

# SISTEMAS NUMÉRICOS: Introducción a la Informática

JUAN DIEGO VELEZ SANCHEZ  
OCTUBRE DE 2020



# 1 CONTENIDO

---

1	CONTENIDO .....	1
2	PRESENTACIÓN .....	2
3	CONVERSIÓN BINARIA A DECIMAL .....	3
4	CONVERSIÓN BASADA EN DIVISIONES SUCEASIVAS .....	7
5	CONVERSIÓN EXTENDIDA.....	10
6	CONVERTIR (0) Y (1) EN LETRAS .....	13
7	CONTADOR DE CARACTERES “1” .....	15
8	CONTADOR DE ÍNDICES DE LA CADENA .....	17
9	CONVERTIR A CUALQUIER BASE .....	19
10	CONCLUSIONES.....	22
11	BIBLIOGRAFÍA.....	23



## 2 PRESENTACIÓN

---

La presente monografía describe la implementación de un conjunto de programas que le dan soporte a la teoría numérica básica de la materia INTRODUCCIÓN A LA INFORMÁTICA.

En los siguientes párrafos se presenta una descripción básica del significado de lo que es un sistema numérico, especialmente el sistema en base 2.

Un sistema numérico es un conjunto de símbolos y reglas, las cuales nos permiten representar números cuyos símbolos son definidos por la base, ya que esta es la cantidad de símbolos que el sistema puede utilizar, también es la que calcula o determina el símbolo que le pertenece a cada número, algunos sistemas numéricos son:

- Decimal
- Binario
- Hexadecimal

**Decimal:** El sistema decimal es la representación de datos numéricos, el cual sus símbolos a utilizar son del 0 al 9, es el sistema que utilizamos a diario, el cual tiene como base 10, ya que se utilizan 10 signos para su representación.

**Binario:** Es el sistema cuya base es dos (2), lo cual el sistema solo permite representar datos numéricos cuya cantidad máxima de símbolos a utilizar son dos. En este caso los símbolos son el cero (0) y uno (1). Todos los números decimales se representan a partir de combinaciones únicas de números binarios, se utiliza en las computadoras ya que trabajan solamente con dos niveles de voltaje.

Los primeros cuatro números binarios se representan:

0 = 00          1 = 01          2 = 10          3 = 11

**Hexadecimal:** Es el sistema que se representan con 16 símbolos diferentes, este incluye tanto símbolos numéricos como letras para su representación, sus símbolos son del 0 al 9, y cuando el valor de estos valen más de 9 se representa con letras de la A hasta la F.

**AUTOR: JUAN DIEGO VELEZ SANCHEZ**

**1004965639**

**juandiego.velez@utp.edu.co**

**<https://github.com/JuanDiegoVelez>**

### 3 CONVERSIÓN BINARIA A DECIMAL

Como dijimos anteriormente es el sistema cuya base es 2, cada cifra binaria tiene un valor el cual depende de su posición o comúnmente llamado índice, la posición da la cifra es contada de derecha a izquierda y los valores que se utilizan para hallar el valor de la cifra es la base elevada su posición, hay que tener en cuenta que la posición comienza a contarse desde cero, no de uno.

**var cadena = "1010";**

1	0	1	0	<del>0</del>
0	1	2	3	
3	2	1	0	

Esto nos lleva a entender que cada posición vale el doble a la posición anterior ejemplo:

Número Binario	0	0	1	0	0	0	0	1
Potencia de la Base	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Equivale a:	128	64	32	16	8	4	2	1
$0.128 + 0.64 + 1.32 + 0.16 + 0.8 + 0.4 + 0.2 + 1.1$								
<b>Es igual a 33</b>								

En la anterior imagen podemos ver que el número tiene una cantidad de cifras que es 8 pero el valor máximo de las posiciones es de 7, para calcular el número a decimal debemos multiplicar la cifra del binario con el resultado de la base elevada a su posición, para



posteriormente sumar el resultado, podemos deducir que los resultados que sumamos son aquellos cuya cifra binaria es 1.

