

Próximo:[Funções de aconselhamento](#), Anterior:[Células de Função](#), Acima:[Funções](#) [[Conteúdo](#)][[Índice](#)]

## 13.10 Fechamentos

Conforme explicado em [Variable Scoping](#) , o Emacs pode opcionalmente habilitar a ligação léxica de variáveis. Quando a vinculação léxica está habilitada, qualquer função nomeada que você cria (por exemplo, com `defun`), bem como qualquer função anônima que você cria usando a `lambda` macro ou a `function` forma especial ou a `#'` sintaxe (consulte [Funções anônimas](#) ), é automaticamente convertida em um *closure* .

Uma closure é uma função que também carrega um registro do ambiente lexical que existia quando a função foi definida. Quando é chamado, qualquer referência de variável léxica dentro de sua definição usa o ambiente léxico retido. Em todos os outros aspectos, os closures se comportam como funções comuns; em particular, eles podem ser chamados da mesma maneira que funções comuns.

Consulte [Lexical Binding](#) , para obter um exemplo de uso de uma closure.

Atualmente, um objeto closure do Emacs Lisp é representado por uma lista com o símbolo `closure` como o primeiro elemento, uma lista representando o ambiente lexical como o segundo elemento e a lista de argumentos e os formulários do corpo como os elementos restantes:

```
;; a ligação léxica está habilitada.  
(lambda (x) (* xx))  
  ⇒ (fechamento (t) (x) (* xx))
```

No entanto, o fato de que a estrutura interna de um encerramento é exposta ao resto do mundo Lisp é considerado um detalhe de implementação interna. Por esse motivo, não recomendamos examinar ou alterar diretamente a estrutura de objetos de fechamento.