

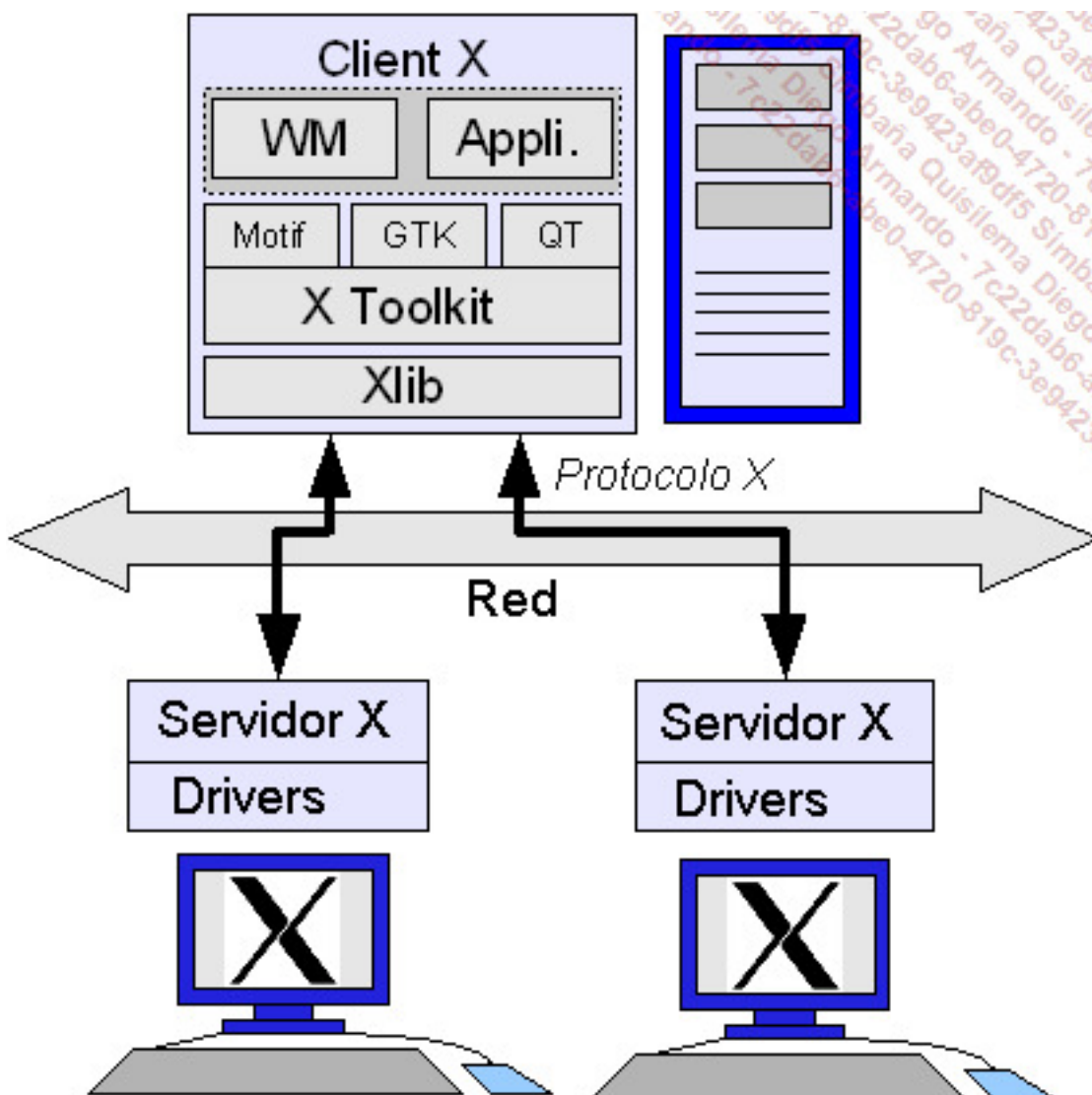
¿Cómo funciona un entorno gráfico?

1. X Window System

a. Un modelo cliente/servidor

Mientras en otros sistemas operativos se integra la interfaz gráfica en lo más profundo del sistema, Unix y Linux disponen de una arquitectura gráfica totalmente diferente. El sistema gráfico básico se llama **X Window System** o, de manera más corriente, **X Window**, **X11** o sencillamente **X**.

