There are no translations available.
Descubre todo sobre X Window, la interfaz gráfica (GUI) para sistemas UNIX y la utilizada por GNU/Linux.

# 1 Introducción

Estamos acostumbrados a encender el ordenador y acceder, previa validación de usuario y contraseña, a un entorno gráfico de escritorio que nos proporciona todo lo necesario para nuestro trabajo: aplicaciones educativas de todo tipo, aplicaciones ofimáticas, herramientas para navegar por Internet, herramientas administrativas, conexiones preparadas para acceder a unidades de red, juegos, etcétera.

Pero, detrás de estos escritorios hay todo un entramado software que, trabajando de forma coordinada, nos permiten disponer de estos entornos integrados. Software que gestiona de qué forma nuestro sistema dialoga con los dispositivos físicos de entrada/salida, software que gestiona la forma como se muestran las ventanas en las que veremos nuestras aplicaciones, software que permite la validación del usuario contra la máquina local o el servidor de red, y, en general software que permite que dispongamos de una forma amigable de trabajar haciendo transparente para el usuario todos estos programas que se están ejecutando por detrás.

En el artículo analizamos estas capas de software y su interrelación, utilizando como base la distribución GNU/Linux Ubuntu 9.04 (Jaunty).

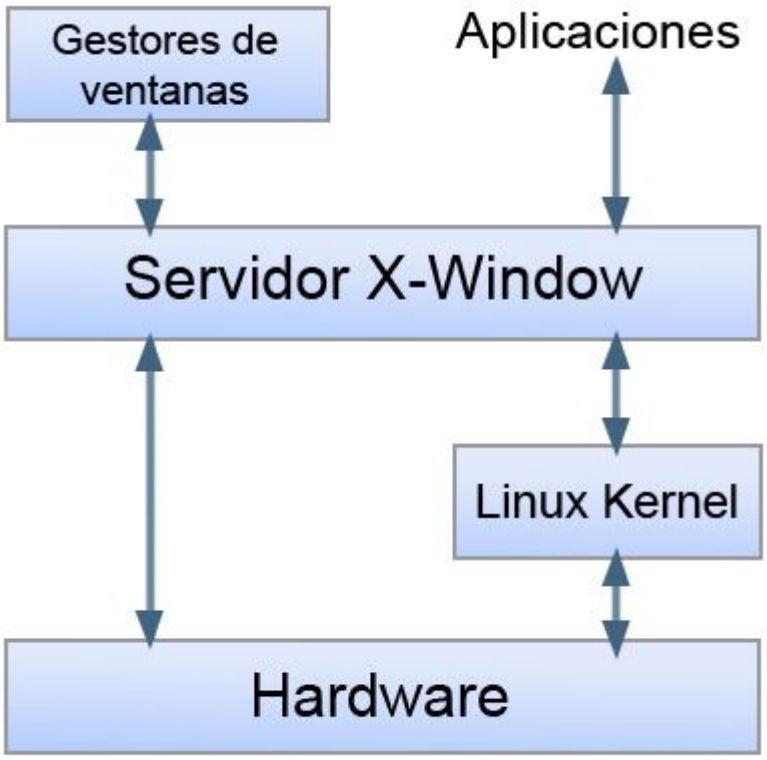
# 2 ¿Qué es X.Org?

X Window es la interfaz gráfica (GUI) para sistemas UNIX y la utilizada por GNU/Linux. Este sistema fue desarrollado por el MIT en los años 80 y define un estándar para la representación gráfica en cualquier sistema. Entre sus muchas características resaltamos que permite la ejecución remota de aplicaciones gráficas a través de redes heterogéneas (máquinas diferentes, sistemas operativos diferentes).

X Window System (abreviado **X11** o **X**) define el estándar para dibujar e interactuar con el usuario a través de ratón y teclado desde la interfaz gráfica de GNU/Linux. Tiene estructura

cliente-servidor. El cliente ejecuta la aplicación X y el servidor gestiona la comunicación con el hardware (tj gráfica, monitor, ratón, teclado) y procesa los gráficos. De esta forma el servidor gráfico X se ejecuta localmente y las aplicaciones X pueden ejecutarse de forma remota desde las máquinas cliente.

El servidor X más utilizado en GNU/Linux es el XFree86, desarrollado y distribuido de forma gratuita por el XFree86 Project (miembro del X Consortium).




Otra implementación del estándar X11 que permite la ejecución de interfaces gráficos de usuario es **X.Org** .

X.Org es una implementación de código abierto del sistema de ventanas X Window y desarrollado por la X.Org Foundation ( http://www.x.org/wiki/ ). El origen de este sistema gráfico está en las XFree86 4.4 y, en concreto, lo que determinó su creación fue el cambio de licencia de las XFree86, que ahora no es completamente compatible con GPL. La primera versión del servidor X.Org fue la X11R6.7 y, en este momento, la versión actual es la X11R7.4.

La mayor parte de las distribuciones GNU/Linux han optado por este servidor gráfico X.Org. Distribuciones como Ubuntu, Debian, OpenSUSE, Mandriva, etc han cambiado su servidor gráfico XFree86 a X.Org. Además, X.Org se integra dentro de los estándares descritos en la Freedesktop ( http://www.freedesktop.org ) cuyo objetivo es establecer las aplicaciones y especificaciones de los entornos de escritorio libres.

# 3 Estructura del sistema gráfico X Window

El sistema gráfico X Window y, por lo tanto heredado por X.Org, consta de varias capas de software que constituyen la interfaz gráfica de usuario:

-

**Sistema de ventanas X**

Es el responsable de las operaciones gráficas básicas como dibujar los iconos, los fondos, y las ventanas en las que se ejecutan las aplicaciones.

-

**Gestor de ventanas**

Añade elementos alrededor de las ventanas, de forma que el usuario pueda cambiar el tamaño, cerrar, ocultar, mover...

-

**Gestor de pantalla**

Gestiona la sesión de usuario. También llamado Display Manager y es lanzado como un servicio del proceso init[1].

-

**Entorno de escritorio**

El entorno de escritorio puede ser, desde un simple gestor de ventanas, hasta un conjunto muy completo de aplicaciones de escritorio. En realidad es lo que 've' el usuario después de abrir la sesión.

## 3.1 Sistema de ventanas X

El sistema de ventanas X es el encargado de mostrar la información gráfica. Es el propio servidor X (puertos TCP y UDP 6000). Utiliza el protocolo Xprotocol para su comunicación con el cliente X, y proporciona un conjunto de órdenes básicas para la generación de ventanas.

El servidor X ejecuta el driver de vídeo real y proporciona una interfaz a través de la cual otros programas pueden ejecutar órdenes de dibujo sin conocer qué hardware está usándose. Los programas que se conectan al servidor X se llaman clientes X, y pueden ser aplicaciones de todo tipo, como procesadores de texto, juegos, etc.

## 3.2 Gestor de ventanas

El gestor de ventanas es un programa intermedio que controla el aspecto, el movimiento y cambio de tamaño de las ventanas, es decir, dibuja bordes, marcos, maximiza, minimiza, usa colores, etc. Controla la barra de títulos, cuál es la ventana activa (comportamiento del foco), recogiendo los comandos del ratón y teclado para ella.

Un gestor de ventanas es, desde el punto de vista del servidor X, una aplicación cliente normal y, por lo tanto, ha de comunicarse con el servidor X como cualquier otra aplicación.

Puede formar parte del entorno de escritorio o ser independiente. Cuando se habla de un gestor de ventanas independiente se está refiriendo a un gestor que no requiere otra capa de software para poder ser funcional.

En http://xwinman.org/ se pueden encontrar diferentes gestores de ventanas así como entornos de escritorios. Por ejemplo, están:

-

**metacity:** es el gestor por defecto de GNOME 2.26 (incluido en Ubuntu 9.04 Jaunty). Anteriormente GNOME utilizaba otros gestores de ventanas, como el **Enlightenment** y **sawfish**. Ambos GPL.

-

**fluxbox:** es el gestor de Knoppix LTD. MIT

-

**kwin:** utilizado por KDE.

-

**twm:** es un gestor minimalista. El menú aparece con el botón izquierdo. MIT.

-

**icewm:** buen gestor de ventanas que sobrecarga poco el sistema y es muy completo.

Los gestores twm e icewm funcionan por si mismos, sin necesidad de estar integrados en un escritorio.

## 3.3 Gestor de pantalla

El gestor de pantalla X (XDM) es una parte opcional del sistema X Window que se usa para el manejo de sesiones, ya que gestiona el registro de usuario (inicio y fin de sesión). Forma parte

del cliente X.

Dentro de una red, utilizando XDM se puede elegir a qué pantalla se quiere conectar y pedir la información de autenticación, como nombre de usuario y contraseña. Hecho esto, arranca el gestor de ventanas y permanece a la espera de que termine, mostrando de nuevo la pantalla de entrada al sistema y determinadas opciones.

Hay gestores de pantalla mas potentes que permiten, además de la conexión y cierre, el cambio de idioma para esa sesión e incluso del escritorio propiamente, si el sistema tiene otro disponible.

Hay tres gestores de pantalla importantes:

1.

