

	<p style="text-align: center;">UNIVERSIDAD DON BOSCO FACULTAD DE INGENIERÍA ESCUELA DE COMPUTACIÓN</p>
<p style="text-align: center;">Ciclo II</p>	<p style="text-align: center;">Guía de Laboratorio #1</p> <p>Nombre de la práctica: Fundamentos de páginas web dinámicas en el cliente con JavaScript.</p> <p>Lugar de ejecución: Centro de cómputo.</p> <p>Materia: Desarrollo de Aplicaciones Web con Software Interpretado en el Cliente</p>

I. OBJETIVOS

Que el estudiante:

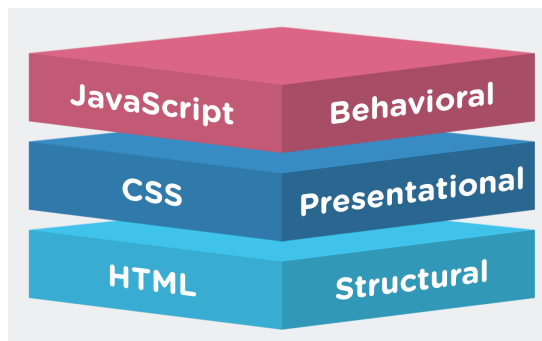
1. Adquiera dominio de los elementos sintácticos básicos del lenguaje JavaScript.
2. Domine de las distintas formas de incluir código JavaScript dentro de un documento web.
3. Sea capaz de generar código HTML 5 desde secuencias de comando de JavaScript.
4. Pueda crear páginas web utilizando métodos básicos de interacción con el usuario.
5. Utilice métodos básicos de depuración de scripts utilizando las herramientas del navegador.

II. INTRODUCCION TEORICA

JavaScript no intrusivo (Unobtrusive JavaScript)

De acuerdo a las buenas prácticas de programación con JavaScript y siguiendo las tendencias de separación en capas manteniendo así la estructura del documento limpia sin código de estilos CSS o sin código de script, JavaScript en este caso, se está poniendo en práctica una forma de introducir JavaScript dentro de las aplicaciones web de forma no invasiva, o como se dice de acuerdo a la traducción directa del inglés, de forma no obstrusiva o no intrusiva (unobtrusive JavaScript).

Básicamente, esta tendencia consiste en evitar incluir el código de JavaScript dentro del marcado HTML. Para hacer esto, se requiere que se adquiera la habilidad de capturar o hacer referencia a los elementos de la página exclusivamente desde el código de JavaScript y una vez hecho esto, agregarle la funcionalidad (el comportamiento) mediante sentencias de programa que estarán dentro del script .js.



Ejemplo:

```
//Usando métodos tradicionales de incluir JavaScript
<a href="#" onclick="openFullImage()">
  
</a>
```

```
//Usando JavaScript no intrusivo
//HTML
<a href="#">
  
</a>
```

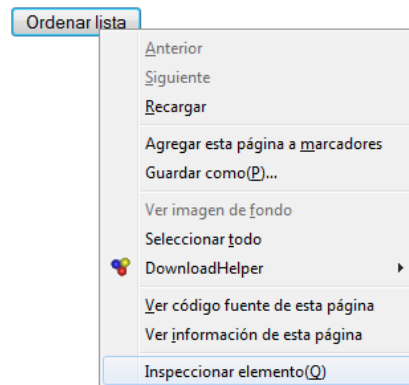
```
//Capturar el elemento <a href="#"></a> con JavaScript y asignar
//la función que será el manejador de eventos al producirse
//el evento click sobre el enlace
document.links[0].onclick = openFullImage;
```

```
function openFullImage(){
  //Código que mostrará la imagen en mayor resolución
}
```

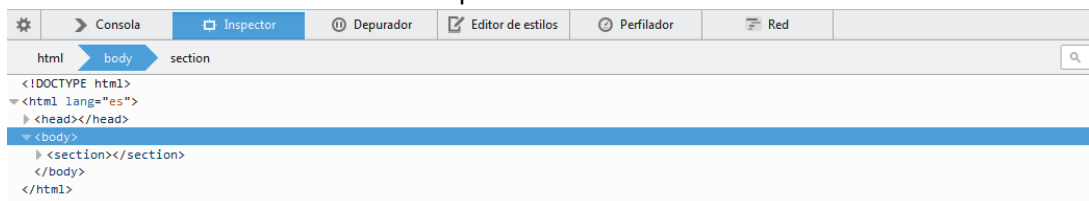
Depuración de errores con el navegador

Frecuentemente nos encontraremos con situaciones en donde será necesario corregir los scripts realizados con JavaScript. Es una tarea común en programación que la página web que contiene código JavaScript no se comporte de la manera esperada o no produzca el resultado que de acuerdo a nuestra lógica es el correcto. Generalmente, esto sucede cuando se produce algún error en la etapa de interpretación ejecutada por el navegador, ya sea debido a algún error de sintaxis o a un error en la lógica de programación. Casi todos los navegadores brindan algún mecanismo de depuración. A continuación se muestran algunos:

