

21. EDITORES DE TEXTO

21.1. Introducción

Introducción

- Los editores de texto crean y modifican ficheros de texto *plano*
- Se emplea en programación y en configuración de sistemas
- Los procesadores de texto crean y modifican ficheros de texto con formato de fuente (negritas, cursivas, tipos de letra, etc), de página (interlineado, márgenes, etc) e imágenes

En cualquier Linux hay disponibles muchos editores
¿Cuál es mejor?

- Depende en buena parte de gustos personales
- Depende de dónde vayamos a usarlos
- Este es un asunto típico para *guerras de religión*



Bilo y Nano. Javier Malonda. (CC BY-NC-ND 3.0)

Tipos de editor de texto

1. Editores en modo gráfico

- Su curva de aprendizaje suele ser más suave
- Adecuados para trabajar como programador en un ordenador *estándar*, local y con gráficos

2. Editores en modo texto (editores de consola)

- Curva de aprendizaje más dura (excepto algunos muy sencillos/simples)
- Permiten trabajar en remoto con la misma facilidad que en local
 - Podemos administrar sin problemas nuestra máquina Linux p.e. desde un Windows prestado y con mala conexión. O incluso una PDA y un teléfono móvil
- Son los únicos disponibles en sistemas empujados, como routers
- Suelen ser los únicos disponibles en ordenadores a medio instalar, averiados, herramientas de rescate, etc

21.2. vi

vi

El editor *estándar* en Unix. Desarrollado por Bill Joy (Co-fundador de Sun Microsystems) en el año 1976.

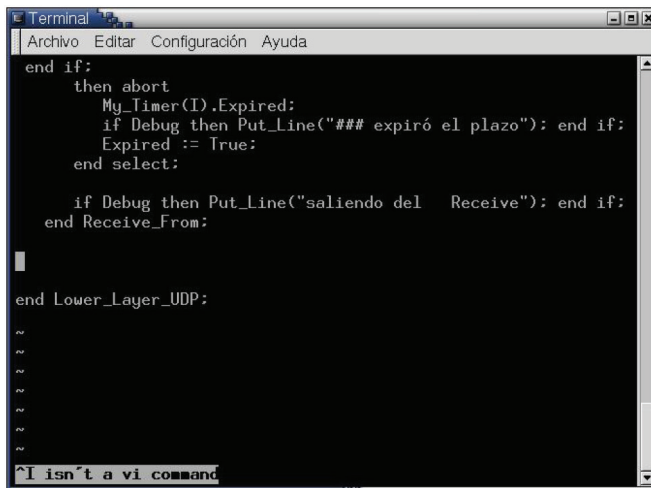
Hoy usamos clones, especialmente vim o más recientemente, neovim

- Si no nos gusta vi/vim/neovim, casi siempre podremos instalar otro
- Pero para poder instalar otro, suele ser imprescindible manejar al menos las órdenes elementales de vi

Ventajas

- Normalmente estará disponible y funcionando en cualquier máquina Unix

- Hay versiones para la mayoría de los SSOO (Windows, macOS...)
- Es muy flexible y potente, conociéndolo bien se puede trabajar a gran velocidad
- Pensado para sesiones remotas con malas conexiones en un *terminal* tonto de los años 70, el ADM-3A
- Si trabajamos en una máquina con gráficos, puede ser conveniente usar un vim en modo gráfico, mejor integrado con el escritorio. Permitirá usar el ratón, funcionará el portapapeles del escritorio y podrá tener menús, de utilidad para ordenes que aún no hemos memorizado
 - En Windows, gvim
 - En Linux, gvim ¹³
 - En OS X, MacVim (mvim)



Inconvenientes

- Interfaz de usuario muy anticuado, el usuario debe memorizar órdenes ¡donde hasta las mayúsculas son significativas!