**gdm** (Gnome Display Manager)**:** utilizado por el escritorio GNOME. Incluye:

-

Conexión: usuario y contraseña

-

Autoconexión: sin validación

-

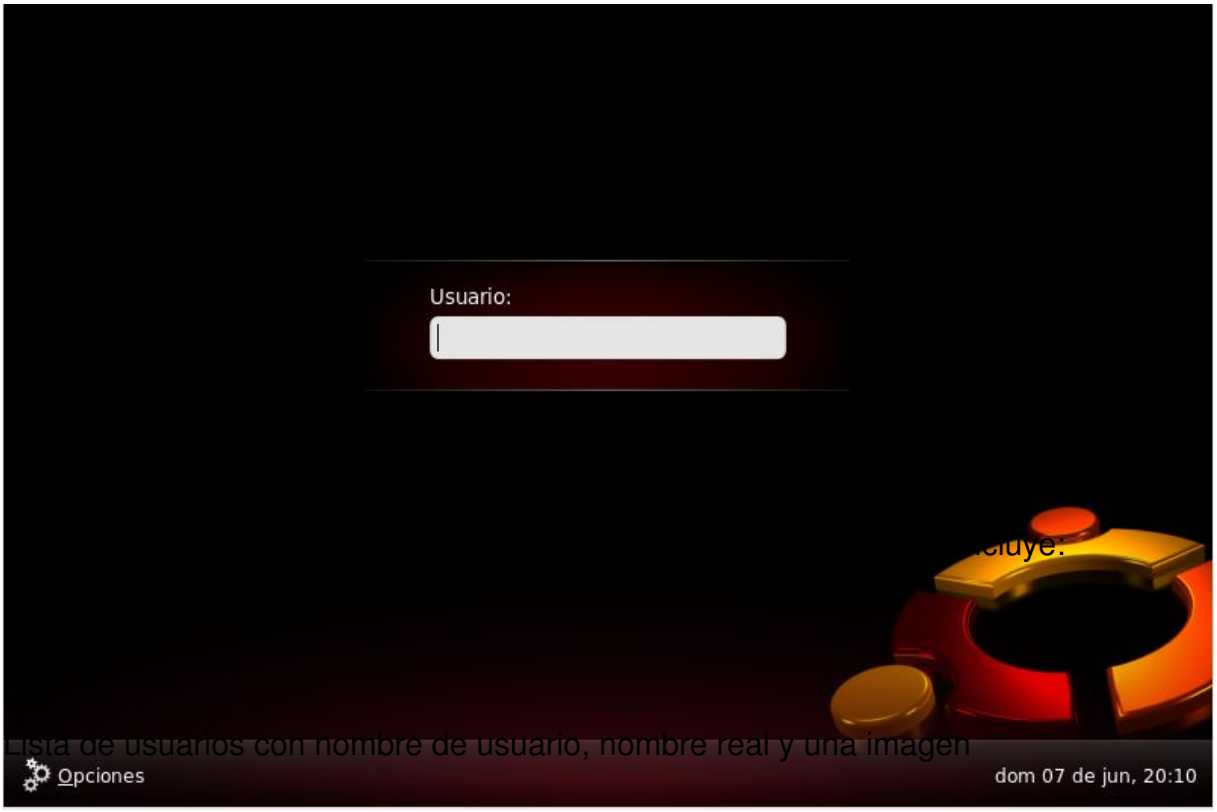
Selección del escritorio: GNOME, KDE

-

Idioma (valor que se asigna a la variable LANG)

-

Reiniciar/Apagar




Lista de usuarios con nombre de usuario, nombre real y una imagen

-

Conexión: usuario y contraseña

-

Autoconexión: sin validación

-

Selección del escritorio: GNOME, KDE,...

-

Reiniciar/Apagar

-

Herramienta para administrar usuarios

1.

**xdm** (X Display Manager)**:** es el mas básico y genérico de X Window

## 3.4 Entorno de escritorio

No hay que confundir el entorno de escritorio con el gestor de ventanas. El gestor de ventanas sólo puede controlar la apariencia y la posición de las ventanas de los clientes de X.

Los entornos de escritorio tienen mucha mas funcionalidad, ya que permiten a los clientes X y otros procesos en ejecución comunicarse entre sí, por ejemplo copiando y pegando texto entre diferentes aplicaciones. En general, permiten la coordinación entre aplicaciones y ofrecen un entorno uniforme de trabajo.

Un entorno de escritorio consta, al menos, de las siguientes aplicaciones clientes X:

-

Gestor de ventanas

-

Administrador de Vistas

-

Barra de tareas

-

Gestor de archivos

-

Conjunto de aplicaciones

Los entornos de escritorio mas conocidos son:

1.

**GNOME (GNU Network Object Model Environment)**

-

Conjunto de herramientas gráficas y basado en las bibliotecas GTK+.

-

El promotor del proyecto fue Miguel de Icaza en 1997.

-

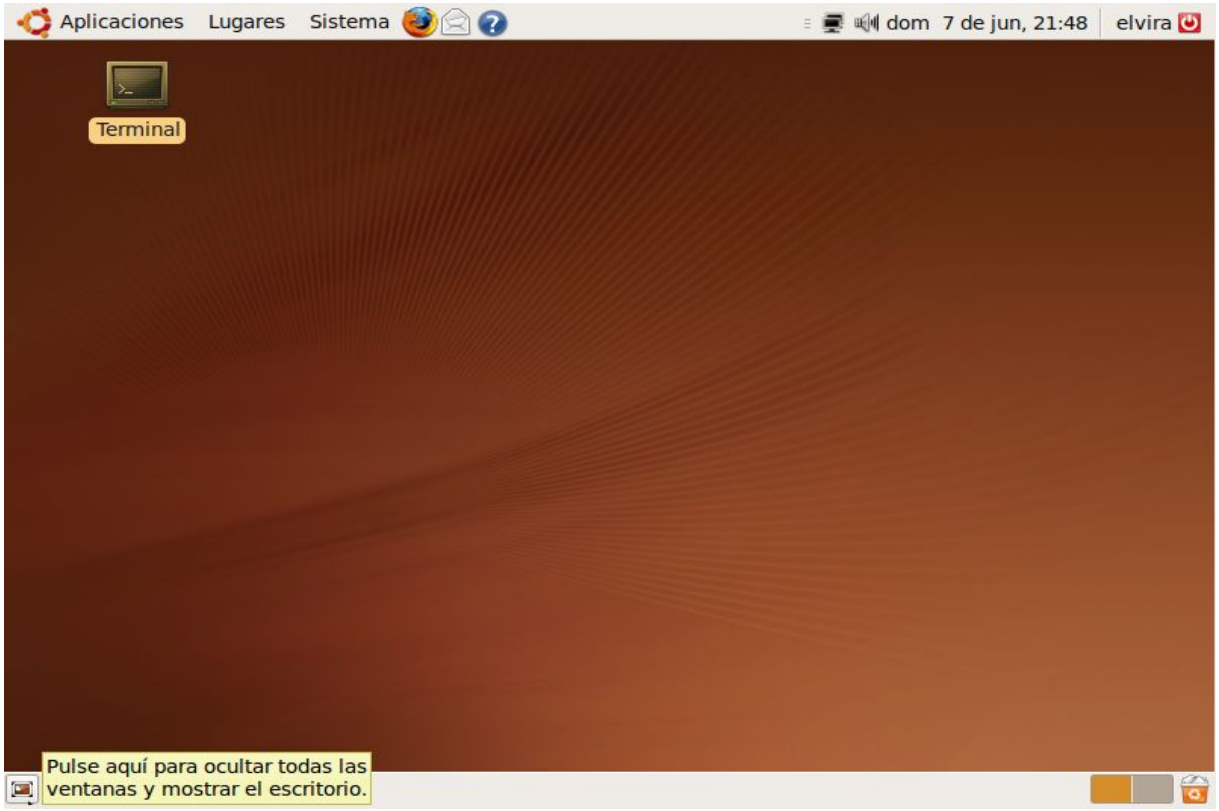
La primera versión completa aparece en 1999.

-

La última versión es la 2.26 (Junio '09).

-

Es GPL/LGPL.




