Bitácora Linux Mint Tara 19.1

Sergio Alvariño [salvari@gmail.com](mailto:salvari@gmail.com)

abril-2019

Bitácora de mi portatil

Solo para referencia rápida y personal.

# Introducción

Mi portátil es un ordenador Acer 5755G con las siguientes características:

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

Mi portátil equipa una tarjeta *Nvidia Geforce GT540M* que resulta pertenecer a una rama muerta en el árbol de desarrollo de Nvidia.

Esta tarjeta provoca todo tipo de problemas de sobrecalientamiento, no importa que versión de Linux uses.

# Programas básicos

## Linux Mint

Linux Mint incluye sudo [[1]](#footnote-24) y las aplicaciones que uso habitualmente para gestión de paquetes por defecto (*aptitude* y *synaptic*).

Tampoco voy a enredar nada con los orígenes del software (de momento)

## Firmware

Instalamos el paquete intel-microcode desde el gestor de drivers.

Instalamos el driver recomendado de nvidia desde el gestor de drivers del *Linux Mint*. Ahora mismo es el *nvidia-driver-390*

Configuramos desde el interfaz del driver para activar la tarjeta intel.

Como a pesar de eso seguimos teniendo problemas de calentamiento:

apt install tlp  
tlp start  
apt install lm-sensors hddtemp  
apt install linux-tools-common linux-tools-generic  
cpupower frequency-set -g powersave  
apt install cpufrequtils

Referencias:

* <https://itsfoss.com/reduce-overheating-laptops-linux/>
* <http://www.webupd8.org/2014/04/prevent-your-laptop-from-overheating.html>

Después de un reinicio **frio** [[2]](#footnote-28) todo parece funcionar de nuevo.

## Parámetros de disco duro

Tengo un disco duro ssd.

Añadimos el parámetro noatime para las particiones de root y /home.

# /etc/fstab: static file system information.  
#  
# Use 'blkid' to print the universally unique identifier for a  
# device; this may be used with UUID= as a more robust way to name devices  
# that works even if disks are added and removed. See fstab(5).  
#  
# <file system> <mount point> <type> <options> <dump> <pass>  
# / was on /dev/sda5 during installation  
UUID=d96a5501-75b9-4a25-8ecb-c84cd4a3fff5 / ext4 noatime,errors=remount-ro 0 1  
# /home was on /dev/sda7 during installation  
UUID=8fcde9c5-d694-4417-adc0-8dc229299f4c /home ext4 defaults,noatime 0 2  
# /store was on /dev/sdc7 during installation  
UUID=0f0892e0-9183-48bd-aab4-9014dc1bd03a /store ext4 defaults 0 2  
# swap was on /dev/sda6 during installation  
UUID=ce11ccb0-a67d-4e8b-9456-f49a52974160 none swap sw 0 0  
# swap was on /dev/sdc5 during installation  
UUID=11090d84-ce98-40e2-b7be-dce3f841d7b4 none swap sw 0 0

Una vez modificado el /etc/fstab no hace falta arrancar:

mount -o remount /  
mount -o remount /home  
mount

En el printado de mount ya veremos si ha cargado el parámetro.

Pasamos el fstrim desde weekly a daily.

Seguimos instrucciones de [aquí](https://easylinuxtipsproject.blogspot.com/p/ssd.html).

Más concretamente de [aquí](https://easylinuxtipsproject.blogspot.com/p/ssd.html#ID8.2)

y cambiamos el parámetro de *swapiness* a 1.

## Fuentes adicionales

Instalamos algunas fuentes desde los orígenes de software:

sudo apt install ttf-mscorefonts-installer  
sudo apt install fonts-noto

Y la fuente [Mensch](https://robey.lag.net/2010/06/21/mensch-font.html) la bajamos directamente al directorio ~/.local/share/fonts

## Firewall

ufw y gufw vienen instalados por defecto, pero no activados.

aptitude install ufw  
ufw default deny  
ufw enable  
ufw status verbose  
aptitude install gufw

## Control de configuraciones con git

### Instalación de etckeeper

sudo su -  
git config --global user.email xxxxx@whatever.com  
git config --global user.name "Name Surname"  
apt install etckeeper

*etckeeper* hara un control automático de tus ficheros de configuración en /etc

Para echar una mirada a los *commits* creados puedes ejecutar:

cd /etc  
sudo git log

### Controlar dotfiles con git

Vamos a crear un repo de git para controlar nuestros ficheros personales de configuración.

