
Ubuntu 13.10 Postinstallation

Un Quick Hacking para Bricolabs

Sergio Alvariño (salvari) <salvari@gmail.com>

2014-01-27

Tabla de contenidos

Introducción	2
Solución de problemas	3
Cursor parpadeante	3
Retardo en el click izquierdo del ratón.	3
Ajustes estéticos	4
Activar espacios de trabajo	4
Modificar formato de fecha	5
Cairo Dock	5
Instalación de Classic Menú	5
Gestión de paquetes	5
Instalación de Synaptic	5
Instalación de Aptitude	5
Herramientas de configuración de sistema	6
Instalación de CCSM	6
Instalar Unity Tweak y Gnome Tweak	6
TLP	6
Utilidades y programas básicos	6
Herramientas de compresión	6
Instalación de restricted extras	6
Codecs	7
Habilitar la opción de Abrir en Terminal en Nautilus	7
Java	7
Seguridad y privacidad	7
Ejecución de fixubuntu	7
Desactivado el reporte automático de errores	8
Desactivación de remote login y guest login	8
TOR	8
Firewall	9
Instalando programas	9
Creando lanzadores	9
Organizando el directorio ~/apps	10
Software de Google	10
Instalación de Chrome	10
Instalación de Chromium	11
Instalación de Google-Earth	11
Gráficos y dibujos	11
Gimp	11
Inkscape	11
Krita	11
MyPaint	12
Alchemy	12
Fotografía	12
Rawtherapee	12
Darktable	12
Luminance	12
Hugin	12
Desarrollo software	13

EMACS	13
VIM	13
Git	14
Mercurial	14
Subversion	14
D programming language	14
Perl	15
Squirrel SQL Client	16
Desarrollo Web	16
Web2py	16
Documentación	16
TeX/LaTeX	16
Docbook 5	16
Scribus	16
Diseño	16
LibreCAD	16
FreeCAD	17
OpenSCAD	17
Electrónica	17
Arduino IDE	17
Pingüino IDE	18
KiCAD	18
Geda	18
Fritzing	18
Astronomía	18
Stellarium	18
Where is M13?	18
Digital Universe	18
Mapas	19

Introducción

Este documento es una descripción de las operaciones de post instalación para Ubuntu 13.10 Saucy Salamander.

La instalación de Ubuntu se hizo en mi portátil, un ordenador Acer 5755G con las siguientes características:

- Core i5 2430M 2.4GHz
- NVIDIA Geforce GT 540M
- 8Gb RAM
- 750Gb HD

La instalación de Ubuntu realizada es una instalación personalizada para permitir editar la tabla de particiones. Las particiones en mi portátil son muy simples 70Gb para / y el resto en una partición montada en /store.

Ciertos directorios de trabajo (p.ej. /var) los convierto más tarde (a mano) en symlinks a /store para que no ocupen espacio en la partición root.

Lo mismo hago con ciertos directorios de mi home, por ejemplo el directorio de música, el de imágenes, el de descargas y la biblioteca de Calibre los tengo en /store con enlaces simbólicos a los mismos en home

Si no me equivoco cuento todo lo que hice para dejar la instalación de Ubuntu a mi gusto, aunque ya se sabe que esta tarea nunca acaba si te gusta trastear.

Además de algunos detalles de configuración, la mayor parte del documento trata de instalar programas, hay dos casos:

- Programas que ya están en Ubuntu pero no están disponibles en la última versión.

Para la mayor parte hay ppa (Personal Package Archive) disponibles, a veces incluso del equipo de desarrollo. Basta con añadir los ppa a nuestra lista de fuentes de software para instalarlos en Ubuntu.

- Programas que no están disponibles en Ubuntu ni siquiera como ppa.

Para instalar estos programas hay diversos métodos, más o menos complicados (e interesantes) para instalarlos.

Todo el documento está basado en la instalación que hago para mi portátil. Eso significa que es un sistema para un solo usuario y por eso cuando instalo un programa del segundo grupo, a veces simplemente lo instalo en mi directorio `~/apps`.

