



Cómo Colaborar con GNOME

Carlos García Campos
carlosgc@gnome.org

III Jornadas telemáticas del Kaslab
8, 9, 10 de julio de 2005



Ciclo de vida de un error

- Durante el uso de una aplicación detectamos un error (bug)
- Obtenemos la última versión de la aplicación para comprobar si el bug ha sido ya corregido
- Reportamos el bug si no ha sido ya reportado
- Buscamos en los fuentes el origen del bug
- Solucionamos el bug
- Preparamos un parche y lo enviamos
- Esperamos noticias del maintainer de la aplicación



Obtener la última versión CVS (Concurrent Version System)

- CVS es el sistema utilizado en GNOME para gestionar los fuentes
- Descargar un módulo

```
$ export CVSROOT=:pserver:anonymous@anoncvs.gnome.org:/cvs/gnome
```

```
$ cvs login
```

```
$ cvs -z3 co <módulo>
```

- Compilar

```
$ cd <módulo>
```

```
$ ./autogen.sh --prefix=/usr
```

```
$ make
```

```
# make install
```



Obtener la última versión Jhbuild

- Jhbuild es un conjunto de scripts para automatizar la compilación de GNOME
- Permite compilar e instalar de forma paralela a la versión de GNOME instalada por nuestra distribución.
- Resuelve dependencias y descarga las últimas versiones (CVS) de los módulos de GNOME



Reportar el bug bugzilla

- Bugzilla es el sistema de seguimiento de errores utilizado en GNOME
- Se trata de una aplicación web que permite:
 - Enviar informes de error (no solo errores como tal)
 - Consultas personalizadas de la base de datos
 - Realizar aportaciones (comentarios, parches, etc.)
 - Notificaciones por correo electrónico
 - etc.



Localizar el origen del bug

- El objetivo es navegar por los fuentes en busca del origen del problema
- No hay una herramienta o receta
- Uso de comandos como find, grep, etc. y sentido común
- Algunos consejos
 - Leer el fichero README
 - Leer el fichero HACKING (si existe)
 - Leer el fichero AUTHORS



Localizar el origen del bug gdb

- Si el error provoca la salida abrupta del programa sin demasiada información, podemos usar el depurador gdb

```
$ gdb programa
```

```
(gdb) run programa [argumentos]
```

```
(se produce el error)
```

```
(gdb) backtrace
```

- Otras opciones útiles

```
(gdb) break
```

```
(gdb) next | step
```

```
(gdb) print | watch
```

```
(gdb) kill
```



Solucionar del bug

Modificación del código

- Es necesario respetar las guías de estilo de GNOME
 - Tabulador de 8 espacios
 - Convenio de nombres (consistencia)
 - Funciones
 - modulo_submodulo_operacion
 - Nombres descriptivos: stwinsz () => set_window_size
 - Minúsculas: funciones y variables (separador: '_')
 - Mayúsculas: macros y enumerados (separador: '_')
 - Mixto: typos y nombres de estructuras
 - Limpieza y claridad
 - General: seguir el convenio existente al modificar



Solucionar del bug

Modificación de la interfaz gráfica

- Seguir la guía de Interfaces Humanas de GNOME (HIG)
- Documento que explica como se deben de realizar los diseños de las interfaces gráficas
- Garantiza una consistencia entre las aplicaciones del escritorio
- Resultado: aplicaciones orientadas al usuario, faciles de usar y con una interfaz agradable



Preparar un parche

El archivo ChangeLog

- El ChangeLog contiene el historial de cambios realizados en cada modulo
- Formato

aaaa-mm-dd Nombre completo <correo_electronico>

* ruta archivos_modificados (funciones): descripción de los cambios

*

- Máximo 72 columnas
- Codificación de caracteres UTF-8



Preparar un parche parche.diff

- Usamos de nuevo el cvs para obtener las diferencias de nuestra versión local con respecto a la versión del cvs

```
$ cvs diff -u [path] > archivo.diff
```

- Acudimos de nuevo al bugzilla y unimos el parche al bug previamente creado
- Es posible que posteriormente el parche necesite de algunas modificaciones para ser aceptado por el maintainer



Y si no soy capaz de solucionarlo?

- Olvida la idea de abandonar!!!
- Busca ayuda
 - Devhelp
 - Busca código en otras aplicaciones
 - Lista de correo del proyecto (si no tiene, gnome-devel-list o gnome-desarrollo puede ser un buen sitio)
 - Canal de irc del proyecto (si no tiene, #gnome-hackers o #gnome-hispano puede ser un buen sitio)



GNOME Love!

- Iniciativa, iniciada por Fernando Herrera para ayudar a la gente que quiere empezar a contribuir en GNOME
- Web: <http://live.gnome.org/GnomeLove>
- Lista de correo: gnome-love@gnome.org
- IRC: [#gnome-love \(irc.gimp.org\)](irc://irc.gimp.org/#gnome-love)



Referencias

- GNOME
<http://www.gnome.org>
- GNOME Hispano
<http://es.gnome.org>
- Recursos para desarrolladores
<http://developer.gnome.org>
- Listas de correo de GNOME
<http://mail.gnome.org>
- Listas de correo de GNOME Hispano
<http://listas.es.gnome.org>