

Métodos Numéricos

Conversor Numérico Multibase

Maria Angelica Cruz Guerrero

NRC: 75397

Carlos Javier Mahecha Gomez

Jonatan Camilo Igua Contreras

Grupo E

Corporación Universitaria Minuto de Dios

Bogota D.C

9 - Marzo - 2025

Método para convertir cualquier dato a decimal

Para el programa de conversión se crea un método para reutilizar código a la hora de convertir cualquier tipo numérico a decimal, ya que en muchas conversiones es necesario emplear el valor decimal del número que se quiere convertir. Se creó el método `convertirDatoDecimal()`, este recibe como parámetro el dato y la potencia base del dato.

```
#Metodo para convertir binario / octal y hexa a decimal
def convertirDatoDecimal(dato,base):
    hex = {'A': 10, 'B': 11, 'C': 12, 'D': 13, 'E': 14, 'F': 15} # Diccionario para valores hexadecimales
    decimal = 0 #almacena el resultado de la conversion
    potencia = 0
    #Con reversed se recorre los digitos de manera inversa
    for digito in reversed(dato.upper()):#convertir mayusculas
        if(digito.isdigit()): #Binario / octal (1 al 7)numero del 0 al 9 a entero
            valor = int(digito)
        else: #tiene una letra A - F hexadecimal
            valor = hex[digito]#buscar valor diccionario

        decimal += valor * (base ** potencia)#conversion
        potencia +=1#aumentar la potencia
    return decimal
```

1. Conversión de Binario a los otros sistemas:

El programa cuenta con el siguiente menú de selección del sistema a convertir si se ingresa la opción de Binario se podrá seleccionar convertir este binario en los otros sistemas.

```
#Opciones para el tipo de conversion
def menuConvertirBinario():
    print("Convertir de binario a: ")
    print(" 1. Octal")
    print(" 2. Hexadecimal")
    print(" 3. Decimal")
    print(" 4. Volver al menu anterior")
```

Para convertir de binario a los demás sistemas numéricos antes se debe validar la entrada del usuario.

Se tiene el método `ValidarDato()` éste recibe como parámetro dato y tipo, si el tipo es Binario entonces se valida con el `if` dentro del ciclo `for` que el número ingresado sea 1 o 0, de lo contrario retorna un valor `False`, si es un binario válido retorna `True`

```
#Metodo para validar los datos ingresados
def validarDato(dato, tipo):
    #Validar Binario
    if tipo == "Binario":
        numeroBi = str(dato)
        for digito in numeroBi: #Recorre el digito (Binarios: 1 y 0)
            if digito != '0' and digito != '1':
                return False
        return True
```

Validar entrada:

Ingresar datos diferentes de 0 y 1

```
<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
  1. Binario
  2. Octal
  3. Hexadecimal
  4. Decimal
Digite su eleccion: 1
Convertir de binario a:
  1. Octal
  2. Hexadecimal
  3. Decimal
  4. Volver al menu anterior
Ingrese la opcion que desea convertir: 1
Ingresar el dato a convertir: 102455
El número ingresado no corresponde, a un número binario.
```

Ingresar datos diferentes a números:

```
<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
  1. Binario
  2. Octal
  3. Hexadecimal
  4. Decimal
Digite su eleccion: 1
Convertir de binario a:
  1. Octal
  2. Hexadecimal
  3. Decimal
  4. Volver al menu anterior
Ingrese la opcion que desea convertir: 1
Ingresar el dato a convertir: 14adEr
El número ingresado no corresponde, a un número binario.
```

- *Binario a Octal*

Si el usuario selecciona Octal que es valor 1, en el menú anterior, si la validación del dato binario fue True entonces se llama convertirBinario para hacer la conversión.

```

#Convertir Binario a Octal
if conver == 1:
    #Ingresar el dato a convertir
    dato = input("Ingresar el dato a convertir: ")
    #Tipo de dato para convertir
    conversor = "Octal"
    #Validar si es (dato) es correcto a tipo
    validacion = validarDato(dato,tipo)
    if validacion == True:#Llama al metodo para la convertir
        print("El resultado es: ",convertirBinario(dato,conversor))
    else:
        print("El número ingresado no corresponde, a un número binario.")
        #Volver al menu principal
        inicio()

```

Metodo de conversion Binario a Octal:

Sí el conversor es Octal ingresa, para convertir un binario a octal es necesario crear paquetes de 3 caracteres del binario.