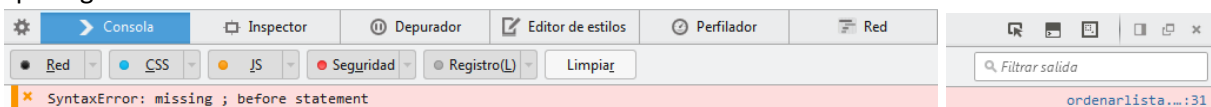
Si utiliza el **Mozilla Firefox 27.0** o superior la depuración resulta más fácil de realizar utilizando el **codemirror** que es la herramienta para desarrolladores (**DevTools**) que ofrece por defecto el navegador de Mozilla. Para activarla, se sigue un proceso similar, primero haciendo clic derecho sobre un elemento que no esté funcionando adecuadamente o que no esté produciendo el resultado esperado y a continuación en el menú desplegable se da clic sobre la opción **Inspeccionar elemento (O)**.



Esa acción abrirá la consola de las herramientas para desarrollador del Firefox:

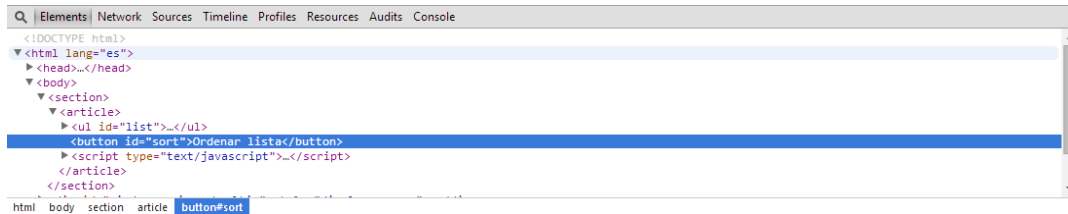


Haciendo clic en la opción Consola, podrá observar alguna notificación de error en el código JavaScript. Si aparece en blanco actualice y de clic en el elemento asociado a la ejecución de código de JavaScript para que le notifique algún error.

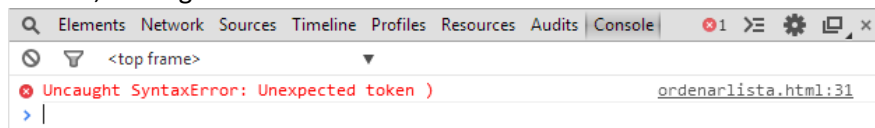


Puede observar que el depurador le muestra el error, su descripción y el número de línea donde está el problema encontrado.

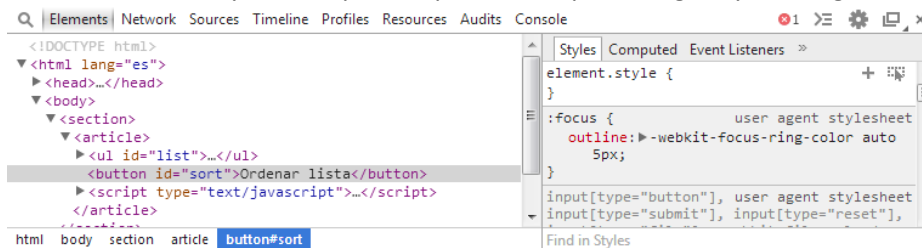
Si está utilizando **Chrome de Google**, puede invocar a las herramientas de desarrollador de Chrome (o **Chrome Developer Tools** o simplemente **DevTools**) para depurar código de JavaScript. El funcionamiento es idéntico y al invocarlo visualizará la siguiente interfaz de usuario donde puede explorar errores de JavaScript siempre en la opción Consola.



