SISTEMAS NUMÉRICOS: Introducción a la Informática

UTP | Pereira

<NOMBRE ESTUDIANTE>

OCTUBRE DE 2020

2020

# CONTENIDO

[1 CONTENIDO 1](#_Toc54269874)

[2 PRESENTACIÓN 2](#_Toc54269875)

[3 CONVERSIÓN BASADA EN DIVISIONES SUCESIVAS 3](#_Toc54269876)

[4 CONVERSIÓN EXTENDIDA 6](#_Toc54269877)

[5 CONVERSIÓN: 0, 1 A PALABRAS: CERO, UNO 7](#_Toc54269878)

[6 CONTAR NÚMERO DE UNOS EN UN BINARIO 9](#_Toc54269879)

[7 CONCLUSIONES 12](#_Toc54269880)

[8 BIBLIOGRAFÍA 13](#_Toc54269881)

# 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.

<UTILIZAR LAS PRESENTACIONES EN POWER POINT Y APUNTES EN EL CLASSROOM, Y EL MATERIAL ADICIONAL QUE CONSIDERE IMPORTANTE>

**AUTOR: <alexander González Ortiz, Juan Esteban Velásquez Molina >**

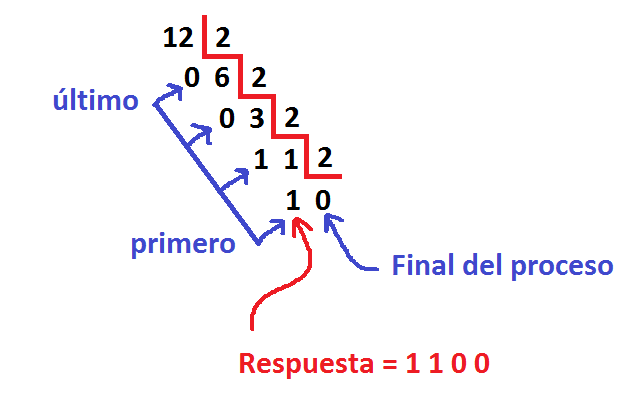
**<CÓDIGO 1004681579, 1006318472 >**

## [<alexander.gonzalez1@utp.edu.co](mailto:<alexander.gonzalez1@utp.edu.co), [juanesteban.velasquez@utp.edu.co](mailto:juanesteban.velasquez@utp.edu.co) >

**<GITHUB https://github.com/AGO2001/segunda\_previa/tree/main/sistemas%20numericos ,** https://github.com/Juaness14 **>**

# CONVERSIÓN BASADA EN DIVISIONES SUCESIVAS

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

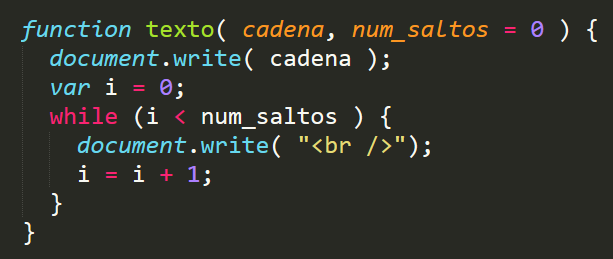


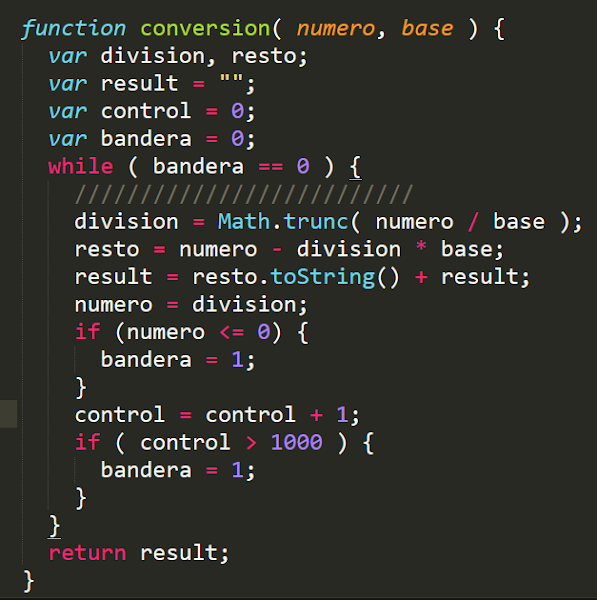
Como se ve en el diagrama, la conversión se realiza dividiendo el número a convertir entre la base seleccionada.

