

**Uma Análise de Viabilidade de Migração para *Software Livre*
Aplicado em Ambientes Integrados de Desenvolvimento das
Linguagens C e Pascal**

Gustavo da Silva Quirino

LAVRAS
MINAS GERAIS – BRASIL
2010

GUSTAVO DA SILVA QUIRINO

**Uma Análise de Viabilidade de Migração para *Software Livre*
Aplicado em Ambientes Integrados de Desenvolvimento das
Linguagens C e Pascal**

Monografia de Pós-Graduação “Lato
Sensu” apresentada ao Departamento de
Ciência da Computação para obtenção do
título de Especialista em “Administração
em Redes Linux”

Orientadora:

Profa. Msc. Kátia Cilene Amaral Uchôa

LAVRAS
MINAS GERAIS – BRASIL
2010

GUSTAVO DA SILVA QUIRINO

Uma análise de viabilidade de migração para *Software Livre* aplicado em Ambientes Integrados de Desenvolvimento das linguagens C e Pascal

Monografia de Pós-Graduação “Lato Sensu” apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em “Administração em Redes Linux”

Aprovada em 24 de Abril de 2010

Prof. DSc. José Monserrat Neto

Prof. Especialista Ramon Simões Abílio

Profª. Msc. Kátia Cilene Uchôa
(Orientadora)

LAVRAS
MINAS GERAIS – BRASIL
2010

*Dedico esta monografia à minha
família, que se orgulha de cada
título conseguido por mim.*

Agradecimentos

Agradeço a Profa. Msc. Kátia Cilene Amaral Uchôa pela confiança e paciência destinada na construção deste trabalho. Aos alunos e professores que contribuíram com a pesquisa.

Sumário

1	Introdução	1
2	IFBA campus Barreiras	6
	2.1 Curso Técnico de Informática IFBA e Grade Curricular.....	7
	2.2 Ferramentas de Programação em Cursos Técnicos no Brasil	10
3	<i>Software Livre</i>	12
	3.1 <i>Software Livre</i> Versus <i>Software</i> Proprietário	16
	3.2 Presença do <i>Software Livre</i> nas Instituições de Ensino	19
	3.3 Linguagem de Programação	23
	3.4 Ambientes Integrados de Desenvolvimento	27
	3.4.1. Linguagem C (Turbo C)	28
	3.4.2. Linguagem Pascal (Turbo Pascal)	31
	3.4.3. Geany	32
	3.4.4. Comparativo das IDE's	34
4	Metodologia	35
	4.1 Público Alvo Envolvido	35
	4.2 Treinamento	36
	4.2.1. Minicurso de <i>Software Livre</i>	37
	4.2.2. Minicurso de Geany (Linguagem C).....	38
	4.2.3. Minicurso de Geany (Linguagem Pascal)	38
	4.3 Programas Desenvolvidos	39
5	Avaliação do Projeto Piloto	40
	5.1 Análise e Discussão	40

6 Conclusão	51
7 Referências Bibliográficas	54
Apêndice A – Questionário-Alunos.....	57

LISTA DE FIGURAS

Figura 3.1 – Popularidade das linguagens de programação.....	26
Figura 3.2 – Modelo de compilação da Linguagem C.....	28
Figura 3.3 – IDE Turbo C++	30
Figura 3.4 – IDE Turbo Pascal	32
Figura 3.5 – IDE Geany	33
Figura 5.1 – Compreensão do significado de liberdade de <i>software</i>	41
Figura 5.2 – Local de utilização do <i>software</i> livre.....	41
Figura 5.3 – Opinião sobre o uso de <i>software</i> livre nos Institutos Federais de Educação.....	42
Figura 5.4 – Importância do <i>software</i> livre na formação profissional ...	43
Figura 5.5 – Uso de <i>software</i> livre nos laboratórios de ensino	44
Figura 5.6 – Tipo de <i>software</i> incentivado pelos professores	44
Figura 5.7 – Uso de <i>software</i> pirata.	45
Figura 5.8 – Intenção dos alunos em aprender mais <i>Softwares</i> Livres.....	46
Figura 5.9 – Dificuldades de adoção do <i>software</i> livre nas escolas	46
Figura 5.10 – Local onde o aluno conheceu o <i>software</i> livre	47
Figura 5.11 – Dificuldades em desenvolver aplicações no Geany	47
Figura 5.12 – Opinião dos alunos sobre adoção do Geany.....	48
Figura 5.13 – Fatores que influenciaram na escolha do Geany	49
Figura 5.14 – Como deve ser o uso de <i>software</i> livre no curso de informática.....	50

LISTA DE TABELAS

Tabela 2.1 – Estrutura dos laboratórios de Informática	7
Tabela 3.1 – <i>Ranking</i> das linguagens mais populares.....	25
Tabela 3.2 – Quadro Comparativo das IDE's	34
Tabela 4.1 – Tópicos do minicurso de <i>software</i> livre	37
Tabela 4.2 – Tópicos do minicurso de <i>Geany (Linguagem C)</i>	38
Tabela 4.3 – Tópicos do minicurso de <i>Geany (Linguagem Pascal)</i>	38

RESUMO

Este trabalho apresenta um estudo da viabilidade de migração para ferramentas livres aplicado em ambientes integrados de desenvolvimento das linguagens C e Pascal. O curso de Informática do Instituto Federal de Educação Ciência e Tecnologia da Bahia, campus Barreiras sempre utilizou IDE's proprietárias no ensino da programação. Este estudo irá verificar tomando como foco os alunos, a possibilidade de utilização de ferramentas livres nas disciplinas técnicas de programação. A IDE Turbo C++ e Turbo Pascal eram utilizadas no desenvolvimento de programas na linguagem C e Pascal respectivamente. A ferramenta livre Geany foi escolhida para fazer parte do estudo de migração por apresentar atualizações constantes, possibilidade de programar em diversas linguagens, interface de fácil manuseio, além de oferecer todas as liberdades de um *software* livre. Nesta monografia estão descritas as fases do estudo de migração, desde o planejamento, treinamento aos alunos, aplicação de questionários, e análise dos dados que visam apontar a viabilidade da migração.

Palavras-Chave: *Software* Livre, Migração, Ambientes Integrados de Desenvolvimento.

Introdução

A rede de Institutos Federais de Educação, Ciência e Tecnologia (IF) foi criada em 2008, herdando toda a estrutura dos antigos CEFET's¹, essa rede conta hoje com 38 Institutos espalhados por todas as regiões do país. O Instituto Federal da Bahia campus Barreiras têm 15 anos de existência e atualmente oferece cursos técnicos nas área de Edificações, Enfermagem, Eletrotécnica, Eletromecânica, Alimentos e Informática.

O curso Técnico de Informática têm uma grade diversificada, incluindo disciplinas como: Banco de Dados, Organização de Computadores, Redes de Computadores, Algoritmo, Projeto e Análise de Sistemas, além de Linguagem de Programação. O desenvolvimento de sistemas é prática obrigatória na maioria dos cursos Técnicos de Informática no Brasil. A construção de aplicativos pode ser feita utilizando diversas linguagens de programação, que são adotadas nas instituições de acordo com o foco de cada curso.

A adoção de uma linguagem de programação em muitos casos requer uma IDE (*Integrated Development Environment*), este tipo de aplicativo é conhecido no Brasil como Ambiente Integrado de Desenvolvimento, que é uma ferramenta que auxilia o programador a desenvolver *software* utilizando uma linguagem de programação. Essas ferramentas são geralmente constituídas por um editor² de código fonte da linguagem, um compilador que transforma a linguagem de programação em linguagem de máquina, e um *linker* que cria um executável, possibilitando que o programa gerado possa ser executado em outras máquinas.

1 CEFET – Centro Federal de Educação Tecnológica, escola de educação profissional extinta pela criação do IFBA.

2 Editor – Aplicativo para edição de texto. Exemplos de editores: Gedit e Emacs.

É possível encontrar no mercado IDE's proprietárias e livres. Entretanto, com as ferramentas de programação proprietárias o usuário não têm acesso ao código fonte da ferramenta, têm que pagar pela licença da IDE, e em alguns casos têm que pagar uma quantia por programa desenvolvido. Nas ferramentas livres de desenvolvimento o usuário têm acesso ao código fonte da ferramenta, têm um custo menor ou até mesmo nulo na aquisição do programa, e ainda pode distribuir e modificar tanto a IDE como os produtos criados como bem entender, desde que siga as condições de licença em uso.

Um outro aspecto importante é que as ferramentas proprietárias de desenvolvimento são produzidas por empresas que podem falir ou deixar de existir no mercado. Assim, os dados críticos de usuários armazenados nos padrões proprietários que não podem ser manipulados por outros aplicativos senão aqueles que os produziram podem tornar-se inacessíveis. Isso não ocorre em IDE's de código aberto, pois é fácil encontrar uma solução alternativa, até mesmo quando empresas, ou programadores que fabricaram a ferramenta deixam de existir. Em último caso, o usuário de IDE livre extinta pode estudar o código fonte do programa e transferir os dados do seu sistema para outro ambiente, ou mesmo fazer atualizações que impeçam que o programa fique obsoleto.

Como exemplo de IDE's proprietárias que deixaram de receber atualizações é possível citar o Turbo Pascal, uma IDE criada pela Borland que foi descontinuada em 1995. Com a descontinuação do Turbo Pascal milhares de programadores tiveram que migrar seus códigos para outros ambientes de desenvolvimento ou até mesmo para outras linguagens. Conforme foi assinalado por [SILVEIRA, 2009] *“em meados da década de 90, quando a Borland descontinuou os compiladores Turbo Pascal e Borland Pascal, deixando para trás uma grande comunidade de programadores..”* .

A decisão de escolher entre uma ferramenta de desenvolvimento livre ou proprietária é complexa e depende do foco, recursos financeiros, e finalidades de cada programador ou empresa. Por outro lado, vale ressaltar que muitas vezes a adoção de uma IDE está ligado a fator cultural, a resistência ao que é apresentado como novo é perceptível, mesmo que este novo traga vantagens como isenção de licença da ferramenta, código fonte aberto, e privilegia a inteligência de seus usuários.

Além de todas as vantagens mencionadas o uso de *software* livre garante liberdade de análise, distribuição, modificação do código fonte podendo diminuir as despesas do instituto com programas e até mesmo retirá-lo da ilegalidade, como é o caso dos órgãos que utilizam programas “pirata”, ou seja, sem licença. O uso de *software* livre também desenvolve o caráter social do usuário uma vez que incentiva a distribuição e troca de conhecimento. A adoção de *software* livre nas disciplinas de programação neste curso trará mais uma vantagem aos alunos e professores, pois esses não terão que trabalhar com duas IDE's proprietárias distintas, Turbo C para desenvolvimento de programas em C, e Turbo Pascal para desenvolvimento de programas na linguagem Pascal. Utilizando Geany³ o usuário pode desenvolver programas tanto na linguagem C quanto na linguagem Pascal.

Nesse sentido, esta pesquisa visa apontar a viabilidade de transição do uso de ferramentas de desenvolvimento proprietárias para ferramentas de desenvolvimento livres tendo como foco alunos do curso Técnico Integrado em Informática do IFBA, e servir de base para futuras migrações que possam ocorrer no instituto.

No IFBA campus Barreiras as ferramentas proprietárias Turbo C++ 3.0⁴

3 Geany: Mais informações sobre a ferramenta em: <http://www.geany.org/> .

4 Turbo C++ - IDE que possibilita desenvolvimento de programas nas linguagens C e C++ criado na década de 90 pela Borland.

e Turbo Pascal 7.0⁵, IDE's das linguagens C e Pascal respectivamente, que são utilizadas no ensino da programação nas séries iniciais do curso Técnico de Informática, por esse motivo fazem parte deste estudo. A ferramenta Geany é uma IDE livre que suporta diversas linguagens de programação, pode ser executada em vários sistemas operacionais, e recebe atualizações freqüentes. Esses fatores foram levados em consideração para que a escolha do *software* livre Geany para estudo do presente trabalho.

