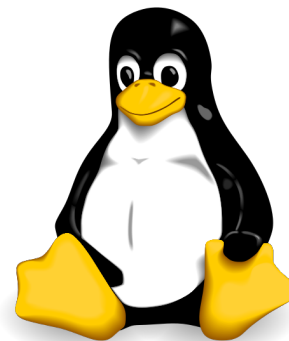


# Trabajar con GNU/Linux



© 2014 José Miguel Santos, Alexis Quesada,  
Fran Santana, Eduardo Rodríguez

# ¿Qué es GNU/Linux?

- Una versión libre de UNIX.

# Vale... ¿y qué es UNIX?

- Un sistema operativo creado en los años 1970 en los laboratorios de AT&T.
- Multitarea, multiusuario.
- años 1970-80: se le añade memoria virtual; redes e internet; interfaces gráficas...
- En los 80 se convierte en estándar internacional (normas POSIX).

# Historia de UNIX (1)

- En 1969, Dennis Ritchie y Ken Thomson, investigadores del **Bell Labs** de AT&T, consiguen que la empresa les apruebe un proyecto de diseño de un nuevo sistema operativo. Lo llaman UNIX en jocosos honor a su anterior proyecto MULTICS.



# Historia de UNIX (2)

- UNIX se desarrolla en AT&T. Como la empresa tiene prohibido vender software, no lo puede comercializar, pero permite a varias universidades usar su código fuente.
- A partir de 1976, UNIX se vuelve tremendamente popular en las universidades.

# Historia de UNIX (3)

- 1974. La Universidad de Berkeley comienza la versión Unix **BSD** con avances novedosos: internet, memoria virtual.
- 1982. Bill Joy, líder del desarrollo BSD, cofunda **Sun Microsystems**, que fabrica sistemas basados en Unix.
- 1983. Microsoft y SCO lanzan **Xenix**, la primera versión de Unix para PC.

# Historia de UNIX (4)

- 1983. Richard Stallman lanza el proyecto **GNU**, para construir una versión de Unix en software libre.
- Años 80. Los grandes fabricantes de hardware desarrollan sus propias versiones de UNIX: IBM(AIX), HP(HP-UX), SGI (Irix), DEC (Ultrix)... AT&T intenta comercializar su UNIX, pero el mercado ya está fragmentado.

# Historia de UNIX (y 5)

- 1987. Empieza la unificación y estandarización. Alianza de AT&T con Sun.
- Se crean especificaciones estándares y neutrales: Posix, OSF.
- 1991. Comienza el desarrollo de **Linux**.
- 1997. Apple basa en BSD el núcleo (Darwin) de su nuevo sistema operativo Mac OS X.



# GNU/Linux



Proyecto **GNU**: liderado por **Richard Stallman**  
**meta** → construir un UNIX totalmente en  
software libre.

A partir de 1983 GNU desarrolla  
compiladores (gcc), editores (EMACS),  
utilidades de shell, etc.

En 1991 **Linus Torvalds** (alumno de la  
Universidad de Helsinki) inicia la  
construcción de **LINUX**, un núcleo libre  
para PC.

**GNU+Linux** → versión totalmente  
funcional y libre de UNIX para PC.



# ¿Qué es software libre?

- Código fuente disponible de forma gratuita.
- El código se puede modificar libremente.
- El código se puede copiar y redistribuir sin pagar licencias.
- Desarrollo basado en una comunidad abierta.  
Liderazgo por méritos.

# Características de GNU/Linux

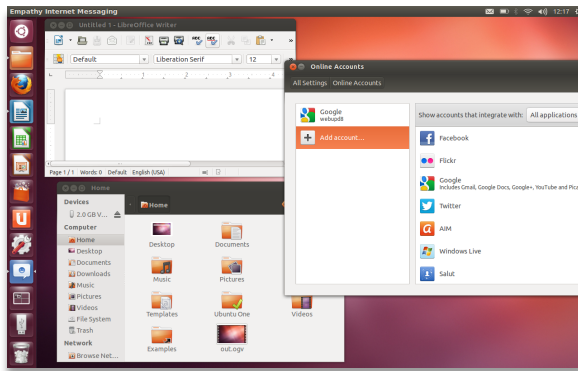
- Soporta la mayoría del software creado para UNIX.
- Funciona en múltiples plataformas (PC, móviles, routers, consolas...)
- Interoperabilidad con otros sistemas y formatos (NTFS, FAT, Samba...)
- Gran cantidad de aplicaciones libres (ofimática, edición de gráficos y vídeo, juegos...)
- Mucha diversidad de versiones o *distribuciones*.

