

SISTEMAS NUMÉRICOS Y RECURSIVIDAD: Introducción a la Informática

<JUAN CAMILO RODRIGUEZ LIBREROS>
OCTUBRE DE 2020

1 CONTENIDO

1 CONTENIDO	1
2 PRESENTACIÓN.....	2
3 CONVERSIÓN BASADA EN DIVISIONES SUCESIVAS	3
4 CONVERSIÓN EXTENDIDA	6
5 SISTEMA BINARIO	8
6 COMPUERTAS LÓGICAS.....	9
7 RECURSIVIDAD.....	10
8 CONCLUSIONES	12
9 BIBLIOGRAFIA Y WEBGRAFIA	13

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.

AUTOR: JUAN CAMILO RODRIGUEZ LIBREROS

CÓDIGO: 1005894155

CORREO: juancamilo.rodriguez1@utp.edu.co

GITHUB LINK: <https://github.com/JuanCRL1>

3 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

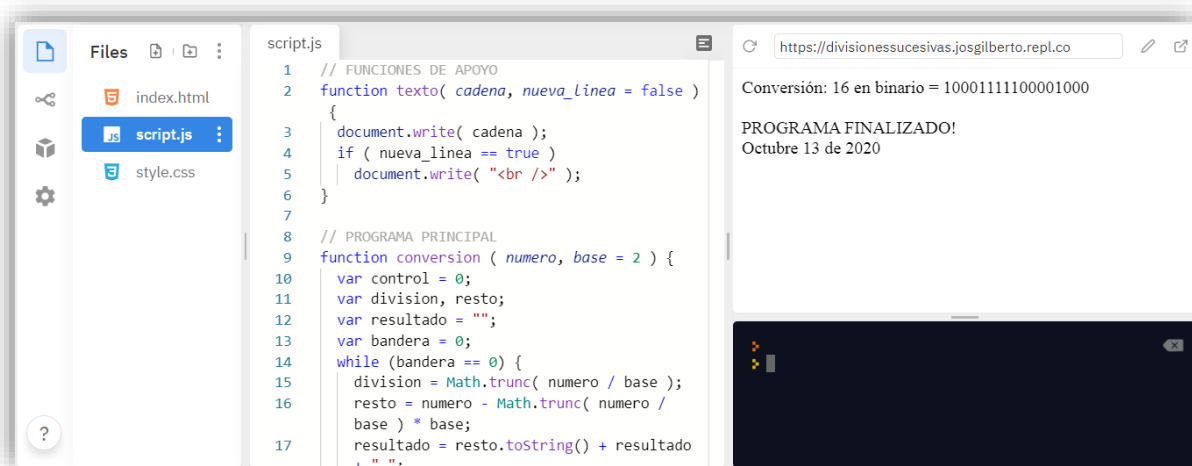
var resultado;
resultado = conversion( n, b );

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.



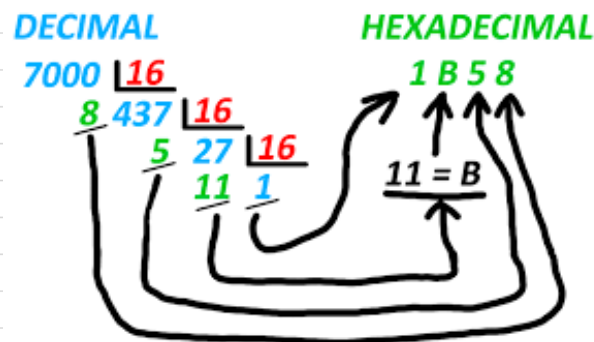
4 CONVERSIÓN EXTENDIDA

A continuación, se va a mostrar la conversión de base 10 a 16.

La base 16 o sistema hexadecimal es un sistema numérico que tiene como base el 16. Para convertir un número a base 16 se aplican las divisiones sucesivas, para convertirlo de base 16 a 10 se multiplica.

