

Herramientas 4GL para GNU/Linux

Víctor Castro Mattei

Universidad de Costa Rica, Facultad de Ingeniería,
San José, Costa Rica
camattei@gmail.com

and

Leonardo Jiménez Quijano

Universidad de Costa Rica, Facultad de Ingeniería,
San José, Costa Rica
crleonardo@gmail.com

Abstract

This document describes the four generation languages, its advantages and disadvantages. It describes 4GL open sources tools for software development, and show some examples about their use.

Keywords: 4GL, Four generation languages, Open Source, Free Software, Anjuta, Glade, SAGE

Resumen

Este documento describe los lenguajes de cuarta generación, sus ventajas y desventajas. Describe herramientas 4GL de código abierto para desarrollo de software, y muestra ejemplos acerca de estos.

Palabras clave: 4GL, Lenguajes de cuarta generación, Código abierto, Software libre, Anjuta, Glade, SAGE.

1 Introducción

Cualquier herramienta que mejore el desempeño en aplicaciones de software es deseable, desde el punto de vista de la industria, cada mejora implica un ahorro, desde el punto de vista del programador, cada mejora en la facilidad implica menos desgaste.

El objetivo de los lenguajes de cuarta generación es facilitar y agilizar el proceso de desarrollo de software, y se posiciona actualmente como herramientas de desarrollo estándar de software. Es casi impensable imaginar un IDE que no posea un editor visual.

Los paquetes de código abierto están presentes en esta nueva generación de lenguajes, ofreciendo una alternativa real y seria, como una solución de desarrollo de software.

2 Definición del Lenguaje de Cuarta Generación

Un lenguaje de programación de cuarta generación (abreviado *4GL*) es un lenguaje diseñado con un propósito específico; maneja un subconjunto de operaciones, del total de operaciones que son posibles realizar con un lenguaje de tercera generación, buscando una mayor abstracción y poder de definición.

Los lenguajes de cuarta generación están más cercanos al lenguaje humano si se compara con otros lenguajes de alto nivel, esto permite un rango mayor de usuarios ya que es posible su utilización por parte de personas que no tienen un entrenamiento formal en programación.

Los lenguajes de cuarta generación están diseñados para reducir esfuerzo, tiempo y costo en el desarrollo de una aplicación, ya que solo es necesario indicar que debe hacerse y no como debe hacerse.

3 Pros y contras del 4GL

Al observar a través del tiempo el fenómeno del desarrollo de proyectos de software, se ha logrado determinar que muchas veces las aplicaciones comparten características similares. Para evitar estar implementando una y otra vez la misma funcionalidad en diferentes programas, surge la necesidad de crear herramientas de propósito específico que automaticen y faciliten tales tareas. Dentro de estas herramientas están los 4GL.

La principal ventaja es que la velocidad en el desarrollo de aplicaciones se incrementa, es decir se disminuye el tiempo de implementación. En un ambiente de desarrollo empresarial lo anterior significa una reducción de costos, tanto para el desarrollador como para el cliente.

Estos lenguajes tienden a estandarizar la forma de implementar aplicaciones, ya que sus interfaces de desarrollo presentan muchas similitudes y además las aplicaciones desarrolladas se generan de forma similar.

Los lenguajes de cuarta generación definen un conjunto menor de operaciones que las disponibles en otros lenguajes, esto al tener funciones más específicas. Esta característica facilita el desarrollo de aplicaciones ya que implica aprender a utilizar menos funciones, si se compara con un lenguaje de generaciones anteriores.

Una consecuencia directa de los aspectos tratados anteriormente, los requerimientos para utilizar una aplicación de desarrollo 4GL, son menos estrictos, permitiendo un rango mayor de usuarios, en los que se incluyen personas con conocimientos escasos de programación.

Como todo nuevo paradigma, sus ventajas vienen acompañadas de desventajas que son importantes tomar en cuenta.

Al manejar un subconjunto de operaciones, su funcionalidad se ve limitada, por tanto es importante conocer el dominio del problema, para así determinar si el uso de la herramienta 4GL se ajusta adecuadamente a la solución buscada, puesto que agregar funcionalidad extra se torna complicado.

