

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

11.4.4 Desestruturando com pcasePadrões

Padrões Pcase não apenas expressam uma condição na forma dos objetos que podem corresponder, mas também podem extrair subcampos desses objetos. Por exemplo podemos extrair 2 elementos de uma lista que é o valor da variável `my-list` com o seguinte código:

```
(pcase minha-lista
  (`(add ,x ,y) (mensagem "Contém %S e %S" xy)))
```

Isso não apenas extrairá `xe y` também testará que `my-list` é uma lista contendo exatamente 3 elementos e cujo primeiro elemento é o símbolo `add`. Se algum desses testes falhar, `pcase` retornará imediatamente `nil` sem chamar `message`.

A extração de vários valores armazenados em um objeto é conhecida como *desestruturação*. O uso `pcase` de padrões permite realizar *vinculação de desestruturação*, que é semelhante a uma vinculação local (consulte [Variáveis locais](#)), mas fornece valores a vários elementos de uma variável extraindo esses valores de um objeto de estrutura compatível.

As macros descritas nesta seção usam `pcase` padrões para realizar a ligação de desestruturação. A condição do objeto ser de estrutura compatível significa que o objeto deve corresponder ao padrão, pois somente assim os subcampos do objeto poderão ser extraídos. Por exemplo:

```
(pcase-let ((`(add ,x ,y) my-list))
  (mensagem "Contém %S e %S" xy))
```

faz o mesmo que o exemplo anterior, exceto que ele tenta extrair diretamente `xe y` de `my-list` sem antes verificar se `my-list` é uma lista que tem o número correto de elementos e tem `add` como primeiro elemento. O comportamento preciso quando o objeto não corresponde realmente ao padrão é indefinido, embora o corpo não seja ignorado silenciosamente: um erro é sinalizado ou o corpo é executado com algumas das variáveis potencialmente vinculadas a valores arbitrários como `nil`.

Os padrões `pcase` que são úteis para desestruturar ligações são geralmente aqueles descritos em [Backquote Patterns](#), uma vez que expressam uma especificação da estrutura dos objetos que irão corresponder.

Para um recurso alternativo para desestruturar a ligação, consulte [seq-let](#).

Macro: corpo de ligações pcase-let ...

Execute a vinculação de desestruturação de variáveis de acordo com as *vinculações* e, em seguida, avalie o *corpo*.

bindings é uma lista de bindings no formato `(binding exp pattern)`, onde *exp* é uma expressão a ser avaliada e *pattern* é um padrão. `(pattern exp) pcase`

Todas as *exp*s são avaliadas primeiro, após o que são comparadas com seus respectivos *padrões*, introduzindo novas associações de variáveis que podem ser usadas dentro de *body*. As vinculações de variáveis são produzidas desestruturando a vinculação de elementos do *padrão* aos valores dos elementos correspondentes da *exp* avaliada.

Macro: `pcase-let*` ...

Execute a vinculação de desestruturação de variáveis de acordo com as *vinculações* e, em seguida, avalie o *corpo* .

bindings é uma lista de bindings no formato , onde *exp* é uma expressão a ser avaliada e *pattern* é um padrão. As vinculações de variáveis são produzidas desestruturando a vinculação de elementos do *padrão* aos valores dos elementos correspondentes da *exp* avaliada . (*pattern exp*)`pcase`

Diferente `pcase-let` , mas similar a `let*` , cada *exp* é comparado com seu *padrão* correspondente antes de processar o próximo elemento de *bindings* , então os bindings variáveis introduzidos em cada um dos *bindings* estão disponíveis nos *exp* s dos *bindings* que o seguem, além de estarem disponíveis no *corpo* .

Macro: `pcase-dolist` (*lista de padrões*) *corpo*...

Executa *body* uma vez para cada elemento da *lista* , a cada iteração realizando uma ligação desestruturante das variáveis no *padrão* aos valores dos subcampos correspondentes do elemento da *lista* . As ligações são executadas como se por `pcase-let` . Quando *pattern* é uma variável simples, isso acaba sendo equivalente a `dolist` (veja [Iteração](#)).

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