

**HERMANO PEREIRA**

**GRADENET  
AMBIENTE VIRTUAL DE  
APOIO AO ENSINO**

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

**Orientador:**

**Prof. MSc. Joaquim Quinteiro Uchôa**

Lavras  
Minas Gerais - Brasil  
2004

**HERMANO PEREIRA**

**GRADENET  
AMBIENTE VIRTUAL DE  
APOIO AO ENSINO**

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

Aprovada em 25 de abril de 2004.

---

Prof. Samuel Pereira Dias

---

Prof. MSc. Fernando Cortez Sica

---

Prof. MSc. Joaquim Quinteiro Uchôa  
(Orientador)

Lavras  
Minas Gerais - Brasil  
2004

Aos meus pais, minha irmã e minha noiva.

## **AGRADECIMENTOS**

Aos alunos e professores do Curso de Análise de Sistemas da Unicentro que me motivaram neste trabalho.

À Professora Josiane pela amizade e pela força nos momentos difíceis.

À Professora Inali pela amizade e incentivo.

Ao Professor Marcos pela amizade e companheirismo.

Aos amigos e familiares pela compreensão.

Aos professores do curso ARL pela dedicação.

Ao Professor Joaquim que dedicou seus momentos preciosos e sua mente brilhante com minhas atividades. E também por ser um verdadeiro mestre.

“deve tornar as coisas fáceis ainda mais fáceis, as coisas difíceis mais fáceis e as coisas impossíveis difíceis”

Larry Wall

## **RESUMO**

Os ambientes virtuais de ensino geralmente são desenvolvidos para atender cursos e aulas de ensino a distância. Poucos ambientes se preocupam apenas com o ensino presencial. A idéia neste trabalho é acompanhar o processo de análise e projeto do desenvolvimento de um ambiente virtual de apoio ao ensino presencial. Para isto foi necessário acompanhar o processo de desenvolvimento do ambiente GradeNet na sua terceira versão, buscando seu histórico e a influência de outros ambientes como Manhattan, TelEduc, CNAMS e Interact.

## SUMÁRIO

<b>1 – INTRODUÇÃO</b>	<b>1</b>
1.1 – <i>A Internet no Meio Acadêmico</i>	2
1.2 – Ambientes Virtuais de Ensino a Distância	2
1.3 – Nasce o GradeNet	3
1.4 – A Nova Versão do GradeNet	4
<b>2 – AMBIENTES VIRTUAIS DE ENSINO</b>	<b>5</b>
2.1 – <i>The Manhattan Virtual Classroom</i>	5
2.2 – Ambiente de Ensino a Distância - TelEduc	8
2.3 – <i>Cisco Networking Academy Management System (CNAMS)</i>	11
2.4 – <i>Interact On-line Learning and Collaboration</i>	13
2.5 – Influência dos Ambientes no GradeNet	16
<b>3 – METODOLOGIAS E FERRAMENTAS UTILIZADAS</b>	<b>17</b>
3.1 – <i>Software e Engenharia de Software</i>	17
3.1.1 – Paradigmas da Engenharia de Software	18
3.1.2 – Ciclo de Vida – Modelo Incremental	18
3.1.3 – Visão Genérica da Engenharia de Software	20
3.2 – Orientação a Objetos	20
3.2.1 – Vantagens da Orientação a Objetos	20
3.2.2 – Programação Estruturada Versus Programação Orientada a Objetos	21
3.3 – <i>Unified Modeling Language (UML)</i>	21
3.4 – Metodologia Rápida para Desenvolvimento de Sistemas Orientado a Objetos (MRDS)	22
3.5 – Banco de Dados Relacional	23
3.5.1 – <i>Structured Query Language (SQL)</i>	24
3.6 – Modelo Canônico	24
3.7 – <i>Hipertext Markup Language (HTML)</i>	25
3.8 – Perl	26
3.9 – MySQL	27
3.10 – Outras Ferramentas	28
<b>4 – DESENVOLVIMENTO DO GRADENET</b>	<b>29</b>
4.1 – Análise de Requisitos	29
4.1.1 – Funcionalidades Modularizadas	29
4.1.2 – Funcionalidades Gerais	32
4.1.3 – Filosofias do GradeNet	33
4.1.4 – Usabilidade	35
4.1.5 – Confiabilidade	35
4.1.6 – Eficiência	35
4.1.7 – Portabilidade	36
4.1.8 – Manutenibilidade	36

4.1.9 – Instalabilidade	36
4.2 – Análise, Projeto e Desenvolvimento do Ambiente	36
4.2.1 – Casos de Uso	37
4.2.2 – Diagramas de Classes	37
4.2.3 – Diagramas de Entidade-Relacionamento	39
4.3 – Análise, Projeto e Desenvolvimento do Módulo Avisos	39
4.3.1 – Casos de Uso	43
4.3.2 – Diagramas de Classes	44
4.3.3 – Diagrama Entidade-Relacionamento	44
4.3.4 – Codificação	46
<b>5 – APRESENTANDO O GRADENET</b>	<b>51</b>
5.1 – Situação Atual	51
5.2 – A Página do Projeto	51
5.3 – Instando o Ambiente	52
5.4 – Acessando o Ambiente	53
5.5 – Módulo Avisos	53
5.6 – Módulo Ambiente	54
5.7 – Módulo Atividades	54
5.8 – Módulo Caderno	55
5.9 – Módulo Calendário	55
5.10 – Módulo Testes	57
5.11 – Módulo Usuários	58
5.12 – Módulos em Desenvolvimento	58
<b>6 – CONCLUSÃO</b>	<b>61</b>
<b>REFERÊNCIAS</b>	<b>62</b>
<b>APÊNDICE A – LISTA DE ABREVIATURAS E SIGLAS</b>	<b>65</b>

## LISTA DE ILUSTRAÇÕES

1 – Menu Principal do Ambiente Manhattan	6
2 – Ambiente TelEduc em uso no Curso ARL	9
3 – Estrutura Básica do Ambiente TelEduc	10
4 – Acesso ao CNAMS	12
5 - Utilizando o Ambiente <i>Interact</i>	14
6 - Interações e Conteúdos do Ambiente <i>Interact</i>	15
7 – Modelo Incremental	19
8 – Modelo Canônico	25
9 - Caso de Uso de Alto Nível – GradeNet	38
10 – Diagrama de Classes de Negócio – GradeNet	40
11 – Diagrama de Classes de Interfaces com o Usuário – GradeNet	41
12 – Diagrama de Classes de Domínio – GradeNet	41
13 – Diagrama de Classes de Controle - GradeNet	42
14 – Diagrama de Classes Diversas - GradeNet	42
15 – Diagrama Entidade-Relacionamento – GradeNet	43
16 – Diagrama de Caso de Uso – Módulo Avisos	44
17 – Diagrama de Classes – Acessando Avisos	46
18 – Classe Aviso -Atributos e Métodos	47
19 – Classe DadosAvisos – Atributos e Métodos	48
20 – Classe HtmlAvisos – Atributos e Métodos	48
21 – Entidade TB_Aviso	49
22 – ClasseAviso.pm	49
23 – DadosAvisos.pm	50
24 – HtmlAvisos.pm	50
25 – Página do Projeto GradeNet	52
26 – ConexaoMySQL.pm	53
27 – Acessando o Ambiente GradeNet	54
28 - Acessando Avisos no GradeNet	55
29 – Visualizando Atividades no GradeNet	56
30 – Acessando o Caderno no GradeNet	56
31 – Visualizando o Calendário no GradeNet	57
32 – Visualizando um Teste Avaliado no GradeNet	58
33 – Acessando o Módulo Usuários do GradeNet	59

## **LISTA DE TABELAS**

1 – Descrição do Caso de Uso – Acessando Avisos

45

## 1 – INTRODUÇÃO

Cada vez mais, professores universitários estão utilizando a *Internet* como meio de comunicação com seus alunos. Professores e alunos trocam informações através de recursos, tais como disquetes, compartilhamento de pastas, correio eletrônico e, até mesmo, através de páginas pessoais. Esse compartilhamento de dados, sejam materiais de aulas ou listas de exercícios, através da *Internet* são de grande serventia, mas a forma adequada seria colocar a sala de aula do mundo real dentro de um mundo virtual.

Com a “explosão” da *Internet*, a comunidade acadêmica voltou-se para o ensino a distância. Como resultado surgiram vários ambientes EaD (Ensino a Distância). Esses ambientes estão evoluindo e atingindo um grande número de universidades brasileiras, e em geral são excelentes ferramentas para facilitar a vida de professores e alunos.

Uma grande preocupação dos professores, entretanto, é tornar esses ambientes em um verdadeiro simulador de aulas onde alunos e professores não precisam estar próximos fisicamente. Isso justifica o surgimento de novos recursos dentro desses ambientes. Ainda sim, fica uma lacuna entre a realidade e a virtualidade, ou seja, os ambientes EaD estão atendendo bem o ensino a distância mas nem sempre são adequadas ao ensino presencial.

Com o propósito de auxiliar o professor em suas aulas presenciais, melhorar a qualidade do ensino, poupando tempo e tornando as aulas mais práticas, surge a idéia de elaborar um ambiente virtual de apoio ao ensino chamado **GradeNet**.

## **1.1 – A *Internet* no Meio Acadêmico**

A história da própria *Internet* tem um grande envolvimento com o meio acadêmico, pois a mesma prosperou dentro de universidades americanas nas décadas de 70 e 80; contudo tornou-se popular na década de 90 devido ao surgimento da WWW (*World Wide Web*) permitindo aos usuários lerem páginas pessoais de outras pessoas ou companhias através de um simples navegador.

Dois serviços de *Internet* merecem destaque: o correio eletrônico e a *web*, onde hipertextos são disponibilizados em páginas. Dentro do ambiente acadêmico é muito comum alunos e professores trocarem mensagens eletrônicas com materiais de aula e exercícios em anexo, ou ainda, disponibilizá-los em uma página. Por serem serviços primitivos da *Internet*, apresentam problemas comuns, que vão desde uma lista de exercícios que não chegou ao aluno à desatualização de uma página de um professor.

Para melhor gerenciamento dos serviços de *Internet* de maneira fácil e confiável, criam-se as ferramentas como ambientes virtuais que cada vez mais vem suprimindo necessidades de alunos e professores.

## **1.2 – Ambientes Virtuais de Ensino a Distância**

Um ambiente virtual é uma ferramenta crucial para um bom curso a distância. Ele deve representar de maneira virtual uma sala de aula real, permitindo aos usuários que interajam entre si através de salas de bate-papo, fóruns de discussões e correio eletrônico. A seguir uma breve apresentação dos ambientes virtuais que inspiraram o desenvolvimento do GradeNet: *Manhattan*, *TelEduc*, *CNAMS* e *Interact*.

### 1.3 – Nasce o GradeNet

Em 2001, em uma aula do Curso de Análise de Sistemas na UNICENTRO - Universidade Estadual do Centro-Oeste, o acadêmico Hermano Pereira e seu professor Fernando Takashi Itakura tiveram a idéia: implementar uma simples página para os professores postarem perguntas aos alunos e que os mesmos pudessem respondê-las. O objetivo era apenas substituir o árduo trabalho de ler e entender grafias nas redações de alunos. Isso resultou no desenvolvimento de um ambiente semelhante a um fórum de discussões.

Em 2002, já graduado, Hermano ingressou na mesma Universidade em que se formou para lecionar aulas de Redes de Computadores como professor colaborador. Ao mesmo tempo cursava especialização a distância na UFLA – Universidade Federal de Lavras. O curso ARL – Administração de Redes Linux já utilizava os ambientes *Manhattan* e *TelEduc* em suas aulas a distância. Esses ambientes o motivaram a melhorar o desenvolvimento do ambiente feito em graduação e criar a primeira versão do ambiente virtual, que passou a ser denominado de GradeNet, oriundo de *Graduar via Internet*. O ambiente já disponibilizava módulos de alunos, avisos, materiais de aula, fórum e exercícios.

Em 2003, Hermano fez o curso de certificação CCNA – *Cisco Certified Network Associate* na NetacadBR em Curitiba, que o motivou a melhorar suas aulas e o ambiente GradeNet. Registrou o domínio <www.gradenet.com.br> disponibilizando subdomínios com o ambiente para os professores do DESIS – Departamento de Análise de Sistemas da UNICENTRO. A segunda versão do ambiente manteve sua simplicidade, mas incorporou funcionalidades importantes tais como: tratamento de avisos, exercícios, trabalhos, anexos, fóruns e principalmente o módulo de testes.

#### **1.4 – A Nova Versão do GradeNet**

O ambiente GradeNet evoluiu a cada ano e foi modificado e melhorado no decorrer dos anos letivos. Na sua terceira versão estão sendo adicionadas novas funcionalidades através de sugestões e experiências do uso do ambiente pelos professores do departamento.

O objetivo desta monografia é documentar e apresentar como a terceira versão do GradeNet está sendo desenvolvida. Para isso: no Capítulo 2 será feita uma revisão sobre ambientes virtuais e, no Capítulo 3, serão apresentadas as metodologias utilizadas. Por sua vez, o Capítulo 4 abordará a visão de análise e projeto de desenvolvimento, enquanto o Capítulo 5 fará uma apresentação do GradeNet.

## 2 – AMBIENTES VIRTUAIS DE ENSINO

Neste capítulo são abordados os ambientes virtuais de ensino que serviram como base para o desenvolvimento do GradeNet. São apresentados desde seus históricos, desenvolvedores, mantenedores, licenças de uso, suporte a idiomas e até suas principais funcionalidades que os destacam dos demais.

### 2.1 – *The Manhattan Virtual Classroom*

O *Manhattan* (Figura 1) é um ambiente virtual de ensino a distância com o objetivo principal de simular uma sala de aula virtualmente. Em desenvolvimento desde 1997 por *Western New England College* em *Springfield – Massachusetts* e utilizado semestralmente por mais de 300 cursos, está licenciado sob GPL e tem recebido colaborações pelo mundo inteiro. Ele encontra-se disponível em várias línguas: inglês, finlandês, alemão, grego, espanhol, português brasileiro<sup>1</sup> e polonês.

