

GNU/Linux, software libre para la comunidad universitaria

# Recursos de programación en GNU/Linux

Ramón Rey Vicente

Universidad de Salamanca, 27 de octubre de 2010



**Departamento de  
Informática y Automática**

<http://informatica.usal.es/>

**GLiSA**

**AUGCYL**

<http://www.augcyl.org/>

# Recursos de programación en GNU/Linux

---

- Lenguajes
- Frameworks
- Entornos de desarrollo
- Otros recursos
- Colaborando



# Lenguajes

---

- C
- C++
- Java / C#
- Python / Perl / Ruby
- Otros: Objective-C, Fortran, Eiffel, Haskell



# C

---

- La base de GNU/Linux y UNIX
- Gran cantidad de bibliotecas
- glibc, GCC



# C++

---

- Permite usar muchos de los recursos disponibles en C
- Problemas de estandarización de compiladores
- Ventajas en el diseño e implementación de aplicaciones
- GCC



# Java

---

- Ya es libre !!!
- Existen implementaciones ajenas al control de SUN
- Independiente de plataforma, en teoría
- El mejor camino para migrar de entornos de desarrollo privativos a entornos libres
- JDK, GCJ



# C#

---

- Java de Microsoft
- MONO
- Buen framework de desarrollo, moderno
- Intenta seguir a Microsoft con su .NET
- Incompleto, siempre por detrás
- Complicado migrar de MONO a .NET y viceversa



# Python

---

- Multiplataforma
- Dinámico
- Orientado a objetos
- Sencillo
- Web y escritorio





# Perl

---

- Multiplataforma
- Hereda features de C, sh, Awk
- Potente manejo de XML
- Fácilmente trabaja con C/C++
- DBI, simplifica acceso a BBDD



# Ruby

---

- Multiplataforma
- Dinámico
- Javascript + Python + Java
- Fácil extender en C



# Otros

---

- Haskell
- Eiffel
- Fortran
- R
- ML
- Objective-C



# Frameworks

---

- Java
- Mono / .NET
- QT / KDE
- Gtk+ / GNOME



# Java (I)

---

- Bytecode
- Recolector de basura
- JDBC, acceso a bases de datos
- Aplicaciones web, servlets
- Introspección, plantillas
- Manejo de xml, cifrado y seguridad
- Programación distribuida



# Java (II)

---

- Swing/AWT para aplicaciones gráficas
- Soporte de Ruby además de Java
- ...hasta tiempo real



# Mono / .Net

---

- Clon de Java
- Integra XML, multihilo, seguridad y cifrado
- Interfaz gráfica con gtk#/gnome#
- Soporta Python, Perl, Javascript, VB.net, Java, etc. además de C#
- No es recomendable para multiplataforma



# QT / KDE

---

- Creación de aplicaciones gráficas QT/KDE
- Manejo de multihilo
- XML
- Red
- Tipos básicos portables
- KDE: multimedia, manejo de archivos, componentes con dcop





# gtk+ / GNOME

---

- Creación de aplicaciones gtk/GNOME
- Tipos básicos portables (glib)
- Manejo de gráficos, animaciones
- GNOME: manejo de archivos, multimedia (gstreamer)



