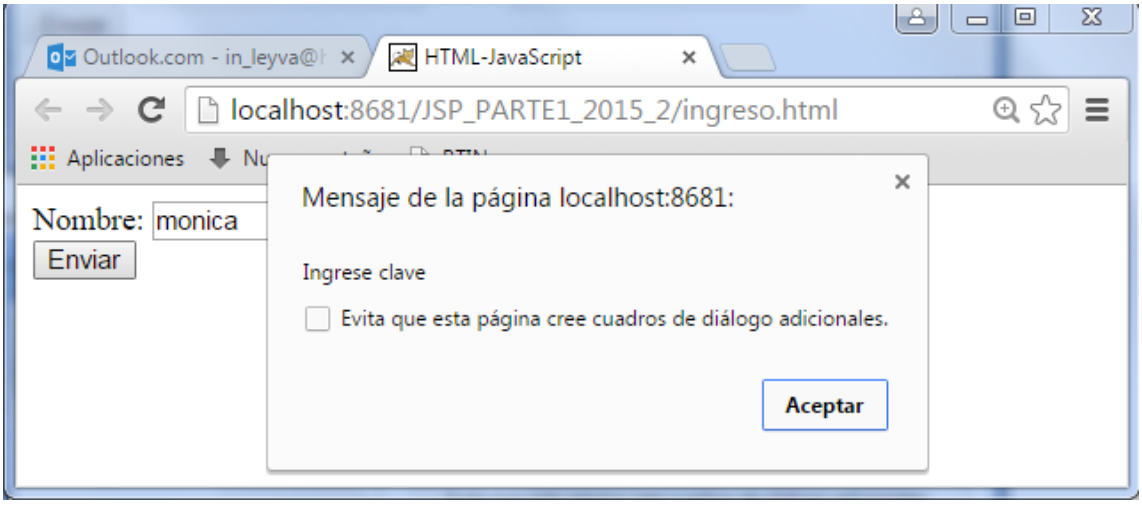
```
<html>
    <head>
        <title>HTML-JavaScript</title>
        <script>
            function validar() {
                var n = document.f1.nom.value;
                var c = document.f1.clave.value;

                if (n == "") {
                    alert('Ingrese nombre');
                    return false;
                } else if (c == "") {
                    alert('Ingrese clave');
                    return false;
                }
                return true;
            }
        </script>
    </head>
    <body>

 <form name="f1" action="" method="post" onSubmit="return validar();">
            Nombre: <input type="text" name="nom">
            Clave: <input type="password" name="clave"> <br/>
            <input type="submit" value="Enviar">
 </form>

    </body>
</html>
```

Ejecución de la aplicación:

Al intentar enviar nombre vacio:



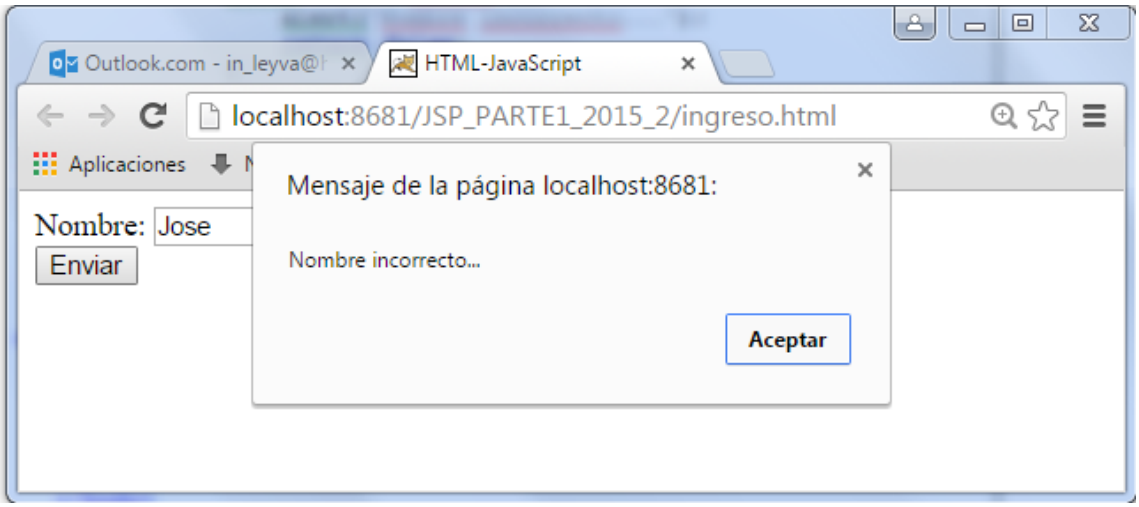
Al intentar enviar clave vacía:



Modifiquemos la función validar (), asumiendo que el usuario es **"monica"** y clave **"123"**, ahora la aplicación además de exigir ingresar nombre y clave deben ser "monica" y "123" respectivamente:

```
<html>
    <head>
        <title>HTML-JavaScript</title>
        <script>
            function validar() {
                var n = document.f1.nom.value;
                var c = document.f1.clave.value;

                if (n == "") {
                    alert('Ingrese nombre');
                    return false;
                } else if (c == "") {
                    alert('Ingrese clave');
                    return false;
                }else if(n!="monica"){
                    alert('Nombre incorrecto...');
                    return false;
                }
                else if(c!="123"){
                    alert('clave incorrecta...');
                    return false;
                }
                return true;
            }
        </script>
    </head>
    <body>
 <form name="f1" action="" method="post" onSubmit="return validar();">
   Nombre: <input type="text" name="nom">
   Clave: <input type="password" name="clave"> <br/>
          <input type="submit" value="Enviar">
  </form>

    </body>
</html>
```

Fíjese la validación por el nombre incorrecto:



Explicación de la función validar:

La principal utilidad de JavaScript en el manejo de los formularios es la validación de los datos introducidos por los usuarios. Antes de enviar un formulario al servidor, se recomienda validar mediante JavaScript los datos insertados por el usuario. De esta forma, si el usuario ha cometido algún error al rellenar el formulario, se le puede notificar de forma instantánea, sin necesidad de esperar la respuesta del servidor.

Normalmente, la validación de un formulario consiste en llamar a una función de validación cuando el usuario pulsa sobre el botón de envío del formulario. En esta función, se comprueban si los valores que ha introducido el usuario cumplen las restricciones impuestas por la aplicación.

Aunque existen tantas posibles comprobaciones como elementos de formulario diferentes, algunas comprobaciones son muy habituales: que se rellene un campo obligatorio, que se seleccione el valor de una lista desplegable, que la dirección de email indicada sea correcta, que la fecha introducida sea lógica, que se haya introducido un número donde así se requiere, etc.

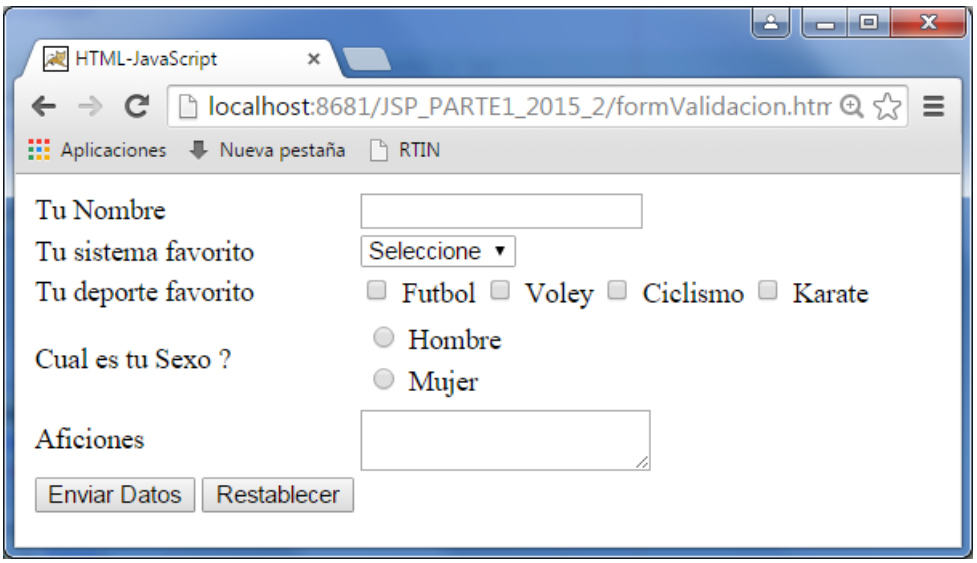
A continuación se muestra el código JavaScript básico necesario para incorporar la validación a un formulario:

```
<form action="" method="" id="" name="" onsubmit="return validar()">
  ...
</form>
```