Si se toma en cuenta el código generado, se hace evidente que es menos elegante y eficiente que uno escrito a mas bajo nivel. El código generado producirá programas que se ejecutan en un tiempo mayor. Pero si se toma en cuenta que los computadores actuales cuentan con gran cantidad de recursos, este aspecto pierde peso.

Desarrollar software en lenguajes de cuarta generación, implica una dependencia de la herramienta utilizada, incluso del proveedor o plataforma, ya que los proyectos generados usualmente necesitan de la misma herramienta para darle mantenimiento. Este aspecto ha hecho que muchas compañías busquen el respaldo de empresas de desarrollo de renombre.

4 Porqué utilizar herramientas 4GL de código abierto

El software libre o de código abierto pone a disposición el código fuente del programa, esto permite la inspección, modificación y extensión de sus funciones. Y de esta manera es posible ajustar un programa a las necesidades específicas del usuario.

Las herramientas de código abierto se desarrollan utilizando el modelo de Bazar[3]. Existe una comunidad mundial que se encarga de la concepción, implementación y mantenimiento de los diferentes proyectos existentes bajo este concepto.

Debido a este desarrollo comunitario, muchas de las aplicaciones tienden a alcanzar un alto grado de madurez, es decir son proyectos robustos, estables y eficientes. Una característica muy importante en este tipo de aplicaciones es que la retroalimentación es muy amplia, en el sentido de que gran cantidad de usuarios acostumbran a informar sobre errores encontrados, aplicar ideas para mejorar e inclusive a adherirse al desarrollo del proyecto.

Las herramientas 4GL libres, proveen las mismas ventajas que las de código cerrado además de las que se mencionan a continuación.

Las licencias de código abierto son más baratas, y usualmente no hay que pagar por el derecho de uso. Estas licencias son adecuadas tanto para un uso académico como comercial, pues es común no distinguir entre estos dos tipos de utilidad, como acostumbra hacerlo el software propietario.

En el ámbito del software libre, la dependencia de una empresa no es problema, pues es una comunidad la que se encarga del mantenimiento, es decir aunque algún desarrollador abandone un proyecto, constantemente se integran nuevos programadores que continúan el desarrollo. Inclusive si el proyecto es abandonado por completo, se cuenta con los fuentes. Esto constituye un respaldo que asegura la continuidad del desarrollo de la herramienta, por parte del usuario o por terceros.

Cabe resaltar que al no depender de una empresa, el soporte se limita al intercambio de mensajes electrónicos con la comunidad, a la consulta de la documentación y a la contratación/consultoría/capacitación de personal.

5 Algunas herramientas 4GL en GNU/Linux

A continuación se presentan unos cuantos ejemplos de herramientas 4GL para GNU/Linux, un sistema operativo libre.

5.1 Anjuta

Anjuta es un ambiente integrado de desarrollo (IDE), para los lenguajes C y C++. Incluye manejador de proyectos, asistentes (wizards), depurador interactivo, y diseñador de interfaces gráficas de usuario basado en 4GL.

Utiliza Glade como componente para el desarrollo de interfaces gráficas. Glade es una herramienta para el desarrollo rápido de aplicaciones (RAD), que utiliza las bibliotecas gráficas GTK+.