Si al crear los paquetes la cantidad de dígitos no es múltiplo de 3 entonces se añade un cero a la izquierda del paquete.

Luego a esos paquetes que se guardaron en una lista, se recorre cada paquete y se convierte ese binario a decimal multiplicando por base 2, después de cada conversión del paquete se suman para crear el valor del octal recordando que se invierten los resultados dados con reversed.

```

def convertirBinario(dato,conversor):
    #Binario a Octal
    if conversor == "Octal":
        """
        -Se añade ceros a la izquierda
        -Se agrupa el numero binario en paquetes de 3 representan base 8
        - Cada grupo se convierte a octal usando base 2
        """
        tamano=3
        while len(dato) % tamano != 0: #Añadir Los "0" a la izquierda para completar el paquete de 3
            dato = '0' + dato
        Lista = []
        rs = ""
        for i in range(0, len(dato), tamano): #Crear la lista con los paquetes de 3 dígitos
            Lista.append(dato[i:i+tamano])
        for j in range(len(Lista)): #Recorrer la lista de paquetes
            decimal = 0
            potencia = 0
            for digito in reversed(Lista[j]): #Convertir de binario a Octal
                decimal += int(digito) * (2 ** potencia)
                potencia += 1
            rs += str(decimal)
        return rs

```

Ejemplo:

Binario original : 101101

Paquete de 3 datos: 101 / 101

Conversión : 5 / 5

Resultado Octal: 55

```

<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
1. Binario
2. Octal
3. Hexadecimal
4. Decimal
Digite su eleccion: 1
Convertir de binario a:
1. Octal
2. Hexadecimal
3. Decimal
4. Volver al menu anterior
Ingrese la opcion que desea convertir: 1
Ingresar el dato a convertir: 101101
El resultado es: 55
}

```

- ***Binario a Hexadecimal***

Metodo de conversion Binario a Hexadecimal:

Primero se crea un diccionario en donde se relaciona la clave con el número y el valor de equivalencia en letras para el sistema hexadecimal. Se empleó un sistema de 4 bloques de 4 bits, en donde en el ciclo while evalúa si el binario es múltiplo de 4 si no, se agregan ceros a la izquierda para completar los paquetes.

Luego en el ciclo for se divide los bloques de “dato”, en 4 bits, luego se crea la variable decimal en la cual se almacenará la conversión de ese bloque de 4 bits a binario, empleado un método llamado convertirDatoDecimal (bloque, 2)

Al final se evalúa si el decimal retornado es mayor o igual a 10, para buscar ese valor en el diccionario y asignarle su letra correspondiente. Al final en la variable hexadecimal se van concatenando los resultados de la conversión de los decimales generados con la técnica de los 4 bits.

```

elif conversor == "Hexadecimal":
    hex_dict = {10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E', 15: 'F'}
    while len(dato) % 4 != 0: # Completar con ceros a la izquierda
        dato = '0' + dato
    hexadecimal = "" #guardar resultado final
    for i in range(0, len(dato), 4): # Tomar bloques de 4 bits
        bloque = dato[i:i+4]
        decimal = convertirDatoDecimal(bloque, 2) # Convertir a decimal usando el metodo
        if decimal >= 10: # Convertir valores 10-15 en A-F
            hexadecimal += hex_dict[decimal] # Buscar en el diccionario
        else:
            hexadecimal += str(decimal)
    return hexadecimal

```

Ejemplo:

Binario original : 10111101

El binario ingresado es de 8 bits el cual es múltiplo de 4 así que no es necesario agregar ceros para completar los paquetes.

Paquete de 4 bits: Primer bloque (1011) , Segundo Bloque (1101)

Convertir cada bloque de binario a decimal

Primer bloque 1011

$$(1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (1 * 2^0) = 8 + 0 + 2 + 1 = 11$$

Buscar el 11 en el diccionario 11 = B

Segundo bloque 1101

$$(1 * 2^3) + (1 * 2^2) + (0 * 2^1) + (1 * 2^0) = 8 + 4 + 0 + 1 = 13$$

Buscar el 13 en el diccionario 13 = D

Resultado hexadecimal: BD

```
<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
1. Binario
2. Octal
3. Hexadecimal
4. Decimal
Digite su eleccion: 1
Convertir de binario a:
1. Octal
2. Hexadecimal
3. Decimal
4. Volver al menu anterior
Ingrese la opcion que desea convertir: 2
Ingresar el dato a convertir: 10111101
El resultado es: BD
```

