I. Sistemas numéricos

#developed by **Roberto Angel Melendez-Armenta**

angelarmenta - Overview

angelarmenta has 6 repositories available. Follow their code on GitHub.

https://github.com/angelarmenta

1.1 Sistemas numéricos (binario, octal, decimal, hexadecimal)

Sistema decimal (base 10)

El **sistema decimal** es el más común en la vida diaria y es el que utilizamos para contar. Se llama **base 10** porque utiliza diez símbolos diferentes (del 0 al 9) para representar cualquier número. Cada dígito en un número decimal tiene un valor posicional que es una potencia de 10. Por ejemplo, el número decimal **3456** puede descomponerse de la siguiente manera:

$$3456 = (3 * 10^3) + (4 * 10^2) + (5 * 10^1) + (6 * 10^0)$$

Sistema binario (base 2)

El **sistema binario** es fundamental en computación, ya que las computadoras representan toda la información mediante dos estados: **1**(encendido) y **0** (apagado). En este sistema, solo se utilizan los dígitos **0** y **1**. Cada posición en un número binario representa una potencia de 2. El número binario **1011** puede descomponerse como:

$$1011 = (1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (1 * 2^0) = 11_{10}$$

Sistema octal (base 8)

El **sistema octal** utiliza ocho dígitos (del 0 al 7). Es menos común, pero se ha utilizado históricamente en la programación de ciertos sistemas. Cada dígito representa una potencia de 8. Por ejemplo, el número octal **145** puede descomponerse como:

$$145_8 = (1 * 8^2) + (4 * 8^1) + (5 * 8^0) = 101_{10}$$

Sistema hexadecimal (base 16)

El **sistema octal** utiliza dieciséis símbolos: los dígitos del 0 al 9 y las letras A-F, donde A representa el valor 10, B representa 11, y así sucesivamente hasta F, que representa 15. Este sistema es muy útil en programación y diseño de sistemas digitales. El número hexadecimal **2F** puede descomponerse como:

$$2F_{16} = (2 * 16^1) + (15 * 16^0) = 47_{10}$$

1.2 Conversiones entre sistemas numéricos

Decimal a binario

El método más común para convertir un número decimal a binario es mediante divisiones sucesivas por 2, guardando los restos.

Por ejemplo, para convertir el número decimal **13** a binario:

1. $\frac{13}{2} = 6$, resto 1
2. $\frac{6}{2} = 3$, resto 0
3. $\frac{3}{2} = 1$, resto 1
4. $\frac{1}{2} = 0$, resto 1

El número binario es **1101**.

Decimal a octal

Para convertir de decimal a octal, se utiliza el mismo procedimiento que con el binario, pero dividiendo sucesivamente entre 8.

Por ejemplo, para convertir el número decimal **83** a octal:

1. $\frac{83}{8} = 10$, resto 3
2. $\frac{10}{8} = 1$, resto 2
3. $\frac{1}{8} = 0$, resto 1

El número octal es **123**.

Decimal a hexadecimal

La conversión de decimal a hexadecimal sigue un proceso similar al de binario, pero dividiendo entre 16.

Por ejemplo, para convertir el número decimal **255** a hexadecimal:

1. $\frac{255}{16} = 15$, resto F
2. $\frac{15}{16} = 0$, resto F

El número hexadecimal es **FF**.

1.3 Operaciones básicas (suma, resta, multiplicación y división)

Operaciones básicas para números binarios

1. **Suma**

Las reglas básicas de la suma binaria son:

Binario	Resultado	Acarreo
$0 + 0$	0	0
$0 + 1$	1	0
$1 + 0$	1	0
$1 + 1$	0	1

La suma binaria sigue estas cuatro reglas. El acarreo se genera cuando la suma de dos dígitos es mayor o igual a la base (2, en este caso), y se "lleva" al siguiente dígito de la izquierda.