Creamos el repo donde queramos

mkdir usrcfg  
cd usrcfg  
git init  
git config core.worktree "/home/salvari"

Y ya lo tenemos, un repo que tiene el directorio de trabajo apuntando a nuestro *$HOME*.

Podemos añadir los ficheros de configuración que queramos al repo:

git add .bashrc  
git add .zshrc  
git commit -m "Add some dotfiles"

Una vez que he añadido los ficheros que quiero tener controlados he puesto un \* en el fichero .git/info/exclude de mi repo para que ignore todos los ficheros de mi $HOME.

Cuando instalo algún programa nuevo añado a mano los ficheros de configuración al repo.

## Aplicaciones variadas

**Nota**: Ya no instalamos *menulibre*, Linux Mint tiene una utilidad de edición de menús.

Keepass2

Para mantener nuestras contraseñas a buen recaudo

Gnucash

Programa de contabilidad

Deluge

Programa de descarga de torrents (acuérdate de configurar tus cortafuegos)

Chromium

Como Chrome pero libre

rsync, grsync

Para hacer backups de nuestros ficheros

Descompresores variados

Para lidiar con los distintos formatos de ficheros comprimidos

sudo apt install keepass2 gnucash deluge rsync grsync rar unrar \  
zip unzip unace bzip2 lzop p7zip p7zip-full p7zip-rar chromium-browser

## Programas de terminal

Dos imprescindibles:

sudo apt install guake terminator

**TODO:** asociar *Guake* a una combinación apropiada de teclas.

## Dropbox

Lo instalamos desde el software manager.

## Chrome

Instalado desde [la página web de Chrome](https://www.google.com/chrome/)

## Varias aplicaciones instaladas de binarios

Lo recomendable en un sistema POSIX es instalar los programas adicionales en /usr/local o en /opt. Yo soy más chapuzas y suelo instalar en ~/apt por que el portátil es personal e intrasferible. En un ordenador compartido es mejor usar /opt.

### Freeplane

Para hacer mapas mentales, presentaciones, resúmenes, apuntes… La versión incluida en LinuxMint está un poco anticuada.

1. descargamos desde [la web](http://freeplane.sourceforge.net/wiki/index.php/Home).
2. Descomprimimos en ~/apps/freeplane
3. Creamos enlace simbólico
4. Añadimos a los menús

### Telegram Desktop

Cliente de Telegram, descargado desde la [página web](https://desktop.telegram.org/).

### Tor browser

Descargamos desde la [página oficial del proyecto](https://www.torproject.org/) Descomprimimos en ~/apps/ y ejecutamos desde terminal:

cd ~/apps/tor-browser  
./start-tor-browser.desktop --register-app

### TiddlyDesktop

Descargamos desde la [página web](https://github.com/Jermolene/TiddlyDesktop), descomprimimos y generamos la entrada en el menú.

## Terminal y Shell

Por defecto tenemos instalado bash.

### bash-git-promt

Seguimos las instrucciones de [este github](https://github.com/magicmonty/bash-git-prompt)

### zsh

Nos adelantamos a los acontecimientos, pero conviene tener instaladas las herramientas de entornos virtuales de python antes de instalar *zsh* con el plugin para *virtualenvwrapper*.

apt install python-all-dev  
apt install python3-all-dev  
apt install python-pip python-virtualenv virtualenv python3-pip