- **Binario a Decimal**

Metodo de conversion Binario a Decimal:

Para convertir el binario en decimal se usó el método de convertirDatoDecimal, para retornar su valor en decimal

```
elif conversor == "Decimal":
    #Se llama al metodo con base 2
    return convertirDatoDecimal(dato,2)
```

Se le pasa a el método el dato = binario y la base 2.

para hacer la conversión se usa la expansión de potencias de 2

Si se ingresa el número binario: 1101

la secuencia de potencias es la siguiente

$$(1 * 2^3) + (1 * 2^2) + (0 * 2^1) + (1 * 2^0) = 8 + 4 + 0 + 1 = 13$$

Valor decimal es : 13

En el método se ingresa al if en donde con reversed se recorre el número de derecha a izquierda, en la parte decimal $\text{+= valor} * (\text{base} ** \text{potencia})$, se multiplica el dígito binario por 2^n y se suma al resultado.

```
#Con reversed se recorre los dígitos de manera inversa
for digito in reversed(dato.upper()): #convertir mayúsculas
    if(digito.isdigit()): #Binario / octal (1 al 7) número del 0 al 9 a entero
        valor = int(digito)
    else: #tiene una letra A - F hexadecimal
        valor = hex[digito] #buscar valor diccionario

    decimal += valor * (base ** potencia) #conversion
    potencia += 1 #aumentar la potencia
return decimal
```

Ejemplo

```
<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
1. Binario
2. Octal
3. Hexadecimal
4. Decimal
Digite su elección: 1
Convertir de binario a:
1. Octal
2. Hexadecimal
3. Decimal
4. Volver al menú anterior
Ingrese la opción que desea convertir: 3
Ingresar el dato a convertir: 1101
El resultado es: 13
```

2. Conversión de Octal a los otros sistemas:

El programa cuenta con el siguiente menú de selección del sistema a convertir si se ingresa la opción de Octal se podrá seleccionar convertir este octal en los otros sistemas.

```
def menuConvertirOctal():
    print("Convertir de Octal a: ")
    print(" 1. Binario")
    print(" 2. Hexadecimal")
    print(" 3. Decimal")
    print(" 4. Volver al menú anterior")
```

- **Octal a Binario**

Método de conversión Octal a Binario:

Para convertir de octal a binario se usa un diccionario con las conversiones del 0 al 7 en binario, se recorre cada dígito del número octal ingresado se reemplaza por su equivalente en binario, al final del reemplazo se eliminan los ceros de la izquierda

```
#Metodo para convertir valores Octal
def convertirOctal(dato, conversor):
    if conversor == "Binario":
        octal_a_binario = {#almacenar conversiones
            '0': '000', '1': '001', '2': '010', '3': '011',
            '4': '100', '5': '101', '6': '110', '7': '111'
        }
        binario = "" # Variable para almacenar el resultado
        # Recorremos cada dígito del número octal
        for digito in dato:
            binario += octal_a_binario[digito]#reemplazo con su valor binario
        binario = binario.lstrip('0') # se usa esto para eliminar lo 0 a la izquierda.
        return binario
```

Ejemplo:

Número Octal: 27

Búsqueda del valor:

Primer valor = 2 [010]

Segundo valor = 7 [111]

Unión de las conversiones = 01011

Quitar los ceros de la izquierda = 10111

Conversion final: 10111

```
<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
1. Binario
2. Octal
3. Hexadecimal
4. Decimal
Digite su eleccion: 2
Convertir de Octal a:
1. Binario
2. Hexadecimal
3. Decimal
4. Volver al menu anterior
Ingrese la opcion que desea convertir: 1
Ingresar el dato a convertir: 27
El resultado es: 10111
```

- **Octal a Hexadecimal**

Método de conversión Octal a Hexadecimal:

Para convertir un octal a un hexadecimal se usa el método anteriormente mencionado `convertirDatosDecimal()`, para poder obtener el equivalente del octal en decimal