# Entornos de desarrollo

---

- Eclipse
- Netbeans
- Anjuta
- Kdevelop
- Vim / Emacs



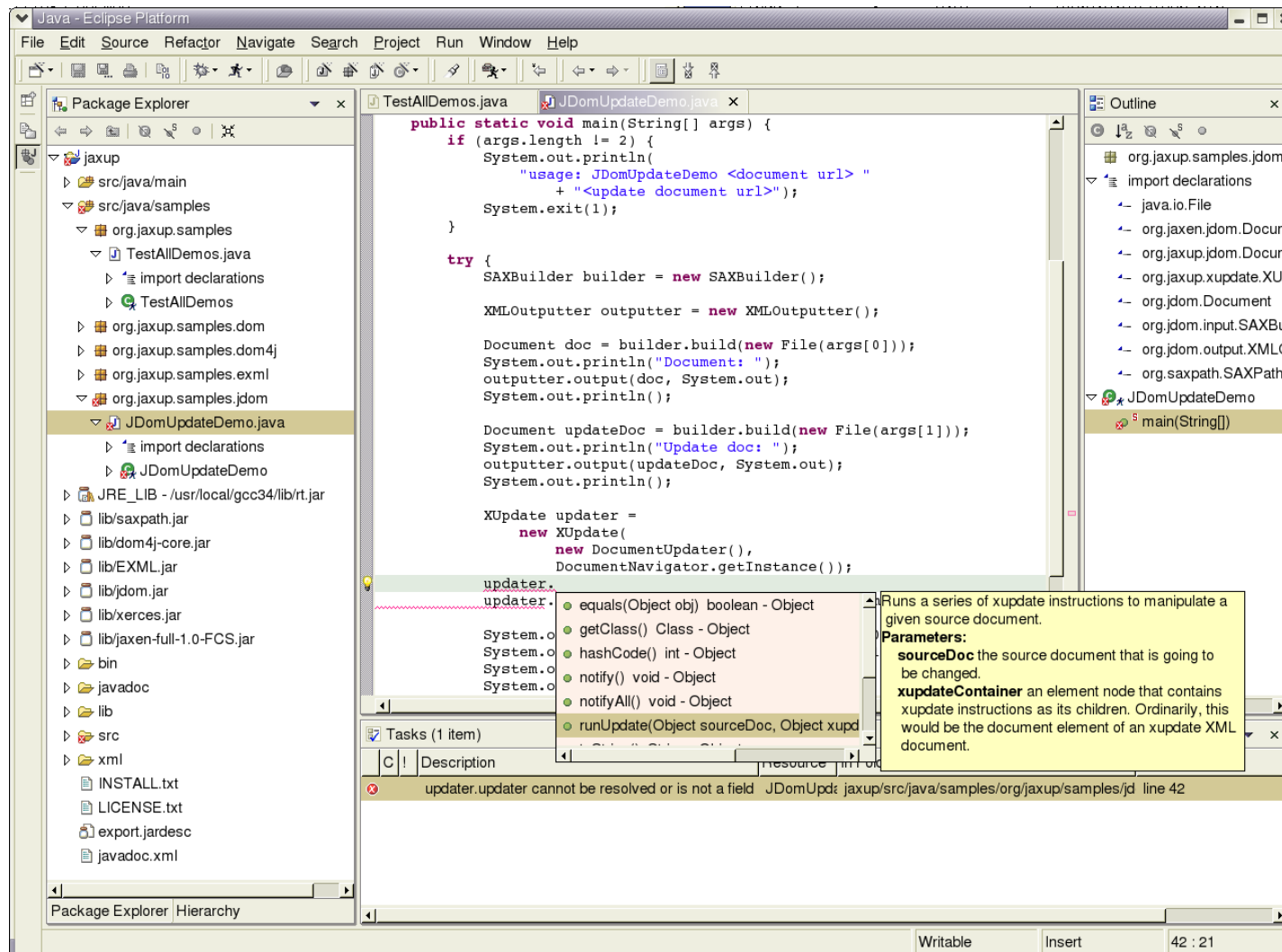
# Eclipse (I)

---

- Multiplataforma
- Toolkit propio SWT, nativo para cada plataforma
- Soporta Java, C++, PHP, a través de plugins
- Soporta múltiples SCM
- Editores visuales (incompletos)