- **KDE (K Desktop Environment)**

-

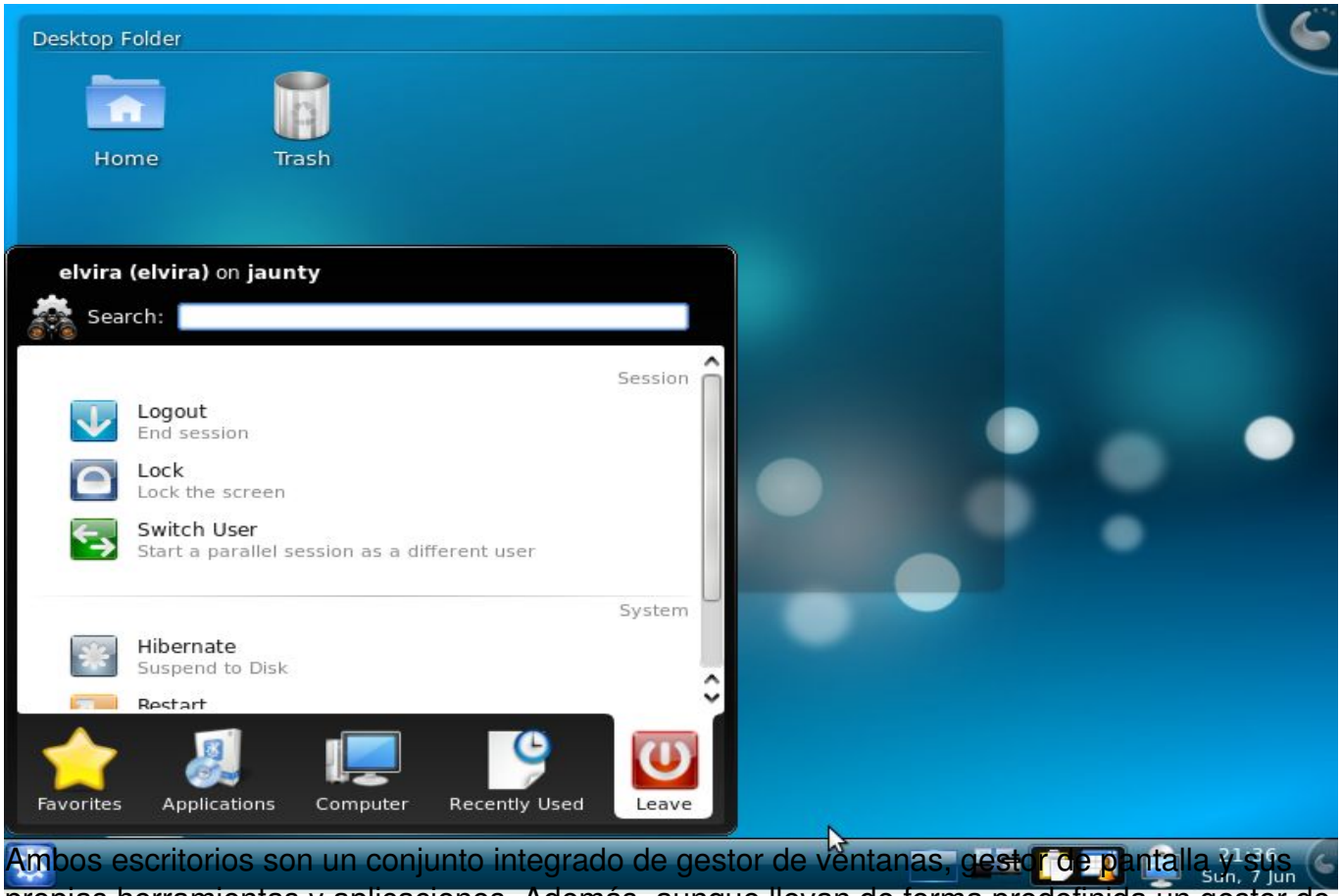
Conjunto de herramientas gráficas y basado en las librerías Qt. En la actualidad las librerías QT son de uso libre para aplicaciones de software libre.

-

El proyecto KDE se inició en 1996.

-

La última versión es la 4.2 (Junio '09). Es GPL/LGPL.




Ambos escritorios son un conjunto integrado de gestor de ventanas, gestor de pantalla y sus propias herramientas y aplicaciones. Además, aunque llevan de forma predefinida un gestor de ventanas se puede cambiar, pero hay que tener en cuenta que no todos funcionan igual de bien.

Si se tiene ambos escritorios instalados, en el menú de **gdm**: *Opciones > Iniciar sesión*, el usuario puede seleccionar el entorno de escritorio con el que quiere trabajar en dicha sesión.

## 4 Características de X.Org. Instalación.

La versión actual de X.Org es la 7.4 y sus características mas importantes son las siguientes:

-

Arquitectura Cliente/Servidor:

-

Cliente X:

-

Realiza una petición de uso de un recurso vía red o vía la interface loopback al Servidor X.

- Servidor X
-

Encargado de comunicarse con el hardware dibujando en pantalla.

-

Envía al cliente los eventos que produzca el usuario.

-

Gestiona los siguientes recursos HW entre los diferentes Clientes X:

-

Tarjeta gráfica.

-

Monitor

-

Teclado y Ratón

- La comunicación se realiza usando el protocolo X11.
- Xorg no es parte del núcleo de Linux.
- Los módulos de vídeo son directamente gestionados por el el servidor gráfico X.
- Soporta *hotplugging* para dispositivos de entrada (teclado, ratón y pantallas táctiles) y de salida (monitor, proyector). Esto permite la comunicación con HAL [2] para determinar cuándo se ha conectado al sistema un nuevo dispositivo.

- Nuevas ventanas translúcidas, con sombra y con animación al minimizarse.
- Soporte para escritorios 3D (Looking Glass de Sun).
- Mejores controladores libres para las tarjetas gráficas de AMD/ATI.

El protocolo X11 fue diseñado para poder ser usado en red. Es decir el usuario puede estar ejecutando un cliente X (aplicación) en una máquina y su interfaz se está mostrando en otra máquina.

Para instalar X.Org ejecutar la orden:

```
$sudo apt-get install xserver-xorg ;o
```

```
$sudo aptitude install xserver-xorg
```

xserver-xorg es un metapaquete que proporciona los componentes para una máquina independiente ejecutando el sistema X Window: librerías X, el servidor X, un conjunto de fuentes, y un conjunto de clientes X (aplicaciones) y utilidades.

# 5 Configuración del servidor gráfico X.Org

El sistema X Window se encuentra localizado en varios directorios y archivos dentro del

sistema de archivos:

-

/**usr**/**X11R6**/ Contiene un enlace al binario del cliente X

-

/**usr**/**lib**/**xorg**/ Contiene los módulos correspondientes a las tarjetas gráficas.

-

/**etc**/**X11**/ Contiene todos los archivos de configuración de los componentes del sistema X Window. Incluye /etc/X11/X que es un enlace simbólico al ejecutable /usr/bin/Xorg.

-

/**var**/**log**/**Xorg.$DISPLAY.log** Archivo que almacena los mensajes de log del servidor X para cada display

-

/**usr**/**bin**/**xclock,xterm,...** Aplicaciones X que residen en este directorio.

El archivo de configuración de Xorg es /**etc**/**X11**/**xorg.conf** y está dividido en secciones cada una de ellas dedicada a la configuración de un aspecto del servicio gráfico.

| **Sección** |
| --- |

| Descripción |
| --- |

**Device**

Define una tarjeta gráfica determinada que se identifica mediante su nombre descriptivo.

**Monitor**

Describe el monitor. Consta de nombre, que luego se referencia en las secciones Screen y las frecuenc

**Screen**

Asocia un Monitor y un Device. En la subsección Display se especifica el tamaño de la pantalla virtual (

**InputDevice**

Configura los dispositivos de entrada, como teclados y dispositivos de entrada especiales (touchpads, j

**ServerFlags**

Establece las opciones generales.

**Modes**

| Almacena los parámetros de modeline para resoluciones de pantalla determinadas. Para mas informaci |
|---|

| **Files** |
|---|

| Indica los directorios donde se encuentran las fuentes y la tabla de colores RGB. |
|---|

| **ServerLayout** |
|---|

| Relaciona los dispositivos de entrada InputDevice con los dispositivos de visualización Screen. |
|---|

*Tabla 1: Secciones de /etc/X11/xorg.conf*

Si editamos el archivo directamente desde una terminal:

```
$sudo gedit /etc/X11/xorg.conf
```

podremos ver las secciones básicas y su contenido por defecto. La configuración específica por usuario se genera al ejecutar la orden del punto 5.1 y se almacena en el directorio home del usuario.

**Section "Device"**

**Identifier "Configured Video Device"**

**EndSection**

**Section "Monitor"**

**Identifier "Configured Monitor"**

**EndSection**

**Section "Screen"**

**Identifier "Default Screen"**

**Monitor "Configured Monitor"**

**Device "Configured Video Device"**

**EndSection**

Para conocer con mas detalle el archivo de configuración, solicitar la página de manual ejecutando desde una terminal:

```
$ man 5 xorg.conf
```

## 5.1 Configuración

En Ubuntu los pasos a realizar para configurar el servidor X.Org son:

1.

Antes de actuar sobre el archivo de configuración se debe hacer una copia del mismo:

```
$sudo cp /etc/X11/xorg.conf /etc/X11/xorg.conf_copia
```

2.

Parar el gestor de ventanas:

```
$sudo /etc/init.d/gdm stop
```

3.

Ejecutar alguna de las órdenes siguientes de configuración del servidor X desde consola:

```
$sudo dpkg-reconfigure xserver-xorg
```

```
$sudo Xorg -configure :0
```

Esta última orden prueba el hardware generando el archivo xorg.conf.new. Trabaja sobre el DISPLAY[3] 0 por defecto.

Al parar el gestor de ventanas perdemos el entorno gráfico y pasamos a modo consola. Si se quiere continuar sin cerrar el escritorio, ejecutar la orden sobre el DISPLAY 1. Muestra la salida:

```
$ sudo Xorg -configure :1

X.Org X Server 1.6.0

Release Date: 2009-2-25

X Protocol Version 11, Revision 0

Build Operating System: Linux 2.6.24-23-server i686 Ubuntu

Current Operating System: Linux servidor 2.6.28-11-generic #42-Ubuntu SMP Fri Apr 17 01:57:59 UTC

Build Date: 09 April 2009 02:10:02AM

xorg-server 2:1.6.0-0ubuntu14 (buildd@rothera.buildd)

Before reporting problems, check http://wiki.x.org
```

to make sure that you have the latest version.

Markers: (--) probed, (**) from config file, (==) default setting,

(++) from command line, (!!) notice, (II) informational,

(WW) warning, (EE) error, (NI) not implemented, (??) unknown.

