

EVOLUCIÓN Y GESTIÓN DE LA CONFIGURACIÓN SOBRE LIBREOFFICE

Secretario:

Fernando Ramos González

Representante:

Nicolás Robayo Moreno

Técnico especialista:

Jesús Rivero Capellán

Project Manager:

Javier García Gallardo

Índice:

➤ Introducción	Pag 3
➤ Gestión del Código Fuente	Pag 4
➤ Gestión de Pruebas	Pag 11
➤ Gestión de la Construcción	Pag 15
➤ Integración/Despliegue continuos	Pag 19
➤ Gestión de entregables	Pag 20
➤ Gestión del despliegue	Pag 20
➤ Gestión de incidencias y depuración	Pag 21
➤ Gestión de la variabilidad	Pag 27
➤ Mapa de herramientas	Pag 29
➤ Conclusiones	Pag 30



Introducción

Hemos elegido para este proyecto analizar esta suite debido a que usa una licencia LGPL, es de código abierto, y permite la participación activa de la comunidad en el desarrollo del mismo.

A continuación resumimos los aspectos fundamentales para entender el resto del documento:

- En cuanto a la gestión del código fuente LibreOffice trabaja con Git, lo que nos va a facilitar su seguimiento.
- En la Gestión de la construcción, LibreOffice es multiplataforma. Vamos a realizarlo en Linux por que la mayoría de los desarrolladores trabajan en Linux y consideramos que habría más documentación actualizada bajo el sistema operativo Linux.
- Como entregables LibreOffice nos brinda el software como tal, el código fuente en C++, manuales de usuario y al menos 6 productos que son Writer, Calc, Impress, Draw, Base y Math.
- LibreOffice se despliega en escritorio y es multiplataforma, como hemos dicho anteriormente.
- Para el reporte de bugs LibreOffice utiliza Bugzilla y listas de correo. En Bugzilla se registran los bugs para ser resueltos, mientras que las listas de correo tienen una índole más informativa y de reporte.
- En cuanto a la gestión de la variabilidad, el paquete de software LibreOffice, contiene una base común a todo el proyecto, que gestiona varios productos sobre ella misma, dando como resultado la suite completa del programa. De esta manera, tenemos un núcleo de la aplicación que da origen a varios productos basados en uno mismo.
- En cuanto a la integración, LibreOffice se integra con otros sistemas de suites ofimáticas en la nube, y además incluye la posibilidad de ampliación personalizada e individualizada a través de un sistema de extensiones, que permite adaptar personalmente el software, y además se mantiene en desarrollo continuo gracias a su naturaleza de desarrollo libre en la comunidad.
- En el desarrollo de LibreOffice se utiliza como herramienta Bugzilla, listas de correos, Git, entornos para C++ (ninguno específicamente).

Con objeto de facilitar la búsqueda de aspectos concretos del desarrollo de LibreOffice, en este trabajo utilizaremos un formato de preguntas/respuestas.

Gestión del código fuente

LibreOffice utiliza Git para la gestión del código.

¿ Cuánto tiempo se tarda en bajarse los archivos del código fuente por git ?

La descarga a través de Git, funciona a una media de 170Kb en una línea promedio de 10Mb de ancho de banda, con un tráfico en condiciones estándar y en una máquina virtual de Ubuntu. El tamaño de datos a descargar es de X Gb, y en total, supone una totalidad de tiempo de 1Gb.

¿Podemos instalar libreOffice en cualquier ordenador con un solo click?

No, LibreOffice gestiona en distintos paquetes su instalación, Instalador Principal, Interfaz de Usuario Traducida (sólo en linux) y la Ayuda Incorporada de LibreOffice, además todos estos paquetes deben instalarse en orden.

¿Se pueden hacer ramas?

Sí, de hecho, LibreOffice las usa para dividir sus líneas de trabajo.

¿Cómo se gestionan las ramas?

LibreOffice está estructurado en varias ramas divididas en funcionalidad:

- **master** - principal rama de desarrollo
- **libreoffice-X-Y** - release branch, suited to release the X.Y product line (like libreoffice-3-4)
- **libreoffice-X-Y-Z** - release branch, suited to release the concrete version (like libreoffice-3-4-0, to release 3.4.0)
- **integration/dev300_mXYZ** - integration branch, to merge code from OpenOffice.org, or elsewhere [obsolete]
- **feature/NAME** - topic (feature) branch, to develop a feature without changing any other branch
- **private/\$GERRITUSERNAME/NAME** private test branches etc.

A continuación, los LibreOffice Master tienen la responsabilidad de hacer merge con el repositorio principal.

¿Cómo se hacen los merge? ¿Con que frecuencia se hacen?

Los merges se hacen de las ramas individuales a las ramas master, libreoffice-X-Y, y libreoffice-X-Y-Z, con las siguientes restricciones:

- Sobre libreoffice-X-Y sólo se solucionan bugs.
- Sobre libreoffice-X-Y-Z sólo se solucionan bugs de bloqueo y traducciones.

La frecuencia de los merges no está definida.

¿Por qué motivos se crean las ramas?

Con el fin de dividir la funcionalidad ya que hay varias líneas de trabajo.

¿Se hacen ramas específicas para Sistemas Operativos?

No, como hemos visto antes sus ramas están gestionadas por funcionalidad y luego durante la construcción al momento de compilar el código, se utilizan unos comandos específicos para determinar el SO.

¿Cuándo se eliminan las ramas?

Nunca, porque el desarrollo es continuo.

¿Qué permisos dan a los usuarios?

Todos los usuarios tienen todos los permisos, commit, pushing pulling, merge, crear un branch sobre su repositorio, etc. Pero los de niveles más bajos efectúan un pushing se realiza sobre un repositorio secundario, no el principal.

¿Cómo se hace un commit?

Se puede hacer commits hacia nuestro repositorio en local según el funcionamiento habitual de Git; la limitación de colaboración está a la hora de hacer un push con el repositorio principal o el repositorio de la comunidad. Hay dos repositorios, un repositorio público comunitario y un repositorio maestro o gestionado. Sólo ciertos usuarios con privilegios, Committers, pueden hacer push directamente sobre el repositorio maestro. El resto de usuarios deben hacer un push al público. Sobre éste se hará una revisión de los cambios realizados por parte de un Committer, y, si los considera apropiados, los aplicará sobre el repositorio maestro.

```
git commit -s --author='Author name <author@mail.address>'
```

¿Qué patrón de compartición de código se usa?

Se usa un patrón de rama principal. Como hemos dicho anteriormente, LibreOffice posee 2 repositorios el principal y el público.

¿Cómo se resuelven los conflictos?

Se pueden presentar los siguientes casos:

- Un documento fue modificado y guardado mientras otro usuario ha abierto el documento. Si el cambio no entra en conflicto, el documento es guardado, pero si el cambio entra en conflicto, aparece el diálogo de resolución de conflictos. El usuario A debe decidir qué versión guardar para el conflicto. Si mantener la suya o mantener la que ya estaba. Cuando todos los conflictos estén resueltos, el documento es guardado. Mientras el usuario A resuelve los conflictos, ningún otro usuario podrá guardar el documento compartido.
- Si otro usuario intenta guardar el documento compartido y resuelve los conflictos del momento. El usuario A ve el mensaje que un merge-in está en progreso, el usuario A puede optar por cancelar la orden de guardar por ahora o volver a intentar guardarlo después.

¿Qué herramientas se utilizan?

No se define ninguna herramienta para programar en concreto, permitiendo a los desarrolladores utilizar la que más les interese.

Para Git puede utilizarse también cualquier herramienta, pero se recomienda utilizar Gerrit, que es una interfaz web para utilizarlo.

Ejercicio: Usar Gerrit



Hemos aprovechado las instrucciones que nos ofrece LibreOffice en su wiki para la instalación y configuración de gerrit para realizar el ejercicio.

Configuración de Gerrit

1. Si ya dispone de un checkout de libreOffice/core, tan sólo tienes que ejecutar `./logerrit setup`, el cual intentará automáticamente instalar las claves y configurar el ssh. Deja la ventana abierta y no contestes a "Which user name did you choose?"

The screenshot shows the Gerrit web interface for LibreOffice. The search results for 'status:open' are displayed in a table with columns for ID, Subject, Project, Branch, Updated, CR, and V. An OpenID registration modal window is overlaid on the search results, titled 'Code Review - Register New Account'. The modal contains the OpenID logo, the text 'Register with Gerrit Code Review at gerrit.libreoffice.org', a text input field with the URL 'https://www.google.com/accounts/08/id', and buttons for 'Register' and 'Close'. Below the input field are options for 'Remember Me', 'Register with a Google Account', and 'Register with a Yahoo! ID'. The modal also includes a section titled 'What is OpenID?' with a brief description and a link to 'Get OpenID'.

