

# GNU/Linux y el FOSS

Por Luis Daniel Casais Mezquida y  
Jose Antonio Verde Jimenez

Jornada de Institutos — T3chFest 2024

Grupo de Usuarios de Linux

[gul.uc3m.es](http://gul.uc3m.es) | @guluc3m



t3chfest

# Tabla de contenidos

- Free Open Source Software
- Licencias
- Git y GitHub
- GNU/Linux
- La Terminal (shell/TTY)

# Free Open Source Software

- **Free:** De uso gratuito
- **Open Source:** Su código fuente es público, y modificable

## Ventajas:

- Cualquiera puede contribuir
- Cualquiera puede ver las posibles vulnerabilidades
- Facilidad para usar y modificar (plugins, APIs, portar a diferentes plataformas...)

## Desventajas:

- El mantenimiento depende de los voluntarios (si no es de una empresa)
- Ciertos softwares no se benefician de tener su código fuente público (seguridad)



# Licencias

Normas que rigen (legalmente) el uso del software. Las más comunes en el OSS son:

- **GPLv3:**
  - Cualquiera puede copiar, usar y distribuir el código fuente
  - No hay garantías de ningún tipo
  - Cualquier otro código que lo incluya y/o modifique debe incluir la misma licencia
- **MIT:**
  - Cualquiera puede copiar, usar y distribuir el código fuente
  - No hay restricciones ni garantías de ningún tipo

<https://choosealicense.com/>

# Git y GitHub



**Git** es un sistema de control de versiones de archivos (VCS)

- Toma *snapshots* (“fotos”) de las versiones de los archivos, permitiendo ver las diferencias entre versiones
- Extremadamente útil y muy usado en la programación (archivos de texto plano)
- Agrupado en “proyectos”, o **repositorios**

**GitHub** es un servicio que permite guardar, publicar y compartir repositorios en la nube

- Permite la colaboración de cualquier persona\*
- Funcionalidades extra para éstos propósitos (*pull requests, issues, wikis, etc.*)
- El lugar de referencia donde conseguir código fuente
- La base del FOSS



# Ejemplos de proyectos FOSS

 [godotengine/godot](#)

Godot Engine – Multi-platform 2D and 3D game engine

 C++ ★ 81.3k  17.3k



 [obsproject/obs-studio](#)

OBS Studio - Free and open source software for live streaming and screen recording

 C ★ 54.4k  7.4k



 [blender/blender](#)

Official mirror of Blender

 C++ ★ 11.0k  1.7k



 [RetroPie/RetroPie-Setup](#)

Shell script to set up a Raspberry Pi/Odroid/PC with RetroArch emulator and various cores

 Shell ★ 9.9k  1.4k



# ¿Qué es GNU/Linux?

- Sistema Operativo
- Open-source y gratuito (FOSS)
- Muy usado
  - Escritorio: 4.1%
  - Servidor: 96.3%
  - Móvil (Android): 70.29%
- Rápido y seguro
- Sólido como una roca
- Infinitamente personalizable (y rompible)



[torvalds/linux](#)

Linux kernel source tree



C



134k



43.5k

# ¿Por qué un SO?

Un ordenador es una máquina que ejecuta instrucciones.

A éstas instrucciones se las agrupa en “programas”, instrucciones con una finalidad determinada.

¿Qué pasa si queremos ejecutar dos o más programas al mismo tiempo?

- Con un sólo procesador no puedes → lo simulas alternando la ejecución rápidamente
- Necesitas un sistema/programa que te permita escalar a tamaño  $n \rightarrow$  SO

El SO trae aparte más funcionalidades como librerías, *drivers*, interfaces, etc.

# Distribuciones (*distros*)

- Sistemas Operativos completos, con módulos y paquetes preinstalados
- Pensados para fines específicos
- Pueden (o no) traer GUI preinstalada
- Desktop Environment intercambiable

- Árbol de distros



# Package Manager

- Instala, desinstala, y actualiza aplicaciones y otros softwares
- Preinstalado con el SO
- Funciona en terminal y con GUI
- Puedes añadir y quitar repositorios
- También se encarga de las dependencias y librerías



# La Terminal (shell/TTY)

- Interfaz de texto entre el usuario y el SO
- Permite dar instrucciones **precisas** y **replicables**
- Con ella, eres básicamente Dios
- Dos modos de uso: usuario (\$) y superusuario / administrador (#)
  - Puedes correr un comando como superusuario con `sudo` (**S**uper **U**ser **DO**)
- Pasa opciones y parámetros con flags `--flag` ó `-f`
- Repetir comandos (navegar por el historial) con las flechitas
- Autocompletado con *tab*
- Las contraseñas no se muestran en pantalla



# Directorios

Un directorio es una dirección (*path*) dentro del sistema de ficheros del ordenador. Están compuestos por carpetas y subcarpetas, separados por `/`, formando un árbol.

