

**ADRIANO JOSÉ ROSA**

**CRIAÇÃO DE AMBIENTE DE  
DESENVOLVIMENTO WEB EM LINUX**

Trabalho de conclusão apresentado ao Curso de Administração em Redes Linux (ARL) da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação Lato Sensu em Administração em Redes Linux, para obtenção do título de Especialização.

LAVRAS  
MINAS GERAIS – BRASIL  
2010

**ADRIANO JOSÉ ROSA**

**CRIAÇÃO DE AMBIENTE DE  
DESENVOLVIMENTO WEB EM LINUX**

Trabalho de conclusão apresentado ao Curso de Administração em Redes Linux (ARL) da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação Lato Sensu em Administração em Redes Linux, para obtenção do título de Especialização.

Orientador:

Prof. D.Sc. Joaquim Quinteiro Uchôa

LAVRAS  
MINAS GERAIS – BRASIL  
2010

**ADRIANO JOSÉ ROSA**

**CRIAÇÃO DE AMBIENTE DE  
DESENVOLVIMENTO WEB EM LINUX**

Trabalho de conclusão apresentado ao Curso de Administração em Redes Linux (ARL) da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação Lato Sensu em Administração em Redes Linux, para obtenção do título de Especialização.

Aprovada em 04 de dezembro de 2010.

---

Prof.<sup>ª</sup> Msc. Kátia Cilene Amaral Uchôa

---

Prof. Msc. Herlon Ayres de Camargo

---

Prof. D.Sc. Joaquim Quinteiro Uchôa  
(Orientador)

LAVRAS  
MINAS GERAIS – BRASIL

*Dedico esta monografia à minha família, em  
especial à minha mãe pelo apoio incondicional  
em todos os momentos*

## **Agradecimentos**

Agradeço ao meu orientador, Professor Dr. Joaquim Quinteiro Uchôa, que me auxiliou no desenvolvimento do presente trabalho.

Agradeço também a todos aqueles que de alguma forma contribuíram na minha formação escolar, acadêmica e profissional.

# Sumário

LISTA DE FIGURAS.....	viii
1 INTRODUÇÃO.....	1
1.1 Considerações Iniciais.....	1
1.2 Motivação.....	2
1.3 Objetivos.....	3
1.4 Estrutura do Trabalho.....	4
2 CONCEITOS BÁSICOS.....	6
2.1 Considerações Iniciais.....	6
2.2 Programação para Internet.....	7
2.3 Gerenciamento de Sistemas Linux.....	9
3 CONFIGURAÇÃO DA ESTAÇÃO DE TRABALHO.....	11
3.1 Considerações Iniciais.....	11
3.2 Definição da Distribuição Linux.....	11
3.3 Contas de Usuários.....	13
3.4 Cotas de Disco.....	16
4 FERRAMENTAS PARA DESENVOLVIMENTO DE APLICAÇÕES WEB... 18	
4.1 Considerações Iniciais.....	18
4.2 Servidores <i>Web</i> .....	19
4.3 Gerenciadores de Banco de Dados.....	20
4.4 Ferramentas de Gerenciamento de Banco de Dados.....	24
4.5 Linguagens de Programação.....	27
4.6 Frameworks.....	29
4.7 Editores de Código.....	31
4.8 Gerenciamento de Projetos.....	35
4.9 Modelagem de Projetos.....	38
4.10 IDE.....	41

5 INSTALAÇÃO DE APLICATIVOS.....	44
5.1 Considerações Iniciais.....	44
5.2 Critérios para a Escolha das Ferramentas.....	46
5.3 Preparando para a Instalação.....	48
5.4 Modelagem e Gerenciamento do Projeto.....	49
5.5 Servidores e Ambientes de Desenvolvimento.....	53
5.6 Ferramentas de Desenvolvimento.....	57
5 CONCLUSÃO.....	61
REFERÊNCIAS BIBLIOGRÁFICAS.....	63

## LISTA DE FIGURAS

Figura 3.1: Tela Inicial.....	13
Figura 3.2: Configuração das cotas de disco.....	17
Figura 4.1: Tela principal do <i>MySQL Administrator</i> .....	25
Figura 4.2: Tela principal do <i>Geany</i> .....	32
Figura 4.3: Tela principal do <i>Anjuta</i> .....	34
Figura 4.4: Editor de códigos <i>ScITE</i> .....	34
Figura 4.5: Tela principal do <i>Planner</i> .....	35
Figura 4.6: Tela principal do <i>Umbrello</i> .....	40
Figura 4.7: Tela principal do <i>ArgoUML</i> .....	40
Figura 4.8: Tela principal do <i>Bluefish</i> .....	42
Figura 4.9: Tela principal do <i>Eclipse</i> .....	43
Figura 5.1: Gerenciador de pacotes <i>Synaptic</i> .....	45
Figura 5.2: Tela principal do <i>OpenProj</i> .....	51
Figura 5.3: Tela principal do <i>Dia</i> .....	53
Figura 5.4: Tela inicial do <i>phpMyAdmin</i> .....	56
Figura 5.5: Editor de imagens <i>Gimp</i> .....	59
Figura 5.6: Cliente <i>FTP Filezilla</i> .....	60
Figura 5.7: Área de trabalho do usuário.....	60



## **CRIAÇÃO DE AMBIENTE DE DESENVOLVIMENTO WEB EM LINUX**

### **RESUMO**

Com o surgimento de novas tecnologias, o desenvolvimento de sistemas teve de adaptar-se à nova realidade. Até então, o funcionamento destes sistemas era restrito ao ambiente das empresas. Com a necessidade de as pessoas passarem a ter de estar constantemente atualizadas, as linguagens de programação tiveram que passar a oferecer recursos para que os dados pudessem ser acessados de qualquer lugar. Esta pesquisa tem por objetivo demonstrar a montagem e configuração de uma estação de trabalho a ser operada por desenvolvedores de aplicações *Web*, através da utilização de *software* livre.

**Palavras-chave:** Desenvolvimento de aplicações *Web*, *software* Livre, Interatividade, Novas Tecnologias.

# 1 INTRODUÇÃO

## 1.1 Considerações Iniciais

Muitos enganam-se quando pensam que a tarefa de um administrador de redes, principalmente um administrador de redes Linux é a de apenas gerenciar servidores e aplicações neles instalados. Além disso, esse administrador tem responsabilidades também sobre os usuários da rede e sobre as estações que operam.

Saber definir quais as funcionalidades devem ser instaladas e ouvir dos usuários quais os aplicativos necessários para que possam desempenhar suas atividades, são atitudes necessárias a um bom administrador de redes. Mas não é apenas a vida do usuário que deve ser facilitada, a comodidade do administrador também deve ser levada em conta. A estação deve estar configurada de modo a permitir que o administrador possa ter controle sobre ela sem a necessidade de estar fisicamente onde ela está instalada.

Por fim, ele deve preocupar-se em permitir que usuários tenham acesso apenas ao que lhes é permitido e evitar que eles se desviem de seu objetivo, para outras atividades indesejadas, como jogos por exemplo.

Tomada como exemplo, uma estação de trabalho destinada a um usuário especialista em desenvolvimento de aplicações para internet, o desafio é mostrar como gerenciar esta estação através da definição de contas de acesso, gerenciamento de cotas de disco e acesso a aplicativos. Por fim, a instalação de todas as ferramentas necessárias ao bom desempenho das atividades deste profissional no seu dia a dia.

## 1.2 Motivação

Com uma economia cada vez mais globalizada, a informação tornou-se um bem extremamente valioso. Administradores de empresas que conseguem ter acesso a informações em tempo ágil, tem uma capacidade maior de tomada de decisões, o que pode definir pelo fracasso ou sucesso de uma transação comercial.

Para que estes administradores possam ter as informações a mão sempre que necessário, as empresas vem modernizando seus sistemas de informação, de modo a criar um ambiente favorável à captação, processamento e disseminação da informação. Isso só é possível com a utilização de sistemas gerenciais bem planejados, que visem a atender pelo menos a grande maioria das necessidades da empresa.

Neste sentido, as empresas de desenvolvimento de sistemas computacionais tem buscado ferramentas de desenvolvimento cada vez mais eficientes para a criação de seus sistemas. Quanto mais recursos uma ferramenta tiver, o sistema desenvolvido por meio dela torna-se mais poderoso, podendo oferecer a possibilidade de um usuário acessar dados de qualquer lugar do mundo, através da comunicação via internet.

Com isso há uma busca constante de novas tecnologias para a criação de ambientes de desenvolvimento de sistemas internet. A busca de novas ferramentas de desenvolvimento visa agilizar e modernizar o processo de desenvolvimento, gerando um sistema com uma melhor qualidade a um custo cada vez menor.

## 1.3 Objetivos

O número de usuários deixando de utilizar sistemas proprietários e passando a usar soluções livres vem crescendo constantemente nos últimos anos. Soluções livres, como o Linux, por exemplo, sempre foram bastante utilizadas em ambientes corporativos. Mas sempre em servidores ou equipamentos de usuários com um nível de conhecimento mais elevado, visto as dificuldades que usuários menos experientes encontravam na sua utilização, principalmente no que diz respeito a instalação e utilização de programas diversos.

Um dos objetivos deste projeto é demonstrar que as distribuições Linux, principalmente a distribuição Ubuntu, desenvolveram-se muito nos últimos anos. Já existe hoje uma facilidade muito grande na instalação de aplicativos e *drivers* de componentes do computador. Uma das grandes dificuldades que era a instalação de aplicativos, visto que a grande quantidade de dependência e a presença muitas vezes de conflitos entre estas dependências já não existe mais. Hoje, é necessário apenas que se saiba o que se quer instalar, o resto é feito pelos gerenciadores de pacotes da própria distribuição.

A presença de aplicativos de uso geral é bastante grande nos repositórios de pacotes das distribuições, mas aplicações gerenciais de uso específico das empresas ainda precisam ser desenvolvidas. Para o desenvolvimento destas aplicações os programadores tendem a utilizar ferramentas de desenvolvimento de licença proprietária, muitas vezes pirateadas. Isso ocorre porque os desenvolvedores consideram que seria mais fácil a utilização de uma ferramenta “fechada”, que mantém sempre o mesmo padrão de instalação e utilização e também pela sensação de impunidade,

por não enfrentar problemas com a utilização de uma ferramenta proprietária sem pagar pela sua licença.

Este trabalho também tem o objetivo de mostrar que é possível criar um ambiente de desenvolvimento de sistemas completamente livre em uma empresa de desenvolvimento de *software* de pequeno porte. Hoje existem ferramentas livres que contemplam todas as necessidades das diferentes fases de construção de um projeto de *software*, desde a sua fase de análise até a fase de testes.

Para atingir estes objetivos, este trabalho visa mostrar o como é simples a instalação e configuração de uma distribuição Ubuntu. Ele objetiva mostrar também as diferentes opções de ferramentas de desenvolvimento disponíveis e a facilidade de instalação destas ferramentas.

No final, dentre as opções de ferramentas pesquisadas, serão escolhidas aquelas que farão parte do ambiente de desenvolvimento da empresa em questão. A escolha das ferramentas será baseada em critérios que levarão em conta a quantidade de recursos computacionais disponibilizada pela empresa, a integração entre as diferentes ferramentas, a facilidade de encontrar ajuda sobre a utilização das ferramentas de desenvolvimento, a opinião pessoal dos membros da equipe de desenvolvimento, entre outros.

## **1.4 Estrutura do Trabalho**

O trabalho está organizado da seguinte forma: o Capítulo 2 apresenta os conceitos que darão um norte à execução deste trabalho, destacando o detalhamento dos conceitos de desenvolvimento de aplicações para internet e gerenciamento de sistemas computacionais nele apresentado.

No Capítulo 3 serão mostradas as configurações realizadas em uma estação de trabalho, descrevendo o processo de instalação de uma distribuição Linux e como foram definidas as permissões e configurações diversas da estação. No Capítulo 3 também são descritas algumas configurações desejáveis, como permissões de acesso a pastas e definição de quantidade de espaço em disco disponibilizado a cada um dos usuários

O Capítulo 4 apresenta algumas das opções de aplicativos que podem servir de ferramenta para a construção de cada uma das etapas do ciclo de vida de um aplicativo. No Capítulo 5 serão detalhados os aplicativos instalados e suas configurações iniciais, instalação de ferramentas de desenvolvimento de aplicações para internet, gerenciamento e armazenamento de dados através de aplicativos de criação e manipulação de bancos de dados. Aplicativos para editoração de imagens e animações também são mostrados neste capítulo.

Por fim, no Capítulo 6 serão apresentadas as conclusões tiradas durante a realização deste trabalho, expondo possíveis dificuldades em todo o processo. desenvolvimento, apontando as estratégias de implementação adotadas e as principais dificuldades encontradas.

## **2 CONCEITOS BÁSICOS**

### **2.1 Considerações Iniciais**

Com o desenvolvimento tecnológico recente, a informação tornou-se um bem extremamente valioso. A necessidade que os indivíduos passaram a ter de estar constantemente atualizados mudou a forma de desenvolvimento de sistemas como era conhecido até então.

Programas de computador que eram originalmente desenvolvidos para ser acessados apenas dentro de um ambiente corporativo, tiveram que passar por transformações. Passou-se a ter a necessidade de que os dados fossem acessados foras das empresas, para que administradores, por exemplo, pudessem ter noção do andamento de seus negócios mesmo estando fora das empresas.

A forma mais eficiente de prover este tipo de acesso a dados de qualquer lugar do mundo passou a ser a internet, em virtude de sua abrangência. Com isso, as linguagens de programação tiveram que passar a oferecer recursos para que os sistemas passassem fazer o compartilhamento dos dados, e ainda, oferecendo uma maior interatividade.

Mas não apenas as linguagens de programação sofreram transformações. Os ambientes de desenvolvimento também tiveram que se adaptar às novas técnicas de desenvolvimento. Consequentemente, os profissionais responsáveis pela configuração destes ambientes também foram se atualizando.

Segue-se então ao detalhamento dos conceitos inerentes ao desenvolvimento de sistemas para internet e também sobre o gerenciamento do sistema responsável pelo ambiente de desenvolvimento dos mesmos.

## 2.2 Programação para Internet

A internet, também conhecida como *World Wide Web*<sup>1</sup> (Rede de Alcance Mundial) ou simplesmente *Web* constitui-se como a mais eficiente forma de disseminação de informação nos últimos anos. Porém, passou, e ainda passa por aprimoramentos para que o conteúdo por ela publicado passe a ser mais atrativo e interativo com o usuário.

No início, a internet não apresentava muitos recursos. O desenvolvimento de páginas, por exemplo, era limitado, sem grandes recursos para torná-las mais funcionais, ou seja, não havia a possibilidade de agregar funcionalidades que as tornassem mais ágeis na apresentação de informações e mais atrativas ao usuário.

