

UNIVERSIDADE FEDERAL DO PARÁ
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JOSÉ ANTÔNIO SARMANHO DOS SANTOS FREIRE

IETS
FERRAMENTA INTERATIVA DE ENSINO DE SQL

DM – 12/2006

UFPA / CT / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2006

**UNIVERSIDADE FEDERAL DO PARÁ
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

JOSÉ ANTÔNIO SARMANHO DOS SANTOS FREIRE

IETS

FERRAMENTA INTERATIVA DE ENSINO DE SQL

Dissertação
submetida à Banca
Examinadora do
Programa de Pós-
Graduação em
Engenharia Elétrica da
UFPA para obtenção
do Grau de Mestre em
Engenharia Elétrica

UFPA / CT / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2006

F866i Freire, José Antônio Sarmanho dos Santos

IETS : ferramenta interativa de ensino de SQL / José Antônio Sarmanho dos Santos Freire, orientador, Eloi Luiz Favero.-2006.

Dissertação (Mestrado) – Universidade Federal do Pará, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2006.

1. Ensino à distância – programação. 2. SQL (Linguagem de programação de computador). 3. Banco de dados relacional. I. Título.

CDD – 21. ed. 371.334

UNIVERSIDADE FEDERAL DO PARÁ
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

IETS

FERRAMENTA INTERATIVA DE ENSINO DE SQL

AUTOR: JOSÉ ANTÔNIO SARMANHO DOS SANTOS FREIRE

DISSERTAÇÃO DE MESTRADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DO PARÁ E JULGADA ADEQUADA PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM 24/03/2006

BANCA EXAMINADORA:

Prof. Dr. Eloi Luiz Favero
(ORIENTADOR – UFPA)

Profa. Dra. Janne Yukiko Yoshikawa Oeiras
(MEMBRO-UFPA)

Prof. Dr. Roberto Célio Limão de Oliveira
(MEMBRO-UFPA)

VISTO:

Prof. Dr. Evaldo Gonçalves Pelaes
(COORDENADOR DO PPGE / CT / UFPA)

PPGE / CT / UFPA

Em memória do meu pai João Freire, um grande homem, incentivador das minhas pesquisas relacionadas a educação.

A minha mãe, Francisca Freire, uma lutadora, que ensinou como conquistar o conhecimento com humildade e força.

A minha esposa Cássia, que me apoiou nos altos e baixos desta jornada da construção do conhecimento.

AGRADECIMENTOS

Agradecimento acima de tudo a Deus, por tornar este sonho uma realidade na minha carreira docente.

Ao professor Dr. Eloi Luiz Favero do PPGEE/CT/UFPA, pela incansável orientação dispensada no trabalho desenvolvido, sempre mostrando os caminhos para o aperfeiçoamento do projeto em questão.

As professoras e minhas grandes amigas Doutorandas Silvana Rossy (IESAM) e Aleksandra Silva (UNAMA), pelo apoio incondicional dispensado durante a construção da ferramenta ora desenvolvida, mostrando uma grande experiência sobre a área de educação à distância.

Ao grupo de pesquisa GPED da UPFA, aonde faço parte, juntamente com os colaboradores Silvana Rossy, Aleksandra Silva, Maria da Penha, Osiel Marlon Negrão, Adriano Del Pino, dentre outros, trabalhando em pesquisa na área de informática na educação, principalmente no ensino a distância (EAD).

Aos meus amigos Ricardo Ferreira, Mauro Souza, Almir Monteiro, chefes e companheiros da Universidade da Amazônia, por me apoiarem, dando o devido incentivo para desenvolver todas as atividades relacionadas aos interesses do mestrado.

SIMPLES, COMO DEVE SER VIVIDA A VIDA.

“O valor das coisas não está no tempo que elas duram, mas na intensidade com que acontecem. Por isso existem momentos inesquecíveis, coisas inexplicáveis e pessoas incomparáveis. Aproveite bem o tempo. Já é bem mais tarde do que pensa.”

FERNANDO PESSOA

SUMÁRIO

I - INTRODUÇÃO	1
1.1 INTRODUÇÃO	1
1.1.1 <i>Dificuldades no ensino de programação.....</i>	<i>1</i>
1.1.2 <i>Problemas com ambientes para EAD em larga escala</i>	<i>2</i>
1.1.3 <i>Soluções em andamento.....</i>	<i>3</i>
1.2 OBJETIVOS	4
1.3 ORGANIZAÇÃO DO TEXTO.....	5
II - TRABALHOS RELACIONADOS.....	7
2.1 SQLATOR (2004) – AN ONLINE SQL LEARNING WORKBENCH	7
2.2 ASSESQL (2004) – AN ONLINE TOOL TO TEST SQL QUERY FORMULATION SKILLS	8
2.3 SQL-TUTOR (1998).....	10
2.4 ANÁLISE DOS TRABALHOS RELACIONADOS	12
III - IETS – VISÃO CONCEITUAL.....	14
3.1 CONTEXTO: PROJETO AMAM.....	14
3.2 ORGANIZAÇÃO DO CONTEÚDO.....	17
3.3 A INTERATIVIDADE NOS TEMAS DE PROGRAMAÇÃO	18
3.4 A ANÁLISE INTELIGENTE DAS ATIVIDADES DE PROGRAMAÇÃO.....	20
3.5 A INTERAÇÃO ALUNO & PROFESSOR	21
IV - PROJETO E IMPLEMENTAÇÃO DO PROTÓTIPO	23
4.1 REQUISITOS	23
4.2 CASOS DE USO.....	24
4.2.1 <i>Caso de Uso Principal.....</i>	<i>25</i>
4.2.2 <i>Casos de Uso por Funcionalidade.....</i>	<i>26</i>
4.2.3 <i>Casos de Uso por Ator.....</i>	<i>28</i>
4.3 PROJETO.....	31
4.3.1 <i>Diagrama de Classes.....</i>	<i>31</i>
4.3.2 <i>Arquitetura do Sistema</i>	<i>32</i>
4.4 IMPLEMENTAÇÃO	35
4.4.1 <i>Exemplo de Código para Classe Cadastro de Curso.....</i>	<i>35</i>
4.4.2 <i>Projeto de Banco de Dados</i>	<i>36</i>
4.5 AS TECNOLOGIAS UTILIZADAS NA FERRAMENTA.....	41
V – AVALIAÇÃO DOS RESULTADOS.....	42
5.1 OBJETIVOS DA PROPOSTA.....	42
5.2 PROTÓTIPO INICIAL	43
5.3 PROTÓTIPO COM AVALIAÇÃO “APROXIMADA” AUTOMÁTICA DE CONSULTAS SQL.....	44
5.4 PROTÓTIPO COM AVALIAÇÃO AUTOMÁTICA APROXIMADA DE CONSULTAS NÃO OBJETIVAS	44
5.4.1 <i>Motivação.....</i>	<i>45</i>

5.4.2	<i>O modelo vetorial para recuperação de informação</i>	45
5.4.3	<i>Conclusão do capítulo</i>	46
VI	- CONCLUSÃO	47
	REFERÊNCIAS BIBLIOGRÁFICAS	49
	APÊNDICE A – CÓDIGOS-FONTES DA FERRAMENTA	53
	APÊNDICE B – DIAGRAMAS DE CLASSES DO AMAM	63
	APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO DO IETS	72
	APÊNDICE D – PUBLICAÇÃO RELACIONADA COM A DISSERTAÇÃO	81

LISTA DE ILUSTRAÇÕES

<i>Figura 2.1 Formulário de Avaliação do SQLator</i>	8
<i>Figura 2.2 AsseSQL: Formulário Principal: Resultado das Tentativas nas Questões</i>	9
<i>Figura 2.3 A análise de saída do SQL-Tutor mostrando o retorno a ser dado ao estudante</i>	10
<i>Figura 2.4 Inabilidade do SQL-Tutor lidar com alguns erros semânticos</i>	11
<i>Figura 2.5 Formulário Principal do SQL-Tutor Web</i>	12
<i>Figura 3.1 AmAm (Ambiente de Aprendizagem Multiparadigmático)</i>	14
<i>Figura 3.2 Contexto de Implementação do IETS</i>	15
<i>Figura 3.3 Apresentação de Exemplo/Conceito detalhe</i>	19
<i>Figura 3.4 Tipo de Exercício realizado na Ferramenta</i>	20
<i>Figura 3.5 Ambiente do Professor (Comentando respostas/Gerando Avaliação)</i>	22
<i>Figura 4.1 Fluxo das atividades de desenvolvimento</i>	23
<i>Figura 4.2 Diagrama de Caso de Uso Principal</i>	25
<i>Figura 4.3 Diagrama de Casos de Uso Referente ao Ator Aluno</i>	28
<i>Figura 4.4 Diagrama de Casos de Uso Referente ao Ator Professor</i>	29
<i>Figura 4.5 Diagrama de Casos de Uso Referente ao Ator Administrador</i>	30
<i>Figura 4.6 Diagrama de classes da ferramenta IETS</i>	32
<i>Figura 4.7 Esquema de interação dos componentes na Ferramenta IETS</i>	33
<i>Figura 4.8 Arquitetura do IETS</i>	34
<i>Figura 4.9 Modelo Físico do Banco de Dados IETS</i>	40
<i>Figura B.1 Diagrama de Classe da Gerência de Dados Institucionais</i>	64
<i>Figura B.1 Diagrama de Classe Geral das Ferramentas – parte I</i>	65
<i>Figura B.3 Diagrama de Classe Geral das Ferramentas – parte II</i>	66
<i>Figura B.2 Diagrama de Classe das Ferramentas do Mural</i>	67
<i>Figura B.3 Diagrama de Classe da Ferramenta Fórum</i>	68
<i>Figura B.4 Diagrama de Classe da Ferramenta Informe</i>	69
<i>Figura B.5 Diagrama de Classe da Ferramenta Correio</i>	70
<i>Figura B.6 Diagrama de Classe da Ferramenta FAQ</i>	71

LISTA DE TABELAS

<i>Tabela 3.1</i>	<i>Possibilidade de organização de uma sessão</i>	18
<i>Tabela 3.2</i>	<i>Exemplos de soluções equivalentes com diferentes métricas</i>	21
<i>Tabela 4.1</i>	<i>Requisitos Funcionais do Sistema</i>	26
<i>Tabela 4.2</i>	<i>UC04: Cadastrar Curso</i>	27
<i>Tabela 4.3</i>	<i>Tabelas utilizadas pelo Ambiente AmAm e ferramenta IETS</i>	37
<i>Tabela 5.1</i>	<i>Resultados da Avaliação</i>	43
<i>Tabela 5.2</i>	<i>Exemplos de soluções equivalentes com diferentes métricas</i>	44
<i>Tabela 5.3</i>	<i>Exemplos de soluções equivalentes com diferentes métricas</i>	46
<i>Tabela C.1</i>	<i>UC01: Autenticar Usuário</i>	72
<i>Tabela C.2</i>	<i>UC02: Cadastrar Aluno</i>	72
<i>Tabela C.3</i>	<i>UC03: Cadastrar Professor</i>	73
<i>Tabela C.4</i>	<i>UC05: Cadastrar Curso</i>	73
<i>Tabela C.5</i>	<i>UC04: Cadastrar Instituição</i>	73
<i>Tabela C.6</i>	<i>UC06: Cadastrar Disciplina</i>	74
<i>Tabela C.7</i>	<i>UC07: Cadastrar Turma</i>	74
<i>Tabela C.8</i>	<i>UC08: Cadastrar Perfil Turma</i>	74
<i>Tabela C.9</i>	<i>UC09: Listar Sumário Disciplina</i>	75
<i>Tabela C.10</i>	<i>UC10: Cadastrar Conteúdo</i>	75
<i>Tabela C.11</i>	<i>UC11: Cadastrar Problema</i>	76
<i>Tabela C.12</i>	<i>UC12: Cadastrar Solução Usuário</i>	76
<i>Tabela C.13</i>	<i>UC13: Gerar Mapeamento Acesso</i>	77
<i>Tabela C.14</i>	<i>UC14: Consultar Desempenho Turma</i>	77
<i>Tabela C.15</i>	<i>UC15: Interagir Aluno</i>	78
<i>Tabela C.16</i>	<i>UC16: Cadastrar / Gerar Avaliação</i>	78
<i>Tabela C.17</i>	<i>UC17: Informar Resposta Problema</i>	79
<i>Tabela C.18</i>	<i>UC18: Exibir Painel Sessão Resposta Aluno</i>	80
<i>Tabela C.19</i>	<i>UC19: Consultar Comentário Professor</i>	80
<i>Tabela C.20</i>	<i>UC20: Exibir Mapa Estatístico Acesso</i>	80

LISTA DE CÓDIGOS-FONTES

<i>Listagem 4.1. Código gerado para a página PHP de Cadastro de Curso</i>	36
<i>Listagem A.1 Código gerado para a página PHP de Cadastro de Curso.....</i>	54
<i>Listagem A.2 Parte do Código da Classe FACHADA</i>	55
<i>Listagem A.3 Código gerado para a classe CadCurso</i>	56
<i>Listagem A.4 Código gerado para a classe Curso.....</i>	57
<i>Listagem A.5 Código gerado para a classe derivada responsável pelas operações básicas de manipulação de dados</i>	58
<i>Listagem A.6 Código gerado para a classe genérica CadBD.....</i>	59
<i>Listagem A.7 Código gerado para classe que realiza a conexão com o banco de dados</i>	62

LISTA DE SIGLAS

AmAm	<i>Ambiente de Aprendizagem Multiparadigmático</i>
API	<i>Application Programing Interface</i>
AVA	<i>Ambiente Virtual de Aprendizagem</i>
EAD	<i>Educação a Distância</i>
E-MAIL	<i>Eletronic Mail</i>
FAQ	<i>Frequently Asked Questions</i>
GPED-NORTE	<i>Grupo de Pesquisa em Educação a Distância do Norte</i>
HTML	<i>Hypertext Mark-up Language</i>
IA	<i>Inteligência Artificial</i>
MySQL	<i>Servidor de banco de dados amplamente difundido na WEB.</i>
PHP	<i>Linguagem de Programação voltada para o ambiente WEB.</i>
SQL	<i>Structured Query Language</i>
TAG	<i>Nome dado a uma marcação do tipo <> dentro de um arquivo HTML</i>
UFPA	<i>Universidade Federal do Pará</i>
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>
WEB	<i>Abreviação de WWW (World Wide WEB)</i>
XML	<i>Extensible Markup Language</i>

RESUMO

Este trabalho propõe uma ferramenta na Web para ensino de programação. O trabalho é baseado em um estudo de caso para ensino da linguagem de programação de bancos de dados SQL e está inserido no contexto de um ambiente interativo de aprendizagem. Esta ferramenta experimental para ensino de linguagens de programação, apresenta como objetivos: 1) completa integração entre o ensino de conceitos de programação com exemplo de fragmentos de programas executáveis on-line; 2) integração com um componente de avaliação de programas verificando se a resposta do aluno é a mesma do professor; 3) monitoramento das atividades do aluno (quantos exemplos foram executados; em cada exercício, quantas tentativas de execução foram feitas, etc). 4) servir como um laboratório para experimentação de duas novas técnicas de avaliação e acompanhamento, que estão sendo pesquisadas em trabalhos em paralelo: (a) avaliação automática de programas com métricas de engenharia de software, permitindo informar o quanto a solução do estudante se aproxima de uma “solução ideal”; e (b) avaliação automática de questões conceituais discursivas, além de permitir as tradicionais perguntas objetivas. Este trabalho é um primeiro passo na direção de construção de um ambiente totalmente assistido (por exemplo com avaliação automática) para ensino da linguagem de programação SQL, onde o professor terá uma carga de trabalho mínima, mais como supervisor.

PALAVRAS-CHAVES:

ensino de programação, SQL, ferramentas de apoio ensino-aprendizagem, exercício-e-prática de programação.

ABSTRACT

This work considers a tool in the Web for programming teaching. The work is based on a study of case for teaching of the programming language of data bases SQL and is inserted in the context of an interactive environment of learning. This experimental tool for teaching of programming languages, presents as objective: 1) complete integration between the teaching of concepts of programming with example of fragments of executable programs on-line; 2) integration with a component of evaluation of programs verifying if the reply of the learner is the same one of the teacher; 3) complete control of the learner's activities (how many examples had been executed; in each exercise, how many execution attempts had been made, etc). 4) to serve as a laboratory for experimentation of two new techniques of evaluation and accompaniment, that are being searched in works in parallel: (a) automatic evaluation of metric programs with of software engineering, allowing informing how much the solution of the student if approaches to a "ideal solution"; e (b) automatic evaluation of discursive conceptual questions, besides allowing the traditional objective questions. This work is a first step in the direction of construction of an environment total attended (for example with automatic evaluation) for teaching of the programming language SQL, where the teacher will have a minimum load of work, more as supervisor.

KEYWORDS:

Teaching of programming, SQL, learning-teaching supporting tools, programming exercise and practice.

I - INTRODUÇÃO

1.1 Introdução

A tecnologia atual vem facilitando a atividade de colecionar e armazenar dados indiscriminadamente, criando o problema de organizá-los e gerenciá-los de forma adequada. A área de bancos de dados visa propor soluções para este problema (Brasil, 1998). Dentre essas soluções, tem sido freqüente a utilização de sistemas de banco de dados, para os quais foi desenvolvida uma linguagem de manipulação de dados que se tornou um padrão de fato - a linguagem SQL (*Structured Query Language*). Com ela, é possível especificar comandos para realizar a pesquisa desejada, como por exemplo, o título de um livro, seu autor, etc.

O ensino de SQL apresenta uma série de desafios:

- a sintaxe e a semântica da linguagem SQL são complexas para estudantes iniciais;
- a habilidade e a competência em programação são adquiridas somente com muita atividade de laboratório;
- o acompanhamento no laboratório sobrecarrega o professor: corrigir os programas, tirar dúvidas, fornecendo um retorno personalizado;
- a dificuldade criada pelas turmas heterogêneas: alguns alunos estão iniciando e outros já são programadores com alguma experiência;
- a dificuldade da educação não presencial: como ensinar SQL para turmas não presenciais.

1.1.1 Dificuldades no ensino de programação

A atividade de programação em geral envolve um grande esforço cognitivo para o estudante e um grande esforço de acompanhamento para o professor. Em quase todos os cursos de programação é comum o uso de monitores para auxiliar o professor no acompanhamento das atividades práticas, principalmente para turmas grandes. Segundo Rodrigues et al. (2000), a

prática de ensino de computação é afetada pela inexistência de uma literatura de referência sobre construção e utilização de metodologias e materiais didáticos para o ensino de computação. Hoje, no Brasil, já temos bons livros para quase todas as disciplinas de programação de um curso de Computação. Entretanto, ambientes de programação completos com recursos para apoiar professores e estudantes ainda estão em desenvolvimento.

Para Castro et al. (2004), a dificuldade é maior para aqueles que ingressam em cursos cujas turmas são heterogêneas, onde alguns estudantes são provenientes de cursos técnicos na área e outros já são programadores que buscam uma educação formal; os primeiros ainda estão iniciando a atividade de programação enquanto que os segundos já são experientes. A questão que se coloca, e que é tratada neste trabalho é: Como conciliar o ensino para estes dois perfis de alunos?

Além disso, problemas como o tempo de prática de laboratório reduzido faz com que, após a realização de uma atividade, o estudante frequentemente não tenha a oportunidade de analisar as tarefas realizadas ou não tenha conseguido formular suas dúvidas e obter respostas. Como consequência, diversos estudantes deixam suas tarefas inacabadas; na próxima aula de laboratório já é outra atividade. Neste sentido, a construção de consultas SQL se torna mais difícil quando o resultado da sua execução realizada pelo aluno contra um banco de dados não é vista: por exemplo, a formulação de consultas em sala de aula ou em testes fora de laboratório. Neste caso, se o estudante é iniciante ele fica dependente do retorno professor/monitor ou de um colega mais experiente.

1.1.2 Problemas com ambientes para EAD em larga escala

Num ambiente de Educação a Distância (EAD), em larga escala, o processo de ensino-aprendizagem da disciplina de Banco de Dados demanda grande esforço por parte dos professores: o elevado número de estudantes para serem acompanhados por um mesmo professor dificulta a mediação da aprendizagem e não atende a necessidade de muitos estudantes em receber feedback imediato de suas ações. Esta sobrecarga de trabalho para o professor na avaliação das atividades dos estudantes remete invariavelmente a uma demora no *feedback* do professor para essas atividades.

1.1.3 Soluções em andamento

Diante desse cenário, a experiência dos profissionais de conteúdo com as práticas de ensino revelam novas formas de apresentar o conteúdo de suas disciplinas, utilizando-se dos modernos recursos de multimídia, de ambientes de EAD e da Internet, entre outros, para alavancar o processo ensino-aprendizagem, agregando novos valores (Moraes et al., 1997). A convergência destas novas tecnologias e metodologias (Souza, 2001) e as novas ferramentas de apoio à instrução e construção de artefatos (produtos desenvolvidos durante o processo de aprendizagem, tal como, documentos), associadas ao material disponível em *sites*, favorecem e facilitam a aprendizagem, visto que permitem ao estudante obter mais informações, além daquelas oferecidas em sala.

Ambientes de EAD possuem ferramentas que auxiliam o professor disponibilizar o material de aula e receber de forma organizada o material dos alunos (ver por exemplo, o Teleduc (2005) e o Aulanet (2005)). Nesses ambientes já existem também ferramentas para fazer avaliação. Por exemplo, as questões objetivas são automaticamente avaliadas e as subjetivas são tipicamente respondidas e manualmente avaliadas. Porém, já existem pesquisas e alguns sistemas comerciais com componentes para avaliar automaticamente questões dissertativas (Shermis, 2003)(Hearst, 2003)(Sukkarieh, 2003), utilizando técnicas tais como regressão múltipla e verificação da similaridade entre dois textos.

Mais especificamente para o ensino de SQL, já existem alguns esforços, tais como: SQLator (Sadiq et al., 2004), SQL-Tutor (Mitrovic, 1998), buscando a construção de ambientes que possibilitem minimizar esse problema: encaminhar tarefas de laboratório para os estudantes e acompanhar estas tarefas. Esses dois sistemas são apresentados mais adiante.

Por outro lado, certas ferramentas destes ambientes permitem lidar com grandes turmas de estudantes: realizando as atividades; recebendo as soluções; avaliando-as; dentro do possível, fornecendo *feedback* imediato às ações do estudante, facilitando as interações e, conseqüentemente, o esclarecimento de dúvidas.

No contexto dos ambientes virtuais para apoiar o ensino presencial e não presencial, o grupo GPED da UFPA (<http://www.deec.ufpa.br/gped/>) está desenvolvendo um ambiente denominado Ambiente de Aprendizagem Multiparadigmático (AmAm), que estabelece uma arquitetura para ambientes virtuais de aprendizagem e o desenvolvimento de um sistema web para apoio ao ensino em diferentes tipos de modalidades utilizando diversas abordagens pedagógicas. O AmAm é desenvolvido a partir de uma arquitetura que propõe o

desenvolvimento de AVA (Ambiente Virtual de Aprendizagem) a partir de agentes, componentes e *framework* (Silva et al., 2003), considerando como princípio básico a integração de todos os aspectos de conhecimento do estudante (Gava, 2002). Nesses ambientes é importante encontrar meios para estimular o estudante a observar seus erros, experimentar e testar novas soluções, formular e registrar suas dúvidas, garantindo feedback para essas ações. Estas tecnologias devem favorecer a análise por parte do estudante, das soluções encontradas.