ID	Subject	Project	Branch	Updated	CR	V
I9f800a97	binaryurp: silently ignore superfluous MUSTREPLY	core	libreoffice-4-1-4	4:26 PM		
I5a20976c	fdoc#71784: Preserve prop "allow row to break across	core	master	3:36 PM		
I2c6d8b1e	sw: add CharInteropGrabBag uno property to charac	core	master	12:17 PM		✗
I88461f73	fdoc#39956: Delete JAB	core	master	11:36 AM		
Ifd51a75a	fdoc#71786: Implemented Glossary folder	core	master	10:05 AM		
I9ff6beb2	fdoc#71936: Add Excel 2010 functions GAMMA.DIST	core	master	7:31 AM		
I515f9c12	fdoc#60698: Merge tvhlp1 into ucpchelp1	core	master	6:02 AM		✓
I038fe43f	fdoc#71434: don't show master text if Placeholder ty	core	libreoffice-4-2	5:39 AM	+1	✓
I038fe43f	fdoc#71434: don't show master text if Placeholder ty	core	libreoffice-4-1	5:39 AM	+1	✓
Iae2bb26a	fdoc#60698: Merge hatchwindowfactory into svf	core	master	5:06 AM		✓
I45d6a249	resolved fdoc#53103: actually use the external data fi	core	libreoffice-4-1	12:22 AM		
Iefbf565f	Fix for LO crash while saving and corruption of file s	core	master	Nov 26		✓
Ifbf5e76	fdoc#69562: make calc functions CEILING and FLOOR	core	master	Nov 26	-1	
Ia278127d	fdoc#70338: Fixed file corruption issue after save fo	core	master	Nov 26		
I0700c735	fdoc#71784: Invalid Sections getting added to the doc	core	master	Nov 26		
I6a785b42	VCL - Remove usage of DBG_CTOR and DBG_DTOR	core	master	Nov 25		
Ic6ce0eb3	fdoc#71911: show the top level dir of a new change	dev-tools	master	Nov 24		
Ic94b41bf	ESC-bug-stats: bugs.freedesktop.org -> bugs.librec	dev-tools	master	Nov 23		
I0fde057b	Remove old outdated gallery images	core	master	Nov 23		
I038b58f2	resolved fdoc#61946: use correct TabDelta for Update	core	libreoffice-4-1	Nov 23		
Ifba91aa8	fdoc#67011: Run Display: Yield through KDEXL: Yield	core	master	Nov 23		
Iaa4a1f77	CID#1078495/1078496: Copy-paste error	core	master	Nov 23		
I61d34859	sw/export docx: add unit test for document with header and section(s)	core	master	Nov 22		
Ie8875bcb	Code clean-up/consolidation task	core	master	Nov 21	-1	

2. Añade la dirección de correo que usas cuando trabajas con git en las preferencias bajo “Contact Information”.

All | My | Projects | Groups | Documentation | LibreOffice Code Review | Javier Garcia <jgchroni@gmail.com> | Settings | Sign Out

Changes | Drafts | Watched Changes | Starred Changes | Draft Comments

Welcome to Gerrit Code Review

Please review your contact information:

The following contact information was automatically obtained when you signed-in to the site. This information is used to display who you are to others, and to send updates to code reviews you have either started or subscribed to.

Full Name: Nicolás Robayo Moreno

Preferred Email: jgchroni@gmail.com | Register New Email ...

Save Changes

Select a unique username:

Username: nicrobmor | Select Username

Register an SSH public key:

Gerrit Code Review uses [public-key cryptography](#) and [SSH](#) to authenticate you during git's push and pull commands to hosted projects. Registering your public key allows Gerrit to identify you whenever you connect through SSH.

This step can also be completed at a later time.

Add SSH Public Key
[\(GitHub's Guide to SSH Keys\)](#)

3. Introduzca un nombre de usuario válido y la clave SSH que le devolvió la ventana de comandos. Acto seguido introduzca el nombre de usuario elegido en la terminal.

https://gerrit.libreoffice.org/#/register/q/status:open,n,z

Aplicaciones | Para acceder rápidamente a una página, arrástrala a esta barra de marcadores. Importar marcadores ahora... | Otros marcadores

Select a unique username:

Username: nicrobmor | Select Username

Register an SSH public key:

Gerrit Code Review uses [public-key cryptography](#) and [SSH](#) to authenticate you during git's push and pull commands to hosted projects. Registering your public key allows Gerrit to identify you whenever you connect through SSH.

This step can also be completed at a later time.

Add SSH Public Key
[\(GitHub's Guide to SSH Keys\)](#)

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDax6C4H+ny6gOUP0ZXS2e3HTJ oWY09Vq85rM1M5S07xzPR7VbMf
ZnZNGU1Iri1sRfUteSh0RihVG0uB1PqHqQCr+3f4p6aCKN3oQT+QBJurzTfzJ7Ke79Fm6cUX2RkKX101t
1cFSYzR+acFQIQVzmspZxebweMGf1CvcG1uVCOct24NZ0TeJDBkzyrOZn84FJgGd4q1Hy1ye27BVVEod
760m21xsoY+y0h1d4N64FXhplacSoeDYm1Vp6ShImnpdzT3JLIRJ0ou/1N/EwyN+h08m818kF56Nw
ueq7RpuunnJzBCABr315xUH0o0ekq3+nbQJcSe1S1P0L nico@nico-Virtua1Box
```

Clear | Add

Server Host Key

Fingerprint:
f5:a8:13:af:a0:d8:2d:4b:db:6c:c6:fd:c1:03:b6:87

Entry for ~/.ssh/known_hosts:
[gerrit.libreoffice.org]:29418 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQD...yDSW0w7J1z

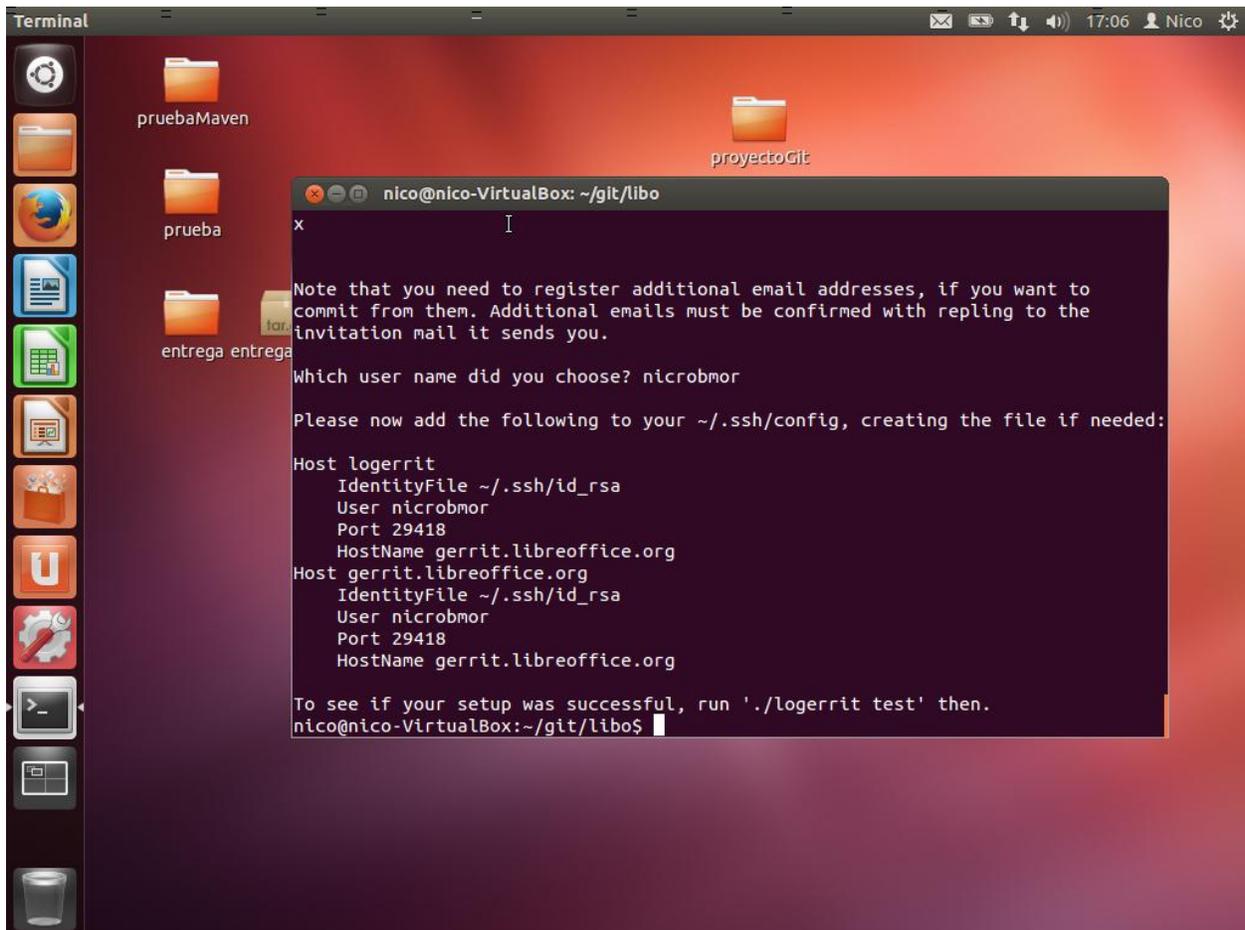
4. Asegúrese de que en `/${HOME}/.ssh/config` aparecen las siguientes secciones (ambas son idénticas salvo la línea de "Host"):