Durante os primeiros anos, a *Web* não permitia a usuários sem um conhecimento mais avançado, contribuir com o conteúdo disponibilizado. A ausência de ferramentas de uso simplificado impediam que todos os usuários publicassem conteúdos na *Web* – não existiam os grandes expoentes da *Web* nos dias de hoje, os *blogs* e *wikis* – gerando assim uma certa exclusão digital destes usuários.

Com o surgimento de novas tecnologias, surgiu então a necessidade de realizar mudanças para que as limitações apresentadas fossem solucionadas permitindo interatividade e agilidade na propagação da informação. Surgiu então a

---

1 [http://pt.wikipedia.org/wiki/World\\_Wide\\_Web](http://pt.wikipedia.org/wiki/World_Wide_Web)



*Web 2.0*<sup>2</sup>. A integração com gerenciadores de bancos de dados, o desenvolvimento de ferramentas gráficas para a criação de animações, componentes para reprodução de vídeos são alguns exemplos de ferramentas desenvolvidas para a *Web 2.0*.

O criador da expressão *Web 2.0*, Tim O'Reilly, a define da seguinte forma:

“*Web 2.0* é a mudança para uma internet como plataforma, e um entendimento das regras para obter sucesso nesta nova plataforma. Entre outras, a regra mais importante é desenvolver aplicativos que aproveitem os efeitos de rede para se tornarem melhores quanto mais são usados pelas pessoas, aproveitando a inteligência coletiva.” (WEB2.0, 2007)

Com isso, a *Web* passou a ser uma plataforma, assim como os diferentes sistemas operacionais existentes, para a execução de aplicações, as quais podem funcionar de forma integrada. O conteúdo dos *websites* também passou por transformações, possibilitando aos usuários participar da elaboração do conteúdo disponibilizado, mesmo que este conteúdo tenha sido gerado por terceiros. Passou-se a ter a possibilidade de enriquecê-lo através de comentários em *blogs*, por exemplo. Há com isso uma expansão no segmento de Programação para Internet.

Em Programação para Internet determinam-se os componentes da interface com o usuário, ou seja, as tecnologias da linguagem de programação e os recursos aplicados na interface gráfica do sistema. De acordo com Hunter & Crauford (2002), o desenvolvimento de aplicações para internet tem sido uma tendência na década de 2000. Ela é utilizada para tornar dinâmica uma página ou estender a funcionalidade de um servidor, criando um aplicativo. Enquanto uma página exibe

2 <http://oreilly.com/web2/archive/what-is-web-20.html>

um conteúdo estático simplesmente, permitindo que o usuário navegue nesse conteúdo, um aplicativo oferece uma experiência mais interativa. Pode ser tão simples como uma pesquisa de palavras chave em um armazenamento de documentos, ou complexo como uma loja virtual. Estes aplicativos estão sendo distribuídos na internet, intranets e extranets de empresas, conseqüentemente mudando a visão de como as empresas fazem seus negócios.

## **2.3 Gerenciamento de Sistemas Linux**

O gerenciamento ou administração de sistemas consiste em conjunto de tarefas necessárias para manter o computador em boas condições de uso. Entre essas tarefas podem ser destacadas a criação e execução de rotinas de cópias de segurança de dados, instalação e atualização de programas, criação de contas de usuários, garantir que os sistemas de arquivos estejam íntegros, entre outras (WIRZENIUS et al., 2005).

O gerenciamento de um sistema computacional, estabelece uma sequência de etapas que devem ser seguidas. A configuração de dispositivos e periféricos, o gerenciamento dos processos e contas de usuários, instalação, configuração e atualização de aplicativos entre outros (UCHÔA, 2007).

No caso em questão, o que se pretende é o gerenciamento de um sistema Linux, onde os usuários são desenvolvedores de sistemas para internet. Portanto aplicam-se, neste caso, as etapas de gerenciamento supra mencionadas, como: a instalação e configuração da estação de trabalho escolhida como ambiente, a instalação e configuração de ferramentas de desenvolvimento e gerenciadores de

bancos de dados, bem como a gestão dos usuários participantes do ambiente serão implementadas durante a execução do trabalho.

No próximo capítulo serão detalhadas a escolha da distribuição utilizada e as configurações nela realizadas para o gerenciamento de usuários e sistemas de arquivos.

## **3 CONFIGURAÇÃO DA ESTAÇÃO DE TRABALHO**

### **3.1 Considerações Iniciais**

Neste capítulo é relatado o processo de instalação e configuração de uma estação de trabalho para um usuário que tem como função, em uma determinada empresa, atuar no desenvolvimento de aplicações para internet. Desta forma será relatado como foram feitas as configurações de permissões de usuários e também o processo de instalação da distribuição Linux escolhida para este estudo.

### **3.2 Definição da Distribuição Linux**

Existem, em 2010, disponíveis uma infinidade de distribuições Linux que serviriam de opção para a realização do estudo a que este trabalho se propõe. A escolha aqui se baseia em uma distribuição com uma boa gama de *drivers* (*software* responsável pelo correto funcionamento dos dispositivos de um computador) de dispositivos disponíveis nativamente para facilitar o trabalho de instalação e configuração do equipamento.

Uma outra questão a ser levada em consideração é o fato de a distribuição possuir uma forma fácil e amigável de instalação e atualização de aplicativos. Usuários sem muita experiência em Linux geralmente tem dificuldades em compilar pacotes de aplicativos. Portanto, a distribuição deve possuir uma interface o mais simples possível para a instalação de aplicativos para tornar facilitada a vida

do usuário.

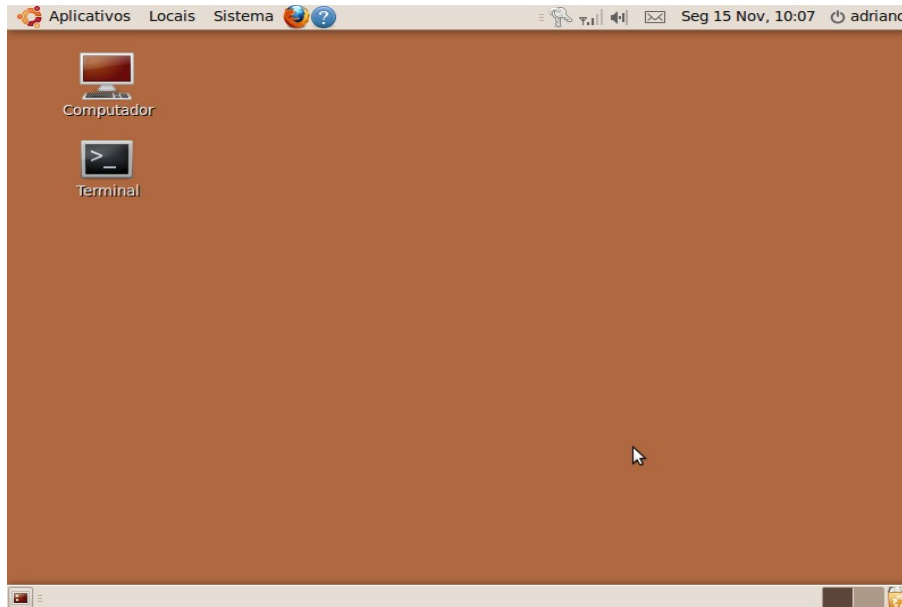
Para o sistema operacional da estação de trabalho, foi utilizada a distribuição Ubuntu, uma das distribuições que mais tem adquirido adeptos nos últimos anos. Usuários iniciantes tem preferência por esta distribuição por apresentar as facilidades supra mencionadas.

O Ubuntu apresenta-se como uma das distribuições Linux que mais se desenvolveu nos últimos anos. Desde 2004 quando foi lançada, o Ubuntu quebrou vários paradigmas das distribuições que a antecederam, principalmente no que diz respeito ao procedimento de instalação ser bastante simples e possuir interface amigável com o usuário (COSTA, 2009).

Outro ponto de destaque do Ubuntu, diz respeito ao lançamento de novas versões, duas por ano, sendo uma em abril e outra em outubro. Isso permite que se tenha um planejamento de atualizações em redes que o utilizam (COSTA, 2009).

A Figura 3.1 mostra a tela inicial da distribuição Ubuntu instalada. O procedimento de instalação da distribuição apresenta-se de forma simples. Seguindo o padrão de assistente de instalação, o usuário vai apenas selecionando as opções de acordo com suas preferências. O que merece uma certa atenção é o particionamento do disco rígido. Apesar de uma interface amigável que define o tamanho da partição em *Mega Bytes* (MB), um usuário pode atrapalhar-se caso o computador já tenha outro sistema operacional instalado.

Dependendo do ambiente em que se encontra o computador, até mesmo as configurações da rede são detectadas automaticamente. No final da instalação é verificada a presença de versões mais recentes de componentes que já são automaticamente instaladas.



**Figura 3.1:** Tela Inicial

Uma observação importante a ser feita é que não serão detalhados neste trabalho, os processos de instalação da distribuição ou dos aplicativos utilizados. Apenas serão mencionados e relatadas possíveis configurações. Para maiores detalhes, basta consultar as referências citadas no texto.

### **3.3 Contas de Usuários**

Terminado o procedimento de instalação, é chegada a hora de se fazer as configurações na distribuição, principalmente no que diz respeito às definições de permissão de contas de usuários. Uma conta principal com um maior número de privilégios é criada durante o procedimento de instalação. Mas, esta conta não

possui poderes para a definição de configurações mais avançadas. Configurações estas que devem ser executadas por meio de uma conta de usuário específica, o usuário root.

Por padrão, a distribuição Ubuntu não vem com a senha do usuário root definida, devendo ser criada assim que possível. Esta senha pode ser criada a qualquer tempo com o usuário acessando o terminal da distribuição e definindo a senha através da utilização do comando *passwd*. Definida a senha do usuário root, passa-se à configuração das demais contas de usuários.

Esta estação irá trabalhar basicamente com desenvolvimento de interfaces de páginas de internet, páginas dinâmicas, criação de imagens, com acesso a banco de dados, enfim tudo a que se refere ao desenvolvimento de aplicações internet. Para tanto foi criado um grupo *webdesenv*, ao qual fará parte o usuário principal deste computador, bem como os demais usuários que possam vir a ter acesso de alguma forma a esta estação.

Feito isso, é chegada a hora de criar as demais contas de acesso a este computador. Com o comando *adduser*, as contas de usuários podem ser criadas de forma simples. Cada uma das contas criadas neste computador que se destinem a usuários de desenvolvimento de aplicações internet devem ser adicionadas ao grupo *webdesenv*.

Em seguida, para que outros usuários possam utilizar os arquivos de projeto armazenados nesta estação de trabalho, cada um com sua própria conta, foi criado um diretório */webdesenv*, que possui como grupo padrão o grupo *webdesenv*. Com isso, basta que se cadastre os possíveis usuários desenvolvedores neste grupo para que eles passem a ter permissões de leitura e escrita a este diretório.

Para os demais usuários foram retiradas todas as permissões de acesso a este diretório, para garantir uma certa privacidade dos desenvolvedores. Para esta configuração foi utilizado o comando *chmod*. Através dele, podem ser definidas permissões de acesso a arquivos e diretórios para o seu dono, os usuários que fazem parte do grupo do dono e para os demais usuários.

Entretanto, há formas mais práticas para gerenciamento de contas de usuários, através da utilização de protocolos e módulos de autenticação, tais como, *PAM*<sup>3</sup> (*Pluggable Authentication Module*), *kerberos*<sup>4</sup> e *OpenLDAP*<sup>5</sup>. Com a utilização destes recursos, pode-se cadastrar a conta de usuário em um servidor e definir suas permissões de acesso apenas uma vez. Com isso, um usuário pode se logar em diferentes estações de trabalho usando sua conta de acesso, sem a necessidade de que a conta seja cadastrada em todas as máquinas. Mas isso não será detalhado, para maiores informações sobre *PAM*, *Kerberos* e *OpenLDAP* e respectivas formas de implementação em redes Linux podem ser vistas em (MORGAN, 2002), (DOMINGUES; SCHNEIDER; UCHÔA, 2001), (JORDÃO, 2005) e (MOTA, 2008).

Continuando as configurações do computador, foram bloqueadas ainda com o *chmod*, as permissões do diretório *games* dentro do diretório */usr*, bloqueando acesso a jogos para usuários comuns. Com isso, definem-se as configurações de permissões de usuários para este computador. Com o decorrer da execução deste trabalho, podem aparecer mais algumas configurações que serão realizadas em momento oportuno.

---

3 <http://www.kernel.org/pub/linux/libs/pam/>

4 <http://www.kerberos.org/>

5 <http://www.openldap.org/>



### 3.4 Cotas de Disco

Uma outra configuração interessante é o gerenciamento de cotas de disco para os usuários. Com as cotas de disco para usuários define-se a quantidade de espaço em disco que cada um irá ter para armazenar seus arquivos. Esta cota de uso de disco pode ser definida não só para usuários individualmente, mas também para grupos de usuários, o que facilita o gerenciamento.

Neste caso, usuários que fazem parte de grupos de usuários diferentes que tem cotas de disco especificadas, podem fazer uso de sua própria cota e também da cota dos grupos aos quais ele faz parte. Por isso, deve-se ter atenção no momento em que a definição das cotas será feita, para que determinados usuários possuam cotas muito grandes e outros usuários possuam cotas com espaço muito restrito.

A finalidade da definição de cotas para este caso é para evitar que, usuários baixem arquivos muito grandes da internet, como músicas, por exemplo, ou armazenem arquivos desnecessariamente, ocupando assim, espaço com arquivos inúteis. Além das cotas para usuários, também foram realizadas as configurações de cotas para os grupos de usuários, uma vez que se poderia utilizar a pasta de projetos para armazenar os tais arquivos. Para isso se faz uso do pacote de aplicativos de gerenciamento de cotas de disco, o *quota*.

A Figura 3.2 mostra como ficou a configuração das cotas de disco na estação, ilustrando qual o espaço em disco para cada um dos usuários e os grupos cadastrados na distribuição.

```

*** Report for user quotas on device /dev/sda5
Block grace time: 7 dias; Inode grace time: 7 dias
                Block limits                File limits
Usuário        used  soft  hard  grace  used  soft  hard  grace
-----
root          -- 44068      0      0          5      0      0
arluser       -- 15832 800000 810000      266      0      0
cortuno       -- 15300 100000 110000      356      0      0

```

**Figura 3.2:** Configuração das cotas de disco

## **4 FERRAMENTAS PARA DESENVOLVIMENTO DE APLICAÇÕES WEB**

### **4.1 Considerações Iniciais**

O segmento de desenvolvimento para *Web*, nos dias atuais, se apresenta bastante amplo nos dias atuais. Pode-se desenvolver um simples sistema de informações gerenciais para empresas de pequeno porte, com processamento e consulta de dados realizado por poucos usuários. Por outro lado, há a possibilidade de desenvolver um complexo sistema de comércio eletrônico, com volume de transações de dados bastante superior, requerendo ferramentas e técnicas que garantam a integridade dos dados.

