

Próximo:[Padrões de aspas](#), Anterior:[pcase Macro](#), Acima:[Condicional de correspondência de padrões](#) [[Conteúdo](#)][[Índice](#)]

11.4.2 Extensão `pcase`

A `pcasemacro` suporta vários tipos de padrões (consulte [Condicional de correspondência de padrões](#)). Você pode adicionar suporte para outros tipos de padrões usando a `pcase-defmacro`.

Macro: `pcase-defmacro name args [doc] &rest body`

Defina um novo tipo de padrão para `pcase`, a ser invocado como `.`. A macro expande isso em uma chamada de função que avalia *body*, cujo trabalho é reescrever o padrão invocado em algum outro padrão, em um ambiente onde *args* está vinculado a *real-args*. (*name actual-args*) `pcase`

Além disso, organize para exibir *doc* junto com a docstring de `pcase`. Por convenção, *doc* deve usar `EXPVAL` para representar o resultado da *expressão* de avaliação (primeiro arg para `pcase`).

Normalmente, *body* reescreve o padrão invocado para usar padrões mais básicos. Embora todos os padrões eventualmente se reduzam a padrões centrais, *body* não é necessário usar padrões centrais imediatamente. O exemplo a seguir define dois padrões, nomeados `less-than` e `integer-less-than`.

```
(pcase-defmacro menor que (n)
  "Corresponde se EXPVAL for um número menor que N."
  `(pred (> ,n)))

(pcase-defmacro inteiro-menor que (n)
  "Corresponde se EXPVAL for um número inteiro menor que N."
  `(and (pred integerp)
        (menor que ,n)))
```

Observe que as docstrings mencionam *args* (neste caso, apenas um: *n*) da maneira usual, e também mencionam `EXPVAL` por convenção. A primeira reescrita (ou seja, *corpo* para `less-than`) usa um padrão central: `pred`. O segundo usa dois padrões principais: `and` e `pred`, assim como o padrão recém-definido `less-than`. Ambos usam uma única construção de backquote (consulte [Backquote](#)).