Na busca por contribuir para o equacionamento das dificuldades dos estudantes na aprendizagem de linguagens de programação de computadores, utilização de softwares, modelagem de problemas, assim como o manuseio de softwares específicos, estamos desenvolvendo um projeto que busca a construção de ambientes telemáticos para apoio à realização de cursos orientados à resolução de problemas (Favero et al., 2003). Nosso foco inicial centra-se nos cursos introdutórios de programação. Neste trabalho focalizamos a utilização da linguagem SQL, num ambiente altamente interativo e adaptativo, integrado a um ambiente de EAD.

1.2 Objetivos

Este trabalho propõe uma ferramenta na Web para ensino de programação. O trabalho é baseado em um estudo de caso para ensino da linguagem de programação de bancos de dados SQL e está inserido no contexto do ambiente interativo de aprendizagem AmAm. Esta ferramenta experimental para ensino de linguagens de programação, apresenta como objetivos:

- integração entre o ensino de conceitos de programação com exemplo de fragmentos de programas executáveis on-line;
- integração com um componente de avaliação de programas verificando se a resposta do aluno é a mesma do professor;
- monitoramento das atividades do aluno (quantos exemplos foram executados; em cada exercício quantas tentativas de execução foram feitas, etc.) ;
- servir como um laboratório para experimentação de duas novas técnicas de avaliação e acompanhamento, que estão sendo pesquisadas em trabalhos em paralelo (como parte de uma Dissertação de Mestrado do estudante Adriano Del Pino e como parte de uma

Tese de Doutorado da estudante Alekssandra da Silva), respectivamente: (a) avaliação automática de programas com métricas de engenharia de software, permitindo informar o quanto a solução do estudante se aproxima de uma “solução ideal”; (b) avaliação automática de questões conceituais discursivas, além de permitir as tradicionais perguntas objetivas.

Este trabalho é um primeiro passo na direção de construção de um ambiente totalmente assistido para ensino da linguagem de programação SQL, onde o professor terá uma carga de trabalho mínima, mais como supervisor. Um desafio deste trabalho é de ser desenvolvido já integrado ao ambiente AmAm.

1.3 Organização do Texto

Este trabalho constitui-se, além desta introdução, de mais 5 capítulos distribuídos como segue.

O Capítulo 2 apresenta os trabalhos relacionados com a ferramenta IETS, mostrando as características peculiares presentes em outros sistemas de ensino de SQL;

O Capítulo 3 descreve a visão conceitual da ferramenta IETS, mostrando a sua interação com o Ambiente de Ensino a Distância denominado AmAm. Como é organizado o conteúdo didático dentro da ferramenta IETS para proporcionar uma maior interatividade em cada tema de programação SQL e a análise inteligente das atividades de programação, ajudando o aluno na resolução dos problemas propostos;

O Capítulo 4 descreve a modelagem da ferramenta IETS, utilizando-se a linguagem UML; descrever os aspectos de implementação;

O Capítulo 5 apresenta as dificuldades na aplicação de uma avaliação e uma solução que poderá ser utilizada para tal problema. Neste capítulo também serão esboçados os resultados alcançados com a utilização da ferramenta IETS (laboratório de informática) como apoio didático à disciplina de Banco de Dados, mostrando os resultados obtidos nas avaliações propostas.

O Capítulo 6 apresenta uma conclusão. São avaliados os resultados obtidos com a implementação dos estudos de caso utilizando a ferramenta, sugerindo as melhorias tecnológicas que a ferramenta poderá ter, dando continuidade a este trabalho.

Por fim, são apresentados alguns materiais opcionais, mas que complementam o texto, no Apêndice.

II - TRABALHOS RELACIONADOS

Neste capítulo analisamos as ferramentas encontradas na literatura especializada. As ferramentas são listadas em ordem cronológica. Nossa ferramenta IETS foi prototipado no primeiro semestre de 2003 e publicado no mesmo ano em (Freire et al.,2003). Portanto, alguns dos trabalhos listados aqui foram desenvolvidos em paralelo ao IETS: a maior parte das funcionalidades deles já estava presente na versão inicial da nossa ferramenta em 2003.

2.1 SQLator (2004) – An Online SQL Learning WorkBench

O SQLator (Sadiq et al., 2004), assim como o IETS, é uma ferramenta web para o ensino de SQL, visando auxiliar o trabalho do professor e enfocando principalmente no acompanhamento e na avaliação das atividades do aluno. Ambos o SQLator e o IETS possuem propostas similares pois foram desenvolvidos em paralelo, inclusive o nosso sistema foi publicado antes, em 2003.

Os dois são ferramentas para ensino de SQL, com as funcionalidades, listadas a seguir:

- apresentar o material teórico na forma de páginas web;
- prover funcionalidades para prática de exemplos ou consultas como respostas de exercícios; os exercícios são classificados em categorias (por exemplo: fácil, médio e difícil);
- gerar exercícios / provas randomicamente a partir de listas de perguntas;
- prover a habilidade de avaliar automaticamente as consultas de SQL dos alunos, provendo um retorno imediato;
- prover recursos para o aluno e o professor acompanharem o progresso no aprendizado através de painéis de monitoramento na ferramenta, respectivamente do aluno e da turma; gerar estatística do aluno e da turma.

O SQLator possui também um sistema para detectar se os alunos fazem plágio (i.e. mesma resposta); são considerados apenas os casos em que várias (mais de 4) respostas são iguais.

A proposta do SQLator é prover um componente inteligente que substitui o especialista na atividade de avaliação. A sua abordagem de avaliação é baseada em algoritmos heurísticos complexos, que determina quando uma consulta SQL responde a uma determinada questão.



The screenshot shows the 'Evaluate' page of the SQLator application. On the left is a navigation menu with links for Home, Database, Data Model, Tables, Queries, Evaluate, Execute, Tutorials, and Profile. The main content area is titled 'Evaluate' and displays the following information:

- Selected Query:** Query Id: 11
- Description:** Find all programs with duration shorter than 4 semesters. List programs and schools offering the programs.
- Difficulty:** Simple
- Status:** ✓ [Correct]
- Evaluated:** 09-Sep-03 1:55:29 PM

Below this information is a text area labeled 'Your SQL Query' containing the query: `Select Program, School from Program where Program.Duration < 4`. There are four buttons: 'Execute', 'Evaluate', 'Solution', and 'Notes'. Underneath, the 'Evaluation Results' section states 'Your query is correct.' and the 'Query Results' section shows a table with the following data:

Program	School
MIS	IS
MET	DC
MSE	SE

Figura 2.1 Formulário de Avaliação do SQLator
Fonte: SQLator (2004)

2.2 AsseSQL (2004) – An Online Tool to Test SQL Query Formulation Skills

O AsseSQL (Coleman & Lister, 2004) é uma ferramenta de avaliação online de SQL: a partir de uma base de questões é aleatoriamente selecionada uma prova ou uma lista de exercício (como um conjunto de questões); o estudante responde livremente qualquer uma das questões, em qualquer ordem. O feedback são os erros de sintaxe e se a resposta traz ou não o resultado correto (ver figura 2.2). As questões da prova ou exercício são marcadas como não respondida, respondida com sucesso, ou incorreta.

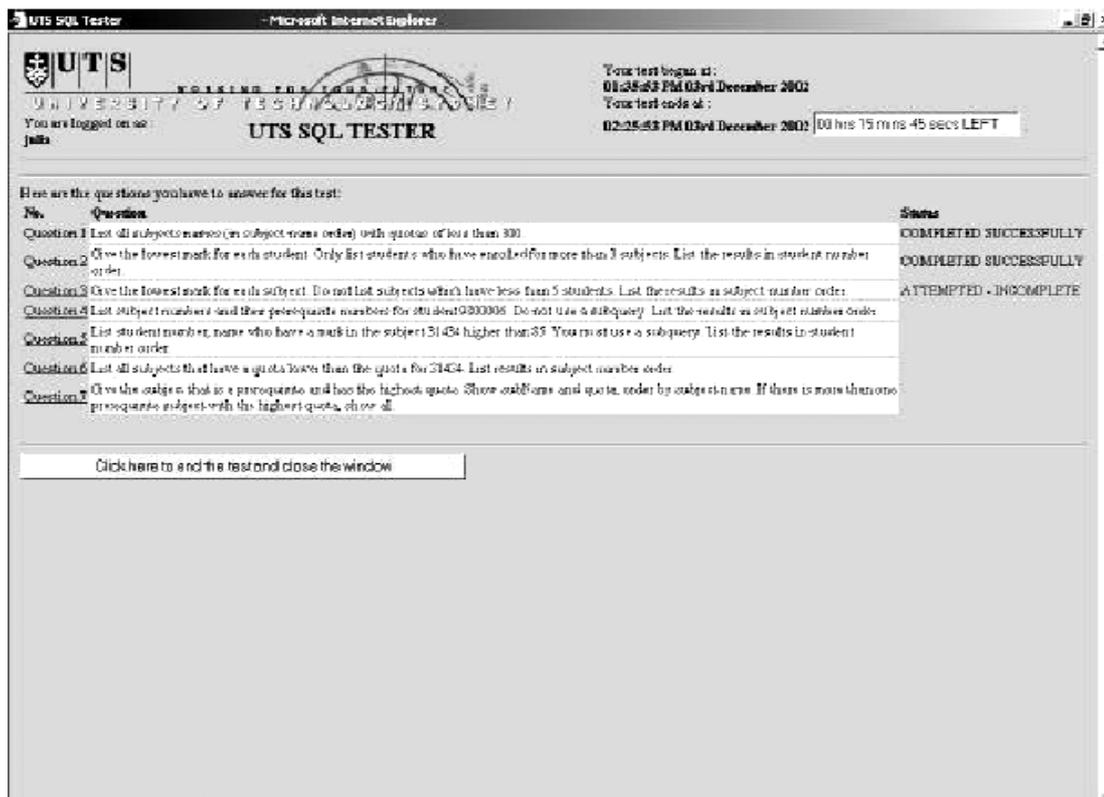


Figura 2.2 AsseSQL: Formulário Principal: Resultado das Tentativas nas Questões
Fonte: AsseSQL (2004)

As perguntas (questões) que compõe uma prova ou exercício são classificadas em categorias, com base em três indicadores (número de tabelas, número de atributos e complexidade das construções). Segue um exemplo (Coleman & Lister, 2004) (supondo 5 categorias) de uma prova com 5 perguntas, uma de cada categoria utilizando o comando “select” da linguagem SQL:

- numa tabela com uma cláusula “where”;
- numa tabela com mais de uma cláusula “where”, união de operadores lógicos;
- numa tabela com uma cláusula “group by” e um “having”;
- em duas tabelas com uma junção natural (INNER JOIN);
- com uma sub-consulta contendo um único “select”.

Para cada estudante é feito um sorteio aleatório diferente; são sorteadas diferentes perguntas em SQL, porém, todas as provas possuem o mesmo número de questões de uma

determinada categoria. Todos os enunciados da base de questões são marcados com a sua categoria.

2.3 SQL-Tutor (1998)

O SQL-Tutor é um sistema tutor inteligente (STI) para auxiliar pedagogicamente o estudante a resolver consultas SQL. A ênfase desta ferramenta é o feedback inteligente, como exemplificado na figura 2.3. Para isso, ele utiliza um *Shell* de um sistema especialista com tecnologia CBM (modelagem baseada em restrição) (Ohlsson, 1994): na versão de 1998 tinha aproximadamente 200 regras na base de conhecimento; atualmente, existe uma versão adaptada para a web (Mitrovic, 2000,2002).

Example 1: For each director, list the director's number and the total number of awards won by comedies he/she directed if that number is greater than 1.

<p>Correct solution:</p> <pre>SELECT DIRECTOR,SUM(AAWON) FROM MOVIE WHERE TYPE='comedy' GROUP BY DIRECTOR HAVING SUM(AAWON)>1</pre>	<p>Student's solution:</p> <pre>SELECT DIRECTOR,SUM(AAWON) FROM DIRECTOR JOIN MOVIE ON DIRECTOR=DIRECTOR.NUMBER WHERE TYPE='comedy'</pre>
---	--

INGRES: E_USOB63 line 1, The columns in the SELECT clause must be contained in the GROUP BY clause.

SQL-Tutor:

- You do not need all the tables you specified in FROM!
- You need to specify the GROUP BY clause! The problem requires summary information.
- Specify the HAVING clause as well! Not all groups produced by the GROUP BY clause are relevant in this problem.
- If there are aggregate functions in the SELECT clause, and the GROUP BY clause is empty, then SELECT must consists of aggregate functions only.

Figura 2.3 A análise de saída do SQL-Tutor mostrando o retorno a ser dado ao estudante (Mitrovic, 1998)

O componente tutor-inteligente possui duas fontes de conhecimento: específico do domínio (incluindo conhecimento pedagógico) em forma de regras e um modelo do estudante, o qual é usado para guiá-lo na escolha de problemas a serem submetidos no ambiente. Este conhecimento é utilizado para diagnosticar as tentativas de soluções dos estudantes e/ou resolver os problemas. O conhecimento está representado no *shell* CBM na forma descritiva: como restrições para diagnosticar erros. As restrições dividem todas as possíveis situações de

problemas dentro de classes de problemas equivalentes; todas as situações numa única classe são consideradas pedagogicamente equivalentes, para as quais é gerada uma mesma ação.

SQL-Tutor abrange apenas o comando SELECT da linguagem SQL, o que não reduz a importância do sistema, porque as consultas cobrem os aspectos mais complexos do SQL. Além disso, os outros comandos SQL (INSERT, DELETE, UPDATE) também se utilizam de subconsultas do tipo SELECT.

O SQL-Tutor também tem suas limitações. Uma limitação acontece porque com regras manualmente codificadas é difícil prever todas as possíveis situações. Por exemplo, a Figura 2.4 mostra um erro semântico simples com um diagnóstico errôneo: o sistema não consegue identificar que ocorreu uma troca de atributos: BORN por DIED.

```
Example 2: List the names of all directors born in or after
1920.

Correct solution:
SELECT LNAME, FNAME
FROM DIRECTOR
WHERE BORN >= 1920

Student's solution:
SELECT LNAME, FNAME
FROM DIRECTOR
WHERE DIED >= 1920

Ingres:
lname      fname
Hitchcock  Alfred
De Mille   Cecil
Ford       John

SQL-Tutor:
Check that you are comparing the numerical constant to the
right attribute in the WHERE clause!
```

Figura 2.4 Inabilidade do SQL-Tutor lidar com alguns erros semânticos

Outra limitação é a entrada de dados dirigida pela sintaxe, o que não é bem aceita para usuários programadores, pois limita muito a digitação da consulta, resultando num certo bloqueio na criatividade: o aluno é forçado a usar o esquema sugerido (Figura 2.5). Este tipo de interface é bom para aprender a sintaxe da linguagem, mas não é bom para o aluno aprender a criar soluções para as consultas.

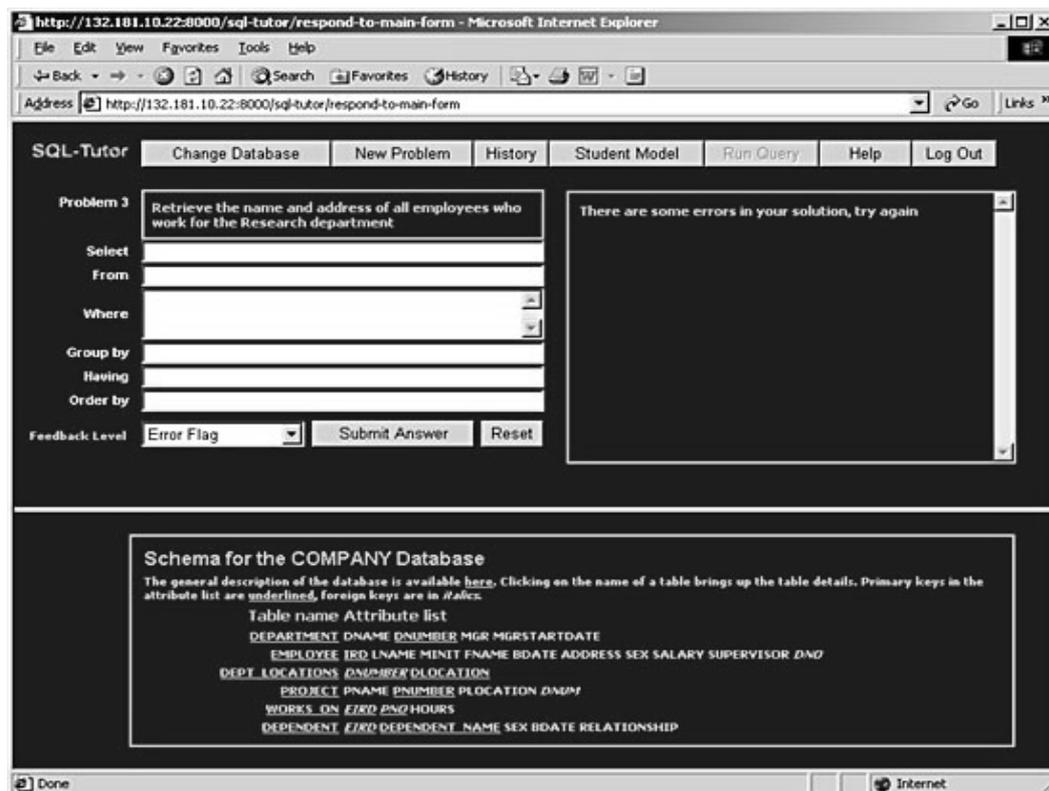


Figura 2.5 Formulário Principal do SQL-Tutor Web
Fonte: SQL-Tutor (1998)

2.4 Análise dos Trabalhos Relacionados

A ferramenta mais sofisticada é o SQLator. O grande diferencial do IETS, com relação a ele, está na proposta mais abrangente de avaliação automática. No SQLator o retorno de uma consulta é apenas correta ou incorreta (componente inteligente); caso seja incorreta, o aluno deverá tentar inserir uma nova resposta; por fim, caso não tenha sucesso ele pode pedir para o sistema a resposta certa ou pode consultar (via e-mail) o professor. Na proposta do IETS, além de tudo isso, existe uma avaliação que compara a resposta do aluno com a resposta do professor, dando oportunidade para o aluno aperfeiçoar a sua resposta mesmo depois da consulta retornar o resultado esperado; esta forma de avaliação é o trabalho de uma dissertação em andamento (Avaliação Automática de Consultas SQL por Adriano Del Pino), implementada sobre o IETS.

Por outro lado, a ferramenta que traz um feedback mais inteligente (SQL-Tutor) depende de uma base de regras escrita manualmente por um especialista em SQL e que dificilmente poderá cobrir todos os casos possíveis. Além disso, em muitos casos o feedback é fraco como ilustrado na Figura 2.4.

Particularmente a ferramenta IETS, considerada como um todo (com os componentes de avaliação sendo desenvolvidos) difere destes trabalhos descritos anteriormente em dois principais aspectos:

- A abordagem é de avaliação profunda (permitindo um feedback imediato em que o aluno pode aperfeiçoar a sua solução) em relação ao SQLator;
- A abordagem é genérica, não depende de regras, em relação ao SQL-Tutor.

III - IETS – VISÃO CONCEITUAL

3.1 Contexto: Projeto AmAm

O IETS foi desenvolvido como uma ferramenta do ambiente AmAm. No estado atual de desenvolvimento, o AmAm está operacional como um laboratório para estudos.

O AmAm é desenvolvido a partir de uma arquitetura que propõe o desenvolvimento de AVAs a partir de agentes, componentes e *framework* (Silva et al., 2003), e considera como princípio básico a integração de todos os aspectos de conhecimento do estudante (Gava, 2002). Em contraste com as abordagens tradicionais dos AVAs, onde disciplinas e cursos (ou grandes áreas de conhecimento) são tratadas de forma isolada, sem uma integração dos assuntos pertencentes as diversas disciplinas que estão sendo estudados por um estudante em um determinado momento, o AmAm, na sua concepção, utiliza uma metáfora de células, aonde cada célula pode ser uma ferramenta de aprendizagem, compondo o espaço de aprendizagem do estudante. A figura 3.1 exemplifica na parte superior da estação de aprendizagem do AmAm, o espaço destinado as ferramentas de aprendizagem.

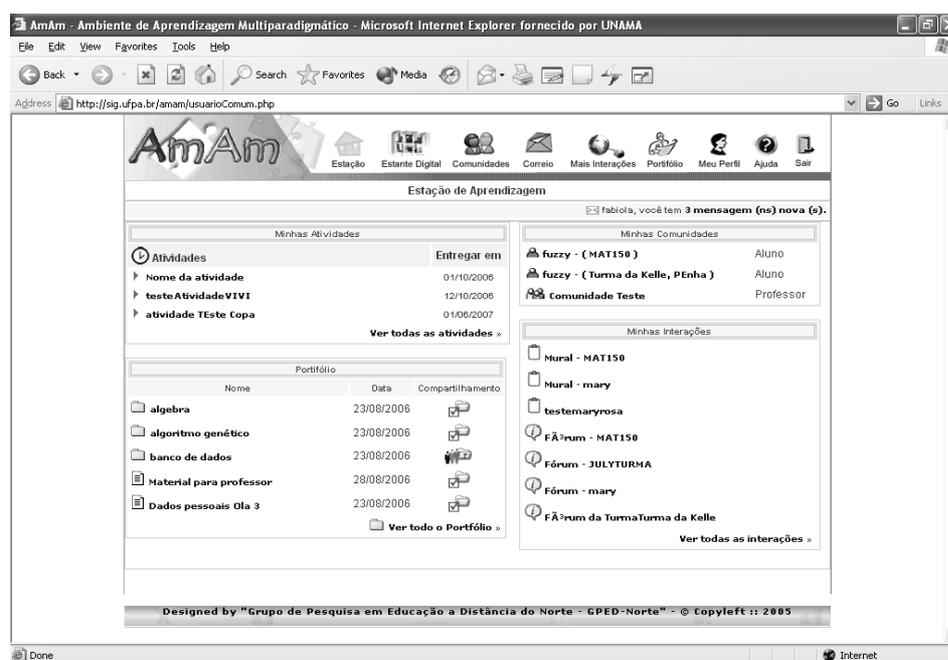


Figura 3.1 AmAm (Ambiente de Aprendizagem Multiparadigmático)

Fonte: (Harb et al., 2003)

O ambiente foi desenvolvido para apoiar comunidades virtuais de aprendizagem (Preece, 2000). A idéia é evitar que o estudante recomece tudo a cada novo período, sem que possa realizar conexões com os materiais estudados anteriormente e também sem poder explorar seus caminhos futuros em relação as disciplinas que ainda serão estudadas. Esse problema, segundo (Gava, 2002), é um fator que priva o estudante na maioria dos AVAs, dificultando sua cognição.

Na figura 3.2 ilustra-se um cenário de uso do AmAm, considerando a IETS como uma ferramenta, constante no repositório de ferramentas, e que pode constituir uma célula na estação de aprendizagem.