É de acordo com o projeto a ser desenvolvido que as ferramentas a serem empregadas são escolhidas. Em aplicações mais complexas, não seria viável, por exemplo, a utilização de um gerenciador de bancos de dados com recursos de controle de transações, integridade e replicação de dados limitados. A possibilidade de ocorrência de violação de integridade de informações seria grande.

Por outro lado, se o que se dispõe em termos de recursos de processamento é limitado, não adianta que se construa um sistema utilizando ferramentas que são verdadeiros devoradores de recursos. O sistema não conseguirá ser executado de forma adequada em termos de velocidade de processamento gerando descontentamento por parte do cliente.

Para o estudo a que se propõe este trabalho será tomado por base o desenvolvimento de sistemas gerenciais mais simples, mais voltados à realidade de um mercado do interior de Minas Gerais. Não serão abordados a fundo, por exemplo, ferramentas de geração de animações e tratamento de imagens. Apenas serão observadas as ferramentas necessárias à modelagem, construção e desenvolvimento serão abordadas.

Existem hoje disponíveis uma infinidade de ferramentas de código aberto que podem servir ao propósito deste trabalho, configuração de uma estação de trabalho para um desenvolvedor de aplicações *Web*. A seguir serão descritas algumas destas ferramentas divididas em categorias de acordo com sua funcionalidade dentro do projeto.

## 4.2 Servidores Web

Um servidor *Web* constitui-se como o programa responsável por disponibilizar páginas, fotos, ou qualquer outro tipo de objeto ao navegador internet do cliente. Tem ainda a capacidade de receber dados do cliente, através da utilização de formulários de entrada de dados, por exemplo, processá-los através de aplicações, e enviá-los de volta ao usuário.

Dentre a gama de opções de programas para instalação disponíveis atualmente, pesquisadas para a realização deste trabalho, talvez os servidores *Web* sejam os que tem o menor número de opções. O mercado de utilização de servidores *Web* livres está restrito basicamente à utilização de apenas um. O servidor *Apache*<sup>6</sup>, é um servidor *Web* extremamente configurável, robusto e de alta performance. Tem

---

<sup>6</sup> <http://www.apache.org/>

como características principais o fato de ter seu código fonte aberto, o que o constitui como o servidor *Web* mais utilizado, com um número de usuários maior do que a soma de todos os usuários de outros servidores (SILVA, 2007).

Segundo dados de uma pesquisa divulgada em novembro de 2010, no site da empresa Netcraft<sup>7</sup>, o servidor *Apache* domina 59,36% da demanda de utilização de servidores *Web* no mundo. Seu principal concorrente, que conta com uma fatia de 22,7% do mercado, é um servidor de licença proprietária, que funciona em apenas uma plataforma de sistema operacional, não sendo portátil para outros sistemas operacionais como o Linux, o que inviabiliza sua utilização neste trabalho.

O restante do mercado é dividido entre outros servidores *Web*, dentre eles o *nginx*<sup>8</sup>. Na página de seu projeto na internet, estão presentes a sua documentação, mas o conteúdo informativo sobre ele é bastante vago. Isso ocorre diferentemente com o Apache, por exemplo, que em sua página oficial possui uma gama muito ampla de informações sobre si, o que acaba por estimular a sua utilização.

Neste caso, a opção pela utilização do *Apache* na execução destes estudo torna-se praticamente inevitável, visto a impossibilidade de utilização de seu principal concorrente e também a precariedade de informações, na visão do autor sobre a utilização do *nginx*.

### **4.3 Gerenciadores de Banco de Dados**

Os sistemas gerenciadores de banco de dados são ferramentas utilizadas no armazenamento, e também em alguns casos, no processamento dos dados

---

<sup>7</sup> <http://news.netcraft.com/>

<sup>8</sup> <http://nginx.org/>

manipulados pelos programas de computador. Além do armazenamento, a realização de consultas de dados através de instruções específicas.

“Um banco de dados por si só pode ser considerado como o equivalente de um armário de arquivamento, ou seja, ele é um repositório ou recipiente para uma coleção de arquivos de dados computadorizados. Os usuários de um sistema podem realizar (ou seja, solicitar que o sistema realize) diversas operações envolvendo tais arquivos. Um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros, em outras palavras, é um sistema computadorizado cuja finalidade é armazenar informações e permitir que usuários acessem e atualizem essas informações quando as solicitar” (DATE, 2003).

Dentre os gerenciadores de banco de dados disponíveis, serão citados 3, o *MySQL*<sup>9</sup>, *FireBird*<sup>10</sup> e o *PostgreSQL*<sup>11</sup>, começando pelo *MySQL*.

Segundo informações de seu site oficial, o *MySQL* conta hoje com mais de 100 milhões de usuários espalhados por todo o mundo. Apoiado em características como, velocidade, agilidade e facilidade de uso, o *MySQL* tornou-se uma opção bastante interessante para empresas nos mais diferentes ramos. O *MySQL* é um banco de dados de código aberto mas, que nos últimos tempos tem despertado a desconfiança e apreensão de seus usuários, pelo fato de que a empresa que o mantinha ter sido adquirida por uma grande empresa de desenvolvimento de *software*, que tem exatamente como seu principal produto um sistema de banco de

---

9 <http://dev.mysql.com/tech-resources/articles/introduction-to-mysql-55.html>

10 <http://www.firebirdsql.org/>

11 <http://www.postgresql.org/>

dados de licença proprietária. O temor dos usuários se baseia no fato de que essa empresa possa fechar seu código fonte obrigando as comunidades de programadores que colaboraram com o projeto a lançar um novo projeto baseado na última versão de código disponibilizada.

O *FireBird* é um sistema gerenciador de bancos de dados que se baseia em oferecer um serviço de armazenamento e gerenciamento de dados com alta performance que suporta características básicas, porém importantes de gerenciadores de bancos de dados mas avançados. *Triggers* (em inglês gatilhos, significa que o *Firebird* tem a capacidade de executar determinadas instruções de manipulação de dados quando um determinado evento acontece em uma tabela do banco de dados.) e *stored procedures* (que são pequenos códigos de programação em linguagem SQL que podem ser disparados diretamente do código fonte das aplicações, evitando que este código tenha de ser agregado ao código fonte da aplicação, melhorando o desempenho da execução da rotina, visto que o código se encontra implementado dentro do banco de dados).

O *Firebird* originado de um projeto anterior, o qual foi adquirido por uma empresa desenvolvedora de *software*, que “fechou” o código fonte. Com isso, os membros da comunidade que mantinham o projeto foram obrigados a criar um novo a partir da última versão disponível do código fonte, mantendo o projeto ativo até hoje.

Já o *PostgreSQL* é um sistema gerenciador de banco de dados que vem desenvolvendo-se ao longo de aproximadamente 15 anos. Portável para diferentes arquiteturas de sistema operacional, o *PostgreSQL* também possui integração com diversas linguagens de programação. Ele tem nele implementadas as mais modernas tecnologias existentes nos mais avançados sistemas de banco de dados, como

controles de transação, replicação de dados entre diferentes servidores, entre outras. Possui ainda uma excelente capacidade de armazenamento de dados, sendo que os bancos de dados nele armazenados são limitados pela capacidade de armazenamento do equipamento que provê o armazenamento destes dados, sendo assim o mais desenvolvido dos 3 gerenciadores de bancos de dados pesquisados.

É importante destacar que as características citadas como presentes no *FireBird* estão presentes também no *PostgreSQL*, que suporta uso de *stored procedures* e *triggers* mesmo antes de o *FireBird* ter sido lançado. Isso mostra que o *FireBird* apresenta apenas as características mais básicas existentes no *PostgreSQL*.

O *PostgreSQL* apresenta ainda uma característica a mais, não presente nos outros gerenciadores de banco de dados, que é a de ser um gerenciador objeto-relacional, que combina as características dos bancos de dados relacionais com o paradigma de orientação a objetos. Maiores detalhes sobre orientação a objetos serão apresentados mais adiante, juntamente com a apresentação das linguagens de programação.

Os 3 gerenciadores de banco de dados apresentados são os mais conhecidos e utilizados pela comunidade de *software* livre, especialmente em aplicativos para *Web*, mas não somente. Cabe ressaltar que há uma grande preferência pelo MySQL por boa parte dos desenvolvedores, especialmente em projetos menos complexos. À medida que o aplicativo tende a tornar-se mais complexo, há uma tendência de preferência pelo *PostgreSQL*, ou ao menos a suporte ao *PostgreSQL* e *MySQL*.



## 4.4 Ferramentas de Gerenciamento de Banco de Dados

Os gerenciadores de bancos de dados tem a total capacidade de construir, manipular e gerenciar os bancos de dados neles contidos sem a necessidade de nenhuma ferramenta externa. Porém, essas tarefas geralmente são executadas em interface modo texto, com isso o administrador de banco de dados ou usuário conheça a sintaxe dos comandos necessários para criar um banco de dados, definir suas tabelas, buscar dados armazenados entre outros.

Em virtude disto, foram criadas diversas ferramentas para auxiliar nestas tarefas, para que os usuários tenham uma interface mais amigável para a confecção e administração do banco de dados. É evidente que alguns destes comandos devem ser de conhecimento, pelo menos básico do usuário, uma vez que eles são utilizados dentro do código fonte do programa para que sejam geradas consultas ao banco de dados, inserção, alteração e exclusão de dados e demais operações necessárias para que o sistema obtenha o máximo de resultados do banco de dados.

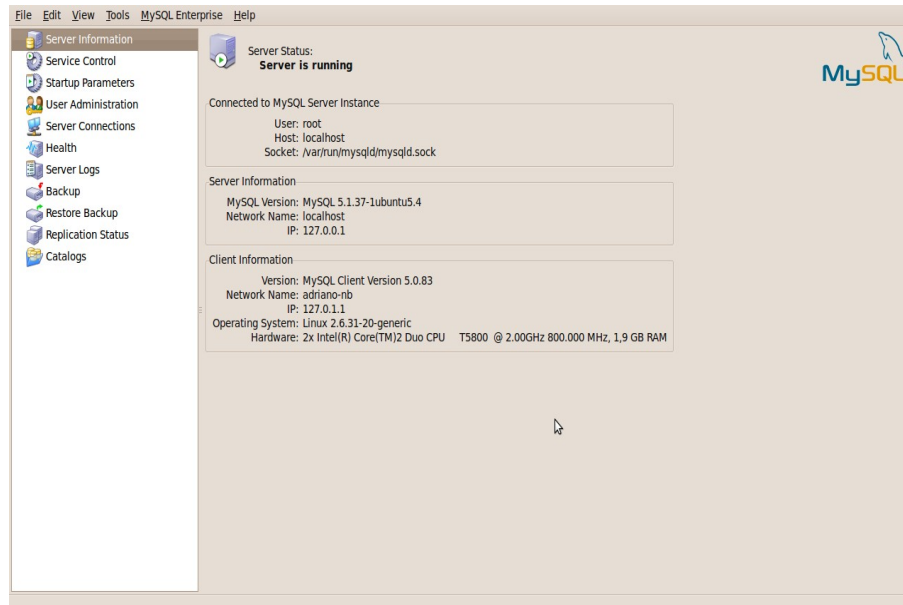
A partir de agora serão descritas algumas destas ferramentas para que delas sejam escolhidas aquelas que mais irão favorecer o estudo deste trabalho.

A primeira das ferramentas pesquisadas é o *MySQL Administrator*<sup>12</sup>, uma ferramenta gráfica capaz de gerenciar todo o funcionamento de um banco de dados *MySQL*. Através dele podem se gerenciar contas de usuários no banco de dados, definir suas permissões de acesso, acessar estatísticas do gerenciador de banco de dados, como tempo de funcionamento e ainda definir a estrutura física dos bancos de dados, suas tabelas e demais componentes do banco. É uma ferramenta

---

<sup>12</sup> <http://dev.mysql.com/downloads/workbench/5.2.html>

disponibilizada pela própria empresa mantenedora do *MySQL*, portanto, possui total compatibilidade com ele. O *MySQL Administrator*, apresentado na Figura 4.1, não possui apenas a manipulação dos dados inseridos na tabela, ou seja, nele é possível definir a estrutura, mas não se manipula os dados.



**Figura 4.1:** Tela principal do *MySQL Administrator*

Em virtude disto passa-se a ter a necessidade de possuir uma ferramenta que fizesse a manipulação destes dados diretamente nas tabelas, em virtude de possíveis manutenções diretamente nos dados. Uma ferramenta que cumpre bem este papel é o *phpMyAdmin*<sup>13</sup>. Ele é uma ferramenta que se destina à administração de bancos de dados *MySQL* por meio de interface *Web*, construindo e manipulando bancos de dados através da execução de qualquer comando em linguagem *SQL*, ou

---

13 [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)

através da utilização de opções gráficas. Uma outra característica interessante é o seu suporte a idiomas.

Até aqui foram apresentadas apenas ferramentas para a utilização em conjunto com o gerenciador de banco de dados *MySQL*. A seguir serão apresentadas ferramentas para auxiliar a utilização dos outros gerenciadores apresentados.

Disponível para diferentes plataformas de sistemas operacionais, o *pgAdmin*<sup>14</sup> se constitui como uma das mais populares ferramentas para a administração de bancos de dados *PostgreSQL*. O *pgAdmin* tem a capacidade de confeccionar desde consultas *SQL* bastante simples até bancos de dados extremamente complexos, suportando todas as funcionalidades presentes no *PostgreSQL*. Ainda possui outras funcionalidades, tais como, editor de códigos *SQL*, agendador de tarefas, suporte a mecanismos de replicação de dados, manipulação dos dados diretamente nas tabelas do banco de dados entre outras.

Assim como existe o *phpMyAdmin*, uma ferramenta que se destina à construção e administração de bancos de dados *MySQL*, por meio de interface *Web*, também existe uma ferramenta para *PostgreSQL*, o *phpPgAdmin*<sup>15</sup>. Essa ferramenta possui as mesmas características presentes no *phpMyAdmin*, no qual o *phpPgAdmin* foi baseado. Execução de instruções *SQL*, execução de cópias de segurança de bancos de dados, interface *Web* para construção e administração de bancos de dados, facilidade de instalação e configuração, entre outras. Sua versão 3 foi lançada em 2002 para que o *phpPgAdmin* pudesse oferecer suporte a uma das principais características do *PostgreSQL*, a utilização de *schemas* (esquemas).

---

14 <http://www.pgadmin.org/index.php>

15 <http://phpgadmin.sourceforge.net/>

Esquemas são conjuntos de objetos de bancos de dados agrupados por usuários e grupos. Entre estes objetos podem ser citados, as tabelas do banco de dados, *triggers*, estruturas de índices, entre outras. Quando se instala o gerenciador de banco de dados é criado um banco de dados e dentro dele podem ser criados diversos esquemas que podem funcionar separadamente ou integrados através de relacionamentos entre seus objetos.

Para a manipulação de bancos de dados do *FireBird* foi escolhido o *FlameRobin*<sup>16</sup>, uma ferramenta que se destina à administração de bancos de dados que pode ser portado para diferentes plataformas de sistema operacional. Possui uma interface gráfica bastante simples, permitindo a administração de bancos de dados de forma bastante intuitiva.