A continuación, se mostrará una tabla del sistema hexadecimal y un ejemplo de conversión con divisiones sucesivas.

Equivalencia	
decimal	hexadecimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F



Esta es una imagen del entorno repl.it para convertir un número de base 10 a 16:

```

if (base == 16 && resto > 9 ) {
  var aux = "";
  switch (resto) {
    case 10:
      aux = "A";
      break;
    case 11:
      aux = "B";
      break;
    case 12:
      aux = "C";
      break;
    case 13:
      aux = "D";
      break;
    case 14:
      aux = "E";
      break;
    case 15:
      aux = "F";
      break;
  }
  aux = aux + aux;
}

```

Número: 175
 Conversión a base: 16
 Resultado: AF

Nota: para convertir un numero de base 16 a base 10 se hace lo siguiente:

A	2	F	7
16^3	16^2	16^1	16^0
			7 x 16^0 = 7
		15 x 16^1 = 240	
	2 x 16^2 = 512		
10 x 16^3 = 40960			
<hr/>			
41719			

Lo mismo sucedería con el sistema binario:

Sistema Binario

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
Equivalente a:	128	64	32	16	8	4	2	1

$$0.128 + 0.64 + 1.32 + 0.16 + 0.8 + 0.4 + 0.2 + 1.1$$

Es igual a 33

5 SISTEMA BINARIO

Este es un sistema en el cual los números decimales (aparte de letras entre otras más cosas) se representan con solo dos cifras: el CERO (0) y el UNO (1), esto se debe a que el computador trabaja con dos niveles de voltaje y es representado de esa manera en términos de 0 y 1.

A continuación, se mostrarán algunas imágenes de programas en el entorno repl.it sobre este tema.

```

1 function sistema_binario() {
2   document.write("EL SISTEMA BINARIO <br/><br/>");
3
4   document.write("Está constituido por un modelo de
operaciones matemáticas sobre dos representaciones
numéricas únicas: El UNO y el CERO. El número UNO se
representa con el cardinal 1, mientras que el número CERO
se representa con el cardinal 0. Todos los números
decimales se representan a partir de combinaciones únicas
de números binarios <br/><br/>");
5
6   document.write("Los primeros cuatro número binarios se
representan así: <br/><br/>");
7
8   document.write("0 => 00 <br/>");
9   document.write("1 => 01 <br/>");
10  document.write("2 => 10 <br/>");
11  document.write("3 => 11 <br/>");
12 }

```

EL SISTEMA BINARIO

Está constituido por un modelo de operaciones matemáticas sobre dos representaciones numéricas únicas: El UNO y el CERO. El número UNO se representa con el cardinal 1, mientras que el número CERO se representa con el cardinal 0. Todos los números decimales se representan a partir de combinaciones únicas de números binarios

Los primeros cuatro número binarios se representan así:

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

```

1 function binarios_con_funcion() {
2   texto("NÚMEROS BINARIOS DEL 0 AL 15");
3   texto("");
4   texto("0 ==> 0000");
5   texto("1 ==> 0001");
6   texto("2 ==> 0010");
7   texto("3 ==> 0011");
8   texto("4 ==> 0100");
9   texto("5 ==> 0101");
10  texto("6 ==> 0110");
11  texto("7 ==> 0111");
12  texto("8 ==> 1000");
13  texto("9 ==> 1001");
14  texto("10 => 1010");
15  texto("11 => 1011");
16  texto("12 => 1100");
17  texto("13 => 1101");
18  texto("14 => 1110");
19  texto("15 => 1111");

```

NÚMEROS BINARIOS DEL 0 AL 15

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

6 COMPUERTAS LÓGICAS

Son dispositivos electrónicos con funciones booleanas (verdadero, falso) o de suma y resta.

Las 3 principales compuertas son: la compuerta AND, compuerta OR, compuerta NOT.

La compuerta and tendrá una salida alta cuando los valores de sus entradas sean altos (1), si una de las dos no es alta tendrá una salida baja (0).