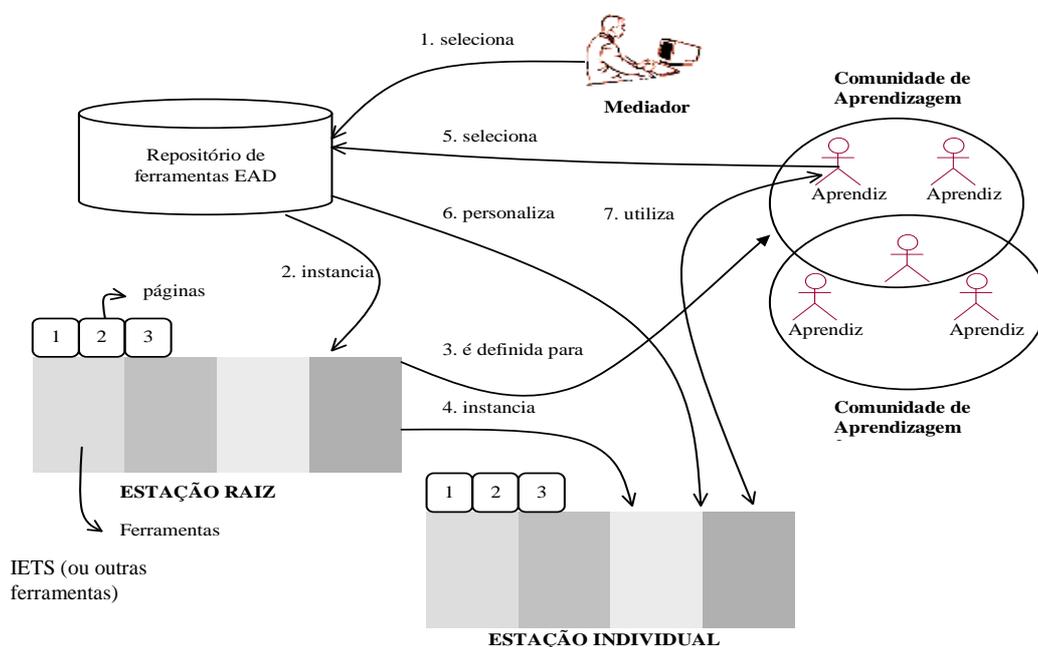


Figura 3.2 Contexto de Implementação do IETS
Fonte: (Harb et al., 2003)

Como citado, a vantagem de inserir uma ferramenta no projeto AmAm está nas facilidades de cooperação e colaboração elicitadas para o AmAm. Na proposta do AmAm, um mediador constrói um espaço de aprendizagem para uma comunidade, selecionando as ferramentas apropriadas para sua utilização (ver figura 3.2). Dentre essas ferramentas encontra-se a ferramenta IETS.

Uma vez selecionadas as ferramentas, o sistema gera uma estação raiz e a torna disponível para os estudantes. Após, um estudante pertencente àquela comunidade pode personalizar sua estação de aprendizagem (estação individual do estudante) acrescentando e

configurando novas ferramentas (ver figura 3.1). Uma comunidade de aprendizagem no contexto do AmAm, pode ser (Lima et al., 2005):

- Comunidades dos Ambientes Virtuais de Aprendizagem (CAVA): formadas por indivíduos (estudantes, professores, colaboradores e outros) que possuam cadastro em qualquer AVA;
- Comunidades das Instituições de Ensino (CIE): formada por indivíduos ligados as instituições de ensino;
- Comunidades das Áreas de Conhecimento (CAC): formada por indivíduos cujos interesses estejam relacionados de forma direta ou indireta a essas áreas e uma comunidade de área de conhecimento poderá conter sub-comunidades de aprendizagem;
- Comunidades dos Cursos (CC): formada por indivíduos membros de curso de qualquer natureza, sendo estes cursos de extensão, graduação, pós-graduação ou outra forma de organização. Uma possível forma de organização, para a Comunidade do Curso de Graduação, seria: Comunidade do Curso de Ciência da Computação, Comunidade do Curso de Pedagogia, etc. Nestas comunidades podem ainda existir sub-comunidades, como por exemplo, comunidades formadas grupos de estudantes de uma respectiva disciplina;
- Comunidades por Afinidades (CA): Lima et al. (2005) sugerem que essas comunidades sejam formadas por indivíduos que possuam interesses ou habilidades afins. A formação dessas comunidades pode acontecer por iniciativa dos próprios indivíduos, de acordo com os objetivos da comunidade ou podem ser sugeridas por um mecanismo inteligente do AVA, baseando-se em diferentes critérios (perfis, habilidades, interações, etc).

No AmAm, uma estação de aprendizagem é única por estudante e está organizada em páginas, que por sua vez, contém células. Cada célula na estação pode conter diferentes recursos para o estudante, como por exemplo: (1) jogos, simuladores, micromundos, sistemas tutores, tutoriais, assistentes, ferramentas para construção de mapas conceituais, entre outros softwares educativos; (2) ferramentas para facilitar a comunicação síncrona ou assíncrona; (3) ferramentas para auxiliar construção cooperativa (transferências de arquivo, edição cooperativa de textos, de figuras geométricas ou de imagens, etc); (4) ferramentas de uso geral, como tradutores e ferramentas de busca.

Adicionalmente, as células podem ser geradas pelo estudante com o objetivo de organizar sua estação de aprendizagem. Nesse caso, uma célula pode ter uma fórmula associada a ela e aplicável a outras células, como por exemplo: (1) uma fórmula que descreve (apresenta), as modificações em outras células nos últimos X dias; (2) uma fórmula que apresenta todas as ações desempenhadas por um outro estudante (esta ação pode ser importante no caso de atividades cooperativas); (3) uma fórmula para levantar palavras (temas) comuns a um conjunto de células; (4) uma fórmula para levantar a frequência de ocorrência de uma determinada palavra em outra célula; (5) uma célula pode ser clonada a partir de outra célula, com fins de reestruturação da sua fórmula.

No contexto da ferramenta IETS, um estudante pode elaborar soluções em SQL enquanto redige textos em outras células (janelas) para outros estudantes ou para o professor, explicando sua solução. Além disso, o estudante pode estar usando outras ferramentas pertencentes ao repositório de ferramentas do AVAs.

3.2 Organização do Conteúdo

Um curso é organizado estruturalmente em três níveis hierárquicos: módulo, sessão e sub-sessão (tema). Por exemplo, o protótipo atual do IETS possui os módulos: Módulo preliminar, Módulo básico e Módulo avançado. O Módulo preliminar é composto de seis sessões: Conceitos Fundamentais, Modelo Conceitual, Modelo Lógico, Normalização de Tabelas, Álgebra Relacional e Modelo Físico.

O Módulo básico, que envolve exemplos executáveis e atividades de programação, é composto de oito sessões. Neste módulo são apresentadas as sessões sobre o conteúdo básico da disciplina, mostrando as principais construções da linguagem, incluindo junções naturais e funções sobre agrupamentos. Os comandos são descritos através de exemplos que podem ser executados on-line, com apenas um clique do mouse, estabelecendo uma vivência entre teoria e a prática acerca do conteúdo ministrado. O Módulo avançado é dividido em cinco sessões, estando o conteúdo relacionado a um estudo mais aprofundado da linguagem SQL, promovendo explorações que permitam ao estudante desenvolver habilidades sobre técnicas avançadas de manipulação de dados e otimização de consultas.

Um módulo compreende várias sessões e uma sessão compreende uma lista-linear de sub-sessões, conforme ilustrado pela tabela 3.1. Assim, o conteúdo de uma sessão (por exemplo, o comando SELECT sobre uma única tabela) pode ser organizado através de sete

possíveis temas descritos na tabela 3.1. Podem-se utilizar todos os temas ou apenas uma parte deles, conforme o assunto a ser explorado. Por exemplo, a maior parte dos assuntos do Módulo básico (em operação) possui os temas II, III, IV, V e VI. Só algumas sessões possuem o tema VII, que compara construções equivalentes em SQL.

Dentro da organização hierárquica, são explorados principalmente três aspectos do ambiente criado pelo IETS: 1) a interatividade nos temas de programação; 2) a análise inteligente das atividades de programação e 3) a flexibilidade no sequenciamento do conteúdo. Estes itens são detalhados abaixo.

Tabela 3.1 Possibilidade de organização de uma sessão

Temas de uma sessão (sub-sessões)	Forma de avaliação	Atividade do aluno
I – Histórico	perguntas objetivas e V/F	Leitura acerca do assunto proposto.
II – Sintaxe	perguntas objetivas e V/F	Leitura acerca do assunto proposto.
III – Conceito Geral/Exemplo	monitorar aula/tempo; monitorar quais exemplos são rodados e quantas vezes	Leitura acerca do assunto proposto; rodar o exemplos
IV – Exemplo/Conceito Detalhe	perguntas objetivas e V/F; monitorar quais exemplos são rodados e quantas vezes	Leitura acerca do assunto proposto; rodar o exemplo
V – Exercícios de Revisão	perguntas objetivas e V/F	Leitura acerca do assunto proposto.
VI – Exercícios de Programação	monitorar aula/tempo; medir a qualidade das repostas	Escrita/execução de programas
VII – Construções Equivalentes	monitorar aula/tempo; medir a qualidade das respostas	Escrita/execução de programas

3.3 A Interatividade nos Temas de Programação

No IETS o estudante está diante de uma interface simples, composta por um menu de capítulos, lembrando o índice de um livro, o que facilita e acelera o aprendizado. Assim que ele termina a leitura do texto descritivo, pode iniciar o treinamento ou uma sub-sessão de avaliação. Para cada tema (sessão) podem ser mostrados vários exemplos, cada qual enfatizando um aspecto ou uma construção sintática do comando sendo apresentado. Os exemplos são apresentados nos temas III– Conceito Geral/Exemplo e IV – Exemplo/Conceito Detalhe (Figura 3.3). A diferença destes temas está que no III é descrito o conceito e no final é apresentado um exemplo geral, enquanto que no IV os conceitos específicos são introduzidos a partir de exemplos (tipicamente, vários exemplos similares introduzem

pequenos detalhes na programação). As sessões apresentadas na ferramenta podem ser configuradas em qualquer ordem (janela esquerda da ferramenta), podendo cadastrar novas sessões para um determinado capítulo. Em cada sessão são apresentadas figuras, o qual ilustram as questões que já foram resolvidas dentro de uma sub-sessão (janela superior). As cores utilizadas nas perguntas ilustram o nível de dificuldade, e nas respostas, a pontuação visual alcançada. As interrogações (?) representam as perguntas que ainda não foram resolvidas pelo aluno em cada sub-sessão.

The screenshot displays the IETS online training environment. The main window is titled "IETS - Treinamento Online - SQL - Microsoft Internet Explorer". It features a sidebar with a navigation menu for sessions, including "1-Preliminar" and "2-Básico". The main content area shows a session titled "Sessão:25 - Select (Estrutura Básica)" with a sub-session "254-Exemplo / Conceito Detalhe". A "Exercitar" button is visible. Below this, there are sections for "Exemplos:" and "Seleções usa...", each containing SQL code snippets. A "Exercitar" window is overlaid, showing a question number "1", a description "(Subjetiva SQL) Selecionar as pessoas da instituicao de ensino.", and a response number "1 de 1" with the SQL query "select * from pessoa". Below the response input, there are buttons for "Avaliar", "Apagar", and navigation controls. A table with the following data is displayed:

COD	NOME	ENDERECO	FONE
1	CARLOS AUGUSTO OLIVEIRA	RUA DAS FLORES,300	2304099
2	MARIA HELENA DE OLIVEIRA	TRAV. 25 DE SETEMBRO,322	2229092
4	JOSÉ ANTONIO SILVA	RUA DO CRUZEIRO,33	2330300

At the bottom of the "Exercitar" window, it shows "Data/Hora Execucao:27/7/2005 - 17:16:55" and "Número de Linhas".

Figura 3.3 Apresentação de Exemplo/Conceito detalhe
Fonte: IETS (2004)

No final de um tema ou de uma sessão são apresentadas listas de exercícios para avaliar o aprendizado do estudante. No IETS temos quatro tipos de exercícios: objetivos de múltipla escolha, objetivo/(V/F), programação SQL e Interpretação de texto. Todos os exercícios são previamente cadastrados pelo professor (perguntas e respostas) dentro do seu ambiente. Toda resposta de um estudante é automaticamente avaliada pelo sistema; a figura 3.4 ilustra uma questão de programação, que será avaliada pela ferramenta IETS. Na janela "Exercitar" (presente na mesma ferramenta) podem ser visualizadas as questões desenvolvidas para cada sub-sessão, através dos botões de navegação, sendo possível visualizar também, as respostas realizadas para cada questão (botões de navegação).

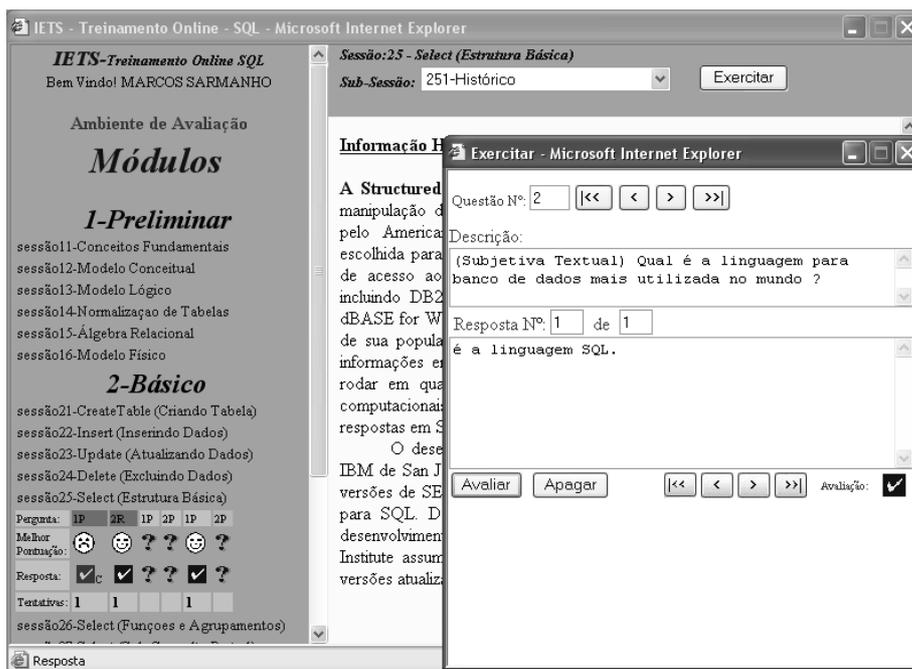


Figura 3.4 Tipo de Exercício realizado na Ferramenta
Fonte: IETS (2004)

3.4 A Análise Inteligente das Atividades de Programação

Existe uma diferença pedagógica entre os exercícios objetivos e os exercícios de programação (ou pergunta não objetiva). Um exercício objetivo é respondido, sendo registrado e analisado pelo sistema, não existindo uma segunda chance para o estudante responder novamente o mesmo. Se for o caso, devem ser submetidos novos exercícios (por exemplo, por sorteio). Por outro lado, nos exercícios de programação o estudante pode responder inúmeras vezes a pergunta de uma consulta, com o objetivo de cada vez se aproximar mais do resultado desejado. Estas respostas podem posteriormente ser examinadas pelo professor ou pelo aluno e podem ser compartilhadas com outros alunos.

No IETS, um módulo de análise inteligente das respostas dos exercícios de programação permite ao programador buscar a melhor resposta possível. Um algoritmo baseado em métricas de Engenharia de Software é utilizado para comparar a resposta do estudante com a resposta do professor, na Tabela 3.2 temos um Exercício de SQL com 3 respostas corretas (1A, 1B e 1C); respostas que produzem o resultado desejado. A última coluna da tabela, RX, mostra o valor de complexidade associado a cada resposta para o mesmo exercício.

Assim, cada uma das três respostas possui uma complexidade diferente: a melhor solução é a 1A tem valor 75,71; depois vem a 1B com valor de 79,61 e depois a 1C com valor 86,14; significa que a solução 1A é mais próxima da resposta do professor; a pior solução 1C está bem distante da solução do professor: $71,80 \times 86,14$; estar mais distante da resposta do professor significa possuir um maior valor de complexidade.

No cálculo da *complexidade* são consideradas diversas métricas de complexidade da disciplina de Engenharia de Software (ES), entre outras: comprimento do texto da solução, número de campos, volume de ES (Silva et al., 2006). A complexidade é calculada a partir de um modelo complexo de regressão linear, calculado em função das métricas mencionadas. O desenvolvimento deste modelo de regressão linear faz parte do trabalho de dissertação de mestrado do aluno Adriano Del Pino (Silva et al., 2006).

*Tabela 3.2 Exemplos de soluções equivalentes com diferentes métricas
Fonte (Silva et al., 2006)*

Exercício	Resposta – SQL	RX
1 P	Select Distinct A.Dept from Sells A,Supplies B Where A.itemno = B.itemno and B.Suppname = 'WARMTH,INC'	71,80
1 A	Select Distinct A.Dept from Sells A inner join Supplies B on A.itemno = B.itemno Where B.Suppname = 'WARMTH,INC'	75,71
1 B	Select Distinct A.Dept from Sells A Where A.itemno in (Select itemno from Supplies B Where B.Suppname = 'WARMTH,INC')	79,61
1 C	Select Distinct A.Dept from Sells A Where exists (Select itemno from Supplies B Where B.Suppname = 'WARMTH,INC' and A.itemno = B.itemno)	86,14

3.5 A Interação Aluno & Professor

A ferramenta IETS implementa dois principais módulos que estão interligados para proporcionar uma interação entre os atores (aluno e professor) dentro da ferramenta.

No módulo Aluno (Figura 3.4), o aluno resolve as questões propostas pelo professor, podendo a qualquer momento receber orientação acerca das respostas que não foram bem sucedidas, como por exemplo, a construção de um comando SQL. O professor envia através do seu ambiente, explicações, informando ao aluno aonde ele precisa melhorar (com os trabalho em andamento esta atividade será automatizada).

No módulo Professor (Figura 3.5), o professor é capaz de interagir com os alunos da sua turma a qualquer momento; enviando mensagens para as respostas mal respondidas; visualizando aqueles que estejam atrasados na resolução de questões ou monitorando os alunos que estejam realizando alguma avaliação na turma.

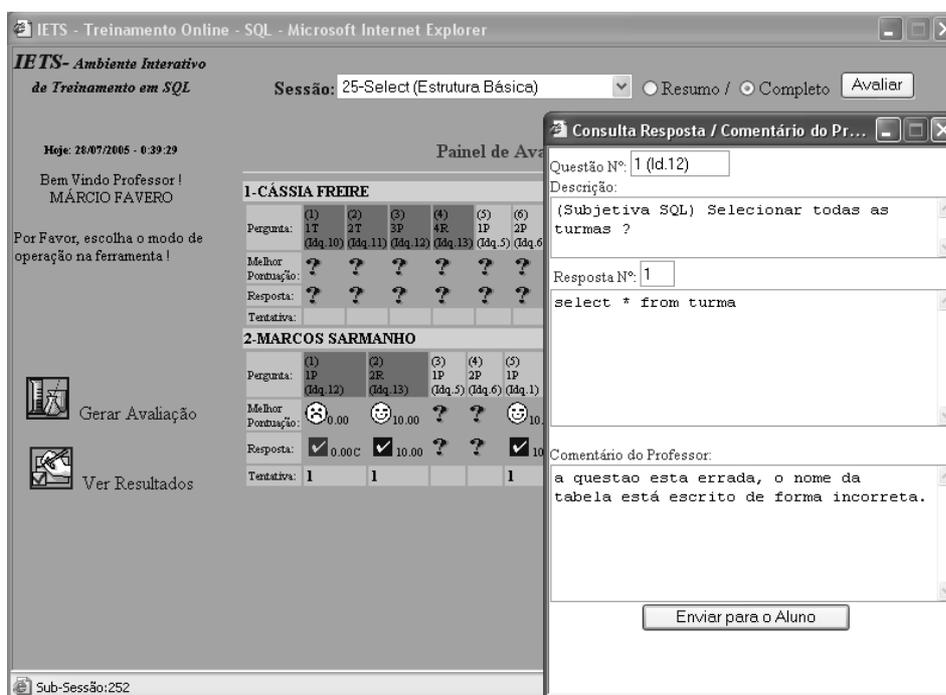


Figura 3.5 Ambiente do Professor (Comentando respostas/Gerando Avaliação)
Fonte: IETS (2004)

O IETS é uma ferramenta que pode ser incorporada a ambientes virtuais de aprendizagem (AVAs), especialmente no AmAm. A vantagem de utilizar o IETS no contexto de AVAs está nos benefícios que outras aplicações podem acrescentar ao aprendizado, facilitando a interdisciplinaridade, as interações (através de ferramentas disponíveis no ambiente) e as facilidades de acompanhamento do processo de aprendizagem que esses ambientes possuem.

Nesta direção, estamos integrando o IETS como uma ferramenta que pode ser adicionada às estações de aprendizagem dos estudantes no AmAm.

IV - PROJETO E IMPLEMENTAÇÃO DO PROTÓTIPO

A ferramenta IETS foi projetada para o aprendizado da linguagem de programação SQL, enfatizando a prática de programação, através do uso de um interpretador de comandos. Além disso, a ferramenta foi originalmente projetada como um ambiente altamente interativo e flexível, como descrito no capítulo III.

4.1 Requisitos

Para criar um software de uma aplicação, é necessária uma descrição do problema e dos seus requisitos, examinando o que é o problema e o que o sistema deve fazer para resolvê-lo. A análise enfatiza uma investigação do problema, de como uma solução é definida. Por exemplo, se o desejo é ter um novo sistema de informação para provar o funcionamento de uma tecnologia, quais os processos de negócio relacionados com o seu uso?

Para desenvolver uma aplicação também é necessário ter descrições de alto nível e descrições detalhadas da solução lógica e de como ela atende os requisitos. O projeto enfatiza uma solução lógica, ou seja, como o sistema atende os requisitos.

Neste estudo adotamos a abordagem Orientada a Objetos (OO) (Ambler, 2004). A essência da análise OO e do projeto OO é enfatizar a consideração de um domínio de problema e uma solução lógica, segundo a perspectiva de objetos (coisas, conceitos ou entidades), conforme a figura 4.1.

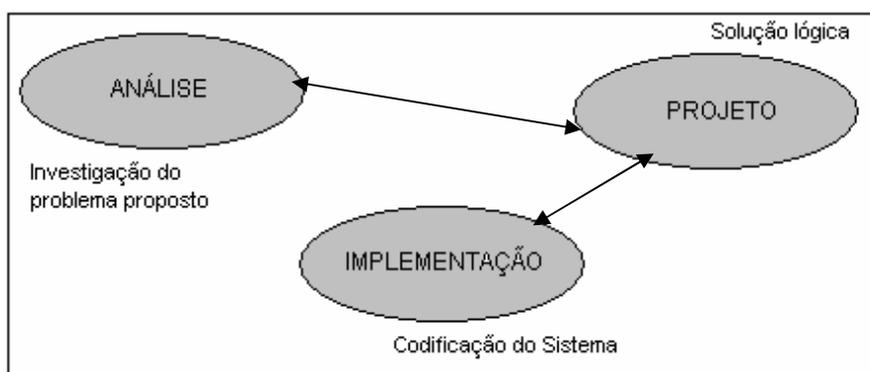


Figura 4.1 Fluxo das atividades de desenvolvimento

Durante a análise orientada a objetos, há uma ênfase na descoberta e na descrição dos objetos ou conceitos do domínio do problema.

Durante o projeto orientado a objetos existe uma ênfase na definição de elementos lógicos de software, os quais, em última instância, serão implementados como classes em uma linguagem de programação orientada a objetos. Estes objetos de software têm atributos e métodos.

