

# GNU Emacs: Cartão de Referência

(para versão 27)

## Iniciando o Emacs

Para entrar no GNU Emacs, digite: `emacs`

## Saindo do Emacs

suspende ou minimiza o Emacs `C-z`  
encerra o Emacs `C-x C-c`

## Arquivos

**abre** um arquivo `C-x C-f`  
**salva** um arquivo em disco `C-x C-s`  
salva **todos** arquivos abertos `C-x s`  
**insere** outro arquivo neste buffer `C-x i`  
substitui este arquivo por outro `C-x C-v`  
salva o buffer em um arquivo especificado `C-x C-w`  
alterna o estado de somente leitura do buffer `C-x C-q`

## Ajuda (Help)

Tecla `C-h` (ou `F1`) e siga as instruções.  
remove a janela de ajuda `C-x 1`  
rola a janela de ajuda `C-M-v`  
apropos: mostra comandos que casam com a string `C-h a`  
descreve função associada a teclas `C-h k`  
descreve uma função `C-h f`  
busca informações específicas do modo `C-h m`

## Recuperando-se de Erros

**aborta** uma operação `C-g`  
**recupera** arquivos após crash `M-x recover-session`  
desfaz uma alteração (**undo**) `C-x u, C-_ or C-/`  
restaura um buffer para o arquivo `M-x revert-buffer`  
redesenha a tela `C-l`

## Busca Incremental

busca para frente `C-s`  
busca para trás `C-r`  
busca por expressão regular `C-M-s`  
busca por expressão regular para trás `C-M-r`  
seleciona a string de pesquisa anterior `M-p`  
seleciona a string seguinte de pesquisa `M-n`  
sai da busca incremental `RET`  
desfaz o efeito do último carácter `DEL`  
encerra a busca `C-g`

Use `C-s` ou `C-r` novamente para repetir a busca. `C-g` cancela apenas o que ainda não foi feito.

## Movimentação

<b>avançar</b>	<b>trás</b>	<b>frente</b>
um carácter	<code>C-b</code>	<code>C-f</code>
uma palavra	<code>M-b</code>	<code>M-f</code>
uma linha	<code>C-p</code>	<code>C-n</code>
para início ou fim de linha	<code>C-a</code>	<code>C-e</code>
sentença	<code>M-a</code>	<code>M-e</code>
parágrafo	<code>M-{</code>	<code>M-}</code>
página	<code>C-x [</code>	<code>C-x ]</code>
semp	<code>C-M-b</code>	<code>C-M-f</code>
função	<code>C-M-a</code>	<code>C-M-e</code>
para início ou fim do buffer	<code>M-&lt;</code>	<code>M-&gt;</code>
rolar para próxima tela	<code>C-v</code>	
rolar para tela anterior	<code>M-v</code>	
rolar para esquerda	<code>C-x &lt;</code>	
rolar para direita	<code>C-x &gt;</code>	
rolar a linha corrente para o centro da tela	<code>C-u</code>	<code>C-l</code>

## Cortando e Apagando

<b>entidade a cortar</b>	<b>trás</b>	<b>frente</b>
caracter (apaga, não corta)	<code>DEL</code>	<code>C-d</code>
palavra	<code>M-DEL</code>	<code>M-d</code>
linha (até o final)	<code>M-O C-k</code>	<code>C-k</code>
sentença	<code>C-x DEL</code>	<code>M-k</code>
semp	<code>M-- C-M-k</code>	<code>C-M-k</code>
corta <b>região</b>	<code>C-w</code>	
copia a <b>região</b>	<code>M-w</code>	
cortar até a próxima ocorrência de <i>char</i>	<code>M-z</code>	<i>char</i>
colar a última coisa cortada	<code>C-y</code>	
substitui a últ. colagem pela cópia anterior	<code>M-y</code>	

## Marcando

posiciona a marca aqui `C-@` or `C-SPC`  
troca a marca pelo ponto e vice-versa `C-x C-x`  
coloca a marca *arg* **palavras** adiante `M-@`  
marca o **parágrafo** `M-h`  
marca a **página** `C-x C-p`  
marca a **semp** `C-M-@`  
marca uma **função** `C-M-h`  
marca todo **buffer** `C-x h`

## Busca e Substituição

Substitui interativamente uma string `M-%`  
usando expressão regular `M-x query-replace-regexp`  
Respostas válidas no modo de busca e substituição  
**substitui** esta, e prossegue `SPC`  
substitui esta e entrada e não avança `,`  
**pula** para a próxima sem substituir `DEL`  
substitui em todo o texto restante `!`  
**volta** para a palavra anterior `~`  
**encerra** `RET`  
entra na edição recursiva (`C-M-c` para sair) `C-r`

