

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

LUIZ HENRIQUE GALVÃO SCHNEIGER

**APLICATIVO PARA AUXÍLIO NA APRENDIZAGEM DE CONCEITOS DE
ÁLGEBRA RELACIONAL**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2015**

LUIZ HENRIQUE GALVÃO SCHNEIGER

**APLICATIVO PARA AUXÍLIO NA APRENDIZAGEM DE CONCEITOS DE
ÁLGEBRA RELACIONAL**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

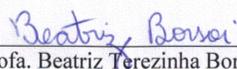
Orientador: Profa. Beatriz Terezinha Borsoi

**PATO BRANCO
2015**

ATA Nº: 005

DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO DO ALUNO LUIZ HENRIQUE GALVÃO SCHNEIGER.

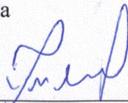
Às 18:35 do dia 25 de novembro de 2015, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (orientadora), Rúbia Eliza de Oliveira Schultz Ascari (convidada) e Fabiano Carniel (convidado), para avaliar o Trabalho de Conclusão de Curso do aluno Luiz Henrique Galvão Schneiger, matrícula 1509187, sob o título **Aplicativo para auxílio na aprendizagem de conceitos de álgebra relacional**; como requisito final para a conclusão da disciplina Trabalho de Conclusão de Curso 2 do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 19:20 hrs foi encerrada a sessão.



Profa. Beatriz Terezinha Borsoi, Dr.
Orientadora



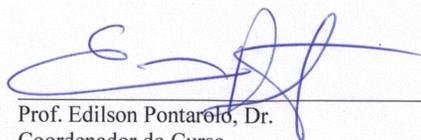
Profa. Rúbia Eliza de Oliveira Schultz Ascari, M.Sc.
Convidada



Prof. Fabiano Carniel, Esp.
Convidado



Profa. Soelaine Rodrigues Ascari, M.Sc.
Coordenador do Trabalho de Conclusão de Curso



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

RESUMO

SCHNEIGER, Luiz Henrique Galvão. Aplicativo para auxílio na aprendizagem de conceitos de Álgebra Relacional. 2015. 59 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

O conhecimento de Álgebra Relacional é essencial para a construção de instruções em *Structured Query Language* (SQL). Embora a Álgebra Relacional esteja diretamente relacionada com a teoria de conjuntos da matemática, em decorrência que as operações realizadas nos dados se referem, também, às operações de conjuntos, o aprendizado desses conceitos pode ser bastante difícil. O alto nível de abstração traz dificuldade para o entendimento dos conceitos e dos resultados produzidos pelas operações. Uma forma de facilitar esse aprendizado poderia ocorrer por meio de ferramentas que permitissem realizar operações de Álgebra Relacional sobre uma base de dados. Assim, por meio da visualização dos resultados o aprendizado poderia ser facilitado. Diante desse contexto, o objetivo deste trabalho é o desenvolvimento de uma ferramenta para elaboração de operações com Álgebra Relacional em um banco de dados e visualização dos seus resultados. O aplicativo também permite a inclusão de listas de exercícios, auxiliando, portanto, o docente na realização de atividades.

Palavras-chave: Álgebra relacional. Banco de dados. *Structured Query Language*.

LISTA DE FIGURAS

Figura 1 – Execução da ferramenta SimAlg.....	16
Figura 2 – Exemplo de projeção na ferramenta SimAlg.....	17
Figura 3 – Ferramenta DBTools	18
Figura 4 – Execução da ferramenta EnsinAR	19
Figura 5 – Ferramenta de Álgebra Relacional.....	20
Figura 6 – Listas de exercícios na ferramenta de Álgebra Relacional.....	21
Figura 7 – Correção de erros na ferramenta de Álgebra Relacional	21
Figura 8 – Execução de consultas.....	22
Figura 9 – Execução mal sucedida: apresentação do erro.....	23
Figura 10 – Diagrama de casos de uso	27
Figura 11 – Diagrama de classes de análise do sistema.....	32
Figura 12 – Diagrama de entidades e relacionamentos do banco de dados.....	35
Figura 13 – Tela principal	39
Figura 14 – Paleta de manipulação de arquivos texto.....	39
Figura 15 – Tabelas e colunas	40
Figura 16 - Área de construção das consultas.....	40
Figura 17 – Paleta de operadores de Álgebra Relacional.....	41
Figura 18 – Visualização de resultados	42
Figura 19 – Configuração de bases de dados.....	42
Figura 20 – Hint nos botões	43
Figura 21 – Lista de bases de dados, tabela e campos	44
Figura 22 – Montagem da consulta.....	44
Figura 23 – Resultado da execução de uma consulta	45
Figura 24 – Mensagem de erro de sintaxe.....	46
Figura 25 – Mensagem de erro na montagem da seleção	46
Figura 26 – Erro de estrutura da consulta	47
Figura 27 – Cadastro de avaliações.....	48
Figura 28 – Cadastro de turmas.....	49
Figura 29 – Cadastro de exercícios	50
Figura 30 – Cadastro de respostas para exercícios.....	51
Figura 31 – Realização de avaliação	52

LISTA DE QUADROS

Quadro 1 – Operações da Álgebra Relacional.....	12
Quadro 2 – Operadores de Álgebra Relacional	14
Quadro 3 – Comparação das ferramentas estudadas e a proposta.....	23
Quadro 4 – Tecnologias e ferramentas utilizadas.....	24
Quadro 5 – Requisitos não funcionais	26
Quadro 6 – Requisitos não funcionais	26
Quadro 7 – Caso de uso compor avaliação	28
Quadro 8 – Caso de uso compor exercícios	29
Quadro 9 – Caso de uso compor respostas.....	29
Quadro 10 – Caso de uso compor usuários.....	30
Quadro 11 – Caso de uso realizar avaliação	31
Quadro 12 – Caso de uso emitir relatórios.....	31
Quadro 13 – Caso de uso realizar consultas	32
Quadro 14 – Descrição da classe avaliacao	33
Quadro 15 – Descrição da classe Exercicios.....	33
Quadro 16 – Descrição da classe Respostas	33
Quadro 17 – Descrição da classe Usuarios	34
Quadro 18 – Descrição da classe Aluno	34
Quadro 19 – Descrição da classe Professor.....	34
Quadro 20 – Descrição da classe Turmas	35
Quadro 21 – Campos da tabela avaliacao	36
Quadro 22 – Campos da tabela exercicios	36
Quadro 23 – Campos da tabela respostas.....	36
Quadro 24 – Campos da tabela turmas.....	37
Quadro 25 – Campos da tabela usuarios	37
Quadro 26 – Campos da tabela alunos	37
Quadro 27 – Campos da tabela professores	38

