



Para saber mais: UTF-8, UTF-16 e UTF-32

Vimos a codificação UTF-8 para representar os vários caracteres do sistema Unicode, mas essa não é a única codificação para isso. Existem outros dois sistemas UTF-16 e UTF-32. Vamos ver a diferença entre esses três.

Atenção: todos os três sistemas conseguem codificar todos os caracteres do Unicode. Apenas a quantidade de bits que eles usarão para fazer isso que muda.

UTF-8

Essa codificação representa os caracteres em pedaços de **8 bits** e sua grande vantagem é manter os textos codificados apenas em ASCII (a grande maioria da Web no passado) intactos. Ele tem um tamanho variável e cada caractere pode ocupar 8, 16, 24 ou 32 bits.

Essa é a codificação mais comum na web hoje.

UTF-16

Essa codificação representa os caracteres em pedaços de **16 bits**. Isso significa que a codificação não é mais compatível com ASCII e ocupa o dobro de memória em textos que possuem apenas caracteres da língua inglesa. Ele tem um tamanho variável e cada caractere pode ocupar 16 ou 32 bits.

Sua grande vantagem é ocupar menos espaço quando o texto possui muitos caracteres asiáticos (UTF-8 usaria 3 bits por caractere e UTF-16 apenas 2). Mesmo

assim, essa vantagem é [bem questionada \(http://utf8everywhere.org/\)](http://utf8everywhere.org/) e muitas pessoas não recomendam o seu uso em muitos casos.

Essa é a [codificação usada em sistemas Windows \(https://docs.microsoft.com/pt-br/windows/win32/intl/unicode?redirectedfrom=MSDN\)](https://docs.microsoft.com/pt-br/windows/win32/intl/unicode?redirectedfrom=MSDN).

UTF-32

Essa codificação representa os caracteres em pedaços de **32 bits**. Ela não é compatível com ASCII e ocupa **4 vezes** mais espaço em textos que só utilizavam ASCII. Mesmo assim, diferentemente do UTF-8 e UTF-16, essa codificação tem um **tamanho fixo** e essa é sua grande vantagem. Como cada caractere ocupa a mesma quantidade de bits, é fácil saber em qual posição cada caractere está em um texto (é só pegar o índice do caractere e multiplicar por 32).