(==) Log file: "/var/log/Xorg.1.log", Time: Thu Jun 4 19:48:54 2009

List of video drivers:

rendition

trident

.....

ati

i810

vesa

(++) Using config file: "/home/elvira/xorg.conf.new"

Xorg detected your mouse at device /dev/input/mice. Please check your config if the mouse is still not op

Your xorg.conf file is /home/elvira/xorg.conf.new

To test the server, run 'X -config /home/elvira/xorg.conf.new'

ddxSigGiveUp: Closing log

La nueva configuración creada se puede probar ejecutando:

$sudo X -config xorg.conf.new :1

Si todo va bien el nuevo archivo de configuración generado se salva con el nombre del auténtico:

$sudo cp xorg.conf.new /etc/X11/xorg.conf

Volver a arrancar el gestor de ventanas:

```
$sudo /etc/init.d/gdm start
```

### *5.2 Ajuste manual del archivo xorg*

Veamos las secciones y su configuración particular:

**Sección Screen**

Una posible sección Screen es la que se incluye.

```
Section "Screen"

Identifier "Default Screen"

Monitor "Monitor0"

Device "Card0"

DefaultDepth 24

SubSection "Display"

Depth 16
```

```
Modes "1024x768" "800x600" "640x480"

EndSubSection

SubSection "Display"

Depth 24

Modes "800x600" "640x480"

EndSubSection

EndSection
```

El identificador se la Sección es "Default Screen". Se indica para luego poder referenciar esta sección desde otras. Las entradas Device y Monitor indican la tarjeta gráfica y el monitor y son enlaces a las secciones Device y Monitor correspondientes al identificador dado.

Se ha establecido una profundidad de color por defecto (DefaultDepth) de 24. Existe una subsección Display para cada profundidad de color. Los valores posibles de Depth son: 8, 15, 16 y 24.

La línea Modes define una lista de resoluciones para cada profundidad establecida. El servidor X lee la lista de izquierda a derecha. Para cada resolución, el servidor X busca un Modeline en la sección Modes. El Modeline depende de la capacidad del monitor y de la tarjeta gráfica.

Para cambiar la resolución de pantalla en caliente (ejecutándose las X) y desde el teclado:

1.

Aumentar la resolución de pantalla: **CTRL+ALT +**

2.

Disminuir la resolución de pantalla: **CTRL+ALT -**

**Sección Device**

Cada sección Device describe una tarjeta gráfica determinada. Se pueden incluir en xorg.conf tantas entradas device como queramos siempre con identificadores diferentes. Si hay más de una tarjeta gráfica instalada, las secciones se numeran de forma consecutiva: Card0, Card1,...

La siguiente sección Device muestra la configuración para una tarjeta gráfica ATI Radeon 9500.

El parámetro Driver especifica el parámetro que se utilizará para la tarjeta gráfica. En el ejemplo, como la tarjeta es una *ATI Radeon* se utiliza el módulo del controlador 'radeon'. El servidor X buscará el módulo en el camino dado en ModulePath (sección Files, subdirectorio drivers). Por defecto es el directorio **/usr/lib/xorg/modules/drivers/**
. Al nombre indicado se le añade _drv.o, de manera que, en el caso del controlador radeon, se cargará el archivo radeon_drv.so.

```
Section "Device"
```

### Available Driver options are:-

### Values: <i>: integer, <f>: float, <bool>: "True"/"False",

### <string>: "String", <freq>: "<f> Hz/kHz/MHz"

### [arg]: arg optional

Identifier "Card0"

Driver "radeon"

VendorName "ATI Technologies Inc"

BoardName "Radeon R300 NE [Radeon 9500 Pro]"

BusID "PCI:1:0:0"

EndSection

**Sección Monitor**

Cada sección Monitor describe un monitor. /etc/X11/xorg.conf puede contener tantas secciones

Monitor como se quiera. La sección ServerLayout especifica qué sección Monitor es la relevante.

Las definiciones de Monitor sólo deben ser ajustadas por usuarios expertos. En esta sección se establecen la sincronización horizontal y vertical para ese monitor, ambos valores son puntos críticos para el buen funcionamiento del monitor.

```
Section "Monitor"

Identifier "Monitor0"

VendorName "Monitor Vendor"

ModelName "LCD Panel 1024x768"

HorizSync 31.5 - 48.5

VertRefresh 40.0 - 70.0

Option "dpms"

EndSection
```

**Sección InputDevice**

Una configuración para ratón puede ser la que se muestra. El ratón es PS/2 y está activada la emulación de 3 botones.

```
Section "InputDevice"

Identifier "Configured Mouse"

Driver "mouse"

Option "CorePointer"

Option "Device" "/dev/input/mice"

Option "Protocol" "ImPS/2"

Option "ZAxisMapping" "4 5"

Option "Emulate3Buttons" "true"

EndSection
```

**Sección ServerLayout**

```
Section "ServerLayout"

Identifier "X.org Configured"

Screen 0 "Default Screen" 0 0

InputDevice "Configured Mouse"

InputDevice "Keyboard0"

EndSection
```

En general desde el sistema X Window se puede volver a reiniciar el servidor ejecutando la combinación de teclas CTRL+BACK+SUPR. Pero por defecto Ubuntu lleva esta combinación de teclas desactivada. Si queremos disponer de ella hemos de incluir en el archivo de configuración /etc/X11/xorg.conf en la sección *ServerFlags* la opción siguiente:

```
Section "ServerFlags"

Option "DontZap" "false"

EndSection
```

Se puede comprobar la detección de la tj gráfica con el comando **lspci**. En la salida generada buscar "VGA".

```
$ lspci|grep VGA
```

```
01:00.0 VGA compatible controller: ATI Technologies Inc Radeon R300 NE [Radeon 9500 Pro]
```

También desde Ubuntu se puede activar/desactivar la opción de aceleración de la tarjeta gráfica. Para comprobarlo ejecutar:

```
$ glxinfo | grep -i render
```

```
direct rendering: Yes
```

```
OpenGL renderer string: Mesa DRI R300 20060815 AGP 4x x86/MMX/SSE2 TCL
```

Si la tarjeta da problemas y necesitamos desactivar la aceleración 3D y los efectos de escritorio, se puede hacer de la forma siguiente:

```
Section "Device"
```

```
Option "DRI" "off"
```

```
EndSection
```

desde el archivo /**etc**/**X11**/**xorg.conf**.

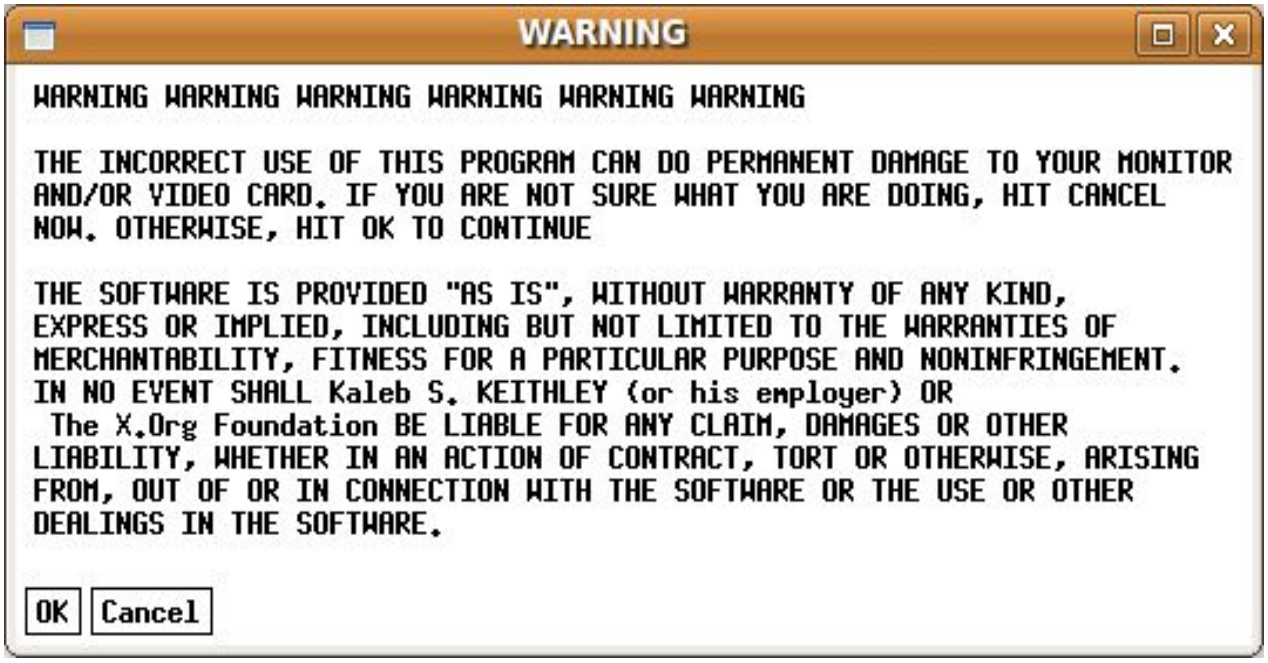
Para el caso de los drivers propietarios de nvidia o ati (fglrx), si se quieren instalar los módulos hay que hacerlo desde un paquete específico cuyo nombre sigue la nomenclatura: **xserver-xorg-video-nv** .
El módulo se instala en el directorio
/**usr**/**lib**/**xorg**/**modules**/**drivers**/
y se llama
**nv_drv.so.**

