





## Máster Software Libre en Galicia, ano II



# Máster Software Libre en Galicia, ano II

Compendio de resultados  
da edición 2008-2009



Título orixinal:

*Máster Software Libre en Galicia, ano II.  
Compendio de resultados da edición 2008-2009*

*Obra colectiva. Copyright 2008, 2009, 2010, 2011 e 2012 de cada un dos artigos incluídos no libro para o autor orixinal de cada un dos mesmos.*

*Obra liberada baixo licenza CreativeCommons Attribution-ShareAlike (3.0)*

D.L. C 2375-2012

ISBN: 978-84-940033-9-4

Edita: Edizer, GHANDALF

<http://www.ghandalf.org>

Dirección e coordinación da edición: Roberto Vieito Raña

Deseño de capa: Ana García Rey

Diagramado, impreso e encuadernado en Galicia por Sacauntos  
Cooperativa Gráfica

[www.sacauntos.com](http://www.sacauntos.com)

Diagramado con Software Libre

# Índice de contidos

|                                                            |    |
|------------------------------------------------------------|----|
| Unhas verbas da Asociación GHANDALF, antes de comezar..... | 12 |
| Prólogo, obra de AGASOL.....                               | 14 |
| Solucións FLOSS para Videoconferencias.....                | 16 |
| 1. Requisitos do sistema.....                              | 18 |
| 2. Pesquisa de solucións.....                              | 19 |
| 3. Solucións.....                                          | 19 |
| 3.1 Access Grid.....                                       | 19 |
| 3.1.1 Software.....                                        | 20 |
| 3.1.2 Hardware.....                                        | 21 |
| 3.2 Appconference.....                                     | 21 |
| 3.2.1 Software.....                                        | 22 |
| 3.2.2 Hardware.....                                        | 22 |
| 3.3 OpenMeetings.....                                      | 23 |
| 3.3.1 Software.....                                        | 24 |
| 3.3.2 Hardware.....                                        | 24 |
| 3.4 Dimdim.....                                            | 25 |
| 3.4.1 Software.....                                        | 25 |
| 4. Avaliación das solucións.....                           | 25 |
| 4.1 Avaliación do produto.....                             | 26 |
| 4.2 Avaliación da comunidade.....                          | 27 |
| 5. Documentación.....                                      | 28 |
| 6. Bibliografía.....                                       | 29 |
| Dinámicas das comunidades de software libre: Python.....   | 30 |
| 1 Introducción a Python.....                               | 32 |
| 1.1 Unha breve introdución a Python.....                   | 32 |
| 1.2 As mellores funcionalidades de Python.....             | 33 |
| 1.3 Unha ollada á historia de Python.....                  | 34 |
| 1.4 Quen usa Python?.....                                  | 36 |

|                                                                   |    |
|-------------------------------------------------------------------|----|
| 2. A Python Software Foundation.....                              | 37 |
| 2.1 Unha ollada á Python Software Foundation.....                 | 37 |
| 2.2 Finalidade da Python Software Foundation.....                 | 37 |
| 2.3 Financiamento e mantemento da Python Software Foundation..... | 38 |
| 2.4 Organización da Python Software Foundation.....               | 39 |
| 2.5 A licenza da Python Software Foundation.....                  | 40 |
| 2.5.1 Contidos da Python Software Foundation License.....         | 42 |
| 3. A comunidade Python.....                                       | 44 |
| 3.1 Comunicación entre os membros da comunidade Python.....       | 44 |
| 3.2 PyCon: a conferencia sobre Python.....                        | 45 |
| 4. Python para desenvolvedores.....                               | 46 |
| 5. Análise de Python.....                                         | 47 |
| 5.1 Datos xerais sobre Python.....                                | 47 |
| 5.2 Historia do desenvolvemento.....                              | 47 |
| 5.3 Linguaxes de programación.....                                | 47 |
| 6. Estatísticas das listas de distribución.....                   | 48 |
| 6.1 A lista de distribución Python announce.....                  | 48 |
| 6.2 A lista de distribución Python dev.....                       | 49 |
| 7. Conclusións sobre Python.....                                  | 50 |
| 8. Referencias.....                                               | 51 |

## Participar e traballar con software libre.....62

|                                                |    |
|------------------------------------------------|----|
| Parte I: Comunidade.....                       | 64 |
| 1.1 Primeiras contribucións.....               | 64 |
| 1.1.1 TexMaker.....                            | 64 |
| 1.1.2 Maemo.....                               | 64 |
| 1.2 Gnome.....                                 | 65 |
| 1.2.1 Primeiras contribucións.....             | 65 |
| 1.2.2 Gnome3.....                              | 65 |
| 1.2.2.1 Proxecto Ridley: obxectivo GTK+ 3..... | 65 |
| 1.2.2.2 GnomeGoals.....                        | 67 |
| 1.2.3 GTK+ 3.....                              | 67 |
| 1.2.4 Bugsquad.....                            | 68 |
| 1.2.5 Integración na comunidade Gnome.....     | 68 |
| 1.2.6 Algunhas estatísticas.....               | 69 |
| Parte II: Modest.....                          | 71 |
| 2.1 Introducción.....                          | 71 |



|                                                    |    |
|----------------------------------------------------|----|
| 2.2 Maemo.....                                     | 71 |
| 2.2.1 Historia.....                                | 72 |
| 2.3 Moblin.....                                    | 74 |
| 2.3.1 Pasado e futuro.....                         | 75 |
| 2.3.2 Ambiente de desenvolvemento de Moblin.....   | 76 |
| 2.4 Modest.....                                    | 79 |
| 2.4.1 Arquitectura.....                            | 79 |
| 2.4.1.1 Tinymail.....                              | 79 |
| 2.4.2 Labor realizado.....                         | 80 |
| 2.4.2.1 Tornar algunhas bibliotecas optativas..... | 81 |
| 2.4.2.2 Depuración de bibliotecas.....             | 81 |
| 2.4.2.3 Problemas.....                             | 82 |
| 2.4.2.4 Futuro.....                                | 83 |
| 3. Bibliografía.....                               | 84 |

## Rich Internet Applications (RIA): comparativa de frameworks en Software Libre.....85

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| 1. Introducción.....                                                      | 87  |
| 2. Tecnoloxías RIA.....                                                   | 88  |
| 2.1 Tecnoloxías baseadas en plugins ou sandboxes.....                     | 89  |
| 2.2 Baseadas nos estándares soportados nativamente polos navegadores..... | 91  |
| 3. GWT.....                                                               | 99  |
| 3.1 Introducción.....                                                     | 99  |
| 3.2 Características.....                                                  | 99  |
| 3.3 Desenvolvendo unha aplicación con GWT.....                            | 102 |
| 3.3.1 Instalación de Google Web Toolkit.....                              | 102 |
| 3.3.2 Creación dun proxecto GWT.....                                      | 104 |
| 3.3.3 Deseñando a aplicación.....                                         | 107 |
| 3.3.4 Construindo a interface de usuario.....                             | 108 |
| 3.3.5 Implementando código no lado do Cliente.....                        | 116 |
| 3.3.6 Desenvolvendo código no lado do servidor.....                       | 120 |
| 3.3.7 Compilando unha aplicación GWT.....                                 | 128 |
| 3.4 Comunidade GWT.....                                                   | 130 |
| 4. Referencias .....                                                      | 133 |

## Análise das aplicacións ERP, CRM e E-COMMERCE baseadas en SFA para PEMES.....134

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| 1. Contexto.....                                                         | 135 |
| 2. Análise das características xerais e funcionais das aplicacións....   | 139 |
| 3. Análise das comunidades que manteñen os proxectos.....                | 153 |
| 3.1 Metodoloxías propostas para levar a cabo o estudo.....               | 153 |
| 3.2 OpenBRR e QSOS.....                                                  | 153 |
| 3.3 QualOSS e SQO-OSS.....                                               | 154 |
| 3.4 Conclusión.....                                                      | 154 |
| 3.5 Análise mediante a metodoloxía QUALOSS.....                          | 155 |
| 3.5.1 Definición previa do obxecto de estudo .....                       | 155 |
| 3.5.2 Ferramentas.....                                                   | 160 |
| 3.5.3 Recursos externos.....                                             | 160 |
| 4. Problemas detectados con MLStats e Bicho.....                         | 161 |
| 4.1 MLStats.....                                                         | 161 |
| 4.2 Bicho.....                                                           | 162 |
| 4.3 Erros.....                                                           | 162 |
| 5. Preguntas e ferramenta empregada para extraer os datos.....           | 165 |
| 6. Problemas detectados con Csvanaly.....                                | 167 |
| 6.1 Dispoñibilidade.....                                                 | 167 |
| 6.2 Incompatibilidade.....                                               | 167 |
| 6.3 Erros.....                                                           | 167 |
| 7. Conclusións.....                                                      | 170 |
| 7.1 Características xerais e funcionais.....                             | 170 |
| 7.2 Grao de robustez e capacidade de ser sustentable a longo prazo ..... | 171 |
| 7.3 Puntuacións obtidas (sobre 4).....                                   | 172 |
| 8. Bibliografía.....                                                     | 173 |

## Comunicación entre aplicacións de área de traballo e web.....175

|                                               |     |
|-----------------------------------------------|-----|
| 1. Introducción.....                          | 177 |
| 2. Últimas novidades na plataforma GNOME..... | 178 |
| 2.1 Online Desktop.....                       | 178 |
| 2.2 Conduit.....                              | 181 |
| 2.3 Mojito.....                               | 182 |
| 2.4 Obxectivos.....                           | 184 |
| 3. A biblioteca: RTM-GLib.....                | 185 |
| 3.1 Dependencias.....                         | 185 |
| 3.2 Desenvolvemento.....                      | 186 |

|                                          |     |
|------------------------------------------|-----|
| 3.3 Licenza.....                         | 186 |
| 3.4 Itinerario.....                      | 187 |
| 3.5 Exemplos de uso.....                 | 187 |
| 3.5.1 Autenticación.....                 | 187 |
| 3.5.2 Lista de tarefas.....              | 188 |
| 4. Ideas futuras.....                    | 189 |
| 4.1 Mojito.....                          | 189 |
| 4.2 Tracker miner.....                   | 189 |
| 4.3 EDS backend.....                     | 190 |
| 5. Conclusión.....                       | 191 |
| Epílogo,<br>obra da Asociación GPUL..... | 193 |

# Unhas verbas da Asociación GHANDALF, antes de comezar

Este libro é un compendio, unha escolma na que se recollen parte dos traballos desenvolvidos polos alumnos e alumnas que cursaron a segunda edición realizada en Galicia do Máster en Software Libre. Deste xeito, o libro está conformado por achegas de Manuel Fontán García, Pedro García Rodríguez, Javier Jardón Cabezas, Beatriz Montero Fernández, José Ponte Fuentes e Manuel Rego Casanovas.

Continuando a liña de publicación comezada o ano pasado coas teses de mestrado da primeira edición do Máster en Galicia, dende GHANDALF axudamos un ano máis a que vexan a luz en lingua galega e en formato libro un rico conxunto de traballos técnicos centrados en diversos aspectos do Software Libre. Como o primeiro ano, a heteroxeneidade dos contidos do libro non é máis que un fiel reflexo da heteroxeneidade enriquecedora presente no mundo do Software Libre, sendo esta unha virtude esencial nun movemento que día a día conta con maior forza, e contribúe de xeito máis acentuado na mellora da nosa sociedade. No momento da súa redacción, todos estes artigos superaron os máis estritos estándares de calidade, pero o lector debe ser consciente que os mesmos elaboráronse entre os anos 2008 e 2009, polo que a día de hoxe pode que algún deles poida verse desactualizado respecto do estado da arte actual; o inevitable paso do tempo aféctalle a todo o que nos rodea, e a todas as nosas creacións, pero non por iso é menor o valor do traballo acometido.

Dende o comezo da nosa andaina, en GHANDALF temos moi claro que o software libre e mailo coñecemento aberto son ferramentas fundamentais para o presente e máis para o futuro de Galicia. Fieis ao principio fundacional de ser axentes activos no fomento do Software Libre e do coñecemento aberto na nosa sociedade, levamos a cabo a

publicación de “Máster Software Libre en Galicia, ano II” como achega dun novo libro técnico en galego sobre o software libre, listo para que calquera persoa, entidade ou institución interesada no increíble mundo do Software Libre teña a man un recurso que desexamos que lle poida ser de utilidade.

As derradeiras verbas desta introdución son de agradecemento, agradecemento aos autores orixinais do artigos, pero tamén aos mecenas que nos permitiron sacar unha tirada en papel, á deseñadora e á imprenta, ás entidades autoras do prólogo e epílogo, e a todos os axentes que permiten que o Software Libre sexa unha realidade.

«Brindemos polo commit, que é o átomo do software libre!»

– Asociación GHANDALF

# Prólogo, obra de AGASOL

É unha honra para a nosa asociación empresarial poder colaborar no prólogo deste libro. As empresas que baseamos o noso modelo de negocio en aplicacións *Open Source* estamos ávidas de profesionais formados e concienciados na defensa e promoción dos dereitos asociados ao código que nos fan fortes e competitivos a todos.

Emprender non é sinxelo, e moito menos nos tempos que corren. Escoitamos a diario noticias desalentadoras de empresas que pechan e autónomos que baixan as súas persianas. Asistimos impasibles á proliferación de grandes xigantes comerciais en detrimento do noso pequeno comercio e vemos como os cerebros que terían que gobernarnos dentro de trinta anos non teñen máis remedio que emigrar en busca de mellores oportunidades. Afortunadamente aqueles que vivimos no apaixonante mundo do Software Libre vemos oportunidades en cada esquina, oportunidades que non serían posibles sen as liberdades que o Software Libre nos brinda e garante.

Grazas ao Software Libre vemos empresas pequenas mellorar a súa competitividade a un custo razoable, emprendedores que compiten con grandes consultoras armados cos seus vellos portátiles con unha Debian e a seu kernel a medida, etc. Nunca teríamos imaxinado ver a xigantes tecnolóxicos co pulso tremeroso ao ler noticias sobre un *droide verde*. Estamos observando como o mercado do software cambia a un ritmo vertixinoso. Quen se atreve a documentar o modelo de negocio do Software Libre? Existe un só modelo ou tantos como emprendedores con coraxe? Non podemos desperdiciar a grande oportunidade que nos brindan os millóns e millóns de liñas de código aberto dispoñibles.

Crear emprego é cousa de dous, oferta e demanda, esta estraña parella que se pelexa todos os días para manter o mundo en funcionamento. A ninguén se lle escapa que é mellor ensinarlle a pescar ao cidadán que comprar o peixe fóra; por esta mesma regra está claro que un aumento na demanda de aplicacións libres xera un maior aproveitamento da riqueza se o comparamos coa mera compra de licenzas ao gigante estranxeiro de

turno. Como principais actores económicos, Administracións e empresarios debemos ser conscientes disto e fomentar a creación de empresas e coñecemento no noso ámbito local; autoabastecernos nas TIC é clave.

Esperamos que os proxectos que este libro contén axuden a achegar o pequeno gran de area que a nosa sociedade do coñecemento necesita.

Alejandro Escobar Díaz

Presidente de AGASOL  
(Asociación de Empresas Galegas de Software Libre)

# Soluciones FLOSS para Videoconferencias

Manuel Fontán García

© 2009 Manuel Fontán García

Os contidos desta obra están licenciados baixo unha licenza atribución – partilla nos mesmos termos de Creative Commons. Para consultares unha copia da licenza completa visita a páxina <http://creativecommons.org/licenses/by-sa/3.0/> ou escribe a este enderezo: Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.



## Resumo

Esta tese de mestrado é parte do Mestrado en software libre e resume o labor realizado durante as prácticas do mestrado levadas a cabo en IGALIA.

O obxectivo da presente obra é presentar algunhas das principais ferramentas FLOSS<sup>1</sup> para videoconferencia existentes na actualidade, estudar as súas diferenzas, avaliar os custes de integración e valorar as súas funcionalidades e como se adaptan ás nosas necesidades.

O proxecto tiña como obxectivo integrar unha solución de videoconferencia baseada en FLOSS para que o persoal de IGALIA puidese comunicarse entre si mediante o uso de tecnoloxía de vídeo.

Para atinxir este obxectivo organizamos o traballo nunha serie de pasos que se enumeran a seguir:

- Definición dos requisitos da solución
- Pesquisa das solucións de software existentes
- Avaliación das solucións existentes
- Teste das solucións
- Integración do sistema
- Documentación

Ao longo da obra explicarei e comentarei cada un deles.

---

1 Siglas en inglés de “software libre e de código aberto” (*free/libre and open source software*)

# 1. Requisitos do sistema

Esta sección é un mero resumo dos requisitos de sistema que se escolleron para este proxecto.

Embora requira de unicamente unhas poucas liñas, é unha parte esencial do proxecto. Se se tomaren decisións incorrectas nesta altura, o volume de traballo e os custos aumentarían innecesariamente.

A idea do proxecto era configurar un sistema de videoconferencia que cumprise os seguintes requisitos:

- FLOSS: condición indispensable
- Lixeireza: o obxectivo é reducir o máximo posible o uso de largo de banda e os requisitos de hardware
- Escalabilidade: o sistema debe ser escalable sen que se incremente demasiado o custe
- O sistema debe aproveitar a infraestrutura existente tanto como for posible
- Fácil de usar
- O sistema debe adaptarse a unha organización de 40 persoas

Estes requisitos, se ben xenéricos, son cruciais á hora de escoller a aplicación axeitada.

## 2. Pesquisa de solucións

A pesquisa de solucións foi unha parte esencial do proxecto. Atopei varias, cada unha coas súas peculiaridades. A idea era abordar todas as solucións existentes e logo filtralas tendo en conta as vantaxes e os custos, non só económicos, senón tamén de tempo.

Con isto en mente comecei a pesquisar as solucións FLOSS multidifusión. Non atopei moitas, polo que decidín testar a solución implementada nas instalacións de IGALIA baseada no *ffmpeg* chamado *MADO*. Foi unha experiencia interesante e aprendín moito sobre parámetros de vídeo e requisitos para a partilla de vídeo, mais embora *ffmpeg* e *vlc* sexan ferramentas potentes, non teñen capacidades de sinalización para administrar conferencias, polo que a implementación dunha solución que permitise a varios usuarios conectárense desde distintas redes levaría moito máis tempo do previsto inicialmente. Embora as funcionalidades multidifusión só sexan posibles en ambientes LAN.

Cos coñecementos sobre sistemas de vídeo adquiridos e xunto coa experiencia que tiña en sinalización, retomei as pesquisas e fixen unhas poucas descubertas interesantes. Embaixo enuméranse as máis interesantes, que serán estudadas na próxima sección.

- *Access Grid*
- *AppConference*
- *OpenMeetings*
- *Dimdim*

## 3. Solucións

Nesta sección resúmense as diversas solucións de videoconferencia estudadas, coas súas vantaxes e inconvenientes.

### 3.1 Access Grid

*Access Grid* é un conxunto de ferramentas e tecnoloxías que fornece unha ferramenta de colaboración virtual. *Access Grid* pode ser empregada

como unha ferramenta avanzada de videoconferencia que permite aos participantes interaxiren en tempo real desde múltiples localizacións na Internet. *Access Grid* é máis que una aplicación de videoconferencia, xa que fornece mecanismos para partillar datos, colaborar empregando diversas aplicacións partilladas (como a partilla de material de presentacións), empregar dispositivos de visualización de gran formato e tamén pode empregar fontes de vídeo múltiples para posibilitar conferencias virtuais. A tecnoloxía inventouse no *Argonne National Laboratory* de Chicago (EUA).

*Access Grid* é unha implementación da camada de aplicación *RTP* (siglas en inglés de *Real-time Transport Protocol*) para definir formatos estándar de pacotes para o envío de vídeo e son a través da Internet. Foi desenvolvida polo *Audio-Video Transport Working Group* da *Internet Engineering Task Force* e publicada en 1996 como *RFC 1889*, e substituída polo *RFC 3550* en 2003.

Hai 287 nodos rexistrados de *Access Grid* en 28 países do mundo, e estímase que o seu número atinxirá 500 en 2010. Existe unha forte concentración de nodos en Australia, nos EUA, no Canadá e en Europa. A lista completa está dispoñible en: <http://www.accessgrid.org/nodes>

*Access Grid* está deseñada para ter soporte nativo de comunicación multidifusión. O enderezamento multidifusión envía información a un grupo de destinos de maneira simultánea, creando copias só se fallaren as ligazóns aos destinos múltiples. A implementación do concepto multidifusión ten lugar no nivel de roteamento IP, no que os routers crean vías óptimas para a distribución de datagramas enviados a unha árbore de extensión de enderezos de destino en tempo real. Se a compararmos co enderezamento unidifusión, este último vai enviar información a cada destino por vez, cunha separación para cada nodo. Segundo este estándar, a multidifusión é unha tecnoloxía superior. Porén, a unidifusión é máis estable e pode ser empregada como *backup* para cando fallar a multidifusión.

### 3.1.1 Software

O software da *Access Grid* é libre e de código aberto, e está licenciado baixo AGTPL, que é similar a unha licenza BSD<sup>2</sup>. Entre os compoñentes centrais están *Robust Audio Tool (RAT)*, unha aplicación de código aberto para conferencias de audio e *streaming*, desenvolvida pola *University*

2 <http://www.mcs.anl.gov/research/projects/accessgrid/about/license.html>

*College London*; a ferramenta para videoconferencias desenvolvida polo *Network Working Group* do *Lawrence Berkeley National Laboratory* en colaboración coa Universidade de California e o *Access Grid Venue Client* que inclúe tamén un *Extensible Messaging and Presence Protocol*, especificado en *RFC 3920* e *RFC 3921* e que son proxectos de normas dentro do *Internet Standards Process*.

AG 3.1 é a última versión estable. Está dispoñible para MS-Windows, Apple OSX 10.4, diversas distribucións de GNU/Linux (Fedora, RHEL, Slackware, Debian/Ubuntu) e FreeBSD.

O software pode descarregarse da URL: <http://www.accessgrid.org/software>

Na sección *Install Guides AG* pódense atopar documentos sobre a instalación: <http://www.accessgrid.org/documentation>

### 3.1.2 Hardware

Para alén dun computador axeitado e de conexión de rede, *Access Grid* require tres compoñentes centrais: vídeo, audio e un monitor. Para iso abonda cun portátil con cámara web e alt falantes de ouvido. Tamén se pode optar por unha versión máis sofisticada e empregar un cuarto dedicado con cámaras múltiples, proxectores e separación de servizos en múltiples máquinas para o produtor de vídeo, o usuario e o audio. Aquí pódese consultar unha lista co hardware recomendado: <http://www.accessgrid.org/hardware>

## 3.2 Appconference

*AppConference* é un *plugin* de alto rendemento de *Asterisk* para conferencias de audio e vídeo.

*Asterisk* é o proxecto de telefonía de código aberto máis popular do mundo. En desenvolvemento desde 1999, *Asterisk* é software libre e de código aberto capaz de transformar un computador común nun servidor de comunicacións de voz con múltiples funcionalidades. *Asterisk* facilita a creación e a implementación dun amplo abano de aplicacións e servizos de telefonía.

*Asterisk* é distribuído baixo a versión 2 da licenza pública xeral GNU (*GNU General Public License, GPL*) e tamén está dispoñible baixo licenzas alternativas negociadas directamente con Digium Inc. Se

adquiriches a túa copia de *Asterisk* licenciada baixo a *GPL*, todos os módulos de *Asterisk* que empregares no teu sistema estarán licenciados tamén baixo a *GPL*. A *GPL* (versión 2) está incluída na árbore de código fonte no arquivo *COPYING*.

*Asterisk* é empregado por máis de dous millóns de usuarios e é compatible cun amplo abano de protocolos de telefonía. É altamente compatible co tratamento e transmisión de voz a través de interfaces tradicionais de telefonía, entre as que se inclúen liñas analóxicas, liñas IDN-BRI e *trunks* dixitais T1/E1. *Asterisk* tamén é compatible cunha gran variedade de protocolos VoIP, entre os que se inclúen SIP, IAX e H.323 entre outros. É compatible cos tipos de sinalización europeos e estadounidenses estándar empregados en sistemas de telefonía empresarial, o que lle permite facer de ponte entre a infraestrutura actual e a futura xeración de redes con integración de voz e datos.

Unha PBX só compensa se pode recibir chamadas. É posible empregar *Asterisk* con chamadas VoIP (SIP, H.323, IAX, etc.) mais tamén é posible conectar co RPTC<sup>3</sup> real mediante unhas serie de tarxetas.

### 3.2.1 Software

*AppConference* está escrito en C e licenciado baixo a licenza pública xeral GNU (*GPL*).

O código fonte está dispoñible no repositorio *svn*<sup>4</sup>.

*Asterisk* está desenvolvido principalmente en GNU/Linux para x/86 e funciona en GNU/Linux para PPC xunto con OpenBSD, FreeBSD e Mac OS X. Para calquera que tiver o tempo e a destreza necesarios debería ser sinxelo portar esta aplicación para outras plataformas e sistemas operativos de tipo UNIX baseados en estándares.

*Asterisk* está dispoñible en *Debian Stable* e é mantido polo equipo *Debian VoIP*.

### 3.2.2 Hardware

*Asterisk*® non necesita hardware adicional para *VoIP*. *Asterisk*® é compatible cunha serie de dispositivos para a interconexión con

---

3 Rede pública de telefonía conmutada (*Public switched telephone network* ou PSTN)

4 <https://appconference.svn.sourceforge.net/svnroot/appconference>  
appconference

equipamentos de telefonía analóxica e dixital, por exemplo, con todo o hardware fabricado por *Digium®*, os creadores de *Asterisk®*.

O hardware compatible divídese en dous grupos xerais: os dispositivos DAHDI e os non DAHDI. O hardware compatible DAHDI soporta conferencias pseudo-TDM e funcionalidades de chamada a través de *chan\_dahdi*, mentres que o hardware compatible non DAHDI pode presentar diferentes funcionalidades.

Pódese consultar unha lista completa en:

- Compatible DAHDI:  
<http://www.asterisk.org/astdocs/node12.html>
- Compatible non DAHDI:  
<http://www.asterisk.org/astdocs/node13.html>

### 3.3 OpenMeetings

*OpenMeetings* é un software libre baseado en navegador que permite crear unha conferencia na rede de maneira instantánea. O usuario pode empregar micro ou cámara web, partillar documentos nun taboleiro ou mesmo partillar a pantalla completa. Está dispoñible como servizo hospedado e tamén pode ser descarregado e instalado no servidor sen restricións de uso nin de usuarios.

*OpenMeetings* está desenvolvido sobre unha plataforma *OpenLaszlo*. *OpenLaszlo* é unha plataforma de código aberto para o desenvolvemento e a distribución de aplicacións de Internet ricas (*Rich Internet Applicatons*) na rede de redes. A plataforma *OpenLaszlo* consta de tres compoñentes principais:

- O compilador de *OpenLaszlo* toma un arquivo fonte de *OpenLaszlo* e compílao nun arquivo Flash ou DHTML que roda en calquera navegador
- O *framework* de tempo de execución de *OpenLaszlo* inclúe varios compoñentes de interface de usuario, ligazón de datos e servizos de rede.
- O *servlet* de *OpenLaszlo* permite soporte en tempo de execución para *media types* adicionais e para *SOAP* e *XML-RPC*

*OpenMeetings* emprega *Red5* como elemento central. *Red5* é un servidor flash de código aberto escrito en *Java* compatible con:

- Transmisión de audio e vídeo (FLV, h264, AAC e MP3)
- Gravación de transmisións de cliente (só FLV)
- Obxectos compartillados (*shared objects*)
- Publicación de transmisión ao vivo (*live stream publishing* )
- *Remoting* (AMF)

*RTMP* e *RTMPT* son os protocolos de comunicación de vídeo e audio<sup>5</sup> que emprega *Red5*.

### 3.3.1 Software

*OpenMeetings* estaba licenciado nun principio baixo a *LGPL* (Licenza pública xeral reducida), mais o proxecto está agora licenciado baixo unha licenza *EPL* (licenza pública de Eclipse) para harmonizalo coa licenza de *OpenLaszlo*.

A plataforma *OpenLaszlo* está publicada baixo a licenza pública común certificada pola OSI (*CPL* 1.0). *OpenLaszlo* emprega a *CPL* sen modificacións. IBM creou esta licenza de finalidade xeral para ser empregada por outras compañías que desexasen lanzar os seus produtos baixo unha licenza de código aberto.

*OpenMeetings* require a versión 5 de *Java* (1.5) ou superior, mais recoméndase *Java* 6 (1.6).

*OpenMeetings* 0.9RC5 é a última versión, pódese descargar xunto con *red5* na URL: <http://openmeetings.googlecode.com/files/red5-openmeetings-rc5.zip>

E na seguinte URL pódese atopar código recente (actualizado cada noite): <http://openmeetings.googlecode.com/svn/trunk/singlewebapp/>

*OpenMeetings* roda en Linux, Unix, Windows e Mac OS X.

### 3.3.2 Hardware

O requisitos mínimos (sen conversores de documentos, gravadora nin funcionalidade de subida de ficheiros) son: 1GHz CPU 1 GB RAM .

---

5 [http://en.wikipedia.org/wiki/Real\\_Time\\_Messaging\\_Protocol](http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol)



Recoméndase un dispositivo 2x/4x 2GHz ++ CPU (32 ou 64Bit) 4GB RAM.

### 3.4 Dimdim

*Dimdim* é un servizo web 2.0 baseado en navegador que permite partillar a área de traballo, mostrar diapositivas, falar, chatear e transmitir vídeo a través dunha cámara web. O servizo hospedado de *Dimdim* é gratuito e pódese empregar desde para pequenas xuntanzas até para sesións con centos de asistentes.

*Dimdim* é similar a *OpenMeetings* desde o punto de vista das conferencias. Tamén é unha solución RTSP baseada na web. A principal diferenza ten a ver coa política comercial. O modelo de negocio de *Dimdim* está baseado na venda de conferencias como servizo, hospedando o servidor e ofrecendo diversos niveis de calidade e funcionalidades en función do que pagar o cliente.

De facto, *Dimdim Pro* e *Dimdim Enterprise* nin son libres nin empregan os mesmos compoñentes da versión FLOSS, polo que a maior parte da información sobre *Dimdim* non explica as funcionalidades da versión libre senón das privativas.

#### 3.4.1 Software

*Dimdim Free* está licenciado baixo unha licenza *GPL* v2. O código fonte está dispoñible en:

`cvs-d:pserver:anonymous@dimdim.cvs.sourceforge.net:/cvsroot/dimdim login`

## 4. Avaliación das solucións

Logo de estudar as distintas solucións, vemos que existen dúas estratexias para fornecer comunicacións de audio/vídeo na Internet.

- Mediante telefonía: esta opción estende as capacidades de telefonía de sistemas VoIP baseados en SIP para dotar o sistema de recursos de vídeo. É o caso de *AccessGrid* e *AppConference*.

- Mediante a web: esta opción beneficiase de *RTMP*, o protocolo de transmisión de audio e vídeo máis empregado na Internet, propiedade de *Adobe Systems*.

## 4.1 Avaliación do produto

Escolleuse *OpenMeetings* porque é compatible con LDAP, é unha solución doada de instalar e de testar, é extensible, o seu prezo adaptábase ás nosas necesidades, é escalable e a documentación dispoñible é bastante boa.

Descartouse *AppConference* porque, embora funcione, non parece que vaia ter soporte nun futuro próximo, xa que non forma parte do proxecto *Asterisk*. A razón pode ser que *Asterisk* estea a considerar fornecer función de vídeo nas súas novas versións, polo que solucións *ad hoc* como o *plugin* de *AppConference* non serían unha alternativa atractiva.

*AccessGrid* é unha tecnoloxía moi madura, desenvolvida en ambientes científicos, principalmente en universidades, e é compatible coa multidifusión. Isto fai que sexa moi atractiva, xa que é a única solución entre todas as estudadas con esta característica. Debido aos requirimentos de largo de banda das comunicacións de vídeo, a multidifusión é unha funcionalidade moi desexable, xa que grazas a ela o número de clientes que están a recibir vídeo é independente do largo de banda. Infelizmente, as comunicacións multidifusión aínda non están dispoñibles para o usuario final en España, polo que *AccessGrid* foi descartada.

Desde o punto de vista do cliente, *Openmeetings* é de novo a solución máis simple, xa que só precisa un navegador web con *plugin* de flash para realizar conferencias de audio e vídeo. A segunda opción nesta lista podería ser *AppConference*, xa que, embora precise software adicional para funcionar, a maioría dos clientes *SIP* soportan videoconferencias. A última da lista sería *AccessGrid*, que precisa clientes *RAT* e *VIC* para ser instalada, o que non é complicado pero que a torna nunha solución non comercial.

## 4.2 Avaliación da comunidade

Esta é unha avaliación subxectiva das comunidades que rodean os diferentes proxectos, baseada na miña experiencia persoal durante as prácticas.

*AccessGrid* é desenvolvida principalmente en universidades. O tempo que se tarda en contactar os desenvolvedores varía en función do subproxecto, as respostas soen ser amables e educadas mais non moi alentadoras. O proxecto posúe moita documentación, mais nalgúns casos esta está demasiado dispersa debido a que diferentes partes do sistema desenvólvense en diferentes universidades, cada una das cales segue as súas propias directrices para a elaboración da documentación.

As listas de *AppConference* no parecen mostrar moita actividade, con menos de 25 mensaxes na lista de desenvolvedores de 2009, moitas das cales non tiveron resposta mentres que outras son simples anuncios de actualizacións. Na miña humilde opinión, unha comunidade precisa algo máis ca un repositorio e noticias esporádicas sobre as actividades principais dos desenvolvedores, especialmente se non houber un gran número destes.

*Dimdim* é un exemplo ideal de proxecto non comunitario. Comezou sendo libre e, por mor de razóns comerciais xa non o é. Sei que existe unha versión de código aberto dispoñible, mais non conta cunha comunidade. Isto é debido a unha serie de factores: a documentación é pobre e difícil de atopar e as respostas son escasas e frecuentemente toscas. Polo que lin en varios foros, a maior parte do persoal contratado de *Dimdim* pasouse a *Openmeetings* cando os primeiros mudaron a súa estratexia comercial.

Para rematar, *Openmeetings* ten, na miña opinión, unha comunidade sá e creativa. A documentación é bastante boa, ten unha *wiki* fácil de usar, con multitude de tutoriais, as listas de distribución mostran bastante actividade (cun gran número de respostas), os contribuidores interaxen entre eles e cos desenvolvedores principais. A miña primeira pregunta foi referida ao procedemento de instalación, tiña algúns problemas con algúns pasos automatizados, e axudáronme moito, explicáronme como ler os *logs* e que partes debía subir á lista. Grazas á súa axuda dei instalado a versión 0.9RC5 en Debian Lenny nunha altura na que a versión funcional era a 0.7. Despois disto tomei confianza coa lista e cos logs e comecei a configurar a conexión LDAP. A primeira vez non funcionou como

acostumaba facelo, mais lendo as listas descubrín que un dos contribuidores tivera o mesmo problema con openLDAP. Pedínlle que me enviase o remendo e logo duns poucos testes solucionei o problema. Sei que non é gran cousa, mais o feito de identificar un bug e de ver como reparalo é moi alentador, e lograr que a aplicación funcionase de novo foi satisfactorio e divertido.

## 5. Documentación

Despois de configurar totalmente o sistema só restaba unha cousa: documentalo. A documentación fíxose seguindo as directrices de IGALIA, isto é, manuais de usuario e administrador elaboradas e funcionais.

Seguindo estes principios, empregouse *reStructuredText* para redixir os manuais. *reStructuredText* é un sistema analizador de linguaxe de marcación fácil de ler e *WYSIWYG*<sup>6</sup>. É útil para documentar linguaxes de programación (como as *docstrings* de *Python*), para crear páxinas web simples e para redixir documentos independentes. *reStructuredText* está deseñado para ser extensible para dominios de aplicación específicos. O analizador de *reStructuredText* é un compoñente de *Docutils*. *reStructuredText* é unha versión modificada dos sistemas de marcación lixeiros *StructuredText* e *Setext*.

O obxectivo principal de *reStructuredText* é definir e implementar unha sintaxe de marcación para ser empregada nas *docstrings* de *Python* e outros dominios de documentación simple e fácil de ler, mais ao mesmo tempo o suficientemente potente como para ser empregado por profesionais. O propósito da marcación é converter os documentos de *reStructuredText* en formatos de datos estruturados e útiles.

*reStructuredText* ten varias vantaxes: permite xerar html con facilidade a partir de fontes .rst. Deste xeito pódense empregar arquivos .css para definir o estilo, pode ser incluído nun repositorio, permitindo así que se fagan commits para reparar os erros. Tamén é fácil de ler, polo que é doado modificar as fontes, algo de gran axuda para os administradores

*reStructuredText* pódese escribir con calquera editor de texto. Para descargar o pacote *docutils* abonda con:

---

6 *What you see is what you get* (ves o que obtés)

```
# apt-get install python-docutils  
$rst2html.pyfile.rstfile.html
```

Se estás interesado en ter unha copia das guías orixinais xeradas, podes poñerte en contacto comigo.