Existem disponíveis para administração de bancos de dados, uma grande quantidade de ferramentas, o que foi mostrado nesta sessão, foram apenas as ferramentas mais conhecidas para a realização desta tarefa. Vejamos agora algumas possibilidades de linguagens de programação disponíveis.

## 4.5 Linguagens de Programação

Uma das partes mais importantes de um projeto, se não a mais importante é a definição da linguagem de programação a ser utilizada. Existem diferentes linguagens de programação disponíveis com características que as tornam mais ou menos eficientes para a criação de um projeto. A definição de qual linguagem de programação a ser utilizada depende de uma série de fatores, tais como, o ambiente em que o programa será executado, o equipamento sobre o qual o sistema será

---

<sup>16</sup> <http://www.flamerobin.org/>

executado, a licença sobre a qual a linguagem de programação e o próprio programa desenvolvido são distribuídos, entre outros.

Existem linguagens de programação específicas para determinadas plataformas de sistema operacional e também aquelas que podem ser executadas em qualquer uma delas, uma característica chamada de portabilidade. Esta característica está fortemente presente em linguagem para desenvolvimento de sistemas *Web*. Para que um sistema seja executado basta que a plataforma de sistema operacional possua um cliente internet e passa-se a ter um ambiente favorável à sua execução.

A seguir serão apresentadas algumas das linguagens de programação que podem servir de ferramentas para o estudo a que este trabalho se propõe.

O *Java*<sup>17</sup> é uma linguagem de programação desenvolvida para rodar em plataformas *desktop* e internet. Seus arquivos *Applet* (arquivos que funcionam como mini-programas que agilizam o desempenho de aplicativos internet em *Java*) são bastante pequenos. O *Java* cria um ambiente que possibilita a execução de aplicações em qualquer ambiente computacional através de uma máquina virtual, podendo ser portátil a diferentes plataformas de sistema operacional e arquiteturas de *hardware* (GONÇALVES, 2006).

Uma segunda possibilidade de linguagem de programação é o *PHP*<sup>18</sup>, uma linguagem essencialmente voltada ao desenvolvimento *Web*. *PHP* é uma linguagem interpretada, ou seja, não necessita ser transformada em linguagem de máquina por um compilador, basta que o sistema possua instalado um navegador internet capaz de apresentar as páginas construídas dinamicamente através do código *PHP*. Os servidores *Web* são os responsáveis pela interpretação do código *PHP*,

---

17 <http://www.oracle.com/technetwork/java/index.html>

18 <http://www.php.net/>

transformando o que foi escrito pelo programador em ações de construção das páginas internet. O código *PHP* fica mesclado ao código *HTML* e sua principal finalidade é a escrita de páginas que serão geradas dinamicamente e muito mais.

Por fim, pode-se citar a linguagem *Python*<sup>19</sup>, uma linguagem extremamente poderosa, interpretada e orientada a objetos. Sua sintaxe de comandos é bastante concisa e clara com recursos poderosos, recursos esses que podem ser expandidos por meio da utilização de módulos desenvolvidos por terceiros. Através da utilização de um interpretador interativo de comandos é possível realizar alterações no código fonte no momento de sua execução, sendo possível a captura de uma exceção causada por um erro na sintaxe de comandos.

Pode-se verificar que cada uma das linguagens mostradas, apresenta características semelhantes, a fim de alcançar o objetivo de desenvolver uma aplicação para atender um objetivo específico.

## 4.6 Frameworks

Um *framework*<sup>20</sup> é uma abstração que une códigos de diferentes projetos provendo uma funcionalidade genérica. Através de uma configuração, um *framework* pode atingir uma funcionalidade específica ditando o fluxo da aplicação diferentemente de bibliotecas de funções que são estáticas, ou seja, tem sempre o mesmo comportamento.

Pode-se citar como vantagens na utilização de *frameworks* uma maior facilidade na detecção de erros. Os *frameworks* possuem um alto nível de abstração,

---

19 <http://www.python.org/>

20 <http://pt.wikipedia.org/wiki/Framework>

tornando assim mais eficiente o processo de desenvolvimento de um *software*, aumentando a produtividade com maior otimização dos recursos de desenvolvimento.

Cada linguagem de programação pode ter *frameworks* específicos para ela, assim como um conjunto de sistemas pode ter *framework* um específico para eles. A seguir serão apresentados alguns *frameworks* para as linguagens de programação descritas anteriormente.

O *Mentawai*<sup>21</sup> é um *framework* para desenvolvimento *Web* para a linguagem de programação *Java*. Ele se diferencia de *frameworks* mais antigos, pois sua configuração é feita através de códigos de programação *Java* e não em arquivos de configuração como era até então. Isso o torna menos suscetível a erros, uma vez que o código *Java* é compilado antes de ser carregado pela aplicação, assim se existir algum problema na configuração estabelecida no código, ela será identificada no ato da compilação do programa. Pode ser integrado com interfaces de desenvolvimento para ampliar suas funcionalidades, como o acréscimo de recursos como auto-completar na digitação do código.

Já o *Django*<sup>22</sup> é um *framework* escrito em *Python* para desenvolvimento de aplicações *Web* de modo rápido e de fácil entendimento. Ele permite que aplicações *Web* sejam desenvolvidas rapidamente, com uma interface bastante amigável e com alto desempenho.

Para terminar, uma outra opção de *framework* disponível é o *CakePHP*<sup>23</sup>, que como o próprio nome sugere, foi desenvolvido para linguagem *PHP*. O *CakePHP*<sup>24</sup> é compatível com diferentes versões do *PHP*, não necessitando sua

---

21 <http://www.mentaframework.org/>

22 <http://www.djangoproject.com/>

23 <http://www.cakephp.com.br/>

24 <http://www.cakephp.com.br/>

atualização em caso de mudança de versão da linguagem de programação. Sua comunidade de desenvolvedores é extremamente ativa, sendo desenvolvido para funcionamento do *PHP* em conjunto com o servidor *Apache* e o gerenciador de banco de dados *MySQL* (SILVA,2007).

Não se pode dizer que existe um *framework* melhor que outro, apenas que existem *frameworks* que implementam uma determinada funcionalidade ou desenvolvidos para resolver um problema específico. Mas a sua principal finalidade é a de agilizar o processo de desenvolvimento de uma aplicação fazendo com que ela seja desenvolvida no menor tempo possível, aumentando o nível de qualidade do projeto.

## **4.7 Editores de Código**

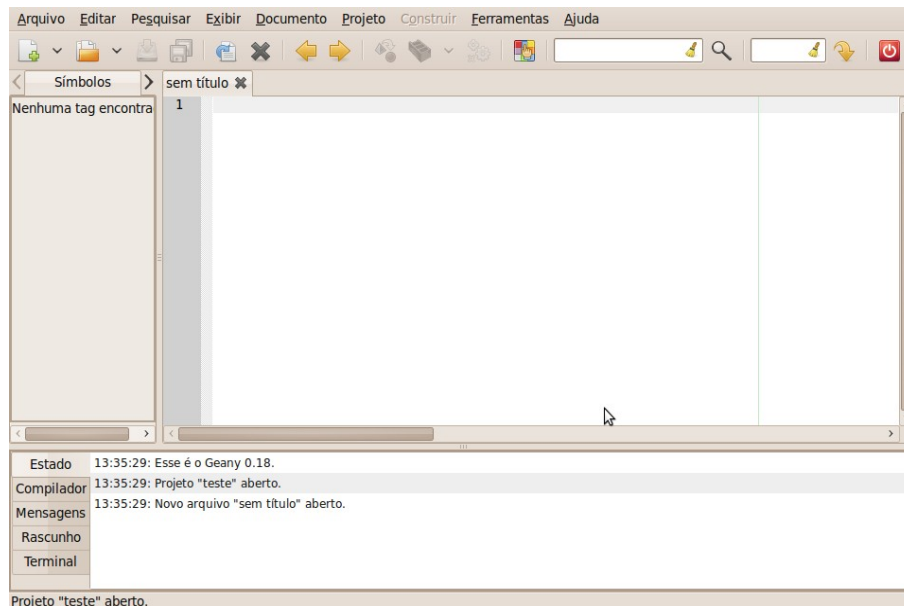
A partir do momento em que se define uma linguagem de programação a ser utilizada em um projeto, deve-se pensar em como será feita a edição do código de programação dos sistema, ou seja, qual editor será utilizado na confecção do código fonte do sistema. Existem diversas ferramentas que trazem consigo as mais diferentes funcionalidades, desde funções implementadas dentro do próprio programa, ou integradas a programas externos.

Algumas das funcionalidades desejáveis a um editor deste tipo são a verificação em tempo de execução da sintaxe dos comandos da linguagem de programação, auxílio na chamada de procedimentos e funções, mostrando ao programador os parâmetros que devem ser passados para esses procedimentos e o tipo de dados que estes parâmetros devem receber. Identificação do tipo de indentação de código utilizada pelo programador ou a opção de configurá-la, para



que a cada vez que se abrir uma nova linha de código, o editor já posicione o cursor na posição adequada tornando o código legível e alinhado.

Um destes editores é o *Geany*<sup>25</sup>, um editor que se apresenta como extremamente leve com suporte a idioma português. Para utiliza-lo em um projeto, basta ao desenvolvedor criar um novo projeto e definir qual a linguagem de programação que será utilizada. Ele ainda auxilia na geração de comentários de código, duplicação de linhas de código entre outras. A Figura 4.2 mostra a tela principal do *Geany*.



**Figura 4.2:** Tela principal do *Geany*

Já o editor *Anjuta*<sup>26</sup>, constitui-se como um ambiente de desenvolvimento bastante versátil, que possui funções básicas de gerenciamento de projeto e controle

<sup>25</sup> <http://www.geany.org/>

<sup>26</sup> <http://projects.gnome.org/anjuta/>

de versões. Também possui uma opção bastante interessante que é a presença de um depurador de códigos interativo, o que facilita o desenvolvimento de aplicações visto que a sintaxe do código já é analisada antes da compilação do sistema. Possui também integração com compiladores. A Figura 4.3 mostra a tela principal do *Anjuta*.

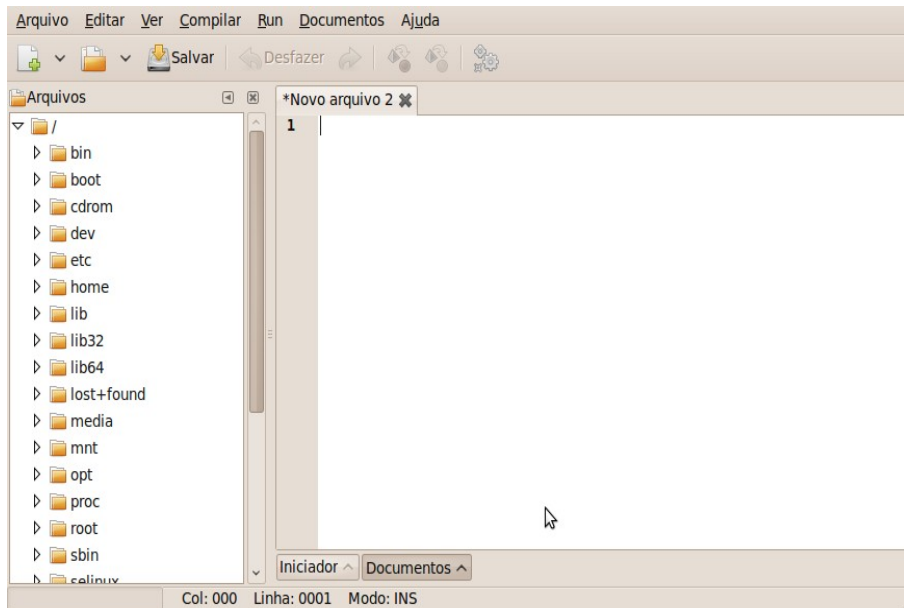
A maioria das funcionalidades do *Anjuta* é obtida pela utilização de *plug-ins* que podem expandir suas funcionalidades diretamente ou o integrado a outros sistema. Um exemplo disso é o controle de versionamento de código oferecido por ele através de *plug-ins* que o interligam a outros programas específicos para esta função. Programas para controle de versionamento de código serão descritos mais adiante.

No mesmo segmento dos editores anteriores, o *SciTE*<sup>27</sup>, ilustrando na Figura 4.4, é um editor de interface mais simples onde se define um projeto e a linguagem de programação a ser utilizada. É mais indicado para a utilização em ambientes de configuração mais simples, visto que não apresenta muitos recursos. A principal funcionalidade encontrada foi a integração com diferentes compiladores de acordo com a linguagem de programação escolhida para a elaboração do projeto se este for o caso.

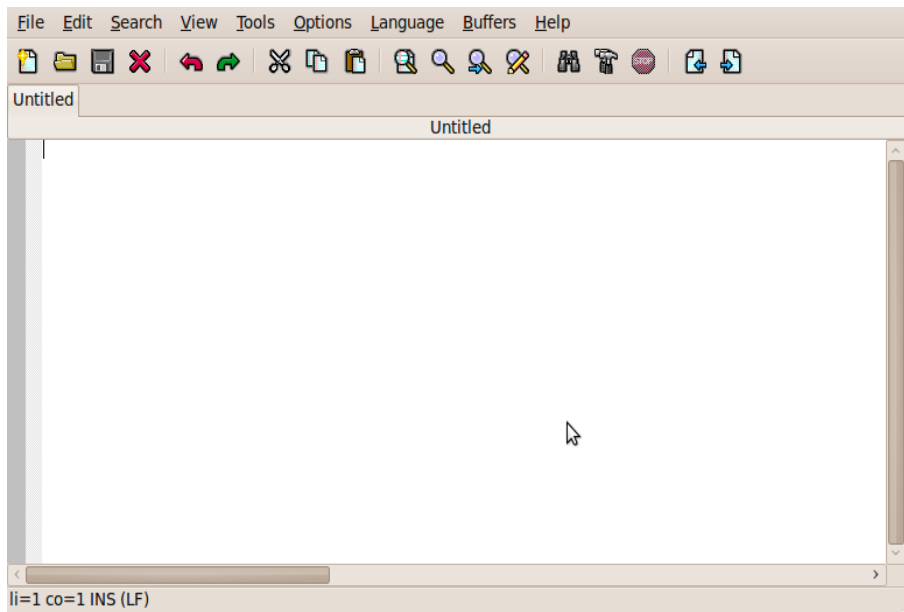
O que se verifica é que os editores apresentam funcionalidades semelhantes, porém, muitas vezes necessitam de integração com sistemas de terceiros ou ainda a utilização de *plug-ins* para estender suas funcionalidades. A seguir serão mostradas ferramentas para um melhor gerenciamento do projeto. Serão apresentadas ferramentas para controle de versões de códigos fonte de projeto e gerenciamento de determinadas fases do desenvolvimento de um projeto.