Entre as funcionalidades do *Manhattan* (MANHATTAN, 2004), encontram-se:

a) *The Post Office*: é um módulo que simula o correio eletrônico convencional, sendo confiável e restrito aos usuários cadastrados na classe. Em poucos cliques é possível enviar mensagens de um usuário ao outro, permitindo anexar arquivos.

b) *Handouts/Notices and Lectures*: são módulos que permitem ao administrador do ambiente inserir informações relevantes a todos os usuários da classe, tais como arquivos, notícias, *links*, artigos, leituras, etc.

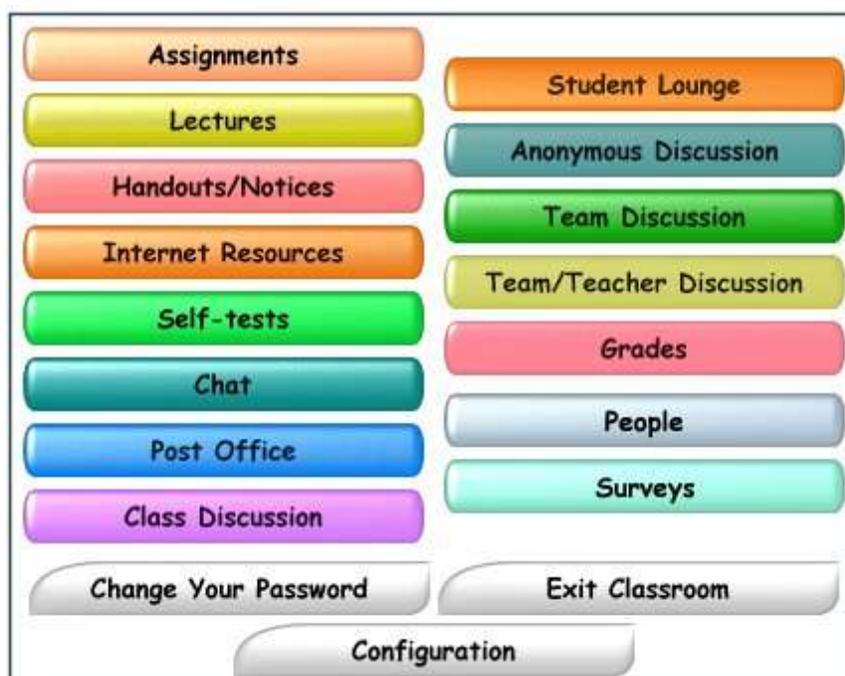
---

<sup>1</sup> - Tradução feita pelo grupo *Linux*, do qual fazem parte membros da Equipe do ARL.

c) *The Self-tests Module*: é um módulo onde o professor pode formular testes com questões objetivas, múltipla escolha, escreva o texto ou complete o texto. Esses testes não são corrigidos no lado do servidor, apenas no lado cliente, ou seja, o processamento é feito no navegador do usuário.

d) *Discussion Groups*: são módulos de fóruns de discussões que podem ser de acesso limitado: para toda a classe, grupo de alunos, grupo de alunos e professores ou até mesmo para postagem anônima.

e) *Chat*: módulo que suporta, em tempo real, o bate-papo entre membros da classe. As salas de bate-papo podem ser criadas pelos usuários que poderão administrar a mesma limitando o acesso de outros usuários.



**Figura 1:** Menu Principal do Ambiente Manhattan

f) Assigments: é um módulo que permite ao professor postar atividades com possibilidade de postar correções das mesmas. Os alunos podem responder às atividades. A idéia é ser um módulo para *paperless*, ou seja, alunos entregam respostas à atividades de maneira *on-line* sem precisarem recorrer à folha de papel convencional.

g) Internet Resources: módulo simples que permite ao professor postar *links* importantes à disciplina juntamente com seus comentários.

h) Grades: o ambiente fornece um formato padrão de planilha de notas para ser utilizada em sua aplicação favorita. Sendo assim o professor poderá importar e exportar notas para o mesmo através da planilha que divulgará as notas para os alunos a partir de suas identificações.

i) Multimedia: não é um módulo, mas sim uma dica de como disponibilizar apresentações e vídeo-aulas no ambiente.

j) Configuration: permite manipular várias informações, como cadastro de alunos, módulos a serem utilizados, organização de grupos, geração das grades de notas e outras configurações do ambiente.

Para ser utilizado, o *Manhattan* deve ser instalado em um servidor *Unix-like*<sup>2</sup>, tais como *FreeBSD*<sup>3</sup> ou *Linux*, no qual também deve estar instalado o servidor de páginas na *Internet: Apache Web Server*. O pacote para instalação do *Manhattan* pode ser baixado no *site*, descompactado e instalado com relativa facilidade. É necessário a senha do superusuário da plataforma para modificar configurações de acesso ao disco e configurações do servidor de páginas.

---

2 - Sistema Operacional semelhante ao *Unix*.

3 - Sistema Operacional baseado em *Unix* de uso livre segundo licença BSD.

## 2.2 – Ambiente de Ensino a Distância – TelEduc

O TelEduc (Figura 2) foi desenvolvido em 1998 pelo NIED – Núcleo de Informática Aplicada à Educação e pelo IC – Instituto de Computação da UNICAMP – Universidade Estadual de Campinas. Está sendo mantido sob orientação da Professora Dra. Heloísa Vieira da Rocha. TelEduc também é parte integrante da dissertação de mestrado de Alessandra de Dutra e Cerceau (CERCEAU, 1998).

É um ambiente para a criação, participação e administração de cursos na *web*. O mesmo está sendo melhorado de forma participativa, ou seja, todas as suas ferramentas foram idealizadas, projetadas e depuradas segundo necessidades relatadas por seus usuários.

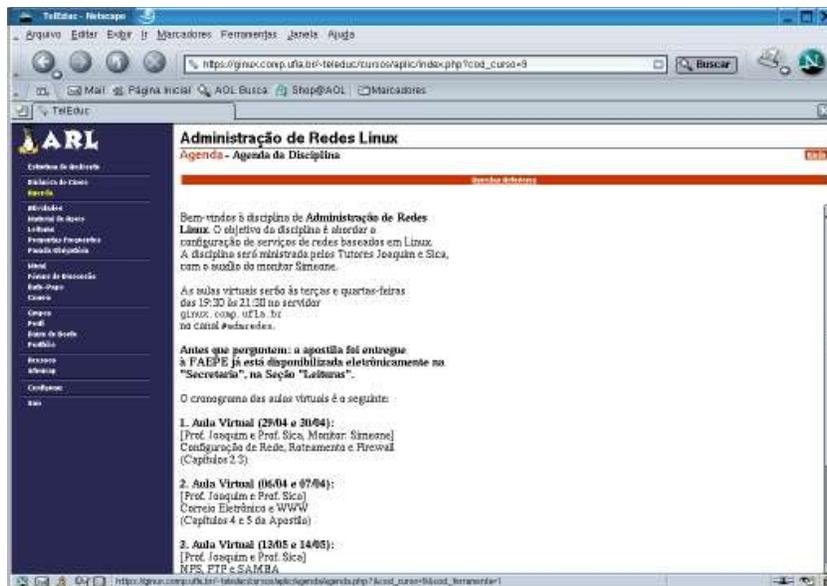
Licenciado sob GPL é um ambiente em português, mas já está disponível em espanhol e também está sendo traduzido para o inglês. Destaca-se o seu uso por mais de 500 entidades de ensino no Brasil, desde cursos, escolas, à universidades.

Atualmente o ambiente provê as seguintes funcionalidades (ROCHA, 2002) e (TELEDUC, 2004):

a) Acessos: módulo que permite acompanhar os acessos dos usuários no ambiente.

b) Agenda: é a página de entrada do ambiente e do curso em andamento. Com a programação de um determinado período do curso.

c) Atividades: para que o professor possa postar atividades a serem realizadas durante o curso. O ambiente TelEduc focaliza o módulo atividades como elemento central (Figura 3).



**Figura 2:** Ambiente Teleduc em uso no Curso ARL.

d) Bate-papo: podendo ser informado na Agenda, é um módulo que permite aos usuários a comunicação em tempo real.

e) Correo: funcional como o correio eletrônico convencional, mas confiável e restrito aos usuários do ambiente, permitindo a troca de mensagens entre os mesmos e, se necessário, anexar arquivos.

f) Diário de Bordo: onde o aluno poderá anotar, como se fosse um diário, seu sucessos, dificuldades, anseios, dúvidas, para servir como reflexão do curso.

g) Dinâmica do Curso: para o professor postar informações em relação ao curso, por exemplo: ementa, disciplina, formas de avaliação, etc.

h) Grupos: módulo que permite aos professores ou alunos formarem grupos para realizarem determinadas tarefas.

i) InterMap: facilitar a visualização de dados quantitativos das interações de mensagens de bate-papo, fórum e correio entre usuários do ambiente (OTSUKA; ROCHA, 2002).

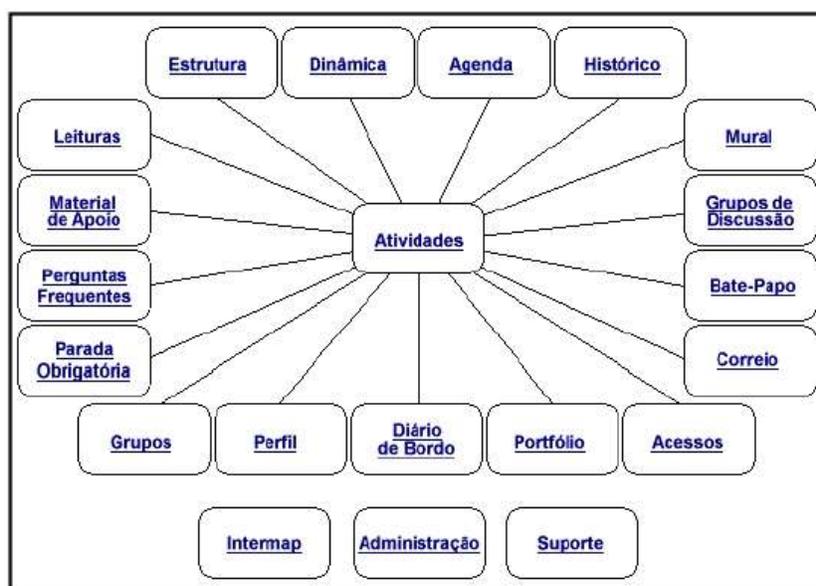
j) Leituras: artigos, publicações e sugestões de leituras para os alunos.

k) Material de Apoio: a idéia é de que o professor possa postar informações úteis para apoio ao curso.

l) Mural: com o real objetivo de “mural” onde qualquer usuário poderá postar algo (informação ou aviso) que interessa à todos que estão ligados ao curso.

m) Perfil: onde cada usuário preenche suas informações pessoais para que outros usuários possam conhecê-lo melhor.

n) Perguntas Frequentes: posta-se apenas uma vez a resposta para dúvidas repetitivas e comuns no ambiente.



**Figura 3:** Estrutura Básica do Ambiente Teleduc (ROCHA, 2002)

o) Portfólio: é uma área onde alunos ou grupos de alunos podem organizar seus trabalhos e arquivos para que possam ser avaliados e comentados.

p) Administração: é um módulo exclusivo do administrador do curso, permitindo ao mesmo configurar o ambiente e cadastrar alunos e senhas.

Os pacotes de instalação e atualização do TelEduc podem ser baixados em (TELEDUC, 2004). O ambiente pode ser instalado em servidores Linux no quais também tenham instalado o servidor de páginas *Apache* com módulos de PHP e servidor de banco de dados MySQL. A versão *off-line* do TelEduc pode ser instalada no MSWindows, mas com muitas limitações. O TelEduc tem uma instalação um tanto exaustiva, mas compensadora. Usuários leigos têm dificuldades de instalar o mesmo, pois exige que o mesmo deva conhecer configurações de Apache e MySQL.

### **2.3 - Cisco Networking Academy Management System (CNAMS)**

O CNAMS (CNAMS, 2004) é um modelo inovador de *e-learning* criado pela Cisco® e faz parte do programa CNAP que tem por objetivo levar cursos de qualidade de certificação CCNA para vários países. O ambiente CNAMS (Figura 4) foi desenvolvido para dar suporte às academias de treinamento autorizadas, sendo assim um *software* proprietário.

Fundado em 1997 o programa CNAP conta com mais de 9.800 academias em mais de 140 países com mais de 260.000 alunos legítimos<sup>4</sup> (NETACAD, 2004).

Existem três tipos principais de usuários no CNAMS, sendo que um usuário pode ser cadastrado em mais de um tipo:

---

4 - Aluno legítimo é aquele que está cadastrado no CNAMS.

a) Administrator: tem permissões de manutenção geral das informações da academia adicionando classes, instrutores, alunos, selecionando o *curriculum*<sup>5</sup> adequado.

b) Instructor: é o instrutor da classe, o qual poderá editar informações da mesma, convidar alunos e aplicar provas aos mesmos.

c) Student: é o aluno da academia que tem acesso aos materiais do curso através do ambiente.

No ambiente CNAMS, existem vários recursos e ferramentas para seus usuários, mas o que realmente diferencia dos outros ambientes é a coerência entre dois módulos principais:



Figura 4 – Acesso ao CNAMS (NETACAD, 2004).

5 - É o currículo *on-line* da Cisco com capítulos detalhados e ilustrados de um determinado curso.

a) *Curriculum On-line*: traduzido para mais de 10 idiomas, é um currículo preparado por profissionais e pedagogos da área de *networking*. Protegido por leis de *copyright*, é um material bem preparado, ilustrado com imagens dinâmicas junto a ferramentas e estudo de casos. É um material de excelente qualidade e fácil assimilação motivando os alunos ao conteúdo do curso e pode ser encontrado na demonstração em (CCNA, 2004).

b) *Assessments/GradeBook*: dos mesmos autores do *Curriculum*, são questões randômicas criadas para serem apresentadas em testes *on-line* (*Assessments*), os alunos são avaliados e a dinâmica e o desempenho da classe e de todos os alunos podem ser acompanhados na grade de notas (*GradeBook*).

