



Gustavo Alberto Maidana Scaroni

Obligatorio

Programación 1

Analista en Tecnologías de la Información

Grupo: M1B

Docente: Benjamin Oholeguy

UNIVERSIDAD ORT

Índice

1	Introducción:.....	3
2	Aplicación:	3
2.1	Niveles de ejercicios:.....	3
2.1.1	Nivel básico:.....	3
2.1.2	Nivel Medio:.....	3
2.1.3	Nivel Avanzado:.....	3
2.2	Codigo de la Aplicación:.....	4
2.2.1	Funciones creadas:	4
2.2.1.1	Funcion numAleatEnt:.....	4
	Esta función toma dos valores máximo y mínimo y retorna un numero entero comprendido entre los valores de entrada.	4
2.2.1.2	Funcion numAleatDec :	4
2.2.1.3	Función auxComb:.....	4
2.2.1.4	Funcion cargoUnidades:.....	4
2.2.1.5	Funcion ejercicio:	4
2.2.1.6	Funcion cargoEventos:.....	4
2.2.1.7	Funcion básicos:.....	4
2.2.1.8	Funcion noBasicos:.....	4
2.2.1.9	Funcion intermedios :	5
2.2.1.10	Funcion avanzados:.....	5
2.2.1.11	Funcion valorNoVacio:.....	5
2.2.1.12	Funcion correcto:	5
2.2.1.13	funcion incorrecto:	5
2.2.1.14	Funcion verificoRes:	5
2.2.1.15	Funcion eliminarTxt:.....	5
2.2.1.16	Funcion mostrarAyuda:.....	5
2.2.2	Codigo javascript:	5
3	Capturas de prueba:.....	12
3.1	FireFox	12
3.2	Opera	12
3.3	Chrome	14

1 Introducción:

Se implementa una aplicación que evalúe el resultado de la conversión de unidades de superficie en alumnos de 5° año de Educación Primaria, generando de modo aleatorio ejercicios de conversión en tres niveles; básico, medio y avanzado.

Cada nivel tiene una particularidad en el ejercicio planteado, en cuanto a los números y unidades que se presentan, aumentando la complejidad del mismo.

2 Aplicación:

2.1 Niveles de ejercicios:

2.1.1 Nivel básico:

Este nivel plantea la conversión de unidades de superficie con hasta un salto en la unidad, es decir de cm^2 a m^2 por ejemplo. Los números propuestos estarán comprendidos entre 99 y 0, pueden ser enteros o con un decimal.

2.1.2 Nivel Medio:

Este nivel agrega un salto a las unidades planteadas, por ejemplo de cm^2 a dam^2 . Los números estarán comprendidos también entre 0 y 99, pueden ser enteros o tener hasta dos decimales.

2.1.3 Nivel Avanzado:

Este nivel aumenta la diferencia entre unidades pudiéndose convertir de cm^2 a km^2 por ejemplo, también aumenta el rango de números entre 100 y 999, pudiendo ser enteros o con hasta 3 decimales.

2.2 Código de la Aplicación:

2.2.1 Funciones creadas:

2.2.1.1 Funcion numAleatEnt:

Esta función toma dos valores máximo y mínimo y retorna un número entero comprendido entre los valores de entrada.

2.2.1.2 Funcion numAleatDec :

Esta función genera un número aleatorio entre 0 y 1 y pasa a la función toFixed el número recibido, recortando los decimales. El retorno puede ser también 0 o 1.

2.2.1.3 Función auxComb:

Esta función recibe un string correspondiente a la unidad y retorna un número con el cual se genera el factor de conversión.

2.2.1.4 Funcion cargoUnidades:

Previamente se genera un vector con las combinaciones posibles de unidades, cada índice del vector vecComb tiene otro vector con dos unidades, que serán en el ejercicio, la unidad de del número propuesto y la unidad a convertir.

La función cargoUnidades recibe dos valores numéricos, el primero solo para cumplir la primer condición del while, el segundo para evaluar la relación entre unidades.

Retorna el par de unidades a emplearse en cada ejercicio.

2.2.1.5 Funcion ejercicio:

Esta función recibe los datos de cada ejercicio y retorna a la función de cada nivel de ejercicio el append para agregar en la página.

2.2.1.6 Funcion cargoEventos:

Carga los eventos necesarios para eliminar el valor ingresado con un click, alertar en caso de no ingresar resultado y llamar a la función verificoRes.

2.2.1.7 Funcion básicos:

Función para generar ejercicios básicos no retorna valor.