## 6. Bibliografía

- OpenMeetings Homepage, 2009, <http://code.google.com/p/openmeetings/>
- AccessGrid Homepage, 2009, <http://www.accessgrid.org/>
- AppConference Forge, 2009, <http://sourceforge.net/projects/appconference/>
- CentOS Homepage, CentOS Project, 2009, <http://www.centos.org/>
- Debian Homepage, Debian Project, 2009, <http://www.debian.org/index.es.html>
- VideoLAN Homepage, VideoLAN team, 2008, <http://www.videolan.org>
- Ubuntu Server Guide, Canonical Ltd., 2006, <https://help.ubuntu.com/8.04/serverguide/C/index.html>
- T. Oetiker et al., The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X $\epsilon$ , 4th ed., 2008

# Dinámicas das comunidades de software libre: Python

Pedro García Rodríguez

© 2009 Pedro García Rodríguez

Os contidos desta obra están licenciados baixo unha licenza atribución – partilla nos mesmos termos de Creative Commons. Para consultares unha copia da licenza completa visita a páxina <http://creativecommons.org/licenses/by-sa/3.0/> ou escribe a este enderezo: Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

## Resumo

*Python* é unha linguaxe de programación dinámica e orientada a obxectos que pode ser empregada en multitude de tipos de desenvolvemento de software. Ofrece un soporte sólido para a integración con outras linguaxes e ferramentas, posúe bibliotecas estándar moi amplas e foi deseñado para que se poida aprender nuns poucos días.

O obxectivo deste artigo é facer unha análise do proxecto *Python* desde un punto de vista histórico, tentando fornecer unha visión global do proxecto desde os estadios iniciais, licenza, fundación da comunidade e finalmente a historia do seu desenvolvemento, para determinar os beneficios de escoller unha ferramenta derivada da análise de software.

Este documento está encamiñado a fornecer unha visión xeral do proxecto e da historia do seu licenciamento, pasando polas características da súa comunidade e da súa fundación, e rematando cun estudo do seu desenvolvemento ao longo do tempo.

A información empregada na redacción deste documento tomouse de diversas fontes, como por exemplo versións lanzadas, xestión do código fonte, listas de distribución, documentos sobre bugs, investigación, etc.

# 1 Introducción a Python

## 1.1 Unha breve introdución a Python

*Python* foi concibido a finais da década dos 80 por Guido van Rossum no CWI dos Países Baixos, como sucesor á linguaxe de programación ABC.

Guido van Rossum é o autor principal de *Python*, e o seu papel ininterrompido decidindo que rumbo deberían seguir *Python* e a súa comunidade fica reflectido no título que lle outorgou a comunidade *Python*: “benévolo ditador perpetuo” (*BDFL* polas súas siglas en inglés). Actualmente, Guido van Rossum está a traballar para Google Inc.

O nome desta linguaxe de programación non ten a ver cos réptiles. Na realidade, vén da comedia da BBC “*Monty Python's flying circus*”. Como deseñador e programador principal, Guido achou que precisaba un nome curto, único e lixeiramente misterioso. Dado que era seguidor do programa, pensou que o nome *Python* sería perfecto.

*Python* é unha linguaxe de programación dinámica e orientada a obxectos que pode ser empregada para múltiples tipos de desenvolvemento de software.

*Python* está dispoñible para os principais sistemas operativos. Funciona en Windows, Linux/Unix, Mac OS X, OS/2, Amiga, PDAs e teléfonos móbiles Nokia. *Python* foi portado tamén para a plataforma .NET, a máquina virtual de *Java* e os móbiles Nokia da serie 60.

Seguro que gustarás de saber que o mesmo código fonte funciona sen modificacións en todos os soportes.

Fornece un soporte sólido para a súa integración con outras linguaxes e ferramentas, posúe bibliotecas estándar moi amplas e pódese aprender en poucos días.

Moitos programadores de *Python* aseguran que a súa produtividade aumentou grazas a esta linguaxe e que senten que os anima a desenvolver un código de maior calidade e máis fácil de manter.

A popularidade de *Python* non deixou de aumentar desde que foi creado, a comunidade de usuarios e desenvolvedores desta linguaxe é enorme. Multitude de proxectos de código aberto como *Zope*, *Django* ou



*Gimp* foron desenvolvidos en *Python*, e un enorme abano de grandes compañías como *Industrial Light and Magic* emprégano diariamente nas súas aplicacións.

## 1.2 As mellores funcionalidades de Python

- **Versatilidade.** Permite crear toda clase de programas. *Python* non foi creado especificamente para a Internet, embora sexa unha potente ferramenta para o desenvolvemento de sitios e de aplicacións web.
- **Multiplataforma.** Hai multitude de versións distintas de *Python* para diversos sistemas operativos. Nos seus inicios foi desenvolvido para *Unix*, mais calquera sistema é compatible con esta linguaxe se houber un intérprete dispoñible para o mesmo.
- **Interpretado.** Quere dicir que o código pode ser executado sen necesidade de compilalo. Na realidade, ten lugar unha compilación, mais esta é transparente para o programador. Nalgúns casos, cando se executa o código pola primeira vez, uns poucos *bytecodes* almacénanse no sistema. Estes *bytecodes* aceleran a compilación que o intérprete realiza cada vez que executa o mesmo código.
- **Interactivo.** *Python* posúe un intérprete de liña de comandos no que se poden colocar sentenzas. Cada sentenza é executada e produce un resultado visible, que facilita a comprensión da linguaxe e permite testar con rapidez os resultados da implementación de anacos de código.
- **Orientado a obxectos.** *Python* soporta a programación orientada a obxectos. *Python* fornece, en moitos casos, un xeito simple de crear programas con compoñentes reutilizables.
- **Funcións e bibliotecas.** *Python* posúe multitude de funcións contidas no código para o tratamento de cadeas de caracteres, números, arquivos, etc. Para alén diso, existen multitude de bibliotecas que poden ser importadas en programas para administrar temas específicos.

- **Sintaxe clara.** Para rematar, a sintaxe de *Python* é moi visual, e posúe notación de identidade (con marxes). En moitas linguaxes, para separar anacos de elementos de código, empréganse unha serie de claves ou palabras clave como inicio e fin. Para separar pedazos de código en *Python* basta con tabular o texto, creando así unha marxe que indica que o código fica dentro dunha función ou dun lazo. Isto axuda aos desenvolvedores a adoptaren as mesmas notacións, de xeito que os programas de cada un deles teñen unha estrutura moi similar.

## 1.3 Unha ollada á historia de Python

Guido van Rossum comezou a traballar no CWI (*Centrum voor Wiskunde and Informatica*) a comezos dos 80. Formaba parte do grupo de programación da linguaxe *ABC*.

O obxectivo principal do grupo de programación da linguaxe *ABC* consistía en desenvolver unha linguaxe de programación que fose fácil de ensinar a usuarios sen ningún tipo de coñecementos de programación nin experiencia no desenvolvemento de software.

A linguaxe *ABC* non tivo éxito, e Guido comezou a traballar no sistema operativo *Amoeba* desenvolvido por Tanenbaum. O SO *Amoeba* tamén era un proxecto do CWI. Guido pensou en crear unha linguaxe de *script* baseada na linguaxe de programación *ABC* que facilitase o desenvolvemento do SO *Amoeba*.

Guido van Rossum comezou a desenvolver a súa idea no Nadal de 1989. Non demorou en descartar a idea de tornar a nova linguaxe nunha linguaxe de implementación específica para o SO *Amoeba*. Finalmente, a primeira versión de *Python* en ver a luz foi a 0.9 e foi lanzada a *USERNET* en febreiro de 1991.

Esta primeira versión consistía na especificación da linguaxe, na implementación dunha máquina virtual, no intérprete da linguaxe e nun conxunto de bibliotecas estándar.

En 1994, Guido van Rossum lanzou a versión 1.0 de *Python*. No mesmo ano creouse o primeiro foro de debate en *USERNET*. Este feito

fixo que o proxecto gañase popularidade, que aumentase a súa relevancia e que continuase a evoluir.

Guido van Rossum traballou no CWI até 1995, ano no que se trasladou a Virxinia (EUA) para traballar no CNRI (siglas en inglés da Corporación para Iniciativas de Investigación Nacional). Formou un equipo de desenvolvemento para traballar en Python, e lanzou varias versións até a 1.6.1 no ano 2000.

En maio dese ano, Guido van Rossum e o equipo de desenvolvemento comezaron a traballar en *BeOpen*, formando o grupo *BeOpen PythonLabs*. A versión 2.0 foi lanzada en outubro e Guido van Rossum e o seu equipo comezaron a traballar en *Digital Creations*, que máis tarde mudou o seu nome para *Zope Corporation*.

A versión actual do produto é, no momento de redacción deste artigo, a 3.1 e foi lanzada o 27 de xuño de 2009. Recoméndase usar a 2.6.2 xa que se considera estable e é máis doado atopar software de terceiros compatible. Na tabela 1 móstranse todas as versións de Python desde o primeiro lanzamento, desde o da versión 0.9 efectuado en febreiro de 1991 até o da última versión estable, a 3.1.

| Python Release | Release Date    | Python Release | Release Date    |
|----------------|-----------------|----------------|-----------------|
| 0.9            | February 1991.  | 2.1            | April 2001.     |
| 1.0            | January 1994.   | 2.2            | December 2001.  |
| 1.1            | October 1994.   | 2.3            | July 2003.      |
| 1.2            | April 1995.     | 2.4            | November 2004.  |
| 1.3            | October 1995.   | 2.5            | September 2006. |
| 1.4            | October 1996.   | 2.5.2          | February 2008.  |
| 1.5            | December 1997.  | 2.6.2          | April 2009.     |
| 1.6            | September 2000. | 3.0.2          | February 2009.  |
| 1.6.1          | September 2000. | 3.1            | June 2009.      |
| 2.0            | October 2000.   |                |                 |

*Tabela 1: versións de Python e datas de lanzamento*

## 1.4 Quen usa Python?

Python emprégase con frecuencia como linguaxe de *script* para aplicacións web, como por exemplo a través de *mod\_python* para o servidor web *Apache*. *Frameworks* de aplicacións web ou servidores de aplicacións como *web2py*, *Zope* e *Django*, desenvolvidos en *Python*, axudan os desenvolvedores a deseñaren e manteren aplicacións complexas.

*Python* é empregado amplamente no sector da seguridade da información.

*Python* ten sido embebido con éxito como linguaxe de *script* nunha serie de produtos de software como *Abaqus 3D*, paquetes de animación 3D como *Maya*, *Softimage XSI* ou *Blender* e en programas de creación de imaxes 2D como *GIMP*, *Scribus*, *Inskape* ou *Paint Shop Pro*.

ESRI, a compañía que produce *ArcGIS*, está a promover *Python* como a mellor escolla para escribir *scripts* en *ArcGIS*.

Tamén ten sido empregado en diversos videoxogos e é un compoñente estándar de multitude de sistemas operativos. Está incluído na maioría das distribucións de GNU/Linux, de NetBSD, de OpenBSD e de Mac OS X. Linux Red Hat e Fedora empregan o instalador Anaconda escrito en Python. Linux Gentoo emprega *Python* en *Portage*, o seu sistema de mantemento de paquetes. Linux Pardus emprégao para tarefas de administración e durante a iniciación do sistema.

Outras aplicacións que empregan *Python* son *YouTube* e o cliente orixinal *BitTorrent*. Entre as grandes empresas que empregan *Python* están Google, Yahoo!, o CERN, a NASA e ITA. A maior parte do software “*Sugar*” dos dispositivos *XO* da iniciativa “Un laptop por neno”, que está a ser desenvolvido en *Sugar Labs*, está escrito en *Python*.

## 2. A Python Software Foundation

### 2.1 Unha ollada á Python Software Foundation

A *Python Software Foundation* é unha entidade sen ánimo de lucro que posúe os dereitos de propiedade intelectual da linguaxe de programación *Python*. Administra o licenciamento de código aberto da versión 2.1 e posteriores de *Python* e posúe e protexe o resto de marcas rexistradas asociadas a *Python*.

A *Python Software Foundation* foi anunciada por Guido van Rossum en 2001 na novena *PyConf* debido ao recente éxito da *Apache Software Foundation* (a ASF formárase en 1998) e ao notable aumento do código e dos usuarios.

Os propietarios do proxecto decidiron que a mellor maneira de estimular o interese da comunidade pasaría por crear unha fundación cuxos principais obxectivos serían promover, protexer e desenvolver a linguaxe de programación *Python*, fomentar o uso e desenvolvemento da mesma, preservar o código, expandir a comunidade e crear unha protección legal contra calquera clase de denuncias.

Outras das tarefas que leva a cabo a *Python Software Foundation* son a organización, o financiamento e o mantemento da fundación. A fundación está formada por 112 membros, a maioría dos cales non son desenvolvedores principais.

Calquera que teña colaborado con *Python* (código, organización, blogs) pode ser nomeado membro por calquera dos membros vixentes. Para iso é necesario ter mostrado algún grao de compromiso. Entre os 112 membros hai 25 membros patrocinadores, entre os que se contan empresas como ActiveState, Canonical, Google Inc., Lucas Films, Microsoft, O'Reilly Press, Sun Microsystems e Zope Corporation.

### 2.2 Finalidade da Python Software Foundation

Quen botar unha ollada á declaración de intencións da *Python Software Foundation*, verá que a súa finalidade é “promocionar, protexer e

desenvolver a linguaxe de programación *Python* e apoiar e facilitar o crecemento da comunidade internacional de programadores de *Python*”.

Como se pode deducir desta declaración de intencións, os principais obxectivos da fundación son producir, manter e distribuír o proxecto libre. Para atinxir este obxectivo, confía no uso dunha licenza, a *Python Software Foundation License* ou *PSFL*, que permite a calquera usuario acceder ás fontes, executalas, modificalas e redistribuílas.

Outra característica da *Python Software Foundation* é a protección tanto do produto como do seu nome, da marca e dos usuarios. Tamén mantén a propiedade intelectual do código desenvolvido e protexe as súas propias contribucións a proxectos externos. Ao mesmo tempo, tenta promover a adaptación da *PSFL* a calquera tipo de código ou produto baseado en *Python*.

Para rematar, a fundación tenta facilitar a promoción e a publicidade e administra un programa de doazóns voluntarias para manter a súas actividades e expandirse.

## 2.3 Financiamento e mantemento da Python Software Foundation

O financiamento que a *Python Software Foundation* precisa para sobrevivir obteno a través do seu programa de bolsas e patrocinio. Grazas a estes ingresos, a *Python Software Foundation* pode pagar salarios a algúns membros que realizan tarefas concretas, como manter a infraestrutura, crear programas de bolsas ou organizar eventos como a *PyCon*.

Segundo as últimas noticias publicadas na web de *Python* (27 de maio de 2008), 881 persoas ou entidades fixeron doazóns dalgunha clase, sendo Google Inc. o principal contribuínte.

Os membros da fundación non son suficientes para realizar todas as tarefas, polo que existe un programa de voluntariado para que individuos alleos á organización poidan colaborar en traballos de documentación, organización e publicación.

## 2.4 Organización da *Python Software Foundation*

A *Python Software Foundation* é unha organización complexa.

Está formada por diversos membros con distintos graos de responsabilidade. Algúns deses membros forman parte do comité de directores, responsable de manter os obxectivos da fundación. O comité de directores conta co apoio de varios comités nos que pode delegar certas tarefas e responsabilidades.

A seguir detállanse os diversos compoñentes e membros da *Python Software Foundation*.

- **Membros:** os membros que poden formar a fundación divídense nas seguintes categorías:
  - *Membros nomeados:* os membros nomeados son membros ou entidades con dereito a voto dentro da fundación. Realizaron ou realizan un papel importante dentro da comunidade *Python*.
  - *Membros patrocinados:* comparten características cos membros nomeados, coa excepción de que patrocinan a fundación mediante un pagamento anual.
  - *Membros eméritos:* non teñen dereito a voto. Para se uniren á fundación teñen que ser nomeados primeiro por un membro da mesma. Unha vez aceptada a solicitude, esa sométese á votación dos membros da fundación.
- **Comité de directores:** os membros elixen os directores unha vez ao ano. Os directores realizan as tarefas administrativas da fundación e son os responsables de manter os obxectivos descritos anteriormente. Na actualidade son oito. Os directores comunícanse entre eles a través de listas de distribución e reúnense como mínimo unha vez ao mes a través do IRC. As súas decisións son públicas e pódense consultar en: <http://www.python.org/psf/records/board/minutes>  
<http://www.python.org/psf/rec>.
- **Oficiais:** despois de elixir o comité de directores, estes elixen á súa vez os oficiais, que deben constar de, como mínimo, un

presidente, un secretario e un tesoureiro. Tamén poden escoller varios vicepresidentes e axudantes dos oficiais. As tarefas destes membros son:

- O *presidente* é o responsable dos asuntos burocráticos.
- O *secretario* é o responsable da xestión da información dos membros da fundación e de redactar e archivar as actas e resolucións das xuntanzas.
- O *tesoureiro* mantén e xestiona as contas e os fondos da fundación.
- **Comités de xestión de proxectos:** en función das necesidades da fundación, os directores poden crear diversos comités para a xestión e realización de tarefas específicas. Estes comités deberán estar formados como mínimo por un individuo e por membros da comunidade para así crear oportunidades. Na actualidade existen os seguintes comités:
  - *Python Conference Committee (PCC)*: realiza tarefas relacionadas coa organización da conferencia anual de desenvolvedores de *Python* (*PyCon*).
  - *Infrastructure Committee (IC)*: responsable da infraestrutura e o mantemento da web de *Python*.

## 2.5 A licenza da Python Software Foundation

O desenvolvemento da licenza da *Python Software Foundation* está intimamente ligado á súa historia. Como expliquei anteriormente, un dos obxectivos da fundación desde a súa creación foi o de tornar *Python* nunha linguaxe accesible para todo o mundo. En consecuencia, a súa implementación foi sempre acompañada dunha licenza que permite “usalo e distribuílo libremente, mesmo con fins comerciais”.

É conveniente repararmos en que o código sempre foi libre (segundo a definición da *OSI*) e compatible coa *GPL* (excepto en caso de emerxencia, como se explica embaixo).



Inicialmente, a licenza de *Python* era propiedade do CWI (centro no que Guido comezou o seu proxecto) e era unha variante da *Historical Permission Notice and Disclaimer* (notificación e declaración de permiso histórico), que permite usar, distribuír e modificar o programa libremente, mais non incluír unha copia da notificación de resignación.

Cando Guido van Rossum abandonou a CNRI, *Python* mantivo a mesma licenza, mais o *copyright* foi cedido á CNRI. No ano 2000, e tras o lanzamento da versión 1.6, a licenza mudou para *licenza Python* (PL polas siglas en inglés). Esta licenza non era compatible coa *GPL* porque unha das cláusulas establecía que en caso de litixio ficaba baixo as leis do estado de Virxinia (EUA).

Logo de varias modificacións, a versión 1.6.1 da licenza pasou a ser compatible coa *GPL*.

Desde aquela, e grazas ao nacemento da *Python Software Foundation*, creouse unha nova licenza, a *Python Software Foundation License (PSFL)*, compatible coa *GPL*. Mais non foi até a versión 2.1 que o código de *Python* ficou totalmente licenciado baixo a *PSF*.

Esta licenza conta coa aprobación da Fundación polo Software Libre e da *Open Source Initiative*, e é unha licenza permisiva. Tamén é compatible coa *GPL*.

A licenza *Python* impón moi poucas restricións a respecto do que un pode facer con *Python*. A *Python Software Foundation* posúe o *copyright* da maior parte do software. Uns poucos arquivos son propiedade de terceiros, mais todos eles fican licenciados baixo a mesma licenza.

| Version     | Year        | License |
|-------------|-------------|---------|
| 0.9 / 1.5.2 | 1991 - 1999 | CWI     |
| 1.6 / 2.0   | 2000 - 2001 | GPL     |
| 2.1 / 3.1   | 2001 - 2009 | FSF     |

*Tabela 2: diversas licenzas de Python*

### 2.5.1 Contidos da Python Software Foundation License

1. Este é un acordo de licenza entre a *Python Software Foundation* (doravante *PSF*) e a organización individual (doravante usuario da licenza) sobre o acceso e uso do software (*Python*) en formato código ou binario así como da documentación asociada.
2. A *PSF* garante ao usuario da licenza, no marco deste acordo de licenza, unha licenza non exclusiva, de título gratuito e de validez internacional para a reprodución, análise, proba, execución, presentación pública, creación de obras derivadas e distribución de *Python* en formato orixinal ou modificado, coa condición de que o programa *Python* así como as súas modificacións fiquen baixo o acordo de licenza e a notificación de *copyright* da *PSF*, isto é, “*Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Python Software Foundation, todos os dereitos reservados*”.
3. Se o usuario da licenza crear unha obra derivada baseada en ou que incluír *Pyhton* total ou parcialmente, e quixer poñer a devandita obra derivada á disposición do público, o usuario da licenza deberá acceder a incluír na devandita obra un resumo breve das modificacións realizadas sobre *Python*.
4. A *PSF* pon *Pyhton* a disposición do usuario da licenza tal como se encontra. A *PSF* non presta representacións nin garantías, implícitas nin explícitas. A título de exemplo, mais non de maneira excluínate, a *PSF* fica eximida de calquera tipo de representación ou garantía de comercialización ou adecuación a calquera propósito particular ou calquera violación de dereitos de terceiros derivada do uso de *Python*.
5. A *PSF* non poderá ser responsabilizada polo usuario da licenza nin por ningún outro usuario por calquera clase de danos ou prexuízos incidentais, especiais ou consecuentes que resultaren da modificación, distribución ou calquera outro uso de *Python* ou calquera programa derivado do mesmo, mesmo cando se tiver advertido da posibilidade de tales danos.
6. Este acordo de licenza expirará automaticamente se se incumpriren os seus termos e condicións.
7. O presente acordo de licenza non establece ningún tipo de relación de axencia, parceria nin empresa común entre a *PSF* e o

usuario da licenza. O presente acordo de licenza non autoriza o uso das marcas comerciais nin do nome comercial da PSF como marca rexistrada para o apoio e/ou promoción de produtos e servizos do usuario da licenza nin de terceiros.

8. Ao copiar, instalar ou usar Python, o usuario da licenza concorda en observar os termos e condicións do presente acordo de licenza.

A seguir enuméranse as características principais da licenza de *Python* en linguaxe comprensible.

As descrições que seguen non teñen validez legal. Para saber que termos da licenza son de aplicación en cada caso particular, recoméndase ler o texto íntegro da licenza e consultar un profesional.

- *Python* é totalmente gratuíto, mesmo para usos comerciais (incluída a revenda)
  - Pódense vender produtos escritos en *Python* ou produtos que inclúen o intérprete *Python*. Ambos casos fican exentos do pagamento de licenzas.
- *Open Software Initiative* certificou a *Python Software Foundation License* como software de código aberto, e inclúea na súa lista de licenzas de código aberto.
- Non existen restricións de *copyleft* do tipo *GPL*
  - É permitida a distribución de versións exclusivamente binarias de *Python*, modificadas ou non. Non é necesario liberar o código fonte. Tamén é permitido escribir módulos de extensión para *Python* e fornecelos só en formato binario.
- Porén, a licenza *Python* é compatible coa *GPL*, segundo a *Free Software Foundation*.
- Non é permitido eliminar a notificación de *copyright* da *PSF* nin do código fonte nin do binario resultante.

## 3. A comunidade Python

Python, como calquera proxecto relevante de software libre, nútrese das colaboracións da comunidade.

Estas colaboracións poden ser de varios tipos, desde o desenvolvemento de código para a implementación da linguaxe a través do código ou a documentación dos manuais de tradución e páxinas web, até a organización de eventos e seminarios para publicitar o proxecto.

Se non existiren estas contribucións desinteresadas, o proxecto nunca tería atinxido a dimensión que posúe na actualidade. Esta sección ofrece un resumo das principais características da comunidade *Python*.

### 3.1 Comunicación entre os membros da comunidade Python

Para que a comunidade medre e manteña unha certa infraestrutura é preciso que flúa a información e a comunicación entre os membros.

A comunidade *Python* emprega principalmente as seguintes canles:

- **Listas de distribución:** son a principal canle de comunicación entre os membros. As listas máis importantes son *python-announce-list* e *python-dev*.
- **Grupos de noticias (*newsgroups*):** Python comezou publicitándose en USENET. O maior grupo de noticias do proxecto, creado en 1994, segue activo. É o grupo *comp.lang.python*.
- **Canais IRC:** existe un canal chamado *#python* dentro da *FreeNode Irc Network* no que os colaboradores do proxecto poden resolver ou aclarar dúbidas aos usuarios en tempo real.
- **Conferencias:** a conferencia principal é a *PyCon*. En todo o mundo celébranse moitas máis conferencias, como por exemplo no Brasil, en Italia, Francia, Reino Unido ou Arxentina.

## 3.2 PyCon: a conferencia sobre Python

*PyCon* é a conferencia da comunidade *Python*, organizada polos seus membros.

*PyCon* está pensada para entusiastas sen importar o seu nivel de coñecementos, desde usuarios recentes a desenvolvedores. *PyCon* ofrece a oportunidade de:

- Asistir a unha gran variedade de presentacións, paneis e debates espontáneos
- Coñecer os avances máis relevantes da comunidade de desenvolvedores de *Python*
- Coñecer desenvolvedores de todo o mundo
- Participar en aulas presentadas por expertos
- Participar en cursos intensivos de desenvolvemento

O obxectivo dos organizadores é lograr que a conferencia sexa accesible e aberta a todo o mundo.

A primeira das conferencias *PyCon* celebrouse nos EUA en 2003. Na actualidade celébranse conferencias en todo o mundo que partillan o espírito da *PyCon*.

Algunhas das principais son:

- A *PyCon* UK 2007 celebrada en Birmingham, Reino Unido
- A *PyCon* Brasil 2007 celebrada en Joinville, Santa Catarina
- A *PyCon* Italia 2007 celebrada en Florencia
- A *PyCon* 2008 Chicago celebrada en Rosemont (Chicago), Illinois, EUA
- As *PyCon* TX2006 e TX2007 celebradas en Addison, Texas, EUA
- As *PyCon* DC2003, DC2004 e DC2005 celebradas en Washington DC, EUA

## 4. Python para desenvolvedores

Esta sección mostra os recursos que fican á disposición dos desenvolvedores de *Python*, tanto dos desenvolvedores principais que posúen acceso *commit* como de aqueles que desexaren participar, contribuír e aprender.

- **Issue tracker**<sup>7</sup>: fornécese un sistema de rastreamento de incidentes para xerir e manter as listas de incidentes.
- **Source code repository (SVN)**<sup>8</sup>: o repositorio do código fonte permite xerir e manter as distintas ramas e versións do código fonte de *Python*.
- **Mailing lists**<sup>9</sup>: fornécense diversas listas de distribución que permiten a comunicación entre os membros e usuarios da comunidade *Python*. Algunhas delas son: *Python-Dev*, *Python-Announce-List* ou *Python-Bugs-List*.
- **Python enhancement proposals (PEPs)**<sup>10</sup>: as propostas de mellora de *Python* son documentos que conteñen información sobre novas funcionalidades da linguaxe e meta-información como por exemplo procedementos ou calendarios de lanzamento, maneiras de adicionar módulos novos a diversas bibliotecas, directrices para a depuración de erros, etc.
- **Documentation development**<sup>11</sup>: contén ligazóns e información relacionadas co desenvolvemento da documentación.

---

7 <http://bugs.python.org/>

8 <http://svn.python.org/projects/python/trunk>

9 <http://mail.python.org/mailman/listinfo/>

10 <http://dinsdale.python.org/dev/peps/>

11 <http://python.org/dev/doc/>

## 5. Análise de Python

### 5.1 Datos xerais sobre Python

As tabelas 6 e 7 (tabelas dispoñibles ao final do capítulo) mostran un resumo do proxecto da linguaxe de programación *Python*. Tamén inclúe unha lista das linguaxes identificadas empregadas na linguaxe de programación *Python* na versión 3.1.

### 5.2 Historia do desenvolvemento

O repositorio de rexistros do código fonte (*SVN*) de *Python* contén toda a historia para estudar a actividade do proxecto. O primeiro *commit* fixose en agosto de 1990 e foi obra de Guido van Rossum. Guido traballou só no proxecto durante case dous anos até que se lle uniron Jack Jansen e Sjoerd Mullender.

O número de *committers* non aumentou de maneira significativa até o desenvolvemento da versión 2.0 no ano 2000. Tras a fundación da *Python Software Foundation* en 2001 o proxecto experimentou un aumento drástico da actividade e da participación, e atinxiu os 42.626 *commits* e 173 *committers*.

A trazos largos, cada mes rexístrase unha media de 186,1397 *commits*.

A ilustración 1 (ilustracións dispoñibles ao final do capítulo) mostra a media de *commits* desde 1990. O número máximo de *commits* nun mes foi de 717 e o mínimo de 2.

A ilustración 2 mostra os programadores que máis *commits* fixeron desde o primeiro lanzamento. 18 programadores fixeron máis de 500 *commits*. É interesante repararmos en que o programador con máis *commits* é Guido van Rossum.

### 5.3 Linguaxes de programación

En *Python* empréganse dúas linguaxes de programación claramente diferenciadas: a linguaxe de programación *C* e *Python*. A linguaxe *C* emprégase na implementación de analizadores léxicos, sintácticos e

semánticos, así como tamén na máquina virtual. *Python* emprégase principalmente na implementación das bibliotecas estándar.

Tamén se empregan outras linguaxes como *Perl*, *Lisp* e mesmo *Montador (assembler)* en tarefas moi específicas. Como se pode ver, *Python* e *C* agrupan case todo o código en porcentaxes similares (50% fronte a 47%).

Curiosamente, esta tendencia repítese na maioría das versións analizadas. Só se interrompe entre as versións 1.6 e 2.0 (inclusive), nas que o código programado en *C* supera 60% e o programado en *Python* é inferior a 40%.

A tabela 3, ao final deste capítulo ao igual que o resto de tabelas e ilustracións, mostra a distribución do código de cada versión de *Python*.

## 6. Estatísticas das listas de distribución

A comunidade *Python* posúe gran cantidade de listas de distribución, cada unha enfocada a un tema concreto.

Algunhas listas céntranse en todo o que ten a ver con *Python*, outras comentan as últimas noticias, noutras débátese as novas funcionalidades, etc. Hai máis de 50 listas de distribución, mais eu voume centrar soamente en dúas.

As listas que vou comentar son a *python announce* (empregada para publicar anuncios sobre a linguaxe de programación *Python*) e *python.dev* (foro de debate dos desenvolvedores principais).

### 6.1 A lista de distribución *Python announce*

Desde a creación desta lista en 1999 recibíronse 7.613 correos electrónicos, enviados por 1.831 individuos diferentes.

O servidor web co maior nivel de actividade nesta lista é *gmail.com*, cun total de 931 correos electrónicos enviados por 253 persoas distintas. Detrás de *gmail.com* están *yahoo.com* (45 usuarios) *python.org* (19



usuarios), *hotmail.com* (18 usuarios) *gmx.net* (17 usuarios) *users.sourceforge.com* (16 usuarios) e *zope.com* (14 usuarios).

Os dominios *.com*, *.net*, *.org* e *.edu* son os máis empregados, con 3.428, 958, 909 e 288 respectivamente. Se botarmos unha ollada á información fornecida por estes dominios, e centrarmos a atención nos dominios nacionais, veremos que a maior parte das persoas que enviaron correos a esta lista viven en Alemaña (dominio *.de*, 514 correos), Australia (dominio *.au*, 253 correos), Francia (dominio *.fr*, 169 correos), Reino Unido (dominio *.uk*, 148 correos). Detrás veñen Canadá (dominio *.ca*, 112 correos), Nova Zelandia (dominio *.nz*, 112 correos) e Rusia (dominio *.ru*, 98 correos).

Se botarmos unha ollada ao número de mensaxes enviadas á lista desde 1999, veremos unha tendencia que vai en aumento. O ano de menos mensaxes é, evidentemente, 1999, xa que foi o ano en que foi creada a lista. 2008 foi o ano de menos mensaxes na tendencia ascendente da lista, mais só se rexistraron 10 mensaxes menos que os anos anteriores, polo que non é relevante á hora de definir a tendencia da lista como ascendente.

## 6.2 A lista de distribución Python dev

Desde que foi creada a lista en 1999 recibíronse 74.431 mensaxes, enviadas por 2.350 persoas diferentes.

O servidor web de maior actividade nesta lista é *python.org* (9.393 mensaxes), seguido de *gmail.com* (7.898 mensaxes), *pobox.com* (2.357 mensaxes), etc.

A tabela 4 mostra os dominios máis activos na lista de distribución *python-dev*. A seguinte parte desde capítulo basearase nos datos de dominio.

O total de individuos que enviaron mensaxes desde *python.org* é de 14. Pode parecer un erro, mais só Guido van Rossum, o creador de *Python*, enviou máis de 6.500 mensaxes. A segunda persoa con maior número de mensaxes enviou 1.537. Tras investigar os datos *SQL* almacenados descubrín a existencia dun *builbot*, o que dá credibilidade ao feito de 14 persoas teren enviado tal cantidade de mensaxes.

É importante repararmos en que Guido enviou mensaxes á lista desde polo menos 3 enderezos de correo electrónico diferentes.

Se botarmos unha ollada ao número de mensaxes enviadas á lista desde 1999 descubriremos que a tendencia é variable. O ano de menos mensaxes é de novo 1999, xa que é o ano en que se creou a lista.

Tamén veremos que 2002 foi o ano de maior número de mensaxes, cun total de 12.447. Descoñezo a causa, xa que a versión lanzada en 2002 foi secundaria.

## 7. Conclusións sobre Python

A presente obra mostrou ao lector unha análise do proxecto *Python*, ofreceulle a posibilidade de analizar os aspectos principais de Python, a información dispoñible na web sobre esta linguaxe, a súa historia, o seu modelo de negocio, as licenzas baixo as que é distribuída, etc.

Podemos concluír que *Python* é unha das principais linguaxes de programación que se poden atopar no universo do software libre na actualidade.

Non debemos esquecer quen está detrás da comunidade *Python*. *Python* conta co apoio dunha das maiores compañías dos nosos días, o que nos convida a pensar que *Python* é unha realidade. Existen multitude de exemplos na web de proxectos de éxito nos que *Python* é empregado no ambiente de produción día tras día, así como múltiples exemplos de como *Python* reduce o tempo de programación, os custos dos proxectos e a curva de aprendizaxe dos novos participantes que se unen a eses proxectos.

Esta obra é de gran axuda para quen precisar escoller unha linguaxe de programación para un novo proxecto. O estudo das métricas e de todo o relacionado co proxecto vaille axudar a tomar a decisión correcta.

