	<b>UNIVERSIDAD DON BOSCO</b> <b>FACULTAD DE INGENIERÍA</b> <b>ESCUELA DE COMPUTACIÓN</b>	
<b>CICLO: 02/2020</b>	<b>GUIA DE LABORATORIO #8</b> <b>Nombre de la Práctica:</b> Objetos en JavaScript <b>Lugar de Ejecución:</b> Centro de cómputo <b>Tiempo Estimado:</b> 2 horas <b>MATERIA:</b> Lenguajes Interpretados en el Cliente	

## I. OBJETIVOS

Que el estudiante:

- Tenga claras todas las características de la Programación Orientada a Objetos con JavaScript.
- Utilice los diversos tipos de objetos incorporados con que cuenta el lenguaje JavaScript.
- Se familiariza con los objetos del navegador más utilizados en los scripts.

## II. INTRODUCCION TEORICA

### Programación Orientada a Objetos en JavaScript

La Programación Orientada a Objetos está basada en el concepto de dos entidades distintas, que son: las clases y las instancias.

Una clase define las propiedades y métodos, a veces denominados simplemente miembros, que caracterizan a un cierto grupo de objetos. Una clase es una abstracción de ciertos objetos que existen en la realidad e intenta ser un modelo o plano de los objetos que representa, definiendo únicamente las características y el comportamiento que debe tener dicho objeto.

La instancia es un objeto concreto creado a partir de las propiedades y métodos definidos en la clase, a veces se prefiere decir que la instancia es un ejemplar concreto del tipo de objeto definido en la clase a partir de la cual se ha creado el objeto. Por ejemplo, Victoria Escobar es una instancia o un ejemplar concreto de la clase empleado, representando a una persona en particular que es o tiene las características y comportamiento de un empleado.

Los lenguajes basados en clases, como Java, C++ o C#, exigen que se deba definir primero una clase y a continuación realizar instancias de esta clase para crear un objeto en particular. JavaScript, es más bien un lenguaje basado en prototipos que no requiere hacer esta distinción, simplemente es un lenguaje que tiene objetos predefinidos, a los que se conoce como prototipos y que son, en la práctica, plantillas a partir de las cuales se obtienen propiedades y métodos para los nuevos objetos que se crean. Sin embargo, cada nuevo objeto puede especificar sus propias propiedades y métodos, tanto al ser creado o en tiempo de ejecución.

### Diferencias entre lenguajes basados en clase en comparación con los basados en prototipos

La siguiente tabla resume las diferencias entre los lenguajes basados en clases, como Java o C#, y los basados en prototipos como JavaScript.

Basado en Clases (Java)	Basado en Prototipos (JavaScript)
Clase e instancia son entidades distintas.	Todos los objetos son instancias.

Define una clase con la definición de clase; instancia una clase con los métodos constructores.	Define y crea un conjunto de objetos con las funciones constructoras.
Crea un único objeto con el operador new.	Igualmente.
Construye una jerarquía de objetos utilizando las definiciones de clases par definir subclases de una clase existente.	Construye una jerarquía de objetos mediante la asignación de un objeto como prototipo asociado con la función constructora.
Hereda las propiedades mediante el seguimiento de la cadena de clases.	Hereda las propiedades mediante el seguimiento de la cadena de prototipos.
La definición de Clases especifica todas las propiedades de todas las instancias de una clase. No puede añadir propiedades dinámicamente en tiempo de ejecución.	La función constructora o prototipo puede especificar un conjunto inicial de propiedades. Puede añadir o remover dinámicamente a objetos individuales o a un conjunto entero de objetos.

## Definición de objeto

Los objetos en JavaScript son tipos de datos compuestos, tal y como las matrices o arreglos, que dentro del lenguaje JavaScript son, más bien, objetos. Se dice que son compuestos porque combinan estado (propiedades que almacenan datos) y comportamiento (procedimientos o métodos para operar los datos).

Un objeto viene siendo una representación concreta, particular y real de algo. Esta representación determina su identidad, su estado y su comportamiento.

## Clasificación de objetos en JavaScript

Los objetos en JavaScript se pueden subdividir en cuatro grupos:

- **Objetos definidos por el usuario.** Son objetos completamente personalizables que puede crear el desarrollador para dar estructura y coherencia a una tarea de programación en particular. Con estos objetos es posible definir propiedades y métodos mediante una sintaxis específica de JavaScript. Las propiedades identifican las características propias del objeto, mientras que los métodos permiten realizar tareas u operaciones con dichas propiedades.
- **Objetos incorporados del lenguaje.** Son objetos proporcionados por el propio lenguaje JavaScript, entre los que se incluyen: (1) objetos asociados a tipos de datos primitivos, como los objetos **String**, **Number** y **Boolean**, (2) objetos que permiten la creación de objetos definidos por el usuario, como los objetos **Object** y **Form**, (3) objetos utilizados para simplificar tareas frecuentes, como los objetos **Date**, **Math** y **RegExp**.
- **Objetos de navegador.** Son objetos que no están especificados como parte del lenguaje JavaScript, pero que son soportados por la mayor parte de navegadores que cumplen con los estándares. Los dos objetos más utilizados que pertenecen a esta categoría son los objetos **window** y **Navigator**. El primero permite el control de la ventana del navegador y, el segundo, proporciona información acerca de la configuración del navegador del usuario.
- **Objetos de documento.** Son los objetos que forman parte del Modelo de Objetos de Documento (Document Object Model, abreviado como DOM) definido por la W3C. Es a través de estos objetos que JavaScript puede manejar Hojas de Estilo en Cascada y la ejecución de HTML dinámico. El acceso a los objetos de documento es proporcionado por el navegador mediante el uso del objeto **document** del objeto **window**.

## Creación de objetos en JavaScript

Existen dos formas de crear objetos en JavaScript:

1. Utilizando la palabra clave **new** y el nombre del **constructor del objeto** a crear. Se pueden crear objetos especializados haciendo uso del operador **new** y a continuación invocar al **método constructor** del objeto que inicializa las propiedades del objeto.

2.

Ejemplo:

```
var carro = new Object();
carro.marca = "Toyota";
carro.modelo = "Corolla";
```

```

carro.npasajeros = 5;
carro.annio = 2007;
carro.placa = "503-618";
carro.nchasis = "JDSXV39Z200750226";
carro.encender = function(numero){
    carro.nchasis == numero ? start = "on" : start = "off";
    return start;
}
carro.apagar = function(){
    start = off;
}

```

3. Utilizando sintaxis de literales de objeto. Esta es la forma más fácil de crear objetos en JavaScript, que consiste de una lista de pares nombre\_propiedad : valor separados por comas y encerrados entre llaves.

Ejemplo:

```

var coord = {x : 0, y : 0};
var circle = {x : coord.x, y : coord.y, r : 2.5};

```

Esta forma de construir objetos en JavaScript es generalizable a todos los tipos de datos del lenguaje. Veamos esto demostrado con dos tipos de objetos como **String** y **Array**.

Utilizando la palabra clave new y el respectivo método constructor:

```

var nombre = new String();
var paises = new Array("El Salvador", "Guatemala", "Honduras");

```

Utilizando sintaxis declarativa conocida como sintaxis de literales de tipos de datos:

```

var nombre = "";
var paises = ["El Salvador", "Guatemala", "Honduras"];

```

## Objetos del lenguaje

Estos son el conjunto de objetos que vienen incorporados por el lenguaje. Entre los principales tenemos los objetos Array, Function, Date, String, Math, Boolean, Number.

### 1. Objeto Array

Este objeto se abordó en guías pasadas, por lo que trataremos únicamente su declaración, propiedades y métodos. La forma de declarar un objeto Array es haciendo uso del constructor Array(). Si se pasan argumentos al constructor, se interpreta que estos serán los elementos del arreglo. La excepción se da cuando se le pasa al constructor un solo valor numérico, en cuyo caso se entiende que se le proporciona al arreglo un tamaño, que podremos acceder mediante la propiedad length.

Los siguientes son ejemplos de declaración de objetos Array:

```

//Crea un arreglo vacío (sin elementos definidos) y crea una referencia
//a él en la variable arreglo1
var arreglo1 = new Array();
//Crea un arreglo con tres elementos
var arreglo2 = new Array("rojo", "verde", "azul");
//Crea un arreglo cuya propiedad length tiene el valor de 5
var arreglo3 = new Array(5);

```

## Propiedades del objeto Array

### *length*

Como su nombre indica esta propiedad nos devuelve la longitud del array, es decir, el número de elementos que puede almacenar. Su uso es muy simple:

```
var lista = new Array(50);
tamagno = lista.length; /*tamagno almacenaría el valor 50 */
```

### *prototype*

Esta es una propiedad muy potente en el sentido que nos permite agregar al objeto Array las propiedades y métodos que queramos.

```
Array.prototype.descriptor = null;
días = new Array ('lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes');
días.descriptor = "Días laborables de la semana";
```

En este ejemplo hemos creado una nueva propiedad para el objeto **Array**, la propiedad descriptor que podría utilizarse para darle un título a la matriz.