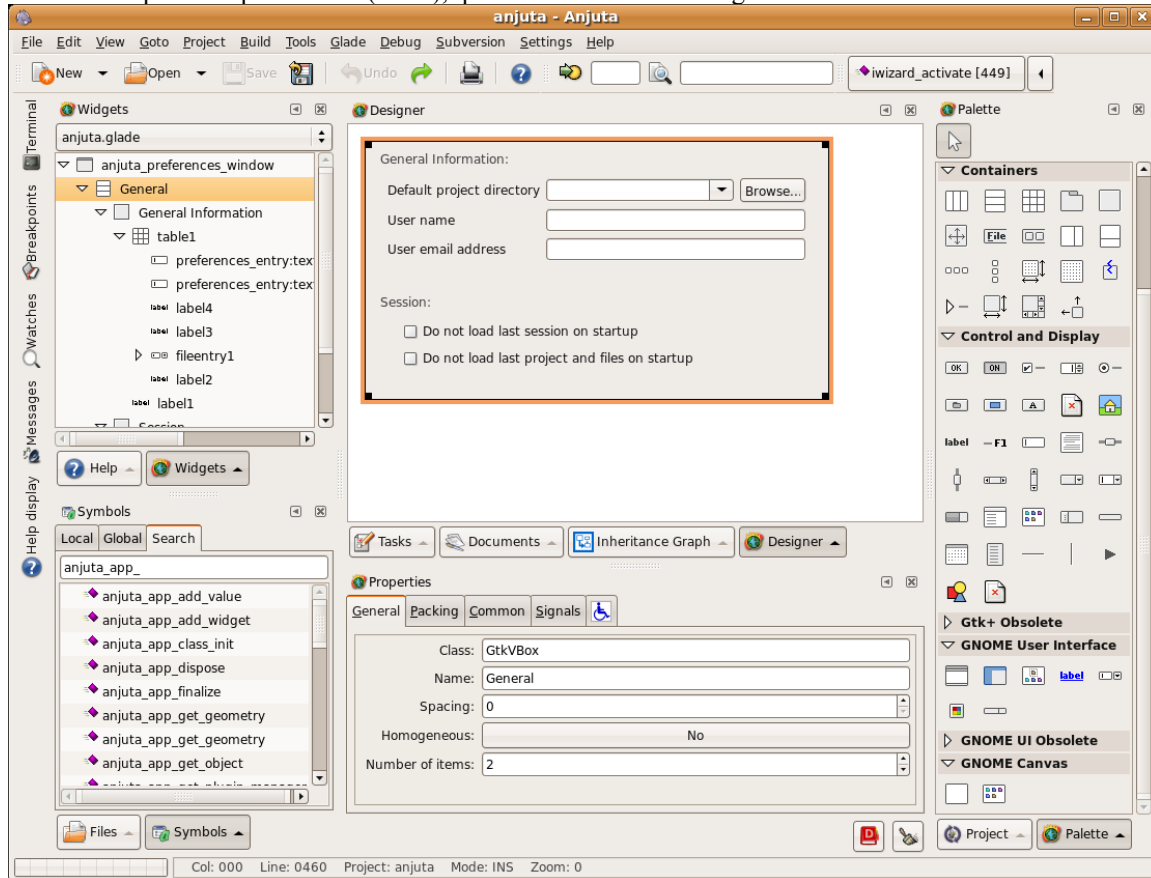


Figura 1. Diseñador de UIs de Anjuta.

Glade se caracteriza por guardar las interfaces gráficas como un archivo XML, y utilizando la biblioteca *libglade*, estas pueden ser cargadas dinámicamente cuando sean necesarias.

Como la mayoría de los IDEs, Anjuta tiene una paleta de widgets que incluye todos los contenedores y controles necesarios para generar GUIs, entre ellos se encuentran listbox, checkbox, buttons, etc.

Anjuta es relativamente liviano, tiene buena documentación, un buen editor de texto e integra de muy buena manera todos los componentes utilizados.

Un ejemplo de una aplicación desarrollada con ayuda del componente Glade es el gPowershot, una herramienta que permite extraer imágenes de una cámara digital y guardarlas utilizando el manejador de bases de datos PostgreSQL.