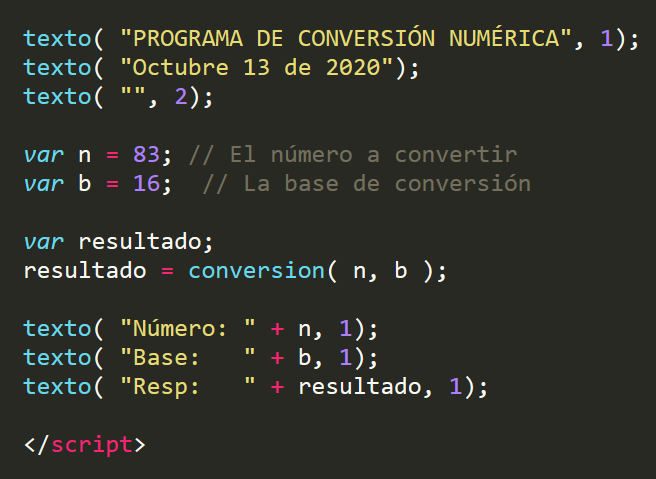
El resultado se obtiene con base en los residuos de las divisiones.

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

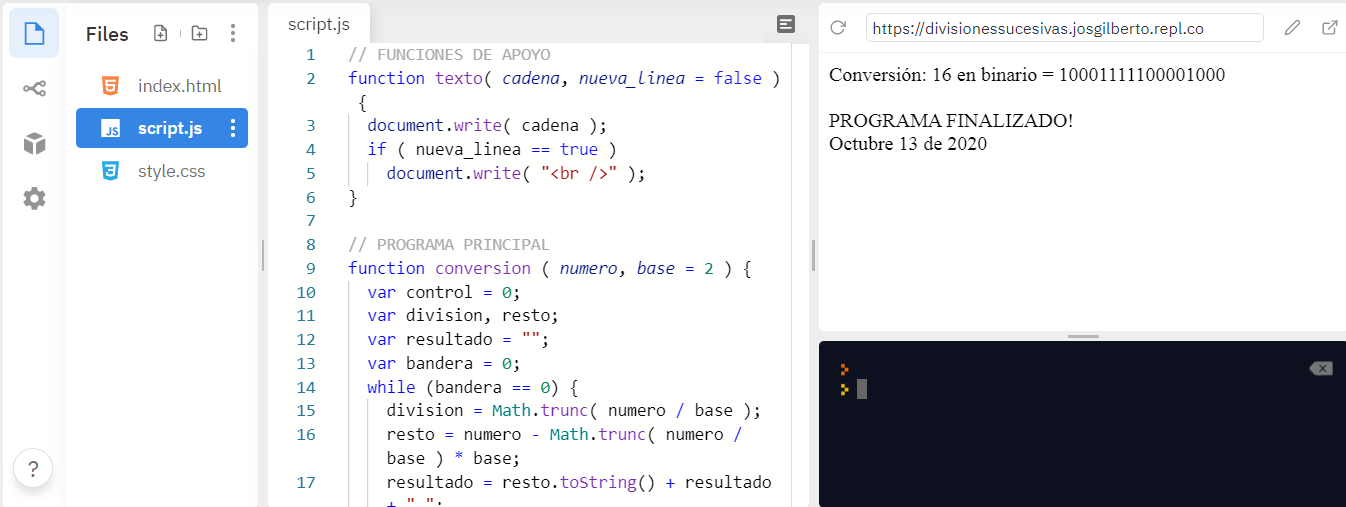
A continuación se presentan las imágenes de los códigos requeridos, para implementar el proceso mostrado en JavaScript. Cada imagen presenta una función distinta, o la ejecución final del programa. Se debe escribir en un solo archivo el código mostrado, y se sugiere un entorno como repl.it.