El que quiera aplicar la instalación de estos programas en un ordenador multiusuario tendrá que currarse una instalación en un directorio de sistema (lo más lógico sería usar `/usr/local` creo yo)

En cualquier caso cuando instalemos programas que no están disponibles en los orígenes de software para Ubuntu (oficiales o no) las únicas operaciones adicionales que nos pueden hacer falta son

- Añadir un lanzador al programa o hacer que el programa aparezca en el Dash
- Hacer que la base de datos de paquetes sepa que hemos instalado el programa.

Del primer caso hay múltiples ejemplos en el documento, del segundo tenemos el caso de TeX, que también incluimos en este documento.

Solución de problemas

Cursor parpadeante

Nada más terminar la instalación me apareció un problema bastante molesto de parpadeo del cursor. Se soluciona fácilmente desactivando el monitor secundario que creó la instalación.

System Settings::Displays: Desactivar el Unknow Monitor

Retardo en el click izquierdo del ratón.

Este problema apareció mucho más tarde después de estar unos días usando la nueva distribución. No tengo ni idea de la causa y la solución la encontré después de un par de horas navegando foros en la web.

Los síntomas eran bastante curiosos. Resumiendo: para que funcionara el click izquierdo del ratón había que hacer: Pulsar Botón Izquierdo - esperar (digamos que medio segundo al menos) - Soltar botón izquierdo

La solución pasa por desactivar la emulación de tres botones para el ratón. Mi ratón es un Microsoft de dos botones mas rueda central.

Para identificar correctamente el ratón lo mejor es lanzar un terminal y monitorizar el `syslog`:

```
$ sudo tail -f /var/log/syslog
```

Ahora desconectamos el ratón y lo volvemos a conectar para ver el código de producto

Veremos algo parecido a esto:

```
Feb 22 15:23:33 rasalhague kernel: [ 8664.300394] usb 2-1.2: new low-speed USB device found
Feb 22 15:23:33 rasalhague kernel: [ 8664.399363] usb 2-1.2: New USB device found, idVendor=8064, idProduct=0003
Feb 22 15:23:33 rasalhague kernel: [ 8664.399375] usb 2-1.2: New USB device string: 'Microsoft Basic Optical Mouse v2.0'
Feb 22 15:23:33 rasalhague kernel: [ 8664.399382] usb 2-1.2: Product: Microsoft Basic Optical Mouse v2.0
Feb 22 15:23:33 rasalhague kernel: [ 8664.399387] usb 2-1.2: Manufacturer: Microsoft
Feb 22 15:23:33 rasalhague kernel: [ 8664.403027] input: Microsoft Microsoft Basic Optical Mouse v2.0
Feb 22 15:23:33 rasalhague kernel: [ 8664.403406] hid-generic 0003:045E:00CB:0000-000000000000
Feb 22 15:23:33 rasalhague mtp-probe: checking bus 2, device 4: "/sys/devices/pci0000:00/0000:00:02.0/usb2/2-1.2/2-1.2:1.0/mtp.0"
Feb 22 15:23:33 rasalhague mtp-probe: bus: 2, device: 4 was not an MTP device
```

La parte que nos interesa es la que dice que tenemos instalado un "Microsoft Basic Optical Mouse v2.0", ojo que en mi caso el nombre de producto tiene un espacio al final y me llevó un rato darme cuenta.

Para ver la lista de dispositivos de entrada del servidor X también podemos hacer:

```
$ xinput --list
```

En mi caso veo que tengo el ratón con el identificador de producto que ya mencionamos asociado al dispositivo 12, para ver las propiedades puedo ejecutar cualquiera de los dos comandos siguientes:

```
$ xinput --list-props "Microsoft Microsoft Basic Optical Mouse v2.0 "
$ xinput --list-props 12
```

Para dejar la emulación de tres botones desactivada editamos el fichero `/usr/share/hal/fdi/policy/mouse-3button.fdi` como sigue:

```
$ sudo gedit /usr/share/hal/fdi/policy/mouse-3button.fdi
```