Primeros quince números binarios:

```
0 ==> 0000
1 ==> 0001
2 ==> 0010
3 ==> 0011
4 ==> 0100
5 ==> 0101
6 ==> 0110
7 ==> 0111
8 ==> 1000
9 ==> 1001
10 ==> 1010
11 ==> 1011
12 ==> 1100
13 ==> 1101
14 ==> 1110
15 ==> 1111
```

El código para convertir números binarios a decimales es el siguiente:

```
1  function calcular(){
2
3      /*
4       Primero debemos obtener una entrada de datos y almacenarlos, en
5       nuestro caso una cade que representa un número binario
6       */
7      var binario = prompt("Introduce un número binario sin espacios: ");
8
9      /* Obtenemos la longitud de la cadena la cantidad de posiciones en la cadena con length
10     se obtiene la cantidad de caracteres po si se pone 10 la cantidad es 2, pero la posición
11     es 1 ya que comenzamos a contar desde cero. */
12
13     var longitud = binario.length - 1;
14
15     var resultado = 0;
16     var error = false;
17 }
```

```

18  /* El bucle recorera toda la cadena aumentando i que es nuestro índice en uno en cada ciclo */
19
20  for(var i = 0; i < longitud + 1; i++){
21
22      if(binario.charAt(longitud - i) === "1"){
23
24          /* El metodo math.pow() nos permite elevar un numero a una potencia determinada, como
25             el valor de una cifra binaria es 2 elevado a su índice o posición, en nuestro caso el
26             contador de la posición es la variable i, este se suma al resultado */
27
28              resultado += Math.pow(2, i);
29
30
31          /* Para obligar al usuario que ponga un número binario, debemos saber si el caracter
32             es diferente de cero (0); o uno (1) */
33
34          if(binario.charAt(longitud - i) !== "1" && binario.charAt(longitud - i) !== "0" ){
35              error = true;
36          }
37      }

```

En el código anterior podemos observar como dentro de una función tenemos el código para pasar de binario a decimal.

Primero debemos obtener una entrada de datos, la cual la conseguimos con `prompt()`, e inicializamos las variables a utilizar, la variable `error` que es booleana es para saber si el conjunto de caracteres es solamente de ceros (0) y unos (1), esto para saber si la entrada que el usuario propuso corresponde a un número binario.

En el bucle que recorre todos los caracteres se evalúan cuáles son iguales a "1", para proceder a calcular su valor, esto con el índice de la variable (`i`) del bucle ya que al recorrer carácter por carácter esta se suma, para esto utilizamos el método `pow()` de la clase `Math`, que nos permite elevar un número, en este caso 2 elevado a la `i`, el resultado de la exponenciación se suma con el valor de la variable `resultado`, que además la misma variable almacena esta suma.

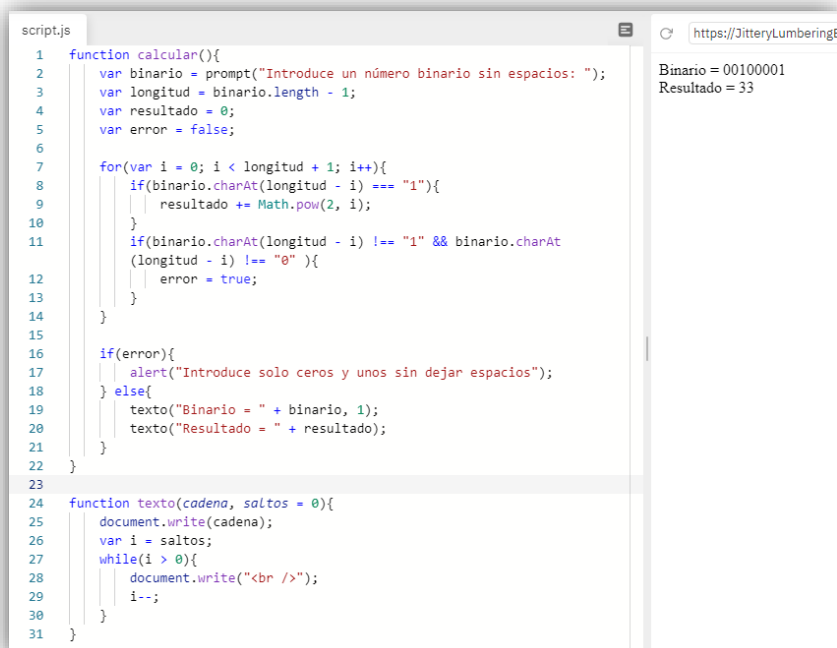
```

31      /* Para obligar al usuario que ponga un número binario, debemos saber si el caracter
32      es diferente de cero (0); o uno (1) */
33
34      if(binario.charAt(longitud - i) !== "1" && binario.charAt(longitud - i) !== "0" ){
35          error = true;
36      }
37  }
38
39  if(error){
40      alert("Introduce solo ceros y unos sin dejar espacios");
41  } else{
42      texto("Binario = " + binario, 1);
43      texto("Resultado = " + resultado);
44  }
45  }
46
47  // Se crea la funcion para escribir en el documento HTML
48
49  function texto(cadena, saltos = 0){
50      document.write(cadena);
51      var i = saltos;
52      while(i > 0){
53          document.write("<br />");
54          i--;
55      }
56  }

