

1. Las computadoras y el sistema binario

El sistema binario es fundamental para el funcionamiento de las computadoras debido a su simplicidad y compatibilidad con los circuitos electrónicos. Este sistema utiliza solo dos estados, representados como 0 y 1, que corresponden a las condiciones de "apagado" y "encendido" de un interruptor eléctrico. Los microprocesadores están diseñados para interpretar estas secuencias binarias como instrucciones o datos.

1.1. Aplicaciones del sistema binario en la tecnología informática

El uso del sistema binario no se limita únicamente al diseño de microprocesadores; también es esencial en diversas áreas de la computación:

1. Memorias de almacenamiento:

- **Memorias RAM y ROM:** Las celdas de memoria almacenan datos en forma de bits (0 o 1), organizados en palabras de longitud fija (por ejemplo, 8, 16 o 32 bits).
- **Soportes ópticos:** En discos compactos (CD) y discos versátiles digitales (DVD), los datos se representan mediante pits y lands (pequeñas hendiduras y superficies planas) que se leen como 0s y 1s.
- **Soportes magnéticos:** En discos duros y cintas magnéticas, los datos se almacenan mediante la magnetización de partículas en dos direcciones, que representan los dos estados binarios.

2. **Buses de comunicación:** Los buses que transportan datos entre el procesador, la memoria y otros componentes utilizan señales binarias para codificar la información que fluye dentro de la computadora.

3. Redes y transmisión de datos:

- Los datos que viajan por las redes se convierten en secuencias binarias, transmitidas como pulsos eléctricos, señales ópticas o radiofrecuencia, según la tecnología utilizada.

- Los protocolos como TCP/IP estructuran y organizan estas secuencias para garantizar la transmisión eficiente y segura de información.

4. Interacción con periféricos:

- Los dispositivos como teclados, pantallas, impresoras y discos externos se comunican con la computadora utilizando instrucciones y datos en formato binario.

1.2. El sistema binario y los tipos de datos en los lenguajes de programación

Como programadores, es fundamental comprender que el sistema binario es la base del almacenamiento y procesamiento de datos en las computadoras. Esto impacta directamente en la forma en que trabajamos con distintos tipos de datos:

1. Enteros:

- Representados como `int` en la mayoría de los lenguajes de programación. Por ejemplo, un entero de 8 bits puede almacenar valores del 0 al 255, mientras que uno de 16 bits puede alcanzar valores hasta 65,535.
- También se emplea para enteros con signo (positivo y negativo) mediante la representación de complemento a dos.

2. Números en punto flotante:

- Utilizados para representar números decimales, como `float` o `double`. Por ejemplo, un número en precisión simple ocupa 32 bits, distribuidos entre un bit para el signo, 8 bits para el exponente y 23 bits para la mantisa.

3. Caracteres:

- Los sistemas ASCII y Unicode mapean caracteres a valores numéricos binarios. Por ejemplo:

- El carácter 'A' corresponde al valor 65 en ASCII (01000001 en binario).
- En Unicode, el carácter 'A' es U+0041.

4. Colores:

- En gráficos y diseño web, los colores se representan mediante el modelo RGB. Cada canal (Rojo, Verde, Azul) se almacena como un valor entre 0 y 255, codificado en binario. Por ejemplo, el color rojo puro (255, 0, 0) sería 11111111 00000000 00000000 en binario.

1.3 Ventajas del sistema binario en la computación

- **Simplicidad:** Facilita la implementación en circuitos digitales mediante puertas lógicas.
- **Fiabilidad:** Reduce los riesgos de error en la transmisión y almacenamiento de datos.
- **Compatibilidad:** Es adaptable a tecnologías como los transistores, que son la base de las computadoras modernas.

1.4. Unidades de medida en el sistema binario

El sistema binario, base del funcionamiento de las computadoras, también define cómo se miden y representan los datos en este entorno. Estas medidas, basadas en múltiplos del bit, son fundamentales para entender el almacenamiento, la transferencia y el procesamiento de información en una computadora. Así como el binario permite la representación de datos, las unidades derivadas de él son esenciales para cuantificar y gestionar esos datos en hardware y software.