Por ser proprietário, o ambiente é armazenado nos próprios servidores da Cisco, mas pode ser utilizado se o interessado buscar uma academia ou um programa para ingressar no CNAMS como *student* ou *instructor*.

#### **2.4 – Interact On-line Learning and Collaboration**

*Interact* (Figura 5) não é só um ambiente virtual de ensino a distância, mas também pode ser usado como meio de interação de grupos de trabalho (INTERACT, 2004), auxiliando a comunicação entre os membros, sendo considerada também uma ferramenta de colaboração. Licenciado sob GPL, o *Interact* está sendo coordenado e desenvolvido em *Christchurch College of Education* – Nova Zelândia, desde 2001 e utilizado por aproximadamente 1000 *sites*. Está disponível em inglês, mas também está sendo traduzido para o alemão e para o português brasileiro.

O principal destaque do ambiente *Interact* é a flexibilidade de modularizar seu curso. A partir de interações e conteúdos é possível criar componentes dentro do ambiente, ilustrado na (Figura 5).

Apresentando as seguintes funcionalidades segundo (INTERACT, 2004):

a) Forum: para discussões entre membros de um grupo onde usuários podem postar comentários, questões, ajuda, etc.

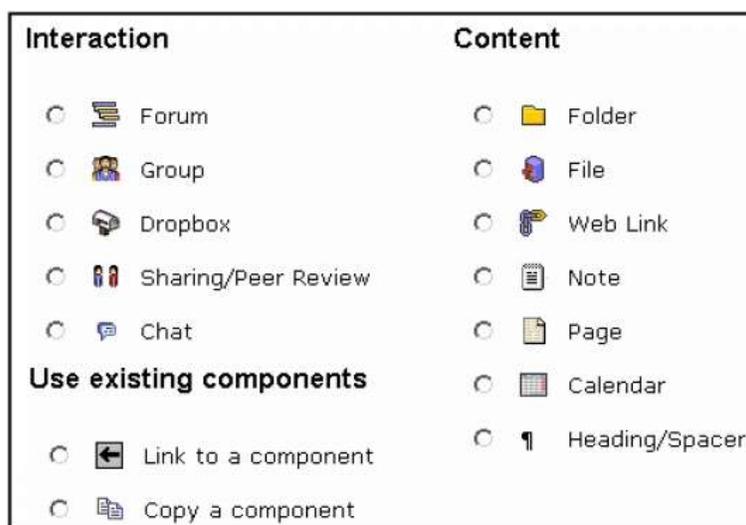
b) Group: para criar grupos de usuários. Esses usuários poderão interagir entre si ou somente com outros componentes que estão no mesmo *Folder*.

c) DropBox: local onde usuários podem submeter suas atividades e documentos de modo privado, exceto o administrador que também poderá visualizá-los.



Figura 5: Utilizando o Ambiente *Interact*.

- d) Sharing/Peer Review: local onde usuários podem submeter arquivos de modo público, e qualquer usuário poderá inserir comentários.
- e) Chat: bate-papo assíncrono para aqueles que estão *on-line*.
- f) Folder: utilizado para distribuir o curso em seções ou, se desejado, em períodos. Qualquer outro componente pode ser adicionado em um componente *Folder*.
- g) File: para usuários enviarem arquivos para o curso em vários formatos.
- h) WebLink: para poder criar um *link* com outras fontes na *Internet*.
- i) Notes: para adicionar anotações.
- j) Page: para adicionar páginas no formato texto ou HTML.



**Figura 6:** Interações e Conteúdos do Ambiente Interact.

- k) Calendar: para adicionar anotações de datas importantes.
- l) Heading/Spacer: se um *Folder* é utilizado para modularizar o ambiente, este componente permite adicionar cabeçalhos e informações.
- m) Link: para poder criar um *link* a outro componente.
- n) Copy: para fazer uma cópia de um componente existente.

O ambiente *Interact* pode ser baixado em (INTERACT, 2004) e, instalado em qualquer plataforma que suporte *Apache*, *PHP* e *MySQL*. Assim como outros ambientes, o *Interact* deve ser instalado com a senha de superusuário, mas sua instalação é simples, bastando apenas editar um ou dois arquivos de configuração.

## 2.5 – Influência dos Ambientes no GradeNet

Como pode ser visto, os quatro ambientes influenciaram o desenvolvimento do ambiente GradeNet nos seguintes aspectos:

a) Manhattan: esse foi o primeiro ambiente virtual em que o autor teve contato, retirando assim uma idéia de como distribuir em módulos e principalmente como funcionam os módulos de *Forum*, *Assigments* e *Notices*.

b) TelEduc: por ser um ambiente com idéias bem modularizadas, foram baseadas algumas idéias e outras funcionalidades em comum, tais como Acessos, Perfil, Senhas (notificação por e-mail), Habilitar Módulos, Últimas Inclusões.

c) CNAMS: é um ambiente voltado para a qualidade e motivação nas aulas, logo, são implementados no GradeNet a filosofia de um material didático, e principalmente a idéia de aplicar testes *on-line*.

d) Interact: é um ambiente flexível e modularizado, e sua influência é a idéia de utilizar ícones para melhor visualização do ambiente. E, também, a idéia de criar uma flexibilidade, não exatamente igual ao do ambiente *Interact*, mas sim ao de anexar um arquivo em algum lugar e poder acessá-lo através de *links* em qualquer outra parte do ambiente.

### **3 – Metodologias e Ferramentas Utilizadas**

Neste capítulo serão apresentados alguns conceitos de Engenharia de *Software*, bem como, metodologias e ferramentas utilizadas para o desenvolvimento do ambiente virtual.

#### **3.1 – *Software* e Engenharia de *Software***

*Software* pode ser definido como o conjunto de instruções que quando executadas produzem a função e o desempenho desejado (PRESSMAN, 95). Sem as instruções fornecidas pelo *software*, o *hardware* é incapaz de executar qualquer das tarefas associadas ao computador (LAUDON, K.; LAUDON, J., 99).

Segundo (PRESSMAN, 95), Engenharia de *Software* é uma disciplina que integra métodos, ferramentas e procedimentos para o desenvolvimento de *software* de computadores.

Os métodos proporcionam os detalhes de como fazer para construir o *software*. Eles envolvem um amplo conjunto de tarefas que incluem: planejamento e estimativas de projeto, análise de requisitos de *software* e de sistemas, projeto da estrutura de dados, arquitetura de programa e algoritmo de processamento, codificação, teste e manutenção.

As ferramentas fornecem o apoio automatizado ou semi-automatizado aos métodos, podendo ser integradas de forma que a informação criada por uma ferramenta possa ser usada por outra. Os procedimentos são o elo de ligação entre métodos e ferramentas que permitem construir o *software* de maneira racional e oportuna.

### 3.1.1 – Paradigmas da Engenharia de Software

Uma série de diferentes paradigmas da engenharia de *software* foram propostos, cada um exibindo potencialidades e fragilidades, mas todos tendo uma série de fases genéricas em comum. A seguir alguns paradigmas segundo (PRESSMAN, 95):

a) Ciclo de Vida Clássico: também conhecido como modelo cascata que requer uma abordagem sistemática, seqüencial ao desenvolvimento do *software*.

b) Prototipação: processo que possibilita o desenvolvedor a criar um modelo de *software*. Passando por fases de análise de requisitos, avaliação de protótipos e produto final.

c) Modelo Espiral: foi desenvolvido para abranger as melhores características do ciclo de vida clássico e da prototipação, acrescentando ao mesmo tempo, um novo elemento - a análise de riscos – que faltava a esses paradigmas.

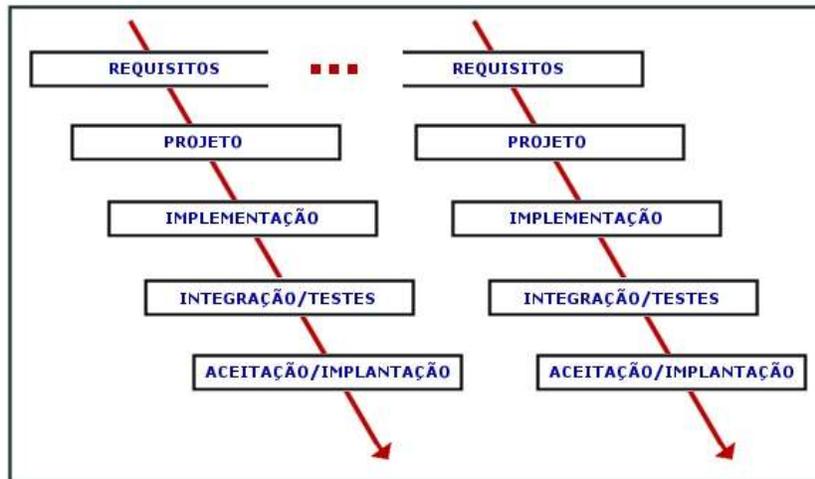
d) Técnicas de 4ª Geração: ferramentas que permitem ao desenvolvedor criar uma parte do *software*, gerando o código automaticamente.

Observe que os paradigmas podem ser combinados para o desenvolvimento do *software* de maneira adequada.

### 3.1.2 – Ciclo de Vida – Modelo Incremental

É uma evolução modelo de cascata, onde o mesmo é implementado em cada módulo do sistema individualmente. Para amenizar os problemas causados pela pouca informação e pelas alterações de requisitos durante o projeto de sistema de informação, o sistema é dividido em pedaços menores.

Este modelo de ciclo de vida é chamado de Modelo Incremental (Figura 7). Um modelo de processo em que cada estágio mostra incrementos na funcionalidade do produto e tem como característica importante antecipar o *feedback*<sup>6</sup> do usuário.



**Figura 7:** Modelo Incremental

Após a especificação dos requisitos, o produto é dividido em módulos, sendo que cada um destes módulos deve oferecer alguma função que agregue valor ao usuário.

Segundo (SCHMITZ; SILVEIRA, 2000), como neste modelo o usuário vai recebendo pedaços do sistema que já podem ser utilizados e como os pedaços são menores, as variações com relação a prazo e custo também são menores, sendo assim as alterações afetam pedaços menores do sistema, diminuindo os impactos no custo e no prazo.

<sup>6</sup> No sentido de: uma resposta ou retorno rápido.

### 3.1.3 – Visão Genérica da Engenharia de *Software*

Algumas fases são genéricas no desenvolvimento de *software*, sendo utilizadas em todos os paradigmas (PRESSMAN, 95):

- a) Fase de definição: identifica quais informações devem ser processadas.
- b) Fase de desenvolvimento: tenta definir a estrutura de dados e a arquitetura de *software* a ser projetada.
- c) Fase de manutenção: referente às mudanças que estão associadas às correções de erros e adaptações exigidas.

### 3.2 – Orientação a Objetos

O projeto e a análise orientados a objetos é uma estratégia de desenvolvimento baseada no conceito de que os sistemas devem ser construídos a partir de um conjunto de componentes reutilizáveis, chamados de objetos. Os objetos procuram incorporar simultaneamente os dados e a funcionalidade de um sistema, em vez de considerá-los isoladamente, tal como é feito no paradigma de programação estruturada (SCHMITZ; SILVEIRA, 2000).

#### 3.2.1 – Vantagens da Orientação a Objetos

A orientação a objetos oferece o benefício potencial de resolver muitos dos problemas atualmente enfrentados pela indústria de informática. Algumas das vantagens segundo (AMBLER, 97) são:

- a) Reusabilidade: maior eficiência no projeto e oportunidade de reutilização.
- b) Extensibilidade: acrescentar novos recursos ao programa basta modificar em apenas um lugar.

c) Aumento da Qualidade: por permitir melhor manutenibilidade do *software* conseqüentemente, o *software* será de melhor qualidade.

d) Vantagens Financeiras: possibilita o desenvolvimento do sistema com rapidez, qualidade e economia.

### **3.2.2 – Programação Estruturada Versus Programação Orientada a Objetos**

O paradigma estruturado é uma estratégia de desenvolvimento baseado no conceito de que um sistema deve ser dividido em duas partes: os dados e as funcionalidades. Assim utilizando a abordagem estruturada o desenvolvimento de aplicações nas quais os dados ficam separados do comportamento tanto no projeto quanto na implementação do sistema (SCHMITZ; SILVEIRA, 2000).

O paradigma de orientação a objetos, em vez de definir sistemas com duas partes separadas (os dados e a funcionalidade), passa a conceber os sistemas como um conjunto de objetos interativos. Os objetos desempenham ações e armazenam informações, ou seja, possuem funcionalidades e contém dados.

### **3.3 – Unified Modeling Language (UML)**

A UML é o resultado da unificação dos métodos de Booch, OMT e OOSE que são apresentados em (FURLAN, 98). É uma notação padronizada para modelagem de sistemas de *software* orientados a objetos, sendo adotada pela indústria de *software* como linguagem padrão, e também, por fornecedores de ferramentas CASE.

A UML foi criada em 1994 mas sua versão 1.0 se oficializou em 1997 quando foi submetida à OMG (*Object Management Group*). Atualmente a UML se encontra na versão 2.0. UML tornou-se a linguagem padrão para especificar, visualizar, documentar e construir artefatos de um sistema que pode ser utilizada com todos os processos ao longo do ciclo de desenvolvimento e através de diferentes tecnologias de implementação. Ela busca unificar as perspectivas entre os diversos tipos de sistemas e fases de desenvolvimento de forma que permite levar adiante determinados projetos que antes não eram possíveis pelos métodos existentes (FURLAN, 98).