*zsh* viene por defecto en mi instalación, en caso contrario:

apt install zsh

Para *zsh* vamos a usar [antigen](https://github.com/zsh-users/antigen), así que nos lo clonamos en ~/apps/

cd ~/apps  
git clone https://github.com/zsh-users/antigen

También vamos a usar [zsh-git-prompt](https://github.com/olivierverdier/zsh-git-prompt), así que lo clonamos también:

cd ~/apps  
git clone https://github.com/olivierverdier/zsh-git-prompt)

Y editamos el fichero ~/.zshrc para que contenga:

# This line loads .profile, it's experimental  
[[ -e ~/.profile ]] && emulate sh -c 'source ~/.profile'  
  
source ~/apps/zsh-git-prompt/zshrc.sh  
source ~/apps/antigen/antigen.zsh  
  
# Load the oh-my-zsh's library.  
antigen use oh-my-zsh  
  
# Bundles from the default repo (robbyrussell's oh-my-zsh).  
antigen bundle git  
antigen bundle command-not-found  
  
# must install autojump for this  
#antigen bundle autojump  
  
# extracts every kind of compressed file  
antigen bundle extract  
  
# jump to dir used frequently  
antigen bundle z  
  
#antigen bundle pip  
  
antigen bundle common-aliases  
  
antigen bundle robbyrussell/oh-my-zsh plugins/virtualenvwrapper  
  
antigen bundle zsh-users/zsh-completions  
  
# Syntax highlighting bundle.  
antigen bundle zsh-users/zsh-syntax-highlighting  
antigen bundle zsh-users/zsh-history-substring-search ./zsh-history-substring-search.zsh  
  
# Arialdo Martini git needs awesome terminal font  
#antigen bundle arialdomartini/oh-my-git  
#antigen theme arialdomartini/oh-my-git-themes oppa-lana-style  
  
# autosuggestions  
antigen bundle tarruda/zsh-autosuggestions  
  
#antigen theme agnoster  
antigen theme gnzh  
  
# Tell antigen that you're done.  
antigen apply  
  
# Correct rm alias from common-alias bundle  
unalias rm  
alias rmi='rm -i'

Antigen ya se encarga de descargar todos los plugins que queramos utilizar en zsh. Todos el software se descarga en ~/.antigen

Para configurar el [zsh-git-prompt](https://github.com/olivierverdier/zsh-git-prompt), que inspiró el bash-git-prompt, he modificado el fichero ~/.zshrc y el fichero del tema en ~/.antigen/bundles/robbyrussell/oh-my-zsh/themes/gnzh.zsh-theme

### fish

**Nota**: No he instalado *fish* dejo por aquí las notas del antiguo linux mint por si le interesa a alguien.

Instalamos *fish*:

sudo aptitude install fish

Instalamos oh-my-fish

curl -L https://github.com/oh-my-fish/oh-my-fish/raw/master/bin/install > install  
fish install  
rm install

Si queremos que fish sea nuestro nuevo shell:

chsh -s `which fish`

Los ficheros de configuración de *fish* se encuentran en ~/config/fish.

Los ficheros de *Oh-my-fish* en mi portátil quedan en ~/.local/share/omf

Para tener la info de git en el prompt de fish al estilo de [bash-git-prompt](https://github.com/magicmonty/bash-git-prompt), copiamos:

cp ~/.bash-git-prompt/gitprompt.fish ~/.config/fish/functions/fish\_prompt.fish

**NOTA**: *fish* es un shell estupendo supercómodo con un montón de funcionalidades. Pero no es POSIX. Mucho ojo con esto, usa *fish* pero aségurate de saber a que renuncias, o las complicaciones a las que vas a enfrentarte.

### tmux

Esto no tiene mucho que ver con los shell, lo he instalado para aprender a usarlo.

sudo apt install tmux

## Utilidades

*Agave* y *pdftk* ya no existen, nos pasamos a *gpick* y *poppler-utils*:

Instalamos *gpick* con sudo apt install gpick

## Codecs

sudo apt-get install mint-meta-codecs

1. ya no incluye gksu pero tampoco es imprescindible [↑](#footnote-ref-24)
2. puede que haya un *bug* que hace fallar el sensor de temperatura si el portatil no arranca frio [↑](#footnote-ref-28)