## 8. Referencias

1. Analisis Python, Santiago Dueñas - GSyC/LibreSoft, Universidad Rey Juan Carlos (Madrid, España). <http://www.libresoft.es>.
2. Python homepage, The Python Foundation. <http://www.python.org>.
3. The Python Software Foundation, The Python Software Foundation. <http://www.python.org/fsf>
4. Ohloh.net : The Open Source Network, Jason Allen and Scott Collison. <http://www.ohloh.net>
5. Libresoft Tools, Libresoft. <http://www.libresoft.es>.
6. The Python Software Foundation: A primer look at, Dave Asher. <http://www.asher.ca>
7. The Perl Software Foundation, The Perl Software Foundation. <http://www.perl-foundation.org>
8. Wikipedia, The Free Encyclopedia, The Wikimedia Foundation. <http://www.wikipedia.org>
9. FlossMetrics Project, GSyC/LibreSoft Research Group. <http://www.flossmetrics.org>

| Year | Total of people |
|------|-----------------|
| 1999 | 6               |
| 2000 | 71              |
| 2001 | 234             |
| 2002 | 336             |
| 2003 | 333             |
| 2004 | 323             |
| 2005 | 325             |
| 2006 | 285             |
| 2007 | 282             |
| 2008 | 280             |
| 2009 | 218             |

*Tabela 3: número total de individuos que participan en python-dev*

| <b>Year</b> | <b>Total of people</b> |
|-------------|------------------------|
| 1999        | 6                      |
| 2000        | 125                    |
| 2001        | 528                    |
| 2002        | 793                    |
| 2003        | 860                    |
| 2004        | 833                    |
| 2005        | 931                    |
| 2006        | 929                    |
| 2007        | 804                    |
| 2008        | 797                    |
| 2009        | 549                    |

*Tabela 4: número total de individuos que participan en python-announce*

| <b>Version</b> | <b>C</b> | <b>Python</b> | <b>Other</b> | <b>Total</b> |
|----------------|----------|---------------|--------------|--------------|
| 1.0.1          | 52.615   | 30.109        | 1.543        | 84.267       |
| 1.1            | 53.304   | 45.515        | 1.674        | 100.493      |
| 1.2            | 61.215   | 74.590        | 1.901        | 137.706      |
| 1.3            | 66.057   | 59.407        | 2.902        | 127.556      |
| 1.4            | 62.364   | 62.288        | 2.421        | 127.073      |
| 1.5            | 89.890   | 76.070        | 2.828        | 168.788      |
| 1.6            | 186.201  | 103.268       | 5.932        | 295.401      |
| 1.6.1          | 186.206  | 103.268       | 5.932        | 295.406      |
| 2.0            | 130.603  | 113.793       | 6.036        | 250.432      |
| 2.1            | 131.437  | 129.714       | 6.359        | 267.510      |
| 2.2            | 213.259  | 198.544       | 6.796        | 418.599      |
| 2.3            | 265.082  | 244.766       | 9.253        | 519.101      |
| 2.4            | 304.233  | 268.782       | 5.616        | 578.631      |
| 2.5            | 353.790  | 317.518       | 17.813       | 689.121      |
| 2.5.2          | 354.616  | 320.812       | 18.852       | 694.280      |
| Rev. 63.637    | 355.104  | 321.624       | 17.744       | 753.917      |
| 2.5.4          | 355.104  | 321.624       | 18.858       | 695.586      |
| 2.6.2          | 387.552  | 370.235       | 17.782       | 775.569      |
| 3.0.1          | 296.215  | 271.109       | 17.807       | 585.131      |
| 3.1            | 311.554  | 284.457       | 17.662       | 613.673      |

*Tabela 5: número total de liñas das distintas linguaxes de programación*

| <b>Domain</b>         | <b>Number of mails</b> |
|-----------------------|------------------------|
| python.net            | 1752                   |
| comcast.net           | 1706                   |
| lemburg.com           | 1584                   |
| acm.org               | 1343                   |
| zope.com              | 1335                   |
| telecommunity.com     | 1332                   |
| pythonware.com        | 1154                   |
| rcn.com               | 1109                   |
| canterbury.ac.nz      | 1085                   |
| pythoncraft.com       | 1076                   |
| digicool.com          | 983                    |
| home.com              | 980                    |
| cosc.canterbury.ac.nz | 913                    |
| udel.edu              | 643                    |
| activestate.com       | 641                    |
| gmx.net               | 633                    |
| egenix.com            | 605                    |
| interlink.com.au      | 595                    |
| yahoo.com             | 523                    |
| alum.mit.edu          | 517                    |
| xs4all.net            | 514                    |
| pitrou.net            | 504                    |

*Tabela 6: domínios de maior actividade em python-dev*

| <b>Year</b> | <b>Total of e-mails</b> |
|-------------|-------------------------|
| 1999        | 666                     |
| 2000        | 2833                    |
| 2001        | 7824                    |
| 2002        | 12447                   |
| 2003        | 9516                    |
| 2004        | 7517                    |
| 2005        | 5810                    |
| 2006        | 8391                    |
| 2007        | 5122                    |
| 2008        | 8508                    |
| 2009        | 5778                    |

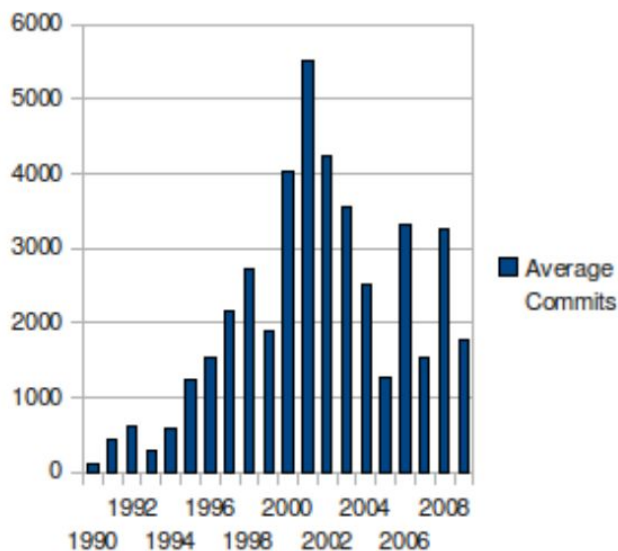
*Tabela 7: total de correos recibidos en python-dev*

| <b>Language</b> | <b>Number of lines</b> | <b>%</b> |
|-----------------|------------------------|----------|
| Python          | 311.554                | 50.77    |
| C               | 284.457                | 46.35    |
| Asm             | 9398                   | 1.53     |
| Sh              | 4.541                  | 0.74     |
| Lisp            | 2.948                  | 0.48     |
| Objc            | 775                    | 0.13     |

*Tabela 8: linguaxes de programación empregadas na última versión de Python*

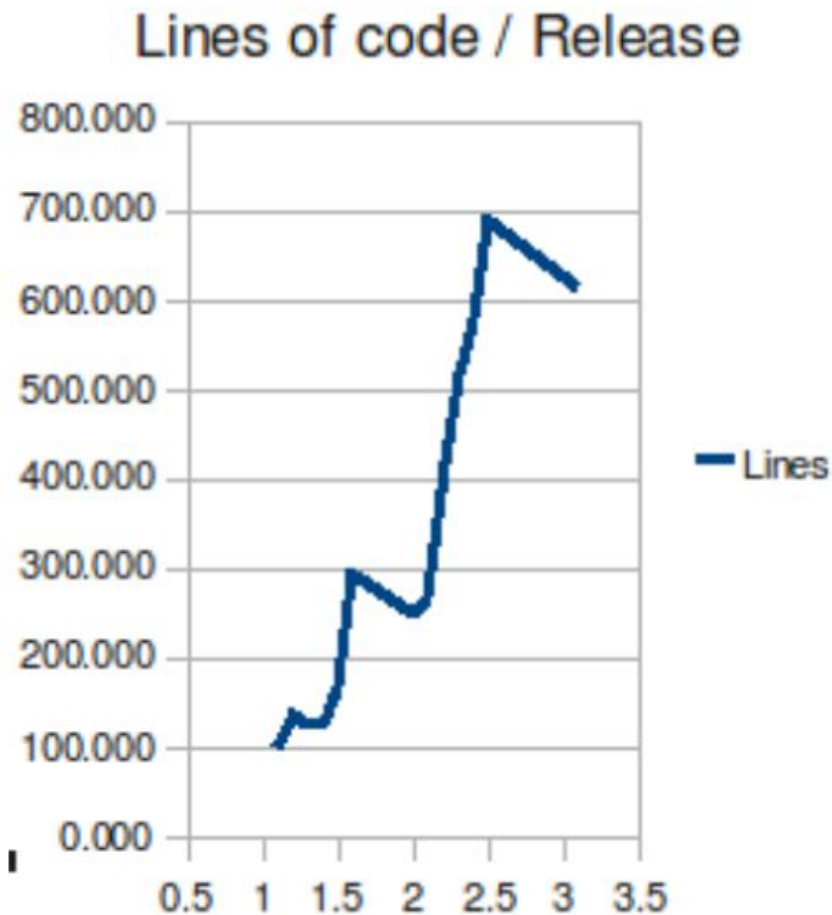
|                                         |                                                           |
|-----------------------------------------|-----------------------------------------------------------|
| <b>Web page:</b>                        | <a href="http://www.python.org">http://www.python.org</a> |
| <b>Current version:</b>                 | 3.1                                                       |
| <b>Current stable version:</b>          | 3.1                                                       |
| <b>Last stable version:</b>             | 2.6.2                                                     |
| <b>License:</b>                         | Python Software Foundation 2.6.2                          |
| <b>Slocs:</b>                           | 613.673                                                   |
| <b>Number of files:</b>                 | 3.608                                                     |
| <b>Effort stimation (person-year):</b>  | 169,19                                                    |
| <b>Effort stimation (person-month):</b> | 2,023                                                     |
| <b>Stimated cost:</b>                   | 22.854.698                                                |
| <b>Average number of developers:</b>    | 44,95                                                     |

*Tabela 9: dados xerais sobre Python*



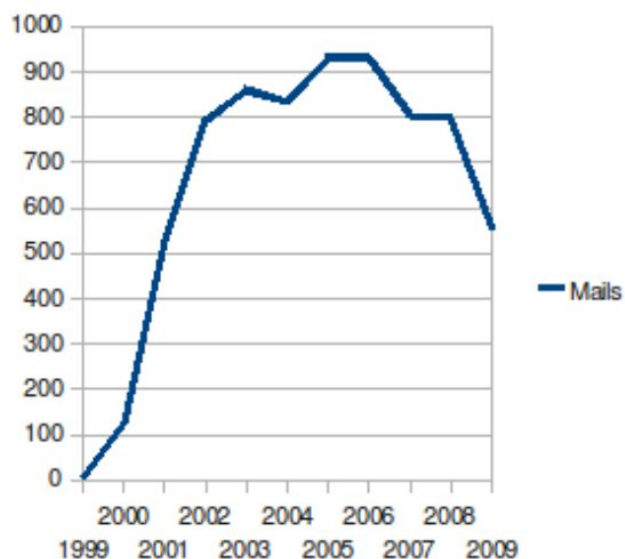
*Ilustración 1: media de commits desde 1990*



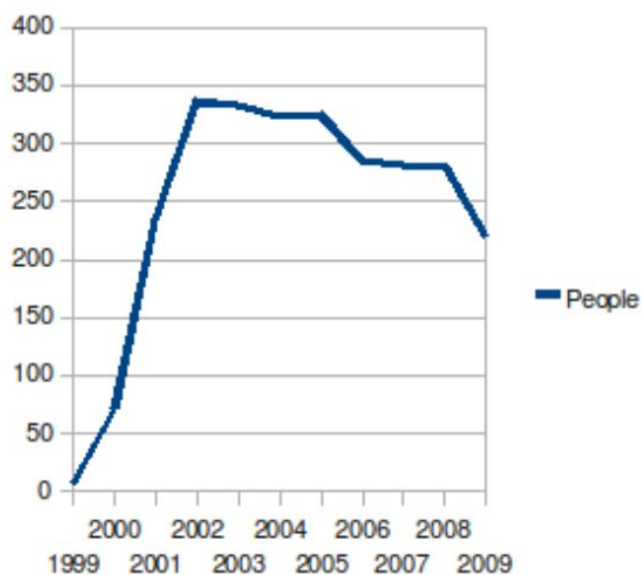


*Ilustración 2: evolución das linguaxes de programación nas versións de Python*

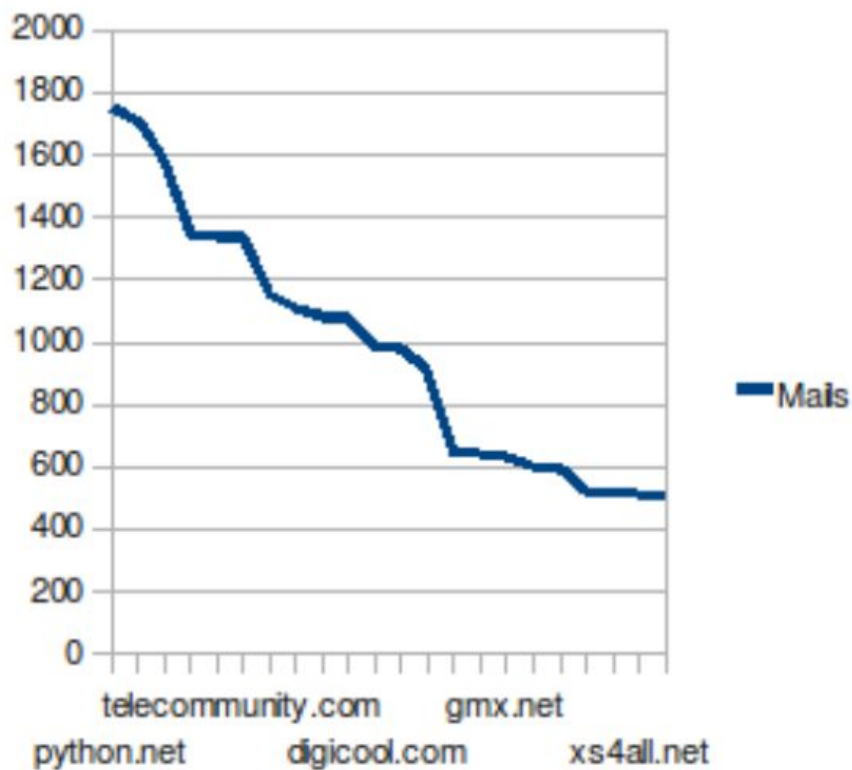




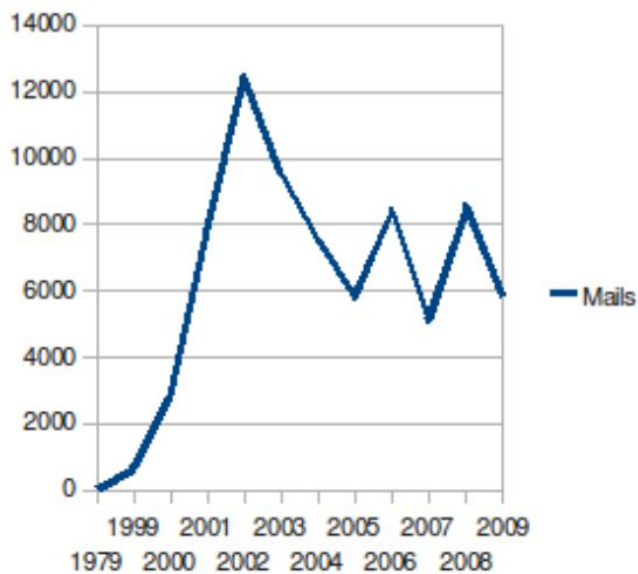
*Ilustración 4: número total de correos recibidos en python-announce*



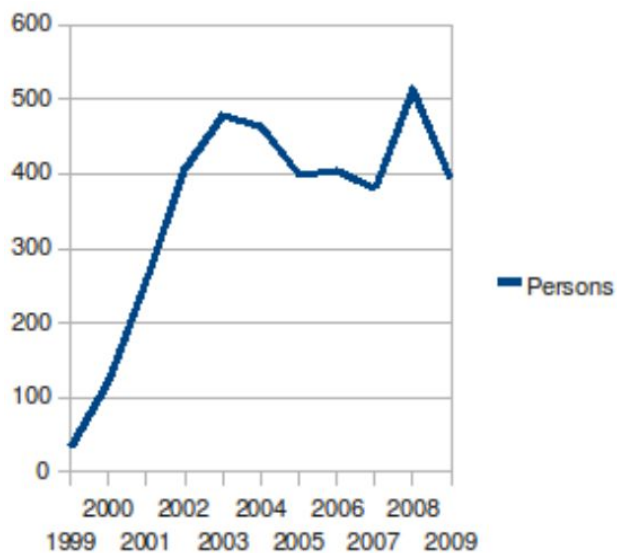
*Ilustración 5: total de individuos que participan en python-announce*



*Ilustración 6: dominios más activos en python-dev*



*Ilustración 7: número total de correos recibidos en python-dev*



*Ilustración 8: total de individuos que participan en python-dev*

# Participar e traballar con software libre

Javier Jardón Cabezas

`jjardon@gnome.org`

© 2009 Javier Jardón Cabezas

Os contidos desta obra están licenciados baixo unha licenza atribución – partilla nos mesmos termos de Creative Commons. Para consultares unha copia da licenza completa visita a páxina <http://creativecommons.org/licenses/by-sa/3.0/> ou escribe a este enderezo: Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

## **Resumo**

O presente documento é un resumo das actividades realizadas durante o Mestrado en software libre organizado por Caixanova e Igalia. Todas elas foron levadas a cabo cos coñecementos adquiridos durante as clases do mestrado.

O documento está dividido en dúas partes. A primeira trata da miña cooperación con comunidades de software libre. A segunda parte trata do labor que realicei durante as miñas prácticas en Igalia, empresa galega de software libre.

# Parte I: Comunidade

## 1.1 Primeiras contribucións

Nesta sección vou describir a miña colaboración nalgúns comunidades de software libre. Un dos meus obxectivos no mestrado é aprender a colaborar na comunidade de maneira máis intensa. Anteriormente xa colaborara en calidade de tradutor no proxecto *Trasno*<sup>12</sup> e tamén participara en tarefas de soporte e tradución na distribución Ubuntu, mais desexaría intensificar aínda máis a miña colaboración.

### 1.1.1 TexMaker

Traducín este editor *TEX* ao galego e ao español como parte da primeira materia do mestrado (*Introdución ao software libre*). As traducións foron aceptadas no proxecto e están dispoñibles na versión actual do programa (máis información na páxina principal<sup>13</sup>).

### 1.1.2 Maemo

O meu primeiro contacto con esta comunidade prodúcese grazas ás *tablets* que Nokia doou aos estudantes do mestrado. Pensei que sería interesante facer unha pequena triaxe de bugs na base de datos de *Maemo* en *bugzilla*.

Logo de triar varios bugs, os administradores déronme permisos para confirmar bugs e para editar todos os campos de bugs no *bugzilla* de *Maemo*.

---

<sup>12</sup> <http://www.trasno.net/>

<sup>13</sup> <http://www.xmlmath.net/texmaker/>



## 1.2 Gnome

Cando comecei a afondar no coñecemento das tecnoloxías *Gnome* e *GTK+*, decidín colaborar coa área de traballo *Gnome* ou con algún proxecto relacionado con *GTK+*. As clases *git* tiñan especial relevancia neste punto porque o proxecto *Gnome* migrara de *Subversion* ao *Git* a comezos de ano. Para alén diso, *Git* é unha ferramenta excepcional que facilita o proceso de desenvolvemento a todos os colaboradores.

### 1.2.1 Primeiras contribucións

Comecei co proxecto *GnomeLove*<sup>14</sup> na procura dunha tarefa simple para novatos. Comecei a limpar e actualizar páxinas na *wiki* de *Gnome*, xeralmente indicando onde ficaban os novos repositorios de *git* e actualizando a formación como mellor podía. Un bo exemplo disto é a páxina de avaliación de *Canvas GTK+*<sup>15</sup>.

### 1.2.2 Gnome3

Comecei a colaborar na transición a *Gnome3* mediante os proxectos *GnomeGoals*. Explicarei o meu labor en *GnomeGoals* na sección 2.2.2. Os principais obxectivos<sup>16</sup> da nova versión de *Gnome* son:

- Renovar a experiencia de usuario
- Tornar a plataforma máis eficiente
- Promocionar *Gnome*

Eu colaboro no segundo punto, retirando da plataforma todas as bibliotecas obsoletas para evitar que se empreguen no código novo. Existe un proxecto relacionado con esta actividade, chamado *ProjectRidley*<sup>17</sup>, que ten como obxectivo a consolidación en *GTK+* dunha serie de bibliotecas externas.

#### 1.2.2.1 Proxecto Ridley: obxectivo GTK+ 3

O proxecto *Ridley* ten como obxectivo a consolidación en *GTK+* dunha serie de bibliotecas externas.

Estas bibliotecas costuman ser pequenas e defectuosas, e con frecuencia están mal mantidas. Non posúen un obxectivo definido (como

---

14 <http://live.gnome.org/GnomeLove>

15 <http://live.gnome.org/ProjectRidley/CanvasOverview>

16 <http://live.gnome.org/ThreePointZero/Plan>

17 <http://live.gnome.org/ProjectRidley/>

*libgnome* ou *libgnomeui*) ou estarían mellor en *GTK+* (como *libgnomeprint* ou *libgnomeprintui*).

Este traballo culminará no lanzamento de *GTK+* 3.

As bibliotecas obxecto de estudo son:

- *libgnome*
- *libgnomeui*
- *libgnomeprint*
- *libgnomeprintui*
- *libglade*
- *libgnomecanvas*
- *libegg*
- *libeel*
- *gtkglext*

Propuxen a inclusión de *DropLibsexy* en *ProjectRidley*<sup>18</sup>. A continuación explico o labor que realicei sobre **DropLibsexy**:

Algúns *widjets* da biblioteca *libsexy* están integrados en *Gtk+*, polo que algunhas aplicacións *GTK+* poden eliminar a dependencia de *libsexy*:

- Substitución da funcionalidade de ícones *SexyIconEntry* por *GtkEntry* (dispoñible en *GTK+* 2.16)
- Substitución da funcionalidade *SexyUrlLabel* por *GtkLabel URL* (dispoñible en *GTK+* 2.17.1)

Decidín crear varios remendos para todas as aplicacións oficiais (e non oficiais) de *Gnome*, un labor que está case concluído. Na páxina web<sup>19</sup> de *DropLibsexy* pódense ver os elementos que fican pendentes.

---

18 Os arquivos *gtk-devel-list* pódense consultar en <http://livegnome.org/archives/gtk-devel-list/2009-June/msg00002.html>

19 <http://livegnome.org/GnomeGoals/DropLibsexy>

### 1.2.2.2 GnomeGoals

Dado que non quería traballar nun proxecto específico, decidín que os proxectos *GnomeGoals*<sup>20</sup> eran un bo comezo. Neles, o colaborador non participa en proxectos concretos de corrección de bugs, senón que leva a cabo tarefas de corrección de bugs que afectan a toda a área de traballo. O colaborador aprende unha tarefa concreta que posteriormente implementa en todos os proxectos, un exercicio excelente para gañar experiencia. As tarefas costuman ser simples e breves, e existen indicacións a disposición do colaborador. Deste xeito, é posible participar sen comprometerse a fondo co proxecto, o que o torna nunha modalidade ideal para quen non quixer contribuír de maneira ininterrompida.

Posteriormente descubrín que a maioría dos obxectivos do *GnomeProject* son moi relevantes para o desenvolvemento de *GTK+* / *Gnome* 3.0.

Actualicei e limpei as páxinas e subpáxinas de *GnomeGoals*.

Tamén enviei algúns remendos e informes de erros.

Limpar *Glib* e *GTK+* inclúe:

- Remover símbolos obsoletos de *Glib* / *GTK+*
- Portar a *GIO*
- Empregar *GtkBuilder* en lugar de *libglade*

Tamén propuxen novos *GnomeGoals* e contribuíu a eles:

- Empregar *g\_timeout\_add\_seconds* en lugar de *g\_timeout\_add*
- Adicionar soporte de *GObject Introspection* á biblioteca
- Empregar a macro *GSEAL* para detectar accesos directos aos campos de obxecto e substituílos cunha chamada a unha función *accessor*
- Empregar un nome xenérico correcto nos ficheiros *.desktop*
- Migrar a *Gsettings* / *dconf* desde *Gconf*

### 1.2.3 GTK+ 3

Tamén comecei a traballar no desenvolvemento de *GTK+*.

---

20 <http://live.gnome.org/GnomeGoals>

Mentres estaba a traballar en *GSEAL GnomeGoal* descubrín que son necesarias unha serie de funcións *accessor* no propio *GTK+*. Comecei a elaborar informes de erros e a enviar remendos para crear algunhas desas funcións *accessor*.

Conforme se intensificaba a miña colaboración na comunidade *GTK+*, propuxen recuperar as xuntanzas de *GTK+*. Nestas xuntanzas decidiuse crear a rama *GTK+ 2.90* para traballar na futura versión 3.0, e comecei a colaborar nela. Na actualidade estou a cargo da tarefas de desenvolvemento de *GTK+ “Sealing structure fields”*<sup>21</sup>.

Tamén creei unha páxina web<sup>22</sup> cun itinerario de *GTK* para seguir o avance das novas funcionalidades das próximas versións de *GTK+*, e reactivei un proxecto co obxectivo de promover a corrección de erros en *GTK+* e *Glib*.

#### 1.2.4 Bugsquad

Tamén comecei a colaborar co equipo de *Bugsquad*.

Limpei e actualicei varias páxinas de *Bugsquad* na *wiki*<sup>23</sup> de *Gnome*. Despois de contribuír facendo triaxe de varios *bugs*, un membro do equipo deume permisos para editar todos os campos dos informes de erros. Tras conversar con algúns membros do equipo, reactivei as xuntanzas de *Bugsquad*<sup>24</sup>. Como resultado das devanditas xuntanzas, mudáronse algunhas das políticas de *bugzilla* para limpar a base de datos de *Gnome* en *bugzilla*.

A sección de estatísticas recolle a páxina co meu perfil de *bugzilla*.

#### 1.2.5 Integración na comunidade Gnome

Cando estaba a traballar nas tarefas relacionadas con *GnomeGoals* e *Bugsquad*, un dos membros da comunidade *Gnome* propuxo que se me nomease membro da Fundación *Gnome*. A petición foi aceptada unhas semanas máis tarde<sup>25</sup> :-).

Logo pedín unha conta de *commit* no repositorio *git* de *Gnome* despois de que algúns desenvolvedores así o suxerisen. A razón foi que estaba a

---

21 <http://live.gnome.org/GTK+/3.0/Tasks>

22 <http://live.gnome.org/GTK+/Roadmap>

23 <http://live.gnome.org/Bugsquad>

24 <http://live.gnome.org/Bugsquad/Meetings>

25 <http://foundation.gnome.org/membership/members.php>

enviar moitos remendos e deste xeito axilizaríase o proceso de facer *commit* dos mesmos.

Finalmente, dado que era membro da Fundación *Gnome* e tiña unha conta de *commit*, recibín un enderezo de correo electrónico de *Gnome*: **`jjardon@gnome.org`**.

### 1.2.6 Algunhas estatísticas

Estas son as miñas estatísticas a 23 de novembro de 2009; probablemente teñan mudado cando leres este documento :-).

Podes ler a páxina co meu perfil de *bugzilla*<sup>26</sup> a modo de resumo breve do meu labor na comunidade *Gnome*.

Repara en que fixen todo o traballo desde que comezou o curso de desenvolvemento en abril (hai 6 meses).

Velaquí algúns datos estatísticos:

- 190 bugs reportados
- 154 bugs con remendos
- fixen 1.079 comentarios sobre bugs
- pechei 166 bugs
- reportei 171 bugs
- sumei 18 puntos

A páxina Ohloh tamén fornece algunhas estatísticas:

---

<sup>26</sup> <https://bugzilla.gnome.org/page.cgi?id=describeuser.html&login=jjardon@gnome.org>

|                                                                                                  | Experience | Measured |
|--------------------------------------------------------------------------------------------------|------------|----------|
|  Autoconf       | 7m         | 7m       |
|  Automake       | 8m         | 5m       |
|  C              | 8m         | 8m       |
|  C++            | 2m         | 2m       |
|  HTML           | 1m         | 1m       |
|  Python         | 7m         | 1m       |
|  Vala           | 6m         | 2m       |
|  XML           | 4m         | 4m       |
|  shell script | 2m         | 2m       |

*Ilustración 1: estadísticas de Ohloh*

## Parte II: Modest

### 2.1 Introducción

*Modest* é un pequeno programa de correo electrónico orientado a dispositivos con recursos modestos, como computadores de escasa potencia e Nokia Internet Tablets N810/N900.

O meu labor noas prácticas que fixen en Igalia centrouse no cliente de correo electrónico *Modest*. O obxectivo das prácticas era portar *Modest* da plataforma *Maemo* a *Moblin* e tal vez actualizar tamén a versión de *Gnome*. Mais, que é *Maemo*? E *Moblin*? Velaquí unha breve introdución.

### 2.2 Maemo

*Maemo* é unha plataforma de software baseada principalmente en código aberto empregada en dispositivos móbiles como por exemplo nas Nokia Internet Tablets N810/N900.

A plataforma *Maemo* está baseada no núcleo do sistema operativo Linux. Linux é un núcleo monolítico compatible con multitude de plataformas de hardware, desde reloxos de pulso até enormes sistemas de servidores. Na actualidade, todos os dispositivos que funcionan coa plataforma *Maemo* posúen un chipset OMAP, que contén un procesador polivalente ARM e unha unidade DSP.

A arquitectura da interface de usuario de *Maemo 5* está baseada no *framework GNOME*, en especial no conxunto de *widgets* de *GTK+*. *GNOME* é un dos principais *frameworks* de aplicacións para sistemas de área de traballo GNU/Linux. A plataforma *Maemo* herdou multitude de compoñentes centrais como por exemplo *GTK+*, o *framework* multimedia *GStreamer*, o xestor de configuración *GConf* ou a biblioteca XML. A plataforma *Maemo* amplía as tecnoloxías *GTK+/GNOME* en canto fornece extensións para áreas de traballo móbiles.

#### **Hildon UI framework**

O *framework* da interface de usuario está baseado no sistema *X Window* co xestor de xanelas *Matchbox*. A API sobre X é o conxunto

ferramentas (*toolkit*) de *widgets* de *GTK+* coas chamadas extensións *Hildon*. *GTK+* é o *framework* da interface de usuario desenvolvido no proxecto *GNOME*. O *framework* da interface de usuario *Hildon* fornece compoñentes adicionais aos de *GNOME* para o panel de control, a barra de estado, o navegador de tarefas e os *home applets*.

A ilustración 2 mostra un diagrama da arquitectura.

## Maemo 5 platform rock stars

|              |                   |                        |
|--------------|-------------------|------------------------|
| GTK+         | Hildon            | Qt (community support) |
| Xorg         | OpenGL ES         | Clutter                |
| SQLite       | Tracker           | D-Bus                  |
| GUPnP        | MAFW              | Telepathy              |
| GStreamer    | PulseAudio        | V4L2                   |
| BusyBox      | GLib              | Gconf                  |
| BlueZ        | linuxwireless.org | OHM                    |
| Linux Kernel | HAL               | upstart                |

5 © 2009 Nokia maemo-formulator-gui-and-more.pdf / 2009-07-04 / Outin G8

**NOKIA**  
Connecting People

Ilustración 2: Maemo 5 “Fremantle”

### 2.2.1 Historia

Até hoxe fabricáronse diversas *Internet Tablets* con pantalla táctil equipadas coa plataforma *Maemo*. O primeiro dispositivo foi a Nokia 770 Internet Tablet, que saíu ao mercado o 25 de novembro de 2005. Coa Nokia 770 Internet Tablet era posible acceder a Internet a través dunha conexión sen fíos (WLAN).

O seguinte dispositivo en empregar a plataforma *Maemo* foi a Nokia 800 Internet Tablet, que funcionaba coa versión 3 de *Maemo*. Con ela podíanse facer chamadas a través de Skype/VoIP, conectarse a Internet sen fíos e estaba equipada cunha cámara integrada.



Outro paso adiante constituíuno a Nokia 810, que funcionaba con *Maemo* 4. Está equipada cun teclado QWERTY para alén da pantalla táctil, cun navegador baseado na tecnoloxía *Mozilla*, con soporte para *Google-Talk* e cun GPS.

*Maemo* 5 marcou un enorme avance na evolución deste sistema. Introduciu unha interface de usuario táctil completamente redeseñada, función de teléfono móbil e función multitarefa en tempo real no panel de *Maemo*. A Nokia N900 funciona con *Maemo* 5.

## Futuro

Tras a adquisición de *Qt* por parte de Nokia, a plataforma está a migrar para as tecnoloxías do *framework Qt*. O *framework* oficial da aplicación de *Maemo* 6, cuxo lanzamento está previsto para 2010, estará baseado en *Qt*. Espérase que sexa a comunidade quen manteña as bibliotecas *Hildon/GTK+*.

A ilustración 3 mostra un diagrama da futura arquitectura.

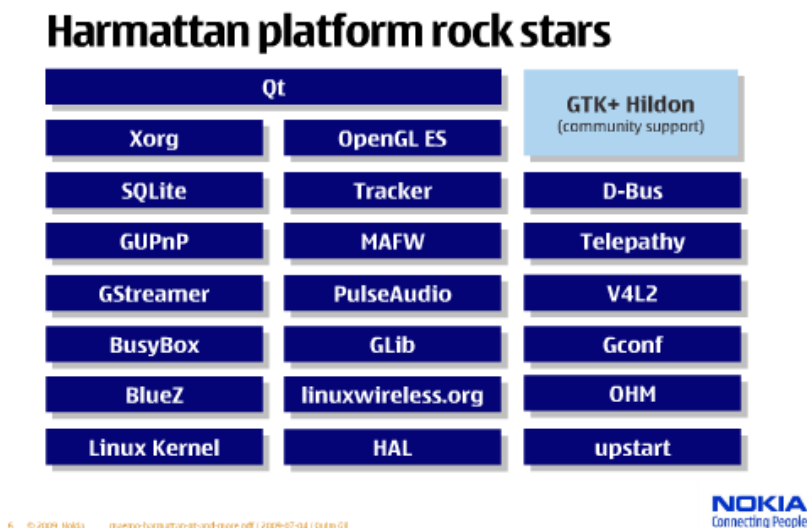


Ilustración 3: *Maemo* 6 “Harmattan”

## 2.3 Moblin

A arquitectura *Moblin* está deseñada para soportar multitude de plataformas e padróns de uso, desde *netbooks* e dispositivos de Internet móbiles, a sistemas integrados, como os sistemas de *infotainment* para automóviles. A parte central desta arquitectura é a chamada camada común ou *Moblin Core*, unha camada independente do hardware e do uso, que fornece unha plataforma común para desenvolver ese tipo de dispositivos. Por debaixo desta camada está o núcleo de Linux, así como os controladores de dispositivo específicos do hardware. Por enriba de *Moblin Core* están a interface específica de usuario e o modelo de interacción co usuario do dispositivo en cuestión.

*Moblin Core* fornece unha serie de bibliotecas de código aberto e de servizos de aplicacións, que axen como módulos altamente funcionais e que permiten crear dispositivos específicos con gran rapidez.

### Moblin Core

*Moblin Core* é o núcleo da plataforma de software *Moblin*. Fornece unha plataforma común altamente funcional que pode ser implementada en multitude de dispositivos, como *netbooks*, dispositivos de Internet móbiles ou mesmo automóviles. *Moblin Core* consta de bibliotecas de plataforma e de servizos de aplicación, polo que fornece módulos extremadamente funcionais para a creación de plataformas. Non fornece interfaces de aplicación de usuario nin interfaces de interacción co usuario. É unha implementación específica da plataforma, creada por *Moblin* ou por un terceiro, a que fornece esa funcionalidade.

*Moblin Core* está construído sobre a plataforma *GNOME Mobile*, que amplía e enriquece mediante tecnoloxías de código aberto como *Clutter*<sup>27</sup>, *GUPnP*<sup>28</sup> e tecnoloxías específicas para *Atom*, como gráficas, de arranque rápido e de aforro de enerxía.

*Moblin Core* seguirá o ciclo de lanzamento bianual de *GNOME Mobile* 6 (que comezará en 2009 coa 2.26). Ficaré sincronizado coa devandita plataforma, manterá a compatibilidade e contribuirá con melloras ás plataformas *GNOME Mobile* e *Desktop*. Espérase que *Moblin Core* medre e madure conforme se creen as implementacións de plataforma actuais, comezando con *Moblin NetBook*.

---

<sup>27</sup> <http://www.clutter-project.org/>

<sup>28</sup> <http://www.gupnp.org/>

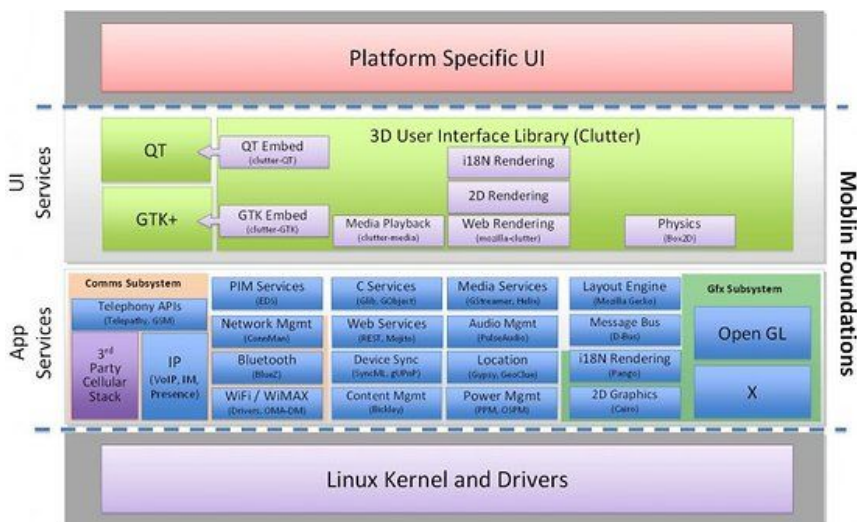


Ilustración 4: Moblin Core

### 2.3.1 Pasado e futuro

Intel lanzou o sitio *moblin.org* en xullo de 2007 e fixo unha actualización importante do mesmo en abril de 2008 co lanzamento da familia de procesadores Intel *Atom* no Intel Developer Forum de Shanghai. En abril de 2009 Intel traspasou *Moblin* á Fundación Linux. O desenvolvemento é supervisado na actualidade polo *Moblin Steering Committee*.