```

Si el usuario puso un carácter que no es un símbolo binario entonces procedemos a mostrar un mensaje, de lo contrario escribimos en el documento el binario y su conversión.

SALIDA o Resultado del anterior código:



```

script.js
1  function calcular(){
2      var binario = prompt("Introduce un número binario sin espacios: ");
3      var longitud = binario.length - 1;
4      var resultado = 0;
5      var error = false;
6
7      for(var i = 0; i < longitud + 1; i++){
8          if(binario.charAt(longitud - i) === "1"){
9              resultado += Math.pow(2, i);
10         }
11         if(binario.charAt(longitud - i) !== "1" && binario.charAt(
12             longitud - i) !== "0" ){
13             error = true;
14         }
15     }
16
17     if(error){
18         alert("Introduce solo ceros y unos sin dejar espacios");
19     } else{
20         texto("Binario = " + binario, 1);
21         texto("Resultado = " + resultado);
22     }
23 }
24
25 function texto(cadena, saltos = 0){
26     document.write(cadena);
27     var i = saltos;
28     while(i > 0){
29         document.write("<br />");
30         i--;
31     }

```

Binario = 00100001  
Resultado = 33

## 4 CONVERSIÓN BASADA EN DIVISIONES SUCESIVAS

A continuación se presenta el algoritmo básico para la conversión numérica basada en divisiones sucesivas.



Como se ve en el diagrama, la conversión se realiza dividiendo el número a convertir entre la base seleccionada.

El resultado se obtiene con base en los residuos de las divisiones.

El proceso finaliza cuando se obtiene cero en el resultado de las divisiones.

A continuación se presenta las imágenes de los códigos requeridos, para implementar el proceso mostrado en JavaScript. Cada imagen presenta una función distinta, o la ejecución final del programa.

```
function texto( cadena, num_saltos = 0 ) {
    document.write( cadena );
    var i = 0;
    while (i < num_saltos ) {
        document.write( "<br />" );
        i = i + 1;
    }
}
```



```
function conversion( numero, base ) {
  var division, resto;
  var result = "";
  var control = 0;
  var bandera = 0;
  while ( bandera == 0 ) {
    ///////////////////////////////////
    division = Math.trunc( numero / base );
    resto = numero - division * base;
    result = resto.toString() + result;
    numero = division;
    if (numero <= 0) {
      bandera = 1;
    }
    control = control + 1;
    if ( control > 1000 ) {
      bandera = 1;
    }
  }
  return result;
}
```

```
texto( "PROGRAMA DE CONVERSIÓN NUMÉRICA", 1);
texto( "Octubre 13 de 2020");
texto( "", 2);

var n = 83; // El número a convertir
var b = 16; // La base de conversión

var resultado;
resultado = conversion( n, b );

texto( "Número: " + n, 1);
texto( "Base:   " + b, 1);
texto( "Resp:   " + resultado, 1);

</script>
```