LISTA DE ABREVIATURAS E SIGLAS

DCL	<i>Data Control Language</i>
DDL	<i>Data Definition Language</i>
DER	Diagrama de Entidades e Relacionamentos
DML	<i>Data Manipulation Language</i>
DQL	<i>Data Query Language</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>

SUMÁRIO

1 INTRODUÇÃO	8
1.1 CONSIDERAÇÕES INICIAIS.....	8
1.2 OBJETIVOS.....	9
1.2.1 Objetivo Geral.....	9
1.2.2 Objetivos Específicos.....	9
1.3 JUSTIFICATIVA.....	9
1.4 ESTRUTURA DO TRABALHO.....	10
2 REFERENCIAL TEÓRICO	11
2.1 ÁLGEBRA RELACIONAL.....	11
2.2 STRUCTURED QUERY LANGUAGE.....	15
2.3 ESTADO DA ARTE.....	16
2.3.1 - SimAgl.....	16
2.3.2 DbTools 2000.....	17
2.3.3 EnsinAR.....	18
2.1.3 Sistema para Apoio a aprendizagem da Álgebra Relacional.....	19
2.1.4 Aplicativo para auxílio na Aprendizagem de conceitos de Álgebra Relacional.....	22
3 MATERIAIS E MÉTODO	24
3.1 MATERIAIS.....	24
3.2 MÉTODO.....	24
4 RESULTADOS	25
4.1 ESCOPO DO SISTEMA.....	25
4.2 MODELAGEM DO SISTEMA.....	26
4.3 APRESENTAÇÃO DO SISTEMA.....	38
4.3.1 Console.....	38
4.3.2 Composição de avaliações.....	47
4.4 IMPLEMENTAÇÃO DO SISTEMA.....	52
5 CONCLUSÃO.....	57
REFERÊNCIAS	58

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais do trabalho, os seus objetivos e a justificativa. O capítulo é finalizado com a apresentação da organização do texto por meio da descrição sumária dos capítulos subsequentes.

1.1 CONSIDERAÇÕES INICIAIS

A Álgebra Relacional é uma linguagem de consulta formal que define operadores para atuar nas tabelas de bancos de dados e é procedimental (REZENDE, 2015). Por ser procedimental são fornecidas as instruções ao sistema para que o mesmo realize uma sequência de operações na base de dados para obter o resultado dessas instruções.

A maneira básica de atuação da Álgebra Relacional é a partir de uma ou mais tabelas como entrada de dados produzir uma nova tabela que é o resultado das operações ou instruções indicadas. Assim, como referenciado por Macoratti (2015), a Álgebra Relacional é uma forma de cálculo sobre conjuntos ou relações. E de acordo com esse autor na terminologia formal de modelo relacional são utilizados os seguintes conceitos (DATE, 1985):

- a) Uma linha é chamada de tupla;
- b) O cabeçalho da coluna é chamado de atributo;
- c) Tabela é chamada de relação;
- d) O tipo de dados que descreve os tipos de valores que podem aparecer em cada coluna é chamado de domínio.

As operações básicas da Álgebra Relacional com o uso de uma ou duas relações como entrada gerando uma relação de saída são (DATE, 1985): seleção, projeção, união, diferença e produto cartesiano.

O aprendizado de Álgebra Relacional pode ser bastante difícil para alunos, pelo alto nível de abstração envolvido. Essa abstração decorre da dificuldade de entendimento dos resultados da execução das suas instruções em um banco de dados. Assim, verificou-se a oportunidade de desenvolver um aplicativo para o ensino desses conceitos de forma prática. Esse aplicativo interpretará operadores lógicos da Álgebra Relacional e executará em bases de dados as consultas em

tabelas trazendo dados semelhantes ao uso de comandos *Structured Query Language* (SQL).

1.2 OBJETIVOS

A seguir são apresentados o objetivo geral e os objetivos específicos do trabalho.

1.2.1 Objetivo Geral

Desenvolver uma ferramenta para auxiliar na aprendizagem de conceitos de Álgebra Relacional.

1.2.2 Objetivos Específicos

- Facilitar o aprendizado de conceitos de Álgebra Relacional por meio da execução das instruções elaboradas em um banco de dados.
- Facilitar a experimentação prática de conceitos de Álgebra Relacional.

1.3 JUSTIFICATIVA

O aplicativo auxilia na representação das instruções em Álgebra Relacional provendo uma forma prática de entendimento dos conceitos. Visando, assim, transformar as instruções aprendidas de forma teórica em uma representação mais prática pela visualização dos resultados das instruções definidas, colaborando no entendimento desses conceitos.

O aplicativo funcionará interpretando os operadores lógicos inseridos no editor da ferramenta, transformando-os em instruções SQL para executar em bases de dados. O aplicativo funcionará também para a inserção de listas de tarefas. Nessas listas serão cadastradas perguntas para resolução. As respostas também

podem estar vinculadas às perguntas, facilitando a verificação dos resultados por parte do próprio aluno. A ferramenta conterà, ainda, um marcador de pontuação conforme o acerto de questões e mostrará as repostas conforme o modo que a mesma for configurada para uso como auto-avaliação ou aprendizado.

A ferramenta justifica-se pela contribuição no ensino tanto para alunos como para professores em cursos nos quais há disciplinas e conteúdos relacionados a banco de dados.

1.4 ESTRUTURA DO TRABALHO

O Capítulo 2 apresenta o referencial teórico que abrange Álgebra Relacional e linguagem estruturada de consulta. O Capítulo 3 apresenta as ferramentas e tecnologias utilizadas na modelagem e desenvolvimento do aplicativo e o método. O método é composto pelas principais etapas e respectivas atividades realizadas. O resultado está apresentado no Capítulo 4. O trabalho é finalizado com as considerações finais seguidas das referências utilizadas no texto.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico do trabalho. E por tratar-se de uma ferramenta para ensino de conceitos de Álgebra Relacional, o referencial teórico é sobre Álgebra Relacional e linguagem estruturada de consulta.

2.1 ÁLGEBRA RELACIONAL

Álgebra Relacional é “um conjunto de operadores que tornam relações como seus operandos e retornam uma relação como seu resultado” (DATE, 2000, p. 132).

Dois conceitos são essenciais às operações em Álgebra Relacional: relações e tuplas. De acordo com Elmasri e Navathe (2011, p. 39), na terminologia formal do modelo relacional, uma linha da tabela de valores é denominada tupla, um cabeçalho da coluna é chamado atributo e a tabela é chamada relação. Para esses autores, uma relação é definida como um conjunto de tuplas. Informalmente, cada relação é semelhante a uma tabela de valores ou, até, certo ponto, a um arquivo plano de registros. É denominado de arquivo plano porque para cada registro possui uma estrutura linear simples ou plana (ELMASRI; NAVATHE, 2011, p. 39).