## Múltiplas Janelas

Quando forem mostrados 2 comandos, o segundo tem comportamento similar para frame.

elimina todas outras janelas	<code>C-x 1</code>	<code>C-x 5 1</code>
divide a janela, acima e abaixo	<code>C-x 2</code>	<code>C-x 5 2</code>
elimina esta janela	<code>C-x 0</code>	<code>C-x 5 0</code>
divide a janela, lado a lado		<code>C-x 3</code>
rola a outra janela		<code>C-M-v</code>
leva o cursor para outra janela	<code>C-x o</code>	<code>C-x 5 o</code>
seleciona um buffer em outra janela	<code>C-x 4 b</code>	<code>C-x 5 b</code>
mostra um buffer em outra janela	<code>C-x 4 C-o</code>	<code>C-x 5 C-o</code>
busca um arquivo em outra janela	<code>C-x 4 f</code>	<code>C-x 5 f</code>
busca arquivo (ro) em outra janela	<code>C-x 4 r</code>	<code>C-x 5 r</code>
executa Dired em outra janela	<code>C-x 4 d</code>	<code>C-x 5 d</code>
busca tag em outra janela	<code>C-x 4 .</code>	<code>C-x 5 .</code>
aumenta a janela na vertical		<code>C-x ^</code>
estrita a janela		<code>C-x {</code>
alarga a janela		<code>C-x }</code>

## Formatando

identifica a **linha** corrente (modo) `TAB`  
identifica a **região** (modo) `C-M-\`  
identifica a **semp** (modo) `C-M-q`  
identifica região rigidamente *arg* colunas `C-x TAB`  
insere uma nova linha após o ponto `C-o`  
move o restante da linha para baixo `C-M-o`  
apaga linhas em branco em torno do ponto `C-x C-o`  
junta a linha com a anterior `M-^`  
apaga todos brancos em torno do ponto `M-\`  
insere um espaço em branco `M-SPC`  
preenche o parágrafo `M-q`  
define a coluna limite de preenchimento `C-x f`  
define um prefixo para cada linha `C-x .`  
formata fonte `M-o`

## Maiúsculas e Minúsculas

Palavra para maiúsculas `M-u`  
Palavra para minúsculas `M-l`  
Primeira letra maiúscula (capitalize) `M-c`  
Região para maiúsculas `C-x C-u`  
Região para minúsculas `C-x C-l`

## O Minibuffer

As teclas seguintes são definidas no minibuffer.

complete o máximo possível	<code>TAB</code>
complete até uma palavra	<code>SPC</code>
complete e execute	<code>RET</code>
mostre as opções para completar	<code>?</code>
busca a entrada anterior no minibuffer	<code>M-p</code>
busca a próxima entrada no minibuffer ou o default	<code>M-n</code>
busca regexp no histórico para trás	<code>M-r</code>
busca regexp no histórico para frente	<code>M-s</code>
encerra o comando	<code>C-g</code>

Tecla `C-x ESC ESC` para editar e repetir o último comando utilizado. Tecla `F10` para ativar o menu.

# GNU Emacs: Cartão de Referência

(para versão 27)

## Buffers

seleciona outro buffer	C-x b
lista todos buffers	C-x C-b
mata um buffer	C-x k

## Transposição

transpõe <b>caracteres</b>	C-t
transpõe <b>palavras</b>	M-t
transpõe <b>linhas</b>	C-x C-t
transpõe <b>sexps</b>	C-M-t

## Verificação Ortográfica

verifica a palavra corrente	M-\$
verifica todas palavras de uma região	M-x ispell-region
verifica todo o buffer	M-x ispell-buffer

## Tags

busca uma tag (uma definição)	M-.
encontra a próxima ocorrência da tag	C-u M-.
especifica um novo arquivo de tags	M-x visit-tags-table
busca por regexp em todos arquivos	M-x tags-search
busca e subst. em todos arquivos	M-x tags-query-replace
continua a última busca ou busca e substi-tuição	M-,

## Shells

executa um comando do shell	M-!
executa um comando do shell na região	M-
filtra uma região por um comando do shell	C-u M-
inicia um shell na janela *shell*	M-x shell