2.2.1.8 Funcion noBasicos:

Funcion para generar ejercicios nivel medio y avanzado recibe parámetros de las funciones de ejercicio avanzado y intermedio.

2.2.1.9 Funcion intermedios :

Pasa a la función no basicos los parámetros para generar el ejercicio.

2.2.1.10 Funcion avanzados:

Pasa a la función no básicos los parámetros para generar el ejercicio.

2.2.1.11 Funcion valorNoVacio:

Alerta si no se ingresa valor al perder el foco.

2.2.1.12 Funcion correcto:

Si el resultado es correcto retorna el after para cargar en la pagina el símbolo correcto.

2.2.1.13 funcion incorrecto:

Si el resultado es incorrecto retorna el after para cargar en la pagina el símbolo correspondiente.

2.2.1.14 Funcion verificoRes:

Extrae los valores ingresados en la pagina, calcula el resultado, lo compara teniendo en cuenta el error de punto flotante y llama a las funciones correcto o incorrecto, en caso de incorrecto llama a la función mostrarAyuda.

2.2.1.15 Funcion eliminarTxt:

Vacia la caja de texto al hacer click sobre ella.

2.2.1.16 Funcion mostrarAyuda:

Carga en la pagina ayuda en caso de resultado incorrecto.

2.2.2 Codigo javascript:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
$(document).ready(iniEventos); //inicializo jQuery

/* asigno evento click para los botones basico,
 * medio y avanzados */

function iniEventos(){
    $('#b1').click(basicos);
    $('#b2').click(intermedios);
    $('#b3').click(avanzados);
}
```

```

/* declaro un vector "unidades" con las unidades a utilizar, luego genero las
 * combinaciones posibles cargandolas por pares en otro vector "vecComb"
 * en vecComb cada indice es otro vector con un par de unidades */

```

```

var unidades=['mm','cm','dm','m','dam','hm','km'];
var vecComb=[];
var cantEjer=10;
for(var i=0;i<unidades.length;i++){
    for(var x=0;x<unidades.length;x++){
        parUni=[unidades[i],unidades[x]];
        vecComb[vecComb.length]=parUni;
    }
}

```

```

/* creo una funcion numAleatEnt, con parametros de entrada maximo y minimo,
 * retorna un entero entre max y min */

```

```

function numAleatEnt(max,min){
    var aleatEnt=Math.floor(Math.random()*((max+1)-min))+min;
    return aleatEnt;
}

```

```

/* creo una funcion numAleatDec que toma como parametro
 * un numero, el cual sera la cantidad de decimales que tenga el
 * numero retornado, tambien puede retornar 0 o 1 */

```

```

function numAleatDec(num){
    return Math.random().toFixed(num);
}

```

```

/* creo funcion auxConv, toma un string que es la unidad (sin superindice)
 * retorna el valor asignado a la unidad, con el cual se calcula el factor
 * de conversion*/

```

```

function auxConv(unidad){
    var mult=0;
    switch(unidad){
        case 'mm':
            mult=1;
            break;
        case 'cm':
            mult=100;
            break;
        case 'dm':
            mult=10000;
            break;
        case 'm':
            mult=1000000;
            break;
        case 'dam':
            mult=100000000;
            break;
    }
}

```

```

    case 'hm' :
        mult=100000000000;
        break;
    case 'km':
        mult=10000000000000;
        break;
}
return mult;
}

```

```

/* creo funcion cargoUnidades, recibe como parametros dos numeros
 * con los cuales se evalua la relacion entre las unidades
 * la funcion llama a la funcion numAleatEnt, le pasa como maximo
 * el largo del vector vecComb -1 y minimo 0.
 * se utiliza para extraer una unidad aleatoria del vecComb
 * retorna la unidad si la relacion es menor o igual a dto2
 * y si el par de unidades extraidas no se repite */

```

```

function cargoUnidades(dto1,dto2){
    var indComb=0;
    var unidades=[];
    var valorUni1=0;
    var valorUni2=0;
    var factor=0;
    factor=dto1;
    while(factor>dto2 || valorUni1==valorUni2){
        indComb=numAleatEnt((vecComb.length-1),0);
        unidades=vecComb[indComb];
        valorUni1=auxConv(unidades[0]);
        valorUni2=auxConv(unidades[1]);
        if(valorUni1>valorUni2){
            factor=(valorUni1/valorUni2);
        }else{
            factor=valorUni2/valorUni1;
        }
    }
    return unidades;
}

```

```

/* funcion que recibe un indice, el numero propuesto de cada ejerc. y las unidades
 * retorna el append para cargar el ejercicio a las funciones de ejercicios basico o no basico */

```

