

# SISTEMAS NUMÉRICOS: Introducción a la Informática

GIAN FRANCO POSSO GIRALDO/ DANIEL ENRIQUE VILLA ARIAS  
OCTUBRE DE 2020



# 1 CONTENIDO

---

1	CONTENIDO .....	1
2	PRESENTACIÓN .....	2
3	TALLER NÚMEROS BINARIOS .....	3
4	CONVERSIÓN BASADA EN DIVISIONES SUCESIVAS .....	8
5	CONVERSIÓN EXTENDIDA.....	11
6	RECURSIVIDAD I .....	15
7	RECURSIVIDAD II .....	19
8	CONCLUSIONES.....	25
9	BIBLIOGRAFÍA.....	26



## 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.

AUTORES: Gian Franco Posso Giraldo/ Daniel Enrique Villa Arias

CÓDIGO: 100468162/ 1010003883

CORREO: f.posso@utp.edu.co/ daniel.villla2@upt.edu.co

GITHUB:



### 3 TALLER NÚMEROS BINARIOS

---

En este punto se realizan trabajos enfocados al sistema binario.

Los números binarios es un sistema numérico basado en dos dígitos, el cero [0] y el uno [1], diferente al sistema que usamos en nuestro día a día que es el sistema decimal el cuál está conformado por diez dígitos (del cero [0] al nueve [9]).

Decimal	Binario
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

A continuación, se presentarán cuatro códigos con diferentes planteamientos sobre el uso del sistema binario.

El primer código transforma el número binario ingresado a texto, es decir si se ingresara el “1001” mostraría en pantalla “Uno Cero Cero Uno”.

```
<script>

// PROGRAMA - 01
// -----
// Programa que permite leer en una cadena de texto, un número binario
// Los ceros y los unos contenidos en la cadena de texto
// Son convertidos a las palabras: Cero y Uno, respectivamente

var cadena = prompt("Introduzca un número binario");
var salida = "";

var i = 0;
while ( i < cadena.length ) {

    if (cadena.charAt(i) === "0") {
        salida = salida + "Cero ";
    }
    if (cadena.charAt(i) === "1") {
        salida = salida + "Uno ";
    }
}

i = i + 1;
}

document.write("Cadena = " + cadena + "<br><br>");
document.write("Salida = " + salida + "<br><br>");
document.write("FINALIZADO!");
</script>
```



Cuando se ejecuta este código salta un recuadro que pide un número binario cualquiera.

Después de que se ingrese el número binario quedará el siguiente diseño de pantalla.

```
Cadena = 100110  
Salida = Uno Cero Cero Uno Uno Cero  
FINALIZADO!
```

El segundo programa cuenta todos los uno [1] de un número binario.

```
<script>
// PROGRAMA - 02
// -----
// Programa que permite leer en una cadena de texto, un número binario
// El programa cuenta el número de unos

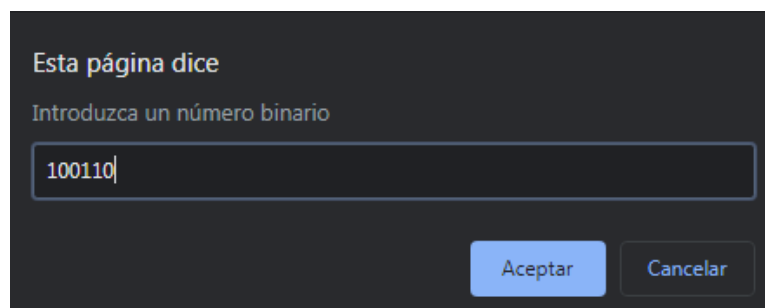
var cadena = prompt("Introduzca un número binario");

var contador = 0; // Aquí se cuenta el número de unos

var i = 0;
while ( i < cadena.length ) {
    if (cadena.charAt(i) === "1") {
        contador = contador + 1;
    }
    i = i + 1;
}

document.write("Cadena   = " + cadena + "<br><br>");
document.write("Contador (número de unos) = " + contador + "<br><br>");
document.write("FINALIZADO!");
</script>
```

Cuando se ejecuta este código salta un recuadro que pide un número binario cualquiera.



