

ENSINO DE INFORMÁTICA USANDO LINGUAGENS DE PROGRAMAÇÃO BASEADAS EM SOFTWARE LIVRE

Fernando Santos Osório

UNISINOS – Universidade do Vale do Rio dos Sinos
Centro de Ciências Exatas e Tecnológicas – Curso de Informática
Av. Unisinos, 950 – São Leopoldo, RS – Cep: 93022-000

E-Mail: osorio@exatas.unisinos.br – Web: <http://www.inf.unisinos.br/~osorio/>

Resumo:

Este trabalho tem por objetivo discutir sobre o uso de linguagens de programação baseadas em ferramentas de Software Livre, para serem aplicadas junto aos cursos de formação de recursos humanos na área de Informática. Dentro deste contexto, vamos apresentar algumas possíveis configurações de ambientes que vem sendo utilizados para este fim. Em particular vamos destacar as experiências práticas do uso deste tipo de ferramentas dentro de um ambiente acadêmico universitário, como é o caso da UNISINOS.

Abstract:

This work presents an overview on using free software programming tools. These tools are mainly used for human resources formation in the field of Computer Science. We will show possible framework configurations used in order to achieve this goal. In particular, we will describe practical experiences accomplished at UNISINOS University.

1. Introdução

O ensino dos princípios básicos sobre programação de computadores necessita de uma aplicação prática destes conceitos teóricos, para uma melhor compreensão por parte dos alunos, através do uso prático de uma linguagem de programação. Atualmente, existem diversas linguagens de programação no mercado, mas no entanto, é interessante questionar se um aluno precisa realmente usar exatamente as mesmas ferramentas que ele poderá vir a empregar em sua vida profissional. Uma corrente seguida por diversos grupos de especialistas na formação de recursos humanos em informática, advoga que o essencial é o ensino dos principais conceitos ligados a programação (algoritmos, tipos e estruturas de dados, programação estruturada e modular, orientação à objetos, etc), independentemente da escolha de uma “*linguagem específica da moda*”. Essa afirmação se justifica pelo fato de que as linguagens de programação estão em constante evolução, e sendo assim, não adianta formar um especialista na linguagem X ou Y, se a tendência é de que muito em breve tais linguagens possam ser substituídas por novos padrões adotados pelo mercado. Afinal, muitas das linguagens largamente utilizadas na atualidade, como VisualC, Borland C++ Builder, Java e Delphi, possuem um curto tempo de aplicação junto ao mercado profissional em torno de 5 a 10 anos.

Outro fator que dificulta o uso das “ferramentas comerciais da moda” para o ensino de programação são os custos. Estas ferramentas possuem um custo bastante elevado, e devem ser constantemente atualizadas com novas versões, o que aumenta ainda mais este custo. O uso destas ferramentas comerciais no desenvolvimento de aplicações onde se necessita de uma alta produtividade e de recursos bastante avançados é plenamente justificável. No entanto, o ensino de programação deve ser realizado usando ferramentas que permitam ao aluno aprender novos conceitos, onde a produtividade e os recursos avançados podem ser colocados em segundo plano. Até mesmo a separação do programa de sua interface com o usuário deve ser incentivada, pois estas são mais um complemento, e não devem constituir a base do ensino de uma linguagem.

Em consequência destes fatores já citados: (i) a necessidade de um aprendizado de conceitos teóricos, e não de particularidades de certos produtos, (ii) o alto custo da aquisição e atualização de ferramentas comerciais; assim como, (iii) a necessidade de treinamento visando criar um profissional apto a trabalhar em múltiplos ambientes e plataformas (e.g. Unix, Apple/Mac, Windows), (iv) a crescente necessidade de desenvolver aplicações portáteis e independentes (“livres”) de plataformas específicas, (v) e devido ao crescente incentivo por parte dos organismos públicos para que se adote o uso de ferramentas baseadas em Software Livre em suas repartições¹; todos estes fatores nos levam à busca de ferramentas e de ambientes de Software Livre que possam ser utilizados no ensino de Informática.

Um outro ponto de grande importância a ser discutido no ensino da Informática é referente a escolha da primeira linguagem a ser ensinada a um aluno. Existe uma grande polêmica a esse respeito, sendo que algumas instituições adotam linguagens como: Fortran, Pascal, C, C++, Modula, Java, Prolog, Lisp, Basic, entre outras. Não temos a intenção de apresentar neste trabalho uma proposta nesse sentido, visto que acreditamos que este tipo de discussão depende das particularidades de cada curso e dos objetivos a que nos propomos ao ensinar uma linguagem. Porém, acreditamos que é possível escolher um conjunto de linguagens que forma um grupo representativo das linguagens/paradigmas mais usados na atualidade. Este conjunto é formado pelas linguagens *Pascal* (Paradigma Estruturado e Procedural), *C / C++* (Paradigmas Procedural / Orientado à Objetos) e *Java* (Paradigma Orientado à Objetos / Programação Concorrente). Neste trabalho não vamos tratar dos Paradigmas Funcional e Lógico, pois pensamos que estes deveriam ser discutidos separadamente dos demais.

