

Emacs em 30 Segundos

Guaracy Monteiro

Contents

Emacs em 30 segundos	3
Alternativas	3
Características do projeto	4
org-mode	4
Arquivo .emacs	4
Configurações iniciais	4
Pacotes	4
Alterações das opções iniciais	5
Inclusão e atualização de fonte de pacotes	5
Seleção e instalação dos pacotes pelo Emacs30	6
Configurações da aparência e dos pacotes	6
Tamanho da janela	6
Arquivo .myemacs	11

TL;DR

Arquivo de configuração para ser utilizado em uma instalação nova do Emacs ou substituir uma antiga (excluir **.emacs** e **.emacs.d**). Automaticamente instala alguns pacotes definidos e configura o ambiente e os pacotes. Tudo bem explicadinho para ser alterado//melhorado facilmente.

1. Grave o arquivo .emacs, no seu diretório **home**. Se desejar, grave também o arquivo .myemacs no seu diretório **home**. Abra o Emacs e espere a configuração terminar.
 2. Baixe e leia a documentação **TBD** .
-

Emacs em 30 segundos

A ideia base deste projeto é a possível dificuldade de instalação e configuração do Emacs por parte dos usuários. A instalação é simples mas, conforme o programa sai de *fábrica*, muitas pessoas encontram uma pequena dificuldade inicial e, como pensam: “- *Ah, é só mais um editor de textos. Vou continuar com o que eu já conheço.*”, e perdem uma ferramenta das mais poderosas que poderiam usar. Apesar de ser possível efetuar algumas alterações pelo menu, muitas necessitam que o usuário edite um arquivo de configuração (**.emacs**). A instalação de novos pacotes, por exemplo, já foi uma tarefa bem mais complicada.

Alternativas

Existem algumas opções que possuem um Emacs configurado para o usuário final. É possível citar:

1. Spacemacs : É uma ótima alternativa e bastante indicada para usuários do Vim. Existem pessoas que tentaram usar o Emacs mas voltaram para o Vim. Depois de instalar o Spacemacs, estão usando e achando ótimo. Basicamente é uma cópia do diretório **.emacs.d** com toda a configuração automática. O usuário precisará ler a documentação se desejar fazer alguma alteração na configuração.
2. super-emacs : Também é uma cópia do diretório **.emacs.d** porém com uma configuração mais simples que o Spacemacs.
3. Ergoemacs : Deve ser uma ótima configuração para o Windows. No Linux é difícil de testar pois muitos atalhos conflitam com o desktop (KDE, Gnome, etc).
4. Emacs30 : Este que você está lendo. Baseia-se apenas em um arquivo de configuração (**.emacs**) que deve ser gravado no diretório home do usuário. Ele irá baixar e configurar os diversos pacotes. Ao término, ele verifica se existe o arquivo (**.myemacs**). Se existir, continua a configuração por este arquivo.

Vejo como principal desvantagem dos itens 1 e 2 a não habilitação do CUA-mode. Esconder o menu principal também pode ser estranho para alguns usuários iniciantes. O item e eu não testei. A habilitação do CUA-mode, entre outras coisas, permite combinações que a maioria dos usuários está acostumada como: *Ctrl+c Ctrl+c* para copiar, *Ctrl+c Ctrl+v* para colar e *Ctrl+z* para desfazer.

Inicialmente o Emacs possui uma tela assim:

```
[ [[file:/images/emacs1.jpg]] ]
```

ou seja, uma barra de ferramentas de gosto duvidoso (depois de um breve período de uso você vai achar melhor os atalhos do que tirar a mão do teclado, pegar o mouse e clicar em um botão) e uma tela inicial com muita informação. Mas tudo isto irá mudar.

Características do projeto

1. Habilitação do CUA-mode para que o usuário não precise aprender que *Alt+w* é usada para copiar um texto, por exemplo.
2. Alterar a aparência do Emacs, utilizando um tema com fundo escuro. O usuário poderá optar por um fundo claro a qualquer momento
3. Deixar a barra de status com uma aparência mais agradável.
4. Alterar diversas opções para deixar o Emacs mais amigável(?).
5. Instalar alguns pacotes para facilitar o trabalho do usuário em diversas áreas, podendo inclusive gerar arquivos .docx (Microsoft Word), .odt (LibreOffice), .pdf, .epub (eBook) entre diversos outros de forma simples (pode necessitar da instalação de programas externos como por exemplo, para gerar .pdf é necessário instalar o **MiKTeX** no Windows ou o **texlive** no Linux).

org-mode

Inicialmente estava previsto a instalação do pacote **markdown** para facilitar a criação e transformação de textos em uma linguagem de marcação simples para outros documentos mais complexos. Como o Emacs já vem com o org-mode e muitos usuários utilizam o Emacs basicamente pelo pacote, nada melhor do que utilizá-lo.