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



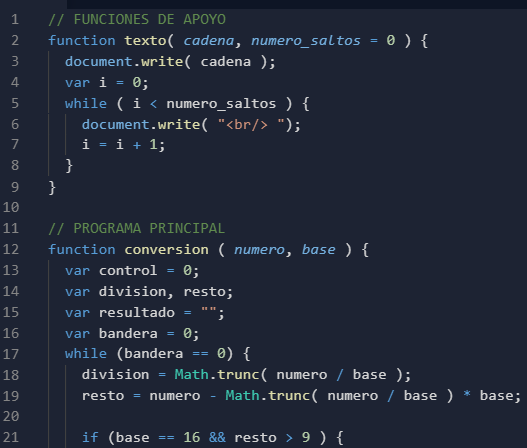
<PRESENTAR UNA EXPLICACIÓN PERSONAL DE ALGUNOS PUNTOS QUE CONSIDERE INTERESANTES EN EL ALGORITMO>

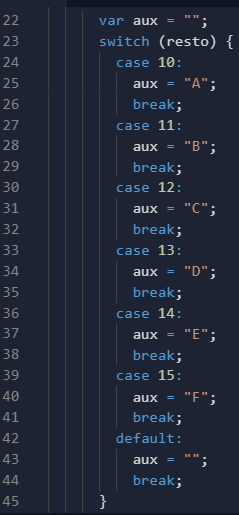
# CONVERSIÓN EXTENDIDA

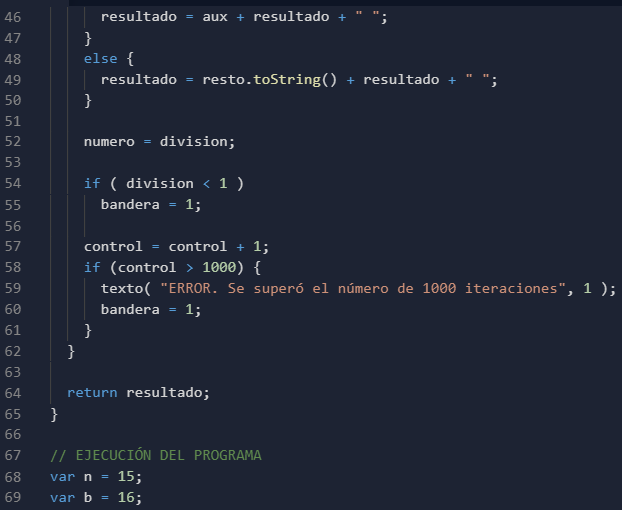
A continuación, presentamos el programa de conversión extendida, la cual se encarga de dar tratamiento a los números en base 16.

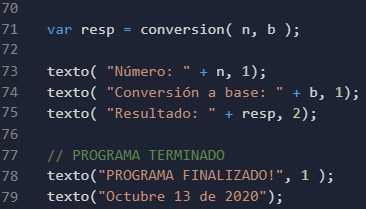
Es similar el código visto en el punto 3, pero ahora el código puede pasar de base 10 a base 16 ya que se incluyen las letras que representan los números del 10 al 15