Después de que se ingrese el número binario quedará el siguiente diseño de pantalla.

```
Cadena = 100110

Contador (número de unos) = 3

FINALIZADO!
```

El tercer programa crea un índice numérico (que inicia en cero [0]) de cada una de las posiciones de la cadena.

```
<script>
// PROGRAMA 03
// -----
// Programa que lee en una cadena de texto un número binario
// El programa crea un índice numérico de las posiciones de la cadena

var cadena = prompt("Introduzca un número binario");

document.write("LA CADENA: " + cadena + "<br><br>");

document.write("SE MUESTRA EL INDICE: <br><br>");

var indice = 0;
var i = 0;
while (i < cadena.length) {

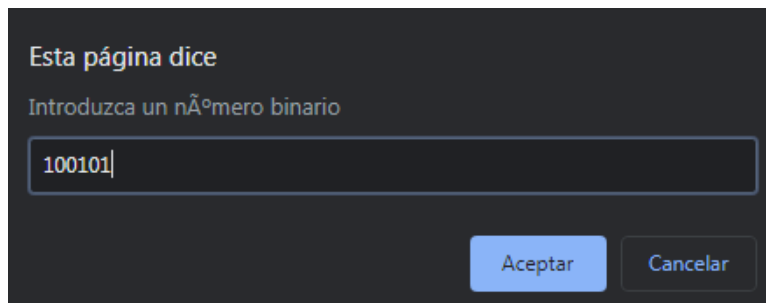
    // Se muestra el índice
    document.write( indice + " ");

    // El índice aumenta
    indice = indice + 1;

    i = i + 1;
}

document.write("<br><br>FINALIZADO");
</script>
```

Cuando se ejecuta este código salta un recuadro que pide un número binario cualquiera.



Después de que se ingrese el número binario quedará el siguiente diseño de pantalla.

```
LA CADENA: 100101

SE MUESTRA EL INDICE:

0 1 2 3 4 5

FINALIZADO
```

El cuarto programa convierte un número binario y lo pasa a su notación decimal.

```
<script>
// PROGRAMA 04
// -----
// Programa que lee en una cadena de texto un número binario
// El programa crea un ÍNDICE INVERSO del número binario
// Este índice inverso se corresponde con la estructura numérica posicional
// Se calcula el equivalente en base 10 del número binario

var cadena = prompt("Introduzca un número binario");

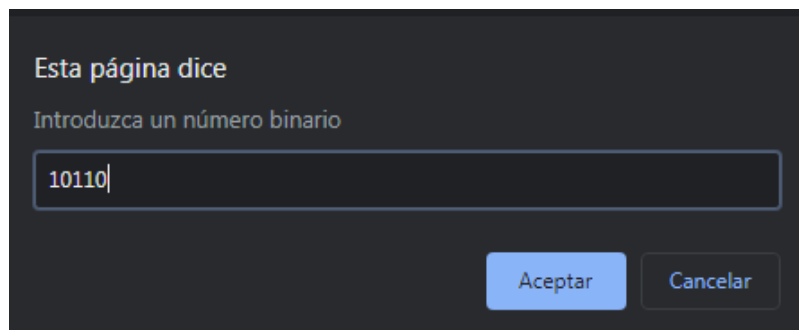
document.write("Número binario: " + cadena + "<br><br>");

var suma = 0;

var indice = cadena.length - 1;
var i = 0;
while (i < cadena.length) {
    suma = suma + parseInt(cadena.charAt(i)) * Math.pow( 2, indice );
    indice = indice - 1;
    i = i + 1;
}

document.write("Número equivalente en base 10 = " + suma);
</script>
```

Cuando se ejecuta este código salta un recuadro que pide un número binario cualquiera.



Después de que se ingrese el número binario quedará el siguiente diseño de pantalla.

