



# **Preparación para el examen LPI 101**

## **Tema 110.4**

# **Configurando el administrador gráfico**

## **Créditos y licencia de uso**

**Coordinación:**

Manuel Guillán (xLekOx) [lpi@xlekox.org](mailto:lpi@xlekox.org)

**Traducción:**

Juan Maria Gil (Smooth) [yo@juanmaria.com](mailto:yo@juanmaria.com)

**Maquetación:**

Manuel Guillán (xLekOx) [lpi@xlekox.org](mailto:lpi@xlekox.org)

Versión 1.0 (02-03-2005 14:00)

Distribuido por FreeUOC ([www.freeuoc.org](http://www.freeuoc.org)) bajo licencia: Attribution-NonCommercial-ShareAlike2.0 de commons creative



<http://creativecommons.org/licenses/by-nc-sa/2.0/>

## ÍNDICE

### Índice de contenido

Tema 110.4

Configurando el administrador gráfico.....	1
Créditos y licencia de uso.....	2
ÍNDICE.....	3
Introducción.....	4
Arrancando X.....	5
Arranque manual de X.....	5
Los ficheros xinitrc y .xinitrc.....	6
Los ficheros Xclients y .Xclients.....	6
Utilizando X.....	6
Utilizando un emulador de terminal.....	8
Personalizando las aplicaciones X.....	9
Utilizando las teclas especiales.....	10
Gestionando aplicaciones con mal comportamiento.....	10
Ejecutando Remotamente X y Clientes .....	11
Configurando la seguridad de X.....	11
Configurando Clientes Remotos.....	12
Bibliografía y enlaces recomendados.....	14

## **Introducción**

En este capítulo se verá como configurar el administrador gráfico de las X. Se verá como arrancar diferentes entornos gráficos, terminales, etc.

Nota: Este tema puede tener contenidos que a fecha de hoy estén desfasados o cambiados.

Este tema tiene un peso (importancia) de 5 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

## Arrancando X

Hay dos formas de arrancar X en el sistema. El primer método consiste en hacerlo manualmente desde la línea de comandos tras el login. La segunda forma es hacer que el sistema arranque en modo gráfico con un login GUI.

## Arranque manual de X

Se puede arrancar manualmente el sistema X Window con el comando *startx*. A continuación seguiremos el flujo de scripts que comienzan tras la ejecución de *startx*. El problema es que las distintas distribuciones no siguen el mismo camino a través de los scripts de inicio de X. Es muy recomendable recorrer los scripts de la distribución y versión que tenemos instalada para ver exactamente lo que hacen. Además es una experiencia excelente.

```
$startx
```

Este comando suele ser un script que inicializa una serie de variables y, finalmente, llama a la aplicación *xinit*. A continuación se incluye un ejemplo de un script *startx* con comentarios al final:

```
#!/bin/sh
#
# (c) 1999 Red Hat Software, Inc.
bindir=/usr/X11R6/bin
userclientrc=$HOME/.xinitrc
userserverrc=$HOME/.xserverrc
sysclientrc=/etc/X11/xinit/xinitrc
sysserverrc=/etc/X11/xinit/xserverrc
clientargs=""
serverargs=""
```

Esta parte inicializa algunas variables que apuntan a otros ficheros de comandos que se invocarán más adelante.

```
if [ -f $userclientrc ]; then
  clientargs=$userclientrc
else if [ -f $sysclientrc ]; then
  clientargs=$sysclientrc
fi
fi

if [ -f $userserverrc ]; then
  server=$userserverrc
else if [ -f $sysserverrc ]; then
  server=$sysserverrc
fi
fi
```

Estas líneas hacen un *test -f* para comprobar si existen los ficheros indicados en las variables

## Tema 110.4 Configurando el administrador gráfico

anteriores.