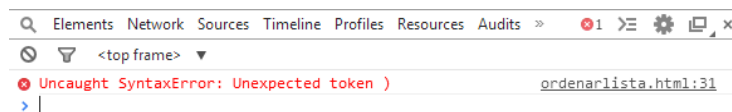
Al hacer clic en la opción Consola (Console) podrá visualizar los errores de JavaScript y las líneas donde se encuentran dichos errores, con alguna información adicional acerca del error.



En el caso de que utilice el navegador **Opera**, puede utilizar el Opera Dragonfly, la propuesta de este navegador para depurar errores de HTML, CSS y JavaScript. El aspecto del Opera Dragonfly es el siguiente:



Igual que con las propuestas de Firefox, Chrome e Internet Explorer se accede a través de los elementos de la página, haciendo clic derecho sobre ellos o invocando mediante método abreviado del teclado, pulsando las teclas CTRL+SHIFT+I. Para observar los errores posibles en JavaScript, debe hacer clic sobre la opción Consola (Console).



Hay que mencionar que en las últimas versiones de Opera, el aspecto del Dragonfly es casi idéntico al de las herramientas para desarrollador de Chrome. Esto es así, debido a que para la última versión de Opera se ha utilizado como núcleo el motor Chromiun, mismo que utiliza el navegador Chrome de Google.

Se plantea que investigue ud. cuál es la herramienta para desarrolladores del navegador Safari de las computadoras MAC.

Interacción básica con el usuario utilizando JavaScript

La forma más básica de interactuar con el usuario desde JavaScript es haciendo uso de los diálogos de entrada y salida simple proporcionados por los navegadores. Existen tres diálogos para entrada y salida simple. Estos son:

- Alerta (alert()),
- Confirmación (confirm()), y

c) Solicitud de información (prompt()).

Veamos cada uno de ellos.

Diálogo de alerta (*alert*)

El cuadro de diálogo de alerta se invoca mediante el método ***alert()*** del objeto window, que conoceremos en futuras prácticas. Este diálogo es simple y muestra un mensaje breve utilizando una cadena de texto, un icono de advertencia y un único botón Aceptar como el que se muestra a continuación.

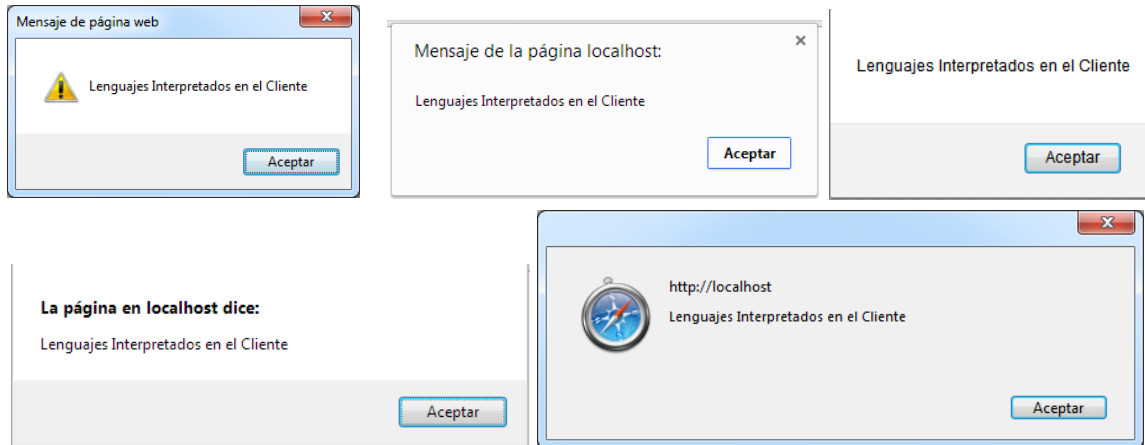


Figura 1. Ventanas de diálogo de alerta de JavaScript (*alert*) en los navegadores de uso más frecuente: Internet Explorer, Firefox, Chrome, Opera y Safari.

La sintaxis básica para un diálogo de alerta es:

```
window.alert(mensaje);
```

De forma abreviada se puede utilizar así:

```
alert(mensaje);
```

Donde mensaje es una cadena de texto, por lo tanto, si se introduce directamente el mensaje como un literal de cadena, debe ingresar el texto entre comillas simples o dobles. Si se utilizan variables declaradas deben ser de tipo cadena. Además, la cadena puede ser el resultado de la evaluación de una expresión, como por ejemplo, una concatenación. Cuando pasa un tipo de dato que no es cadena JavaScript lo convertirá en cadena.