A UML busca uma simples padronização de notação unificada, uma vez que contém conceitos novos que não são encontrados em outros métodos orientados a objetos. A UML recebeu influência das técnicas de modelagem de dados (Diagrama de Entidade e Relacionamento), modelagem de negócio, modelagem de objetos e componentes (FURLAN, 98).

### **3.4 – Metodologia Rápida para Desenvolvimento de Sistemas Orientados a Objetos (MRDS)**

A MRDS tem por objetivo auxiliar o projetista no desenvolvimento de sistema de informação. Ela cobre as fases de desenvolvimento que vão desde a captura inicial de requisitos até a especificação completa da interface de todos os módulos componentes do sistema. Segundo (SCHMITZ; SILVEIRA, 2000), a MRDS se divide em duas fases:

a) 1ª Fase – Definição dos Requisitos: o objetivo desta fase é definir a funcionalidade esperada pelo usuário do sistema criando um Modelo de Requisitos que é composto pelo Modelo de Caso de Uso, Maquete e Glossário. O Modelo de Caso de Uso por sua vez é composto pelos Diagramas de Caso de Uso e as Descrições do Caso de Uso.

b) 2ª Fase – Análise e Projeto: o objetivo desta fase é de produzir um plano que permita a construção do sistema criando um Modelo de Análise e Projeto que é composto pelos Modelos Estático e Dinâmico. O Modelo Estático é composto por um conjunto de Diagramas de Classes. O Modelo Dinâmico é composto por um conjunto de Diagramas de Seqüências e Diagramas de Estados.

### **3.5 – Banco de Dados Relacional**

A estrutura de banco de dados desenvolvida mais recentemente, o modelo de banco de dados relacional, foi desenvolvida para superar limitações dos outros modelos na representação de relacionamentos de dados. O modelo relacional representa todos os dados do banco de dados em tabelas simples bidimensionais denominadas relações. As tabelas parecem semelhantes a arquivos simples, mas as informações em mais de um arquivo podem ser extraídas e combinadas com facilidade.

A vantagem do modelo relacional está em que um elemento de um dado de um arquivo ou tabela pode ser relacionado a qualquer fragmento em outro arquivo ou tabela desde que ambas compartilhem um elemento de dado comum (LAUDON, K.; LAUDON, J., 99).

Um SGBD (Sistema Gerenciador de Banco de Dados) Relacional permite que dados sejam armazenados em um só lugar, tornando-os disponíveis para diferentes aplicações. O SGBD serve como uma interface entre o banco de dados comum e diversos programas aplicativos.

A técnica utilizada para modelar relacionamentos entre dados independente de SGBD é utilizada para criar diagramas ER (Entidades e Relacionamentos) (HARRINGTON, 2002) .

### **3.5.1 – *Structured Query Language (SQL)***

A SQL foi desenvolvida pela IBM em meados de 70 para ambientes de *mainframes* e minicomputadores e foi apresentada comercialmente em 1979. Ela foi incorporada a alguns ambientes de PC. Em 1986 a ANSI publicou o padrão SQL com o objetivo de se tornar uma linguagem padrão para SGBD's relacionais (LAUDON, K.; LAUDON, J., 99).

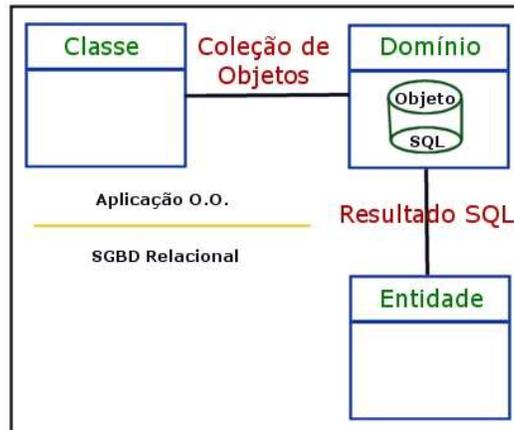
Como uma linguagem de manipulação de dados completa, a SQL contém instruções que permitem inserir, modificar, excluir e recuperar dados (HARRINGTON, 2002).

A SQL é uma linguagem orientada a transformações que aceitam uma ou mais relações como entrada e produz uma relação única como entrada e produz uma relação única como saída. O resultado de toda consulta de SQL é uma relação; mesmo que o resultado seja um único número, este é considerado uma relação com uma linha e uma única coluna (KROENKE, 1998).

### **3.6 – Modelo Canônico**

Observe que ao utilizar uma linguagem orientada a objetos e um SGBD relacional torna-se necessário adotar o modelo canônico (Figura 8). Esta forma de mapeamento permite tanto o isolamento das características do SGBD particular como também segue os princípios de encapsulamento e ocultação da informação. Isso pode ser resolvido criando três classes: uma no modelo UML e

outra no modelo ER, e ainda, uma terceira que traduz do formato orientado a objetos para relacional ou vice-versa (SCHMITZ; SILVEIRA, 2000).



**Figura 8:** O Modelo Canônico

### 3.7 – *Hipertext Markup Language (HTML)*

Em 1990 no laboratório europeu CERN (*European Organization for Nuclear Research*), Tim Beners-Lee desenvolveu uma linguagem chamada HTML (W3C, 2004). Logo após a sua criação, ela foi utilizada como linguagem padrão pela NCSA (*National Center for Supercomputing Applications*) em seu navegador chamado Mosaic.

Em 1995 a IETF (*Internet Engineering Task Force*) lançou a versão 2.0, e a partir daí, empresas como a Netscape© Communications Corporation e a Microsoft© Corporation inseriram novos comandos no HTML para executar funções em seus navegadores. Atualmente na versão 4.0, o HTML é hoje uma aplicação ISO 8879 (TANENBAUM, 97).

HTML é uma linguagem que determina um padrão para a definição de vinculação entre documentos (hipertexto), sendo assim, uma linguagem básica para a construção de páginas de Internet. O HTML trata de várias características tais como: inserção de figuras, textos, vinculação, figuras animadas, formulários, entre outros (RITCHEY, 96).

### 3.8 – Perl

Desenvolvido em 1986 por Larry Wall que era um programador de sistemas da NSA (*National Security Agency*), em 1987 anunciou sua linguagem PERL (*Practical Extraction and Report Language*) (PERL, 2004). Adotou duas licenças, a GPL e a *Artistic License* que impõe restrições a produtos proprietários.

A utilidade, velocidade, flexibilidade e portabilidade de Perl tornaram-na ideal para a *World Wide Web*. Sendo uma linguagem baseada em processamento de texto, é adaptável para manipular o meio baseado em textos da *Web*. Perl é particularmente eficaz no uso com o protocolo CGI (*Common Gateway Interface*), que possibilita que servidores da *Web* interajam com outras aplicações e passem sua saída para aplicações clientes sob a forma de conteúdo dinâmico (DEITEL, 2002).

Segundo (GUELICH; GUNDAVARAM; BIRZNIEKS, 2001), CGI e Perl se tornaram sinônimos, mesmo CGI podendo ser programados em outras linguagens. Pode-se então qualificar o seguinte:

- a) Perl é fácil de aprender, por se parecer com outras linguagens populares;
- b) Perl permite rápido desenvolvimento por ser interpretado, ou seja, não precisa de compilação antes da execução;

- c) Perl é facilmente portátil;
- d) Perl contém expressões regulares que são operadores poderosos para manipulação de *strings*;
- e) Perl manipula dados binários tão fácil quanto texto;
- f) Perl não exige tipagem em suas variáveis;
- g) Perl se conecta facilmente a aplicativos externos;
- h) Existem incontáveis módulos perl de código aberto na *Internet*.
- i) Perl é de uso livre.

### 3.9 – MySQL

Em 1995, uma equipe de programação utilizava o mSQL (HUGHES, 2004) para conectar suas tabelas, usando suas próprias rotinas de baixo nível, mas notaram que o mSQL não era rápido e nem flexível o suficiente. Então, os mesmos resolveram desenvolver uma nova interface SQL surgindo o MySQL (MYSQL, 2004).

MySQL é um sistema gerenciador de banco de dados, multi-tarefas e robusto. Ele é um dos bancos de dados mais usados para a *Internet*, principalmente por servidores Linux. Por ser livre, apresenta uma grande vantagem sobre diversos gerenciadores de bancos de dados, entrando em grande disputa neste mercado (DIAS, 2000).

Características do MySQL segundo (STOCO, 2000):

- a) Permite um número ilimitado de utilização por usuários simultâneos;
- b) Trabalha em diferentes plataformas;
- c) Pode associar tabelas de banco de dados diferentes em uma mesma consulta;
- d) Fornece um sistema de privilégio de senhas flexível e seguro;
- e) Permite até 32 indexadores por tabela;

- f) Apresenta capacidade de manipulação de tabelas com mais de 50.000.000 registros;
- g) Permite a conexão ao servidor utilizando *TCP/IP*<sup>7</sup>;
- h) Licença de uso GPL, sendo cobrado apenas o suporte se o fizer necessário.

### 3.10 – Outras Ferramentas

Outras ferramentas foram utilizadas para o desenvolvimento do ambiente, às quais serão citadas:

- a) *Poseidon for UML*: para modelagem de Casos e Uso e Diagramas de Classes.
- b) *DBDesigner*: para modelagem de Diagramas de Entidade-Relacionamento.
- c) *Apache Web Server*: servidor de páginas na *Internet* integrando Perl ao MySQL.
- d) *Free Icons*: coleção de ícones fornecidas livremente por (JONES, 2004)

---

<sup>7</sup> Arquitetura padrão da Internet.

## **4 – DESENVOLVIMENTO DO GRADENET**

Este capítulo apresenta as atividades realizadas para o desenvolvimento do ambiente GradeNet. Serão apresentados os requisitos do ambiente, bem como a análise e projetos através de diagramas. Com o propósito de demonstrar como o ambiente foi desenvolvido, serão detalhados os diagramas do Módulo de Avisos.

### **4.1 – Análise de Requisitos**

Buscou-se através de definição de requisitos especificar as funcionalidades do ambiente a ser desenvolvido. Foram recolhidos dados de outros ambientes, experiências de professores que o utilizaram e suas dificuldades no ensino presencial.

#### **4.1.1 – Funcionalidades Modularizadas**

Nesta seção serão especificadas as funcionalidades do ambiente, divididas em módulos que estão descritos a seguir:

##### **Módulo Ambiente:**

No módulo Ambiente o professor poderá configurar as informações do ambiente, tais como cores, nome da disciplina, ementa e plano de ensino entre outras informações sem precisar recorrer aos arquivos de configuração.

### **Módulo Atividades:**

Um dos módulos mais importantes no ambiente, permitirá que o professor cadastre qualquer tipo de atividade para os alunos, desde simples exercícios à trabalhos de conclusão. É um módulo que controlará a postagem de informações, ou seja, a atividade poderá ser bloqueada até certa data e hora, e só poderá ser respondida até uma data e hora limite. Outra funcionalidade do módulo de atividades é a permissão para que os alunos possam postar arquivos. Após a data limite, o professor poderá fazer a correção e/ou atribuir as notas.

### **Módulo Avisos:**

A idéia deste módulo é que os professores poderão disponibilizar para os alunos qualquer tipo de aviso, desde data de provas à *links* para materiais de aulas ou notas finais. Ou seja, será um módulo que permitirá ao professor postar qualquer coisa que deseja passar aos alunos. Será a funcionalidade mínima do ambiente, mas de grande importância, pois será a primeira informação que o aluno verá quando acessar o ambiente.

### **Módulo Biblioteca:**

Será um módulo de material de apoio à disciplina em geral, onde cada item poderá ser de um simples *link* de indicação de livro, à um *e-book*<sup>8</sup> completo. O professor poderá incluir itens que quiser, adicionando uma descrição de cada para facilitar a pesquisa dos alunos.

### **Módulo Caderno:**

Objetivo deste módulo será substituir o caderno e o disquete convencional, onde o aluno poderá fazer as suas anotações e armazenar seus arquivos.

---

8 Livro em formato eletrônico.

### **Módulo Calendário:**

Um importante módulo que permitirá aos professores e alunos se situarem dentro do cronograma da disciplina. O módulo será vinculado aos módulos de avisos e atividades, permitindo ao professor agendar as informações que deseja dentro do ano corrente.

### **Módulo Fórum:**

Neste módulo somente professores poderão postar perguntas, é um fórum relevante aos assuntos de interesse do professor onde qualquer usuário poderá participar.

### **Módulo Fórum Livre:**

Qualquer usuário poderá postar perguntas, funcionando de maneira semelhante ao módulo anterior.

### **Módulo Notícias:**

Talvez o módulo mais simples, mas muito requisitado, pois professores gostam de estar atualizados em relação à algum assunto. Permitirá ao professor postar notícias externas ao processo atual da disciplina, ou seja, notícias atuais que estão na mídia, tais como revistas, jornais, *sites*, etc.

### **Módulo Testes:**

É um módulo que permitirá ao professor cadastrar questões objetivas para gerar testes de conhecimento para seus alunos. Os testes serão gerenciados pelo professor. Ou seja, o aluno só terá acesso ao teste se for permitido. Quando encerrar um teste o ambiente corrigirá automaticamente e apresentará a nota do aluno.

### **Módulo Usuários:**

É um módulo que permitirá gerenciar as informações dos usuários cadastrados no ambiente, tais como nomes, senhas, fotos e acessos. Serão permitidos apenas dois tipos de usuários: professores e alunos.

### **4.1.2 – Funcionalidades Gerais**

Nesta seção serão apresentadas as funcionalidades gerais do ambiente, ou seja, funcionalidades comuns a todos os módulos do ambiente.

### **Destacar últimas informações:**

Quando acessar o ambiente, o usuário poderá procurar pelas últimas inclusões que estarão destacadas em negrito, começando pelo menu principal.

### **Destacar calendário e testes:**

Quando houver alguma informação na data corrente, ou algum teste estiver aberto, deverá haver algum destaque para esses módulos.

### **Habilitar módulos:**

O professor poderá habilitar apenas os módulos que deseja utilizar em sua disciplina.