A Álgebra Relacional define operações para realizar consultas e alterações em banco de dados (KINAIPP; ALVES, 2010). Date (2008) destaca que Codd (1972) definiu 8 operadores, normalmente considerados como sendo da álgebra original. Contudo, Date (2000) ressalta que novos operadores podem ser definidos dependendo das necessidades das relações de entrada e saída. Essas operações podem ser divididas em dois grupos (ELMASRI; NAVATHE, 2005, DATE 2000):

a) Operações da teoria de conjuntos da matemática – são os operadores de conjuntos relacionais tradicionais definidos pela teoria de conjuntos da matemática. Essas operações se aplicam aos elementos de conjuntos visando gerar um terceiro conjunto a partir de operações realizadas sobre os elementos desses conjuntos. Essas operações são: união (*union*), interseção (*intersection*), diferença de conjuntos (*set difference*) e produto cartesiano (*cross product*);

b) Operações desenvolvidas especificamente para bancos de dados - são operadores relacionais especiais para seleção ou restrição (*select*), projeção (*project*), junção (*join*) e divisão (*division*).

Paes (2004) destaca que as operações fundamentais na Álgebra Relacional são: seleção, projeção, união, diferença e produto cartesiano. Além dessas existem as operações definidas a partir das operações fundamentais e são: renomeação, interseção, junção natural, divisão e atribuição. Em Álgebra Relacional, tanto os operandos como o resultado das operações são relações. E, assim, no resultado das operações pode ser aplicada qualquer operação da Álgebra Relacional.

As operações no Modelo Relacional abordam 4 categorias (TAKAI, 2015): estrutura, dados, conjunto e tabela, como representado no Quadro 1. Essas operações devem obedecer a restrições de integridade. Porém, quando restrições forem violadas em algumas das operações, o Sistema Gerenciador de Banco de Dados (SGBD) pode tomar uma ação adaptativa (com essa ação a operação é executada, mas com restrições) ou uma ação de informe da impossibilidade de execução (e a ação não é executada) (TAKAI, 2015).

Agrupamentos de operações	Operações de cada agrupamento			
Estrutura	Inserção	Remoção	Atualização	
Dados	Inserção	Remoção	Atualização	
Conjunto	União	Interseção	Diferença	Produto cartesiano
Tabela	Seleção	Projeção	Junção	

Quadro 1 – Operações da Álgebra Relacional

Fonte: Takai (2015, p. 5).

A descrição dos operadores é apresentada a seguir, tendo como base os conceitos constantes em Date (2000), Takai (2015) e Paes (2004).

a) Operações relacionadas à estrutura - as operações sobre a estrutura auxiliam na definição e manutenção do esquema de dados relacional.

a.1) Inserção - adição de novas tabelas ao esquema de dados.

a.2) Remoção - retirada de tabelas e atributos do esquema de dados.

a.3) Atualização - adição de atributos às tabelas existentes no esquema de dados.

b) Operações relacionadas aos dados - operações sobre as tuplas de uma relação já existente no banco de dados.

b.1) Inserção: adição de uma ou mais tuplas em uma relação.

b.2) Remoção: retirada de tuplas de uma relação.

b.3) Atualização: alteração de algum valor de atributo de uma tupla.

c) Operações de conjuntos - operações que se aplicam a duas relações que obedecem à compatibilidade de união. Na união, ambas as relações devem apresentar atributos pertencentes simultaneamente aos mesmos domínios. As relações (A e B) usadas nas operações podem estar ambas fisicamente armazenadas no mesmo banco de dados ou bancos de dados distintos; uma estar armazenada e a outra originada (resultante) de uma operação realizada; ou ambas resultarem de operações realizadas.

c.1) União - o resultado da união de duas relações é um conjunto das tuplas que pertencem a ambas ou a cada uma das relações especificadas, mas sem redundância. A não redundância é estabelecida pela própria definição de conjunto.

A operação de união cria uma relação a partir de outras duas relações, sendo o conjunto resultante composto pelas tuplas comuns e não-comuns a ambos os conjuntos envolvidos na relação. O resultado da união é composto por linhas únicas de cada uma das relações, mas as tuplas que possam estar duplicadas constarão apenas uma vez no conjunto resultante.

c.2) Interseção - o resultado da interseção de duas relações é um conjunto composto por todas as tuplas que pertençam simultaneamente às duas relações. Como resultado da interseção o conjunto resultante é composto por todas as tuplas que pertencem simultaneamente a ambas as relações da operação.

c.3) Diferença - a diferença entre duas relações é um conjunto formado pelas tuplas que pertencem a primeira relação, mas não à segunda. A diferença entre conjuntos permite encontrar as tuplas que estão no primeiro operando, mas não no segundo.

c.4) Produto cartesiano - aplica-se a duas relações que não necessitam ser compatíveis para a operação de união. O resultado do operador produto cartesiano é uma relação contendo todas as tuplas resultantes da combinação de duas relações especificadas. No resultado, "para cada tupla da primeira relação haverá a combinação com todas as tuplas da segunda relação ou vice-versa" (REZENDE, 2012, p. 1).

d) Operações de tabela - operações que se aplicam para quaisquer relações.

d.1) SELEÇÃO - operação aplicada sobre uma relação que visa selecionar um sub-conjunto de tuplas que satisfaçam a uma determinada condição (simples ou composta). O sub-conjunto retornado contém todas as tuplas de uma relação definida e que satisfazem a condição especificada por essa relação. Esta condição

aplica-se apenas em uma única relação sendo verificada individualmente para cada tupla da relação.

d.2) **Projeção** - operação aplicada sobre uma relação para selecionar os atributos de uma relação de acordo com um conjunto de atributos especificados. No resultado as tuplas iguais não são repetidas. Como resultado da projeção é retornada uma relação que contém todas as tuplas que permanecem após removidos os atributos especificados e de forma a satisfazer uma relação dada.

d.3) **Junção** - a junção resulta em uma relação contendo todas as tuplas possíveis da combinação de duas tuplas. Esta combinação é realizada de acordo com uma condição indicada, ou seja, é utilizada para combinar tuplas de duas relações que atendam o predicado de junção especificado partindo dos atributos comuns a ambas.

d.4) **Operação Divisão**: “dadas duas relações unárias e uma relação binária, retorna uma relação contendo todas as tuplas de uma única relação unária presentes na relação binária, coincidindo com todas as tuplas da outra relação unária” (DATE, 2000, p. 133).