La siguiente sección, que aquí omitiremos, ejecuta los ficheros de comandos indicados en las variables, los ficheros *xinitrc* y *xserverrc*.

```
# set up default Xauth info for this machine
mcookie=`mcookie`
serverargs="$serverargs -auth $HOME/.Xauthority"
xauth add $display . $mcookie
xauth add `hostname -f`$display . $mcookie
```

Este bloque configura la información de seguridad de X ejecutando la utilidad *xauth*.

```
xinit $clientargs -- $server $display $serverargs
```

Finalmente, el script *startx* ejecuta *xinit*.

Consejo para el examen: El comando *startx* es un script que ejecuta el proceso *xinit*.

### Los ficheros *xinitrc* y *.xinitrc*

Cada vez que arranca el proceso *xinit* se ejecuta un fichero de comandos. Cada usuario puede crear su fichero personal *.xinitrc* en su directorio home. Si este fichero no existiese se ejecutaría en su lugar el fichero genérico *xinitrc* que se encuentra en */etc/X11/xinit* o en */usr/X11/xinit/xinitrc*. Si se incluyen varias tareas dentro del fichero *.xinitrc*, la última no debería enviarse al background porque el servidor X podría terminar.

Consejo para el examen: Cuando un usuario arranca X con el comando *startx* se ejecuta el script *.xinitrc* del usuario, si no existiese se ejecutaría el script genérico *xinitrc*.

### Los ficheros *Xclients* y *.Xclients*

Algunas distribuciones hacen que el fichero *xinitrc* llame al script *Xclients*. Red Hat utiliza este fichero para chequear y saber que manejadores de ventanas están instalados y para arrancar aquel que el usuario ha configurado como favorito. Un usuario puede tener su propio fichero *Xclients* en *~/.Xclients*.

### Utilizando X

Tras librar la batalla de la instalación y configuración de X nos encontramos con la recompensa de poder disfrutarlo. El entorno X es altamente personalizable hasta el punto que sorprenderá a algunos usuarios nuevos de Linux y X.

### Eligiendo un entorno o manejador de ventanas

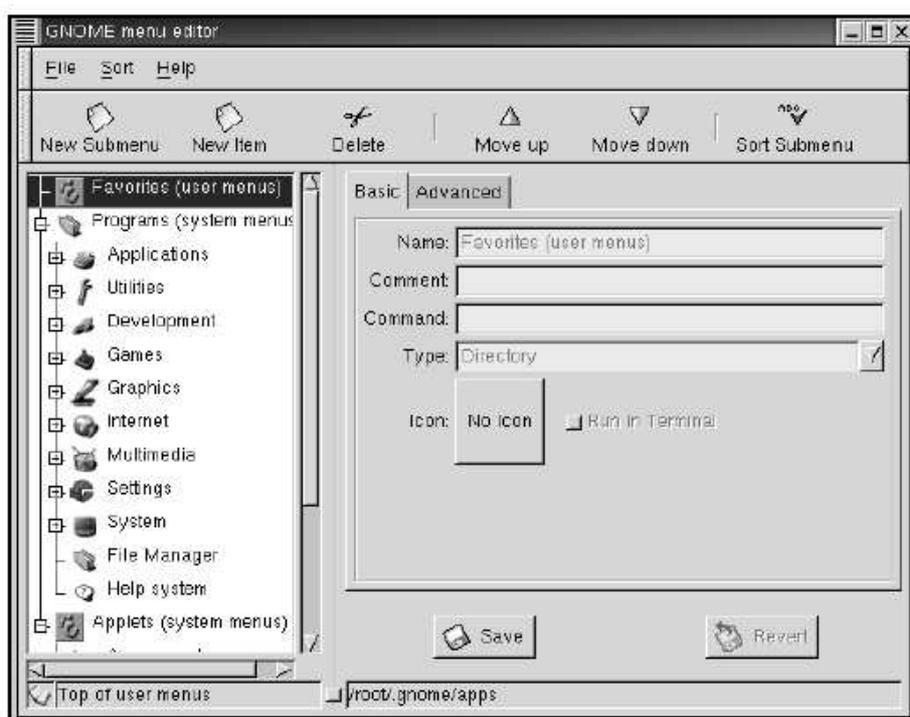
Hay disponible una amplia variedad de manejadores de ventanas, unos más conocidos que otros. Muchas personas utilizan solo un manejador de ventanas determinado mientras otras utilizan un entorno completo de escritorio como como KDE o GNOME, que combina un manejador de ventanas junto con una serie de aplicaciones y herramientas integradas en el mismo.