## Métodos del objeto Array

Para ver estos métodos consulte la guía sobre arreglos realizada recientemente. Entre dichos métodos se encuentran: concat(), join(), reverse(), sort(), pop(), push(), etc.

## 2. Objeto Function

Las funciones también son consideradas como objetos por JavaScript. Aunque normalmente no estamos acostumbrados a definir las de esta forma, JavaScript proporciona una forma de definir funciones en donde explícitamente se declaran como objetos. Para ello se utiliza el operador new y el constructor Function, tal y como se muestra a continuación:

```
sumar = new Function("a", "b", "return a + b");
```

La llamada a la función se realiza de la misma forma que conocemos. Puede ver el siguiente ejemplo de llamada:

```
document.write(sumar(90, 100));
```

## Métodos de Function

Los métodos de Function son todos los heredados del objeto Object, que veremos más adelante.

## Propiedades del objeto Function

Básicamente se tienen tres propiedades que se pueden usar con el objeto Function y son:

### *arguments*

Se trata de un array que contiene los argumentos pasados a la función. Esta propiedad permite el uso de funciones con un número variable de argumentos.

### *caller*

Contiene una referencia a la función que llamó a la actual.

### *constructor*

Heredada de la clase Object.

## 4. Objeto String

El objeto **String** se usa para manipular cadenas de caracteres. En JavaScript todo texto encerrado entre comillas, dobles o simples, se interpreta como una cadena, así '45' no es el número cuarenta y cinco sino la cadena formada por los caracteres 4 y 5. Este objeto contiene una gran cantidad de métodos para el manejo y el análisis de cadenas, la extracción de subcadenas e incluso para la conversión de cadenas a texto de marcado HTML.

Como en los anteriores objetos se utiliza el operador new y el constructor String(), como se muestra a continuación:

```
var cadena = new String();
```

También se puede pasar un argumentos al constructor, que será su valor inicial, como se muestra en este otro ejemplo:

```
var cadena = new String("No me importa.");
```

## Propiedades del objeto String

### *length*

Valor numérico que nos indica la longitud en caracteres de la cadena dada.

### *prototype*

Nos permite asignar nuevas propiedades al objeto String.

## Métodos del objeto String

### *anchor(name)*

Crea un enlace (o ancla) a partir de un objeto String con el atributo name igual a la cadena que se le pasa como argumento al método.

Vea el siguiente ejemplo:

```
var refer "referencia1";  
var ancla1 = refer.anchor("anclaje1");
```

Después de ejecutadas las instrucciones anteriores el valor de ancla será:

```
<a name="anclaje1">referencia1</a>
```

La sintaxis de este método permite usar una constante String en lugar del nombre de un objeto String. El ejemplo anterior podría haber escrito como:

```
var ancla = "referencia1".anchor("anclaje1");
```

### *charAt(position)*

Este método aplicado a una cadena devuelve el carácter que se encuentra en la posición dada por el atributo *position* teniendo en cuenta que el índice del primer carácter a la izquierda de la cadena es 0 y el último es una unidad menor que longitud de la cadena. Si el valor del atributo no es válido (igual o mayor que la longitud de la cadena o negativo) el método devuelve el valor *undefined*. Por ejemplo, el siguiente código devuelve el carácter en la tercera posición de la cadena nombre:

```
var nombre = "abcdefghij";  
var car3 =  
nombre.charAt(2);
```

Devolverá "c", que es el tercer carácter por la izquierda (índice igual a 2).

## 5. Manejo de fechas

El objeto Date contiene un valor que representa fecha y hora de un instante dado. Para crear una instancia de este objeto usamos alguna de las siguientes sintaxis:

```
var fecha = new Date()
var fecha = new Date(número)
var fecha = new Date(cadena)
var fecha = new Date(año, mes, día[, hora[, minutos[, seg[, ms]]]])
```

Los argumentos encerrados entre corchetes son opcionales. En la primera forma la variable fecha contendrá la fecha del día actual. La segunda opción almacena en fecha la fecha dada por el argumento como el número de milisegundos transcurridos desde la media noche del 1 de Enero de 1970. El tercer tipo se usa cuando la fecha se pasa en forma de cadena. Por último la fecha puede crearse pasándole como argumento los números de año, mes, día, hora y opcionalmente, hora, minuto, segundo y milisegundo. Los años posteriores a 1970 puede escribirse con dos dígitos, pero es aconsejable usar siempre cuatro dígitos por aquello de los efectos 2000.

```
var hoy = new Date() /*fecha del día en hoy */
var evento = new Date("November 10 1990");
var otro = new Date("10 Nov 1990");
var otro = new Date("10/02/2000"); //Oct, 2, 2000
var instante = new Date(1990, 11, 10, 20,00);
```

Estas son las posibles formas de declarar objetos de tipo fecha. Las dos últimas almacenan el mismo día, pero en la última además se guarda la hora.

Donde se usen cadenas para indicar una fecha podemos añadir al final las siglas GMT (o UTC) para indicar que la hora se refiere a hora del meridiano Greenwich, si no se toma como hora local, o sea, según la zona horaria configurada en el ordenador donde se ejecute el script.

## Método

### ***getDate()***

Nos devuelve el día del mes del objeto fecha al que se aplica. Este método controla por supuesto el número de días de cada mes y contempla el caso de años bisiestos, incluida la excepción del 2000. En el siguiente ejemplo se presenta en pantalla Hoy es día 2, suponiendo que la fecha del sistema es 2-10-200. Primero creamos la variable fecha instanciada como un objeto Date() para a continuación escribir directamente el valor de getDate() aplicado a fecha

```
var fecha = new Date();
document.write("Hoy es día: "+fecha.getDate());
```

### ***setFullYear()***

Nos permite cambiar el año del objeto fecha por el valor pasado como argumento, un número de dos dígitos que se interpreta como decenas dentro del siglo, o sea, que para poner el año 1995 se debe pasar 95. El ejemplo pone precisamente este valor en el campo año de la variable fecha.

```
var fecha = new Date();
fecha.setYear(95)
document.write(fecha.toString());
```

Ojo si pasamos el valor 00 en el argumento el año obtenido es el 1900 (uno de los efectos 2000), por esto es recomendable usar la función setFullYear(agno).

### ***setDate()***

Nos permite cambiar el día del mes del objeto fecha al que se aplica para poner el valor que se pasado en el argumento días. Este método controla por supuesto el número de días de cada mes y contempla el caso de años bisiestos, incluida la excepción del 2000, de forma que si pasamos como argumento 31 y el mes es de 30 días la función corrige la fecha completa pasándola al día 1 del mes siguiente. Esto lo vemos en el ejemplo que sigue: creamos una variable como

un objeto `Date` con el último día de Septiembre (mes de 30 días) e intentamos poner el día a 31, luego comprobamos la fecha almacenada:

```
var fecha = new Date("1 Sep 2008");
fecha.setDate(31);
document.write("Hoy es día: " + fecha.toString());
```

Como puede verse, si se prueba el ejemplo la fecha es corregida y pasa a 1 de Octubre.

## Objetos del navegador.

Los objetos del navegador se encuentran en el nivel más alto de la jerarquía de objetos que define JavaScript. Estos son **window, frame, location, history, y navigator**.

El objeto más alto en la jerarquía de objetos que define JavaScript es **window**. El resto de objetos (excepto el objeto **navigator**) se ubican siempre dentro de una ventana.

Los métodos y propiedades de los objetos **window** y **navigator** pueden utilizarse sin necesidad de declarar variables.

Ejemplo:

```
//Mostrar la hora segundo a segundo
function hourStatus(){
  var miFecha = new Date();
  window.status = formato(miFecha.getHours()) + ":" + formato(miFecha.getMinutes()) + ":" + formato(miFecha.getSeconds());
  //Llamando a la función segundo a segundo
  setTimeout("hourStatus()", 1000);
}
//Función para formatear las partes de la hora
function formato(valor){
  if(valor<10) valor = "0" + valor;
  return valor;
}
```

## Objetos de documento

Los objetos del documento se encuentran en el segundo nivel de la jerarquía de objetos que define JavaScript. Estos son **document, link, anchor e image**. Estos objetos permiten a JavaScript, entre otras cosas, el manejo de Hojas de Estilo en Cascada y la ejecución de HTML dinámico (DHTML).

El acceso a los objetos del documento es proporcionado por el navegador mediante el uso de la **propiedad document** del **objeto window**.

