Gian Franco Posso Giraldo/ Daniel enrique villa arias

OCTUBRE DE 2020

SISTEMAS NUMÉRICOS: Introducción a la Informática

UTP | Pereira

2020

# CONTENIDO

[1 CONTENIDO 1](#_Toc56036108)

[2 PRESENTACIÓN 2](#_Toc56036109)

[3 CONVERSIÓN BASADA EN DIVISIONES SUCESIVAS 3](#_Toc56036110)

[4 CONVERSIÓN EXTENDIDA 6](#_Toc56036111)

[5 RECURSIVIDAD I 10](#_Toc56036112)

[6 RECURSIVIDAD II 14](#_Toc56036113)

[7 CONCLUSIONES 17](#_Toc56036114)

[8 BIBLIOGRAFÍA 18](#_Toc56036115)

# 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**

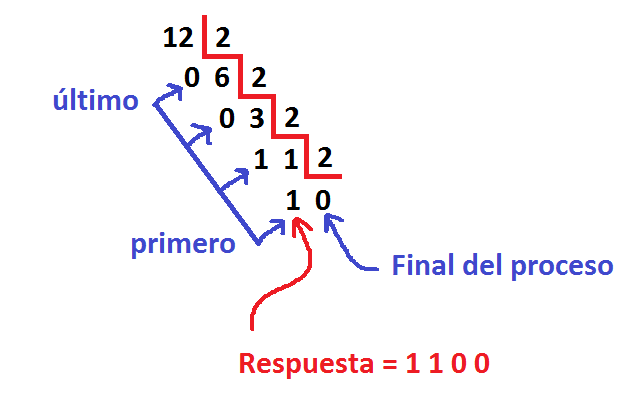
**CÓDIGO: 100468162/ 1010003883**

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

**GITHUB:**

# CONVERSIÓN BASADA EN DIVISIONES SUCESIVAS

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

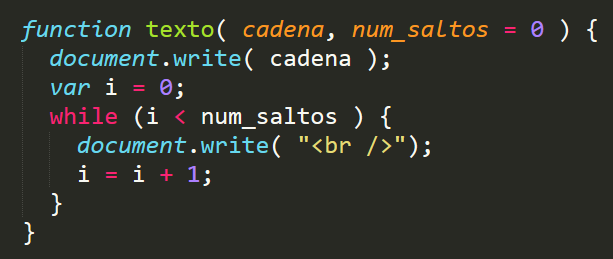


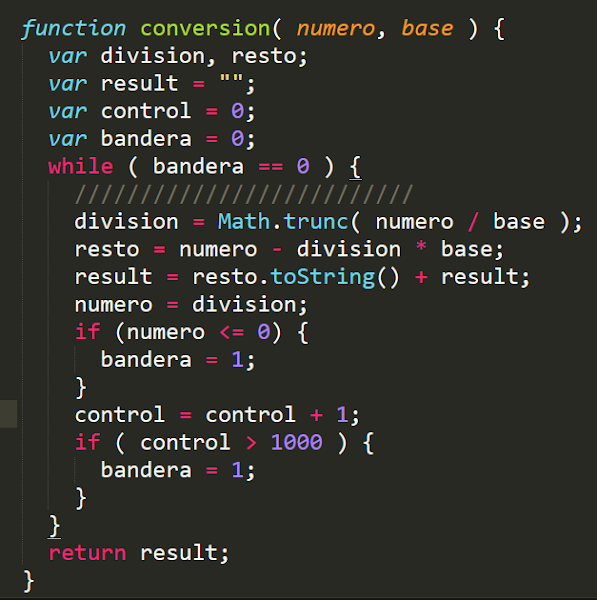
Como se ve en el diagrama, la conversión se realiza dividiendo el número a convertir entre la base seleccionada.

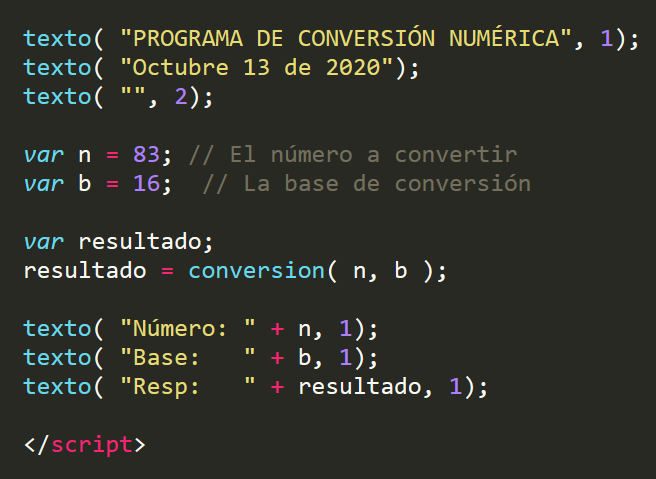
El resultado se obtiene con base en los residuos de las divisiones.