Ejemplo: sumar **1011** (11 en decimal) y **1101** (13 en decimal).

```
   1011   (11 en decimal)
 + 1101   (13 en decimal)
 -------
  11000   (24 en decimal)
```

2. **Resta**

Las reglas básicas de la resta binaria son:

Binario	Resultado	Préstamo
$0 - 0$	0	0

$1 - 0$	1	0
$1 - 1$	0	0
$0 - 1$	1	1

La resta binaria también sigue reglas específicas. Cuando intentamos restar 1 de 0, necesitamos "pedir prestado" del siguiente dígito más significativo, al igual que en el sistema decimal.

Ejemplo: restar **1011** (11 en decimal) y **1001** (9 en decimal).

```
  1011  (11 en decimal)
- 1001  (9 en decimal)
-------
  0010  (2 en decimal)
```

3. **Multiplicación**

Las reglas básicas de la multiplicación binaria son:

Binario	Resultado
$0 * 0$	0
$0 * 1$	0
$1 * 0$	0
$1 * 1$	1

La multiplicación binaria sigue un procedimiento similar al de la multiplicación en decimal. Sin embargo, dado que solo hay dos dígitos posibles (0 y 1), las reglas son más simples.

Ejemplo: multiplicar **101** (5 en decimal) y **11** (3 en decimal).

```
    101  (5 en decimal)
×    11  (3 en decimal)
-------
    101
 + 101
```

```
-------
   1111  (15 en decimal)
```

4. **División**

La división en binario sigue el mismo principio que la división en decimal. Se busca cuántas veces el divisor cabe en el dividendo, pero solo con los dígitos 0 y 1. Los pasos básicos de la división binaria son:

- **Compara el divisor con las primeras cifras del dividendo**: Si el divisor cabe, colocas un 1 en el cociente, y si no cabe, colocas un 0.
- **Baja el siguiente dígito del dividendo**: Repites este proceso hasta terminar con todos los dígitos del dividendo.
- **Resta el divisor del número actual**: Si el divisor cabe, realizas la resta y bajas el siguiente dígito.
- **Cociente y residuo**: El resultado de la división se da en forma de cociente y residuo.

Ejemplo: dividir **1101** (13 en decimal) entre **11** (3 en decimal).

- Comienza por ver cuántos dígitos del dividendo necesitas para que el divisor quepa.
  - El divisor tiene 2 dígitos (**11**), por lo que comenzamos comparando las dos primeras cifras del dividendo: **11** (las dos primeras cifras de **1101**).
  - **11** (3 en decimal) cabe una vez en **11**, así que colocamos un **1** en el cociente.
- Baja el siguiente dígito del dividendo (**0**), de manera que ahora tienes **00**.
  - **11** no cabe en **00**, por lo que colocamos un **0** en el cociente.
- Ahora baja el siguiente dígito del dividendo, que es **1**, formando **01**.
  - **11** sigue sin caber en **01**, por lo que colocamos otro **0** en el cociente.
  - Finalmente, baja el último dígito del dividendo, formando **101**.
  - **11** (3 en decimal) cabe una vez en **101** (5 en decimal), así que colocamos un **1** en el cociente.
  - Restamos **11** de **101**, lo que da un residuo de **10** (2 en decimal).

- El cociente es **100** en binario (que es **4** en decimal), y el residuo es **10** en binario (que es **2** en decimal).

Operaciones básicas para números octales

1. **Suma**

La suma en octal sigue reglas similares a la suma en decimal, pero cuando el resultado de una suma es 8 o mayor, debes llevar al siguiente dígito como un acarreo, ya que estamos en base 8. A continuación, se muestra el algoritmo para la suma en números octales.

```
Entrada: Dos números octales A y B
Salida: La suma de A y B en octal

Pasos:
        1. Alinea ambos números octales de derecha a izquierd
        los dígitos de la misma posición estén alineados.
        2. Suma los dígitos comenzando desde la derecha (posi
        3. Si la suma de los dos dígitos es menor que 8, escr
        de ellos.
        4. Si la suma es mayor o igual a 8, escribe la difere
        y acarrea 1 a la siguiente columna.
        5. Repite los pasos 2-4 para cada par de dígitos hast
        a la izquierda.
        6. Si hay un acarreo restante después de sumar el últ
        agrégalo como un dígito adicional a la izquierda del
```