Ejemplo:

```
//Función que obtiene el texto seleccionado
function getTextSelected(){
  var selText = "";
  //Internet Explorer
  if(document.selection){
    selText = document.selection.createRange().text;
  }
  //Otros navegadores (Chrome, Firefox, Opera, etc.)
  if(window.getSelection){
    selText = window.getSelection();
  }
  if(document.getSelection){
    selText = document.getSelection();
  }
  //Asignar el texto seleccionado a un elemento P
  document.getElementById("paragraph").innerHTML = "<p>Texto seleccionado: " + selText + "</p>";
}
```

Estas últimas dos categorías de objetos se estudiarán con más detalle en próximas guías de práctica.

### III. MATERIALES Y EQUIPO

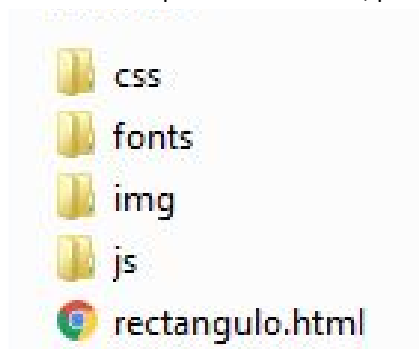
Para la realización de la guía de práctica se requerirá lo siguiente:

No.	Requerimiento	Cantidad
1	Guía de práctica #8: Objetos en JavaScript	1
2	Computadora con Sublime Text   PHP Designer 2007 y navegadores instalados	1
3	Memoria USB o disco flexible	1

### IV. PROCEDIMIENTO

Para los ejemplos del procedimiento de esta guía se ha utilizado el enfoque de desarrollo con JavaScript no invasivo. Esto significa que el manejo de eventos no se realizará en casi ningún caso con atributos manejadores de eventos dentro del código HTML. El manejo de eventos se hace con el modelo de eventos estándar, conocido como modelo de eventos Nivel 2 del DOM, que dicho sea de paso, es mucho más eficiente y propicia la completa separación de la funcionalidad del documento web de su estructura, algo muy similar a la separación de la apariencia de la estructura del documento lograda con el correcto manejo de las hojas de estilo en cascada.

Se debe de respetar la estructura, para todos los ejercicios :



**Ejemplo #1:** El ejemplo solicita dos valores numéricos (pueden ser con punto flotante) para calcular, presionando los botones respectivos, el perímetro y el área de un rectángulo con esas dimensiones. Se utilizan campos de formulario de tipo number, incorporados en HTML5 y se establecen valores mínimo, máximo y el desplazamiento para manejar los valores ingresados con el control que algunos navegadores modernos incorporan. También es factible ingresar los valores directamente.

#### Guion 1: rectangulo.html

```

<!DOCTYPE html>
<html lang="es">
<head>
  <title>Áreas de rectángulos</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="css/slick.css" />
  <link rel="stylesheet" href="css/demo.css" />
  <script src="js/calculo.js"></script>
</head>
<body>
<section id="slick">
  <!-- Área de formulario -->
  <div class="contact-form">
    <!-- Título -->

```



```

<div class="title">Cálculo de área y perímetro</div>
<!-- Texto indicativo -->
<p class="intro">Ingrese los siguientes datos</p>
<!-- Formulario -->
<div class="w-100">
  <!-- Campos de formulario -->
  <form name="frmrectangulo" id="frmrectangulo" action="javascript:void(0);">
    <!-- Campo para la base -->
    <div class="field">
      <input type="number" name="base" id="txtbase" min="0" max="1000"
step=".1" placeholder="(Base)" required />
      <span class="entypo-base icon"></span>
      <span class="slick-tip left">Ingrese la base:</span>
    </div>
    <!-- Campo para la altura -->
    <div class="field">
      <input type="number" name="altura" id="txtaltura" min="0" max="1000"
step=".1" placeholder="(Altura)" required />
      <span class="entypo-altura icon"></span>
      <span class="slick-tip left">Ingrese la altura:</span>
    </div>
    <!-- Botones para hacer los cálculos -->
    <input type="submit" value="area" class="send" name="btnarea" id="area" />
    <input type="submit" value="perimetro" class="send" name="btnperimetro"
id="perimetro" />
  </form>
  <!-- Fin de campos de formulario -->
</div>
</div>
<!-- Final del área del formulario -->
</section>
</body>
</html>

```

### Guión 2: calculo.js

```

//Registrar evento click del ratón al presionar botones de envío
function iniciar(){
  var btnarea = document.getElementById("area");
  var btnperim = document.getElementById("perimetro");
  if(btnarea.addEventListener){
    btnarea.addEventListener("click", calculararea, false);
  }
  else{
    btnarea.attachEvent("onclick", calculararea);
  }
  if(btnperim.addEventListener){
    btnperim.addEventListener("click", calcularperimetro, false);
  }
  else{
    btnperim.attachEvent("onclick", calcularperimetro);
  }
}

function calculararea(){
  var rect = new rectangulo(document.frmrectangulo.txtbase.value, document.frmrectangulo.txtaltura.value);
  rect.mostrar(rect.carea(), " área");
  return false;
}

function calcularperimetro(){
  var peri = new rectangulo(document.frmrectangulo.txtbase.value, document.frmrectangulo.txtaltura.value);

```

```

    peri.mostrar(peri.cperimetro(), 'perímetro');
    return false;
}

//Creando una clase rectángulo
function rectangulo(base, altura){
    //Propiedades de la clase
    this.base = parseFloat(base);
    this.altura = parseFloat(altura);
    //Métodos de la clase
    //definidos usando el constructor Function()
    this.carea = new Function("return this.base * this.altura");
    this.cperimetro = new Function("return 2*this.base + 2*this.altura");
    this.mostrar = new Function("valor","tipoc","alert('El ' + tipoc + ' es: ' + valor)");
}

//Asociando función que manejará el evento load al cargar la página
if(window.addEventListener){
    window.addEventListener("load", iniciar, false);
}
else if(window.attachEvent){
    window.attachEvent("onload", iniciar);
}
}

```

Resultado de la visualización en un navegador:

**Cálculo de área y perímetro**

Ingrese los siguientes datos

Base: 7.4

Altura: 5

PERIMETRO AREA

El perímetro es: 24.8

El área es: 37

Aceptar Aceptar

**Ejemplo #2:** El siguiente ejemplo muestra una forma de poder calcular la edad exacta de una persona, ingresando la fecha de nacimiento mediante campos de formulario donde debe ingresarse el día, el mes y el año de nacimiento. El cálculo de la edad exacta se hace con respecto a la fecha actual, que tendría que estar bien configurada en el sistema operativo del servidor o computadora donde se ejecuta el script.

#### Guión 1: edadexacta.html

```

<!DOCTYPE html>
<html lang="es">
<head>
    <title>Introducir fecha</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/fonts.css" />

```

```

<link rel="stylesheet" href="css/formstyle.css" />
<script src="js/calculoedad.js"></script>
<!-- Para las versiones anteriores -->
<!--[if lt IE 9]>
    <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>
<body>
<div class="cover">
    <form action="javascript:void(0)" name="frmFechaNac" id="contactform">
        <h2 class="contact">Tu fecha de nacimiento</h2>
        <span class="contact">
            <label for="dia">Día:</label>
        </span>
        <select name="seldia" id="dia" disabled>
        <script>
            for(i=1; i<=31; i++){
                document.write("<option value='" + i + "'">" + i + "</option>");
            }
        </script>
        </select>
        <span class="contact">
            <label for="mes">Mes:</label>&nbsp;
        </span>
        <select name="selmes" id="mes">
            <option value="Enero" selected>Enero</option>
            <option value="Febrero">Febrero</option>
            <option value="Marzo">Marzo</option>
            <option value="Abril">Abril</option>
            <option value="Mayo">Mayo</option>
            <option value="Junio">Junio</option>
            <option value="Julio">Julio</option>
            <option value="Agosto">Agosto</option>
            <option value="Septiembre">Septiembre</option>
            <option value="Octubre">Octubre</option>
            <option value="Noviembre">Noviembre</option>
            <option value="Diciembre">Diciembre</option>
        </select>
        <span class="contact">
            <label for="txtano">Año:</label>&nbsp;
        </span>
        <input type="text" value="2014" name="txtano" id="ano" size="4" maxlength="4" /><br />
        <input type="button" name="btncalcular" id="calcular" value="Tu edad" />
    </form>
</div>
</body>
</html>