Se usa la base 8 para hacer multiplicaciones sucesivas, por ejemplo

Número Octal: 764

$$(7 * 8^2) + (6 * 8^1) + (4 * 8^0) = 448 + 48 + 4 = 500$$

Después de tener el número decimal, se divide el decimal entre 16, se guarda el residuo, se repite hasta que del cociente sea cero, se toman los residuos de manera inversa para formar el número hexadecimal, al final se evalúa si es mayor o igual a 10 para cambiarlos por las letras


```

elif conversor == "Hexadecimal":
    #convertir octal a decimal
    decimal = convertirDatoDecimal(dato, 8)
    hex = {10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E', 15: 'F'}
    hexadecimal = "" #dato convertido
    #Convertir de decimal a hexadecimal
    while decimal > 0:
        residuo = decimal % 16 # Obtener el residuo (dígito hexadecimal)
        if residuo >= 10:
            hexadecimal = hex[residuo] + hexadecimal # Convertir a A-F
        else:
            hexadecimal = str(residuo) + hexadecimal # Mantener números 0-9
        decimal //= 16 # Dividir por 16 para seguir con el siguiente dígito
    return hexadecimal if hexadecimal else "0" # Retorna "0" si el número es 0
else:
    return "No se encontro el tipo de convertidor"

```

Ejemplo

Número Octal: 764

Número decimal: 500

División:

$500 \% 16 = \text{cociente} \Rightarrow 31 \text{ residuo} \Rightarrow 4$

$31 \% 16 = \text{cociente} \Rightarrow 1 \text{ residuo} \Rightarrow 15$

$1 \% 16 = \text{cociente} \Rightarrow 0 \text{ residuo} \Rightarrow 1$

Número dado = 1 15 4

Número convertido= 1F4

```

<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
1. Binario
2. Octal
3. Hexadecimal
4. Decimal
Digite su eleccion: 2
Convertir de Octal a:
1. Binario
2. Hexadecimal
3. Decimal
4. Volver al menu anterior
Ingrese la opcion que desea convertir: 2
Ingresar el dato a convertir: 764
El resultado es: 1F4

```

- *Octal a Decimal*

Para convertir el octal en decimal se usó el método de convertirDatoDecimal, para retornar su valor en decimal

```

elif conversor == "Decimal":
    #Se llama al metodo con base 8
    return convertirDatoDecimal(dato,8)

```

Se le pasa a el método el dato = octal y la base 8.

para hacer la conversión se usa la expansión de potencias de 8

Si se ingresa el número octal: 745

la secuencia de potencias es la siguiente

$$(7 * 8^2) + (4 * 8^1) + (5 * 8^0) = 448 + 32 + 5 = 485$$

Valor del decimal es : 485

```
<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
1. Binario
2. Octal
3. Hexadecimal
4. Decimal
Digite su eleccion: 2
Convertir de Octal a:
1. Binario
2. Hexadecimal
3. Decimal
4. Volver al menu anterior
Ingrese la opcion que desea convertir: 3
Ingresar el dato a convertir: 745
El resultado es: 485
```

3. Conversión de Decimal a los otros sistemas

El programa cuenta con el siguiente menú de selección del sistema a convertir si se ingresa la opción de Decimal se podrá seleccionar convertir este decimal en los otros sistemas.

```
def menuConvertirDecimal():
    print("Convertir de Decimal a: ")
    print(" 1. Binario")
    print(" 2. Octal")
    print(" 3. Hexadecimal")
    print(" 4. Volver al menu anterior")
```

- **Convertir de decimal a binario:**

Cuando el usuario elige convertir un número decimal a binario, el código primero separa el número en dos partes: la parte entera que corresponde a la variable “p_entera” y la parte decimal que es la “p_decimal”.

```
def convertirDecimal(dato, conversor):
    dato = float(dato) # convertir dato a decimal

    if conversor == "Binario": # decimal a binario
        p_entera = int(dato) # parte entera del dato
        p_decimal = dato - p_entera # parte decimal
        # convertir parte entera
        binario_entero = "" # guardar binario entero
        if p_entera == 0:
            binario_entero == "0"
        while p_entera > 0: # residuo division
            binario_entero = str(p_entera % 2) + binario_entero
            p_entera //= 2 # dividir el numero por 2 mas pequeño
        # cuando se termine el ciclo la variable binario_entero
        # tiene los valores de los residuos de la parte entera del numero

        # Convertir parte decimal
        binario_decimal = "" # guardar binario decimal
        precision = 10 # numeros de cifras significativas a tomar
        while p_decimal > 0 and len(binario_decimal) < precision:
            p_decimal *= 2
            digito = int(p_decimal) # parte entera se obtiene mutiplicar * 2
            binario_decimal += str(digito)
            p_decimal -= digito
        # unir cada parte
        if binario_decimal: # si binario tiene algo se ingreso un numero decimal
            return binario_entero + '.' + binario_decimal
        return binario_entero # se ingreso un entero
```