---

<sup>27</sup> <http://www.scintilla.org/SciTE.html>



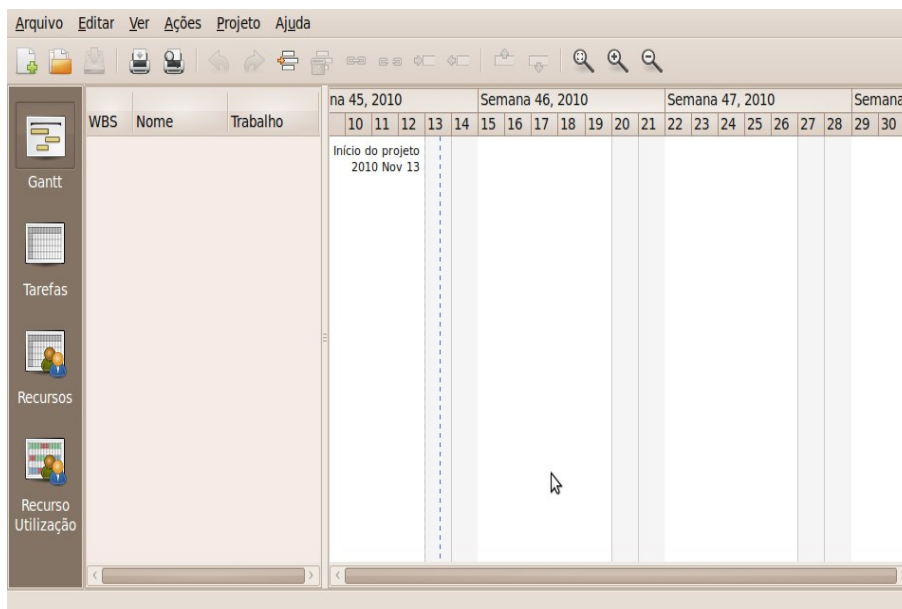
**Figura 4.3:** Tela principal do *Anjuta*



**Figura 4.4:** Editor de códigos ScITE

## 4.8 Gerenciamento de Projetos

Gerenciar um projeto significa ter o controle sobre diferentes áreas que o compõem. Quando se inicia um projeto, geralmente se estabelece um tempo estimado de produção, um custo de desenvolvimento, dimensionamento de pessoas e ferramentas para desenvolvimento, entre outros. Para um melhor gerenciamento é necessário que se possua ferramentas para organizar e dar suporte para saber se o projeto caminha de acordo com o esperado ou se existe algum tipo de atraso. Para esta tarefa merecem destaque o *Planner*<sup>28</sup> e o *OpenProj*<sup>29</sup>. A tela principal do *Planner* é apresentada na Figura 4.5.



**Figura 4.5:** Tela principal do *Planner*

<sup>28</sup> <http://live.gnome.org/Planner>

<sup>29</sup> <http://openproj.org/>

A utilização das duas ferramentas dá-se de forma bastante parecida. Nela são cadastrados os diferentes projetos a serem gerenciados. Para cada um dos projetos são definidas tarefas com prazos determinados ou alocados recursos. Mediante os lançamentos de dados de acordo com o andamento do projeto podem ser gerados gráficos ou relatórios para que o gerente de projeto passe a ter a noção exata em relação ao andamento do projeto, ou seja, se seus prazos estão sendo cumpridos, se os custos de desenvolvimento estão dentro do esperado entre outros.

Quando a equipe de desenvolvedores está trabalhando no mesmo projeto, é necessário que se tenha um controle sobre o desenvolvimento. Os códigos fonte ficam localizados nas máquinas de cada um dos desenvolvedores e muitas vezes há uma dependência entre diferentes partes do sistema. Um exemplo disso são sistemas que usam bibliotecas de funções. Uma biblioteca deste tipo é usada ao longo de todo o desenvolvimento do sistema.

Portanto, se a equipe conta com diferentes desenvolvedores e estes passam a fazer alterações nesta biblioteca, cada um em sua própria máquina, quando o projeto for finalizado, o trabalho para organizar estas alterações para gerar uma única que atenda todas as necessidades será bastante grande. Para resolver este problema, foram desenvolvidas ferramentas de sincronização de códigos, para que os desenvolvedores tenham sempre a noção exata das alterações realizadas em uma determinada parte do projeto.

O funcionamento destas ferramentas baseia-se em um repositório onde os arquivos de códigos são armazenados e daí podem ser distribuídos para diversas máquinas dentro da rede. Conforme os desenvolvedores vão fazendo alterações no código fonte a ferramenta encarrega-se de comparar a versão do código que está no repositório com a versão da máquina do desenvolvedor. Se o desenvolvedor

sincronizar o código fonte presente na sua máquina com o que está no repositório, é feita uma cópia da versão antiga e criada uma nova.

Se um segundo desenvolvedor resolver fazer o mesmo com o mesmo arquivo do código fonte, a ferramenta irá informar que existe já uma versão diferente daquela que ele estava trabalhando, oferecendo-lhe a possibilidade de comparar os dois códigos e também a de igualá-los, fazendo uma mescla entre as duas versões, criando uma nova e assim por diante

Neste contexto merecem destaque duas ferramentas, o *Git*<sup>30</sup> e o *Subversion*<sup>31</sup>. O funcionamento das duas ferramentas é basicamente o mesmo, descrito anteriormente. Através de suas opções, os desenvolvedores definem um repositório central, onde serão armazenados os históricos de atualizações de cada um dos projetos e a partir dele vão sendo atualizados de acordo com as implementações feitas no código fonte.

Ainda dentro da área de gerenciamento de projeto, a fase de testes é uma das mais importantes. Na fase de desenvolvimento, os desenvolvedores testam o programa de forma linear, ou seja, tem a tendência de seguir uma sequência que muitas vezes evita que o sistema apresente algum tipo de falha. Já na fase de testes, a função dos analistas é verificar se o sistema trata todas, ou pelo menos, a grande maioria das ocorrências que podem vir a causar algum tipo de falha no sistema.

Para tanto, é necessário que se disponha de ferramentas que façam este tipo de gerenciamento, de modo que os analistas de teste e até mesmo os usuários possam ajudar na documentação das falhas dos sistema para que os desenvolvedores possam corrigi-las assim que possível.

---

30 <http://git-scm.com/>

31 <http://subversion.apache.org/>

Para esta tarefa também podem ser citadas duas ferramentas, o *Mantis*<sup>32</sup> e o *Bugzilla*<sup>33</sup>. Ambas as ferramentas possuem características bastante parecidas. São ferramentas com interface *Web*, que portanto podem ser utilizadas independentemente da arquitetura de sistema que se esteja utilizando. Neles são cadastrados cada um dos sistemas que se está desenvolvendo e em seguida os lançamentos de falhas ocorridas durante a fase de testes ou durante a utilização do sistema.

A grande vantagem destes sistemas é a de que eles podem ser adaptados de acordo com as necessidades da empresa. As alterações nas funcionalidades existentes ou mesmo a implementação de outras por parte dos desenvolvedores são possíveis tendo em vista que o código fonte está disponível para ser baixado da internet e também é muito fácil de ser alterado.

## 4.9 Modelagem de Projetos

Modelagem de projetos consiste na elaboração de diagramas para que se apresente graficamente as diferentes partes de um projeto. Nesses diagramas são demonstradas todas as classes de objetos presentes no sistema, a sequência em que cada uma das funcionalidades do sistema será executadas, os casos de uso entre outros elementos do projeto.

A *UML (Unified Modeling Language* ou Linguagem de Modelagem Unificada) é uma linguagem visual utilizada para modelar sistemas computacionais por meio de paradigmas de orientação a objetos. A linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem de *software*, adotada

---

32 <http://www.mantisbt.org/>

33 <http://www.bugzilla.org/>

internacionalmente pela indústria de engenharia de *software*.

*UML* não é uma linguagem de programação e sim uma linguagem de modelagem, cujo objetivo é auxiliar os engenheiros a definir características do *software*, tais como: seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos, e até suas necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implantado. Todas as características são definidas por meio da *UML*, antes do *software* começar a ser realmente desenvolvido (GUEDES, 2005).

Os diagramas são desenvolvidos utilizando-se de uma ferramenta *CASE* (*Computer-Aided Software Engineering* ou Engenharia de *Software* Auxiliada por Computador). Uma ferramenta *CASE* é uma ferramenta destinada a auxiliar a engenharia de *software*, permitindo a modelagem das partes de um projeto, podendo acompanhá-lo em todo o seu ciclo de vida.

Dentre essas ferramentas de modelagem podemos citar o *ArgoUML*<sup>34</sup>, *Umbrello*<sup>35</sup> e o *Dia*<sup>36</sup>. Através da utilização de símbolos gráficos, ambas as ferramentas permitem a confecção de diagramas em linguagem *UML* para a modelagem das classes do sistema, casos de uso entre outros. Alguns dos diagramas que podem ser modelados por estas ferramentas e utilizados na análise de um projeto são diagrama de casos de uso, diagrama de classe, diagrama de combinação, diagrama de estados, entre outros. O *Dia* vai ainda um pouco além, permitindo a confecção de fluxogramas e diagramas de modelagem de banco de dados, como o diagrama entidade relacionamento (DER). A Figura 4.6 ilustra a tela principal do *Umbrello* e a Figura 4.7 apresenta o *ArgoUML*.

---

34 <http://argouml.tigris.org/>

35 <http://www.umbrello.org/>

36 <http://live.gnome.org/Dia>



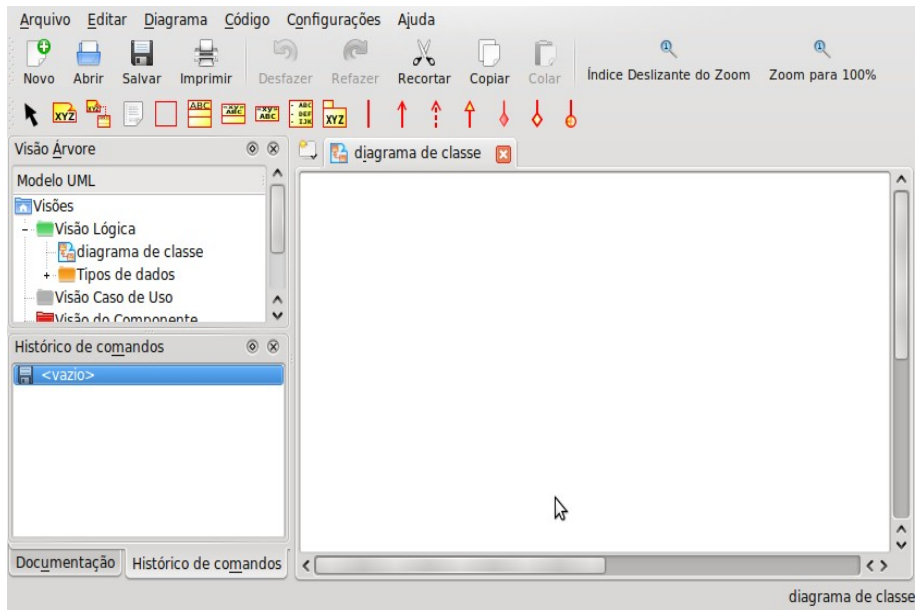


Figura 4.6: Tela principal do *Umbrello*

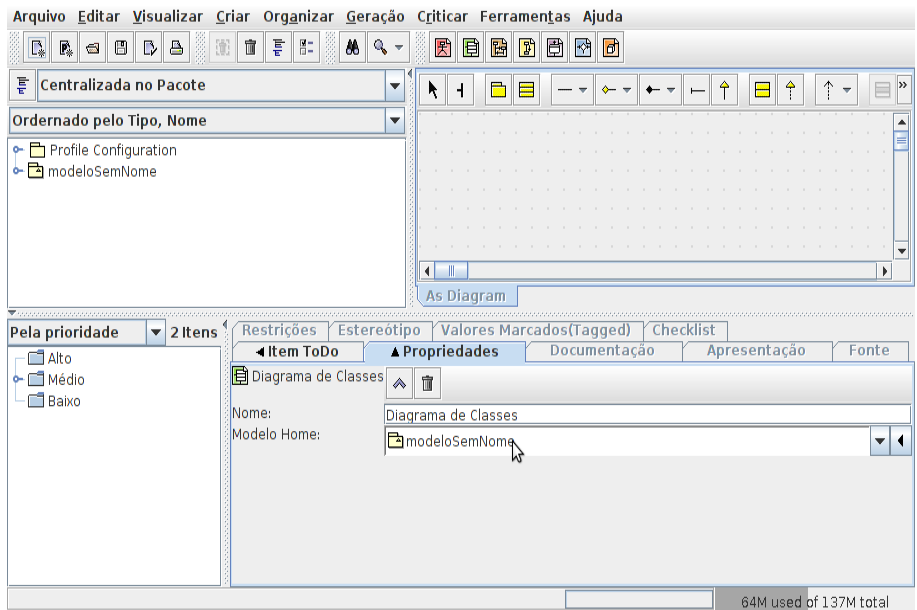


Figura 4.7: Tela principal do *ArgoUML*

## 4.10 IDE

No princípio, e até hoje pode ser assim, muitos desenvolvedores utilizavam editores de texto puro para construir a tela que seria apresentada ao usuário. O posicionamento dos campos de entrada de dados, tamanho de tais campos entre outras características eram definidos linha a linha do código fonte por parte dos desenvolvedores. Isso geralmente resultava em um trabalho exaustivo, muitas vezes desestimulante para o desenvolvedor e a interface muitas vezes não era atraente para o usuário.

Hoje em dia existem ferramentas com o objetivo de agilizar o desenvolvimento de aplicativos, as chamadas *IDEs*. (*Integrated Development Environment* ou em português Ambiente Integrado de Desenvolvimento). Editores de código fonte, compiladores, depuradores de código, geradores de código são exemplos de ferramentas presentes nas IDEs.

Nesta sessão serão apresentadas algumas ferramentas de grande utilidade na construção de um *software*. Para esta tarefa foram escolhidas três ferramentas, o *NetBeans*<sup>37</sup>, o *Eclipse*<sup>38</sup> e o *Bluefish*<sup>39</sup>.