Número binario: 10110  
Número equivalente en base 10 = 22



## 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. Se debe escribir en un solo archivo el código mostrado, y se sugiere un entorno como repl.it.

```
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.



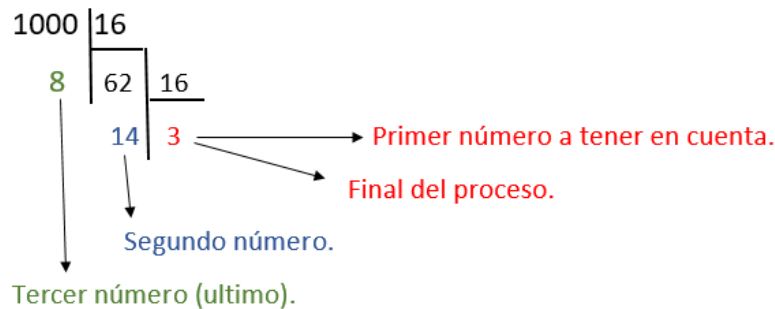
```
1 // FUNCIONES DE APOYO
2 function texto( cadena, nueva_linea = false )
3 {
4   document.write( cadena );
5   if ( nueva_linea == true )
6     document.write( "<br />" );
7 }
8
9 // PROGRAMA PRINCIPAL
10 function conversion ( numero, base = 2 ) {
11   var control = 0;
12   var division, resto;
13   var resultado = "";
14   var bandera = 0;
15   while (bandera == 0) {
16     division = Math.trunc( numero / base );
17     resto = numero - Math.trunc( numero /
18     base ) * base;
19     resultado = resto.toString() + resultado
20     + " ";
```

Conversion: 16 en binario = 10001111100001000

PROGRAMA FINALIZADO!  
Octubre 13 de 2020

<PRESENTAR UNA EXPLICACIÓN PERSONAL DE ALGUNOS PUNTOS QUE CONSIDERE INTERESANTES EN EL ALGORITMO>

## 5 CONVERSIÓN EXTENDIDA



Como podemos observar en la división, para poder pasar cualquier número a sistema hexadecimal lo dividimos entre 16, si es un número muy grande se requieren de varias divisiones, el proceso termina cuando el cociente ya sea menor que 16, como podemos observar en la división que está arriba, el ultimo cociente fue el número 3, ya que el 62 sigue siendo mayor a 16. El sistema hexadecimal consta de 16 caracteres, del 0 al 9 se escriben normal, del 10 al 15 se remplazan por las letras de la A hasta la F, el 16 sería el número 10 en sistema hexadecimal, por ejemplo, en el sistema decimal cada 10 números el primer número de izquierda a derecha cambia aumentando de a 1, en el 10 el primer número es 1, después de otros 10 números llegarían al 20 siendo el primer número 2, en el sistema hexadecimal cambian cada 16 números, el 16 seria 10, al sumar otros 16 llegaríamos al número 32 que se escribiría como 20, el 64 seria 30, así nos podemos guiar para encontrar los otros, por ejemplo si el 16 es 10, el 17 es 11, si el 32 es 20 el 33 sería el 21. Cuando son números más grandes se nos puede complicar por eso hicimos una división de ejemplo donde mostramos como pasar cualquier número a sistema hexadecimal, tomamos el número que sea y lo dividimos entre 16, nosotros usamos el número 1000, el primer cociente nos dio 62 y el residuo 8, como 62 es mayor que 16 seguimos con el proceso, al dividir 62 entre 16 nos dio cociente 3 y residuo 14, 3 es menor que 16 entonces terminamos el proceso, el ultimo cociente es el primer número que se tiene en cuenta y el único de los cocientes, y todos los residuos de derecha a izquierda entonces los números a tener en cuenta serian el ultimo cociente que es el 3 y los residuos 14 y 8, en ese orden, el 3 se escribe así normal en sistema hexadecimal el 14 sería la letra E, el 8 si queda igual entonces el número 1000 en sistema hexadecimal se escribiría 3E8.

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

