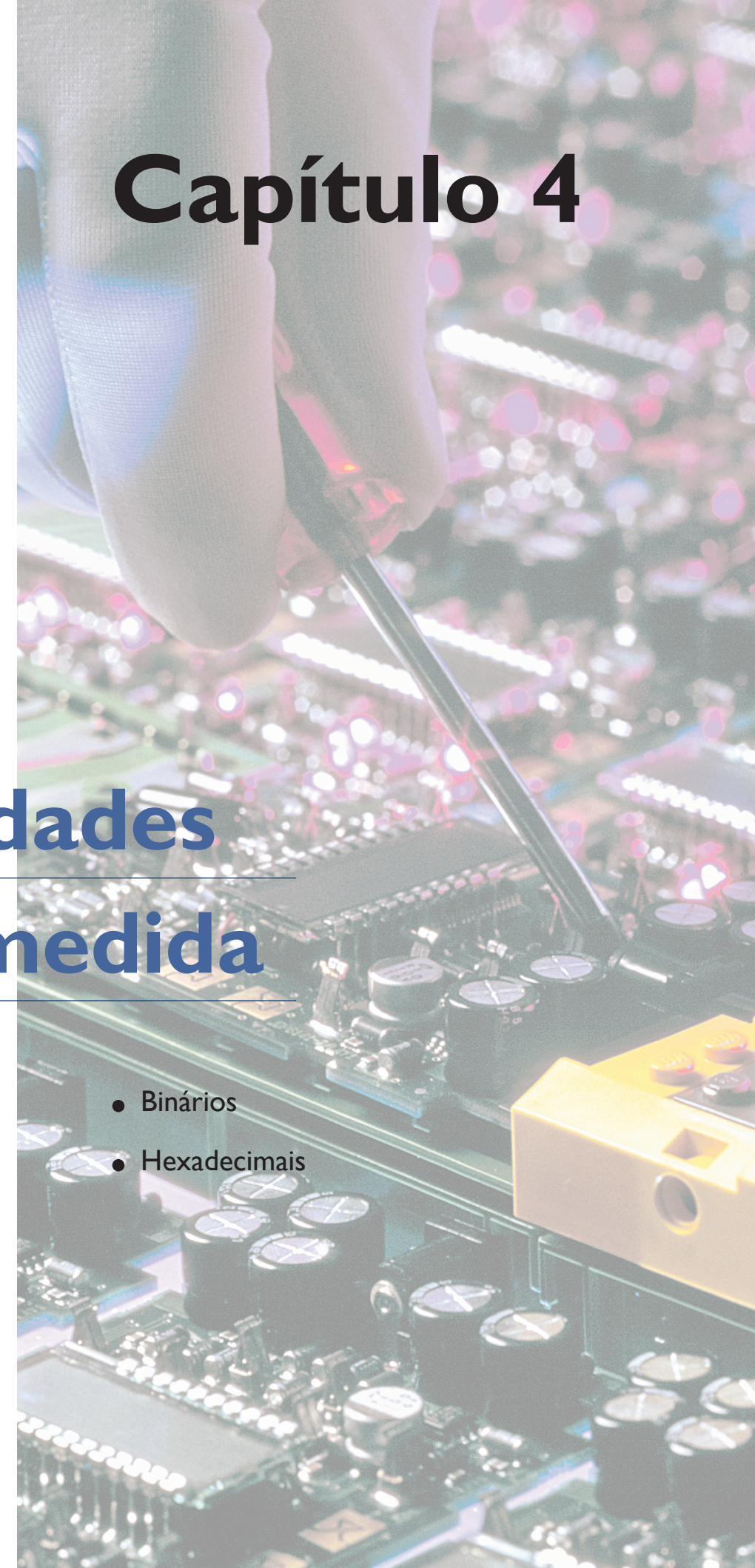


Capítulo 4

Unidades de medida

- Binários
- Hexadecimais



Para representar o mundo real utilizamos unidades de medida. Para medir o tempo empregamos horas. Para medir pequenas distâncias, metros, e mais longas, quilômetros. Para medir volume de líquidos, recorremos a litros, etc. Em computadores utilizamos bits para medir o tamanho das informações. Isso é importante, por exemplo, quando precisamos saber quantas músicas caberão no CD, qual é o espaço livre no HD ou ainda se a foto é muito grande para ser enviada por e-mail.