```
Host logerrit
```

```
IdentityFile /path/to/your/private-key  
User YOUR_USERNAME_HERE (cf. above)  
Port 29418  
HostName gerrit.libreoffice.org
```

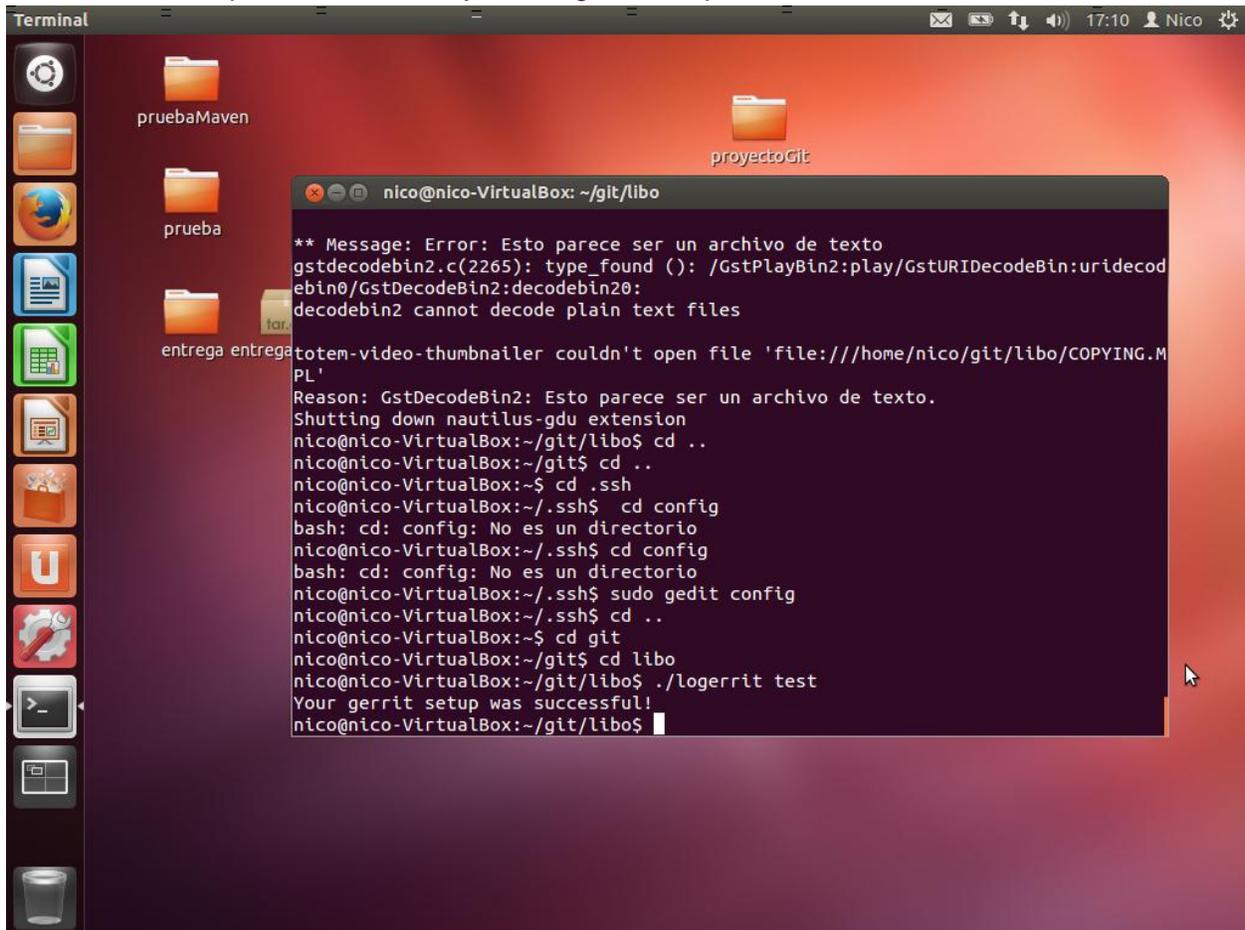
```
Host gerrit.libreoffice.org
```

```
IdentityFile /path/to/your/private-key  
User YOUR_USERNAME_HERE (cf. above)  
Port 29418  
HostName gerrit.libreoffice.org
```



5. Compruebe en que `/path/to/your/private-key` debe incluir el nombre de fichero también. Por ejemplo: `~/.ssh/id_rsa`.

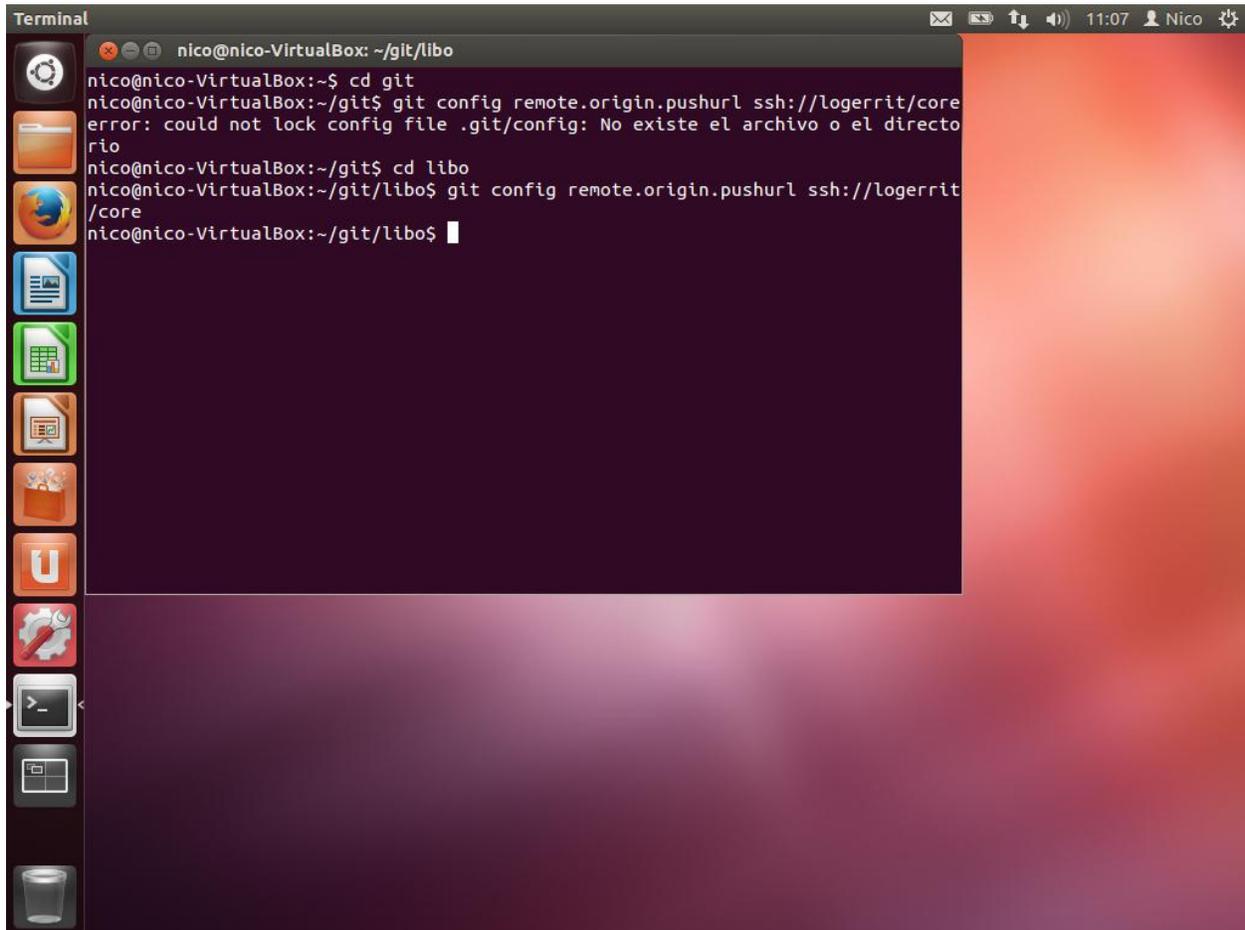
6. Para probar la instalación ejecute run `./logerrit test` (en el repositorio de LibreOffice). Debería aparecer un mensaje: `Your gerrit setup was successful!`



The screenshot shows a Linux desktop environment with a terminal window open. The terminal displays the following output:

```
nico@nico-VirtualBox: ~/git/libo
** Message: Error: Esto parece ser un archivo de texto
gstdecodebin2.c(2265): type_found (): /GstPlayBin2:play/GstURIDecodeBin:uridecod
ebin0/GstDecodeBin2:decodebin20:
decodebin2 cannot decode plain text files
totem-video-thumbnailer couldn't open file 'file:///home/nico/git/libo/COPYING.M
PL'
Reason: GstDecodeBin2: Esto parece ser un archivo de texto.
Shutting down nautilus-gdu extension
nico@nico-VirtualBox:~/git/libo$ cd ..
nico@nico-VirtualBox:~/git$ cd ..
nico@nico-VirtualBox:~$ cd .ssh
nico@nico-VirtualBox:~/.ssh$ cd config
bash: cd: config: No es un directorio
nico@nico-VirtualBox:~/.ssh$ cd config
bash: cd: config: No es un directorio
nico@nico-VirtualBox:~/.ssh$ sudo gedit config
nico@nico-VirtualBox:~/.ssh$ cd ..
nico@nico-VirtualBox:~$ cd git
nico@nico-VirtualBox:~/git$ cd libo
nico@nico-VirtualBox:~/git/libo$ ./logerrit test
Your gerrit setup was successful!
nico@nico-VirtualBox:~/git/libo$
```

7. Prepare su repositorio de git para que pushee a gerrit usando la clave ssh: `git config remote.origin.pushurl ssh://logerrit/core`



The image shows a terminal window titled "Terminal" with the user "nico" in a "nico-VirtualBox" environment. The terminal output is as follows:

```
nico@nico-VirtualBox: ~/$ cd git
nico@nico-VirtualBox:~/git$ git config remote.origin.pushurl ssh://logerrit/core
error: could not lock config file .git/config: No existe el archivo o el directorio
nico@nico-VirtualBox:~/git$ cd libo
nico@nico-VirtualBox:~/git/libo$ git config remote.origin.pushurl ssh://logerrit/core
nico@nico-VirtualBox:~/git/libo$
```

Fuentes:

- <http://es.libreoffice.org/caracteristicas/>
- <https://help.libreoffice.org/Common/Collaboration>. 
- [https://wiki.documentfoundation.org/Git For LibreOffice Developers](https://wiki.documentfoundation.org/Git_For_LibreOffice_Developers)
- <https://wiki.documentfoundation.org/Development/gerrit/setup>
- <https://wiki.documentfoundation.org/Development/BuildingOnLinux>

Gestión de pruebas

¿Cómo hace las pruebas LibreOffice o cómo propone que se podrían hacer ?

LibreOffice deja a libre albedrío de los usuario las distintas pruebas que pueden realizar, esto es, desde bich  por la aplicación **clickando** en iconos y botones hasta probar nuevas características.

LibreOffice organizó en 2012, para la versión Beta 4.0.0, un maratón de Test que duró unos 6 días.

¿Podría proponer u observar qué técnicas de caso de prueba se usan?

Generalmente se basan en la técnica error Guessing basándose en su experiencia, también usan técnicas de cobertura CRUD, **ya que se evalúa cada funcionalidad definiendo s ** **objetivo y los resultados esperados.**

Por otra parte, usan las pruebas Smoke Tests, que deben realizarse obligatoriamente 1 ó 2 días antes de liberar una versión. Las pruebas de humo, en español, son las pruebas preliminares para revelar fallos simples lo suficientemente graves como para rechazar una versión del software prospectivo. Son pruebas simples, como instalar la aplicación, abrir Writer y guardar un fichero simple, etcétera.

También se utilizan tests unitarios (automáticos). Estos pueden ejecutarse después de construir el proyecto haciendo un simple **make check** p  a terminal.

Sin embargo, los tests más importantes son los tests de regresión (manuales). Pueden hacerse tests básicos para solventar bugs o tests exhaustivos para versiones importantes. Éstos se realizan mediante MozTrap, que es una aplicación web que permite probar, generar y mantener casos de prueba y ver los resultados (dependiendo del rol de la persona).

Ejercicio: Probar casos de prueba usando MozTrap

Para poder probar el software, resulta evidente que tenemos que tenerlo construido en nuestra máquina. Sin embargo, el uso de la herramienta web MozTrap para casos de prueba funciona aparte. En este ejercicio nos pondremos en el papel de un usuario ejecutando casos de prueba e informando de resultados:

1. Lo primero que debemos hacer es entrar en la página principal de MozTrap <http://manual-test.libreoffice.org/> . Nos aparecerá una pantalla tal como la siguiente:

2. Acto seguido, debemos crearnos una cuenta o iniciar sesión usando OpenID o Persona.

3. Navegamos a Run Tests:

4. Seleccionamos el producto, versión y conjunto de tests a probar en la nueva pantalla. Después especificamos algunas propiedades de nuestro sistema y hacemos click en el botón verde:

The screenshot shows the MozTrap LibreOffice web application. The header includes the MozTrap logo, the text "MozTrap LibreOffice", and a button "Install as Open Web App". Navigation links "Run Tests", "View Results", and "Manage" are visible. A user greeting "Welcome microbmor [Sign Out, Password]" and a "Help" button are on the right. Below the header, there are tabs for "Test Runs" and "Test Cases". The main content area is divided into three columns: "Products", "Versions", and "Runs". Each column contains a dropdown menu, all of which are currently empty.

5. Nos aparecerán muchos casos de prueba posibles junto con las valoraciones de usuarios, elegimos uno de los casos para probarlo (en principio elegir al azar facilitaría que entre todos los testers se ejecutasen todos los casos)

The screenshot shows the MozTrap LibreOffice web application with the "Runs" tab selected. The "Products" dropdown is set to "LibreOffice". The "Versions" dropdown is set to "4.2.0". The "Runs" dropdown is set to "4.2.0 beta2". Below the dropdowns, there is a table with columns "Start" and "End". The "Start" column shows "2013-12-02" and "2013-12-18", and the "End" column shows "2013-12-15" and "2014-01-05". Below the table, there is a section titled "SET YOUR VALUES TO RUN TESTS IN 4.2.0 BETA2" with three dropdown menus: "Archs" (set to "x86_64"), "Locale" (set to "EN_US"), and "Platforms" (set to "Windows"). A green button labeled "run tests in 4.2.0 beta2!" is located at the bottom right of this section. The footer contains copyright information: "Copyright © 2011-13 Mozilla, 650 Castro St, Suite 300, Mountain View, CA 94041-2021".

6. Leemos las instrucciones y pasos del test y lo ejecutamos. Entonces, pueden darse 3 situaciones:
 - a. El test se ha ejecutado correctamente y sin problemas. Entonces deberemos pinchar en “PASS TEST”.
 - b. El test ha fallado. Debemos identificar en qué paso ha fallado y entonces clicar en el botón “FAIL TEST” correspondiente al paso en concreto y dar un breve informe de lo que ha ocurrido.
 - c. El test es inválido, no tiene sentido o no están claros los pasos a realizar: entonces la acción a realizar es pinchar el botón amarillo, y dar un informe adecuado.

RUN TESTS:

Regressions tests for the second beta of the 4.2.0 version

Enter filter terms Advanced Filtering »

Results	Name	Suites	Order ▲
<input checked="" type="checkbox"/> PENDING <input checked="" type="checkbox"/> Install LibreOffice general p1	Priority 1	1	

SUBMIT YOUR TEST RESULT FOR CASE #28

The test is targeting on testing clean installation of LibreOffice built in all supported platforms. Execute platform native installer (requires root/Administrator privileges) based on your environment:

- 1. On Solaris – pkgadd
 - » all of installation goes well. Normal user can execute the application.
- 2. On Linux – rpm or dpkg
 - » all of installation goes well. Normal user can execute the application.
- 3. On Windows – setup.exe
 - » all of installation goes well. Normal user can execute the application. Make sure the package is signed properly.
- 4. On Mac – open the .dmg file. Drag and drop the executable into the Applications folder.
 - » If an existing version of LibreOffice is present in Applications folder, you should get a cancel/replace dialog in your native language. If LibreOffice did not exist there yet, it should be copied to this folder.

7. Es posible cambiar de idea, esto es, deshacer la valoración para incluir otra. Esto se hace mediante el nuevo icono que aparece en la lista de tests.

RUN TESTS:

Regressions tests for the second beta of the 4.2.0 version

Enter filter terms Advanced Filtering »

Results	Name	Suites	Order ▲
<input checked="" type="checkbox"/> PASSED <input type="button" value="Refresh"/> <input checked="" type="checkbox"/> Install LibreOffice general p1	Priority 1	1	
<input checked="" type="checkbox"/> PENDING <input checked="" type="checkbox"/> Create a new Writer document p1 writer	Priority 1	2	
<input checked="" type="checkbox"/> PENDING <input checked="" type="checkbox"/> Spell and grammar checker in your own language p1 writer	Priority 1	3	
<input checked="" type="checkbox"/> PENDING <input checked="" type="checkbox"/> Hyphenation tool p1 writer	Priority 1	4	
<input checked="" type="checkbox"/> PENDING <input checked="" type="checkbox"/> Uninstall LibreOffice general p1	Priority 1	5	
<input checked="" type="checkbox"/> PENDING <input checked="" type="checkbox"/> First launch of LibreOffice general p1	Priority 1	6	
<input checked="" type="checkbox"/> PENDING <input checked="" type="checkbox"/> Launch components in different ways general p1	Priority 1	7	
<input checked="" type="checkbox"/> PENDING <input checked="" type="checkbox"/> Translation check for installing LibreOffice general p1	Priority 1	8	

Fuentes:

<http://lists.freedesktop.org/archives/libreoffice-ga/2011-August/000020.html>

<https://wiki.documentfoundation.org/QA/Testing/Subsequenttests>

https://wiki.documentfoundation.org/QA/Testing/Regression_Tests

<https://wiki.documentfoundation.org/MozTrap>

https://wiki.documentfoundation.org/MozTrap/User_Guide

Gestión de la construcción

¿Qué herramientas se usan ? ¿ Cómo se usan esas herramientas ?

Construir LibreOffice toma tiempo, mucho tiempo. Exactamente cuánto depende de cómo de rápida sea tu máquina, aunque hay herramientas que puedes usar para acelerar las cosas. 

ccache

Por defecto, ccache se habilitará automáticamente si se encuentra en tu sistema. Para mejores resultados, puedes querer incrementar el tamaño de la caché o habilitar la compresión de la caché; `man ccache` es tu amigo.

distcc / Icecream

Si estás en un entorno donde tienes acceso a múltiples máquinas con ciclos de cpu sobrantes, puedes echarle un ojo a [distcc](#) o [icecream](#), que son herramientas para construcción distribuida.

El soporte para Icecream esta dentro de LibreOffice, es suficiente añadir `--enable-icecream` a `./autogen.sh`, y el proceso de configuración preseleccionará el número de trabajos a usar, e intentará encontrar los icecream gcc wrappers en `/opt/icecream/bin`. Podrías tenerlos en algún otro lugar, por favor usa `--with-gcc-home=/path/to/your/icecream/bin` para sobrescribir eso.

--with-max-jobs / --with-num-cpus

El proceso de construcción puede realizarse ejecutando múltiples tareas en paralelo. El paralelismo total se controla desde el parámetro de autogen `--with-max-jobs` y el paralelismo de cada máquina en subtareas se controla con `--with-num-cpus`. Si tienes suficiente memoria, descubrirás que usando `--with-max-jobs=n --with-num-cpus=n` donde n es tu número de núcleos, o 2 para `--with-max-jobs` si solo tienes 1 núcleo, reduce significativamente el tiempo de construcción.

El comando `--with-max-jobs` está por defecto a 1, y `--with-num-cpus` está por defecto para el número de núcleos/cpus en tu sistema. Nótese que su efecto en red es multiplicativo, ahí podemos acelerar su producto con procesos de trabajo (compilador) de manera simultánea. La construcción del proyecto se realiza de forma paralela a la compilación del mismo. Para ello se utilizan los siguientes comandos:

Autogen.sh

Autogen.sh es un script que genera el makefile y configura la construcción según los parámetros que se le pasen. Tiene una gran cantidad de opciones, aunque pueden consultarse mediante el parámetro --help

Make

Este makefile descarga las dependencias que queden y realiza la construcción del proyecto, compilándolo después.

¿Qué utilizan y cómo gestionan las dependencias?

Windows

Windows gestiona sus dependencias mediante cygwin mediante el siguiente comando:

```
setup.exe -P autoconf,automake,bison,cabextract,flex,gcc-g++,git,gnupg,gperf,libxm2-  
devel,libpng12-devel,make,mintty,openssh,openssl,patch,perl,pkg-  
config,readline,rsync,unzip,vim,wget,zip
```

Esto lanzará la GUI y te pedirá que elijas manualmente las rutas de instalación etc..., pero los paquetes pasados por línea de comandos están preseleccionados lo que significa que sólo debes aceptar la selección.

Debian

Linux las gestiona de una manera más sencilla usando los siguientes comandos:

```
sudo apt-get build-dep openoffice.org && sudo apt-get install git libgnomeui-dev gawk  
junit4
```

Ubuntu Natty

Mediante el siguiente comando:

```
sudo apt-get build-dep libreoffice
```

Ubuntu Maverick/Lucid

Mediante el siguiente comando:

```
sudo apt-get build-dep openoffice.org
```

Ubuntu Fedora/RedHat

Mediante el siguiente comando:

```
sudo yum-builddep libreoffice
```

Mac OS X

Necesita previamente tener instalado XCode con el SDK 10.4 e instalar cada dependencia mediante comandos, para más información consulta las fuentes más abajo en esta misma sección.

¿Cada cuánto tiempo se realiza una construcción del proyecto?

Es variable, depende si la comunidad de desarrolladores dispone de tiempo para reunirse. Una vez reunidos observarán todos aquellos cambios realizados desde la fecha de la última versión en el repositorio maestro y los estudiarán en base a su utilidad en el actual estado de desarrollo, decidiendo cuáles serán incluidos en la próxima versión.

¿Los ejecutables están en control de versiones?

No, no están en el repositorio. Los ejecutables se generan de manera ajena al control de versiones.

Ejercicio: compilar y ejecutar LibreOffice en Linux

El proceso de construcción y compilación de LibreOffice es una operación costosa. Se recomienda disponer de mucho tiempo para realizarla y disponer de un equipo capaz, con al menos 2 GB de RAM, 20 GB de disco duro libres y un procesador del nivel de Intel Core i3.

Para la realización de este ejercicio, se ha hecho uso de una Máquina Virtual donde se ha instalado una versión limpia de Ubuntu v12.04.

Utilizando 2 núcleos de un i7, 2 GB de RAM, 30 GB de disco duro asignados a la MV y a una velocidad de descarga media de 500 KiB/s (con una conexión de unos 15 MB de ancho de banda aproximadamente), se han tardado 5-6 horas en tener LibreOffice preparado.

1. Descargar código fuente

Lo primero a realizar es descargar las dependencias. Pueden utilizarse los siguientes comandos de terminal:

```
sudo apt-get build-dep libreoffice # on Debian/Ubuntu  
sudo zypper si -d libreoffice # on OpenSUSE 11.4 or later  
sudo yum-builddep libreoffice # only on Fedora 15+ & derivatives
```

Una vez estén descargadas, es hora de descargar el código fuente en sí. Puede descargarse clonando el repositorio de git:

```
mkdir git
cd git
git clone git://anongit.freedesktop.org/libreoffice/core libreoffice
```

Esto clonaría el repositorio en una nueva carpeta llamada libreoffice.

El repositorio de LibreOffice es muy lento. Con el objeto de este trabajo hemos preparado un nuevo repositorio en BitBucket. El repositorio está localizado en la siguiente dirección: <https://bitbucket.org/jesrivcap/libreoffice.git>

2. Compilar el código fuente

En la carpeta donde está el repositorio de git, ejecutar

```
cd libreoffice
./autogen.sh
```

Este comando genera el makefile para compilar LibreOffice

```
make
```

Ejecutamos el makefile anteriormente creado, construyendo y compilando el proyecto.

3. Montar el entorno de desarrollo

Para arrancar LibreOffice es necesario lanzar el siguiente comando:

```
make debugrun
```

En ese momento el programa solicitará un comando adicional (se puede reconocer porque la línea de comando empieza por (gdb)). Aquí seleccionamos la aplicación concreta de LibreOffice que queremos ejecutar. Por ejemplo, podemos arrancar Writer:

```
run --writer
```

Nótese que se incluyen dos '-'. En el caso de utilizar uno tratará de arrancar una versión de Writer vieja, la cual dará un error.

Si todo ha ido bien, Writer entrará en ejecución, mostrando su interfaz gráfica.

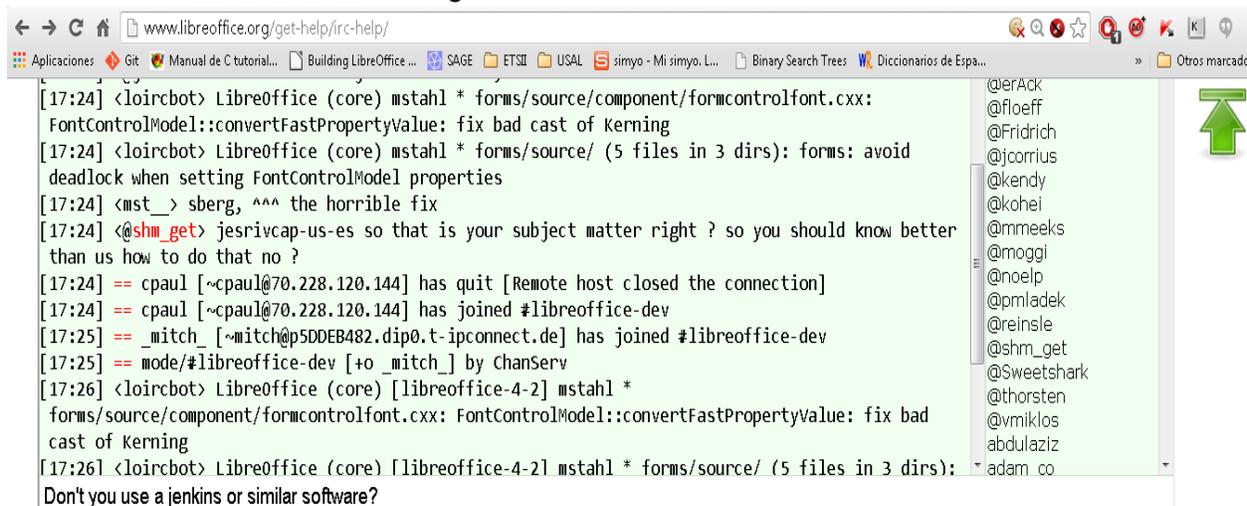
Fuentes:

<https://wiki.documentfoundation.org/Development/BuildingOnLinux>
https://wiki.documentfoundation.org/Development/Install_Mac_OS_10.6.4_Dependencies/es#Construyen do las Dependencias para Mac OS X 10.6.4 .2864bits.29
https://wiki.documentfoundation.org/Development/Native_Build/es#Primera_Construcci.C3.B3n
https://wiki.documentfoundation.org/Development/Windows_Build_Dependencies/es
https://wiki.documentfoundation.org/Development/Native_Build/es
https://wiki.documentfoundation.org/Development/Linux_Build_Dependencies/es
https://wiki.documentfoundation.org/Development/Install_Mac_OS_10.6.4_Dependencies/es

Integración/despliegue continuos

¿Qué tipo de Integración utilizan: Big Bang o Incremental?

Hemos hecho una investigación donde se ha preguntado a algunos desarrolladores si desarrollaban sobre una integración continua, obteniendo como respuesta que no automatizan su integración. LibreOffice posee demasiado código fuente y el tiempo de compilación ronda por las 4-5 horas, así que proponemos el uso de una herramienta para automatizar esto y que al momento de trabajar se pueda coger directamente el código compilado y ahorrar tiempo ello. A continuación mostramos las imágenes de la conversación:



The screenshot shows an IRC chat window with the following text:

```
[17:24] <loircbot> LibreOffice (core) mstahl * forms/source/component/formcontrolfont.cxx: FontControlModel::convertFastPropertyValue: fix bad cast of Kerning
[17:24] <loircbot> LibreOffice (core) mstahl * forms/source/ (5 files in 3 dirs): forms: avoid deadlock when setting FontControlModel properties
[17:24] <mst_> sberg, ^^^ the horrible fix
[17:24] <@shm_get> jesrivcap-us-es so that is your subject matter right ? so you should know better than us how to do that no ?
[17:24] == cpaul [~cpaul@70.228.120.144] has quit [Remote host closed the connection]
[17:24] == cpaul [~cpaul@70.228.120.144] has joined #libreoffice-dev
[17:25] == _mitch_ [~mitch@p5DDEB482.dip0.t-ipconnect.de] has joined #libreoffice-dev
[17:25] == mode/#libreoffice-dev [+o _mitch_] by Chanserv
[17:26] <loircbot> LibreOffice (core) [libreoffice-4-2] mstahl * forms/source/component/formcontrolfont.cxx: FontControlModel::convertFastPropertyValue: fix bad cast of Kerning
[17:26] <loircbot> LibreOffice (core) [libreoffice-4-2] mstahl * forms/source/ (5 files in 3 dirs): Don't you use a jenkins or similar software?
```

Other Channels

There are other IRC channels reserved for LibreOffice. If you want to connect to them just go to the "Status" and type "/join channelname" replacing channelname with the name of the channel you need. Some of these channels are:

- ▼ #libreoffice-dev: Development Channel
- ▼ #tdf-infra: Related to all TDF-hosted websites and infrastructures

Como herramienta para integración continua, en principio proponemos el uso de Jenkins, sin embargo, durante las pruebas realizadas para estudiar su utilización nos hemos encontrado con obstáculos insalvables que han impedido su uso. Entre ellos:

- Timeout de 10 minutos en el plugin Git de Jenkins, no configurable, que impide hacer un checkout de LibreOffice desde un repositorio externo (ya sea el propio de la comunidad, o cualquier otro que se cree con objeto de este trabajo) o incluso en local, debido al gran tamaño. Como fuente para este problema consúltese la siguiente dirección: <https://issues.jenkins-ci.org/browse/JENKINS-20387>
- Incapacidad de LibreOffice de realizar ejecutar el makefile desde Jenkins correctamente, ya que, de algún modo, no es capaz de realizar peticiones web para recoger las dependencias necesarias.
- No es posible realizar la integración con ningún servidor Git y Jenkins debido a que el Jenkins que utilizamos es local. Hacer esto posible requeriría modificar la configuración del router local para redireccionar el tráfico. Esta posibilidad se observa con el servicio de Bitbucket, pero no disponemos de infraestructura suficiente como para enlazar el sitio que ofrece el servicio a nuestro equipo a través de nuestro router y un servicio de dirección estática.
- Jenkins está muy orientado a proyectos escritos en Java y sus plataformas. Esto hace difícil adaptarlo a este proyecto, escrito en C++.
- Hemos encontrado dificultades para usar los plugins de Jenkins, creemos que por desconocimiento de cómo configurarlos (advertimos fallas de documentación en los mismos) o por falta de soporte específico.

Por tanto, consideramos que no es la plataforma de integración más recomendada para este proyecto. El tiempo de realización de este trabajo impide realizar nuevas búsquedas, sin embargo, no descartamos que otras herramientas puedan funcionar adecuadamente para su uso con LibreOffice.

Gestión de entregables

Los entregables de Libreoffice son básicamente los ficheros binarios que distribuye con todos los componentes, los cuales se colocan en la página web de LibreOffice para su acceso público.

Gestión del despliegue

LibreOffice, como aplicación de escritorio que es, ofrece binarios para cada posible plataforma que los usuarios pueden utilizar para instalarlo en sus máquinas.

Gestión de incidencias y depuración

Si trabajamos con Ubuntu

Para reportar un problema específico a Ubuntu, primero verifique si no se encuentra en la lista actualizada, luego **llene éste formulario y especifique libreoffice** bajo **In what package did you find this bug?**. Recuerde agregar el "tag" **lo33** a su reporte.

También es útil reportar el mismo problema en FreeDesktop.org como se indica **anteriormente** pues dichos reportes "upstream" pueden ser verificados (y posiblemente probados y arreglados) en otras versiones de LibreOffice.

¿Quién y cómo se priorizan las incidencias?

Las incidencias las resuelven los desarrolladores como a ellos mejor les parezca, tomando como criterios la prioridad del fallo y su propia experiencia.

¿Cómo se encaja esto con la gestión del código?

Cada *minor* de una versión incorpora las correcciones que se detectan en las incidencias, pero solamente en las versiones que incrementan el *major*, se incluyen nuevas funcionalidades. **Cada una de estas versiones se organiza en un fork en git, generándose uno por cada versión que se obtiene.**

¿Cuándo y cómo vamos a liberar versiones?

No se tiene una fecha en concreto para realizar una liberación de una versión, cuando una versión es estable siempre tiene que tener el formato: x.y. Los compilados que contienen cambios de la X son aquellos que incluyen nuevas funcionalidades, y los que incluyen cambios en la Y, son los que incluyen revisiones de fallos, pero no añaden funcionalidad.

¿Cómo vamos a gestionar los cambios?

Hasta el momento, solamente 60 personas deciden los cambios, siendo este un sistema muy centralizado. De este modo proponemos un sistema más colaborativo en cuanto a la toma de decisiones, de forma que los usuarios puedan aportar y valorar ideas y se aprueben por votación, decidiéndose a su vez la prioridad de las mismas.

¿Cómo se informa de una incidencia?

LibreOffice utiliza Bugzilla para gestionar los informes de incidencias.

¿Cómo se informa de un bug? ¿hay plantillas?

Mediante correo electrónico o por la herramienta Bugzilla antes mencionada, lamentablemente no usan plantillas predefinidas al realizar un reporte, esto ayudaría mucho para poder sintetizar y poder hacer el seguimiento fácilmente.

¿Cómo se gestiona el reporte de bug?

Para el reporte de bugs LibreOffice utiliza Bugzilla y listas de correo. LibreOffice ofrece una interfaz en su web donde debes seguir los siguientes pasos:

- *Sign in* : nos pide las credenciales de la cuenta de Bugzilla.
- *Component* : tipo del bug.
- *Details* : tipo del componente. (subtipo del bug)
- *Description* : en qué consiste el bug.
- *Submit* : enviamos el reporte para que se añada a la base de datos de Bugzilla.

Si Bugzilla no te gusta, puedes utilizar las listas de correo que ofrece LibreOffice para reporte de bug users@global.libreoffice.org o en todo caso el foro LibreOfficeForum.org.

Ejercicio: Enviar un bug en LibreOffice

Entramos en [Bug Submission Assistant](#), necesitaremos habernos hecho una cuenta de Bugzilla. Para crearla entra aquí <https://bugs.freedesktop.org/createaccount.cgi>.

1: Iniciar sesión

Utiliza la cuenta de Bugzilla.

2: Componente

En el combobox de Componente, elegimos el componente que queramos reportar. Si no estás seguro de que componente se adecúa a tu problema elige el componente LibreOffice y luego los usuarios con mayor experiencia cambiarán esta opción a la adecuada.

