

Aprendiendo Emacs

Mario Román

<2018-05-25 Fri 18:34>

Emacs

Emacs es un editor de texto construido sobre un intérprete del lenguaje **Elisp** para hacerlo extensible. Cada acción del editor constituye un comando sobre el intérprete, por lo que podemos reescribir sus comandos o crear nuevos comandos simplemente programando sobre el intérprete.

<https://github.com/libreim/blog/raw/post-emacs/images/emacs.png>

Emacs editando este mismo artículo en Markdown

Emacs sirve como IDE para la mayoría de lenguajes de programación y como editor para programación literaria y ciencia reproducible; se integra con git y con el sistema de archivos y tiene extensiones disponibles para usarse como aplicación para organizar listas de tareas, leer el correo o servir como hoja de cálculo.

I use Emacs, which might be thought of as a thermonuclear word processor.

– **Neal Stephenson**, *In the Beginning...* was the command line.

En este artículo haré una referencia breve a todo lo que me ha ido sirviendo para aprender Emacs mientras enlazo a fuentes que tratan cada uno de los temas más extensamente. Como recursos generales para aprender Emacs, puedo recomendar:

- **.Emacs Tutorials** de *jekor*, explican todo lo necesario para aprender Emacs en 10 videotutoriales que pueden seguirse progresivamente.
- **Emacs Meetups** de Thoughtbot, que profundizan sobre temas concretos de uso de Emacs.

- [Emacs Rocks](#), vídeos breves sobre casos concretos de uso de Emacs.
- [Emacs wiki](#), una wiki que documenta todo lo relacionado con Emacs.
- [sachachua.com](#), donde se hace referencia periódicamente a noticias de Emacs, y tiene escritas hojas de referencia de atajos de teclado de Emacs.
- [C'est la z](#), otro blog en el que se tratan temas relacionados con Emacs.
- [GNU Emacs](#), la documentación oficial de Emacs.
- [El baúl del programador](#), que ha empezado una recopilación colaborativa de los comandos y paquetes más útiles para Emacs.

Instalación

Emacs puede encontrarse en la mayoría de gestores de paquetes, pero puede no estar en su versión más actualizada.

```
sudo apt install emacs
```

Para escribir este artículo estoy usando GNU Emacs 25.1, pero la última versión estable es la **24.5**. La versión estable actual puede descargarse desde [GNU](#).

Atajos de teclado

En Emacs se usa una [notación específica](#) para escribir un atajos de teclado. La mayoría de documentación que consultes usará **C-x** en lugar de **Control+x**; usará **C-x C-s** para indicar que debes dejar pulsado **Control** mientras pulsas **x** y **s**; y por último, usará **M-x**, donde la **M** se referirá a la tecla **Meta**. La tecla **Meta** en Emacs se refiere normalmente a dos opciones equivalentes, o bien pulsar **alt izq.** mientras se pulsa **x**, o pulsar **Esc** y luego pulsar **x**. En resumen:

Atajo	Descripción	Comando
C-n	Mantener control pulsado mientras se pulsa n	Next line
C-x C-s	Mantener control pulsado pulsando x y luego s	Save file
M-x	Mantener alt o pulsar esc para luego pulsar x	Execute Command
RET	Salto de línea, pulsar enter antes de seguir	

Usar atajos de teclado facilita mucho usar Emacs rápidamente después del tiempo de aprendizaje. [1]

Cuando abras el programa por primera vez, te ofrecerá seguir un tutorial de Emacs escrito en Emacs. El tutorial es muy útil para aprender a moverse dentro de Emacs, pero la mayoría de lo que cuenta no es especialmente fácil de aprender de una sola vez. Lo más chocante para un usuario nuevo puede ser el sistema de copiar-pegar; que de forma muy simplificada se resume en: [2]

- M-w copia.
- C-w corta.
- C-y pega.