O bit é a menor porção de informação possível em informática. Um único bit representa somente duas informações, 0 ou 1, parecendo pouco em relação a outros formatos como o decimal, em que 1 dígito pode representar 10 valores diferentes, de 0 a 9. Mas essa representação é inerente à maneira como as informações podem ser escritas e lidas pelo computador. No início, utilizavam-se cartões de papel para armazenar informações (veja figura 11 e o quadro *Informática do século 19*). Um furo em determinada posição do papel representava a informação 1. Ausência de furos significava 0. Essa mesma lógica pode ser utilizada depois nos computadores digitais, que utilizam sinais elétricos de tensões diferentes, baixa e alta, para identificar o valor 0 e 1. Mídias magnéticas podem armazenar essas informações substituindo o furo do cartão por um ponto que pode ser ou não magnetizado.

4.1. Binários

Aprendemos a enxergar o mundo em forma decimal, e portanto temos dificuldade para compreender e nos comunicar utilizando a forma binária. Já imaginou alguém dizendo que sua idade é 1111? É o mesmo que 15, em formato decimal. Portanto, precisamos compreender como ler um número binário e transformá-lo em decimal para conhecer o valor da informação.

Quando aparece sozinho, o número 0 vale 0 em decimal e também em binário, assim como o número 1, que vale 1 em ambos os sistemas. Mas o binário 10 não vale 10 em decimal, e sim 2. O número vai aumentando à medida que se coloca mais dígitos à esquerda, como no formato decimal, em que, conforme aumentamos uma casa, o número cresce na base de 10. Por exemplo: o número 1 na primeira casa mais à direita vale 1 mesmo, mas se estiver na segunda casa, vale 10, na terceira, 100 e assim por diante. O sistema binário funciona na base 2 – o dígito mais à direita vale no máximo 1,

da segunda vale 2 (10), da terceira vale 4 (100). Ou seja: para converter um binário em decimal, multiplicamos dígito a dígito pela base 2 elevando-o à potência de sua posição. Exemplo:

| CÁLCULO DE BINÁRIOS | | |
|---------------------|--|---------|
| Binario | Cálculo | Decimal |
| 0 | 0×2^0 | 0 |
| 1 | 1×2^0 | 1 |
| 10 | $1 \times 2^1 + 0 \times 2^0$ | 2 |
| 11 | $1 \times 2^1 + 1 \times 2^0$ | 3 |
| 100 | $1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$ | 4 |
| 101 | $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ | 5 |
| 110 | $1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$ | 6 |
| 111 | $1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$ | 7 |

Se para converter binário em decimal utilizamos a multiplicação, para decompor decimais em binários recorremos à divisão, que é a função inversa. E como inverso da potência é raiz quadrada, chegamos ao valor binário agregando os dígitos obtidos nos restos de divisões sucessivas pelo número 2 (base binária), a partir do último resultado até o da primeira divisão. Acompanhe na figura 12.

Informática no século 19

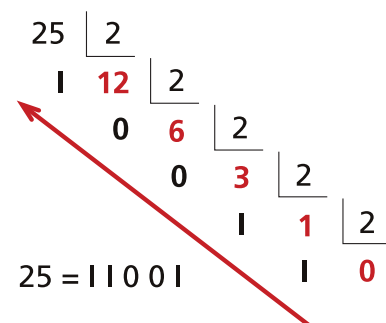
A origem dos cartões perfurados remonta aos Estados Unidos do final do século 19 e à criação da gigante da informática IBM. A técnica foi concebida por Herman Hollerith, nascido em 1860, que partiu do princípio de comando de teares automáticos. Em 1880, Hollerith trabalhava no National Census Office, que fazia pesquisas demográficas e levava 10 anos para tabular as informações. Naquele ano ele inventou uma máquina leitora de cartões perfurados em código binário. No início, Hollerith usou fitas de papel perfuradas divididas em campos para cada grupo da população. Depois adotou cartões para cada indivíduo. Hollerith conseguiu suas primeiras patentes em 1884 e continuou aperfeiçoando o sistema. Começou processando dados estatísticos de saúde pública e fazendo levantamentos para a administração do exército. A consolidação do empreendimento viria em 1889, quando sua empresa venceu a concorrência do United States Census Bureau para fornecer o sistema processador do censo do ano seguinte. Em 1911, a empresa de Hollerith juntou-se a outras três corporações para formar a Computing Tabulating Recording Corporation, que depois passou a se chamar IBM.