O SO *Moblin 2* foi deseñado especificamente para funcionar nun procesador Intel *Atom* dun *netbook*. Este segundo gran lanzamento (*Moblin 2.0*) representou unha mudanza da área de traballo *Xfce* a unha interface de usuario de *GNOME Mobile* creada adhoc e baseada no *clutter* de *OpenedHand*, unha peza chave do ambiente gráfico de *Maemo*, construído ao redor do sistema de xanelas *X*.

Futuras versións de *Moblin* funcionarán tamén en teléfonos móbiles.

### Nbtk

*Nbtk* é unha biblioteca (conxunto de ferramentas para *netbook*) para a implementación de elementos comúns de interface de usuario baseada en *Clutter*. Porén, *Nbtk* desenvolveuse seguindo obxectivos a moi curto

prazo para que o equipo de *Moblin* puidese rematar a interface de usuario *Moblin* 2.0 o antes posible.

Agora que *Moblin* 2.0 (e tamén 2.1) xa foi lanzado, o equipo de *Moblin* está a desenvolver un conxunto de ferramentas máis serio.

## **Mx**

O primeiro obxectivo foi poñerlle á biblioteca *Ntk* un nome menos intimamente ligado ás plataformas para “*netbook*”. Escolleron *mx* como nome de espazo, xa que o conxunto de ferramentas está orientado a mellorar a experiencia de usuario de *Moblin*. *Mx* fornece un conxunto de elementos estándar de interface de usuario, entre outros, botóns, barras de progreso, *tooltips* e barras de rola. Tamén implementa unha serie de xestores de espazo (*layout managers*) estándar, embora algúns deles vaian estar dispoñibles no propio *Clutter* cando sexa lanzada a nova versión 1.2. Outra funcionalidade interesante é a posibilidade de configurar propiedades de estilos desde un ficheiro similar a un *.css*.

### **2.3.2 Ambiente de desenvolvemento de Moblin**

Dado que *Moblin* é unha creación moi recente, non se sabía con claridade cal era o proceso de desenvolvemento de aplicacións para esta plataforma. O método estaba a mudar rapidamente en cuestión de poucos meses.

## **Imaxe de desenvolvemento ISO**

Intel ofrecía unha imaxe de desenvolvemento para as versións de *Moblin* anteriores á 2 baseada na área de traballo *Xfce*, que permitía desenvolver aplicacións ben instalando a devandita imaxe nun *netbook* ou ben empregando algún ambiente de virtualización. Este método xa non se emprega na actualidade porque *Moblin* 2 mudou para un ambiente totalmente novo baseado en *Clutter* e porque existen unha serie de problemas cos gráficos tridimensionais en ambientes virtuais.

## **Jhbuild**

A seguinte estratexia consistiu en empregar *jhbuild*<sup>29</sup> para descargar pacotes específicos de *Moblin*. A seguir, podíaase empregar *Xephyr*<sup>30</sup> para desenvolver aplicacións nun ambiente *Gnome* normal. Esta estratexia xa

---

29 <http://live.gnome.org/Jhbuild>

30 <http://www.freedesktop.org/wiki/Software/Xephyr>

non se pon na práctica porque o ambiente *Xephyr* tamén presenta problemas coa aceleración 3D.

### **A estratexia que seguín nas prácticas**

Existen unha serie de maneiras de configurar o ambiente de desenvolvemento, das que describirei dúas:

- Desenvolver na estación de traballo Linux realizando validacións remotas ocasionais e tarefas de depuración de erros nun dispositivo de destino
- Desenvolver nun ambiente *Moblin* v2 empregando, por exemplo, un *netbook* como plataforma.

*Moblin* suxire desenvolver en ambos tipos de ambientes.

A maior parte do desenvolvemento pode ser feito nunha distribución recente e estendida, como por exemplo Ubuntu 8.04, Fedora 9, equipadas coas ferramentas de desenvolvemento estándar. Unha das tarefas principais consiste en aliñar as versións das bibliotecas de desenvolvemento que precisa a aplicación que se desexa desenvolver coas que están recollidas na lista de bibliotecas de *Moblin*<sup>31</sup>. A maior parte das tarefas de desenvolvemento da aplicación, de compilación, de execución e de depuración de erros levaranse a cabo neste ambiente.

De cando en vez (de preferencia nas primeiras fases) é recomendable executar a aplicación nun ambiente *Moblin* v2 (un *netbook*). Para alén diso, é moi probable que o desenvolvedor desexe aprender a depurar a súa aplicación de forma remota executándoa nun *netbook* ou nun ambiente virtual desde a súa estación de traballo.

### **Estratexia actual**

Esta é a estratexia actual (en novembro de 2009) para configurar un ambiente de construción para compilar e depurar aplicacións de *Moblin*. Os pasos son os seguintes:

- Configurar a estación de traballo Linux
- Descargar e descomprimir o kit de desenvolvemento de software (*SDK*)

---

31 <http://moblin.org/documentation/moblin-sdk/moblin-v2-development-libraries>

- Instalar unha imaxe de *Moblin* 2.1 no *netbook*
- Configurar o *IDE* e o ambiente de construción
- Comezar a depurar erros

O obxectivo final vai ser depurar a aplicación como se mostra embaixo:

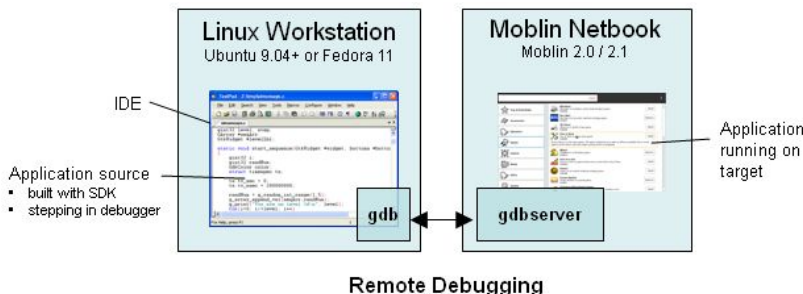


Ilustración 5: ambiente de desenvolvemento de Moblin

## Moblin SDK Sysroot

O *SDK* de *Moblin* emprega un *sysroot* para o desenvolvemento. É un directorio que contén o sistema operativo *Moblin* e tamén arquivos cabezallo (*header files*), bibliotecas e información do pacote. Permite executar *Moblin* na estación de traballo, construír e depurar código no editor que o desenvolvedor escoller e depurar erros de maneira remota.

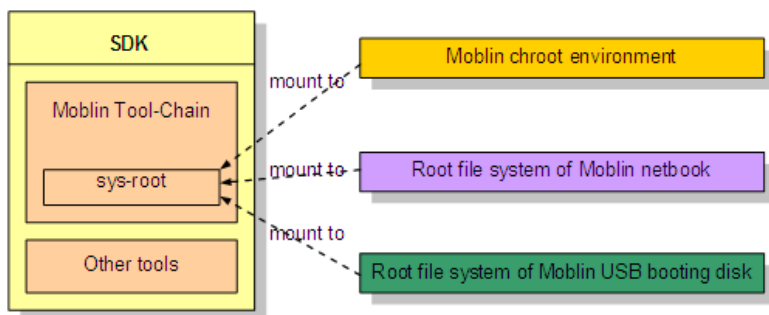


Ilustración 6: Sysroot do SDK de Moblin

Empregando o *sysroot* coa imaxe de desenvolvemento de *Moblin*, os desenvolvedores teñen a posibilidade de facer as seguintes tarefas:

- Actualizar os contidos do *sysroot* no *tarball* do *SDK* de *Moblin*
- Montar o ambiente *chroot* de *Moblin* no *sysroot*
- Montar o sistema de arquivos raíz de *Moblin* para *netbook* a través de *sshfs* en *sysroot*
- Montar o sistema de arquivos raíz do disco de arranque USB de *Moblin* en *sysroot*

## 2.4 Modest

*Modest* é un cliente de correo electrónico gratuito e de código aberto desenvolvido polo proxecto *Maemo* de Nokia. Na actualidade está a ser desenvolvido polos *hackers* de Igalia.

O beta público do *Modest* foi lanzado o 11 de decembro de 2007 con actualizacións semanais até o primeiro lanzamento oficial como parte do lanzamento *Diablo* do SO *Internet Tablet* 2008. Substituíu o cliente de correo-e *osso-email*.

Na actualidade, *Modest* require o *SDK* de *Maemo 5* para Nokia *Internet Tablets*. É posible que outras versións máis antigas (*Diablo*) tamén funcionen, mais non foron testadas recentemente.

### 2.4.1 Arquitectura

Cando comecei a colaborar con *Modest* existían varias versións: *gnome*, *Maemo* (para versións de *Maemo*  $\geq 4$ ) e *hildon2* (para *maemo5*, *Fremantle*). É posible compilar a versión desexada cos parámetros *./configure*. *Modest* está baseado no *framework* de correo electrónico *Tinymail*.

#### 2.4.1.1 Tinymail

*Tinymail* é unha biblioteca para o desenvolvemento de aplicacións móbiles con funcionalidade de correo electrónico. O *framework* fornece compoñentes que axudan ao desenvolvedor coa interface de usuario mais tamén coa conectividade e a almacenaxe temporal de mensaxes.

Implementa unha serie de compoñentes estándar de interface de usuario que poden ser empregados logo despois da instalación. Ofrece soporte inmediato para *Maemo*, *GPE*, *OLPC* e *Gnome*.

### Arquitectura

**A camada de servizo** (*service layer*). A camada de servizo abstrae o acceso cos servizos de correo electrónico e é implementada por defecto empregando un *Camel* modificado.

O *Camel* que emprega *Tinymail* está embebido no procedemento de compilación e foi adaptado para consumir menos memoria e menos largo de banda.

**A camada de interface de usuario.** A camada de interface de usuario abstrae e implementa funcionalidades relacionadas coa interface de usuario. As implementacións estándar de *tinymail* implementan compoñentes da interface de usuario mediante o conxunto de ferramentas *Gtk+*. Existe soporte específico para dispositivos como os *laptops* da iniciativa “Un *laptop* por neno” ou dispositivos baseados na plataforma *Maemo* como as Nokia 810/900.

#### 2.4.2 Labor realizado

O primeiro paso foi substituír o ambiente de desenvolvemento. Na sección 3.2.2 explícase como configurar o ambiente de desenvolvemento de *Moblin*. Subín unha serie de remendos para actualizar os repositorios de *modest* e *tinymail* en *jhbuild*.

Despois tentei compilar *modest* e *tinymail*, pero xurdiron unha serie de problemas:

- *Tinymail* tenta compilar sempre con soporte para *NetworkManager*
- *Modest* e *Tinymail* empregan bibliotecas obsoletas, que non están incluídas nas imaxes oficiais de *Moblin* nin nas de *GNOME3*

Todos os remendos que fixen están dispoñibles no *bugzilla* de *Maemo* para os bugs de *modest* e no *bugzilla* de *gnome* para os de *tinymail*. Reportei uns 20 bugs de *modest* e uns 12 de *tinymail*, para os que fixen os remendos correspondentes.



### 2.4.2.1 Tornar algunhas bibliotecas optativas

#### **Tinymail**

Fixen un remendo para tornar optativo o soporte de *NetworkManager*<sup>32</sup>. A razón é que *moblin* ten o seu propio xestor de rede, *ConnMan*<sup>33</sup>, un novo xestor de rede patrocinado por Intel e Nokia.

Para facelo modifiquei os arquivos das *autotools* necesarios.

#### **Modest**

Dado que o desenvolvemento de *modest* está intimamente relacionado coa plataforma *Maemo*, ás veces non podo compilar a aplicación porque o desenvolvemento principal faise no *Maemo SDK*. Solucionouse subindo remendos.

### 2.4.2.2 Depuración de bibliotecas

É preciso eliminar as bibliotecas obsoletas de *modest* e *Tinymail* para que funcionen axeitadamente en ambientes *moblin* e/ou *GNOME3*. O problema é que tanto *Tinymail* como *modest* empregaban moitas desas bibliotecas e símbolos obsoletos, que posúen multitude de dependencias indirectas, polo que eran precisas multitude de bibliotecas para compilar *modest* e *Tinymail*.

Contactei algúns desenvolvedores de *Gnome* para que se creasen esforzos orientados a eliminar as bibliotecas obsoletas de *Tinymail* e *modest*<sup>34</sup>.

#### **Tinymail**

*Tinymail* empregaba algunhas bibliotecas e símbolos obsoletos:

- *libgnome*
- *libgnomeui*
- *libglade*
- *libgnomevfs*
- Algúns símbolos obsoletos de *glib* e *gtk+*

---

32 <http://projects.gnome.org/NetworkManager/>

33 <http://connman.net/>

34 <http://www.gnome.org/fpeters/299.html>

- Uso de funcións *accessor* en lugar de acceso directo para tornar o código compatible con *GTK+3*

Todos estes problemas xa foron resolto.

**Gio port.** Substituír *libgnomevfs* foi o máis complicado, xa que a documentación non está completa.

Outro dos problemas radica na implementación de *GnomeVfs streams* e a nova *api*: o *Gio stream*. Na *API* antiga (*GnomeVfs*), todos os *streams* eran bidireccionais, en *GIO* xa non é así. En *Gio* existe unha clase para os *streams* de entrada e outra clase para os *streams* de saída.

Logo de investigar, e coa axuda da comunidade de *tinymail*, fixen un remendo<sup>35</sup> para portar *tinymail* a *Gio*.

### Modest

*Modest* tamén empregaba algunhas bibliotecas e símbolos obsoletos:

- *libgnomeui*
- *libgnomevfs*
- Algúns símbolos obsoletos de *glib* e *gtk+*
- Algúns símbolos obsoletos de *hildon*
- Uso de funcións *accessor* en lugar de acceso directo para tornar o código compatible con *GTK+3*

Todos estes problemas xa foron resolto, mais algúns dos remendos aínda teñen que ser revisados antes de seren introducidos no código de *modest*.

#### 2.4.2.3 Problemas

Decatámonos de que a versión *Gnome* de *modest* era moi incompleta.

Unha das ideas pasaba por empregar a versión *hildon2* de *modest* (a que ten maior soporte) en todas as plataformas. Pero esta estratexia presenta un problema, xa que nin *Gnome* nin *Moblin* inclúen bibliotecas *hildon*.

---

35 [https://bugzilla.gnome.org/show\\_bug.cgi?id=594234](https://bugzilla.gnome.org/show_bug.cgi?id=594234)

Así que decidimos que a mellor solución sería ter un código común xunto con unha factoría de *widgets* para cada plataforma. Os *hackers* de Igalia están a traballar nesta estratexia.

#### 2.4.2.4 Futuro

Velaquí algunhas ideas para o desenvolvemento futuro de *modest*:

- **Mer**: unha versión operativa de *modest* para esta distribución de *Maemo*, aberta e administrada en comunidade, sería de grande interese (de feito empregaban *modest* como cliente de correo-e principal mais mudaron para *claws* despois de sufrir problemas de compilación). Unha vez rematadas as tarefas de depuración non debería ser complicado. Pódese consultar máis información na páxina web do proxecto<sup>36</sup>.
- **Mx widgets**: con este conxunto de *widgets* pódese integrar a interface de usuario de *modest* en *Moblin* para *netbooks*. Aínda temos que esperar o primeiro lanzamento desta biblioteca de *widgets* baseada en *clutter*.

---

36 <http://wiki.Maemo.org/Mer>

### 3. Bibliografía

1. Proxecto GNOME: <http://www.gnome.org/>
2. Proxecto Ridley: <http://live.gnome.org/ProjectRidley>
3. Proxecto GnomeGoals: <http://live.gnome.org/GnomeGoals>
4. Equipo Bugsquad: <http://live.gnome.org/Bugsquad>
5. Conxunto de ferramentas de GTK+: <http://www.gtk.org/>
6. Cliente de correo-e Modest: <http://modest.garage.maemo.org/>
7. Plataforma Maemo: <http://maemo.org/>
8. Proxecto Moblin: <http://moblin.org/>
9. A miña páxina wiki: <http://live.gnome.org/JavierJardon>
10. Perfil de bugzilla: <http://bugzilla.gnome.org/describeuser.cgi?login=javierjc1982%40gmail.com>

# Rich Internet Applications (RIA): comparativa de frameworks en Software Libre

Beatriz Montero Fernández

© 2009 Beatriz Montero Fernández

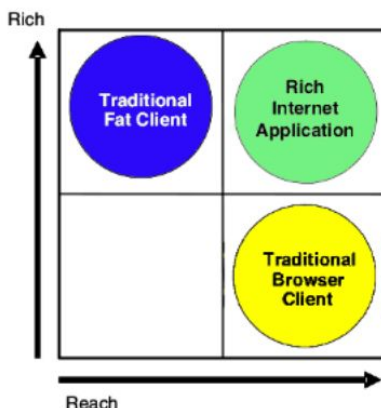
Os contidos desta obra están licenciados baixo unha licenza atribución – partilla nos mesmos termos de Creative Commons. Para consultares unha copia da licenza completa visita a páxina <http://creativecommons.org/licenses/by-sa/3.0/> ou escribe a este enderezo: Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

## Resumo

Esta obra mostra un percorrido polas diversas alternativas en Software Libre dispoñibles na actualidade para o desenvolvemento de *Rich Internet Applications*. Farase unha exploración en profundidade dunha das alternativas expostas: GWT (*Google Web Toolkit*), expondo as súas principais características mediante o desenvolvemento dunha pequena aplicación. Por último, faremos unha breve análise da comunidade que está detrás deste *framework*.

# 1. Introducción

O acrónimo RIA, de *Rich Internet Applications* (Aplicacións Ricas de Internet), foi concibido en 2002 por Jeremy Allaire de Macromedia, agora Adobe. É un termo que abrangue unha nova xeración de aplicacións que combinan as propiedades das aplicacións de área de traballo tradicionais coas propiedades das aplicacións Web. Isto quere dicir, unha RIA combina a rica interactividade dunha aplicación de área de traballo co amplo alcance dunha aplicación web. Hoxe en día considérase como unha tecnoloxía que forma parte da Web 2.0.



*Ilustración 1: representación visual dunha aplicación RIA. Nesta representación podemos observar como a unificación dunha aplicación de área de traballo (Fat client) e as tradicionais aplicacións web (browser client) permiten obter un novo tipo de aplicación coñecida como RIA*

A descrición gráfica desta definición móstrase na ilustración 1, onde podemos observar a unificación entre as aplicacións de área de traballo e as aplicacións web. A seguir cítanse as principais características que debe cumprir unha aplicación RIA:

- Interface de usuario máis rica, tornando máis fácil a interacción co usuario final, incluíndo funcionalidades como arrastrar, pegar e cálculos no lado do cliente sen a necesidade de enviar a información ao servidor.

- Comunicación asíncrona. Non se producen recargas de páxina, xa que desde o principio cárgase a páxina completa e logo vanse actualizando partes da mesma, e só se produce comunicación co servidor cando se necesitan datos externos como datos dunha base de datos ou doutros ficheiros externos.
- Redución da carga do servidor, xa que parte do procesamento trasládase ao cliente.
- Redución do tráfico de rede, xa que o cliente asume parte do procesamento e non é necesario realizar tantas peticións ao servidor.

Alguns exemplos de aplicacións RIA existentes na rede na actualidade son:

- Visualización de mapas con Goolge Maps ou Yahoo! Maps ou Mapas de Live Search
- Xestión de *feeds* con Google Reader
- Xestión de correo con Gmail
- Edición de documentos con GoogleDocs
- Etherpad, procesador de texto colaborativo
- Bepin, editor de ficheiros de código de Mozilla

## 2. Tecnoloxías RIA

Na actualidade existen multitude de tecnoloxías (*frameworks*, bibliotecas...) que nos axudan a desenvolver aplicacións RIA e que podemos encadrar en dous grupos:

- Baseadas en *plugins* ou *sandboxes*
- Baseadas en estándares soportados nativamente polos navegadores (DOM, Javascript, DHTML)



## 2.1 Tecnoloxías baseadas en *plugins* ou *sandboxes*

Este tipo de tecnoloxías necesitan un engadido instalado no cliente para poderen fornecer a potencialidade típica das RIA.

### Vantaxes

- Este tipo de tecnoloxías caracterízanse por forneceren uns efectos de usuario extraordinarios, especialmente ao manexarmos contido multimedia.
- Os desenvolvedores séntense cómodos ao implementaren unha aplicación con este tipo de tecnoloxías, xa que permiten realizar un desenvolvemento rápido de aplicacións cuns resultados moi rechamantes.
- Son as que teñen maior cota de mercado.

### Inconvenientes

- Obrigan ao cliente a instalar complementos no navegador para poder visualizalas.
- Non son independentes da plataforma, xa que os *plugins* dependen tanto do sistema operativo como do propio navegador.
- Non son accesibles, xa que non están integradas con HTML.

A seguir lístanse algunhas destas tecnoloxías:

- **Adobe Flex.** É a solución de Adobe para dar soporte ao despregamento e desenvolvemento de Aplicacións Ricas de Internet. Nun desenvolvemento multicamada, as aplicacións Flex correspóndense coa camada de presentación, mais permiten comunicarse co servidor (lóxica e modelo de datos) a través de servizos web XML e obxectos remotos (tales como Coldfusion CFCs, propio de Adobe mais privativo, clases Java, e calquera que soportar o formato de mensaxes de accións).

|                |                                   |
|----------------|-----------------------------------|
| <b>Licenza</b> | MPL                               |
| <b>Plugin</b>  | Adobe Flash Player<br>(Privativo) |

|                             |                                    |
|-----------------------------|------------------------------------|
| <b>Versión</b>              | 3.0.0                              |
| <b>Linguaxe do frontend</b> | MXML                               |
| <b>Linguaxe do backend</b>  | Java, PHP, .NET...                 |
| <b>Plataformas</b>          | GNU/Linux, Mac, Microsoft, Solaris |

*Cadro 1: ficha Adobe Flex*

- **Sun Java FX.** É a resposta de Sun Microsystems a Adobe (Adobe Flex) e a Microsoft (Silverlight) para o desenvolvemento de aplicacións RIA, coa vantaxe de que permite realizar desenvolvementos para PCs e dispositivos móbiles e permite executar as aplicacións fóra do navegador, sen necesidade de instalar ningún complemento adicional. JavaFX inclúe JavaFX SDK, un compilador e ferramentas en tempo de execución, gráficos, medios, servizos web e bibliotecas de texto ricas, Netbeans IDE for JavaFX, un ambiente de desenvolvemento completo e integrado e, para os deseñadores, JavaFX Production Suite, unha suite con *plugins* e ferramentas para exportar outros compoñentes gráficos (desde GIMP, Photoshop, ...) ás aplicacións JavaFX.

|                             |                                    |
|-----------------------------|------------------------------------|
| <b>Licenza</b>              | Parte en GPLv2                     |
| <b>Plugin</b>               | JRE Java                           |
| <b>Versión</b>              | 1.2                                |
| <b>Linguaxe do frontend</b> | JavaFX                             |
| <b>Linguaxe do backend</b>  | Java                               |
| <b>Plataformas</b>          | GNU/Linux, Mac, Microsoft, Solaris |

*Cadro 2: ficha Java FX*

- **Open Laszlo.** É unha plataforma completamente software libre para o desenvolvemento de aplicacións RIA para PC e móbiles. OpenLaszlo inclúe a linguaxe de *scripting* LZX e o Servidor OpenLaszlo, que é un *servlet* Java. A diferenza de Adobe Flex e JavaFX, permite compilar o código LZX a un ficheiro SWF e tamén a DHTML, esta característica fai que se atope a medias na clasificación de tecnoloxías RIA seguida neste documento. Sen conseguir unha interface tan vistosa e impresionante como as de AdobeFlex ou JavaFX, a curva de aprendizaxe de OpenLaszlo é menos acusada que os dous anteriores, xa que o código LZX é moito máis lexible e de alto nivel orientado a obxectos.
- **Novell Moonlight.** É a implementación en código aberto de Microsoft Silverlight e está incluído dentro do proxecto Mono. O seu obxectivo é permitir que aplicacións Silverlight roden sobre plataformas GNU/Linux e fornecer un SDK en GNU/Linux para desenvolver aplicacións Silverlight. En virtude dun acordo con Microsoft, Novell ten e acceso ás especificacións, aos test oficiais e a empregar os *codecs* que utiliza Silverlight, aínda que, para aqueles que non quixeren usar os *codecs* binarios de Microsoft, Moonlight pode ser compilado directamente polos usuarios finais con implementacións *open source* deses *codecs*. A diferenza das anteriores, Moonlight non é unha plataforma para o desenvolvemento de aplicacións RIA, é un completo en código aberto para executar Silverlight en plataformas GNU/Linux.

## 2.2 Baseadas nos estándares soportados nativamente polos navegadores

Este tipo de tecnoloxías non necesita instalar ningún *plugin* no cliente, xa que están baseadas en HTML. Máis concretamente, a súa base tecnolóxica é Ajax.

### Vantaxes

- Non necesitan de ningún engadido no cliente
- Son multiplataforma. Ao estaren baseadas en HTML, calquera navegador pode visualizalas.

|                             |                                    |
|-----------------------------|------------------------------------|
| <b>Licenza</b>              | Licenza Pública Común              |
| <b>Plugin</b>               | Adobe Flash Player (privativo)     |
| <b>Versión</b>              | 4.0.9                              |
| <b>Linguaxe do frontend</b> | LZX                                |
| <b>Linguaxe do backend</b>  | Java, PHP...                       |
| <b>Plataformas</b>          | GNU/Linux, Mac, Microsoft, Solaris |

*Cadro 3: ficha OpenLaszlo*

|                    |           |
|--------------------|-----------|
| <b>Licenza</b>     | LGPLv2    |
| <b>Versión</b>     | 1.0.1     |
| <b>Linguaxe</b>    | C++       |
| <b>Plataformas</b> | GNU/Linux |
| <b>Navegadores</b> | Firefox   |

*Cadro 4: ficha Moonlight*

### **Inconvenientes**

- Non todos os navegadores implementan de igual maneira os estándares (HTML, CSS, Javascript), polo que poden xurdir diverxencias que obriguen a realizar distintas implementacións en función do navegador.
- Non son accesibles, xa que o usuario necesita ter activado JavaScript, de non ser así, non se visualizarán.

- Existe unha gran variedade de *frameworks* dispoñible, o que obriga o desenvolvedor a facer un forte esforzo inicial para explorar as distintas alternativas e escoller a que mellor se adaptar ás súas necesidades.

Dentro deste tipo de tecnoloxías podemos facer unha clasificación, en función do ámbito ao que están enfocadas. Así, temos:

- **Tecnoloxías centradas no cliente.** Son as tecnoloxías RIA que se executan completamente no navegador, e a comunicación co servidor faise a través de mecanismos independentes da tecnoloxía RIA, por exemplo, a través de servizos web.
- **Tecnoloxías centradas no servidor.** Son as tecnoloxías que, executándose no navegador, inclúen código que é necesario executar no servidor.

No cadro agrupamos varios *frameworks* deste tipo de tecnoloxías baseadas en estándares segundo a clasificación anterior:

| Centrados no cliente | Centrados no servidor |
|----------------------|-----------------------|
| jQuery               | GWT                   |
| Prototype            | RichFaces             |
| ExtJS                | IceFaces              |
| YUI                  | ZK                    |
| Dojo                 |                       |
| Capuccino            |                       |

*Cadro 5: frameworks*

A seguir, citaremos algúns dos *frameworks* máis utilizados:

- **jQuery.** É unha biblioteca de Javascript que permite simplificar a maneira de interaxir cos documentos HTML, manexar eventos, manipular a árbore DOM, desenvolver animacións e agregar

interacción coa tecnoloxía Ajax a páxinas web. Na actualidade conta cunha ampla comunidade grazas á cal existe un gran número de funcionalidades ou *plugins* xa implementados grazas ás cales desenvolver unha aplicación con jQuery tórnase nun proceso moi rápido que, unido á sinxeleza da biblioteca fan que este *framework* RIA teña unha penetración moi alta no mercado.

|                 |            |
|-----------------|------------|
| <b>Licenza</b>  | MIT e GPL  |
| <b>Versión</b>  | 1.3.2      |
| <b>Linguaxe</b> | Javascript |

*Cadro 6: ficha jQuery*

- **Prototype.** É o *framework* Javascript máis lonxevo, orientado a obxectos, fornécelle ao desenvolvedor técnicas Ajax listas para ser usadas no desenvolvemento áxil e sinxelo de páxinas web. O seu potencial maximízase ao utilizalo con Ruby On Rails no lado do servidor, de feito, este xa inclúe Prototype.

|                 |            |
|-----------------|------------|
| <b>Licenza</b>  | MIT        |
| <b>Versión</b>  | 1.6        |
| <b>Linguaxe</b> | Javascript |

*Cadro 7: ficha Prototype*

- **ExtJS.** *Framework* Javascript que permite construír aplicacións complexas en Internet. Está baseado en compoñentes (*layouts*, *paneis*, *grids*...) de alto rendemento, personalizados e extensibles. Como biblioteca Javascript, do mesmo xeito que jQuery e Prototype, é independente de calquera tecnoloxía ao lado servidor, no entanto é un *framework* capaz de integrarse con outros *frameworks*, en particular con GWT, que veremos a seguir,

o cal facilita a vinculación entre a presentación e o modelo de datos.

|                 |            |
|-----------------|------------|
| <b>Licenza</b>  | GPLv3      |
| <b>Versión</b>  | 3.0.0      |
| <b>Linguaxe</b> | Javascript |

*Cadro 8: ficha EXTJS*

- **YUI.** *Yahoo User Interface* é unha biblioteca Javascript que permite desenvolver aplicacións RIA utilizando DOM, Ajax e DHTML. Para alén diso, fornece un conxunto de recursos CSS e *widgets* predefinidos que permiten realizar un desenvolvemento rápido de aplicacións. Unha das principais vantaxes de YUI a respecto de outros *frameworks* é que está apoiada polos desenvolvedores de Yahoo, que poñen á nosa disposición a potencialidade dos recursos Yahoo xa existentes, dunha maneira moi ben documentada.

|                 |            |
|-----------------|------------|
| <b>Licenza</b>  | BSD        |
| <b>Versión</b>  | 1.7        |
| <b>Linguaxe</b> | Javascript |

*Cadro 9: ficha YUI*

- **Dojo.** É un *framework* Javascript que contén APIs e *widgets* para facilitar o desenvolvemento de aplicacións web que utilicen tecnoloxía Ajax. Permítelles aos desenvolvedores realizaren compoñentes de forma rápida e estandarizada, de maneira que estes sexan tamén reutilizables e compatibles con todos os navegadores. Un trazo identificador de Dojo é que fornece un sistema de pacotes para facilitar o desenvolvemento modular,

inicializando unha serie de xerarquías de paquetes e espazos de nome (*io, event*, etc.) baixo o pacote *dojo*.

|                 |            |
|-----------------|------------|
| <b>Licenza</b>  | BSD e AFL  |
| <b>Versión</b>  | 1.3.1      |
| <b>Linguaxe</b> | Javascript |

*Cadro 10: ficha DOJO*

- **Capuccino.** Cappuccino está construído sobre tecnoloxías web estándar como Javascript e implementa APIs coñecidas como as dos *frameworks* GNUstep e Apple Cocoa. Permite ao desenvolvedor abstraerse das complexidades asociadas tradicionalmente á web, HTML, CSS e mesmo DOM. Capuccino está implementado usando unha nova linguaxe de programación chamada Object-J, baseada en Object-C, que é unha linguaxe interpretada que se executa no navegador.

|                 |          |
|-----------------|----------|
| <b>Licenza</b>  | LGPL     |
| <b>Versión</b>  | 0.7.1    |
| <b>Linguaxe</b> | Object-J |

*Cadro 11: ficha Capuccino*

- **GWT (*Google Web Toolkit*).** *Framework* desenvolvido por Google para desenvolver aplicacións complexas baseadas en Ajax, utilizando a linguaxe de programación Java. Unha vez desenvolvido o *frontend* da aplicación en Java, o compilador GWT tradúceo a HTML e Javascript, isto permite ao desenvolvedor obter a potencialidade de Ajax sen necesidade de escribir código Javascript, para alén diso, o compilador GWT



asegúrase de que o código HTML e Javascript xerado sexa compatible con calquera navegador. GWT soporta unha gran cantidade de *widgets* que son útiles nos desenvolvementos de aplicacións Ajax, incluíndo árbores, pestanas, barras de menú e menús de diálogo. GWT tamén soporta chamadas de procedementos remotos (RPC) e outras características que veremos no seguinte apartado. Todas elas, fan que GWT sexa un *framework* RIA que, embora sexa recente, está a crecer a un ritmo exponencial.

|                 |            |
|-----------------|------------|
| <b>Licenza</b>  | Apache 2.0 |
| <b>Versión</b>  | 1.7        |
| <b>Linguaxe</b> | Java       |

*Cadro 12: ficha GWT*

- **JSF.** É unha especificación do JCP (*Java Community Process*), identificado como JSR 314 que define o desenvolvemento de interfaces de usuario en aplicacións Java EE. JSF fornece un marco de traballo que facilita o desenvolvemento de aplicacións, separando as diferentes camadas dunha arquitectura: presentación, regras e entidades de negocio. JSF fornece un conxunto de APIs para representar compoñentes dunha interface de usuario e administrar o seu estado, administrar eventos, validar entrada, definir un esquema de navegación das páxinas e dar soporte para internacionalización e accesibilidade e un conxunto de etiquetas para despregar JSF en páxinas JSP. Entre estas implementacións de JSF hai dúas que dan soporte a aplicacións RIA: *IceFaces* e *RichFaces*. Todas elas están orientadas a compoñentes almacenados en servidor e seguen unha programación orientada a eventos. A diferenza de moitas das bibliotecas vistas con anterioridade, os eventos xerados no cliente son xestionados polo servidor, o que as converte en tecnoloxías centradas no servidor.

| <b>Produto</b>   | <b>Licenza</b> | <b>Versión</b> |
|------------------|----------------|----------------|
| <b>RichFaces</b> | <b>GPL</b>     | 3.3.2          |
| <b>IceFaces</b>  | <b>MPL</b>     | 1.8.2          |

*Cadro 13: diversas implementacións de JSF*

- **ZK.** É un *framework* escrito en Java que permite o desenvolvemento de aplicacións RIA sen necesidade de escribir Javascript. Caracterízase por ser unha tecnoloxía centrada no servidor, orientada a eventos e baseada en compoñentes. Nun modelo multicamada correspóndese coa camada de presentación, para iso utiliza unha linguaxe de marcado denominada ZUML.

|                                    |            |
|------------------------------------|------------|
| <b>Licenza</b>                     | GPL e LGPL |
| <b>Versión</b>                     | 3.6        |
| <b>Linguaxe do <i>frontend</i></b> | ZUML       |
| <b>Linguaxe do <i>backend</i></b>  | Java       |

*Cadro 14: ficha ZK*

Outros *frameworks* RIA baseados en estándares que podemos atopar na actualidade son: *Pyjamas*, *Echo3*, *midori*, *MochiKit*, *MooTools*, *gooxdoo*, *Rialto Toolkit*, *Rico*...

## 3. GWT

### 3.1 Introducción

Nesta sección faremos unha exploración máis profunda do *framework* GWT para axudar aos desenvolvedores na súa decisión á hora de escoller un *framework* para desenvolver unha aplicación RIA. Escolleuse GWT en detrimento de outros, por tres razóns:

1. Por ser un *framework* baseado en estándares
2. Entre os baseados en estándares, foi o *framework* escollido, porque o desenvolveu Google e porque o utilizou nas súas aplicacións máis coñecidas: Gmail, GoogleDocs, Google Wave, etc., sendo así un caso de éxito indiscutible.
3. Porque todas as ferramentas necesarias para facilitar o desenvolvemento dunha aplicación en GWT son tamén software libre.

Esta sección é unha tradución parcial e non oficial de *Getting Started* de GWT

### 3.2 Características

Google Web Toolkit (GWT) é un *framework* de desenvolvemento en Java de código aberto, que permite escapar da amálgama de tecnoloxías usadas actualmente para escribir aplicacións Ajax, difíciles de utilizar e propensas a erros.

Con GWT, é posible desenvolver e depurar aplicacións Ajax usando Java en calquera ambiente de desenvolvemento (sistema operativo e IDEs). Unha vez implementada a aplicación, GWT compila e traduce o programa a Javascript e HTML compatible con calquera navegador web.

O ciclo de desenvolvemento dunha aplicación GWT é o seguinte:

1. Nun ambiente de desenvolvemento integrado (IDE) escribir e depurar unha aplicación en Java, usando as bibliotecas GWT necesarias.
2. Usar o compilador de Java a Javascript de GWT para transformar a aplicación nun conxunto de arquivos Javascript e HTML que se

colgan en calquera servidor e executalos desde un navegador web.

