

Instituto Federal de Santa Catarina
CST em Análise e Desenvolvimento de Sistemas
Introdução à Computação

VJ e VIM

Prof. Diego da Silva de Medeiros

São José, outubro de 2023

Editor/processador de texto

- Editor de texto:
 - Permite editar textos sem formatação
 - Bloco de notas, gedit, etc
- Processador de texto
 - Edita textos com formatação, tabelas, figuras
 - Word, BrOffice, etc

VI - Visual Interface // VIM - VI Improved

- Um dos EDITORES de texto Linux mais usados
- Possui recursos avançados de edição
 - Abrir vários arquivos ao mesmo tempo
 - Autocorreção
 - Auto-identação
 - Seleção visual
 - Macros
 - Seleção vertical de texto
 - Interação total com o sistema operacional
 - Uso de expressões regulares
 - Sintaxe colorida
- VIM é uma versão melhorada do VI



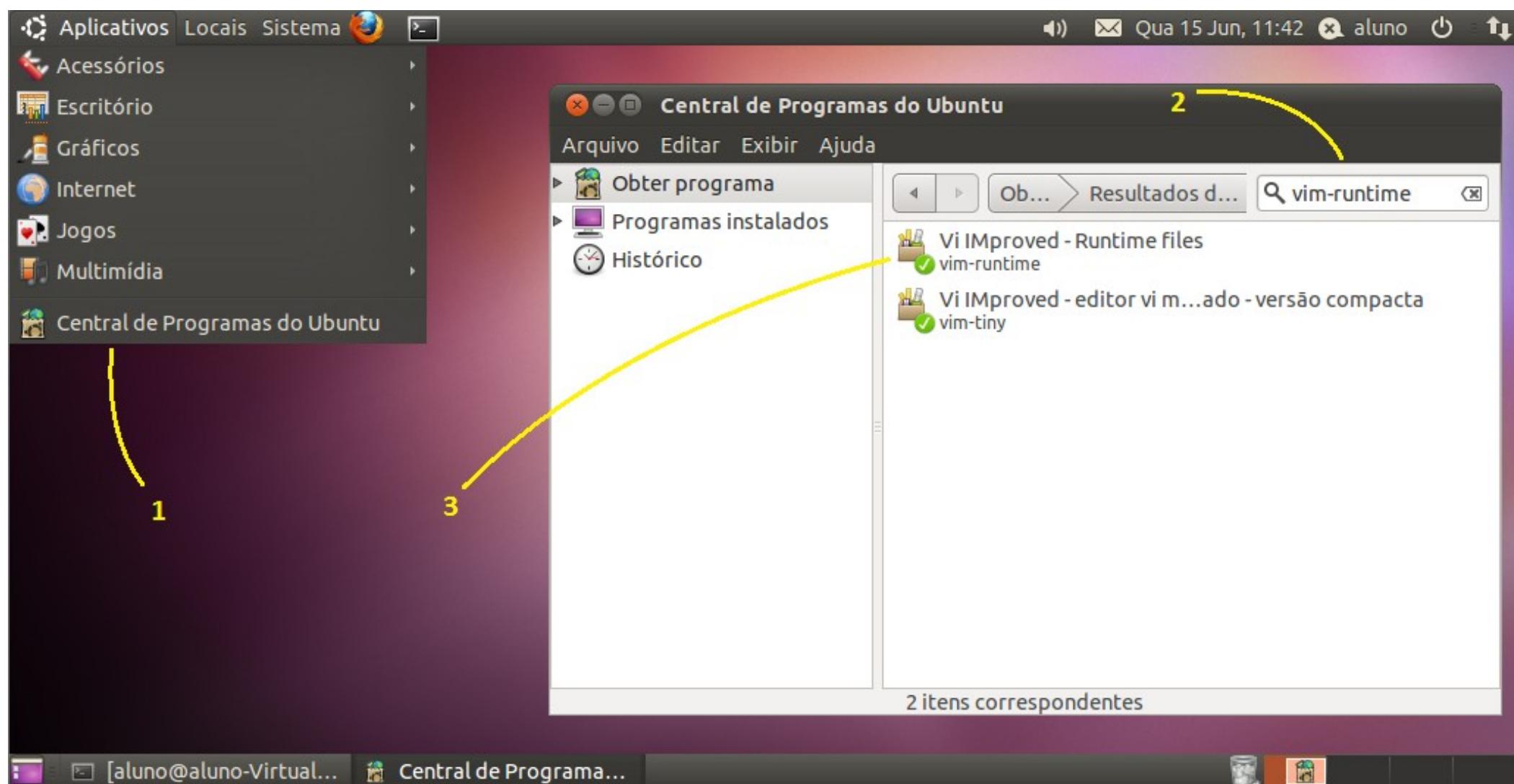


VI gang sign

VI/VIM - Instalação