# Distribuciones

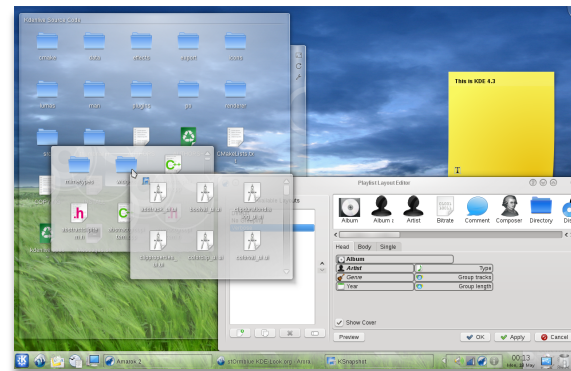
- Empresas y organizaciones se dedican a preparar versiones instalables de Linux:
  - RedHat, SuSe, Debian, Ubuntu...
  - Ahorran el esfuerzo de compilación, configuración, instalación, actualización de módulos, etc.
  - Aunque Linux es gratuito, las distribuciones no siempre lo son, debido a su valor añadido (ej. servicio de soporte técnico).

# Entornos gráficos

- Varias plataformas de escritorio. Las más populares son **Gnome** y **KDE**.
- Funcionan de forma similar a los escritorios de Windows/Mac.



Gnome 2



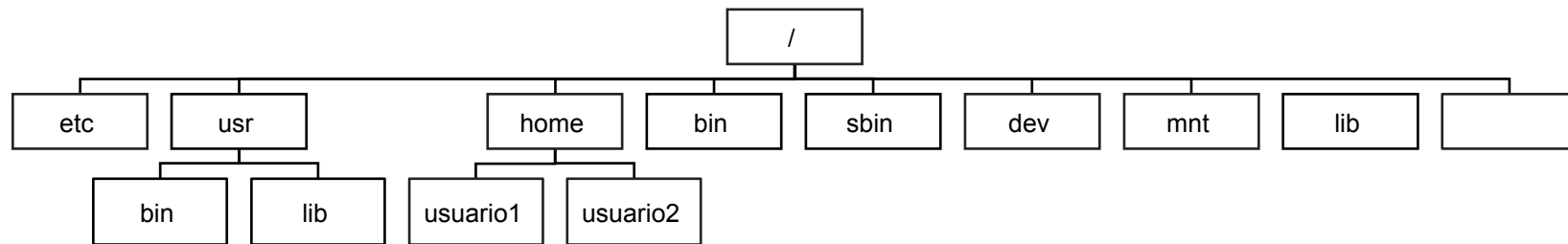
KDE 4

# ESTRUCTURA DE UNIX

# Elementos del sistema

- El núcleo o *kernel*
- El intérprete de órdenes o *shell*
- Los programas (estándares) del sistema
- Otras aplicaciones

# Organización de archivos



**/bin, /sbin, /usr/bin** -> Programas ejecutables (binarios)

**/etc** -> Ficheros de configuración del sistema, p.ej. passwd, fstab, /etc/X11, /etc/skel

**/usr** -> Ficheros importantes del sistema, p.ej. /usr/man, /usr/doc, /usr/doc, /usr/lib

**/home** -> Directorios de usuarios (puede variar según la configuración)

**/dev** -> Archivos especiales (archivos de dispositivos)

**/mnt** -> dispositivos montados (puede variar según la configuración)

**/lib** -> bibliotecas compartidas que utilizan los programas al ejecutarse

**/proc, /sys** -> archivos para obtener información de los procesos, el hardware, etc.



# Árbol de directorios

- Unix muestra el almacenamiento como un árbol con archivos y directorios.
- No se ven «unidades», ni «carpetas de red».
- Bajo esta apariencia simple, muchos de los directorios pueden estar en diferentes particiones, diferentes discos e incluso diferentes computadoras.
- → «montar» un dispositivo en un directorio.