3. Verificar que a aplicación se execute en todos e cada un dos navegadores existentes.

As aplicacións GWT poden ser executadas en dous modos:

- **Modo *hosted* (*hosted mode*).** Onde a aplicación roda como bytecodes de Java sobre unha máquina virtual. Por norma xeral, gástase máis tempo desenvolvendo en modo *hosted*, xa que neste caso cóntase con todas as vantaxes que fornece Java para depurar usando un IDE como Eclipse. Para soportar o modo *hosted*, GWT conta cun navegador especial que está “enganchado” á máquina virtual de Java.
- **Modo *web* (*web mode*).** Onde a aplicación corre como HTML + Javascript sobre un navegador, traducido desde o código fonte Java orixinal co compilador de GWT.

Outras características:

- **RPC realmente simple.** Para comunicarse desde o navegador que lanza a aplicación co servidor web, soamente é necesario definir clases de Java serializables para as peticións e respostas. En produción, GWT serializa automaticamente as peticións do navegador a deserializa as respostas dende o servidor web. O mecanismo de RPC de GWT pode mesmo administrar xerarquías de polimorfismo en clases, e pódense manexar as posibles excepcións.
- **Optimización de código Javascript.** É unha característica do compilador GWT que permite xerar varias versións do código en tempo de compilación, mais só unha delas carregarase en tempo de execución en función dos perfís de usuario do navegador concreto. Por exemplo, ao desenvolver unha aplicación que soporte internacionalización, co módulo de internacionalización de GWT, o compilador GWT crea versións para o Firefox en galego, para o Firefox en inglés, para o Opera en galego, ... mais logo se a aplicación se executa desde un Firefox en galego só carregará o código Javascript para ese navegador.

- **Compoñentes da interface de usuario dinámicos e reutilizables.** Pódense crear *widgets* reutilizables por composición doutros *widgets* xa existentes. Despois pódese cooperar coa comunidade de desenvolvedores enviándoos como un jar.
- **Uso doutras bibliotecas Javascript ou código Javascript nativo.** Se as bibliotecas de clases de GWT non foren suficientes para o que se necesitar, pódese mesturar Javascript no código da aplicación usando JSNI (Javascript Native Interface) que permite escribir código javascript con Java. Mesmo é posible tratar os obxectos Javascript con obxectos Java, isto facilita o traballo con JSON.
- **Administración do historial do navegador.** Nas aplicacións Ajax, cando muda a interface, non muda o URL que se está visitando no navegador. Isto produce unha incompatibilidade coa forma de administrar o historial de páxinas visitadas dentro dos navegadores. GWT non é a excepción, porén, proporciona unha clase para administrar o historial dunha aplicación Ajax e para que o usuario poida seguir utilizando os botóns “Atrás” e “Seguinte” do seu navegador sen problemas.
- **Soporte para internacionalización.** Pódense crear rápida e facilmente aplicacións multidioma.
- **Soporte para accesibilidade.** GWT dá soporte a ARIA, que é unha especificación para tornar accesibles as aplicacións RIA baseadas no estándar DOM.
- **Integración con JUnit** mediante a integración de JUnit en GWT. É posible probar as aplicacións e depuralas nun navegador mentres se constrúen e mesmo é posible testar chamadas asíncronas a procedementos remotos RPC.
- **Maior produtividade.** É posible usar calquera IDE (Eclipse, Netbeans, ...) polo que se pode escoller aquel con o que se estiver máis familiarizado. Doutra banda, os erros comúns en Javascript (erros de sintaxes, por exemplo) son facilmente detectados mentres se desenvolve a aplicación, e non cando o usuario final a está executando. E, por ultimo, os deseños en Java baseados na

programación orientada a obxectos son fáciles de comunicar e entender, polo que a base do código Ajax é máis comprensible con menos documentación.

## 3.3 Desenvolvendo unha aplicación con GWT

### 3.3.1 Instalación de Google Web Toolkit

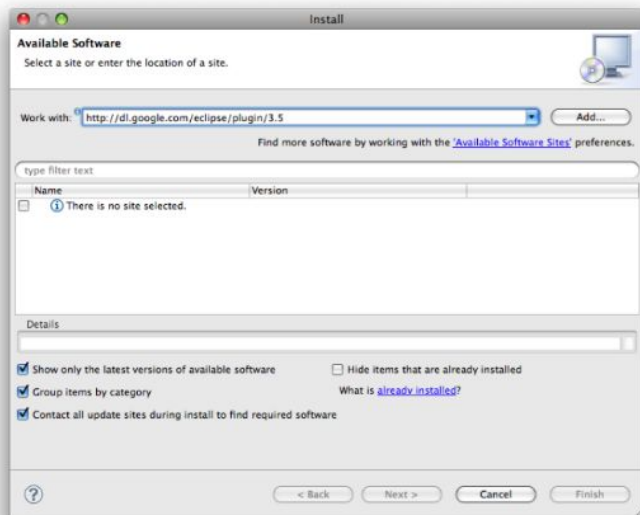
Para o desenvolvemento da aplicación empregaremos o IDE Eclipse Galileo 3.5 e o *plugin* GWT para este IDE. Seguiremos estes pasos:

1. En Eclipse, *Help - Install new software*
2. Na xanela que aparece no campo *Working with* colocamos: <http://dl.google.com/eclipse/plugin/3.5> e facemos clic en *Add*.
3. Na nova xanela deixamos *Name* baleiro, xa que o nome vai ser preenchido automaticamente, e prememos *OK*.
4. Na seguinte xanela na caixa central, asegurámonos de que estean marcados todos os checks relativos a **Plugins** e **SDK**, así, van ser instalados **Google App Engine Java SDK** e **Google Web Toolkit SDK**. Despois prememos *Next*.
5. Na seguinte xanela, revisamos que se van instalar os seguintes elementos:

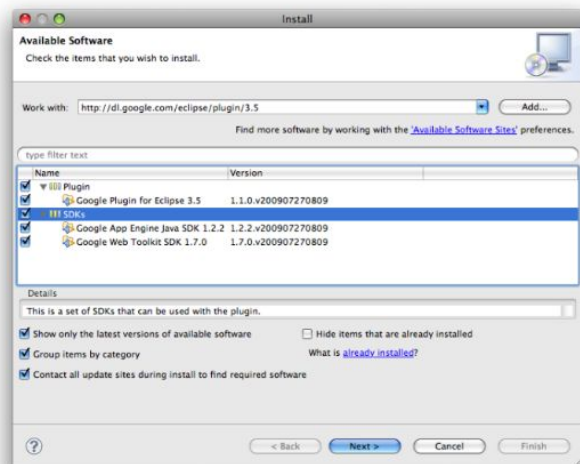
Google App Engine Java SDK 1.2.2

Google Plugin for Eclipse 3.5

Google Web Toolkit SDK 1.7.0



*Ilustración 2: xanela de instalación de GWT onde inserir o repositorio*



*Ilustración 3: xanela de instalación de GWT onde se comproba que elementos instalar*

6. A seguir, captamos os termos das licenzas.
7. Para rematar, reiniciamos Eclipse.

### 3.3.2 Creación dun proxecto GWT

Unha das vantaxes de GWT é que podemos aproveitar as múltiples ferramentas que nos fornece un IDE como Eclipse: *refactoring*, preenchemento automático de código, depuración... Para alén disto, o *plugin* GWT para Eclipse fornece un *wizard/Asistente* para a creación de aplicacións GWT. A seguir, enumeramos os pasos a seguir para crear a nosa primeira aplicación con GWT que chamaremos *TarefasGWT* :

1. Na barra de ferramentas, prememos o botón *New Web Application Project*



*Ilustración 4: botón do wizard de GWT*

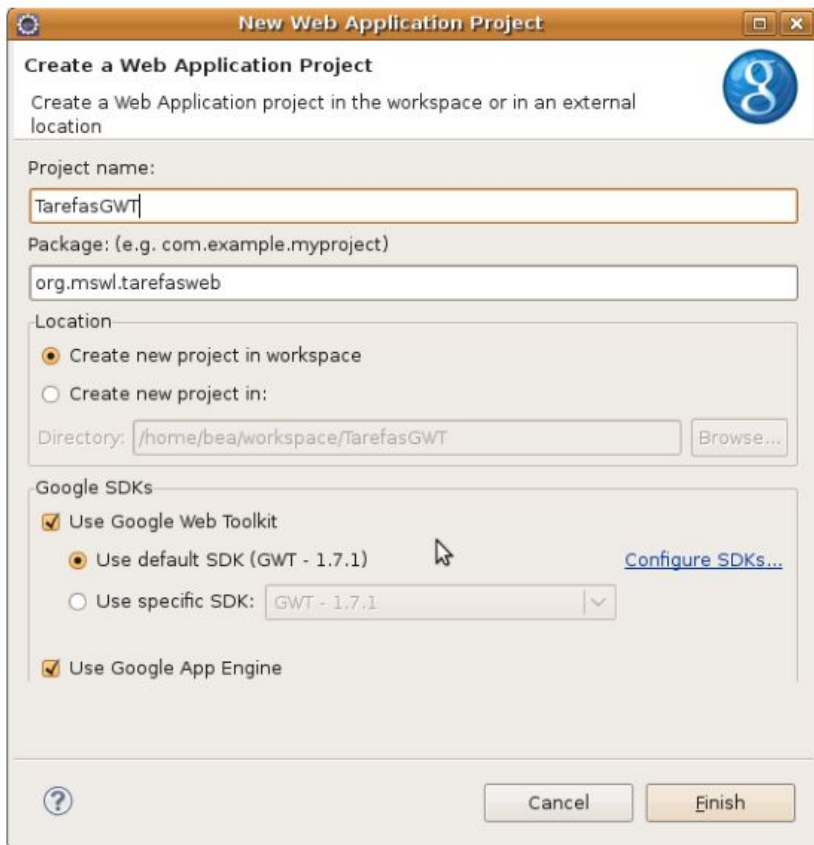
2. Inserimos como nome do proxecto *TarefasGWT*
3. Como pacote inserimos `org.mswl.tarefasgwt`
4. Debemos asegurarnos de que as opcións *Use Google Web Toolkit* e *Use Google App Engine* están marcadas, a última é opcional mais altamente recomendada.
5. Para rematar, prememos *Finish*.

Analicemos a estrutura de directorios que se xerou na nosa pasta do proxecto (*workspace/TarefasGWT/*):

- `/src/org/mswl/tarefasgwt` Contén `TarefasGWT.gwt.xml` O módulo XML de definición do proxecto GWT e os directorios *client* e *server* con arquivos que conteñen o código do cliente e o servidor respectivamente.
- `/war` Contén recursos estáticos que poden ser servidos publicamente como CSSs, imaxes...
- `/war/WEB-INF` Contén os ficheiros de configuración da aplicación Web Java.



- `/war/WEB-INF/lib` Contén as bibliotecas necesarias para o desenvolvemento da aplicación.



*Ilustración 5: creación do proxecto TarefasGWT*

Analícemos agora algúns dos ficheiros xerados polo wizard:

- `/src/org/mswl/tarefasgwt/TarefasGWT.gwt.xml`  
Ficheiro de configuración da aplicación GWT.
- `/war/WEB-INF/TarefasGWT.html` Páxina web HTML que carregará a aplicación, doravante Host Page.

- /war/WEB-INF/TarefasGWT.css Folla de estilos que se aplicará á aplicación.
- /src/org/mswl/tarefasgwt/client/TarefasGWT.java Código fonte Java do lado do servidor que implementa unha clase *entry point*.
- En /src/org/mswl/tarefasgwt/client/ os ficheiros *GreetingService.java* e *GreetingServiceAsync.java*, e en /src/org/mswl/tarefasgwt/server/ o ficheiro *GreetingServiceImpl.java* que implementa o RPC de GWT.

Neste contexto aparecen dous conceptos novos: módulo e clase *entry point*. Un módulo é a unidade básica de configuración dunha aplicación GWT onde se definen todos os datos que o compilador GWT necesita saber para xerar o código HTML e Javascript a partir do código Java. Unha aplicación GWT pode conter varios módulos, mesmo admite herdanza de módulos, por exemplo, o noso *TarefasGWT.gwt.xml* herda de *com.google.gwt.user.User*, e debe incluír unha clase *entry point*. *TarefasGWT.gwt.xml* inclúe por defecto o estilo estándar de GWT aínda que pon á nosa disposición outros dous estilos *Dark* e *Chrome*.

```
<?xml version="1.0" encoding="UTF-8"?>
<module rename-to='tarefasgwt'>
  <!-- Inherit the core Web Toolkit stuff. -->
  <inherits name='com.google.gwt.user.User' />

  <!-- Inherit the default GWT style sheet. You can change -->
  <!-- the theme of your GWT application by uncommenting -->
  <!-- any one of the following lines. -->
  <inherits name='com.google.gwt.user.theme.standard.Standard' />
  <!-- <inherits name='com.google.gwt.user.theme.chrome.Chrome' /> -->
  <!-- <inherits name='com.google.gwt.user.theme.dark.Dark' /> -->

  <!-- Other module inherits -->

  <!-- Specify the app entry point class. -->
  <entry-point class='org.mswl.tarefasgwt.client.TarefasGWT' />
</module>
```

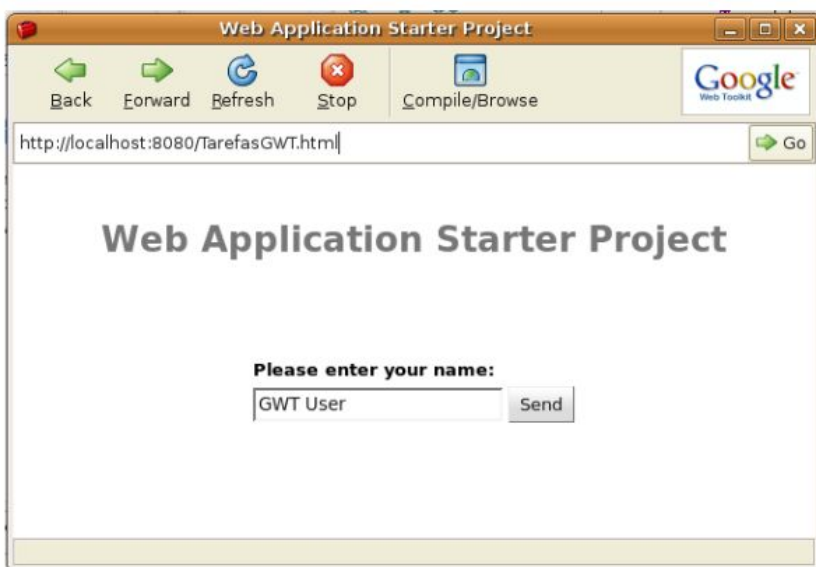
Ilustración 6: módulo *TarefasGWT.gwt.xml*

Pola súa banda, unha clase entry point, como a nosa TarefasGWT.java, implementa a interface com.google.gwt.core.client.EntryPoint e cada vez que se carrega o módulo correspondente esta clase é instanciada automaticamente e o seu método EntryPoint.onModuleLoad() é invocado.

Execución do proxecto en Modo hosted

Para iso tamén utilizaremos as facilidades do plugin de Eclipse.

1. No panel Project Explorer seleccionamos o noso proxecto TarefasGWT.
2. Botón dereito, e seleccionamos Run As Web Application. A seguir, deberíamos ver algo como o que se mostra na Figura.



*Ilustración 7: execución de TarefasGWT en modo hosted*

### 3.3.3. Deseñando a aplicación

#### **Requisitos funcionais**

A nosa aplicación TarefasGWT implementará as seguintes funcionalidades:

- Permitirá ao usuario engadir tarefas

- Listará todas as tarefas, incluíndo nome e data de creación
- Permitirá eliminar tarefas da lista

Identificación dos elementos da interface de usuario

Analizados os requisitos funcionais, necesitamos:

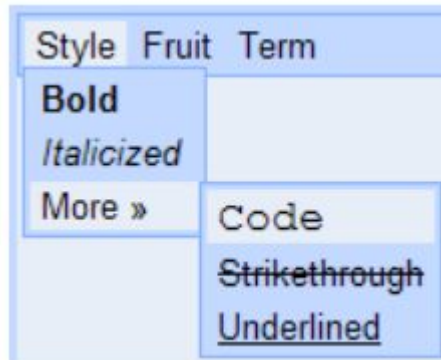
- Unha tabela para listar as tarefas
- Dous botóns, un para adicionar tarefas e outro para borrarlas
- Unha caixa de texto para inserir unha tarefa
- Incluiremos tamén unha serie de elementos estáticos, como un logo e o cabezal da páxina.

### 3.3.4. Construíndo a interface de usuario

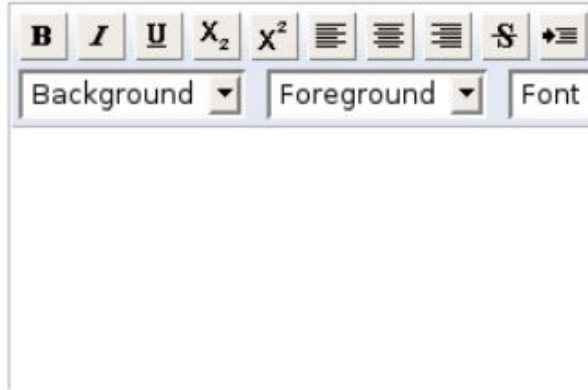
Selección dos widgets GWT que implementarán a interfaz

Antes de implemetarmos ningún widget, botemos unha ollada á galería de widgets que nos fornece GWT. Aquí expomos unha pequena mostra:

#### Menu bar



## Rich text area



## Suggest box



GWT ten un tipo especial de *widget* chamado FlexTable que permite crear novas celas baixo demanda, o cal se axusta perfectamente á nosa tabela de tarefas, xa que en principio descoñecemos cantas tarefas quererá introducir o usuario.

Por outra banda, empregaremos *Button* para os botóns de *Adicionar tarefa* e *Borrar tarefa*, *TextBox* para adicionar o nome dunha tarefa.

## Selección de paneis para agrupar os elementos UI

Unha vez seleccionados os *widgets* a utilizar, o seguinte paso consiste en decidir como se van a despregar na páxina, para iso utilizaremos os paneis de GWT. GWT fornece diversos tipos de paneis para agrupar os

elementos UI, mesmo un panel pode agrupar outros paneis. Podemos establecer unha analoxía con HTML, mediante a utilización de elementos *table* ou *div*.

Para *TarefasGWT* utilizaremos un *Panel Vertical* para listar tarefas. No noso código Java, crearemos unha instancia da clase *VerticalPanel* e chamáremola *mainPanel*. Logo utilizaremos un *Panel Horizontal* para inserir unha nova tarefa, e no noso código Java, crearemos unha instancia da clase *HorizontalPanel* e chamáremola *addPanel*.

Existe un panel máis que necesitamos, chámase *Root Panel*. Un *RootPanel* é o elemento que utiliza GWT para conter todos os compoñentes dinámicos dunha aplicación. É a raíz da xerarquía de paneis. O *RootPanel* por defecto envolve o corpo do documento HTML, e é obtido invocando o método `RootPanel.get()`. Se for necesario envolver outro elemento do documento HTML con *RootPanel*, usaremos o método `RootPanel.get(String)`.

### Embebendo os elementos na *Host Page*

Agora necesitamos introducir todos estes elementos na nosa aplicación, iso faremolo a través de nosa *HostPage*, *TarefasGWT.html*, que xerou automaticamente o wizard GWT. Para comezar, a nosa *TarefasGWT.html* terá o corpo baleiro. Isto implica que o *RootPanel* envolverá todo o corpo do documento HTML e, en consecuencia, todo o mostrado no navegador será xerado dinamicamente por GWT.

Nós introduciremos dous elementos estéticos, o título e un logo. Despois, para embeber a aplicación GWT dentro de *TarefasGWT.html*, utilizaremos un *div* e identificáremolo con *tarefasList*. Indicaremos paso a paso como facelo:

1. Abrimos a nosa *host page*, `/war/WEB-INF/TarefasGWT.html`.
2. No cabezal, mudamos o título por *TarefasGWT*, mais conservamos a referencia á folla de estilos e ao ficheiro Javascript.
3. No corpo, engadimos unha etiqueta `H1` que envolva o texto *TarefasGWT*.

4. A seguir, engadiremos un elemento *div* cuxo identificador sexa *tarefasList*
5. Conservamos o *iframe* para dar soporte ao histórico do navegador.
6. Borramos todo o código HTML e comentarios restantes, de xeito que debería quedar unha páxina como a que se mostra na ilustración 8.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <link type="text/css" rel="stylesheet" href="TarefasGWT.css">
    <title>Tarefas GWT</title>
    <script type="text/javascript" language="javascript"
      src="tarefasgwt/tarefasgwt.nocache.js"></script>
  </head>
  <body>
    
    <h1>Tarefas GWT</h1>
    <div id="tarefasList"></div>

    <!-- OPTIONAL: include this if you want history support -->
    <iframe src="javascript:''" id="__gwt_historyFrame" tabIndex='-1'
      style="position:absolute;width:0;height:0;border:0"></iframe>
  </body>
</html>
```

Ilustración 8: a hosted page *TarefasGWT.html*

## Implementando os widgets e paneis

A maioría dos elementos UI mostraranse ao ser carregada a páxina *TarefasGWT.java*. Despois implementarémolos dentro do seu método *onModuleLoad*.

Vaiamos paso a paso:

1. Instanciamos cada un dos *widgets* e paneis a utilizar, utilizando os seus respectivos construtores. Para iso abrimos */src/org/mswl/tarefasgwt/client/TarefasGWT.java* e introducimos o seguinte

código que aparece na ilustración 9. Utilizamos as utilidades de Eclipse para organizar os *imports*.

2. Creamos a tabela para listar as tarefas. Para iso introducimos dentro do método `onModuleLoad` o código amosado na ilustración 10.

```
/**
 * Entry point classes define <code>onModuleLoad()</code>.
 */
public class TarefasGWT implements EntryPoint {
    private VerticalPanel mainPanel = new VerticalPanel();
    private FlexTable stocksFlexTable = new FlexTable();
    private HorizontalPanel addPanel = new HorizontalPanel();
    private TextBox newSymbolTextBox = new TextBox();
    private Button addStockButton = new Button("Add");

    public void onModuleLoad() {
    }
}
```

*Ilustración 9: código para instanciar os widgets e os paneis*

```
public void onModuleLoad() {
    // Create table for stock data.
    stocksFlexTable.setText(0, 0, "Name");
    stocksFlexTable.setText(0, 1, "Data");
    stocksFlexTable.setText(0, 2, "Remove");
}
```

*Ilustración 10: código para instanciar os widgets e paneis*

3. Organizar os *widgets*. Para iso montarémolos sobre os paneis, máis concretamente, primeiro montaremos o *addPanel* que é o panel a horizontal o cal envolverá a caixa de texto para adicionar unha tarefa. Logo montaremos o *mainPanel* que envolverá a tabela de tarefas e o panel *addPanel*. En código, isto tradúcese como aparece na ilustración 11.



4. Asociar *mainPanel* co *Root Panel*. Para embeber todos os *widgets* e paneis xerados na *host page*, temos que vincularlos a través do *Root Panel*. Para iso asociaremos o elemento *div* da *host page* *TarefasGWT.html* e que identificamos como *tarefasList* co noso *Root Panel*. Isto tradúcese no código Java mostrado na ilustración 12.
5. Fixar o foco da páxina na caixa de texto, para que o usuario poida introducir tarefas tan pronto como sexa carregada a páxina.
6. Comprobamos o resultado do aquí implementado. Para iso, lanzamos a aplicación en *Modo Hosted*: Seleccionamos o noso proxecto e executamos *Run as - Web Application*. O resultado pode verse na ilustración 14.

```
public void onModuleLoad() {  
    // Create table for stock data.  
    stocksFlexTable.setText(0, 0, "Name");  
    stocksFlexTable.setText(0, 1, "Data");  
    stocksFlexTable.setText(0, 2, "Remove");  
  
    // Assemble Add Stock panel.  
    addPanel.add(newSymbolTextBox);  
    addPanel.add(addStockButton);  
  
    // Assemble Main panel.  
    mainPanel.add(stocksFlexTable);  
    mainPanel.add(addPanel);  
}
```

Ilustración 11: código para montar os paneis

```
// Associate the Main panel with the HTML host page.  
RootPanel.get("stockList").add(mainPanel);
```

Ilustración 12: código para montar os paneis

```
// Move cursor focus to the input box.  
newSymbolTextBox.setFocus(true);
```

Ilustración 13: fixar o foco na caixa de texto

Deixamos TarefasGWT executándose en *Modo Hosted*, xa que as sucesivas mudanzas que realizaremos probarémolas en *Modo Hosted* premendo unicamente *Refresh*.

### Manipulando eventos no cliente

1. **Analizando os eventos que se poden realizar.** GWT fornece unha gran variedade de manipuladores de eventos. Para xestionarmos o evento de facer clic nos botóns *Add* e *Delete* utilizaremos `ClickHandler` e para manipularmos o evento de premer *Enter* utilizaremos `KeyPressHandler`.



Ilustración 14: páxina web resultante

2. **Rexistrando os eventos do rato.** Para iso pasaremos un obxecto que implemente a interface `ClickHandler` que neste caso forneceremos mediante unha clase anónima. A interface `ClickHandler` ten un método `onClick` o cal se invoca cando o usuario fai un clic sobre o *widget* onde rexistramos o evento. Para iso, no método `onModuleLoad` da nosa clase *TarefasGWT.java* engadimos o código que rexistra o evento e o asocia ao botón *Add* e, na clase, definimos un novo método que chamaremos `addTarea()`, tal e como se mostra no cadro 15.
  
3. **Rexistrando os eventos de teclado.** De igual xeito asociaremos o evento de premer *Enter* no botón *Add*, mais neste caso utilizaremos a interface `KeyPressHandler` que ten un método chamado `onKeyPress`.

| Acción                    | Evento                                   | Resposta                                                                                                                         |
|---------------------------|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| O usuario insire un texto | Preme o botón <i>Add</i> ou <i>Enter</i> | Verificar a entrada<br>Verificar se a entrada xa é existente<br>Adicionar unha nova fila á tabela<br>Crear o botón <i>Delete</i> |

*Cadro 15: requisitos funcionais e eventos asociados*

4. **Respondendo os eventos.** Cando o usuario preme o botón *Add*, ou cando preme *Enter*, a nosa aplicación debe responder realizando as seguintes accións:
  - Validando o texto de entrada
  - Comprobando que o texto inserido non se corresponde cunha tarefa inserida previamente. Adicionando unha fila máis á tabela que lista as tarefas

- Adicionando un novo botón *Delete*

Neste apartado codificaremos o primeiro punto, é dicir, verificaremos que o texto inserido non contén caracteres estraños. En primeiro lugar deberemos recuperar a cadea de texto inserida polo usuario no `TextBox` utilizando o seu método `getText()`. A seguir, mediante a utilización de expresións regulares, comprobaremos a non existencia de caracteres non permitidos. Se o texto non for válido, mostraremos unha mensaxe de advertencia, e se for correcto, deixaremos a caixa de texto baleira para que o usuario poida inserir unha nova tarefa. Vexamos o código Java que implementa o explicado na ilustración 17.

```
/**
 * Add tarea to FlexTable. Executed when the user clicks
 * the addStockButton or presses enter in the newSymbolTextBox.
 */
private void addTarea() {
    final String symbol = newSymbolTextBox.getText().toUpperCase().trim();
    newSymbolTextBox.setFocus(true);

    // Stock code must be between 1 and 10 chars that are numbers, letters, or dots.
    if (!symbol.matches("[0-9A-Z\\.]{1,10}$")) {
        Window.alert("'" + symbol + "' is not a valid symbol.");
        newSymbolTextBox.selectAll();
        return;
    }
    newSymbolTextBox.setText("");
}
```

Ilustración 17: validando o texto de entrada

5. **Comprobando o desenvolvido até o momento.** Para iso, posto que aínda conservamos a execución de `TarefasGWT` en *Modo Hosted*, prememos *Refresh*. Inserimos caracteres estraños na caixa de texto e prememos *Add*, a ilustración 18 recolle o resultado desta operación.

### 3.3.5. Implementando código no lado do Cliente

Até agora construímos a interface utilizando *widgets* e paneis de GWT, rexistramos certos eventos asociados aos devanditos *widgets* e implementamos unha validación de texto inserido polo usuario en resposta a un evento. Con todo, aínda temos que implementar as seguintes funcionalidades ao lado cliente:

- Comprobar que o texto inserido non se corresponde cunha tarefa inserida previamente.
- Adicionar unha fila máis á tabela de lista as tarefas
- Adicionar un novo botón *Delete*

Na próxima sección a adicionaremos chamadas ao servidor para recuperar as tarefas.

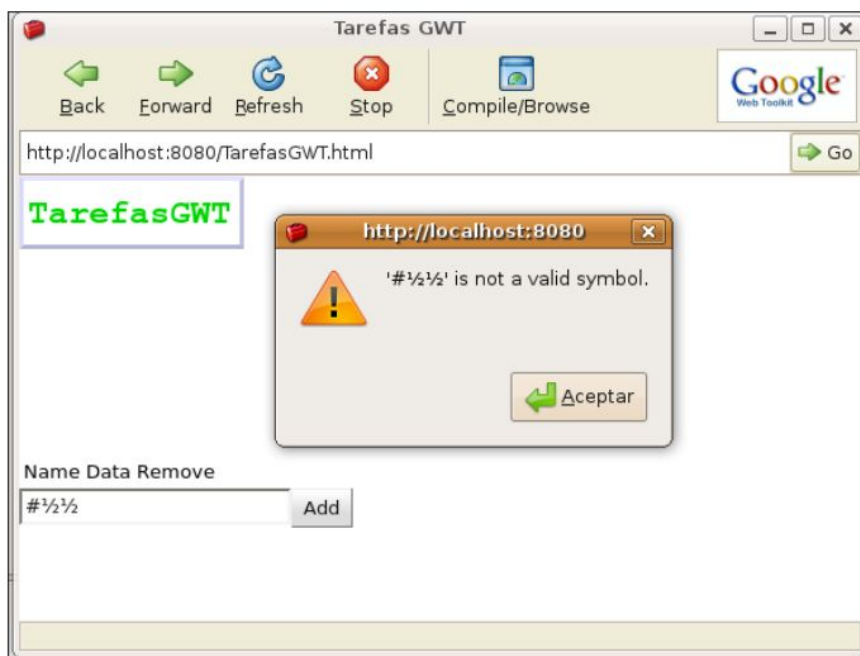


Ilustración 18: inserindo texto incorrecto en TarefasGWT

### Adicionando e eliminando tarefas da lista

Para listar as tarefas empregamos un *widget* especial chamado *FlexTable*, que permite aumentar celas de maneira dinámica sen necesidade de definir previamente o tamaño da tabela. Deste xeito, para implementarmos esta funcionalidade seguiremos os seguintes pasos:

1. **Crear unha estrutura de datos para almacenar a lista de tarefas.** Para iso utilizaremos a clase `ArrayList` de Java e chamáremoslle *tarefas*.
2. **Adicionar unha tarefa á lista de tarefas.** Despois de que o usuario inserise o nome dunha tarefa e premese *Enter* ou fíxese clic en *Add*, é necesario comprobar que o nome da tarefa non existe na lista e, se non existir, engadilo á tabela, incluíndo a súa data de creación, que recuperaremos utilizando a clase `Date` de Java. Para isto, no método `addTarea()` adicionamos o código da ilustración 20.
3. **Adicionar o botón *Delete* para borrar unha tarefa de lista de tarefas.** Un usuario pode eliminar tarefas da lista de tarefas, para iso adicionaremos un botón *Delete* nunha cela de nosa *FlexTable* chamando o método `setWidget()`. Para alén diso, é necesario rexistrar un evento, o de pulsar o botón *Delete*, para iso utilizaremos novamente a interface `ClickHandler`, e para rematar eliminaremos a tarefa da lista. Na ilustración 21 está o código para ser adicionado ao método `addTarea()`.

```
/**
 * Entry point classes define <code>onModuleLoad()</code>.
 */
public class TarefasGWT implements EntryPoint {
    private VerticalPanel mainPanel = new VerticalPanel();
    private FlexTable stocksFlexTable = new FlexTable();
    private HorizontalPanel addPanel = new HorizontalPanel();
    private TextBox newSymbolTextBox = new TextBox();
    private Button addStockButton = new Button("Add");
    private ArrayList<String> tarefas = new ArrayList<String>();
```

Ilustración 19: `ArrayList` para almacenar a lista de tarefas

```
// Don't add the tarefa if it's already in the table.
if (tarefas.contains(symbol))
    return;

// Add the tarefa to the table.
int row = stocksFlexTable.getRowCount();
tarefas.add(symbol);
stocksFlexTable.setText(row, 0, symbol);
stocksFlexTable.setText(row, 1, new Date().toString());
```

Ilustración 20: adicionar unha tarefa á lista

```
// Add a button to remove this tarea from the table.
Button removeStockButton = new Button("x");
removeStockButton.addClickHandler(new ClickHandler() {
    public void onClick(ClickEvent event) {
        int removedIndex = tareas.indexOf(symbol);
        tareas.remove(removedIndex);
        stocksFlexTable.removeRow(removedIndex + 1);
    }
});
stocksFlexTable.setWidget(row, 2, removeStockButton);
```

Ilustración 21: borrado de tarefa da lista de tarefas

4. **Comprobando o desenvolvido até o momento.** Para isto, dado que aínda conservamos a execución de *TarefasGWT* en *Modo Hosted*, prememos *Refresh*. Inserimos unha nova tarefa e prememos *Add*, a ilustración 22 recolle o resultado desta operación.



Ilustración 22: resultado de adicionar tarefas

### 3.3.6 Desenvolvendo código no lado do servidor

Nesta sección implementaremos un procedemento GWT remoto para realizar unha petición ao servidor que nos devolva a lista de tarefas.

#### **RPC: invocación remota de métodos en GWT**

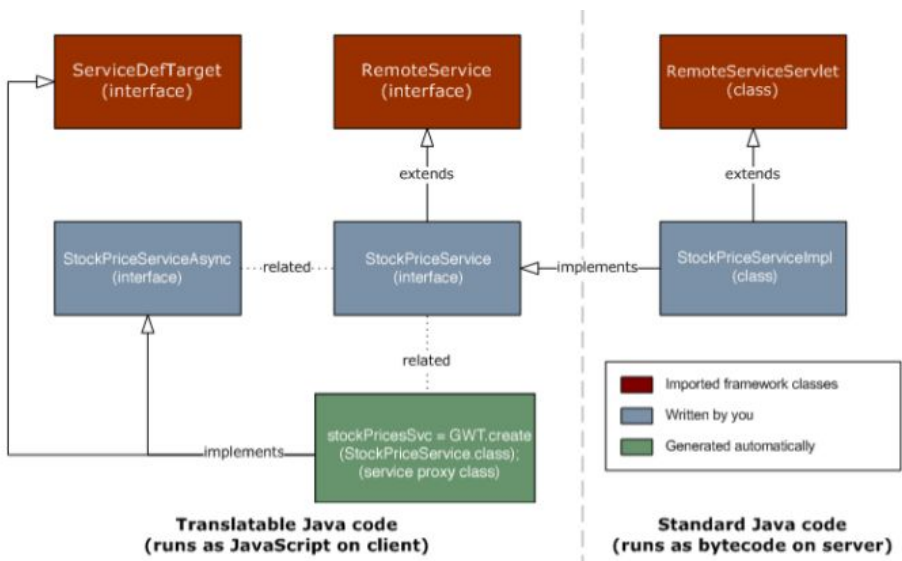
Como todas as aplicacións cliente/servidor, os programas en Google Web Toolkit necesitarán pedir certos datos ao servidor para realizaren determinadas tarefas. O mecanismo para interaxir co servidor a través da rede é chamado RPC (*Remote Procedure Call*). O RPC en Google Web Toolkit permite facilmente ao cliente enviar e recibir obxectos de Java sobre HTTP.

O código do lado do servidor que é invocado desde o cliente é frecuentemente chamado servizo, polo que chamar a procedementos remotos é comunmente chamado invocación de servizos. Embora sexa necesario termos claro que o termo servizo neste contexto non é o mesmo que *web-service*. En realidade, os servizos Google Web Toolkit non son o mesmo que SOAP (*Simple Object Access Protocol*) ou REST (*Representational State Transfer*).

Na ilustración 23 podemos ver o tres elementos chave para entendermos a invocación remota de métodos:

- O servizo que corre no servidor
- O código cliente que invoca o servizo
- Os obxectos Java que se intercambian entre cliente e servidor. Tanto o cliente como o servidor poden serializar e deserializar os obxectos transmitidos.





*Ilustración 23: anatomía de servicios en RPC de GWT*

1. **Definindo a clase para o intercambio de datos entre cliente e servidor.** Para iso imos crear unha clase chamada *Tarea* que debe ser serializable, para que poida ser utilizada como colector de datos na invocación remota de procedementos. Para isto, no pacote `org.mswl.tarefasgwt.client` definimos unha clase como a da ilustración 24.