```

### Guión 2: calculoedad.js

```

//Variable para determinar si un año es bisiesto o no
var bisiesto = false;

//Registrar evento click al presionar el botón de envío
function iniciar(){
    var dia, mes;
    var btncalcular = document.getElementById("calcular");
    //Al producirse un click sobre el botón de envío invocar
    //a la función edad para calcularla con los datos ingresados
    if(btncalcular.addEventListener){
        btncalcular.addEventListener("click", function(){
            dia = document.frmFechaNac.seldia.options[frmFechaNac.seldia.selectedIndex].value;

```

```

        mes = document.frmFechaNac.selm.es.options[frmFechaNac.selm.es.selectedIndex].value;
        edad(dia, mes, document.frmFechaNac.txtano.value);
    }, false);
}
else{
    alert(mes);
    btncalcular.attachEvent("onclick", function(){
        dia = document.frmFechaNac.seldia.options[frmFechaNac.seldia.selectedIndex].value;
        mes = document.frmFechaNac.selm.es.options[frmFechaNac.selm.es.selectedIndex].value;
        edad(dia, mes, document.frmFechaNac.txtano.value);
    });
}
//Si se cambia de mes en el campo select name=selm.es invocar
//a la función llenarDias() para volver a rellenar el campo
//select dependiente de los días de acuerdo al mes seleccionado
var selm.es = document.getElementById("mes");
if(selm.es.addEventListener){
    selm.es.addEventListener("change", function(){
        mes = document.frmFechaNac.selm.es.options[document.frmFechaNac.selm.es.selectedIndex].value;
        llenarDias(mes, document.frmFechaNac.seldia);
    }, false);
}
else if(selm.es.attachEvent){
    selm.es.attachEvent("onchange", function(){
        mes = document.frmFechaNac.selm.es.options[document.frmFechaNac.selm.es.selectedIndex].value;
        llenarDias(mes, document.frmFechaNac.seldia);
    });
}
}
//Del mismo modo, si se ingresa otro año, volver a rellenar
//el campo select de los días para considerar si se cambia
//a un año bisiesto o no
var txtano = document.getElementById("ano");
if(txtano.addEventListener){
    txtano.addEventListener("change", function(){
        mes = document.frmFechaNac.selm.es.options[document.frmFechaNac.selm.es.selectedIndex].value;
        llenarDias(mes, document.frmFechaNac.seldia);
    }, false);
}
else if(txtano.attachEvent){
    txtano.attachEvent("onchange", function(){
        mes = document.frmFechaNac.selm.es.options[document.frmFechaNac.selm.es.selectedIndex].value;
        llenarDias(mes, document.frmFechaNac.seldia);
    });
}
}
}

//Función para determinar si un año es bisiesto o no
function esBisiesto(anno){
    if(anno % 4 == 0 && (anno % 400 == 0 || anno % 100 != 0)){
        bisiesto = true;
    }
    else{
        bisiesto = false;
    }
    return bisiesto;
}

//Función para limpiar el menú de selección Días
function quitarDias(menuDias){
    for(i=0; i<menuDias.options.length; i++){
        menuDias.options[i] = null;
    }
}

```

```

    }
}

//Función para llenar la caja de texto de los días
function llenarDias(mes, menuDias){
    document.frmFechaNac.seldia.disabled = false;
    var i;
    quitarDias(menuDias);
    switch(mes){
        case 'Enero':
        case 'Marzo':
        case 'Mayo':
        case 'Julio':
        case 'Agosto':
        case 'Octubre':
        case 'Diciembre':
            for(i=0; i<31; i++){
                menuDias[i] = new Option(parseInt(i+1), parseInt(i+1));
            }
            break;
        case 'Abril':
        case 'Junio':
        case 'Septiembre':
        case 'Noviembre':
            for(i=0; i<30; i++){
                menuDias[i] = new Option(parseInt(i+1), parseInt(i+1));
            }
            break;
        case 'Febrero':
            if(esBisiesto(document.frmFechaNac.txtano.value)){
                for(i=0; i<29; i++){
                    menuDias[i] = new Option(parseInt(i+1), parseInt(i+1));
                }
            }
            else{
                for(i=0; i<28; i++){
                    menuDias[i] = new Option(parseInt(i+1), parseInt(i+1));
                }
            }
            break;
    }
}

//Función para obtener la edad exacta de la persona
function edad(dia, mes, annio){
    var tusdias, tusmeses, tusannios;
    var fecActual = new Date();
    var year = fecActual.getFullYear();
    var month = parseInt(fecActual.getMonth()) + 1;
    var day = fecActual.getDate();
    tusdias = day - dia;

    switch(mes){
        case 'Enero':
            mes = 1;
            if(tusdias < 0){
                tusdias = day - dia + 31;
                month--;
            }
            break;
        case 'Febrero':

```

```

mes = 2;
if(tusdias < 0 && esBisiesto(annio)){
    tusdias = day - dia + 29;
    month--;
}
else
    if(tusdias < 0 && !esBisiesto(annio)){
        tusdias = day - dia + 28;
        month--;
    }
    break;
case 'Marzo':
    mes = 3;
    if(tusdias < 0){
        tusdias = day - dia + 31;
        month--;
    }
    break;
case 'Abril':
    mes = 4;
    if(tusdias < 0){
        tusdias = day - dia + 30;
        month--;
    }
    break;
case 'Mayo':
    mes = 5;
    if(tusdias < 0){
        tusdias = day - dia + 31;
        month--;
    }
    break;
case 'Junio':
    mes = 6;
    if(tusdias < 0){
        tusdias = day - dia + 30;
        month--;
    }
    break;
case 'Julio':
    mes = 7;
    if(tusdias < 0){
        tusdias = day - dia + 31;
        month--;
    }
    break;
case 'Agosto':
    mes = 8;
    if(tusdias < 0){
        tusdias = day - dia + 31;
        month--;
    }
    break;
case 'Septiembre':
    mes = 9;
    if(tusdias < 0){
        tusdias = day - dia + 30;
        month--;
    }
    break;
case 'Octubre':

```

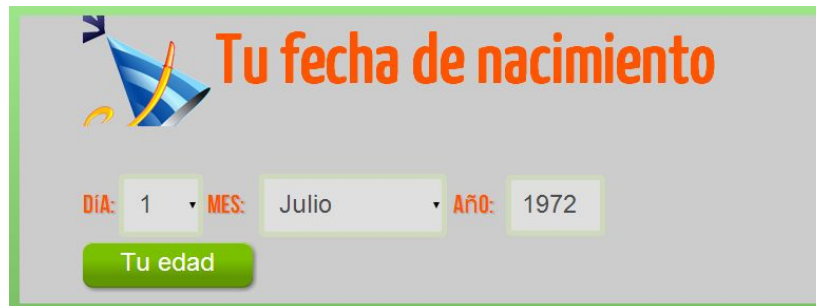
```

        mes = 10;
        if(tusdias < 0){
            tusdias = day - dia + 31;
            month--;
        }
        break;
    case 'Noviembre':
        mes = 11;
        if(tusdias < 0){
            tusdias = day - dia + 30;
            month--;
        }
        break;
    case 'Diciembre':
        mes = 12;
        if(tusdias < 0){
            tusdias = day - dia + 31;
            month--;
        }
        break;
    }
    tusmeses = month - mes;
    if(tusmeses < 0){
        tusmeses = month - mes + 12;
        year--;
    }
    tusannios = year - annio;
    alert("Tu edad exacta es:\n" +
        tusannios + " años,\n" +
        tusmeses + " meses, y\n" +
        tusdias + " días.");
}

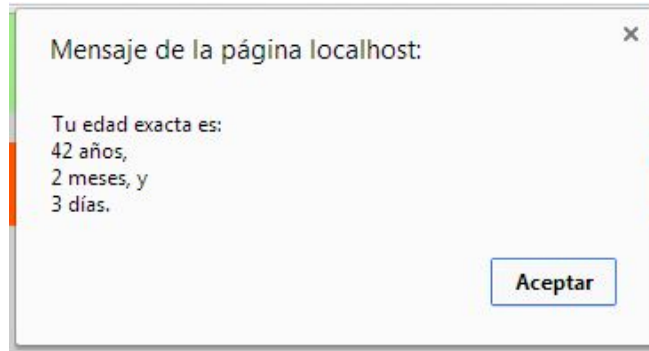
//Asociando función que manejará el evento load al cargar la página
if(window.addEventListener){
    window.addEventListener("load", iniciar, false);
}
else if(window.attachEvent){
    window.attachEvent("onload", iniciar);
}

```

Resultado al ejecutar el script activado con los controles de formulario:



The screenshot shows a web form titled "Tu fecha de nacimiento" (Your birth date) in orange text. To the left of the title is a blue and yellow party hat icon. Below the title, there are three input fields: "DÍA:" with the value "1", "MES:" with the value "Julio", and "AÑO:" with the value "1972". Each field has a small downward arrow indicating it's a dropdown menu. Below these fields is a green button with the text "Tu edad" (Your age).