### **Números de usuários *on-line*:**

Os usuários poderão saber quem mais está acessando o ambiente ao mesmo tempo.

### **Pseudo-HTML:**

Será permitido aos usuários que utilizem uma simulação de HTML para melhor formatarem seus textos. Também será permitido fazer *links* para imagens e páginas na *Internet*.

### **Registro vinculando perfil:**

A cada informação cadastrada no ambiente, haverá um *link* para acesso ao perfil do usuário.

### **Registro de data e hora:**

A cada informação cadastrada no ambiente, haverá a data e hora da inclusão.

### **Visualização *bottom-up*:**

Qualquer informação adicionada no site deverá ser apresentada sempre a última por primeiro, ou seja, LIFO (*Last In First Out*).

## **4.1.3 – Filosofias do GradeNet**

As decisões foram tomadas com base em algumas constatações no uso de versões anteriores do GradeNet. Essas constatações foram feitas pelo autor e por outros professores que utilizaram o ambiente. As constatações são as seguintes:

- a) Módulo Aviso é suficiente para armazenar informações relevantes a todos os usuários.
- b) Apresentar as últimas informações inclusas no ambiente (LIFO) economiza tempo dos usuários.

---

9 De baixo para cima.

c) Listar informações juntamente com data e hora, e ainda o nome dos autores, motiva os outros a participarem do ambiente também.

d) Utilizar o Módulo Caderno, evita perder arquivos e anotações em disquetes e *e-mails*.

e) No ambiente não existe bate-papo pois existem várias ferramentas na *Internet* com este propósito, e além do mais as aulas são presenciais.

f) No ambiente não existe correio eletrônico, no máximo uma listagem de *e-mails* dos usuários. A idéia é não utilizar o correio eletrônico convencional, por não ser confiável procurando acostumar o aluno a acessar o ambiente periodicamente.

g) Os alunos não poderão ler respostas de atividades uns dos outros, a não ser que tenha expirado a data limite. Sendo assim, um bom argumento para professores não aceitarem respostas atrasadas.

h) Apenas a inclusão de uma correção para uma atividade é suficiente para todas as respostas. Pois se as respostas são abertas para todos, fica mais fácil para o professor citar erros de uma ou mais respostas apenas uma vez.

i) O ambiente estabelece uma maneira aberta de comunicação entre os alunos, pois todos estão em um ambiente virtual vinculado ao mundo real, ou seja, todos se conhecem.

j) A biblioteca é flexível, mas a idéia por trás de tudo isso é que o aluno tenha acesso à várias fontes para resolver seus exercícios de aula.

k) A idéia de apresentar um *ranking* é apenas uma questão de competitividade entre os alunos, ou seja, todos querem estudar para conquistar uma boa colocação.

l) Saber o número de usuários *on-line* e quem está *on-line* motiva os alunos, pois os mesmos vêem seus colegas estudando e fazendo atividades.

m) Disponibilizar materiais de aulas aos aluno pelo ambiente, melhora a qualidade das aulas e melhora o rendimento dos alunos dentro da disciplina.

#### **4.1.4 – Usabilidade**

O ambiente fornecerá uma interface padrão, utilizando somente recursos simples de HTML. Tais como, *links*, botões, tabelas comuns sem implementar formatações de textos avançadas, e outras linguagens. Para permitir que o ambiente possa ser acessado de navegadores com poucos recursos.

#### **4.1.5 – Confiabilidade**

Do ambiente: o ambiente deverá ser confiável para não permitir que usuários não autorizados tenham acessos à informações da base.

Do servidor *web*: o servidor deverá implementar políticas de segurança para uma boa proteção ao ambiente, e *backups* periódicos.

#### **4.1.6 – Eficiência**

Do ambiente: o ambiente deverá ser eficiente para apresentar as informações ao usuário, ou seja, colocar as informações de maneira mais rápida para que o mesmo possa ter um acesso rápido e eficiente.

Do servidor *web*: ter uma largura de banda adequada para suportar as aulas presenciais onde haverão vários alunos postando arquivos e fazendo testes ao mesmo tempo.

#### **4.1.7 – Portabilidade**

O ambiente será portátil para qualquer plataforma que suporte: Apache, Perl e MySQL, com suas bibliotecas e módulos integrados.

#### **4.1.8 – Manutenibilidade**

O ambiente deverá obedecer o ciclo de vida do Modelo Incremental, de acordo com o MRDS, sendo assim suficiente para ser um *software* de boa manutenção.

#### **4.1.9 – Instalabilidade**

O ambiente deverá implementar classes que criem tabelas automaticamente na base de dados, para facilitar o uso do ambiente pelos usuários.

### **4.2 – Análise, Projeto e Desenvolvimento do Ambiente**

A partir da análise orientada objetos, buscou-se planejar um conjunto de atividades e procedimentos, aliado aos métodos e a documentação específica de cada fase, que permitissem a construção do ambiente.

O Modelo Dinâmico, que é composto pelos diagramas de seqüências e de estados, não foi implementado.

#### **4.2.1 – Casos de Uso**

Um diagrama de casos de uso representa um conjunto de cenários identificados, que seja útil aos usuários de um sistema. Um caso de uso descreve cada cenário possível para um sistema, composto por seqüências de passos em que há interação entre os usuários e o sistema. E o ator é quem solicita os serviços disponíveis em casos de uso (DIAS, 2000).

Existem dois tipos de Casos de usos:

- a) Caso de Uso de Alto Nível: que apresenta apenas a visão geral do ambiente.
- b) Caso de Uso Detalhado: que apresenta uma visão mais aprofundada de um módulo ou ação no ambiente.

A partir da análise de requisitos é possível modularizar as funcionalidades e criar um caso de uso de alto nível, como representado na (Figura 9).

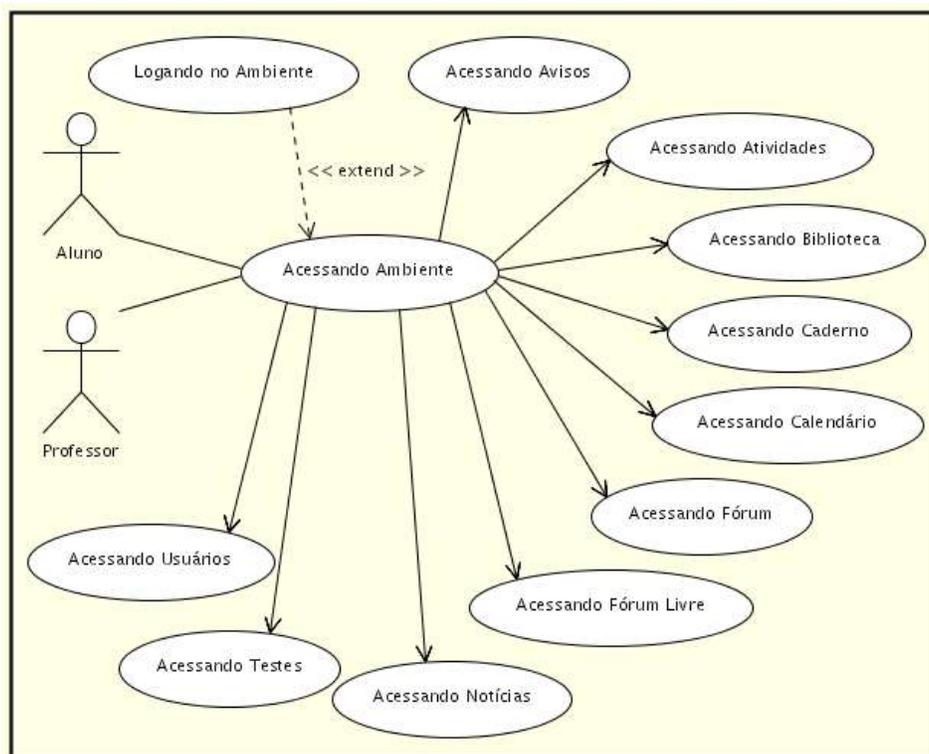
#### **4.2.2 – Diagramas de Classes**

Depois de modularizar o ambiente em diagramas de Caso de Uso de Alto Nível e Detalhado, pode-se criar os Diagramas de Classes de acordo com as Metodologias e padrões UML. As classes representam o empacotamento de uma estrutura de dados com suas rotinas de acesso e se constituem numa forma natural de modularizar o sistema.

De acordo com o MRDS podemos ter vários tipos de classes e diagramas de classes:

a) Classes de Negócio: são as classes do negócio, gerando o principal diagrama que tem a visão geral do negócio.

b) Classes de Interface com o Usuário: são classes que formam interfaces para suas aplicações.



**Figura 9:** Caso de Uso de Alto Nível - GradeNet

c) Classes de Domínio: são classes que formam o modelo canônico, e ficam entre as Entidades da Base de Dados e os Objetos das Classes em execução.

e) Classes de Controle: são classes simples que apenas instanciam as classes principais, iniciando suas operações.

f) Classes Diversas: são outras classes abstratas ou criadas para solucionar problemas não projetados anteriormente.

Na sequência serão apresentados as classes do ambiente em seus respectivos pacotes: na (Figura 10) estão as Classes do Negócio, na (Figura 11) as Classes de Interface, na (Figura 12) as Classes de Domínio, Classes de Controle na (Figura 13) e Classes Diversas na (Figura 14).

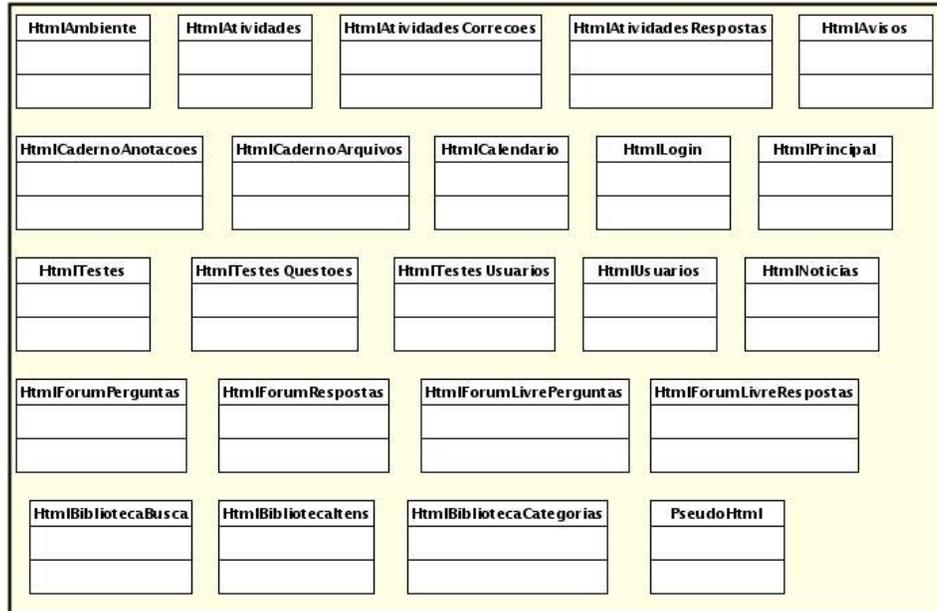
#### **4.2.3 – Diagrama de Entidade-Relacionamento**

É um diagrama voltado para descrição gráfica das entidades e seus relacionamentos, caracterizando uma visão lógica de alto nível dos dados do sistema. A (Figura 15) apresenta o Diagrama ER do Ambiente.

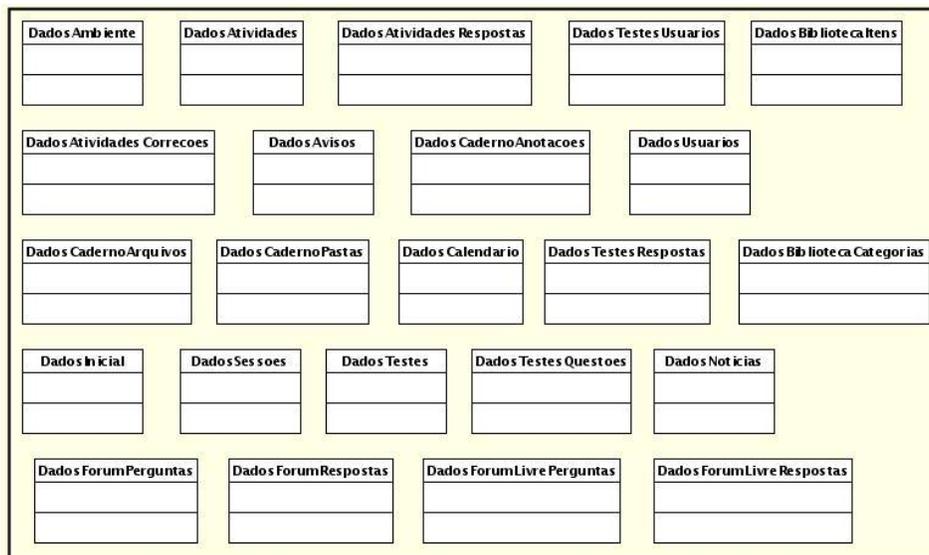
#### **4.3 - Análise, Projeto e Desenvolvimento do Módulo Avisos**

Depois de criar os diagramas principais do GradeNet, já é possível aplicar o MRDS no primeiro módulo, o Módulo de Avisos. Para ficar bem ilustrado nesta seção serão abordado os detalhes de apenas um módulo.