```

1  <script type="text/javascript">// FUNCIONES DE APOYO
2  ▼ function texto( cadena, numero_saltos = 0 ) {
3      document.write( cadena );
4      var i = 0;
5  ▼  while ( i < numero_saltos ) {
6      document.write( "<br/> " );
7      i = i + 1;
8  }
9  }
10
11  // PROGRAMA PRINCIPAL
12  ▼ function conversion ( numero, base = 2 ) {
13      var control = 0;
14      var division, resto;
15      var resultado = "";
16      var bandera = 0;
17  ▼  while (bandera == 0) {
18      division = Math.trunc( numero / base );
19      resto = numero - Math.trunc( numero / base ) * base;
20
21  ▼      if (base == 16 && resto > 9 ) {
22          var aux = "";
23  ▼          switch (resto) {
24  ▼              case 10:
25                  aux = "A";
26                  break;
27  ▼              case 11:
28                  aux = "B";
29                  break;
30  ▼              case 12:
31                  aux = "C";
32                  break;
33  ▼              case 13:
34                  aux = "D";
35                  break;
36  ▼              case 14:
37                  aux = "E";
38                  break;
39  ▼              case 15:
40                  aux = "F";
41                  break;
42  ▼              default:

```

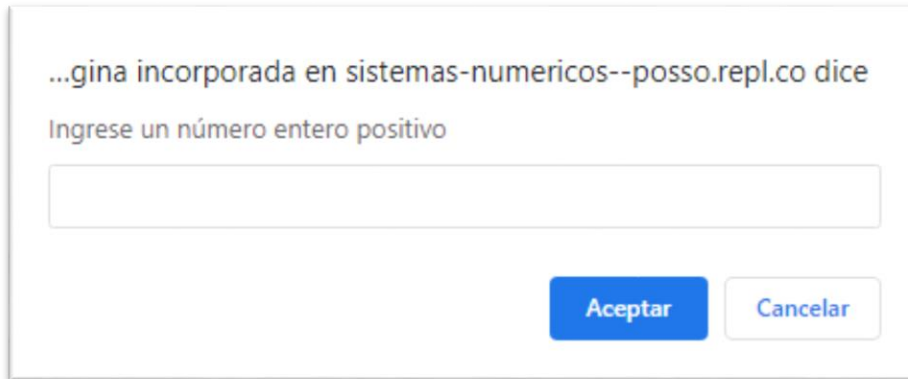
```

43         aux = "";
44         break;
45     }
46     resultado = aux + resultado + " ";
47 }
48 else {
49     resultado = resto.toString() + resultado + " ";
50 }
51
52 numero = division;
53
54 if ( division < 1 )
55     bandera = 1;
56
57 control = control + 1;
58 if (control > 1000) {
59     texto( "ERROR. Se superó el número de 1000 iteraciones", 1 );
60     bandera = 1;
61 }
62 }
63
64 return resultado;
65 }
66
67 // EJECUCIÓN DEL PROGRAMA
68 var n = prompt ("Ingrese un número entero positivo");
69 var b = 16;
70
71 var resp = conversion( n, b );
72
73 texto( "Número: " + n, 1);
74 texto( "Conversión a base: " + b, 1);
75 texto( "Resultado: " + resp, 2);
76
77 // PROGRAMA TERMINADO
78 texto("PROGRAMA FINALIZADO!", 1 );
79 texto("Octubre 13 de 2020");</script>

```

A continuación, se dará una breve explicación del funcionamiento del código.

Cuando se ejecuta el código, salta un cuadro emergente que le solicita al usuario que ingrese un número entero positivo.

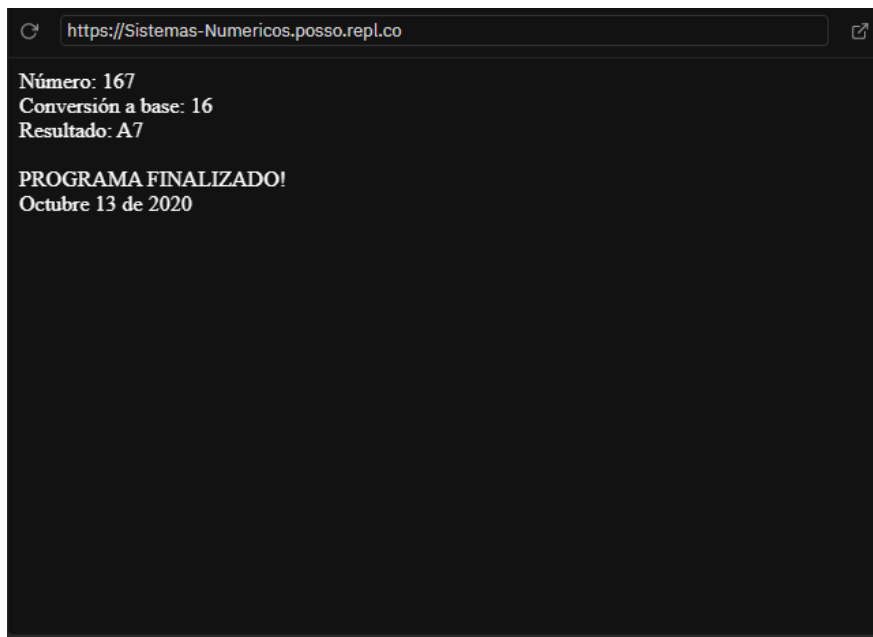


...gina incorporada en sistemas-numericos--posso.repl.co dice

Ingrese un número entero positivo

**Aceptar** Cancelar

Después de que el usuario ingrese el número, el programa realizará la conversión de dicho número a hexadecimal y lo mostrará en pantalla.