Veamos algunos ejemplos:

```
alert("¡Hola desde JavaScript!");
var usuario="Ricardo Ernesto";
alert("Hola " + usuario + ". Bienvenido a JavaScript");
```

Diálogo de confirmación (*confirm*)

Un diálogo de confirmación utiliza el método ***confirm()*** del objeto Window. Este diálogo muestra una pequeña ventana con un mensaje para que el usuario responda si está de acuerdo, pulsando un botón Aceptar, o si no está de acuerdo, pulsando un botón Cancelar (OK y Cancel, en un navegador en inglés). El diálogo también muestra un icono de interrogación. El diálogo tiene un aspecto como el que se muestra a continuación:

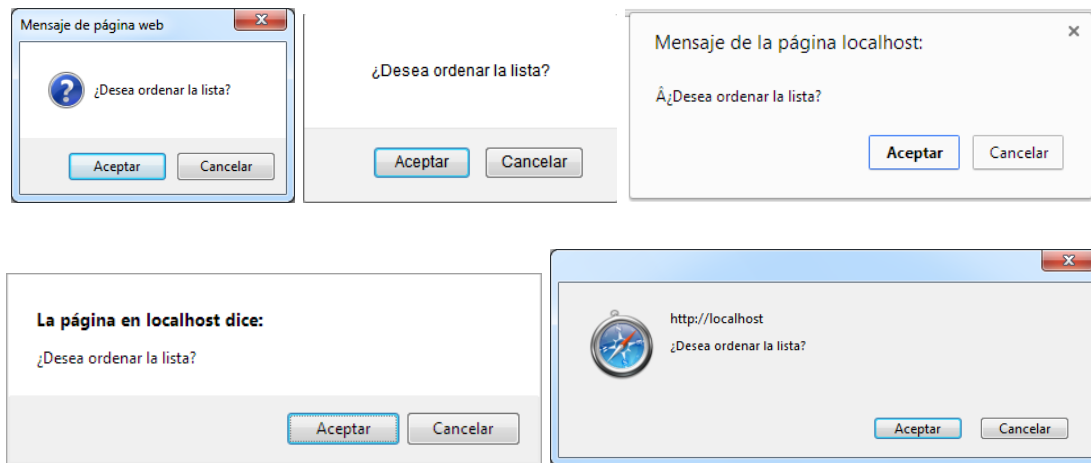


Figura 2. Ventanas de diálogo de confirmación (confirm) de los navegadores de uso más frecuente

La sintaxis básica de un diálogo de confirmación es:
`window.confirm(mensaje);`

O en forma resumida
`confirm(mensaje);`

Diálogo de solicitud de información (*prompt*)

Este diálogo utiliza el método ***prompt()*** del objeto Window. Esta ventana pequeña contiene un cuadro de texto donde el usuario debe escribir la información que se le solicita con un mensaje y un par de botones, uno para ingresar la información escrita y otro para cancelar la operación. Es importante mencionar que el dato ingresado por el usuario en este cuadro de texto, siempre será procesado en el script con un dato de tipo cadena de texto, sin importar, que se hayan introducido valores numéricos. El aspecto es el que se muestra a continuación:

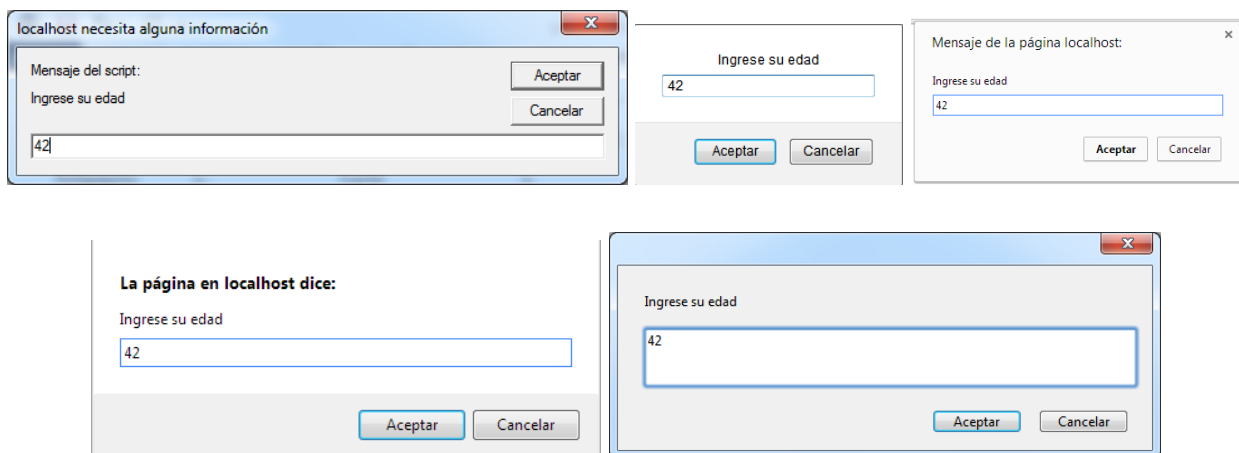


FIGURA 3. VENTANAS DE DIÁLOGO DE SOLICITUD DE INFORMACIÓN DE JAVASCRIPT (PROMPT) EN MICROSOFT INTERNET EXPLORER 6.0 Y NETSCAPE 7.0, RESPECTIVAMENTE.

El método ***prompt()*** tiene dos argumentos. El primero es una cadena de texto que se muestra como etiqueta para indicar al usuario el tipo de información solicitada y el segundo es un valor predeterminado que aparece en el cuadro de texto donde se escribe la información solicitada. El método **devuelve un valor de cadena** que contiene el valor que introduce el usuario en la solicitud.

La sintaxis básica del diálogo solicitud de información es:

```
var result = prompt(mensaje, valor);
```

Nótese que para poder utilizar el valor devuelto por el método **prompt()** debe utilizarse una variable a la cual asignárselo. En la sintaxis anterior **mensaje** es la cadena de texto que aparece como indicación del tipo de información que se está solicitando y **valor** es una cadena con un valor predeterminado que aparecerá en el cuadro de texto del diálogo solicitud de información. Si no coloca un valor predeterminado en la llamada aparecerá por defecto, en algunos navegadores el valor **undefined**. Si no desea esto coloque como valor una cadena vacía. Como se muestra a continuación:

```
var edad = prompt("¿Cuál es tu color favorito? ", "");
```

Imprimir en la página web desde JavaScript

Con JavaScript se puede mandar a escribir texto, datos y código HTML de forma dinámica en una página web. Para poder hacer uso de esta característica debe hacer uso del método write del objeto document de JavaScript.

La sintaxis a utilizar para aplicar este método es la siguiente:

```
document.write(cadena o datos a imprimir);
```

Alternativamente, se puede utilizar también:

```
document.writeln(cadena o datos a imprimir);
```

Ejemplo:

```
document.write("Esta página fue modificada por última vez: " + document.lastModified);
```

OBSERVACIÓN: Aunque estos han sido los métodos tradicionales en los comienzos de JavaScript para generar contenido dinámico, con el paso del tiempo y las mejoras en los navegadores el contenido dinámico se genera de otras maneras utilizando principalmente el método innerHTML del objeto document. En este punto hay que aclarar que innerHTML fue un invento de Microsoft para su navegador Internet Explorer, que se hizo muy popular de tal manera que hoy en día, todos los navegadores le dan soporte. Sin embargo, de acuerdo a la W3C no forma parte de ningún estándar, de modo que su alternativa es la manipulación del documento mediante el DOM (Document Object Model), que estaremos desarrollando más adelante en este curso.

III. MATERIALES Y EQUIPO

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

No.	Requerimiento	Cantidad
1	Guía de práctica #1: Fundamentos de páginas web en el cliente con JavaScript	1
2	Computadora con navegadores actualizados y editor de texto instalado	1
3	Memoria USB	1
4	Editor con navegadores actualizados	

IV. PROCEDIMIENTO

Indicaciones: Crear una carpeta con el nombre DAW y dentro de ella crear otra carpeta a la que nombrará guia1 y dentro de esta todas las carpetas de los ejercicios que se le muestran a continuación:

- ejemplo1
- ejemplo2
- ejemplo3

Ejemplo 1: Cálculo de la prima y la cuota a pagar por una casa, solicitando al usuario su ingreso salarial por mes.