Os alunos do curso de informática responderam questionários com perguntas abertas e fechadas a respeito de conhecimento sobre *software* livre, Ambientes Integrados de Desenvolvimento, e impactos de migração de programas. Os dados foram coletados por abordagem probabilística. Antes de responderem ao questionário os alunos receberam um minicurso em 3 etapas. A primeira a respeito do conceito e características do *software* livre, e suas vantagens frente ao proprietário. Na segunda etapa falou-se a respeito da linguagem C com o desenvolvimento de uma aplicação em Turbo C++ e em Geany. Na terceira foram abordados aspectos da linguagem Pascal com o desenvolvimento de uma aplicação em Turbo Pascal e Geany. Cada uma das etapas teve duração de 3 horas, totalizando 9 horas de minicurso. Ao final os alunos foram instigados a responder um questionário a respeito do conhecimento sobre *software* livre, facilidades e dificuldades encontradas no uso da nova ferramenta, e opinião a respeito da migração de Turbo C++ e Turbo Pascal para Geany nas disciplinas de programação. A análise e interpretação dos dados foi feita utilizando métodos estatísticos e construção de gráficos.

A monografia encontra-se organizada como se segue: no segundo capítulo será abordado a respeito do IFBA campus Barreiras, o curso Técnico de

5 Turbo Pascal – Ambiente Integrado de Desenvolvimento que possibilita desenvolvimento de programas na linguagem Pascal criado na década de 90 pela Borland.

Informática na modalidade integrada, e as ferramentas de programação utilizadas. O terceiro aborda o histórico e conceitos do *software* livre e dos Ambientes Integrados de Desenvolvimento envolvidos no estudo. No quarto é explanado como foi feita a pesquisa, e quais processos foram tomados para a obtenção dos resultados. No quinto o leitor pode encontrar informações sobre como os dados foram analisados. O sexto contém a conclusão e o resultado da pesquisa.

2. IFBA campus Barreiras

A antiga rede de Centros Federais de Ensino Técnico vêm sendo transformada gradativamente em Institutos Federais de Educação (IF). Atualmente os Institutos Federais estão presentes nos 26 estados da Federação e no Distrito Federal, os institutos são conhecidos pela sigla IF seguido da sigla do estado a qual estão instalados. A sigla do Instituto Federal de Educação, Ciência e Tecnologia da Bahia é IFBA. O IFBA está subdividido em 8 *campi*, sendo que este estudo realizar-se-a no campus Barreiras, oeste da Bahia. Dentre as modalidades de ensino oferecidas pelo institutos destacam-se os níveis Médio integrado com Técnico, Técnico Subsequente, e Superior.

O campus Barreiras conta com 919 alunos matriculados, 60 professores, 49 técnicos e oferece cursos em todas as modalidades citadas. A Instituição têm 15 anos de história e é referência no oeste da Bahia como escola pública, gratuita e de qualidade. O processo seletivo acontece anualmente, sendo que alguns cursos recebem novos alunos semestralmente e outros recebem alunos anualmente, incluindo o curso Técnico de Informática.

O corpo docente efetivo é formado por 5 graduados, 30 especialistas, e 5 mestres. O corpo Técnico efetivo é formado por 10 servidores com Ensino Fundamental, 20 servidores com Ensino Médio, e 19 servidores com Ensino Superior.

No IFBA campus Barreiros existem laboratórios de elétrica, mecânica, edificações, enfermagem, desenho e informática. Os laboratórios de informática são numerados em 26, 27, e 28. A Tabela 2.1 mostra a estrutura dos laboratórios de informática da Instituição.

Tabela 2.1 – Estrutura dos Laboratórios de Informática

Laboratório	Máquinas	Processador	Memória RAM	HD
26	12	Celeron 1,6Ghz	1GB	80GB
27	30	Celeron D 2,6Ghz	512MB	80GB
28	25	Core 2 Duo 1.8Ghz	1GB	80GB

O laboratório 26 serve de estudos para as disciplinas de Sistemas Operacionais e Organização de Computadores. O laboratório 27 serve de estudos para as disciplinas de Introdução a Computação, Redes, e Estrutura de Dados. O laboratório 28 serve de estudos para as disciplinas de Programação, Banco de Dados e Projeto de Sistemas, além de Sistemas Operacionais II.

2.1 Curso Técnico de Informática IFBA e Grade Curricular

O curso Técnico Integrado em Informática têm 4 anos de existência, e conta com 186 alunos. Conforme consta no projeto do curso Técnico em Informática na modalidade Integrada,

“Aos alunos que concluírem todas as séries do curso, com assiduidade e aproveitamento mínimos previstos na legislação vigente, serão conferidos diplomas de técnico de nível médio em informática na forma integrada, nos termos do Artigo 7º do Decreto nº 5.154/2004” [BAQUEIRO, 2006].

O curso Técnico Integrado em Informática conta atualmente com 16 professores das disciplinas do núcleo comum, e 3 professores das disciplinas técnicas. O principal objetivo do curso é formar cidadãos com habilidades técnicas na área de informática para atuarem no ramo da indústria, comércio e agricultura, como é assinalado no projeto do curso.

“O Curso Técnico de Nível Médio em Informática na forma Integrada oferecido pelo CEFET-BA/UE-Barreiras tem como objetivo formar cidadão com preparação humanísticas, sociológicas, filosóficas, artísticas, científicas e tecnológicas e com competência para a habilitação profissional técnica em Informática”. [BAQUEIRO, 2006]

A forma de ingresso é feita por meio de um exame de seleção emanado pelo Conselho de Ensino, Pesquisa e Extensão, do Regimento Interno e da legislação vigente. Entre as competências esperadas de um egresso estão a capacidade de manipular os dispositivos quem formam um computador, criar, instalar e manipular *softwares*, identificar e manipular tipos de redes, administrar sistemas de banco de dados, além de executar ações de treinamento em sistemas computacionais.

A grade curricular do curso Técnico em Informática é formada por disciplinas do núcleo comum, isto é, disciplinas básicas do ensino médio como Português, Matemática, Química, Biologia, Artes, entre outras. As disciplinas técnicas são específicas do curso, dentre as principais pode-se destacar Eletrônica Digital, Arquitetura de Computadores, Linguagem de Programação, Estrutura de Dados, Sistemas Operacionais, Banco de Dados, e Redes de Computadores.

“A Organização Curricular observará as Diretrizes Curriculares Nacionais definidas pelo Conselho Nacional de Educação para o Ensino Médio e para a Educação Profissional Técnica de

Nível Médio através da Resolução CNE/CEB 1/05, pelo Parecer 15/98 e pelas Resoluções CNE/CEB 3/98 e 04/99, observadas as alterações emanadas em conformidade com o Decreto 5.154/04 pela Resolução 01/05.”[BAQUEIRO, 2006]

No curso Técnico de Informática do IFBA campus Barreiras diversas disciplinas utilizam a linguagens de programação. Porém, como o objetivo deste estudo é avaliar a viabilidade de migração das ferramentas de programação das linguagens C e Pascal, esta seção abordará apenas as disciplinas que estão inseridas neste processo.

A disciplina de Algoritmo é ministrada no 2º ano do curso, momento em que o aluno já está habituado ao Instituto. O objetivo desta disciplina é fazer com que o aluno aprenda lógica de programação, e construa pequenos programas. No início do curso é abordado o conceito de algoritmos utilizando uma linguagem de fácil entendimento, desde a entrada da primeira turma o Portugol⁶ é a linguagem ensinada. O aplicativo Visalg⁷ é utilizado para compilar os códigos fontes desenvolvidos nesta linguagem. No segundo semestre letivo o aluno começa a aprender os conceitos e fundamentos da linguagem Pascal, desenvolvendo aplicativos mais complexos. A IDE Turbo Pascal é utilizada na elaboração de programas na disciplina.

A disciplina Estrutura de Dados ministrada no 3º ano do curso objetiva fazer com que o aluno aprenda a manipular dados de forma organizada, que facilita a inserção, remoção e atualização de informações. Nas duas primeiras

6 Portugol: é uma pseudo-linguagem de programação, criada para demonstrar o uso de algoritmos e programação estruturada.

7 Visalg: é um programa que edita, interpreta e executa algoritmos com uma linguagem próxima do Português Estruturado, como um programa normal de computador. Recurso valiosos para quem está iniciando no aprendizado de algoritmos, não só para praticar a sua criação mas também para melhor entender a sua execução através do visualizador de variáveis que funciona como um depurador.

unidades o aluno aprende os conceitos e fundamentos da linguagem C. Nas duas últimas unidades as aplicações envolvendo estruturas de dados são mais freqüentes, e o aluno desenvolve programas com pilhas, filas, listas, entre outras estruturas. A IDE Turbo C++ versão 3.0 é utilizada no desenvolvimento de programas na disciplina em questão.

Todas as disciplinas seguem os mesmos critérios para aprovação dos alunos. O ano letivo é dividido em 4 unidades, em cada uma delas o aluno faz avaliações que totalizam dez pontos no final. Está aprovado o aluno que conseguir vinte e quatro pontos no somatório das quatro unidades.

2.2 Ferramentas de programação utilizadas em cursos Técnicos no Brasil

A informação a respeito das IDE's utilizadas nos cursos Técnicos do Brasil foi adquirida através contatos com professores do IFMT, IFTO, e IFSC. Esses professores mencionaram como programas utilizados o DevCPP⁸, GCC⁹, NetBeans¹⁰.

Bloodshed DevCPP é uma IDE livre e gratuita escrito em Delphi¹¹, que pode ser executado apenas no sistema operacional Windows, e é utilizado no

8 DevCPP – Mais informações em: <http://www.bloodshed.net/devcpp.html>

9 GCC – Mais informações em: <http://gcc.gnu.org/>

10 NetBeans – Mais informações em: <http://netbeans.org/>

11 Delphi – É um compilador e uma IDE, atualmente produzido pela Embarcadero. Delphi é direcionado para o Sistema Operacional Windows. Mais informações em: <http://www.embarcadero.com/products/delphi>

desenvolvimento de aplicações nas linguagens C e C++.

O GNU Compiler Collection (gcc) é um compilador que oferece possibilidade de desenvolvimento de aplicações em C, C++, Objective-C, Fortran, Java, e Ada. É *software* livre e gratuito criado originalmente para ser o compilador do Sistema Operacional GNU.

NetBeans é uma IDE livre e gratuita, que pode ser executada em Linux, Windows, MacOS, e Solaris, em que o usuário pode desenvolver aplicações nas linguagens Java, PHP, JavaScript, Ajax, Ruby, Ruby on Rails, Groovy and Grails, C/C++.

A ferramenta DevCPP apesar de ser *software* livre foi escrito em um *software* proprietário, e só pode ser executado em um sistema operacional proprietário, o que dificulta sua adoção em um ambiente que preza pela liberdade de *software*. GCC é o compilador livre mais difundido entre os programadores. O próprio Geany necessita do GCC para ser instalado no Linux. GCC também é utilizado no Geany para compilação de programas na linguagem C. NetBeans é uma IDE robusta, multiplataforma, que oferece ao usuário a possibilidade de desenvolver aplicações em diversas linguagens, porém requer *hardware* poderoso para ser executada. Para se ter uma idéia, a última versão do NetBeans pode ser executada no Ubuntu 9.04, desde que o *hardware* tenha as seguintes configurações mínimas: Processador de 2.6 GHz Intel Pentium IV ou equivalente e Memória RAM de 2 GB. Esse pré-requisito inviabiliza a sua utilização em escolas que não possam investir freqüentemente em *hardware*.

3. *Software Livre*

Antes do surgimento das linguagens de programação de alto nível, os fornecedores de computadores davam pouca importância para o *software*, uma vez que a maior parte de sua receita era advinda da venda do *hardware*. A utilização de um programa só era possível se o usuário adquirisse uma máquina igual a que o programador tinha utilizado para desenvolver o programa. A troca de informações e códigos entre programadores era prática incentivada pelas empresas, pois isto impulsionava a venda de mais computadores.[UCHÔA, 2008] afirma que:

“o relacionamento entre os fabricantes de hardware e periféricos com os usuários de computadores somente mudaram quando os programas começaram a serem comercializados de forma independente do hardware.”

A partir desse momento o *software* ganhou valor, e as empresas não eram beneficiadas com a troca de programas entre os usuários, os programas passaram a ser vendidos com restrições que impediam que o usuário pudesse reproduzi-lo ou distribuí-lo para outras pessoas, além de ser proibido a divulgação de problemas e o estudo de seu funcionamento. As restrições impostas pelos fabricantes começavam a conflitar com os interesses dos usuários surge aí o movimento do *software* livre. Segundo [SILVEIRA, 2003]:

“O movimento do software livre começou pequeno. Reunia e distribuía programas e ferramentas livres, com o código-fonte aberto. Assim, todas as pessoas poderiam ter acesso não só aos programas, mas também aos códigos em que foram escritos”.