# Nombres de archivos

- Similar a Windows, pero con algunas diferencias importantes.
- Distingue mayúsculas: “pepe”, “PEPE”, “Pepe”
- No existe el concepto de «extensión»
- El separador de rutas es: /
- No hay *unidades lógicas*: es un único árbol de directorios
- Los nombres de archivos con espacio van entre “” o bien usando espacio como “hola\ mundo”
- Otros caracteres especiales son: \\*, \?

# Tipos de archivos

- Archivos regulares.
- Directorios.
- Enlaces (*links*). Como los accesos directos.
- Archivos especiales: dispositivos, tuberías (*pipes*), sockets.

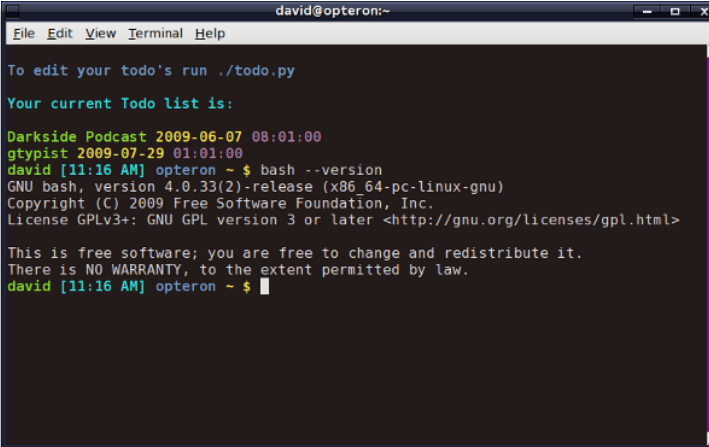
# Archivos de dispositivo

- UNIX nos permite tratar a todos los dispositivos físicos como ficheros:  
    /dev/mouse      /dev/console  
    /dev/sda        /dev/tty
- Las operaciones de E/S se realizan leyendo o escribiendo en esos ficheros.

¡Indaguemos en **/dev** y busquemos archivos especiales!

# El shell

- Es la interfaz de consola (CLI) de los sistemas UNIX.
- Acepta órdenes desde el teclado.
- Como el programa **cmd** de Windows.



```
david@opteron:~  
File Edit View Terminal Help  
To edit your todo's run ./todo.py  
Your current Todo list is:  
Darkside Podcast 2009-06-07 08:01:00  
gtypist 2009-07-29 01:01:00  
david [11:16 AM] opteron ~ $ bash --version  
GNU bash, version 4.0.33(2)-release (x86_64-pc-linux-gnu)  
Copyright (C) 2009 Free Software Foundation, Inc.  
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>  
  
This is free software; you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
david [11:16 AM] opteron ~ $
```

# El shell

- No forma parte del núcleo, se ejecuta como un proceso más.
- En UNIX existen diferentes shells, más o menos estándares:
  - sh (el estándar original)
  - **bash** (el que Linux trae por defecto)
  - ksh, csh

# ÓRDENES DEL SHELL

# Órdenes (*commands*)

- Sentencias que escribimos en la consola y que el shell ejecuta.
- Los nombres de las órdenes suelen ir en minúsculas: **ls**, **cp**, **rm**, **clear**...
- Forma de invocarlas:

*orden -opciones arg1 arg2 ...*

Ejemplo: `ls -l /etc`



# ¡Ayuda!

- Manual sobre una orden: **man *orden***
- Opción **--help**: (sólo en Linux):  
`ls --help`