A operação de divisão é utilizada nas consultas que exigem que todos os valores dos atributos do divisor estejam presentes no dividendo. Produz como resultado a projeção de todos os atributos da primeira relação exceto aqueles presentes no divisor.

O Quadro 2 apresenta os operadores utilizados em Álgebra Relacional

Operação	Notação	Sintaxe
Seleção	σ	<condição de seleção> (Relação)
Projeção	π	<lista de atributos> (Relação)
Junção natural	\bowtie	(Relação A \times Relação B)
Divisão	\div	(Relação A \div Relação B)
União	\cup	(Relação A \cup Relação B)
Interseção	\cap	(Relação A \cap Relação B)
Diferença	-	(Relação A - Relação B)
Produto Cartesiano	\times	(Relação A \times Relação B)

Quadro 2 – Operadores de Álgebra Relacional
Fonte: Rezende (2013, p. 1).

2.2 STRUCTURED QUERY LANGUAGE

Structured Query Language, a SQL, “é um conjunto de comandos de manipulação de banco de dados utilizado para criar e manter a estrutura desse bando de dados, além de incluir, excluir, modificar e pesquisar informações nas suas tabelas” (OLIVEIRA, 2002, p. 18).

A linguagem SQL implementa conceitos definidos no modelo relacional e com essa linguagem é possível (DAMAS, 2007):

- a) criar, alterar e remover elementos de um banco de dados como tabelas, *views* e índices;
- b) inserir, alterar e excluir dados;
- c) consultar (obter dados de acordo com critérios de pesquisa);
- d) consultar o acesso de usuários ao banco de dados e as operações realizadas por esses usuários;
- e) obter garantia da consistência e integridade do banco de dados.

A SQL pode ser dividida em algumas linguagens, que são (PAES, 2004, OLIVEIRA, 2002):

a) *Data Definition Language* (DDL) - Linguagem de Definição de Dados que permite a criação dos componentes do banco de dados, como, por exemplo, tabelas e índices. A DDL também fornece comandos para alteração de esquemas de relação e exclusão de relações. Comandos da DDL: CREATE, ALTER e DROP para table e index;

b) *Data Manipulation Language* (DML) - Linguagem de manipulação de dados armazenados no banco de dados. É baseada na álgebra e no cálculo relacional e é composta por comandos para inserir, remover e modificar tuplas em banco de dados. Comandos básicos da DML: INSERT, UPDATE e DELETE;

c) *Data Query Language* (DQL) – Linguagem de consulta de dados que permite extrair dados do banco de dados. Comando: SELECT.

d) *Data Control Language* (DCL) - Linguagem de controle de dados que provê a segurança interna do banco de dados. Por meio dessa linguagem usuários podem ser cadastrados no bando de dados e privilégios atribuídos. Comandos são: CREATE USER, ALTER USER, CREATE SCHEMA, GRANT e REVOKE.

2.3 ESTADO DA ARTE

A seguir é apresentada a análise de ferramentas similares de consulta em Álgebra Relacional. Foram analisadas as seguintes ferramentas simuladoras de consultas em Álgebra Relacional disponíveis: SimAgl, DbTools 2000, EnsinAR e sistema para apoio a aprendizagem da Álgebra Relacional.

2.3.1 - SimAgl

A ferramenta SimAgl foi desenvolvida em 2010 pela acadêmica Larissa Lauter como trabalho final de graduação em Ciência da Computação da Universidade Federal de Santa Maria (LAUTER, 2010). Essa ferramenta contém uma interface interativa e de fácil utilização, porém todos os dados manipulados são estáticos em tela.

A execução de uma projeção em Álgebra Relacional e o resultado em forma de grade logo abaixo da execução pode ser visualizada na Figura 1.

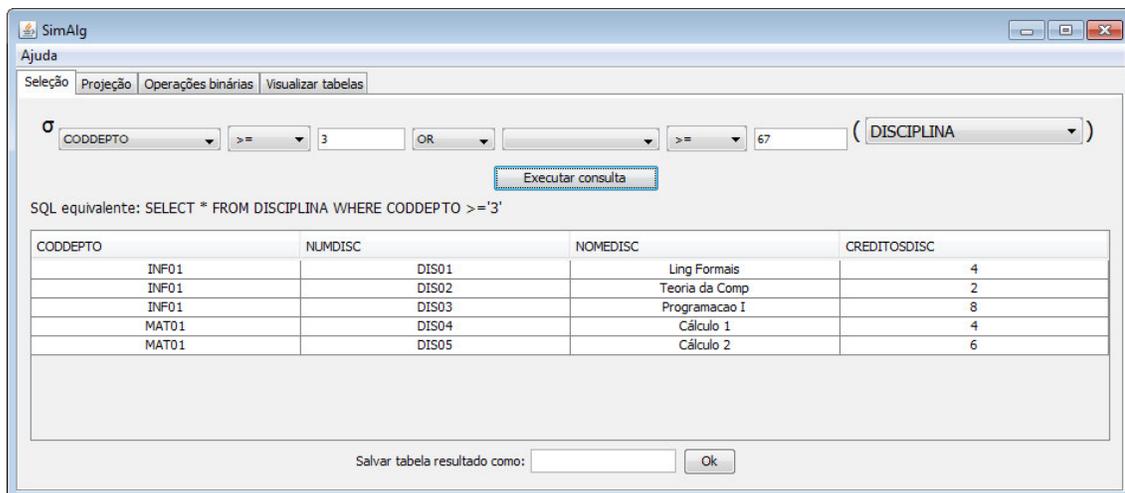


Figura 1 – Execução da ferramenta SimAgl

Na Figura 2 está o exemplo de uma projeção em Álgebra Relacional.

