

# GNOME 2.0

**Rodrigo Moya**  
Proyecto GNOME, Madrid

**rodrigo@gnome-db.org**

Este documento presenta la ponencia sobre GNOME 2.0 presentada para el IV Congreso Hispalinux, a celebrarse entre los días 15, 16 y 17 de noviembre de 2001.

En él se pretende introducir a los asistentes las maravillas que estarán presentes en la siguiente gran versión del escritorio GNOME, que debería estar disponible unas semanas después del congreso.

## 1. Introducción

Desde sus primeros inicios, uno de los objetivos principales del proyecto GNOME ha sido el de ofrecer una plataforma de desarrollo completa y moderna que pueda ser utilizada para el desarrollo de aplicaciones de gran calidad y que cumplan las necesidades de los usuarios de entornos de escritorio.

Ya con la versión 1.4 de GNOME, aparecida en el mes de marzo-abril de 2001, vimos la introducción de nuevas tecnologías en esta plataforma de desarrollo, que, durante todos estos meses han ido asentándose. En esta versión 2.0, que marca la madurez de muchas de estas tecnologías, veremos la consolidación de uno de los desarrollos más importantes que se han hecho dentro del proyecto GNOME en particular y del software libre en general: Bonobo, el sistema de componentes desarrollado para el proyecto GNOME.

Así, GNOME 2.0 se presenta como la plataforma más completa y compacta presentada hasta el momento por el proyecto GNOME, lo cual significa no sólo la disponibilidad de un montón de tecnologías para los desarrolladores, sino también la disponibilidad de

un buen número de aplicaciones que hacen uso de estas tecnologías, y que facilitarán enormemente el trabajo de los usuarios.

## 2. Bonobo, el sistema de componentes

La consolidación de Bonobo como una de las tecnologías más importantes del proyecto GNOME no es más que el cumplimiento de uno de los objetivos marcados inicialmente dentro de GNOME: la creación de un sistema de componentes para UNIX/Linux que permitiera el desarrollo de aplicaciones "profesionales", de gran envergadura, al estilo de lo que nos encontramos en otros sistemas operativos.

Esta consolidación se traduce, en GNOME 2.0, en que todas, absolutamente todas las aplicaciones GNOME usan Bonobo, con las ventajas que esto conlleva, como el fácil reemplazo de cada una de las partes sin afectar al resto del sistema, etc. Cuando una aplicación almacena su configuración, o cuando intenta visualizar un fichero en determinado formato, o cuando interactúa con el sistema, en todos estos casos, la comunicación entre los distintos componentes del sistema se hace mediante Bonobo. Esto es algo que no es nuevo dentro del proyecto GNOME, donde desde el principio, de una u otra forma, casi todas las aplicaciones usaban CORBA (la tecnología sobre la que se basa Bonobo). La diferencia es que antes el uso era un tanto "opcional", por lo que cada aplicación lo implementaba como mejor le convenía al/los programador(es) de ese proyecto determinado. Ahora, mediante Bonobo, todas las interfaces entre las distintas aplicaciones están claramente definidas por Bonobo, por lo que cualquier aplicación puede comunicarse con cualquier otra sin necesidad de conocer las interfaces CORBA que esa aplicación determinada usa para ello. Sólo tiene que conocer las interfaces de Bonobo, que son públicas y están ampliamente documentadas.

Uno de los pasos más importantes que se han dado con respecto a la versión 1.4, es la separación de todas las librerías en dos partes: la parte no gráfica, perfectamente utilizable por aplicaciones totalmente ajenas a GNOME y al sistema X Window, y la parte gráfica, que esa sí que es dependiente del sistema gráfico y de GNOME. Esto está facilitando el uso de la tecnología base de GNOME en cualquier tipo de aplicación, por lo que se están empezando a ver multitud de aplicaciones/soluciones que integran, por ejemplo, entornos de servidores, donde instalar el sistema gráfico es una aberración, con entornos de oficina, donde priman las estaciones de trabajo con GNOME (o cualquier otro sistema de escritorio).