Puedes especificar la dirección de forma absoluta, o relativa (a partir del directorio actual). E.g.:

- `/opt/gul/mi-carpeta` (absoluto)
- `gul/mi-carpeta` (estando en `/opt`)

Directorios comunes:

- `/`: Carpeta raíz del sistema
- `~`: Carpeta raíz del usuario (`/home/user`)
- `.`: Directorio actual
- `..`: Directorio padre (superior)

# Comandos básicos de terminal

- `ls` (lis): muestra los contenidos del directorio
- `cd dir` (change directory): moverse al directorio *dir*
- `cp origin destination` (copy): copia el archivo *origin* al path *destination*
- `rm file` (remove): elimina el archivo *file* (utilizar flag `-r` para carpetas)
- `mkdir dir` (make directory): crea el directorio *dir*
- `gedit file` (Graphical editor): abre el archivo *file* con el “bloc de notas”

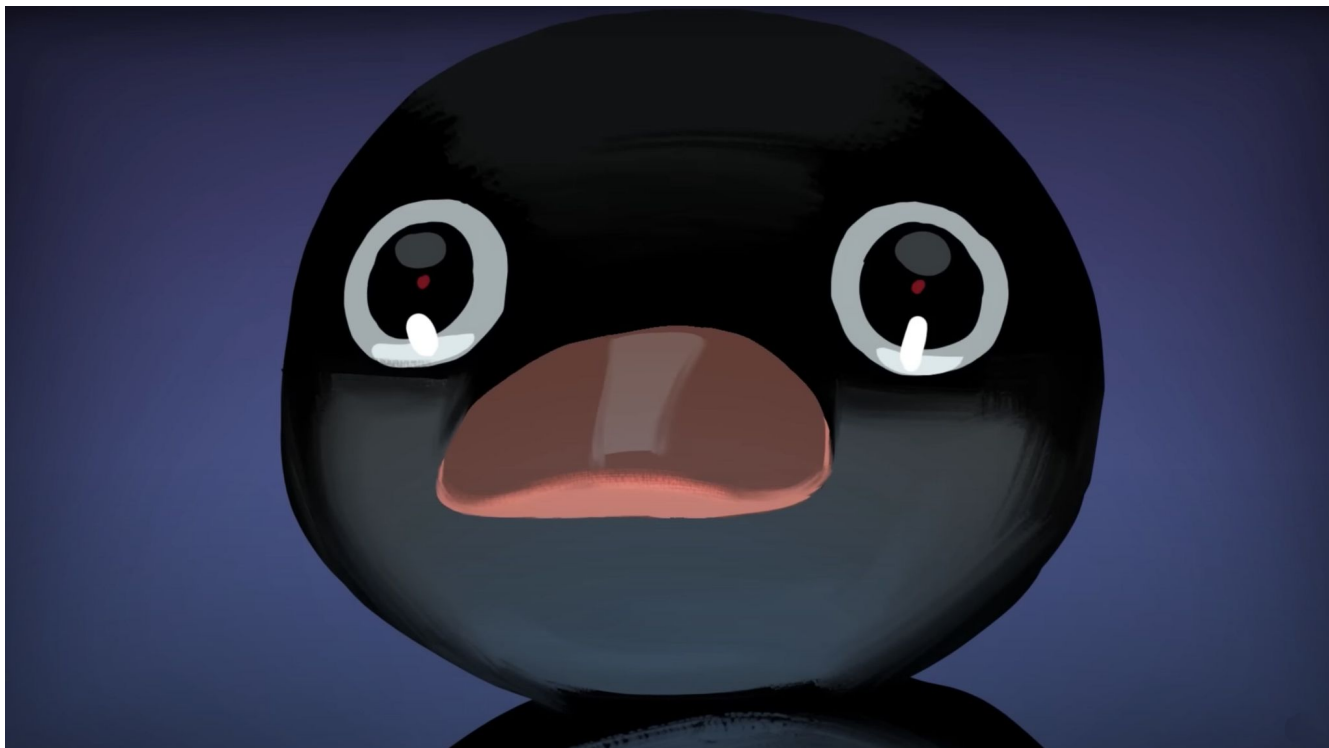
**Ahora... ¡Abrid el navegador y la terminal!**

[github.com/guluc3m/foss-example](https://github.com/guluc3m/foss-example)

# Más información

- [It's FOSS](#)
- [Arch Wiki](#)
- [aprendolinux.com](#)
- [github.com/acaldero/uc3m\\_linux](#)
- [github.com/guluc3m/tutorial\\_git](#)
- [github.com/guluc3m/linux404](#)
- [Te has instalado Linux... ahora, ¿qué?](#)
- [Linux en 90' para no desesperarse en las prácticas](#)

**¿Preguntas, reclamos, improperios?**



***fin***