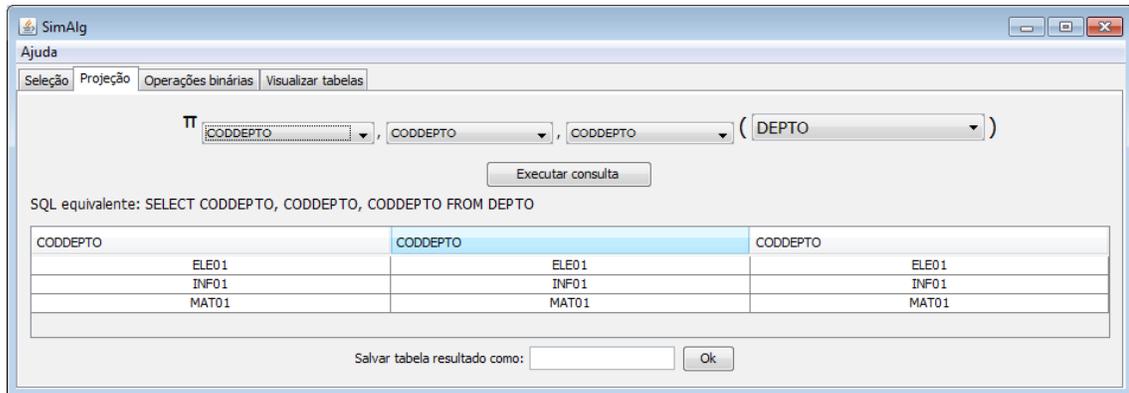


Figura 2 – Exemplo de projeção na ferramenta SimAlg

2.3.2 DbTools 2000

O simulador de consultas DbTolls 2000 foi desenvolvido com intuito de ensinar em sala de aula alguns conceitos de banco de dados. Essa ferramenta, disponível em <http://www.cc.gatech.edu/computing/Database/dbTools/> foi criada utilizando VisualCafé 4 Expert Edition e Java SE 1.2.2. Ela contém uma interface intuitiva e de fácil utilização, apresenta uma hierarquia de tabelas e campos no banco de dados para ajudar no auxílio da elaboração das consultas e contém um ambiente livre para executar as consultas.

A Figura 3 apresenta a ferramenta em execução, podendo ser visualizada a consulta sendo executada e o resultado sendo mostrado logo abaixo na forma de colunas e linhas.

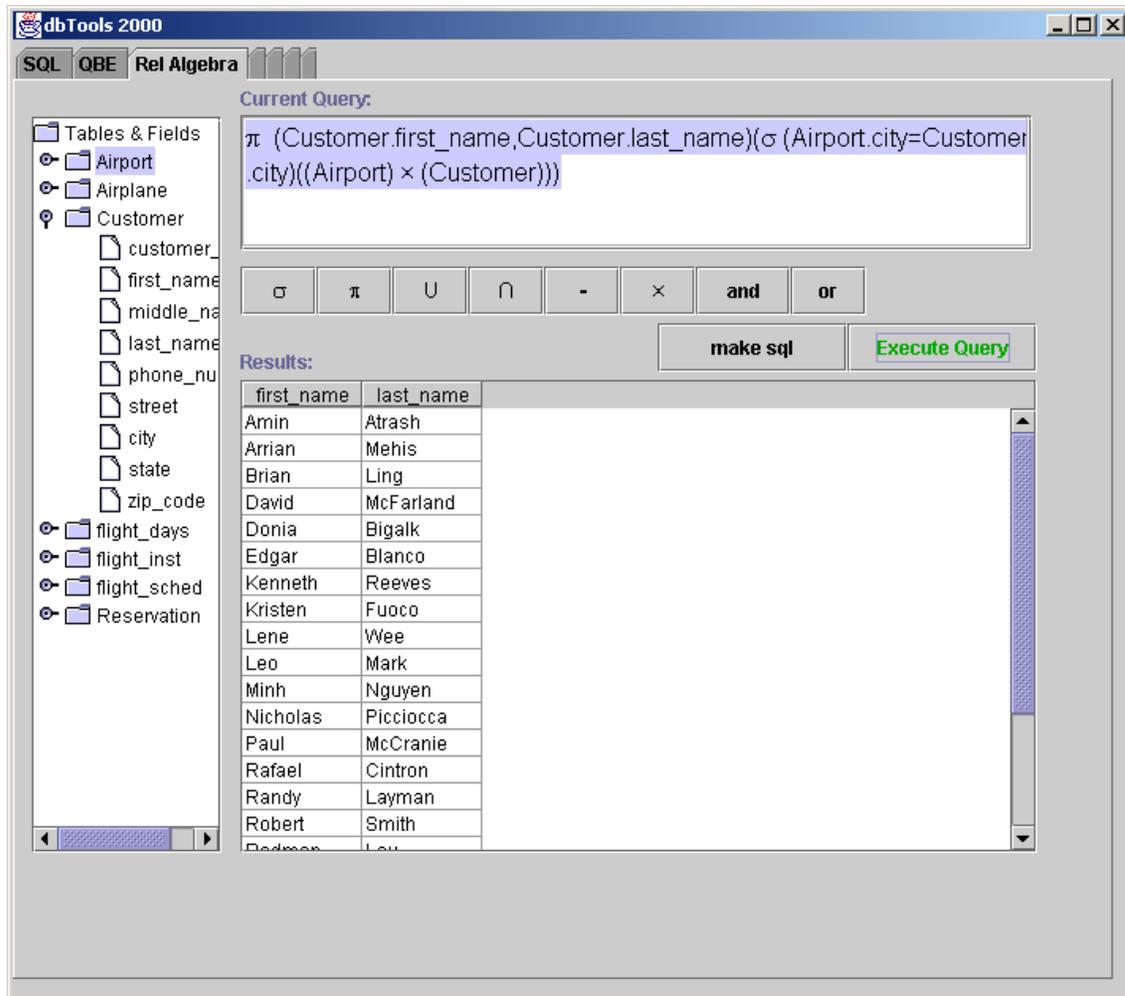


Figura 3 – Ferramenta DBTools

2.3.3 EnsinAR

Simulador EnsinAR, foi desenvolvido por Ederson Luiz Paes (PAES, 2004). Essa ferramenta didática possui uma interface intuitiva e de bastante ajuda com os operadores e com a definição da função de cada um deles. Contém uma interface livre para escrita de consultas e a cada execução possibilita visualizar o comando a ser executado em SQL.

Na Figura 4 é apresentada a execução de uma instrução em Álgebra Relacional, podendo ser visualizado o resultado sendo mostrado logo abaixo em forma de colunas e linhas e uma caixa de mensagem mostrando o comando em SQL.