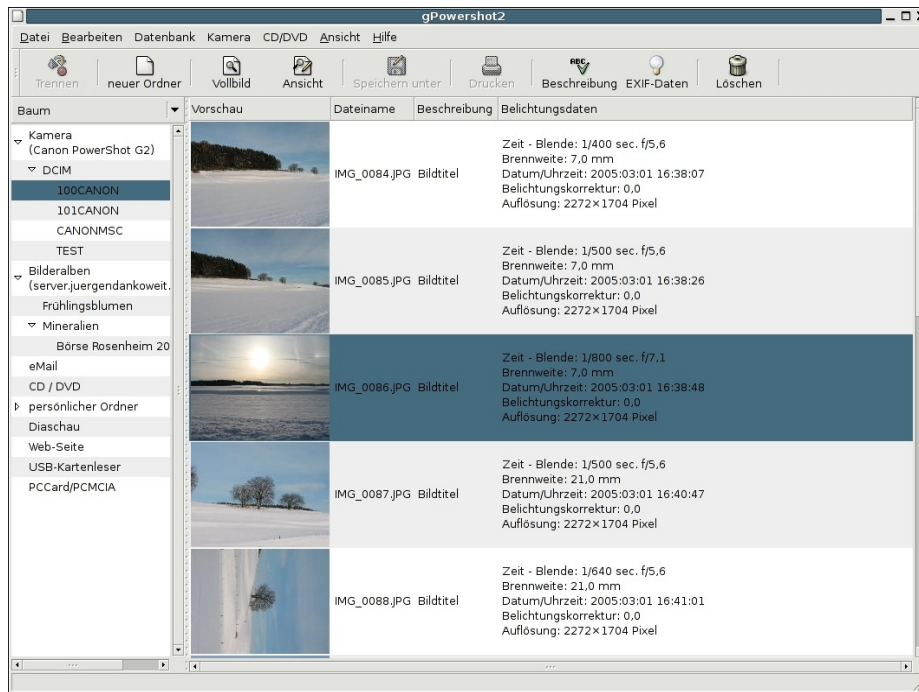


Figura 2. gPowershot.

PHOEBE es otro ejemplo del uso de Glade, es un software astronómico que ayuda a construir modelos a partir de datos reales referentes a velocidad radial.

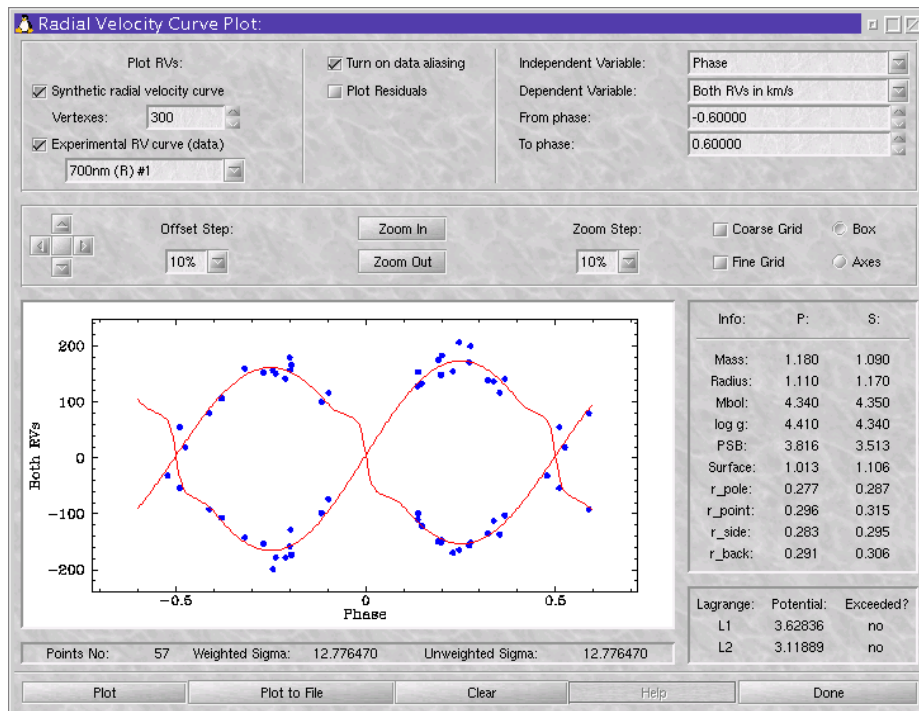
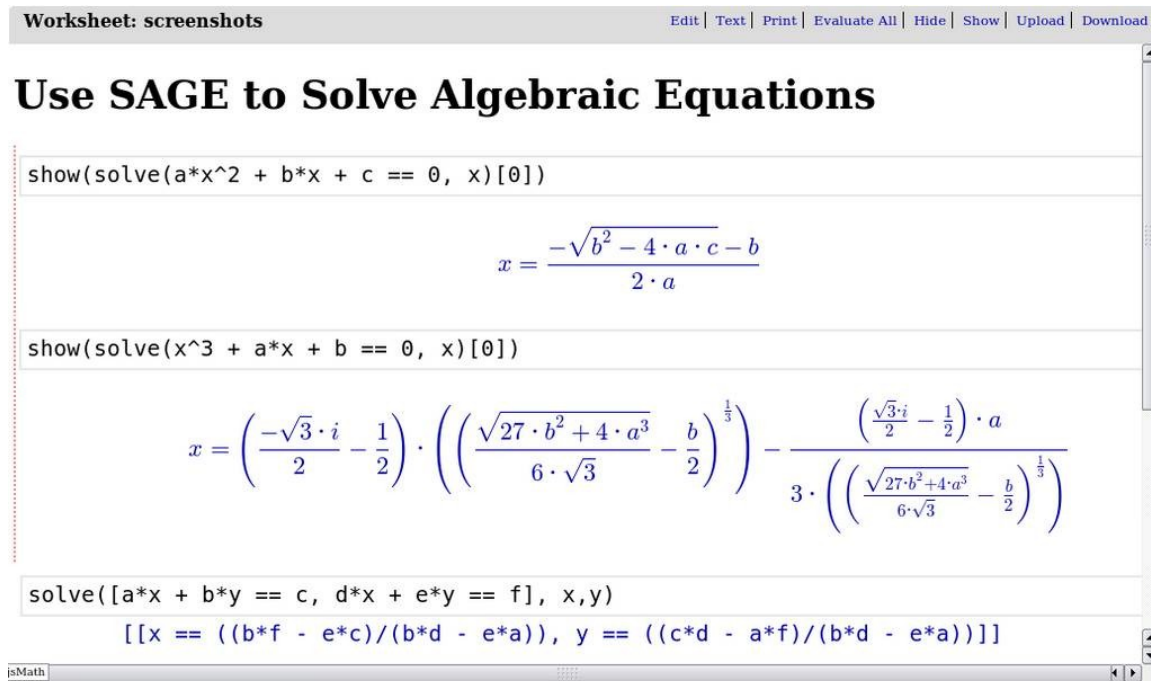


