

Próximo:[Formulários Especiais](#), Anterior:[Formulários de função](#), Acima:[Formulários](#) [[Conteúdo](#)][[Índice](#)]
]

10.2.6 Avaliação de Macro Lisp

Se o primeiro elemento de uma lista que está sendo avaliada for um objeto de macro, a lista será uma *chamada de macro*. Quando uma chamada de macro é avaliada, os elementos do restante da lista *não* são avaliados inicialmente. Em vez disso, esses próprios elementos são usados como argumentos da macro. A definição da macro calcula uma forma de substituição, chamada de *expansão* da macro, a ser avaliada no lugar da forma original. A expansão pode ser de qualquer forma: uma constante auto-avaliadora, um símbolo ou uma lista. Se a expansão for em si uma chamada de macro, esse processo de expansão se repete até que algum outro tipo de forma resulte.

A avaliação comum de uma chamada de macro termina avaliando a expansão. No entanto, a macro expansão não é necessariamente avaliada de imediato, ou de todo, porque outros programas também expandem as chamadas de macro, podendo ou não avaliar as expansões.

Normalmente, as expressões de argumento não são avaliadas como parte do cálculo da expansão da macro, mas aparecem como parte da expansão, portanto, são calculadas quando a expansão é avaliada.

Por exemplo, dada uma macro definida da seguinte forma:

```
(defmacro cadr (x)
  (lista 'carro (lista 'cdr x)))
```

uma expressão como `(cadr (assq 'handler list))` é uma chamada de macro e sua expansão é:

```
(carro (cdr (assq 'lista de manipuladores)))
```

Observe que o argumento `(assq 'handler list)` aparece na expansão.

Veja [Macros](#), para uma descrição completa das macros do Emacs Lisp.