Figura 11
Imagem de um cartão perfurado.

Figura 12

Cálculo do valor binário.



Entre os programadores mais antigos era comum o termo nibble. Um nibble representa meio byte (4 bits), a quantidade de bits que o sistema de codificação BCD (Binary-coded decimal ou codificação binário decimal) usa para representar valores de 0 a 15. O nibble é largamente utilizado em sistemas digitais.

Capacidade

Não é comum usar bits para representar a capacidade de módulos de memória ou de um disco rígido: o padrão é o **byte**, também chamado de octeto. O byte é formado por 8 bits, que representam um único caractere na tabela ASCII (American Standard Code for Information Interchange – código padrão americano para troca de informações). Por isso passou a ser empregado – sabia-se que um arquivo de 50 bytes continha um texto de aproximadamente 50 letras. A ASCII é uma tabela de códigos de 8 bits que representam 128 caracteres com base no alfabeto da língua inglesa.

Múltiplos

Sempre que precisamos representar números muito grandes ou muito pequenos, costumamos utilizar múltiplos para simplificar. Por exemplo, quando queremos falar de 1000 metros, dizemos simplesmente 1 quilômetro. Essas grandezas são baseadas em uma potência. Metro é 10^3 de quilômetro. Ou seja, a cada número de potência temos um prefixo para identificar seu grau de simplificação.

Os números decimais levam prefixos baseados em letras gregas, chamados de greco-latinos. Veja a tabela de relacionamento abaixo:

| SIMBOLOGIA DOS NÚMEROS DECIMAIS | | |
|---------------------------------|---------|-----------|
| Prefixo | Símbolo | Fator |
| Exa | E | 10^{18} |
| peta | P | 10^{15} |
| tera | T | 10^{12} |
| giga | G | 10^9 |
| mega | M | 10^6 |
| quilo | k | 10^3 |
| hecto | h | 10^2 |
| deca | da | 10 |
| deci | d | 10^{-1} |
| centi | c | 10^{-2} |

Apesar de essa notação ser baseada em números decimais (note que a base é 10^x), ela costuma ser utilizada também para reproduzir grandezas binárias que são baseadas em 2. Isso para que se possa compreender melhor as quantidades que estão sendo expressas.

Existe uma notação baseada em 2 (binária), desenvolvida pela IEC (International Electrotechnical Commission ou Comissão Eletrotécnica Internacional), uma empresa Suíça, que desenvolve padrões elétricos e eletrônicos.

| MÚLTIPLOS DE BYTES | | | | | |
|-----------------------|---------|----------|---------------|---------|-----------|
| Prefixo binário (IEC) | | | Prefixo do SI | | |
| Nome | Símbolo | Múltiplo | Nome | Símbolo | Múltiplo |
| byte | B | 2^0 | byte | B | 10^0 |
| kibibyte(kilobyte) | KiB | 2^{10} | kilobyte | kB | 10^3 |
| mebibyte(megabyte) | MiB | 2^{20} | megabyte | MB | 10^6 |
| gibibyte (gigabyte) | GiB | 2^{30} | gigabyte | GB | 10^9 |
| tebibyte(terabyte) | TiB | 2^{40} | terabyte | TB | 10^{12} |
| pebibyte(petabyte) | PiB | 2^{50} | petabyte | PB | 10^{15} |
| exbibyte(exabyte) | EiB | 2^{60} | exabyte | EB | 10^{18} |
| zebibyte(zettabyte) | EiB | 2^{70} | zettabyte | ZB | 10^{21} |
| yobibyte(yottabyte) | YiB | 2^{80} | yottabyte | YB | 10^{24} |

4.1.1. Hexadecimais

A base hexadecimal foi adotada para facilitar a representação de números binários. Os dígitos hexadecimais vão de 0 a F (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D e F) e podem representar os números por meio de menor quantidade de dígitos.