1. Utiliza uma linguagem de marcação relativamente simples.
2. É possível utilizar como arquivo de entrada e exportar para diversos formatos.
3. De brinde, o usuário ganha um poderoso organizador pessoal que permite até a sincronização dos dados com seu smartphone ou tablet.
4. Por último mas não menos importante, o GitHub aceita diretamente um arquivo README.org como o README.md.

Arquivo .emacs

Configurações iniciais

Pacotes

É onde tudo acontece. Achei que seria melhor explicar com mais detalhes tudo o que acontece durante a execução do arquivo para que o usuário possa efetuar alterações básicas para deixar o Emacs mais ao seu gosto. As linhas que iniciam com ponto e vírgula indicam que são comentários e não serão interpretadas. Para um entendimento melhor, seria interessante que o usuário aprendesse um

pouco sobre a linguagem **emacs-lisp** (uma variação de lisp) de onde vem toda a flexibilidade do Emacs.

Alterações das opções iniciais

Deixei estas alterações no início pois, se for feita alguma alteração utilizando o menu **Options** e o usuário selecionar **Options/Save Options**, esta parte do arquivo **.emacs** será alterada. Ficando no início é mais fácil de visualizar e não causa tanta confusão.

```
(custom-set-variables
 '(cua-mode t nil (cua-base)) (ref:cua)
 '(custom-enabled-themes (quote (misterioso))) (ref:theme)
 '(indicate-empty-lines t) (ref:empty)
 '(show-paren-mode t) (ref:paren)
 '(tool-bar-mode nil)) (ref:tool)
```

Ativamos o CUA-mode (*cua*), inicializamos um tema (cores utilizadas para fundo, fontes e salientar diversas sintaxes no texto) diferente do original (*theme*), indicamos que linha vazias devem conter um símbolo no início para diferencia de linhas que possuam espaço (*empty*), dizemos que queremos uma visualização para abertura e fechamento de chaves, parentesis e colchetes (muito útil para programação) (*paren*) e, finalmente, que não desejamos ver a barra de ferramentas (as teclas de atalho são mais eficientes e nada que dois níveis do menu não resolvam) (*tool*).

Inclusão e atualização de fonte de pacotes

```
(require 'package)
(add-to-list 'package-archives
  '("melpa" . "http://melpa.milkbox.net/packages/")
  t)
(package-initialize)
```

Adiciona o repositório MELPA que contém um maior número de pacotes e com uma atualização constante.

```
(when (not package-archive-contents)
  (package-refresh-contents))
```

Atualiza o conteúdo das fontes de pacotes se não existe. Para você atualizar os pacotes, utilize o menu **Options/Manage Emacs Packages**. Na janela de gerenciamento de pacotes, pressione **U** para atualizar os pacotes (irá excluir o anterior e instalar a versão nova), **I** para instalar algum pacotes desejado (veja **.myemacs**), **D** para excluir algum pacote (atenção para o que você excluir) e,

quando tudo estiver pronto, pressione **X** para executar as ações de inclusão e exclusão.

Seleção e instalação dos pacotes pelo Emacs30

```
(defvar gbm-required-packages
  '(which-key
    hl-line+
    powerline
    hlinum
    hiwin
    ido-grid-mode
    ido-select-window
    imenu-anywhere
    smex
    pandoc-mode
    org-cua-dwim
    org-pandoc
    auto-complete
    smartparens
    goto-chg
    indent-guide
    theme-looper))
```

Não inclua nenhum pacote neste ponto. Utilize o arquivo **.myemacs** se deseja incluir outros pacotes.

```
;;-----
;; ## Instalação/atualização dos pacotes
;;
(mapc (lambda (p)
      (package-install p))
      gbm-required-packages)
```

Configurações da aparência e dos pacotes

Tamanho da janela

```
;;-----
;; # Configuração da aparência e pacotes
;;-----
;; ##Tamanho inicial da janela
;;
(setq initial-frame-alist
  '(
```

```

        (width . 130) ; characters
        (height . 40) ; lines
    ))

;;-----
;; ## Which-key
;;
(which-key-mode)
(which-key-setup-minibuffer)
(setq max-mini-window-height 10)
(setq which-key-idle-delay 0.5)
(set-face-attribute 'which-key-local-map-description-face nil :weight 'bold)

;;-----
;; ## Numeração de linhas
;;
(global-linum-mode t)

;;-----
;; ## Realçar linha do cursor
;;
(hl-line-mode t)
(toggle-hl-line-when-idle)
(set-face-attribute hl-line-face nil :background "Grey25")
(set-cursor-color "yellow")

;;-----
;; ## Realça numeração da linha do cursor
;;
(require 'hlinum)
(hlinum-activate)

;;-----
;; ## Realçar parêntesis
;;
(show-paren-mode)

;;-----
;; ## Esconde barra de rolamento ##
;;
(scroll-bar-mode -1)

;;-----
;; ## Ativa smartparens
;;
(show-smartparens-global-mode t)
;;(add-hook 'prog-mode-hook 'turn-on-smartparens-strict-mode)
;;(add-hook 'markdown-mode-hook 'turn-on-smartparens-strict-mode)

;;-----
;; ## Salva estado atual ao sair