```

function ejercicio(nl,unil,uniO,ind){
    return $('#ejercicios').append('<div id="dv'+ind+'"> <spam id="spn'+ind+'">'+nl+'</spam> <spam id="spui'+ind+'">'+unil+'</spam><sup>2</sup> son <input id="tx'+ind+'"' class="valor" type="text" value="Ingrese resultado"/> <spam id="spuo'+ind+'">'+uniO+'</spam><sup>2</sup> <input id="bt'+ind+'"' class="calcular" value="Calcular" type="button"/></div>');
}

```

```

/* funcion llamada por las funciones de ejercicios para inicializar eventos
* eliminar texto de las cajas de texto, alertar si no se carga valor y llamar
* a la funcion verificoRes para el calculo y verificacion de resultado*/

```

```

function cargoEventos(){
    $('#valor').click(eliminarTxt);
    $('#valor').blur(valorNoVacio);
    $('#calcular').click(verificoRes);
}

```

```

/* funcion de ejercicios basicos, llama a la funcion cargoUnidades
* pasando como parametro la relacion aceptada max entre unidades y
* un valor por encima para la primer condicion
* llama a las fuciones de numeros aleatorios para definir el numero
* a proponer. Llama a la funcion ejercicio y recibe el append y por ultimo
* carga los eventos de la funcion cargoEventos */

```

```

function basicos(){
    $('#ejercicios').empty();
    var unidad=cargoUnidades(10001,10000);
    var cantDec=0;
    var numDec=0;
    var numEnt=0;
    numIn=0;
    for(var i=0;i<cantEjer;i++){
        cantDec=numAleatEnt(1,0);
        numDec=parseFloat(numAleatDec(cantDec));
        numEnt=parseFloat(numAleatEnt(99,0));
        numIn=numEnt+numDec;
        ejercicio(numIn,unidad[0],unidad[1],i);
    }
    cargoEventos();
}

```

```

/* funcion de ejercicios no basicos recibe parametros de las
* funciones de ejercicios medio y avanzado
* evalua que las unidades recibidas no se repitan
* llama a la funcion ejercicio y carga eventos */

```



```

function noBasico(vl1,vl2,vl3,vl4,vl5,vl6){
    $('#ejercicios').empty();
    var auxV='si';
    var auxUni=[];

    for(var i=0;i<cantEjer;i++){
        unidad=cargoUnidades(vl1,vl2);
        for(x=0;x<auxUni.length+1;x++){
            if(unidad==auxUni[x]){
                auxV='no'
            }
        }
        if(auxV=='si'){
            auxUni[auxUni.length]=unidad;
            cantDec=numAleatEnt(vl5,vl6);
            numDec=parseFloat(numAleatDec(cantDec));
            numEnt=parseFloat(numAleatEnt(vl3,vl4));
            numIn=numEnt+numDec;
            ejercicio(numIn,unidad[0],unidad[1],i);
        }else{i=i-1;
            auxV='si';
        }
    }

    cargoEventos();
}

```

```

/* funcion de ejercicios nivel medio
* pasa parametros a la funcion noBasicos */

```

```

function intermedios(){
    noBasico(1000001,1000000,99,0,2,0);
}

```

```

/* funcion ejercicios nivel avanzado, idem anterior*/

```

```

function avanzados(){
    noBasico(100000000001,10000000000,999,100,3,2);
}

```

```

/* funcion con la que verifico no vacio y alerto */

```

```

function valorNoVacio(){
    if($('#this').val()==''){
        alert('¡No ingresó el resultado!');
    }
}

```

```

/* funcion que retorna el after para resultado correcto */

```

```

function correcto(dato){
    return $('#bt'+dato).after('<label id="lb'+dato+'" class="bien">iCorrecto!</label>');
}

/* funcion que retorna el after para incorrecto */

function incorrecto(dato){
    return $('#bt'+dato).after('<label id="lb'+dato+'" class="mal">iIncorrecto! <br><br><p
id="p'+dato+'"> Para obtener ayuda haga <a style="color: blue" title="Entrar a la ayuda"><u><b> click
aqui </b></u></a></p><br>');
}