**Ejemplo #3:** En el siguiente ejemplo se muestra cómo crear un script JS que permita indicar en base a una zona horaria mundial seleccionada por el usuario la hora y fecha de esa zona horaria. En este ejemplo se hace uso de los objetos incorporados de JavaScript Date para manejar la fecha y el objeto Math para realizar los cálculos de desfase horario entre las zonas.

#### Guión 1: zonahoraria.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Zona horaria internacional</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="css/elegant.css" />
  <script src="js/zonahoraria.js"></script>
  <!-- Para las versiones anteriores -->
  <!--[if lt IE 9]>
    <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
</head>
<body>
<header>
  <h1>Zonas del horario internacional</h1>
</header>
<section>
<article>
<form name="zonahoraria" class="elegant-aero">
  <h1>
    Zonas horarias del mundo
    <span>Seleccione su zona horaria.</span>
  </h1>
  <label>
    <span>Zonas horarias :</span>
    <select name="zonas" id="zhselect">
      <option value="-12">Línea internacional de fecha del oeste</option>
      <option value="-11">Isla Midway, Samoa</option>
      <option value="-10">Hawai</option>
      <option value="-9">Alaska</option>
      <option value="-8">Hora del Pacífico (USA y Canadá)</option>
      <option value="-7">Hora de las montañas rocosas (USA y Canadá)</option>
      <option value="-6">América Central</option>
      <option value="-5">Hora central (USA y Canadá)</option>
      <option value="-4">Hora del Atlántico (Canadá)</option>
      <option value="-3">Buenos Aires, Asunción, Brasilia, Montevideo</option>
      <option value="-2">Atlántico Central</option>
      <option value="-1">Azores</option>
    </select>
  </label>
</form>
</article>
</section>
</body>
</html>
```



```

        <option value="0" selected="selected">Hora del meridiano de Greenwich, Londres,
Dublin</option>
        <option value="+1">Paris, Madrid, Barcelona, Roma</option>
        <option value="+2">El Cairo</option>
        <option value="+3">Nairobi</option>
        <option value="+4">Bakú</option>
        <option value="+5">Ekaterimburgo</option>
        <option value="+6">Astana</option>
        <option value="+7">Bangkok</option>
        <option value="+8">Ulán Bator</option>
        <option value="+9">Tokio</option>
        <option value="+10">Sidney</option>
        <option value="+11">Islas Salomón</option>
        <option value="+12">Wellington</option>
    </select>
</label>
<label>
    <span>La hora en esta zona es :</span>
    <input type="text" name="hour" id="hour" placeholder="(zona horaria)"
readonly="readonly" />
</label>
<label>
    <span>&nbsp;</span>
    <input type="button" class="button" value="Cambiar" />
</label>
</form>
</article>
</section>
</body>
</html>

```

## Guión 2: zonahoraria.js

```

//La fecha del día en la zona del cliente web
//usando el objeto Date de JavaScript
var fechaHoy = new Date();

//Cálculo del desfase en el huso horario para hacer
//luego el ajuste con la zona horaria seleccionada
//en el elemento select del formulario
var desfase = -(Math.round(fechaHoy.getTimezoneOffset()/60)) + 12;

//Registrar evento click del ratón al presionar botones de envío
function iniciar(){
    var select = document.getElementById("zhselect");
    if(select.addEventListener){
        select.addEventListener("change", getHoraLocal, false);
    }
    else if(select.attachEvent){
        select.attachEvent("onchange", getHoraLocal);
    }
}

//Función getHoraLocal()
//Calcula la hora en la zona horaria seleccionada
function getHoraLocal(){
    var fechaHoy = new Date();
    //Ajustar el desfase horario respecto a la zona seleccionada.
    //Por ejemplo, la zona horaria para América Central está
    //en el índice 6 de la colección de opciones del elemento select.
    //Si un usuario seleccionara la zona horaria de Tokio que está
    //en el índice 21 de la colección el cálculo se realizaría así:

```

```
//21 - desfase del cliente (zona horaria de América Central) = 15
var zh = document.zonahoraria.zonas.selectedIndex - desfase;

//Sumar (o restar) las horas de desfase respecto a la hora
//local del cliente
fechaHoy.setHours(fechaHoy.getHours() + zh);

//Informar la hora local del huso horario elegido
document.zonahoraria.hour.value = fechaHoy.toLocaleString();
}

//Asociando función que manejará el evento load al cargar la página
if(window.addEventListener){
    window.addEventListener("load", iniciar, false);
}
else if(window.attachEvent){
    window.attachEvent("onload", iniciar);
}
```

Resultado al ejecutar el script con un navegador:

**Zonas del horario internacional**

**Zonas horarias del mundo**  
 Seleccione su zona horaria.

Zonas horarias : Paris, Madrid, Barcelona, Roma

La hora en esta zona es : 31/8/2015 18:54:53

Cambiar

---

**Zonas horarias del mundo**  
 Seleccione su zona horaria.

Zonas horarias : Hora del meridiano de Greenwich, Londres, Dublin

La hora en esta zona es :

- Línea internacional de fecha del oeste
- Isla Midway, Samoa
- Hawai
- Alaska
- Hora del Pacífico (USA y Canadá)
- Hora de las montañas rocosas (USA y Canadá)
- América Central
- Hora central (USA y Canadá)
- Hora del Atlántico (Canadá)
- Buenos Aires, Asunción, Brasilia, Montevideo
- Atlántico Central
- Azores
- Hora del meridiano de Greenwich, Londres, Dublin**
- Paris, Madrid, Barcelona, Roma
- El Cairo
- Nairobi
- Bakú
- Ekaterimburgo
- Astana
- Bangkok

**Ejemplo #4:** Ejemplo que permite ingresar en un formulario datos de un libro que luego son almacenados en propiedades de un objeto libro definido con el objeto Object de JavaScript. Se verifica que los datos obtenidos sean correctos, se limpian los campos del formulario y finalmente se construye una tabla HTML con uno de los métodos del objeto donde se muestran los datos ya almacenados en las propiedades de ese objeto creado.

#### Guión 1: libros.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Creación de objetos</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="css/font.css" />
  <link rel="stylesheet" href="css/masked.css" />
  <link rel="stylesheet" href="css/libro.css" />
  <script src="js/libro.js"></script>
</head>
<body>
<div class="masked">
  <h1>Información del libro</h1>
</div>
<form action="javascript:void(0)" name="frmbook" id="frmbook">
  <table>
    <tr>
      <td>Autor: </td>
      <td>
        <input type="text" name="txtautor" id="txtautor" maxlength="60" />
      </td>
    </tr>
    <tr>
      <td>Título: </td>
      <td>
        <input type="text" name="txttitulo" id="txttitulo" maxlength="90" />
      </td>
    </tr>
    <tr>
      <td>Editorial: </td>
      <td>
        <select name="seleditorial" size="1">
          <option value="a" selected="selected">Mc-Graw Hill</option>
          <option value="b">Pearson</option>
          <option value="a">Alfa-omega</option>
        </select>
      </td>
    </tr>
  </table>
</form>
```

```

        <option value="a">Anaya</option>
    </select>
</td>
</tr>
<tr>
    <td>Edición: </td>
    <td>
        <select name="seledicion" size="1">
            <option value="a" selected="selected">1ra</option>
            <option value="b">2da</option>
            <option value="c">3ra</option>
            <option value="d">4ta</option>
            <option value="e">5ta</option>
            <option value="f">6ta</option>
        </select>
    </td>
</tr>
<tr>
    <td>País: </td>
    <td>
        <input type="text" name="txtpais" id="txtpais" maxlength="36" />
    </td>
</tr>
<tr>
    <td colspan="2" class="combinar">
        <input type="button" name="mostrar" id="mostrar" value="Enviar" class="boton" />
    </td>
</tr>
</table>
</form>
</body>
</html>

```

## Guión 2: libro.js

```

//Creando el objeto
var book = new Object();

//Registrar evento click del ratón al presionar botones de envío
function iniciar(){
    var showinfo = document.getElementById("mostrar");
    if(showinfo.addEventListener){
        showinfo.addEventListener("click", function(){
            createObject(document.frmbook);
        }, false);
    }
    else if(showinfo.attachEvent){
        showinfo.attachEvent("onclick", function(){
            createObject(document.frmbook);
        });
    }
}

// Creando el nuevo objeto
function createObject(form){
    book.autor = form.txtautor.value;
    book.titulo = form.txttitulo.value;
    book.editorial = form.seleditorial.options[form.seleditorial.selectedIndex].text;
}