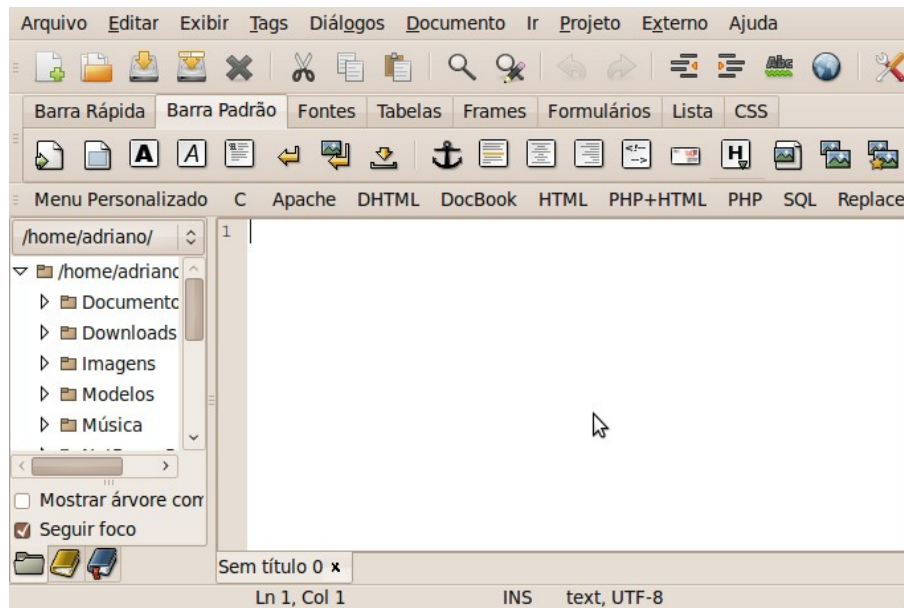
No caso do *Bluefish*, apresentado na Figura 4.8, possui funções de abas onde o usuário seleciona o objeto que quer na página, e ele cria as tags (sessões presentes no código *HTML*) apenas para que sejam preenchidas. Esses objetos podem ser: tabelas, botões, formulários, entre outros. Outra função interessante é a edição de folhas de estilo para facilitar a agilizar o desenvolvimento da interface da página e padronizá-la.

---

37 <http://netbeans.org/>

38 <http://www.eclipse.org/org/>

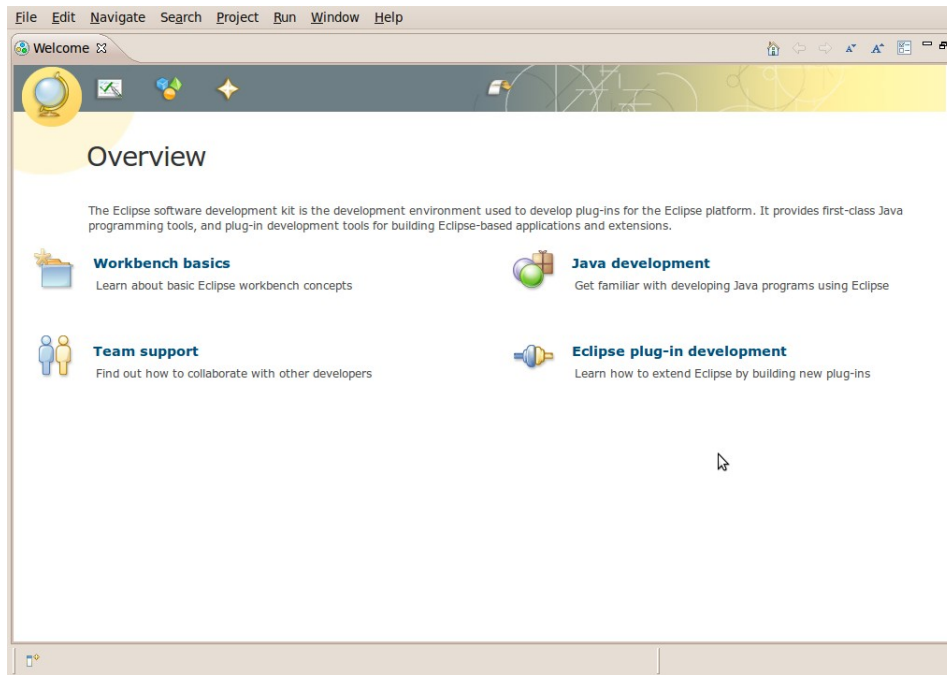
39 <http://bluefish.openoffice.nl/>



**Figura 4.8:** Tela principal do *Bluefish*

Já o *NetBeans* tem as mesmas funções apresentadas no *Bluefish*, mas com a vantagem de ajudar a desenvolver *Applets*, mini aplicativos que rodam dentro das páginas por meio da máquina virtual *Java* instalada no cliente. Este recurso tem como vantagem agregar funcionalidades à página desenvolvida e em virtude dessas *Applets* terem um tamanho reduzido, agilizam a operação do aplicativo.

Por fim, o *Eclipse* é um ambiente de desenvolvimento de *software* integrado, multi-linguagem que pode ter suas funcionalidades expandidas através da utilização de *plug-ins*. O *Eclipse* foi desenvolvido originalmente para criação de programas em linguagem *Java*, mas também pode ser utilizado no desenvolvimento de programas em linguagem *PHP*, *Python*, entre outras. De acordo com a linguagem de programação utilizada, a interface de desenvolvimento do *Eclipse* pode ter denominações diferentes. Por exemplo, a interface de desenvolvimento para



**Figura 4.9:** Tela principal do *Eclipse*

linguagem *Java* é chamada de *Eclipse JDT*, para linguagem *PHP* é chamada de *Eclipse PDT*, entre outras. Na Figura 4.9 é apresentada a imagem da tela principal do *Eclipse*.

A partir de agora passa-se à especificação e instalação de aplicativos e gerenciadores para que esta estação de trabalho passe a desempenhar as funções para ele destinadas. No capítulo a seguir são apresentadas algumas das ferramentas utilizadas no projeto e desenvolvimento de aplicações internet.

## 5 INSTALAÇÃO DE APLICATIVOS

### 5.1 Considerações Iniciais

Definidas as configurações e permissões dos usuários do computador, segue agora o processo de instalação dos aplicativos utilizados. As aplicações em questão devem atender à necessidade dos usuários no desenvolvimento de interfaces de páginas internet e aplicações para esta plataforma, tais como, servidor *Web*, ferramentas de gerenciamento e modelagem de projeto. Ainda, há a necessidade de um gerenciador e ferramentas de construção e modelagem de banco de dados, linguagem de programação e *IDEs* de desenvolvimento.

Deve-se ainda, instalar ferramentas de edição de imagens e animações, para possibilitar que os desenvolvedores incrementem a interface dos projetos desenvolvidos por eles. Isso possibilita o desenvolvimento de projetos mais avançados e utilizando as tecnologias mais atuais de editoração de imagens.

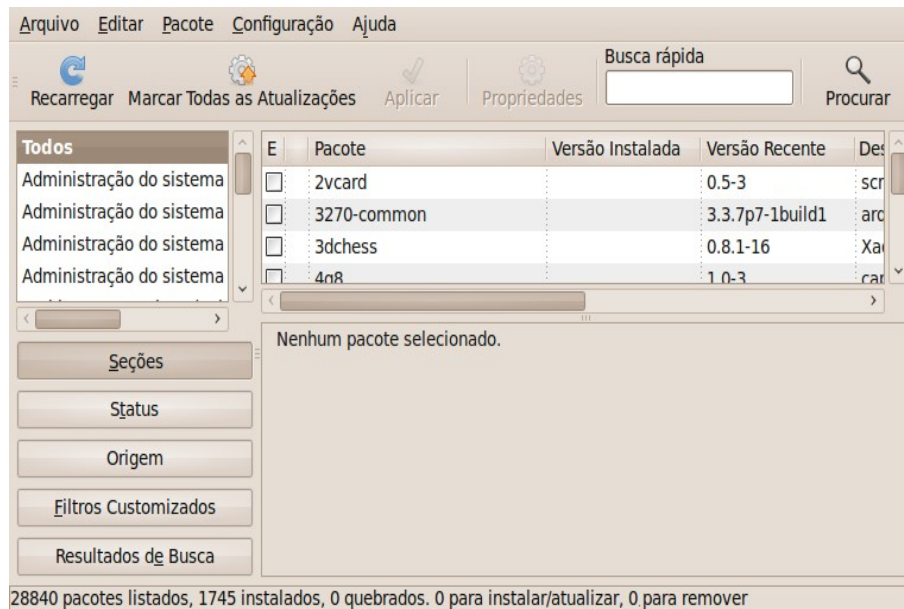
Empresas que realizam serviços de desenvolvimento de sistemas, geralmente definem sua plataforma e ferramentas de desenvolvimento no princípio de suas atividades e geralmente não são alteradas com o tempo. No caso específico deste trabalho, esta definição leva em conta o desenvolvimento de sistemas para plataforma internet com a utilização de ferramentas livres, ou seja, não há a utilização de programas com licenças proprietárias.

Para a instalação dos gerenciadores e ferramentas utilizadas, a opção é a utilização do Gerenciador de Pacotes *Synaptic*<sup>40</sup>, visto que este gerenciador opta

---

40 <http://wiki.ubuntu-br.org/Synaptic>

sempre pela instalação das versões mais recentes disponíveis. Faz também toda a verificação de instalação de componentes necessários ao funcionamento das aplicações (dependências), facilitando bastante o trabalho. O gerenciador de pacotes Synaptic é apresentado na Figura 5.1.



**Figura 5.1:** Gerenciador de pacotes *Synaptic*

Em alguns casos a instalação também poderá ser feita através de outro gerenciador de pacotes que é o *apt*<sup>41</sup>. Na realidade, os dois gerenciadores trabalham em conjunto, mas como o *Synaptic* oferece uma interface gráfica com o usuário, se constituindo como a opção preferencial.

<sup>41</sup> <https://help.ubuntu.com/8.04/serverguide/C/apt-get.html>

## 5.2 Critérios para Escolha das Ferramentas

Quando se inicia o projeto de desenvolvimento do *software*, é necessário que sejam cumpridas algumas etapas. Fases de análise, modelagem, desenvolvimento e testes são algumas das etapas do ciclo de vida de um *software*. Para cada uma das fases do ciclo de vida são aplicadas a utilização de ferramentas de construção e gerenciamento.

A escolha destas ferramentas deve seguir alguns critérios definidos pela equipe de desenvolvedores, levando em consideração a arquitetura e recursos de *hardware*, sistemas operacionais disponíveis, plataforma sobre a qual o sistema será executado, domínio do uso das diferentes ferramentas e a opinião pessoal dos membros da equipe de desenvolvimento. Este último critério citado talvez seja o que mais apresente polêmicas, uma vez que uma ferramenta considerada adequada por um membro da equipe, pode não ser para outro.

Nesta sessão serão apresentados alguns critérios que servirão de base para justificar a escolha de uma ferramenta em detrimento de outra. Os critérios são:

- Adequação ao projeto: Neste caso as ferramentas escolhidas devem atender a pelo menos uma necessidade do projeto, ou seja, devem satisfazer pelo menos a uma condição do seu desenvolvimento. Não adianta a escolha de uma linguagem de programação que não ofereça recursos de desenvolvimento para plataforma *Web* ou que o software construído a partir dela não possa ser executado no sistema operacional disponibilizado.
- Análise de mercado: A quantidade de usuários que utilizam uma determinada ferramenta diz muito sobre ela. Quanto mais usuários uma ferramenta possuir, maior a possibilidade de troca de experiências em relação a proble-

mas de utilização, limitações, entre outros. Como neste trabalho o foco é a utilização de *software* livre, a facilidade de se encontrar soluções para dúvidas de utilização e constantes atualizações em suas comunidades de desenvolvimento também são bastante importantes.

- Facilidade de instalação: Quanto mais fácil a instalação da ferramenta, melhor para os desenvolvedores. Isso faz com que não se perca tempo com pesquisas e tentativas de instalação de ferramentas que apresentam um grande número de dependência, ou ainda que se tenha a necessidade de compilar a ferramenta para ser usada no sistema.
- Opinião pessoal dos desenvolvedores: Deve ser levada em conta, mas não pode se sobrepor a outros critérios mais importantes. A opinião pessoal de um desenvolvedor ou seu conhecimento na utilização de uma determinada ferramenta podem contribuir para a agilidade na condução do processo de desenvolvimento de um *software*. Porém, se uma ferramenta apresenta maiores recursos e facilidades ao desenvolvimento do sistema, do que outra mais apreciada pela equipe de desenvolvimento, a ferramenta que melhor contribuir ao processo de desenvolvimento deve ser preferida.
- Consumo de recursos: Uma ferramenta que possua um maior consumo de recursos de sistema, que tenha opção de ser substituída por outra que tenha desempenho melhor sem comprometer a qualidade do desenvolvimento do *software*, deve ter sua utilização questionada dentro do projeto. Deve-se avaliar a quantidade de recursos disponíveis e se sua utilização é primordial ao sucesso do projeto.
- Integração entre as diferentes ferramentas: A possibilidade de as diferentes ferramentas trabalharem em conjunto para facilitar e agilizar o processo de



desenvolvimento é bastante interessante. Este é o caso de *frameworks* que tem como função estender funcionalidades de determinadas *IDEs* de desenvolvimento.

Vale a pena destacar que o estudo apresentado neste trabalho se baseia em uma empresa de desenvolvimento de *software* de pequeno porte, contando com uma equipe constituída por um gerente de projeto, um analista de sistemas, dois técnicos de suporte e quatro programadores. Sendo uma empresa de pequeno porte, situada em uma cidade do interior da região sul do estado de Minas Gerais, atende a um mercado bastante restrito, desenvolvendo projetos de pequeno e médio porte. Uma questão primordial para esta empresa é custo de desenvolvimento de um sistema. A utilização de ferramentas livres contribui para que não se tenha despesas com aquisição de ferramentas de desenvolvimento. Por outro lado, a escolha das ferramentas deve levar em conta que a empresa não dispõe de grandes recursos de *hardware*, portanto as ferramentas escolhidas devem preferencialmente requisitar uma quantidade menor de recursos, para que a empresa não tenha que arcar com aquisição de um *hardware* com maior poder de processamento.

### **5.3 Preparando para a Instalação**

Antes de iniciar a instalação dos pacotes, será necessária a atualização dos pacotes do *apt-get*, para atualizar a lista de pacotes disponíveis para *download*. Esta atualização se faz necessário para que o *apt* atualize a lista de repositórios disponíveis, quais pacotes estão disponíveis em cada um e qual a sua versão. Com isso, fica mais fácil identificar onde buscar as dependências necessárias à instalação de um determinado programa e, caso aquela dependência já se encontre instalada, verificar se não há a necessidade de atualização.

Um outro serviço instalado foi o *OpenSSH*<sup>42</sup>, para que o responsável pelo gerenciamento dos sistemas possa realizar suas atividades remotamente, sem a necessidade do administrador estar fisicamente presente onde está instalado o computador. Isso facilita suas atividades, visto que pode realizar as tarefas de instalação, configuração de aplicativos e gerenciamento do sistema de onde ele estiver, poupando assim tempo e esforço.

Primeiramente serão instalados os programas servidores e os ambientes de desenvolvimento. Em seguida as ferramentas de gerenciamento e operação de tais programas servidores e por fim as ferramentas de desenvolvimento utilizadas pelos usuários da estação em questão.

## 5.4 Modelagem e Gerenciamento de Projeto

Durante a concepção de um projeto, definem-se os diferentes personagens que atuarão em todo o seu ciclo de vida. Gerente de projeto, analistas, desenvolvedores, usuários são alguns exemplos. Dentre estes, a figura mais importante talvez seja a do gerente de projeto. É ele quem define os rumos que o projeto irá tomar, e está sob sua responsabilidade o sucesso ou o fracasso de um projeto.