El manejador de ventanas escogido se arranca normalmente desde el fichero Xclient, Xsession, o .xinitrc, dependiendo de la distribución y de la elección del usuario. Red Hat utiliza el fichero Xclient mientras que Debian usa un fichero Xsession genérico para los usuarios locales y remotos. Para ejecutar un manejador de ventanas distinto a los incorporados en esos ficheros se puede especificar manualmente en el fichero .xinitrc dentro del directorio home. Este script se ejecutará cada vez que se arranque el proceso *xinit*.

Una vez escogido el entorno X podemos trabajar para personalizar el sistema. Un método podría ser la utilización de temas que cambiarán el aspecto general del sistema. <http://www.themes.org> es un sitio web muy popular dedicado a temas.

Aunque los manejadores de ventanas necesitaban modificar a mano los ficheros de configuración para hacer hasta el cambio más simple, los manejadores de ventanas actuales suelen incluir un panel de control con herramientas para realizar estas tareas más fácilmente.

A continuación se muestra una imagen del editor de menús de GNOME.



### Utilizando los clientes X

Hay varios parámetros de línea de comandos utilizados por la mayor parte de las aplicaciones clientes X. Estos parámetros nos permiten establecer la configuración de inicio mediante una sintaxis estandar. Son conocidos como “X Toolkit options” (*Opciones de la colección de herramientas de X*), y aunque puedan ser un poco pesados para teclearlos a mano si que son muy prácticos para utilizarlos en un script o en un menú

## Tema 110.4 Configurando el administrador gráfico

La siguiente tabla muestra estos parámetros.

<i>Opción</i>	<i>Función</i>
-bg <color> o -background <color>	Establece el color de fondo por defecto para la aplicación.
-fg <color> o -foreground <color>	Establece el color de texto por defecto para la aplicación.
-bd <color> o -bordercolor <color>	Establece el color del borde de ventana por defecto para la aplicación.
-bw <número> o -borderwidth <número >	Establece la anchura en pixels del borde de la ventana.
-display nombrehost:display.pantalla	Especifica el nombre del host y los números de display y pantalla en los que se mostrará la aplicación.
-fn <fuente> o -font <fuente>	Especifica la fuente utilizada para el texto.
-geometry <anchura>x<altura>+<x>+<y>	Especifica el tamaño y posición inicial de la ventana.
-iconic	Le indica a la aplicación que se inicie en modo icono si fuese posible.
-name <nombre>	Especifica el nombre bajo el cual se encontrarán los recursos de las aplicaciones.
-rv o -reverse	Le indica a la aplicación que simule vídeo inverso si fuese posible.
+rv	Le indica a la aplicación que no intente simular vídeo inverso.

### **Utilizando un emulador de terminal**

Muchos usuarios ejecutan X como una interfaz gráfica de múltiples emuladores de terminal simultáneos en una sola pantalla. Existen diferentes emuladores de terminal como para satisfacer las necesidades de todo el mundo. A continuación se indican varios de los más populares.

#### **xterm**

El emulador de terminal “estándar” xterm existe desde hace mucho tiempo. Proporciona emulaciones DEC VT102/VT220 y Tektronix 4104 para aplicaciones . Soporta un gran número de opciones de línea de comandos para configurar el emulador en tiempo de ejecución y también soporta las opciones estándar X Toolkit que se mostraron anteriormente

#### **rxvt**

Para aquellos usuarios que no necesiten toda la funcionalidad de xterm existe rxvt. rxvt no emula tantos sistemas como hace xterm y tampoco incluye soporte de las opciones X Toolkit. El beneficio es una menor utilización de memoria puesto que fue diseñado para ser utilizado en sistemas con muchos emuladores de terminal abiertos simultáneamente.