- Algumas distribuições do Linux podem vir apenas com uma versão reduzida do VI/VIM
- Para a versão completa, instalar o pacote “**vim-runtime**”:
 - Abrir a Central de Programas do Ubuntu (1)
 - Procurar por “**vim-runtime**” (2)
 - Instalar o programa “**Vi IMproved - Runtime files**” (3)

VI/VIM - Instalação



VI/VIM - Modos

- **Modo de inserção/substituição**

- Teclas do teclados são letras que serão inseridas no arquivo
- Modo identificado por um -- INSERT --/-- REPLACE -- no canto inferior esquerdo

- **Modo de comandos**

- Teclas do teclado são comandos para o editor
- Modo por padrão ativo quando se abre um arquivo

- **Modo visual**

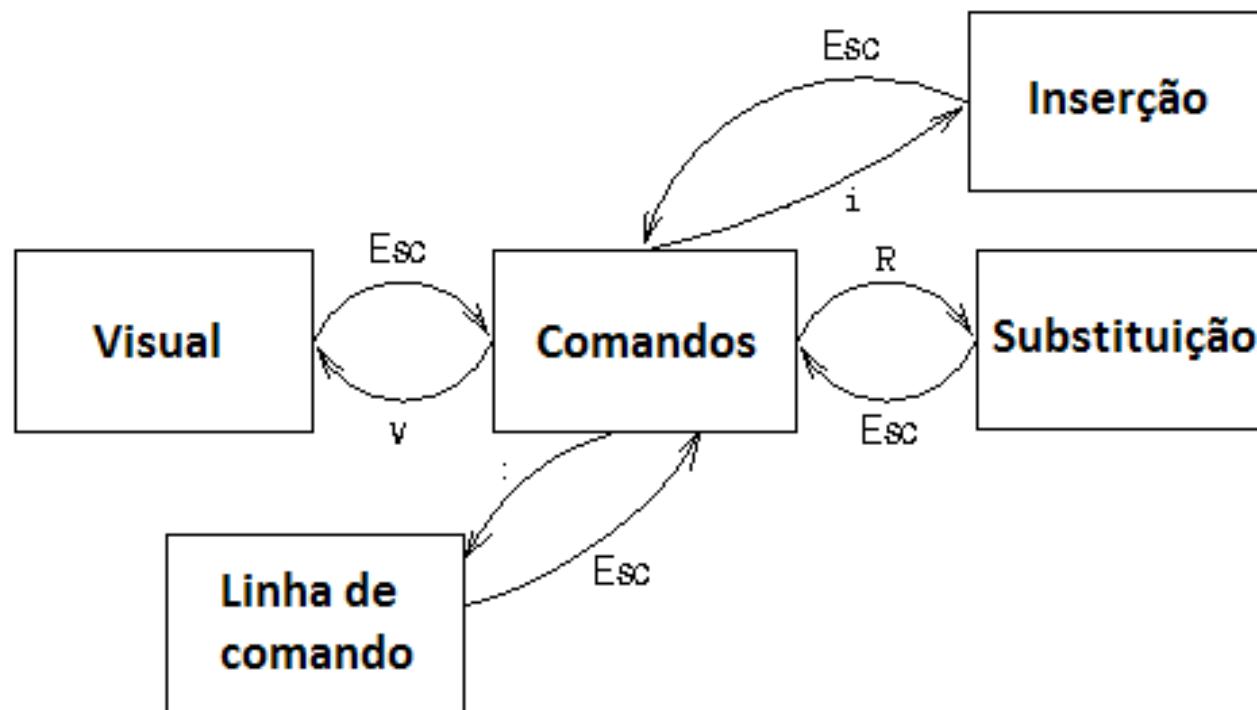
- Permite selecionar um texto dentro do arquivo
- Semelhante ao arrastar do mouse num editor de textos visual
- Identificado por um -- VISUAL -- no canto inferior esquerdo da tela

