

Anterior:[Conversão de Caso](#), Acima:[Strings e Personagens](#) [[Conteúdo](#)][[Índice](#)]

## 4.10 A Tabela de Casos

Você pode personalizar a conversão de caixa instalando uma *tabela de caixa* especial . Uma tabela de maiúsculas e minúsculas especifica o mapeamento entre letras maiúsculas e minúsculas. Ela afeta as funções de conversão de maiúsculas e minúsculas para objetos Lisp (consulte a seção anterior) e aquelas que se aplicam ao texto no buffer (consulte [Alterações de maiúsculas](#) e minúsculas ). Cada buffer tem uma tabela de casos; há também uma tabela de casos padrão que é usada para inicializar a tabela de casos de novos buffers.

Uma tabela de casos é uma tabela de caracteres (consulte [Tabelas de caracteres](#) ) cujo subtipo é `case-table`. Esta tabela de caracteres mapeia cada caractere no caractere minúsculo correspondente. Possui três slots extras, que contêm tabelas relacionadas:

### *maiúscula*

A tabela de letras maiúsculas mapeia cada caractere no caractere maiúsculo correspondente.

### *canonizar*

A tabela *canonicalize* mapeia todo um conjunto de caracteres relacionados a maiúsculas e minúsculas em um membro específico desse conjunto.

### *equivalências*

A tabela de equivalências mapeia cada um de um conjunto de caracteres relacionados ao caso para o próximo caractere desse conjunto.

Em casos simples, tudo o que você precisa especificar é o mapeamento para letras minúsculas; as três tabelas relacionadas serão calculadas automaticamente a partir daquela.

Para alguns idiomas, as letras maiúsculas e minúsculas não estão em correspondência um-para-um. Pode haver duas letras minúsculas diferentes com o mesmo equivalente em maiúsculas. Nesses casos, você precisa especificar os mapas para letras minúsculas e maiúsculas.

A tabela extra *canonicalize* mapeia cada caractere para um equivalente canônico; quaisquer dois caracteres relacionados por conversão de maiúsculas e minúsculas têm o mesmo caractere equivalente canônico. Por exemplo, uma vez que 'uma' e 'UMA' estão relacionados por conversão de maiúsculas e minúsculas, eles devem ter o mesmo caractere equivalente canônico (que deve ser 'uma' para ambos, ou 'UMA' Para ambos).

A tabela extra de *equivalências* é um mapa que permuta ciclicamente cada classe de equivalência (de caracteres com o mesmo equivalente canônico). (Para ASCII comum , isso mapearia 'uma' para dentro 'UMA' e 'UMA' para dentro 'uma', e da mesma forma para cada conjunto de caracteres equivalentes.)

Ao construir uma tabela de casos, você pode `nil` fornecer *canonicalize* ; então o Emacs preenche este slot a partir dos mapeamentos de letras minúsculas e maiúsculas. *Você também pode* fornecer `nil` equivalências ; então o Emacs preenche este slot de *canonicalize* . Em uma tabela de casos que está realmente em uso, esses componentes não são . Não tente especificar *equivalências* sem também especificar *canonicalize* . `nil`

Aqui estão as funções para trabalhar com tabelas de casos:

**Função: *objeto case-table-p***

Esse predicado retorna um *objeto* `nil` não- se é uma tabela de casos válida.

**Função: *tabela set-standard-case- table***

Esta função torna a *tabela* a tabela de casos padrão, para que seja usada em quaisquer buffers criados posteriormente.

**Função: *tabela de caso padrão***

Isso retorna a tabela de casos padrão.

**Função: *tabela de caso atual***

Esta função retorna a tabela de casos do buffer atual.

**Função: *tabela set-case- table***

Isso configura a tabela de casos do buffer atual para *table* .

**Macro: *corpo de mesa com estojo ...***

A *with-case-table* macro salva a tabela de casos atual, torna *table* a tabela de casos atual, avalia os formulários do *corpo* e, finalmente, restaura a tabela de casos. O valor de retorno é o valor do último formulário em *body* . A tabela de casos é restaurada mesmo em caso de saída anormal via *throw* ou erro (ver [Saídas Não Locais](#) ).

Alguns ambientes de idioma modificam as conversões de maiúsculas e minúsculas de caracteres ASCII ; por exemplo, no ambiente de língua turca, o ASCII maiúsculo I é reduzido para um turco sem ponto i ('eu'). Isso pode interferir no código que requer conversão comum de maiúsculas e minúsculas ASCII , como implementações de protocolos de rede baseados em ASCII . Nesse caso, use a *with-case-table* macro com a variável *ascii-case-table* , que armazena a tabela de maiúsculas e minúsculas não modificada para o conjunto de caracteres ASCII .

**Variável: *ascii-case-table***

A tabela de casos para o conjunto de caracteres ASCII . Isso não deve ser modificado por nenhuma configuração de ambiente de idioma.

As três funções a seguir são sub-rotinas convenientes para pacotes que definem conjuntos de caracteres não ASCII . Eles modificam a tabela de casos especificada *case-table* ; eles também modificam a tabela de sintaxe padrão. Consulte [Tabelas de sintaxe](#) . Normalmente, você usaria essas funções para alterar a tabela de casos padrão.

**Função: *set-case-syntax-pair uc lc case-table***

Esta função especifica um par de letras correspondentes, uma maiúscula e uma minúscula.

**Função: *set-case-syntax-delims l r case-table***

Essa função torna os caracteres *l* e *r* um par correspondente de delimitadores invariantes de maiúsculas e minúsculas.

**Função: *set-case-syntax char sintaxe case-table***

Esta função torna *char* invariável entre maiúsculas e minúsculas, com sintaxe de *sintaxe* .

**Comando: describe-buffer-case-table**

Este comando exibe uma descrição do conteúdo da tabela de casos do buffer atual.

Anterior:[Conversão de Caso](#), Acima:[Strings e Personagens](#) [[Conteúdo](#)][[Índice](#)]