O Projeto e Implementação do IETS são documentados com as seguintes notações de um processo simplificado de desenvolvimento de software:

- **Análise de Requisitos:** Com os Casos de Uso: a) Diagrama de Casos de Uso; b) Lista de Funcionalidades.
- **Projeto:** Com o Diagrama de Classes; Com um desenho da arquitetura em Componentes.
- **Implementação:** Mostrando de forma genérica como as Classes são projetadas via programas.

Nas seções que seguem, estas etapas são exemplificadas. Nos limitamos aqui a apresentar um exemplo significativo de cada artefato de documentação produzido. O leitor deverá procurar o Apêndice para encontrar maiores detalhes sobre a especificação dos componentes do sistema.

4.2 Casos de Uso

A primeira etapa do processo consiste no levantamento do que se deseja que o sistema faça. Trata-se de uma lista de necessidades agrupadas de modo que caracterize cenários de utilização do futuro sistema. O objetivo é separar as necessidades e identificar os objetivos de cada uma, sem entrar em detalhes de como fazer isto no sistema. É apenas “o que” deve ser feito e não o “como”.

Com base nas necessidades apontadas, pode ser elaborado um ou mais diagramas de casos de uso (UML) com objetivo de representar graficamente os usuários (atores) interagindo com o sistema (casos de uso) em várias situações (cenários).

Cada caso de uso deve representar um processo do início ao fim em relação à atuação de um determinado ator com o sistema, descrevendo todos os passos (eventos) desta atuação.

4.2.1 Caso de Uso Principal

O caso de uso principal do sistema IETS está representado na figura 4.2, este caso de uso foi desenvolvido tomando como base o ator Usuário, mais genérico na hierarquia de atores envolvidos no sistema. O ator Aluno tem por propósito utilizar o sistema para desenvolver suas atividades de aprendizado, o ator Professor gerencia treinamentos e avaliações para suas turmas e o ator Administrador mantém o funcionamento geral do sistema, com tarefas de cadastros básicos do sistema.

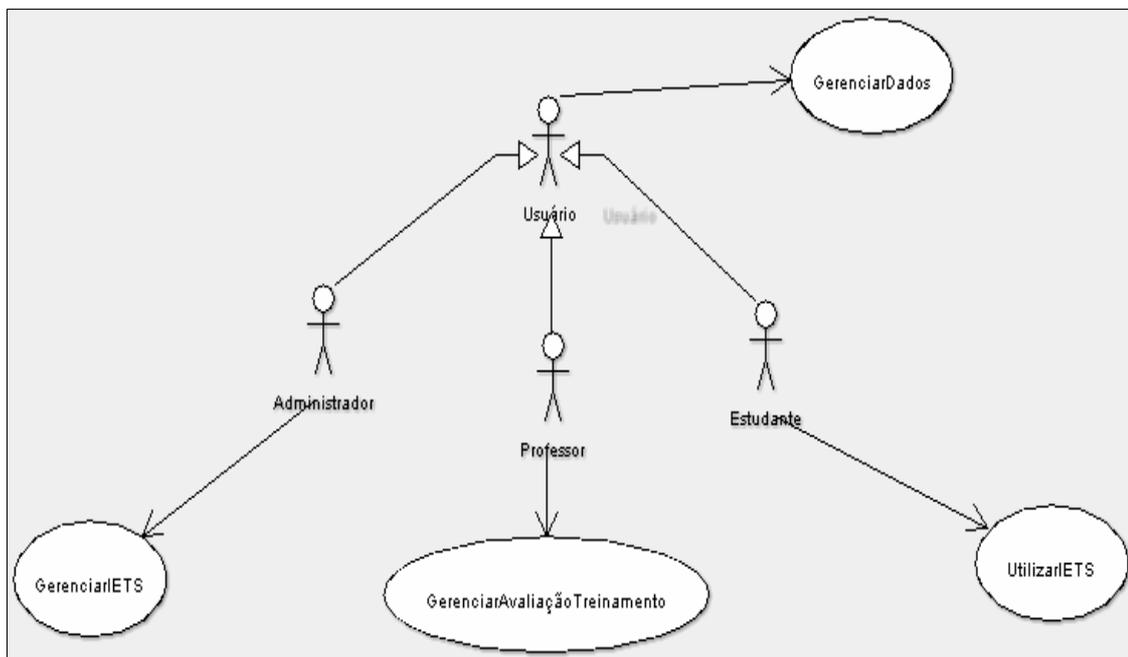


Figura 4.2 Diagrama de Caso de Uso Principal

A partir da especificação do caso de uso principal, os outros diagramas que o detalham são especificados de forma gradativa para facilitar o entendimento do sistema. Os casos de uso identificados na figura 4.2 representam quatro sub-sistemas que são decompostos, criando outros diagramas de casos de uso detalhados. Cada Ator envolvido em algum caso de uso possui uma identificação única, que será utilizada para acessar o sistema. Abaixo estão detalhados os atores envolvidos no sistema:

Ator Administrador: é responsável pelas operações de gerência sobre todo sistema, engloba todas as operações de manutenção de alunos, professores, instituições, cursos, disciplinas e turmas no sistema.

Ator Professor: é responsável pela gerência das turmas criadas, a qual os alunos serão anexados. O sistema permite que o papel do professor possa ser desempenhado por professores credenciados de diversas instituições de ensino superior, através do acesso via internet. O papel principal do professor neste ambiente é monitorar (gerenciar) todo o aprendizado do aluno, através de uma interface de consulta de turmas, interagindo, se necessário, via sistema, através de comentários dentro de questões já resolvidas pelo aluno, ou interagir pelo modo tradicional (corpo a corpo). O professor Cadastra novas avaliações, informando o seu período de abertura e fechamento, bem como a sua pontuação, podendo também cadastrar novos problemas para serem resolvidos pelos alunos nas diversas turmas de um curso.

Ator Aluno: participa das turmas de banco de dados propostas pela instituição; participa de avaliações, quando forem liberadas pelo sistema, e de treinamentos utilizando a ferramenta com o auxílio do respectivo professor da disciplina. As informações de cada treinamento ou avaliação são visualizadas através de um painel que mostra informações sobre o progresso alcançado no desenvolvimento de tais atividades.

O detalhamento dos Casos de Uso acontece em duas dimensões, por funcionalidade como fluxo de funções (Autenticar Usuário, Cadastrar Aluno, etc.) e por ator como diagramas de Caso de Uso (Administrador, Aluno e Professor). Este detalhamento acontece respectivamente nas seções 4.2.2 e 4.2.3.

4.2.2 Casos de Uso por Funcionalidade

O Caso de Uso principal é decomposto em 20 Casos de Uso mais específicos, ilustrados na tabela 4.1.

Tabela 4.1 Requisitos Funcionais do Sistema

Referência	Funcionalidade
UC01	Autenticar Usuário
UC02	Cadastrar Aluno
UC03	Cadastrar Professor
UC04	Cadastrar Curso

UC05	Cadastrar Instituição
UC06	Cadastrar Disciplina
UC07	Cadastrar Turma
UC08	Cadastrar Perfil Turma
UC09	Listar Sumário Disciplina
UC10	Cadastrar Conteúdo
UC11	Cadastrar Problema
UC12	Cadastrar Solução Usuário
UC13	Gerar Mapeamento Acesso
UC14	Consultar Desempenho Turma
UC15	Interagir Aluno
UC16	Cadastrar / Gerar Avaliação
UC17	Informar Resposta Problema
UC18	Exibir Painel Sessão Resposta Aluno
UC19	Consultar Comentário Professor
UC20	Exibir Mapa Estatístico Acesso

Os casos de uso são descrições das funcionalidades do sistema, informando o que deve ser feito para que eles sejam realizados. Utilizamos um padrão de documentação aonde são descritos o nome do ator, objetivo do caso de uso, nome da funcionalidade e os passos que descrevem as ações entre o ator e o sistema.

Segue abaixo a descrição dos requisitos para um dos 20 Casos de Uso. Este documento nos dá uma noção dos demais 19 casos de uso que são listados no Apêndice.

Tabela 4.2 UC04: Cadastrar Curso

Caso de Uso:	Cadastrar Curso
Objetivo:	Cadastra um novo curso para uma instituição com a finalidade de agrupar as turmas para a realização de treinamentos e avaliações
Ator(es):	Administrador
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. Administrador identificado fornece os dados do curso para efetivação do cadastro <ol style="list-style-type: none"> a. Se os dados forem válidos <ol style="list-style-type: none"> i. Inserir dados do curso no sistema; ii. Mostrar mensagem: dados cadastrados com sucesso b. Senão forem válidos <ol style="list-style-type: none"> i. Mostrar mensagem: falha no cadastro

4.2.3 Casos de Uso por Ator

Servem para representar graficamente os casos de uso e os seus relacionamentos. É composto por atores que se comunicam com suas funcionalidades.

Ator: Aluno

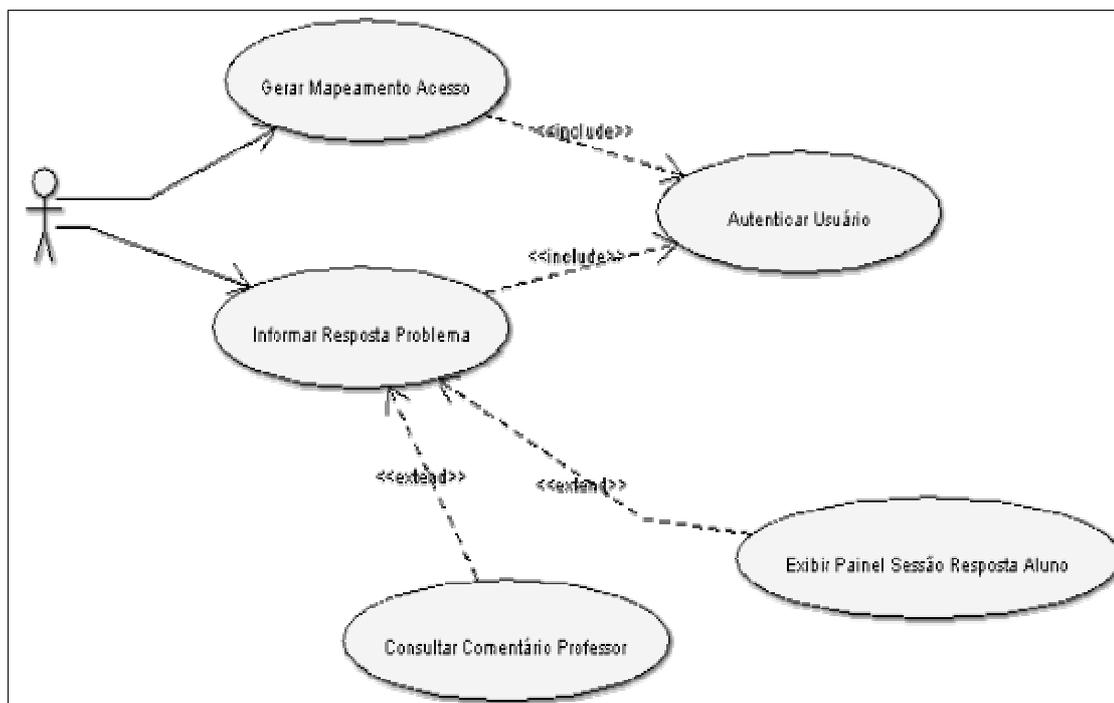


Figura 4.3 Diagrama de Casos de Uso Referente ao Ator Aluno

A figura 4.3 mostra o diagrama de casos de uso para o ator Aluno, explicando as suas funções desempenhadas dentro do sistema.

O ator Aluno informa a resposta aos problemas expostos nas avaliações e treinamentos, interagindo como o sistema, visualizando o resultado num painel exposto e na própria interface de resposta do aluno, podendo ainda consultar comentários do professor acerca da resposta aproximada ou não da ideal, possibilitando o desenvolvimento de novas respostas (tentativas) para determinada questão, estas funcionalidades descritas anteriormente estão expostas nos Casos de Uso **“Informar Resposta Problema”**, **“Consultar Comentário Professor”** e **“Exibir Painel Sessão Resposta Aluno”**.

O Caso de Uso **“Gerar Mapeamento Acesso”** está relacionado ao monitoramento dos passos do aluno nas diversas sessões e sub-sessões. Esta funcionalidade inicia a partir da entrada do aluno no sistema, logo após a sua autenticação (Caso de Uso: Autenticar Usuário).

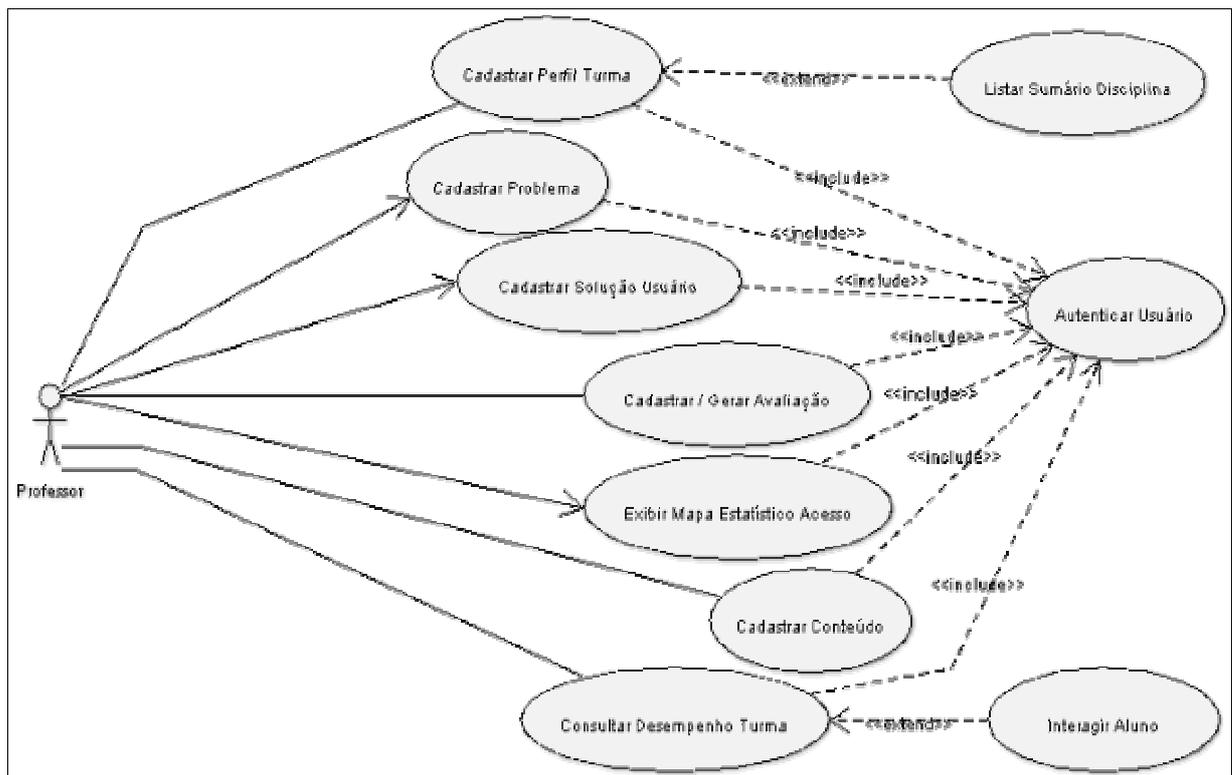
Ator: Professor

Figura 4.4 Diagrama de Casos de Uso Referente ao Ator Professor

A figura 4.4 mostra o diagrama de casos de uso para o ator Professor, explicando as suas funções desempenhadas dentro do sistema.

O professor cadastra os itens de conteúdo para a disciplina (sessões e sub-sessões) que será ministrada (Caso de Uso: Cadastrar Conteúdo), importando também o texto que será ligado a tal item de conteúdo (Caso de Uso: Importar Conteúdo); após os procedimentos anteriores, são ativadas somente as sessões e sub-sessões que estão disponíveis num determinado momento para turma, podendo dispará-las de acordo com a finalização de assuntos dentro da disciplina do curso (Caso de Uso: Cadastrar Perfil Turma e Caso de Uso de Extensão: Listar Sumario Disciplina).

O ator Professor cadastra os problemas com o objetivo de ligá-los, posteriormente, as avaliações e treinamentos; os problemas são do tipo: V ou F, alternativas, textual e execução de comandos SQL (Caso de Uso: Cadastrar Problema). Cada problema exposto tem uma ou várias soluções que são cadastradas também pelo professor da disciplina (Caso de Uso: Cadastrar Solução Professor).

As avaliações são cadastradas pelo professor de determinada disciplina e turma, informando os dados referentes ao período de sua realização, a quantidade de questões por grupo (Caso de Uso: Cadastrar / Gerar Avaliação), dentre outros dados referentes a sua realização.

O professor pode consultar o desempenho da turma, durante uma avaliação, com o objetivo de verificar o andamento da resolução dos problemas (Caso de Uso: Consultar Desempenho Turma) e interagir com os alunos da turma, através de comentários traçados para as questões no qual determinado aluno não tenha conseguido alcançar a nota máxima exigida (Caso de Uso de Extensão: Interagir Aluno).

Outro monitoramento que pode ser feito pelo professor da disciplina é a consulta do mapa de acesso (Caso de Uso: Exibir Mapa Estatístico Acesso), que tem como objetivo mostrar o trajeto realizado pelo aluno no percurso dentro da ferramenta.

O monitoramento verifica também, se o aluno está realizando as suas atividades, podendo corrigir eventuais desvios de percurso, como por exemplo, a quantidade excessiva de passos para percorrer um determinado assunto dentro da ferramenta IETS.

Ator: Administrador

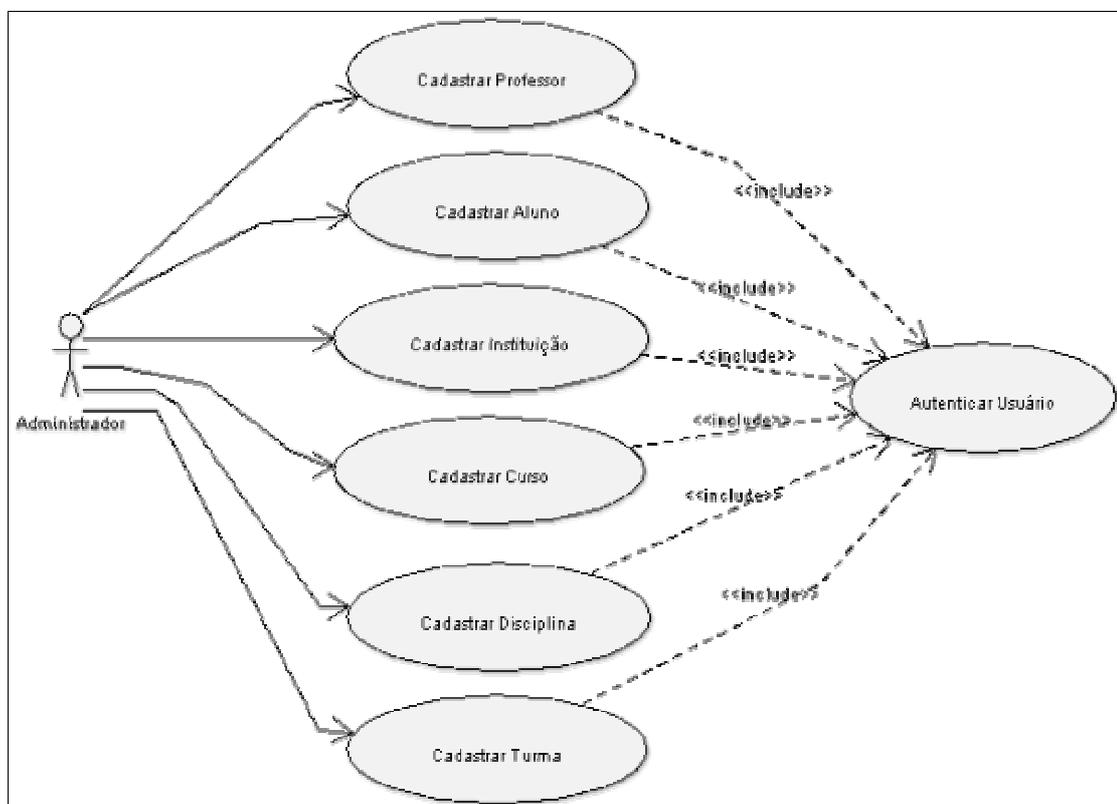


Figura 4.5 Diagrama de Casos de Uso Referente ao Ator Administrador

A figura 4.5 mostra o diagrama de casos de uso para o ator Administrador, explicando as suas funções desempenhadas dentro do sistema.

O Ator Administrador cadastra os professores (Caso de Uso: Cadastrar Professor), alunos (Caso de Uso: Cadastrar Aluno) que serão usuários do sistema, além de realizar outros cadastros, tais como: instituições (Caso de Uso: Cadastrar Instituição), cursos (Caso de Uso: Cadastrar Curso), disciplinas (Caso de Uso: Cadastrar Disciplina) e turmas (Caso de Uso: Cadastrar Turma) no sistema.

4.3 Projeto

Na etapa do projeto da aplicação, torna-se necessário modelar o conjunto de objetos, que de forma conjunta, representam as necessidades e processos apontados pela etapa de análise.

4.3.1 Diagrama de Classes

Os diagramas de classes agrupam objetos que possuem atributos e comportamentos e mostram os relacionamentos existentes entre os objetos. Desta forma, é possível agrupá-los em pacotes de modo a segmentar o desenvolvimento dos componentes.

Uma vez que o foco deste trabalho está na ferramenta “IETS”, o diagrama de classes é apresentado a seguir (figura 4.6).