/* funcion que extrae unidades y numero propuesto
* calcula resultado evalua teniendo en cuenta el error
* de punto flotante y llama a las funciones correcto o incorrecto
* si es incorrecto llama a mostrar ayuda
* sino se ingresa resulta o es isNaN alerta */

function verificoRes(){
    var idt=$(this).attr('id').charAt(2);
    $('#lb'+idt).remove();
    uniIn=$('#spui'+idt).text();
    uniOut=$('#spuo'+idt).text();
    var numPro=parseFloat($('#spn'+idt).text());
    var vaUni1=auxConv(uniIn);
    var vaUni2=auxConv(uniOut);
    factConv=vaUni1/vaUni2;
    var result=numPro*factConv;
    var numIng=$('#'+ 'tx'+idt).val();
    if(!isNaN(numIng)){
        numIng=parseFloat(numIng);
        if((1/numIng)*result<1.0000000001&&(1/numIng)*result>0.9999999999){
            correcto(idt);
        }else{
            incorrecto(idt);
            $('#p'+idt).click(mostrarAyuda);
        }
    }else{
        alert('¡No ingresó el resultado!');
    }
}

/* elimino valor ingresado */

function eliminarTxt(){
    $(this).attr('value','');
}

```

```
/* muestro ayuda si valor ingresado es incorrecto */
```

```
function mostrarAyuda(){
  $(this).empty();
  var negrita='<b style="color: black">';
  var supIndice='<sup style="font-size: 9pt">2</sup></b>';
  var uniInicial;
  var uniFinal;
  var factor;
  uniInicial=uniIn
  uniFinal=uniOut
  factor=factConv

  $(this).after('<div><h3 style="color: blue;"><u>Conversión de unidades de superficie: </u></h3>\n\
    <p id="recuadroResaltado" style="background-color: #F8F8FF; width: 35%;color: blue;font-
size: 14pt;">\n\
      Debes convertir el número propuesto de: '+negrita+' '+uniInicial+' '+supIndice+' \n\
      a '+negrita+' '+uniFinal+' '+supIndice+' <br><br> Recuerda que: <br>\n\
      1 '+negrita+' '+uniInicial+' '+supIndice+' es igual a: '+factor+' '+negrita+'
'+uniFinal+' '+supIndice+' \n\
      <br><br>Debes multiplicar el número propuesto por: '+factor+'<br></p></div>');
}
```

3 Capturas de prueba:

3.1 FireFox

Conversor de superficies - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

Conversor de superficies +

file:///E:/ORT/ATI/S1/P1/cd/aplicacion/conversor.html

Realizar las siguientes conversiones

Utilice el punto como separador de decimales

34 m ²	son	3400	dm ²	Calcular	✓ ¡Correcto!
58.9 m ²	son	5890	dm ²	Calcular	✓ ¡Correcto!
47.8 m ²	son	4780	dm ²	Calcular	✓ ¡Correcto!
92 m ²	son	9200	dm ²	Calcular	✓ ¡Correcto!
73.9 m ²	son	7390	dm ²	Calcular	✓ ¡Correcto!
33.8 m ²	son	3380	dm ²	Calcular	✓ ¡Correcto!
31.4 m ²	son	3140	dm ²	Calcular	✓ ¡Correcto!

3.2 Opera

Opera

Conversor de superficies x

Conversor de superficies x

Local localhost/E:/ORT/ATI/SL/P1/testing/aplicacion/conversor.htm

Realizar las siguientes conversiones

Utilice el punto como separador de decimales

111.6 hm² son dm² ¡Correcto!

150.1 dam² son hm² ¡Correcto!

885.86 dm² son m² ¡Correcto!

720.5 m² son hm² ¡Correcto!

133.38 dam² son cm² ¡Correcto!

902.1 m² son dam² ¡Correcto!

682.9 dm² son cm² ¡Correcto!

198.4 cm² son mm² ¡Correcto!

924.52 dam² son dm² ¡Correcto!

3.3 Chrome

Conversor de superficies

file:///E:/ORT/ATI/S1/P1/cd/aplicacion/conversor.html

Realizar las siguientes conversiones

Utilice el punto como separador de decimales

267.204 hm ²	son	<input type="text" value="2.67204"/>	km ²		Calcular	✓ ¡Correcto!
840.88 mm ²	son	<input type="text" value="0.000000084088"/>	hm ²		Calcular	✓ ¡Correcto!
472.29 mm ²	son	<input type="text" value="4.7229"/>	cm ²		Calcular	✓ ¡Correcto!
236.97 dam ²	son	<input type="text" value="23697"/>	m ²		Calcular	✓ ¡Correcto!
128.88 m ²	son	<input type="text" value="0.012888"/>	hm ²		Calcular	✓ ¡Correcto!
213.35 m ²	son	<input type="text" value="21835"/>	dam ²		Calcular	✓ ¡Correcto!