Y que contenga lo siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>

<deviceinfo version="0.2">
  <device>
    <match key="info.product" string="Microsoft Microsoft Basic Optical Mouse v2.0">
      <merge key="input.x11_options.Emulate3Buttons" type="string">false</merge>
    </match>
  </device>
</deviceinfo>
```

Con esta modificación a mi se me resuelve el problema.

Ajustes estéticos

Activar espacios de trabajo

1. En el Dash (**Super**+**A**) buscamos Appearance (Apariencia)

2. En la pestaña Behavior activamos la opción de Workspaces

Modificar formato de fecha

Cambio el formato del reloj para poder ver el día del mes y de la semana. Con botón derecho del ratón sobre el reloj accedemos a Settings.

Cairo Dock

Cuando Canonical decidió brindar una nueva experiencia de usuario, adoptando Unity, superé el trauma instalando Cairo Dock.

Tengo que admitir que hoy en día estoy bastante cómodo con Unity pero "le he cogido cariño" al dock y lo mantengo en mi portátil aunque ya no arranca por defecto al inicio.

```
$ sudo add-apt-repository ppa:cairo-dock-team/ppa
$ sudo add-apt-repository ppa:xubuntu-dev/xfce-4.10
$ sudo apt-get install cairo-dock
```

Para todos los que necesiten tener un menú de aplicaciones pero no quieran el Dock les recomiendo que prueben el appindicator Classic Menú. Así que...

Instalación de Classic Menú

```
$ sudo add-apt-repository ppa:diesch/testing
$ sudo apt-get update
$ sudo apt-get install classicmenu-indicator
```

Gestión de paquetes

Instalación de Synaptic

Generalmente uso la línea de comandos o Aptitude para instalar pero Synaptic también está muy bien, mucho mejor que el centro de software de Ubuntu para la mayor parte de las tareas. Desde Synaptic podemos visualizar fácilmente los orígenes de software que tengamos configurados para nuestro sistema

```
$ sudo apt-get install synaptic
```



Importante

Una vez configurado Synaptic es importantísimo entrar en las opciones y deshabilitar la opción de instalar los paquetes recomendados. En el menú Opciones en la opción Preferencias desactivamos la opción Considerar paquetes recomendados como dependencias

Instalación de Aptitude

```
$ sudo apt-get install aptitude
```

Una vez instalado Aptitude, lo configuro para que no instale los paquetes recomendados, arrancamos con **sudo aptitude** y en el menú Options seleccionamos Dependency Handling y desactivamos la opción Install Recommended Packages Automatically

Aptitude es un atavismo de mis tiempos Debianitas, cuando los hombres eran hombres y usaban la línea de comandos para casi todo, yo era uno de los timoratos que usaba un interfaz pseudo gráfico para la gestión de paquetes. En cualquier caso tiene la ventaja de que también lo podemos usar como sustituto de apt-get en la línea de comandos y una vez configurado no nos va a instalar los paquetes recomendados.

Herramientas de configuración de sistema

Instalación de CCSM

Esta aplicación nos permite configurar un montón de opciones de Compiz, el gestor de ventanas de Ubuntu. (aunque yo, la verdad, lo mantengo con los efectos al mínimo).

```
$ sudo apt-get install compizconfig-settings-manager
$ sudo apt-get install compiz-plugins-extra
```

Instalar Unity Tweak y Gnome Tweak

Nos dan más facilidades para configurar opciones de Unity y de Gnome.

```
$ sudo apt-get install unity-tweak-tool gnome-tweak-tool
```

TLP

Para mantener un consumo de batería ajustado en el portátil. Yo antes usaba Jupiter pero ya no existe.

```
$ sudo add-apt-repository ppa:linrunner/tlpsudo
$ apt-get updatesudo
$ apt-get install tlp tlp-rdw
$ sudo tlp start
$ sudo tlp stat
```

Utilidades y programas básicos

Herramientas de compresión

```
$ sudo apt-get install p7zip-rar p7zip-full unace unrar zip unzip \
sharutils rar uudeview mpack arj cabextract file-roller
```

Instalación de restricted extras

```
$ sudo apt-get install ubuntu-restricted-extras
```

Codecs

Me da la impresión de que todo esto lo han incluido en los restricted extras, pero por si acaso ejecutamos:

```
$ sudo apt-get install gstreamer0.10-plugins-ugly \
gstreamer0.10-ffmpeg libxine1-ffmpeg gxine mencoder libdvdread4 \
totem-mozilla icedax tagtool easytag id3tool lame \
nautilus-script-audio-convert libmad0 mpg321
$ sudo /usr/share/doc/libdvdread4/install-css.sh
```

Habilitar la opción de Abrir en Terminal en Nautilus

Si queremos tener la opción de abrir un terminal de comandos en el directorio que estamos visitando con Nautilus.