Se presentará ahora el código de la conversión extendida

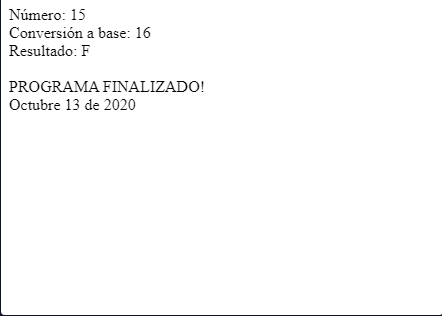








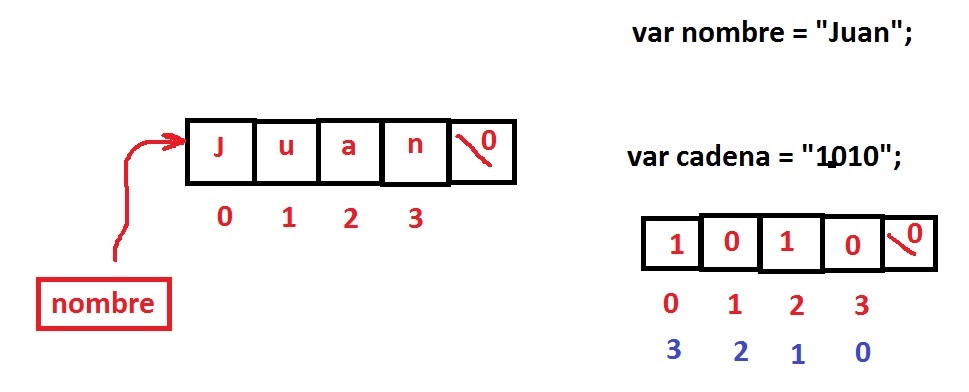
Al ejecutar el programa con base 16 se puede ver efectivamente cumple con lo que se ve escrito en el código



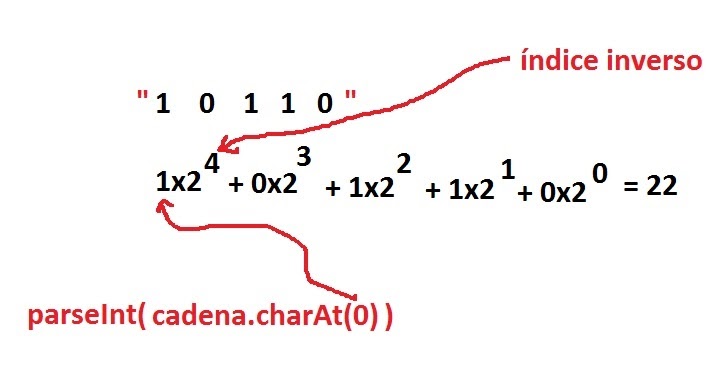
# CONVERSIÓN: 0, 1 A PALABRAS: CERO, UNO

Vamos a presentar el programa que convierte los dígitos binarios 0, 1 a las palabras Cero, Uno.

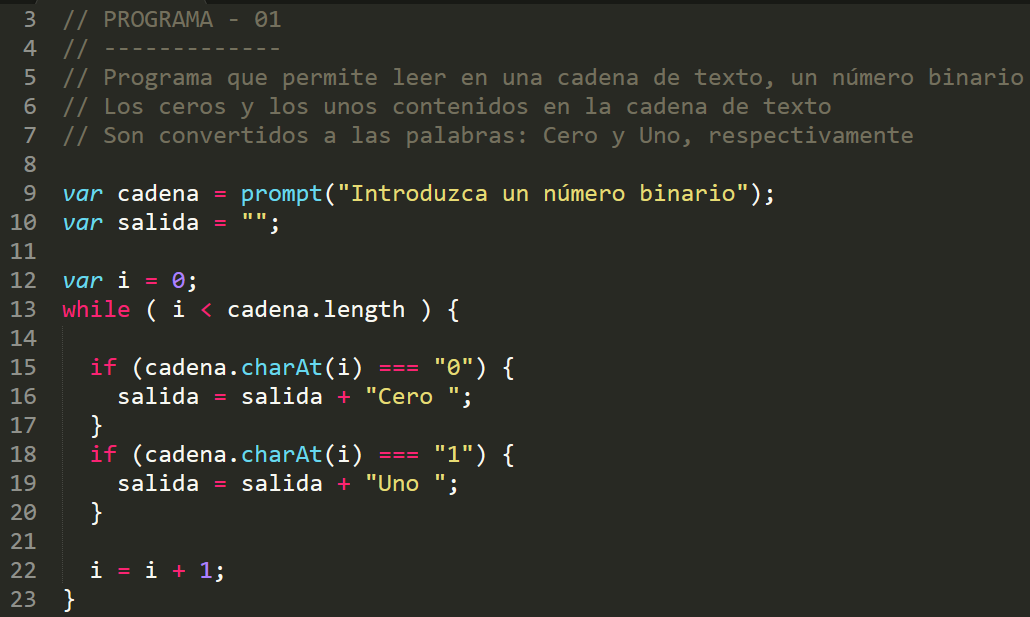
Algunos elementos teóricos. En primer lugar, observamos cómo se almacena una cadena en la memoria:



En la siguiente gráfica, presentamos el índice inverso:



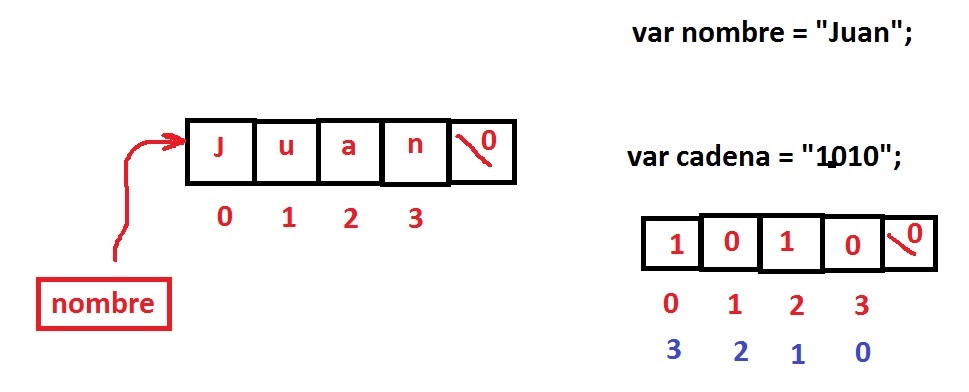
A continuación, presentamos el código fuente, seccionado por partes:



En estas líneas se comenta el programa, se crean algunas variables, se pregunta por una cadena binaria y se procede luego a realizar la conversión, dígito a dígito.

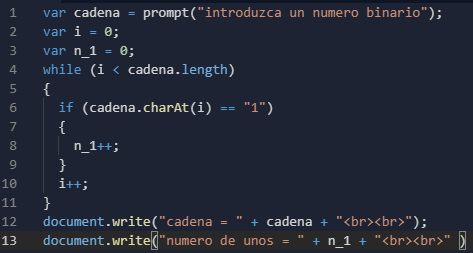
# CONTAR NÚMERO DE UNOS EN UN BINARIO

Vamos a presentar el programa que cuenta el número de unos disponibles en un binario. Veamos la estructura de la cadena que contiene el binario:

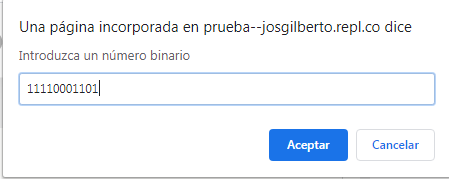


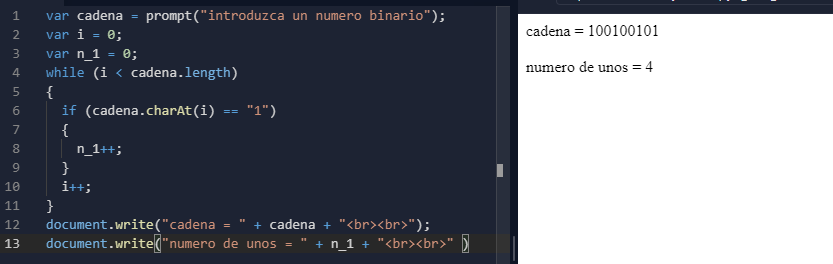
El número de unos se calcula recorriendo la cadena de izquierda a derecha, evaluando cada una de las posiciones en la cadena, y cada vez que el contenido de dicha posición sea uno, sumaremos 1 a un contador preparado para ello.

Seguidamente presentamos el código fuente.