2. **Resta**

La resta en octal también sigue un procedimiento similar a la resta en decimal. Si el número superior es menor que el número inferior en una columna, debemos pedir prestado de la siguiente columna a la izquierda.

```
Entrada: Dos números octales A y B, donde A > B
Salida: La resta de A menos B en octal

Pasos:
        1. Alinea ambos números octales de derecha a izquierd
```

```
        2. Resta los dígitos de derecha a izquierda.
        3. Si el dígito superior es mayor o igual que el infe
        resultado de la resta.
        4. Si el dígito superior es menor que el inferior, pi
        columna inmediatamente a la izquierda. Luego, suma 8 
        realiza la resta.
        5. Repite los pasos 2-4 para cada par de dígitos hast
        izquierda.
        6. Si hubo préstamos durante la resta, asegúrate de r
        columna.
```

3. **Multiplicación**

La multiplicación en octal también sigue las reglas básicas de la multiplicación, pero cualquier resultado que sea 8 o mayor debe dividirse entre 8, dejando el residuo y llevando el cociente como acarreo.

```
Entrada: Dos números octales A y B
Salida: El producto de A * B en octal

Pasos:
        1. Alinea ambos números de derecha a izquierda como l
        multiplicación decimal.
        2. Multiplica el primer dígito del número de abajo po
        número de arriba, comenzando de derecha a izquierda.
        3. Si el resultado es menor que 8, escribe el resulta
        a 8, divide el resultado por 8, escribe el residuo y 
        4. Repite el proceso con el siguiente dígito del núme
        los resultados una posición hacia la izquierda.
        5. Suma todos los resultados parciales obtenidos en e
        algoritmo de suma en octal.
        6. El resultado final es el producto en octal.
```

4. **División**

La división en octal sigue el mismo proceso que en el sistema decimal, pero los cálculos se hacen en base 8.

```
Entrada: Dos números octales A y B, donde A ≥ B
Salida: El cociente y residuo de A ÷ B en octal

Pasos:
        1. Toma el número octal A (dividendo) y B (divisor).
        2. Compara el divisor con las primeras cifras del div
        mayor que esas cifras, toma más dígitos del dividendo
        3. Divide las cifras seleccionadas del dividendo entr
        división en base 8 (similar a la división en decimal)
        4. Escribe el cociente de esta operación como el prim
        5. Multiplica el divisor por este cociente y resta el
        6. Baja el siguiente dígito del dividendo y repite el
        multiplicación y resta.
        7. Continúa el proceso hasta que no queden más dígito
        8. El cociente es el resultado de la división y lo qu
        el residuo.
```

Operaciones básicas para números hexadecimales

1. **Suma**

La suma en hexadecimal se realiza de manera similar a la suma en decimal. Si el resultado de la suma de dos dígitos es mayor o igual que 16, se acarrea 1 a la siguiente posición (igual que cuando sumamos en decimal y el resultado es 10 o mayor).

```
Entrada: Dos números hexadecimales A y B
Salida: La suma de A y B en hexadecimal

Pasos:
        1. Alinea ambos números hexadecimales de derecha a iz
        2. Comienza sumando los dígitos de derecha a izquierd
        3. Si la suma es menor que 16, escribe el resultado.
        4. Si la suma es mayor o igual a 16, convierte la sum
        (por ejemplo, 18 en decimal es 12 en hexadecimal), es
        resultante y lleva 1 como acarreo.
        5. Repite el proceso de suma para todos los dígitos, 
        si es necesario.
```

```
        6. Si hay un acarreo después de sumar el último par d
        como un dígito adicional a la izquierda del resultado
```