Y el esquema de la función **validar()** es el siguiente:

```
function validar() {

  if (condicion que debe cumplir el primer campo del formulario){
    // Si no se cumple la condicion...
    alert('[ERROR] El campo debe tener un valor de...');
    return false;
  }
  else if (condicion que debe cumplir el segundo campo del formulario){
    // Si no se cumple la condicion...
    alert('[ERROR] El campo debe tener un valor de...');
    return false;
  }
  ...
  else if (condicion que debe cumplir el último campo del formulario){
    // Si no se cumple la condicion...
    alert('[ERROR] El campo debe tener un valor de...');
    return false;
  }

  // Si el script ha llegado a este punto, todas las condiciones
  // se han cumplido, por lo que se devuelve el valor true
  return true;
}
```

Realizar el siguiente formulario y validarlo con JavaScript:



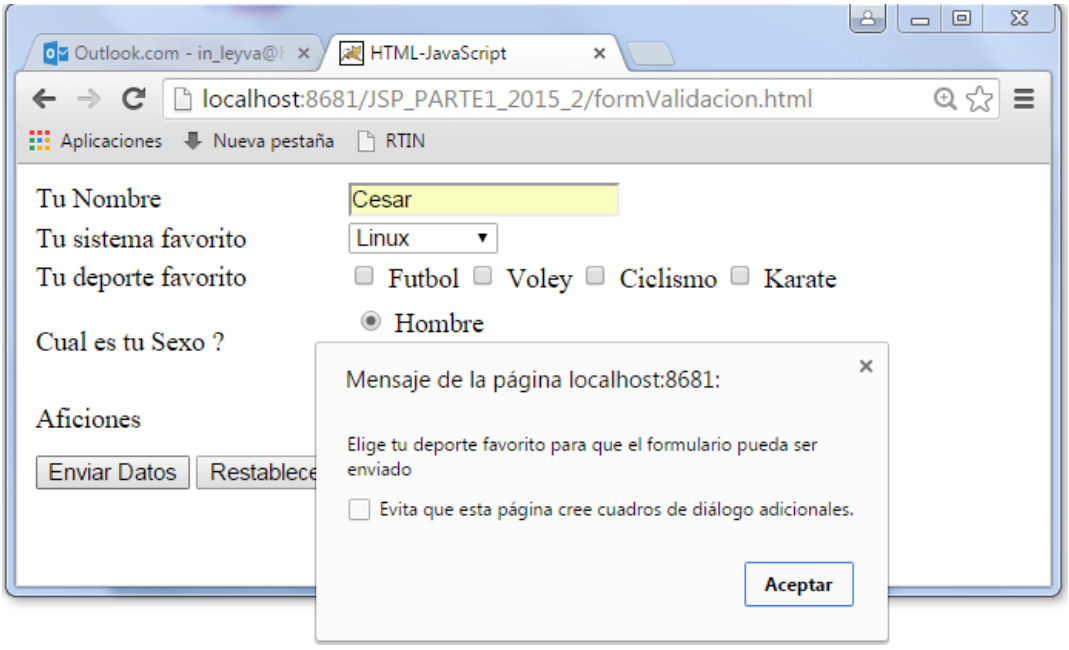
Desarrollo:

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>HTML-JavaScript</title>
        <script>
     //funcion que retona valor si boton ha sido activado o no
            function getValueRadioActivado() {
                //variable de comprobación
                var pulsado = false;
                //array de elementos
                var opciones = document.f1.rdsexo;
                //bucle de comprobación
                for (i = 0; i < opciones.length; i++) {
                    if (opciones[i].checked == true) {
                        pulsado = true;
                    }
                }
                return pulsado;
            }
    //funcion para determinar si un deporte favorito ha sido activado
            function getValueChekActivado() {
                //variable de comprobación
                var pulsado = false;
                //array de elementos
                var opciones = document.f1.chkdep;
                //bucle de comprobación
                for (i = 0; i < opciones.length; i++) {
                    if (opciones[i].checked == true) {
                        pulsado = true;
                    }
                }
                return pulsado;
            }
```

