

Cliente de Mensajería Instantánea

Guerra, Luis Mariano

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

El objetivo del proyecto es el desarrollo de un cliente de mensajería instantánea libre, multiplataforma compatible con la red Msn Messenger.

La finalidad es brindar un cliente libre que permita al usuario utilizar las últimas características brindadas por el cliente oficial a través de una interfaz de usuario agradable e intuitiva, para esto es necesario el desarrollo de una librería que soporte el protocolo de comunicación en una de sus últimas versiones (MSN Protocol versión 13), para luego diseñar una interfaz que utilice esta librería.

Con el objetivo de brindar la mejor experiencia a la mayoría de usuarios se intenta desarrollar una plataforma flexible que se adapte a las necesidades de cada usuario, tanto en configuración, preferencias, lenguajes y características brindadas.

Teniendo en cuenta los objetivos planteados se elige como plataforma de desarrollo el lenguaje de programación Python[1], y la librería GTK[2][3].

Palabras Clave

Cliente, mensajería instantánea, multiplataforma, Msn Messenger, python, gtk, software libre, GPL[4]

Introducción

El desarrollo del programa surge debido a la ausencia de un cliente de mensajería instantánea libre que tenga soporte para las últimas características brindadas por el cliente Windows Live Messenger. Esta sensación de vacío se puede apreciar al interactuar con usuarios que deciden probar alternativas y no encuentran ninguno que satisfaga sus expectativas. Como primer alternativa se plantea la posibilidad de ampliar el desarrollo de un proyecto existente, surgiendo como alternativas aMSN y Gaim, la primera opción se descarta debido a que dicho cliente utiliza un lenguaje de programación no muy conocido (TCL)[5] y una librería gráfica muy poco intuitiva para el usuario y con un look and feel no muy amigable (Tk)[6]. En cuanto al segundo proyecto se descarta debido a que el objetivo es mucho más amplio al planteado aquí.

Sobre la decisión de que el desarrollo se realice siguiendo el modelo de software

libre, se debe a que el propósito es el de crear una comunidad de desarrollo al rededor del proyecto en el cual todos sus integrantes puedan brindar sus opiniones y aportes para acelerar el desarrollo y crear un producto de mejor calidad y adaptado a las necesidades de sus usuarios. Además el concepto de investigación, desarrollo y aprendizaje no es viables en ambientes cerrados, donde no se ejercita el concepto de libre acceso a la información, debido a que gran parte del estado actual del proyecto se encuentra en los aportes hechos por otras personas que decidieron hacer accesibles sus aportes a quien este interesado.

Elementos del Trabajo y metodología

Ante la decisión de comenzar el desarrollo desde cero se plantea la elección del lenguaje de programación y librería gráfica, para ello se selecciona el lenguaje de programación Python y la librería gráfica GTK. Esta elección se fundamenta en la necesidad de un desarrollo ágil y cuyos resultados se hagan disponibles en múltiples plataformas. El lenguaje de programación Python esta dotado de una gran potencia expresiva que permite realizar tareas complicadas de una manera simple y legible, esto permite un desarrollo veloz de la aplicación en la cual las características nuevas son agregadas rápidamente lo que genera interés por parte de la comunidad al notar una evolución constante. Cabe notar también que el lenguaje de programación es interpretado por lo cual el mismo programa funciona sin ninguna modificación en distintos sistemas operativos. La elección de la librería se debe a los mismos elementos, el soporte de esta librería es amplia y cuenta con una documentación detallada. Cabe aclarar que otra de las razones de la elección es que ambas herramientas tienen licencias compatibles con el concepto de software libre por lo cual su desarrollo es

independiente de cualquier empresa y permite a cualquier interesado el libre acceso a dichas herramientas. Para el desarrollo se utiliza la programación orientada a objetos, utilizando un enfoque modular con bajo acoplamiento entre módulos lo que permite realizar modificaciones en los elementos sin afectar el funcionamiento global, esto es necesario ya que el desarrollo se realizara de manera distribuida teniendo los desarrolladores un repositorio con sistema de control de versiones centralizado (SVN)[7] en el cual los últimos cambios son agregados para que sean accesibles por los demás.

Siguiendo el concepto de modularidad y bajo acoplamiento planteados por el paradigma orientado a objetos se concibe el desarrollo de la librería y el cliente como dos proyectos separados con interfaces bien definidas con lo cual otros proyectos pueden disponer de la librería desarrollada para utilizarla, ampliarla o simplemente estudiarla. Para evitar una base de código de gran tamaño, solo las características esenciales son integradas en el cliente, el cual es dotado de la capacidad de ser extendido a través de complementos (Plugins) para todas aquellas características opcionales, o dependientes de la arquitectura. En este conjunto de plugins se encuentran incluidos el sonido, comunicación entre procesos (DBUS)[8], notificaciones de eventos, configuración personalizada, interacción con programas de reproducción de audio y corrección ortográfica, los cuales se comunican con el cliente a través de un sistema de señales y métodos de retro llamadas. Este sistema permite que los complementos se acoplen a una interfaz definida y documentada pudiendo ser agregados, activados o desactivados en cualquier momento sin necesidad de actualizar el código de base. En este caso la programación orientada a objetos y la modularidad en conjunto con la extensibilidad mediante plugins han permitido un desarrollo veloz y un programa que se adapta a los cambios con facilidad a los cambios sugeridos.