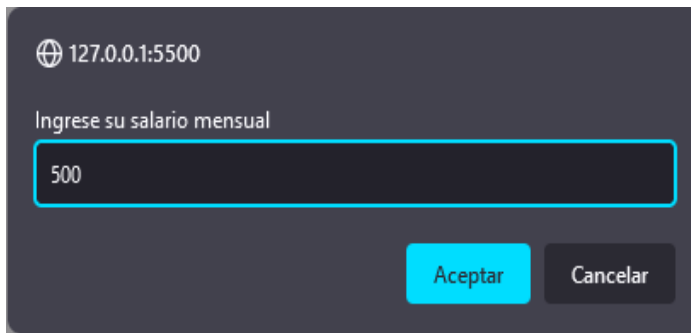
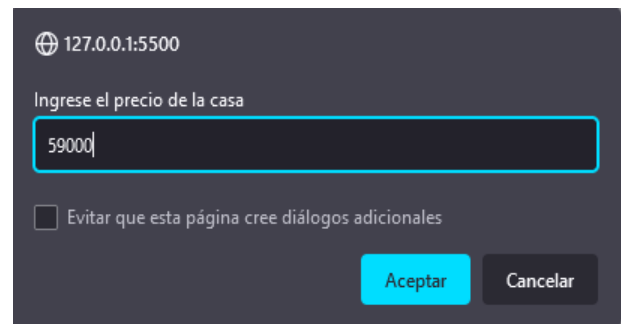
Guión 1: cuotacasa.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <title>Cuota prima de casa</title>
  <link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Oswald:400,700" />
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css"
  integrity="sha384-Vkoo8x4CGsO3+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
  crossorigin="anonymous">
</head>
<body class="container">
<script>
  var salario, preciocasa, prima, cuotamensual;
  var annios;
  var h1, totalh1;
  salario = parseFloat(prompt('Ingrese su salario mensual, '));
  annios = 12*15; //numero de años a pagar por la casa
  preciocasa = parseFloat(prompt('Ingrese el precio de la casa, '));
  if(salario <= 400){
    if(preciocasa > 8000){
      alert('Lo siento el valor de la casa es muy alto para su sueldo');
    }
    else{
      prima = (preciocasa*0.1);
      cuotamensual = (preciocasa-prima)/annios;
      document.write("<h1>Calculo de Prima de Casa</h1>");
      document.write("<img src='img/house.png'>");
      document.write("<h1>Calculo de Prima de Casa</h1>");
      document.write("<h1 class='alert alert-primary'>La prima que usted debe pagar por la casa es = $ " +
Math.round(prima * Math.pow(10,2)) / Math.pow(10,2) + "</h1><hr>");
      document.write("<h1 class='alert alert-primary'>La cuota mensual a pagar durante 30 años es de = $ " +
Math.round(cuotamensual * Math.pow(10,2)) / Math.pow(10,2) + "</h1><hr>");
    }
  }
  else{
    prima = (preciocasa*0.2);
    cuotamensual = (preciocasa-prima)/annios;
    document.write("<h1>Calculo de Prima de Casa</h1>");
    document.write("<img src='img/house.png'>");
    document.write("<h1 class='alert alert-primary'>La prima que usted debe pagar por la casa es = $ " +
Math.round(prima * Math.pow(10,2)) / Math.pow(10,2) + "</h1><hr>");
    document.write("<h1 class='alert alert-primary'>La cuota mensual a pagar durante 15 años es de = $ " +
Math.round(cuotamensual * Math.pow(10,2)) / Math.pow(10,2) + "</h1><hr>");
  }
  //Buscar todos los elemento h1 dentro del documento
  h1 = document.getElementsByTagName('h1');
  //Obtener el total de los elementos h1 encontrados
  totalh1 = h1.length;
  //Crear un ciclo o lazo para asignar el manejador
```

```
//de eventos onmouseover y onmouseout
for(var i=0; i<totalh1; i++){
  h1[i].onmouseover = function(){
    this.className = 'alert alert-info';
  }
  h1[i].onmouseout = function(){
    this.className = 'alert alert-primary';
  }
}
</script>
</body>
</html>
```

- Crear una carpeta con el nombre **img** y dentro de esta colocar las imagen brinda en los recursos de la guía.

Resultado:

Calculo de Prima de Casa



La prima que usted debe pagar por la casa es = \$ 11800

La cuota mensual a pagar durante 15 años es de = \$ 262.22

Ejemplo 2: Realización de un calculador de áreas geométricas, en base a un tipo de área seleccionada y a las medidas ingresadas de acuerdo con el área ingresada. Las áreas a considerar serán: círculo, triángulo y rectángulo.

Guión 1: calculoareas.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8"/>
  <title>Cálculo de áreas</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css"
  integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
</head>
<body>
<section id="menu">
<h1 class="text-center">Áreas</h1>
<hr>
<nav id="menu-options">
  <div class="container" style="text-align: center;">
    <div class="row">
      <div class="col-sm">
        <a href="circulo.html" title="Calcular área del círculo">
          
          <h6>Área de Círculo</h6>
        </a>
      </div>
      <div class="col-sm">
        <a href="#">
          
          <h6>Área de Triángulo</h6>
        </a>
      </div>
      <div class="col-sm">
        <a href="#">
          
          <h6>Área de rectangulo</h6>
        </a>
      </div>
    </div>
  </div>
</nav>
</section>
</body>
</html>
```