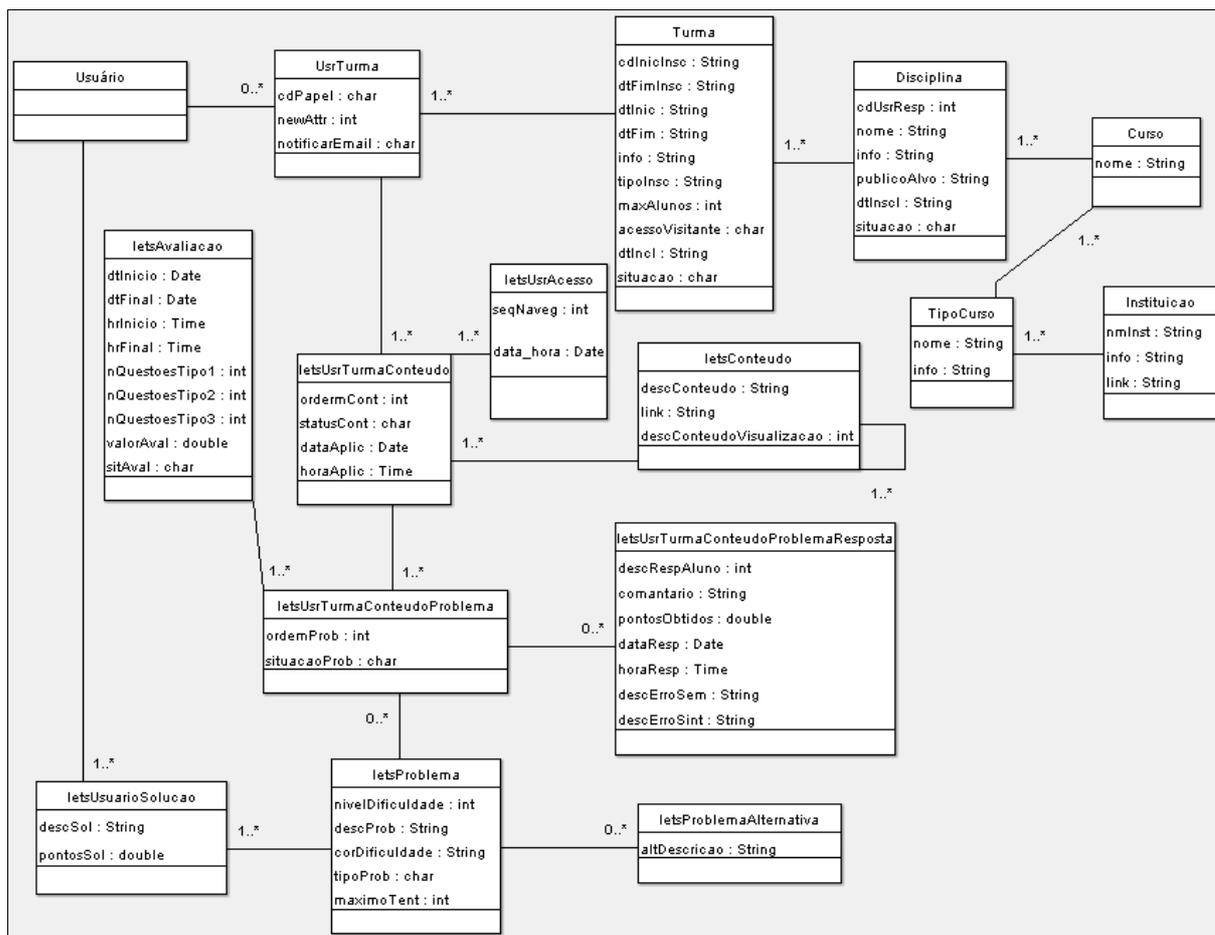


Figura 4.6 Diagrama de classes da ferramenta IETS

4.3.2 Arquitetura do Sistema

O IETS é um sistema desenvolvido para WEB, criado utilizando HTML, PHP e JavaScript, sendo utilizado o padrão de projeto Fachada (Gamma, 1994), que diminui a complexidade do sistema, minimizando sua dependência e comunicação entre subsistemas. Este padrão de projeto tem por finalidade estabelecer uma única interface de comunicação entre as classes clientes e as classes de negócio (Gamma, 1994). A figura 4.7 mostra uma classe genérica utilizando o padrão Fachada, contendo um fluxo de dados em sua direção, demonstrando a arquitetura.

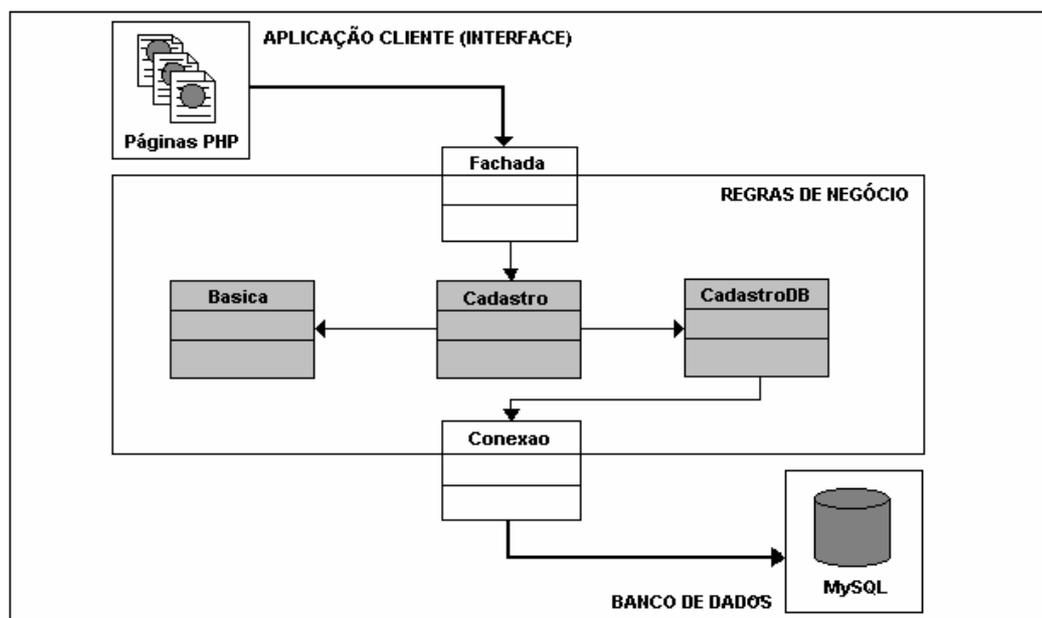


Figura 4.7 Esquema de interação dos componentes na Ferramenta IETS

As classes que contemplam as regras de negócio foram ilustradas com a utilização da cor cinza e as classes que possuem a finalidade de intermediar a comunicação entre a aplicação cliente e as regras de negócio, e estas com o banco de dados, estão na cor branca.

O lado cliente da ferramenta é composto de páginas PHP, sendo necessárias para o acesso as funcionalidades do ambiente.

O lado servidor da ferramenta é composto de um grupo de programas em PHP no qual residem as classes de regras de negócio. A classe “**Fachada**” é responsável pela comunicação entre as classes de regras de negócio e a aplicação, e a classe “**Conexao**” é responsável pela comunicação entre o banco de dados e as classes de regras de negócio.

A classe de regras de negócio denominada “**Basica**” é composta pelas classes: “**Basica**”, “**Cadastro**” e “**CadastroBD**”. A primeira fornece os atributos e para cada um deles é atribuído um método “**get**” e “**set**”, e também um método denominado “**setAll**” que faz a edição de todos os atributos. A classe “**Cadastro**”, por sua vez, é responsável pelo instanciamento da classe “**CadastroBD**”, disponibilizando o método correspondente. Por último, a classe “**CadastroBD**” realiza o instanciamento da classe “**Conexao**”, o qual realiza a conexão com o banco de dados correspondente, executando os comandos SQL, retornando resultados, que são tratados pelo sistema.

A arquitetura do IETS é composta, conforme a figura 4.8, por quatro módulos principais: Cadastros Básicos, Cadastros Específicos, Atividades e Avaliação. As classes que implementam a comunicação dos módulos com as regras de negócio e o banco de dados estão ilustradas no centro da figura.

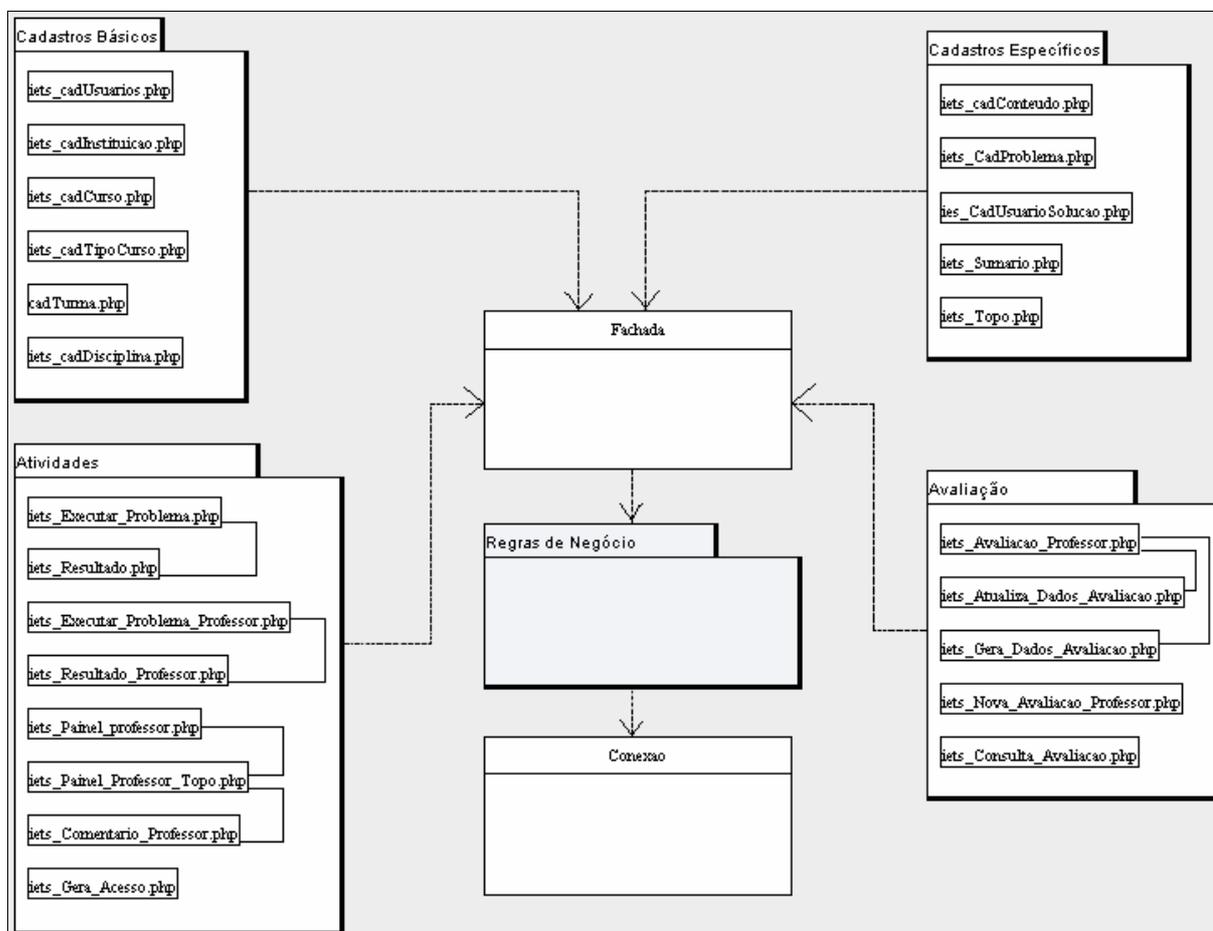


Figura 4.8 Arquitetura do IETS

O módulo de Cadastros Básicos tem como finalidade o cadastramento dos dados referentes aos usuários (administradores, professores e alunos), instituições de ensino, cursos a serem disponibilizados, turmas e as disciplinas que serão ministradas num determinado curso na instituição de ensino. Estes cadastros são operados pelo administrador do sistema.

O módulo de Cadastros Específicos está relacionado com o fornecimento de dados para a composição do conteúdo que será ministrado, a composição de problemas que serão utilizados nas avaliações e treinamentos e o cadastramento das possíveis soluções ligadas a determinado problema estabelecido. Estes cadastros são operados pelo professor da disciplina.

O módulo de Atividades é responsável pelas atividades que o aluno desempenha na ferramenta referentes a execução de problemas que serão avaliados pelo professor da disciplina. O professor ao longo de uma atividade de avaliação interage com a turma na ferramenta, analisando as respostas, e sugerindo, se necessário, alterações nas mesmas, possibilitando ao aluno alcançar a solução ideal para o problema proposto. Este módulo é operado pelos professores e alunos.

O módulo de Avaliação está relacionado à manutenção dos dados referentes às avaliações, o qual são utilizados pelo professor num determinado período, com o objetivo de testar os conhecimentos da turma em relação a um ou mais conteúdos ministrados com ou sem auxílio da ferramenta IETS. Este módulo é manipulado pelo professor.

4.4 Implementação

Esta seção contém duas subseções: uma mostrando um exemplo de código em PHP e a outra detalhando as tabelas do banco de dados. Na seção sobre o código do sistema exemplificamos apenas uma classe, pois o sistema tem milhares de linhas de código.

A implementação do sistema segue um modelo de arquitetura baseado em padrões e componentes que se repetem para muitas das classes codificadas. Portanto no Apêndice B mostramos um exemplo de programa para cada uma das classes do modelo da arquitetura. Com este conjunto de classes o leitor tem uma idéia de como é codificado o sistema. É claro que existem classes especiais que fogem deste padrão, mas o padrão representa mais ou menos 80% do código produzido.

4.4.1 Exemplo de Código para Classe Cadastro de Curso

O código da Listagem 4.1 mostra uma classe PHP para o cadastramento de cursos na ferramenta IETS.

```

1  Iets_CadCurso.php
2  <?
3  // ClassGenerator 1.5 Lite - www.classgenerator.net
4  // Classe gerada em 21/09/2004 10:10:04
5  include_once dirname(__FILE__).'/classes/class.Fachada.php';
6  session_start();
7  $fachada = new Fachada();
8  if($_POST['btnSalvar']){
9  $curso = new Curso($_POST['txtCdInst'], $_POST['txtCdTipoCurso'],
10     $_POST['txtCdCurso'], $_POST['txtNome']);
11  $res = $fachada->inserirCurso($curso);
12  $msg = ($res)?(Util::getMsg('S')):(Util::getMsg('E'));
13  header('location:admCurso.php?msg='.urlencode($msg));
14  }
15  ?>
16  <HTML>
17  <HEAD><? include_once dirname(__FILE__).'/template/templateHeaders.php';?>
18  </HEAD>
19  <BODY>
20  <? include_once dirname(__FILE__).'/template/templateComeco.php';?>
21  <div id='divConteudo'>
22  <FORM name='frmCurso' method='POST' action='cadCurso.php' onSubmit='return
23  checarDados(this)'>
24  <table><caption>Cadastro Curso</caption>
25  <tr>
26  <td class='itemForm'>CdInst:</td>
27  <td><INPUT type='text' name='txtCdInst' size='60' value="" maxlength='40'
28  class='textForm'></td>
29  </tr>
30  <tr>
31  <td class='itemForm'>CdTipoCurso:</td><td><INPUT type='text' name='txtCdTipoCurso'
32  size='60' value="" maxlength='40' class='textForm'></td>
33  </tr>
34  <tr>
35  <td class='itemForm'>CdCurso:</td><td><INPUT type='text' name='txtCdCurso' size='60'
36  value="" maxlength='40' class='textForm'></td>
37  </tr>
38  <tr>
39  <td class='itemForm'>Nome:</td>
40  <td><INPUT type='text' name='txtNome' size='60' value="" maxlength='40' class='textForm'></td>
41  </tr>
42  <tr>
43  <td colspan='2'><INPUT type='submit' name='btnSalvar' value='Salvar' class='buttonForm'></td>
44  </tr>
45  </table>
46  </form>
47  </div><? include_once dirname(__FILE__).'/template/templateFim.php';?>
48  </BODY></HTML>

```

Listagem 4.1. Código gerado para a página PHP de Cadastro de Curso

4.4.2 Projeto de Banco de Dados

Apesar de haverem diferenças entre a abordagem relacional e o paradigma de objetos, os Banco de Dados Relacionais têm sido utilizados pela maioria dos desenvolvedores orientado a objetos para armazenar objetos (Ambler, 2004).

De acordo com a figura 4.9, o modelo de dados do IETS é composto de dezesseis tabelas, parte delas são do modelo do AmAm e parte específica do IETS, como descrito nas tabelas 4.3 e 4.4.

Tabela 4.3 Tabelas utilizadas pelo Ambiente AmAm e ferramenta IETS

Tabela	Descrição
usuario	armazena os dados referentes aos usuários que estão aptos à utilizar o sistema IETS. Estes dados são: Endereço completo, data de nascimento, sexo, local de trabalho, profissão, login, senha, entre outros.
usrturna	armazena os dados relativos as turmas nas quais os alunos e professores estão alocados para desempenhar seu papel dentro da ferramenta IETS. Estes usuários estão alocados apenas numa instituição de ensino, dentro de um determinado curso, para uma determinada disciplina.
turna	guarda os dados referentes as turmas que estão disponibilizadas para determinado número de alunos, tendo um prazo de duração determinado e um coordenador responsável pelos alunos alocados.
disciplina	armazena os dados das disciplinas que podem ser alocadas para determinada turma.
curso	guarda os dados referentes aos cursos disponíveis na instituição. Para o sistema IETS esta tabela é limitada aos cursos de informática que utilizem disciplinas voltadas ao ensino de SQL.
tipocurso	armazena os dados acerca do tipos de curso de uma determinada instituição.
instituicao	armazena os dados relativos as instituições de ensino aonde são ministradas as disciplinas que utilizam o ensino de SQL como parte integrante do seu conteúdo programático.

Tabela 4.4 Tabelas geradas especificamente para atender os requisitos da ferramenta IETS

Tabela	Descrição
iets_conteudo	guarda os dados referentes ao conteúdo para a disciplina que será ministrada, tais como: os módulos, sessões e sub-sessões que são utilizados para gerar o sumário de opções da ferramenta.
iets_problema	contém os dados acerca dos problemas que são utilizados para formar as questões que são utilizadas nas avaliações e treinamentos.

iets_problema_alter nativa	é responsável pelo armazenamento das alternativas utilizadas nos problemas cadastrados. O conteúdo possível esta tabela será os itens A,B,C... ou V/F, o professor escolhe se quer colocar outros tipos de letras ou números para sua composição.
iets_usrturma_con teudo	armazena o conteúdo que será utilizado para cada aluno na composição do sumário da ferramenta, o professor pode desta forma configurar um conteúdo alternativo de acordo com o aprendizado do aluno na linguagem SQL.
iets_usrturma_con teudo_problema	armazena os dados relativos aos problemas que estão ligados a determinado aluno.
iets_usrturma_con teudo_problema_res posta	responsável por armazenar os dados relativos a respostas cadastradas pelos alunos numa avaliação ou treinamento. Os dados são: a descrição da resposta do aluno ao problema proposto, um comentário escrito pelo professor, os pontos obtidos na questão, a descrição dos erros semânticos e/ou sintáticos que possam ocorrer numa questão SQL e a data e hora da resposta para monitoração.
iets_usuario_solucão	armazena os dados referentes as soluções ideais cadastradas para os problemas expostos numa avaliação ou treinamento, estas soluções podem ser cadastradas pelo professor da disciplina ou incorporadas pelo aluno, após uma avaliação ou treinamento.
iets_avaliacao	guarda os dados referentes as avaliações que são disparadas num determinado período para avaliar o desempenho do aluno no ensino de SQL.
iets_usracao	armazena os dados referentes aos passos realizados pelo aluno na ferramenta, para posterior manipulação pelo professor.

Abaixo segue uma breve descrição dos relacionamentos entre as tabelas.

A tabela **“usuario”** se relaciona com duas tabelas: **“usrturma”** (várias turmas podem ser alocadas para um determinado usuário), **“iets_usuario_solucão”** (uma ou mais soluções podem ser fornecidas por um determinado usuário).

A tabela **“turma”** se relaciona com duas tabelas: **“usrturma”** (vários usuários devem estar alocados para uma determinada turma), **“disciplina”** (uma determinada disciplina é utilizada por uma ou mais turmas de alunos).

A tabela **“usrturma”** se relaciona com três tabelas: **“usuario”** (cada usuário está alocado a uma ou várias turmas), **“turma”** (cada turma está alocada para um grupo de

usuários), **“iets_usrturma_conteudo”** (uma seqüência de conteúdo é alocada para um aluno numa determinada turma).

A tabela **“iets_usrturma_conteudo”**, por sua vez, se relaciona com quatro tabelas: **“usrturma”** (um ou mais conteúdos são alocados para um determinado aluno), **“iets_conteudo”** (um conteúdo pode ser alocado para um ou mais alunos de determinada turma), **“iets_usrturma_conteudo_problema”** (um ou mais problemas podem ser alocados para um determinado conteúdo de um usuário da turma), **“iets_usracesso”** (o usuário de uma turma realiza um ou mais acessos à conteúdos pertencentes aos módulos do sistema).

A tabela **“iets_conteudo”** se relaciona com ela própria (auto-relacionamento) visando agrupar conteúdos gerais (módulos) e específicos (sessões e sub-sessões) dentro do sistema.

A tabela **“iets_usrturma_conteudo_problema”** se relaciona com quatro tabelas: **“iets_usrturma_conteudo_problema_resposta”** (um determinado problema alocado para um conteúdo pode ter zero a várias respostas criadas pelos alunos de uma turma dentro de uma avaliação ou treinamento), **“iets_usrturma_conteudo”** (vários problemas podem ser alocados para um determinado conteúdo ligado a um aluno), **“iets_problema”** (um problema se relaciona a um ou vários conteúdos para os alunos de uma turma), **“iets_avaliacao”** (uma avaliação pode ser aplicada para uma ou várias turmas num determinado período).

A tabela **“iets_problema”**, por sua vez, se relaciona com três tabelas: **“iets_usrturma_conteudo_problema”** (um problema pode ser utilizado por um ou mais alunos de uma determinada turma), **“iets_problema_alternativa”** (um problema pode ter uma ou várias alternativas, dependendo do tipo de problema, que pode ser: objetivo alternativa, subjetivo SQL ou subjetivo textual), **“iets_usuario_solucão”** (um problema pode ter uma ou várias soluções ligadas a determinado usuário, o qual poderá ser professor ou aluno).

Os outros relacionamentos do diagrama não foram descritos anteriormente, ou envolverem tabelas auxiliares (disciplina, curso, tipocurso e instituicao).

4.5 As Tecnologias Utilizadas na Ferramenta

A ferramenta é desenvolvida como aplicação baseada na *Web*, onde o acesso às informações é disponibilizado a partir de um navegador, sendo, por este motivo, desnecessária a instalação de qualquer *software* adicional na estação dos usuários finais.

Para tornar dinâmica a apresentação do conteúdo, foi empregada a linguagem PHP (2005), que permitiu a implementação do projeto, além de ser flexível e independente de plataforma. Como servidor de aplicação, é utilizado o Apache (2005) para a disponibilizar os serviços *Web*. Essa linguagem, assim como a escolha do banco de dados MySQL, foram escolhas da equipe GPED-NORTE, em razão da experiência da equipe de desenvolvimento.

O PHP é uma linguagem que permite criar *sites Web* dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros da URL e *links*. A diferença de PHP com relação a linguagens semelhantes a *Javascript* é que o código PHP é executado no servidor, sendo enviado para o cliente apenas *Hypertext Mark-up Language* (HTML) puro (PHP, 2005).

O *MySQL* foi projetado pela empresa sueca TcX para aplicações baseadas na *Web*, cujos os dados são mantidos por um pequeno conjunto de programas. Dentre as vantagens na sua utilização pode-se destacar (Tcx, 2002):

- natureza econômica. O *MySQL* é gratuito para a maioria das aplicações;
- a administração do produto é simples e fácil, extremamente rápido e fácil de customizar;
- suporta programação em C, Perl, Java, PHP, dentre outras.

V – AVALIAÇÃO DOS RESULTADOS

Neste capítulo inicialmente comentamos como foram cumpridos os objetivos propostos na introdução. Num segundo momento apresentamos um estudo de caso em laboratório.

5.1 Objetivos da Proposta

Este trabalho propõe uma ferramenta na Web para ensino de programação. O trabalho é baseado em um estudo de caso para ensino da linguagem de programação de bancos de dados SQL e está inserido no contexto de um ambiente interativo de aprendizagem chamado AmAm.

Na proposta foram apresentados 5 objetivos:

- completa integração entre o ensino de conceitos de programação com exemplo de fragmentos de programas executáveis on-line;
- integração com um componente de avaliação de programas verificando se a resposta do aluno é a mesma do professor;
- monitoramento das atividades do aluno (quantos exemplos foram executados; em cada exercício quantas tentativas de execução foram feitas, etc.);
- servir como um laboratório para experimentação de duas novas técnicas de avaliação e acompanhamento, que estão sendo pesquisadas em trabalhos em paralelos (como parte de uma Dissertação do estudante Adriano Del Pino e como parte de uma Tese da estudante Aleksandra Silva): avaliação automática de programas com métricas de engenharia de software, permitindo informar o quanto a solução do estudante se aproxima de uma “solução ideal”; e avaliação automática de questões conceituais discursivas, além de permitir as tradicionais perguntas objetivas.