X no es sólo un simple programa. Es un sistema gráfico completo encargado de dibujar y gestionar los eventos de los componentes habituales de un entorno gráfico de usuario **GUI** (*Graphical User Interface*): ventanas, botones, menús, listas, menús desplegables, casillas de selección, cursores de ratón, etc. Tome nota: X puede gestionar y mostrar estos componentes gráficos, pero no se encarga de instalarlos. X sólo gestiona las interacciones entre el hombre y la máquina.



Arquitectura X Window

X tiene otra particularidad: es cliente/servidor. A menudo, el servidor X es un componente de software en un ordenador que dispone de un teclado, un ratón y una pantalla. Recibe y responde a órdenes de visualización, o procedentes del teclado o ratón. El cliente X se conecta al servidor y le manda órdenes de visualización, peticiones de entrada a través del teclado o el estado del ratón. Dicho de otro modo, un cliente X es un programa capaz de dialogar con el servidor X. En realidad, un cliente X es un programa gráfico. Para poder comunicarse con el servidor, utiliza un componente llamado Xlib. El cliente y el servidor no siempre están en la misma máquina. El servidor que gestiona la visualización puede encontrarse en un primer ordenador y el programa gráfico en otro. Las órdenes, llamadas peticiones, entre el cliente y el servidor pasan por la red.



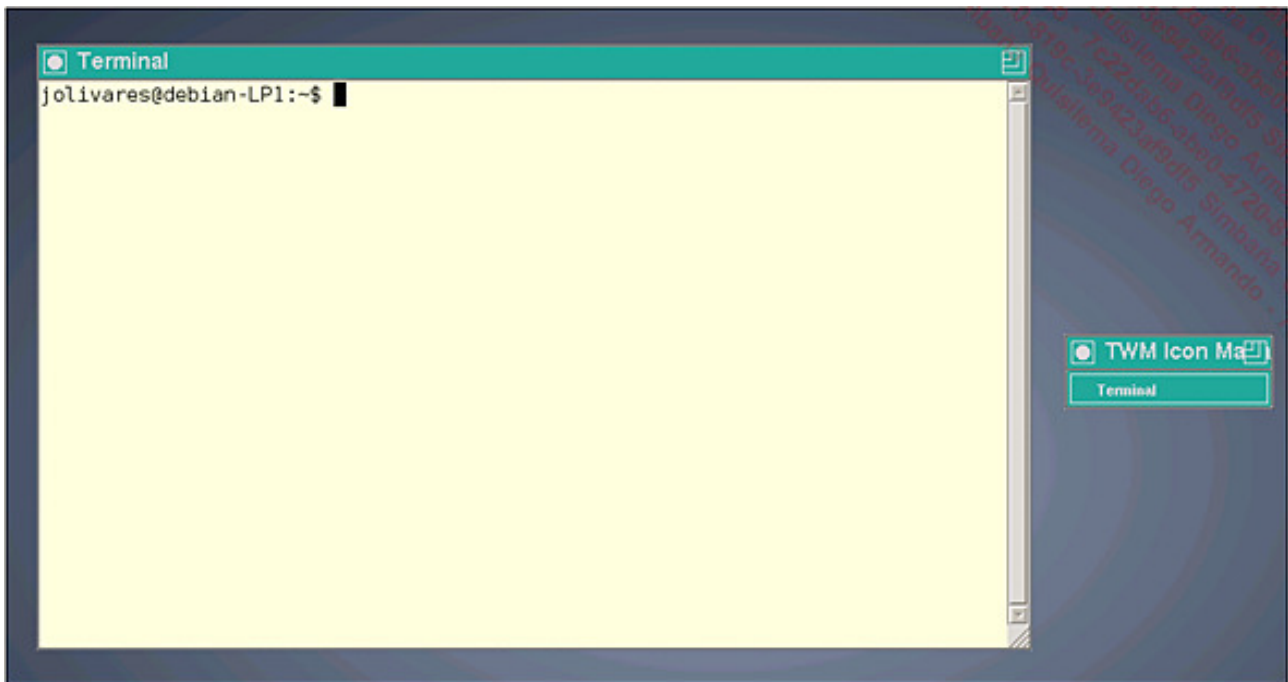
No confunda X Window con Windows. El primero es un sistema de visualización cliente/servidor; el segundo, un sistema operativo. El «Window» de X Window no tiene «s». Además, Windows y X Window no son compatibles entre sí, aunque se pueda instalar un servidor X en Windows.