Figura 3. PHOEBE.

5.2 SAGE

SAGE es un paquete de software matemático que incluye soporte para álgebra, cálculo, teoría de números, criptografía, computación numérica, álgebra conmutativa, teoría de grupos, combinatorias, teoría de grafos y álgebra lineal.



The screenshot shows the SAGE web interface with a worksheet titled "3d (sage_notebook)". The interface includes a menu bar with options like "Edit", "Text", "Print", "Evaluate All", "Hide", "Show", "Upload", and "Download". The main content area displays the following code and results:

```
show(solve(a*x^2 + b*x + c == 0, x)[0])
```

$$x = \frac{-\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}$$

```
show(solve(x^3 + a*x + b == 0, x)[0])
```

$$x = \left(\frac{-\sqrt{3} \cdot i}{2} - \frac{1}{2} \right) \cdot \left(\left(\frac{\sqrt{27 \cdot b^2 + 4 \cdot a^3}}{6 \cdot \sqrt{3}} - \frac{b}{2} \right)^{\frac{1}{3}} \right) - \frac{\left(\frac{\sqrt{3} \cdot i}{2} - \frac{1}{2} \right) \cdot a}{3 \cdot \left(\left(\frac{\sqrt{27 \cdot b^2 + 4 \cdot a^3}}{6 \cdot \sqrt{3}} - \frac{b}{2} \right)^{\frac{1}{3}} \right)}$$

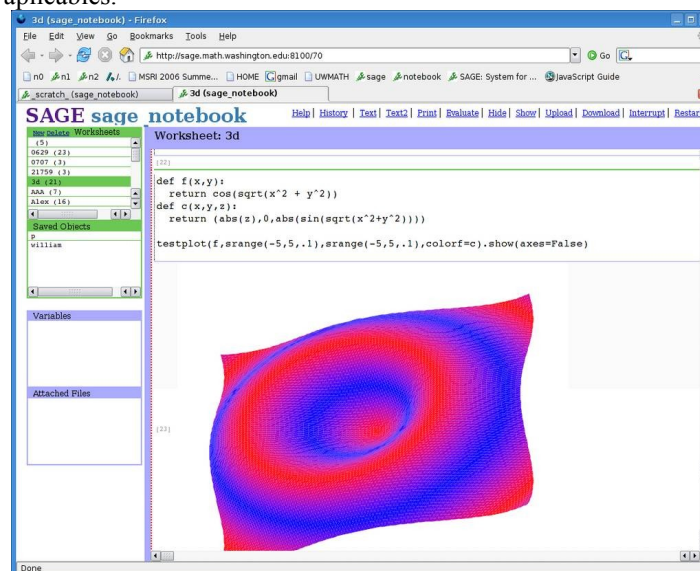
```
solve([a*x + b*y == c, d*x + e*y == f], x,y)
```