A continuación se muestra el programa en el entorno repl.it, con los datos de ejecución del programa.



```
1 // FUNCIONES DE APOYO
2 function texto( cadena, nueva_linea = false )
3 {
4   document.write( cadena );
5   if ( nueva_linea == true )
6     document.write( "<br />" );
7 }
8
9 // PROGRAMA PRINCIPAL
10 function conversion ( numero, base = 2 ) {
11   var control = 0;
12   var division, resto;
13   var resultado = "";
14   var bandera = 0;
15   while (bandera == 0) {
16     division = Math.trunc( numero / base );
17     resto = numero - Math.trunc( numero /
```

En la función texto vemos que si no se pone nada en la cantidad de saltos este por defecto vale 0, se logra inicializando el argumento en cero, lo mismo ocurre en el segundo argumento de la función conversion() que si no se pone algún valor este por defecto será 2.

En la función conversión, tenemos dos variables control y bandera, estas nos permitirán controlar las iteraciones de los bucles, las cuales evitaban que el bucle se ejecute infinita veces.

Como debemos tener un número entero para poder ir descomponiéndolo en la división, se logra con el método trunc(), de la clase Math, los argumentos son el número a dividir y el divisor de esta, esto para eliminar los números decimales.

Hay que tener en cuenta que el resultado final no es un número, el resultado será una cadena que simboliza un número a una base dada, lo cual el resultado que nos arroja la expresión matemática en la variable resto no es sumado, sino que es convertido a cadena con el método toString(), y concatenado al valor del resultado, ya que esta variable es una cadena.

## 5 CONVERSIÓN EXTENDIDA

---

A continuación presentamos el programa de conversión extendida, la cual se encarga de dar tratamiento a los números en base 16.

Hasta el momento hemos hecho un programa que es capaz de convertir un número decimal a otro número con una determinada base, pero que pasa con los números convertidos a una base mayor de 10, si recordamos con una base de 10 tenemos 10 símbolos diferentes los cuales van del 0 a 9, pero los sistemas con una base mayor a 10, comienzan a representarse tanto con números como letras, si miramos el código de la sección anterior podemos ver que la variable resto almacena un resultado de una expresión matemática, el cual este será el símbolo a utilizar, para bases del 2 al 10 el resultado tendrá un valor entre 0 y 9. Si utilizamos bases mayores a 10, necesitamos más símbolos ya que el resultado de la variable resto puede ser mayor a 9, lo cual utilizamos letras para simbolizarlos (A = 10), (B = 11) y así sucesivamente hasta (F = 16).

El código JavaScript es el siguiente:

```
9
10 // PROGRAMA PRINCIPAL
11 function conversion ( numero, base = 2 ) {
12     var division, resto;
13     var resultado = "";
14     var parar = 0;
15
16     // Parar evitar que el bucle se ejecute infinitas veces
17
18     while (numero > 0 && parar < 1000) {
19
20         division = Math.trunc( numero / base );
21         resto = numero - Math.trunc( numero / base ) * base;
22     }
```



```
23     if (base > 9 && resto > 9 ) {
24         var aux = "";
25
26         /* Si el resultado en resto es mayor a 9, procedemos a saber su simbolo
27         ejemplo A es el simbolo de 10, y asi hasta 16 lo cual esta letra la
28         guardamos en una variable para concatenarla mas adellante. */
29
30         switch (resto) {
31             case 10:
32                 aux = "A";
33                 break;
34             case 11:
35                 aux = "B";
36                 break;
37             case 12:
38                 aux = "C";
39                 break;
40             case 13:
41                 aux = "D";
42                 break;
43             case 14:
44                 aux = "E";
45                 break;
46             case 15:
47                 aux = "F";
48                 break;
49             default:
50                 aux = "";
51                 break;
52         }
53     }
```

```
54     //Concatenamos el valor de aux que sera una variable con el resultado
55
56     resultado = aux + resultado + " ";
57 }
58
59 /* Si la base o el resultado de resto es menor a 10, debemos convertir
60 el numero a cadena y concatenarlo */
61
62 else {
63     resultado = resto.toString() + resultado + " ";
64 }
65
66 numero = division;
67
68 //Si el bucle se repite 1000 veces se procede a pararlo y mostrar un mensaje
69
70 if (parar > 1000) {
71     texto( "ERROR. Se superó el número de 1000 iteraciones", 1 );
72     bandera = 1;
73 }
74 }
75
76 return resultado;
77 }
78
79 // EJECUCIÓN DEL PROGRAMA
80 var n = 16545;
81 var b = 15;
82
83 var resp = conversión( n, b );
84
85 texto( "Número: " + n, 1);
86 texto( "Conversión a base: " + b, 1);
87 texto( "Resultado: " + resp, 2);
88 }
```