Para auxiliar no gerenciamento ou na tomada de decisões sobre as diferentes etapas do projeto, o gerente de projeto pode fazer uso de ferramentas que irão auxiliá-lo na definição de tarefas a serem executadas dentro do projeto. Dentro destas tarefas incluem-se os prazos de execução de cada uma delas, o dimensionamento de recursos e material humano para execução entre outros. No final, o

---

42 <http://www.openssh.org/>

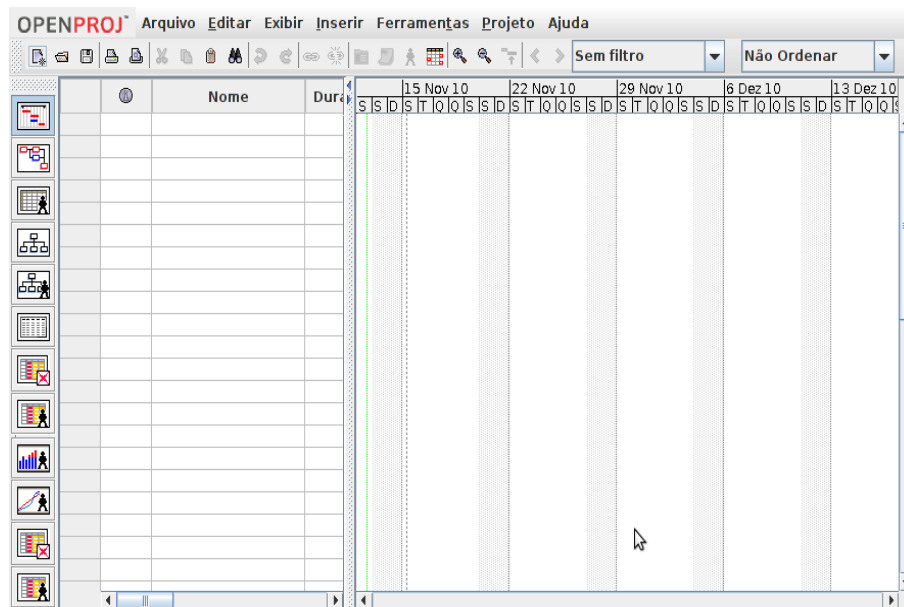
gerente terá condições de avaliar o desempenho de toda a equipe de projeto, bem como saber se o seu desenvolvimento está acontecendo conforme o planejado.

Para o ambiente de desenvolvimento que está sendo criado dentro da execução deste trabalho, há a necessidade de que este ambiente possua uma ferramenta para gerenciamento de projetos. Para esta tarefa foi escolhido o *OpenProj*. A opção pelo *OpenProj* foi feita analisando-se a quantidade de recursos oferecidos. Possui uma grande quantidade de gráficos para o acompanhamento das atividades do projeto. Uma função interessante que poderia ser disponibilizada pelo *OpenProj*, que existe implementada no *Planner*, por exemplo, é a customização de campos no projeto, o que possibilitaria especificar valores para as tarefas e recursos alocados ao projeto.

O processo de instalação desta ferramenta difere um pouco das demais que foram instaladas. Na versão da distribuição Linux utilizada para a criação do ambiente de desenvolvimento, o *OpenProj* não se encontra disponível nos repositórios do gerenciador de pacotes *Synaptic*. O que não significa que sua instalação seja complicada. Está disponível na internet um pacote de instalação para a distribuição que está sendo utilizado. Basta apenas baixar este pacote e clicar duas vezes no arquivo para que o instalador de pacotes faça todo o processo de instalação. A Figura 5.2 apresenta a imagem do *OpenProj*.

O processo de desenvolvimento de um sistema passa primeiramente por sua fase de análise. Nesta fase são definidas os casos de uso de um sistema, as classes de objetos a serem implementadas, entre outras. Para isso é necessária a presença de ferramentas que tenham a capacidade de criar os diagramas que servirão de base para que os desenvolvedores possam visualizar a análise do projeto graficamente e assim definir as estratégias de desenvolvimento.

Para esta função a ferramenta escolhida foi o *Dia*, visto que tem a capacidade de gerar tanto os diagramas de fluxo de dados, os diagramas de modelagem XML das classes e casos de uso do sistema, e ainda a modelagem entidade relacionamento do banco de dados. A tela principal da ferramenta Dia está representada na Figura 5.3



**Figura 5.2:** Tela principal do *OpenProj*

Definida a análise do projeto, passa-se à fase de codificação do sistema. As ferramentas de edição de código serão definidas mais adiante, mas é necessário que se defina a ferramenta responsável pela sincronização do código elaborado pelos desenvolvedores. As ferramentas pesquisadas mostraram-se muito iguais em seu funcionamento, em virtude disso foi observado a complexidade de sintaxe dos comandos de sincronização a serem executados e a ferramenta escolhida foi o *Subversion*. Um outro ponto observado a favor da utilização do *Subversion* foi a

quantidade de usuários utilizando o sistema. Existe uma grande quantidade de fóruns na internet de com questões sobre a utilização do *Subversion* no dia a dia das empresas. Na região onde a empresa que serve de base para o estudo deste trabalho está situada, existem outras empresas de mesmo ramo de atividade utilizando o *Subversion*.

Quando estiver terminando a fase de codificação do sistema, será chegada a hora de realizar os testes no sistema. Para tanto torna-se necessária a presença de uma ferramenta de gerenciamento de *Bugs* (erros presentes na codificação do sistema), isso permite aos analistas de teste relatar os erros na hora em que ocorrerem, definindo níveis de prioridade sem a necessidade de reportá-los diretamente aos programadores. Para isso, a ferramenta *Mantis* foi a escolhida por possuir uma interface mais simples e amigável e por ser bastante configurável, inclusive no que diz respeito a biblioteca de idiomas.

O processo de instalação usando o gerenciador de pacotes *Synaptic* é bastante simples. Os pacotes de instalação possuem os mesmos nomes das ferramentas, portanto bastou apenas informá-los para executar o processo. No caso do *Mantis* e do *Subversion*, houve apenas a necessidade de configurar um usuário primário definindo sua senha. As demais configurações como repositórios de dados e demais contas de usuários ficarão a cargo da equipe de desenvolvimento.

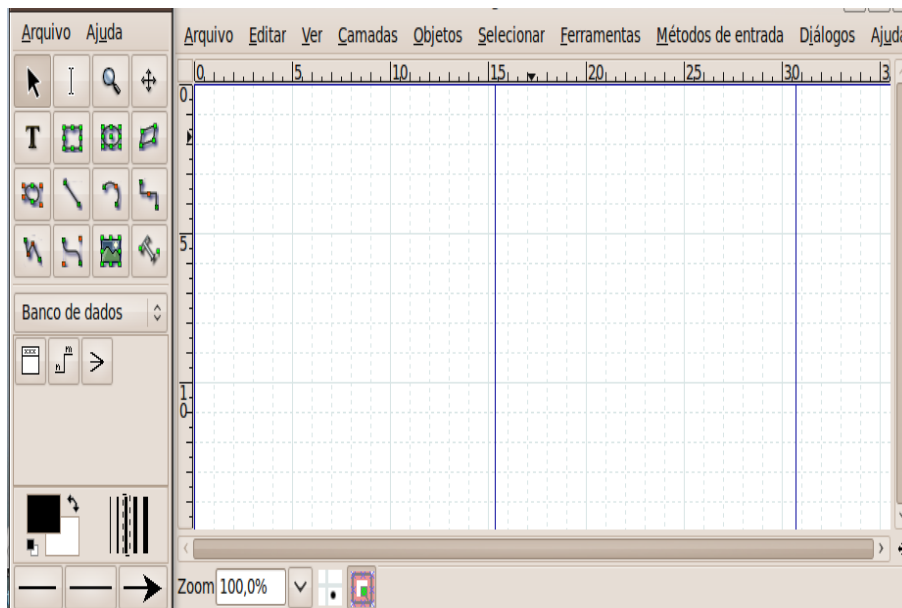


Figura 5.3: Tela principal do *Dia*

## 5.5 Servidores e Ambientes de Desenvolvimento

A instalação de aplicações tornou-se uma tarefa relativamente simples, com gerenciadores de pacotes como o *Synaptic* e o *apt-get*. Não há mais a necessidade de baixar os códigos fonte e compilá-los, de modo que basta apenas saber o nome do pacote que se quer instalar, para que ele seja baixado e instalado diretamente por esses gerenciadores.

No caso das aplicações que vão ser instaladas no computador em questão, algumas delas como o *PHP* por exemplo, possuem configurações específicas que controlam seu funcionamento, como os parâmetros de variáveis públicas, sendo que

alguns usuários preferem habilitar esses parâmetros e outros não. Em virtude disto, as configurações específicas destes programas não serão tratados, apenas os processos de instalação e configurações para sua inicialização e funcionamento.

As aplicações inicialmente instaladas serão o servidor *Web Apache* o ambiente *PHP* para desenvolvimento das aplicações, juntamente com o ambiente de desenvolvimento *Java* e os gerenciadores de bancos de dados *MySQL* para o desenvolvimento de aplicações de pequeno porte, sem grandes recursos, e o *PostgreSQL*, caso seja desenvolvida uma aplicação de maior porte, que exija maiores recursos de banco de dados. Complementando esta primeira parte, serão instaladas ferramentas para gerenciamento de bancos de dados, que são o *MySQL Administrator* e o *phpMyAdmin* que são ferramentas mais comuns para o gerenciamento e manipulação de dados do *MySQL*.

A escolha do gerenciador de banco de dados foi a mais complicada. Os gerenciadores *MySQL* e *PostgreSQL* apresentaram características que os colocaram como opção. Mas, a grande quantidade de usuários utilizando *MySQL* em conjunto com *Apache*, *PHP* e *Linux* foi o fator que definiu a sua escolha. Na internet, inclusive, este ambiente de desenvolvimento é denominado de LAMP, sigla formada pelas iniciais de cada uma das ferramentas (*Linux*, *Apache*, *MySQL* e *PHP*). Porém, toda empresa visa crescimento, e a expansão do mercado de desenvolvimento de aplicações para internet é visível, o que pode levar a empresa a expandir suas atividades e atacar o mercado de desenvolvimento de aplicações de grande porte. Para isso, há necessidade de um gerenciador de bancos de dados mais poderoso. Por isso, na necessidade de desenvolvimento de aplicações de grande porte será adotada a utilização do *PostgreSQL*.

O *MySQL* e o *PostgreSQL* são gerenciadores de bancos de dados com interface em modo texto, ou seja, sua operação é feita através de linhas de comando. Por esse motivo foram instaladas duas ferramentas gráficas para auxiliar o processo de confecção e gerenciamento do banco de dados. A primeira delas é o *MySQL Administrator*, uma ferramenta gráfica que se destina ao gerenciamento do servidor *MySQL* como um todo. Desenvolvida pela própria empresa do gerenciador de bancos de dados. Com ele é possível a administração total do banco de dados, exceto dos dados armazenados nele, ou seja, ele não tem a capacidade de visualizar os dados armazenados no banco, mas apenas sua estrutura. Em virtude disso, optou-se pela instalação do *phpMyAdmin*, que também tem funções de gerenciamento e construção de bancos de dados, como o *MySQL Administrator*, com a vantagem de rodar em plataforma internet e ter a capacidade de visualizar os dados armazenados nas tabelas.

Para a instalação foram necessários o pacote *phpMyAdmin* e suas dependências, o pacote *mysqserver5* e o pacote *mysqladmin*. Efetuado o *download* dos mesmos, foram necessárias apenas duas configurações. Para o *MySQL* a definição da senha do usuário *root*, o usuário que possui todos os poderes de administração do gerenciador, e para o *phpMyAdmin*, qual o servidor *Web* a ser utilizado, neste caso o *Apache2*. A Figura 5.4 mostra a tela inicial do *phpMyAdmin*.

Como houve a opção pela utilização também do *PostgreSQL*, passa-se a ter a necessidade de instalação de suas ferramentas gráficas de administração assim como foi feito com o *MySQL*. Para isso foram instalados além do pacote *PostgreSQL* que corresponde à instalação do gerenciador de banco de dados, os pacotes *pgadmin* e *phpPgadmin* para a administração dos bancos de dados criados. Juntamente com estes pacotes foram instaladas suas respectivas dependências.



A opção pelo *Apache* se deu pelo fato de que ele é um servidor *Web* extremamente configurável, robusto e de alta performance. Tem como características principais o fato de ter seu código fonte disponível na internet, o que

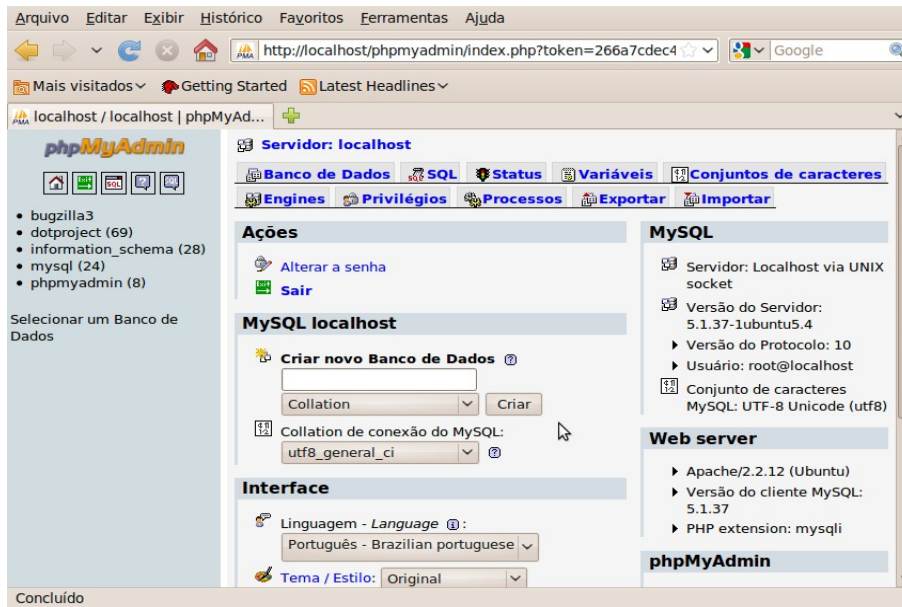


Figura 5.4: Tela inicial do *phpMyAdmin*

o constitui como o servidor *Web* mais utilizado na internet com um número de usuários maior do que a soma de todos os usuários de outros servidores (SILVA, 2007). A instalação do *Apache* é relativamente simples, utilizando o *Synaptic*, basta selecionar os pacotes desejados, no caso o pacote e *apache2* e o *lib-apache-php* para interligar o servidor *Web* com o *PHP*. Os pacotes e suas dependências foram baixadas e instaladas sem problemas.

Em seguida foi a vez da linguagem de programação *PHP*, que se constitui hoje numa das mais utilizadas linguagens para desenvolvimento de aplicações internet. É uma linguagem de código fonte aberto assim como o *Apache*, que tem

como principal característica ser incorporada ao código *HTML*, a linguagem padrão para desenvolvimento de páginas de internet.