$$[[x == ((b*f - e*c)/(b*d - e*a)), y == ((c*d - a*f)/(b*d - e*a)]]$$

Figura 4. SAGE.

SAGE es compatible con aplicaciones como Magma, Maple, Mathematica y MATLAB entre muchos otros. Este paquete ofrece una interfaz web que permite su control desde un navegador web, inclusive remotamente.

Este programa ofrece una buena ayuda que permite resolver rápidamente las dudas que surgen durante el trabajo. Por ejemplo escribiendo Plot? se puede obtener ayuda sobre el comando Plot, y escribiendo Plot?? se podrá ver el código fuente. Escribiendo una palabra, seguida de un punto y la tecla tab, se sugerirán todos los comandos aplicables.



The screenshot shows the SAGE web interface with a worksheet titled "3d (sage_notebook)". The interface includes a menu bar with options like "Help", "History", "Text", "Text2", "Print", "Evaluate", "Hide", "Show", "Upload", "Download", "Interrupt", and "Restart". The main content area displays the following code and a 3D plot:

```
def f(x,y):  
    return cos(sqrt(x^2 + y^2))  
def c(x,y,z):  
    return (abs(z), 0, abs(sin(sqrt(x^2+y^2))))  
testplot(f, xrange(-5,5,1), xrange(-5,5,1), color=c).show(axes=False)
```

The 3D plot shows a surface with a central peak and a surrounding valley, colored with a gradient from blue to red.

Figura 5. SAGE.

6 Conclusiones

- Las herramientas de cuarta generación agilizan los procesos de desarrollo de software, reduciendo el costo monetario y en cuando a tiempo de los mismos.
- Los 4GL proveen un subconjunto de funciones de las que son posibles realizar con el uso de un 3GL.
- Los 4GL tienden a estandarizar las interfaces de usuario.
- Los 4GL ofrecen la posibilidad de que usuarios no programadores puedan utilizarlos.
- Las herramientas libres 4GL liberan al usuario de la dependencia de una empresa.
- El software libre ofrece soluciones reales al proceso de desarrollo de software a través del uso de herramientas 4GL.

Referencias

- [1] WIKIPEDIA, The free encyclopedia. *Fourth-generation programming language*. Visitado: 20 de junio, 2007. URL: <http://en.wikipedia.org/wiki/4GL>.
- [2] ENCYCLOPEDIA BRITANNICA. *Fourth-generation language (4GL)*. Visitado: 20 de junio, 2007. URL: <http://www.britannica.com/ebc/article-9364778>.
- [3] RAYMOND, Eric S. *The Cathedral and the Bazaar*. Visitado: 21 de junio, 2007. URL: <http://gnuwin.epfl.ch/articles/en/cathedralbazaar/cathedral-bazaar.pdf>.
- [4] ANJUTA, integrated development enviroment. Visitado: 22 de junio, 2007. URL: <http://www.anjuta.org>.
- [5] GLADE, user interface builder. Visitado: 22 de junio, 2007. URL: <http://glade.gnome.org>
- [6] GTK+, The GIMP Toolkit. Visitado: 22 de junio, 2007. URL: <http://www.gtk.org>
- [7] GPOWERSHOT. 2. Visitado: 23 de junio, 2007. URL: <http://www.geocities.com/juergendankoweit/gpowershot.html>
- [8] PHOEBE, Physics Of Eclipsing BinariEs. Visitado: 23 de junio, 2007. URL: <http://phoebe.fiz.uni-lj.si>
- [9] SAGE, open source mathematics software. Visitado: 23 de junio, 2007. URL: <http://www.sagemath.org>