```

```

    book.edicion = form.seledicion.options[form.seledicion.selectedIndex].text;
    book.pais = form.txtpais.value;
    showProperties(book, "InfoBook");
}

function showProperties(objeto, objName){
    var infBook = "";
    var tblBook = "";
    for(var i in objeto){
        infBook = infBook + objName + "." + i + " = " + objeto[i] + "\n";
    }
    if(!confirm(infBook + "\n\n¿Es esta información correcta?")){
        frmbook.txtautor.value = "";
        frmbook.txttitulo.value = "";
        frmbook.seleditorial.value = "a";
        frmbook.seledicion.value = "a";
        frmbook.txtpais.value = "";
    }
    tblBook += "<!DOCTYPE html>\n";
    tblBook += "<html lang='es'>\n";
    tblBook += "<head>\n\t";
    tblBook += "<title>Información del libro</title>\n";
    tblBook += "<meta charset='utf-8' />";
    tblBook += "<link rel='stylesheet' href='css/infolibro.css' />\n";
    tblBook += "</head>\n";
    tblBook += "<body>\n";
    tblBook += "<table class='bookinfo'>";
    tblBook += "<thead>\n\t<tr>\n\t\t";
    tblBook += "<th>Título</th>\n\t\t";
    tblBook += "<th>Autor</th>\n\t\t";
    tblBook += "<th>Editorial</th>\n\t\t";
    tblBook += "<th>Edición</th>\n\t\t";
    tblBook += "<th>País</th>\n\t\t";
    tblBook += "</tr>\n\t</thead>\n";
    tblBook += "<tbody>\n\t";
    tblBook += "<tr>\n\t\t";
    tblBook += "<td>" + book.titulo + "</td>\n\t\t";
    tblBook += "<td>" + book.autor + "</td>\n\t\t";
    tblBook += "<td>" + book.editorial + "</td>\n\t\t";
    tblBook += "<td>" + book.edicion + "</td>\n\t\t";
    tblBook += "<td>" + book.pais + "</td>\n\t\t";
    tblBook += "</tr></tbody>\n";
    tblBook += "</table>\n";
    tblBook += "</head>\n";
    tblBook += "</html>\n";
    document.write(tblBook);
}

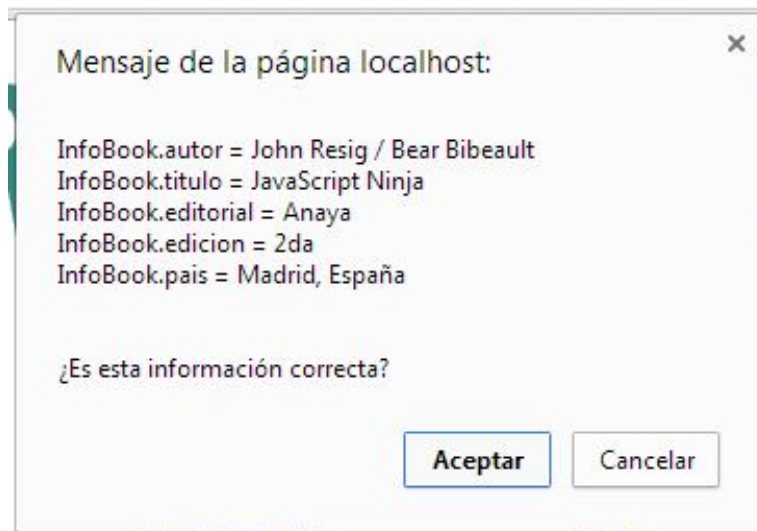
//Asociando función que manejará el evento load al cargar la página
if(window.addEventListener){
    window.addEventListener("load", iniciar, false);
}
else if(window.attachEvent){
    window.attachEvent("onload", iniciar);
}

```

Resultado:

# INFORMACIÓN DEL LIBRO

Autor:	John Resig / Bear Bibeault
Título:	JavaScript Ninja
Editorial:	Anaya ▼
Edición:	2da ▼
País:	Madrid, España
<input type="button" value="Enviar"/>	



Título	Autor	Editorial	Edición	País
JavaScript Ninja	John Resig / Bear Bibeault	Anaya	2da	Madrid, España

**Ejemplo #5:** Este ejemplo muestra cómo controlar con JavaScript las opciones de un campo select dependiente de otro campo select. De modo que al cambiar la opción seleccionada en el campo select principal, se limpia por completo el campo select dependiente y se vuelve a llenar inmediatamente con las opciones que se corresponden con la opción seleccionada.

## Guión 1: carro.html

```

<!DOCTYPE html>
<html lang="es">
<head>
  <title>Clases y objetos</title>

```

```

<meta charset="utf-8" />
<link rel="stylesheet" href="css/font.css" />
<link rel="stylesheet" href="css/carform.css" />
<script src="js/showcar.js"></script>
</head>
<body>
<header id="insetBgd">
  <h1>Ejemplo del uso de objetos</h1>
</header>
<section>
<article>
<form name="frmcar" id="frmcar">
<fieldset id="fldcar" class="group">
  <legend><span class="labelgroup">Datos del carro</span></legend>
  <label for="selfab">Fabricante: </label>
  <select name="selfab" id="selfab">
    <option value="Toyota">Toyota</option>
    <option value="Nissan">Nissan</option>
    <option value="Hyundai">Hyundai</option>
    <option value="Volkswagen">Volkswagen</option>
    <option value="Chevrolet">Chevrolet</option>
    <option value="Honda">Honda</option>
    <option value="Mitsubishi">Mitsubishi</option>
  </select>
  <label for="selmod">Modelo: </label>
  <select name="selmod" id="selmod">
    <option value="Corolla">Corolla</option>
    <option value="Echo">Echo</option>
    <option value="Yaris">Yaris</option>
    <option value="Avensis">Avensis</option>
    <option value="Camry">Camry</option>
    <option value="Land Cruiser">Land Cruiser</option>
    <option value="4 Runner">4 Runner</option>
    <option value="Hilux">Hilux</option>
  </select>
  <label for="radcolor" class="colores">Color:</label><br />
  <input type="radio" value="Blanco" alt="Marque el color del carro" id="radcolor" name="radcolor" />
  <label for="radcolor" class="colores">Blanco</label><br>
  <input type="radio" value="Negro" alt="Marque el color del carro" id="radcolor" name="radcolor" />
  <label for="radcolor" class="colores">Negro</label><br>
  <input type="radio" value="Gris" alt="Marque el color del carro" id="radcolor" name="radcolor" />
  <label for="radcolor" class="colores">Gris</label><br>
  <input type="radio" value="Azul" alt="Marque el color del carro" id="radcolor" name="radcolor" />
  <label for="radcolor" class="colores">Azul</label><br>
  <input type="radio" value="Verde" alt="Marque el color del carro" id="radcolor" name="radcolor" />
  <label for="radcolor" class="colores">Verde</label><br>
  <input type="radio" value="Rojo" alt="Marque el color del carro" id="radcolor" name="radcolor" />
  <label for="radcolor" class="colores">Rojo</label><br>
  <input type="radio" value="Amarillo" alt="Marque el color del carro" id="radcolor" name="radcolor" />
  <label for="radcolor" class="colores">Amarillo</label><br>
  <input type="radio" value="Beige" alt="Marque el color del carro" id="radcolor" name="radcolor" />
  <label for="radcolor" class="colores">Beige</label> <br />
  <label for="txtanio">Año: </label>
  <input type="number" name="txtanio" placeholder="2014" maxlength="4" />
  <input type="button" name="btnEnviar" id="enviar" value="Enviar" />
</fieldset>
</form>
</article>
</section>
</body>
</html>