The screenshot shows a web-based JavaScript editor and REPL interface. The editor on the left contains a JavaScript script named 'script.js'. The script defines a 'texto' function for printing text and a 'conversion' function for converting a decimal number to a specified base. The 'conversion' function uses a while loop to repeatedly divide the number by the base, and a switch statement to map remainders to hexadecimal digits (A-F). The REPL on the right shows the output of the script: 'Número: 16545', 'Conversión a base: 16', 'Resultado: 40A1', and 'PROGRAMA FINALIZADO!'. The URL in the browser's address bar is 'https://JitteryLumberingEmulator.juandiegodieg90.repl.co'.

```
script.js
1 function texto( cadena, numero_saltos = 0 ) {
2   document.write( cadena );
3   var i = 0;
4   while ( i < numero_saltos ) {
5     document.write( "<br/> " );
6     i = i + 1;
7   }
8 }
9
10 // PROGRAMA PRINCIPAL
11 function conversion ( numero, base = 2 ) {
12   var division, resto;
13   var resultado = "";
14   var parar = 0;
15
16   while (numero > 0 && parar < 1000) {
17     division = Math.trunc( numero / base );
18     resto = numero - Math.trunc( numero / base ) * base;
19
20     if (base > 9 && resto > 9 ) {
21       var aux = "";
22       switch (resto) {
23         case 10:
24           aux = "A";
25           break;
26         case 11:
27           aux = "B";
28           break;
29         case 12:
30           aux = "C";
31           break;
32         case 13:
33           aux = "D";
34           break;
```

Número: 16545  
Conversión a base: 16  
Resultado: 40A1  
PROGRAMA FINALIZADO!

https://JitteryLumberingEmulator.juandiegodieg90.repl.co

En el programa anterior podemos ver que agregamos una condición, la cual evaluara si la base y el valor de la variable resto es mayor a nueve, esto para proceder a calcular su símbolo, ya que si el valor de resto es 10 se simbolizara con la letra A, así sucesivamente hasta 16 = F, esto lo logramos con la expresión switch(), que nos permite evaluar diferentes casos, por ejemplo en el caso que de qué resto tenga un valor de 10 una variable diferente almacenara una letra A que luego será concatenada con el resultado, lo mismo ocurre si resto vale 11 la variable diferente toma como valor una B para ser concatenada.

En el ejemplo podemos ver que el número a convertir es 16545 en base 16, lo cual el resultado es 40A1, lo cual la letra A esta simbolizando al número 10.



## 6 CONVERTIR (0) Y (1) EN LETRAS

---

Este programa no permitirá convertir números (cifras) a letras ejemplo:

10 = uno cero

El código es el siguiente:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>taller 1</title>
5    </head>
6    <body>
7      <script>
8        // PROGRAMA - 01
9        // -----
10       // Programa que permite leer en una cadena de texto, un número binario
11       // Los ceros y los unos contenidos en la cadena de texto
12       // Son convertidos a las palabras: Cero y Uno, respectivamente
13
14       var cadena = prompt("Introduzca un número binario");
15       var salida = "";
16
17       var i = 0;
18       while (i < cadena.length) {
19         if (cadena.charAt(i) === "0") {
20           salida = salida + "Cero ";
21         }
22         if (cadena.charAt(i) === "1") {
23           salida = salida + "Uno ";
24         }
25
26         i = i + 1;
27       }
28
29       document.write("Cadena = " + cadena + "<br><br>");
30       document.write("Salida = " + salida + "<br><br>");
31       document.write("FINALIZADO!");
32     </script>
33   </body>
34 </html>
35
```

El programa anterior al ser ejecutado mostrara un mensaje donde el usuario puede introducir un numero binario para que luego muestre en la pantalla su conversión para a letras en el ejemplo usaremos el numero binario 1101.



...página insertada en classroom--juandiegodieg90.repl.co dice

Introduzca un número binario