# Órdenes (muy) básicas

| orden                 | acción  |
|-----------------------|---|
| <b>cd</b> <i>ruta</i> | Cambia el directorio de trabajo                     |
| <b>pwd</b>            | Muestra la ruta del directorio de trabajo           |
| <b>ls</b>             | Muestra el contenido del directorio actual          |
| <b>ls -l</b>          | Como ls, pero muestra los atributos de los ficheros |
| <b>ls *.c</b>         | Muestra los ficheros que acaban en “.c”             |
| <b>clear</b>          | Limpia la ventana de la consola                     |
| <b>whoami</b>         | ¿quién soy?   |
| <b>date</b>           | Fecha y hora actuales                               |

# Caracteres comodín

- Asterisco (\*) → vale por cero o más caracteres cualesquiera
- Interrogación (?) → un carácter cualquiera
- [a-h] → cualquiera de las letras entre “a” y “h”
- Ejemplo: ls -l [a-z]\*.doc

*Muestra la rutas que empiecen por una letra minúscula y acaben en “.doc”*

# Rutas especiales

- .. el directorio padre
- . el directorio actual
- ~ el directorio de nuestro usuario
- / el directorio raíz del sistema

# Trabajar con ficheros (1)

| orden  | acción  |
|--|---|
| <b>cp</b> <i>origen destino</i>  | Copia el fichero <i>origen</i> en <i>destino</i>  |
| <b>cp</b> <i>ruta1 ruta2 ruta3 destino</i>   | Copia los ficheros <i>ruta1</i> , <i>ruta2</i> , <i>ruta3</i> en el directorio <i>destino</i> |
| <b>cp -r</b> <i>ruta1 ruta2 destino</i>  | Copia recursivamente <i>ruta1 ruta2</i>   |
| <b>mv</b> <i>origen destino</i><br><b>mv [-r]</b> <i>ruta1 ruta2 ruta3 destino</i> | Igual que <b>cp</b> , pero mueve o renombra   |
| <b>rm</b> <i>ruta1 ruta2 ...</i>   | Borra los ficheros <i>ruta1</i> , <i>ruta2</i> ...  |
| <b>rm -r</b> <i>dir1 dir2 ...</i>  | Borra recursivamente los directorios, ¡ojo!   |

# Trabajar con ficheros (y 2)

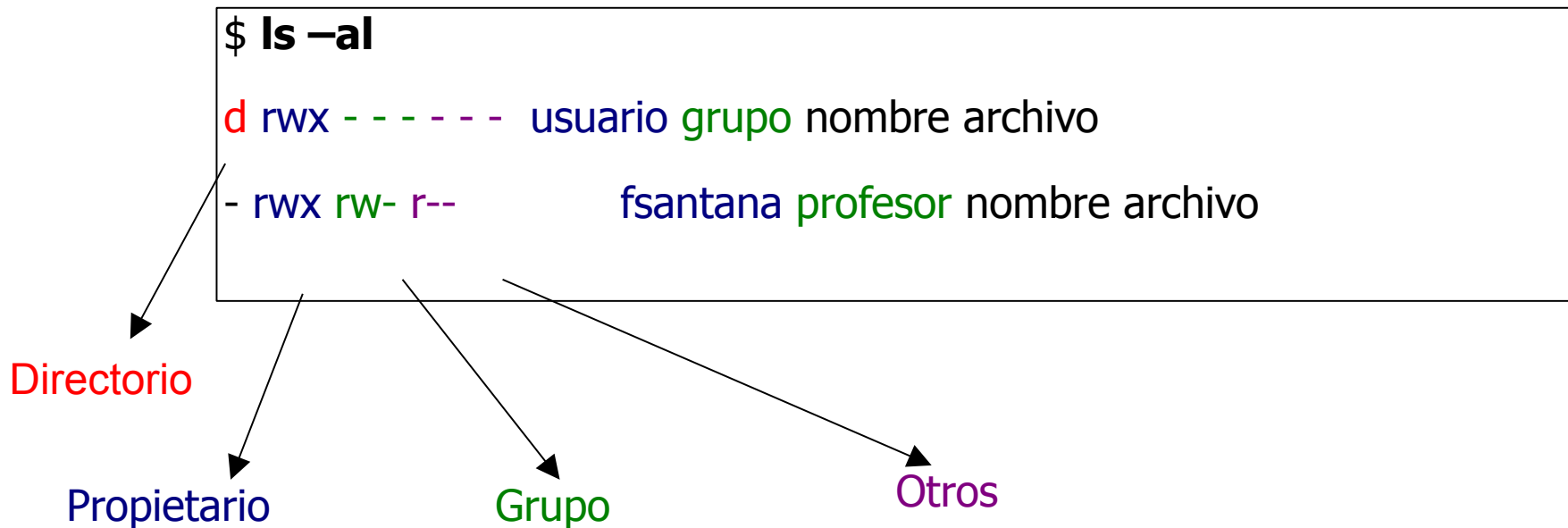
| orden  | acción  |
|--|---|
| <b>mkdir</b> <i>ruta</i><br><b>rmdir</b> <i>ruta</i> | Crea/Borra directorio. rmdir sólo funciona si el directorio está vacío.                                       |
| <b>cat</b> <i>fichero</i>                            | Muestra el contenido del fichero.   |
| <b>touch</b> <i>fichero</i>                          | Si el fichero no existe, lo crea con longitud cero.<br>Si el fichero existe, le actualiza la fecha de acceso. |