```
https://Sistemas-Numericos.posso.repl.co
Número: 167
Conversión a base: 16
Resultado: A7

PROGRAMA FINALIZADO!
Octubre 13 de 2020
```

## 6 RECURSIVIDAD I

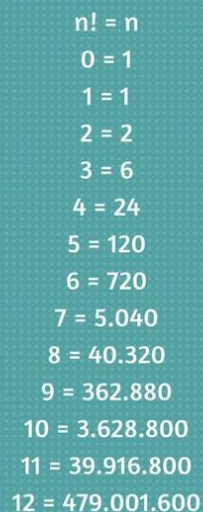
---

En este punto se da una introducción a la recursividad.

La recursividad es una técnica de programación que consiste en que una función se llame así misma en una especie de bucle hasta que se cumpla o se deje de cumplir una condición, esto depende de cómo se estructure.

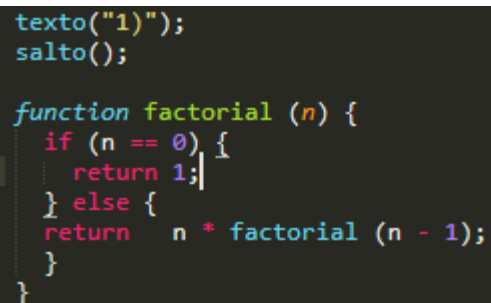
A continuación, se mostrarán tres ejemplos de cómo puede ser empleada la recursividad.

En el primer ejemplo se pide hallar el factorial de “x” número. Un número factorial consiste en multiplicar un número “n” por sus antecesores, por ejemplo, el cuatro [4] se multiplicaría por los números del cero [0] al cuatro [4].



n!	=	n
0	=	1
1	=	1
2	=	2
3	=	6
4	=	24
5	=	120
6	=	720
7	=	5.040
8	=	40.320
9	=	362.880
10	=	3.628.800
11	=	39.916.800
12	=	479.001.600

A continuación, se presenta el código que funciona para conseguir la factorial de un número.

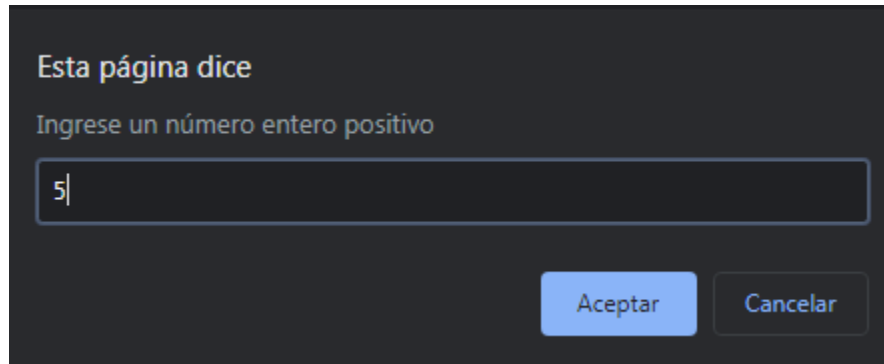


```
texto("1");
salto();

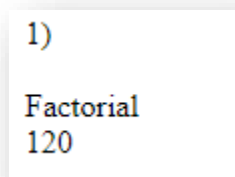
function factorial (n) {
  if (n == 0) {
    return 1;
  } else {
    return n * factorial (n - 1);
  }
}
```



Cuando se ejecuta el código, salta un cuadro emergente que le solicita al usuario que ingrese un número entero positivo



Después de que el usuario ingrese el número, se mostrará en pantalla el factorial de dicho número.



1)  
Factorial  
120

En el segundo ejemplo se pide hallar la sumatoria entre dos números (En este caso “a” y “n”). En este caso hay que sumar todos los números entre “a” y “n”, por ejemplo, uno [1] y cinco [5].

$$a=1 \quad n=5$$
$$1+2+3+4+5=15$$

A continuación, se presenta el código que funciona para conseguir la sumatoria entre dos números.

