

Próximo:[Descarregando](#), Anterior:[Recursos nomeados](#), Acima:[Carregando](#) [Conteúdo][Índice]

## 16.8 Qual arquivo definiu um determinado símbolo

### Função: símbolo de arquivo de símbolo e tipo opcional

Esta função retorna o nome do arquivo que definiu o *símbolo*. Se *type* for `nil`, qualquer tipo de definição será aceitável. Se *type* for `defun`, `defvar`, ou `defface`, isso especifica definição de função, definição de variável ou apenas definição de face.

O valor normalmente é um nome de arquivo absoluto. Também pode ser `nil`, se a definição não estiver associada a nenhum arquivo. Se *símbolo* especifica uma função carregada automaticamente, o valor pode ser um nome de arquivo relativo sem extensão.

A base para `symbol-file` são os dados na variável `load-history`.

### Variável: histórico de carregamento

O valor desta variável é uma lista que associa os nomes dos arquivos de biblioteca carregados com os nomes das funções e variáveis que eles definiram, bem como os recursos que eles forneceram ou exigiram.

Cada elemento nesta lista descreve uma biblioteca carregada (incluindo bibliotecas que são pré-carregadas na inicialização). É uma lista cujo CAR é o nome absoluto do arquivo da biblioteca (uma string). O restante dos elementos da lista tem estas formas:

#### `var`

O símbolo *var* foi definido como uma variável.

#### `(defun . fun)`

A função *fun* foi definida.

#### `(t . fun)`

A função *fun* era anteriormente um carregamento automático antes que essa biblioteca a redefinisse como uma função. O elemento a seguir é sempre , que representa a definição de *fun* como uma função. `(defun . fun)`

#### `(autoload . fun)`

A função *fun* foi definida como um autoload.

#### `(defface . face)`

O rosto do *rosto* foi definido.

#### `(require . feature)`

O recurso de *recurso* era necessário.

#### `(provide . feature)`

O recurso de *recurso* foi fornecido.

#### `(cl-defmethod method specializers)`

O método nomeado foi definido usando `cl-defmethod`, com *especialistas* como seus especialistas.

**(`define-type` . *type*)**

O tipo de *tipo* foi definido.

O valor de `load-history` pode ter um elemento cujo CAR é `nil`. Este elemento descreve as definições feitas `eval-buffer` em um buffer que não está visitando um arquivo.

O comando `eval-region` atualiza `load-history`, mas faz isso adicionando os símbolos definidos ao elemento para o arquivo que está sendo visitado, em vez de substituir esse elemento. Veja [Eval](#).

Próximo:[Descarregando](#), Anterior:[Recursos nomeados](#), Acima:[Carregando](#) [Conteúdo][Índice]