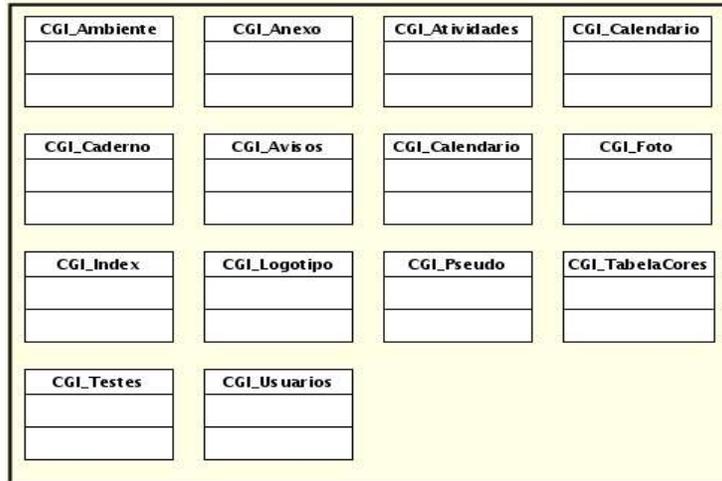




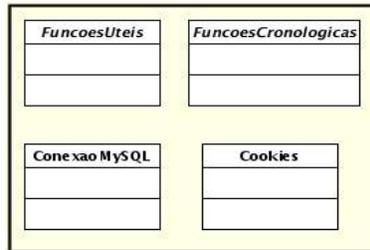
**Figura 11:** Diagrama de Classes de Interfaces com o Usuário - GradeNet



**Figura 12:** Diagrama de Classes de Domínio - GradeNet



**Figura 13:** Diagrama de Classes de Controle - GradeNet



**Figura 14:** Diagrama de Classes Diversas - GradeNet

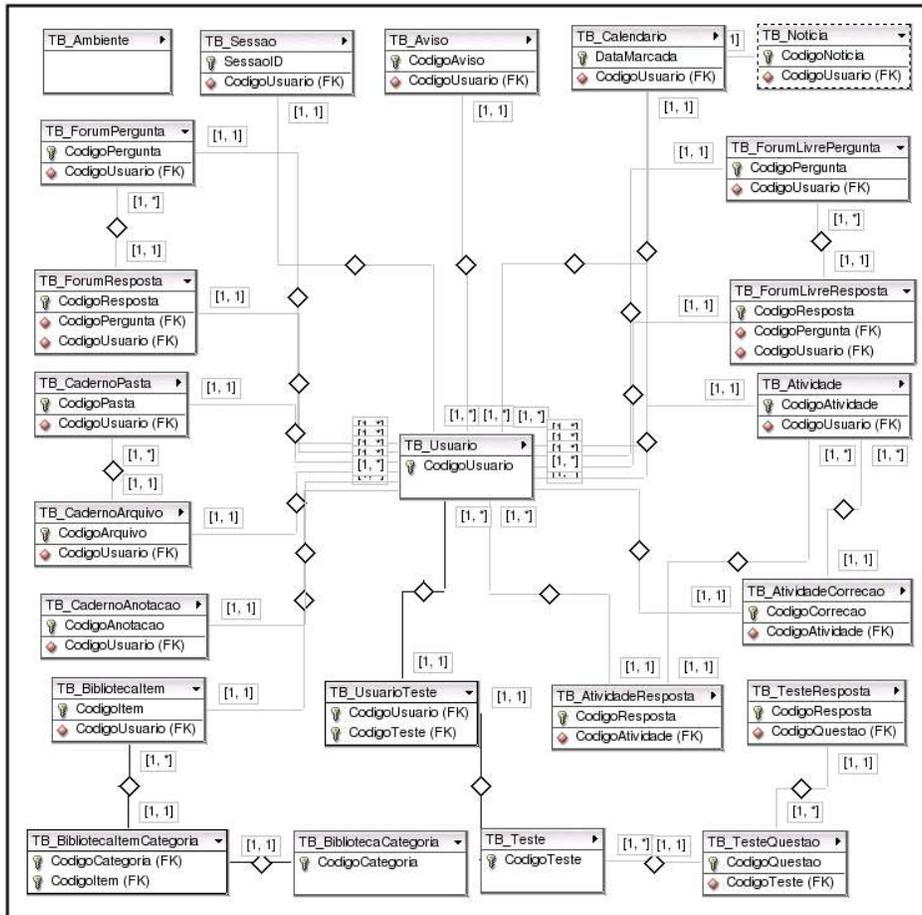
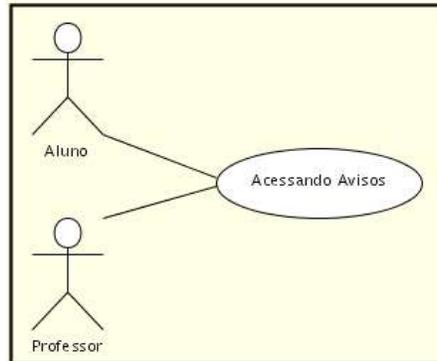


Figura 15: Diagrama Entidade-Relacionamento - GradeNet

### 4.3.1 – Casos de Usos

A partir do caso de uso de alto nível, inicia-se o processo do modelo incremental detalhando o Módulo de Avisos, conforme (Figura 16). Os detalhes dessa figura serão apresentadas na (Tabela 1).



**Figura 16:** Diagrama de Caso de Uso – Módulo Avisos

### 4.3.2 – Diagramas de Classes

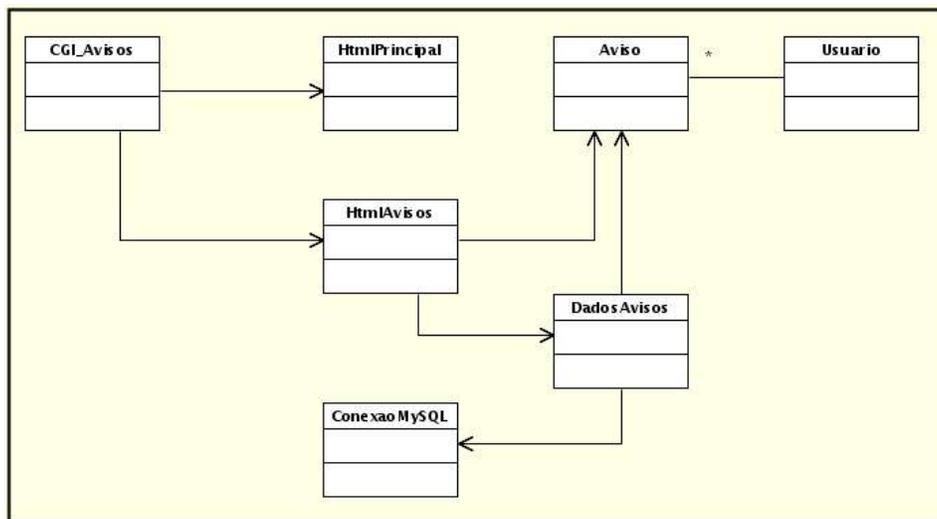
Após a descrição do caso de uso *Acessando Avisos*, tornou fácil a identificação das classes e entidades envolvidas, sendo assim identificadas dentro do diagrama de classes na (Figura 17). E com mais detalhes, apresentando atributos e métodos, estão as classes de Aviso (Figura 18), DadosAvisos (Figura 19) e HtmlAvisos (Figura 20).

### 4.3.3 – Diagrama Entidade-Relacionamento

Após criar um Diagrama ER, basta apenas modificá-lo quando preciso, não necessitando criar novos diagramas para modularização. Nesse caso esta seção apenas apresentará a entidade TB\_Aviso com seus atributos na (Figura 21).

Atores:	Professor e Aluno
Pré-condição:	1 - O ator quer manipular ou ler avisos no ambiente.
Roteiro:	<p>1 – O ambiente apresenta os últimos avisos inclusos no ambiente e cria um sub-menu para alunos (2) ou para professores (3).</p> <p>2 – O aluno poderá ler (3), navegar (4) ou estender (5) os avisos.</p> <p>3 – O ambiente apresentará o conteúdo do aviso permitindo que o professor possa alterar (8) ou excluir (9).</p> <p>4 – O ator poderá navegar através de setas e páginas para os avisos antigos ou mais recentes.</p> <p>5 – O ator poderá estender os avisos para uma melhor leitura.</p> <p>6 – O professor poderá ler (3), navegar (4), estender (5), incluir (7), alterar (8), excluir (9).</p> <p>7 – O professor poderá incluir um novo aviso inserindo as seguintes informações (Título, Texto, Data para vincular ao calendário).</p> <p>8 – O professor poderá alterar as informações do aviso selecionado.</p> <p>9 – O professor poderá confirmar a exclusão do aviso selecionado.</p> <p>10 – O ator poderá escolher outra opção (1)</p>
Pós-condição:	1 - O ator manipulou e/ou leu os avisos no ambiente.
Exceção:	<p>1a - O ator acessou o ambiente com sessão inválida, e deverá ser redirecionado para o caso de uso #Logando no Ambiente.</p> <p>7a/8a – O título do aviso está vazio, o ambiente deverá negar a inclusão ou alteração.</p>

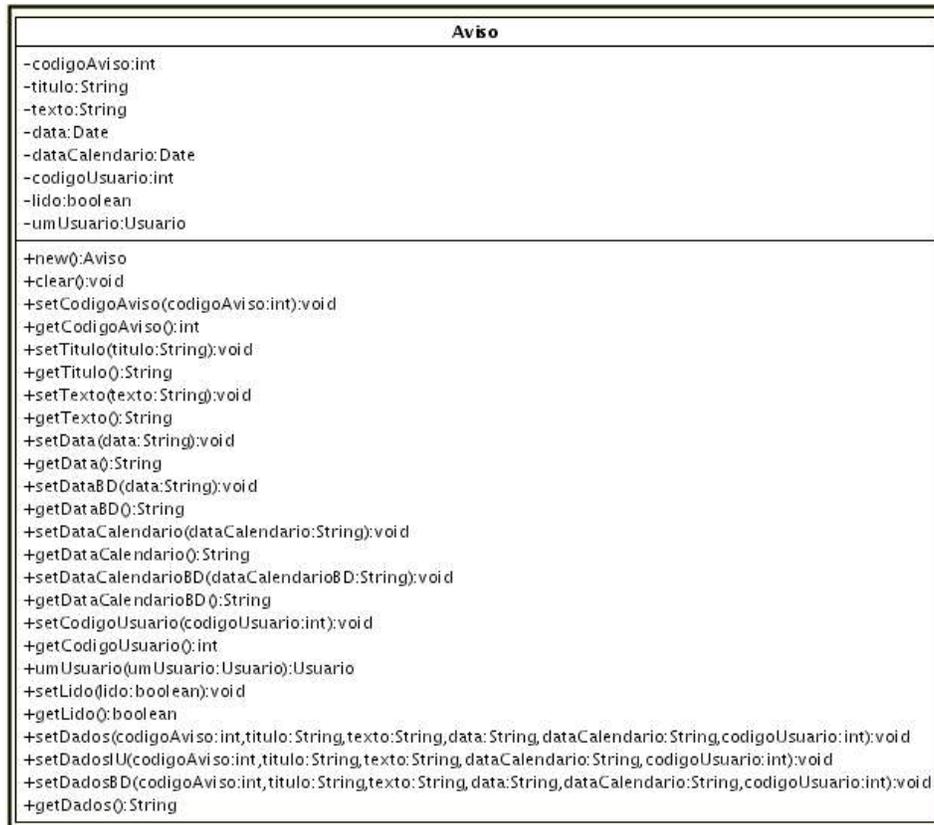
**Tabela 1 – Descrição do Caso de Uso – Acessando Avisos**



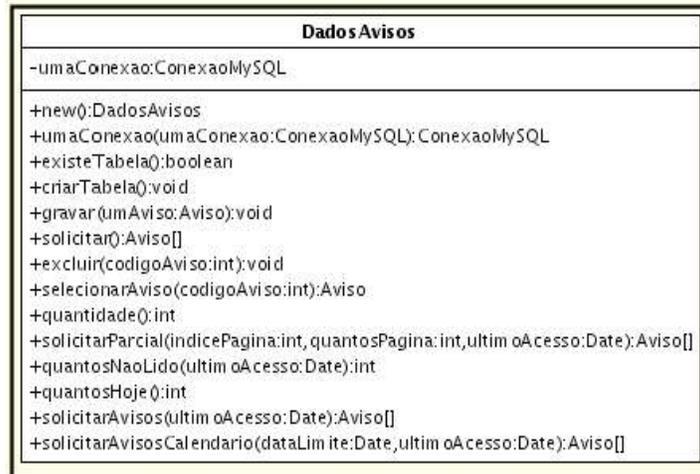
**Figura 17:** Diagrama de Classes – Acessando Avisos

#### 4.3.4 – Codificação

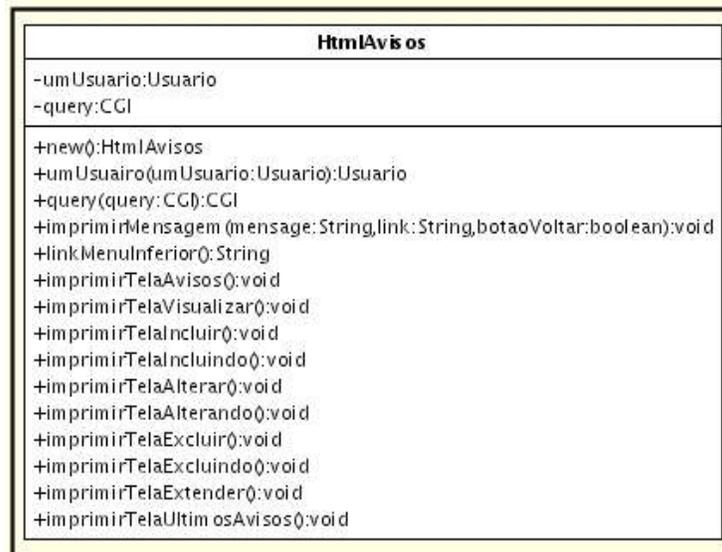
O ambiente foi codificado em Perl. Nesta seção serão apresentados trechos de código de três classes: ClasseAviso.pm (Figura 22), DadosAvisos.pm (Figura 23) e HtmlAvisos (Figura 24):