1.4.1. Unidades de medida en el sistema binario

Los datos se miden en múltiplos del **bit**, la unidad básica de información en binario, donde un bit puede representar un 0 o un 1. A partir de esta unidad elemental, se derivan otras más grandes:

- **1 Byte (B) = 8 bits:** Un byte puede representar 256 valores diferentes (0 a 255 en decimal).
- **1 Kilobyte (KB) = 1024 Bytes:** Utilizado para medir pequeños archivos, como documentos de texto.
- **1 Megabyte (MB) = 1024 KB:** Adecuado para archivos de mayor tamaño, como imágenes y canciones.
- **1 Gigabyte (GB) = 1024 MB:** Común en discos duros y memorias USB.
- **1 Terabyte (TB) = 1024 GB:** Utilizado para medir la capacidad de almacenamiento de discos duros de gran tamaño.

1.4.2. Ejemplos de uso práctico

- **RAM:** Las computadoras modernas suelen contar con memorias RAM de entre 8 GB y 32 GB, donde la RAM permite almacenar datos temporales para su rápido acceso.
- **Discos duros:** Los discos SSD (Solid State Drive) más comunes tienen capacidades de 512 GB o 1 TB, adecuadas para almacenar sistemas operativos, aplicaciones y datos.
- **Velocidades de transferencia:** Las conexiones de internet, por ejemplo, suelen expresarse en megabits por segundo (Mbps), indicando la cantidad de datos que pueden transmitirse en un segundo.

1.4.3. Conversión entre decimal y binario

El sistema binario utiliza solo dos dígitos, 0 y 1, mientras que el decimal utiliza diez dígitos, del 0 al 9. A continuación, se explican los pasos básicos para convertir números entre ambos sistemas:

1. **De decimal a binario:** Divide el número decimal entre 2 y anota el residuo. Luego, divide el cociente obtenido nuevamente entre 2, registrando los residuos en cada paso. Repite el proceso hasta que el cociente sea 0. Los residuos leídos de abajo hacia arriba forman el número binario.

- Ejemplo: Convertir 13 (decimal) a binario.
 - $13 \div 2 = 6$, residuo = 1
 - $6 \div 2 = 3$, residuo = 0
 - $3 \div 2 = 1$, residuo = 1
 - $1 \div 2 = 0$, residuo = 1 Resultado: **1101**

2. **De binario a decimal:** Multiplica cada dígito binario por 2 elevado a la posición que ocupa, comenzando desde 0 (de derecha a izquierda), y suma los resultados.

- Ejemplo: Convertir 1101 (binario) a decimal.
 - $1 \times 2^3 = 8$
 - $1 \times 2^2 = 4$
 - $0 \times 2^1 = 0$
 - $1 \times 2^0 = 1$ Resultado: $8 + 4 + 0 + 1 = 13$ (decimal).

Para entender cómo convertir un número binario (base 2) a decimal (base 10), podemos hacer la siguiente analogía con el sistema decimal:

1. Sistema decimal (base 10)

En el sistema decimal, cada cifra (de derecha a izquierda) tiene un valor de

- $10^0 = 1$
 - $10^1 = 10$
 - $10^2 = 100$
 - $10^3 = 1000$
- y así sucesivamente.

Por ejemplo, el número 342 en decimal se descompone como:

$$342=3\times 100+4\times 10+2\times 1.$$

2. Sistema binario (base 2)

En el sistema binario, cada dígito (bit) puede ser 0 o 1, y cada posición (de derecha a izquierda) representa potencias de 2, de la siguiente manera:

- $2^0=1$
- $2^1=2$
- $2^2=4$
- $2^3=8$
- $2^4=16$

y así sucesivamente.

Si tomamos como ejemplo el número binario 1101, lo interpretamos así, de derecha a izquierda:

- El primer dígito (1) vale $2^0=1$.
- El segundo dígito (0) vale $2^1=2$, pero está multiplicado por 0, así que aporta 0.
- El tercer dígito (1) vale $2^2=4$.
- El cuarto dígito (1) vale $2^3=8$.

Entonces, 1101 en binario se convierte a decimal como:

$$8\times 1+4\times 1+2\times 0+1\times 1=8+4+0+1=13.$$

De esta manera, en el sistema binario cada posición vale una potencia de 2, al igual que en el sistema decimal cada posición vale una potencia de 10.