Otra opción de configuración es cuando se quieren activar extensiones que permiten trabajar con sombras y transparencias. En este caso y en el archivo /**etc**/**X11**/**xorg.conf** añadimos la sección:

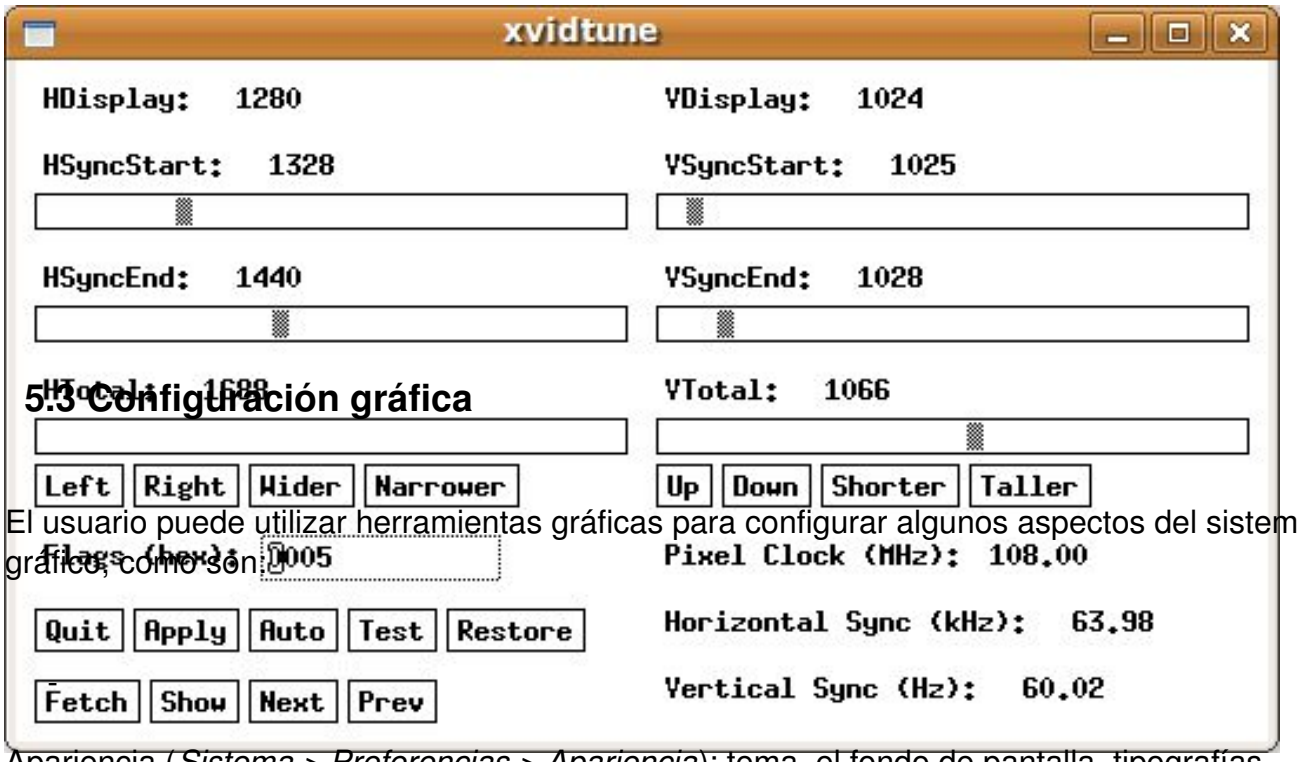
```
Section "Extensions"

Option "Composite" "Enable"

Option "RENDER" "Enable"

EndSection
```

Por último, para hacer ajustes de configuración del monitor tenemos la orden **xvidtune**. Esta herramienta proporciona una interfaz con controles para configurar las frecuencias de actualización del monitor y la resolución de la pantalla.

Al ejecutar la orden muestra una ventana de aviso indicando que puede ser peligroso hacer un uso arbitrario de esta herramienta.




Y la interfaz de trabajo es la siguiente:




## 5.3 Configuración gráfica

El usuario puede utilizar herramientas gráficas para configurar algunos aspectos del sistema gráfico, como son:

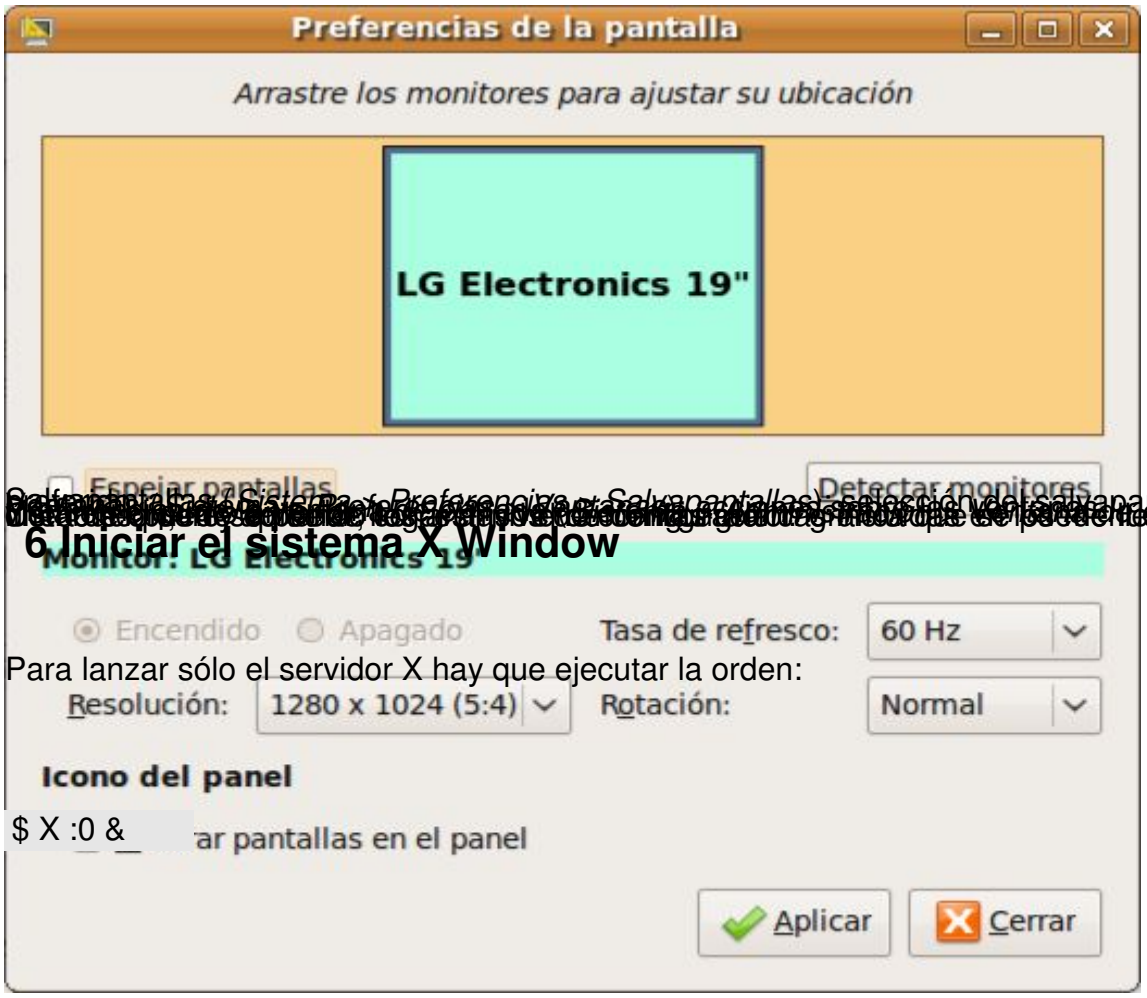
Apariencia (*Sistema > Preferencias > Apariencia*): tema, el fondo de pantalla, tipografías, efecto visuales...

-

Escritorio remoto (*Sistema > Preferencias > Escritorio remoto*)

-

Pantalla (*Sistema > Preferencias > Pantalla*): detección de monitores, resolución utilizada, frecuencias de refresco,...




## 6 Iniciar el sistema X Window

Para lanzar sólo el servidor X hay que ejecutar la orden:

```
$ X :0 &
```

El servidor por defecto se ejecuta sobre la primera pantalla gráfica (DISPLAY :0) a la que se puede acceder mediante la combinación de teclas CTRL+ALT+F7. Los servicios están disponibles, pero al no haber ningún cliente y ningún gestor de ventanas no podemos hacer nada con ellos. En realidad no muestra nada, pero el proceso se está ejecutando y se puede comprobar con la orden:

```
$ps aux |grep X
```

```
root 12034 1.3 2.7 31816 27476 tty7 Ss+ 09:02 2:12 /usr/bin/X :0 -br -audit 0 -auth /var/lib/gdm/:0.Xauth
```

Existen varios métodos para iniciar el sistema X Window:

1.

Ejecutar las X cuando lo necesitamos

2.

Ejecutar en las X en secuencia de arranque

3.

Ejecutar las X en una máquina remota

Para cualquiera de los tres planteamientos suponemos que estamos en la consola. Es decir, no existe ningún servidor X lanzado.

La opción 3 la explicamos después del apartado de Seguridad.

Además de la orden X tenemos la orden **xinit** que se utiliza para iniciar el servidor X y un primer programa cliente que puede ser una terminal xterm para, desde ella, ir lanzando aplicaciones. Es configurable para cada usuario mediante el archivo ~/.xinitrc.