Otro punto en el que se ha puesto especial énfasis es en solucionar el problema de los "bindings" (enlaces con otros lenguajes de programación), que se estaba empezando a convertir en algo bastante problemático. Y es que, por cada librería de la plataforma

GNOME, hay que escribir una versión para cada uno de los lenguajes de programación soportados (que, en el caso de GNOME, siempre han sido numerosos). Así, a partir de ahora se pretende que las aplicaciones exporten sus funcionalidades internas no a través de librerías en C que luego tengan que ser "enlazadas" a otros lenguajes, si no a través de los interfaces CORBA definidos por Bonobo, de forma que simplemente haya que añadir soporte para Bonobo en cada uno de los lenguajes soportados para que aplicaciones escritas en dichos lenguajes puedan acceder a todas esas funcionalidades exportadas por las aplicaciones. Este objetivo se está cumpliendo perfectamente, y ya podemos usar Bonobo desde lenguajes tales como C (en el que está desarrollado el propio Bonobo), C++, Guile, Perl, Python, Java.

También es de destacar la fusión que han experimentado en estos últimos meses OAF (Object Activation Framework), el sistema de activación de objetos CORBA de GNOME 1.4, y el propio Bonobo. La razón para esta fusión es que no estaba muy clara la línea que separaba a ambas librerías, más bien parecía que, si bien con objetivos aparentemente distintos, en algunas cosas se solapaban la una con la otra. Así, ahora tenemos bonobo-activation, que consiste en el antiguo OAF (sistema de activación de objetos, como un DNS para objetos CORBA) junto con la parte básica del sistema de componentes, que no es más que el famoso interfaz CORBA Bonobo::Unknown. El resto de interfaces CORBA de Bonobo están en las dos librerías de las que se compone ahora Bonobo: `libbonobo` (los interfaces no gráficos) y `libbonoboui` (los interfaces con salida gráfica, como controles, documentos compuestos, etc).

### 3. GTK+/GLib 2.0

Aparte de Bonobo, la otra estrella de GNOME 2.0 es el binomio compuesto por GLib y GTK+ que, para esta nueva versión traen muchas novedades.

GLib es una librería de funciones para programación en C, que contiene multitud de utilidades para facilitar la programación en este lenguaje. Incluye cosas como la gestión de estructuras de datos como listas enlazadas, arrays dinámicos, tablas de búsqueda, árboles binarios, o la gestión de Entrada/Salida asíncrona. Pero, puesto que se busca que esta librería contenga todo lo (razonablemente) posible, para esta versión se ha añadido el sistema de objetos de GTK a GLib, de forma que no sea necesario enlazar con las librerías GTK (que dependen de X Window) para usar este potente sistema de objetos, que permite simular, de una forma bastante transparente para los programadores de C, la programación orientada a objetos, paradigma sobre el que se basan todas las librerías que componen la plataforma GNOME.

GTK+ es la librería que se encarga de los "widgets", tales como los botones, ventanas,

cajas de texto, etc. Incluye todo tipo de "widgets" que nos permitieran realizar fácilmente todo tipo de aplicaciones con un interfaz gráfico. En esta nueva versión, las novedades en cuanto a "widgets" se traducen en una limpieza generalizada de los "widgets" existentes en versiones anteriores (procediendo incluso a la calificación como obsoletos de algunos de ellos, como por ejemplo algunos de los tipos de listas), y, por fin, la introducción de un "widget" que permite mostrar datos clasificados en filas y columnas mucho más avanzado que el ya obsoleto GtkCList (que estaba muy, muy limitado).

Pero, la novedad más importante de GTK+ es el uso de Pango para la representación del texto en pantalla. Pango es una librería que permite la representación de texto en distintos alfabetos, teniendo en cuenta todos los detalles de cada uno de estos alfabetos, como la dirección de la escritura, la separación entre caracteres, etc. El uso de Pango en GTK+ es obligatorio, pero totalmente transparente al programador, que sólo tiene que preocuparse de usar caracteres Unicode (un estándar para la representación de caracteres en múltiples alfabetos) para todo el texto que quiera representar en pantalla; GTK+ y Pango se encargaran del resto. Pango viene a cubrir las necesidades que muchos usuarios de países con alfabetos distintos al occidental para poder usar el sistema X Window. Con Pango, todos estos usuarios podran usar sus escritorios GNOME en su idioma nativo.

Tambien es de destacar la creación de ATK (Accessibility Tool Kit), por parte de la empresa Sun™, que es un conjunto de clases abstractas, integradas en GTK, que permiten el desarrollo de aplicaciones "accesibles" (para personas con deficiencias físicas). Con la introducción de ATK al nivel de GTK+, se espera que esto lleve a la capacitación de TODAS las aplicaciones GNOME de forma que puedan ser usadas sin problemas por personas discapacitadas, algo que, junto con la internacionalización conseguida con Pango, va a hacer aumentar, aun más, el número de usuarios y las posibilidades de GNOME.