Resultados

Siguiendo los lineamientos planteados en el apartado anterior y luego de 11 meses de desarrollo el cliente a pasado por dos versiones (rama 0.2.x y 1.0.x), actualmente se encuentra en una etapa de desarrollo con la finalidad de realizar el lanzamiento de la versión 1.0 final[9]. El cliente cuenta con la mayoría de las funcionalidades planteadas en el comienzo de su desarrollo. El mismo se encuentra traducido a 12 idiomas (Ingles, Español, Italiano, Portugués, Francés, Alemán, Holandés, Portugues-Brasil, Sueco, Turco, Chino simplificado y Taiwanes), la pagina del proyecto cuenta con 46000 visitas y el programa en todas sus versiones ha sido descargado 6365 veces[10].

En la actualidad el proyecto cuenta con 3 desarrolladores, 2 beta testers y un empaquetador.

Discusión

Como una siguiente etapa del proyecto se han planteado nuevas metas, como el soporte para transferencia de archivos, audio y vídeo y soporte para otros protocolos principalmente Jabber[11]. Algunos usuarios han planteado la utilización de la librería para el desarrollo de otros proyectos, como una capa de transporte para MSN en Jabber, lo que permite la interoperabilidad entre estas dos redes brindando las características innovadoras aportadas por la libreria. Todo esto con la finalidad de desarrollar un cliente para el sistema operativo mac OS X.

Otra propuesta ha sido la de unir esfuerzos para el desarrollo de un framework libre estandarizado con soporte de múltiples protocolos llamado telepathy[12], para así evitar la duplicación de esfuerzos. Todas estas alternativas serán analizadas luego del lanzamiento de la versión 1.0 final.

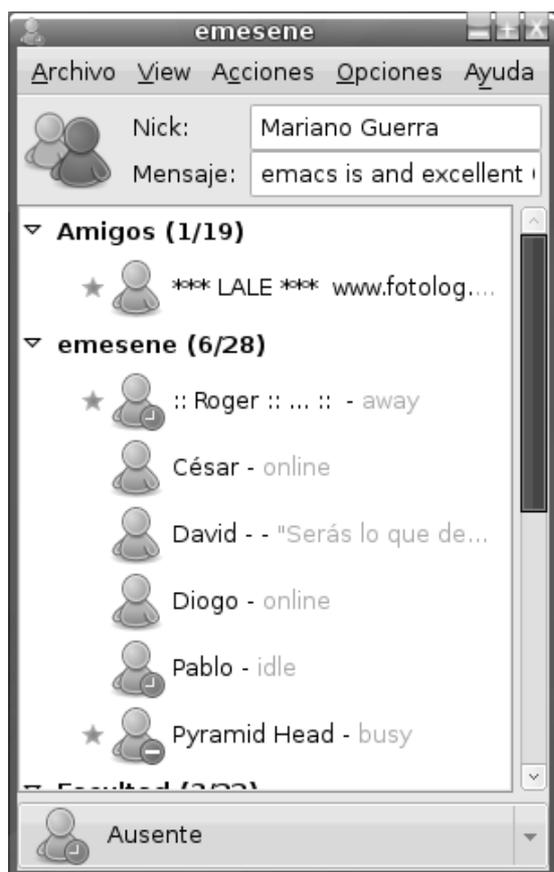


imagen 1

Conclusión

Como resultado se ha logrado el desarrollo de un cliente de mensajería instantánea flexible, configurable, con una interfaz amigable y con soporte de las últimas características brindadas por el cliente oficial. También se ha creado una pequeña comunidad de usuarios que utilizan y ayudan al desarrollo de la aplicación, se ha traducido la aplicación a 12 idiomas y ha obtenido una respuesta interesante por parte de los usuarios. Los objetivos planteados en el inicio han sido logrados y las herramientas seleccionadas han permitido que el desarrollo de la aplicación sea continuo e

incremental. También se han incorporado nuevos conocimientos y herramientas como HIG (Human Interface Guidelines) [13] la cual brinda un marco teórico para la producción de interfaces de usuario intuitivas y uniformes, gettext[14] una librería que facilita la tarea de traducción de aplicaciones y permite a personas sin conocimientos de programación participar sin problemas en la traducción de la misma, desarrollo de módulos para el tratamiento de mensajes asíncronos mediante el uso de hilos, también se han incorporado al desarrollo herramientas como ser Subversion, un sistema de control de versiones, eclipse[15], una plataforma para el desarrollo de aplicaciones etc.

Referencias

- [1] www.python.org
- [2] www.gtk.org
- [3] www.pygtk.org
- [4] <http://www.gnu.org/licenses/gpl.txt>
- [5] <http://www.tcl.tk/>
- [6] <http://www.tcl.tk/software/tcltk/>
- [5] http://msnpiki.msnfanatic.com/index.php/MSN_Protocol_Version_13
- [7] <http://subversion.tigris.org/>
- [8] <http://www.freedesktop.org/wiki/Software/dbus>
- [6] www.amsn-project.net
- [7] www.gaim.sourceforge.net
- [9] <http://sourceforge.net/projects/emesene>
- [10] http://sourceforge.net/project/stats/?group_id=168206&ugn=emesene&type=&mode=alltime
- [11] <http://www.jabber.org/>
- [12] <http://telepathy.freedesktop.org/wiki/>
- [13] <http://developer.gnome.org/projects/gup/hig/>
- [14] <http://www.gnu.org/software/gettext/>
- [15] www.eclipse.org

Datos de Contacto:

Luis Mariano Guerra. Universidad Tecnológica Nacional – Facultad Regional Córdoba. Tucuman 312 2B. luismarianoguerra@gmail.com. Tel: 4217205