```
        //funcion para validar fomulario
            function validar() {
                var n = document.f1.nom.value;
                var s = document.f1.sis.value;
                var d = document.f1.chkdep.value;
                if (n== "") {
                    alert('Ingrese nombre');
                    return false;
                } else if (s == "") {
                    alert('Seleccione:\nTu sistema favorito');
                    return false;
                }
                else if (!getValueChekActivado()) {
                    alert("Elige tu deporte favorito para que el
formulario pueda ser enviado "+d);
                    return false;
                }
                else if (!getValueRadioActivado()) {
                    alert("Elige una opción de sexo para que el
formulario pueda ser enviado");
                    return false;
                }
                return true;
            }
        </script>
    </head>
    <body>


        <form name="f1" action="" method="post" onSubmit="return
validar();">
            <table>
                <tr>
                    <td>Tu Nombre</td>
                    <td><input type="text" name="nom"> </td>
                </tr>
                <tr>
                    <td>Tu sistema favorito</td>
                    <td>
                        <select name="sis">
                            <option value=""> Seleccione</option>
                            <option value="1"> Linux</option>
                            <option value="2"> Windows</option>
                            <option value="3">Mac</option>
                        </select>
                    </td>
                </tr>
                <tr>
                    <td>Tu deporte favorito</td>
                    <td>
        <input type="checkbox" name="chkdep" value="Futbol"> Futbol
        <input type="checkbox" name="chkdep" value="Futbol"> Voley
        <input type="checkbox" name="chkdep" value="Ciclismo"> Ciclismo
        <input type="checkbox" name="chkdep" value="Karate"> Karate
                    </td>
                </tr>

                <tr>
                    <td>Cual es tu Sexo ?</td>
                    <td>
                        <table>
```

```
                        <tr>
                    <td><input type="radio" name="rdsexo"> Hombre</td>
                        </tr>
                        <tr>
                     <td><input type="radio" name="rdsexo"> Mujer</td>
                        </tr>
                    </table>

                    </td>
                </tr>
                <tr>
                    <td>Aficiones</td>
                    <td>

                        <textarea name="txta"> </textarea>
                    </td>
                </tr>
                <tr>
                    <td>
                        <input type="submit" value="Enviar Datos">
                        <input type="submit" value="Restablecer">
                    </td>
                    <td>

                    </td>
                </tr>

            </table>
        </form>

    </body>
</html>
```

**********EJECUCION DE LA APLICACIÓN**********



Nota

Una de las principales funciones de Javascript y uno de los motivos por los cuales se creó este lenguaje son las validaciones de formulario.

Todo usuario de internet alguna vez ha completado un formulario y al tratar de enviarlo le apareció un mensaje mostrando que había algún campo vacío o incorrecto. Pero gracias a Javascript podremos validar formularios desde el lado del cliente (navegador) evitando que llegue información inútil al servidor.

Sin embargo voy a aclarar algo muy importante, si bien validar el envío de datos antes de ser procesados con Javascript es muy importante, también tener en cuenta que las validaciones en el servidor (por ejemplo con JAVA) nunca deberían faltar, ya que el usuario puede tener Javascript deshabilitado o bien éste puede burlar fácilmente esta seguridad con conocimientos medios de internet.

Validar formularios con Javascript lleva demasiado tiempo creando las funciones para cada regla y muchas veces se torna una tarea tan necesaria como densa. Sin embargo **jQuery(** Libreria de JavaScript) a través de su librería **validate.js** lo hace demasiado fácil, por lo que se recomienda investigar sobre este tema.

Se sugiere también Investigar sobre Validación de formularios con HTML5, permite definir patrones de validación para los diferentes campos de un formulario sin utilizar ni una sola línea de javascript. Esta característica nos permite ahorrarnos algunas solicitudes inadecuadas y el consiguiente gasto innecesario de recursos del servidor, aunque sin duda la principal ventaja es la mejora de la experiencia del usuario.

Es importante anotar que la validación de formularios con HTML5, al igual que la validación con javascript, se realiza en el lado del cliente (navegador web) y, por tanto, puede ser modificada por el usuario o ignorada totalmente. Esto quiere decir que las cuestiones relativas a la seguridad del formulario no deben basarse en esta validación ni es sustituta de la validación server-side.