```
$ sudo apt-get install nautilus-open-terminal
```

Abrimos dconf y: hacemos click en Org, hacemos click en Desktop, hacemos click en Interface, habilitamos la opción can-change-accel

Una vez hecho esto cerramos dconf y reiniciamos nautilus:

```
$ nautilus -q
```

Java

Necesitamos tener Java instalado para que posteriormente nos funcionen varios programas y el plugin del navegador. Yo personalmente uso el openjdk:

```
$ sudo aptitude install icedtea-7-plugin openjdk-7-jre
```

Para desarrolladores hay que instalar también:

```
$ sudo apt-get install openjdk-7-jdk
```

Seguridad y privacidad

Ejecución de fixubuntu

Información reciente disponible en <https://fixubuntu.com>

Basicamente desactivamos las opciones que permiten al Dash buscar on-line en sitios como Amazon, E-Bay o Ubuntu Shop.

Lo que yo hice fue copiar esto a un terminal y ejecutar

```
V=`/usr/bin/lsb_release -rs`; \  
if [ $V \< 12.10 ]; then \  
    echo "Good news! Your version of Ubuntu doesn't invade your privacy."; \  
else gsettings set com.canonical.Unity.Lenses remote-content-search none; \  
if [ $V \< 13.10 ]; \  
then sudo apt-get remove -y unity-lens-shopping; \  
else gsettings set com.canonical.Unity.Lenses disabled-scopes \  
"['more_suggestions-amazon.scope', 'more_suggestions-ulms.scope', \  
  'more_suggestions-populartracks.scope', 'music-musicstore.scope', \  
  'more_suggestions-ebay.scope', 'more_suggestions-ubuntushop.scope', \  
  'more_suggestions-skimlinks.scope']"; \  
fi; \  
if ! grep -q productsearch.ubuntu.com /etc/hosts; \  
then echo -e "\n127.0.0.1 productsearch.ubuntu.com" \  
| sudo tee -a /etc/hosts >/dev/null; \  
fi; \  
echo "All done. Enjoy your privacy."; fi
```

Desactivado el reporte automático de errores

Si te molesta que Ubuntu esté todo el tiempo pidiendo permiso para enviar un informe.

Editamos el fichero `/etc/default/apport` y ponemos `enabled=0`

```
$ sudo gedit /etc/default/apport  
$ sudo restart apport
```

Desactivación de remote login y guest login

A mi no me hacen falta (no lo uso), además lo de permitir simultaneamente las opciones de login remoto y login anónimo no parece muy buena idea. Pero que conste que no lo he investigado en profundidad.

```
$ echo allow-guest=false \  
| sudo tee -a /etc/lightdm/lightdm.conf.d/50-unity-greeter.conf  
$ echo greeter-show-remote-login=false \  
| sudo tee -a /etc/lightdm/lightdm.conf.d/50-unity-greeter.conf
```

TOR

Para utilizar la red TOR, que garantiza la navegación anónima, descargamos el Bundle desde la página web del proyecto en el idioma deseado: <https://www.torproject.org/projects/torbrowser.html.en>

Descomprimos el fichero recibido en donde queramos instalar (en mi caso `~/apps/`)

Para que me funcionase he tenido que editar el fichero `~/apps/tor-browser/star-tor-browser` y añadir la línea

```
export GTK_IM_MODULE=xim
```


Tenemos que instalarnos un lanzador para TOR, podemos hacerlo en Cairo o integrarlo en el propio Ubuntu. En la siguiente sección describiremos como hacerlo.

Firewall

Configuración del cortafuegos.