Guión 2: circulo.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <title>Área círculo</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css"
  integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
  <script src="js/circulo.js"></script>
</head>
<body class="container">
<section id="figura">
```

```
<div class="circulo"></div>
</section>
<nav>
  <a href="calculoareas.html" class="badge badge-primary"><h4>Regresar</h4></a>
</nav>
</body>
</html>
```

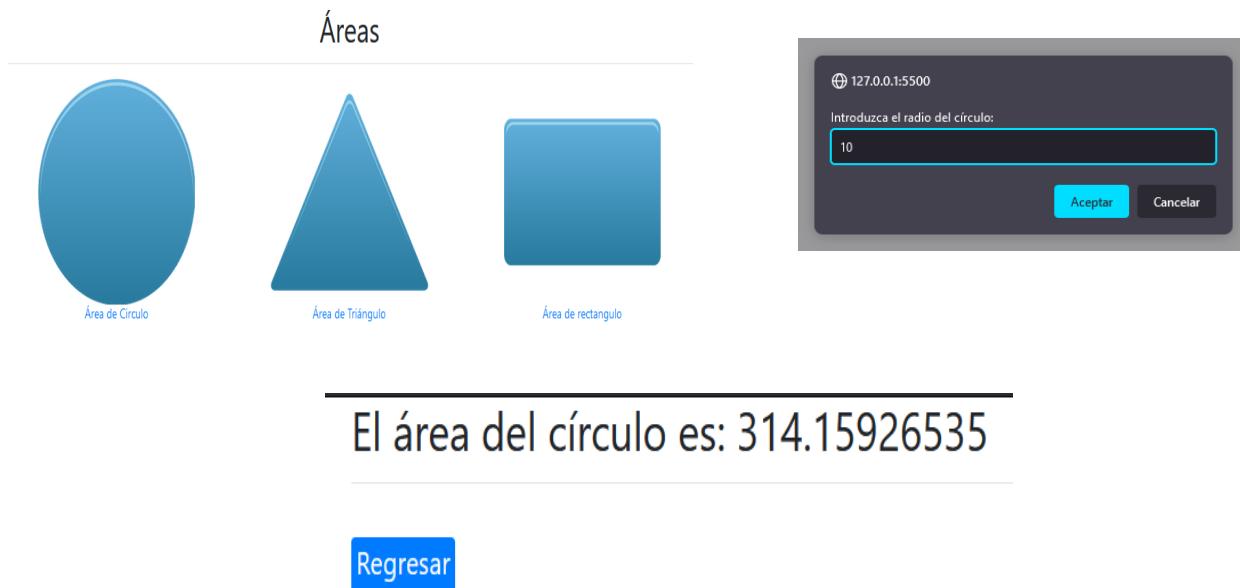
Crear carpeta js y dentro de ella cree y coloque el siguiente script.

Guión 3: circulo.js

```
const PI = 3.1415926535;
var radio = prompt('Introduzca el radio del círculo:');
var area;
area = PI*radio*radio;
document.write("<header><h1>El área del círculo es: " + area + "</h1><hr /><br /></header>");
```

- Crear una carpeta con el nombre **img** y dentro de esta colocar las imagen brinda en los recursos de la guía.
- En base a los ejemplos **circulo.html** y **circulo.js**, creados anteriormente realizar los archivos:
 - triangulo.html y triangulo.js $(base*altura)/2$
 - rectángulo.html y rectángulo.js $(base*altura)$

Resultado:



Ejemplo 3: Cálculo del número de días para el mes y año seleccionado.

Guión 1: diasmes.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <title>Días de cada mes</title>
  <link rel="stylesheet" async href="http://fonts.googleapis.com/css?family=Anton" data-generated="http://enjoycss.com" />
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css"
  integrity="sha384-Vkoo8x4CGsO3+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
</head>
<body class="container">
<div id="info"></div>
</body>
<script src="js/calculodias.js"></script>
</html>
```

- Crear una carpeta con el nombre **img** y dentro de esta colocar las imagen brinda en los recursos de la guía.
- Crear carpeta js y dentro de ella cree y coloque el siguiente script.