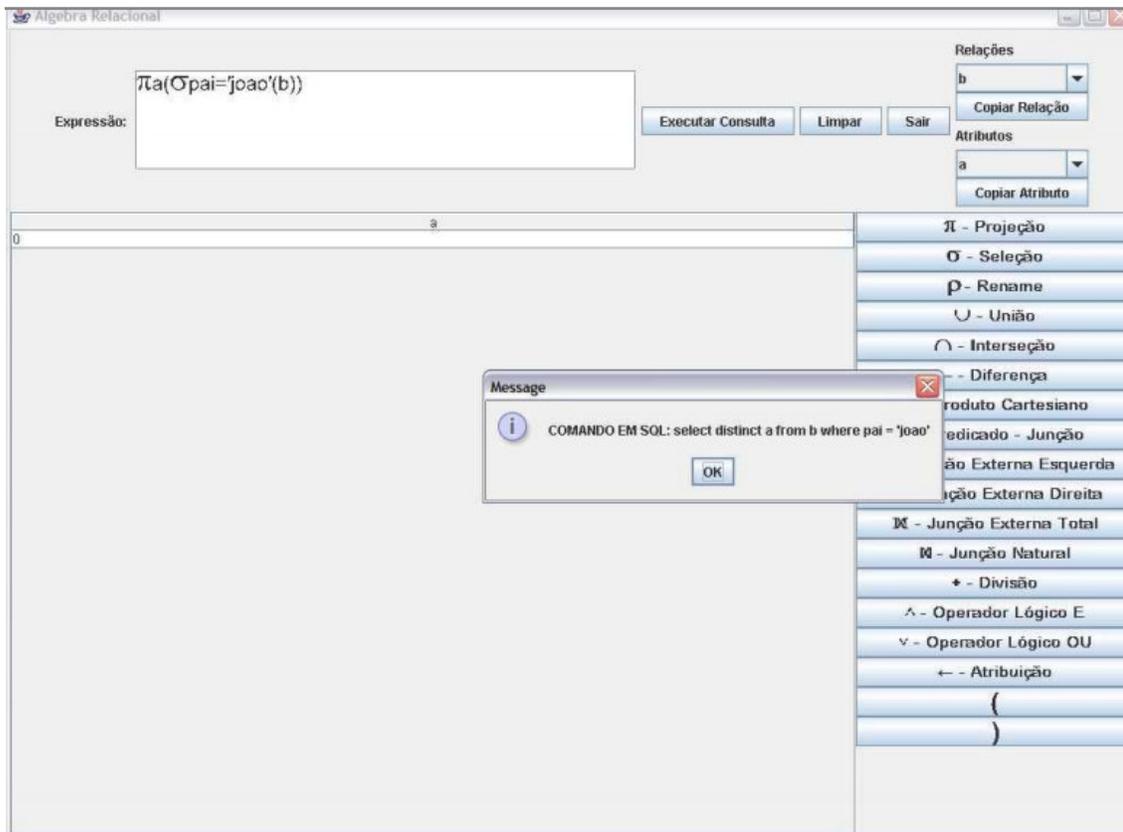


Figura 4 – Execução da ferramenta EnsinAR

2.1.3 Sistema para Apoio a aprendizagem da Álgebra Relacional

O sistema desenvolvido pelo acadêmico Renato Vidal Kinaipp (KINAIPP, 2009) funciona como uma ferramenta de auxílio na elaboração de exercícios em Álgebra Relacional. Por meio dessa ferramenta são cadastrados: a base de dados, usuários, nível de permissões por usuário e exercícios. A Figura 5 apresenta a interface principal desta ferramenta.

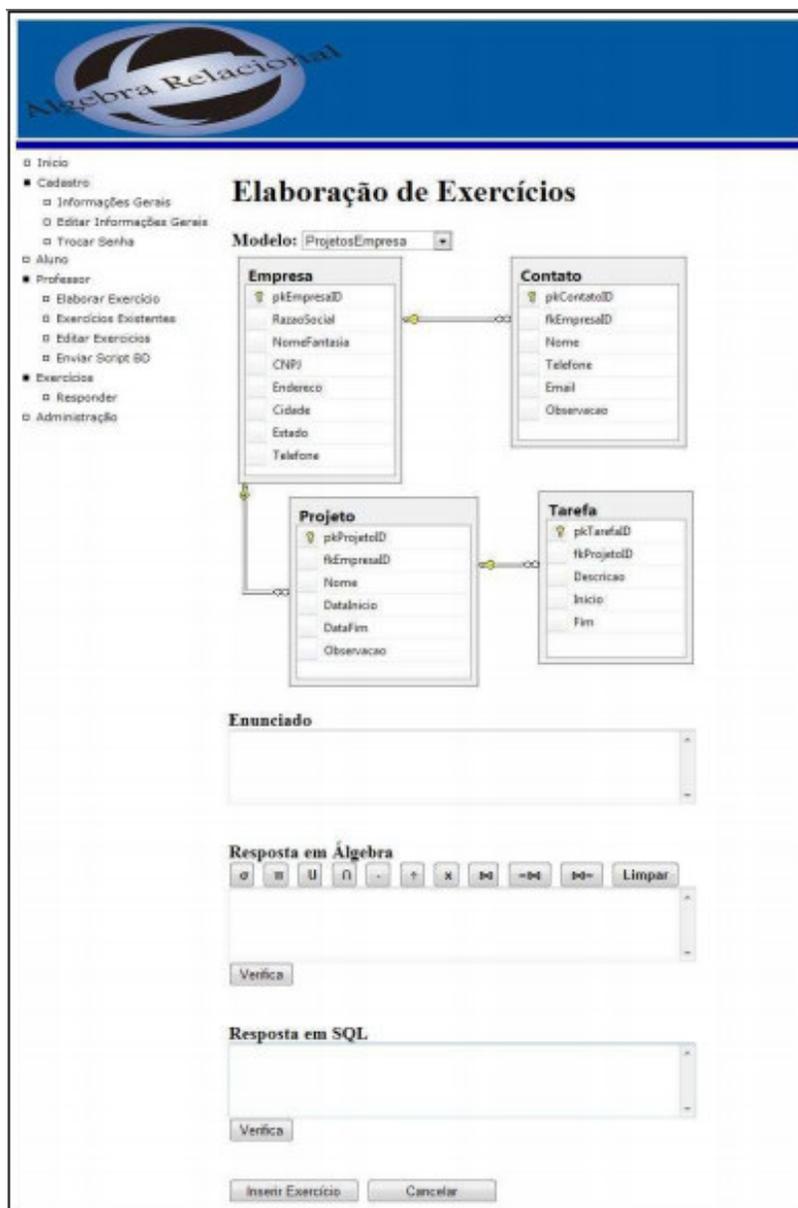


Figura 5 – Ferramenta de Álgebra Relacional

A ferramenta desenvolvida por Kinaipp (2009) contém cronograma de começo e término da resolução dos exercícios, como pode ser visualizado na Figura 6. O sistema é bastante usual como ferramenta didática para professores executarem trabalhos com alunos.

Pergunta
Selecione as descrições das tarefas.

π descricao, inicio, fim(tarefa)

Resposta Incorreta!

