

# Usando Emacs como “IDE” para Ruby on Rails

**Versión 0.1** (04/2007)

Gabriel Arellano  
gabrielarellano[arroba]gmail.com

(2007) Gabriel E. Arellano

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. The GNU Free Documentation License as applicable to this document can be found at: <http://www.gnu.org/copyleft/fdl.html>

# Usando Emacs como "IDE" para Ruby on Rails

## Obtener e Instalar Emacs

Descargar y/o descomprimir y/o instalar Emacs (esto dependerá de su S.O. y distribución)

## ECB (Emacs Code Browser)

ECB (Emacs Code Browser) ....

Descargar ECB de <http://ecb.sourceforge.net/> y descomprimirlo en el directorio que contiene las librerías de emacs (en mi caso `~/emacs.d/site-lisp/`).

Además deberá descargar y descomprimir en el mismo directorio las siguientes librerías que son necesarias para que ECB funcione:

- Semantic <http://cedet.sourceforge.net/semantic.shtml>
- Eieio <http://cedet.sourceforge.net/eieio.shtml>
- Speedbar <http://cedet.sourceforge.net/speedbar.shtml>

Al descomprimirlos quite los números de versión a los directorios para facilitar futuras actualizaciones (entonces no hará falta modificar el archivo `.emacs`)

La estructura del directorio `~/emacs.d/site-lisp/` debería ser algo así:

```
~/emacs.d/  
  + site-lisp/  
    + ecb/  
    + semantic/  
    + eieio/  
    + speedbar/
```

Modifique el archivo de configuración de Emacs (en mi caso `~/emacs`) agregando:

```
;; Definimos las rutas donde copiamos las librerías.  
(add-to-list 'load-path "~/emacs.d/site-lisp")  
(add-to-list 'load-path "~/emacs.d/site-lisp/ecb")  
(add-to-list 'load-path "~/emacs.d/site-lisp/eieio")  
(add-to-list 'load-path "~/emacs.d/site-lisp/semantic")  
(add-to-list 'load-path "~/emacs.d/site-lisp/speedbar")  
  
;; Cargamos ECB.  
(require 'ecb)  
(require 'ecb-autoloads)  
;; Desactivamos el Tip of the Day de ECB...  
(setq ecb-tip-of-the-day nil)  
;; Abrir los archivos con click del mouse...  
(setq ecb-primary-secondary-mouse-buttons (quote mouse-1--mouse-2))  
;; Que ECB se cargue automáticamente...  
(ecb-activate)  
  
;; Activamos Semantic. (Resaltado de sintaxis)  
(setq semantic-load-turn-everything-on t)  
(require 'semantic-load)
```

## Soporte Ruby on Rails

### Obtener e Instalar emacs-rails

Descargar el paquete emacs-rails de <http://rubyforge.org/projects/emacs-rails> y descomprimirlo en el directorio que contiene de las librerías de Emacs.

Además deberá descargar en el mismo directorio estas librerías que son necesarias para que emacs-rails funcione:

- Snippets (a la TextMate®)  
<http://www.kazmier.com/computer/snippet.el>
- Buscar de manera recursiva dentro de los directorios  
<http://www.webweavertech.com/ovidiu/emacs/find-recursive.txt>  
(recuerden cambiarle la extensión de **.txt** a **.el** luego de descargarlo)
- Permite ejecutar código Ruby dentro de Emacs:  
<http://svn.ruby-lang.org/cgi-bin/viewvc.cgi/trunk/misc/inf-ruby.el?view=co>
- El ruby minor mode  
<http://svn.ruby-lang.org/cgi-bin/viewvc.cgi/trunk/misc/ruby-mode.el?view=co>  
Ver error de las últimas revisiones del repositorio de ruby-lang.org
- Modo “elétrico”<sup>1</sup> en Ruby  
<http://svn.ruby-lang.org/cgi-bin/viewvc.cgi/trunk/misc/ruby-electric.el?view=co>

La estructura del directorio `.emacs.d` debería parecerse a esto:

```
~/emacs.d/  
+ site-lisp/  
  + ecb/  
  + semantic/  
  + eieio/  
  + speedbar/  
  + emacs-rails/  
- find-recursive.el  
- inf-ruby.el  
- ruby-electric.el  
- ruby-mode.el  
- snippet.el
```

---

<sup>1</sup> En Emacs ciertos caracteres son “elétricos” esto quiere decir que además de autocompletarse (por ej. al poner un `def` que agregue el `end`), reindentan la línea actual y en algunos casos insertan saltos de página.