# Editar ficheros

- Editores de consola:
  - Los clásicos: **vi**, **emacs**  
*(muy potentes, pero cuesta aprender a usarlos)*
  - más sencillos (no estándares): **nano**, **pico**
- Editores gráficos:
  - De serie con Gnome: **gedit**
  - De serie con KDE: **kwrite**

# Permisos de acceso

- Sólo tres:  
lectura(R), escritura(W) y ejecución(X)





# Cambiar los permisos de un archivo

```
chmod [ugoa] [+ -=] [rwx] fichero1 fichero2...
```

u = propietario

g = grupo

o = otros usuarios

a = todos

```
chmod a+rwx semana.txt
```

```
chmod go-wx ordenls.txt
```

```
chmod a=r semana2.f
```

```
chmod 644 ordenls.txt
```

# Órdenes para procesar textos

| Orden  | acción   |
|--|--|
| <b>cat</b> <i>fichero</i>                                      | Muestra el contenido del fichero.  |
| <b>cat</b> <i>f1 f2 f3...</i>                                  | Muestra la concatenación de varios ficheros.   |
| <b>less</b> <i>fichero</i>                                     | Muestra el fichero en un visor que permite moverse arriba y abajo con los cursores, buscar texto con “/ <i>texto</i> ”, etc. |
| <b>head -N</b> <i>fichero</i><br><b>tail -N</b> <i>fichero</i> | Muestra las <i>N</i> primeras líneas de un fichero (head), o las últimas (tail).   |
| <b>grep</b> <i>cadena fichero</i>                              | Busca las líneas en <i>fichero</i> que contengan la <i>cadena</i> .  |
| <b>sort -[nr]</b> <i>fichero</i>                               | Muestra las líneas de <i>fichero</i> ordenadas.<br><b>-n</b> ordena numéricamente. <b>-r</b> invierte la ordenación.         |
| <b>wc</b> <i>fichero</i>                                       | Cuenta caracteres, palabras y líneas de <i>fichero</i> .   |

# Redirigir la entrada y la salida

- El texto que normalmente sale por la pantalla se puede *redirigir* a un fichero:

```
cat fichero >salida
```

```
ls -l /etc/a* >ficheros_de_etc.txt
```

- Usar **>>fichero** para no destruir el contenido original del fichero de salida.
- También se puede redirigir la entrada de datos:  

```
sort <desordenado >ordenado
```

# Tuberías (*pipes*)

- Se puede redirigir la salida de un proceso a la entrada de otro:

```
ls -Rl /etc | less
```

```
ls | wc -l      (para saber cuántos ficheros hay)
```

```
cat listas* | sort -n | head -1
```

*(ordena numéricamente el contenido de todos los ficheros listas\* y muestra sólo la primera línea ordenada)*

# Buscar archivos: find

| ejemplo                                   | acción  |
|---|---|
| <b>find</b> /home -name "*.c"             | Busca recursivamente desde /home todos los ficheros cuyo nombre encaje con "*.c"  |
| <b>find</b> . -iname "*.doc"<br>-mtime -7 | Busca desde el directorio actual todos los ficheros que terminen en ".doc", sin distinguir mayúsculas, y que se hayan modificado en los últimos siete días. |
| <b>find</b> . -type d -user pepe          | Busca todos los directorios por debajo de la carpeta actual que sean propiedad del usuario pepe.  |