- **Modo de linha de comando**

- Permite salvar o arquivo, abrir novos arquivos, fechar um dos arquivos abertos
- Recurso de procurar e substituir

VI/VIM - Operação

- Modo de comando = Padrão
- Modo de comando → modo de inserção: i
- Modo de comando → modo visual: v
- Sair do modo de inserção/visual, voltando para o de comando: ESC
- Sair do VI (a partir do modo de comando): SHIFT+zz



VI/VIM - Mapa dos comandos

ESC
normal mode

~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
\ goto mark	1 ²	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste after	• misc	• misc	• misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	• ex cmd line	" reg spec ¹	! goto mk. bol	bol/ goto col
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	• not used!	•	
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³ indent	> indent ³	? find (rev.)	/ find		
Z extra ⁵	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	, reverse t/T/f/F	• repeat cmd			

- motion** moves the cursor, or defines the range for an operator
- command** direct action command, if red, it enters insert mode
- operator** requires a motion afterwards, operates between cursor & destination
- extra** special functions, requires extra input
- Q·** commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line,
mk = mark, yank = copy

words: guux([foo, bar, baz];
WORDS: guux(foo, bar, baz);

Main command line commands ('ex'):

:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:%s/x/y/g (replace 'x' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important commands:

CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line
(dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

VI/VIM - Help

- A partir do modo de comandos, digitar *:help*
- A navegação é dada como se fosse um texto normal
- Seções do help destacadas permitem “pular” diretamente para elas através das teclas *CTRL+J*

VI/VIM - Abertura

- Comandos *vi* e *vim* são o mesmo programa, o *vim*

```
aluno@Maquina:~$ whereis vi  
vi: /usr/bin/vi /usr/share/man/man1/vi.1.gz
```

```
aluno@Maquina:~$ ls -l /usr/bin/vi  
lrwxrwxrwx 1 root root 20 2011-06-17 01:07 /usr/bin/vi -> /etc/alternatives/vi
```

```
aluno@Maquina:~$ ls -l /etc/alternatives/vi  
lrwxrwxrwx 1 root root 17 2011-06-17 01:06 vi -> /usr/bin/vim.tiny
```

- Para abrir um arquivo:

- **vi** Abre o vim vazio, sem arquivo (pouco usado)
- **vi arquivo** **Abre o arquivo "arquivo". Se ele não existir, o cria**
- **vi arquivo +** Abre com cursor no fim do arquivo
- **vi arquivo +10** Abre com cursor na linha 10
- **vi arquivo +/amigo** Abre com cursor na 1ª ocorrência de "amigo"