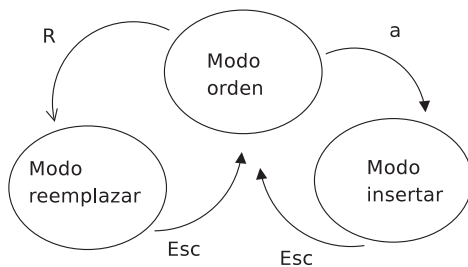
¹³ el nombre del paquete es vim-gtk

21.2.1. Modos de vi

Modos de vi

1. Modo orden (también llamado modo comando, modo normal)
En este modo guardamos el fichero, leemos otro, salimos, copiamos, pegamos, etc
2. Modo insertar (también llamado modo texto o modo entrada)
En este modo insertamos texto
3. Modo reemplazar (también llamado modo texto o modo entrada, sin distinguirlo del modo insertar)

En este modo reemplazamos texto



21.2.2. Órdenes imprescindibles

Órdenes imprescindibles

Desde la shell

```
koji@mazing:~$ vi nombre_fichero.txt
```

(Edita el fichero del nombre indicado. Si no existe, lo crea)

Desde vi

a	Pasar de modo orden a modo insertar
R	Pasar de modo orden a modo reemplazar
Esc	Volver a modo orden

x	Borrar un carácter
J	Unir la línea actual con la línea siguiente
:wq	Escribir el fichero y salir
:q!	Salir sin guardar el fichero

Este conjunto de órdenes es suficiente para editar cualquier fichero

21.2.3. Órdenes básicas

Órdenes básicas

:r nombre	leer un fichero
:w nombre	escribir fichero
u	Deshacer último cambio
ctrl r	Rehacer lo último deshecho
D	Borrar hasta final de línea
dd	Borrar línea actual
yy	copiar (yanc) línea
p	pegar lo ultimo copiado o borrado
.	Repetir la última orden
/patron	Busca un patrón (hacia adelante)
n	Repetir búsqueda
N	Buscar en dirección inversa a anterior
G	Ir a Final del archivo
5G	Ir a línea 5
%	Salta al paréntesis que se corresponda con el paréntesis actual (o llave, corchete...)

Casi todas las órdenes permiten anteponer un número, que indica cuántas veces se repetirá

dd	Borrar línea actual
10dd	Borrar 10 líneas
u	Deshacer un cambio
3u	Deshacer últimos 3 cambios
cw	Cambiar una palabra
5cw	Cambiar 5 palabras

21.2.4. Otras órdenes

Otras órdenes

0	ir a principio línea
\$	ir a fin línea
w	ir a siguiente palabra
b	ir a palabra anterior
r	Sustituir 1 carácter
cw	Cambiar palabra (change word)
dw	Borrar hasta fin palabra (delete word)
yw	Copiar palabra
*	Buscar palabra igual a la palabra sobre la que está el cursor
ma	Poner marca de texto a
mb	Poner marca de texto b
'a	ir a marca a
'b	ir a marca b
Ctrl G	Indicar línea actual
~	Pasar de may. a minusc. o al revés
:49,53 w! fichero	Escribir en fichero líneas de 49 a 53
.,53 w! fichero	Escribir en fichero desde línea actual hasta línea 53
:1,\$ s/digo/diego/g	Buscar todas las cadenas "digo" desde la línea 1 hasta el final, y reemplazarlas por "diego"
:set nu	Indicar el n° de línea
:set nonu	Desactivar n° de línea
:set ic	Ignore case (Insensible a mayus/min)
:set noic	Desactiva ic

Podemos configurar vim de forma persistente creando un fichero de configuración

- En Unix/Linux
~/.vimrc
- En Windows
c:\Archivos de programa\vim_vimrc (XP/Vista)
c:\Program File (x86)\vim_vimrc (Windows 7)

Por ejemplo, el fichero de configuración puede contener:

```
set vb
set ic
set tabstop=4
syntax on
```

Esto activa la *visual bell* (que elimina los molestos pitidos del terminal), ignora mayúsculas/minúsculas, fija el tabulador en 4 espacios y colorea el texto si reconoce la sintaxis

