

Próximo:[Carregamento automático](#), Anterior:[Formulários Macro](#), Acima:[Formulários](#) [[Conteúdo](#)][[Índice](#)]  
]

## 10.2.7 Formulários Especiais

Uma *forma especial* é uma função primitiva marcada especialmente para que seus argumentos não sejam todos avaliados. A maioria dos formulários especiais define estruturas de controle ou executa ligações de variáveis — coisas que as funções não podem fazer.

Cada forma especial tem suas próprias regras para as quais os argumentos são avaliados e quais são usados sem avaliação. A avaliação de um determinado argumento pode depender dos resultados da avaliação de outros argumentos.

Se o primeiro símbolo de uma expressão for de uma forma especial, a expressão deve seguir as regras dessa forma especial; caso contrário, o comportamento do Emacs não é bem definido (embora não falhe). Por exemplo, `((lambda (x) x) 3) 4)` contém uma subexpressão que começa com `lambda` mas não é uma expressão bem formada `lambda`, então o Emacs pode sinalizar um erro, ou pode retornar 3 ou 4 ou `nil`, ou pode se comportar de outras maneiras.

### Função: *objeto de forma especial-p*

Esse predicado testa se seu argumento é uma forma especial e retorna em `t` caso afirmativo, `nil` caso contrário.

Aqui está uma lista, em ordem alfabética, de todas as formas especiais no Emacs Lisp com uma referência para onde cada uma é descrita.

#### **and**

veja [Condições de Combinação](#)

#### **catch**

veja [Pegar e jogar](#)

#### **cond**

veja [Condicionais](#)

#### **condition-case**

veja [Tratamento de Erros](#)

#### **defconst**

veja [Definindo Variáveis](#)

#### **defvar**

veja [Definindo Variáveis](#)

#### **function**

veja [Funções Anônimas](#)

#### **if**

veja [Condicionais](#)

**interactive**

veja [Chamada interativa](#)

**lambda**

veja [expressões lambda](#)

**let****let\***

veja [Variáveis Locais](#)

**or**

veja [Condições de Combinação](#)

**prog1****prog2****progn**

veja [Sequenciamento](#)

**quote**

veja [Cotação](#)

**save-current-buffer**

veja [buffer atual](#)

**save-excursion**

veja [Excursões](#)

**save-restriction**

veja [Estreitamento](#)

**setq**

veja [Configurando Variáveis](#)

**setq-default**

veja [Criando Buffer Local](#)

**unwind-protect**

veja [Saídas Não Locais](#)

**while**

veja [Iteração](#)

**Nota do Common Lisp:** Aqui estão algumas comparações de formas especiais no GNU Emacs Lisp e Common Lisp. `setq`, `if`, e `catch` são formas especiais tanto no Emacs Lisp quanto no Common Lisp. `save-excursion` é uma forma especial no Emacs Lisp, mas não existe no Common Lisp. `throw` é uma forma especial em Common Lisp (porque deve ser capaz de lançar vários valores), mas é uma função em Emacs Lisp (que não possui vários valores).

Próximo:[Carregamento automático](#), Anterior:[Formulários Macro](#), Acima:[Formulários](#) [[Conteúdo](#)][[Índice](#)]  
]