Modifique el archivo de configuración de Emacs (en mi caso ~/.emacs) agregando al final:

```
.....

;; Activamos Rails Mode
(add-to-list 'load-path "~/.emacs.d/site-lisp/emacs-rails")

(require 'rails)

;; Activamos el resaltado de sintaxis en Ruby Mode...
(add-hook 'ruby-mode-hook 'turn-on-font-lock)

;; Activamos el autocompletar (Ruby Electric).
(defun try-complete-abbrev (old)
  (if (expand-abbrev) t nil))
(setq hippie-expand-try-functions-list
      '(try-complete-abbrev
        try-complete-file-name
        try-expand-dabbrev))
(add-hook 'ruby-mode-hook
          (lambda()
            (add-hook 'local-write-file-hooks
                      '(lambda()
                         (save-excursion
                           (untabify (point-min) (point-max))
                           (delete-trailing-whitespace)
                           )))
              (set (make-local-variable 'indent-tabs-mode) 'nil)
              (set (make-local-variable 'tab-width) 2)
              (imenu-add-to-menubar "IMENU")
              (require 'ruby-electric)
              (ruby-electric-mode t)
              ))
```

## Soporte para archivos *.rhtml*

Los archivos **.rhtml** tienen una mezcla de código ruby y código html....

Descargar el paquete *rinari* en cualquier directorio usando el comando:

```
svn checkout svn://viewvc.rubyforge.org/var/svn/rinari/trunk rinari
```

(a la fecha el proyecto no a liberado ningún paquete... sólo está disponible vía SubVersion...)

Copiar la carpeta *rinari* recién descargada al directorio que contiene de las librerías de Emacs. La estructura del directorio **.emacs.d** debería parecerse a esto:

```
~./emacs.d/  
+ site-lisp/  
  + ecb/  
  + semantic/  
  + eieio/  
  + speedbar/  
  + emacs-rails/  
  + rinari/  
  - find-recursive.el  
  - inf-ruby.el  
  - ruby-electric.el  
  - ruby-mode.el  
  - snippet.el
```

Agregar las siguientes líneas al final del archivo **.emacs**:

```
.....  
  
;; Soporte de archivos RHTML  
(add-to-list 'load-path "~/emacs.d/site-lisp/rinari")  
(add-to-list 'load-path "~/emacs.d/site-lisp/rinari/rhtml")  
  
(require 'rinari)
```

### Funcionamiento:

Este modo agrego las siguientes combinaciones de teclas:

<b>C-c C-s</b>	Open script/console in its own buffer
<b>C-c C-v</b>	Toggle between view and action
<b>C-c C-t</b>	Toggle between model/controller and test
<b>C-c C-r</b>	Invoke Rake
<b>C-c C-f</b>	Find config file
<b>C-c C-M-t</b>	Run current test file
<b>C-c C-S-t</b>	Run current test
<b>C-x C-M-f</b>	Find file in project

## Chiches varios...

### Control de versiones (con Subversion).

Descargar la extensión *vc-svn* de <http://svn.collab.net/repos/svn/trunk/contrib/client-side/vc-svn.el>

Copiar el archivo en el directorio que contiene de las librerías de Emacs. La estructura del directorio **.emacs.d** debería parecerse a esto:

```
~./emacs.d/  
+ site-lisp/  
  + ecb/  
  + semantic/  
  + eieio/  
  + speedbar/  
  + emacs-rails/  
  + rinari/  
- find-recursive.el  
- inf-ruby.el  
- ruby-electric.el  
- ruby-mode.el  
- snippet.el  
- vc-svn.el
```

Agregar las siguientes líneas al final del archivo **.emacs**:

```
;;;;  
  
;; Agregamos soporte de Subversion  
(require 'vc-svn)
```

### Funcionamiento:

Botón derecho del mouse sobre un archivo bajo control de versiones y elegir alguna opción del menú “Version Control”, o bien, usando las siguientes combinaciones de teclas:

- C-x v i**    *vc-register* -- add a new file to version control
- C-x C-q**   *vc-toggle-read-only* -- check in or out, depending on the current state
- C-x v v**    *vc-next-action* -- same thing, check in or out, depending on the current state
- C-x v ~**    *vc-version-other-window* -- look at other revisions
- C-x v =**    *vc-diff* -- diff with other revisions
- C-x v u**    *vc-revert-buffer* -- undo checkout
- C-x v c**    *vc-cancel-version* -- delete the latest revision
- C-x v d**    *vc-directory* -- show all files which are not up to date
- C-x v g**    *vc-annotate* -- show when each line in a vc file was added and by whom
- C-x v s**    *vc-create-snapshot* -- tag all the files with a symbolic name
- C-x v r**    *vc-retrieve-snapshot* -- undo checkouts and return to a snapshot with a symbolic name
- C-x v l**    *vc-print-log* -- show log (not in ChangeLog format)
- C-x v a**    *vc-update-change-log* -- update ChangeLog
- C-x v m**    *vc-merge*
- C-x v h**    *vc-insert-headers*
- M-x**        *vc-resolve-conflicts* -- pop up an ediff-merge session on a file with conflict markers

## Ruedita del mouse.

Agregar las siguientes líneas al final del archivo **.emacs**:

```
;;;;  
  
;; La ruedita ahora mueve el texto en lugar de solo hacer ruido...  
(defun up-slightly () (interactive) (scroll-up 5))  
(defun down-slightly () (interactive) (scroll-down 5))  
(global-set-key [mouse-4] 'down-slightly)  
(global-set-key [mouse-5] 'up-slightly)  
  
(defun up-one () (interactive) (scroll-up 1))  
(defun down-one () (interactive) (scroll-down 1))  
(global-set-key [S-mouse-4] 'down-one)  
(global-set-key [S-mouse-5] 'up-one)  
  
(defun up-a-lot () (interactive) (scroll-up))  
(defun down-a-lot () (interactive) (scroll-down))  
(global-set-key [C-mouse-4] 'down-a-lot)  
(global-set-key [C-mouse-5] 'up-a-lot)
```

### Funcionamiento:

**Ruedita Arriba /Abajo:** desplaza el texto de a 5 líneas.

**Shift + Ruedita Arriba /Abajo:** desplaza el texto de a 1 línea.

**Control + Ruedita Arriba /Abajo:** desplaza el texto de a pantallas completas.

## Recursos (de qué lugares me robé esto...)

### Screencasts (cómo podría quedar esto...)

- <http://platypope.org/yada/emacs-demo/>
- <http://emacsonrails.drozdov.net/>

### Emacs + Ruby + Rails

- Cómo usar Emacs con Rails (Wiki de RoR)  
<http://wiki.rubyonrails.org/rails/pages/HowToUseEmacsWithRails>
- Sitio del proyecto Rails Minor Mode  
<http://dima-exe.ru/rails-on-emacs>
- Emacs y RoR (Wiki de Emacs)  
<http://www.emacswiki.org/cgi-bin/wiki/RubyOnRails>
- Rinari (otro modo rails con mejor soporte .rhtml)  
<http://technomancy.us/article/emacsrails>

### ECB

- Sitio de ECB (Emacs Code Browser)  
<http://ecb.sourceforge.net/>

### Otros Chiches

- Control de Versiones en ECB  
<http://www.emacswiki.org/cgi-bin/wiki/VersionControl>  
<http://ecb.sourceforge.net/docs/Version-control-support.html>
- Ruedita del mouse  
<http://www.esdebian.org/article.php/2006051511570443>
- Plegado de bloques (code folding) ----- TO DO -----  
<http://emacs.wordpress.com/2007/01/16/quick-and-dirty-code-folding/>
- FileNameCache ----- TO DO -----  
<http://www.emacswiki.org/cgi-bin/wiki/RubyOnRails>  
<http://www.emacswiki.org/cgi-bin/wiki/FileNameCache>
- Mejor apariencia ----- TO DO -----  
<http://www.brockman.se/software/zenburn/zenburn.el>