The screenshot shows the LibreOffice Bug Submission Assistant web interface. At the top, there is a navigation bar with links: Home, Download, Features, Get Help, Get Involved, Developers, International Sites, About Us, Security, Nabble, Mailing Lists, Installation, Documentation, Accessibility, Bug, Ask LibreOffice, Online Help, and IRC. Below the navigation bar, there is a green banner with the text: "Thank you for using LibreOffice and for helping us improve our software. This assistant will lead you step-by-step through the bug-reporting process. Please submit a separate bug for each issue! For our full-featured bugtracker go to bugs.libreoffice.org".

The main content area is divided into two columns. On the left, there is a vertical sidebar with five steps: 1 Sign in, 2 Component, 3 Details, 4 Description, and 5 Submit. Step 2, "Component", is currently selected. On the right, there is a form with a "Component:" label and a dropdown menu. The dropdown menu is open, showing a list of components: (Choose one), Drawing, Extensions, Formula Editor, Installation, Libreoffice (highlighted), Linguistic, Localization, and Presentation. To the right of the dropdown menu, there is a text box with the text: "Component Libreoffice", "If you can't find any other appropriate component: [MinGW](#) for problems with the [MinGW builds](#) from [Tinderbox @7 Master](#) and [daily Branch](#), [\[1\]](#) for all other issues.", and a "Read more ... >" link.

3: Detalles

Ahora seleccionaremos un “Sub componente” en la sección correspondiente.

Si no sabes el subcomponente apropiado ve a [Componentes](#). En esa página, clickea el componente apropiado.

Components

- [Android Impress Remote](#), the LibreOffice Impress Remote app for Android
- [BASIC](#), programming using the Basic language
- [Chart](#), data charts and plots
- [ci-infra](#), internal feature request for the tinderbox/gerrit infrastructure
- [contrib](#), to be used if you want to contribute tools, templates, or something similar to LibreOffice
- [Database](#), the database application LibreOffice Base
- [deletionrequest](#), for bugs which have been created only for testing Bugzilla or the Bug Submission Assistant
- [Documentation](#), for problems with LibreOffice [user guides](#), localized Help packages, and [LibreOffice Help](#)
- [Drawing](#), the drawing application LibreOffice Draw
- [Extensions](#), for some specific problems with extensions; but not for the extensions website, which uses the component WWW
- [filters and storage](#), import and export code, and I/O infrastructure shared between applications; mostly interesting for **developers**
- [Formula Editor](#), the formula editor, which is probably LibreOffice Math
- [framework](#), the application [framework](#), ie. code shared between applications not part of the graphical toolkit pieces
- [graphics stack](#), the VCL toolkit, drawing layer, and all non-print specific rendering functionality, interesting for **developers**
- [Installation](#), for problems with installing LibreOffice
- [Libreoffice](#), to be used if you can't find any other appropriate component
- [Linguistic](#), for problems with spell check, right-to-left, hyphenation, or the thesaurus, not specific to one language
- [Localization](#), for all problems that are related to a particular language version of LibreOffice. Alternatively you can use one of the other components and set the I10n subcomponent.
- [Presentation](#), the presentation application LibreOffice Impress
- [Printing and PDF export](#), for general problems with Printing or PDF export that are not limited to a single application (like WRITER or CALC)
- [sdk](#), the UNO runtime environment, UNO API and SDK, interesting for **developers**
- [Spreadsheet](#), the spreadsheet application LibreOffice Calc
- [UI](#), for problems concerning User Interface. You should also use this component if your problem affects more than 1 Application or if you can't decide what the correct component might be.
- [ux-advise](#), to be used only by QA staff or developers
- [Writer](#), the word processor LibreOffice Writer
- [WWW](#), for problems concerning website (bad links, for example), Bug-Tracker, software downloads, not responding or incomplete mirrors, mailing lists, ...

Lee las descripciones de todos los subcomponentes de la página de ese componente.

Sub-components

Some Sub-components are specific to a particular Component. In that case, they'll be listed under their parent Component. For example [ASKLIBREOFFICE](#) is listed under [WWW](#).
All other sub-components are described here!

How to use subcomponents

On [BugReport Details](#) lots of *subcomponents* are listed with links to [Bugzilla](#) Search and [Bugzilla](#) bug submission. In the following you find hints for what kind of problems what Subcomponents should be used in the [Bugzilla](#) Subject line. In the following listing a click on the Sub Component heading will start a Bugzilla query for that sub component.

CONFIGURATION

Use for instance, if the configuration of LibreOffice applications (*WRITER*, *DRAW*, *CALC*, ...) is broken or if the *Installation* associates file extensions wrongly.

FILEOPEN

For all problems related to opening documents, as well as for problems when opening non-ODF file types like MS-Office, XHTML or XML, Picture formats like PNG, SVG, WMF, ... Please generally mention the document type and if known and appropriate the Version of the document or of the software with what it has been created in parenthesis. [Examples](#)

- [CRASH](#) when FILEOPEN particular XLSX (MSO2007)
- [FILEOPEN](#) particular .CDR (X6) shows ...

FILESAVE

For all problems related to saving documents, as well as for problems when exporting to the file types XHTML or XML and Picture formats like PNG, SVG, WMF, ...

MAILMERGE

For problems with the LibreOffice Mailmerge function from menu 'Tools -> Mail Merge Wizard' or with the Form Letter function in menu 'File -> Print'.

MinGW

[MinGW](#) (Minimalist GNU for Windows) is a free toolchain based on GCC (the GNU Compiler Collection). So far we are using the Microsoft VisualStudio C++ compiler to provide the Windows builds, but the intent is to switch to MinGW to produce the Windows binaries. At the current stage, the MinGW builds start, and are somewhat usable, but contain bugs, and need testing.

Bugs should be marked with `mingw` only when they appear in the builds from [here](#), but not in other Windows, or Linux builds.

Please note that MinGW builds are binary incompatible with the [MSVC](#), which means that extensions written in C++ built with MSVC cannot work in MinGW builds.

PDF

For PDF export problems only visible in one component like Drawing, Presentation, Spreadsheet or Writer. In DRAW together with [FILEOPEN](#) for PDF import problems.

Si no ves el subcomponente adecuado, clickea en Extended Help, y lee las descripciones de los subcomponentes en esa página. Una vez que sepas cuál es el subcomponente que buscas, selecciónalo en el Bug Submission Assistant. Si no puedes descubrir qué subcomponente tienes que usar, selecciona “(All other problems)” en el Bug Submission Assistant.

Escoge la versión de la aplicación donde apareció el bug. Para comprobarlo en LibreOffice, selecciona **Help > About > LibreOffice**

En “Operating system” o “OS”, escoge tu sistema operativo.

Si hay una sección de “Hardware”, rellénala.