```

public class Tarea implements Serializable {
    private String name;
    private String date;

    public Tarea() {
    }

    public Tarea(String name) {
        this.name = name;
        this.date = new Date().toString();
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }
}

```

*Ilustración 24: Obxecto 'Tarea' serializado para o intercambio de datos entre cliente e servidor*

2. **Creando o servizo.** En GWT, un servizo defínese mediante unha interface que se estende á interface RemoteService de GWT. Para isto, no pacote `org.mswl.tarefasgwt.client` definiremos nosa interface `GetTareasService` que definirá un único método `getTareas()`. A ilustración 25 recolle a definición do noso servizo. Na ilustración 25, a anotación `RemoteServiceRelative` `Path("getTareas")`

emprégase para indicar o *path* relativo á hora de invocar o servizo. Entenderémolo mellor máis adiante.

```
package org.mswl.tarefasgwclient;

import com.google.gwt.user.client.rpc.RemoteService;

@RemoteServiceRelativePath("getTareas")
public interface GetTareasService extends RemoteService {
    Tarea[] getTareas();
}
```

*Ilustración 25: interface que define o servizo RPC.*

3. **Implementando o servizo.** Agora imos implementar o servizo que se executará no servidor. Trátase de recuperar unha lista de tarefas, que o navegador deberá cargar logo que un usuario faga unha petición. Poderanse recuperar as tarefas dunha base de datos almacenada no mesmo servidor ou noutro, ou ben recuperalas de ficheiro. No noso caso, crearemos en memoria varias tarefas e almacenarémolas nun *array*. Para isto, crearemos en `org.mswl.tarefasgwserver` unha clase chamada `GetTareasServiceImpl` que estenderá a clase `RemoteServiceServlet` de GWT e implementará a interface que definimos previamente. A ilustración 26 recolle a implementación da devandita clase. É importante salientarmos que o compilador de GWT non traducirá este código a HTML nin a Javascript.

```
package org.mswl.tarefasgwserver;

import org.mswl.tarefasgwclient.GetTareasService;

public class GetTareasServiceImpl extends RemoteServiceServlet
    implements GetTareasService{

    public Tarea[] getTareas() {
        Tarea tarea1 = new Tarea("Trabajo");
        Tarea tarea2 = new Tarea("Proyecto");
        Tarea tarea3 = new Tarea("Dentista");

        Tarea[] tareas = new Tarea[3];
        tareas[0] = tarea1;
        tareas[1] = tarea2;
        tareas[2] = tarea3;

        return tareas;
    }
}
```

*Ilustración 26: Clase que implementa o servizo e que se executará no servidor*

4. **Incluindo o código do servidor no módulo de GWT.** Neste punto temos que indicarlle ao noso Colector de Servlets<sup>37</sup> onde está a implementación do servizo. Para isto hai que asegurarse de que o código do cliente estea configurado para invocar o servizo usando o URL para a cal o noso *servlet* é mapeado na configuración de web.xml. Para iso temos que incluír os elementos `<servlet>` e `<servlet-mapping>` á web.xml da nosa aplicación, tal e como mostra a ilustración 27, que nos axuda a entender mellor a anotación

`RemoteServiceRelativePath("getTareas")` da ilustración 25. O URL de invocación do servizo é `/tarefasgwt/getTareas/`, xa que `tarefasgwt` é o nome que lle demos ao noso módulo en `TarefasGWT.gwt.xml`, e `getTareas` porque é o *path* relativo que indicamos cando definimos o servizo.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

  <!-- Servlets -->
  <servlet>
    <servlet-name>getTareasServiceImpl</servlet-name>
    <servlet-class>org.mswl.tarefasgwt.server.GetTareasServiceImpl</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>getTareasServiceImpl</servlet-name>
    <url-pattern>/tarefasgwt/getTareas</url-pattern>
  </servlet-mapping>

  <!-- Default page to serve -->
  <welcome-file-list>
    <welcome-file>TarefasGWT.html</welcome-file>
  </welcome-file-list>

</web-app>
```

Ilustración 27: web.xml onde se especifica o servizo, que o implementa e como se invoca

---

<sup>37</sup> **Colector de Servlets.** O colector é parte do Servidor Web ou do Servidor de Aplicacións e encárgase realizar todo o traballo das conexións. Exemplo: Tomcat. A *Google App Engine* fornécenos o seu propio colector, *Jetty*, para executarmos a aplicación en *Modo Hosted*.

5. **Invocando o servizo no cliente.** Para isto, primeiro temos que definir unha chamada asíncrona ao servidor e despois realizar a chamada ao método remoto.

- a) **Definir chamadas asíncronas ao servidor.** Para poder realizar unha invocación a un método remoto é necesario pasarlle un obxecto da clase GWT AsyncCallback. Para iso é necesario definirmos unha nova interface que terá o mesmo nome que a nosa GetTareasService mais rematada en Async. Nesta interface definiremos o mesmo método getTareas(), mais neste caso témoslle que pasar un obxecto da clase GWT AsyncCallback que será notificado cando a chamada asíncrona for completada. Unha vez completada, o resultado é devolvido a través do mesmo obxecto. Deste xeito, no pacote org.mswl.tarefasgwclient definimos unha nova interface tal e como definimos na ilustración 28.

```
package org.mswl.tarefasgwclient;

import com.google.gwt.user.client.rpc.AsyncCallback;

public interface GetTareasServiceAsync {

    void getTareas(AsyncCallback<Tarea[]> callback);

}
```

*Ilustración 28: interface que define o método asíncrono.*

6. **Facendo a chamada ao método remoto.** O proceso de realizar un RPC desde un cliente envolve sempre os mesmos pasos:

- a) Instanciar a interface de servizo usando GWT.create(). Para isto, en TarefasGWT.java adicionamos un novo atributo tal e como mostra a ilustración 29.

b)

```
public class TarefasGWT implements EntryPoint {  
    private VerticalPanel mainPanel = new VerticalPanel();  
    private FlexTable stocksFlexTable = new FlexTable();  
    private HorizontalPanel addPanel = new HorizontalPanel();  
    private TextBox newSymbolTextBox = new TextBox();  
    private Button addStockButton = new Button("Add");  
    private ArrayList<String> tareas = new ArrayList<String>();  
    private GetTareasServiceAsync getTareasSvc = GWT.create(GetTareasService.class);  
}
```

Ilustración 29: instanciar o proxy do cliente

c) Crear un obxecto *callback* asíncrono para ser notificado cando o RPC for completado. O obxecto *callback* terá dous métodos, que se chamarán en función de se a chamada ao procedemento foi completada ou non. Isto tradúcese no código da ilustración 30.

d) Realizar a invocación ao método. Sobre o *proxy* invocamos o método remoto pasándolle o obxecto *callback*, podémolo ver na ilustración 31.

## 7. Recodificando TarefasGWT para adaptala á invocación remota. Neste punto faremos dúas mudanzas:

```
public void onModuleLoad() {  
    // Initialize the service proxy.  
    if (getTareasSvc == null) {  
        getTareasSvc = GWT.create(GetTareasService.class);  
    }  
  
    // Set up the callback object.  
    AsyncCallback<Tarea[]> callback = new AsyncCallback<Tarea[]>() {  
        public void onFailure(Throwable caught) {  
            // Do something with errors  
        }  
  
        public void onSuccess(Tarea[] result) {  
            for (int i = 0; i < result.length; i++) {  
                tareas.add(result[i]);  
            }  
            updateTable();  
        }  
    };  
};
```

Ilustración 30: obxecto Callback

```

// Make the call to the get tareas service.
getTareasSvc.getTareas(callback);

} // End of onModuleLoad Method

```

*Ilustración 31: chamada ao método remoto*

- Mudamos o atributo tareas, para transformalo nun ArrayList de obxectos Tarea, para que almacene os datos que devolve o servidor na invocación remota (ilustración 32).

```

/**
 * Entry point classes define <code>onModuleLoad()</code>.
 */
public class TareasGWT implements EntryPoint {
    private VerticalPanel mainPanel = new VerticalPanel();
    private FlexTable stocksFlexTable = new FlexTable();
    private HorizontalPanel addPanel = new HorizontalPanel();
    private TextBox newSymbolTextBox = new TextBox();
    private Button addStockButton = new Button("Add");
    private ArrayList<String> tareas = new ArrayList<String>();

```

*Ilustración 32: ArrayList de Tareas*

- Implementamos un novo método updateTable() que pintará a interface coas tarefas devolvidas na invocación remota. Para isto, copiamos todo o código restante do método onModuleLoad() e updateTable() e inserimos o código da ilustración 33.

```
// Associate the Main panel with the HTML host page.
RootPanel.get("tarefasList").add(mainPanel);

for (final Tarea tarea: tareas) {
    final String symbol = tarea.getName();
    int row = stocksFlexTable.getRowCount();
    stocksFlexTable.setText(row, 0, symbol);
    stocksFlexTable.setText(row, 1, tarea.getDate());
    Button removeStockButton = new Button("x");
    removeStockButton.addClickHandler(new ClickHandler() {
        public void onClick(ClickEvent event) {
            int removedIndex = tareas.indexOf(tarea);
            tareas.remove(tarea);
            stocksFlexTable.removeRow(removedIndex + 1);
        }
    });
    stocksFlexTable.setWidget(row, 2, removeStockButton);
}
```

*Ilustración 33: inserindo a lista de tarefas enviada polo servidor na tabela*

8. **Comprobando o desenvolvido até o momento.** Para isto, posto que aínda conservamos a execución de TarefasGWT en *Modo Hosted*, prememos *Refresh*. Veremos como a lista de tarefas que devolve o servidor é carregada automaticamente (34).

### 3.3.7. Compilando unha aplicación GWT

Neste punto, xa desenvolvimos o noso TarefasGWT, mais até agora o executamos en *Modo Hosted*, e agora debemos executalo en *Modo Web*, é dicir, onde a aplicación roda como HTML + Javascript sobre un navegador, traducido desde o noso código fonte Java orixinal co compilador de GWT.

1. Compilar a nosa aplicación. Para iso utilizaremos a utilidade que fornece o *plugin* GWT de Eclipse *GWT Compile Project* (ilustración 35). O compilador GWT xera unha serie de ficheiros que almacena en `/workspace/TarefasGWT/war/tarefasgwt/`, o nome de moitos deles conteñen un



GUIDs que almacena o código Javascript xerado polo Compilador GWT.

2. Executamos a aplicación en *Modo Web*. Para iso abrimos o noso navegador e inserimos o URL <http://localhost:8080/TarefasGWT.html>.



*Ilustración 34: lista de tarefas recuperadas por RPC*



*Ilustración 35: ícone GWT Compiler Project*

3. Despregando a aplicación nun servidor web. Abonda con copiar todos os arquivos baixo o directorio `/workspace/TarefasGWT/war/` ao servidor.

## 3.4 Comunidade GWT

Nesta sección analizaremos o grao de actividade do sistema control de versións utilizado pola comunidade de desenvolvedores que está detrás de GWT. Para iso utilizaremos a ferramenta *CVSAnaly*, que se aloxa na forxa *Morfeo*, do grupo *GsyC/LibreSoft*, que analiza os *logs* do sistema control de versións (*CVS*, *Subversion* ou *Git*) e estrutura os datos nunha base de datos de tipo MySQL. En particular, GWT utiliza *Subversion*.

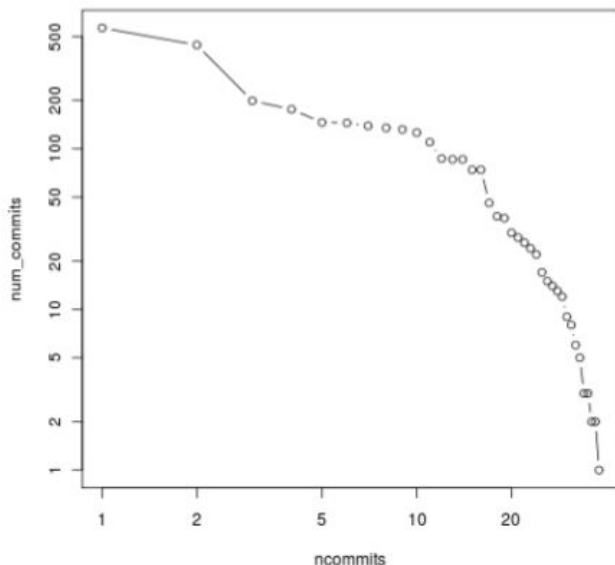
|                                    |      |
|------------------------------------|------|
| <b>Número de <i>committers</i></b> | 38   |
| <b>Número de <i>commits</i></b>    | 3085 |

*Cadro 16: datos xerais Subversion de GWT*

| <b>Committer</b>    | <b>Número de <i>commits</i></b> |
|---------------------|---------------------------------|
| scottb@google.com   | 566                             |
| gwt.team.scottb     | 443                             |
| jlabanca@google.com | 199                             |
| ...                 | ...                             |
| adamw               | 86                              |
| fabbott@google.com  | 74                              |
| ...                 | ...                             |

*Cadro 17: principais committers*

Como podemos comprobar, todos os committers de GWT son desenvolvedores de Google.



*Ilustración 36: número de commits por committer*

Na ilustración 36 podemos observar que hai un grupo pequeno de desenvolvedores que fai un gran número de contribucións, e despois hai moitos desenvolvedores que fan poucas contribucións. Aínda que non hai unha diferenza moi grande entre un grupo e outro, podemos deducir que GWT achégase ao **Modelo de Cebola**<sup>38</sup> que seguen moitos proxectos de Software Libre.

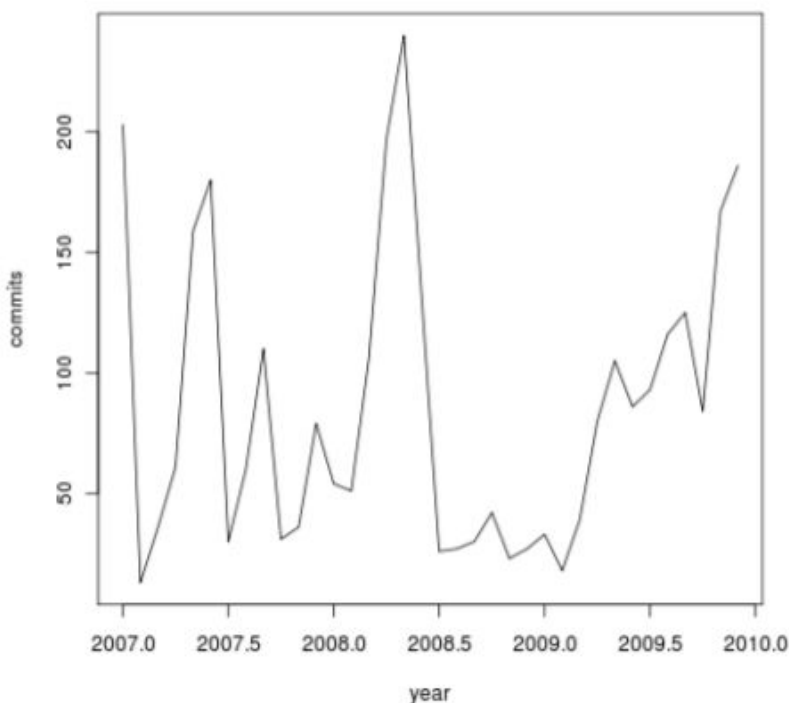
O **Coefficiente Gini**, coeficiente que mide a desigualdade a respecto da contribución dos distintos colaborados do proxecto, é de 0.6185021 e

---

38 **Modelo de Cebola.** Define unha estrutura en camadas, onde existe un núcleo de participantes que teñen unha implicación co proxecto verdadeiramente elevada, se a medirmos en termos de actividade que desenvolven (cantidade de mudanzas no código, número de mensaxes enviadas a listas de correo, *bugs* reportados e corrixidos, etc). E despois defínense outras camadas con grupos onde o número de desenvolvedores e o seu grao de implicación medran de maneira inversa, isto é, a máis desenvolvedores por camada, menos contribucións por *committer*

fica por baixo da media da maioría dos proxectos de software libre, que atinxen un valor de aproximadamente 0.7. Isto significa que no caso de GWT, a contribución dos distintos *committers* é máis equitativa que a maioría dos proxectos de software libre.

Na ilustración 37 podemos observar que o número de *commits* sufriu varios altos e baixos, despois do punto mais alto de contribucións que coincide co lanzamento da versión 1.5, aparece un período durante ultimo semestre de 2008 cun número de contribucións moi baixo. Doutra banda, neste ano 2009 o número de *commits* aumentou considerablemente, algo que moitos vinculan coa presentación de Google Wave<sup>39</sup>.



*Ilustración 37: evolución do número de commits no tempo*

---

39 Novo produto de Google que integra todas as ferramentas de Google: gmail, google docs, gtalk, etc.

## 4. Referencias

- Macromedia Flash MX-A next-generation rich client, Jeremy Allaire, March 2002.
- RIA Tecnologies overview, Javier Morán (Igalia)
- Exploring Ajax Runtime Offerings, Gary Horen 2006, <http://www.oracle.com/technology/pub/articles/dev2arch/2006/11/exploring-ajax.html>
- Adobe Flex, <http://www.adobe.com/es/products/flex/>
- JavaFX, <http://javafx.com/>
- OpenLaszlo, <http://www.openlaszlo.org/>
- Moonlight, <http://mono-project.com/Moonlight>
- JQuery, <http://jquery.com/>
- Prototype, <http://www.prototypejs.org/>
- ExtJS, <http://www.extjs.com/>
- GWT Google Web Toolkit, <http://code.google.com/intl/gl/webtoolkit/>
- Getting Started GWT, <http://code.google.com/intl/gl/webtoolkit/gettingstarted.html>
- YUI (Yahoo User Interface) <http://developer.yahoo.com/yui/>
- Dojo, <http://www.dojotoolkit.org/>
- JSF, <http://java.sun.com/javaee/jaserverfaces/>
- IceFaces, <http://www.icefaces.org/>
- RichFaces, <http://www.jboss.org/richface>
- Capuccino, <http://cappuccino.org>

# Análise das aplicacións ERP, CRM e E-COMMERCE baseadas en SFA para PEMES

José Puente Fuentes

© 2009 José Puente Fuentes

Os contidos desta obra están licenciados baixo unha licenza atribución – partilla nos mesmos termos de Creative Commons. Para consultares unha copia da licenza completa visita a páxina <http://creativecommons.org/licenses/by-sa/3.0/> ou escribe a este enderezo: Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

# 1. Contexto

Estudo elaborado para ser publicado polo CENATIC, Centro Nacional de Referencia de Aplicación das Tecnoloxías da Información e a Comunicación (TIC) baseadas en fontes abertas, cuxo obxectivo é a análise das aplicacións de tipo ERP, CRM e E-Commerce de fontes abertas. O estudo pretende ofrecer ás pequenas e medianas empresas resultados sobre a calidade a respecto da comunidade que as sostén e aos procedementos de desenvolvemento, avaliando así a súa robustez, indicativa da súa capacidade de resolver os posibles problemas que puideren xurdir, e de dilucidar se se tratan de proxectos sustentables, entendendo isto como a súa capacidade de manterse no tempo, sendo estes puntos esenciais para estes tipos de aplicacións tendo en conta as funcións críticas que van levar a cabo dentro das empresas.

As empresas fornecedoras de software de xestión empresarial son testemuñas de como nos últimos anos está a esgotarse o mercado das grandes compañías, e vense obrigadas a orientárense tamén a pequenas e medianas empresas. Para alén disto, a forte competitividade, o grande dinamismo e a flexibilidade que se esixe na xestión de calquera negocio, conduciron a unha forte demanda de solucións de xestión por parte da maioría de empresas que configuran o tecido empresarial español, composto maioritariamente por Pemes. Tendo en conta as limitacións propias das pequenas e medianas empresas, tanto económicas como de recursos humanos, os sistemas de xestión deben fornecer solucións integrais, completas, de rápida implantación, fáciles de usar, baratas e ofrecer proximidade ou atención personalizada. Neste contexto, o software aberto pode ofrecer solucións que se adaptan a estas necesidades, xa que o modelo de desenvolvemento que segue o software libre encaixa perfectamente con estes requirimentos.

Por outra banda, embora o software de fontes abertas experimentase un gran crecemento nos últimos anos, é agora cando se pode observar un aumento das oportunidades de negocio que xorden en torno a este modelo de desenvolvemento. Considerados este tipo de sistemas enfocados á xestión empresarial como pouco atractivos para as comunidades de software libre, e posta en cuestión a súa capacidade para xerar beneficios, este estudo pretende mostrar como de activas se atopan estas comunidades e as oportunidades de negocio que se xeran en torno ao

software libre, fornecendo solucións profesionais con diversos esquemas de negocio.

CENATIC é unha Fundación Pública Estatal, promovida polo Ministerio de Industria, Turismo e Comercio (a través da Secretaría de Telecomunicacións e para a Sociedade da Información e a entidade pública Red.es) e a Junta de Extremadura, que tamén conta no seu Padroado coas comunidades autónomas de Andalucía, Aragón, Asturias, Cantabria, Cataluña, Galicia e Illas Baleares.

CENATIC é o único proxecto estratéxico do Goberno de España para impulsar o coñecemento e uso do software de fontes abertas, en todos os ámbitos da sociedade.

Por outra banda, Caixanova, en colaboración coa Universidad Rey Juan Carlos e Igalia, organiza a segunda edición do Mestrado de software libre en Galicia, que pretende completar a formación de novos titulados ou profesionais que desexen profundar no desenvolvemento e xestión de software libre.

Da unión de ambas as entidades e proxectos nace unha bolsa que financia este estudo sobre solucións de xestión empresarial orientadas a Pemes, alicerce da economía española, que fornece como resultado unha análise de calidade dalgunhas das principais aplicacións deste tipo.

Dado que se trata precisamente dun estudo que se centra nas necesidades das Pemes a respecto deste tipo de solucións de software, a análise encádrase nun contexto de adquisición no que non se prevé unha colaboración destas empresas e a comunidade que xestionan os proxectos, xa que xeralmente non disporán dos recursos económicos e/ou humanos necesarios, sendo máis axeitado referírmonos a un contexto de só uso e integración na infraestrutura da empresa.

Neste contexto, habería que centrarse no estudo da comunidade, que nos mostrará a rapidez con que se solucionan posibles problemas ou a actividade que presenta e a tendencia que segue, e os seus procesos de desenvolvemento, como a súa capacidade de xerar documentación ou a frecuencia coa que se liberan novas versións, e non tanto nas ferramentas que se empregan dentro da comunidade e a calidade do código desenvolvido, xa que non se vai interaxir directamente con estas últimas.

Realizarase unha análise comparativa de proxectos que, embora teñan funcións heteroxéneas, partillan o mercado cara ao que van dirixidos. E



trataranse como proxectos únicos, nos que non se analizarán subconxuntos nin dependencias de software, xa que se trata de avaliar o que ese software en conxunto ofrece, é dicir, unha solución ERP, CRM ou E-commerce.

O obxecto deste estudo é a avaliación de dezanove aplicacións para solucións empresariais orientadas a PEMES. Esta análise dividirase en dous bloques:

Por unha banda, o estudo de aplicacións desde o punto de vista funcional e as características xerais de cada aplicación.

E pola outra, a análise das comunidades que manteñen as devanditas aplicacións, centrándose na robustez e a sustentabilidade do proxecto a longo prazo. Para esta análise foi escollida a metodoloxía QualOSS, considerada a máis axeitada polo crítica que resulta a tarefa de seleccionar para unha empresa unha aplicación deste tipo.

Os resultados obtidos serán publicados na web do observatorio de CENATIC, na que se poderán consultar os datos relativos ás aplicacións, así como o conxunto de datos necesarios para realizar a devandita análise, e tentarase que sexan de fácil consulta e poidan facilitar a toma de decisións no proceso de adquisición de software por parte de pequenas e medianas empresas.

As aplicacións obxecto desta análise son as que se listan a continuación:

- AbanQ (antes FacturaLUX)
- ADempiere
- Compierre
- ConcourseSuite (antes Centric CRM)
- ERP5
- Hipergate
- IntarS
- Magento
- Open ERP (antes Tiny ERP)

- Open for Business (Apache OFBiz)
- Openbravo
- openCRX
- opentaps
- osCommerce
- project-open
- SQL-Ledger
- SugarCRM
- vtigerCRM
- webERP

## 2. Análise das características xerais e funcionais das aplicacións

Nesta sección descríbense, para cada unha das aplicacións, os seguintes aspectos:

1. Descrición xeral da aplicación.
2. Licenza, elixida esta característica porque dela dependerá o modelo de negocio polo que se poderá optar e o grao de liberdade outorgado ao usuario. Por outra banda é importante coñecermos este aspecto por se trataren de aplicacións dirixidas a empresas que poderían ver como un valor engadido a posibilidade de pecharen o código ou adquiríreno baixo licenza comercial.
3. Soporte e *Partners*. Dado o crítica que resulta a implantación deste tipo de aplicacións na infraestrutura de calquera empresa, considérase importante que exista unha empresa patrocinadora, ou calquera tipo de organismo ou fundación, e *partners* detrás do desenvolvemento do proxecto, que garantan uns servizos de mantemento, soporte e integración de calidade. Para unha visión máis completa do soporte e fontes de información que ofrece a comunidade, ver o anexo de recursos (cumprindo cun dos requirimentos QualOSS) tamén dispoñible no traballo orixinal no CENATIC.
4. Tecnoloxía e linguaxes de programación utilizadas.
5. Modelo de negocio. En función de se estiver orientado ao servizo, soporte, adaptacións e implantación ou de se se ofreceren diferentes edicións segundo funcionalidades, o prezo poderá aumentar se se pretende adquirir unha solución integral. Estando formado o tecido empresarial español na súa maioría por PEMES, que demandan solucións completas, co consecuente aforro ao non teren que adquirir diferentes aplicacións para desempeñar as mesmas funcións, isto pode determinar a elección dunha ou doutra solución polas limitacións económicas existentes nestas pequenas e medianas empresas.
6. Funcionalidades.

7. Idiomas soportados. Importante xa que o estudo está orientado a solucións para PEMES, e neste tipo de empresas poden existir barreiras lingüísticas na utilización destas aplicacións por parte dos usuarios finais.

A información sobre cada unha das aplicacións pódese atopar no anexo de aplicacións tamén dispoñible no traballo orixinal no CENATIC.

Como conclusión e resumo dos aspectos máis importantes no que respecta ás características e funcionalidades destas aplicacións, móstranse a seguir as seguintes tabelas das licenzas, soporte, arquitecturas e linguaxes de programación e funcións que implementan cada unha delas.

- Sistema de licenzas que seguen

| TIPO GPL                                                                                             | BSD         | PROPIA LIBRE            | DUAL     | APACHE | LIBRE/COMERCIAL                 | NON LIBRE      | DUAL/COMERCIAL |
|------------------------------------------------------------------------------------------------------|-------------|-------------------------|----------|--------|---------------------------------|----------------|----------------|
| Abanq<br>Adempiere<br>ERP 5<br>Hipergate<br>IntarS<br>Open ERP<br>osCommerce<br>SQL-Ledger<br>webERP | OpenC<br>RX | Openbra<br>vo<br>vtiger | Opentaps | Ofbiz  | Compiere<br>Magento<br>SugarCRM | ConcourseSuite | ]Project-open[ |

*Tabela 1: licenzas*

- Base do soporte fornecido

| Empresas | Comunidade | Mixto    |
|----------|------------|----------|
| Abanq    | Adempiere  | Compiere |

|                                        |                               |                                                                                                            |
|----------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------------------|
| ConcourseSuite<br>IntarS<br>SQL-Ledger | Ofbiz<br>OsCommerce<br>WebErp | Hipergate<br>ERP5<br>Magento<br>Openbravo<br>OpenCRX<br>Open ERP<br>Opentaps<br>]Project-open[<br>SugarCRM |
|----------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------------------|

*Tabela 2: soporte*

- Tipo de arquitecturas e linguaxes de programación

| <b>APLICACIÓNS</b>                | <b>ARQUITECTURA</b>  | <b>LINGUAXE</b> |
|-----------------------------------|----------------------|-----------------|
| AbanQ                             | Cliente-Servidor     | C++/QT          |
| Adempiere                         | Web                  | Java            |
| Compiere                          | Cliente-Servidor/Web | Java            |
| ConcouseSuite (antes Centric CRM) | Web                  | Java            |
| ERP5                              | Web                  | Python          |
| Hipergate                         | Web                  | Java            |
| IntarS                            | Web                  | Objective-C     |
| Magento                           | Web                  | PHP             |

|                                  |                      |               |
|----------------------------------|----------------------|---------------|
| Open ERP (antes Tiny ERP)        | Cliente-Servidor/Web | Python/GTK/QT |
| Open for Business (Apache OFBiz) | Web                  | Java          |
| Openbravo                        | Web                  | Java          |
| Open CRX                         | Web                  | Java          |
| Opentaps                         | Web                  | Java          |
| osCommerce                       | Web                  | PHP           |
| Project-open                     | Web                  | Tcl           |
| SQL-Ledger                       | Web                  | Perl          |
| Sugar CRM                        | Web                  | PHP           |
| vtiger CRM                       | Web                  | PHP           |
| web ERP                          | Web                  | PHP           |

*Tabela 3: arquitecturas e linguaxes*

### • Funcións ERP

Para a elaboración da seguinte tabela tivéronse en conta os módulos básicos e primordiais que debe conter unha solución deste tipo. A continuación detállanse cada unha destas funcionalidades:

#### Finanzas

Neste módulo débense procesar as informacións xeradas noutros módulos, por exemplo, os títulos por pagar que se xeran en compras ou os títulos por cobrar que se xeran na facturación, impostos, alugueres ou préstamos.

Por outra banda, débese ocupar doutros procedementos como a transmisión electrónica de títulos a bancos, a emisión automática de

cheques, o cálculo de novo vencemento, a baixa automática de títulos na data de vencemento, a análise de crédito con consultas a clientes que mostran as súas compras, como se fixo o pagamento e outros datos importantes.

### Contabilidade

Exerce un papel fundamental nun sistema ERP, pois é aquí onde converxen todos os datos que dalgunha maneira modifican o patrimonio da empresa. Por medio dos asentamentos automáticos, ao recibir as compras, solicitudes, producións ou vendas, este módulo deberá actualizarse automaticamente.

Deberá tamén xerar os informes como o libro diario, o balanço parcial e o libro maior, e deben poder ser encadernados ou almacenados en medios electrónicos para que estean dispoñibles para calquera consulta futura, principalmente fiscal.

### Fiscal

Este módulo debe poder cumprir coa lexislación vixente neste ámbito en cada rexión ou país, polo que as principais características que debe fornecer son facilidade de adaptación e a capacidade de emisión dos informes axeitados.

### Custos

Debe estar totalmente integrado coa contabilidade. O custo dun produto debe poder verse baixo diferentes ópticas, como a estándar (ou de reposición), a real ou a media. Deberá contar tamén coa posibilidade de cálculos en diferentes moedas. O cálculo do custo de reposición ou estándar baséase na estrutura do produto e a súa implantación permite non só este cálculo senón tamén o uso do MRP I e II, as variacións de consumo de materia prima e a eficiencia da man de obra, a determinación dos prezos de venda e a toma de decisións sobre o proceso de fabricación dun determinado produto. Poderase visualizar durante a súa produción ou ao fin da mesma, deste xeito avaliarase o custo do produto en cada etapa produtiva.

A actualización dos custos esténdese por todo o proceso de produción e chega até o custo da mercadoría vendida, onde se poderá comparar co estándar e definir a ganancia da venda.

## Stock

Debe ser un sistema que axude no efectivo control dos produtos e posibilite tanto a súa consignación automática como o control de entradas/saídas, a localización do stock, a mellor asignación de recursos humanos e físicos, a identificación da prioridade de carga e descarga de produtos, a xestión do almacén, etc.

## Compras

O obxectivo da informatización do proceso de compras é suplir automaticamente o stock baseándose en criterios preestablecidos e o suficientemente flexibles para atender as mudanzas repentinas que ocorren na previsión de vendas.

Os custos dependen das compras e este intégrase co PCP, o cal contén á súa vez o MRPI (*Material Requirement Planning* ou Planificación de Necesidades de Materiais) que debe definir que, cando e canto se debe comprar de cada elemento dentro dun determinado período.

Ao chegar a mercadería a recepción será monitorizada polo sistema. A emisión da factura actualiza a carteira de pedidos, actualiza o stock, inclúe o título en contas por pagar e xera os asentamentos contables para os libros fiscais, para alén de comparar a factura non só con relación aos seus cálculos senón tamén ao establecido nos respectivos pedidos de compra.

Se o sistema conta con ferramentas de concepto BI, como o *Workflow*, os pedidos de compras poderán aprobarse de forma automática.

## PCP (Planificación e Control da Producción)

Debe calcular as necesidades de cada elemento que se producirá (MRPI) e emite as respectivas ordes de produción en conxunción co programa de operacións da carga de máquina (MRPII *Manufacturing Resources Planning* ou Planificación dos Recursos de Manufactura). Este último deberá conter as informacións relativas a horas extras, fins de semana, ausencias, variación na produtividade, etc. Estes datos deben servir para actualizar o stock e tamén os custos en número de horas reais investidas no proceso.



## BI (*Bussiness Intelligence*)

Refírese ao uso dos datos para facilitar a toma de decisións. Abrangue a comprensión do funcionamento actual da empresa e a anticipación de acontecementos futuros, co obxectivo de fornecer información para apoiar as decisións empresariais.

Mediante as ferramentas e técnicas ETL (extraer, transformar e cargar) débese poder extraer os datos de distintas fontes, depuralos e preparalos (homoxeneización dos datos) e así cargalos no almacén de datos.

## SCM (*Supply chain management*)

A administración de redes de fornecemento é o proceso de planificación, posta en execución e control das operacións da rede de fornecemento co propósito de satisfacer as necesidades do cliente con tanta eficacia como for posible. A xerencia da cadea de fornecemento abrangue todo o movemento e almacenaxe de materias primas, o correspondente inventario que resulta do proceso e as mercadorías procesadas desde o punto de orixe ao punto de consumo. A correcta xestión da cadea de fornecemento debe considerar todos os acontecementos e factores posibles que puideren causar unha interrupción.

## RR.HH.

Ocúpanse da correcta administración de usuarios involucrados na actividade da empresa. É preciso permitir ao departamento de RR.HH. a xestión básica dos empregados, como control de horas de traballo, vacacións, ausencias ou emisión de nóminas.

## Facturación

Área onde normalmente hai necesidade de fortes adaptacións: a condición de pagamento, a política de reaxustamento de prezos e descontos, o pagamento de comisións, a lexislación específica de impostos, etc. Porén, todas as excepcións ás que nos referimos deben incorporarse ao sistema de xeito que só se necesite programar a factura de forma específica e controlar a carteira de pedidos, administrar a aprobación de crédito e de stock de forma automática, emitindo todos os informes necesarios para o sector de vendas.

Podemos dicir que dentro do ERP, a facturación xera actualizacións despois da emisión da factura: a baixa de stock, a contabilización, o

registro de datos fiscais, a actualización financeira, a baixa da carteira de clientes e, no caso de contar con este módulo, xera as informacións necesarias para un bo control dentro do CRM.

### Vendas

Módulo para a xestión das transaccións con clientes, orientado a curto prazo, de carácter puntual no que non se obtén demasiada información do cliente. Considérase un axente pasivo, ofrecéndolle un trato xenérico no que non se fai especial énfase no servizo. Este módulo céntrase no produto, prezo, custo e transaccións, sendo a forza de vendas o axente activo.

### Post-venda

Establecer os canais axeitados para un correcto *feedback* co cliente para a resolución de posibles problemas ou dúbidas, efectuando un labor de fidelización.

### POS (*Point Of Sale*)

Axudar nas tarefas de xestión dun negocio de vendas ao público mediante unha interface accesible para os vendedores, permitir a creación e impresión de tickets de venda mediante as referencias de produtos, realizar diversas operacións durante todo o proceso de venda como mudanzas no stock e xerar diversos informes para a xestión do negocio.

### CRM/*Marketing* Relacional

Fai referencia aos sistemas informáticos que dan soporte a unha estratexia de negocio baseada principalmente na satisfacción dos clientes. Tamén coñecidos como sistema de abertura de datos (*front office*), trata directamente cos clientes, e o seu labor é apoiar a xestión das relacións cos clientes, venda e *marketing*. Debe tratarse dun sistema que administre un *Data Warehouse* (Almacén de Datos) coa información da xestión de vendas e dos clientes da empresa.