O IETS vem sendo liberado em diversas versões cada qual com novas funcionalidades. Abaixo descrevemos a versão atual, comentado o estado atual do sistema.

5.2 Protótipo Inicial

A primeira versão do IETS implementa as funcionalidades propostas nos três objetivos iniciais: é uma versão que integra o ensino de conceitos de programação com exemplo de fragmentos de programas executáveis on-line; inclui também a avaliação de consultas contra a resposta do professor; e um módulo de monitoramento das atividades do aluno (quantos exemplos foram executados; em cada exercício quantas tentativas de execução foram feitas, etc.). Na tabela 5.1 temos um fragmento de uma tela de monitoração de um campeonato de programação executado na ferramenta IETS entre três equipes.

No IETS temos o ambiente do aluno e o ambiente do professor. Numa prova (ou campeonato de programação), um conjunto de questões é sorteado e submetido aos alunos que se conectam no sistema via Módulo do Aluno. O Módulo do Professor permite monitorar o andamento de cada aluno (ou equipe) através da utilização de painéis presentes na ferramenta. Por exemplo, é possível mostrar a tela de monitoramento para a turma num laboratório através de um projetor de imagens, focalizando o andamento geral dos alunos/equipes envolvidas numa prova.

Em particular, esta abordagem de avaliação via sorteio de questões, implementada pelo IETS, tem sido usada pelo professor Eloi Favero em inúmeras turmas de BD, a vários anos (comunicação pessoal): “dado um BD com 4 ou 5 tabelas, são elaboradas aproximadamente 30 questões sobre consultas de SQL, que são distribuídas com uma semana de antecedência para os alunos exercitarem (em laboratório). No dia da prova cada aluno conecta na máquina, e para ele são sorteadas 5 questões e é estipulado em tempo de aproximadamente 30 a 40 minutos.” Com o IETS, já na versão inicial, o processo de sorteio de questões é completamente automatizado, economizando muito trabalho para o professor.

Tabela 5.1 Resultados da Avaliação

Questões	(1)				(2)				(3)				(4)				(5)			
Equipes	C	I	NT	TQ	C	I	NT	TQ	C	I	NT	TQ	C	I	NT	TQ	C	I	NT	TQ
(1)	C		2	06:48		I	4	04:39	C		1	05:47	C		1	06:25	C		7	08:35
(2)	C		1	07:44	C		1	07:12	C		2	06:45	C		1	05:33	C		1	06:48
(3)	C		1	05:47	C		1	06:09	C		1	06:01		I	3	08:37		I	3	07:02
Totais	3	0	4	20:19	2	1	6	18:00	3	0	3	18:33	2	1	5	20:47	2	1	11	22:25
Médias	1	1	1,3	06:46	1	1	2	06:00	1	1	1	06:11	1	1	1,7	06:57	1	1	3,7	07:28

C = Correta / I = Incorreta / NT = Tentativas / TQ = Tempo de Permanência na Questão

5.3 Protótipo com Avaliação “Aproximada” Automática de Consultas SQL

A versão atual IETS possui um sistema de avaliação da complexidade das consultas SQL, desenvolvido pelo aluno de mestrado Adriano Del Pino. Para esta atividade o IETS serve como um ambiente onde são cadastradas as questões e são feitos experimentos. Segue um resumo de um artigo que descreve a proposta de dissertação do estudante:

Este trabalho apresenta uma abordagem para avaliação automática de consultas SQL. Esta abordagem soluciona o desafio de avaliar automaticamente as consultas de SQL possibilitando o aluno aperfeiçoar a sua solução buscando, além de uma resposta que retorna o resultado correto, uma solução com uma complexidade similar a da solução do professor (ou a melhor solução da turma). Esta proposta pode ser utilizada em ambientes de educação à distância ou na educação presencial nas atividades de laboratório, incluindo as avaliações. A solução proposta tem como vantagens: o estudante recebe um feedback instantâneo durante a atividade prática de programação; o professor é liberado do árduo trabalho de correção de comandos SQL (principalmente os complexos). O método, fundamentado em estatística e métricas da Engenharia de Software, pode ser adaptado para outras linguagens tais como Java e Pascal

Abaixo temos um exemplo de tabela com os resultados de avaliação de questões pelo método proposto no trabalho do Adriano.

*Tabela 5.2 Exemplos de soluções equivalentes com diferentes métricas
Fonte: (Silva et al., 2006)*

Exercício	Resposta – SQL	RX
1 P	Select Distinct A.Dept from Sells A,Supplies B Where A.itemno = B.itemno and B.Suppname = 'WARMTH,INC'	71,80
1 A	Select Distinct A.Dept from Sells A inner join Supplies B on A.itemno = B.itemno Where B.Suppname = 'WARMTH,INC'	75,71
1 B	Select Distinct A.Dept from Sells A Where A.itemno in (Select itemno from Supplies B Where B.Suppname = 'WARMTH,INC')	79,61
1 C	Select Distinct A.Dept from Sells A Where exists (Select itemno from Supplies B Where B.Suppname = 'WARMTH,INC' and A.itemno = B.itemno)	86,14

5.4 Protótipo com avaliação automática aproximada de consultas não objetivas

Está em andamento outra versão do IETS com o sistema de avaliação automática aproximada de consultas não objetivas, sendo implementado pela estudante de doutorado Aleksandra

Silva. Para esta atividade o IETS serve como um ambiente onde são cadastradas as questões e são retornados os resultados de forma automática.

O trabalho explora o uso do modelo vetorial de recuperação de informação (Salton, 1971) adaptado a um processo de avaliação automática de questões discursivas. O modelo vetorial estabelece uma representação de cada documento como um vetor num espaço n -dimensional, assim os documentos podem ser comparados (por similaridade) entre si ou com uma expressão de busca. Nós propomos avaliar questões discursivas onde as respostas são representadas por vetores e comparadas entre si e com a resposta do professor.

5.4.1 Motivação

Uma tarefa trabalhosa para professores é avaliação de um grande número de provas discursivas. Com os avanços da informática tem surgido ambientes de auxílio a educação presencial e/ou a educação à distância (amam), os quais possuem ferramentas para cadastrar questões e aplicar provas. O problema é que nesses ambientes ainda não temos sistemas para avaliar questões discursivas. No trabalho da estudante Alekssandra (Silva et al., 2006), é apresentada uma proposta de um modelo de *avaliação aproximada* de questões discursivas. Com este modelo ainda não é possível fazer uma avaliação de forma precisa, por isso a chamamos de avaliação aproximada. Apesar disso, acreditamos que o resultado da avaliação aproximada é satisfatório (dá uma boa indicação dos conceitos presentes nas respostas) e o professor com o seu uso economizará muito no tempo de correção de questões discursivas.

5.4.2 O modelo vetorial para recuperação de informação

O modelo vetorial (Salton, 1971) é um modelo fundamentado em álgebra linear, desenvolvido para se implementar sistemas de recuperação de informações. Na recuperação de informações existe uma busca descrita por um vetor de termos (palavras-chave) que são comparados com os vetores dos termos existentes em cada documento. Neste modelo tanto a expressão da busca como cada um dos documentos da base são representados por vetores com os pesos dos termos. Na tabela 5.3 são exemplificados três vetores de termos, num espaço 3D, onde existem dois documentos (D1,D2) e uma expressão de busca (B1).

Tabela 5.3 Exemplos de soluções equivalentes com diferentes métricas

	T1	T2	T3	
D1	0.3	0.0	0.5	
D2	0.5	0.4	0.2	
B1	0.3	0.2	0.2	

A partir da tabela 5.3 é possível calcular a similaridade entre a expressão de busca (B1) e os dois documentos (D1, D2) para saber qual está mais próximo da expressão de busca. A similaridade é calculada pelo co-seno entre os diferentes vetores:

	t1	t2	t3	
X	$W_{1,1}$	$W_{2,1}$	$W_{3,1}$	
Y	$W_{1,2}$	$W_{2,2}$	$W_{3,3}$	

$$\text{Sim}(x,y) = \frac{\sum(w_{i,x} * w_{i,y})}{\sqrt{(\sum(w_{i,x})^2 * \sum(w_{i,y})^2)}}$$

D1	0.3	0.0	0.5	
B1	0.3	0.2	0.2	

$$\begin{aligned} \text{Sim}(D1,B1) &= (0.3*0.3+0.0*0.2+0.5*0.2) / \{ (0.3^2+0.0^2+0.5^2)+ (0.3^2+0.2^2+0.2^2) \} \\ &= 0.19 / 0.52 \end{aligned}$$

$$\text{Sim}(D1,B1) = 0,36$$

D2	0.5	0.4	0.2	
B1	0.3	0.2	0.2	

$$\begin{aligned} \text{Sim}(D2,B1) &= (0.5*0.3+0.4*0.2+0.2*0.2) / \{ (0.5^2+0.4^2+0.2^2)+ (0.3^2+0.2^2+0.2^2) \} \\ &= 0.27 / 0.62 \end{aligned}$$

$$\text{Sim}(D2,B1) = 0,43$$

Portanto, pelas equações acima, conclui-se que o documento D2 é mais similar à expressão de busca que o documento D1. Na essência o modelo vetorial trabalha com este conceito simples de comparação de vetores.

5.4.3 Conclusão do capítulo

Este trabalho é um primeiro passo na direção da construção de um ambiente totalmente assistido para ensino da linguagem de programação SQL, onde o professor terá uma carga de trabalho mínima, mais como supervisor. Por fim, outro desafio deste trabalho é de ser desenvolvido já integrado ao ambiente AmAm.

VI - CONCLUSÃO

Este trabalho nos permitiu, a partir de uma análise dos vários aspectos da metodologia adotada pelos professores no ensino de SQL, desenvolver uma ferramenta chamada IETS, integrada a um ambiente interativo de aprendizagem, o AmAm (Favero et al., 2003). O IETS é uma ferramenta flexível e amigável, onde o estudante pode navegar entre conceitos, exemplos e exercícios práticos de programação. Ao mesmo tempo, o professor é apoiado nas atividades de acompanhamento dos estudantes. A ferramenta IETS pode ser utilizada como apoio ao processo de aprendizagem (modalidades: presencial, semi presencial ou à distância), sem pretender substituir o professor em sala de aula.

IETS, com relação a outras ferramentas de ensino de programação (Clements et al., 2001), (Haynes, 1998), apresenta novas idéias, entre outras:

- completa integração de ensino de conceitos teóricos com técnicas de programação: a cada momento o estudante pode executar exemplos de um código visualizando o efeito da programação; responder questões práticas utilizando em parte estes exemplos e/ou elaborando as suas soluções;
- análise inteligente das respostas do estudante, podendo, por exemplo, saber quantos exemplos o estudante executou e quantas vezes tentou responder o exercício até concluí-lo; monitorando se a resposta se aproxima ou não da resposta do professor;
- permite flexibilizar o sequenciamento do conteúdo.

Para tornar o espaço de aprendizagem colaborativo, a ferramenta está sendo adaptada para a utilização no AmAm, de forma que seja possível privilegiar as estratégias de grupo na aprendizagem do SQL, como por exemplo, a viabilidade da comparação e análise das soluções obtidas por diferentes estudantes, a possibilidade de ter uma base de casos (com as soluções apresentadas) para auxiliar na busca de novas soluções, entre outras.

Dentre as possibilidades que surgem para minimizar a sobrecarga de trabalho do professor no acompanhamento dos estudantes, temos: a possibilidade de acompanhar vários estudantes simultaneamente (painéis de monitoramento da ferramenta IETS) e utilizar uma

estratégia didática para um grupo de estudantes ao mesmo tempo em que se possam tomar decisões (sobre as estratégias) personalizadas, comentários do professor sobre soluções de estudantes que podem ser disponibilizados para todos.

Além disso, já existem inúmeros trabalhos sobre a utilização da Web, não apenas para ensino de programação, mas como um ambiente educacional altamente interativo e adaptativo (Brusilovsky, 1998). Segundo Brusilovsky, em um ambiente destes, devemos abordar aspectos tais como: 1) sequenciamento inteligente e adaptativo do material do curso - navegação adaptativa; 2) solução de problemas a partir de exemplos; 3) análise inteligente das soluções do estudante. Neste trabalho, nós focalizamos estes três aspectos de interatividade e adaptabilidade para uma ferramenta de programação, com ênfase nos dois últimos.

Adicionalmente, pretendemos incorporar às estratégias didáticas, através da tecnologia de agentes, características pró-ativas, que permitam tornar a ferramenta mais dinâmica e flexível, adicionando facilidades tanto para os estudantes, quanto para os professores. Dentre as possibilidades que a tecnologia de agentes pode oferecer está a execução automática de estratégias pedagógicas, adaptando-as para o perfil do estudante, modelado conforme o seu desempenho no processo, tornando assim o ambiente altamente adaptativo.

A ferramenta IETS foi projetada para ser utilizada via WEB. Sistemas operacionais que suporta as tecnologias PHP e MySQL.

Para os trabalhos futuros, existem diversas propostas para a ampliação das funcionalidades da ferramenta, tais como:

- A utilização de algoritmos mineração textual para a resolução de questões subjetivas, o qual está em andamento;
- Utilizar uma abordagem similar para outras linguagens, por exemplo, Java ou linguagens para programação de algoritmos (português estruturado).

REFERÊNCIAS BIBLIOGRÁFICAS

AMBLER, Scott W. **The Fundamentals of Mapping Objects to Relational Databases**. 2000 Copyright 2003-2004 Scott W. Ambler, Disponível em: <<http://www.agiledata.org/essays/mappingObjects.html>>. Acesso em: 06 nov. 2004.

APACHE 2005. **The Apache Software Foundation**. Disponível em: <<http://www.apache.org>>. Acesso em: 15 out. 2005.

AULANET 2005. **AulaNet**. Disponível em: <<http://www.aulanet.com.br>>. Acesso em: 15 out. 2005.

BRASIL. MEC: Secretaria de Educação Superior. Comissão de Especialistas de Ensino de Computação e Informática – CEECInf. **Diretrizes Curriculares de cursos da área de computação e informática (1998)**. Disponível em: <http://www.mec.gov.br/sesu/ftp/curdiretriz/computacao/co_diretriz.rtf>. Acesso em: 15 dez. 2004.

BRUSILOVSKY, P. **Adaptive Educational Systems on the World Wide Web: A Review of Available Technologies**, in Proc. of workshop "WWW-Based Tutoring" at 4 th Intern. Conference on Intelligent Tutoring Systems (ITS'98), 1998.

CASTRO, T. H. C.; CASTRO JR., A. N. MENEZES, C. S. **Aprende - um Ambiente Cooperativo de Apoio à Aprendizagem de Programação**. In: Simpósio Brasileiro de Informática na Educação, 15., 2004, Manaus-AM, **Anais...** Manaus: EDUA, 2004, v 1, p.71-79.

CASTRO, T. H. C.; CASTRO JR., A. N. MENEZES, C. S. **Representando padrões em um ambiente de apoio a aprendizagem de programação**. In: Simpósio Brasileiro de Informática na Educação, 13., 2002, São Leopoldo-RS, **Anais...** São Leopoldo: UNISINOS, 2002, p.558-561.

CLEMENTS, R. B., J.; C. FLANAGAN, M. FLATT, S. KRISHNAMURTHI, P. STECKLER AND M. FELLEISEN. DRSCHEME. **A programming environment for Scheme**. Journal of Functional Programming, 2001.

COLEMAN, J. P. **AsseSQL-An Online Tool to Test SQL Query Formulation Skills**. Australasian Computing Education Conference , 2003, Adelaide, Australia, 2003.

- FAVERO, E. L. et al. **AmAm: Ambiente de Aprendizagem Multiparadigmático**. Disponível em: <<http://www.deec.ufpa.br/gped/publicacoes.htm>>. Acesso em: 10 maio 2003.
- FREIRE, JOSÉ A.S. et al. **IETS: Interactive Environment for Teaching SQL**. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 2004, Santos. 2004.
- GAMMA, E. et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley, 1994.
- GAVA, T. B. S. **Um Framework para Organização e Integração de Conhecimento em Ambientes Virtuais de Aprendizagem**. Exame de qualificação. Universidade Federal do Espírito-Santo. 2002.
- HARB, M.P.A.A. et al. **AmAm: ambiente de aprendizagem multiparadigmático**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. 14., 2003, NCE-UFRJ. P. 223-232.
- HAYNES, CHRISTOPHER T., **Experience with an Analytic Approach to Teaching Programming Languages**. Proceedings of the 29th. SIGCSE Technical Symposium on Computer Science Education, Vol. 30, 1998. Findler.
- HEARST, MARTI A. **The debate on automated essay grading**. University of California, Berkeley. <hearst@sims.berkeley.edu>.
- LIMA, P. S. R. BRITO, S. R. FONSECA, O. FAVERO, E. L. **Adaptação de Interfaces em Ambientes Virtuais de Aprendizagem com Foco na Construção Dinâmica de Comunidades**. Revista Novas Tecnologias na Educação (RENOTE). Porto Alegre: UFRGS. v.3, n.1, maio, 2005.
- MITROVIC, A. (1998). **A Knowledge-Based Teaching System for SQL**, Proc. ED-MEDIA'98, T. Ottmann, I. Tomek (eds.), 1027-1032.
- MITROVIC, A.; HAUSLER, K. **Porting SQL-Tutor to the Web**. Proc. ITS'2000 workshop on Adaptive and Intelligent Web-based Education Systems, p.37-44, 2000.
- MITROVIC, A.; MARTIN, B.; MAYO, M. **Using evaluation to shape ITS design: Results and Experiences with SQL-Tutor**. Int. J. User Modeling and User-Adapted Interaction, v. 12, n.2-3, p.243-279, 2002.
- MORAES, C.D; VACCARO, G.L.R.; FERREIRA, A.L.A. **Usando recursos computacionais no ensino de matemática**. In: Simpósio Brasileiro de Informática na Educação, 7., 1997, São José dos Campos-SP. **Anais...** São José dos Campos: ITA, 1997.

- MYSQL. **MySQL**. Disponível em:<<http://www.mysql.com>>. Acesso em: 20 set. 2004.
- OHLSSON, S. **Constraint-based Student Modelling. Proc. Student Modelling: the Key to Individualized Knowledge-based Instruction**, Berlin, Springer-Verlag, pp. 167-189, 1994.
- PREECE, J. **Online Communities: Designing Usability and Supporting Socialbility**. John Wiley & Sons, Inc., New York, NY, 2000.
- PHP. **PHP: Hypertext Pre-processor**. Disponível em:<<http://www.php.net>>. Acesso em: 21 set. 2005.
- RORIGUES, J. A., BARBOSA, P. R., BRITO, A.F. **O Papel da Pedagogia e da Andragogia no Ensino de Computação em Engenharia**. Proceedings of the International Conference on Engineering and Computer Education (ICECE2000). São Paulo/BR, 2000.
- SADIQ, SHAZIA et al. **SQLator-An Online SQL Learning Workbench**. In: ITi CSE, 2004, Leeds, United Kingdom, 2004.
- SALTON, G. **The SMART retrieval system - experiments in automatic document processing**. Prentice-Hall, Englewood Cliffs N.J. ,1971.
- SHERMIS, M.D.; BURSTEIN, J.C. **Automated essay scoring.**, 2003.
- SILVA, A. S. et al. **Avaliação automática de consultas SQL em ambiente virtual de ensino-aprendizagem** (em andamento). RBIE - Revista Brasileira de Informática na Educação. 2006.
- SILVA, A.S.; BRITO, S.R.; FAVERO, E.L.; HERNÁNDEZ-DOMÍNGUEZ, A. TAVARES, O.L.; FRANCES, C.R.L. **Uma arquitetura para desenvolvimento de ambientes interativos de aprendizagem baseado em agentes, componentes e framework**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. 14., 2003, NCE-FRJ.
- SOUZA, R.R. **Usando mapas conceituais na educação informatizada rumo a um aprendizado significativo**. Disponível em: <<http://www.edutecnet.com.br/biblioteca/virtual/edtextos.htm>>. Acesso em: 15 dez. 2004.
- SUKKARIEH1, JANA Z.; PULMAN1, STEPHEN G.; RAIKES2, NICHOLAS. **Auto-marking: using computational linguistics to score short, free text responses**. TCX. MySQL Reference Manual for version 4.0.3, 2002. Disponível em: <http://www.mysql.com/documentation/mysql/bychapter/manual_Introduction.html/Features>. Acesso em: 28 dez. 2004.

TCX. MySQL Reference Manual for version 4.0.3, 2002. Disponível em: <http://www.mysql.com/documentation/mysql/bychapter/manual_Introduction.html/Features>. Acesso em: 28 dez. 2004.

TELEDUC. Disponível em: <<http://teleduc.nied.unicamp.br/teleduc>>. Acesso em: 01 jul. 2005.

APÊNDICE A – CÓDIGOS-FONTES DA FERRAMENTA

Os códigos-fontes descritos neste Apêndice fazem referência ao funcionamento apenas do cadastro de cursos na ferramenta IETS. Os outros cadastros estão implementados na ferramenta.

O código-fonte total deste exemplo é de aproximadamente 334 linhas em linguagem PHP, contendo também, códigos em linguagem SQL para a manipulação do banco de dados.