Alguns estudiosos acreditam que o movimento do *software* livre não ganharia tamanha proporção se não tivesse apoio de empresas, universidades e governos. De acordo com [REIS, 2004]:

“o movimento do software livre é um movimento social bastante complexo, envolvendo não apenas alguns poucos lunáticos, mas também empresas, universidade, e governos. O que une esses agentes com interesses tão diversos é fornecer à sociedade sistemas computacionais mais estáveis, baratos e que finalmente sirvam aos interesses das pessoas e das empresas consumidoras e não exclusivamente aos interesses dos monopólios.”

Um dos maiores incentivadores do movimento de *software* livre foi Richard Stallman, que criou uma comunidade especializada no desenvolvimento de programas de código aberto, pois segundo [SILVEIRA, 2003]:

“Foi com base na indignação ativa de um então integrante do MIT, Richard Stallman, contra a proibição de se acessar o código-fonte de um software, certamente desenvolvido a partir do conhecimento acumulado de tantos outros programadores, que em 1985 foi criada a Free Software Foundation.”

Essa comunidade tinha o objetivo de criar novos *softwares* livres e manter o sistema operacional GNU proposto por Richard, pois segundo [SILVEIRA,2003]:

“A idéia era produzir um sistema operacional livre que tivesse a lógica do sistema Unix, que era proprietário, ou seja, pertencia a uma empresa. Por isso, os vários

esforços de programação eram reunidos em torno do nome GNU – GNU's Not Unix.”

Uma característica importante do projeto GNU é que ele valoriza o conhecimento criado anteriormente, uma vez que o programador não precisa iniciar seu programa do zero, pois alguma função ou trecho de código pode ser reaproveitado de outro sistema. Richard Stallman escreveu em um de seus artigos que:

“O projeto GNU não é somente desenvolvimento e distribuição de alguns softwares livres úteis. O coração do projeto GNU é uma idéia: que software deve ser livre, e que a liberdade do usuário vale a pena ser defendida. Se as pessoas têm liberdade, mas não a apreciam conscientemente, não irão mantê-la por muito tempo. Se quisermos que a liberdade dure, precisamos chamar a atenção das pessoas para a liberdade que elas têm em programas livres.”[SILVA, 2005]

O objetivo do projeto de GNU de criar um Sistema Operacional livre foi realizado em 1992 com a ajuda de Linus Torvalds, pois segundo [SILVEIRA, 2003]:

“Com a difusão da internet, o movimento de software livre ganhou o mundo logrou produzir um sistema operacional livre, completo e multifuncional, o GNU/Linux. Em 1992, o finlandês Linus Torvalds conseguiu compilar todos os programas e ferramentas do movimento GNU em um kernel, um núcleo central, o que viabilizou o sistema operacional.”

O projeto GNU não está limitado ao desenvolvimento de um sistema operacional, existe uma gama de programas criados pela comunidade, entre eles

pode-se destacar o compilador GCC (GNU Compiler Collection), o editor de imagens GIMP, e o ambiente Gráfico GNOME.

Todo programa de computador é formado por instruções que determinam qual a sua funcionalidade. Um conjunto de instruções que formam um *software* é conhecido como código fonte¹². O processo de desenvolvimento de um programa consiste na transformação do código fonte, ou seja, instruções em linguagem de programação que são convertidas em linguagem de máquina. O programador pode restringir ou liberar o acesso ao código fonte na venda ou distribuição do seu produto, o acesso irrestrito ao código fonte é a principal característica de um *software* livre, essa informação é ratificada por [SILVEIRA, 2003] quando define este tipo de *software* como sendo “*um programa de computador com o código-fonte aberto, possibilitando que qualquer técnico possa estudá-lo, alterá-lo, adequá-lo às suas próprias necessidades e redistribuí-lo, sem restrições*”. Segundo a FSF, um *software* é tido como livre quando oferece ao usuários a liberdade de estudar, copiar, executar, e distribuir.

Estes termos têm o intuito de garantir a liberdade do usuário ao utilizar programas livres, porém Richard Stallman percebeu que muitos programadores e empresas desenvolvedoras de programas estavam se apropriando de *softwares* livres, fazendo modificações e fechando o código fonte, impedindo que o mesmo pudesse ser estudado e modificado por outros usuários. Segundo [SILVEIRA, 2003]:

“Para evitar que os esforços do movimento fossem apropriados indevidamente e patenteados por algum empreendedores oportunista, novamente bloqueando o desenvolvimento compartilhado, a Free Software Foundation inventou a Licença

¹² Código Fonte - é um conjunto de palavras escritas em uma determinada linguagem de forma ordenada e lógica. Esse conjunto de palavras é transformado em linguagem de máquina através de um compilador ou interpretador.

Pública Geral, GPL em inglês, conhecida como copyleft em contraposição ao copyright.”

O conceito de *copyleft* surgiu então para impedir que oportunistas adquirissem um *software* livre e tentassem transformá-lo em um programa proprietário.

3.1 Software Livre Versus Software Proprietário

Software livre e *software* proprietário têm características distintas em relação ao acesso ao código fonte, disseminação do conhecimento, custo do produto, entre outras.

[SILVEIRA, 2003] afirma que

“O software livre nasce como oposição ao software proprietário, programas de computador com código-fonte fechado, patenteado por uma única empresa, que cobra direito de propriedade intelectual.”

Nesse modelo de desenvolvimento proprietário o usuário acaba aprisionado e refém dos principais fabricantes.

Quando um usuário adquire a licença de uso de um programa proprietário, ele não tem acesso ao código fonte, e não sabe quais as funções estão contidas nele. Existe uma série de regras e normas que regem as licenças de *software* proprietário, não sendo permitida a análise, modificação ou distribuição do código fonte. Outra característica deste tipo de programa é que o usuário paga por cópia instalada, ou seja, quanto maior o número de máquinas

em que o usuário deseja ter o programa, mais licenças devem ser compradas. [BACIC, 2003] diz que:

“o software proprietário é regido por uma série de normas que visam limitar o seu uso ao número de licenças adquiridas. São estabelecidas licenças pelas quais é necessário pagar por cópia instalada, não sendo permitida a alteração do código; ademais, é ilícita a livre distribuição do programa e normalmente não se tem acesso ao código fonte destes programas.”

Como a grande maioria dos programas proprietários têm seu código fonte fechado, fica impossível para o usuário realizar melhorias, corrigir erros ou mesmo adicionar alguma função específica de sua necessidade.

O *software* livre pode ser gratuito ou não, a filosofia livre vai além da cobrança pelo desenvolvimento ou distribuição de um programa. A característica mais importante de um *software* livre é que o código fonte dele estará disponível para seus usuários, disseminando assim o conhecimento, e quebrando paradigmas criados pelo *software* Proprietário. Segundo [SILVEIRA, 2003]: “Free software é uma questão de liberdade e não somente de preço! Software livre não significa ser contrário à comercialização, mas sim, assumir uma postura rigorosa contra a falta de liberdade.”

O *software* livre permite que o usuário altere ou contrate um programador para alterar e adicionar funcionalidades ao seu programa, evitando que o usuário tenha que adquirir uma nova versão do programa que supra suas necessidades. O *software* livre valoriza a troca de conhecimento uma vez que os programadores podem aproveitar trechos de códigos existentes em outros programas. Após adquirir um *software* livre o usuário pode distribuí-lo e instalá-lo em quantas máquinas quiser, pois ele têm posse do programa e não apenas têm a licença para usá-lo. Neste contexto [SILVEIRA, 2003] diz que:

“O movimento de software livre é a maior expressão da imaginação dissidente de uma sociedade que busca mais do que a sua mercantilização. Trata-se de um movimento baseado no princípio do compartilhamento do conhecimento e na solidariedade praticada pela inteligência coletiva conectada na rede mundial de computadores. “

A disseminação do *software* livre só foi possível devido a internet, pois é através desta rede que programadores de todo o mundo trocam informações e códigos de programas. Quando um defeito ou brecha de segurança é descoberto, pessoas de várias partes do planeta têm acesso ao código e podem ajudar uns aos outros a resolver problemas, contribuindo para o aperfeiçoamento de seus programas. O *software* livre contribui para o avanço da sociedade, uma vez que não restringe o conhecimento ao criador do programa, mas incentiva que o saber seja coletivo, criado e administrado por todos. De acordo com [SILVEIRA, 2003]:

“O movimento do software livre é expressão autêntica desse potencial da rede e o grande modelo para a consolidação de soluções compartilhadas diante de questões complexas, a partir da interação multiétnica, multinacional e multicultural. É a afirmação da possibilidade de a internet consolidar-se também como uma esfera global planetária, evitando a condição hegemônica de supermercado global.”

3.2 Presença do *software* livre nas Instituições de Ensino

Através de incentivo político e aumento no número de adeptos aos *software* livre, é cada vez maior o número de instituições de ensino que utilizam programas de código aberto.

Fatores como, liberdade de acesso ao código fonte, aumento no número de aplicativos disponíveis, e diminuição de custos com licenças influenciaram esse fenômeno. Apesar desta crescente as ferramentas proprietárias de programação ainda têm um maior número de usuários em relação as ferramentas livres, neste sentido Chrispiniano assinala em [CHRISPINIANO, 2005].

“É verdade que tanto o número de soluções como a qualidade dos programas têm evoluído rapidamente, conforme crescem o número de programadores trabalhando com programas de código aberto e o de instituições que os utilizam. Mesmo assim, as versões livres de programas com número reduzido de usuários ainda não conseguem competir com as proprietárias.”

Nos dias atuais ainda é possível encontrar ferramentas livres que não conseguem atrair programadores, seja por falta de *plugins* e extensões fechadas ou até mesmo pela crença de que IDE's livres não oferecem tantos recursos como as proprietárias. Ainda é preciso mencionar que fatores históricos e culturais influenciam no motivo pelo qual as IDE'S livres tem sido pouco disseminadas em ambientes educacionais. Segundo [SILVEIRA, 2003]:

“A Microsoft consolidou-se no Brasil apoiada em compras governamentais e na disseminação de sua cultura com o mercado

pirata. Depois que essa cultura se consolidou, começaram as campanhas antipirataria patrocinadas, em geral, por entidades fomentadas pela MS, direcionadas a empresas médias e grandes. Até mesmo nas universidades houve campanha em que o MEC entrava de forma sub-reptícia para regularizar a pirataria existente, numa parceria escondida com o setor privado.”

Nesse contexto é possível observar sem muito esforço que megaempresas produtoras de programas proprietário estão inseridas nas universidades, oferecendo licenças gratuitas de seus produtos, bolsas de estudos, patrocinando eventos, doando equipamentos. Todas essas vantagens oferecidas as instituições educacionais acabam servindo para a permanência de seus produtos, e consequentemente propiciando a alienação desses neste espaço educacional. Por sua vez, esta prática gera lucro a essas megaempresas que acabam alienando cada vez mais usuários para seus produtos. Por fim, um profissional formado nessas instituições quando entra no mercado de trabalho irá convencer a empresa a pagar pela ferramenta de desenvolvimento utilizada por ele na universidade. Entretanto, essa cadeia passou a perder força com o aumento da difusão do *software* livre no mercado. Conforme afirma [LOZANO, 2004]:

“O software livre acaba com a venda de "licenças de uso" e não com a venda de software ou de serviços agregado desta forma o software livre se encaixa a favor da tendência do mercado, e não contra ela .”

É possível ganhar dinheiro com *software* livre, seja na venda do programa, manutenção do sistema, ministrando cursos, entre outras alternativas.

Na internet é cada vez mais comum serviços gratuitos que lucram apenas com propagandas e anúncios, desta forma o modelo de cobrança de licença se encontra defasado e provavelmente não irá durar muito tempo.

Geralmente existem mais de um tipo de versão para um determinado *software* proprietário, entre esses tipos é possível citar a versão acadêmica, direcionada para instituições de ensino. Esta versão dispensa o pagamento de licença, sendo liberada somente a fins de estudo. Neste sentido, muitas das instituições de ensino não pagam licença dos programas, isto acaba perpetuando a presença desses produtos no mercado e alienando conhecimento desses usuários a um único *software*.

Instituições de ensino públicas e privadas não conseguem acompanhar a evolução dos programas proprietários, pois cada nova versão necessita de *hardware* de última geração. Isso gera um custo, que muitas vezes inviabiliza a compra da licença, ou até mesmo obriga a instituição a deixar de investir em setores estratégicos. O *software* livre têm preocupação com a eficiência, e não têm ligação com fabricantes de *hardware*. “*Para o setor educacional, muitas vezes carente de recursos, o software livre é uma alternativa viável e que deve ser considerada seriamente.*” [ALMEIDA Q.,2000]

As instituições de ensino têm um papel fundamental na sociedade que é a geração e difusão do conhecimento, e as liberdades que permeiam o *software* livre estão intimamente ligadas a essas duas características. Por este motivo a maioria das Instituições públicas e privadas do Brasil, estão migrando suas aplicações para *software* livre.