En Windows podemos añadir

```
set enc=utf-8
```

De esta forma, empleará por omisión la misma codificación que en Unix/Linux. Para más información sobre vi, consulta la página web *vi lovers home page*

21.3. Editores ligeros

Editores ligeros

Hemos visto que vi tiene muchas ventajas. Pero si nos *asusta* su interfaz de usuario y necesitamos un editor en modo texto, disponemos de editores ligeros como

- mcedit (editor del mc, midnight commander)
- nano (clon de pico)
- joe

21.4. Emacs / XEmacs

Emacs / XEmacs

Editor clásico en Unix. Uno de los más conocidos, se populariza a mediados de los 80

Emacs trabaja en modo texto, XEmacs en modo gráfico

Ventajas

- Completísimo, es mucho más que un editor. Permite leer correo, news, se integra con gran cantidad de herramientas...
- Módulos para muchos lenguajes de programación
- Da formato y color al fuente, con mucha calidad.
- Completamente personalizable (en lisp)
- Puede emular a vi

Inconvenientes

- Muy grande y pesado, consume muchos recursos.
- Su uso resulta complicado
- Aún para las tareas sencillas, tiene alguna peculiaridad que lo hace poco intuitivo al usuario actual

Usando emacs

```

\begin{figure}
\centerline{\includegraphics[width=5cm]{figs/xemacs}}
\end{figure}
\end{minipage} \hfill
\
\begin{minipage}{4cm}
\begin{itemize}
\item menu
\item pantalla edición
\item línea de modo
\item línea comandos
\end{itemize}
\end{minipage} \hfill
\end{hslide}

\begin{hslide}
\subsection{emacs $\neq$ xemacs}[]
\begin{figure}
\centerline{\includegraphics[width=9cm]{figs/emacs}}
\end{figure}
\end{hslide}

\begin{hslide}
\subsection{atajos de teclado}
\end{hslide}

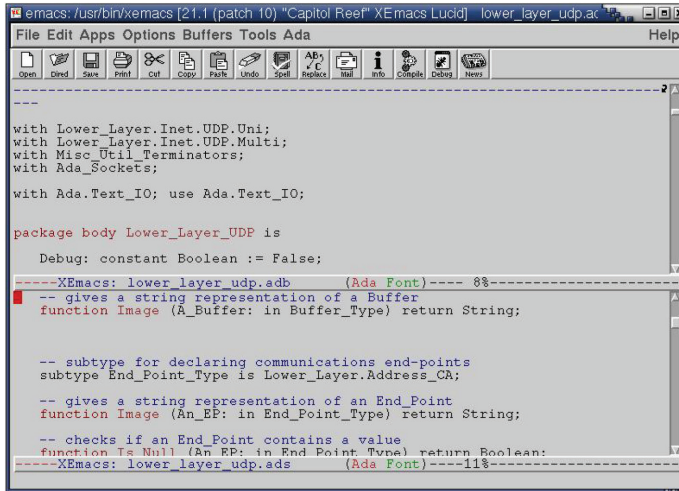
```

Wrote /home/jmp/laza/docencia/cursos/inaf.2002/editores/editores.tex

emacs ≠ xemacs

- menu
- pantalla edición

- línea de modo
- línea comandos



Atajos de teclado

- CTRL-K borrar línea
- ESC-X query-replace, ESC-X replace
- ESC-X goto-line
- CTRL-X-S salvar
- CTRL-X-F encontrar fichero
- CTRL-W=cortar, CTRL-Y=pegar
- CTRL-@=marca

