

Anterior:[Combinadores de conselhos](#), Acima:[Funções de aconselhamento](#) [Conteúdo][Índice]

### 13.11.4 Adaptando o código usando o antigo defadvice

Muito código usa o defadvice mecanismo antigo, que é amplamente tornado obsoleto pelo novo advice-add, cuja implementação e semântica são significativamente mais simples.

Um conselho antigo, como:

```
(defadvice linha_anterior (antes da próxima linha no final
                               (&opcional argumento try-vscroll))
  "Insira uma linha vazia ao subir da linha superior."
  (se (e next-line-add-newlines (= arg 1)
        (salvar-excursão (início da linha) (bobp)))
      (prog
        (início de linha)
        (nova linha))))
```

poderia ser traduzido no novo mecanismo de aconselhamento em uma função simples:

```
(defun previous-line--next-line-at-end (&opcional arg try-vscroll)
  "Insira uma linha vazia ao subir da linha superior."
  (se (e next-line-add-newlines (= arg 1)
        (salvar-excursão (início da linha) (bobp)))
      (prog
        (início de linha)
        (nova linha))))
```

Obviamente, isso não modifica o previous-line. Para isso, o conselho antigo precisava:

```
(ad-activate 'previous-line)
```

Considerando que o novo mecanismo de aconselhamento necessita:

```
(advice-add 'previous-line :before #'previous-line--next-line-at-end)
```

Observe que ad-activate teve um efeito global: ativou todos os conselhos habilitados para essa função especificada. Se você quisesse ativar ou desativar apenas uma peça em particular, você precisaria *habilitá-la* ou *desativá-la* com ad-enable-advice e ad-disable-advice. O novo mecanismo acaba com essa distinção.

Em torno de conselhos como:

```
(defadvice foo (em torno de foo-around)
  "Ignore maiúsculas e minúsculas em 'foo'."
  (seja ((caso-dobra-pesquisa t)))
  (ad-do-it))
  (ativar anúncio 'foo')
```

poderia traduzir em:

```
(defun foo--foo-around (orig-fun & rest args)
  "Ignore maiúsculas e minúsculas em 'foo'."
  (seja ((caso-dobra-pesquisa t)))
  (aplicar argumentos orig-fun)))
(advice-add 'foo :around #'foo--foo-around)
```

*Em relação à classe* do conselho , observe que o novo :before não é exatamente equivalente ao antigo before, pois no conselho antigo você poderia modificar os argumentos da função (por exemplo, com ad-set-arg), e isso afetaria os valores dos argumentos vistos pela função original, enquanto que no conselho new :before, modificar um argumento via setq no conselho não tem efeito sobre os argumentos vistos pela função original. Ao portar before um conselho que dependia desse comportamento, você precisará transformá-lo em novo :around ou :filter-args conselho.

Da mesma forma, o conselho antigo after pode modificar o valor retornado alterando ad-return-value, enquanto o novo :after conselho não pode, portanto, ao portar esse after conselho antigo, você precisará transformá-lo em novo :around ou :filter-return conselho.

Anterior:[Combinadores de conselhos](#), Acima:[Funções de aconselhamento](#) [Conteúdo][Índice]