```
function sumatoria (a, n) {  
  if (n == 0 ) {  
    return 0;  
  } else {  
    return n + sumatoria (1, n - 1);  
  }  
}  
  
texto ("Sumatoria");  
res = sumatoria(1,5);  
texto (res.toString());  
salto();
```

Después de ejecutar el código se mostrará la sumatoria de dichos números.

2)

Sumatoria  
15

En el tercer ejemplo se pide calcular “x” elevado a la “y”.

$$x=4$$

$$= 4^5 = 4*4*4*4*4 = 1024$$

$$y=5$$

A continuación, se presenta el código que funciona para conseguir la potencia de “x” a la “y”.

```
function potencia (x, y, z) {
  if (y == 1) {
    return z;
  } else {
    return potencia (x , y - 1, x * z);
  }
}

texto (" Potencia");
res = potencia (2, 8, 2); //El valor de la variable "x" debe tener el mismo valor que la variable "z",
// es decir si "x" vale dos[2], "z" deberá de tener el mismo valor.
texto (res.toString());
salto()
```

Se utilizan tres variables “x” “y” “z”, la primera se utiliza para el número al que se le desea sacar potencia, la segunda para indicar el exponente de dicho número y la tercera se utiliza para almacenar cada ciclo hecho.

Cuando se ejecute el código se mostrará en pantalla la potencia de dicho número.

3)

Potencia

256

## 7 RECURSIVIDAD II

Se sigue realizando trabajos enfocados en la recursión. Se mostrarán los programas hechos sobre el manejo de la recursividad.

Se mostrará a continuación un programa que suma los elementos de una lista mediante la iteración.

```

1
2
3 <script type="text/javascript">
4 //Esta parte el código suma los elementos de una lista iterativamente.
5 function texto ( cadena ) {
6     document.write( cadena );
7 }
8
9 function salto ( numero ) {
10     for(let i = 0; i < numero; i ++ ) {
11         document.write("<br>")
12     }
13 }
14
15
16 const numeros = [0,1,2,3,4,5,6,7,8,9];
17
18 let acc = 0;
19 for(let i = 0; i < numeros.length; i ++){
20     acc += numeros[i];
21 }
22 document.write("Sumatoria de los elementos de una lista por iteración:<br>")
23 document.write("La lista es:<br>" + numeros)
24 document.write("<br>")
25 texto ("sumatoria elementos de la lista = " + acc );
26 document.write("<br>")
27 document.write("-----");

```

Cuando se ejecuta este programa quedará el siguiente diseño de pantalla.

```

Sumatoria de los elementos de una lista por iteración:
La lista es:
0,1,2,3,4,5,6,7,8,9
sumatoria elementos de la lista = 45

```



En el primer programa se pide sumar los elementos de una lista.

```
//Sumar los elementos de una lista por recursión
function sumatoria (sum){
  if(sum.length === 0){
    return 0;
  }
  else{
    const [first, ...rest] = sum;
    return first + sumatoria (rest);
  }
}
respuesta = sumatoria ([0,1,2,3,4,5,6,7,8,9]);
document.write("Sumatoria de una lista por recursión.<br>" + numeros);
document.write("<br>");
document.write ("Sumatoria de la lista =" + respuesta);
document.write("<br>");
document.write("-----");
document.write ("<br>");
```

Cuando se ejecuta el código, se sumarán los números que están adentro de la lista, que en este caso serían del cero [0] al nueve [9], dando el siguiente diseño de pantalla.

```
Sumatoria de una lista por recursión.
0,1,2,3,4,5,6,7,8,9
Sumatoria de la lista =45
```



En el segundo programa se pide mostrar los elementos de una lista, es decir que muestre en pantalla los elementos que pertenecen a “x” lista, en este caso sería los números del cero[0] al nueve[9].

```
//Mostrar elementos de una lista
document.write ("Mostrar elementos de una lista:<br>");
function mostrar (numeros){
  if (numeros.length === 0) {
    return;
  }
  else{
    const [first, ...rest] = numeros;
    document.write (first);
    mostrar (rest);
  }
}
lista = [0,1,2,3,4,5,6,7,8,9];
mostrar (lista);
document.write("<br>");
document.write("-----");
document.write("<br>");
```

Cuando se ejecuta el código se mostrará en pantalla los elementos pertenecientes a dicha lista.

Mostrar elementos de una lista:  
0123456789

En el tercer programa se pide encontrar los elementos de una lista, es decir que en una lista “x” busque un elemento el que queramos y nos lo muestre en pantalla, en caso de no encontrar ese elemento lo deberá decir.