```
index.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5    <meta name="viewport" content="width=device-width">
6    <title>repl.it</title>
7    <link href="style.css" rel="stylesheet" type="text/css" />
8  </head>
9  <body>
10   <script>
11     var cadena = prompt("Introduzca un número binario");
12     var salida = "";
13
14     var i = 0;
15     while ( i < cadena.length ) {
16
17       if (cadena.charAt(i) === "0") {
18         salida = salida + "Cero ";
19       }
20       if (cadena.charAt(i) === "1") {
21         salida = salida + "Uno ";
22       }
23
24       i = i + 1;
25     }
26
27     document.write("Cadena = " + cadena + "<br><br>");
28     document.write("Salida = " + salida + "<br><br>");
29     document.write("FINALIZADO!");
30   </script>
31 </body>
32 </html>
```

https://classroom.juandiegodieg90.repl.co

Cadena = 1101

Salida = Uno Uno Cero Uno

FINALIZADO!

El programa anterior funciona gracias a un bucle, este es el encargado de recorrer carácter por carácter de la cadena introducida por el usuario gracias al método `prompt()`, sabemos la cantidad de caracteres de la cadena gracias a la instrucción `cadena.length`, además dentro de este bucle encontramos dos condicionales, las cuales evaluarán si el carácter es igual a "1" o igual a "0" por medio del método `charAt(índice)`, para proceder a concatenar la palabra "uno" o "cero" con el valor de la variable `salida`, líneas 18 y 21 en el anterior código. Por último cuando el bucle recorra toda la cadena imprimir el resultado en la pantalla.



## 7 CONTADOR DE CARACTERES “1”

---

El programa permitirá al usuario introducir una cadena, el cual se pueda decir cuántos caracteres iguales a 1 se encuentran en esta.

El código es el siguiente:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>taller 2</title>
5    </head>
6    <body>
7      <script>
8        // PROGRAMA - 02
9        // -----
10       // Programa que permite leer en una cadena de texto, un número binario
11       // El programa cuenta el número de unos
12
13       var cadena = prompt("Introduzca un número binario");
14
15       var contador = 0; // Aquí se cuenta el número de unos
16
17       var i = 0;
18       while ( i < cadena.length ) {
19
20         if (cadena.charAt(i) === "1") {
21           contador = contador + 1;
22         }
23
24         i = i + 1;
25       }
26
27       document.write("Cadena   = " + cadena + "<br><br>");
28       document.write("Contador (número de unos) = " + contador + "<br><br>");
29       document.write("FINALIZADO!");
30     </script>
31   </body>
32 </html>
```





```
index.html
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width">
6      <title>repl.it</title>
7      <link href="style.css" rel="stylesheet" type="text/css" />
8    </head>
9    <body>
10     <script>
11       var cadena = prompt("Introduzca un número binario");
12
13       var contador = 0; // Aquí se cuenta el número de unos
14
15       var i = 0;
16       while ( i < cadena.length ) {
17
18         if (cadena.charAt(i) === "1") {
19           contador = contador + 1;
20         }
21
22         i = i + 1;
23       }
24
25       document.write("Cadena = " + cadena + "<br><br>");
26       document.write("Contador (número de unos) = " + contador +
27         "<br><br>");
28       document.write("FINALIZADO!");
29     </script>
30   </body>
31 </html>
```

https://classroom.juandiegodieg90.repl.co

Cadena = 11011011

Contador (número de unos) = 6

FINALIZADO!

En el código anterior podemos ver que al usuario se le pide un número binario por medio del método `prompt()` para ser almacenado en una variable, tenemos un bucle que recorrerá toda la cadena, gracias al método `charAt()`, podemos evaluar los caracteres de esta y verificar cuales son iguales a "1" para poder aumentar el contador en uno, si no es así el contador no aumentara, el valor almacenado en la variable `contador` corresponde a la cantidad de unos (1) almacenados en la cadena, luego se imprimirá en la pantalla.