Pero si no te convence, puedes usar **CUA Mode**, que te permite volver a usar C-c y C-v para copiar y pegar. Otros comandos útiles de aprender antes de empezar con nada más son C-x C-s para guardar y C-x C-f para abrir un archivo.

Documentación

Emacs es un editor autodocumentado, es decir, la documentación del editor puede consultarse dentro del propio programa. Para llamar a la ayuda se puede pulsar C-h ?, que nos dejará elegir si necesitamos ayuda sobre comandos, atajos de teclados, variables, licencias, etc. Especialmente útiles son:

- C-h c <atajo de teclado> nos da el nombre de la función que se ejecuta al pulsar esas teclas.
- C-h f <nombre de función> documenta la función.

Buffers y ventanas

Cada vez que abrimos un archivo, o pedimos un apartado de documentación, o abrimos la configuración, se abre un nuevo buffer. Un **buffer** es el equivalente a un documento o un espacio de trabajo en otros editores. Podemos movernos entre los buffers actualmente abiertos pulsando C-x <left> o C-x <right>; y podemos mostrar la lista de buffers actualmente abiertos con C-x C-b (¡en un nuevo buffer!).

Además de los buffers que tengamos actualmente abiertos, tenemos ventanas que los muestran. Podemos partir la pantalla de Emacs en varias ventanas con `C-x 2` y `C-x 3`, que la parten horizontal y verticalmente respectivamente. Para volver a quedarnos sólo con la ventana en la que está el cursor, podemos pulsar `C-x 1`; y para cambiar de ventana sobre la que actúa el cursor podemos usar `C-x o`.

Modos de Emacs

El comportamiento de Emacs sobre cada buffer que abra será distinto dependiendo normalmente de la extensión del archivo. Esto le permite colorear de manera distinta distintas sintaxis, o tener comportamientos específicos (indentación, atajos de teclado, formateo) cuando está editando cada lenguaje.

Cada una de estas formas de edición se llama **modo**, y el **modo** actual aparece resaltado entre paréntesis en la barra inferior de Emacs. El modo básico es **Fundamental**, pero para cada propósito existen modos específicos. Para casi todos los lenguajes de programación tendremos un modo. Existen, por ejemplo, **Ruby-mode**, **Python-mode** o **CC-mode**.

Además de los *modos mayores* de los que hemos hablado hasta ahora, existen *modos menores* que son opcionales y complementan a los modos mayores. Por ejemplo, mientras escribo este artículo estoy usando **Markdown** como modo mayor y **ARev** (**Auto-revert mode**) como modo menor.

Personalización

Prácticamente todos los parámetros que uses en Emacs pueden ser ajustados a tu necesidad. Desde los atajos de teclado hasta el tema de color y fuentes que usa el editor.

La forma más básica de editar todas estas configuraciones es `M-x customize group`, que accede a un menú en el que se pueden modificar todas ellas. Todos los cambios que aquí se hagan se guardarán en un archivo `.emacs` (o `init.el`, en las versiones nuevas de Emacs). Este archivo es la otra forma de configurar Emacs; el archivo de inicio `.emacs` contiene código en Emacs Lisp que se ejecutará al iniciar el editor y podemos incluir allí todo lo que queramos configurar. Algunos paquetes, por ejemplo, necesitarán configuración adicional que habrá que incluir en este archivo.

.Emacs #2 - Customizations and themes - jekor

Sistemas de paquetes

Melpa

Es conveniente añadir un repositorio más grande que el que trae GNU por defecto, y **MELPA** es uno de los repositorios de paquetes de Emacs más grandes y actualizados. El repositorio de MELPA se añade desde `M-x customize-group RET package`. Dentro de la pestaña de repositorios puede insertarse la dirección de MELPA:

Archive name: `melpa-stable`

URL or directory name: `https://stable.melpa.org/packages/`

Para salir de cualquiera de las pantallas de personalización se usa `q`.

