

Próximo:[Contras Tipo de Célula](#), Anterior:[Tipo de símbolo](#), Acima:[Tipos de programação](#) [[Conteúdo](#)][[Índice](#)]

2.4.5 Tipos de Sequência

Uma *sequência* é um objeto Lisp que representa um conjunto ordenado de elementos. Existem dois tipos de sequência no Emacs Lisp: *listas* e *arrays* .

As listas são as sequências mais usadas. Uma lista pode conter elementos de qualquer tipo e seu comprimento pode ser facilmente alterado adicionando ou removendo elementos. Veja a próxima subseção para mais informações sobre listas.

Arrays são sequências de comprimento fixo. Eles são subdivididos em strings, vetores, char-tables e bool-vectors. Os vetores podem conter elementos de qualquer tipo, enquanto os elementos string devem ser caracteres e os elementos bool-vector devem ser `nil`. Tabelas de char são como vetores, exceto que são indexadas por qualquer código de caractere válido. Os caracteres em uma string podem ter propriedades de texto como caracteres em um buffer (consulte [Propriedades de texto](#)), mas os vetores não suportam propriedades de texto, mesmo quando seus elementos são caracteres.

Listas, strings e outros tipos de array também compartilham semelhanças importantes. Por exemplo, todos têm um comprimento *l* e todos têm elementos que podem ser indexados de zero a *l* menos um. Várias funções, chamadas funções de sequência, aceitam qualquer tipo de sequência. Por exemplo, a função `length` relata o comprimento de qualquer tipo de sequência. Veja [Sequências Arrays Vetores](#) .

Geralmente é impossível ler a mesma sequência duas vezes, uma vez que as sequências são sempre criadas de novo na leitura. Se você ler a sintaxe de leitura de uma sequência duas vezes, obterá duas sequências com conteúdo igual. Há uma exceção: a lista vazia `()` sempre representa o mesmo objeto, `nil`.