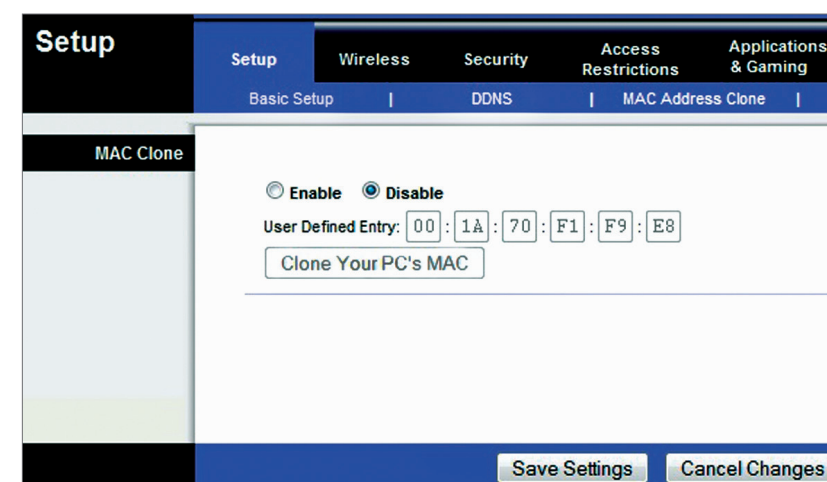
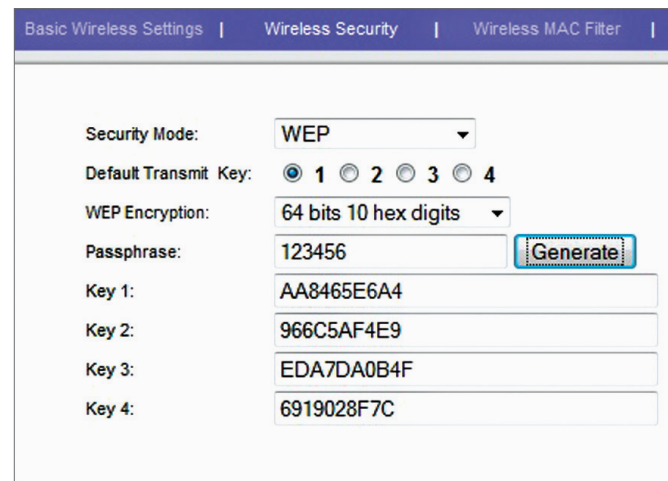


Figura 13

Tela administrativa de um roteador – Endereços MAC.

Figura 14
Definição de senhas de rede sem fio.



| TABELA DE CONVERSÃO DE NÚMEROS | | | |
|--------------------------------|---------|-------|-------------|
| Decimal | Binário | Octal | Hexadecimal |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |

Por exemplo: o número binário 111110101000 em hexadecimal é representado por FA8, e equivale a 4008 em decimal. Cada dígito hexadecimal sempre representa 4 bits. Para diferenciar a representação de um número decimal de um binário, e evitar confusão, foi convencionado que os números da base 16 (hexadecimais) teriam um \$ na frente e os binários um b. Assim, podemos diferenciar, b10, \$10 e 10, que valem 2, 16 e 10, consecutivamente.

No dia a dia o técnico em informática encontrará números hexadecimais em endereços MAC de equipamentos de rede, senhas de redes sem fio (figuras 13 e 14), números de série, assinatura digital, para definir cores para componentes em vários tipos de linguagens de programação, números de IPv6 etc.

A forma de conversão de números decimais em hexadecimais é idêntica à de binários. Do mesmo modo que dividimos o número decimal por 2, em hexadecimal dividimos por 16. E vamos fazendo divisões consecutivas até que o resultado inteiro seja 0. O número hexadecimal será igual ao formado pela agregação dos restos das divisões, a partir da última para a primeira.

| TABELA DE CONVERSÃO DE NÚMEROS | | | |
|--------------------------------|---------|-------|-------------|
| Decimal | Binário | Octal | Hexadecimal |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 10 | 2 | 2 |
| 3 | 11 | 3 | 3 |
| 4 | 100 | 4 | 4 |
| 5 | 101 | 5 | 5 |
| 6 | 110 | 6 | 6 |
| 7 | 111 | 7 | 7 |