“O emprego de software livre na educação é uma alternativa imprescindível a qualquer projeto educacional, tanto no setor público como privado. Fatores tais como liberdade, custo, flexibilidade são estratégicos para a condução bem sucedida de

projetos educacionais mediados por computador.
“[ALMEIDA Q.,2000]

Cases de sucesso de migração para *software* livre em instituições de ensino são cada vez mais frequentes. Professora de uma escola de Ensino Fundamental de Santa Maria – RS, Vanessa Nogueira promoveu a migração na instituição em que trabalha, e conta quais os frutos colhidos de seu trabalho.

“Além de contribuir para a comunidade, de apoiar uma cultura de liberdade ao conhecimento, temos softwares estáveis, seguros, sem a ameaça de vírus. A aceitação foi “excelente” entre professores e alunos e os pais passaram a fazer parte do processo. Os estudantes tiveram a oportunidade de mostrar novas descobertas aos pais, ampliando a rede de conhecimento, e hoje muitos usam em suas casas” [ALMEIDA, 2009].

O Ministério da Educação incentiva o uso de ferramentas livres nas instituições de ensino através do Programa Nacional de Informática na Educação (Proinfo¹³). Com isto diversas soluções livres estão disponíveis no Portal do *Software* Público Brasileiro¹⁴.

13 Proinfo: é um programa educacional criado pela Portaria nº 522/MEC, de 9 de abril de 1997, para promover o uso pedagógico de Tecnologias de Informática e Comunicações (TICs) na rede pública de Ensino Fundamental e Médio.

14 Portal do *software* Público Brasileiro: é utilizado como um dos alicerces para definir a política de uso e desenvolvimento de *software* pelo setor público no Brasil. Tal política compreende a relação entre os entes públicos, em todas as unidades da federação e demais esferas de poder, e destes com as empresas e a sociedade. O endereço do portal é: <http://www.softwarepublico.gov.br>

3.3 Linguagem de Programação

Para que um computador possa executar alguma tarefa é preciso descreve-la na linguagem em que a máquina compreende. Computadores são máquinas elétricas, portanto a única coisa que processam é ligado quando há passagem de corrente elétrica, e desligado quando não há passagem de corrente. Para facilitar a compreensão, foi definido que o número 1 representa ligado e o número 0 representa desligado, essa convenção recebeu o nome de números binários. Através destes dois estados é possível representar qualquer tipo de dados em um computador, como texto, imagens, vídeos, etc. Entretanto seria impossível ao programador construir seu *software* tendo que decorar instruções em binário como 1001110010101001.

As linguagens de programação surgiram para facilitar a vida dos programadores, pois possibilita escrever instruções na linguagem humana. Para que a linguagem de programação possa ser transformada em linguagem de máquina é preciso utilizar um compilador, este programa têm a função de transformar palavras da nossa linguagem em linguagem de máquina. Além dos compiladores o programador pode fazer uso do editor, que têm a função de ajudar na escrita do código, avisando aos desenvolvedores caso exista alguma palavra que não faça parte da linguagem. *“A vantagem é que você poderá trabalhar com instruções como if, else, etc. Além de todas as facilidades oferecidas pela linguagem ao invés de gigantescos endereços binários”*. [MORIMOTO, 2004].

As linguagens diferem em sintaxe e recursos, existem linguagens específicas para desenvolvimento de programas científicos, programas comerciais, *Web sites*, etc. Antes de iniciar um programa o desenvolvedor deve

estar atento para a popularidade da linguagem de programação, pois caso a linguagem deixe de ser utilizada, provavelmente o programador terá de migrar para outra linguagem.

A companhia Tiobe¹⁵ é especializada em medir a cada mês a popularidade das linguagens de programação. A classificação é feita baseada em consulta a engenheiros de *software*, fornecedores, mecanismos de busca, como: Google, MSN, Yahoo, além de Wikipedia e Youtube. A Tiobe disponibiliza uma tabela contendo a posição que cada linguagem ocupa no *ranking* das mais populares, a posição ocupada no mesmo período do ano anterior, a classificação da linguagem em porcentagem, a variação de uso em relação ao período anterior, e o *status* que determina quais as principais linguagens de acordo com índice de popularidade. Observando a Tabela 3.1 de março de 2010 é possível verificar que a linguagem C ocupa a 2º posição no *ranking* das linguagens mais populares, tornando-a assim uma boa linguagem a ser inserida nos programas de cursos Técnicos de Informática do Brasil.

15 TIOBE: é uma companhia especializada na avaliação e monitoramento da qualidade de *software*. A empresa surgiu em 2000 e o nome vêm da seguinte frase "The Importance Of Being Earnest" em português: A impotência de ser sério. Maiores informações em: <http://www.tiobe.com/index.php/content/company/Home.html> .

Tabela 3.1 – *Ranking* das linguagens mais populares. Fonte: [TIOBE, 2010]

Position Mar 2010	Position Mar 2009	Delta in Position	Programming Language	Ratings Mar 2010	Delta Mar 2009	Status
1	1	=	Java	17.509%	-2.29%	A
2	2	=	C	17.279%	+1.42%	A
3	4	↑	PHP	9.908%	+0.42%	A
4	3	↓	C++	9.610%	-0.75%	A
5	5	=	(Visual) Basic	6.574%	-1.71%	A
6	7	↑	C#	4.264%	-0.06%	A
7	6	↓	Python	4.230%	-0.95%	A
8	9	↑	Perl	3.821%	+0.40%	A
9	10	↑	Delphi	2.684%	-0.03%	A
10	8	↓↓	JavaScript	2.651%	-0.96%	A
11	11	=	Ruby	2.327%	-0.27%	A
12	32	↑↑↑↑↑↑↑↑	Objective-C	1.970%	+1.79%	A
13	-	↑↑↑↑↑↑↑↑	Go	0.921%	+0.92%	A
14	15	↑	SAS	0.769%	-0.03%	A
15	13	↓↓	PL/SQL	0.737%	-0.31%	A
16	22	↑↑↑↑↑	MATLAB	0.661%	+0.20%	B
17	17	=	ABAP	0.639%	+0.00%	B
18	16	↓↓	Pascal	0.603%	-0.13%	B
19	19	=	ActionScript	0.594%	+0.11%	B
20	27	↑↑↑↑↑↑	Fortran	0.563%	+0.24%	B

É possível observar também que a linguagem Pascal está em 18ª no *ranking* e teve uma queda de 0,13% em relação ao mesmo período de 2009. Mesmo não ocupando posições de destaque no ranking TIOBE a linguagem Pascal é de fácil entendimento, sendo indicada para disciplinas de introdução a programação. Além das tabelas a empresa TIOBE também disponibiliza um Gráfico que mostra a evolução das linguagens nos últimos sete anos. O Gráfico

referente ao período de 2002 á 2010 pode ser visto na Figura 3.1.

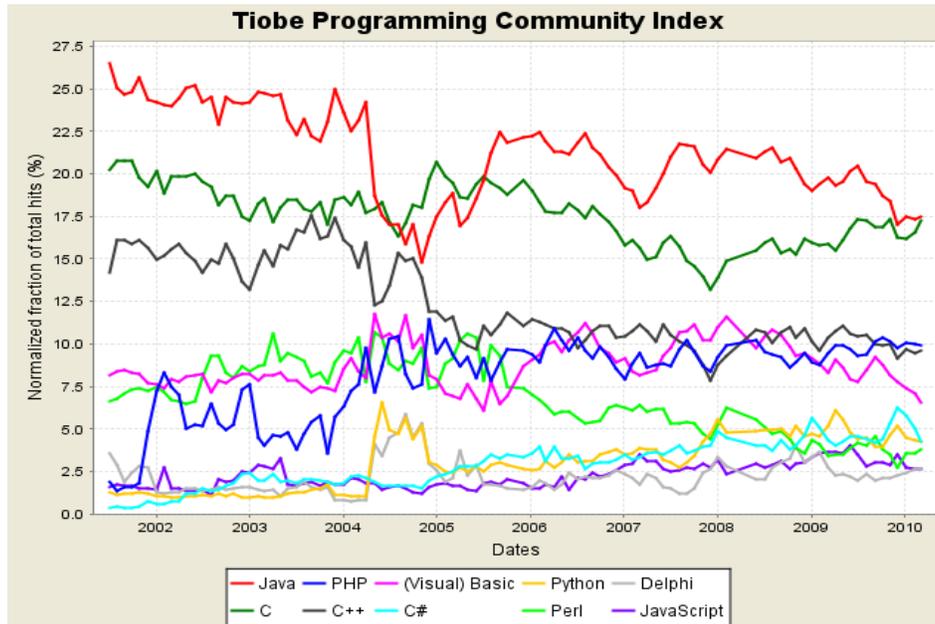


Figura 3.1: Popularidade das linguagens de programação. Fonte: [TIOBE, 2010]

Analisando a Figura 3.1 é possível observar que a linguagem C vêm disputando a liderança de popularidade com a linguagem JAVA desde 2002. A linguagem C, a linguagem Pascal, e a linguagem Java necessitam de programas que transformem suas instruções em linguagem de máquina, pois a linguagem de programação “pura e simples” serve somente para escrever instruções e funções a serem executadas pelo computador. O trabalho de transformar o código fonte em linguagem de máquina é feito pelo compilador ou interpretador. Existem diversos compiladores disponíveis para as mais variadas linguagens. Existem também programas que oferecem maior facilidade de trabalho para programadores, contendo compilador, editor de texto, depurador para corrigir problemas, entre outras ferramentas em um só *software*. A esse tipo de ferramenta da-se o nome de Ambiente Integrado de Desenvolvimento.

3.4 Ambientes Integrados de Desenvolvimento

O termo Ambiente Integrado de Desenvolvimento é uma tradução do termo em inglês IDE ou *Integrated Development Environment*. A IDE é um conjunto de ferramentas que auxiliam o programador no desenvolvimento de aplicações. Entre as ferramentas mais comuns encontradas em uma IDE pode-se destacar:

- *Editor: edita o código-fonte do programa escrito na linguagem de programação.*
- *Compilador: Compila o código fonte do programa, editando em uma linguagem específica e a transforma em linguagem de máquina*
- *Linker: Liga os vários pedaços de código-fonte, compilados em linguagem de máquina, em um programa executável que pode ser em um computador, ou outro dispositivo computacional*
- *Depurador: auxilia no processo de encontrar e corrigir erros (bugs) no código-fonte do programa, na tentativa de aprimorar a qualidade de software;*
[WIKIPEDIA, 2009]

A função de uma IDE é oferecer maior comodidade ao programador. Funções como auto-completar, lista de bibliotecas, e teclas de atalho são comuns neste tipo de aplicação. É possível encontrar no mercado uma grande lista de IDE's, proprietárias ou livres, que atendam uma ou diversas linguagens de programação. As IDE's proprietárias Turbo C++ e Turbo Pascal fazem parte do contexto de desenvolvimento desta pesquisa porque estas ferramentas já foram ensinadas aos alunos envolvidos nesta investigação. A IDE livre Geany faz parte deste estudo por apresentar atualizações constantes, interface agradável, recursos de navegação por funções, ser multiplataforma, além de oferecer todas as liberdades de um *software* livre.

3.4.1 Linguagem C (Turbo C++)

“A linguagem C foi desenvolvida no início da década de 1970 para viabilizar a implementação de software básico – mais precisamente de um sistema operacional Unix.” [REIS, 2004]

A linguagem C não têm aplicação específica, sendo mais apropriada a programação estruturada, porém é mais utilizada no desenvolvimento de compiladores, editores de texto, banco de dados, entre outras aplicações. [GACLI, 2009] afirma que:

“A linguagem C pertence a uma família de linguagens cujas características são: portabilidade, modularidade, compilação separada, recursos de baixo nível, geração de código eficiente, confiabilidade, regularidade, simplicidade e facilidade de uso.”

A transformação do código fonte em linguagem C num programa executável segue uma série de procedimentos, que podem ser vistos na Figura 3.2. O pré-processador atua no nível de código fonte, nesta etapa o comentário do programa é removido e as diretivas que começam com o caracter # são interpretadas.