```

```

;;
(require 'saveplace)
(setq-default save-place t)
(setq save-place-file (expand-file-name ".places" user-emacs-directory))

;;-----
;; ## Desabilita buffer de mensagem inicial
;;
(setq initial-buffer-choice
      t)

;;-----
;; ## Troca mensagem do buffer de rascunho
;;
(setq initial-scratch-message
      ";; Nada neste buffer será salvo. Use:\n;; Ctrl+x Ctrl+r / Ctrl+x Ctrl+f para ler um arquivo")

;;-----
;; ## Realça frame ativo
;;
(require 'hiwin)
(hiwin-activate)
(set-face-background 'hiwin-face "black")

;;-----
;; ## Configura powerline
;;
(powerline-center-theme)
(setq powerline-default-separator
      'wave)

;;-----
;; ## ido no modo grade
;;
(setq ido-enable-flex-matching t)
(setq ido-everywhere t)
(ido-mode t)
(ido-grid-mode t)
(global-set-key (kbd "C-x o") 'ido-select-window)
(global-set-key (kbd "<f4>") 'ido-select-window)

;;-----
;; ## Configura atalho *Ctrl+.* para imenu-anywhere
;;
(global-set-key (kbd "C-.") 'imenu-anywhere)

;;-----
;; ## Configura atalhos *Alt+x* e *Alt+X* para smex
;;

```



```

(global-set-key (kbd "M-x") 'smex)
(global-set-key (kbd "M-X") 'smex-major-mode-commands)

;;-----
;; ## Configura o autocomplete
;;
(ac-config-default)
(ac-linum-workaround)

;;-----
;; ## Configura o markdown
;;
;;(autoload 'markdown-mode "markdown-mode"
;;  "Major mode for editing Markdown files" t)
;;(add-to-list 'auto-mode-alist '("\\.text\\|" . markdown-mode))
;;(add-to-list 'auto-mode-alist '("\\.markdown\\|" . markdown-mode))
;;(add-to-list 'auto-mode-alist '("\\.md\\|" . markdown-mode))

;;-----
;; ## Configura pandoc
;;(load "pandoc-mode")
;;(add-hook 'org-mode-hook 'pandoc-mode)

;;-----
;; ## Indent guide
;;
(indent-guide-global-mode)

;;-----
;; ## Configura theme-looper
;;
(theme-looper-set-theme-set '(adwaita
                             deeper-blue
                             dichromacy
                             misterioso
                             tango-dark
                             tango
                             tsdh-dark
                             wheatgrass
                             wombat))

(theme-looper-set-customizations 'powerline-reset)
(global-set-key (kbd "C-\\") 'theme-looper-enable-next-theme)

;;-----
;; ## Configura goto last change
;;
(global-set-key (kbd "C-x .") 'goto-last-change)
(global-set-key (kbd "C-x ,") 'goto-last-change-reverse)

```

```

;;-----
;; ## Ctrl+x Ctrl+r abre lista de arquivos recentes
;;
(require 'recentf)
(recentf-mode t)
(setq recentf-max-menu-items 25)
(defun recentf-ido-find-file ()
  "Find a recent file using Ido."
  (interactive)
  (let ((file (ido-completing-read "Choose recent file: " recentf-list nil t)))
    (when file
      (find-file file))))
(global-set-key (kbd "C-x C-r") 'recentf-ido-find-file)

;;-----
;; Se o arquivo .myemacs existe, carregar
;; configurações do usuário
;;
(setq myconfig "~/myemacs")
(if (file-exists-p myconfig)
    (load-file myconfig))

;;-----
;; Define F3 para pesquisar e Shift+F3 para pesquisar próxima
(global-set-key (kbd "S-<f3>") 'isearch-backward)

;;-----
;; Define F3 para iniciar busca
;; F3 novamente para próxima ocorrência
;; Shift+F3 para ocorrência anterior
(global-set-key (kbd "<f3>") 'isearch-forward)
(define-key isearch-mode-map (kbd "<f3>") 'isearch-repeat-forward)
(define-key isearch-mode-map (kbd "S-<f3>") 'isearch-repeat-backward)

;;-----
;; ## Movimentação entre frames
;;
(windmove-default-keybindings) ;; 'meta);
;; Make windmove work in org-mode:
(add-hook 'org-shiftup-final-hook 'windmove-up)
(add-hook 'org-shiftright-final-hook 'windmove-right)
(add-hook 'org-shiftleft-final-hook 'windmove-left)
(add-hook 'org-shiftdown-final-hook 'windmove-down)

(setq org-CUA-compatible t)
(setq org-support-shift-select t)
(setq org-src-fontify-natively t)
(setq org-startup-truncated nil)

```

;;-----;; FIM DO ARQUIVO .emacs ;;-----

Arquivo .myemacs