**Figura 18:** Classe Aviso – Atributos e Métodos



**Figura 19:** Classe DadosAvisos – Atributos e Métodos



**Figura 20:** Classe HtmlAvisos – Atributos e Classes

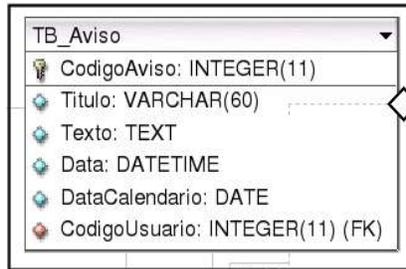


Figura 21: Entidade TB\_Aviso

```

UW PICO(tm) 4.4          File: ClasseAviso.pm
#####
# setTitulo e getTitulo
#####
# métodos para manipular o atributo titulo
#####
sub setTitulo {
    my $self = shift;
    if (@_) {
        my $titulo = &TiraEspacos($_[0]);
        $titulo = &RetiraTags($titulo);
        if (length($titulo) > 0) {
            if (length($titulo) <= 60) {
                $self->{titulo} = $titulo;
            } else {
                die "Título do aviso pode conter no máximo 60 caracteres!\n";
            }
        } else {
            die "Título não pode estar vazio!\n";
        }
    } else {
        die "Título está inválido!\n";
    }
}

sub getTitulo {
    my $self = shift;
    return $self->{titulo};
}
#####
^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Pg     ^X Cut Text     ^C Cur Pos
^X Exit          ^J Justify      ^W Where is    ^V Next Pg     ^U UnCut Text  ^T To Spell

```

Figura 22: ClasseAviso.pm

```

#####
# solicitarParcial
#-----
# Método que solicita parcialmente os avisos da base de dados.
#####
sub solicitarParcial {
  my $self = shift;
  my $indicePagina = $_[0];
  my $quantosPagina = $_[1];
  my $ultimoAcesso = $_[2];
  my $posicao = ($indicePagina-1) * $quantosPagina;
  my $tSQL = "SELECT *, (Data < \'$ultimoAcesso\') AS Lido ";
  $tSQL .= "FROM ". $self->umaConexao->getTab(). 'Aviso as A ";
  $tSQL = $tSQL. "INNER JOIN ". $self->umaConexao->getTab(). "Usuario as U ";
  $tSQL = $tSQL. "ON A.CodigoUsuario = U.CodigoUsuario ";
  $tSQL .= "ORDER BY A.Data Desc LIMIT $posicao, $quantosPagina ";
  my @colecacaoAvisos = [];
  my $SQL = $self->umaConexao->executar->prepare($tSQL);
  $SQL->execute or die "Problemas ao selecionar avisos!\n";
  my $i = 0;
  while (my $r = $SQL->fetchrow_hashref) {
    $colecacaoAvisos[$i] = new ClasseAviso;
    $colecacaoAvisos[$i]->setDadosBD($r->{'CodigoAviso'}, $r->{'Titulo'}, $r->{'Texto'});
    $r->{'Data'}, $r->{'DataCalendario'}, $r->{'CodigoUsuario'});
    $colecacaoAvisos[$i]->unUsuario->setCodigoUsuario($r->{'CodigoUsuario'});
    $colecacaoAvisos[$i]->unUsuario->setNome($r->{'Nome'});
    $colecacaoAvisos[$i]->setLido($r->{'Lido'});
    $i++;
  }
  $SQL->finish;
  return @colecacaoAvisos;
}
#####
DadosAvisos.pm lines 292-324/478 67%

```

Figura 23: DadosAvisos.pm

```

      <td height="2">&nbsp;</td>
    </tr>
  </tr>
  <td>
    <table width="746" border="0" cellspacing="2" cellpadding
="0">
      l;
#####
HTML
my $itemAtual = $intervaloIni;
my $titulo = "";
my $iconeAviso = "";
if ($totalItens > 0) {
  foreach $umAviso (@colecacaoAvisos) {
    if ($umAviso->getLido()) {
      $titulo = $umAviso->getTitulo();
      $iconeAviso = "icones/avisos_pb.jpg";
    } else {
      $titulo = "<b>". $umAviso->getTitulo(). "</b>";
      $iconeAviso = "icones/avisos.jpg";
    }
  }
#####
HTML
print qq |
      <tr>
        <td width="25">
          <div align="right"></div>
        </td>
        <td width="30">
          <div align="center"><font face="Arial, Helvetica, s
#####
HtmlAvisos.pm lines 337-367/1708 16%

```

Figura 24: HtmlAvisos.pm

## **5 – APRESENTANDO O GRADENET**

Neste capítulo será apresentado o resultado de todo o processo de desenvolvimento e motivo desta documentação: o Ambiente Virtual de Apoio ao Ensino, chamado GradeNet.

### **5.1 – Situação Atual**

Por seguir a metodologia MRDS, vários módulos do GradeNet estão sendo desenvolvidos e modificados no decorrer do ano letivo. O ambiente não está pronto em sua totalidade, mas suas principais funcionalidades já estão implementadas.

O ambiente está sendo aplicado e testado em 17 disciplinas da UNICENTRO no ano de 2004. Esse processo levará à adequar o ambiente ao ensino presencial. Faltam apenas quatro módulos (Biblioteca, Fórum, Fórum Livre e Notícias) para que o processo de MRDS esteja completo desde sua criação. Após isso será implementado o Modelo Incremental, cada vez que se tornar necessária a manutenção no ambiente.

### **5.2 – A Página do Projeto**

A página principal (Figura 25) do *site* <[www.gradenet.com.br](http://www.gradenet.com.br)> apenas apresenta o projeto. Explica o funcionamento do ambiente, vinculando às disciplinas nas quais já está em uso.



Figura 25: Página do Projeto

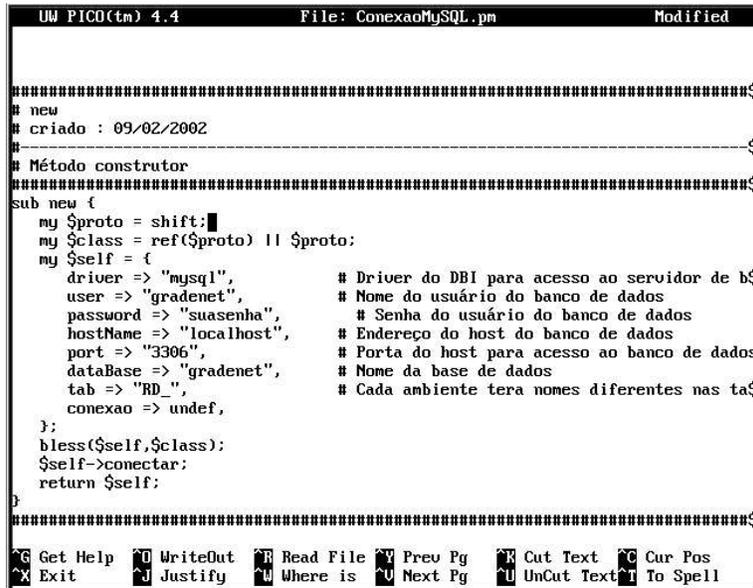
### 5.3 – Instalando o Ambiente

O ambiente pode ser instalado ou atualizado, apenas copiando os arquivos corretamente para a pasta do servidor *web* onde está localizado *site*, tornando necessário apenas modificar os valores de atributos da classe *ConexaoMySQL.pm* (Figura 26):

- a) *user*: usuário na base MySQL;
- b) *password*: a senha deste usuário;
- c) *dataBase*: o nome da base de dados;
- d) *tab*: sigla, para o nome da disciplina.

Ao definir um valor ao atributo *tab*, note que será possível ter mais de uma disciplina dentro da mesma base de dados, ou seja, o ambiente poderá ser instalado mais de uma vez com apenas uma conta em um servidor *web*, e ainda, sem necessitar da senha de superusuário.

Após atribuir as informações necessárias, basta acessar o *site* e todas as tabelas serão criadas automaticamente.



```
UW PICO(tm) 4.4      File: ConexaoMySQL.pm      Modified
#####
# new
# criado : 09/02/2002
#####
# Método construtor
#####
sub new {
  my $proto = shift;
  my $class = ref($proto) || $proto;
  my $self = {
    driver => "mysql",          # Driver do DBI para acesso ao servidor de b$
    user => "gradenet",        # Nome do usuário do banco de dados
    password => "suasenha",    # Senha do usuário do banco de dados
    hostName => "localhost",    # Endereço do host do banco de dados
    port => "3306",            # Porta do host para acesso ao banco de dados
    dataBase => "gradenet",    # Nome da base de dados
    tab => "RD_",              # Cada ambiente tera nomes diferentes nas ta$
    conexao => undef,
  };
  bless($self,$class);
  $self->conectar;
  return $self;
}
#####
G Get Help      W WriteOut      R Read File      Y Prev Pg      K Cut Text      C Cur Pos
X Exit         J Justify       U Where is      U Next Pg      U UnCut Text   T To Spell
```

Figura 26: ConexaoMySQL.pm

#### 5.4 – Acessando o Ambiente

O ambiente simplifica o acesso listando todos os *logins* dentro de uma *combo*. E o aluno poderá saber antecipadamente se já existe alguma informação de grande importância no ambiente, com os avisos logo na página inicial (Figura 27).

#### 5.5 – Módulo Avisos

O Módulo de Avisos (Figura 28) é o primeiro a ser acessado, após o *login*, e também é o principal meio de veiculação de informações importantes dentro do ambiente. O professor ainda tem a opção de cadastrar o aviso e vinculá-lo ao calendário.



**Figura 27:** Acessando o Ambiente GradeNet.

## 5.6 – Módulo Ambiente

Após a instalação do Ambiente é necessário acessar o Módulo Ambiente para a configuração do mesmo, sendo possível trocar a cor de fundo, e modificar informações gerais sobre a disciplina. Basta o professor confirmar e o ambiente se modificará.

## 5.7 - Módulo Atividades

O Módulo Atividades (Figura 29) permite ao professor veicular exercícios e trabalhos para seus alunos. Os alunos tem uma visualização das atividades e seus *status* em relação às datas e respostas. Este módulo é considerado um dos mais importantes, pois é a parte onde é mais exigida a dedicação dos alunos.

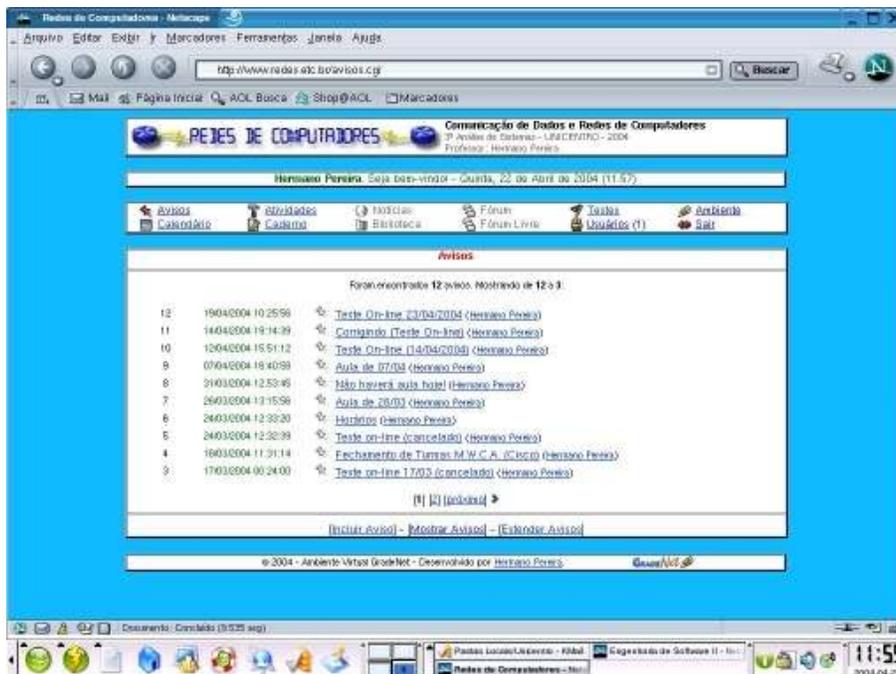


Figura 28: Acessando Avisos no GradeNet.

## 5.8 – Módulo Caderno

O Módulo Caderno (Figura 30) é um módulo pessoal de cada usuário, onde poderão fazer suas anotações, e guardar arquivos. Esses arquivos podem ser publicados através de *pseudo*-HTML em qualquer parte do ambiente.

## 5.9 – Módulo Calendário

O Módulo Calendário (Figura 31) fornece uma visualização de um calendário real com todos os meses do ano, destacando as datas que tem vínculo aos avisos ou atividades.