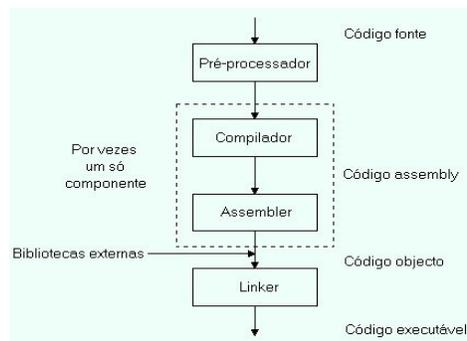


Figura 3.2 – Modelo de compilação da Linguagem C. Fonte: [FELGUEIRAS, 2009]

Os compiladores têm a função de transformar código fonte escrito em alguma linguagem de programação em uma linguagem binária, entendida pelos computadores. Alguns compiladores C transformam o código fonte em C (texto) na linguagem Assembly (texto). O Assembly traduz o código para linguagem binária. O código gerado é armazenado em arquivos de extensão .o (Unix) ou .obj (MS-DOS). O *linker* têm a função de combinar todos os objetos com o resultado compilado das funções em um único arquivo com código executável. Felgueiras em [FELGUEIRAS,2009] descreve algumas características que levaram a linguagem C a tornar-se uma das mais populares linguagens de programação, são elas:

- *Pequeno tamanho da sua definição*
- *Subdivisão do código e grande utilização de funções*
- *Alguma conversão automática entre tipos de dados (ao contrário do Pascal)*
- *Linguagem estruturada*
- *Disponibilidade de operadores para programação de baixo nível*
- *Utilização fácil e extensa de apontadores para aceder memória, vetores, estruturas e funções.*

O Turbo C++ é uma IDE que teve sua primeira versão lançada em 1990 e que podia ser executado no sistema OS/2. No ano seguinte, a versão 1.1 e 3.0 foram lançadas, especula-se que o salto da versão 1.x para 3.x deve-se a tentativa de acompanhar as versões do Sistema Operacional Windows. A versão 4.0 chegou a ser lançada em 1993, porém com o sucesso das ferramentas Delphi, e C++ Builder¹⁶, a Borland resolveu encerrar as atividades de desenvolvimento

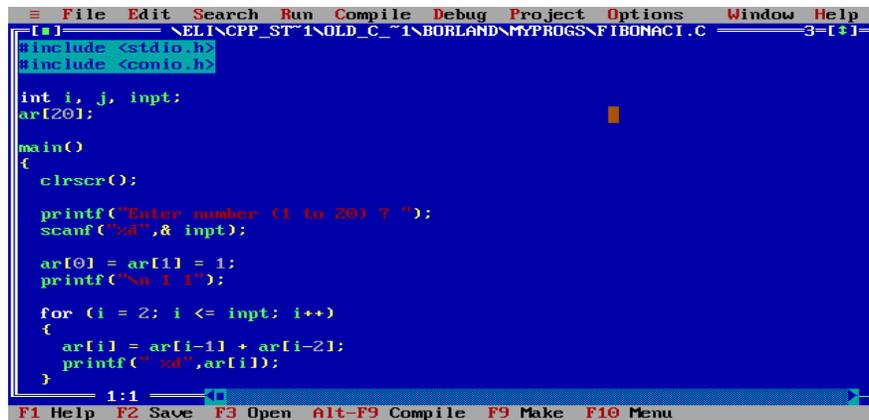
¹⁶ C++ Builder:(ou abreviado por BCB) é um ambiente de desenvolvimento integrado (IDE) produzido pela Embarcadero para a escrita de programas na linguagem C++.

da família Turbo C++.

O código fonte dessas IDE's é fechado, e a ferramenta pode ser executada apenas no Sistema Operacional Windows. O site da Borland informa que:

“Turbo Delphi, Turbo Delphi para .NET, Turbo C++ e Turbo C# estarão disponíveis no terceiro trimestre de 2006. As edições Turbo Explorer desses produtos será gratuita. Os preços para as edições Turbo Professional estará por volta de US\$500. Estudante pagará preços acadêmicos para as edições Turbo Professional que ficará em torno de US\$ 100.” [Borland, 2009]

A versão 3.0 do Turbo C++ é utilizada no IFBA Campus Barreiras desde 2007, através desta IDE o aluno edita, compila e executa códigos fontes transformando-os em programas executáveis. A Figura 3.3 mostra a interface do Turbo C++ 3.0



```
File Edit Search Run Compile Debug Project Options Window Help
\ELI\CPP_ST1\OLD_C_1\BORLAND\MYPROGS\FIBONACI.C 3-1:1
#include <stdio.h>
#include <conio.h>

int i, j, inpt;
ar[20];

main()
{
  clrscr();
  printf("Enter number (1 to 20) ? ");
  scanf("%d",& inpt);

  ar[0] = ar[1] = 1;
  printf("No. 1 1");

  for (i = 2; i <= inpt; i++)
  {
    ar[i] = ar[i-1] + ar[i-2];
    printf(" %d",ar[i]);
  }
}
```

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

Figura 3.3 - IDE Turbo C++ 3.0

Assemelha-se ao Delphi, sendo considerado sua versão em C++, de forma que muitos componentes desenvolvidos para Delphi podem ser utilizados no C++ Builder sem modificação. Maiores Informações em: <http://www.embarcadero.com/products/cbuilder>

3.4.2 Linguagem Pascal (Turbo Pascal)

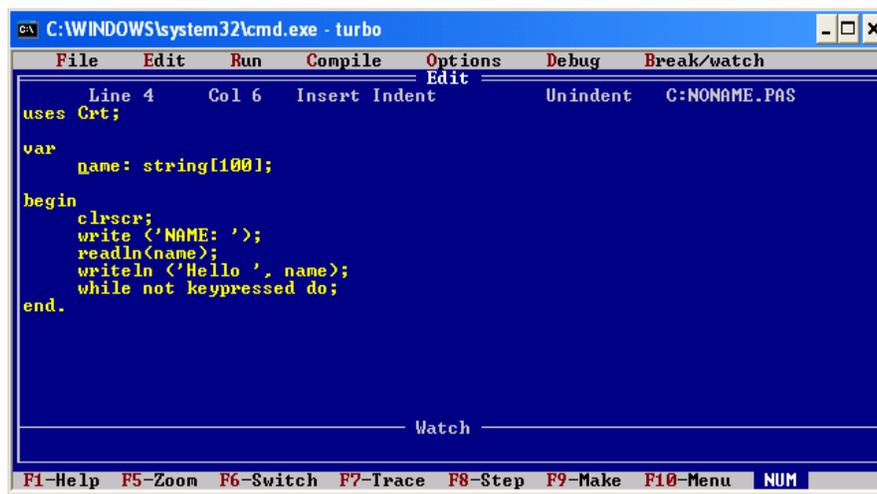
A linguagem Pascal foi criada em 1971 pelo Dr. Niklaus Wirth do *Swiss Federal Institute of Technology* (Zurich), como alternativa para programadores da linguagem ALGOL. O nome Pascal deve-se a homenagem que o criador fez ao filósofo e matemático francês do século 17 que construiu um computador mecânico-digital.

“Pascal pretendia ser ser uma linguagem educacional, e foi amplamente adotada como tal. Pascal é de escrita mais livre, diferente de FORTRAN, assim os estudantes não tinham que se preocupar com formatação”.[Turbo pascal, 2009].

Por esse motivo a linguagem Pascal foi adotada na maioria das universidades, além disso revistas especializadas em programação publicavam a maioria dos códigos nesta linguagem. Porém quando Dennis Ritch criou a linguagem C e contribuiu para criação do sistema UNIX a linguagem Pascal começou a perder adeptos. O sistema Windows também escrito em C contribuiu para que grande parte dos programadores de Pascal fossem obrigados a mudar de linguagem. Muitas universidades que antes ensinavam Pascal nas disciplinas avançadas, passaram a ensinar C, C++, e posteriormente Java. Porém, por ser uma linguagem fácil de ser aprendida e muito próxima da linguagem natural, Pascal é a linguagem preferida de universidades e escolas técnicas para ensino da introdução á programação.

Turbo Pascal é uma IDE e um compilador para linguagem Pascal. A primeira versão da ferramenta foi lançada em 1983 pela Borland, sua licença custava U\$49,99. O baixo preço fez com que o Turbo Pascal desbancasse outros compiladores como o Microsoft Pascal. Turbo Pascal se disseminou na década

de 80 e no início da década de 90 por provê facilidade no uso e baixo preço, porém em 1995 devido a mudanças nos objetivos da empresa, a Borland parou de atualizar o Turbo Pascal. A última versão da ferramenta é a 7.0, utilizada no laboratório de programação do IFBA campus Barreiras. A empresa criadora do Turbo Pascal não cobra pela licença de uso, porém seu código fonte continua fechado. A Figura 3.4 mostra a interface do Turbo Pascal 7.0 utilizado no IFBA campus Barreiras.



```
Line 4 Col 6 Insert Indent Unindent C:\NONAME.PAS
uses Crt;
var
  name: string[100];
begin
  clrscr;
  write ('NAME: ');
  readln(name);
  writeln ('Hello ', name);
  while not keypressed do;
end.
```

Watch

F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu NUM

Figura 3.4 - IDE Turbo Pascal 7.0

3.4.3 O Ambiente Integrado de Desenvolvimento Geany

O Geany foi criado para ser rápido e leve, ele pode ser executado em qualquer sistema operacional que dê suporte à biblioteca GTK, entre os principais estão Linux, FreeBSD, NetBSD, OpenBSD, MacOS X, AIX v5.3, Solaris Express e Windows. Para usuários Windows é disponibilizado um instalador que agrupa todas as dependências necessárias.

O código fonte do Geany é licenciado pela GNU GPL, com isto o usuário pode, estudar seu código fonte, modificá-lo, e distribuí-lo, desde que mantidas as regras da licença. Geany suporta uma gama de linguagens de programação, entre elas, pode-se destacar C, Java, PHP, HTML, Python, Perl, Pascal. Geany é independente de compiladores, quando o usuário insere um código e salva ele em uma extensão, o programa escolhe o compilador que irá transformar aquele código fonte em uma linguagem de máquina. Como exemplo, se é salvo um programa com extensão .c, o Geany provavelmente irá chamar o compilador GCC (*GNU Compiler Collection*), caso seja salvo um programa com extensão .pas, o Geany irá chamar o compilador Pascal fpc. A Figura 3.5 mostra a interface do Geany.

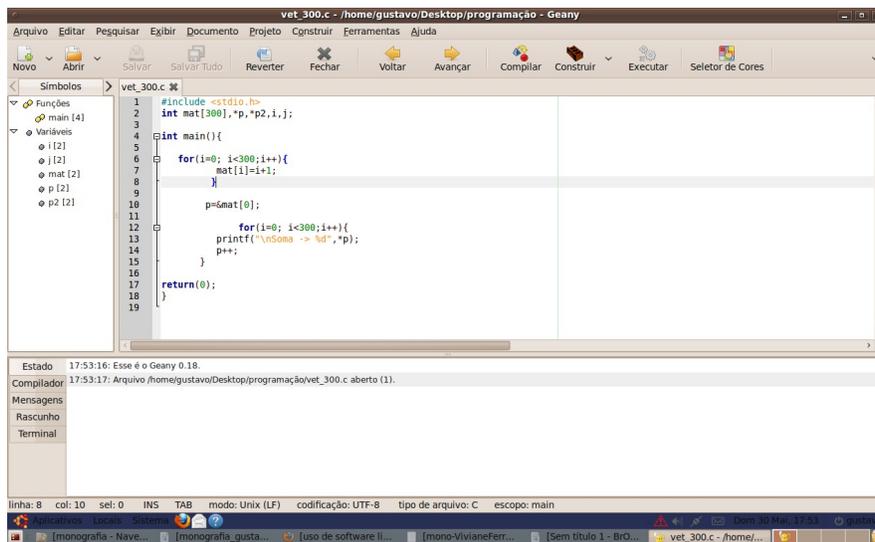


Figura 3.5 – IDE Geany

No *site* da ferramenta é possível encontrar um manual de auxílio para aqueles que desejem contribuir com melhorias, ou resolver falhas encontradas. Se o usuário não é programador é possível ainda contribuir com traduções ou construção de manuais e guias.

3.4.4 Comparativo das IDE's

A Tabela 3.2 mostra um comparativo entre as IDE's proprietárias utilizadas no IFBA campus Barreiras, e a IDE livre Geany.