Os programas “**Class.Fachada.php**”, “**Class.Conexao.php**” e “**Class.CadBD.php**”, estão presentes no funcionamento dos cadastros e consultas existentes na ferramenta IETS.

```

iets_CadCurso.php
1  <?
2  // ClassGenerator 1.5 Lite - www.classgenerator.net
3  // Classe gerada em 21/09/2004 10:10:04
4  include_once dirname(__FILE__).'/classes/class.Fachada.php';
5  session_start();
6  $fachada = new Fachada();
7  if($_POST['btnSalvar']){
8  $curso = new Curso($_POST['txtCdInst'], $_POST['txtCdTipoCurso'],
9      $_POST['txtCdCurso'], $_POST['txtNome']);
10 $res = $fachada->inserirCurso($curso);
11 $msg = ($res)?(Util::getMsg('S')):(Util::getMsg('E'));
12 header('location:admCurso.php?msg='.urlencode($msg));
13 }
14 ?>
15 <HTML>
16 <HEAD><? include_once dirname(__FILE__).'/template/templateHeaders.php';?>
17 </HEAD>
18 <BODY>
19 <? include_once dirname(__FILE__).'/template/templateComeco.php';?>
20 <div id='divConteudo'>
21 <FORM name='frmCurso' method='POST' action='cadCurso.php' onSubmit='return
22 checarDados(this)'>
23 <table><caption>Cadastro Curso</caption>
24 <tr>
25 <td class='itemForm'>CdInst:</td>
26 <td><INPUT type='text' name='txtCdInst' size='60' value="" maxlength='40'
27 class='textForm'></td>
28 </tr>
29 <tr>
30 <td class='itemForm'>CdTipoCurso:</td><td><INPUT type='text' name='txtCdTipoCurso'
31 size='60' value="" maxlength='40' class='textForm'></td>
32 </tr>
33 <tr>
34 <td class='itemForm'>CdCurso:</td><td><INPUT type='text' name='txtCdCurso'
35 size='60' value="" maxlength='40' class='textForm'></td>
36 </tr>
37 <tr>
38 <td class='itemForm'>Nome:</td>
39 <td><INPUT type='text' name='txtNome' size='60' value="" maxlength='40'
40 class='textForm'></td>
41 </tr>
42 <tr>
43 <td colspan='2'><INPUT type='submit' name='btnSalvar' value='Salvar'
44 class='buttonForm'></td>
45 </tr>
46 </table>
47 </form>
48 </div><? include_once dirname(__FILE__).'/template/templateFim.php';?>
49 </BODY>
50 </HTML>

```

Listagem A.1 Código gerado para a página PHP de Cadastro de Curso

```
Class.Fachada.php

1 <?
2 // ClassGenerator 1.5 Lite - www.classgenerator.net
3 // Classe gerada em 21/09/2004 10:10:04
4 ...
5 include_once dirname(__FILE__).'/class.CadCurso.php';
6 ...
7 include_once dirname(__FILE__).'/class.Conexao.php';
8 include_once dirname(__FILE__).'/class.Util.php';

9 class Fachada {
10 var $conexao;
11 function Fachada() {
12 $this->conexao = NULL;
13 }
14 function getConexao() {
15 return $this->conexao;
16 }
17 function iniciarTransacao() {
18 $this->conexao = new Conexao();
19 $this->conexao->iniciarTrans();
20 }
21 function commitTransacao() {
22 $this->conexao->commitTrans();
23 $this->conexao = NULL;
24 }
25 function rollBackTransacao() {
26 $this->conexao->rollBackTrans();
27 $this->conexao = NULL;
28 }
29 ...
30 // Métodos da Classe Curso
31 function inserirCurso($curso) {
32 $cadCurso = new CadCurso($this->getConexao());
33 return $cadCurso->inserirCurso($curso);
34 }
35 function alterarCurso($curso) {
36 $cadCurso = new CadCurso($this->getConexao());
37 return $cadCurso->alterarCurso($curso);
38 }
39 function excluirCurso($curso) {
40 $cadCurso = new CadCurso($this->getConexao());
41 return $cadCurso->excluirCurso($curso);
42 }
43 function getCurso($cdInst, $cdTipoCurso, $cdCurso) {
44 $cadCurso = new CadCurso();
45 return $cadCurso->getCurso($cdInst, $cdTipoCurso, $cdCurso);
46 }
47 function getAllCurso($ini=NULL, $num=NULL) {
48 $cadCurso = new CadCurso();
49 return $cadCurso->getAllCurso($ini, $num);
50 }
51 ...
52 }
53 ?>
```

Listagem A.2 Parte do Código da Classe FACHADA

```
Class.CadCurso.php
1 <?
2 // ClassGenerator 1.5 Lite - www.classgenerator.net
3 // Classe gerada em 21/09/2004 10:10:04
4 include_once dirname(__FILE__).'/class.CadCursoBD.php';
5 include_once dirname(__FILE__).'/class.Curso.php';
6 class CadCurso{
7 var $cadCursoBD;
8 function CadCurso($conexao = "") {
9 $this->cadCursoBD = new CadCursoBD($conexao);
10 }
11 function inserirCurso($curso) {
12 return $this->cadCursoBD->inserirCurso($curso->getCdInst(), $curso->getCdTipoCurso(),
13     $curso->getCdCurso(), $curso->getNome());
14 }
15 function alterarCurso($curso) {
16 return $this->cadCursoBD->alterarCurso($curso->getCdInst(), $curso->getCdTipoCurso(),
17     $curso->getCdCurso(), $curso->getNome());
18 }
19 function excluirCurso($curso) {
20 return $this->cadCursoBD->excluirCurso($curso->getCdInst(), $curso->getCdTipoCurso(),
21     $curso->getCdCurso());
22 }
23 function getCurso($cdInst, $cdTipoCurso, $cdCurso) {
24 if($rs = $this->cadCursoBD->getCurso($cdInst, $cdTipoCurso, $cdCurso)) {
25     $va = array_shift($rs);
26     return new Curso($va['cdinst'], $va['cdtipocurso'], $va['cdcurso'], $va['nome']);
27 }else
28     return false;
29 }
30 function getAllCurso() {
31 if($rs = $this->cadCursoBD->getAllCurso()){
32 while($va = array_shift($rs))
33     $vet[] = new Curso($va['cdinst'], $va['cdtipocurso'], $va['cdcurso'], $va['nome']);
34 return $vet;
35 }else
36 return false;
37 }
38 }
39 ?>
```

Listagem A.3 Código gerado para a classe CadCurso

```
Class.Curso.php
1 <?
2 // ClassGenerator 1.5 Lite - www.classgenerator.net
3 // Classe gerada em 21/09/2004 10:10:04
4 class Curso{
5     var $cdInst;
6     var $cdTipoCurso;
7     var $cdCurso;
8     var $nome;
9     function Curso($cdInst, $cdTipoCurso, $cdCurso, $nome){
10     $this->cdInst = $cdInst;
11     $this->cdTipoCurso = $cdTipoCurso;
12     $this->cdCurso = $cdCurso;
13     $this->nome = $nome;
14     }
15     function setAll($nome = false){
16     if($nome != false)
17     $this->setNome($nome);
18     }
19     function getCdInst(){ return $this->cdInst;}
20     function getCdTipoCurso(){ return $this->cdTipoCurso;}
21     function getCdCurso(){ return $this->cdCurso;}
22     function getNome(){ return $this->nome;}
23     function setCdInst($x){ $this->cdInst = $x;}
24     function setCdTipoCurso($x){ $this->cdTipoCurso = $x;}
25     function setCdCurso($x){ $this->cdCurso = $x;}
26     function setNome($x){ $this->nome = $x;}
27     }
28 ?>
```

Listagem A.4 Código gerado para a classe Curso

```

Class.CadCursoBD.php
1  <?
2  // ClassGenerator 1.5 Lite - www.classgenerator.net
3  // Classe gerada em 21/09/2004 10:10:04

4  include_once dirname(__FILE__).'/class.CadBD.php';
5  class CadCursoBD extends CadBD {

6  function CadCursoBD($conexao = '') {
7  $this->CadBD($conexao);
8  }
9  function inserirCurso($cdInst, $cdTipoCurso, $cdCurso, $nome) {
10 $sql = "INSERT INTO curso
        i.          VALUES ('$cdInst', '$cdTipoCurso', '$cdCurso', '$nome)";
11 return $this->executarIUD($sql);
12 }
13 function alterarCurso($cdInst, $cdTipoCurso, $cdCurso, $nome) {
14 $sql = "UPDATE curso
        SET
        nome = '$nome'
        WHERE
        cdInst = '$cdInst' AND
        cdTipoCurso = '$cdTipoCurso' AND
        cdCurso = '$cdCurso'";
15 return $this->executarIUD($sql);
16 }
17 function excluirCurso($cdInst, $cdTipoCurso, $cdCurso) {
18 $sql = "DELETE FROM curso
        WHERE
        cdInst = '$cdInst' AND
        cdTipoCurso = '$cdTipoCurso' AND
        cdCurso = '$cdCurso'";
19 return $this->executarIUD($sql);
20 }
21 function getCurso($cdInst, $cdTipoCurso, $cdCurso) {
22 $sql = "SELECT * FROM curso
        WHERE
        cdInst = '$cdInst' AND
        cdTipoCurso = '$cdTipoCurso' AND
        cdCurso = '$cdCurso'
        ORDER BY
        cdInst, cdTipoCurso, cdCurso";
23 $rs = $this->con->execute($sql);
24 return $this->con->fetch_array($rs);
25 }
26 function getAllCurso() {
27 $sql = "SELECT * FROM curso
        ORDER BY
        cdInst, cdTipoCurso, cdCurso";
28 $rs = $this->con->execute($sql);
29 return $this->con->fetch_array($rs);
30 }
31 }
32 ?>

```

Listagem A.5 Código gerado para a classe derivada responsável pelas operações básicas de manipulação de dados

```
Class.CadBD.php
1 <?
2 // ClassGenerator 1.5 Lite - www.classgenerator.net
3 // Classe gerada em 02/05/2004 22:53:09
4 include_once dirname(__FILE__).'/class.Conexao.php';
5 class CadBD{
6     var $con;
7     function CadBD($conexao){
8         if($conexao)
9             $this->con = $conexao;
10        else
11            $this->con = new Conexao();
12        }
13    function executarIUD($sql){
14        if($rs = $this->con->execute($sql)){
15            $rs->Close();
16            if($this->con->getAutoCommit())
17                $this->con->close();
18            return true;
19        }else
20            return false;
21        }
22    function executarInsAuto($sql){
23        if($rs = $this->con->execute($sql)){
24            $sid = $this->con->insertId();
25            $rs->Close();
26            if($this->con->getAutoCommit())
27                $this->con->close();
28            return $sid;
29        }else
30            return false;
31        }
32    }
33    ?>
```

Listagem A.6 Código gerado para a classe genérica CadBD

Class.Conexao.php

```
1 <?
2 include_once dirname(__FILE__)."/adodb/adodb.inc.php";
3 class Conexao {
4     var $conexao;
5     var $usuario;
6     var $senha;
7     var $bdPath;
8     var $bd;
9     var $tipoBd;
10    var $charSet;
11    var $buffers;
12    var $dialect;
13    var $autoCommit;
14    function Conexao() {
15        $this->usuario = "root";
16        $this->senha = "";
17        $this->bdPath = "localhost";
18        $this->bd = "iets";
19        $this->tipoBd = "mysql";
20        $this->charSet = "ASCII";
21        $this->buffers = 100;
22        $this->dialect = 3;
23        $this->autoCommit = true;
24        // Cria uma instancia do objeto
25        // Configura uma conexao com a biblioteca
26        $this->conexao = NewADOConnection($this->tipoBd);
27        $this->conexao->charSet = $this->charSet;
28        $this->conexao->buffers = $this->buffers;
29        $this->conexao->dialect = $this->dialect;
30        // Abre a conexao com o banco
31        if (!$this->conexao->Connect($this->bdPath,$this->usuario,$this->senha,$this->bd)) {
32            header("Location: manutencao.php");
33        }
34    function &fetch_array(&$rs) {
35        $numeroColunas = $rs->FieldCount();
36        while (!$rs->EOF) {
37            for($i=0;$i < $numeroColunas;$i++){
38                $coluna = $rs->FetchField($i);
39                $nomeColuna = strtolower($coluna->name);
40                $tipoColuna = $rs->MetaType($coluna->type);
41                if ( ($tipoColuna == 'D') || ($tipoColuna == 'T') ){
42                    $vetor["$nomeColuna"] = $this->converteData($rs->fields[$i]);
43                }else{
44                    $vetor["$nomeColuna"] = $rs->fields[$i];
45                }
46            }
47            $vetorRegistro[] = $vetor;
48            $rs->MoveNext();
49        }
50        return $vetorRegistro;
51    }
52    function converteData($data){
```

```

50     return $this->converteAmdParaDma($data);
51 }
52 function &converteMdaParaDma(&$data){
53     $data = substr($data,0,10);
54     list($mes,$dia,$ano) = explode("/", $data);
55     $data = $dia."/".$mes."/".$ano;
56     return $data;
57 }
58 function &converteDmaParaMda(&$data){
59     $data = substr($data,0,10);
60     list($dia,$mes,$ano) = explode("/", $data);
61     $data = $mes."/".$dia."/".$ano;
62     return $data;
63 }
64 function &converteDmaParaAmd(&$data){
65     $data = substr($data,0,10);
66     list($dia,$mes,$ano) = explode("/", $data);
67     $data = $ano."-".$mes."-".$dia;
68     return $data;
69 }
70 function &converteAmdParaDma(&$data){
71     $data = substr($data,0,10);
72     list($ano,$mes,$dia) = explode("-", $data);
73     $data = $dia."/".$mes."/".$ano;
74     return $data;
75 }
76 function close(){
77     $this->conexao->Close();
78 }
79 function execute($sql){
80     $rs = $this->conexao->Execute($sql) or
            die("Erro na consulta: $sql. <br>" . $this->conexao->ErrorMsg());
81     $_SESSION['lastSQL'] = $sql;
82     return $rs;
83 }
84 function getAutoCommit(){
85     return $this->autoCommit;
86 }
87 function setAutoCommit($bol){
88     $this->autoCommit = $bol;
89 }
90 function insertId(){
91     return $this->conexao->Insert_id();
92 }
93 function iniciarTrans(){
94     $this->setAutoCommit(false);
95     $this->conexao->BeginTrans();
96 }
97 function commitTrans(){
98     $this->setAutoCommit(true);
99     $this->conexao->CommitTrans();

```

```
100 $this->close();
101 }
102 function rollBackTrans(){
103 $this->setAutoCommit(true);
104 $this->conexao->RollbackTrans();
105 $this->close();
106 }
107 ?>
```

Listagem A.7 Código gerado para classe que realiza a conexão com o banco de dados

APÊNDICE B – DIAGRAMAS DE CLASSES DO AmAm

Este apêndice apresenta parte dos diagramas de classes do AmAm.

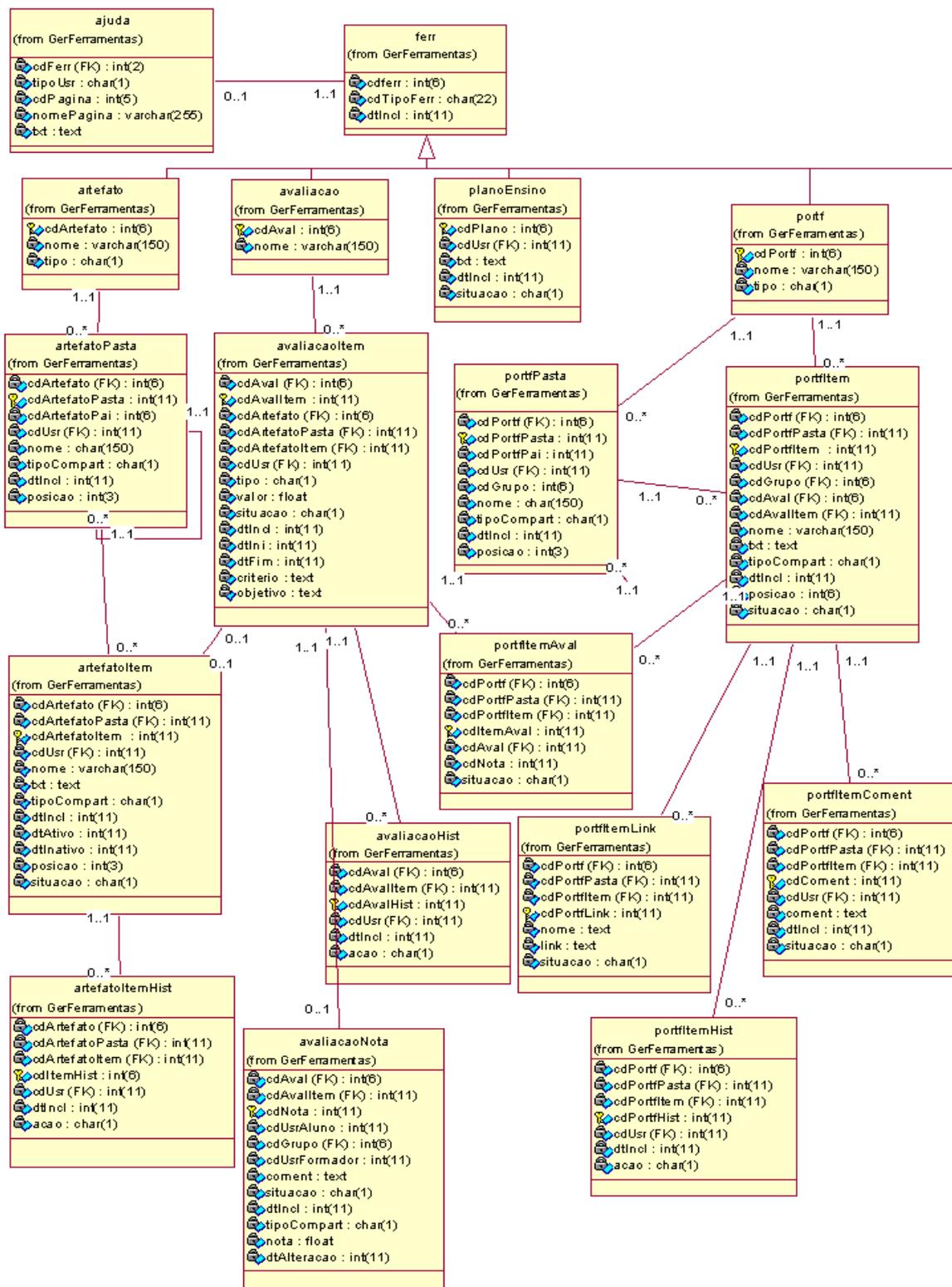


Figura B.1 Diagrama de Classe Geral das Ferramentas – parte I

A Figura B.3 apresenta a parte II do diagrama de classes geral das Ferramentas.

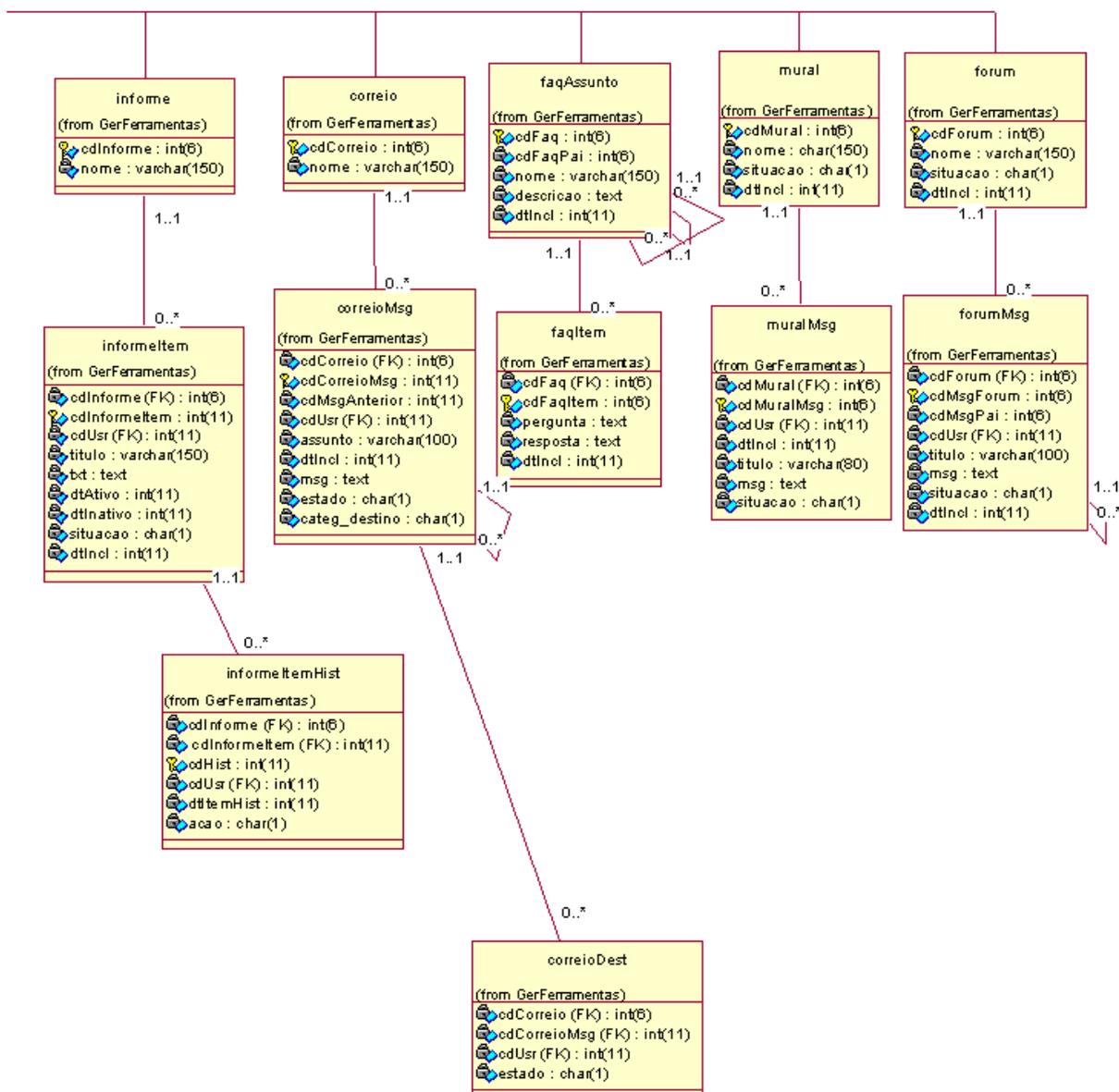


Figura B.3 Diagrama de Classe Geral das Ferramentas – parte II

A Figura B.2 apresenta o diagrama de classes da ferramenta Mural.

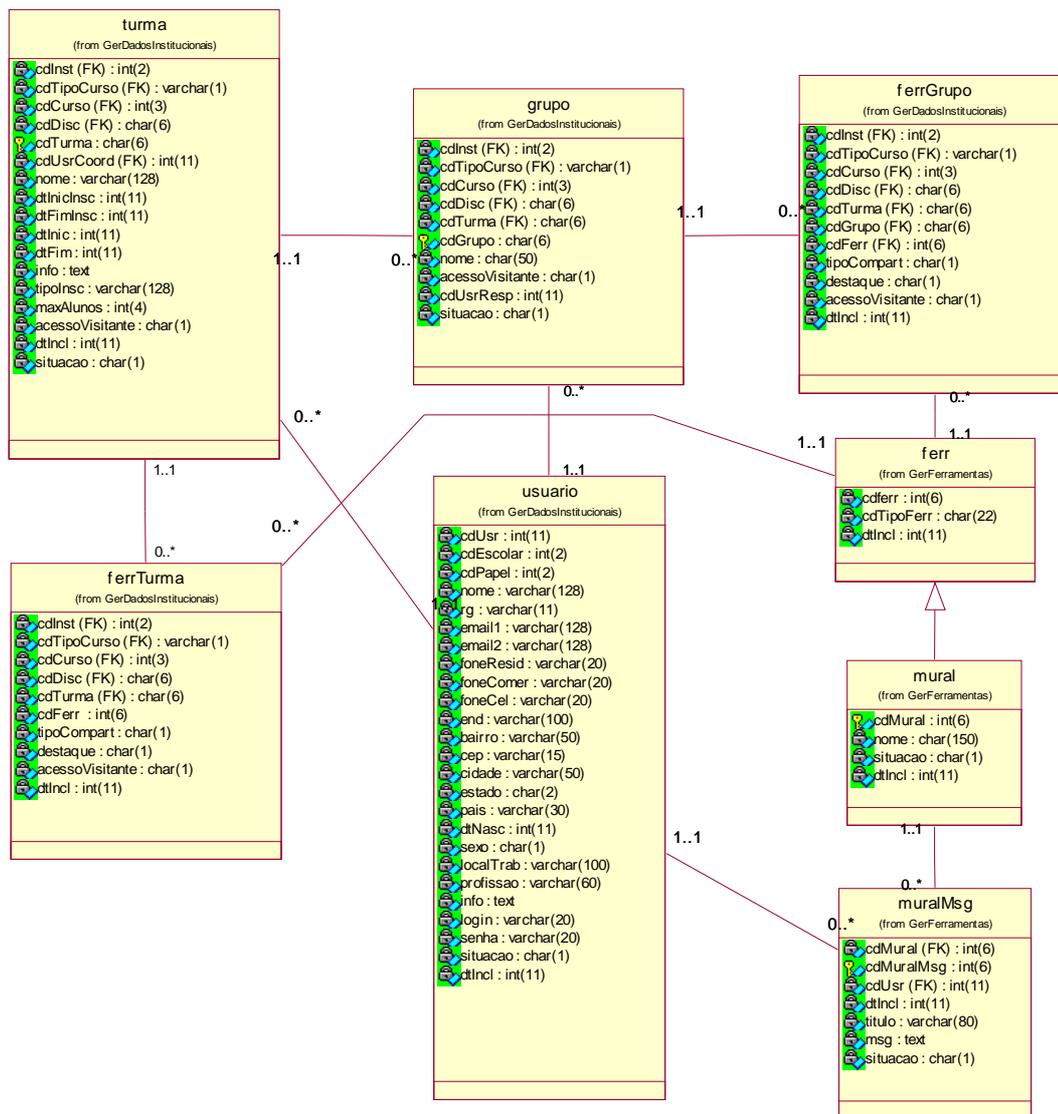


Figura B.2 Diagrama de Classe das Ferramentas do Mural

A Figura B.3 apresenta o diagrama de classes da ferramenta Fórum.