# Eclipse (II)



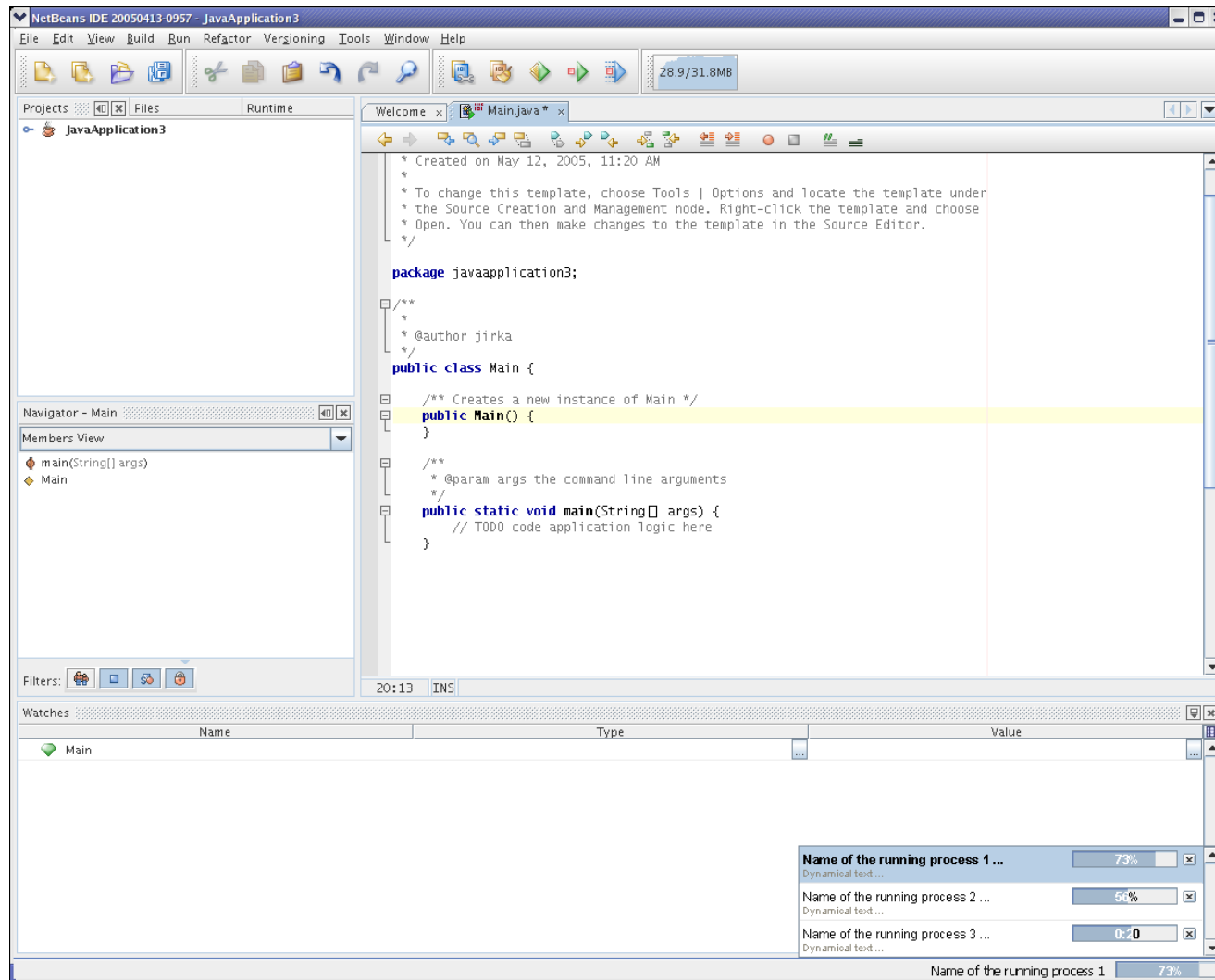
# Netbeans (I)

---

- Java, C++ y Ruby
- Editor visual de aplicaciones Java
- Muy extensible: plugins
- Multiplataforma
- Requiere bastantes recursos