#### **aterm**

aterm es similar a rxvt en tanto que no soporta tantas emulaciones como xterm. Tampoco soporta

las opciones X Toolkit. Fue diseñado para trabajar con el manejador de ventanas AfterStep, pero no es indispensable. aterm ofrece varias opciones que no incluye xterm, entre ellas un modo de transparencia en el cual se ve el fondo del escritorio a través de la ventana.

### **Eterm**

Si buscamos el emulador de terminal de mejor aspecto y más configurable, Eterm es ese. Fue diseñado para trabajar con el manejador de ventanas Enlightenment, y se nota. Necesita más memoria que otros emuladores pero admite tantas configuraciones y personalizaciones como se pueda desear. También soporta temas.

### **gnome-terminal**

El entorno de escritorio GNOME viene con su propio emulador de terminal GNOME-compatible. Está escrito utilizando las librerías GTK (GIMP Tool Kit) así que tendrá el mismo aspecto y funcionamiento del tema GNOME que se tenga instalado. Es un emulador de terminal muy completo y una buena elección si se utiliza GNOME.

### **konsole**

Para no ser menos, el grupo KDE incluye en su entorno de escritorio un emulador de terminal llamado konsole. Se trata de un emulador completo que hereda el aspecto y funcionamiento de KDE.

## **Personalizando las aplicaciones X**

Como ya se ha comentado, el entorno X es altamente personalizable. Las aplicaciones escritas para utilizar entornos de escritorio como GNOME y KDE heredan los aspectos y funcionalidades que hayan sido configurados en esos entornos, ¿Pero que pasa con el resto de las aplicaciones?. Hay muchas aplicaciones que no han sido escritas para un entorno de escritorio y solamente utilizan las librerías estándar X Toolkit, pero aun así nos proporcionan métodos para personalizar su apariencia.

Las aplicaciones que utilizan las librerías X Toolkit pueden personalizarse utilizando el formato Xresource. En /usr/X11R6/lib/X11/appdefaults o en /etc/X11/app-defaults podemos encontrar numerosos ficheros de ejemplo de Xresource. El nombre de cada uno de ellos es el de la aplicación a la que se corresponden. A continuación veremos un ejemplo tomado del fichero Xman, el cual se corresponde con la aplicación *xman* que visualiza las páginas de *man*.

```
*manualBrowser.Title: Manual Page
*manualBrowser.IconName: Manual Page
*manualBrowser.geometry: 600x600
```

Aquí podemos ver el título por defecto, el nombre del icono y la geometría de la ventana. De hecho, la sintaxis de estas entradas puede ser muy compleja, pero las más simples que serán las que más probablemente tengamos que cambiar siguen el formato de éste ejemplo que es el nombre de la aplicación seguido por la propiedad a cambiar.

Al cambiar algún ajuste en un fichero de los que se encuentran en apps-default, este ajuste quedará alterado para todos los usuarios del sistema. Para cambiar el comportamiento de la aplicación solo para un usuario deberíamos poner las entradas del fichero Xresource en otro fichero

llamado .Xdefaults dentro del directorio home del usuario correspondiente.

Consejo para el examen: El fichero .Xdefaults guarda las personalizaciones del usuario para las aplicaciones X.

### Utilizando las teclas especiales

Hay varias combinaciones especiales de teclas que pueden utilizarse dentro de X. Los usuarios que solían cambiar de consolas virtuales en Linux con la combinación Alt-TecladeFunción se sorprenderán al ver que esto ya no funciona en X.

La siguiente tabla nos muestra las combinaciones especiales.

#### Combinaciones de Teclas

Combinación	Función
Ctrl-Alt-<+ del Teclado numérico>	Cambia a una resolución de mayor vídeo si estuviese configurada.
Ctrl-Alt-<- del Teclado numérico>	Cambia a una resolución menor de vídeo si estuviese configurada.
Ctrl-Alt-Retroceso	Salida rápida de X a no ser que esta opción estuviese desactivada en XF86Config.
Ctrl-Alt-<F1 hasta F6>	Cambia a las consolas de texto.
Ctrl-Alt-<F7>	Cambia de nuevo al modo gráfico desde una consola de texto.

