

Próximo:[Registros](#), Anterior:[Listas](#), Acima:[Topo](#) [Conteúdo][Índice]

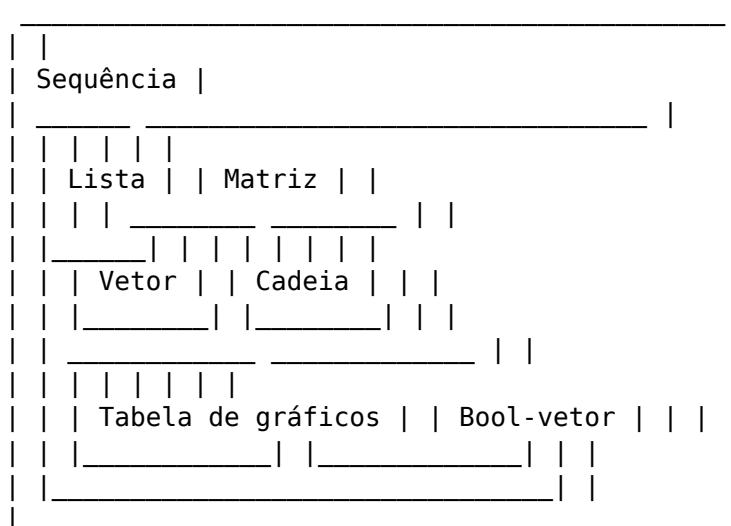
6 Sequências, Matrizes e Vetores

O tipo de *sequência* é a união de dois outros tipos de Lisp: listas e arrays. Em outras palavras, qualquer lista é uma sequência e qualquer array é uma sequência. A propriedade comum que todas as sequências têm é que cada uma é uma coleção ordenada de elementos.

Um *array* é um objeto de tamanho fixo com um slot para cada um de seus elementos. Todos os elementos são acessíveis em tempo constante. Os quatro tipos de arrays são strings, vetores, char-tables e bool-vectors.

Uma lista é uma sequência de elementos, mas não é um único objeto primitivo; é feito de células contra, uma célula por elemento. Encontrar o n -ésimo elemento requer olhar através de n cons células, então os elementos mais distantes do início da lista demoram mais para serem acessados. Mas é possível adicionar elementos à lista ou remover elementos.

O diagrama a seguir mostra a relação entre esses tipos:



- [Funções de Sequência](#) Funções que aceitam qualquer tipo de sequência.
- [Matrizes](#) Características de arrays em Emacs Lisp.
- [Funções de matriz](#) Funções específicas para arrays.
- [Vetores](#) Características especiais dos vetores Emacs Lisp.
- [Funções Vetoriais](#) Funções específicas para vetores.
- [Tabelas de Caracteres](#) Como trabalhar com tabelas de caracteres.
- [Bool-Vetores](#) Como trabalhar com vetores bool.
- [Anéis](#) Gerenciando um anel de objetos de tamanho fixo.

Próximo:[Registros](#), Anterior:[Listas](#), Acima:[Topo](#) [Conteúdo][Índice]