```
//Buscar un elemento de una lista
function buscar (dato, lista) {
  if (lista.length === 0) {
    return false;
  }
  else {
    const [first, ...rest] = lista;
    if (first == dato)
      return true;
    else
      return buscar (dato, rest);
  }
}
lista = [0,1,2,3,4,5,6,7,8,9];
if ( buscar( 5, lista )) {
  document.write ("Encontrado el 5 en la lista<br>");
}

if ( buscar( 28, lista)) {
  document.write ("Encontrado el 28 en lista");
}
else{
  document.write ("El 28 no esta en la lista");
}
```

Cuando se ejecuta este código aparecerá en pantalla el elemento que buscamos en la lista, en este caso sería el número cinco [5], también buscará en la lista el número veintiocho[28], en caso de que no encuentre este número deberá mostrar en pantalla que este elemento no está en la lista.

```
Encontrado el 5 en la lista
El 28 no esta en la lista
```



En este nuevo código se crean tres funciones para encontrar el mayor, el menor y el promedio de una lista.

En esta parte del código esta la función para encontrar el mayor elemento de una lista con dos parámetros, se hace por medio de la recursión, si la "i" es mayor o igual a los elementos de la lista retorna "m", "i" toma los elementos de la lista, si es mayor que "m" se le suma uno [1] a "i" hasta que llegue al número mayor de la lista.

```
1 <script type="text/javascript">
2   var lista=[ 1, 30, 6, 5, 40, 2, 3];
3   document.write("Codigo que dice el mayor, el menor elemento de una lista y su promedio.<br>");
4   document.write("La lista es:<br>" + lista);
5   document.write("<br>");
6   document.write ("el mayor es : "+mayor(lista[0],1));
7   document.write("<br>")
8
9   //Funcion para encontrar el mayor
10  function mayor(m,i){
11
12      if (i>=lista.length) return m;
13
14      if (lista[i]>m) return mayor(lista[i],i+1);
15
16      else return mayor(m,i+1);
17  }
18
```

Esta parte del código se encuentra la función para encontrar el menor elemento de una lista, es casi igual a la función para encontrar el mayor, solo que en el segundo "if" se cambia el mayor que por el menor que.

```
18
19   document.write("El menor es:" + menor(lista[0],1));
20   document.write("<br>");
21
22
23   //funcion para encontrar el menor
24   function menor(e,a) {
25       if(a>=lista.length) return e;
26
27       if (lista[a]<e) return menor(lista[a],a+1);
28
29       else return menor(e,a+1)
30   }
```





Esta parte del código se encuentra la función para encontrar el menor elemento de una lista, es casi igual a la función para encontrar el mayor, solo que en el segundo if se cambia el “mayor que” por el “menor que”.

```
31 //codigo para encontrar el promedio
32 var suma = lista.reduce((previus, current) => current += previus);
33 var promedio = suma / lista.length;
34 document.write("El promedio de la lista es:<br>" + promedio);
35
36 </script>
```

Cuando se ejecuta este programa, se mostrará en pantalla el siguiente diseño.

```
Codigo que dice el mayor, el menor elemento de una lista y su promedio.
La lista es:
1,30,6,5,40,2,3
el mayor es : 40
El menor es:1
El promedio de la lista es:
12.428571428571429
```



## 8 CONCLUSIONES

---

La monografía nos muestra diferentes tipos de código y explicaciones para solucionar diferentes tipos de ejercicios, la mayoría con recursión y otros también con iteración es interesante que nos muestren dos formas para resolver estos ejercicios ya que en muchos ejercicios serán complejos y conociendo la lógica de dos diferentes formas se puede lograr llegar más fácil a la solución. También es mu útil porque al realizar aquellos ejercicios se adquiere mayor conocimiento para programar en el lenguaje de JavaScript y se conocieron algunas funciones muy útiles de la propia librería de JavaScript. 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.



## 9 BIBLIOGRAFÍA

---

<https://repl.it>