Tabela 3.2: Quadro Comparativo das IDE's

----- -	Turbo C++	Turbo Pascal	Geany
Código	Fechado	Fechado	Aberto
Licença	Freeware	Freeware	GPL
Linguagens Suportadas	C / C++	Pascal	C, Java, PHP, HTML, Python, Perl, Pascal, ente outras.
Sistemas Operacionais	Windows	Windows	Linux, FreeBSD, NetBSD, OpenBSD, MacOS X, AIX v5.3, Solaris Express e Windows
Desenvolvedor	Borland	Borland	Enrico Tröger, Nick Treleaven
Última Atualização	1991	1995	Agosto de 2009

Analisando a Tabela 3.2 é possível observar que o ambiente de desenvolvimento Geany apresenta vantagens frente ao Turbo C++ e ao Turbo Pascal. Pode-se destacar que Geany têm código fonte aberto, pode ser executado em diversos Sistemas Operacionais, suporta uma grande quantidade de linguagens de programação, e recebe atualizações periódicas.

4. Metodologia

Optou-se por fazer uma análise de aceitação das novas ferramentas pelos alunos do curso técnico de informática. No início do levantamento foram ministrados minicursos abordando a importância do *software* livre e suas vantagens frente ao proprietário, e a utilização da ferramenta Geany aplicado no desenvolvimento de programas em C e Pascal. Após os minicursos foi proposto um questionário aos alunos com perguntas fechadas que pudessem mensurar o grau de aceitação da nova ferramenta. Os dados foram tabulados e transformados em gráficos para facilitar a análise.

4.1 Público Alvo Envolvido

O público alvo envolvido nesta pesquisa consiste em 18 alunos que cursaram o 4º ano do curso Técnico de Informática na modalidade integrada. Estes alunos têm experiência de programação em linguagem C e Pascal utilizando ferramentas proprietárias de desenvolvimento, por isso foram escolhidos para fazer parte da pesquisa, uma vez que poderiam responder perguntas de comparação entre as ferramentas.

4.2 Treinamento

O treinamento ocorreu entre os dias 19 e 21 de outubro no ano 2009, no laboratório de programação do IFBA campus Barreiras. Em cada dia foram ministradas 3 horas de minicurso totalizando 9 horas. Todas as máquinas estavam com o sistema operacional Linux, distribuição Ubuntu, com a ferramenta Geany instalada, além dos compiladores GCC¹⁷ e Free Pascal¹⁸. Foi usado o Virtual Box¹⁹ com uma máquina virtual contendo o Sistema Operacional Windows para que as ferramentas Turbo C++ 3.0 e Turbo Pascal 7.0 pudessem ser utilizadas.

Apesar de já terem contato com programas Livres em algumas disciplinas técnicas do curso, no dia 19 de outubro, os alunos receberam um minicurso abordando o histórico, os conceitos que envolvem este tema, as vantagens frente ao *software* proprietário, e empresas de sucesso que utilizam *software* livre. No dia 20 de outubro, os alunos foram instigados a desenvolver

17 Gcc: O GNU *Compiler Collection* (chamado usualmente por GCC) é um conjunto de compiladores de linguagens de programação produzido pelo projecto GNU. É *software* livre distribuído pela *Free software Foundation* (FSF) sob os termos da GNU GPL, e é um componente-chave do conjunto de ferramentas GNU. É o compilador padrão para sistemas operativos UNIX e Linux e certos sistemas operativos derivados tais como o Mac OS X. Maiores Informações em: <http://gcc.gnu.org/>

18 *Free Pascal*(FPC): é um compilador Pascal profissional de 32 e 64bits. Está disponível para diferentes processadores Intel x86, Amd64/x86_64, PowerPC, PowerPC64, Sparc, ARM. Os sistemas operacionais a seguir são suportados: Linux, FreeBSD, Haiku Mac OS X / Darwin, DOS, Win32, Win64, WinCE, OS / 2. Maiores Informações em: www.freepascal.org/

19 VirtualBox: VirtualBox é um *software* de virtualização desenvolvido pela Sun Microsystems que, como o VMware Workstation, visa criar ambientes para instalação de sistemas distintos. Ele permite a instalação e utilização de um sistema operativo dentro de outro dando suporte real a programas de outros sistemas. Maiores Informações em: <http://www.virtualbox.org/>

uma aplicação em C, utilizando a IDE Turbo C++ 3.0, receberam o treinamento do Geany, e desenvolveram a mesma aplicação na ferramenta livre. No dia 21, os alunos foram instigados a desenvolver uma aplicação em Pascal utilizando a IDE Turbo Pascal 7.0, não foi necessário ensinar como compilar o código em Pascal, já que Geany escolhe o compilador automaticamente assim que lhe é informado a extensão do arquivo. Os alunos desenvolveram uma aplicação em Pascal, utilizando a IDE Geany.

4.2.1 Minicurso *Software* Livre

O minicurso de *software* livre foi realizado para 18 alunos, no dia 19 de outubro no laboratório de programação do IFBA. As apresentações foram executadas com o programa de Apresentações do pacote BrOffice, com duração de 3 horas, dentre os tópicos trabalhados constam os mencionados na Tabela 4.1.

Tabela 4.1 - Tópicos do minicurso de *software* livre

1	Apresentação da Pesquisa
2	Introdução
3	Conceito de <i>software</i> livre e <i>software</i> Proprietário
4	Histórico do <i>software</i> livre
5	Vantagens do <i>software</i> livre frente ao Proprietário.
6	Exemplo de <i>Softwares</i> Livres de sucesso e empresas que adotam este tipo de programa.

4.2.2 Mini-Curso de Geany (Linguagem C)

O treinamento em Geany aplicado no desenvolvimento de programas em C aconteceu para 18 alunos presentes, no dia 20 de outubro, no laboratório de programação do IFBA. As apresentações foram feitas com o programa de Apresentações do pacote BrOffice durante 3 horas seguindo os tópicos presentes na Tabela 4.2.

Tabela 4.2 - Tópicos do minicurso de *Geany (Linguagem C)*

1	Definição de Ambiente Integrado de Desenvolvimento
2	Componentes de uma IDE
3	Revisão da estrutura de um programa em C
4	Características das IDE's Turbo C++ 3.0 e Geany
5	Desenvolvimento de uma aplicação em Turbo C++
6	Desenvolvendo aplicações em C com Geany
7	Desenvolvimento de uma aplicação no Geany

4.2.3 Minicurso de Geany (Linguagem Pascal)

O treinamento em Geany aplicado no desenvolvimento de programas em Pascal aconteceu no dia 21 de outubro, no laboratório de programação do IFBA, estavam presentes 18 alunos. A apresentação durou 3 horas, foram seguidos os tópicos presentes na Tabela 4.3.

Tabela 4.3 - Tópicos do minicurso de Geany (Linguagem Pascal)

1	Revisão da estrutura de um programa em Pascal
2	Características da IDE Turbo Pascal 7.0
3	Desenvolvimento de uma aplicação em Turbo Pascal

4	Desenvolvendo aplicações em Pascal no Geany
5	Desenvolvimento de uma aplicação no Geany

4.3 Programas Desenvolvidos

Para que fosse possível comparar facilidades e dificuldades de manuseio das ferramentas foi recomendado fazer a mesma aplicação em todas as etapas dos minicursos. Essa aplicação consiste em uma agenda estática, com informações de nome, endereço e telefone do usuário. As funções requisitadas foram inserir e mostrar. O resultado final deveria ser um programa em que o usuário pudesse escolher entre inserir dados, ou mostrar as informações já cadastradas, até o limite de 100.

5. Avaliação do Projeto Piloto

Para avaliação do projeto piloto de migração para a IDE Geany foram aplicados questionários a uma amostra de 17 alunos. Esse questionário foi elaborado tendo por base a Avaliação do uso de *software* livre em Faculdades de Computação feita por membros do Projeto *Software* Livre Brasil²⁰ com parceria e apoio do ENEC²¹. As perguntas que compõem o questionário podem ser verificadas no Apêndice A.

5.1 Análise e Discussão

Os alunos mostraram conhecimento do significado de liberdade de *software*, como é possível observar na Figura 5.1, todos escolheram a opção que melhor define o conceito deste tipo de programa.

20 Projeto *software* livre Brasil é uma rede social, mantida pela Associação *software* livre.org, que reúne universidades, empresários, poder público, grupos de usuários, *hackers*, ONG's e ativistas pela liberdade do conhecimento. O objetivo do projeto é a promoção do uso e do desenvolvimento do *software* livre como uma alternativa de liberdade econômica e tecnológica. Maiores informações em: <http://softwarelivre.org/>.

21 ENEC: A Executiva Nacional dos Estudantes de Computação foi fundada em 10 de Setembro 1993,. ENEC é o órgão de associação, coordenação, orientação e representação dos estudantes dos cursos universitários de graduação e pós-graduação na área de computação do Brasil. Maiores Informações em: <http://www.enec.org.br/>.



Figura 5.1 – Compreensão do significado de liberdade de *software*.

Essa porcentagem de acerto deve-se ao fato de que além do minicurso de *software* livre recebido pelos alunos, professores do curso técnico já haviam comentado sobre as liberdades de *software*.

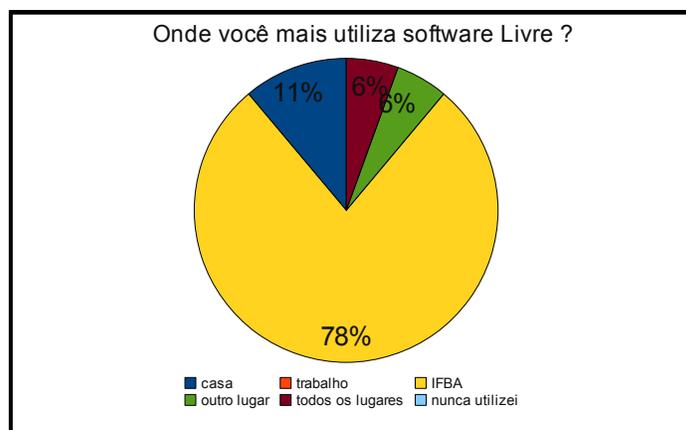


Figura 5.2 – Local de utilização do *software* livre

É possível observar na Figura 5.2 que a maioria dos alunos utilizam *software* livre somente na escola. Grande parte dos alunos do 4º ano tiveram suas primeiras experiências com *software* livre no Instituto, portanto ainda existe

dificuldades para quebrar os laços que alguns deles carregam com o uso de *software* proprietário. O *software* livre que é utilizado na escola, muitas vezes não pode ser transferido para a residência ou trabalho, lugares onde o computador é compartilhado com outras pessoas.

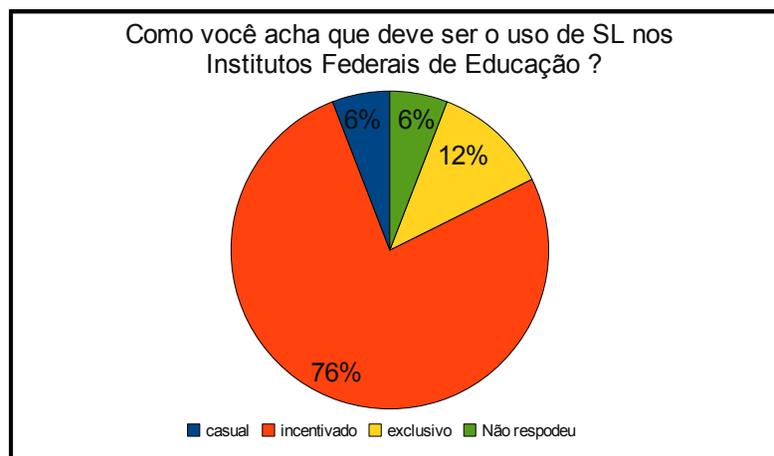


Figura 5.3 – Opinião sobre o uso de *software* livre nos Institutos Federais de Educação.

A Figura 5.3 mostra que os alunos são conhecedores da importância do *software* livre em sua formação acadêmica. A maioria dos entrevistados, 76% responderam que o uso de *software* livre nos Institutos Federais de Educação deve ser incentivado, inclusive outros 12% dos alunos mencionaram que o uso deste tipo de *software* deve ser exclusivo. Analisando as informações do Figura 5.4 é ratificada a informação que os alunos consideram o *software* livre importante na sua formação profissional. Isto porque a Figura 5.4 mostra que 94% dos alunos consideram o *software* livre necessário ou essencial para sua formação profissional.