Luego en el proceso para la parte entera, se usa un bucle while que continúa mientras la parte entera sea mayor a 0, en cada iteración, se obtiene el residuo de dividir por 2, este residuo representa el dígito binario menos significativo y se coloca al inicio de la cadena “binario_entero”. Por último, la parte entera se divide entre 2 para seguir obteniendo los siguientes dígitos binarios. Para la parte decimal, se utiliza otro bucle while para manejar la parte decimal, que se repite hasta alcanzar 10 cifras significativas o hasta que la parte decimal llegue a 0, donde en cada iteración, la parte decimal se multiplica por 2. La parte entera del resultado se convierte en el siguiente dígito binario y se agrega a la cadena “binario_decimal”. Por último, esta parte entera se resta de la parte decimal original para continuar el proceso, y para el resultado concatenarlo, si tenemos una parte del dato en decimal, se une con la parte entera usando un punto “.” para formar el número binario completo, por ejemplo, si el decimal es 5.625 se convierte en 101.101 que corresponde al número binario.

- **Convertir de decimal a octal:**

Cuando se selecciona la opción de convertir un número decimal a octal, el código utiliza la base 8 para realizar la conversión. El proceso se divide en dos partes: la conversión de la parte entera y la parte decimal del número. Esto garantiza que el resultado final sea preciso tanto para los números enteros como para los números con decimales.

```
elif conversor == "Octal":
    p_entera = int(dato) # parte entera del dato
    p_decimal = dato - p_entera # parte decimal

    residuos = [] # guardar los residuos
    # parte entera a octal
    while p_entera > 0:
        residuos.append(p_entera % 8) # guardar residuo
        p_entera //= 8 # dividir el numero por 8
    # Invertir la lista y unir los valores en una cadena
    # si residuos esta vacio se usa el if, para evitar la union de un valor vacio en residuos []
    octal_entero = ''.join(map(str, residuos[::-1])) if residuos else "0"
    # parte decimal a octal
    octal_decimal = "" # guardar los valores decimales
    precision = 10
    while p_decimal > 0 and len(octal_decimal) < precision:
        p_decimal *= 8
        digito = int(p_decimal)
        octal_decimal += str(digito)
        p_decimal -= digito
    # union de partes
    if octal_decimal:
        return octal_entero + '.' + octal_decimal
    return octal_entero
```

El primer paso consiste en convertir la parte entera del número decimal a octal. Para lograr esto, se utiliza un bucle while que continúa ejecutándose mientras la parte entera sea mayor a 0. En cada iteración, el número se divide entre 8 y el residuo de esta división se almacena en una lista llamada residuos. Estos residuos representan los dígitos del número octal, comenzando por el dígito menos significativo. Una vez que el bucle termina, la lista de residuos se invierte para formar el número octal en el orden correcto. Si la lista está vacía lo que sucede si el número ingresado es 0, se asigna directamente "0" a la variable "octal_entero". Para la parte decimal, el proceso es diferente. Se utiliza otro bucle while que se repite hasta alcanzar un máximo de 10 cifras significativas o hasta que la parte decimal se convierta en 0. En cada iteración, la parte decimal se multiplica por 8. El resultado de esta operación se descompone en dos partes: la parte entera, que se convierte en el siguiente dígito octal y se agrega a la cadena "octal_decimal", y la nueva parte decimal, que se obtiene restando la parte entera al resultado original. Este ciclo continúa hasta cumplir las condiciones mencionadas, asegurando que los dígitos octales generados sean precisos. El resultado de la conversión se construye combinando la parte entera y la parte decimal del número octal. Si existe una parte decimal, se utiliza un punto "." para separarlas. Por ejemplo, si el número decimal ingresado es 78.34, el código lo convierte a 116.26 en formato octal.

