

Próximo:[Iteração](#), Anterior:[Combinando Condições](#), Acima:[Estruturas de controle](#) [[Conteúdo](#)][[Índice](#)]

11.4 Condicional de correspondência de padrões

Além das quatro formas condicionais básicas, o Emacs Lisp também possui uma forma condicional de correspondência de padrões, a `pcasemacro`, um híbrido de `cond` e `cl-case` (consulte [Condicionais](#) em extensões de Lisp comuns) que supera suas limitações e introduz o *estilo de programação de correspondência de padrões* . As limitações que `pcases` supera são:

- A `cond` escolhe entre alternativas avaliando a *condição* de predicado de cada uma de suas cláusulas (veja [Condicionais](#)). A principal limitação é que as variáveis `let-bound` in *condition* não estão disponíveis para os *body-forms* da cláusula .

Outro incômodo (mais um inconveniente do que uma limitação) é que quando uma série de predicados de *condição* implementam testes de igualdade, há muito código repetido. (`cl-case` resolve esse inconveniente.)

- A `cl-casemacro` escolhe entre alternativas avaliando a igualdade de seu primeiro argumento em relação a um conjunto de valores específicos.

Suas limitações são duplas:

1. Os testes de igualdade usam `eq`.
2. Os valores devem ser conhecidos e escritos com antecedência.

Eles se tornam `cl-case` inadequados para strings ou estruturas de dados compostas (por exemplo, listas ou vetores). (`cond` não tem essas limitações, mas tem outras, veja acima.)

Conceitualmente, a `pcasemacro` toma emprestado o foco do primeiro argumento de `cl-case` e o fluxo de processamento de cláusula de `cond`, substituindo *condição* por uma generalização do teste de igualdade que é uma variante de *correspondência de padrões* e adicionando recursos para que você possa expressar de forma concisa o predicado de uma cláusula e organize para compartilhar ligações `let` entre o predicado de uma cláusula e as *formas do corpo* .

A expressão concisa de um predicado é conhecida como *padrão* . Quando o predicado, chamado no valor do primeiro arg, retorna não-`nil`, dizemos que “o padrão corresponde ao valor” (ou às vezes “o valor corresponde ao padrão”).

- | | |
|--|---|
| • A <code>pcasemacro</code> | Inclui exemplos e advertências. |
| • Extensão <code>pcase</code> | Defina novos tipos de padrões. |
| • Padrões de estilo <code>backquote</code> | Correspondência de padrões estruturais. |
| • Desestruturando com Padrões <code>pcase</code> | Usando padrões <code>pcase</code> para extrair subcampos. |

Próximo:[Iteração](#), Anterior:[Combinando Condições](#), Acima:[Estruturas de controle](#) [[Conteúdo](#)][[Índice](#)]