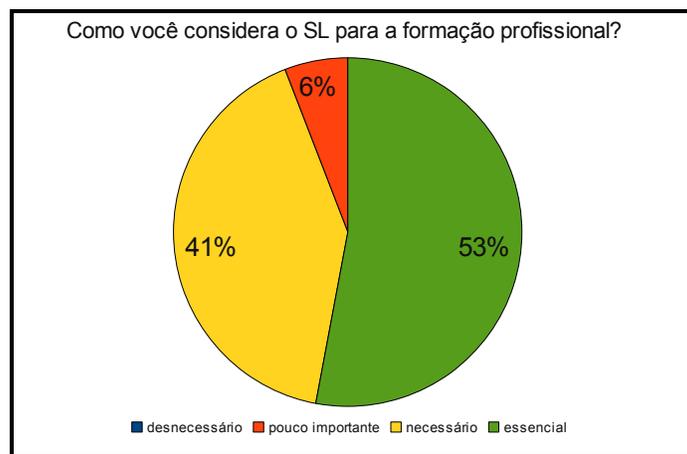


Figura 5.4 – Importância do *software* livre na formação profissional

Por serem incentivados a trabalhar com *software* livre durante o curso Técnico, de acordo com a Figura 5.5, 71% dos alunos entrevistados responderam que o *software* livre é muito utilizado nos laboratórios de ensino. Esse dado é importante para o estudo de migração pois mostra que os alunos já têm experiência com uso de *software* livre e são incentivados a utilizar este tipo de programa. O sucesso da migração para Geany torna-se mais fácil, uma vez que os alunos conhecem outros programas livre.

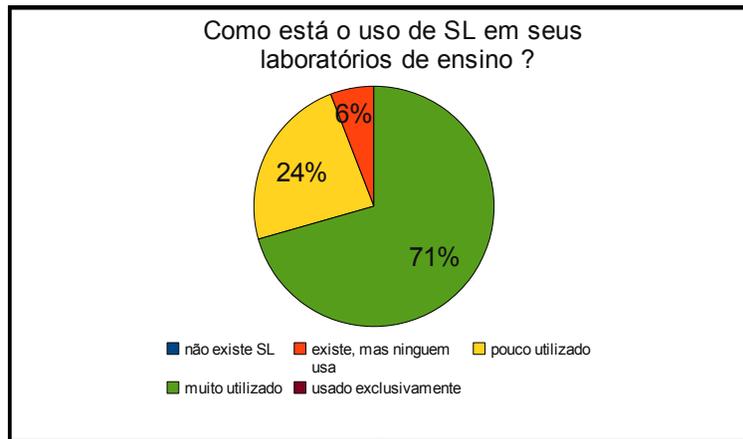


Figura 5.5 – Uso de *software* livre nos laboratórios de ensino

Ainda de acordo com a Figura 5.5, 24% dos alunos concordam que este tipo de *software* é pouco utilizado. Esse dado pode ser explicado pelo fato de alguns alunos acharem que o *software* livre poderia ser mais utilizado nos laboratórios do Instituto.

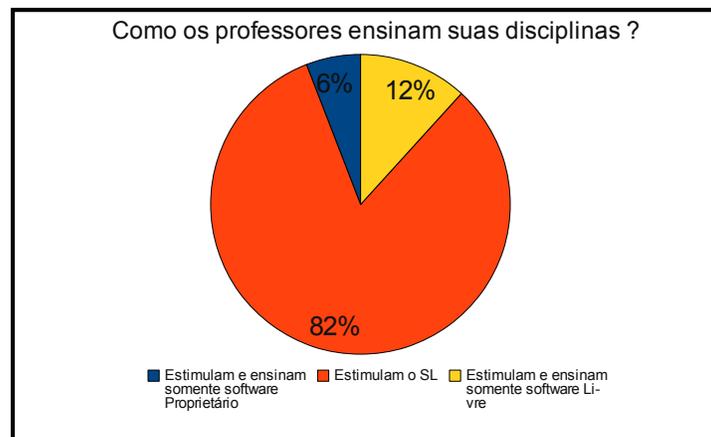


Figura 5.6 – Tipo de software incentivado pelos professores

A Figura 5.6 mostra que 94% dos alunos consideram que são estimulados a usar *software* livre pelos seus professores. Esse dado revela que os

professores de informática do campus Barreiras são conhecedores da importância do *software* livre na formação do aluno, e as vantagens que este tipo de programa traz para o Instituto.

Nenhum aluno respondeu que nunca tivesse utilizado *software* pirata, isso mostra que eles sabem que utilizam alguns programas de forma ilegal.

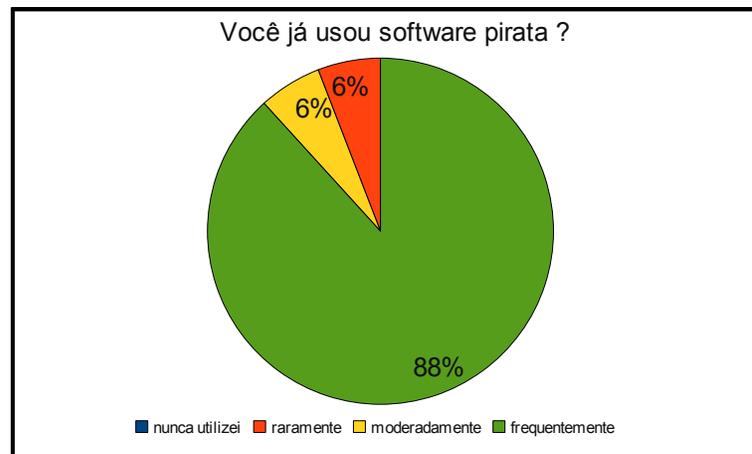


Figura 5.7 – Uso de *software* pirata.

A Figura 5.7 mostra que 100% dos alunos já utilizaram *software* pirata, confirmando a tese de que o uso de software proprietário, principalmente na sua forma ilegal ainda é grande. Entretanto, 94% dos alunos entrevistados gostariam que fosse utilizado mais *software* livre na instituição em que estudam conforme demonstra Figura 5.8. Esse fato demonstra que o conceito de liberdade de *software* e as vantagens do *software* livre frente ao proprietário foram bem assimiladas durante o curso técnico, e também durante o minicurso.

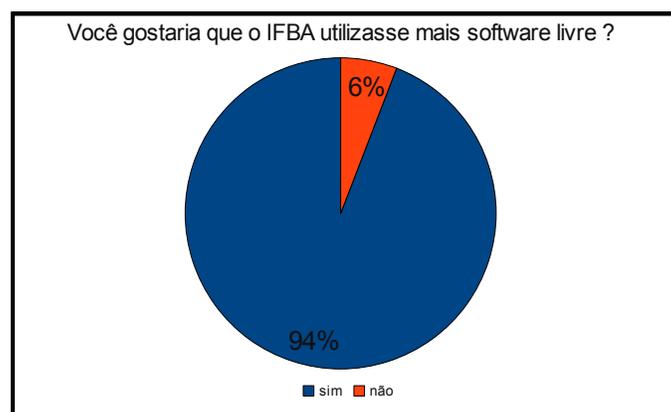


Figura 5.8 – Intenção dos alunos em aprender mais *Softwares Livres*

A Figura 5.9 mostra que 82% dos alunos apontam a falta de cultura de utilização do *software* livre como principal fator que explica o não uso deste tipo de programa nas escolas.

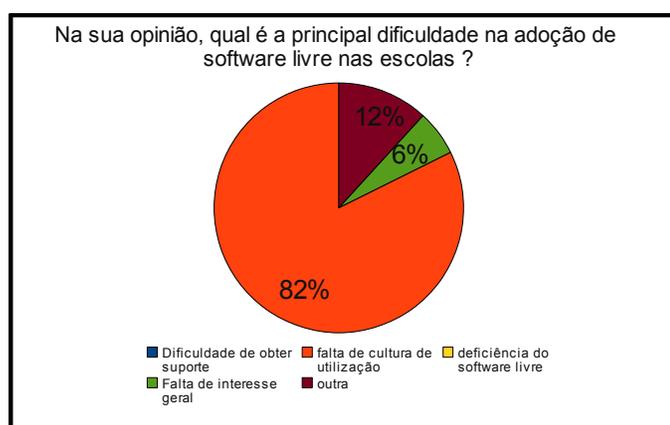


Figura 5.9 – Dificuldades de adoção do *software* livre nas escolas

A Figura 5.10 comprova que a maioria dos alunos 94% tiveram suas primeiras experiências com *software* livre através do IFBA.

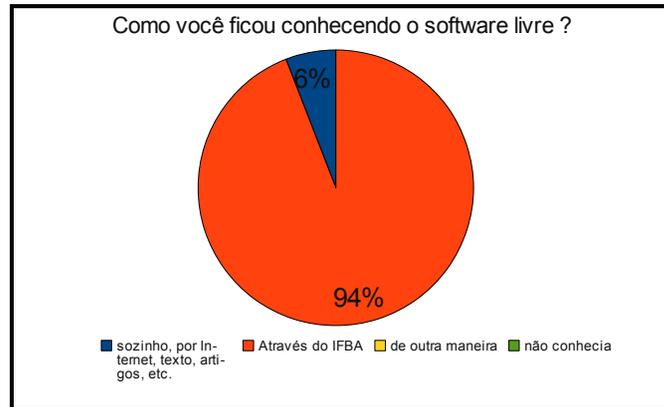


Figura 5.10 – Local onde o aluno conheceu o *software* livre

Grande parte dos alunos não sentiu dificuldades em desenvolver aplicações no Geany, isso mostra que este *software* é fácil de ser manuseado, uma vez que o tempo que os alunos tiveram para aprender a programar na ferramenta foi pequeno. Conforme Figura 5.11, 82% dos alunos não sentiram nenhuma dificuldade para programar com a IDE Geany, e apenas 18% sentiram um pouco de dificuldade.



Figura 5.11 – Dificuldades em desenvolver aplicações no Geany

Quando questionados sobre uma possível adoção do Geany nas disciplinas técnicas em que Turbo C++ e Turbo Pascal eram utilizados, podemos observar na Figura 5.12 que 72% dos alunos concordam com a idéia. Enquanto 17% dos alunos discordam e responderam que os programas proprietários devem continuar sendo usados, contra 11% que responderam não fazer diferença.

E desta forma a Figura 5.12 comprova que a maioria dos alunos assimilou a importância do *software* livre e reconhece as vantagens e liberdades oferecidas pelos Geany em relação ao Turbo C++ e Turbo Pascal. Essa informação aponta que os alunos do curso de informática aprovam a migração para Geany no desenvolvimento de aplicações em C e Pascal.

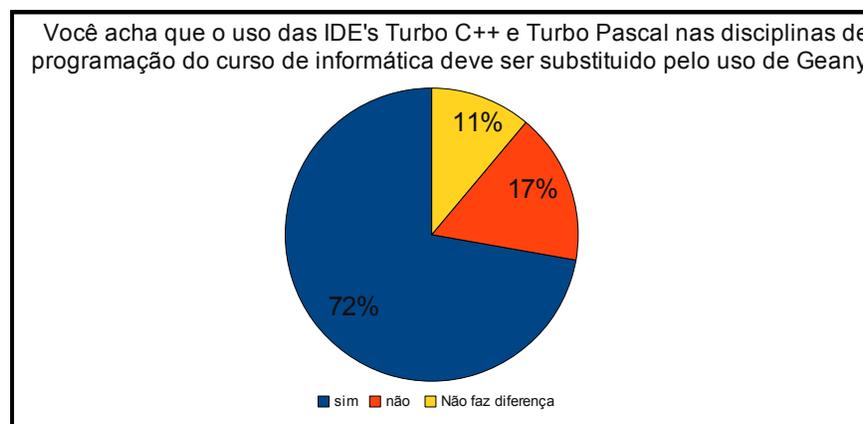


Figura 5.12 – Opinião dos alunos sobre adoção do Geany

Aos alunos que foram a favor da migração foi questionado quais os fatores que os levaram fazer essa escolha, 12 alunos responderam essa pergunta. De acordo com a Figura 5.13, 1/3 (33%) dos entrevistados acham que a interface do Geany influenciou no apoio a migração. Inclusive, essa mesma porcentagem de entrevistados 1/3 (33%) acham que as facilidades oferecidas pelo Geany são grandes com relação a quantidade de linguagens suportadas, a constante atualização, fatores esses que justificariam essa migração. Entretanto, 27% dos

entrevistados consideram as liberdades oferecidas pelo *software* livre como o principal fator que os levaram a ser a favor da migração. Apenas 7% escolheram outra opção.