### Gestionando aplicaciones con mal comportamiento

Debido a la naturaleza gráfica de las aplicaciones X, éstas son normalmente mas complejas y difíciles de desarrollar que las aplicaciones de texto. Por esta causa muchas aplicaciones X se han ganado la mala reputación de ser inestables o de causar problemas.

Una de las aplicaciones que más quejas ha reportado es el navegador Netscape Navigator, pero no es el único culpable.

Algunas aplicaciones, simplemente caen mientras otras no se cierran cuando se les indica, se quedan abiertas y siguen consumiendo recursos que podrían utilizarse para otras cosas.

Si detectásemos que el sistema rueda más lento de lo normal o escuchamos continuamente el disco duro trabajar mientras intercambia memoria con la partición de swap, deberíamos comprobar la lista de procesos. Asegurémonos de que ninguna aplicación que pensamos que debería estar cerrada aparece como abierta y ejecutándose en el background.

Consejo para el examen: Muchas de las aplicaciones inestables ruedan en X, los objetivos del examen mencionan específicamente *Netscape Navigator* y *tkrat* (un agente de correo gráfico), así que es conveniente recordarlas.

## Ejecutando Remotamente X y Clientes

Una característica muy buena de X es la posibilidad de mostrar transparentemente aplicaciones a través de una red. De esta forma podemos configurar equipos económicos de sobremesa para que los usuarios manejen aplicaciones que se ejecutan en un gran servidor de red. X permite tanto ejecutar remotamente aplicaciones sueltas como iniciar un escritorio completo desde un sistema remoto.

## Configurando la seguridad de X

X soporta varios mecanismos de autenticación, algunos muy complejos. Para la mayoría de los usuarios el método básico basado en host funciona bien y es fácil de configurar y manejar. La seguridad basada en el host se configura con el comando *xhost*. Se pueden consultar los ajustes de seguridad actuales ejecutando *xhost* sin opciones.

```
[root@redhat /root]# xhost
access control enabled, only authorized clients can connect
INET:brain
INET:marvin
```

También podemos utilizar el comando *xhost* para ver si disponemos de permisos para mostrar clientes en un sistema remoto. Esto se consigue cambiando el valor de la variable DISPLAY al del host remoto y después ejecutando *xhost*. Por ejemplo para ver si podemos mostrar aplicaciones en el host `redhat.the-nashes.net` escribiríamos:

```
export DISPLAY=redhat.the-nashes.net:0.0
xhost
```

En el siguiente apartado volveremos con la variable DISPLAY.

El comando *xhost* emplea solo unas pocas opciones de línea de comandos. Para habilitar la autenticación basada en host y limpiar todos los hosts permitidos utilizaremos *xhost -*. Por ejemplo:

```
[root@redhat /root]# xhost -
access control enabled, only authorized clients can connect
```

Para desconectar todas las autenticaciones y permitir a todo el mundo que se conecte a nuestro sistema y a nuestras aplicaciones escribiríamos *xhost +*. Por ejemplo:

```
[root@redhat /root]# xhost +
access control disabled, clients can connect from any host
```

¡Precaución!: Ejecutar *xhost +* no es una buena idea ya que cualquier usuario podría conectarse a nuestro servidor X.

Para autorizar a determinados hosts a conectarse utilizamos también la opción *+*. Por ejemplo, para dar permisos de conexión a tres hosts escribiríamos lo siguiente:

## Tema 110.4 Configurando el administrador gráfico

```
[root@redhat /root]# xhost +deedee marvin brain
deedee being added to access control list
marvin being added to access control list
brain being added to access control list
```

Para eliminar un host de la lista de permisos utilizaremos la opción -.  
Por ejemplo, para eliminar a *deedee* escribiríamos lo siguiente:

```
[root@redhat /root]# xhost -deedee
deedee being removed from access control list
```

Finalmente, para asegurarnos de que nuestra autenticación es correcta escribiríamos solamente *xhost*:

```
[root@redhat /root]# xhost
access control disabled, clients can connect from any host
INET:brain
INET:marvin
```

Consejo para el examen: La autenticación basada en host se configura con el comando *xhost*.