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

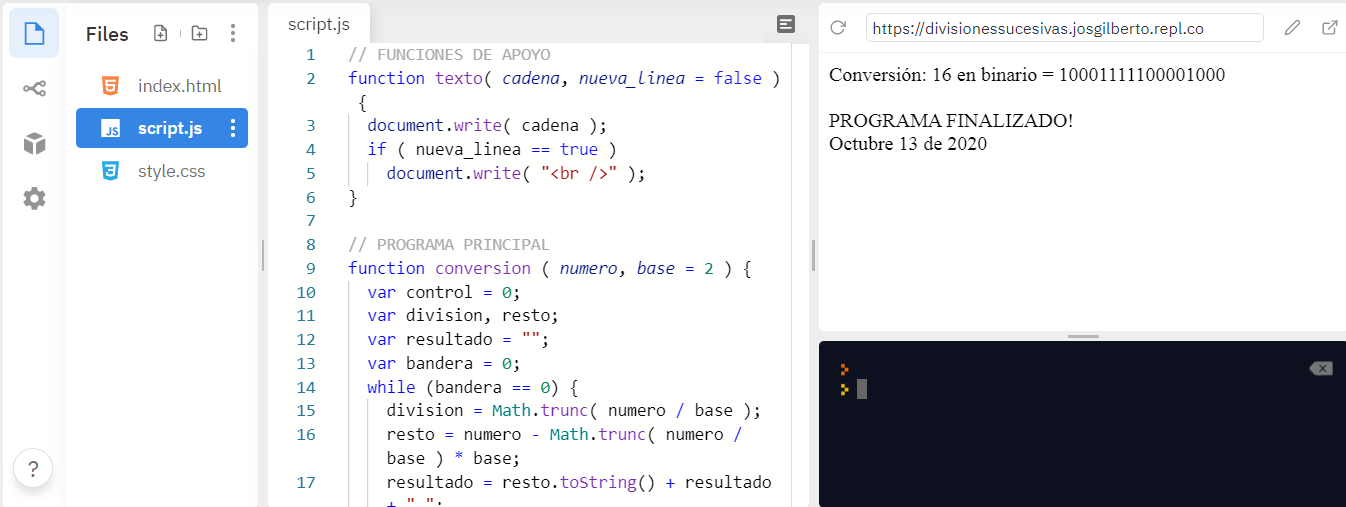
A continuación, se presenta las imágenes de los códigos requeridos, para implementar el proceso mostrado en JavaScript. Cada imagen presenta una función distinta, o la ejecución final del programa. Se debe escribir en un solo archivo el código mostrado, y se sugiere un entorno como repl.it.