# Netbeans (II)



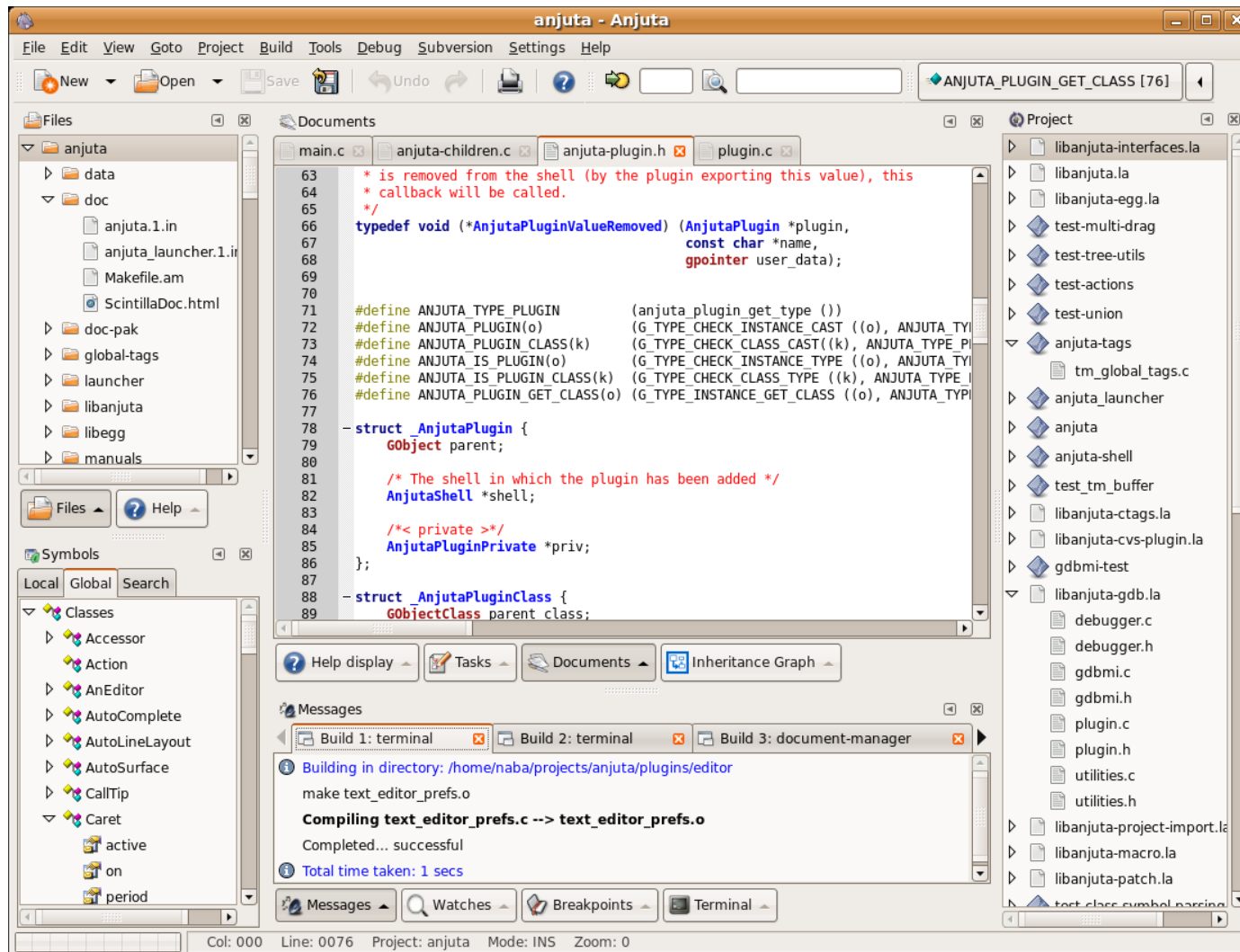
# Anjuta (I)

---

- Orientado a gtk+/GNOME
- Integración con autotools (automake, autoconf, etc)
- Integración con gdb
- Soporte para Pascal, Perl, Java, por archivo
- Ligero, ideal para máquinas modestas