Resposta Atual:
π descricao, inicio, fim(tarefa)

Resultado da Consulta:

descricao	inicio	fim
Trabalho1	10/05/2009 00:00:00	10/06/2009 00:00:00
Trabalho2	20/04/2009 00:00:00	10/05/2009 00:00:00
Trabalho3	15/04/2009 00:00:00	10/06/2009 00:00:00
Trabalho4	10/05/2009 00:00:00	10/06/2009 00:00:00
Trabalho5	11/06/2009 00:00:00	10/07/2009 00:00:00
Trabalho6	14/04/2009 00:00:00	21/07/2009 00:00:00
Trabalho7	15/04/2009 00:00:00	21/07/2009 00:00:00

Figura 6 – Listas de exercícios na ferramenta de Álgebra Relacional

A forma de sugestão de correção de perguntas é bastante interessante, contendo uma representação declarativa do erro que ocorreu. A correção das perguntas é exemplificada na Figura 7.

Pergunta
Selecione as descrições das tarefas.

π descricao="tarefa1"(tarefa)

Erro! Causado pelo Token: '='

Resposta Atual:
π descricao="tarefa1"(tarefa)

Figura 7 – Correção de erros na ferramenta de Álgebra Relacional

2.1.4 Aplicativo para auxílio na Aprendizagem de conceitos de Álgebra Relacional

O aplicativo para auxílio na Aprendizagem de conceitos de Álgebra Relacional foi desenvolvido pelo acadêmico Luiz Henrique Galvão Schneiger no curso de análise e desenvolvimento de sistema na UTFPR no Câmpus de Pato Branco – PR, como, trabalho de conclusão de curso. Esse aplicativo contém uma interface interativa para livre construção de consultas, paletas de ajuda e auxílio na elaboração de consultas e sugestões para correção de erros ocasionados pela execução mal sucedida das consultas, porém não é disponível *online* por ser um sistema *desktop*.

O resultado da execução de instruções em Álgebra Relacional é apresentado em forma de grade logo abaixo da área de montagem das consultas, como pode ser visualizada na Figura 8.

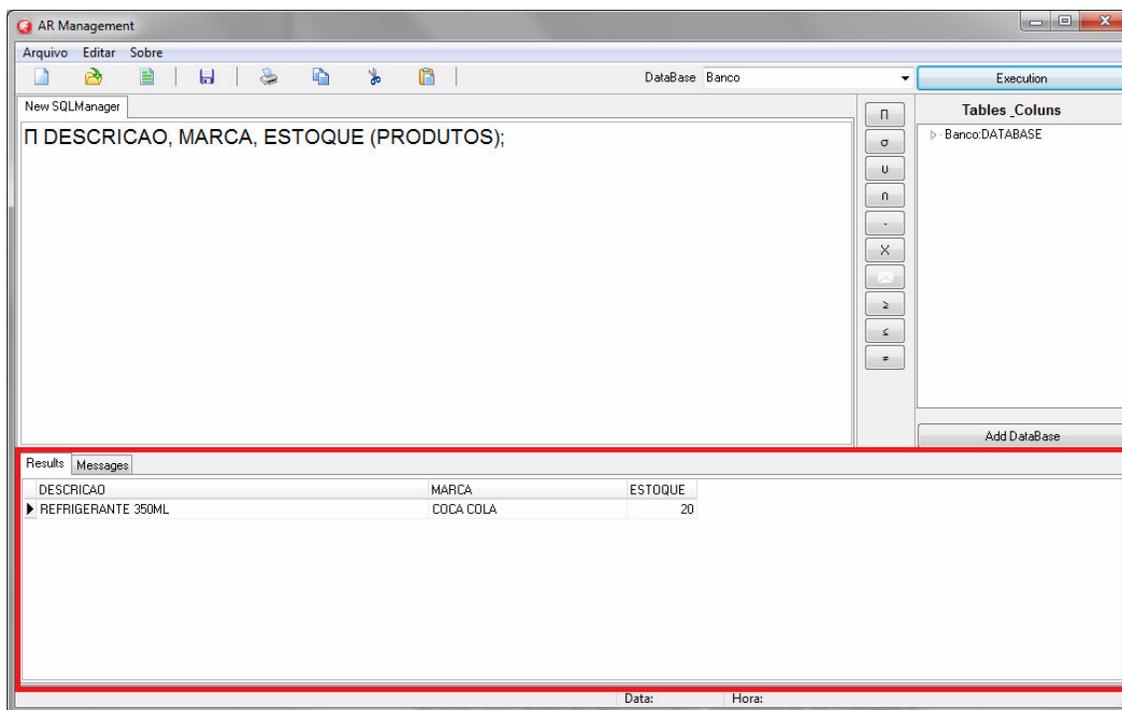


Figura 8 – Execução de consultas

E a execução mal sucedida pode ser visualizada em forma de texto com o explicativo do erro ocorrido, como mostra na Figura 9.

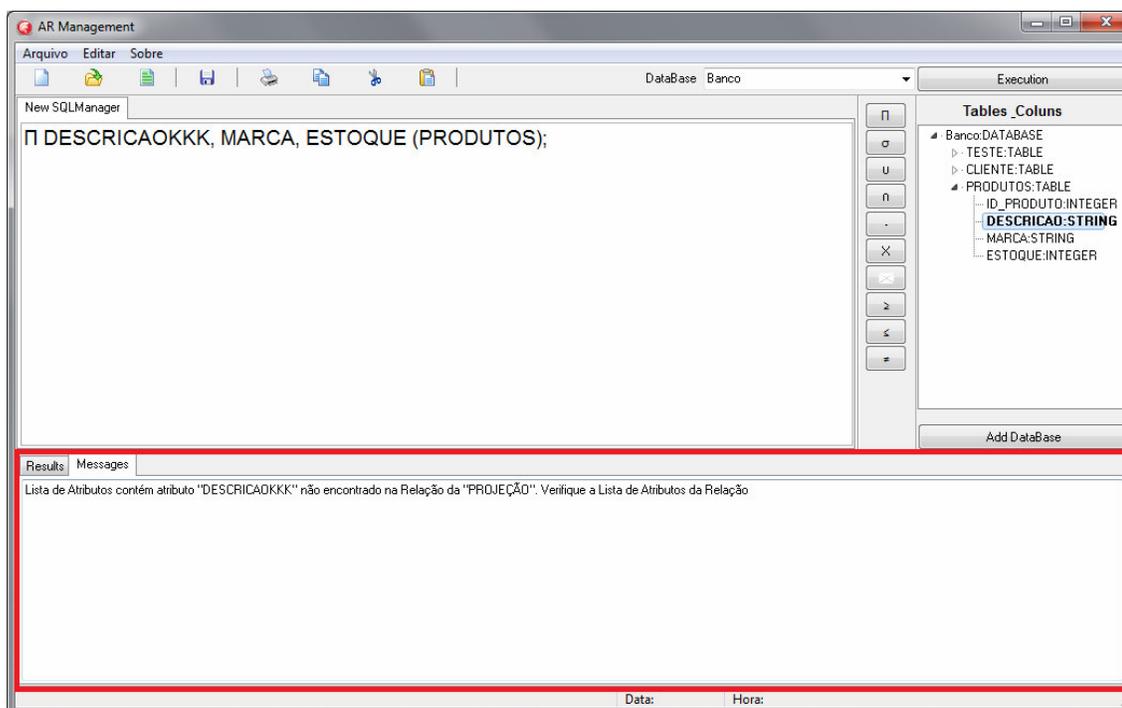


Figura 9 – Execução mal sucedida: apresentação do erro

O Quadro 3 apresenta uma comparação das principais características das ferramentas analisadas.