Un usuario puede iniciar un programa (cliente) en su máquina local conectada por la red a un servidor remoto donde se está ejecutando realmente el servidor de ventanas.

Si ejecuta solo X Window, ejecuta únicamente el servidor. El resultado puede sorprenderle: una pantalla gris con una cruz como cursor de ratón. Por mucho que lo intente, no habrá ventana que venga a alegrarle la vista y los clics del ratón no tienen efecto alguno. Aunque iniciara un cliente, notaría que no hay ventanas.

b. El gestor de ventanas

Entre las muchas peticiones gráficas, la más importante es la de crear una ventana y decorarla dibujando los diversos elementos de ésta: la barra de título, el marco, los diversos botones. Como X sólo facilita lo imprescindible, se limita a dibujar la ventana, pero no le corresponde determinar cómo se debe renderizar el resto de los elementos. Otro programa cliente X debe decir al servidor cómo dibujar la ventana: es el **gestor de ventanas** o *Window Manager*. El servidor X visualiza el resultado dibujado por este gestor: ventanas, selecciones, movimientos y decoraciones (estilos, colores, etc.).



Un gestor simple: twm

Esto quiere decir que no hay un único gestor de ventanas, sino varios. Algunos son muy sencillos y básicos, y se limitan a lo mínimo; por ejemplo TWM, que aquí mostramos con una ventana y un menú. Observe su austeridad espartana. Sin duda, no es la mejor manera de sorprender a sus amigos con la calidad gráfica de Linux.

Otros son muy completos y permiten trabajar muy cómodamente porque, además de las ventanas básicas, proponen temas visuales agradables y personalizables, menús contextuales y, a veces, paneles de configuración. Es el caso de WindowMaker, con el que incluso se pueden cambiar menús, temas visuales, añadir botones en la barra de la derecha, etc. Durante mucho tiempo, este gestor de ventanas fue el más utilizado: rápido, eficaz, frugal en el consumo de recursos.



WindowMaker un gestor de ventanas evolucionado

c. Los widgets y los toolkits

Los componentes y su librería

Incluso a los gestores de ventanas más desarrollados les falta algo: una integración más avanzada de los programas (y de sus estilos) en el entorno. La razón estriba en el hecho de que, salvo casos muy especiales, los **WM** (*Windows Manager*) usan librerías gráficas propias con funcionalidades y componentes específicos. A los elementos que forman parte de la interfaz gráfica (botón, menú, campo de entrada, etc.) se les conoce como **Widget** (*Window gadget*), o sea, «chisme» para ventanas. Las librerías gráficas contienen juegos completos.

Todas estas librerías gráficas deben ofrecer funciones para dibujar y gestionar widgets. Cuando la librería contiene un kit completo y extendido de widget, se llama **Widget Toolkit**,

o de manera más sencilla **toolkit** o caja de herramientas para la interfaz gráfica. Existen varios, que se diferencian según su complejidad, su uso, su estética (relativamente), el lenguaje de programación para utilizarlos, etc. X Window dispone de un toolkit por defecto llamado **Xt** (*X Toolkit*). El más conocido y utilizado ha sido durante mucho tiempo **MOTIF**, más aún porque es un estándar POSIX IEEE, hasta que, liberado en torno al año 2000, MOTIF fue considerado software propietario y, por esa razón, los programadores gráficos crearon otros conjuntos de componentes. La consecuencia es de lo más interesante: una pléyade de toolkits diferentes, incompatibles entre sí, que dibujan de manera totalmente diferente los widgets. Todavía encontrará en Linux muchos ejemplos en los que un componente aparece visualmente diferente en varios programas, dando la impresión de caos de estilos heterogéneos.