```
$ sudo apt-get install gufw
$ gufw
```



TODO

Completar esta sección con la configuración del cortafuegos

Más información en <http://debianhelp.wordpress.com/2013/11/19/to-do-list-after-installing-ubuntu-13-10-aka-saucy-salamander-os-2/>

Instalando programas

Creando lanzadores

Como comentamos en la introducción a la hora de instalar programas nos vamos a encontrar con dos casos:

- Programas integrados en los orígenes de software de Ubuntu (ya sean oficiales o no).
- Programas no disponibles en paquetes para Ubuntu, o que decidimos instalar fuera de los orígenes de software para Ubuntu (por ejemplo para usar una versión más actualizada).

Para los programas en el segundo caso ya he comentado que yo suelo instalar en `~/apps`. Ya he explicado que lo hago así en mi portátil por que solo lo uso yo, si fuera un ordenador multiusuario lo propio sería usar un directorio por debajo de `/usr/local` para cada aplicación.

Una vez instalado el programa podemos crear el lanzador en el dock Cairo y/o integrarlo en los menús de Ubuntu.

La primera opción, añadir un lanzador al dock Cairo no tiene demasiada ciencia, basta con explorar el menú que aparece con el botón derecho del ratón, para ver como se hace. El Cairo nos permite crear separadores, carpetas de lanzadores que actúan como submenús, etc. etc. Yo suelo descargar un icono que esté bien para el programa (si es que no viene incluido) y descargarlo en el directorio `~/apps/icons`, para que no desmerezca el aspecto gráfico de Cairo.

La segunda opción se basa en que tanto Gnome, como Kde cumplen con el menú estándar especificado por FreeDesktop. Eso implica que basta con crear un fichero (con un formato xml específico) en el directorio `/usr/share/applications` para que esté disponible para todos los usuarios, o en el directorio `~/local/share/applications` para que solo le aparezca a nuestro usuario, lo lógico en nuestro portátil como solo instalamos para nuestro usuario es usar esta última opción.

Hay multitud de herramientas para crear este fichero. Nosotros vamos a usar `gnome-panel`.

```
sudo apt-get install --no-install-recommends gnome-panel
```

Una vez instalado podemos crear nuestro lanzador (por ejemplo para el programa TOR que acabamos de instalar, sin más que ejecutar:

```
$ sudo gnome-desktop-item-edit ~/.local/share/applications/ \
--create-new
```

Organizando el directorio ~/apps

Cuando instalamos programas en ~/apps nos podemos ahorrar trabajo organizándolo un poco.

Cuando actualizamos nuestros programas descargando nuevas versiones puede que nos vayan apareciendo directorios nuevos, por ejemplo para el IDE Arduino al descomprimir nos aparece el directorio: ~/apps/arduino-1.0.5. Si después instalamos la nueva versión beta para Arduino Yun nos aparecerá un directorio diferente, en el momento de escribir esto sería: arduino-1.5.6-r2. Si no cambiamos los nombres de directorios tendremos que estar editando los lanzadores, de lo contrario dificultamos el seguimiento de la versión instalada.

Nos puede quedar más organizado si:

- Creamos el directorio ~/apps/arduino

```
$ mkdir ~/apps/arduino
```

- Descomprimos las versiones que descargamos en este directorio.
- Creamos un link `current` a la versión actual (también se puede hacer en modo gráfico desde tu gestor de ficheros favorito).

```
$ cd ~/apps/arduino
$ ln -s ./arduino-1.5.6-r2 current
```

- En este punto el directorio ~/apps/arduino tendría esta pinta:

```
$ tree -L 1 -d arduino
arduino
### arduino-1.0.5
### arduino-1.5.6-r2
### current -> arduino-1.5.6-r2/
```

Ahora podemos crear nuestros lanzadores asociándolos al path ~/apps/arduino/current. Cuando descarguemos nuevas versiones actualizamos el link `current` y todos los lanzadores seguirán funcionando.

Software de Google

Instalación de Chrome

Instalamos el navegador de Google

```
$ wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub \
```

```
| sudo apt-key add -
```

Instalación de Chromium

También podemos instalar la versión libre de Chrome aunque Google está limitando algunas funcionalidades.

```
$ sudo apt-get install chromium
```

Instalación de Google-Earth



TODO

Completar esta sección con la instalación del Google-Earth

Gráficos y dibujos

Gimp

El software libre de manipulación de imágenes.