No item seguinte, em função das três linguagens principais que selecionamos – Pascal, C/C++ e Java, vamos apresentar como constituir ambientes de ensino para cada uma destas linguagens, baseados na utilização de Software Livre atualmente disponíveis. Juntamente com esta apresentação, iremos discutir sobre as algumas das particularidades (limitações, características e vantagens) de cada ambiente. Ao final iremos relatar as experiências que tem sido feitas na UNISINOS no sentido de adotar ambientes baseados em Software Livre nesta instituição, para fins educacionais e de pesquisa.

2. Ambientes de ensino de programação baseados em Software Livre

Um ambiente de ensino de uma linguagem de programação deve possuir alguns requisitos básicos para que o aluno tenha um mínimo de “conforto” ao usá-lo. Um conjunto ideal de ferramentas é obtido quando podemos integrar dentro de um único ambiente², com uma única interface amigável e padronizada, as seguintes ferramentas:

1. Editor de textos e de programas que seja flexível e configurável;
2. Compilador de programas, com fácil ajuste das bibliotecas e de seus parâmetros;
3. Depurador (*debugger*) de programas, permitindo execução passo-à-passo, uso de *break-points* e exibição/alteração do conteúdo das variáveis;
4. Construtor de interfaces, permitindo a criação de interfaces através de uma “linguagem visual” de forma interativa;
5. Acesso a informações de ajuda (*Help On-line*);

Além destas características, vamos também considerar a possibilidade de escrever programas que sejam:

¹ Medidas neste sentido vem sendo incentivadas e adotadas pelo Governo do Estado do RS, pelo Governo Federal, pela Comunidade Européia e pela República da China.

² Ambiente Integrado de Desenvolvimento – **IDE** (Integrated Development Environment). Um exemplo de IDE é o RHIDE [8] que permite a integração de diferentes compiladores sob uma mesma interface de edição, compilação e depuração de programas.

6. Modulares com possibilidade de integração de extensões (gráficas, multimídia, banco de dados, Internet, etc);
7. Portáteis para diferentes plataformas: (a) Linux, (b) Dos, (c) Windows 95/98, (d) Windows NT, (e) Windows em Geral (Janela Dos), (f) Solaris, (g) Unix em Geral, (h) MacOS, (i) OS/2;
8. Compactos e se possível bem otimizados;

Também vamos considerar algumas das características das ferramentas de desenvolvimento empregadas, no que diz respeito aos seguintes itens:

9. Disponibilidade do código fonte da ferramenta, possibilitando que esta seja portada para outros ambientes, ou mesmo adaptada segundo necessidades específicas do usuário;
10. Documentação completa e atualizada da ferramenta;
11. Atualizações regulares com correções de problemas (*bugs*) e implementação de novos recursos;

A tabela abaixo procura resumir as possibilidades de composição de ambientes integrados de desenvolvimento de programas nas linguagens Pascal, C/C++ e Java, utilizando Softwares Livres, e considerando os itens citados acima.

Linguagem	Software	Características (baseadas nos itens acima)
PASCAL	GNU Pascal GPC [5]	1,2,3,5 - Através do uso do IDE Rhide[8] em Linux ou Dos 7(a,b,e,f,g) - Disponível em ambientes Unix e Dos (DJGPP) 9, 10, 11 - Mantido pela organização GNU [5]
	Free Pascal [4]	- Software que busca ser compatível com o Turbo Pascal 7 e o Delphi 4 da Borland (recursos do Delphi ainda em desenvolvimento). 1,2,3,5 - Através do uso do IDE Rhide[8] em Linux ou Dos 7(a,b,d,e,i) - Disponível em plataformas Windows/Intel 9,10,11 - Muito bem documentado e atualizado regularmente.
C / C++	GNU C e C++ GCC e GPP/G++ [5] DJGPP (Dos) [3]	- O gcc/gpp possui versões em ambientes Unix e Dos que são compatíveis, o que facilita o porte de aplicações entre estes dois ambientes. No Dos é usado o DJGPP [3]. 1,2,3,5 - Através do uso do IDE Rhide[8] em Linux ou Dos 4 - Em ambiente Linux/Unix é possível o uso do Glade [6] 6 - Biblioteca em C para integração de recursos gráficos e de multimídia em ambientes Linux/Dos - Allegro [9] 7(a,b,e,f,g) - Disponível em ambientes Unix e Dos (DJGPP) 9, 10, 11 - Mantido pela organização GNU [5]
	Borland C++ 5.5 [1] Command-Line Compiler	- A Borland (empresa comercial) disponibilizou o compilador oferecido em seus produtos, mas apenas na versão de linha de comandos (<i>command-line</i>), não possuindo nesta distribuição um ambiente IDE. 7(b,e) -Disponível em ambientes Dos (ou no Windows) 8, 10 - Código bastante eficiente, compatível com as demais ferramentas da Borland.
JAVA	Sun Javasoft [7] Java 1.2 SDK Java Forte IDE	- O SDK é a versão linha de comandos, onde recentemente a Sun disponibilizou uma IDE para o Java2 (1.2), o Forte. 1,2,3,4,5 - Java Forte IDE (inclui editor, debugger, etc) 6 - Diversas bibliotecas de extensão (gráficas, Internet, etc) 7(a,b,c,d,e,f,h) - Programas portáteis para outros ambientes que rodem Java (ambientes que executam o JVM) 8 (pouco otimizado), 9 (parcial), 10 (boa), 11 (muito atualizado)