```

### Guión 2: showcar.js

```
//Registrar evento click al presionar botones de envío
//y evento change al cambiar de opción en el elemento select
function iniciar(){
    var select = document.getElementById("selfab");
    var button = document.getElementById("enviar");
    //Al producirse en evento change en el elemento select
    //invocar a la función addOptions para volver a llenar
    //el select dependiente con los datos adecuados
    if(select.addEventListener){
        select.addEventListener("change", function(){
            addOptions(marcas[this.options[this.selectedIndex].text], document.frmcar.selmod);
        }, false);
    }
    else{
        select.attachEvent("onchange", function(){
            addOptions(marcas[this.options[this.selectedIndex].text], document.frmcar.selmod);
        });
    }
    //Al producirse un click sobre el botón de envío
    //invocar los métodos del objeto carro que mostrarán
    //los valores ingresados en el formulario
    if(button.addEventListener){
        button.addEventListener("click", function(){
            var seleccion = showRadioSelected(document.frmcar.radcolor);
            carro.pedido(document.frmcar.selfab.value, document.frmcar.selmod.value, seleccion, document.frmcar.txtanio.value);
            carro.mostrar();
        }, false);
    }
    else{
        button.attachEvent("onclick", function(){
            var seleccion = showRadioSelected(document.frmcar.radcolor);
            carro.pedido(document.frmcar.selfab.value, document.frmcar.selmod.value, seleccion, document.frmcar.txtanio.value);
            carro.mostrar();
        });
    }
}

//Inicializando matriz para manejar las marcas y sus respectivos modelos
var marcas = new Array(7);
marcas["Toyota"] = ["Corolla", "Echo", "Yaris", "Avensis", "Camry", "Land Cruiser", "4 Runner", "Hilux"];
marcas["Nissan"] = ["Sentra", "Platina", "Almera", "Altima", "Tiida", "Pathfinder", "Patrol", "X-Trail", "Frontier"];
marcas["Hyundai"] = ["Elantra", "Accent", "Coupé", "Santa Fe", "i30"];
marcas["Volkswagen"] = ["Golf", "Jetta", "Passat", "Phaeton", "Thunder Bunny", "Touareg", "Saveiro"];
marcas["Chevrolet"] = ["Optra", "Aveo", "Cobalt", "Malibu", "Corvette", "Chevy", "Avalanche", "Trailblazer"];
marcas["Honda"] = ["Civic", "Acura", "Accord", "Fit", "Odyssey", "CR-V", "Pilot", "RidgeLine"];
marcas["Mitsubishi"] = ["Lancer", "Galant", "Eclipse", "Montero", "Nativa", "Outlander", "L200"];

//Creando el objeto carro con el constructor Object()
var carro = new Object();
//Propiedades del objeto
carro.fabricante = "";
carro.modelo = "";
carro.color = "";
carro.anio = "";
//Métodos del objeto
carro.pedido = function(fab, mod, col, an){
    carro.fabricante = fab;
    carro.modelo = mod;
    carro.color = col;
```



```

        carro.anio = an;
    }
    carro.mostrar = function(){
        document.write("<!DOCTYPE html>\n");
        document.write("<html lang='es'>\n");
        document.write("<head>\n\t");
        document.write("<title>Datos del carro</title>\n");
        document.write("<meta charset='utf-8' />");
        document.write("<link rel='stylesheet' href='css/carinfo.css' />\n");
        document.write("</head>\n");
        document.write("<body>\n");
        document.write("<table class='carinfo'><tr>\n");
            document.write("<th colspan='2'>Datos del carro</th>\n");
            document.write("<tr><td>Fabricante: </td>\n");
            document.write("<td>" + carro.fabricante + "</td></tr>\n");
            document.write("<tr><td>Modelo: </td>\n");
            document.write("<td>" + carro.modelo + "</td></tr>\n");
            document.write("<tr><td>Color: </td>\n");
            document.write("<td>" + carro.color + "</td></tr>\n");
            document.write("<tr><td>Año: </td>\n");
            document.write("<td>" + carro.anio + "</td></tr>\n");
            document.write("</tr></table>\n");
            document.write("</head>\n");
            document.write("</html>\n");
        }
    }

    function showRadioSelected(radiogroup){
        var seleccionado;
        var numradios = radiogroup.length;
        for(var i=0; i<numradios; i++){
            if(radiogroup[i].checked){
                seleccionado = radiogroup[i].value;
            }
        }
        return seleccionado;
    }

    function removeOptions(optionMenu){
        for(i=0; i<optionMenu.options.length; i++){
            optionMenu.options[i] = null;
        }
    }

    function addOptions(optionList, optionMenu){
        var i=0;
        removeOptions(optionMenu); //Limpia las opciones
        for(i=0; i<optionList.length; i++){
            optionMenu[i] = new Option(optionList[i], optionList[i]);
        }
    }

    //Asociando función que manejará el evento load al cargar la página
    if(window.addEventListener){
        window.addEventListener("load", iniciar, false);
    }
    else if(window.attachEvent){
        window.attachEvent("onload", iniciar);
    }
}

```

Resultado en el navegador:

### Ejemplo del uso de objetos

**Datos del carro**

Fabricante:   
 Toyota

Modelo:   
 Corolla

Color:   
☐ Blanco   
☐ Negro   
☐ Gris   
☐ Azul   
☐ Verde   
☐ Rojo   
☒ Amarillo   
☐ Beige

Año:   
 2007

Enviar

Datos del carro	
Fabricante:	Toyota
Modelo:	Corolla
Color:	Beige
Año:	2007

**Ejemplo #6:** El siguiente ejemplo simula el proceso de matrícula de un alumno. Para ello se llenan los campos de un formulario con datos básicos. Con un script JS se obtienen los datos ingresados en los campos del formulario y se crean propiedades para un

#### Guión 1: matricula.html

```

<!DOCTYPE html>
<html lang="es">
<head>
  <title>Crear y editar tablas con JS</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="css/modernform.css" />
  <script type="text/javascript" src="js/alumnodb.js"></script>
  <!-- Para las versiones anteriores -->
  <!--[if lt IE 9]>
    <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
</head>
<body>
<div class="container">
<section class="tabblue">
  <ul class="tabs blue">
    <li>
      <input name="tabs blue" id="tab1" checked="" type="radio">
      <label for="tab1">Datos del alumno</label>
      <div id="tab-content1" class="tab-content">
        <p>
          Ingresa los siguientes datos para completar tu registro.
        </p>
        <form name="frmmat" id="pais">
        <!-- <form name="register" action="" --> -->
      </div>
    </li>
  </ul>
</section>
</div>

```

```

<span class="tabaddon"><i class="fa fa-user fa-2x"></i></span>
<input type="text" name="txtname" id="txtname" required="required" placeholder="(Su nombre)" class="field" />
<span class="tabaddon"><i class="fa fa-envelope fa-2x"></i></span>
<input type="text" name="txtlastname" id="txtlastname" required="required" placeholder="(Su apellido)" class="field" />
<span class="tabaddon"><i class="fa fa-lock fa-2x"></i></span>
<input type="number" name="txtage" id="txtage" required="required" placeholder="(Su edad)" class="field" />
<div class="accounttype">
  <p>Género:</p>
  <input type="radio" value="Masculino" id="radioOne" name="chkgender" checked="checked" />
  <label for="radioOne" class="radio" chec>Masculino</label>
  <input type="radio" value="Femenino" id="radioTwo" name="chkgender" />
  <label for="radioTwo" class="radio">Femenino</label>
</div>
<span class="tabaddon"><i class="fa fa-degree fa-2x"></i></span>
<select name="seldegree" id="seldegree">
  <option disabled="" selected="" value="0">(Seleccione una carrera)</option>
  <option value="Ingeniería en Computación">Ingeniería en Computación</option>
  <option value="Ingeniería en Mecatrónica">Ingeniería en Mecatrónica</option>
  <option value="Ingeniería Industrial">Ingeniería Industrial</option>
  <option value="Ingeniería Biomédica">Ingeniería Biomédica</option>
  <option value="Ingeniería en Telecomunicaciones">Ingeniería en Telecomunicaciones</option>
  <option value="Ingeniería Aeronáutica">Ingeniería Aeronáutica</option>
  <option value="Licenciatura en Mercadotecnia">Licenciatura en Mercadotecnia</option>
  <option value="Licenciatura en Idiomas">Licenciatura en Idiomas</option>
  <option value="Licenciatura de Diseño Gráfico">Licenciatura de Diseño Gráfico</option>
  <option value="Licenciatura en Comunicación">Licenciatura en Comunicación</option>
  <option value="Licenciatura en Administración de Empresas">Licenciatura en Administración de Empresas</option>
  <option value="Técnico en Computación">Técnico en Computación</option>
  <option value="Técnico en Multimedia">Técnico en Multimedia</option>
  <option value="Técnico en Ortesis y Prótesis">Técnico en Ortesis y Prótesis</option>
</select>
<div class="btn">
  <input type="button" name="btnSend" id="btnSend" value="Matricular" />
</div>
</form>
</div>
</li>
</ul>
</section>
</div>

<!-- Formulario original -->
<!-- <form name="frmmat" id="pais">
  <fieldset id="wrapper">
    <legend>
      <span>Datos del alumno:</span>
    </legend>
    <label for="txtname">Nombre:</label>
    <input type="text" name="txtname" id="txtname" maxlength="30" value="" placeholder="(Nombre)" /><br />
    <label for="txtlastname">Apellido:</label>
    <input type="text" name="txtlastname" id="txtlastname" maxlength="30" value="" placeholder="(Apellido)" /><br />
    <label for="txtage">Edad:</label>
    <input type="number" name="txtage" id="txtage" maxlength="2" value="" placeholder="(Su edad)" /><br />
    <label for="txtdegree">Carrera:</label>
    <select name="txtdegree" id="txtdegree">
      <option value="Ingeniería en Computación">Ingeniería en Computación</option>
      <option value="Ingeniería en Mecatrónica">Ingeniería en Mecatrónica</option>
      <option value="Ingeniería Industrial">Ingeniería Industrial</option>
      <option value="Ingeniería Biomédica">Ingeniería Biomédica</option>
      <option value="Ingeniería en Telecomunicaciones">Ingeniería en Telecomunicaciones</option>
      <option value="Ingeniería Aeronáutica">Ingeniería Aeronáutica</option>
    </select>
  </fieldset>
</form>
-->