## 8 CONTADOR DE ÍNDICES DE LA CADENA

En este programa lograremos contar los índices o posiciones de caracteres en una cadena introducida por el usuario.

El código es el siguiente:

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>taller 3</title>
5    </head>
6    <body>
7      <script>
8        // PROGRAMA 03
9        // -----
10       // Programa que lee en una cadena de texto un número binario
11       // El programa crea un índice numérico de las posiciones de la cadena
12
13       var cadena = prompt("Introduzca un número binario");
14
15       document.write("LA CADENA: " + cadena + "<br><br>");
16
17       document.write("SE MUESTRA EL INDICE: <br><br>");
18
19       var indice = 0;
20       var i = 0;
21       while (i < cadena.length) {
22
23         // Se muestra el índice
24         document.write( indice + " ");
25
26         // El índice aumenta
27         indice = indice + 1;
28
29         i = i + 1;
30       }
31
32       document.write("<br><br>FINALIZADO");
33     </script>
34   </body>
35 </html>

```

En el anterior código podemos observar que por el método `prompt()` anteriormente visto para pedir datos por medio del teclado, se pide un número binario, utilizamos un bucle y dos variables, una de ellas para utilizarla en el bucle, nos permite recorrer la cantidad de caracteres de la cadena y la otra para mostrar el resultado en la línea 24, ya que esta corresponde al índice. Al final obtenemos lo siguiente:

...página insertada en classroom--juandiegodieg90.repl.co dice

Introduzca un número binario

Aceptar
Cancelar

script.js

```

1 // PROGRAMA 03
2 // -----
3 // Programa que lee en una cadena de texto un número binario
4 // El programa crea un índice numérico de las posiciones de la cadena
5
6 var cadena = prompt("Introduzca un número binario");
7
8 document.write("LA CADENA: " + cadena + "<br><br>");
9
10 document.write("SE MUESTRA EL INDICE: <br><br>");
11
12 var indice = 0;
13 var i = 0;
14 while (i < cadena.length) {
15
16     // Se muestra el índice
17     document.write( indice + " ");
18
19     // El índice aumenta
20     indice = indice + 1;
21
22     i = i + 1;
23 }
24
25 document.write("<br><br>FINALIZADO");
26

```

https://classroom.juandiegodieg90.repl.co

LA CADENA: 100101

SE MUESTRA EL INDICE:

0 1 2 3 4 5

FINALIZADO

Podemos ver que en el anterior código introducimos una cadena de 6 caracteres, pero la el índice solo cuenta hasta 5, esto porque se debe comenzar a contar desde cero (0), además podemos ver en el código, que el índice esta inicializado en cero y aumenta en uno en cada iteración del bucle.

## 9 CONVERTIR A CUALQUIER BASE

En este programa podremos poner un número escrito en una base diferente y convertirlo a un número decimal, anteriormente en la unidad de conversión extendida, el programa nos permite convertir un número decimal a un número en determinada base, ejemplo convertir el 5454 en número hexadecimal nos dará 154E, pero que pasa si queremos calcular el número 154E escrito en hexadecimal a un número decimal, para eso debemos realizar el siguiente código:

```

8      // PROGRAMA 05
9      // -----
10     // Programa que lee en una cadena de texto un número
11     // El número está escrito en diferente base
12     // El resultado será un número decimal
13
14     var numero = prompt("Introduce un número: ").toUpperCase();
15     var base = parseInt(prompt("Escrito en base: "));
16     var resultado = 0;
17     var error = false;
18
19     for (var i = numero.length - 1; i >= 0; i--) {
20         var c = numero.charAt(i);
21         if (!isNaN(c)) {
22             resultado += parseInt(c) * Math.pow(base, numero.length - 1 - i);
23         } else if (base > 10 && base <= 16) {
24             var n = 0;
25
26             if (base > 10 && c == 'A') {
27                 n = 10;
28             } else if (base > 11 && c == 'B') {
29                 n = 11;
30             } else if (base > 12 && c == 'C') {
31                 n = 12;
32             } else if (base > 13 && c == 'D') {
33                 n = 13;
34             } else if (base > 14 && c == 'E') {
35                 n = 14;
36             } else if (base > 15 && c == 'F') {
37                 n = 15;
38             } else {
39                 error = true;
40             }
41
42             resultado += n * Math.pow(base, numero.length - 1 - i);
43         } else {
44             error = true;
45         }
46     }

```