Como último, comentar que la versión de GTK+ para Windows, existente desde hace más de dos años, forma ya parte de la distribución oficial de GTK+, lo cual significa que Windows es, desde ya, otra de las plataformas soportadas "oficialmente" por GTK+. Ya hay aplicaciones de la talla del GIMP o Glade que tienen sus correspondientes versiones para el sistema de las ventanas.

## 4. Nuevas tecnologías

Como en cada versión de GNOME, GNOME 2.0 trae consigo alguna que otra nueva tecnología que se añade a la plataforma de desarrollo. Hay que constatar que algunas de estas librerías/tecnologías existen desde hace tiempo, pero es en GNOME 2.0 cuando

han pasado a formar parte "oficialmente" de la plataforma GNOME.

## 4.1. Bonobo-config

Si ya en GNOME 1.4 se substituyó gnome-config (el sistema de configuración basado en archivos del estilo de los .INI de MS Windows, usado en todas las versiones anteriores a GNOME 1.4) con GConf, un sistema de configuración distribuido basado en CORBA, ahora en GNOME 2.0 hemos presenciado, si no una substitución, sí un relevo generacional que se irá afianzando en versiones posteriores (GNOME 2.2, etc).

Este relevo lo protagoniza bonobo-config, que es un sistema de acceso a bases de datos de configuración basado en Bonobo. De momento, bonobo-config simplemente actúa como una capa basada en Bonobo para acceder a las bases de datos de configuración manejadas por GConf, pero se espera que en próximas versiones se unifiquen las funcionalidades de GConf y bonobo-config en un solo sistema. Precisamente por este "futuro relevo", se aconseja a todos los programadores de aplicaciones que usen bonobo-config para acceder a la configuración de sus aplicaciones, pues GConf, en su estado actual (es decir, la librería en C libgconf), es más que probable que desaparezca en posteriores versiones de GNOME.

Lo interesante de bonobo-config con respecto a GConf es que el acceso a la configuración se hace a través de interfaces CORBA definidos en Bonobo, de forma que se cumple el objetivo comentado en la sección de Bonobo de exportar toda la funcionalidad de las aplicaciones (en este caso, del sistema de configuración) a través de Bonobo, para así permitir el acceso a dicha funcionalidad a través de los interfaces bien conocidos (y documentados) de Bonobo.

## 4.2. libgnomedb/libgda

El acceso a bases de datos es algo especialmente importante en el mundo corporativo, por tanto, en GNOME 2.0 se han incluido, como parte de la plataforma de desarrollo GNOME, libgnomedb y libgda, dos librerías que han sido desarrolladas durante más de tres años como parte del proyecto GNOME-DB.

Hace pocos meses, se hizo un gran trabajo dentro del proyecto GNOME-DB para pulir todas las asperezas y presentar, por fin, un producto listo para ser usado por cualquier persona con un mínimo de conocimientos de BBDD. Este trabajo incluyó, por supuesto, las librerías libgnomedb y libgda, que son dos librerías que ofrecen toda la funcionalidad del proyecto GNOME-DB para que pueda ser reutilizada en cualquier proyecto.

libgda es la librería que implementa el acceso a bajo nivel (apertura/cierre de conexiones, ejecución de comandos y obtención de resultados, etc) a todas las bases de datos soportadas por el proyecto GNOME-DB (PostgreSQL, MySQL, ODBC, Oracle, MS SQL Server, Sybase, Primebase). Está basada, como no, en Bonobo, y permite un fácil acceso a todas las operaciones básicas de una base de datos, como es la ejecución de comandos, la obtención de resultados, la ejecución de transacciones, etc.

libgnomedb ofrece, basados en libgda, un conjunto de "widgets" orientados al desarrollo de aplicaciones visuales de acceso/manejo de datos. Estos "widgets" incluyen cajas de entrada de datos para distintos formatos, rejillas para la visualización de filas devueltas por las bases de datos, listas, o incluso cosas más avanzadas como cuadros de conexión o visualizadores de errores, que permiten fácilmente desarrollar aplicaciones de acceso a datos.