## 6.1 Ejecutar las X cuando lo necesitamos

Se trata de entrar en el sistema sin servidor gráfico, en modo consola y cuando interesa a un usuario *sudo* lanza el entorno gráfico.

Comprobaremos que /**usr**/**bin**/**X11**/ está incluido en el contenido de la variable PATH.

```
$echo $PATH
```

Iniciamos las X ejecutando la orden:

```
$sudo startx
```

Los archivos implicados en este arranque del modo gráfico son:

**/etc/X11/xinit/xinitrc** (run command de xinit) archivo desde el cual se lanza /etc/X11/Xsession.

**/etc/X11/Xsession** arranca la sesión X para el usuario conectado teniendo en cuenta una serie de opciones de configuración ( **/etc/X11/Xsession.options)**tanto del sistema como del propio usuario. En concreto este script inicia el entorno de escritorio y el gestor de pantalla seleccionado en el archivo **/etc/X11/default-display-manager**
.

Si se quiere utilizar un gestor de ventanas personalizado para el usuario se puede visualizar y modificar con la orden siguiente:

```
$ sudo update-alternatives --config x-window-manager

[sudo] password for elvira:

Sólo hay un programa que provee x-window-manager

(/usr/bin/metacity). No se configurará nada.
```

En Ubuntu sólo se incluye como gestor de ventanas Metacity y por ese motivo no se puede cambiar. En otras distribuciones puede haber varios gestores, o incluso en Ubuntu se pueden instalar otros, en cuyo caso, con esta orden se podría elegir el gestor de ventanas a utilizar.

El usuario puede lanzar una sesión GNOME 'a mano' ejecutando las órdenes siguientes estando en la consola:

```
$ X :0 &
```

```
$ xterm -display :0 &
```

```
$ gnome-session
```

Esta secuencia ejecuta el servidor X en background (&) sobre el display 0, lanza un terminal x en el display :0 y ejecuta desde dentro de dicho terminal una sesión GNOME. Todo ello sin pasar por el gestor de pantalla gdm.

## 6.2 Ejecutar las X en secuencia de arranque

Al instalar Ubuntu el sistema arranca siempre con el entorno gráfico. Ejecuta el servidor X, el gestor de ventanas Metacity y el X Display Manager **xdm**/**kdm**/**gdm** según el escritorio instalado e independientemente del nivel de ejecución con el que se entre en el sistema. Por último lanza el entorno de escritorio GNOME o KDE.

En los directorios /**etc**/**rcx.d**/ (x=nº nivel) hay un script **S30gdm** (o su equivalente para KDE) que es el responsable de que se lance el gestor de pantalla para GNOME, excepto para el nivel 1 en el que no se lanza el entorno gráfico. El nivel 1 se corresponde con el modo de trabajo monousuario y multitarea, para tareas administrativas que requieran que no haya ningún otro usuario conectado.

Este es el 'modo normal' de trabajo que, por defecto, es el nivel 2. Se puede comprobar ejecutando la orden:

```
$ runlevel
```

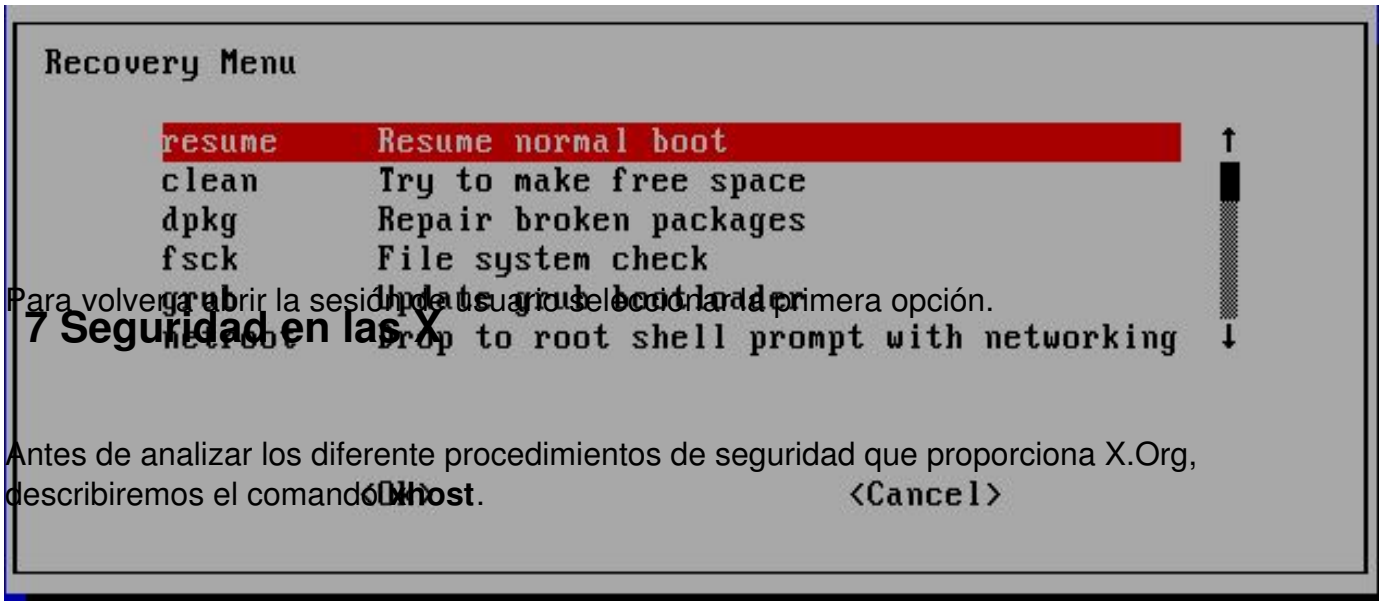
```
N 2
```

Que indica que estamos en el nivel 2 de ejecución, es decir multiusuario, multitarea, con servicios de red activos y entorno gráfico lanzado.

Pero sí se puede modificar el nivel de ejecución en caliente ejecutando la orden:

```
$ sudo init 1
```

En este caso el sistema matará todos los procesos implicados en la gestión de usuarios y sus tareas, la red y el entorno gráfico y pasaremos a la consola del sistema. ¡Ojo! si se prueba y se tiene algún archivo o aplicación abiertos lo cerrará todo. Muestra un *Recovery Menu* donde el usuario puede seleccionar la forma de seguir conectado al sistema.




Para volver a abrir la sesión de usuario seleccionaría la primera opción.

## 7 Seguridad en las X

Antes de analizar los diferente procedimientos de seguridad que proporciona X.Org, describiremos el comando **xhost**.

xhost permite indicar al servidor X que programas de otras máquinas remotas pueden acceder a su pantalla, teclado y ratón. Este método no distingue usuario sino máquina, con lo cual la seguridad se puede ver comprometida.

Los pasos para la utilización de xhost son:

1.

Indicar al servidor X que permita el acceso desde una máquina concreta:

>maquinalocal$ xhost +maquinaremota.dominio.com (o IP)

2.

Indicar en la máquina remota dónde tienen que mostrar su salida las aplicaciones ejecutadas. Para ello hay que configurar la variable DISPLAY:

> maquinaremota$ export DISPLAY=maquinalocal.dominio.com:1.0 (o IP:1.0)

Los dos procedimientos de seguridad al utilizar las X se basan en:

1.

**Control de acceso al host.**

Este método permite decidir qué equipos pueden tener acceso a nuestra pantalla. Utiliza la orden xhost:

```
# xhost +192.168.1.100

192.168.1.100 being added to access control list

# xhost -192.168.1.100

192.168.1.100 being removed from access control list

$ xhost +

access control disabled, clients can connect from any host
```

```
$ xhost -

access control enabled, o-nly authorized clients can connect
```

Las órdenes son lo suficientemente autoexplicativas.

Para ejecutar una aplicación sobre un display de esa IP hay que exportar el display y lanzar la aplicación. Esta acción la veremos en el punto siguiente.

2.

**Control de acceso Mit-Magic-Cookie.**

Este método genera una clave cada vez que se arranca el servidor X, y sólo se permite el acceso a los clientes que comunican esa clave.

Para habilitar este método de acceso hay que editar el archivo **/etc/X11/xinit/xserverrc** y añadir la opción
**-auth $HOME/.Xauthority**
en la línea que arranca el servidor X:

```
$ sudo vi /etc/X11/xinit/xserverrc

#!/bin/sh

exec /usr/bin/X11/X -nolisten tcp -auth $HOME/.Xauthority
```

Cada vez que un usuario arranca el servidor X se genera una cookie de 128 bits y se almacena en el archivo ~/*.Xauthority*.

¿Cómo se utiliza? Para acceder como root al display del *usuario1* hay que hacer que los clientes X comuniquen la misma Magic-Cookie que ha creado el arranque del servidor X. Eso se hace con el comando **xauth**.

En el ejemplo se ejecuta la aplicación *xeyes* como root en el escritorio del *usuario1* (display 1):

```
$ sudo xauth merge usuario1/.Xauthority
```

```
$ export DISPLAY=:1.0
```

```
$ xeyes &
```

dónde *usuario1* es el id. del usuario que ha ejecutado las X.