```
$ sudo add-apt-repository ppa:otto-kesselgulasch/gimp
$ sudo apt-get update
$ sudo apt-get install gmic gimp-gmic
```

Instalamos también el GPS (Gimp Paint Studio). Para añadir más funcionalidades de dibujo a Gimp. Descargamos el paquete con el GPS 2.0 desde aquí:

<http://www.ramonmiranda.com/2012/07/gps-20-disponible.html>

Y descomprimos en nuestro directorio `~/gimp-2.8`

Inkscape

El programa libre de gráficos vectoriales. Instalado desde los orígenes de software de Ubuntu.

```
$ sudo apt-get install inkscape
```

Un día de estos habrá que probar sozi y scour.

Krita

Un programa de dibujo completísimo. No hay que dejarse engañar por los menús minimalistas. Instalado desde los orígenes de software de Kubuntu.

```
$ sudo add-apt-repository ppa:kubuntu-ppa/backports
$ sudo apt-get install krita
```

MyPaint

Un programa de dibujo artístico muy completo.

Instalamos desde ppa

```
$ sudo add-apt-repository ppa:achadwick/mypaint-testing
$ sudo apt-get update
$ sudo apt-get install mypaint mypaint-data-extras
```

Alchemy

Un programa para buscar inspiración a la hora de ponerse a dibujar. Echad una mirada a la página web <http://al.chemy.org/gallery/> para entender lo que digo.

Pues como siempre, descargamos, descomprimos en ~/apps y creamos el lanzador a nuestro gusto.

Fotografía

Rawtherapee

Programa para revelado digital de imágenes raw.

```
$ sudo add-apt-repository ppa:dhor/myway
$ sudo apt-get update
$ sudo apt-get install rawtherapee
```

Darktable

Programa para revelado digital de imágenes raw.

```
$ sudo add-apt-repository ppa:pmjdebruijn/darktable-release-plus
$ sudo apt-get update
$ sudo apt-get install darktable
```

Luminance

Para generar imágenes HDR

```
$ sudo apt-get install luminance
```

Hugin

Para generar imágenes panorámicas o HDR.

```
$ sudo apt-get install hugin
```

Desarrollo software

EMACS

Si no eres de EMACS puedes pasar a la sección siguiente. Instalamos el paquete emacs y a mayores instalamos la última versión emacs24

```
$ sudo apt-get install emacs
$ sudo apt-get install emacs24
```

La verdad es que sólo estoy usando el emacs24, tengo que comprobar si es posible hacer una instalación más limpia.

Al margen de tener o no las dos versiones de Emacs instaladas, la última versión tiene un problema con el Saucy, no parece llevarse nada bien con el Ibus. A causa de ese problema no se pueden escribir por ejemplo: "x^2"

Para soslayar el problema tendremos que establecer la variable de entorno XMODIFIERS=""



TODO

Describir el fichero .emacs

Auctex

Instalación de yasnippets

Yasnippets es una utilidad para Emacs que uso bastante. Se puede instalar desde Ubuntu pero no me gusta como se instala, es muy difícil de configurar a tu gusto, así que la instalo fuera del sistema de paquetes.

```
$ mkdir ~/.emacs.d/plugins
$ mkdir ~/.emacs.d/snippets
$ cd ~/.emacs.d/plugins
$ git clone -recursive https://github.com/capitaomorte/yasnippet
```



Nota

COMPLETAR <https://github.com/capitaomorte/yasnippet>

Instalación AucTex



Nota

COMPLETAR

VIM

Si no eres de Emacs serás de Vi ¿no?. Instalamos VIM desde Ubuntu:

```
$ sudo apt-get install gnome-vim
```

Git

Lo del git-cola es opcional.

```
$ sudo add-apt-repository ppa:git-core/ppa
$ sudo apt-get update
$ sudo apt-get install git
$ sudo apt-get install git-cola
```

Además hacemos la configuración inicial del git:

```
$ git config --global user.name "Sergio Alvariño"
$ git config --global user.email "salvari@gmail.com"
$ git config --global core.editor emacs
$ git config --global color.ui true
$ git config --global credential.helper cache
$ git config --global credential.helper 'cache --timeout=3600'
```

Mercurial

Tortoise Hg no es imprescindible.