Para desenvolvimento em *Java* é utilizado o pacote *OpenJDK*, que são objetos gráficos para desenvolvimento de interfaces, não sendo necessárias configurações específicas para o pacote. O único procedimento foi o de seleção no *Synaptic* que efetuou o *download* do *OpenJDK* e realizou o processo de instalação automaticamente.

Configurações e ajustes para utilização de funcionalidades dos mesmos serão feitas pelo usuário da estação. Pode-se ainda realizar a instalação de *frameworks* para estender a funcionalidade das linguagens de programação utilizadas. O *framework* em questão é o *CackePHP*. A instalação dele é extremamente simples bastando informar o seu nome no gerenciador de pacotes e ele irá baixar e instalar o pacote. Outros *frameworks* podem e devem ser necessários ao desenvolvimento de aplicativos, mas a tarefa de escolher qual o mais adequado se necessário ficará a cargo da equipe de desenvolvimento.

## 5.6 Ferramentas de Desenvolvimento

Finalizando o processo de instalação faltam apenas a instalação das ferramentas de desenvolvimentos para as linguagens de programação. Para esta função foram escolhidos o Bluefish, Geany e NetBeans, para edição de código HTML, PHP e Java.

Para executar o procedimento de instalação das duas ferramentas, bastou apenas informar o nome de seus pacotes no *Synaptic*, *netbeans* e *bluefish* respectivamente para executar seu processo de instalação automaticamente. A

função do *Netbeans* é auxiliar no desenvolvimento de interfaces e edição de códigos das aplicações que demandem uso da linguagem de programação *Java*. Já a do *Bluefish* é a criação da interface das páginas de código *HTML*.

Ainda faz-se necessária a instalação de um editor de códigos para o *PHP*. Para esta tarefa foi escolhido o *Geany*, que dentre as ferramentas mencionadas no capítulo anterior, foi a que apresentou um melhor comportamento no que diz respeito a consumo de recurso. A utilização para a qual o *Geany* se destina, não demanda integração com compiladores ou outros recursos externos, visto que será utilizado principalmente para edição de código *PHP* e *HTML*, o que demanda apenas a funcionalidade de verificação de erros de sintaxe de comandos.

Instaladas as ferramentas para desenvolvimento de código, seria interessante também a instalação de uma ferramenta de editoração gráfica, para tratamento de imagens a serem inseridas nos projetos. Para esta tarefa foi escolhido o *Gimp*<sup>43</sup>, ilustrado Figura 5.5. No caso do *Gimp*, ele já vem instalado por padrão na distribuição Ubuntu, não sendo necessário nenhum processo adicional. O *Gimp* se constituiu como uma alternativa livre a soluções proprietárias, tornando-se bastante popular entre os membros da comunidade de software livre.

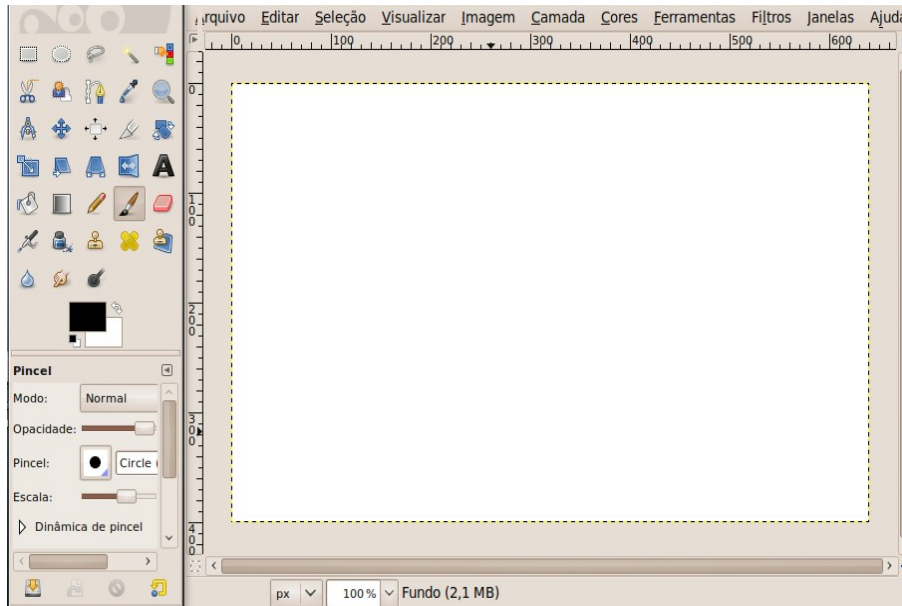
Uma outra ferramenta interessante para este ambiente seria um cliente FTP. Aplicativos de FTP (File Transfer Protocol ou Protocolo de Transferência de Arquivos) em modo texto são comuns no Linux, mas, como a opção é por aplicativos gráficos, foram realizadas algumas pesquisas em fóruns na internet, e o *filezilla*<sup>44</sup> é referência constante nestes fóruns, o que constituiu um fator determinante à sua adoção. Com isso, há a possibilidade de *upload* de arquivos para servidores externos ou outras máquina internas à rede, implementando uma

---

43 <http://www.gimp.org/>

44 <http://filezilla-project.org/>

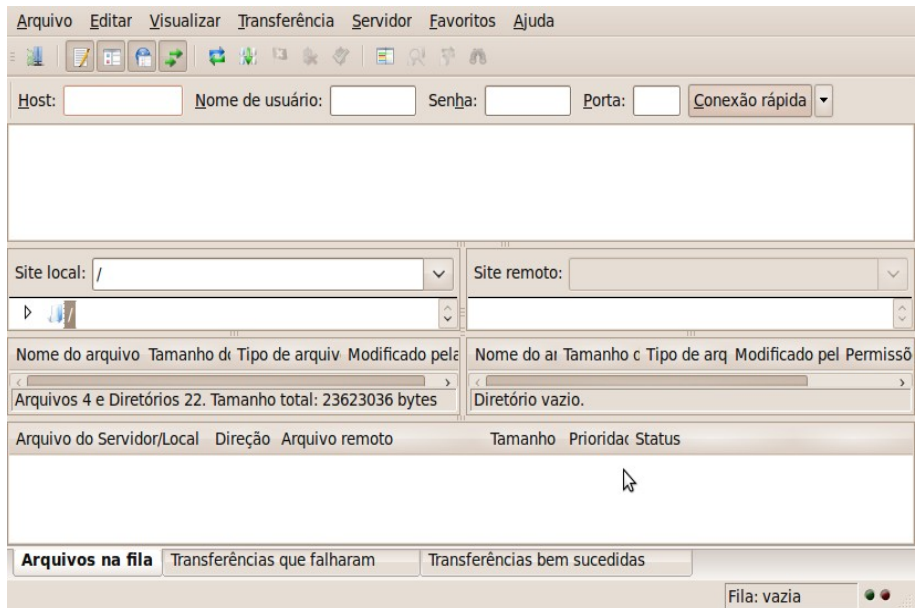
funcionalidade a mais ao computador em questão. A Figura 5.6 ilustra a tela principal do Filezilla



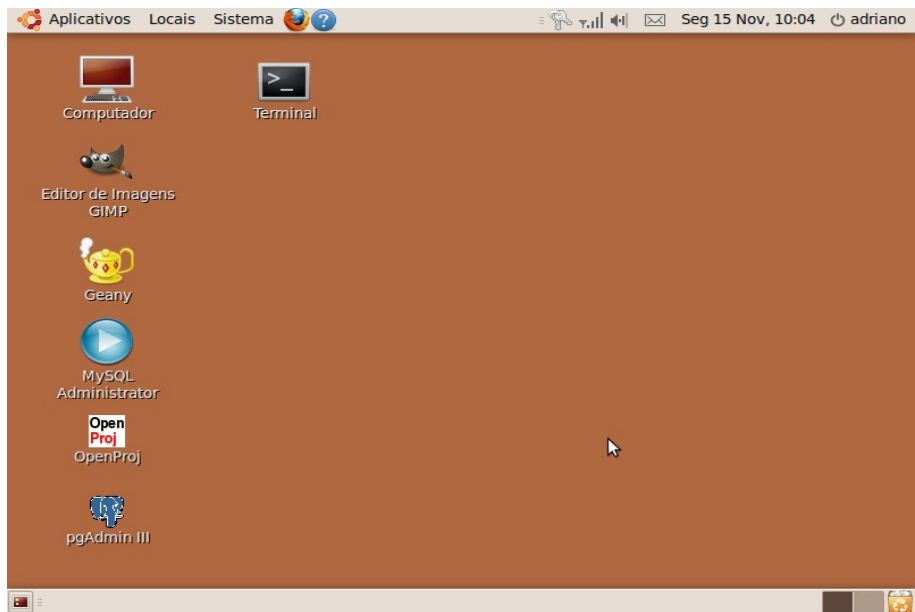
**Figura 5.5:** Editor de imagens *Gimp*

Com isso é terminada a fase de configuração e instalação sendo necessária agora uma conferência no computador de modo a verificar se as aplicações estão funcionando corretamente e dar por terminada a configuração. Criados os atalhos na área de trabalho, o processo de instalação estará terminado. A Figura 5.7 mostra como ficou a área de trabalho com o final das configurações, já com os atalhos dos programas configurados

Com o final das configurações e com as conferências realizadas, segue agora a descrição das conclusões obtidas com o desenvolver do trabalho.



**Figura 5.6:** Cliente *FTP Filezilla*



**Figura 5.7:** Área de trabalho do usuário

## 6 CONCLUSÃO

A utilização de ambientes de desenvolvimento baseados em *software* livre só oferece vantagens às empresas de desenvolvimento de *software*. Se fosse levada em conta apenas a questão financeira, a utilização de *software* livre já constituiria uma vantagem competitiva sobre ferramentas de desenvolvimento com licença proprietária.

Porém, além da relação custo benefício, muitas ferramentas livres oferecem funcionalidades que não estão presentes em soluções proprietárias. Um exemplo que pode ser citado é a capacidade que a linguagem de programação *Java* tem de gerar aplicativos que podem ser executados em diferentes plataformas de sistema operacional. Isso possibilita o desenvolvimento de um sistema sem a preocupação de qual sistema operacional o cliente adota, se é Linux ou não.

Existe disponível uma infinidade de aplicativos para que se construa um ambiente de desenvolvimento utilizando apenas ferramentas livres, contemplando todas as fases de desenvolvimento de um *software*. Cabe à equipe de desenvolvimento estabelecer critérios para a adoção destas ferramentas de acordo com os recursos disponíveis e também analisando as necessidades do projeto. A escolha de uma ferramenta em detrimento de outra deve levar em conta os benefícios que ela trará ao desenvolvimento do projeto de acordo com os critérios de avaliação.

Um outro ponto que contribui para a implantação de ambientes de desenvolvimento usando Linux e *software* livre é a facilidade de instalação de aplicativos dentro da distribuição Linux. Antigamente perdia-se muito tempo preparando um ambiente Linux em virtude da complexidade de instalar aplicativos que não vinham

instalados na distribuição. A necessidade de solucionar problemas com falta ou conflito de dependências acabava por desestimular a sua utilização.

Com as distribuições Linux mais recentes, isso é coisa do passado. Para que se faça a customização necessária para que o ambiente atenda as necessidades do usuário, basta que se saiba apenas o que se quer instalar e a distribuição através de seu gerenciador de pacotes padrão faz todo o trabalho, sem que usuário tenha que compilar programas, instalar pacotes de dependência, entre outras.

Devido a facilidade de instalação e utilização das ferramentas analisadas, conclui-se que a criação de ambientes de desenvolvimento baseados em Linux e *software* livre é perfeitamente viável. Tanto o sistema operacional quanto as ferramentas pesquisadas oferecem uma base de conhecimento bastante extensa, seja pela documentação disponível nas páginas dos projetos das ferramentas, seja por meio de fóruns e *wikis* que mostram as experiências dos usuários com a utilização das ferramentas.

Como trabalhos futuros, o que se propõe é o aprimoramento do ambiente de desenvolvimento através da pesquisa em implantação de ferramentas para desenvolvimento de aplicações mais voltadas para a área de multimídia. Também a pesquisa de novos *frameworks* para estender ainda mais o poder de desenvolvimento das ferramentas já utilizadas. Fora do escopo do trabalho apresentado, pode-se propor também trabalhos paralelos na área de gerenciamento de sistemas Linux voltados para a área de controle de usuários e segurança da informação.

## REFERÊNCIAS BIBLIOGRÁFICAS

COSTA, E. **O Ubuntu abraça a nuvem**. Info Exame. 286: 12(2009) 75

DATE, C. J. **Introdução a sistemas de banco de dados**. 5 ed. Rio de Janeiro: Elsevier, 2003.

DOMINGUES, M. A.; SCHNEIDER, B. de O.; UCHÔA, J. Q. **Autenticação em sistemas Linux usando OpenLDAP**. In: Semac2001 - XII Semana da Computação - IV Workshopem Linux, Internet e Aplicações. São José do Rio Preto: UNESP, 2001. URL: <http://www.ginux.ufla.br/~joukim/extensao/semac.pdf>.

GONÇALVES, Edson. **Dominando o NetBeans**. Rio de Janeiro: Ciência Moderna, 2006.

GUEDES, Gilleanes T. A. **UML 2: guia de consulta rápida**. 2 ed. São Paulo: Novatec, 2005.

HUNTER, Jason; CRAUFORD, Willian. **JAVA servlet: programação**. Rio de Janeiro: Ciência Moderna, 2002.

JORDÃO, Leonardo de Brito. **Uso do Protocolo de Autenticação Kerberos em Redes Linux**: Lavras, 2005. URL: <http://www.ginux.ufla.br/files/mono-LeonardoJordao.pdf>.

MORGAN, A. G. **The Linux PAM System Administrators' Guide**; Draft v0.76. [S.l.]:Linux-PAM, 2002. URL: <http://www.us.kernel.org/pub/linux/libs/pam/>.



MOTA, Adriano Pinheiro. **Integrando LDAP com SAMBA para Utilização como Solução de PDC na Rede**: Lavras, 2008. URL: <http://www.ginux.ufla.br/files/mono-AdrianoMota.pdf>

SILVA, Gleydson Mazioli. **Gua Foca GNU/Linux - Apache. Foca Linux**. 2007. [Online; acessado em 2 de junho de 2010] URL: <http://focalinux.cipsga.org.br/download/avancado/focalinux3-pdf.zip>

UCHÔA, J. Q. **Gerenciamento de Sistemas Linux. 3. ed.** Lavras: UFLA/FAEPE, 2007. (Curso de Pós Graduação “Lato Sensu” (Especialização) a Distância em Administração em RedesLinux).

WEB2.0. **Conceituando o que é web 2.0**: 2007. URL: <http://web2.0br.com.br/conceito-web20>.

WIRZENIUS, L. et al. **The Linux System Administrator’s Guide**. Version 0.9. Chapel Hill (North Carolina): The Linux Documentation Project, 2005. URL: <http://www.tldp.org/guides.html>.