GTK, Qt

Al no poder utilizar Motif (y su equivalente libre Lesstif, que no estaba todavía muy preparado), los desarrolladores de interfaces gráficas en Linux concentraron sus esfuerzos en dos toolkits concretos, excelentemente programados y que permiten hacer abstracción totalmente de X Window. Existen varias versiones que no necesitan siquiera X Window (para Windows, Macintosh, Wayland, los asistentes personales, los teléfonos, etc.).

- ˘ El primero se llama **GTK** (*GIMP toolkit*). Fue creado para dibujar la interfaz gráfica del programa de retoque de imágenes GIMP. Gracias a su eficacia gráfica, muchos programadores lo han incorporado a sus propios programas y GTK con el tiempo se ha vuelto independiente. Hoy es la librería por defecto del entorno de escritorio GNOME. GTK está escrito en lenguaje C.
- ˘ El segundo se llama **QT** (*cute, mono*). La Q no significa nada: los desarrolladores de este toolkit pensaban que esta consonante, Q, era muy bonita en el editor de textos que utilizaban... Está desarrollado en C++, pero es compatible con una decena de lenguajes. Además de las funciones de dibujo y gestión de los widgets, QT propone un entorno completo de desarrollo de aplicaciones gráficas y no gráficas: bases de datos SQL, XML, multithreading, gestión de archivos, internacionalización, etc. La librería QT es la utilizada por el entorno ofimático KDE.
- ˘ Hay otros muchos toolkits, como **MOTIF**, **AWT**, **ATHENA**, **VCL**. Aunque son menos conocidos, todavía hay aplicaciones que los utilizan (como LibreOffice que usa VCL).

GTK y QT no son compatibles, al menos de manera nativa. Si ejecuta un programa desarrollado con GTK en KDE, el contorno de la ventana tendrá el estilo de KDE (QT), pero GTK dibujará su contenido.

Algunos programas utilizan sus propios toolkits. Es el caso de LibreOffice.org y es la principal razón de su lentitud durante su primera descarga. Cuando ejecuta este programa, éste debe cargar también su toolkit, lo cual resulta redundante con el de su gestor de ventanas o de su entorno ofimático.

d. Los escritorios virtuales

Si hay una característica de Linux que se ha echado de menos en sus competidores (ha aparecido solamente a partir de Windows 10), es la del escritorio virtual. ¿Qué hacer cuando hay tantas ventanas abiertas que estamos obligados a minimizarlas para recuperar las escondidas, o volver a dimensionarlas para verlas? Pues utilizar los escritorios virtuales.

X Window permite la utilización de varios espacios de trabajo en una misma sesión. En vez de tener un escritorio, puede tener dos, tres, cuatro... En efecto, en teoría puede obtener tantos escritorios como desee si dispone de bastante memoria. Es el gestor de ventanas, o el entorno de escritorio, el que elige (y por lo tanto, usted).

El interés es obvio. En vez de concentrar todas sus ventanas en un solo escritorio, repártalas en los escritorios disponibles. Más aún: sólo precisa de un simple clic para ir de un escritorio a otro, o incluso para desplazar una ventana de un escritorio a otro. Además, es posible hacer que una ventana se siga visualizando en varios escritorios virtuales (es decir, «colgarla» en la pantalla).

2. Los entornos de escritorio

Si Unix, y Linux en particular, han conservado durante mucho tiempo la fama de sistema operativo para profesionales, universitarios, informáticos o excéntricos gurús, es porque faltaba un entorno gráfico orientado a la ofimática y de fácil manejo. Ya en los primeros capítulos se comentaba que los hábitos que los usuarios han ido adquiriendo con las interfaces de Windows o MacOS no pueden desaparecer de un día a para otro. Durante

años, los gestores de ventanas fueron difíciles de configurar. ¿Se le puede pedir a alguien acostumbrado al ratón que modifique un archivo de configuración en modo texto, cuya sintaxis no es precisamente sencilla?

Hacia 1996, Linux alcanzó la madurez entre las empresas, pero sigue habiendo un mercado por conquistar. En el otoño de 1995, Windows puso un contundente final a la carrera de la mayoría de los sistemas operativos para PC de oficina. Los DR-DOS, PC-DOS y sobre todo OS/2, muy superiores, han sobrevivido primero y, finalmente, desaparecido. Los entornos gráficos de Unix y Linux no pueden competir en términos de facilidad de manejo y oferta de programas para el gran público con los sistemas de Microsoft y de Apple. Todo queda por hacer: hay que alcanzar a los competidores, mejorar e innovar.