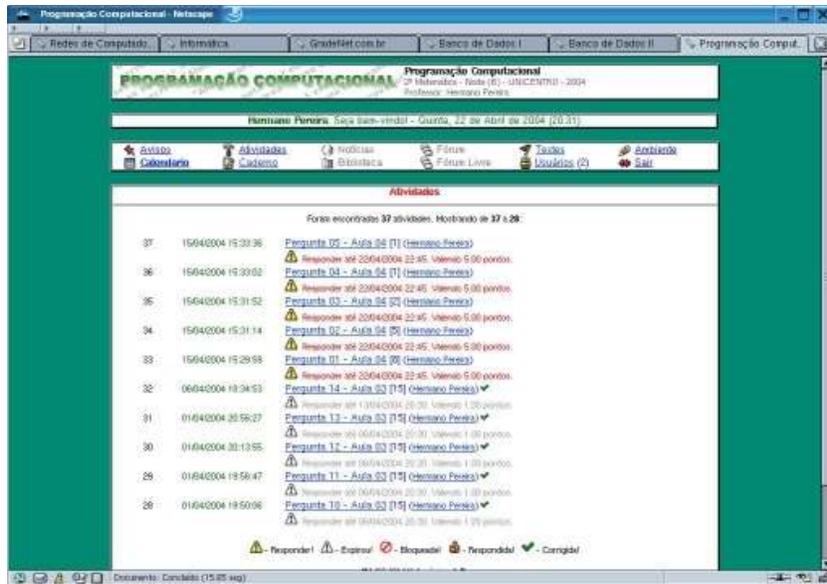


Figura 29: Visualizando Atividades no GradeNet

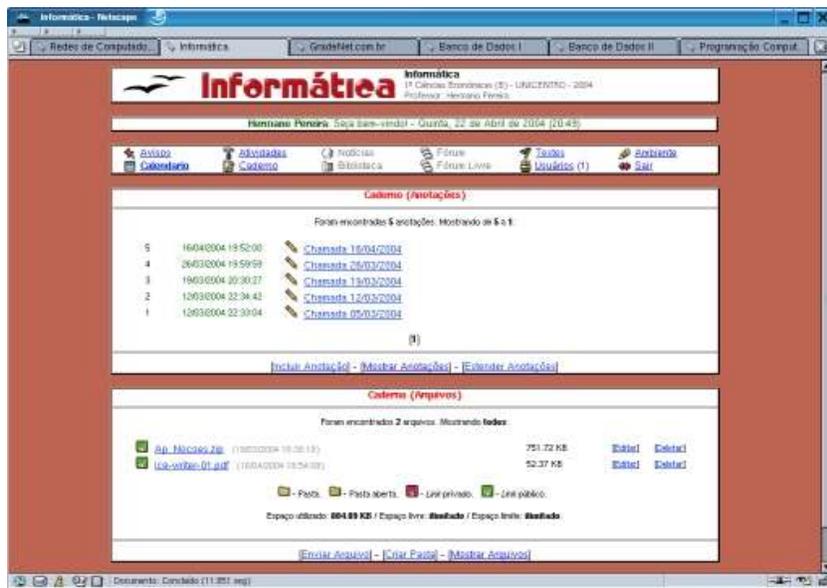
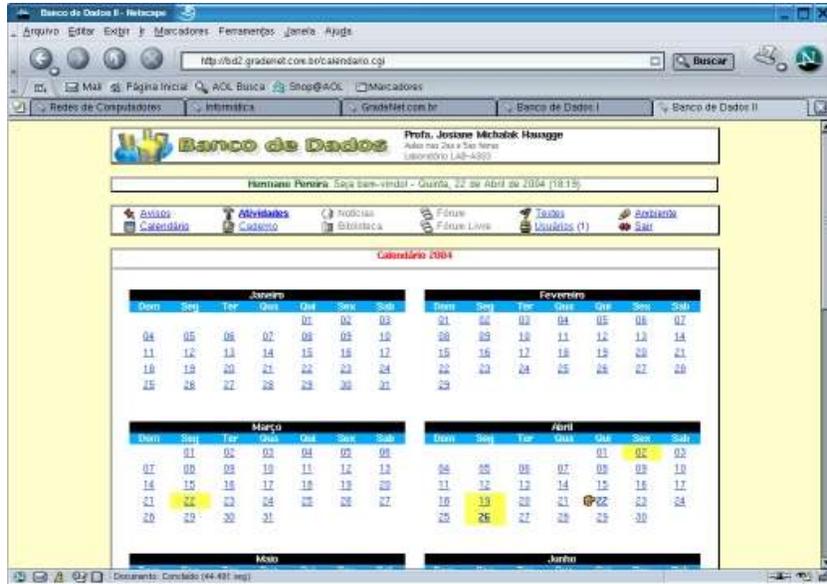


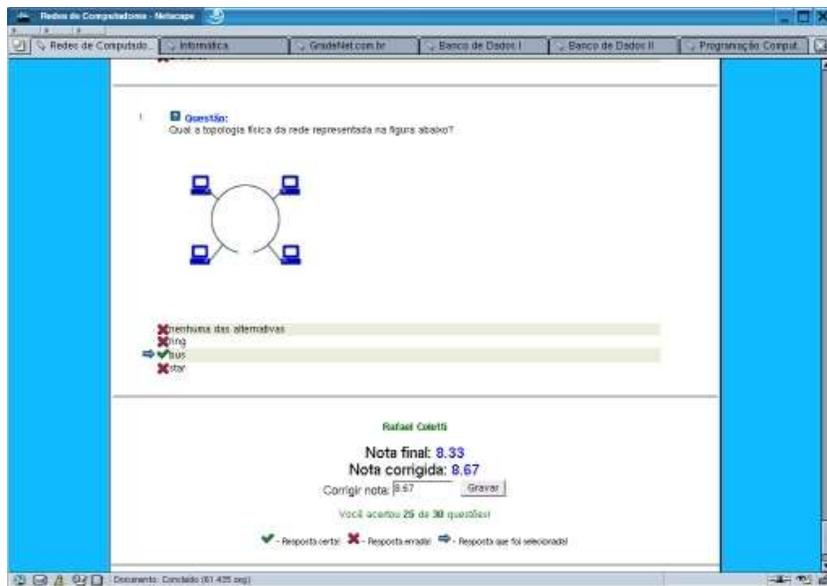
Figura 30: Acessando o Caderno no GradeNet



**Figura 31:** Visualizando o Calendário no GradeNet

## 5.10 – Módulo Testes

O Módulo de Testes permite ao professor cadastrar questões com uma resposta certa e até nove respostas erradas. Depois de cadastrar as questões, é possível gerenciar o testes, solicitando que as questões e as respostas randômicas, e sejam gerados para os alunos. Cada aluno terá um teste de ordem diferente. O próprio módulo fará a correção do teste e apresentará a nota ao aluno (Figura 32). O professor pode habilitar apresentar os resultados em um *ranking* geral.



**Figura 32:** Visualizando um Teste Avaliado no GradeNet

### 5.11 – Módulo Usuários

O Módulo Usuários é o módulo onde o professor poderá cadastrar alunos e outros professores, gerenciar seus acessos, e saber se o mesmo está *on-line* ou não. A partir do cadastro os usuários tem acesso às informações tais como perfil do colega, lista de *e-mails*, álbum com fotos.

### 5.12 – Módulos em Desenvolvimento

No processo de desenvolvimento utilizando o MRDS, os módulos Biblioteca, Fórum, Fórum Livre e Notícias ainda serão implementados. A requisição maior é pelo Fórum Livre, que será o próximo módulo a ser desenvolvido e incorporado ao ambiente.



Figura 33: Acessando o Módulo Usuários do GradeNet



## 6 – CONCLUSÃO

Os ambientes virtuais de ensino apresentados não são suficientes, ou pelo menos, não são adequadas para as aulas presenciais. Muitas de suas funcionalidades são dispensáveis, e faltam outras que são importantes para o dia-a-dia em sala de aula.

Esta monografia apresentou o Ambiente Virtual GradeNet voltado ao ensino presencial, e já tem resultados positivos. Várias disciplinas dentro da UNICENTRO estão usufruindo da versão 3.0. Alunos e professores estão contribuindo com idéias para esta e para a próxima versão.

Como parte da experiência em desenvolvimento de *software* e de sua aplicação prática, o ambiente poderá melhorar na próxima versão:

- a) Novas sugestões de professores e alunos em ambientes diferentes de aula, sempre trazem novidades para o mesmo, mas é bom lembrar que o ambiente deve manter sua simplicidade.
- b) Estudos para migrar de linguagem tais como Java utilizando JSP (*Java Server Pages*), para permitir uma melhor manutenibilidade.
- c) Estudos para migrar de banco de dados tais como PostgreSQL para simplificar implementações que podem ser feitas a nível de SGBD.
- d) Documentar e modificar o código-fonte para internacionalização e principalmente para ser disponibilizado sob GPL.

## REFERÊNCIAS

AMBLER, S. W.: *Análise e Projeto Orientados a Objeto*. Rio de Janeiro: Editora Infobook, 1997.

CERCEAU, A. D.: *Formação a Distância de Recursos Humanos para Informática Educativa*. 1998. 129f. Dissertação de Mestrado – UNICAMP, Campinas.

CCNA: *Demonstração Curriculum CCNA*. Disponível em <[http://cisco.netacad.net/public/academy/catalog/demos/demo\\_portuguese/start.html](http://cisco.netacad.net/public/academy/catalog/demos/demo_portuguese/start.html)>. Acesso em 17 abr. 2004.

CNAMS: *O que é CNAMS?* Disponível em <[http://www.cisco.com/global/BR/netacad/perfil\\_academias.shtml](http://www.cisco.com/global/BR/netacad/perfil_academias.shtml)>. Acesso em 17 abr. 2004.

DEITEL, H.M.: *Perl – Como Programar*. Porto Alegre: Bookman, 2002

DIAS, A. S.: *Delphi & MySQL*. Rio de Janeiro: Editora Ciência Moderna, 2000.

FURLAN, J. D.: *Modelagem de Objetos através da UML – the Unified Modeling Language*. São Paulo: Editora Makron Books, 1998.

GUELICH, S.; GUNDAVARAM, S.; BIRZNIEKS, G.: *Programação CGI com Perl*. Rio de Janeiro: Editora Ciência Moderna, 2001.

HARRIGTON, J. L.: *Projetos de Bancos de Dados Relacionais: teoria e prática*. Rio de Janeiro: Editora Campus, 2002.

HUGHES: *Hughes Technologies: the home of Mini SQL (mSQL)*. Disponível em <<http://www.hughes.com.au>>. Acesso em 20 abr. 2004.

INTERACT: *About Interact*. Disponível em <<http://cce-interact.sourceforge.net>>. Acesso em 17 abr. 2004.

JONES, D. S.: *Free Icons*. Disponível em <dean@gallant.com>. Acesso em 15 jan. 2004.

KROENKE, D. M.: *Banco de Dados – Fundamentos, Projeto e Implementação*. 6.ed. Rio de Janeiro: LTC, 1998.

LAUDON, K. C.: LAUDON, J. P.: *Sistemas de Informação*. 4. ed. Rio de Janeiro: LTC, 1999.

MANHATTAN Virtual Classroom: *What is Manhattan?* Disponível em <<http://manhattan.sourceforge.net>>. Acesso em 15 abr. 2004.

MYSQL: *MySQL – The World's Most Popular Open Source Database* Disponível em <<http://www.mysql.com>>. Acesso em 20 abr. 2004.

NETACAD: *Academy Connection*. Disponível em <<http://cisco.netacad.net/public/index.html>>. Acesso em 17 abr. 2004.

OTSUKA, J. L.; ROCHA, H. V.: *Avaliação Formativa em Ambientes de EaD*. In: XIII Simpósio Brasileiro de Informática na Educação (SBIE 2002). São Leopoldo, 12-14 de novembro, 2002.

PERL: *Perl.com – The source of Perl*. Disponível em <<http://www.perl.com>>. Acesso em 20 abr. 2004.

PRESSMAN, R. S.: *Engenharia de Software*. São Paulo: Editora Makron Books do Brasil, 1995.

RITCHEY, T.: *Programando Java e JavaScript para Netscape 2.0*. São Paulo: Quark Editora, 1996.

ROCHA, H. V.: *O Ambiente TelEduc para Educação a Distância baseada na Web: Princípios, Funcionalidades e Perspectivas de Desenvolvimento*. Apud: Moraes, M.C.: Educação a Distância: Fundamentos e práticas. Campinas, 2002.

SCHMITZ, E. A.; SILVEIRA, D. S. *Desenvolvimento de Software Orientado a Objetos Utilizando UML e Delphi 5*. Rio de Janeiro: Editora Brasport, 2000.

STOCO, L. M.: *Integrando PHP com MySQL - Guia de Consulta Rápida*. São Paulo: Novatec Editora, 2000.

TANENBAUM, A. S.: *Redes de Computadores*. Rio de Janeiro: Editora Campus, 1997.

TELEDUC: *Ambiente de Ensino a Distância*. Disponível em <<http://teleduc.nied.unicamp.br>>. Acesso em 16 abr. 2004.

W3C: *Introducion to HTML 4*. Disponível em <<http://www.w3.org/TR/REC-html40/intro/intro.html>>. Acesso em 23 abr. 2004.

## APÊNDICE A - LISTA DE ABREVIATURAS E SIGLAS

ANSI	<i>American National Standards Institute</i>
ARL	<i>Administração de Redes Linux</i>
BSD	<i>Berkeley Software Distribution</i>
CASE	<i>Computer Aided Software Engineering</i>
CCNA	<i>Cisco Certified Network Associate</i>
CERN	<i>European Organization for Nuclear Research</i>
CGI	<i>Common Gateway Interface</i>
CNAMs	<i>Cisco Networking Academy Management System</i>
CNAP	<i>Cisco Networking Academy Program</i>
EaD	<i>Ensino a Distância</i>
ER	<i>Entidade-Relacionamento</i>
GPL	<i>GNU Public License</i>
HTML	<i>HyperText Markup Language</i>
IC	<i>Instituto de Computação</i>
IETF	<i>Internet Engineering Task Force</i>
ISO	<i>International Organization for Standardization</i>
JSP	<i>Java Server Pages</i>
LIFO	<i>Last In First Out</i>
MRDS	<i>Metodologia Rápida de Desenvolvimento de Software</i>
NIED	<i>Núcleo de Informática Aplicada à Educação</i>
NCSA	<i>National Center for Supercomputing Applications</i>
NSA	<i>National Security Agency</i>
OMG	<i>Object Management Group</i>
OMT	<i>Object Modeling Technique</i>
OOSE	<i>Object Oriented Software Engineering</i>
PC	<i>Personal Computer</i>
PERL	<i>Practical Extraction and Report Language</i>
PHP	<i>Personal Home Page</i>
PM	<i>Perl Module</i>
SGBD	<i>Sistema Gerenciador de Banco de Dados</i>
SQL	<i>Structured Query Language</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
UFLA	<i>Universidade Federal de Lavras</i>
UML	<i>Unified Modeling Language</i>
UNICAMP	<i>Universidade Estadual de Campinas</i>
UNICENTRO	<i>Universidade Estadual do Centro-Oeste</i>
WWW	<i>World Wide Web</i>
W3C	<i>World Wide Web Consortium</i>