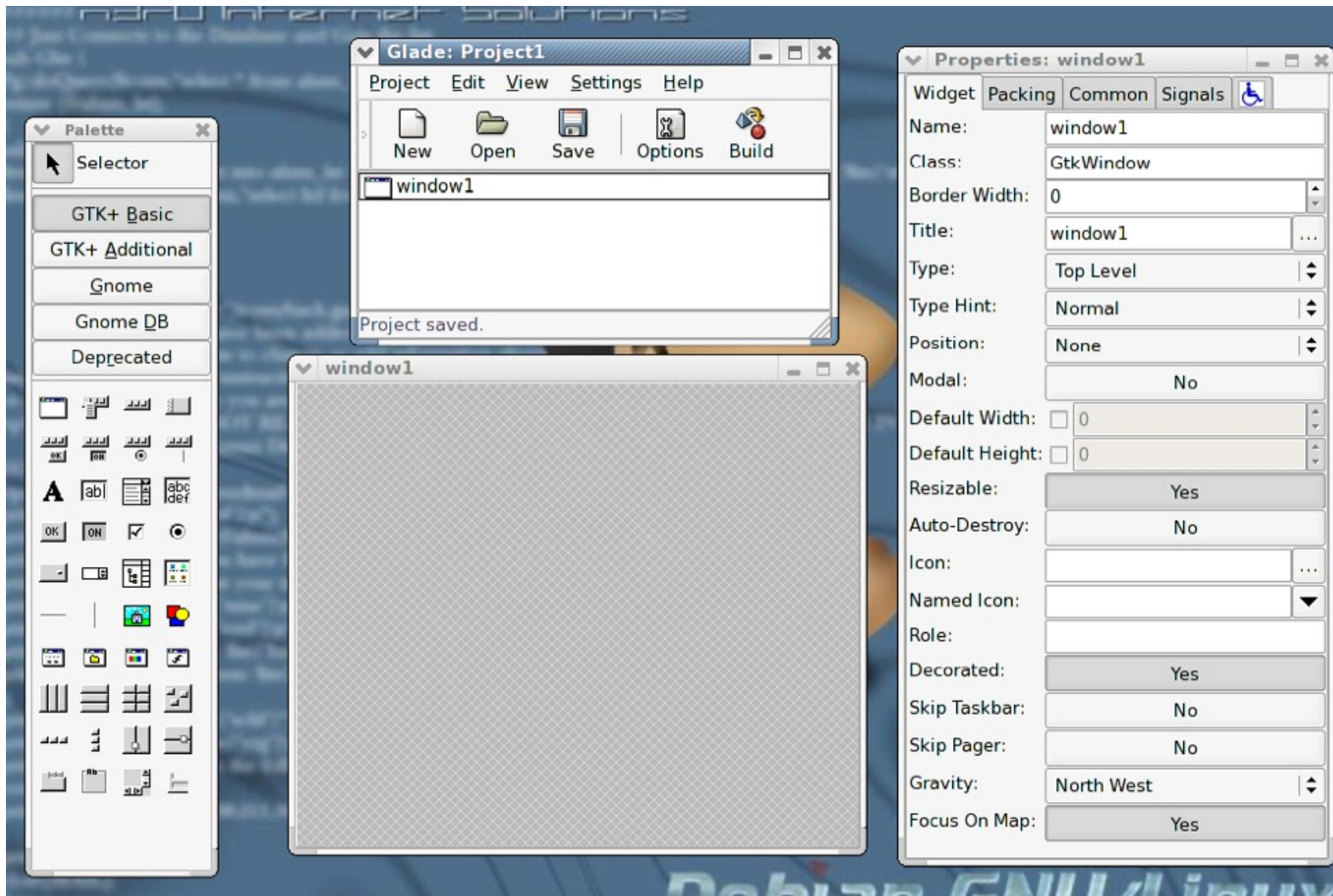


# Anjuta (II)





# GLADE



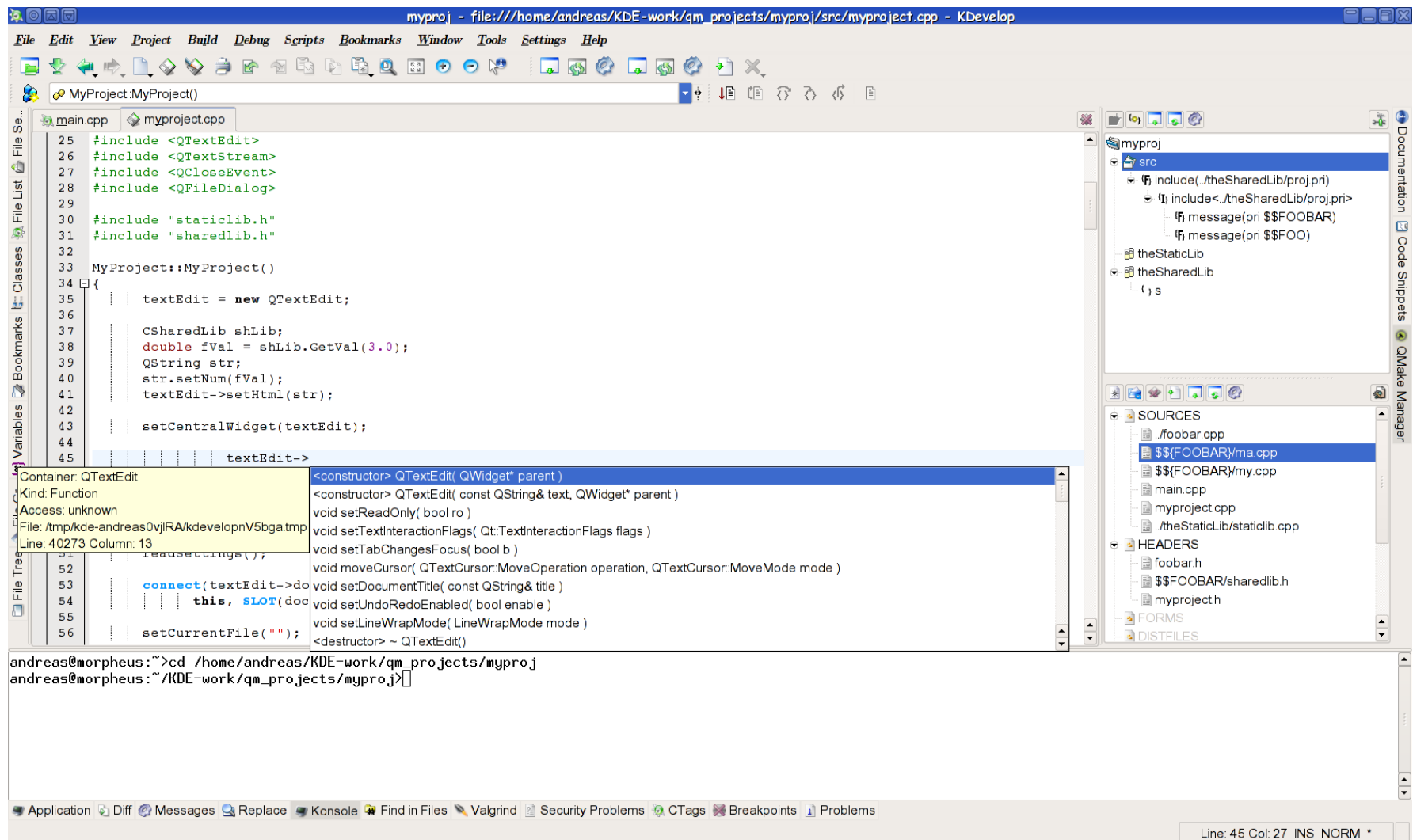
# Kdevelop (I)

---

- Orientado a QT/KDE
- Gran soporte de C++
- Soporta C, Java, Pascal, PHP, Ruby, etc
- Integración con autotools, qmake y ant
- Ligero



# Kdevelop (II)



# Vim (I)

---

- Editor con muchas posibilidades
- Soporta virtualmente cualquier lenguaje
- Únicamente para editar código
- Obliga a conocer autotools, etc.
- Ideal para máquinas muy modestas