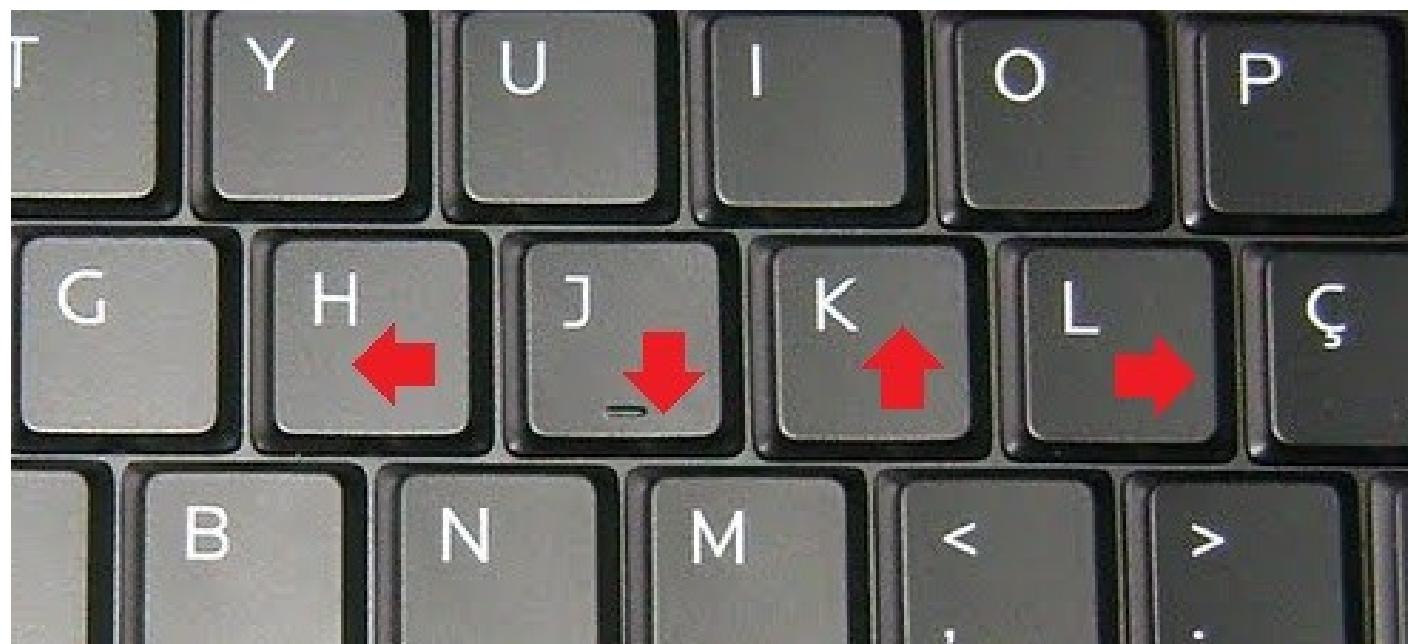
VI] - Salvando/saindo

- Para sair do *vi*, ir para o modo de comandos [ESC]

• :w	Salva
• :q	Sai
• :wq ou :x ou ZZ	Salva e sai
• :w!	Salva forçado
• :q!	Sai forçado
• :wq!	Salva e sai forçado

VI - Movendo o cursor do teclado

- Dependendo da versão do vi é possível que as setas não movam o cursor do teclado pelo arquivo
- Neste caso, para movê-lo, usar as teclas *h, j, k, l*



VII - Movendo o cursor do teclado

- Além disso, outras teclas são usadas para mover o cursor:

- w início da próxima palavra
- b início da palavra anterior
- e final da palavra atual
- H início da tela
- M meio da tela
- L final da tela
- qq início do arquivo
- G final do arquivo
- { início do parágrafo
- } final do parágrafo

VII - Movendo o cursor do teclado

- (início da sentença atual (sentença é um texto com um ponto final)
-) final da sentença atual
- - início da linha anterior
- + início da próxima linha
- \$ final da linha atual
- | início da linha atual
- : N linha de número N dentro do arquivo

VII - Copiando

- **Copiando uma linha:**

- Digitar duas vezes a letra `y` de *yank* (arrancar)

- **Copiando uma linha**

- Digitar a letra `Y`

- **Copiando mais de uma linha:**

- Digitar o número de linhas e duas vezes a letra `y`
 - Selecionar as linhas no modo visual e digitar `y`

VII - Deletando/recortando

- Excluindo letras:
 - x minúsculo apaga um caracter à frente, como o *Delete*
 - X maiúsculo apaga um caracter atrás, como o *Backspace*
- Excluindo uma linha:
 - Digitar duas vezes a letra d de *delete*
- Excluindo toda a linha:
 - Digitar a letra D
- Excluindo mais de uma linha:
 - Digitar o número de linhas e duas vezes a letra d
 - Selecionar as linhas no modo visual e digitar d
- As linhas excluídas ficam armazenadas, podendo ser coladas