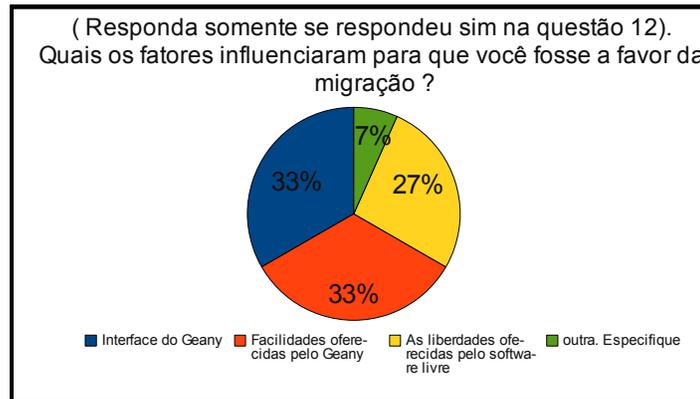


Figura 5.13 – Fatores que influenciaram na escolha do Geany

A Figura 5.14 aponta que 88% dos alunos concordam que o uso de Ambientes Integrados de Desenvolvimento livre deve ser incentivado. Enquanto que 6% dos alunos responderam que o uso deste tipo de IDE deve ser exclusivo, contra outros 6% que responderam que o uso de ferramentas livres de programação deve ser casual.



Figura 5.14 – Como deve ser o uso de *software* livre no curso de informática.

A Figura 5.14 reafirma o desejo dos alunos do curso de Informática em utilizar ferramentas de programação que recebem atualizações frequentes, que podem ter seu código fonte estudado, modificado e distribuído.

6. Conclusão

O processo de desenvolvimento de *software* torna-se mais fácil com o uso dos Ambientes Integrados de Desenvolvimento. Os alunos do curso Técnico de Informática do IFBA campus Barreiras sempre desenvolveram seus programas nas linguagens C e Pascal utilizando IDE's proprietárias. Este estudo de viabilidade de migração para IDE's livres mostrou que a maioria dos alunos tiveram seu primeiro contato com *software* livre no instituto. O poder econômico dos fabricantes de *software* proprietário fez com que este tipo de programa estivesse presente na maioria dos computadores pessoais. A falta de cultura de utilização leva muitas pessoas a terem medo de um modelo de software em que o código fonte é aberto, os usuários podem contribuir para sua melhoria, e muitas vezes não têm custo de aquisição. Porém, os resultados desse estudo de migração comprovaram que quando usuários passam a conhecer os princípios do *software* livre, e as vantagens da IDE Geany, como por exemplo: atualizações frequentes, possibilidade de programação em várias linguagens, visualização rápida de funções, liberdades do *software* livre entre outras. Essas vantagens, inclusive, não são encontradas nas IDE's Turbo C++ e Turbo Pascal, o que propicia para que o usuário perca o medo do novo e passa a preferir a ferramenta que lhe oferece mais recursos.

Essa pesquisa revelou que apenas 6% dos alunos entrevistados conhecia um *software* livre antes de entrar no Instituto. Esse fato comprova que falta cultura de utilização do *software* livre, porém é papel da escola fazer com que os alunos não fiquem aprisionados a um único modelo de *software*, como o: *software* proprietário. O IFBA campus Barreiras está empenhado nesta tarefa, uma vez que 82% dos alunos sentiram-se estimulados por seus professores a utilizar *software* livre e 71% concordam que este tipo de programa é muito

utilizado nos laboratórios da escola. Através das informações coletadas pode ser percebido que depois que o aluno conhece as vantagens e liberdades oferecidas pelo *software* livre, ele passa a admitir que este tipo de programa será importante para a sua formação acadêmica.

Os dados mostraram que as vantagens oferecidas pelo Geany superam a “eficiência” ilusória que se tem programando em IDE's proprietárias, a escolha dos alunos envolvidos na pesquisa comprova que quando os mitos que cercam o *software* livre são derrubados e pessoas passam a conhecer seus princípios e vantagens, o medo do novo deixa de existir e o *software* livre passa a ser preferência. As facilidades, liberdades, e a interface do Geany foram fatores decisivos para que alunos preferissem este *software* a continuar com os programas gratuitos, porém de código fonte fechado que não recebem atualizações a mais de 10 anos.

O Governo Federal investe na migração para *software* livre, pois não quer ficar aprisionado a empresas fabricantes de *software*, além disso economiza com a utilização deste tipo de programa. O Ministério da Educação apóia iniciativas que incentivem o uso de *software* livre nas escolas, portanto esta pesquisa realizada no IFBA só vem a fortalecer uma ação que logo será feita de maneira mais ampla envolvendo não só os programas de laboratório, mas também de escritório, gestão, entre outros presentes no instituto.

Este trabalho revelou que é possível utilizar a IDE Geany no ensino da programação em linguagem C e Pascal no curso Técnico de Informática do IFBA campus Barreiras. A maioria dos alunos entrevistados reconheceram a importância do *software* livre em sua formação profissional, apontaram vantagens em relação ao *software* proprietário que era utilizado, e manifestaram a vontade de que mais migrações possam ser feitas no instituto. Como trabalhos futuros sugere-se um estudo da implantação de ferramentas livres nos

laboratórios de programação do IFBA campus Barreiras.

7. Referências Bibliográficas

ALMEIDA Q., Rubens. **Software livre na Educação**. 2000. Disponível na internet via www. url:http://www.ccuec.unicamp.br/ead/impresao_html?foco2=Publicacoes/78095/291721&focomenu=Publicacoes&imprimir=1.

Boletim EAD - Unicamp / Centro de Computação / Equipe EAD. Arquivo capturado em: 11 de novembro de 2009.

ALMEIDA, Rafania. **Migração para *software* livre provoca descobertas na escola**. 2009. Disponível na internet via www. url: <http://biblioceha.blogspot.com/2009/08/migracao-para-software-livre-provoca.html> Arquivo capturado em 07 mar. 2010.

BAQUEIRO, Diciola F. A. **Projeto do Curso Técnico em Informática na Modalidade Integrada do CEFET-BA unidade Barreiras-BA**. Barreiras, 2006.

BACIC, Nicolas Michel. **O software livre como alternativa ao aprisionamento tecnológico imposto pelo software Proprietário**. 2003. Disponível na internet via www. url:<http://www.rau-tu.unicamp.br/nou-rau/softwarelivre/document/?code=107> Arquivo capturado em: 02 de mar. 2010.

BORLAND. **Borland's Developer Tools Group Announces Plans to Rev Up Classic Turbo™**. Disponível na internet via www. url: http://www.Borland.com/br/company/news/press_releases/2006/08_08_06_Borland_announces_plans_to_rev_up_classic_turbo.html Arquivo capturado em: 20 de out. 2009.

CHRISPINIANO, Jose. **Governos testam possibilidades do *software* livre**. SP.GOV, no. 04. FUNDAP, Fundação do Desenvolvimento Administrativo, Secretaria de Estado do Governo, São Paulo, Brasil: Marzo. 2005 1807-3131. Disponível na internet via www. url: <http://www.revista.fundap.sp.gov.br/> Arquivo capturado em 02 de outubro de 2009.

FELGUEIRAS. **O programa em C**. Disponível na internet via www. url:

http://www.carlosfelgueiras.hpg.com.br/Cursos/LinguagemC/Cap_1.html#s1_3. Arquivo capturado em 10 de agosto de 2009.

GACLI. **Introdução á Linguagem C**. Disponível na internet via www. url: <ftp://ftp.unicamp.br/pub/apoio/treinamentos/linguagens/c.pdf>. Centro de Computação. Unicamp. Arquivo capturado em 09 de agosto de 2009.

LOZANO, Fernando. **O Software Livre e o mercado de Software**. 2004. Disponível em: www.lozano.eti.br/palestras/mercado.pdf Arquivo capturado em 02 de outubro de 2009.

MORIMOTO, Carlos E.; **Linguagens de Programação**. Disponível em <http://www.gdhpress.com.br/entendendo/leia/index.php?p=cap1-20> 2004. Acesso em 07 mar. de 2010.

REIS, Adriana N. et. al. **Desenvolvimento em Software Livre**. São Leopoldo: Unisinos. 2004. 280p.

SILVA. G. M. **Guia Foca GNU/Linux**. Julho 2005. Disponível na internet via www. url: <http://www.linuxdicas.com.br/ch-intro.htm>. Arquivo capturado em 02 de março de 2010.

SILVEIRA, Sérgio. A. et. al. **software livre e Inclusão Digital**. São Paulo: Conrad, 2003. 340p.

SILVEIRA, Diego. **Free Pascal no Ubuntu**. Disponível na internet em : <http://diegosilveira.blogspot.com/2009/08/free-Pascal-no-ubuntu.html>. Arquivo capturado em 29 de setembro de 2009.

UCHÔA, Kátia Cilene Amaral. **Cibercultura e software livre**. 4º edição. Editora UFLA/FAEPE. Lavras, 2008.

TIOBE. **Programming Community Index for march 2010**. Disponível em: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. Arquivo capturado em 06 de outubro de 2009.

TURBO pascal, 2009. Disponível em: http://encyclopedia-pt.snyke.com/articles/turbo_pascal.html#Extens.C3.B5es. Acesso em 06 de

março de 2010.

WIKIPEDIA **Ambiente de Desenvolvimento Integrado**. Disponível em:
http://pt.wikipedia.org/wiki/Ambiente_de_Developolvimento_Integrado.
Arquivo capturado em 07 de outubro de 2009.

Apêndice A

Questionário - Alunos

N	Pergunta	Opções
1	Qual é a sua compreensão do significado de Liberdade de <i>software</i> ?	(a) que não tem preço
		(b) não possui autor
		(c) que possui código fonte aberto
2	Onde você mais utiliza <i>software</i> livre (SL)?	(a) casa
		(b) trabalho
		(c) faculdade
		(d) outro lugar
		(e) todos lugares
3	Como você acha que deve ser o uso de SL nos Institutos Federais de Educação?	(a) não deve ser usado
		(b) casual
		(c) incentivado
		(d) exclusivo
4	Como você considera o <i>software</i> livre (SL) para a formação profissional?	(a) desnecessário
		(b) pouco importante
		(c) necessário
		(d) essencial
5	Como está o uso de SL em seus laboratórios de ensino?	(a) não existe SL.
		(b) existe, mas ninguém usa.
		(c) pouco utilizado
		(d) muito utilizado
		(e) usado exclusivamente

6	Como os professores ensinam suas disciplinas?	(a) estimulam e ensinam somente <i>software</i> Proprietário
		(b) estimulam o SL
		(c) estimulam e usam apenas <i>software</i> livre
7	Você já usou <i>software</i> pirata?	(a) nunca
		(b) raramente
		(c) moderadamente
		(d) freqüentemente
8	Você gostaria que o IFBA utilizasse mais <i>software</i> livre?	(a) sim
		(b) não
9	Na sua opinião, qual é a principal dificuldade na adoção de <i>software</i> livre nas escolas?	(a) dificuldade de obter suporte
		(b) falta de cultura de utilização
		(c) deficiência do <i>software</i> livre
		(d) falta de interesse geral
		(e) outra
10	Como você ficou conhecendo o <i>software</i> livre?	(a) sozinho, por Internet, textos, artigos etc.
		(b) pelo IFBA.
		(c) de outra maneira.
		(d) não conhecia.
11	11. Você sentiu dificuldades em desenvolver aplicações no Geany ?	() sim
		() não
		() pouca dificuldade
12	Você acha que a uso das IDE's Turbo C++ e o	() sim

	Turbo Pascal nas disciplinas de programação do curso de informática deve ser substituído pelo uso do Geany ?	<input type="checkbox"/> não
		<input type="checkbox"/> Não faz diferença
13	(Responda somente se respondeu sim na questão 12). Quais os fatores influenciaram para que você fosse a favor da migração ?	<input type="checkbox"/> Interface do Geany
		<input type="checkbox"/> Facilidades oferecidas pelo Geany
		<input type="checkbox"/> As liberdades oferecidas pelo <i>software</i> livre
		<input type="checkbox"/> Outra Especifique
14	Como você acha que deve ser o uso de IDE's livres no ensino da programação no curso de informática ?	(a) não deve ser usado
		(b) casual
		(c) incentivado
		(d) exclusivo