A venda non se resume só a unha simple orde de pedido e a súa consecuente emisión de factura, envolve o *telemarketing*, a pre-venda, o soporte post-venda, a asistencia técnica, o historial das últimas compras, o control das pendencias é dicir, a dirección e atención personalizada do cliente.

## PM (*Project Managament*) ou PMS (*Project Management System*)

Planificación, organización e xestión dos recursos para lograr a culminación con éxito das metas dun proxecto e os seus obxectivos tendo en conta as limitacións previas, como alcance, tempo e orzamento. Por outra banda, debe optimizar a asignación e a integración dos recursos necesarios para cumprir os obxectivos previamente definidos.

O sistema debe incluír o control de orzamentos, asignación de recursos, o control das fases do proxecto e seguimento do progreso físico e financeiro e índices de rendemento, fluxo de caixa do proxecto e cadros cuantitativos que mostren o previsto e realizado.

## DMS (*Document Management System*)

Xestión de grandes cantidades de documentos: rastrexar e almacenar documentos electrónicos ou imaxes de documentos en papel. Debe permitir o acceso rápido mediante diversos métodos á información contida nos documentos.

## *Groupware*

Refírese ao conxunto de utilidades que integran o traballo nun só proxecto con usuarios concorrentes que se atopan en diversas estacións de traballo conectadas a través dunha rede (Internet ou Intranet). Estas utilidades deben ofrecer unha comunicación na empresa a través das seguintes opcións: *to-dos*, notas, calendario compartido, xuntanzas, eventos, tarefas ou notificacións mediante correos electrónicos que garantan a colaboración e o traballo en equipo.

## *Mobile*

Adaptacións para os principais dispositivos móbiles para a correcta comunicación co sistema.

## *E-commerce*

Sobre todo neste contexto, este módulo de comercio electrónico envolve non só a venda, senón tamén o *marketing*, a loxística de entrega, o pagamento, o soporte post-venda e asistencia técnica e principalmente, a integración coas solucións ERP.

*Páxina seguinte: Tabela 4: funcións ERP*

| Aplicaciones                       | Funcionalidades xerats |              |        |        |       |         |     |     |    |      |             |        |            |
|------------------------------------|------------------------|--------------|--------|--------|-------|---------|-----|-----|----|------|-------------|--------|------------|
|                                    | Finanzas               | Contabilidad | Fiscal | Custos | Stock | Compras | PCP | SCM | BI | RRHH | Facturación | Vendas | Post-venta |
|                                    |                        |              |        |        |       |         |     |     |    |      |             |        |            |
| AbanQ                              | Si                     | Si           | \$     | Si     | Si    | Si      | \$  |     |    | \$   | Si          | Si     | \$         |
| Adempiere                          | Si                     | Si           | Si     | Si     | Si    | Si      |     | Si  | Si | Si   | Si          | Si     | Si         |
| Compiere                           | Si                     | Si           | Si     | Si     | Si    | Si      | \$  | Si  | \$ | Si   | Si          | Si     | \$         |
| ConcourseSuite (antes Centric CRM) |                        |              |        |        |       |         |     |     |    |      |             | \$     | \$         |
| ERP5                               | Si                     | Si           | Si     | Si     | Si    | Si      | Si  | Si  |    | Si   | Si          | Si     | Si         |
| Hipergate                          |                        |              |        |        | Si    |         |     |     |    | Si   | Si          | Si     | Si         |
| IntraS                             |                        |              |        |        | \$    | \$      | \$  |     |    | \$   | \$          | \$     | \$         |
| Magento                            |                        |              |        |        |       |         |     |     |    |      |             | Si     | Si         |
| Open ERP (antes Tiny ERP)          | Si                     | Si           | Si     | Si     | Si    | Si      | Si  | Si  | Si | Si   | Si          | Si     | Si         |
| Open for Business (Apache OFBiz)   | Si                     | Si           | Si     | Si     | Si    | Si      | Si  | Si  | Si | Si   | Si          | Si     | Si         |
| Openbravo                          | Si                     | Si           | Si     | Si     | Si    | Si      | Si  | Si  | Si | Si   | Si          | Si     | Si         |
| OpenCRX                            |                        |              |        |        |       |         |     |     |    |      |             |        |            |
| OpenTaps                           | Si                     | Si           | Si     | Si     | Si    | Si      | Si  | Si  | Si | Si   | Si          | Si     | Si         |
| OSCommerce                         |                        |              |        |        |       |         |     |     |    |      |             |        |            |
| Project-open                       |                        |              |        | Si     |       | Si      |     | Si  | Si | Si   | Si          | Si     | Si         |
| SQL-Ledger                         | Si                     | Si           | Si     | Si     | Si    | Si      |     |     |    |      | Si          | Si     |            |
| SugarCRM                           |                        |              |        |        |       |         |     |     |    |      | Si          | Si     | Si         |
| Vtiger CRM                         |                        |              |        |        |       | Si      |     |     |    |      | Si          | Si     | \$         |
| WebERP                             | Si                     | Si           | Si     | Si     | Si    | Si      |     |     |    |      | Si          | Si     | Si         |

- **Funcións CRM**

Tivéronse en conta como características básicas os seguintes módulos funcionais:

*Portal/Self-service*

A capacidade dos clientes actuais e potenciais de acceder a partes específicas do sistema para dispoñen de soporte, efectuaren compras ou ofrecerem datos de *marketing* útiles para a empresa. Debe dispor de acceso aos servizos que a aplicación implementar, como *knowledge base*, *tracking* ou reportar información relevante dunha forma directa a través deste portal e ser integrada automaticamente no sistema.

*Dashboards*

Proporciona información en tempo real a todos os axentes involucrados nunha determinada tarefa ou proceso e segundo o seu perfil de usuario. Debe ser unha información integral que abranza todo o proceso de vendas e relacione as contas asociadas para proporcionar a información necesaria aos empregados para o correcto desempeño do seu labor.

*Meetings*

Xestión de reunións.

*Calendario*

Permitir unha rápida visión do calendario da empresa e axilizar as procuras. Nel débense integrar as tarefas, xuntanzas, oportunidades e outros eventos que envolven o axente en cuestión.

*Previsión vendas*

Darase por cuberta esta funcionalidade se dispor da capacidade de xerar previsións baseadas nos datos de vendas e medir a súa precisión en relación ás vendas efectuadas.

*Leads*

Defínese como *lead* de vendas a entidade potencialmente interesada na compra dun produto ou servizo e representa no primeiro estadio do proceso de vendas e o punto de entrada no fluxo de vendas. Esta funcionalidade cobre a correcta recolección, interpretación, xestión e

asignación dos *leads* de máis calidade desde o punto de vista da probabilidade de convertérense nunha oportunidade de venda.

### *Contas*

A capacidade de mostrar unha visión integral e relacional dos clientes, na que se inclúen produtos, contactos, oportunidades, actividades, historial, citas, localización, etc, proporcionando o estado no que se atopa un cliente e o seu status.

### *Contactos*

Xestión dos contactos asociados aos clientes actuais e potenciais.

### *Campañas*

Funcionalidade que fornece a planificación, execución e monitorización das campañas de mercadotecnia a través de múltiples canais.

### *Análise efectividade*

Ofrecer medicións sobre a efectividade e rendibilidade das campañas de *marketing* e establecer como estes programas impactan no fluxo de vendas.

### *Mailing*

Capacidade de executar campañas de correo electrónico de forma masiva segundo os criterios fornecidos na análise das oportunidades de vendas.

### *Knowledge Base*

Capacidade de xestionar dunha forma eficiente información estruturada ou non, dándolle ao cliente a oportunidade de crear *FAQs*, procurar e inserir información e administrar arquivos dunha forma rápida e sinxela. Por outra banda, isto deberá reportar información relevante á empresa sobre quen, cando e en que están interesados os clientes.

### *Tracking*

Reporte de problemas que son categorizados e enviados á persoa responsable. Debe axilizar a resolución de problemas e indicar a evolución do produto ou servizo en termos de calidade.

### *Mail*

Fornece as funcións de cliente de correo electrónico e establece os filtros de entrada e saída axeitados según roles/responsabilidades dentro do organigrama da empresa.

### *Tarefas/Fluxo traballo*

Asignación de eventos (xuntanzas, correos-e, chamadas...) aos empregados axeitados e a asociación ás contas ou clientes aos que van dirixidos.

### *Reporting*

Capacidade de converter datos en información procesable que permite a monitorización do rendemento de todas as áreas da organización que cobre o sistema: tendencia das vendas, *marketing*, incidencias, perfís dos clientes e contas.

| Aplicacións                               |                     |           |                   |               |            |                |              |                                   |         |                       |              |        | Reporting      |      |               |                            |
|-------------------------------------------|---------------------|-----------|-------------------|---------------|------------|----------------|--------------|-----------------------------------|---------|-----------------------|--------------|--------|----------------|------|---------------|----------------------------|
|                                           | Previsión<br>ventas | Lea<br>ds | Oportu<br>nidades | Contact<br>os | Conta<br>s | Dashb<br>oards | Campañ<br>as | Análise<br>de<br>efectivida<br>de | Mailing | Knowle<br>dge<br>base | Tracki<br>ng | Portal | Calend<br>ario | Mail | Xuntanz<br>as | Tarefas/Flux<br>o traballo |
| AbanQ                                     | \$                  |           | \$                | \$            |            |                | \$           |                                   | \$      |                       | \$           |        |                | \$   |               | \$                         |
| Adempiere                                 | \$                  | Si        | Si                | Si            | Si         | Si             | Si           | Si                                | Si      |                       | Si           | Si     | Si             | Si   |               | Si                         |
| Compiere                                  | \$                  | Si        | Si                | Si            | Si         | \$             | Si           | Si                                | Si      |                       | Si           | Si     | Si             | Si   |               | Si                         |
| ConcourseSuite<br>(antes Centre<br>CRM)   | \$                  | \$        | \$                | \$            | \$         | \$             | \$           | \$                                | \$      | \$                    | \$           | \$     | \$             | \$   |               | \$                         |
| ERP5                                      |                     | Si        | Si                | Si            | Si         |                | Si           |                                   | Si      | Si                    | Si           |        | Si             | Si   | Si            | Si                         |
| Hipergate                                 | Si                  |           | Si                | Si            |            |                | Si           |                                   | Si      | Si                    | Si           |        | Si             | Si   | Si            | Si                         |
| IntarS                                    | -                   | -         | -                 | -             | -          | -              | -            | -                                 | -       | -                     | -            | -      | -              | -    | -             | -                          |
| Magento                                   | -                   | -         | -                 | -             | -          | -              | -            | -                                 | -       | -                     | -            | -      | -              | -    | -             | -                          |
| Open ERP<br>(antes Tiny<br>ERP)           | Si                  | Si        | Si                | Si            |            | Si             | Si           | Si                                | Si      |                       | Si           | Si     | Si             | Si   | Si            | \$                         |
| Open for<br>Business<br>(Apache<br>OFBiz) | Si                  | Si        |                   | Si            |            | Si             | Si           | Si                                |         |                       | Si           | Si     | Si             | Si   | Si            | Si                         |
| Openbravo                                 | Si                  | Si        | Si                | Si            | Si         | Si             |              |                                   |         |                       |              |        |                |      |               | Si                         |
| OpenCRX                                   |                     | Si        | Si                | Si            | Si         | Si             |              |                                   | Si      |                       | Si           | Si     | Si             | Si   | Si            | Si                         |
| Openups                                   | Si                  | Si        | Si                | Si            | Si         | Si             | Si           | Si                                | Si      |                       | Si           |        | Si             | Si   | Si            | Si                         |
| OSCommerce                                | -                   | -         | -                 | -             | -          | -              | -            | -                                 | -       | -                     | -            | -      | -              | -    | -             | -                          |
| Project-open                              |                     |           | Si                | Si            |            |                |              |                                   | Si      | Si                    | Si           | Si     | Si             | Si   | Si            | Si                         |
| SQL-Ledger                                | -                   | -         | -                 | -             | -          | -              | -            | -                                 | -       | -                     | -            | -      | -              | -    | -             | -                          |
| SugarCRM                                  | \$                  | Si        | Si                | Si            | Si         | Si             | Si           | Si                                | Si      | Si                    | Si           | \$     | Si             | Si   | Si            | \$                         |
| Vtiger CRM                                |                     | Si        | Si                | Si            | Si         | Si             | Si           |                                   | Si      | Si                    | Si           | Si     | Si             | Si   | Si            | Si                         |
| WebERP                                    | -                   | -         | -                 | -             | -          | -              | -            | -                                 | -       | -                     | -            | -      | -              | -    | -             | -                          |

Tabela 5: funcións CRM



## 3. Análise das comunidades que manteñen os proxectos

### 3.1 Metodoloxías propostas para levar a cabo o estudo

- OpenBRR (*Open Business Readiness Rating*)
- QSOS (*Qualification and Selection of Open Source Software*)
- QalOSS (*Quality in Open Source Software*)
- SQO-OSS (*Software Quality Observatory for Open Source Software*)

Tendo en conta as características das tres principais metodoloxías de análise da calidade do software libre, as aplicacións obxecto de estudo e os destinatarios da análise, o presente estudo decantouse por seguir a metodoloxía QualOSS polos motivos que se describen a seguir.

### 3.2 OpenBRR e QSOS

Ambas as metodoloxías propoñen un sistema manual de puntuación que calquera persoa cuns coñecementos básicos pode seguir, sendo a principal diferenza entre elas, que QSOS tamén avalía o contexto e as diferentes versións do produto. Estas metodoloxías son axeitadas para axudar ao usuario final a decidir que aplicación instalar e usar. Porén, a súa proposta, demasiado superficial e cun método demasiado ríxido (non contemplan diferentes escenarios), non sería profunda dabondo para axudar na toma de decisións se se pensar na integración dun compoñente de software dentro dun produto de software. Neste caso, os riscos dunha mala selección son máis elevados, e estas dúas metodoloxías non responden as preguntas necesarias, por exemplo, a análise da calidade do software. Por outra banda, tamén carecen da trazabilidade desexada para este tipo de estudo.

Por tanto, OpenBRR e QSOS son axeitadas para obter unha avaliación para a implantación de aplicacións de software libre cando o risco dunha selección incorrecta terá pouco impacto na empresa. Con todo, cando os riscos desta selección foren elevados, como a selección de produtos de

software empresarial, tanto OpenBRR como QSOs carecen da minuciosidade e rigor adecuados.

### 3.3 QualOSS e SQO-OSS

O principal obxectivo destas metodoloxías é fornecer axuda no proceso de selección do compoñente de software máis axeitado para a integración nunha solución de software máis ampla. A principal diferenza entre estas metodoloxías é que SQO-OSS só se centra na calidade do produto resultante, mentres que QualOSS tamén avalía outros aspectos importantes do proceso de desenvolvemento de software.

En calquera proceso de adquisición de software (libre ou privativo), o usuario non só está interesado no produto, senón tamén na seriedade e reputación da empresa que está detrás do desenvolvemento. No caso dun proxecto de software libre, no proceso de adquisición débese valorar tamén esta seriedade na iniciativa FLOSS que leva a cabo o desenvolvemento e soporte. QualOSS expón unha metodoloxía que abrangue toda a iniciativa FLOSS, estudando os produtos resultantes do traballo (incluíndo código, documentación e test), membros da comunidade, procesos de software e ferramentas e bibliotecas. Todo isto vains dar unha visión da seriedade da iniciativa FLOSS mediante dous claros indicadores: robustez, entendida como a capacidade de resolver problemas rapidamente, e a capacidade do proxecto de ser sustentable a longo prazo.

### 3.4 Conclusión

Nun escenario de adquisición no que os riscos da implantación non sexan demasiados elevados, OpenBRR e QSOS serán suficientes para a elección dunha determinada aplicación. Tendo en conta que a implantación dunha aplicación de tipo ERP/CRM é crítica para calquera empresa, xa que afecta a súa infraestrutura e os riscos son elevados, será necesario utilizar unha metodoloxía que nos asegure un contexto de desenvolvemento e soporte axeitados, avaliando todo o ente que estiver detrás do desenvolvemento da aplicación e que nos asegure un entorno estable, sendo máis apropiadas QualOSS ou SQO-OSS. Ante estas dúas opcións, QualOSS é máis completa a respecto da análise da calidade da

comunidade que contribúe, polo que esta foi elixida para realizar o estudo de calidade destas aplicacións.

## 3.5 Análise mediante a metodoloxía QUALOSS

### 3.5.1 Definición previa do obxecto de estudo

**Iniciativa (ou esforzo) FLOSS** son as tarefas realizadas polos membros da comunidade, seguindo uns procesos co propósito de crearen un produto e valéndose das ferramentas necesarias. É dicir, Qualoss divide os proxectos de software libre en:

- CM (*Community Members*), conxunto de membros da comunidade.
- WP (*Work Products*), conxunto de produtos obtidos como resultado do seu traballo: código, documentación, etc.
- SP (*Software Processes*), procesos seguidos para realizar os produtos.
- TL (*Tools & Libraries*), bibliotecas para compilar o produto e as ferramentas das que se axudan, como o software de control de versións, seguimento de erros, etc.

**Escenario segundo a amplitude** do estudo refírese ao alcance do obxecto de estudo. Dependendo do que se trate, o estudo terase que centrar máis ou menos nos axentes mencionados anteriormente e especificar os obxectos de estudo en cada un deles. Pódense enumerar catro tipos de escenarios deste tipo:

- Un só proxecto: se non se tiver previsto contribuír ao proxecto, é preciso centrarse na calidade do software producido. En caso contrario, débense abranguer os catro aspectos: CM, WP, SP e TL. Se só se desexar integrar/usar e asegurar a posibilidade de actualizacións, será preciso centrarse na comunidade (CM), a cal tamén nos indicará a capacidade de xerar documentación e os procesos de software (SP).
- Conxunto de proxectos: centrarase na familia de proxectos e como se interrelacionan entre si, avaliando se existen

procedementos comúns, se partillan algún tipo de organismo que os aglutine, etc.

- Subcompoñente: a amplitude redúcese ao compoñente en cuestión e, para alén dos catro aspectos anteriores, débese considerar a independencia do código, o encapsulamento, a subcomunidade que o rodea, etc.
- Versións: os tres escenarios anteriores fan referencia ao momento da selección, neste caso atopámonos ante a situación de actualización a unha nova versión. Débese avaliar a calidade do código e as diferenzas entre versións nos catro aspectos.

### **Escenarios de adquisición:**

- Colaboración: escenario no que se prevé contribuír ao proxecto. Neste escenario baséase o método estándar de QualOSS, xa que é o máis amplo, cubrindo todos os aspectos (CM, WP, SP e TL). De todos os xeitos, este método é o suficientemente flexible para poder adaptarse a outras situacións e escenarios.
- Explotación: Debe asegurar un bo soporte e lonxevidade do proxecto.
- *Fork*: A calidade do produto debe ser o centro do estudo.
- Adquisición: Céntrase na calidade do produto.
- En infraestrutura: Neste caso, podería ser suficiente QSOS ou OpenBRR, mais se o esforzo da empresa for moi grande, tanto en recursos humanos como económicos, complexo de comprender ou administrar e necesítase formación, ou os clientes vanse a relacionar co sistema, QualOSS será máis axeitado.
- Servizo: Interesante estudar a robustez e capacidade de sosterse no tempo dun proxecto pouco coñecido para avantaxar a competencia.

Unha vez coñecidos os diferentes axentes que interveñen no desenvolvemento de software libre e os diferentes escenarios e contextos nos que as empresas se poden atopar no proceso de selección dunha aplicación, podemos definir en que aspectos se debe centrar este estudo.

O escenario de negocio, ou de adquisición é “en infraestrutura”, e segundo amplitude, dun só proxecto. Neste caso, os aspectos a estudar máis importantes serán CM (Membros da comunidade) e SP (Procesos de software) e serán os que nos revelen a capacidade de resolver problemas, tanto internos como externos, o soporte que obteremos, a capacidade de xerar documentación e como de sustentable é o proxecto a longo prazo.

**Como calquera outra metodoloxía, QualOSS componse de requirimentos, métodos e postulados.**

Entre os requirimentos máis importantes podemos salientar:

1. Indicar breve e claramente a adquisición de software libre para o que está dirixido.
2. Establecer o esforzo esperado en persoas/mes.
3. Trazabilidade: dise que é trazable se é posible recuperar todos os *input-output* e relacións, incluíndo información relativa aos procedementos, técnicas e ferramentas usados no proceso de entrada e saída.
4. Débese respectar o **método QualOSS**. Trátase dun método xenérico estándar que se encadra dentro dun contexto de colaboración no que se estudan de forma ampla os catro elementos que compoñen unha iniciativa FLOSS, embora este método sexa o suficientemente flexible para adaptarse a diferentes situacións ou escenarios. Consta de:

**1. Inicialización da avaliación:**

- a) Describir o contexto.
- b) Por que é necesario o estudo desde o punto de vista do negocio.
- c) Explicar como serán usados os resultados nese contexto.
- d) Enumerar as limitacións que a avaliación debe respectar en tempo, custo e esforzo.
- e) Validar os resultados producidos por estes obxectivos.

**2. Creación e planificación do estudo:**

- a) Identificar a iniciativa do software libre exacta, é dicir, o ámbito (*scope endeavor*) ten que ser claro e exacto.
- b) Definir quen o fará e a quen afectará.
- c) Establecer a carga de traballo.
- d) Identificar as ferramentas, foren estas manuais, software ou parcialmente automáticas, e as súas medidas ou configuracións.
- e) Planificar a supervisión.
- f) Validar os resultados producidos.

### **3. Recoller e analizar datos:**

- a) Recoller, analizar e manter a relación entre a información sen procesar, os métodos e as ferramentas usadas para a análise e os resultados.
- b) Validación. Se for necesario adaptar as ferramentas e métodos, débense validar de novo e gardar as adaptacións ou mesmo mudar a versión do método ou ferramenta.

### **4. Interpretar os resultados:**

- a) Interpretar a información analizada na etapa anterior usando os métodos seleccionados na etapa 2.
- b) Manter a relación entre a información analizada, os métodos de interpretación e a interpretación dos resultados, é dicir, manter a trazabilidade.
- c) Validar a interpretación dos resultados.

### **5. Supervisión do estudo:**

- a) Debe ser transversal e estar presente en todas as etapas do método.
- b) Monitorizar que todas as tarefas sexan executadas como se planificaron, incluíndo que cada tarefa realice a validación dos seus resultados.

- c) Rexistrar a tensión durante o desenvolvemento, é dicir, se por exemplo, sempre se atopan erros na mesma tarefa ou etapa.

O obxectivo final deste método é conseguir uns indicadores que cuantifiquen a calidade do proxecto en termos de robustez e posible lonxevidade, mais, a diferenza doutras metodoloxías, QualOSS pretende responder a preguntas concretas que son máis fáciles de interpretar polos destinatarios dos estudos, e así ofrecer unha forma de percibir o risco asociado a certos aspectos importantes dentro da comunidade que desenvolve o proxecto.

**O procedemento ou fluxo de traballo en QualOSS poderíase resumir da seguinte forma:**

1. Identificar a persoa responsable do estudo.
2. Planificar a supervisión.
3. Identificar os colaboradores no estudo.
4. Identificar con precisión a amplitude (*scoping*) do proxecto a estudar.
5. Engadir regras específicas a estudo unha vez acoutado.
6. Adaptar o modelo estándar, e refinar as preguntas que se tentan contestar, ao escenario específico.
7. Identificar e contactar con membros da comunidade a estudar.
8. [Opcional] Rápida avaliación da calidade dos datos.
9. Seleccionar e adaptar indicadores e ferramentas.
10. Crear información adicional da configuración das ferramentas se for necesario.
11. Recoller e analizar datos.
12. Interpretar resultados.
13. Entrevistas cos avaliadores.
14. Compartir resultados e conseguir opinións dos membros da comunidade.

### 3.5.2 Ferramentas

QualOSS apóiase en varias ferramentas de extracción de datos para atopar os indicadores de calidade:

- CVSanaly: ferramenta que garda o *log* dun sistema de control de versións (CVS, *Subversion* ou *Git*) nunha base de datos de tipo MySQL.
- MailingListStats: garda listas de correo en formato *mbox* a unha base de datos MySQL.
- Bicho: recolle datos de sistemas de control de erros gárdalos nunha base de datos de tipo MySQL. Actualmente é capaz de recoller datos da forxa *SourceForge*, a través dun *parser* de HTML e de *Bugzillas* de GNOME e KDE. Por outra banda, tamén conta con *scripts* que axudan no procesamento da información obtida a través das anteriores ferramentas.
- runQueriesQualOSS.py: automatiza a execución de consultas ás bases de datos creadas por CVSanaly, MailingListStats e Bicho.
- normalize.py: dálle o formato axeitado aos arquivos creados polo anterior *script*.
- calcIndicators.py: calcula unha regresión lineal mediante R, a cal nos mostrará a pendente/tendencia de determinados eventos e axentes, por exemplo, *commits*, novos *committers*, etc.

No anexo de recursos, tamén dispoñible no traballo orixinal no CENATIC, enuméranse as fontes dos diferentes proxectos a analizar, de onde foron obtidos os datos necesarios para aplicar as ferramentas anteriormente mencionadas, cos datos requiridos por QualOSS na súa fase de “inicialización”.

### 3.5.3 Recursos externos

Para alén das anteriores ferramentas, QualOSS tamén obtén información do proxecto FLOSSMETRICS, que fornece datos sobre centos de proxectos de software libre.



## 4. Problemas detectados con MLStats e Bicho

### 4.1 MLStats

As listas de correo da maioría das aplicacións deste estudo están aloxadas en *SourceForge* e non é posible acceder aos arquivos en formato *mbox*, xa que só os administradores dos proxectos teñen permisos para facelo. Aplicacións que contan con listas en *SourceForge*:

- AbanQ. Embora estean en desuso desde a mudanza de *FacturaLux*. Actualmente os foros son a súa maior ferramenta de soporte para os usuarios.
- Adempiere
- Hipergate
- Openbravo
- Opentaps
- ]project-open[
- SugarCRM
- vtigerCRM
- webERP

Outras non dispoñen de listas de correo:

- Compiere
- Concoursesuite
- IntarS
- Magento
- SQL\_Ledger

Noutros casos, embora non estean en *SourceForge*, tampouco teñen formato *mbox*.

- Open ERP. Actualmente sen conexión.

- osCommerce. Só dispón dunha lista destinada a anuncios sobre a aplicación.

Disponíbles

- ERP5
- Apache Ofbiz

Por mor da falta de fontes de datos axeitadas para esta ferramenta, non se puideron extraer datos das listas de correo da maioría dos proxectos, por tanto tampouco se poden obter os indicadores que derivan destes.

## 4.2 Bicho

Por mor de problemas de compatibilidade cos sistemas de seguimento de erros, só se puideron crear as bases de datos das seguintes aplicacións (aínda que como se verá máis adiante, erros nos datos obtidos fan imposible o uso desta ferramenta para o estudo):

- Abanq
- ADempiere
- Compiere
- Hipergate
- Openbravo
- Opencrx
- Opentaps
- ]project-open[
- webERP

## 4.3 Erros

```
bicho --type sf -p /tmp/bicho --db-password_out
root --db-user_out root --db-database_out bicho
```

```

"http://sourceforge.net/tracker/?
func=detail&aid=685047&group_id=29057&atid=410215
"

mysql

/usr/lib/python2.6/dist-
packages/storm/database.py:342: Warning: Table
'GeneralInfo' already exists

return function(*args, **kwargs)

/usr/lib/python2.6/dist-
packages/storm/database.py:342: Warning: Table
'Bugs' already exists

return function(*args, **kwargs)

/usr/lib/python2.6/dist-
packages/storm/database.py:342: Warning: Table
'Comments' already exists

return function(*args, **kwargs)

/usr/lib/python2.6/dist-
packages/storm/database.py:342: Warning: Table
'Attachments' already exists

return function(*args, **kwargs)

/usr/lib/python2.6/dist-
packages/storm/database.py:342: Warning: Table
'Changes' already exists

return function(*args, **kwargs)

mysql

Obtaining bug links, from url:
http://sourceforge.net/tracker/?
func=detail&aid=685047&group_id=29057&atid=410215

Analysing: http://sourceforge.net/tracker/?
func=detail&aid=685047&group_id=29057&atid=410215

Traceback (most recent call last):
File "/usr/local/bin/bicho", line 27, in <module>

```

```

retval = Bicho.main.main (sys.argv[1:])
File "/Tools/bicho-0.3/Bicho/main.py", line 266,
in main
bich.run ()
File "/Tools/bicho-0.3/Bicho/Bicho.py", line 39,
in run
self.backend.run()
File "/Tools/bicho-0.3/Bicho/backends/sf.py",
line 158, in run
dataBug = self.analyzeBug(bug)
File "/Tools/bicho-0.3/Bicho/backends/sf.py",
line 100, in analyzeBug
dataBug = parser.getDataBug()
File "/Tools/bicho-
0.3/Bicho/backends/ParserSFBugs.py", line 481, in
getDataBug
ch.Field = change.Field
AttributeError: 'str' object has no attribute
'Field'

```

Erro no parseamento, que ao atopar na páxina a palabra “*Changes:*” dentro da sección “*Comments*”, que é igual á tabela que debe gardar na base de datos, carrega un *string* que non é correcto, dando como resultado este erro.

Por outra banda, esta tabela *Changes* non é correctamente carregada na base de datos, o que fai que os seus campos estean baleiros, provocando o seguinte erro:

```

python runQueriesQualOSS.py localhost root root
empty empty bicho_weberp
runQueriesQualOSS.py:93: Warning: Incorrect
datetime value: ''
cursor.execute(queriesQualOSS.cm_sral)

```

```
runQueriesQualOSS.py:94: Warning: Incorrect
datetime value: ''
cursor.execute(queriesQualOSS.cm_sra3_bts)
runQueriesQualOSS.py:97: Warning: Incorrect
datetime value: ''
cursor.execute(queriesQualOSS.cm_iwa1_bts)
```

Embora remate o proceso sen lanzar un erro na execución, os arquivos de datos que xera están baleiros, non podendo obter os indicadores sobre a resolución de *bugs*.

Polos problemas xa mencionados coas ferramentas *Mlstats* e *Bicho*, soamente utilizarase *Cvsanaly* para obter os indicadores de calidade baseándose nos datos extraídos do repositorio de desenvolvemento principal.

## 5. Preguntas e ferramenta empregada para extraer os datos

|                                                                                                                                                                                                      |              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| Mantívose estable ou aumentou o número de membros que contribúen ao código ao longo da historia do proxecto?                                                                                         | Cvsanaly 2.0 |
| Mantívose estable ou aumentou o número de persoas que contribúen con outros datos que non sexan código ou reporte de <i>bugs</i> ao longo do tempo?                                                  | Cvsanaly 2.0 |
| Mantívose estable ou aumentou o número dos membros máis activos que contribúen ao proxecto? (Entendendo isto como <i>committers</i> que pertencen grupo que efectúan máis de 80% de <i>commits</i> ) | Cvsanaly 2.0 |

|                                                                                                                                                                                                                                                      |              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| Cal é a evolución dos membros máis activos que deixaron ese “ <i>core team</i> ” durante un período significativo de tempo?                                                                                                                          | Cvsanaly 2.0 |
| Foron compensados os períodos de tempo en os que os membros máis activos deixaron o “ <i>core team</i> ” coa incorporación de novos membros principais?                                                                                              | Cvsanaly 2.0 |
| Cal é a media de tempo de permanencia dos membros?                                                                                                                                                                                                   | Cvsanaly 2.0 |
| Cal é a evolución do número de membros que proporcionan remendos ou <i>commits</i> ?                                                                                                                                                                 | Cvsanaly 2.0 |
| É o número de eventos suficiente para considerala unha comunidade activa? (Entendendo evento como unha acción rexistrada que tivo lugar no repositorio de datos)                                                                                     | Cvsanaly 2.0 |
| É axeitado o número de <i>commits</i> ao código para considerala unha comunidade activa desde o punto de vista de <i>committers</i> ?                                                                                                                | Cvsanaly 2.0 |
| Existen subgrupos na comunidade? Se for así, están estes desconectados ou existen membros que actúan como ponte entre eses subgrupos? (Subgrupos poden ser a nivel de roles, como <i>bug reporter</i> ou <i>committer</i> e tamén a nivel de código) | Cvsanaly 2.0 |

Unha vez executados os *scripts* sobre as bases de datos, obtemos os indicadores que nos permiten responder ás anteriores preguntas, o cal nos indicará o grao de adecuación das comunidades a respecto de:

- Tamaño e rexeneración: o grao en que a evolución e rexeneración da comunidade de software libre mantén unha proporción adecuada para que sexa sustentable.
- Interacción e carga de traballo: o grao en que a comunidade interaxe de maneira axeitada e no que a carga de traballo entre os membros da comunidade efectúase de forma que se mantén a cohesión e motivación.

## 6. Problemas detectados con Csvanaly

### 6.1 Disponibilidade

Os repositorios das seguintes aplicacións non están dispoñibles, polo que foi imposible atopar os indicadores de calidade:

- AbanQ
- IntarS (só se pode acceder a unha versión antiga e sen actividade. Só dispoñible para publicidade do proxecto)
- SQL-Ledger
- SugarCRM

### 6.2 Incompatibilidade

Open ERP. O sistema de control de versións que utiliza é *Bazaar*, o cal non é compatible coa ferramenta *Cvsanaly*. Soamente é posible obter datos antigos de *TinyERP*.

### 6.3 Erros

- Openbravo

```

Erro en Cvsanaly-2.0
cvsanaly2 --db-user root --db-password root --db-
database -l openbravo.log
Parsing log for openbravo (git)
Traceback (most recent call last):
File "/usr/local/bin/cvsanaly2", line 37, in
<module>
retval = pycvsanaly2.main.main (sys.argv[1:])
File "/usr/local/lib/python2.6/dist-
packages/pycvsanaly2/main.py", line 321, in main
parser.end ()
File "/usr/local/lib/python2.6/dist-
packages/pycvsanaly2/Parser.py", line 62, in end
self.handler.end ()
File "/usr/local/lib/python2.6/dist-
packages/pycvsanaly2/DBProxyContentHandler.py",
line 80, in end
self.db_handler.commit (item)
File "/usr/local/lib/python2.6/dist-
packages/pycvsanaly2/DBContentHandler.py", line
343, in commit
committer = to_utf8 (commit.committer.name)
AttributeError: 'NoneType' object has no
attribute 'name'

```

Este erro foi corrixido para que no caso de non existir, o obxecto reportase un erro mais seguise co proceso de extracción de datos.

Despois de comprobar o número total de erros, só se atopou 1, polo que se toma en conta a información obtida, xa que, no peor dos casos, só se perdeu un *commit*, non sendo relevante nin facendo que o cálculo final estea terxiversado.



- **Compiere**

```
$cvsanaly2 --db-user root --db-password root
--db-database compiere_trunk
http://svn.compiere.com/core/trunk
[...]
Warning (31891): parsing svn log, unexpected line
M /trunk/xx.1
Warning (31896): parsing svn log, unexpected line
Rutas cambiadas:
Warning (31897): parsing svn log, unexpected line
A /trunk/xx.2
Warning (31902): parsing svn log, unexpected line
Rutas cambiadas:
Warning (31903): parsing svn log, unexpected line
M /trunk/xx.1
Warning (31908): parsing svn log, unexpected line
Rutas cambiadas:
[...]
20
$python runQueriesQualOSS.py localhost root root
compiere_trunk empty empty
Error 1146: Table 'compiere_trunk.file_types'
doesn't exist
```

Neste caso, o *parseamento* do repositorio non o realiza correctamente porque o *log* non ten o formato axeitado. Embora cree a base de datos, non insire os datos necesarios para a extracción de información do repositorio.

- **Script normalize.py**

Engade e pon a cero os meses do ano nos que non se produce actividade no repositorio. Modificado para evitar que se, por exemplo, a

data de inicio do proxecto é novembro do ano X, enchese os meses anteriores deste ano e os puxese a cero. Isto provocaba un erro no valor da pendente/tendencia que se calcula no *script calcIndicators.py*.

## 7. Conclusións

### 7.1 Características xerais e funcionais

Este estudo pretende servir como referencia ás Pemes no proceso de selección dunha solución informática para a mellora dos seus procesos de negocio. Tendo en conta isto, desde o punto de vista das funcionalidades que cobren, obviamente será preferible unha solución o máis integral posible, que abranga o maior número de áreas da empresa. Sobre todo é importante no caso das Pemes, as cales non dispoñen dos recursos económicos e humanos das grandes corporacións. Con todo, hai que sinalar que isto depende das necesidades da empresa. Se a intención é incorporar á empresa un sistema CRM, a maior facilidade de uso e interfaces máis amigables, para alén de conter, na maioría dos casos, máis funcionalidades específicas deste tipo de aplicacións, fan preferible a elección de solucións especialistas neste ámbito. O mesmo sucede nas aplicacións de tipo E-commerce, embora as diferenzas cos sistemas integrais ERP/CRM sexan menos significativas.