# Empaquetar ficheros (tar)

- El programa **tar** empaqueta ficheros y directorios completos en un único archivo (ej. para adjuntarlo a un correo).
- Los ficheros suelen usar la terminación **.tar**

| ejemplo                           | acción  |
|-----------------------------------|---|
| <b>tar cf</b> pepe.tar /home/pepe | Empaqueta recursivamente el contenido de /home/pepe en el archivo “pepe.tar”. |
| <b>tar tf</b> pepe.tar            | Visualiza el contenido de “pepe.tar”.   |
| <b>tar xvf</b> pepe.tar           | Extrae el contenido de “pepe.tar”.  |

# Empaquetar y comprimir ficheros

- El programa **tar** con la opción **z** trabaja con ficheros comprimidos (similar a un RAR o un ZIP).
- Terminaciones habituales: **.tgz** y **.tar.gz**

| ejemplo                            | acción  |
|------------------------------------|---|
| <b>tar czf</b> pepe.tgz /home/pepe | Genera el archivo comprimido “pepe.tgz”.            |
| <b>tar xvzf</b> pepe.tar.gz        | Descomprime y extrae el contenido de “pepe.tar.gz”. |

# Comprimir y descomprimir

- **gzip / gunzip.** Utilidades de GNU.
- **zip / unzip.** Utilidades compatibles con PKZip, Winzip, etc.
- **Ojo:** el formato RAR es propietario y no se garantiza que pueda funcionar en Linux (aunque existe el programa rar/unrar).



# Otras órdenes útiles

| Orden                                      | Acción   |
|--|--|
| <b>passwd, yppasswd</b>                    | Cambia la contraseña del usuario.  |
| <b>ln -s <i>original</i> <i>enlace</i></b> | Crea un <i>enlace</i> (acceso directo o alias) al fichero <i>original</i> .                |
| <b>shutdown -h now</b>                     | Apaga el sistema de inmediato.   |
| <b>shutdown -r +5</b>                      | Reinicia el sistema después de una espera de 5 minutos.                                    |
| <b>ssh</b> usuario@micacharro.com          | Abre una conexión remota con la cuenta <i>usuario</i> a la máquina <i>micacharro.com</i> . |

# Procesos

- Proceso = programa en ejecución.
- **PID** = cada proceso tiene un identificador único (un número entero).

| Orden                  | Acción  |
|------------------------|---|
| <b>ps</b>              | Muestra los procesos de la terminal actual.                                   |
| <b>ps -e</b>           | Muestra todos los procesos del sistema.                                       |
| <b>kill <i>pid</i></b> | Elimina el proceso con identificador <i>pid</i> .                             |
| <b>top</b>             | Lista de los procesos que más recursos consumen. Se actualiza en tiempo real. |

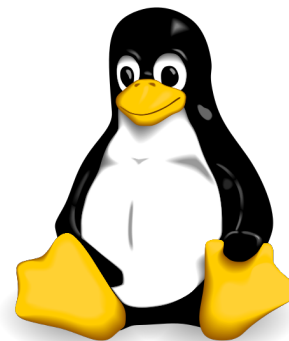
# Procesos en segundo plano

- Sintaxis: *orden* &
- Lanza un programa como un proceso en segundo plano. La terminal queda libre para seguir haciendo otras tareas.
- Ejemplo: `gedit` &

# Procesos

- Si hemos lanzado un programa desde la consola, se puede alterar su ejecución con estas combinaciones de teclas:
- **Ctrl-C** lo mata.
- **Ctrl-Z** lo detiene y libera la consola.
- **bg** lo reanuda y la consola sigue libre (como si lo hubiéramos lanzado con **&**).
- **fg** lo reanuda y vuelve a acaparar la consola.

# FIN



© 2014 José Miguel Santos, Alexis Quesada,  
Fran Santana, Eduardo Rodríguez