```

```

<option value="Licenciatura en Mercadotecnia">Licenciatura en Mercadotecnia</option>
<option value="Licenciatura en Idiomas">Licenciatura en Idiomas</option>
<option value="Licenciatura de Diseño Gráfico">Licenciatura de Diseño Gráfico</option>
<option value="Licenciatura en Comunicación">Licenciatura en Comunicación</option>
<option value="Licenciatura en Administración de Empresas">Licenciatura en Administración de Empresas</option>
<option value="Técnico en Computación">Técnico en Computación</option>
<option value="Técnico en Multimedia">Técnico en Multimedia</option>
<option value="Técnico en Ortesis y Prótesis">Técnico en Ortesis y Prótesis</option>
</select>
<input type="button" name="btnSend" id="btnSend" value="Matricular" class="css3dbutton" />
</fieldset>
</form> -->
</body>
</html>

```

## Guión 2: alumnoudb.js

```

//Registrar evento click al presionar el botón de envío
function iniciar(){
    var btnenviar = document.getElementById("btnSend");
    //Al producirse un click sobre el botón de envío
    //invocar los métodos del objeto carro que mostrarán
    //los valores ingresados en el formulario
    if(btnenviar.addEventListener){
        btnenviar.addEventListener("click", function(){
            var chkvalue, selvalue, nuevoalumno;
            var radiofield = document.frmmat.elements["chkgender"];
            for(var i=0; i<radiofield.length; i++){
                if(radiofield[i].checked){
                    chkvalue = radiofield[i].value;
                }
            }
            selvalue = document.frmmat.seldegree.options[frmmat.seldegree.selectedIndex].value;
            nuevoalumno = new alumnoUDB(document.frmmat.txtname.value, document.frmmat.txtlastname.value,
document.frmmat.txtage.value, chkvalue, selvalue);
            nuevoalumno.matricular();
            nuevoalumno.imprimir();
        }, false);
    }
    else{
        btnenviar.attachEvent("onclick", function(){
            var chkvalue, nuevoalumno;
            var radiofield = document.frmmat.elements["chkgender"];
            for(var i=0; i<radiofield.length; i++){
                if(radiofield[i].checked){
                    chkvalue = radiofield[i].value;
                }
            }
            selvalue = document.frmmat.seldegree.options[frmmat.seldegree.selectedIndex].value;
            nuevoalumno = new alumnoUDB(document.frmmat.txtname.value, document.frmmat.txtlastname.value,
document.frmmat.txtage.value, chkvalue, selvalue);
            nuevoalumno.matricular();
            nuevoalumno.imprimir();
        });
    }
}

//Definiendo la clase alumnoUDB haciendo uso de sintaxis de función
function alumnoUDB(nombre, apellido, edad, genero, carrera){
    //Propiedades de la clase
    this.nombre = nombre;
    this.apellido = apellido;

```

```

this.edad = edad;
this.genero = genero;
this.carrera = carrera;
this.numCarnet = null;
//Métodos de la clase
this.matricular = function(){
    var fecha = new Date();
    var year = fecha.getFullYear();
    var day = fecha.getDate();
    var sec = fecha.getSeconds();
    this.numCarnet = this.nombre.substring(0,1) + this.apellido.substring(0,1) + this.formato(sec) + this.formato(day) + year;
}
this.imprimir = function(){
    document.write("<!DOCTYPE html>\n");
    document.write("<html lang='es'>\n");
    document.write("<head>\n\t");
    document.write("<title>Alumno matriculado</title>\n");
    document.write("<meta charset='utf-8' />");
    document.write("<link rel='stylesheet' href='css/infoalumno.css' />\n");
    document.write("</head>\n");
    document.write("<body>\n");
    document.write("<table class='carinfo'><tr>\n");
    document.write("<th colspan='2'>Datos del alumno</th>\n");
    document.write("<tr><td>Carnet: </td>\n");
    document.write("<td>" + this.numCarnet + "</td></tr>\n");
    document.write("<tr><td>Nombre: </td>\n");
    document.write("<td>" + this.nombre + " " + this.apellido + "</td></tr>\n");
    document.write("<tr><td>Edad: </td>\n");
    document.write("<td>" + this.edad + "</td></tr>\n");
    document.write("<tr><td>Género: </td>\n");
    document.write("<td>" + this.genero + "</td></tr>\n");
    document.write("<tr><td>Carrera: </td>\n");
    document.write("<td>" + this.carrera + "</td></tr>\n");
    document.write("</tr></table>\n");
    document.write("</head>\n");
    document.write("</html>\n");
}
this.formato = function(valor){
    if(valor<10) valor = "0" + valor;
    return valor;
}
}

//Asociando función que manejará el evento load al cargar la página
if(window.addEventListener){
    window.addEventListener("load", iniciar, false);
}
else if(window.attachEvent){
    window.attachEvent("onload", iniciar);
}
}

```

Resultado en el navegador:

Datos del alumno

Ingresa los siguientes datos para completar tu registro.

Género:

☐ Masculino

☒ Femenino

Matricular

Datos del alumno	
Carnet:	ID38042014
Nombre:	Irma Aracely Dueñas R.
Edad:	36
Género:	Femenino
Carrera:	Licenciatura en Comunicación

## V. DISCUSIÓN DE RESULTADOS

1. Realice una aplicación orientada a objetos llamada potencia. Su aplicación debe crear una clase utilizando funciones, en donde las propiedades deben ser tres: la base y la potencia que deben obtener sus valores de dos argumentos enviados a la definición de la función que hará las veces de la clase. Una tercera propiedad para obtener la potencia de elevar la base a la potencia recibida. Su clase debe poseer dos métodos, uno para retornar el cálculo de la potencia con las propiedades de la clase y otro para mostrar en la página web mediante el uso de la propiedad innerHTML el resultado del cálculo de la potencia haciendo uso del método para tal propósito. Debe realizar una interfaz de formulario apropiada para este problema y no usar cuadros prompt para obtener los datos.

2. Siguiendo la metodología de los ejemplos 5 y 6 (función anónima u objeto Object) del procedimiento, cree un objeto llamado miembro que sirva para registrar usuarios en un sitio web. El usuario debe ingresar cierta información básica, como la que se muestra en el formulario de la siguiente imagen y al simular el envío de los datos un script JS debe almacenar la información ingresada en las propiedades del objeto registro que deben coincidir con los nombres o ids de los campos del formulario. Además de esos campos, debe existir una propiedad identificadora en el objeto que deberá crear de la siguiente forma: las iniciales de los dos apellidos (si sólo tiene un apellido duplicar la primera letra del primer apellido) concatenadas con el año de registro y un número aleatorio de 4 cifras que deberá crear mediante el uso del método random del objeto Math. Investigue, si es necesario. Al final, después de simular el envío de los datos debe mostrar todas las propiedades del objeto y sus valores asignados en una tabla con estilos.

## VI. BIBLIOGRAFIA

- Deitel, Paul / Deitel, Harvey / Deitel, Abbey. Internet & World Wide Web. Cómo programar. 5a. Edición. Editorial Pearson. 2014. México D.F..
- John Resig / Bear Bibeault. JavaScript Ninja.1a. Edición. Editorial Anaya Multimedia / Manning. Septiembre 2013. Madrid, España.
- Flanagan, David. JavaScript La Guía Definitiva. 1ra Edición. Editorial ANAYA Multimedia. 2007. Madrid, España.
- Tom Negrino / Dori Smith. JavaScript & AJAX para diseño web. 6ta. Edición. Pearson/Prentice Hall. 2007. Madrid, España.
- Scott McCracken. Curso de Programación Web. 1ra. Edición. Inforbooks ediciones. 2011. Barcelona, España.
- Comparación entre lenguajes basados en clases contra los lenguajes basados en prototipos. Página de soporte para desarrolladores de JavaScript orientado al navegador Mozilla Firefox. [http://developer.mozilla.org/es/docs/Gu%C3%ADa\\_JavaScript\\_1.5:Lenguajes\\_basados\\_en\\_clases\\_frente\\_a\\_basados\\_en\\_prototipos](http://developer.mozilla.org/es/docs/Gu%C3%ADa_JavaScript_1.5:Lenguajes_basados_en_clases_frente_a_basados_en_prototipos).