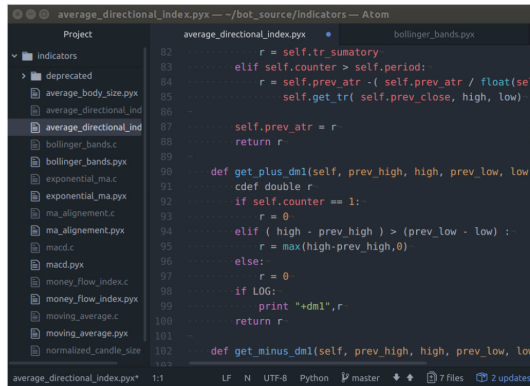
Enlaces sobre Emacs/XEmacs

- Emacs <http://www.gnu.org/software/emacs>
- XEmacs <http://www.xemacs.org>

21.5. Otros editores

21.5.1. Atom

Atom



- Editor de texto, libre y gratuito, disponible para Windows, Linux y macOS

Ventajas

- Más que un editor, es un IDE (Integrated development environment) con mucha funcionalidad: da formato, color, autocompleta, se integra con el compilador, con git, incluye colaboración en tiempo real (teletype)
- Ampliable mediante paquetes, que se pueden instalar desde el terminal (apm)
- Desarrollado por GitHub
- Moderno: la primera versión es de 2014, se ha vuelto muy popular

Inconvenientes

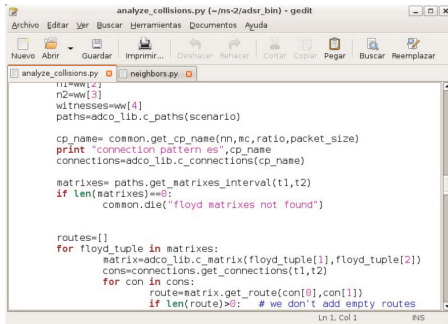
- Exige una sesión gráfica

enlaces

- <https://atom.io/>

21.5.2. gedit

gedit



Inconvenientes

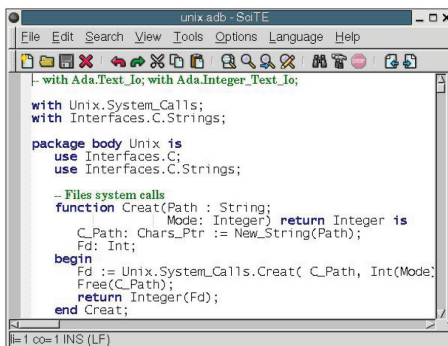
Editor de texto de propósito general, es el *block de notas* de gnome

Ventajas

- Muy sencillo y fácil de manejar
- Exige una sesión gráfica
- Ha mejorado mucho, pero sigue teniendo poca funcionalidad
- Tal vez no sea la mejor opción si tenemos disponible editores como atom, scite...

21.5.3. SciTE

SciTE



Editor de texto multiplataforma Ventajas

- Muy completo: Da formato, color, se integra con el compilador...
- Versiones para Win32 y X Window
- Muy fácil de manejar
- Es el editor de *anjuta*, el IDE de gnome

Inconvenientes

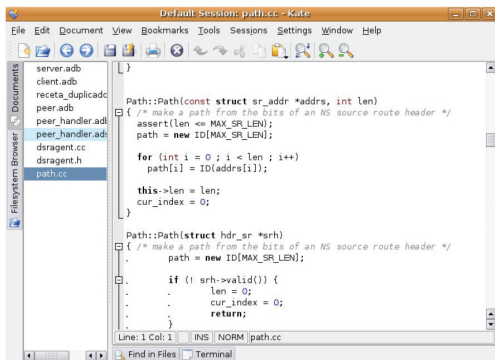
- Exige una sesión gráfica
- No muy extendido
- Hay editores más avanzados

enlaces

- <http://www.scintilla.org/SciTE.html>

21.5.4. Kate

Kate



Es el editor del escritorio KDE Ventajas

- Muy completo: Da formato, color, se integra con el compilador...
- Muy buen *pretty printing*
- Muy fácil de manejar

Inconvenientes

- Exige una sesión gráfica
- No muy extendido
- Hay cosas editores más avanzados hacen mejor
- Es necesario tener instalado KDE (o al menos buena parte)
- No disponible en otras plataformas

Enlaces

- <http://kate-editor.org>