## Retângulos

copia o retângulo para o registrador	C-x r r
corta o retângulo	C-x r k
cola o retângulo	C-x r y
abre o retângulo, move o texto para direita	C-x r o
troca por espaços o conteúdo do retângulo	C-x r c
antepõe uma linha a string	C-x r t

## Abreviaturas

adiciona uma abreviatura global	C-x a g
adiciona abreviatura ao modo local	C-x a l
adiciona globalmente expansão de abrev.	C-x a i g
adiciona localmente expansão de abrev.	C-x a i l
explicitamente expande uma abrev	C-x a e
completa com base em palavras anteriores	M-/

## Expressões Regulares

qualquer caracter exceto nova linha	.	(dot)
zero ou mais repetições	*	
uma ou mais repetições	+	
zero ou uma repetição	?	
protege o caracter especial <i>c</i> (“or”)	\c	
	\\	
agrupamento	\( ... \)	
mesmo texto que <i>n</i> -ésimo grupo	\n	
quebra de palavra	\b	
sem quebra de palavra	\B	
<b>entidade</b>	<b>casa início</b>	<b>casa fim</b>
linha	^	\$
palavra	\<	\>
buffer	\‘	\’
<b>classe de caracteres</b>	<b>casa esses</b>	<b>casa os outros</b>
conjunto explícito	[ ... ]	[^ ... ]
caracter de sintaxe de palavra	\w	\W
caracter de sintaxe de <i>c</i>	\sc	\Sc

## Conjuntos de Carac. Internacionais

especifica uma língua principal	C-x RET l
mostra todos métodos de inserção	M-x list-input-methods
habilita/desabilita um método de inserção	C-\
determina o sistema de codificação	C-x RET c
mostra sistemas de codificação	M-x list-coding-systems
escolhe a codificação preferida	M-x prefer-coding-system

## Info

entra no leitor de Info	C-h i
busca função ou arquivo no Info	C-h S

Movimentação em um nodo:

rola para frente	SPC
rola para trás	DEL
início do nodo	. (dot)

Movimentação entre nodos:

<b>próximo</b> nodo	n
nodo <b>anterior</b>	p
mover cima <b>cima</b>	u
seleciona item do menu pelo nome	m
seleciona <i>n</i> -ésimo item do menu	<i>n</i>
segue referência cruzada (retorna com l)	f
retorna último nodo visitado	l
retorna ao diretório de nodos	d
ir para o topo do arquivo Info	t
ir para qualquer nodo por nome	g

Outros:

executar <b>tutorial</b> do Info	h
busca pelo assunto no índice	i
busca por expressão regular	s
<b>sair</b> Info	q

## Registrador

salva região em um registrador	C-x r s
insere o conteúdo do registrador no buffer	C-x r i
salva valor do ponto no registrador	C-x r SPC
salta para o ponto salvo no registrador	C-x r j

## Macros de Teclado

<b>inicia</b> a definição de uma macro	C-x (
<b>encerra</b> a definição de uma macro	C-x )
<b>executa</b> a última macro definida	C-x e
adiciona a última macro definida	C-u C-x (
nomeia a última macro definida	M-x name-last-kbd-macro
insere uma definição em Lisp	M-x insert-kbd-macro

## Lidando com Emacs Lisp

avalia <b>sexp</b> antes do ponto	C-x C-e
avalia a <b>defun</b> corrente	C-M-x
avalia a <b>região</b>	M-x eval-region
lê e avalia o minibuffer	M-:
carrega do diretório padrão do sistema	M-x load-library

## Personalização Simples

personaliza variáveis e fontes M-x customize

Fazendo teclas de atalho globais em Emacs Lisp (exemplos):

```
(global-set-key "\C-cg" 'goto-line)
(global-set-key "\M-#" 'query-replace-regexp)
```

## Escrevendo Comandos

```
(defun command-name (args)
  "documentation" (interactive "template")
  body)
```

Um exemplo:

```
(defun this-line-to-top-of-window (line)
  "Reposition line point is on to top of window.
With ARG, put point on line ARG."
  (interactive "P")
  (recenter (if (null line)
                0
                (prefix-numeric-value line))))
```

A especificação **interactive** explica como ler interativamente argumentos. Tecla C-h f **interactive** para mais detalhes.

Copyright © 2021 Free Software Foundation, Inc.  
For GNU Emacs version 27  
Designed by Stephen Gildea  
Translated by Rodrigo Real

Released under the terms of the GNU General Public License version 3 or later.

For more Emacs documentation, and the TeX source for this card, see the Emacs distribution, or <https://www.gnu.org/software/emacs>