JAVA	Borland Jbuilder 3 Foundation [2]	- O JBuilder permite a criação de programas portáveis nas plataformas Linux, Solaris e Windows. Assim como as ferramentas da Sun, esta é uma aplicação bastante pesada (em termos de hardware) e gera código não muito otimizado. 1,2,3,4,5- IDE com ferramentas visuais de desenvolvimento 6, 7 - Possui as características da linguagem Java 8 (pouco otimizado), 10 (boa)
------	-----------------------------------	---

* Os itens *não* indicados em alguns dos softwares presentes nesta tabela são, em alguns casos, itens onde não possuímos uma indicação clara da existência/possibilidade do software possuir tal item.

O quadro acima demonstra que temos ótimos softwares para o aprendizado de programação baseados em Software Livre. Uma das maiores restrições impostas é a falta de um ambiente de desenvolvimento visual de interfaces multi-plataforma nas linguagens C/C++ e Pascal, o que pode ser compensado pela existência deste tipo de ferramentas para a linguagem JAVA. No entanto, a linguagem JAVA requer maiores recursos de hardware (mais memória e CPUs mais rápidas), o que pode representar *atualmente* um custo mais elevado na implantação de um ambiente usado no aprendizado baseado nesta linguagem.

3. Experiências e resultados do ensino de Informática usando Software Livre

A Unisinos, e mais especificamente os cursos ligados a área de Informática, tem utilizado tanto softwares livres, como também softwares comerciais em suas atividades de ensino e pesquisa. O curso de graduação em Informática utiliza como primeira linguagem o Pascal, onde atualmente vem se estudando a possibilidade de adoção do Free Pascal no lugar do Turbo Pascal da Borland. O ensino da linguagem Pascal permite a criação de bons hábitos de programação, bem como a sua proximidade da linguagem Delphi (amplamente utilizada no meio empresarial) se constitui num atrativo a mais para os alunos do curso de graduação. Nos cursos de extensão universitária e no mestrado em Computação Aplicada, tem sido usado o ambiente baseado no GNU C/C++ com sucesso. Os alunos tem apreciado bastante a possibilidade de trabalhar em uma mesma linguagem programando em dois sistemas operacionais distintos (Windows/DOS com o DJGPP/Rhíde e Linux com o Gcc/Gpp - onde o uso do Rhíde é opcional devido a existência de outras ferramentas de edição e de depuração). No ambiente de pesquisa o GNU C/C++ e o JAVA tem sido largamente utilizado, pois permitem o desenvolvimento de softwares que poderão ser utilizados em diferentes plataformas, facilitando ainda mais o intercâmbio e difusão das pesquisas realizadas.

As experiências na Unisinos com o GNU C/C++ e o DJGPP, demonstraram a viabilidade da adoção de Softwares Livres para o ensino de programação. Estamos plenamente satisfeitos com a qualidade das ferramentas de Software Livre utilizadas no que se refere a documentação, atualização, facilidade de uso, entre outros aspectos. De acordo com a tendência atual, esperamos muito em breve ampliar o uso deste tipo de ferramentas em nossa instituição.

Referências:

- [1] Borland Community. Borland C++ Compiler 5.5. Web: <http://community.borland.com>
- [2] Borland Community. JBuilder 3 Foundation. Web: <http://community.borland.com>
- [3] DJGPP - Delorie Software, Gnu GPP. Web: <http://www.delorie.com>
- [4] Free Pascal. On-Line Help Manuals: Users, Programmers and Reference Guide. Web: <http://www.freepascal.org>
- [5] GNU Project - Gnu is Not Unix. Web: <http://www.gnu.org>
- [6] Glade GTK+ User Interface Builder. Web: <http://glade.pn.org>
- [7] JAVASOFT - Sun Corporation. Linguagem JAVA - Java 1.2 SDK e Forte Java IDE. Web: <http://www.javasoft.com>
- [8] RHIDE - Robert Hölme IDE. Web: <http://www.tu-chemnitz.de/~sho/rho/rhide.html>
- [9] Allegro - A game programming library. Web: <http://www.talula.demon.co.uk/allegro/>