Si hay una sección “Severity”, ignóra esta sección sino tienes experiencia. Seleccionando “Blocker” no hará que el bug se solucióne más rapido. Si quieres saber las definiciones de los items en la sección Severity, échale un ojo a su carta.

También, puedes ignorar la sección “latest known-working version”.

The screenshot shows the LibreOffice Bug Submission Assistant interface. At the top is a green navigation bar with links: Home, Download, Features, Get Help, Get Involved, Developers, International Sites, About Us, Security, Nabble, Mailing Lists, Installation, Documentation, Accessibility, Bug, Ask LibreOffice, Online Help, IRC. Below this is a 'Bug' section header. A green banner reads: 'Bug Submission Assistant. Thank you for using LibreOffice and for helping us improve our software. This assistant will lead you step-by-step through the bug-reporting process. Please submit a separate bug for each issue! For our full-featured bugtracker go to bugs.libreoffice.org'. On the left is a vertical sidebar with five steps: 1 Sign in, 2 Component, 3 Details, 4 Description (highlighted), 5 Submit. The main area shows a 'Component' dropdown set to 'Libreoffice'. Below it is a grid of icons representing various LibreOffice components like Writer, Calc, Impress, etc. To the right of the grid is a text box titled 'Component Libreoffice' with instructions: 'If you can't find any other appropriate component: [MinGW](#) for problems with the *MinGW builds* from [Tinderbox @7 Master](#) and [daily Branch](#), [\[1\]](#) for all other issues.' Below the grid are four dropdown menus: 'Sub component:' (Choose one), 'Version the bug appeared:' (4.1.3.2 release), 'Operating system:' (Windows 7), and 'Latest known-working version:' (NONE). A 'Read more ... >' link is at the bottom right of the component info box.

4: Descripción

En la sección “Subject” :

- No incluya información ya conocida de los campos anteriores.
- Si no estás usando el Bug Submission Assistant: Incluye los nombres de los subcomponentes de los [Componentes](#).(Si estás usando este asistente, la sección “Subcomponente” es el lugar para incluir los subcomponentes).
 - Pon mayúsculas en los subcomponentes.
 - El subcomponente puede ser confundido con partes de una palabra(por ejemplo, UI es parte de la palabra **quit**), por lo que te recomendamos que rodees los subcomponentes con corchetes.
 - Usa al menos 2 subcomponentes.
- Resume el problema con la máxima precisión.
 - Un resumen incorrecto: “El fichero está roto”.
 - Un resumen más adecuado: El menú **File > Save as** no está disponible (está en gris).
- Evita las formas cortas como “doesn’t” or “isn’t” (para el inglés) para relajar los strings en el Sumario; en vez de eso, usa la forma completa como “does not” or “is not”.
- Si el problema escrito en el reporte es que LibreOffice se crasha o para de responder(se cuelga), agrega la palabra CRASH al sumario, así esos bugs pueden resolverse más fácil.

En la sección “Long Description” o “Description”, hay que dar una descripción de los hechos más larga.

En esta misma sección, hay que dar una lista de pasos para reproducir el bug.

- Usar una lista enumerada.
- Declarar el método exacto que hizo que ocurriera el bug. Por ejemplo, en vez de escribir “Abrir documento”, escribir “En nuevo documento vacío LibO Spreadsheet”, usa menu File > Abrir (diálogo LibO) > archivo tipo “Documento de texto” > seleccionar el documento de muestra adjunto > doble click”.

Si estás usando una pre-construcción de LibreOffice bajo Linux, declara la versión exacta de los paquetes de LibreOffice de tu sistema de gestión de paquetes.

- Incluir información sobre donde lo instalaste y la localización (UI idioma, documentos de idioma) podría ser útil.
- Incluir si se está usando un LibreOffice de 32 bits en una sistema de 64 bits.
- Incluir si el paquete es una construcción no oficial de LibreOffice.

Escribe el comportamiento esperado y el comportamiento actual.

Puedes incluir un adjunto, como una screenshot o un documento de ejemplo.

- Si creas unas screenshots, cambia el idioma a inglés antes de hacerla. Para ello dirígete a **Tools ▶ Options ▶ Language Settings ▶ Languages**.
- Puedes hacer screenshots más útiles añadiendo comentarios y marcando las áreas relevantes con Draw de LibreOffice.
- Si quieres juntar mas de una screenshot, deberías recopilar todas ellas en un documento y adjuntarlo como PDF. Por favor añadir un comentario breve por cada screenshot para decir lo que quieres demostrar con ella.
- Si quieres unir más de un documento, deberías crear un archivo .zip conteniendo todos los documentos y un adjuntar eso como .zip.
- Si tu adjunto es demasiado grande como para almacenarlo en Bugzilla (mayor de 1 MB) puedes usar la [Página Experimental de Subida](#).

5: Envío

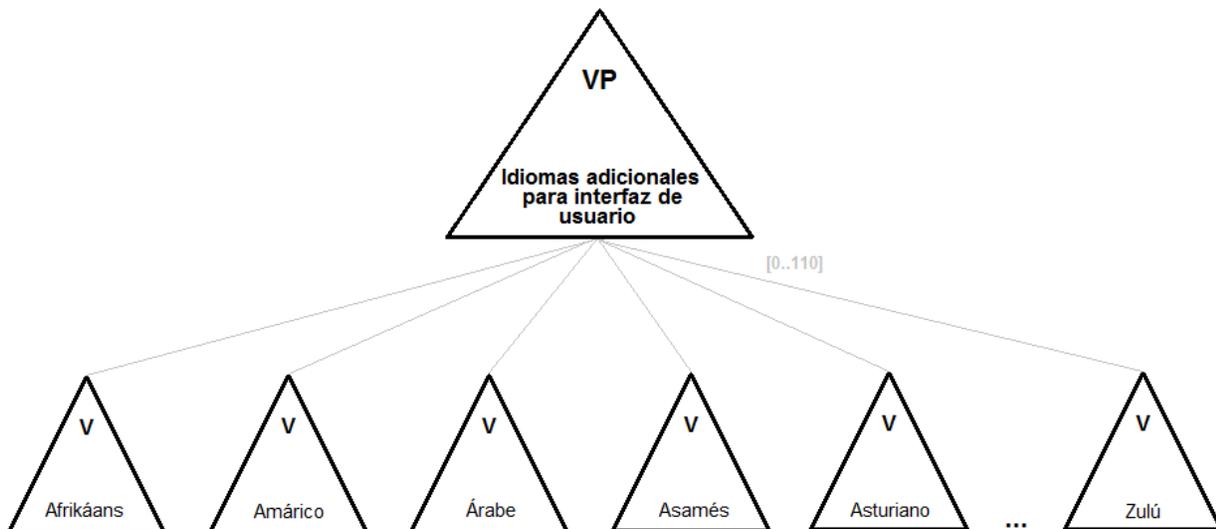
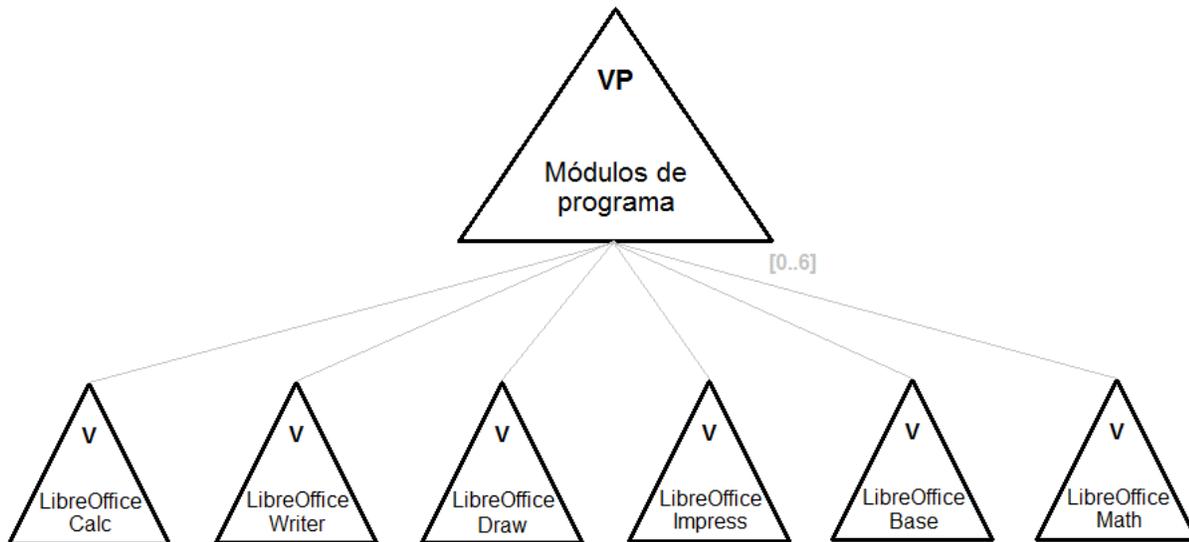
Clicka en “Submit” y tu reporte será agregado a la base de datos de Bugzilla.

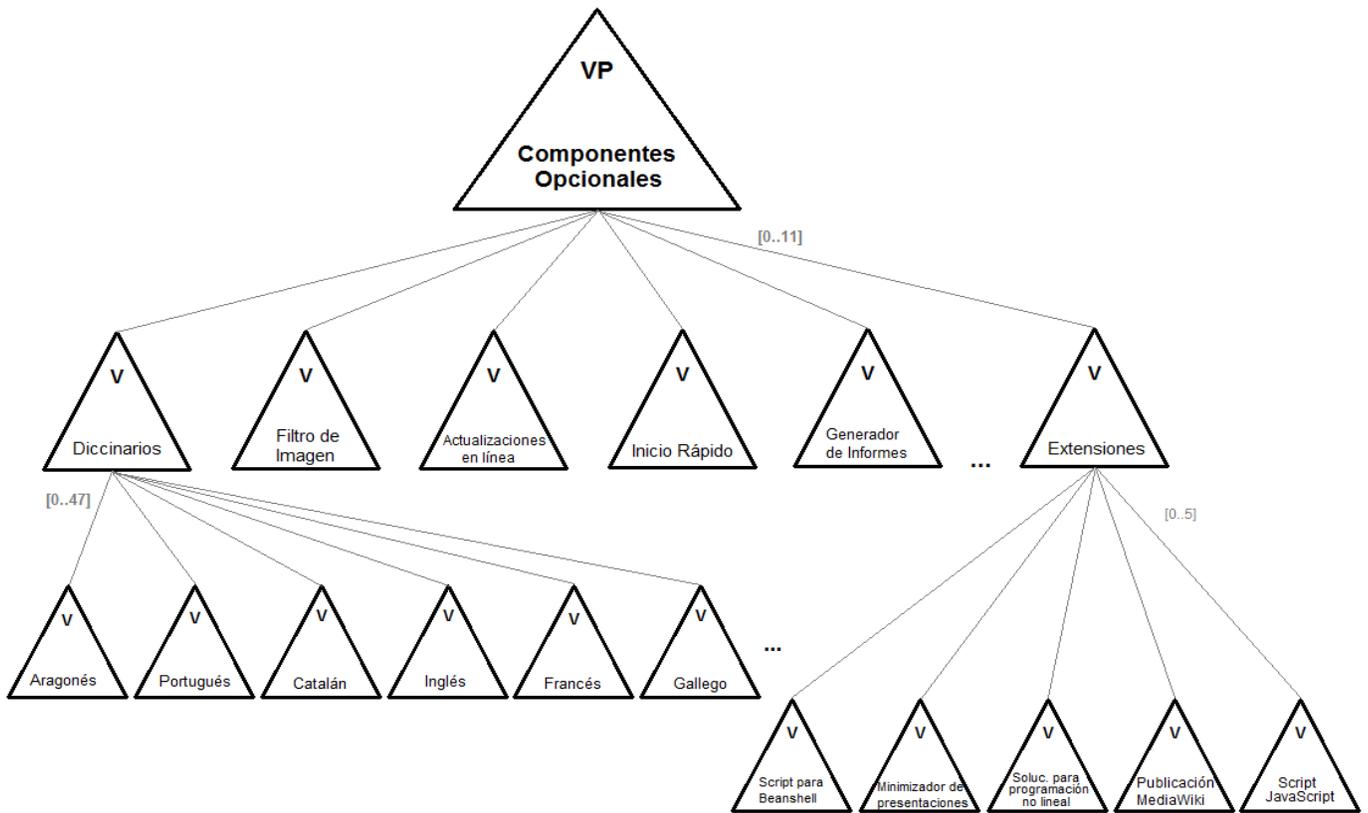
Si Bugzilla te parece desalentador

Si el sistema del bug tracking de Bugzilla parece desalentador o muy difícil de comprender, deberías postear tu problema aquí: users@global.libreoffice.org [user support list](#) o LibreOfficeForum.org.

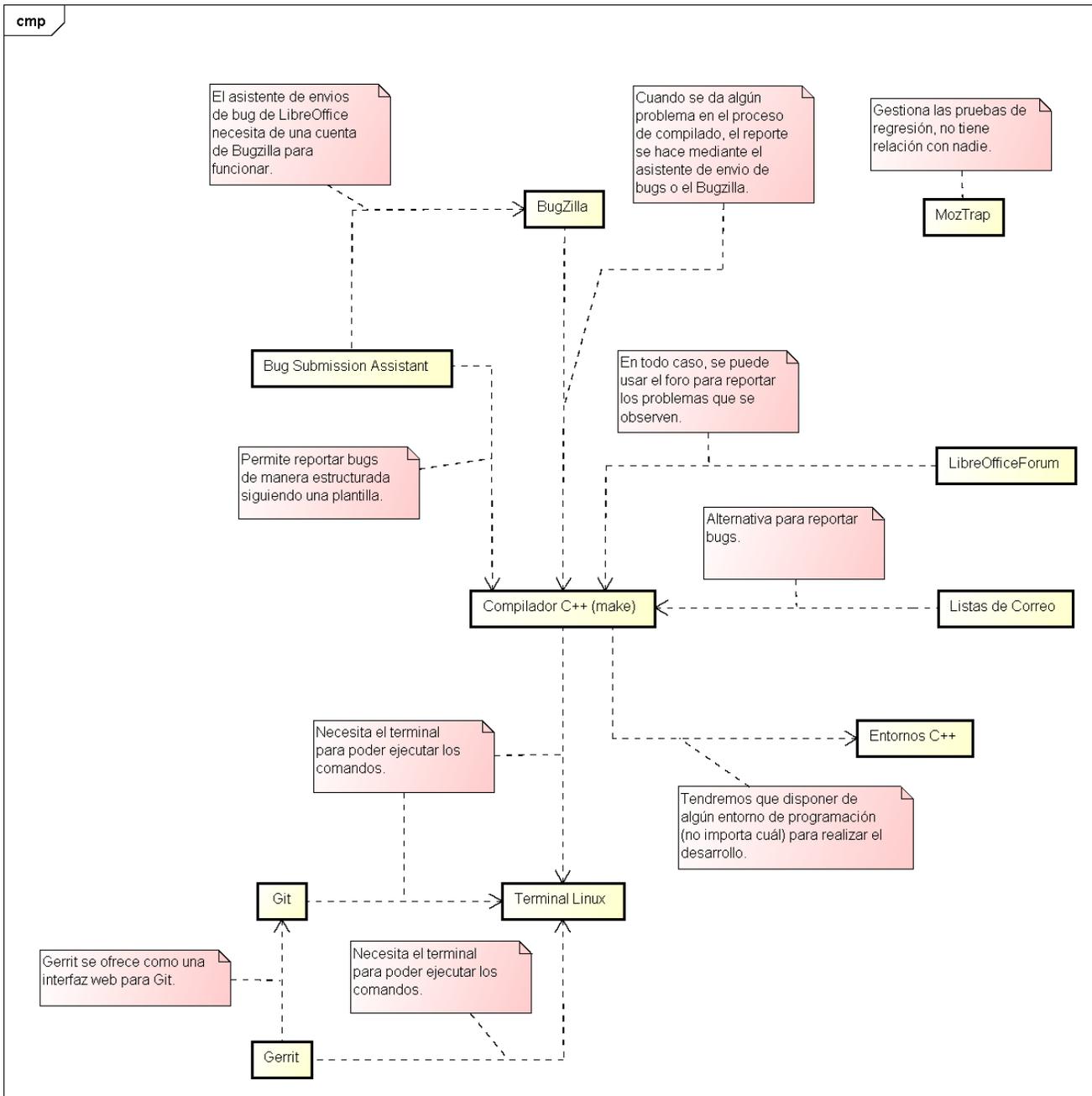
Gestión de la variabilidad

Tomamos como ejercicio la instalación de LibreOffice, la que nos ofrece los distintos productos de variabilidad (VP): Módulos de Programa, Idiomas Adicionales para Interfaz de Usuario y Componentes Opcionales, que éstos a su vez tienen distintos niveles, de los cuales los representamos mediante los siguientes diagramas:





Mapa de herramientas



Conclusiones

LibreOffice es un proyecto colaborativo en el que participa mucha gente, pero tiene un proceso centralizado en el que pocas personas gestionan los cambios. Mucha gente realiza cambios pero sólo unos pocos que se seleccionan llegan a aprobarse e integrarse. Sus procesos tanto de pruebas como de integración son mayoritariamente manuales, y estos procesos están bien asentados debido a la gran cantidad de contribuyentes. Añadiendo esto a la inherente dificultad, siendo un proyecto de gran envergadura, de implementar estos procesos, no esperamos que vayan a cambiar.

Su despliegue y entrega son simples: colocar ficheros de instalación con todo lo necesario en la página web de LibreOffice. Esto es debido a que es fundamentalmente una aplicación de escritorio.

Un detalle interesante es la carencia de herramientas estandarizadas en la mayoría de áreas, principalmente MozTrap y Git son las de uso obligatorio. No se considera un entorno de desarrollo estándar, ni tampoco la interfaz de Git, e incluso no se llega a usar una única herramienta para el reporte de incidencias, debido al carácter libre y fuertemente colaborativo de la aplicación.

Una gran acierto al desarrollar la aplicación es el sistema  de plugins. De este modo puede beneficiarse enormemente de los aportes de la comunidad, existiendo infinidad de ellos que cubren muchas funcionalidades sin explorar, idiomas nuevos, y más.

LibreOffice tiene una gestión de la configuración asentada y eficaz, pero creemos que mejorar la organización y la toma de decisiones haría mucho bien por su desarrollo. Sin embargo, hemos de tener en cuenta que para un proyecto de software libre, **es una auténtica  odisea** solventar estos aspectos.