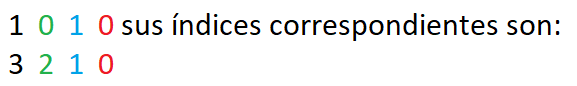
En el código fuente se introduce una variable llamada cadena el cual es donde se va almacenar el numero binario, la variable i es el contador y la variable n\_1 es en donde se va a sumar el número de unos que tiene el código para luego mostrar la cadena y el número total de unos, la (cadena.CharAt(i)) permite al código saber si en la posición en la que esta hay un 1 o un 0



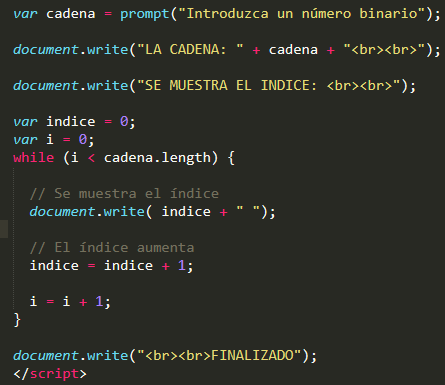


# CONTAR NÚMERO DE UNOS EN UN BINARIO

El siguiente programa a partir de un número binario dado, va a dar el índice correspondiente de cada uno. Un ejemplo ilustrativo seria el siguiente:



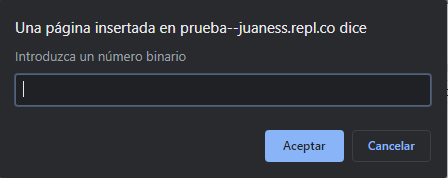
Como se puede observar, los índices están dados de forma inversa a como se escribe el número binario por el usuario, esto debido a que se empieza a contar desde derecha a izquierda los índices de los números binarios.

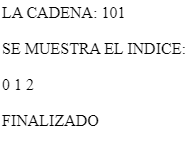


Explicación del código previamente mostrado:

Primero se necesitará que se ingrese un número binario, por ejemplo, el “101”; después que se haya ingresado, se declaran un par de variables, las cuales son, índice y ‘i’, ambas iniciadas en 0. Después de todo esto, se declarará un ciclo while. Mientras la variable ‘i’ sea menor al largo de la cadena ingresada, gracias al. length, se empezará a sumar +1 a la variable índice para ir mostrando en pantalla. Una vez el ciclo while se termine, acabará el programa.

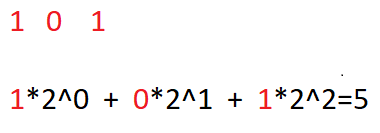
Muestra de cómo compila el programa en repli.it

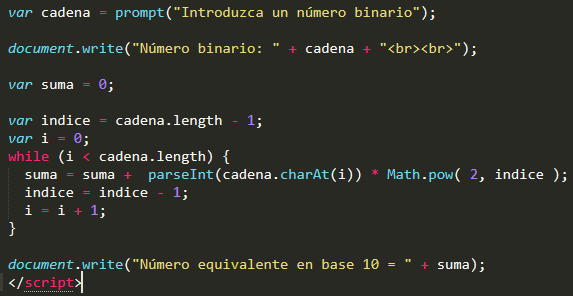




# CONVERTIR NÚMERO BINARIO A BASE 10

En esta ocasión, se convertirá a partir de un número binario a base 10. Haciendo la suma de los números en base 2 elevados a la potencia del índice en el que están del número binario de izquierda a derecha (ver punto 7 para mejor explicación).

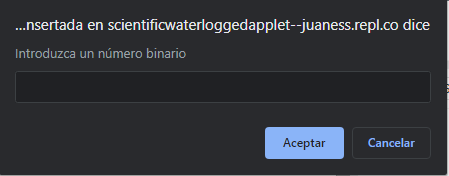


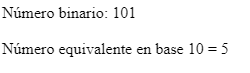


Explicación del código:

Sin ir a muchos tecnicismos, en pocas palabras, el código anteriormente mostrado lo que hace es obtener el largo de la cadena ingresada (en binario), para nuestro ejemplo usaremos el “101”. Luego de obtener ese valor, se va a adentrar a un ciclo while, mientras ese índice sea menor al largo de la cadena “101”, seguirá iterando, mientras eso ocurre, dentro del ciclo está pasando lo que se ilustró arriba, sumar la conversión de binario a base decimal de cada índice.

Aquí el código compilado en rpl.it:





# 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.

# BIBLIOGRAFÍA

<https://repl.it>