# Vim (II)

```

file.c (/storage/projects/nimf/contents/nimf) - GVIM
File Edit Tools Syntax Buffers Window Extended Help

file_lastdir
file_ext
file_nicesize
file_niceusername
file_nicegroupname
file_nicemode
file_copy
file_move
file_del
file_mkdir
file_lnk
file_chmod
file_chown
dir_chmod
dir_chown
dir_copy
dir_del
dir_move
report_done
report_error
report_progress
action_read
filelist_copy
filelist_move
filelist_mkdir
filelist_del
filelist_lnk
filelist_chmod
filelist_chown

operations.c (/storage/projects/nimf/contents/nimf/operations.c)
function
op_start
op_progress
op_progress_readerror
op_progress_readprogress
op_edit

dialogs.c (/storage/projects/nimf/contents/nimf/dialogs.c)
tag List
Press ? for keyboard shortcuts
Sorted by name (.bak, ~, .swp,
"/storage/projects/nimf/contents/nimf/contents/nimf/008680170000000b
BUGS
CHANGELOG
COPYING
INSTALL
Makefile
README
RELEASENOTES
TODO
bookmarks.c
bookmarks.h
cmdline.c
cmdline.h
config.c
config.h
context.c
dialogs.c
dialogs.c-backup
dialogs.h
file.c
file.h
gtkaddons.c
gtkaddons.h
logo.h
nimf
nimf.c
nimf.h
operations.c
operations.h
panel.c
panel.h
settings.c
settings.h
sysinfo.c
sysinfo.h

file.c
report_done (reporttype);
report_progress (reporttype, fullpath, NULL, (int) ((float)i/roffiles+100));
while (cur && op_stop != 1) {
    char *fullpath = malloc (sizeof(char) * (strlen(cur->data)+strlen(srcpath)
    + 2));
    int error = 0;
    report_progress (reporttype, fullpath, NULL, (int) ((float)i/roffiles+100));
    if (strcmp(cur->data, ".") != 0) {
        /* XXX: Fullpath function? */
        sprintf (fullpath, "%s/%s", srcpath, (char *)cur->data);
        printf ("fullpath %s", fullpath);
        i++;
        switch (file_type_withlink (fullpath)) {
            case FTYPE_NORMAL : error = chown (fullpath, uid, gid); break;
            case FTYPE_DIR :
                if ((options & OPT_CHOWN_RECURSE) == OPT_CHOWN_RECURSE) {
                    printf ("filelist_chown: recurse\n");
                    error = dir_chown (actionpipe, reporttype, fullpath, uid, gid);
                }
                error = file_chown (fullpath, uid, gid);
                break;
            case FTYPE_LINKNORMAL : error = file_chown (fullpath, uid, gid); break;
            /* XXX: default: unknown file? */
        }
        report_progress (reporttype, fullpath, NULL, (int) ((float)i/roffiles+100));
        switch (error) {
            case -1:
                report_error (reporttype, fullpath, NULL);
                if (action_read (actionpipe) == 1) {
                    op_stop = 1;
                }
                break;
            case 1:
                op_stop = 1;
                break;
            default:
                break;
        }
        free (fullpath);
    }
    cur = cur->next;
}
report_done (reporttype);

dialogs.c
#include <sys/stat.h>
#include <stdlib.h>
#include <sys/types.h>
#include <pwd.h>
#include <grp.h>
#include <sys/types.h>
#include "dialogs.h"
#include "logo.h"
#include "file.h"
/* These must be static for some reason, or the dialog can't be
 * re-used. If anybody knows why, please let me know */
static GtkWidget *dialog;
static GtkWidget *entry;
void dialog_error_ev_button_clicked (
    GtkWidget *widget,
    GtkWidget *win) {
    gtk_widget_destroy (win);
}
void dialog_about_btn_ok_cb (GtkWidget *widget, GtkWidget *win) {
    gtk_widget_destroy (win);
}
void dialog_about (GtkWidget *widget, gpointer data) {
    GtkWidget *win, *pixmapwid;
    GtkPixmap *logo;
    GtkStyle *style;
    GtkWidget *label;
    GtkWidget *vbox;
    GtkWidget *frame;
    GtkWidget *btn_ok;
}
dialogs.c
43,37
32
Nimf: ChangeLog
-----
Please do not spread the email addresses in this file.
[XXVERSION] [XXEMAIL] ( XXDATE )
 * Panel : Optimizations in the calling of panel_info_update. Makes large
          selections faster.
 * Panel : Added chmod dialog and functionality.
 * Panel : Refresh can now be forced so complete dir is reread instead of
          diffed with in-memory version.
 * Operations: Current file that is being worked on is now shown in progress.
[0.1.0] [XXEMAIL] ( Wed May 14 18:22:31 CEST 2003 )
 * Config : The F6 is rename instead of move bug has been fixed. You'll
          have to change the keysum for F6 to operation_move in the
          key-settings dialog or remove your old ~/nimf/keys config.
 * Panel : Bug fix for the bug where hidden files wouldn't show.
          panel->showhiddenfiles is now initialized at 0.
[0.0.21] [XXEMAIL] ( Mon May 5 15:33:05 CEST 2003 )
 * Panel : Letters are no longer added to Jumpsearch if they yield no
          result.
 * Panel : Fixed the toggle-hidden-button bug in which the key-combo for
          toggling hidden file-view wouldn't work because it recursively
          triggered the 'clicked' event for the button, which in turn
          called the panel.toggle_hidden() function. The key handling
          now just emits a clicked signal.
 * Jumpsearch: Fixed bug where you couldn't search for files starting with
          a period.
 * Config : Fixed buffer overflow bug in config_get_key().
 * Settings : Multiple user interface enhancements were made. Keyboard
          handling is better. Connect fields are automatically focussed.
 * Cmdline : Initial version of the commandline bar.
file.c 1282,1-4 Bot #1792103
|| gcc -Wall -c -x gtk-config --cflags file.c
|| cc nimf.o sysinfo.o panel.o cmdline.o file.o operations.o dialogs.o config.o bookmarks.o settings.o gtkaddons.o -o nimf -g -Wall -x gtk-config --cflags --libs `glib-config --libs gthread`
knts/nimf [RO] 4,1 Top [Error List]
1,1 All