# Anexo I

El archivo .emacs completo:

```
;; Definimos las rutas donde copiamos las librerias.
(add-to-list 'load-path "~/.emacs.d/site-lisp")
(add-to-list 'load-path "~/.emacs.d/site-lisp/ecb")
(add-to-list 'load-path "~/.emacs.d/site-lisp/eieio")
(add-to-list 'load-path "~/.emacs.d/site-lisp/semantic")
(add-to-list 'load-path "~/.emacs.d/site-lisp/speedbar")

;; Cargamos ECB.

(require 'ecb)
(require 'ecb-autoloads)
;; Desactivamos el Tip of the Day de ECB...
(setq ecb-tip-of-the-day nil)
;; Abrir los archivos con click del mouse...
(setq ecb-primary-secondary-mouse-buttons (quote mouse-1--mouse-2))
;; Que ECB se cargue automaticamente...
(ecb-activate)

;; Activamos Semantic. (Resaltado de sintaxis)
(setq semantic-load-turn-everything-on t)
(require 'semantic-load)

;; Activamos Rails Mode
(add-to-list 'load-path "~/.emacs.d/site-lisp/emacs-rails")

(require 'rails)

;; Activamos el resaltado de sintaxis en Ruby Mode...
(add-hook 'ruby-mode-hook 'turn-on-font-lock)

;; Activamos el autocompletar (Ruby Electric).
(defun try-complete-abbrev (old)
  (if (expand-abbrev) t nil))

(setq hippie-expand-try-functions-list
      '(try-complete-abbrev
        try-complete-file-name
        try-expand-dabbrev))

(add-hook 'ruby-mode-hook
          (lambda()
            (add-hook 'local-write-file-hooks
                      '(lambda()
                         (save-excursion
                           (untabify (point-min) (point-max))
                           (delete-trailing-whitespace)
                           )))
              (set (make-local-variable 'indent-tabs-mode) 'nil)
              (set (make-local-variable 'tab-width) 2)
              (imenu-add-to-menubar "IMENU")
              (require 'ruby-electric)
              (ruby-electric-mode t)
              )))
```

Continuación...

```
;; Soporte de archivos RHTML
(add-to-list 'load-path "~/.emacs.d/site-lisp/rinari")
(add-to-list 'load-path "~/.emacs.d/site-lisp/rinari/rhtml")

(require 'rinari)

;; Soporte de Subversion
(require 'vc-svn)

;; La ruedita ahora mueve el texto en lugar de solo hacer ruido...
(defun up-slightly () (interactive) (scroll-up 5))
(defun down-slightly () (interactive) (scroll-down 5))
(global-set-key [mouse-4] 'down-slightly)
(global-set-key [mouse-5] 'up-slightly)

(defun up-one () (interactive) (scroll-up 1))
(defun down-one () (interactive) (scroll-down 1))
(global-set-key [S-mouse-4] 'down-one)
(global-set-key [S-mouse-5] 'up-one)

(defun up-a-lot () (interactive) (scroll-up))
(defun down-a-lot () (interactive) (scroll-down))
(global-set-key [C-mouse-4] 'down-a-lot)
(global-set-key [C-mouse-5] 'up-a-lot)
```

## Anexo II (incompleto aun...)

Cheatsheet de teclas:

### **Rails**

<b>C-c C-s</b>	Open script/console in its own buffer
<b>C-c C-v</b>	Toggle between view and action
<b>C-c C-t</b>	Toggle between model/controller and test
<b>C-c C-r</b>	Invoke Rake
<b>C-c C-f</b>	Find config file
<b>C-c C-M-t</b>	Run current test file
<b>C-c C-S-t</b>	Run current test
<b>C-x C-M-f</b>	Find file in project

### **Control de Versiones**

<b>C-x v i</b>	vc-register -- add a new file to version control
<b>C-x C-q</b>	vc-toggle-read-only -- check in or out, depending on the current state
<b>C-x v v</b>	vc-next-action -- same thing, check in or out, depending on the current state
<b>C-x v ~</b>	vc-version-other-window -- look at other revisions
<b>C-x v =</b>	vc-diff -- diff with other revisions
<b>C-x v u</b>	vc-revert-buffer -- undo checkout
<b>C-x v c</b>	vc-cancel-version -- delete the latest revision
<b>C-x v d</b>	vc-directory -- show all files which are not up to date
<b>C-x v g</b>	vc-annotate -- show when each line in a vc file was added and by whom
<b>C-x v s</b>	vc-create-snapshot -- tag all the files with a symbolic name
<b>C-x v r</b>	vc-retrieve-snapshot -- undo checkouts and return to a snapshot with a symbolic name
<b>C-x v l</b>	vc-print-log -- show log (not in ChangeLog format)
<b>C-x v a</b>	vc-update-change-log -- update ChangeLog
<b>C-x v m</b>	vc-merge
<b>C-x v h</b>	vc-insert-headers
<b>M-x</b>	vc-resolve-conflicts -- pop up an ediff-merge session on a file with conflict markers