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

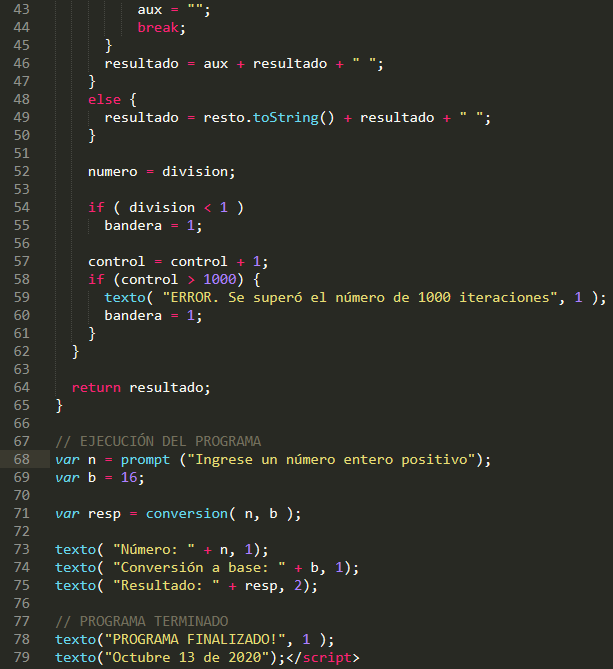


<PRESENTAR UNA EXPLICACIÓN PERSONAL DE ALGUNOS PUNTOS QUE CONSIDERE INTERESANTES EN EL ALGORITMO>

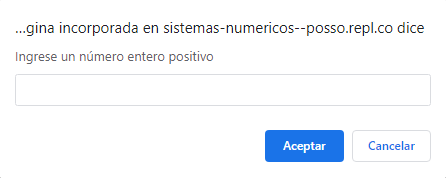
# CONVERSIÓN EXTENDIDA

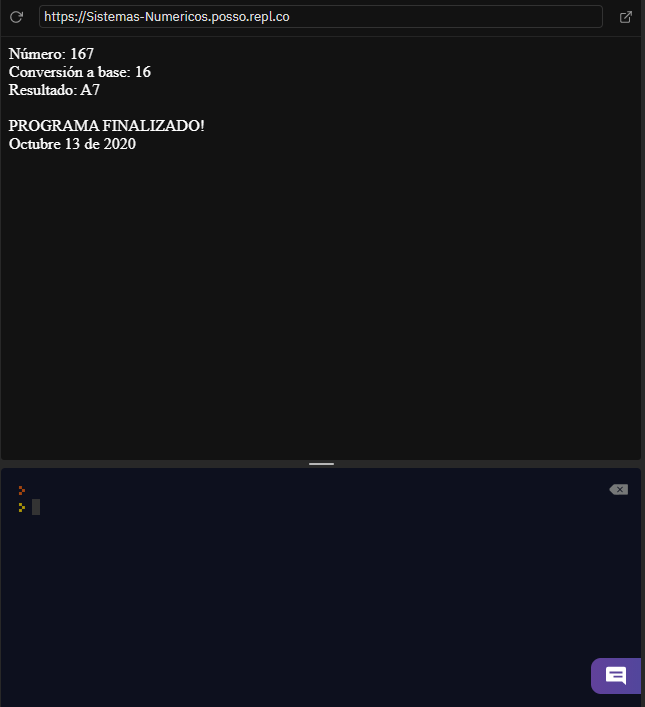
Como podemos observar en la división, para poder pasar cualquier número a sistema hexadecimal lo dividimos entre 16, si es un número muy grande se requieren de varias divisiones, el proceso termina cuando el cociente ya sea menor que 16, como podemos observar en la división que está arriba, el ultimo 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.



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.

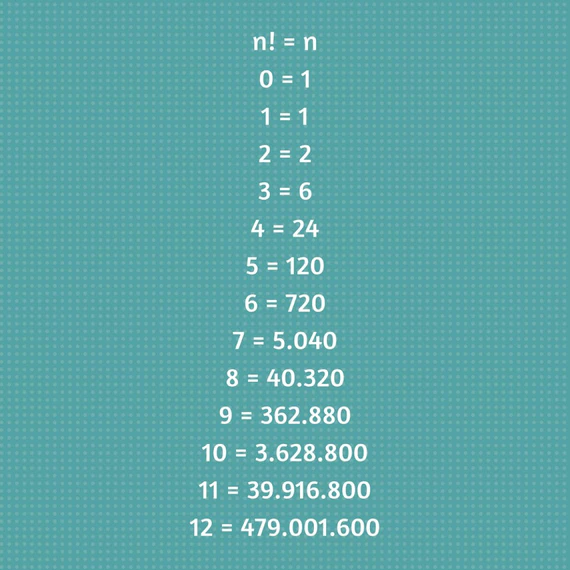
Después de que el usuario ingrese el número, el programa realizará la conversión de dicho número a hexadecimal y lo mostrará en pantalla.

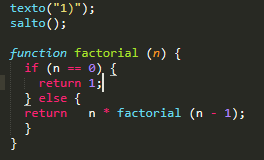
# RECURSIVIDAD I

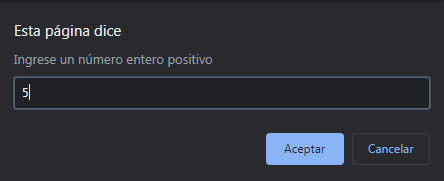
En este punto se da una introducción a la recursividad.

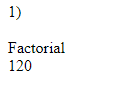
La recursividad es una técnica de programación que consiste en que una función se llame así misma en una especie de bucle hasta que se cumpla o se deje de cumplir una condición, esto depende de cómo se estructure.