- **Convertir de decimal a hexadecimal:**

La conversión de un número decimal a hexadecimal utiliza la base 16 y un enfoque similar al de la conversión a octal, pero con algunas diferencias clave. En hexadecimal, los dígitos que van del 10 al 15 se representan con las letras A, B, C, D, E y F. El código utiliza un diccionario especial llamado “hex” para manejar esta conversión de manera eficiente.

```
elif conversor == "Hexadecimal":
    dato = float(dato) # Convertir a flotante
    p_entera = int(dato) # Parte entera
    hexa_entero = "0" # Inicializar en caso de que el número sea menor a 1
    hex = {10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E', 15: 'F'}
    # Convertir parte entera
    if p_entera > 0:
        residuos = []
        while p_entera > 0:
            hexadecimal = p_entera % 16
            if hexadecimal >= 10:
                residuos.append(hex[hexadecimal]) # Convertir a A-F
            else:
                residuos.append(str(hexadecimal)) # Mantener números 0-9
            p_entera //= 16 # Dividir por 16 para seguir con el siguiente dígito
        hexa_entero = ''.join(residuos[::-1]) # Orden correcto

    # Convertir parte decimal
    p_decimal = dato - int(dato) # Extraer parte decimal
    hexa_decimal = ""
    precision = 10 # Número de cifras decimales
    while p_decimal > 0 and len(hexa_decimal) < precision:
        p_decimal *= 16
        digito = int(p_decimal)
        if digito >= 10:
            hexa_decimal += hex[digito]
        else:
            hexa_decimal += str(digito)
        p_decimal -= digito

    # Unir ambas partes
    if hexa_decimal:
        return hexa_entero + '.' + hexa_decimal
    return hexa_entero
```

Para convertir la parte entera, el código emplea un bucle while que sigue ejecutándose mientras el número sea mayor a 0. En cada iteración, el número se divide por 16 y el residuo se guarda en una lista llamada “residuos”. Si el residuo es menor a 10, se almacena como está; pero si es 10 o mayor, se utiliza el diccionario “hex” para reemplazarlo con la letra correspondiente (A-F). Una vez completadas todas las iteraciones, la lista se invierte para formar la cadena “hexa_entero”, en el orden correcto. Si el número ingresado era 0, se asigna directamente "0" como resultado. Para convertir la parte decimal, se sigue un procedimiento similar al utilizado en la conversión a octal, pero multiplicando la parte decimal por 16 en lugar de 8. El bucle while se repite hasta alcanzar un máximo de 10 cifras significativas o hasta que la parte decimal se convierta en 0. En cada iteración, la parte entera del resultado se convierte en el siguiente

dígito hexadecimal y se agrega a la cadena "hexa_decimal". Si este dígito es 10 o mayor, se reemplaza usando el diccionario "hex" para obtener la letra correspondiente. El proceso continúa hasta que se cumplen las condiciones del bucle, asegurando una conversión precisa de los decimales. El resultado final se construye uniendo la parte entera y la parte decimal con un punto "." en el medio, si es necesario. Por ejemplo, el número decimal 47.8125 se convierte en 2F.D en hexadecimal.

4. Conversión de Hexadecimal a los otros sistemas

El programa cuenta con el siguiente menú de selección del sistema a convertir si se ingresa la opción de Hexadecimal se podrá seleccionar convertir este hexadecimal en los otros sistemas.

```
def menuConvertirHexadecimal():  
    print("Convertir de Hexadecimal a: ")  
    print(" 1. Binario")  
    print(" 2. Octal")  
    print(" 3. Decimal")  
    print(" 4. Volver al menu anterior")
```

Después que la persona seleccione que desea convertir un dato hexadecimal a otro tipo de dato, la persona digitara el número hexadecimal.. Posteriormente el aplicativo procederá a llamar el método "validarDato" el cual se le pasa como parámetro "tipo" que corresponde a hexadecimal ya que el valor ingresado es hexadecimal siempre y el valor, este proceso se realiza debido a que primero se valida que el número que desea convertir a hexadecimal, una vez el dato sea validado como hexadecimal seguirá con el proceso de conversión, en caso de no ser así, lo devolverá al inicio a realizar nuevamente el proceso. A continuación, se mostrará los dos posibles casos:

- Caso en digitar correctamente un número hexadecimal:

```
<<Convertidor de Numeros>>  
¿Selecciona el tipo de dato a ingresar?:  
 1. Binario  
 2. Octal  
 3. Hexadecimal  
 4. Decimal  
Digite su eleccion: 3  
Convertir de Hexadecimal a:  
 1. Binario  
 2. Octal  
 3. Decimal  
 4. Volver al menu anterior  
Ingresa la opcion que desea convertir: 1  
Ingresa el dato a convertir: 2F  
El resultado es: 101111
```

- Caso en digitar incorrectamente un número hexadecimal:

```
<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
1. Binario
2. Octal
3. Hexadecimal
4. Decimal
Digite su eleccion: 3
Convertir de Hexadecimal a:
1. Binario
2. Octal
3. Decimal
4. Volver al menu anterior
Ingresa la opcion que desea convertir: 1
Ingresar el dato a convertir: MN20
El número ingresado no corresponde, a un número Hexadecimal.
<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
1. Binario
2. Octal
3. Hexadecimal
4. Decimal
Digite su eleccion: █
```

Después de conocer como válida el número hexadecimal y el menú, conoceremos cómo realiza el proceso de conversión para cada tipo de dato ya sea binario, decimal y octal.

- Convertir de hexadecimal a decimal:

```
elif conversor == "Decimal":
    #Se llama al metodo base 16
    return convertirDatoDecimal(dato,16)
```

Después que la persona seleccione la opción de convertir de decimal a hexadecimal, y digitar el dato, lo lleva al método “convertirHexadecimal” que recibe el dato y el conversor, donde este último indica a qué dato se va a convertir, en este caso es decimal por lo tanto entra en el if que le corresponde. Aquí esta llama al método “convertirDatoDecimal” enviándole el dato y 16, tener en cuenta que el 16 indica la base del sistema hexadecimal.

```
Tabnine | Edit | Test | Explain | Document
def convertirDatoDecimal(dato,base):
    hex = {'A': 10, 'B': 11, 'C': 12, 'D': 13, 'E': 14, 'F': 15} # Diccionario para valores hexadecimales
    decimal = 0 #almacena el resultado de la conversion
    potencia = 0
    #Con reversed se recorre los digitos de manera inversa
    for digito in reversed(dato.upper()):#convertir mayusculas
        if(digito.isdigit()): #Binario / octal (1 al 7)numero del 0 al 9 a entero
            valor = int(digito)
        else: #tiene una letra A - F hexadecimal
            valor = hex[digito]#buscar valor diccionario

        decimal += valor * (base ** potencia)#conversion
        potencia +=1#aumentar la potencia
    return decimal
```

Luego en el método `convertirDatoDecimal`, en este primer paso guarda los valores numéricos de las letras en hexadecimal en la variable “Hex”, para después recorrer el dato en inverso para conocer sus potencias, es decir, la posición empezando desde 0, luego le da el valor a las letras dependiendo de la variable “Hex” y para finalizar lo convierte a decimal con la fórmula (valor * Base elevada a la potencia). Para entender mejor este proceso de convertir de hexadecimal a decimal, daremos un ejemplo:

Número hexadecimal: 2F

Pasos:

1. Primer paso, agarra el primer dígito que es F (15) $\rightarrow 15 \cdot 16^0 = 15$
2. Segundo paso Toma el segundo dígito que es 2 $\rightarrow 2 \cdot 16^1 = 32$
3. Tercer paso, suma los resultados $32 + 15 = 47$ decimal

```
#Metodo convertir valores hexadecimal
Tabnine | Edit | Test | Explain | Document
def convertirHexadecimal(dato, conversor):
    if conversor == "Binario":
        decimalHex = convertirDatoDecimal(dato, 16)
        binario = ""
        while decimalHex > 0:
            residuo = decimalHex % 2
            binario = str(residuo) + binario #al inicio el residuo
            decimalHex //= 2 #dividimos el numero entre dos division sin decimales
        return binario
```

- Convertir de hexadecimal a binario:

```
def convertirHexadecimal(dato, conversor):
    if conversor == "Binario":
        decimalHex = convertirDatoDecimal(dato, 16)
        binario = ""
        while decimalHex > 0:
            residuo = decimalHex % 2
            binario = str(residuo) + binario #al inicio el residuo
            decimalHex //= 2 #dividimos el numero entre dos division sin decimales
        return binario
```