### Configurando Clientes Remotos

Hay dos formas básicas de indicar a una aplicación que se muestre en un sistema remoto. La primera emplea la variable de entorno DISPLAY. El formato de uso de la variable DISPLAY es el siguiente:

```
DISPLAY=nombrehost:display.pantalla
```

Por ejemplo:

```
DISPLAY=marvin.the-nashes.net:0.0
```

El campo *nombrehost* puede ser tanto un nombre DNS como una dirección IP. Si solo hubiese un usuario utilizando un servidor X en el sistema remoto el número de display sería 0.

El número de pantalla se utiliza solo en entornos multipantalla y puede omitirse si solo se utiliza una pantalla.

Después de modificar la variable DISPLAY, cualquier cliente X que se inicie se mostrará automáticamente en el sistema especificado en la variable, si la autenticación así lo permitiese.

El segundo método para mostrar clientes remotamente consiste en emplear la opción de línea de comando *-display* reconocida por la mayoría de los clientes.

La sintaxis es la siguiente:

```
xclient -display nombrehost:display.pantalla [argumentos de xclient]
```

El formato, como se puede observar, es el mismo que el empleado con la variable DISPLAY.

## Tema 110.4 Configurando el administrador gráfico

Éste método es práctico para cuando solo deseamos mostrar uno o dos clientes en otro sistema.

Configurando el login remoto

Otra función muy práctica de XDM es la de permitirnos conectarnos remotamente a otro sistema y utilizar nuestro sistema local solo para visualización. También podemos configurar un sistema para proporcionar un cliente con un menú de distintos sistemas de login. La funcionalidad subyacente la proporciona y controla XDMCP (X Display Manager Control Protocol).

Estos ajustes deberían funcionar con todos los clientes X y servidores con soporte XDMCP.

Referencia cruzada: El “Linux Terminal Server Project” (Proyecto de Servidor de Terminales para Linux) ha creado un sistema de login que puede ser empleado en estaciones de trabajo sin discos. Podemos encontrar más información en <http://www.ltsp.org>.

Conectándonos a sistemas remotos

Si un sistema remoto rueda XDM, es fácil conectar con dicho sistema desde X y ejecutar aplicaciones desde él.

Se utiliza la siguiente sintaxis:

```
X -query <servidor nombre>
```

X también soporta la posibilidad de buscar en la red local cualquier servidor que ruede XDM. Esto se consigue con el siguiente comando:

```
X -broadcast
```

Los sistemas Linux también puede configurarse para proporcionar una lista de hosts con los que podemos conectarnos, esta lista se llama chooser. Para solicitar un chooser emplearíamos el siguiente comando:

```
X -indirect <nombre servidor>
```

Consejo para el examen: Asegurémonos de conocer la diferencia entre estos tres comandos para el examen.

***Bibliografía y enlaces recomendados***

LPI 1 Certification Bible (Bible) by Angie Nash, Jason Nash  
John Wiley & Sons; Bk&CD-Rom edition (July 1, 2001) ISBN: 0764547720

LPI Linux Certification in a Nutshell by Jeffrey Dean  
O'Reilly & Associates; 1st ed edition (May 15, 2001) ISBN: 1565927486

CramSession's LPI General Linux Part 1 : Certification Study Guide  
CramSession.com; ISBN: B000079Y0V; (August 17, 2000)

Referencias Unix Reviews  
<http://www.unixreview.com/documents/s=7459/uni1038932969999/>

Página LPI: [www.lpi.org](http://www.lpi.org)

Apuntes IBM: <http://www-106.ibm.com/developerworks/edu/l-dw-linux-lpir21-i.html>

Manuales GPL: <http://www.nongnu.org/lpi-manuals/>