Este titánico trabajo no desanima a los programadores. Ya que Linux, para el gran público, no es de fácil manejo, hay que crear un entorno ofimático gráfico adaptado a sus necesidades. Con pocos meses de diferencia dos equipos distintos asumen el proyecto.

KDE



Logo de KDE

En octubre de 1996 un primer equipo liderado por Matthias Ettrich anuncia el lanzamiento del proyecto **KDE** (*K Desktop Environment*). La K ya no tiene significado particular (salvo al principio: kool) y sólo es la letra más cercana a la L de Linux en el alfabeto. KDE ofrece una interfaz gráfica y aplicaciones unificadas alrededor de un toolkit llamado QT y desarrolladas en C++. KDE 1 ya era impresionante en 1997 y permitía utilizar Linux como si fuera una mezcla de Windows 3.1 y 95. Pero fue a partir de KDE 2, ya en el año 2000, y gracias a una nueva arquitectura, cuando KDE reveló su potencia e igualó a los entornos gráficos de sus competidores. La salida de la versión 3 va aún más lejos y, superando a todos sus competidores, integra los diversos componentes de forma avanzada. La versión 3 se encuentra todavía disponible a través del proyecto Trinity. KDE es el entorno gráfico favorito de la mayoría de los usuarios de Linux; pretende ser el más completo en términos de integración, de configuración y de oferta de software (a riesgo de tener numerosos duplicados y dejar la opción al usuario).

Las versiones 4 y 5 han representado una verdadera ruptura en cuanto a las versiones anteriores. Donde KDE 3.0 se aproximaba a un Windows XP o un Windows 7, las versiones posteriores ponen en plano el concepto mismo de entorno de escritorio. KDE platform, y luego KDE plasma, ofrecen un entorno moderno y de gran rendimiento. Se han efectuado enormes esfuerzos de integración y KDE no tiene nada que envidiar a las interfaces de Windows y de Mac OS X.

KDE se sigue conociendo bajo ese nombre, pero hoy en día se presenta como una plataforma llamada a su vez Plataforma KDE o KDE Plasma.

GNOME



Logo de Gnome

En agosto de 1997, un equipo liderado por Miguel de Icaza y Federico Mena decide crear un entorno de escritorio totalmente libre para competir con KDE (cuyo QT no era libre en aquel momento). **GNOME** (*Gnu Network Object Model Environment*) es un entorno de escritorio basado en el toolkit GTK y programado en C. Es el entorno gráfico de escritorio oficial del proyecto GNU. La filosofía de GNOME difiere de manera radical de la de KDE. GNOME tiene un estilo voluntariamente depurado, que se centra en las funcionalidades esenciales de un entorno y que privilegia algunas aplicaciones en detrimento de otras. La integración no es tan exhaustiva, ya que las aplicaciones desarrolladas en GTK no son todas aplicaciones GNOME. GNOME 3, un remake completo, salió en abril de 2011.

La evolución es lenta. La última versión, de septiembre de 2020, es la 3.38. Su lentitud de evolución y las opciones a veces sorprendentes de sus desarrolladores han provocado la aparición de muchos derivados (forks), como Mate o Cinnamon. Este también ha estado en el origen de Unity en Ubuntu.

El proyecto Freedesktop



Logo de Freedesktop

Los desarrolladores de KDE y GNOME decidieron trabajar en una base común de configuración para una mejor integración de las aplicaciones GNOME en KDE, y viceversa, con el fin de armonizar la infraestructura del conjunto. El objetivo no es diseñar un escritorio único, ya que cada uno tiene sus ventajas. El resultado es la creación de **Freedesktop**, una zona de comunicación y colaboración informales destinada a trabajar en favor de la interoperabilidad de los diversos entornos gráficos para Linux y Unix. Freedesktop es neutral a nivel tecnológico, no se privilegia ningún entorno. Pero, a menudo, lo que sale de Freedesktop se integra en GNOME y KDE: del menú principal común al programa de detección de nuevo hardware, pasando por los copiar y pegar, los atajos de teclado y una multitud de mejoras que Freedesktop ha aportado.