

Próximo:[Tipo de matriz](#), Anterior:[Tipo de sequência](#), Acima:[Tipos de programação](#) [Conteúdo][Índice]

2.4.6 Tipos de Célula e Lista de Contras

Uma *célula de contras* é um objeto que consiste em dois slots, chamados slot CAR e slot CDR . Cada slot pode *conter* qualquer objeto Lisp. Também dizemos que o CAR desta célula cons é qualquer objeto que seu slot CAR contenha atualmente, e da mesma forma para o CDR .

Uma *lista* é uma série de células de contras, ligadas entre si de modo que o slot CDR de cada célula de contras contenha a próxima célula de contras ou a lista vazia. A lista vazia é na verdade o símbolo *nil*. Consulte [Listas](#) , para obter detalhes. Como a maioria das células cons são usadas como parte de listas, nos referimos a qualquer estrutura feita de células cons como uma *estrutura de lista* .

Uma nota para os programadores de C: uma lista Lisp funciona assim como uma *lista encadeada* construída de células cons. Como os ponteiros em Lisp são implícitos, não distinguimos entre um slot de célula contra que contém um valor versus apontar para o valor.

Como as células cons são tão centrais para o Lisp, também temos uma palavra para um objeto que não é uma célula cons. Esses objetos são chamados de *átomos* .

A sintaxe de leitura e a representação impressa para listas são idênticas e consistem em um parêntese esquerdo, um número arbitrário de elementos e um parêntese direito. Seguem exemplos de listas:

```
(A 2 "A"); Uma lista de três elementos.
(); Uma lista sem elementos (a lista vazia).
nada ; Uma lista sem elementos (a lista vazia).
("UMA ()"); Uma lista de um elemento: a string "A ()".
(UMA ()); Uma lista de dois elementos: A e a lista vazia.
(Um nulo); Equivalente ao anterior.
((ABC)); Uma lista de um elemento
; (que é uma lista de três elementos).
```

Ao ler, cada objeto dentro dos parênteses torna-se um elemento da lista. Ou seja, uma célula de contras é feita para cada elemento. O slot CAR da célula cons contém o elemento e seu slot CDR se refere à próxima célula cons da lista, que contém o próximo elemento da lista. O slot CDR da última célula contra está configurado para manter *nil* .

Os nomes CAR e CDR derivam da história do Lisp. A implementação original do Lisp rodava em um computador IBM 704 que dividia as palavras em duas partes, o endereço e o decremento; CAR era uma instrução para extrair o conteúdo da parte de endereço de um registro, e CDR uma instrução para extrair o conteúdo do decremento. Por outro lado, as células cons são nomeadas pela função *cons* que as cria, que por sua vez foi nomeada por seu propósito, a construção de células.

- [Diagramas de Caixa](#) Desenho de listas.
- [Notação de par pontilhado](#) Uma sintaxe geral para células contras.
- [Tipo de Lista de Associação](#) Uma lista especialmente construída.

Próximo:[Tipo de matriz](#), Anterior:[Tipo de sequência](#), Acima:[Tipos de programação](#) [Conteúdo][Índice]