La compuerta or tendrá una salida alta si una sola de las 2 entradas es alta, si las dos entradas son bajas tendrá una salida baja (0).

La compuerta not o inversor tiene una sola entrada cuando esta está baja la vuelve alta, si esta alta la vuelve baja de ahí su nombre inversor.

A continuación, se mostrarán unos gráficos descriptivos sobre este tema.

```

1 function and (a,b){
2   if (a===0 && b===0)
3     return 0;
4   if (a===0 && b==1)
5     return 0;
6   if (a==1 && b===0)
7     return 0;
8   if (a==1 && b==1)
9     return 1;
10  }
11 function or (a,b){
12   if (a===0 && b===0)
13     return 0;

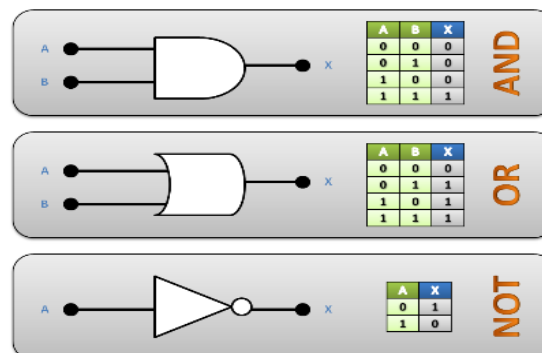
```

COMPUERTAS BINARIAS

AND entre 0 y 1= 0
 AND entre 1 y 1= 1
 OR entre 0 y 0= 0
 OR entre 1 y 0= 1
 NOT de 0= 1
 NOT de 1= 0



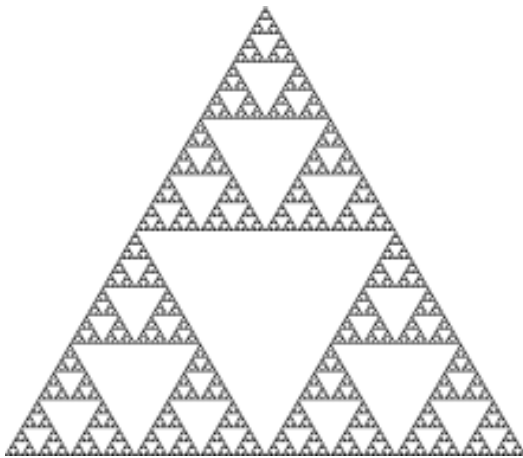
Compuertas AND, OR, NOT



7 RECURSIVIDAD

Recursión o recursividad es la forma en la cual se especifica un proceso basado en su propia definición.

Para entender esto se presentarán algunas imágenes recursivas:



A continuación, se presentarán unas imágenes del entorno repl.it (código y ejecución):

```
function numeros ( a, b ) {
  if ( a > b ) {
    return;
  }
  else {
    texto ( a );
    numeros ( a + 1, b );
  }
}
```

```
numeros ( 1, 5 );
```

```
function multiplicar ( a, b ) {
  if ( a == 0 ) {
    return 0;
  }
  else {
    return b + multiplicar ( a - 1, b );
  }
}
```

```
var x = 35;
```

```
var y = 7;
```

```
respuesta = multiplicar ( x, y );
```

RECURSIVIDAD

1

2

3

4

5

MULTIPLICAR RECURSIVO

$35 * 7 = 245$

8 CONCLUSIONES

Al hacer programas en el entorno HTML, CSS y JavaScript para cada uno de los temas se facilita la comprensión de estos.

9 BIBLIOGRAFIA Y WEBGRAFIA

<https://repl.it>

<https://repl.it/@JuanCRL1>

https://es.wikipedia.org/wiki/Puerta_l%C3%B3gica

https://es.wikipedia.org/wiki/Compuerta_AND

https://es.wikipedia.org/wiki/Puerta_OR

https://es.wikipedia.org/wiki/Puerta_NOT

https://es.wikipedia.org/wiki/Recurso_l%C3%B3gico