VII - Colando e desfazendo alterações

- Colando antes do cursor:

- Digitar P de *paste* (colar)

- Colando depois do cursor:

- Digitar p

- Colando repetidamente:

- Digitar o número de vezes que o texto deve ser colado e digitar p

- Desfazendo última alteração (caracter)

- Digitar u de *undo*

- Desfazendo última alteração (linha)

- Digitar U de *undo*

VII - O modo de inserção/substituição

- i modo de inserção (-- INSERT --)
- I insere no início da linha
- a insere depois do caractere atual (*append*)
- A insere no final da linha
- s insere, excluindo o caracter atual
- S insere, excluindo a linha atual
- o insere entre a linha atual e a próxima, criando uma linha
- O insere entre a linha atual e a anterior, criando uma linha
- r substitui o caractere atual (não abre o modo de inserção)
- R modo de substituição (-- REPLACE --)

VII - Busca texto

- Para procurar uma palavra ou frase dentro do texto:
 - */palavra* procura por ocorrências da *palavra* dentro do texto
 - */frase normal* procura por ocorrências da *frase normal* dentro do texto
 - */palavra1\|palavra2* procura ao mesmo tempo por ocorrências da palavra1 e da palavra2
- Navegando entre os resultados
 - *n* próximo resultado
 - *N* resultado anterior

VII - Substituições

- Substituições são feitas no modo de linha de comando
 - `:s/velho/novo` troca para *novo* a 1a ocorrência de *velho* na linha corrente
 - `:%s/velho/novo` troca para *novo* a 1a ocorrência de *velho* na linha, em todo o arquivo (%)
 - `:%s/velho/novo/g` troca para *novo* todas (*g*) as ocorrências de *velho* em todo o arquivo (%)
 - `:%s/velho/novo/gc` igual ao anterior, pedindo confirmação em cada substituição
- Assim como na procura, é possível navegar nos resultados:
 - *n* próximo resultado
 - *N* resultado anterior

VII - Maiúsculas e minúsculas

- $N\sim$ converte os próximos N caracteres para maiúsculo
- $g\sim\$$ converte até o final da linha
- No modo visual:
 - u converte para minúsculas
 - U converte para maiúsculas
 - \sim inverte

VI/VIM - Alinhamento

- É possível escolher o alinhamento para o texto
- Para alinhar uma linha, deixar o cursor na linha e digitar :<*alinhamento*>
- As opções disponíveis são
 - Centralizado :*center*
 - Esquerdo :*left*
 - Direito :*right*
- Para alinhar mais de uma linha, selecionar as linhas no modo visual

VI/VIM - Opções

- É possível alterar algumas opções dentro do *vi*
- O comando *:set* é usado:
 - *:set hlsearch* ilumina a última procura
 - *:set ignorecase* case insensitive na busca
 - *:set incsearch* ilumina procura enquanto digita
 - *:set number* mostra número da linha
 - *:set smartcase* assume "noignorecase" quando tiver maiúsculas na palavra procurada
 - *:set lines=N* número de linhas na tela
- Com exceção do *:set lines=N*, todos possuem o complemento com o *no* na frente, indicando o inverso

VI/VIM - O .vimrc

- O *vi* é um editor de textos altamente configurável
- Há muitas opções que podem ser utilizadas na personalização do aplicativo
- As configurações do *vi* são escritas no arquivo `~/.vimrc`
- Há vários exemplos de `.vimrc` na internet
- Um exemplo de `.vimrc` pode ser encontrado [aqui](#)
 - Copie o conteúdo de um arquivo de exemplo e crie um arquivo `.vimrc` dentro de seu diretório *home*
 - Abra o *vi* e veja a diferença...

Referências

- [Blog do aurélio](#)
- [O editor ED](#)
- [O mapa das teclas do VI/VIM](#)
- Comando *vimtutor* - tutorial sobre o *vim* (inglês)