Este método no necesita el puerto tcp 6000 abierto.

```
$ xauth list
```

```
elvira-jaunty/unix:0 MIT-MAGIC-COOKIE-1 e839ce41769fc6ead1b10437dbdb4951

localhost.localdomain/unix:0 MIT-MAGIC-COOKIE-1 e839ce41769fc6ead1b10437dbdb4951
```

**IMPORTANTE:**

Como norma de seguridad no conectarse nunca al servidor X como root. Si alguna aplicación gráfica necesita ejecutarse como root se puede hacer de la forma siguiente:

1. Iniciar las X como usuario normal
2. Abrir una terminal y ejecutar:

```
$ sudo XAUTHORITY=$HOME/.Xauthority

$ export XAUTHORITY

$ su root
```

1. Ejecutar la aplicación

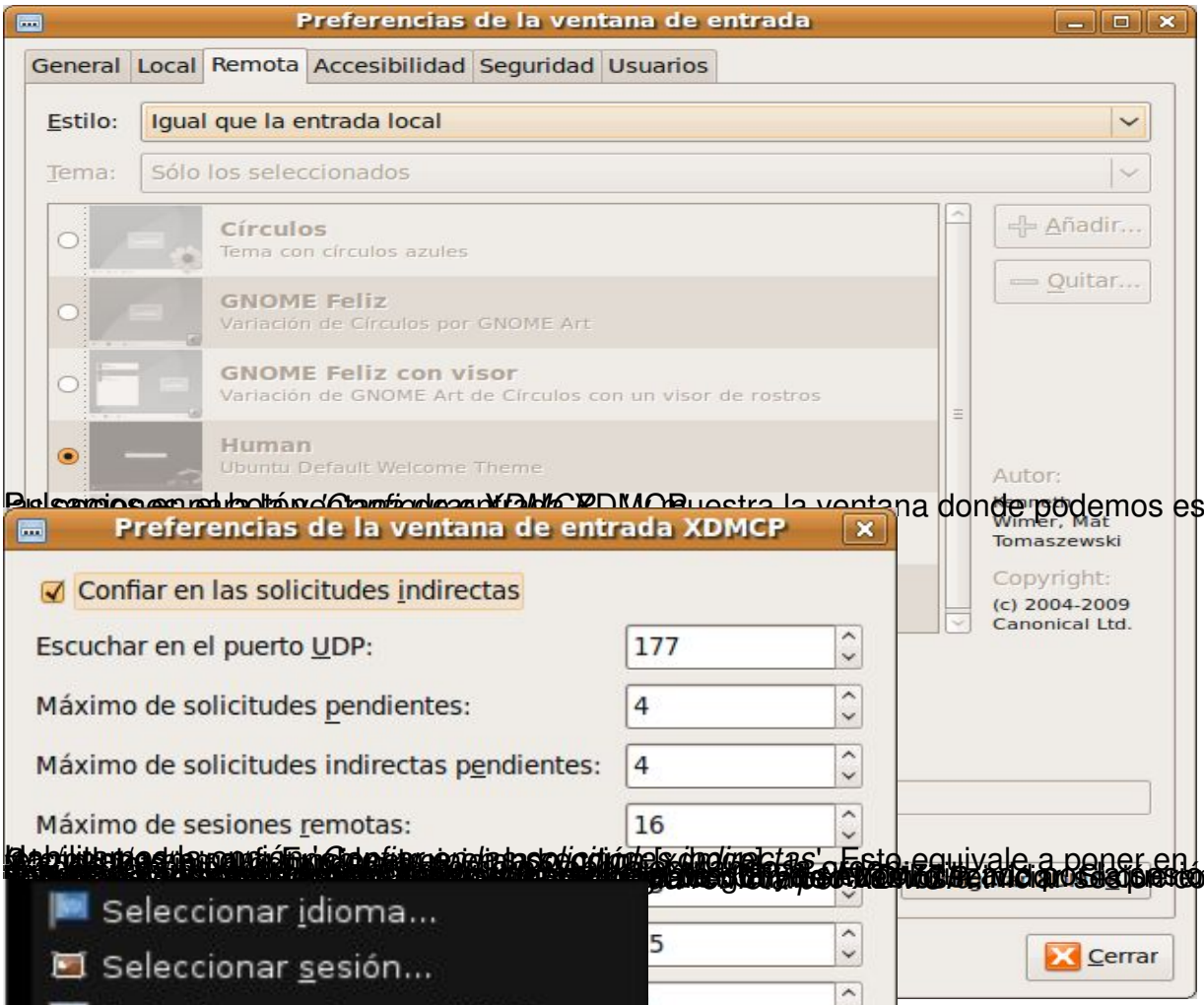
Cuando se utilice este método hay que asegurarse de que el archivo ~/.Xauthority tenga permiso de lectura para el grupo al que pertenece el usuario.

# 8 Ejecutar las X en una máquina remota

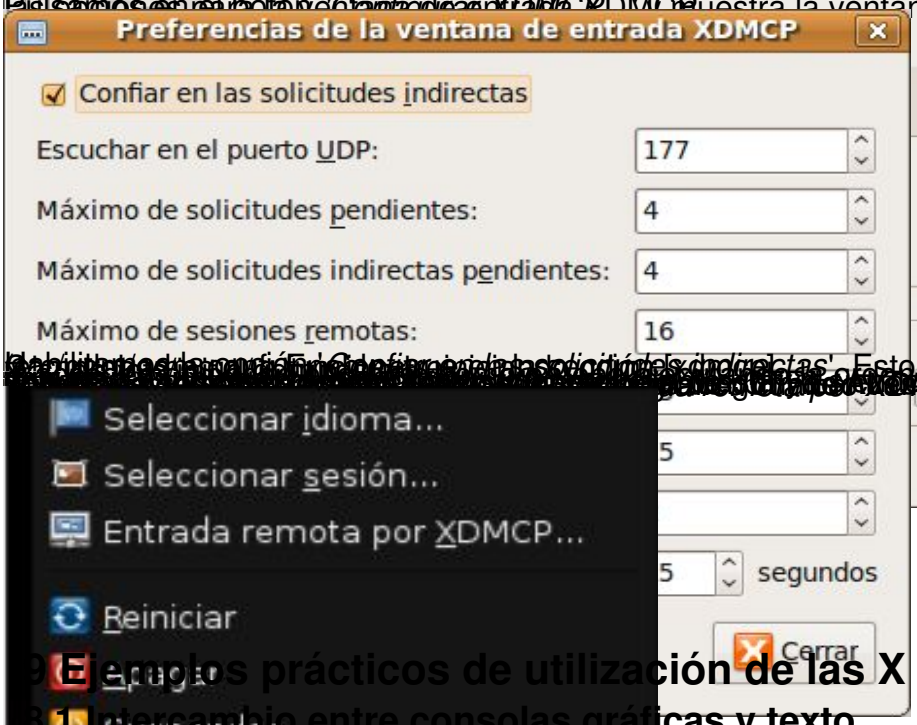
Podemos ejecutar sesiones remotas habilitando en XDMCP las solicitudes indirectas. XDMCP es el protocolo que permite el login remoto. Para ello ejecutar en una terminal la orden **gdmsetup** o ir a:

*Sistema > Preferencias > Ventana de entrada > Pestaña 'Remota'*

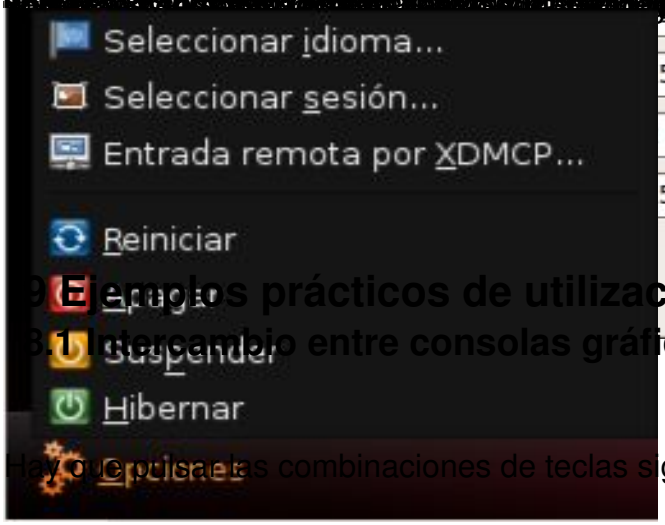
En Estilo seleccionamos *Igual que la entrada local* y muestra la pantalla siguiente:




Pulsando en el botón Configurar XDMCP nos muestra la ventana donde podemos establecer




'Confiar en las solicitudes indirectas'. Esto equivale a poner en




Y de lo

## 9. Ejemplos prácticos de utilización de las X

### 9.1. Intercambio entre consolas gráficas y texto.

Hay que pulsar las combinaciones de teclas siguientes:

-

Estando en consola gráfica pasar a consola de texto:

CTRL+ALT+F1 ..... CTRL+ALT+F6

-

Estando en consola texto pasar a consola gráfica:

CTRL+ALT+F7 ..... CTRL+ALT+F12

Donde CTRL+ALT+F7 se corresponde con el DISPLAY 0, CTRL+ALT+F8 el DISPLAY 1,...

## *8.2 Varias sesiones X-Window simultáneas.*

Se pueden abrir hasta 6 consolas gráficas concurrentes. Los cambios entre consolas gráficas se hacen como en el punto 8.1.