```
$ sudo apt-get install mercurial
$ sudo apt-get install tortoisehg tortoisehg-nautilus
```

Creado fichero ~/.hgrc con el contenido:

```
[ui]
username = Sergio Alvariño <salvari@gmail.com>
editor = emacs

[web]
cacerts = /etc/ssl/certs/ca-certificates.crt
```

Subversion

No lo uso para programar pero hay veces en que te bajas programas con el cliente.

```
$ sudo apt-get install subversion
```

D programming language

Visitamos la página e instalamos el paquete para Debian/Ubuntu (<http://dlang.org/download.html>).

DUB

También conviene que nos instalemos el DUB <http://code.dlang.org/download>, con este no hay que partirse la cabeza, nos descargamos el "binary tarball" para linux 64 bits y lo descomprimos en ~/bin

DUB será en breve el instalador oficial para el D. Conviene a empezar a usarlo cuanto antes. Además tiene ventajas obvias a la hora de gestionar las dependencias de nuestros programas.

vibe.d

Instalamos dependencias

```
$sudo apt-get install libevent-dev libssl-dev
```

D completion daemon - DCD

Instalamos también este programa que nos va a permitir configurar varios editores para la edición de código en D, de momento lo he probado solo en emacs y va muy bien.

```
$ git clone https://github.com/Hackerpilot/DCD
$ cd DCD
$ ./build.sh
```

Una vez compilados copiamos el `dcd-server` y `dcd-client` a nuestro path, por ejemplo a `~/bin`.

Para completar la instalación tenemos que crear el fichero `~/.config/dcd/dcd.conf`. Cada línea del fichero es un directorio para imports desde D, si has hecho la instalación como yo deberá tener estas dos líneas:

```
/usr/include/dmd/phobos
/usr/include/dmd/druntime/import
```

Perl

Perl viene instalado por defecto (se usa para un montón de cosas en linux)

En Debian, y consecuentemente en Ubuntu, hay versiones empaquetadas de la mayor parte de los módulos de Perl, pero no estarán completamente actualizadas. Como es más rápido instalar las versiones empaquetadas yo suelo instalar las empaquetadas la primera vez, después uso las herramientas internas de Perl para descargar desde CPAN las últimas versiones cuando me interesa.

He instalado los siguientes módulos de Perl en sus versiones empaquetadas para Debian/Ubuntu:

<code>libmodern-perl-perl</code>	Atajos para varios pragmas usados comunmente (ver http://modernperlbooks.com/books/modern_perl/)
<code>liblog-log4perl-perl</code>	Logs para Perl
<code>libtemplate-perl</code>	Template toolkit para Perl
<code>libmoose-perl</code>	Extensión para facilitar OOP en Perl.

Padre, un IDE para Perl

También me lo he instalado, aunque la verdad suelo usar Emacs para todo.

```
$ sudo aptitude install Padre
```

Squirrel SQL Client

Un cliente SQL muy versátil. Como va sobre Java me vale para el trabajo (Windows) y para uso personal.

Descargamos el instalador de la última versión (en el momento de escribir esto la 3.5.2) desde la página web <http://squirrel-sql.sourceforge.net/#installation>. Le damos permisos para ejecución y lo ejecutamos. El instalador ya se encarga de crear los lanzadores correspondientes. Si queremos instalarlo para todos los usuarios del sistema hay que ejecutarlo como root.

Yo lo uso sobre todo con MySQL, así que necesitamos descargarnos el driver para el MySQL desde la página web de MySQL <http://dev.mysql.com/downloads/connector/j/> y lo descomprimos (por ejemplo en ~/apps)

Una vez descomprimido basta con informar al Squirrel SQL de la ruta al conector. En el menú Windows escogemos View Drivers. Picamos dos veces en el driver de MySQL y añadimos el jar que descargamos en el paso anterior a la pestaña: Extra Class Path.

A partir de aquí ya podremos conectarnos a bases de datos MySQL.

Desarrollo Web

Wordpress y Drupal

Web2py

Descargamos la última versión desde la página web de la aplicación <http://www.web2py.com/init/default/download> (yo descargo la versión para usuarios normales - source code)

Como de costumbre descomprimos en nuestro directorio ~/apps, y tendremos el directorio ~/apps/web2py

Documentación

TeX/LaTeX

Esto va a ser largo

Docbook 5

Esto es fácil pero con dependencias.

Scribus

Instalar Scribus

Diseño

LibreCAD