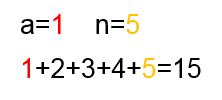
A continuación, se mostrarán tres ejemplos de cómo puede ser empleada la recursividad.

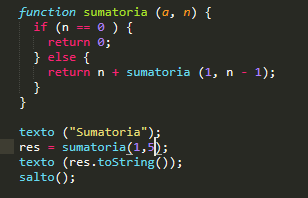
En el primer ejemplo se pide hallar el factorial de “x” número. Un número factorial consiste en multiplicar un número “n” por sus antecesores, por ejemplo, el cuatro [4] se multiplicaría por los números del cero [0] al cuatro [4].

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

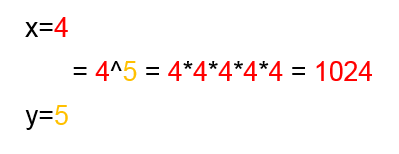
Cuando se ejecuta el código, salta un cuadro emergente que le solicita al usuario que ingrese un número entero positivo

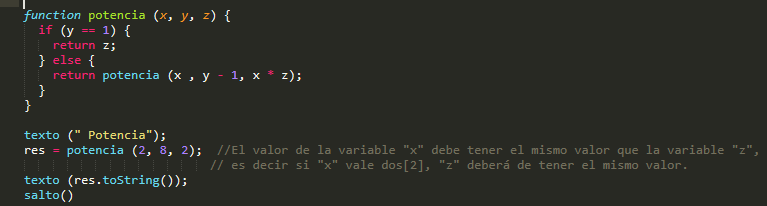
Despues de que el usuario ingrese el número, se mostrara en pantalla el factorial de dicho número.

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

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

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

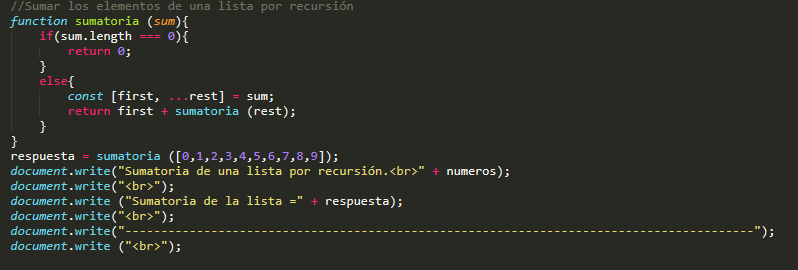
A continuación, se presenta el código que funciona para conseguir la potencia de “x” a la “y”.

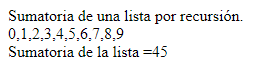
Se utilizan tres variables “x” “y” “z”, la primera se utiliza para el número al que se le desea sacar potencia, la segunda para indicar el exponente de dicho número y la tercera se utiliza para almacenar cada ciclo hecho.

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

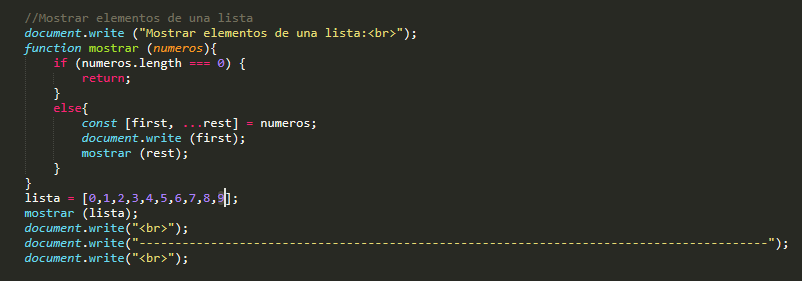
# RECURSIVIDAD II

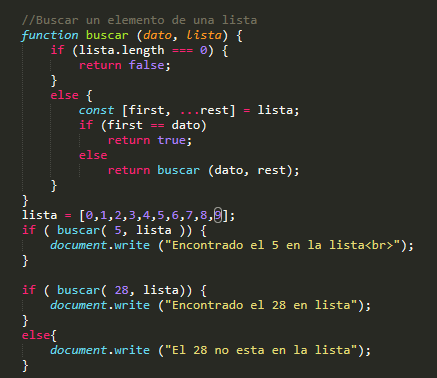
Se sigue realizando trabajos enfocados en la recursión. Se mostrarán tres ejemplos sobre el manejo de la recursividad.

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

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

En el segundo ejemplo 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].

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

En el tercer ejemplo 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.

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 esté elemento no está en la lista.

# 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.

# BIBLIOGRAFÍA

<https://repl.it>