2. **Resta**

La resta en hexadecimal sigue el mismo procedimiento que en decimal. Si el dígito superior es menor que el dígito inferior, se debe pedir prestado de la columna siguiente. Cada préstamo en hexadecimal agrega 16 al dígito superior (en lugar de 10 como en decimal).

```
Entrada: Dos números hexadecimales A y B, donde A > B
Salida: La resta de A menos B en hexadecimal

Pasos:
        1. Alinea ambos números hexadecimales de derecha a iz
        2. Comienza restando los dígitos de derecha a izquier
        3. Si el dígito superior es mayor o igual que el infe
        normalmente.
        4. Si el dígito superior es menor que el inferior, pi
        siguiente dígito más significativo (igual que en la r
        Esto añade 16 al dígito superior antes de realizar la
        5. Repite los pasos 2-4 para todos los dígitos.
        6. Si hubo préstamos, ajusta los dígitos superiores p
        esos cambios.
```

3. **Multiplicación**

La multiplicación en hexadecimal es similar a la multiplicación en decimal. Multiplicamos los dígitos normalmente y si el producto es mayor o igual que 16, lo convertimos a hexadecimal y llevamos el acarreo a la siguiente posición.

```
Entrada: Dos números hexadecimales A y B
Salida: El producto de A * B en hexadecimal

Pasos:
        1. Alinea los números hexadecimales de derecha a izqu
        en la multiplicación decimal.
        2. Multiplica el primer dígito del número inferior po
        dígitos del número superior, de derecha a izquierda.
```

```
        3. Si el producto es mayor o igual que 16, convierte 
        hexadecimal y lleva el acarreo necesario.
        4. Multiplica el siguiente dígito del número inferior
        número superior, desplazando los resultados a la izqu
        5. Suma todos los productos parciales, usando el algo
        hexadecimal.
        6. El resultado final es el producto en hexadecimal.
```

4. **División**

La división en hexadecimal se realiza dividiendo el número de mayor valor (dividendo) entre el divisor, tal como en decimal. Los resultados se expresan en términos hexadecimales.

```
Entrada: Dos números hexadecimales A y B, donde A ≥ B
Salida: El cociente y residuo de A ÷ B en hexadecimal

Pasos:
        1. Compara el divisor con las primeras cifras del div
        2. Divide las cifras seleccionadas del dividendo entr
        la división en base 16.
        3. Escribe el cociente de esta operación como el prim
        4. Multiplica el divisor por este cociente y resta el
        5. Baja el siguiente dígito del dividendo y repite el
        multiplicación y resta.
        6. Continúa el proceso hasta que no queden más dígito
        7. El cociente es el resultado de la división y el re
        al final.
```

1.4 Aplicación de los sistemas numéricos en la computación

Los sistemas numéricos son esenciales en diversas áreas de la computación:

- **Programación y desarrollo de software**: Los sistemas binario y hexadecimal se utilizan para representar instrucciones de máquina y datos.
- **Redes de computadoras**: Las direcciones IP y las máscaras de subred se representan a menudo en binario.
- **Diseño de hardware y electrónica digital**: Los circuitos lógicos trabajan en base binaria.

Referencias

Arzate Hernández, E., Barjas Arzate, E. A. M. S., & Dhaity Dhaity, G. (2022). *Fundamentos de los sistemas numéricos*. Editorial Académica Española. ISBN: 978-3-330-09889-3.

Mano, M. M., & Kime, C. R. (2008). *Lógica digital y diseño de computadoras* (4.ª ed.). Pearson Prentice Hall.

Sebesta, R. W. (2016). *Conceptos de lenguajes de programación* (11.ª ed.). Pearson.

Tanenbaum, A. S., & Bos, H. (2015). *Arquitectura de computadoras: Un enfoque cuantitativo* (5.ª ed.). Morgan Kaufmann.

Tocci, R. J., Widmer, N. S., & Moss, G. L. (2011). *Sistemas digitales: Principios y aplicaciones* (11.ª ed.). Pearson Educación.