As tabelas referentes ás funcionalidades, que nos fornecen información da relación funcionalidades/prezo, servirán como indicador de se a solución encaixa coas necesidades e limitacións da empresa. Por outra banda, estas as limitacións económicas fan que sexa especialmente importante o tipo de edicións e soporte, xa que as diferenzas de prezo poden ser moi importantes. Isto tamén fai necesaria unha análise das diferentes alternativas ofrecidas e comprobar se existe unha versión axeitada a respecto da empresa cara á que se dirixe, soporte e prezo. O tipo de licenza, edicións e o modelo de negocio mostrados para cada aplicación daranos a información necesaria sobre estas cuestións.

As limitacións humanas tamén son un punto importante a ter en conta cando se pretende integrar na empresa unha solución deste tipo, polo que tamén é importante dispor da información relativa ao soporte profesional

de empresas especializadas que é ofrecido sobre a aplicación, xa que as empresas interesadas poden ter diferentes niveis de capacitación.

Como conclusión do anterior, desde o punto de vista das características e funcionalidades das aplicacións, non é posible establecer que solución é mellor, xa que cada empresa ten unhas necesidades específicas. Segundo estas necesidades e contrastándoas coa información exposta no estudo espérase que sexa posible obter unhas conclusións sobre cal é a elección óptima en cada caso.

## 7.2 Grao de robustez e capacidade de ser sustentable a longo prazo

Neste apartado, non foi posible obter datos das seguintes aplicacións por non estar dispoñible a devandita información:

- AbanQ: non está dispoñible o repositorio, o sistema de seguimento de erros está case inactivo e tampouco dispón de listas de correo. A tentativa de crear unha comunidade a través de *Sourceforge* e un repositorio privado, baseándose na aplicación base, non logrou atraer colaboradores.
- Concoursesuite: desde 2007 non se rexistran mudanzas na edición Comunidade, en 2009 case non existe actividade no sistema de seguimento de erros e non dispón de listas de correo.
- IntarS: non ofrece acceso ao repositorio, e non conta con sistema de seguimento de erros nin listas de correo. Soamente está dispoñible unha versión antiga e moi básica (sen mudanzas desde 2007) en *Sourceforge* a modo de proba.
- SQL-Ledger: non é posible acceder ao repositorio, e non dispón de sistema de seguimento de erros nin listas de correo.
- SugarCRM: conta con sistema de seguimento de erros, mais non é posible acceder ao repositorio e a lista de correo da que dispón non está o suficientemente integrada co desenvolvemento da aplicación.

Se nos basearmos nun dos principios básicos sobre os que se asenta o software libre, “moitos ollos ven máis que poucos”, estas solucións obteñen unha valoración desfavorable por careceren dos medios

axeitados para unha correcta comunicación, crecemento e enriquecemento a través dunha comunidade saudable.

Como xa se mencionou, Open ERP, embora conte con todas as ferramentas para a correcta comunicación da comunidade, non foi posible analízalo por non ser compatible coas ferramentas usadas neste estudo. Para alén diso, tampouco foi posible a extracción de información a través de *cvsanaly* do proxecto *Compiere* porque no parseamento dos datos do repositorio prodúcense erros.

Os datos cualitativos sobre as aplicacións que foi posible avaliar, pódense consultar nas tabelas anexas QualOSS. Estas tabelas conteñen os datos e as explicacións axeitadas para obter unhas conclusións sobre a robustez e capacidade dos proxectos de mantérense a longo prazo.

### 7.3 Puntuacións obtidas (sobre 4)

|                |      |
|----------------|------|
| ADempire       | 2,4  |
| ERP5           | 2,7  |
| Hipergate      | 2,9  |
| Magento        | 2,5  |
| Apache Ofbiz   | 2,3  |
| Openbravo      | 3,3  |
| OpenCRX        | 2,86 |
| Opentaps       | 2,9  |
| osCommerce     | 3,22 |
| ]project-open[ | 3,1  |
| vtigerCRM      | 2,4  |
| webERP         | 2,6  |

## 8. Bibliografía

- Haberkorn, Ernesto. “*Gestión Empresarial con ERP*”.
- Aloy , Raimond. “*Soluciones ERP para Pymes. Hacia el alineamiento entre fabricantes y empresas usuarias*”.
- Renart Cava, Lluís G. “*CRM: Tres estrategias de éxito*”.
- Deprez, Jean-Christophe, Haaland, Kirsten e Kamseu, Flora. “*QualOSS Methodology & QUALOSS assessment methods*”.
- Principais paxinas web consultadas:
  - <http://www.ohloh.net>
  - <http://sourceforge.net>
  - <http://guide.conecta.it>
  - <http://www.eosdirectory.com>
  - <http://www.qsos.org>
  - <http://www.wikipedia.org>
  - <http://abanq.org>
  - <http://www.adempiere.com>
  - <http://www.compiere.com>
  - <http://www.concursive.com>
  - <http://www.erp5.org>
  - <http://www.hipergate.org>
  - <http://www.seat-1.com>
  - <http://www.magentocommerce.com>
  - <http://openerp.com>
  - <http://ofbiz.apache.org>
  - <http://www.openbravo.com>

- <http://www.opencrx.org>
- <http://www.opentaps.org>
- <http://www.oscommerce.com>
- <http://www.project-open.org>
- <http://www.sql-ledger.org>
- <http://www.sugarforge.org/content/open-source>
- <http://www.vtiger.com>
- <http://www.weberp.org>

# Comunicación entre aplicacións de área de traballo e web

Manuel Rego Casasnovas

mrego@igalia.com

© 2009 Manuel Rego Casasnovas

Os contidos desta obra están licenciados baixo unha licenza atribución – partilla nos mesmos termos de Creative Commons. Para consultares unha copia da licenza completa visita a páxina <http://creativecommons.org/licenses/by-sa/3.0/> ou escribe a este enderezo: Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

## **Resumo**

Na actualidade, todo o mundo emprega aplicacións web. Embora teñan vantaxes, como por exemplo a de estaren dispoñibles en calquera lugar grazas á Internet, presentan problemas de usabilidade no que respecta ás aplicacións comúns de área de traballo.



# 1. Introducción

Existen numerosos servizos web na Internet. A definición xeral de servizo web é a dun sistema que fornece unha interface que permite a interacción entre máquinas a través dunha rede.

Cada vez máis aplicacións web fornecen algún tipo de API para acceder aos seus servizos. Este número está a aumentar e parece que a tendencia se vai manter durante un tempo. Algúns exemplos destas aplicacións son: Flickr<sup>40</sup>, Facebook<sup>41</sup>, Twitter<sup>42</sup>, etc.

As aplicacións web presentan vantaxes en comparación coas aplicacións de área de traballo:

- Os datos pódense partillar e están centralizados.
- Non precisan de seren configurados nin instalados, un simple navegador é suficiente.
- É posible acceder a elas desde calquera lugar

Por outra banda, as aplicacións de área de traballo presentan as seguintes vantaxes:

- Son máis rápidas.
- Acostuman ter interfaces de usuario mellores e máis elaboradas.

Na actualidade conviven varias tendencias. Algunhas delas céntranse no desenvolvemento de aplicacións para seren executadas nun navegador, mentres que outras céntranse no desenvolvemento de aplicacións de área de traballo ou móbiles que empregan servizos web.

Na presente obra trataremos o segundo grupo. O principal problema é a conexión entre as aplicacións de área de traballo e eses servizos web. Aínda non existe unha solución estándar para todas as aplicacións de área de traballo.

Na actualidade, cada aplicación de área de traballo desenvolve a súa propia solución para acceder a algúns deses servizos remotos, e hai que

---

40 <http://www.flickr.com/>

41 <http://www.facebook.com/>

42 <http://www.twitter.com/>

loitar unha e outra vez cos mesmos problemas relacionados coa implementación.

Estas aplicacións web acostuman fornecer algún tipo de conxunto de APIs (ou bibliotecas) para facilitar o acceso aos servizos web desde a área de traballo (ou desde outras aplicacións web). Isto permite partillar unha parte do código, mais non constitúe unha solución final para a integración total entre área de traballo e web.

## 2. Últimas novidades na plataforma GNOME

Existen diversos proxectos de GNOME<sup>43</sup> encamiñados a solucionar os problemas relacionados co acceso a servizos web remotos e que tentan fornecer unha solución común para todas as aplicacións de área de traballo. Chegados a este punto, revisaremos brevemente tres destes proxectos: *Online Desktop*, *Conduit* e *Mojito*.

En *live.gnome.org* hai unha páxina (*Gnome Integration with Online Services*<sup>44</sup>) que tenta agrupar toda a información sobre as diversas tecnoloxías que permiten integrar a área de traballo e aplicacións web. A idea consiste en englobar todas as distintas bibliotecas para definir unha maneira común de acceder a servizos web na futura versión de GNOME, a 3.0<sup>45</sup>. Os usuarios van esperar nesa altura unha mellor comunicación entre a área de traballo e os servizos web máis habituais.

### 2.1 Online Desktop

*Online Desktop*<sup>46</sup> foi unha importante iniciativa que tentou unir as aplicacións da área de traballo cos servizos web. A idea principal consiste en empregar aplicacións da área de traballo para acceder as distintas aplicacións web, en lugar dun simple navegador web.

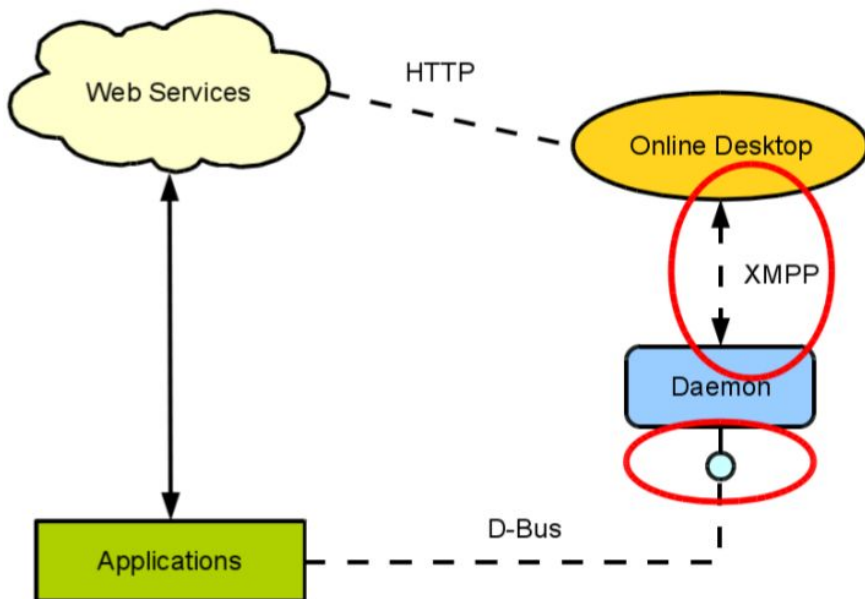
---

43 <http://www.gnome.org>

44 <http://live.gnome.org/OnlineIntegration>

45 <http://live.gnome.org/ThreePointZero>

46 <http://live.gnome.org/OnlineDesktop>



*Ilustración 1: Arquitectura de Online Desktop*

Un componente importante é o servidor web central, escrito en *Java*, que administra o acceso aos diferentes servizos web. Este servidor principal vai interaxir coas aplicacións web, implementando as interfaces necesarias para realizar esta tarefa. Este servidor principal tamén vai almacenar algúns datos dos usuarios como por exemplo contas de servizos web, configuracións das aplicacións da área de traballo, etc. Mesmo podería almacenar as señas de acceso das aplicacións web.

Isto é unha idea excelente, xa que todo o traballo necesario para conectarse con calquera aplicación web partillárase no servidor central. Deste xeito, cando una aplicación mudar a súa API, só habería que modificar o servidor. Aínda máis, desde o punto de vista do usuario final, a principal funcionalidade é que só tería que autenticarse unha vez (no servidor central) para poder acceder a todos os seus datos e servizos. Deste xeito, o usuario non tería por que se preocupar de que computador está a usar, xa que toda a información ficaría almacenada no servidor principal.

Desde o punto de vista das aplicacións da área de traballo, *Online Desktop* fornece un *daemon*, que define certos métodos e sinais *D-Bus*<sup>47</sup>, para que estas o empreguen á hora de interaxiren cos diversos servizos web. O *daemon* está conectado co servidor principal mediante o protocolo XMPP<sup>48</sup>, o que quere dicir que existe unha comunicación bidireccional entre o *daemon* e o servidor central. Isto permítelle ao servidor enviar sinais a aplicacións de área de traballo cando se modificaren datos. O servidor principal podería mesmo pesquisar os diversos servizos web de cando en vez para logo enviar sinais ás aplicacións de área de traballo.

Velaquí algunhas das vantaxes desta solución:

- É moi cómoda para as aplicacións de área de traballo, xa que só teñen que traballar cunha interface *D-Bus* para interaxiren coas aplicacións web.
- A lóxica de interaxiren co servizo web é implementada unha soa vez no servidor principal.

E algúns dos inconvenientes:

- O servidor central podería tornarse nun gargalo se houber demasiados usuarios. Poderían mesmo xurdir problemas de escalabilidade.
- Poderían xurdir problemas de seguridade, xa que o servidor principal almacena unha gran cantidade de información do usuario.
- O proxecto está abandonado na actualidade.

*Online Desktop* naceu a meados de 2007 e durou aproximadamente un ano. Foi iniciado por algúns *hackers* importantes de GNOME (Havoc Pennington, Owen Taylor e Colin Walters) que estaban a traballar en Red Hat<sup>49</sup> na altura. A meados de 2008 a actividade no proxecto comezou a decaer e fica parada desde hai un ano.

---

47 <http://dbus.freedesktop.org/>

48 <http://en.wikipedia.org/wiki/Xmpp>

49 <http://www.redhat.com/>

## 2.2 Conduit

Conduit<sup>50</sup> é una aplicación de sincronización xenérica para GNOME. Non é unha simple ponte entre aplicacións web e da área de traballo. Por exemplo, permite sincronizar: dous cartafoles no sistema, a biblioteca de música co *iPod*, os contactos co teléfono móbil, etc. *Conduit* permite crear distintos módulos para cada tipo de sincronización.

Por certo, *Conduit* soporta o protocolo *SyncML*<sup>51</sup>, un estándar de sincronización para teléfonos móbiles, polo que permite partillar contactos entre o computador e o teléfono móbil.

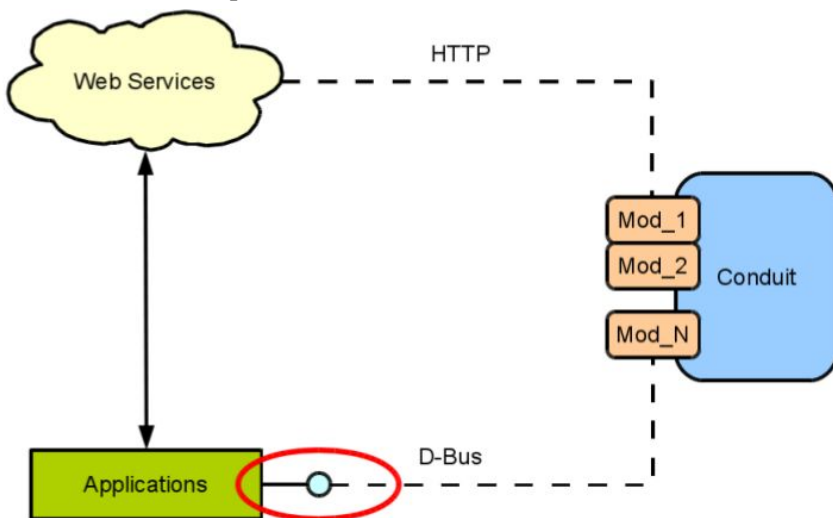


Ilustración 2: Módulos de Conduit (desde o punto de vista das comunicacións web e de área de traballo)

Debido a que o seu ámbito vai para alén dos servizos web, trátase dunha abordaxe totalmente diferente. Porén, fornece unha nova idea para solucionar o problema relacionado coa sincronización web.

Para a sincronización de aplicacións e servizos web emprega as interfaces de *D-Bus* fornecidas polas aplicacións de área de traballo, ou os arquivos almacenados polas devanditas aplicacións para obteren información. Isto quere dicir que debería coñecer a interface *D-Bus* de

<sup>50</sup> <http://live.gnome.org/Conduit>

<sup>51</sup> <http://en.wikipedia.org/wiki/Syncml>

cada aplicación ou o tipo de formato dos arquivos que emprega. Por tanto, se houber mudanzas na implementación da aplicación, sería preciso modificar o módulo de *Conduit*.

As aplicacións de área de traballo non coñecen *Conduit*, polo que non saben se alguén está a sincronizar os seus datos. Isto pode tornarse nunha fonte de problemas considerable, especialmente se a funcionalidade que fornece *Conduit* non for tan potente como se as aplicacións accedesen aos servizos web de maneira directa.

Se falarmos da sincronización entre escritorio e a web as vantaxes principais son:

- Xa existe certa cantidade de código fonte para acceder a APIs de servizos web en varios módulos.

Algúns dos inconvenientes son:

- Non está pensado para o propósito específico de conectar áreas de traballo e aplicacións web.
- Depende de interfaces *D-Bus* ou dos formatos dos arquivos das aplicacións de área de traballo, que poderían mudar en calquera altura.
- As aplicacións de escritorio non administran a interacción cos servizos web.

## 2.3 Mojito

*Mojito*<sup>52</sup> é un proxecto orientado a trasladar as redes sociais á área de traballo. É un proxecto recente e forma parte do proxecto *Moblin*<sup>53</sup>, desenvolvido por Intel<sup>54</sup>.

Na altura, non é máis ca un agregador de redes sociais, que extrae contidos dunha serie de servizos web. *Mojito* fornece unha interface *D-Bus* que pode ser aproveitada polas aplicacións de área de traballo para tiraren datos desas aplicacións web. Para alén diso, fornece aínda un

---

<sup>52</sup> <http://moblin.org/projects/mojito>

<sup>53</sup> <http://moblin.org/>

<sup>54</sup> <http://www.intel.com/>

método para actualizar o estado do usuario nos servizos web que o soportaren. Esta abordaxe é similar á de *Online Desktop*.

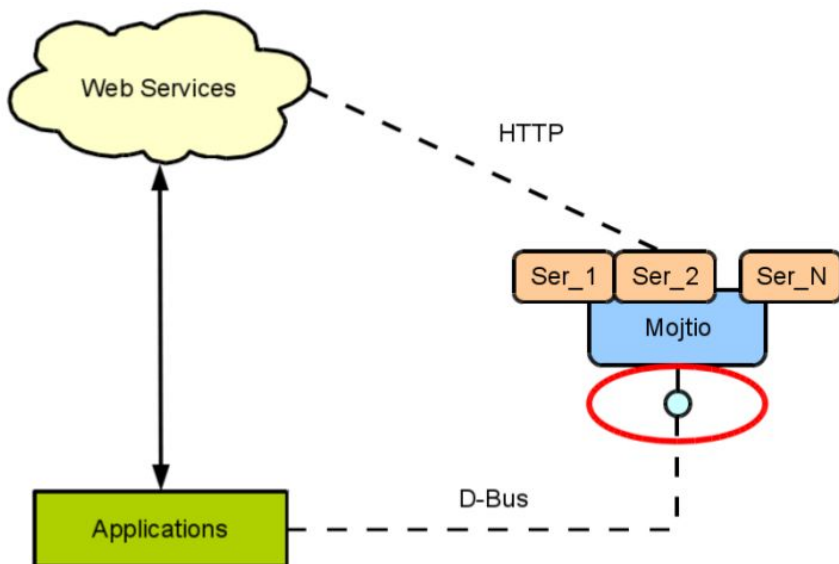


Ilustración 3: Estrutura de Mojito

A modo de resumo, poderíase dicir que se trata dun lugar no que se centraliza toda a interacción entre a área de traballo e diversas aplicacións web. Toda a lóxica necesaria para conectarse a eses servizos web está implementada en *Mojito*. Aínda máis, pesquisa os diversos servizos web de cando en vez e envíalles sinais *D-Bus* cando se produciren mudanzas.

Vantaxes:

- Desde o punto de vista das aplicacións de área de traballo que tentan interaxir con algún servizo web, é bastante similar á abordaxe de *Online Desktop*.
- Toda a lóxica relacionada cos servizos web está implementada na mesma aplicación, polo que é doado partillar parte do código común para acceder á web.

Inconvenientes:

- Na altura só soporta unhas poucas funcionalidades, e non é seguro que vaia aumentar o seu ámbito actual.
- Aínda está nos primeiros pasos e aínda non forma parte do proxecto GNOME.

## 2.4 Obxectivos

Logo de revisar a situación actual deste tema, a idea é crear unha biblioteca baseada en GNOME para fornecer acceso á aplicación web *Remember The Milk*<sup>55</sup>. E, despois, tratar de fornecer a través de *Mojito* (vía *D-Bus*) todos os métodos necesarios para interaxir con este servizo web, facilitando o acceso de calquera aplicación a *Remember The Milk*.

A razón de desenvolver unha biblioteca para este servizo (e non outro) é que non existía unha boa biblioteca escrita en *C* para acceder a esta aplicación web. Para alén diso, durante o Mestrado en software libre<sup>56</sup> desenvolveuse un cliente de área de traballo para *Remember The Milk*. Esta aplicación podería acabar por empregar a biblioteca (de maneira directa ou a través de *Mojito*) para facilitar a interacción co servizo web.

*Remember The Milk* é unha aplicación web para xerir tarefas e administrar o tempo. Aspira a mellorar a experiencia de usuario neste sentido. Fornece multitude de funcionalidades que permiten crear listas de tarefas múltiples, organizar as tarefas en función de etiquetas, especificar a localización dunha tarefa, partillar tarefas entre grupos de persoas, etc. Case todas estas funcionalidades están dispoñibles na súa web API.

---

<sup>55</sup> <http://www.rememberthemilk.com/>

<sup>56</sup> <http://www.mastersoftwarelibre.com/>



## 3. A biblioteca: RTM-GLib

A biblioteca creada chámase *RTM-GLib*. Fornece acceso á API de *Remember The Milk*.

A páxina web da biblioteca foi creada na *wiki* de GNOME<sup>57</sup>. Nela está recollida a información sobre o proxecto, como por exemplo as noticias máis recentes, as notas de versión, o itinerario, etc.

Cómpre salientar que, tras o anuncio do primeiro lanzamento de *RTM-GLib* (versión 0.1.0), publicado nas listas de distribución de GNOME e *Remember The Milk*, a biblioteca tornouse nun dos conxuntos de APIs recomendados na páxina web de *Remember the Milk*.

### 3.1 Dependencias

*RTM-GLib* depende unicamente de *librest*, unha biblioteca deseñada para facilitar o acceso a servizos web que din ser *REST*<sup>58</sup> (transferencia de estado representacional), como a de *Remember The Milk*.

Esta biblioteca programouse empregando *libsoup* e *libxml* co fin de facilitar o máximo posible o acceso a calquera servizo web, sen necesidade de procesar manualmente a resposta da aplicación web cando se envía unha petición. *Librest* forma parte do proxecto *Moblin* e foi desenvolvida por Rob Bradford e Ross Burton, os autores do proxecto *Mojito*, que tamén emprega *librest*.

A razón principal para empregar esta biblioteca en lugar da combinación habitual de *libsoup* (ou *libcurl*) e *libxml* radica na súa simplicidade. Grazas a esta biblioteca, as peticións a servizos web son moi fáciles de procesar, polo que todo o esforzo se concentra na implementación do material particular de *Remember The Milk*. Para alén diso, cumpre con todos os requisitos para desenvolver *RTM-GLib*.

---

57 <http://live.gnome.org/RTMGlib>

58 [http://en.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://en.wikipedia.org/wiki/Representational_State_Transfer)

## 3.2 Desenvolvemento

*RTM-GLib* está escrita en C<sup>59</sup>, como a maioría das bibliotecas e aplicacións de GNOME.

Como calquera outra biblioteca de GNOME, toma *GLIB*<sup>60</sup> e *Gobject*<sup>61</sup> como base. Define algunhas clases que representan os diversos obxectos que precisa unha aplicación que desexa interaxir con *Remember The Milk*.

Para alén diso, a biblioteca emprega *Check*<sup>62</sup> (un *framework* de teste de unidades para C) para comprobar o comportamento das distintas clases definidas pola biblioteca.

De maneira adicional, empregouse *GTK-Doc*<sup>63</sup> para definir a referencia da API, co obxectivo de permitir que calquera poida comezar a empregar a biblioteca sen problema.

Para alén diso, durante o desenvolvemento de *RTM-GLib* empregouse un repositorio *Git*<sup>64</sup> (<http://gitorious.org/rtm-glib/>). E lanzáronse dúas versións, a primeira versión (0.1.0) e a de mantemento (0.1.1), que reparaba algúns erros, tentando seguir unha das máximas do software libre: “lanza logo, lanza con frecuencia”.

## 3.3 Licenza

A licenza que se escolleu para *RTM-GLib* foi a *GNU Lesser General Public License 2.1*<sup>65</sup> (Licenza pública xeral reducida). A razón desta decisión é que a maior parte das bibliotecas de GNOME empregan esta licenza, polo que, en certa maneira, é máis compatible con GNOME.

Para alén diso, este tipo de licenza (*non-copyleft*) ten sentido se un está a desenvolver unha biblioteca ou un conxunto de ferramentas e o que busca estender o seu uso, sen se preocupar se aplicacións públicas ou privadas se desenvolven a partir dela.

---

59 [http://en.wikipedia.org/wiki/C\\_programming\\_language](http://en.wikipedia.org/wiki/C_programming_language)

60 <http://library.gnome.org/devel/glib/stable/>

61 <http://library.gnome.org/devel/gobject/stable/>

62 <http://check.sourceforge.net/>

63 <http://www.gtk.org/gtk-doc/>

64 <http://git-scm.com/>

65 <http://www.gnu.org/licenses/lgpl-2.1.html>

## 3.4 Itinerario

Na actualidade, *RTM-GLib* pódese empregar para acceder á maior parte dos métodos da API de *Remember The Milk* (algúns deles aínda teñen que ser implementados).

Sexa como for, poderíanse levar a cabo multitude de melloras, como en calquera proxecto de software libre. As máis relevantes a curto prazo poderían ser as seguintes:

- Finalizar a implementación de todos os métodos fornecidos por *Remember The Milk*.
- Adicionar soporte para chamadas asíncronas á biblioteca. Actualmente, a biblioteca só fornece métodos síncronos.
- Adicionar soporte para *Gobject introspection*<sup>66</sup> para facilitar a creación de *bindings* para a biblioteca.

## 3.5 Exemplos de uso

Nesta sección móstranse dous exemplos para comprender como empregar esta biblioteca. O primeiro dos exemplos explica como administrar a autenticación no servizo web de *Remember The Milk*. No segundo, móstrase unha simple chamada a un dos métodos da biblioteca que devolve unha lista de tarefas.

Na pasta de exemplos no arquivo *rtm-glib-example.c* pódese ver un exemplo completo no código fonte.

### 3.5.1 Autenticación

- Crear un novo obxecto *RtmGLib*:

```
RtmGlib *rtm = rtm_glib_new  
(YOUR_RTM_API_KEY,  
YOUR_RTM_SHARED_SECRET);
```

*É necesario pedir unha clave API na páxina web de Remember The Milk*<sup>67</sup>.

---

<sup>66</sup> <http://live.gnome.org/GObjectIntrospection>

<sup>67</sup> <http://www.rememberthemilk.com/services/api/keys.rtm>

- Obter o frob de autenticación:

```
gchar *frob = rtm_glib_auth_get_frob
(rtm, NULL);
```

*Nesta altura, o usuario debería acceder á URL recibida e aceptar os permisos requiridos pola aplicación.*

- Obter o testemuño de autenticación:

```
gchar *auth_token =
rtm_glib_auth_get_token (rtm, frob,
NULL);
```

- Comprobar que o usuario se conectou correctamente:

```
gchar *username = rtm_glib_test_login
(rtm, auth_token, NULL);
```

### 3.5.2 Lista de tarefas

- Chamar o método `rtm.tasks.getList` cun obxecto *RTMGLib* autenticado:

```
GList *glist = rtm_glib_tasks_get_list
(rtm, NULL, NULL, NULL, NULL);

for (GList *item = glist; item; item =
g_list_next (item)) {
    RtmTask *task = (RtmTask *) item-
>data;
    g_print ("%s", rtm_task_to_string
(task));
}
```

## 4. Ideas futuras

### 4.1 Mojito

A idea expresada anteriormente de integrar de algunha maneira *RTM-GLib* e *Mojito* aínda non se levou a cabo. O principal atranco ten a ver co ámbito do proxecto *Mojito*, e que aínda non se decidiu se fornecer ás aplicacións de área de traballo de métodos de acceso a calquera servizo web a través da súa API *D-Bus*.

Na actualidade, *Mojito* centra a súa atención nas redes sociais. Por exemplo, están a engadir soporte para *Facebook*. Porén, podería funcionar para calquera servizo web. Sexa como for, é algo que habería que debater cos desenvolvedores do proxecto (aínda non recibimos noticias da súa parte).

Para fornecer acceso a *Remember The Milk* é necesario implementar un novo servizo de *Mojito*. Xa existe un servizo *dummy* que podería empregarse como base para desenvolver o novo *plugin*, que empregaría *RTM-GLib* como biblioteca base.

Na miña opinión, a integración de *RTM-GLib* en *Mojito* debería constituír un dos principais obxectivos de futuro da biblioteca, polo menos se entrar dentro do ámbito do proxecto e despois de comentalo cos autores. Deste xeito facilitaríase a interacción de calquera plataforma co servizo web de *Remember The Milk*. E sería importante para a biblioteca, xa que implicaría que esas aplicacións estarían a empregar *RTM-GLib* de maneira indirecta, o que podería incrementar o *feedback* sobre *bugs*, novas funcionalidades, etc. Infelizmente, a falta de tempo fixo imposible avanzar neste caso de estudo até hoxe.

### 4.2 Tracker miner

*Tracker*<sup>68</sup> é un proxecto que aspira a recoller información e meta-información sobre datos persoais. Esta información indéxase para poder procurala con rapidez e comodidade. Fornece tanto unha ferramenta de procura como un sistema de almacenaxe para toda esta información.

---

68 <http://projects.gnome.org/tracker/>

Desde o punto de vista do usuario final, *Tracker* non é máis ca un pequeno *applet* que lle permite inserir calquera palabra relacionada con algo do que gusta, por exemplo, palabras dun documento ou o compositor dunha canción. Mais outras aplicacións empregan *Tracker* para, por exemplo, procurar arquivos de vídeo no sistema.

Un *daemon* roda no sistema e indexa cada mudanza que ten lugar no sistema de arquivos, e almacena a meta-información relacionada cos diferentes arquivos na memoria de *Tracker*. Este tipo de *daemons* chámanse *miners*.

Na actualidade, só existe un *miner* oficial, o *miner* do sistema de ficheiros. Porén, no futuro preténdese que aparezan máis *miners*, cada un deles coa función de indexar distintas fontes de datos. Por exemplo, xa hai alguén (Adrien Bustany) que está a traballar nun *miner* para Facebook escrito en *Vala*, que tamén emprega *librest* para interaxir co servizo web. O código fonte deste novo *miner* está nunha rama do repositorio de desenvolvemento de *Tracker*.

Se houber máis *miners*, o usuario podería configurar que *miners* están a ser executados nunha altura determinada e podería activar ou desactivar calquera deles desde un *applet* común. Este *applet* tamén mostraría información sobre o progreso do proceso de indexación.

A respecto dos detalles da implementación á hora de desenvolver un novo *miner*, cómpre estender unha clase abstracta (*TrackerMiner*) e implementar unha serie de métodos para realizar operacións habituais como iniciar, deter, pausar e retomar o *miner*.

Por exemplo, un novo *miner* para indexar as tarefas de *Remember The Milk* podería ser desenvolvido empregando *RTM-GLib* como biblioteca base.

## 4.3 EDS backend

*Evolution Data Server (EDS)*<sup>69</sup> fornece un *backend* común para programas que teñen que traballar con contactos, tarefas e información de calendarios. No inicio era parte de *Evolution* (o cliente de correo electrónico estándar de GNOME), mais na actualidade é empregado por outras aplicacións externas.

---

69 <http://projects.gnome.org/evolution/>

Un exemplo interesante desde o punto de vista de *RTM-GLib* é que o *applet* de calendario do panel GNOME emprega *EDS* para fornecer unha lista dos compromisos e tarefas. Sería estupendo poder sincronizar esas tarefas coas tarefas gardadas na páxina web de *Remember The Milk*.

Para desenvolver esta idea sería necesario crear un *backend* de *EDS*. Este *backend* podería estar baseado no *backend* do calendario de Google, que xa está implementado en *EDS*.

## 5. Conclusión

Na actualidade, as aplicacións web están a vivir un bo momento. O amplo abano de aplicacións dispoñibles na Internet son máis que suficientes para o usuario medio.

Parece que a integración entre área de traballo e aplicacións web ten un futuro prometedor. Actualmente, algunhas aplicacións de área de traballo comezan a interaxir con diversos servizos web. Existen unha serie de estratexias orientadas a posibilitar esta comunicación, e tamén existen multitude de tecnoloxías enfocadas a solucionar este problema dalgunha maneira.

O desenvolvemento da biblioteca foi unha práctica interesante para aprender a tratar con este tipo de problemas de interacción. Tamén contribuíu a profundar os coñecementos sobre a plataforma GNOME e como funciona por dentro. Aínda máis, é un exemplo sinxelo de como se pode solucionar este problema no universo GNOME, no que algunhas ideas como as propostas a respecto de *librest* e *Mojito* poderían converterse en estándar.

Para rematar, dicir que o futuro da comunicación entre a área de traballo e a web é certamente prometedor.





# Epílogo, obra da Asociación GPUL

Moito cambiaron as cousas dende que o GPUL naceu, hai case 15 anos na Facultade de Informática da Universidade da Coruña. Eramos entón un grupo dispar de soñadores, que evanxelizábamos polos corredores da Facultade a todo o que nos quixera escoitar... ben, e tamén aos que non... Confiábamos cegamente nun sistema operativo e nunhas ferramentas libres, nun xeito de aprender, de compartir coñecemento, de desenvolver software.

É posible que naquel momento esa confianza cega nos mostrara ante o mundo coma uns inxenuos. É posible que naquel entón o mundo do software privativo nos superara en medios e en penetración social. Pero iso non nos botou para atrás. Creamos unha infraestrutura para compartir información, organizamos hackmeetings, iniciamos proxectos propios ou iniciámonos nos grandes proxectos de software libre. E como non, comprendemos a importancia de que o software que facíamos ou usábamos estivera na nosa querida lingua galega. Como bos evanxelizadores, medramos lonxe da nosa alma máter, chegando incluso a crear unha sucursal en Ourense, que posteriormente converteuse nun grupo independente, pero irmandado: o GULO.

15 anos no mundo da Informática poden chegar a ser un ciclo. Hoxe xa non é estraño para calquera pequena, mediana ou grande empresa falar de Linux, e aínda que moitos usuarios de a pé non o saiban, levan un kernel libre no seu teléfono móbil. O software libre e, en concreto, os sistemas GNU/LINUX, están máis presentes que nunca na vida de todos. E iso débese ó bo facer da filosofía que impulsou o movemento do software libre dende os seus inicios.

Un dos síntomas de que as cousas cambiaron é que, ademais dos nosos meetings máis ou menos informais, hoxe vemos con alegría iniciativas como o Máster en Software Libre, que proporcionan unha sólida formación en tecnoloxías libres. Deste Máster cónstanos que saíron

magníficos profesionais, a maioría dos cales se reúnen nunha asociación que comparte principios coa nosa. Con GHANDALF existiu feeling de contado, e despois de colaborar na organización dalgúns pequenos eventos, certificamos o irmandamento entre asociacións ó contar coa súa colaboración nun dos mellores e máis grandes eventos sobre software libre que tiveron lugar na nosa terra: a GUADEC 2012.

Dende o principio, GHANDALF nas súas accións mostra unha necesaria preocupación pola documentación do software. É algo que tendemos a esquecer cando estamos perdidos entre miles de liñas de código: que tan importante é desenvolver o proxecto como documentalo. E así chega ás nosas mans esta nova publicación, coa que aprenderemos moito, por suposto, e tamén disfrutaremos. Dende GPUL queremos agradecer e felicitar ós autores, editores e á asociación GHANDALF en global, por este traballo que amplía a documentación sobre software libre en lingua galega. Un paso máis cara a unha sociedade máis libre.

Francisco J. Tsao Santín

Presidente da Asociación GPUL