### 4.3. SOUP, SOAP en GNOME

SOUP es una implementación, en C, del protocolo SOAP (Simple Object Access Protocol), que es un protocolo basado en XML para la ejecución de procedimientos remotos a través de Internet. El modo de funcionamiento de SOAP es muy sencillo: simplemente se usa un transporte de red (normalmente HTTP, aunque está previsto que funcione también sobre otros protocolos como SMTP) para enviar ficheros XML en los que se describen una serie de operaciones y sus parámetros correspondientes. Así, un cliente SOAP genera un fichero XML en el que especifica el nombre de una operación a ser ejecutada, junto con sus parámetros de entrada, los envía a un servidor SOAP, y recibe, como respuesta, otro fichero XML en el que se especifican los resultados de salida de la operación ejecutada en el servidor.

Este protocolo, y más concretamente, la implementación de SOAP en SOUP, se utilizará más bien para hacer servicios web (como el famoso .NET de Microsoft, que también está basado en SOAP para la parte de servicios a través de Internet), que podrán integrarse fácilmente en aplicaciones GNOME a través de SOUP. Un ejemplo de las cosas que puede ofrecernos SOAP puede encontrarse en

XMethods

(ver sección de referencias), donde podemos encontrar distintos servidores que ofrecen multitud de servicios a través de SOAP, como son por ejemplo servicios de búsqueda, de traducciones, etc.

## 4.4. GStreamer, multimedia en GNOME

Una de las principales carencias que ha habido desde siempre en GNOME era en el campo de la multimedia, donde durante mucho tiempo apenas hubo nada. Hace tiempo, por fin, apareció GStreamer, que es una librería para el tratamiento de datos multimedia bastante avanzada y con unos objetivos bastante interesantes.

GStreamer no está pensada para usuarios finales, si no para programadores, pero es de esperar que cuando GNOME 2.0 esté por fin disponible, ya tendremos algunas aplicaciones (reproductores de sonido o video, tratamiento de imagen, etc) que hagan uso de esta potente librería.

## 4.5. libxslt

XML es, junto con CORBA, una de las tecnologías más usadas en GNOME. Y si ya desde hace tiempo disponíamos de libxml (una librería en C que implementa un analizador XML con la que podemos manejar y crear documentos XML), desde hace poco tenemos también a nuestra disposición libxslt, que es una librería que nos permite realizar, desde nuestros programas, transformaciones XSLT, que es un mecanismo mediante el cual podemos convertir documentos XML en otros formatos, todo a través de hojas de estilo.

Al estar esta funcionalidad disponible en una librería que forma parte de la plataforma GNOME, muchas aplicaciones están empezando a hacer uso de ella para distintos cometidos. Destaca, por ejemplo, el uso de XSLT en el proyecto GNOME-DB, en el que se usa XSLT para convertir la salida generada por la ejecución de un informe en distintos formatos, tales como PDF, HTML, PostScript, etc.

## 5. Conclusiones

GNOME 2 es, sin duda, uno de los mayores avances que se van a producir en los próximos meses dentro del mundo del software libre, que va a acercarse aun más si cabe su uso a un número muy elevado de personas, tanto principiantes como expertos informáticos provenientes de otros sistemas operativos, que, al encontrarse frente a GNOME, tendrán a su disposición un escritorio moderno y potente que cubre todas sus necesidades. Pero para que esto sea cierto, es necesario el desarrollo de nuevas aplicaciones más potentes y fáciles de usar, todo ello basándose en las tecnologías descritas en esta ponencia.

Es por ello primordial conseguir más y más gente que desarrolle para GNOME, uno de los objetivos principales de esta ponencia.

## Bibliografía

N. Walsh y L. Muellner, *DocBook: The Definitive Guide*, 1a edición, O'Reilly & Associates, Inc., octubre de 1999, 156592-580-7, disponible en <http://www.docbook.org> .

*Documentación para desarrolladores GNOME*: <http://developer.gnome.org> .

*Pango*: <http://www.pango.org> .

*GTK+*: <http://www.gtk.org> .

*GNOME-DB*: <http://www.gnome-db.org> .

*GStreamer*: <http://www.gstreamer.net> .

*libxml y libxslt*: <http://www.xmlsoft.org> .

*Proyecto GNOME*: <http://www.gnome.org> .

*GNOME Hispano*: <http://www.es.gnome.org> .