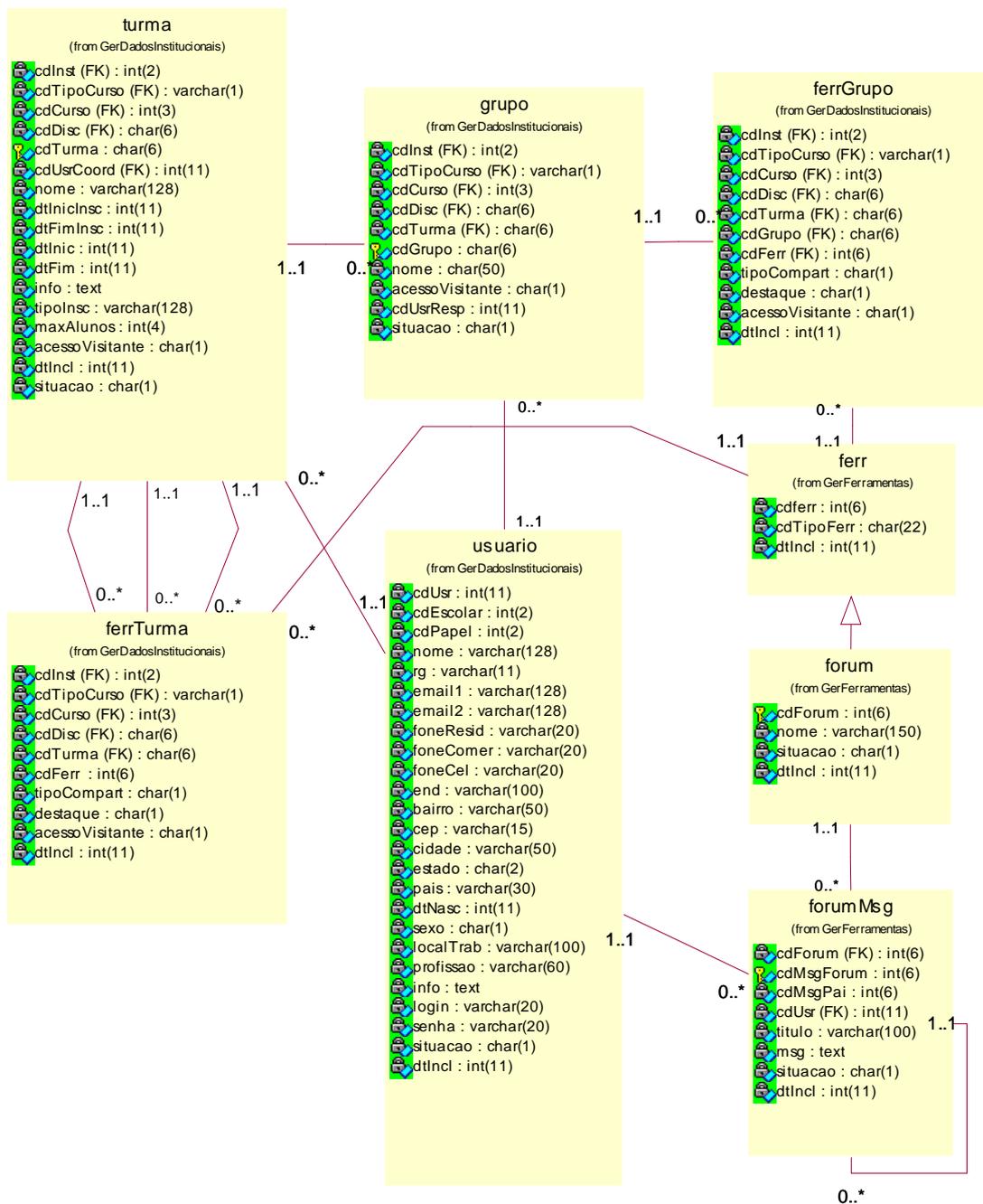


Figura B.3 Diagrama de Classe da Ferramenta Fórum

APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO DO IETS

As tabelas C.1, C.2, C.3, C.4, C.5, C.6 e C.7, detalham os requisitos para os Casos de Uso, destinados as funções básicas desempenhadas pelo módulo de Cadastros Básicos dentro da ferramenta IETS.

Tabela C.1 UC01: Autenticar Usuário

Caso de Uso:	Autenticar Usuário
Objetivo:	Realiza a autenticação do usuário no sistema
Atores:	Aluno, Professor e Administrador
Fluxo Básico:	<ol style="list-style-type: none"> 1. Usuário fornece os dados de login e senha <ol style="list-style-type: none"> a. Se autenticação correta <ol style="list-style-type: none"> i. Exibir opções disponíveis de acordo com as suas permissões b. Se autenticação incorreta <ol style="list-style-type: none"> i. Exibir mensagem de acesso negado.

Tabela C.2 UC02: Cadastrar Aluno

Caso de Uso:	Cadastrar Aluno
Objetivo:	Realiza o cadastramento dos dados pessoais do Aluno
Atores:	Administrador
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O Administrador identificado fornece os dados do Aluno para efetivação do cadastro <ol style="list-style-type: none"> a. Se os dados forem válidos <ol style="list-style-type: none"> i. Inserir dados pessoais no sistema; ii. Mostrar mensagem: dados cadastrados com sucesso b. Senão forem válidos <ol style="list-style-type: none"> i. Mostrar mensagem: falha no cadastro

Tabela C.3 UC03: Cadastrar Professor

Caso de Uso:	Cadastrar Professor
Objetivo:	Cadastrar os professores que irão operar o sistema com o objetivo de avaliar as turmas de uma determinada disciplina.
Ator(es):	Administrador
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O Administrador seleciona a opção de Adicionar Professor <ol style="list-style-type: none"> a. Se já existir problema cadastrado <ol style="list-style-type: none"> i. Exibir Mensagem: Cadastro já Existente b. Senão existir o Conteúdo cadastrado <ol style="list-style-type: none"> i. Dados são informados para o cadastro de Professor <ol style="list-style-type: none"> 1. Se os dados forem válidos <ol style="list-style-type: none"> a. Inserir dados do conteúdo no sistema; b. Mostrar mensagem: dados cadastrados com sucesso 2. Senão forem válidos <ol style="list-style-type: none"> a. Mostrar mensagem: falha no cadastro

Tabela C.4 UC05: Cadastrar Curso

Caso de Uso:	Cadastrar Curso
Objetivo:	Cadastra um novo curso para uma instituição com a finalidade de agrupar as turmas para a realização de treinamentos e avaliações
Ator(es):	Administrador
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. Administrador identificado fornece os dados do curso para efetivação do cadastro <ol style="list-style-type: none"> a. Se os dados forem válidos <ol style="list-style-type: none"> i. Inserir dados do curso no sistema; ii. Mostrar mensagem: dados cadastrados com sucesso b. Senão forem válidos <ol style="list-style-type: none"> i. Mostrar mensagem: falha no cadastro

Tabela C.5 UC04: Cadastrar Instituição

Caso de Uso:	Cadastrar Instituição
Objetivo:	Cadastra uma nova instituição aonde serão aplicadas as avaliações e treinamentos.
Ator(es):	Administrador
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O Administrador identificado fornece os dados da instituição para efetivação do cadastro <ol style="list-style-type: none"> a. Se os dados forem válidos <ol style="list-style-type: none"> i. Inserir dados da instituição no sistema; ii. Mostrar mensagem: dados cadastrados com sucesso b. Senão forem válidos <ol style="list-style-type: none"> i. Mostrar mensagem: falha no cadastro

Tabela C.6 UC06: Cadastrar Disciplina

Caso de Uso:	Cadastrar Disciplina
Objetivo:	Cadastrar as disciplinas que serão alocadas para o ambiente, apesar de utilizar como alvo principal a disciplina de Banco de Dados. Todas estas disciplinas serão relacionadas ao ensino de SQL
Ator(es):	Administrador
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. Administrador identificado fornece os dados da disciplina para efetivação do cadastro <ol style="list-style-type: none"> a. Se os dados forem válidos <ol style="list-style-type: none"> i. Inserir dados da disciplina no sistema; ii. Mostrar mensagem: dados cadastrados com sucesso b. Senão forem válidos <ol style="list-style-type: none"> i. Mostrar mensagem: falha no cadastro

Tabela C.7 UC07: Cadastrar Turma

Caso de Uso:	Cadastrar turma
Objetivo:	Cadastrar novas turmas de uma disciplina
Ator(es):	Administrador
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O Administrador identificado fornece os dados da turma para efetivação do cadastro <ol style="list-style-type: none"> a. Se os dados forem válidos <ol style="list-style-type: none"> i. Inserir dados da turma no sistema; ii. Mostrar mensagem:dados cadastrados com sucesso b. Senão forem válidos <ol style="list-style-type: none"> i. Mostrar mensagem: falha no cadastro

A tabela C.8 descreve o Caso de Uso destinado ao cadastro do perfil da turma, de acordo com a seleção da turma realizada pelo professor da disciplina.

Tabela C.8 UC08: Cadastrar Perfil Turma

Caso de Uso:	Cadastrar Perfil Turma
Objetivo:	Criar um perfil para determinada turma na ferramenta
Ator(es):	Professor
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O sistema lista as turmas para seleção 3. O professor seleciona uma turma na lista 4. O sistema informa o sumário contendo os módulos com suas respectivas sessões e sub-sessões (Extend: Listar Sumário Disciplina). 5. O professor, então realiza a seleção das opções. 6. O sistema grava o perfil para turma selecionada.

A tabela C.9 esboça a descrição do Caso de Uso de extensão denominado “Listar Sumário Disciplina” para o caso de Uso “Cadastrar Perfil Turma”.

Tabela C.9 UC09: Listar Sumário Disciplina

Caso de Uso:	Listar Sumário Disciplina
Objetivo:	(Caso de Uso de Extensão) realiza a listagem dos módulos, sessões que compõe parte do ambiente de uma determinada disciplina.
Ator(es):	Aluno
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O sistema lista os itens do sumário para que o aluno comece a operação no sistema.

A tabela C.10 objetiva mostrar o Caso de Uso “Cadastrar Conteúdo” , tendo com finalidade a visualização dos passos necessários para o cadastramento do conteúdo, que deverá compor o sumário do IETS para determinada turma ou aluno.

Tabela C.10 UC10: Cadastrar Conteúdo

Caso de Uso:	Cadastrar Conteúdo
Objetivo:	Cadastrar Conteúdos que comporão o sumário do ferramenta IETS, este conteúdo será selecionado de acordo com o perfil de aluno imposto pelo professor da turma.
Ator(es):	Professor
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O Professor seleciona a opção de Adicionar conteúdo <ol style="list-style-type: none"> a. Se já existir Conteúdo cadastrado <ol style="list-style-type: none"> i. Exibir Mensagem: Cadastro já Existente b. Senão existir o Conteúdo cadastrado <ol style="list-style-type: none"> i. Dados são informados para o cadastro de Conteúdo <ol style="list-style-type: none"> 1. Se os dados forem válidos <ol style="list-style-type: none"> a. Inserir dados do conteúdo no sistema; b. Mostrar mensagem: dados cadastrados com sucesso 2. Senão forem válidos <ol style="list-style-type: none"> a. Mostrar mensagem: falha no cadastro

A tabela C.11 mostra os passos necessários para o cadastramento de um problema que poderá ser utilizado numa sub-sessão.

Tabela C.11 UC11: Cadastrar Problema

Caso de Uso:	Cadastrar Problema
Objetivo:	Realiza o cadastramento dos problemas que serão utilizados em cada sub-sessão da ferramenta IETS, estes problemas poderão ser de quatro tipos: Teórico: V ou F, Alternativas e Textual; Prático: Execução de comandos SQL.
Ator(es):	Professor
Fluxo Básico:	<ol style="list-style-type: none"> 2. INCLUDE Autenticar Usuário 3. O Professor seleciona a opção de Adicionar Problema <ol style="list-style-type: none"> a. Se já existir problema cadastrado <ol style="list-style-type: none"> i. Exibir Mensagem: Cadastro já Existente b. Senão existir o Problema cadastrado <ol style="list-style-type: none"> i. Dados são informados para o cadastro de Problema <ol style="list-style-type: none"> 1. Se os dados forem válidos <ol style="list-style-type: none"> a. Inserir dados do problema no sistema; b. Mostrar mensagem: dados cadastrados com sucesso 2. Senão forem válidos <ol style="list-style-type: none"> a. Mostrar mensagem: falha no cadastro

A tabela C.12 ilustra o Caso de Uso “Cadastrar Solução Usuário”, que tem como finalidade direcionar os passos necessários para a manutenção dos dados da solução do usuário para os problemas cadastrados na ferramenta.

Tabela C.12 UC12: Cadastrar Solução Usuário

Caso de Uso:	Cadastrar Solução Usuário
Objetivo:	Realiza o cadastramento dos dados referentes a solução do usuário para determinado problema que será resolvido pelo aluno em alguma avaliação ou treinamento.
Ator(es):	Professor
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O Professor seleciona a opção de Adicionar Solução Usuário <ol style="list-style-type: none"> a. Se já existir a solução cadastrada <ol style="list-style-type: none"> i. Exibir Mensagem: Cadastro já Existente b. Senão existir a solução cadastrada <ol style="list-style-type: none"> i. Dados são informados para o cadastro de Solução Usuário <ol style="list-style-type: none"> 1. Se os dados forem válidos <ol style="list-style-type: none"> a. Inserir dados da Solução do Usuário no sistema; b. Mostrar mensagem: dados cadastrados com sucesso 2. Senão forem válidos <ol style="list-style-type: none"> a. Mostrar mensagem: falha no cadastro

A tabela C.13 mostra os passos para a realização da geração do mapeamento da utilização da ferramenta pelo aluno.

Tabela C.13 UC13: Gerar Mapeamento Acesso

Caso de Uso:	Gerar Mapeamento Acesso
Objetivo:	Realiza a geração do mapeamento dos passos que um aluno realiza no sistema (passos: autenticação, acesso sessões e sub-sessões)
Ator(es):	Aluno
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O sistema registra o acesso do aluno ao sistema, sessões e sub-sessões para auxiliar o professor no aprendizado do aluno.

A tabela C.14 ilustra os passos necessários para a Consulta da avaliação realiza numa determinada turma pelo professor da disciplina.

Tabela C.14 UC14: Consultar Desempenho Turma

Caso de Uso:	Consultar Desempenho Turma
Objetivo:	Mostrar ao professor o desempenho de uma determinada turma em relação a avaliação realizada, esta consulta poderá ser visualizada na forma completa aonde todas as tentativas são visualizadas, ou na forma reduzida, no qual são visualizadas apenas informações acerca da pontuação final alcançada por cada aluno.
Ator(es):	Professor
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O professor seleciona opção Ver resultados 3. O sistema lista as turmas no qual o professor esta alocado e que existam avaliações a serem consultadas 4. O professor seleciona a turma desejada 5. O sistema lista as avaliações cadastradas para a turma selecionada anteriormente 6. O professor seleciona a avaliação 7. O sistema exibe um painel contendo as pontuações alcançadas nas questões selecionadas pelo professor, analisa as questões desenvolvidas por cada aluno. Podendo interagir com os alunos. (Extend: Interagir Aluno)

A tabela C.15 ilustra o Caso de Uso de Extensão que é utilizado pelo Caso de Uso “Consultar Desempenho Turma”, sendo ativado quando o professor clicar numa questão do painel do aluno, comentando-a.

Tabela C.15 UC15: Interagir Aluno

Caso de Uso:	Interagir Aluno
Objetivo:	(Caso de Uso de Extensão) permite a interação do professor com a turma escrevendo um comentário em qualquer questão já concluída pelo aluno.
Ator(es):	Professor
Fluxo Básico:	<ol style="list-style-type: none"> 1. O Professor seleciona a questão que deseja comentar no painel de pontuação, devendo escolher apenas a questão que seja diferente da imagem (arquivo GIF) de interrogação. 2. O sistema exibe um formulário informando o tipo de questão (teórica: V ou F, Alternativa ou Textual; Prática: Comando SQL), a sua descrição, o N° da tentativa e a resposta informada pelo aluno. 3. O professor escreve e envia o comentário levando em consideração a resposta do aluno 4. O sistema mostra no painel a questão já comentada pelo professor.

A tabela C.16 mostra o Caso de Uso destinado a ilustrar os passos para o cadastramento e geração de dados para a avaliação de uma turma.

Tabela C.16 UC16: Cadastrar / Gerar Avaliação

Caso de Uso:	Cadastrar / Gerar Avaliação
Objetivo:	Cadastrar e Gerar dados para a avaliação que será realizada por determinada turma no qual o professor está alocado
Ator(es):	Professor
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. Professor seleciona a opção de Gerar Avaliação <ol style="list-style-type: none"> a. Se já existir o cadastrado de Avaliação <ol style="list-style-type: none"> ii. Exibir Mensagem: Avaliação já foi Cadastrada e gerada para esta turma b. Senão existir a Avaliação cadastrada <ol style="list-style-type: none"> i. Dados são informados para o cadastro de Avaliação <ol style="list-style-type: none"> 1. Se os dados forem válidos <ol style="list-style-type: none"> a. Inserir dados da avaliação no sistema; b. Mostrar mensagem: geração de dados para avaliação iniciada. c. Mostrar mensagem: dados da avaliação cadastrados / gerados com sucesso. 2. Senão forem válidos <ol style="list-style-type: none"> a. Mostrar mensagem: falha no cadastro.

A tabela C.17 ilustra o Caso de Uso “Informar Resposta Problema”, que tem como finalidade informar os passos necessários para o desenvolvimento da resposta do aluno numa avaliação ou treinamento desenvolvido na ferramenta.

Tabela C.17 UC17: Informar Resposta Problema

Caso de Uso:	Informar Resposta Problema
Objetivo:	Quando as perguntas estão prontas para uma determinada turma, os alunos registram suas respostas para um determinado treinamento ou atividade com o objetivo de medir seu aprendizado.
Ator(es):	Aluno
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O aluno clica na opção exercitar localizada dentro de uma sub-sessão ou pelo sumário 3. O sistema mostrará uma janela com as informações acerca do número da questão o qual o aluno está posicionado naquele momento, o nível de dificuldade da questão, além da descrição do problema e uma caixa de entrada de dados para registrar a resposta. 4. O aluno informa a resposta a pergunta atual. 5. O sistema registra a resposta e envia uma mensagem ao aluno informando o resultado: correta ou incorreta, além de informar através de uma cor a pontuação aproximada do aluno na questão. 6. O sistema atualiza o sumário com as informações acerca da pontuação do aluno (Extend: Exibir Painel Sessão Resposta Aluno) 7. aluno poderá navegar nas tentativas que realizou numa pergunta, com o objetivo de poder refazê-la a qualquer momento. 8. O sistema informará ao aluno o comentário do professor (Extend: Consultar Comentário Professor), caso o mesmo registre-o no sistema. 9. O aluno navega para a próxima pergunta ou anterior, dentro da janela de perguntas.

As tabelas C.18 e C.19 ilustram os Casos de Uso de Extensão que são utilizados com o Caso de Uso visualizado na tabela C.17, o primeiro tendo como finalidade a exibição dos dados da resposta do aluno numa avaliação ou treinamento realizados com a utilização da ferramenta; o segundo Caso de Uso destinado a exibição de um comentário do professor para uma das respostas que o aluno desenvolver numa avaliação dentro da ferramenta.

Tabela C.18 UC18: Exibir Painel Sessão Resposta Aluno

Caso de Uso:	Exibir Painel Sessão Resposta Aluno
Objetivo:	(Caso de Uso de Extensão) atualiza e exibe com o resumo das informações acerca de cada resposta de um determinado aluno, este resumo será visualizado dentro do sumário da ferramenta.
Ator(es):	Aluno
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O sistema informa os dados sobre a situação do aluno na ferramenta dentro do painel que pertence ao sumário. A cada resposta realizada pelo aluno, o painel será atualizado, mostrando sempre informações atualizadas.

Tabela C.19 UC19: Consultar Comentário Professor

Caso de Uso:	Consultar Comentário Professor
Objetivo:	(Caso de Uso de Extensão) Mostrar o comentário realizado pelo professor em uma resposta desenvolvida pelo aluno
Ator(es):	Aluno
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O sistema exibe uma imagem (arquivo GIF) no formulário de resposta do aluno informando a existência de um comentário feito pelo professor, a referida imagem somente aparecerá, caso exista comentário para a resposta do aluno. 3. O Aluno clica na imagem. 4. O sistema exibe o referido comentário.

A tabela C.20 ilustra o Caso de Uso destinado a visualização da trajetória dos alunos no ambiente, para que o professor analise os problemas ocorridos na navegação de um alunos dentro dos conteúdos expostos na ferramenta.

Tabela C.20 UC20: Exibir Mapa Estatístico Acesso

Caso de Uso:	Exibir Mapa Estatístico Acesso
Objetivo:	Mostrar um painel contendo a trajetória do aluno dentro da ferramenta numa determinada sessão de acesso ao sistema
Ator(es):	Professor
Fluxo Básico:	<ol style="list-style-type: none"> 1. INCLUDE Autenticar Usuário 2. O sistema exibe uma lista de turmas para seleção. 4. O professor seleciona uma das turmas. 5. O sistema exibe uma lista de alunos para seleção. 6. O professor escolhe o aluno que será analisado. 7. O sistema exibe todas as sessões de acesso ao sistema. 8. O professor escolhe a sessão desejada. 9. O sistema exibe um mapa do percurso do aluno dentro da ferramenta (ambiente).

APÊNDICE D – PUBLICAÇÃO RELACIONADA COM A DISSERTAÇÃO

1) FREIRE, José Antônio Sarmanho; SILVA, Aleksandra Do Socorro; BRITO, Silvana Rossy de; FAVERO, Elói Luiz; HARB, Maria da Penha de Andrade Abi. *IETS: Interactive Environment for Teaching SQL*. In: *INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION*, 2004, Santos. 2004.

Este trabalho descreve uma ferramenta virtual *Web* para o ensino da linguagem para programação de bancos de dados SQL e foi introduzido no contexto de um ambiente de aprendizado interativo. Este ambiente experimental apresenta três grandes vantagens em relação ao ambiente tradicional para educação à distância: 1) Integração completa do material, com exemplos executáveis dos comandos on-line; 2) Integração completa com um componente para avaliação, que além de avaliar as respostas objetivas tradicionais, mostra a diferença entre as respostas do aluno e a solução ideal e 3) Controle completo das atividades (quantos exemplos foram executados, quantas experiências foram feitas em cada atividade).