Otra forma de conseguir este mismo efecto es añadirlo directamente a nuestro archivo de configuración (`.emacs` o `init.el`), como se indica en las **instrucciones de instalación** del repositorio.

Paquetes

Podemos listar los paquetes que podemos instalar usando `M-x list-packages` y podemos buscar entre los paquetes pulsando varias veces el comando `C-s`. Si pulsamos `i` al lado de uno de ellos se marcará para instalar y al pulsar `x` se ejecutará la instalación de todos los paquetes marcados.

.Emacs #3 - Installing packages and extensions - jekor

Paquetes útiles

Dired

Dired viene instalado por defecto con Emacs y permite navegar la estructura de directorios del sistema operativo. Podemos empezar a navegarla usando `M-x dired` y pulsando `RET` cada vez que queramos abrir un archivo o una carpeta.

Podemos además afectar a los archivos. Por ejemplo, si queremos eliminar algunos archivos, podemos marcarlos con `d` y eliminarlos definitivamente con `x`.

.Emacs #4 - Exploring the filesystem - jekor

org-mode

org-mode es un modo de Emacs que se creó originalmente para gestionar listas de tareas, agendas y calendarios; pero además, contiene en su interior un completo lenguaje de marcado. Permite exportar documentos a una gran variedad de formatos (pdf, html, latex o markdown) e incluir internamente trozos de código y ejecutarlos. Además, tiene un sistema de tablas en texto plano capaz de sustituir la hoja de cálculo para tareas sencillas. Por todo esto, puede ser usado en tareas como la ciencia reproducible o la programación literaria cuando Latex es demasiado complejo, ayudando además a manejar la bibliografía y los enlaces tanto externos como entre archivos.

Getting started with org-mode - Harry Schwartz

Especialmente útil para matemáticas es la **previsualización de Latex** y el poder incluir los paquetes de la **AMS** para marcar teoremas o definiciones.

<https://github.com/libreim/blog/raw/post-emacs/images/org-math.png> *Apuntes de matemáticas en org-mode*

magit

magit permite integrar Emacs con **git** fácilmente para incluir los commits desde dentro del mismo editor. Usando **magit-status** llegamos a una pantalla en la que podemos elegir qué ficheros añadir al commit con **s** [3] y visualizar las diferencias con el commit anterior usando **tab**; ejecutar el commit con **c** c, que nos mostrará el buffer con el mensaje de commit y por último usar **C-c C-c** para enviarlo. El push y pull los haremos desde **magit-status** con **P** u y **F** u, respectivamente.

Es útil asignar un atajo de teclado al comando **magit-status**, que es el que muestra la ventana desde la que controlamos el añadir y hacer commit de ficheros. Por ejemplo, podemos fijarlo en **f5** añadiendo a nuestro archivo de configuración:

```
(global-set-key (kbd "<f5>") 'magit-status)
```

Macros de teclado

Las macros de teclado nos dejan grabar una secuencia de acciones y volver a repetirla tantas veces como sea necesaria. Se puede empezar a grabar con **f3** y terminar la grabación y repetirla tantas veces como sea necesario con **f4**.

.Emacs #9 - Keyboard macros - *jekor*

Elisp

Emacs Lisp, o Elisp, es un lenguaje de programación diseñado específicamente para escribir un editor de texto. Facilita el tratamiento de texto y el manejo de archivos y buffers.

Podemos escribir scripts en Elisp que se encarguen de tareas repetitivas en nuestro editor de texto y asignarlas a atajos de teclado o ampliarlo con más funcionalidad. Un tutorial básico sobre Elisp es [Learn Emacs Lisp in 15 minutes](#).

Notas

[1] Realmente solo puedo decir que a mí me funciona y que en general los atajos de teclado parecen ser **mejores que usar el ratón**.

[2] El **sistema** que Emacs usa para esto es bastante más sofisticado.

[3] De hecho, podemos seleccionar qué párrafos dentro de un fichero queremos añadir al commit.