```
48     if (error) {  
49         alert("Verifique (número invalido)");  
50     } else {  
51         alert("Resultado: " + resultado);  
52     }  
53 }
```

Primero que todo le pedimos al usuario un número que será introducido con el método `prompt()`, si notamos la variable que almacena este número no es convertida a número natural con el método `parseInt()`, ya que esta cadena puede tener letras. Como el programa entiende que la letra “a” y “A” son diferentes utilizamos el método `toUpperCase()`, que nos permite convertir todas las letras en mayúsculas, también pedimos la base que le pertenece al número escrito por el usuario, ya que no es lo mismo tener 1A escrito en base 16 que en base 13, además inicializamos una variable para guardar el resultado y otra para comprobar que el usuario ha introducido solo los caracteres disponibles, como los números del (0) al (9) y las letras desde (A) hasta (F), estos son los únicos símbolos o caracteres que puede introducir, si introduce un punto (.) o una letra que no está entre A y F generara un error.

En las condicionales debemos saber qué base es la que se utiliza, porque si recordamos la base 11 tiene símbolos desde el cero hasta el nueve pero además contiene la letra (A), lo cual si el usuario a introducido una letra desde B hasta F generara un error, ya que B no es un símbolo que le pertenece a la base 11, lo cual debemos evaluar las bases hasta la 16 ya que esta última es la que le pertenece todos los caracteres como número y letras, o sea:

Base 11 sus símbolos son: del (0) al (9) además de la letra (A).

Base 12 sus símbolos son: del (0) al (9) además de las letras A, B.

Base 13 sus símbolos son: del (0) al (9) además de la letra A, B, C.

Base 14 sus símbolos son: del (0) al (9) además de la letra A, B, C, D.

Base 15 sus símbolos son: del (0) al (9) además de la letra A, B, C, D, E.

Base 16 sus símbolos son: del (0) al (9) además de la letra A, B, C, D, E, F.

Como podemos ver si se introduce una F solo hay una base a la que le corresponde tal símbolo, por lo tanto si ese símbolo se introduce en una base inferior a 16 generara un error y si no se trata el resultado será diferente, por eso utilizamos las condicionales.



En el ejemplo vamos a utilizar el número 123ABC, recordemos que este número solo puede estar escrito en bases mayores a 12, si no es así generara un error porque el símbolo no existe, en este caso el número 123ABC está escrito en hexadecimal, el resultado será un numero decimal.

127.0.0.1:5500 dice

Introduce un número:

Aceptar Cancelar

127.0.0.1:5500 dice

Escrito en base:

Aceptar Cancelar

127.0.0.1:5500 dice

Resultado: 1194684

Aceptar

Como vemos en las anteriores imágenes no importa si ponemos mayúsculas con minúsculas ya que el método `toUpperCase()` se encarga de ponerlas en mayúsculas, recordemos que el numero decimal será diferente si se pone una base inferior y genera un error en bases menores a 13.

**Convertidor numérico realizado:**

<https://repl.it/@JuanDiegoDieg90/CONVERSION-NUMERICA-Web>



## 10 CONCLUSIONES

---

El desarrollo de las temáticas elaboradas en clase utilizando el lenguaje JavaScript prueba ser un mecanismo de gran valor para el aprendizaje de los conceptos básicos de la materia, aprendemos los diferentes sistemas numéricos, que embarcan desde el sistema binario hasta el sistema hexadecimal. Además de ampliar el conocimiento de manejar y modificar las cadenas de texto.

También nos brinda una nueva forma de pensar, un desarrollo en la lógica y de cómo solucionar un problema con lo aprendido hasta el momento.

Como vemos utilizamos lo esencial de cada programa variables, funciones, condiciones, bucles, métodos, etc.



## 11 BIBLIOGRAFÍA

---

<https://repl.it>

<https://es.slideshare.net/MariaVegasArraez/parte-2-conversiones-1>