```



# Emacs (I)

---

- Editor altamente extensible
- Soporta cualquier lenguaje
- Integración con sistemas de control de versiones, compiladores, debuggers, etc.
- Es un IDE sin ser IDE



# Emacs (II)



```
ADPAgentC.java

public class ADPAgentC extends ADPAgentC {
    Random r;

    int[] fixedpolicy;
    double[] utilities;
    double[] reward;

    int numStates;
    int numActions;
    double discount;

    double[][][] TransModel;
    double[][] Nsa;
    double[][][] Nsasp;

    int[] statevisited;
    int prevState;
    int prevAction;

    PolicyEvaluation policyEval;

    public ADPAgentC(Mdp mdp) {
        super(mdp);

        this.r = new Random();

        this.numStates = mdp.numStates;
        this.numActions = mdp.numActions;
    }
}

public class ADPAgentC extends ADPAgentC {
    Random r;

    int[] policy;
    double[] utilities;
    double[] reward;

    int numStates;
    int numActions;
    double discount;

    double[][][] TransModel;
    int[][] Nsa;
    int[][][] Nsasp;

    int[] statevisited;
    int prevState;
    int prevAction;

    PolicyEvaluation policyEval;

    ExplorationFunction explorationFunction;

    final int threshold = 5;
    final double unexploredValue = 1000.0;
}

A: --:%% !home!phil!school!6.825!proj3! B: --:-- ADPAgentC.java 4% of 3.4k (
```



# Otros recursos

---

- SCM: control de versiones
- Buscando software
- Alojamiento
- Licencias





# SCM

---

- Gestión del código fuente
- Control de versiones
- SVN, el bueno
- git, arch, bazaar, los feos
- CVS, el malo



# Subversion (SVN)

---

- Desarrollo centralizado
- Sencillo
- Compatible con comandos del CVS
- Soporta directorios
- El nuevo estándar para los habituados al CVS



# CVS

---

- Desarrollo centralizado
- Sencillo
- Muy estable
- Un estándar
- Obsoleto



# SCM's distribuidos

---

- git
- arch
- bazaar



# git

---

- Sencillo
- Complejo
- Potente
- Flexible
- Escalable



# Buscando software

---

- Freshmeat
- Sourceforge
- Directorio del proyecto GNU
- Savannah
- Proyecto Debian
- GNOME, KDE...



# Alojamiento

---

- Savannah
- Sourceforge



# Licencias

---

- GPL todos somos iguales ante el código
- LGPL somos iguales excepto los desiguales
- BSD / MIT X11 somos iguales los que queremos serlo





# Colaborando

---

- Elegir un proyecto
- Elegir un módulo o componente
- Elegir un tarea o bug
- Desarrollar la solución
- Exponerla a los miembros del proyecto
- Reimplementar



# Elegir un proyecto

---

- Según lenguaje de programación
- Según la licencia
- Según el tamaño
- Según la ayuda y la comunidad creada



# Elegir un módulo

---

- Según el nivel de cada uno
- Sencillez para empezar
- Pensar funcionalidades no presentes



# Elegir una tarea

---

- Bugzilla
- TODO
- Roadmap
- Bounties (¬)



# Implementar solución

---

- Seguir las convenciones y guías de estilo
- Realizar los cambios mínimos necesarios en el código de los demás
- El software libre no es solo Linux
- Considerar la licencia del proyecto



# Mostrar el trabajo

---

- Mostrar el trabajo a la comunidad
- Buscar la aceptación del mantenedor del proyecto
- Modificar y reimplementar todo en función de la respuesta
- Trabajo colaborativo



# Reimplementar

---

- Aceptar las críticas
- Argumentar las decisiones
- Buscar la excelencia técnica
- Y finalmente, conseguir que acepten el parche



# Fin



**Departamento de  
Informática y Automática**  
<http://informatica.usal.es/>

**GLISA**

**AUGCyL**  
<http://www.augcyl.org/>





Esta obra está bajo una licencia Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 España de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/es/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