Guión 2: calculodias.js

```
var mes = prompt("Ingrese el mes: ", "");
var year = prompt("Ingrese el año: ", "");
var diasdelmes;
var bisiesto = false;
if(year%4 == 0 && (year%100 != 0 || year%400 == 0)){
  bisiesto = true;
}
switch(mes){
  case "Enero": case "enero": case "Marzo":
  case "marzo": case "Mayo": case "mayo":
  case "Julio": case "julio": case "Agosto":
  case "agosto": case "Octubre": case "octubre":
  case "Diciembre": case "diciembre":
    diasdelmes = 31;
    break;
  case "Abril": case "abril": case "Junio":
  case "junio": case "Septiembre": case "septiembre":
  case "Noviembre": case "noviembre":
    diasdelmes = 30;
    break;
  case "Febrero": case "febrero":
    if(!bisiesto)
      diasdelmes = 28;
    else
      diasdelmes = 29;
    break;
  default:
    alert("El mes " + mes + " no existe");
    mes = "No existe";
    diasdelmes = 0;
}
if(bisiesto == true){
  var msg = "El año " + year + " es bisiesto.<br />El mes de " + mes + " tiene " + diasdelmes + " días.";
}
else{
  var msg = "El año " + year + " no es bisiesto.<br />El mes de " + mes + " tiene " + diasdelmes + " días.";
}
//Variable para hacer referencia al elemento DIV
//donde se mostrará el número de años, meses y días
```

```
//que ha vivido la persona
var dias = document.getElementById('info');
dias.innerHTML = '<h3>' + mes + ' ' + year + '</h3>';
dias.innerHTML += '<div class="card" style="width: 18rem;">';
dias.innerHTML += '';
dias.innerHTML += '<div class="card-body">';
dias.innerHTML += ' <h5 class="card-title">' + diasdelmes + ' dias</h5>';
dias.innerHTML += ' <p class="card-text">' + msg + '</p>';
dias.innerHTML += '</div>';
dias.innerHTML += '</div>';
```

Resultado:

file://
Ingrese el año:
2020
☐ Evitar que esta página cree diálogos adicionales
Aceptar Cancelar

file://
Ingrese el mes:
Febrero
Aceptar Cancelar

Febrero 2020

29 días
El año 2020 es bisiesto.
El mes de Febrero tiene 29 días.

V. DISCUSION DE RESULTADOS

1. Crear una aplicación web realizar cálculo de presupuesto en un hospital. Dado un valor de presupuesto por el usuario, el presupuesto se calcula el total asignado a tres áreas del hospital: Dermatología 40%, Traumatología 30% y Pediatría 25%.
2. Similar al ejemplo 2 del procedimiento de esta guía de práctica, cree un sitio web con las páginas que sean necesarias para realizar conversiones de una cantidad dada en metros a pies, pulgadas y yardas (investigue los factores de conversión). Debe realizar una página principal donde seleccionar en un menú a qué unidades desea convertir la cantidad en metros que ingresará y luego, mediante mensaje prompt() deberá solicitar esa cantidad y luego mostrar en la página web de la unidad de conversión seleccionada la cantidad ingresada en metros y la cantidad convertida.
3. Crear un programa donde el usuario ingrese en un prompt la temperatura en Celcius y el programa muestre en la página la temperatura en Fahrenheit.

VI. INVESTIGACION COMPLEMENTARIA

1. Investigue a qué se le llama JavaScript no invasivo, no obstructivo o no intrusivo (unobtrusive JavaScript). Muestre un ejemplo funcionando correctamente en todos los navegadores que utilizamos en este curso (Internet Explorer 11, Firefox, Chrome, Opera y Safari).
2. Investigue los tipos de datos null y undefined en JavaScript indicando en qué se diferencian, para qué los podemos utilizar y qué cuidados deberíamos tener al acceder a datos con estos valores.

VII. BIBLIOGRAFIA

Flanagan, David. JavaScript La Guía Definitiva. 1ra Edición. Editorial ANAYA Multimedia. 2007. Madrid, España.

Powell, Thomas / Schneider, Fritz. JavaScript Manual de Referencia. 1ra Edición. Editorial McGraw-Hill. 2002. Madrid, España.

McFedries, Paul. JavaScript Edición Especial. 1ra Edición. Editorial Prentice Hall. 2002. Madrid, España.

Tom Negrino / Dori Smith. JavaScript & AJAX Para Diseño Web. 6ª Edición. Editorial Pearson – Prentice Hall. 2007. Madrid España.