Hay que indicar el display al cual irá cada sesión:

```
$ startx -- :0 &
```

```
$ startx -- :1 &
```

```
.....
```

```
$ startx -- :6 &
```

## *8.3 Diferentes entornos de escritorio en diferentes consolas gráficas.*

Con la orden startx siempre ejecutamos el entorno de escritorio por defecto. Pero el usuario tiene el control sobre qué entorno de escritorio quiere utilizar cada vez, con la condición, lógicamente, de tener instalados los paquetes correspondientes.

¿Cómo podemos lanzar escritorios diferentes en diferentes consolas gráficas y poder pasar de una a otra sin cerrar sesión?

**Método-A**: Desde el modo texto, iniciamos el servidor X y una terminal en la consola gráfica 0:

```
$ xinit -- :0 &
```

Desde la terminal creada ejecutar:

```
$ startkde
```

Hacemos lo mismo para otra consola gráfica:

```
$ xinit -- :1 &
```

Desde la terminal creada ejecutar:

```
$ gnome-session
```

**Método-B**: Iniciamos el servidor X en la consola gráfica 0 ( sin terminal ):

```
$ X :0 &
```

Volver a la consola de texto y lanzar una terminal en el mismo display:

```
$ xterm -display :0 &
```

Desde este terminal lanzar el escritorio:

```
$ startkde
```

Hacer lo mismo para GNOME en otra consola gráfica.

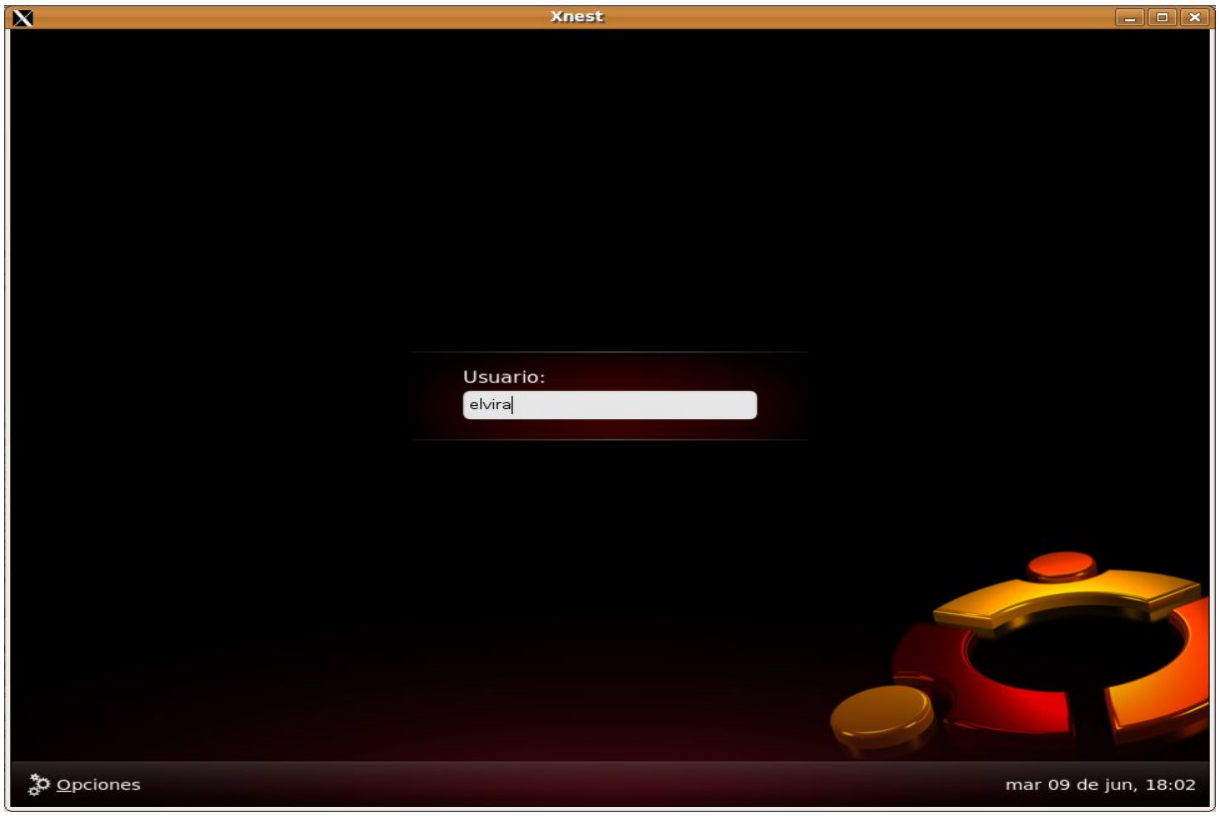
**Método-C:** Utilizamos la orden xnest.

Xnest es a la vez un servidor X y una aplicación cliente X, ya que abre una ventana de conexión en la que el usuario puede trabajar dentro de la misma sesión abierta. Es como una sesión de usuario anidada. Incluso esta herramienta permite abrir ventanas en equipos remotos. Para trabajar con xnest hay que instalar:

```
$ sudo aptitude install xnest
```

Para probarla podemos lanzar gdm dentro de xnest desde la sesión abierta:

`$gdmflexiserver --xnest`




## *8.4 Sesiones gráficas remotas.*

**Método-A**: suponemos que la

-

Máquina A sirve los recursos gráficos pero no tiene lanzado el entorno gráfico.

-

Máquina B ejecuta las aplicaciones (startkde).

Como usuario:

1.

En la máquina A ejecutar:

```
$ xinit -- :0 &
```

2.

En la terminal creada en la máquina A, ejecutar:

```
$ xhost +IP_maquinaB
```

3.

Hacer una conexión vía telnet a la máquina B:

```
$ telnet IP_maquinaB
```

4.

Realizar el proceso de login con cualquier usuario de la máquina cliente :

```
login: usuario1
```

```
password: *******
```

5.

Ya en la máquina B exportar el DISPLAY para que sepa donde tiene que escribir:

```
[usuario1@CLIENTE usuario1]$ export DISPLAY=IP_maquinaA:0.0
```

6.

Ya en la máquina B podemos anzar el entorno de escritorio preferido:

```
[usuario1@CLIENTE usuario1]$ startkde &
```

7.

No cerrar la ventana xterm sino se cerrará la aplicación.

Comprobar qué máquina está ejecutando las aplicaciones con el comando ps. Comprobamos que el servidor después de lanzar los servicios gráficos sobre su DISPLAY 0 lo ha exportado a la máquina B (cliente) que ejecuta en local la aplicación KDE haciendo uso del servidor X que se está ejecutando sobre la máquina A (servidor).

**Método-B**: Ejecución segura de aplicaciones gráficas X remotas con ssh. No se necesita ejecutar xhost porque ssh abre un canal seguro.

1.

En el archivo de configuración /etc/ssh/sshd_config (equipo remoto) activar el reenvío por X11:

```
X11Forwarding yes
```

```
AllowTcpForwarding yes
```

2.

Arrancar el servidor X en la máquina local.

3.

Abrir una xterm en la máquina local. Ejecutar:

```
$ ssh -l usuario maquina_remota
```

4. En la máquina remota:

```
$ xedit &
```

En esta secuencia de órdenes lo que se hace es activar el reenvío X11 en la máquina remota. Igual que en el Método A, arrancamos el servidor X en la máquina local y desde una terminal xterm nos conectamos vía ssh a la máquina remota. Ya en la máquina remota ejecutamos una aplicación X (xedit).

# 9 Conclusión

Hemos comprobado la potencia del sistema gráfico X Window y su capacidad para soportar diferentes entornos de escritorio adaptándose al gusto o preferencias del usuario. Ello contrasta con la rigidez del sistema gráfico de Windows que no permite mas que una débil personalización a nivel de cambio de fondo de pantalla, resolución, colores y poco mas.

El sistema X Window permite trabajar con múltiples escritorios por sesión de usuario, trabajar con múltiples consolas gráficas y cada una de ellas con un entorno de escritorio diferente, establecer de forma directa y segura conexiones remotas, múltiples efectos de escritorio, etcétera, además de las consabidas adaptaciones y preferencias de los usuarios.

Pero también tiene sus inconvenientes y uno de ellos y de peso, es que el sistema X usa mucho ancho de banda, lo cual es crítico cuando varios usuarios están conectados a través de una conexión lenta.

Otro inconveniente es que la información que se transmite no está cifrada, quedando a disposición de los sniffers. Si apostamos por la seguridad, cuando se utilizan los servicios X deben ser reenviados sobre un protocolo seguro como SSH.

De hecho ya se están estudiando alternativas a este sistema gráfico que está vigente desde el año 1987 y que empieza a dar muestras de agotamiento en sus prestaciones. El horizonte parece estar en el 2010...

## Notas

Init [1]: es el primer proceso en ejecución después de la carga del Kernel, y ejecuta los scripts de arranque para inicializar los diferentes servicios, programas o registros que sean necesarios para que el sistema funcione como el usuario quiere o como el administrador estableció. http://es.wikipedia.org/

HAL [2]: **capa de abstracción de hardware** es un elemento del sistema operativo que funciona como interfaz entre el software y el hardware del sistema, proveyendo una plataforma de hardware consistente sobre la cual correr las aplicaciones. (http://es.wikipedia.org)

La variable DISPLAY [3] identifica la máquina (nombre o IP) en la que se está ejecutando el servidor X y el monitor y nº de pantalla virtual. La sintaxis es: DISPLAY=Nombre_del_Servidor:display.screen display: nº de monitor conectado al servidor (comienza en 0); screen: nº de pantalla virtual (comienza en 0).