```
$ sudo add-apt-repository ppa:librecad-dev/librecad-stable
$ sudo apt-get install librecad librecad-data
```


FreeCAD

```
$ sudo add-apt-repository ppa:freecad-maintainers/freecad-stable
$ sudo apt-get install freecad freecad-doc
```

OpenSCAD

He intentado instalarlo desde el ppa recomendado pero falla para Saucy

```
$ sudo add-apt-repository ppa:chrysn/openscad
```

Electrónica

Diversos programas.

Arduino IDE

Instalamos dependencias:

```
$ sudo aptitude install gcc-avr avr-libc
```

También tendremos que dar privilegios a nuestro usuario para que pueda acceder a los puertos usb, lo más sencillo es añadirlo al grupo dialout

```
$ sudo usermod -a salvari -G dialout
```

Descargamos el programa desde la página web <http://arduino.cc>

Descomprimos el programa en el directorio ~/apps

Ahora tenemos que crear un lanzador para el programa, podemos hacerlo facilmente en el cairo, es bastante intuitivo.

O podemos currarnos un lanzador para Ubuntu. Ubuntu almacena los lanzadores en ficheros xml en el directorio /usr/share/applications. Nos podemos coger un fichero de los existentes y currarnos uno o podemos instalar la antigua aplicación para crearlos y crear uno nuevo con los siguientes comandos:

```
$ sudo apt-get install --no-install-recommends gnome-panel
$ sudo gnome-desktop-item-edit /usr/share/applications/ --create-new
```

Una vez creado el lanzador nos aparecerá en el Dash cuando busquemos aplicaciones. Y también podemos arrastrarlo desde el dash al dock para dejarlos permanentes.

Evidentemente también aparecerá en el menú global del Cairo o el applet de Classic Menú si lo hemos instalado.

Pingüino IDE

Pingüino

KiCAD

KiCAD

Geda

Geda

Fritzing

Fritzing

Astronomía

Stellarium

Excelente programa para explorar los cielos nocturnos, lo instalamos directamente desde Ubuntu:

```
$ sudo aptitude install stellarium
```

El programa es muy fácil de usar. Solo comentar que desde el menú Options en la opción Tools se pueden descargar catalogos adicionales.

Where is M13?

Otra aplicación interesante de astronomía, se trata de un mapa tridimensional de nuestra galaxia implementado en java.

El programa libre y gratuito se puede descargar desde aqui: <http://www.thinkastronomy.com/M13/common/download.html>

La instalación también es muy sencilla, sencillamente lo descomprimos en nuestro directorio ~/apps. Basta con dar permisos de ejecución al fichero WhereIsM13.jar y ya lo podemos ejecutar desde el explorador de ficheros con doble click o crearnos un lanzador para el programa de la manera que más nos guste.

Digital Universe

Una aplicación que se basa en un visor de partículas en tres dimensiones para hacer implementar un mapa de nuestra galaxia y otro de nuestra galaxia y sus alrededores.

El programa puede descargarse desde la web del Museo Americano de Historia Natural: <http://www.amnh.org/our-research/hayden-planetarium/digital-universe/download>. Ojito que no es software libre. Es gratuito, pero si quereis usarlo para otra cosa distinta del uso personal hay que leerse la licencia.

Para instalarlo, lo mismo que el anterior, los descomprimos en nuestro directorio ~/apps/ y veremos que en el directorio ~/apps/Digital Universe tendremos el manual de la aplicación, el manual del visualizador de partículas en 3D, un par de scripts de ejemplos y dos scripts para visualizar los datos astronomicos:

- milkyway.sh
- extragalactic.sh

Como de costumbre, podemos crearnos lanzadores para los programas que nos interesen.

Mapas

MOBAC