Después que la persona eligió la opción de hexadecimal a binario y digitó el dato, este va al método de “`convertirHexadecimal`”, recibiendo los parámetros correspondiente al dato, y tipo de convertidor que desea. Una vez allí este llama al método “`convertirDatoDecimal`” para recibir el número en decimal (Este proceso se explicó a detalle en “convertir hexadecimal a decimal”), luego que este le retorne el decimal procesado a convertirlo en binario. Para realizar este proceso de decimal a binario lo que hace es dividir el número entre 2 y va guardando el residuo en una cadena que se declaró antes del ciclo while llamado “binario” para después mostrar el resultado. Para más claridad se muestra este último proceso de decimal a binario:

Número hexadecimal: 2F

Pasos:

1. Primero realiza el proceso de mandarle este al método “convertirDatoDecimal” que retorna un decimal, que en este caso es 47.
2. Segundo divide este entre 2 hasta que el dividendo sea menor al divisor y el residuo lo va guardando en la variable binario:
 - a. $47/2 = 1$ binario = “1”
 - b. $23/2 = 1$ binario = “11”
 - c. $11/2 = 1$ binario = “111”
 - d. $5/2 = 1$ binario = “1111”
 - e. $2/2 = 0$ binario = “01111”
 - f. $\frac{1}{2} = 1$ binario = 101111”
3. Resultado del número binario: 101111

- Convertir de hexadecimal a octal:

```
elif conversor == "Octal":
    #convertir hexa a decimal
    decimal = convertirDatoDecimal(dato, 16)
    #decimal a octal
    if decimal == 0:
        return "0"
    octal = ""#guardar valor
    while decimal > 0:
        octal = str(decimal % 8) + octal # Obtener el residuo para crear el octal
        decimal //= 8 # Dividir por 8 para seguir con el siguiente dígito
    return octal
```

Después que la persona eligió la opción de hexadecimal a octal y dígitó el dato, este va al método de “convertirHexadecimal”, recibiendo los parámetros correspondiente al dato, y tipo de convertidor que desea. Una vez allí este llama al método “convertirDatoDecimal” para recibir el número en decimal (Este proceso se explicó a detalle en “convertir hexadecimal a decimal”), luego que este le retorne el decimal procese a convertirlo en octal. Para realizar este proceso de decimal a octal lo que hace es que si el número decimal es 0 retorna 0 directamente, pero si no es 0 por medio de un ciclo while va dividiendo el decimal en 8 y va guardado el el residuo en la variable “octal” creada con anterioridad y el resultado lo sigue dividiendo hasta que este sea menor que 8. Para dar más claridad con el ejercicio, daremos un ejemplo del proceso:

Número hexadecimal: 2F

Pasos:

1. Primero realiza el proceso de mandarle este al método “convertirDatoDecimal” que retorna un decimal, que en este caso es 47.
2. Segundo divide este entre 8 hasta que el dividendo sea menor al divisor y el residuo lo va guardando en la variable binario:
 - a. $47/8 = 5$ residuo 7 octal= “7”
 - b. $5/8 = 0$ residuo 5 octal= “57”
3. Resultado del número octal: 57

Posibles de errores de cálculos

Los cálculos a la hora de convertir un decimal a otro tipo numérico pueden tener errores de aproximación si el dato decimal es un número fraccionario, ya que en el programa se tiene un ciclo que limita las cifras decimales de conversión del número, para evitar que el número resultante sea muy grande o se generen ciclos infinitos, se estableció un límite de solo 10 cifras significativas.

Por ejemplo, si se ingresa el número decimal 0.4 el cual al convertir a octal se generan ciclos infinitos de conversión el programa nunca terminaría para estos casos se estableció un límite de cifras de 10 para evitar ciclos infinitos.

```
<<Convertidor de Numeros>>
¿Selecciona el tipo de dato a ingresar?:
1. Binario
2. Octal
3. Hexadecimal
4. Decimal
Digite su eleccion: 4
Convertir de Decimal a:
1. Binario
2. Octal
3. Hexadecimal
4. Volver al menu anterior
Ingrese la opcion que desea conlvertir: 2
Ingresar el dato a convertir: 0.4
El resultado es: 0.3146314631
```

Al implementar una forma de limitar las cifras decimales permite aumentar las cifras o disminuirlas dependiendo si es necesario comparar los datos con otra fuente.

```
precision = 10
```