Ferramenta	SimAlg 2.3.1	DbTools 2.3.2	EnsinAR 2.3.3	Sistema de Aprendizagem em AR 2.3.4	Aplicativo proposto por Kinaipp	Ferramenta proposta
Interface intuitiva	X	X	X	X	X	X
Liberdade de construção de consultas		X	X	X	X	X
Sintaxe e simbologias tradicionais	X	X	X	X	X	X
Correção de instruções ou explicativo do erro.		X		X	X	X
Seleção de banco de dados.		X	X	X	X	X
Não necessita de componentes adicionais para execução.					X	X
Auto completar nas instruções	X					

Quadro 3 – Comparação das ferramentas estudadas e a proposta

3 MATERIAIS E MÉTODO

A ênfase deste capítulo está em reportar o que e como será feito para alcançar o objetivo do trabalho. Este capítulo pode ser subdividido, inicialmente, em duas seções, sendo uma para os materiais e outra para o método.

3.1 MATERIAIS

Para a modelagem e a implementação do sistema serão utilizadas as ferramentas e as tecnologias apresentadas no Quadro 4.

Ferramenta / Tecnologia	Versão	Finalidade
Astah Community	6.9.0	Para a modelagem do aplicativo.
Delphi	XE5	Linguagem de programação e ambiente de desenvolvimento
SQLite		Banco de dados

Quadro 4 – Tecnologias e ferramentas utilizadas

3.2 MÉTODO

Inicialmente foram estudadas ferramentas disponíveis no mesmo contexto e que apresentam características semelhantes à que será desenvolvida como resultado deste trabalho. Do estudo dessas ferramentas foi elaborado um quadro com as características fundamentais dessas ferramentas e a identificação da ferramenta que a possui.

Esse estudo auxiliou a definir os requisitos pretendidos para a ferramenta a ser desenvolvida. Esses requisitos foram modelados e posteriormente implementados.

4 RESULTADOS

Neste capítulo são apresentadas as funcionalidades para a implementação de uma ferramenta para aprendizagem de conceitos de Álgebra Relacional, contendo a modelagem para cadastro de turmas, avaliações, exercícios, respostas e usuários (alunos e professores). Como foco deste capítulo está o projeto, priorizado nesta versão do sistema, com a definição dos requisitos, a modelagem dos mesmos com o auxílio de diagramas.

4.1 ESCOPO DO SISTEMA

O sistema auxiliará no aprendizado de conceitos relacionados à Álgebra Relacional. O sistema auxiliará na representação e na compreensão das instruções em Álgebra Relacional, de forma prática e intuitiva, visando colaborar no entendimento dos alunos. A solução computacional proposta considera o contexto apresentado a seguir.

O sistema funcionará com uma tela principal contendo os principais operadores da Álgebra Relacional para montagem de consulta. O console ou área de texto, assim chamada, conterà um espaço livre para criação das instruções. Essa área de consulta será disponibilizada para usuários com cadastro no sistema.

O sistema disponibilizará uma forma de uso para trabalhos ou avaliações. Cada usuário professor terá a possibilidade de cadastrar avaliações, exercícios e repostas para cada turma. Na elaboração das avaliações, o usuário professor pode optar por um ambiente que permite a aplicação de todos os procedimentos de uma avaliação ou por um ambiente para aplicação de trabalhos acadêmicos. Para cada modelo utilizado como avaliativo ou somente voltado para trabalhos acadêmicos são disponibilizados relatórios com as respostas das questões realizadas pelos alunos para serem analisadas pelo professor.

4.2 MODELAGEM DO SISTEMA

O Quadro 5 apresenta a listagem dos requisitos funcionais identificados para o sistema.

Identificação	Nome	Descrição
RF01	Cadastrar avaliações	Cadastrar as avaliações que serão realizadas pelas turmas. Exemplo: 5º A - TADS, 6º C - SI
RF02	Cadastrar exercícios	Cadastrar os exercícios para compor as avaliações.
RF03	Cadastrar respostas	Cadastrar as respostas para os exercícios e avaliações.
RF04	Cadastrar usuários	Cadastrar os usuários com níveis de permissões diferentes para executar as funcionalidades. Os níveis de usuários são: professor e aluno.
RF05	Realizar avaliação	A composição de avaliações é realizada por um usuário que contém a permissão de professor. O usuário compõe a avaliação vinculando ou criando novos exercícios e respostas. As avaliações são realizadas pela própria ferramenta contendo estimativa de tempo e submissão da avaliação realizada.
RF06	Emitir relatórios	Emitir relatórios de avaliações de exercícios e respostas.
RF07	Realizar consultas	A realização de consultas é feita no console do sistema. As consultas podem ser elaboradas na forma de instruções SQL ou Álgebra Relacional.

Quadro 5 – Requisitos não funcionais

A listagem do Quadro 6 apresenta os requisitos não-funcionais identificados para o sistema. Os requisitos não funcionais explicitam regras de negócio, restrições ao sistema de acesso, requisitos de qualidade, desempenho e segurança, dentre outros.

Identificação	Nome	Descrição
RNF01	Acesso ao sistema	O acesso ao sistema será realizado por meio de <i>login</i> e senha.
RNF02	Acesso avaliação	A avaliação será liberada somente na data e horário definidos para a realização da avaliação pelo professor responsável.

Quadro 6 – Requisitos não funcionais

O diagrama de casos de uso apresentado na Figura 8 contém as funcionalidades do sistema realizadas pelos seus atores, os usuários do sistema, exemplificados na descrição de cada um que são: professor e alunos. O professor é

responsável em compor a avaliação do aluno contendo as questões a serem respondidas e suas respectivas respostas, a forma de avaliação que será cobrada, data e horário da avaliação e é responsável por visualizar os relatórios de avaliações realizadas pelos alunos. Os alunos são responsáveis por realizar as avaliações propostas pelos professores e consultar os relatórios das suas avaliações. Ambos têm acesso às funcionalidades de realizar consultas no console do sistema. As funcionalidades dos casos de uso foram agrupadas em console e realização de exercícios, caracterizados por avaliações, conforme apresenta a Figura 10.

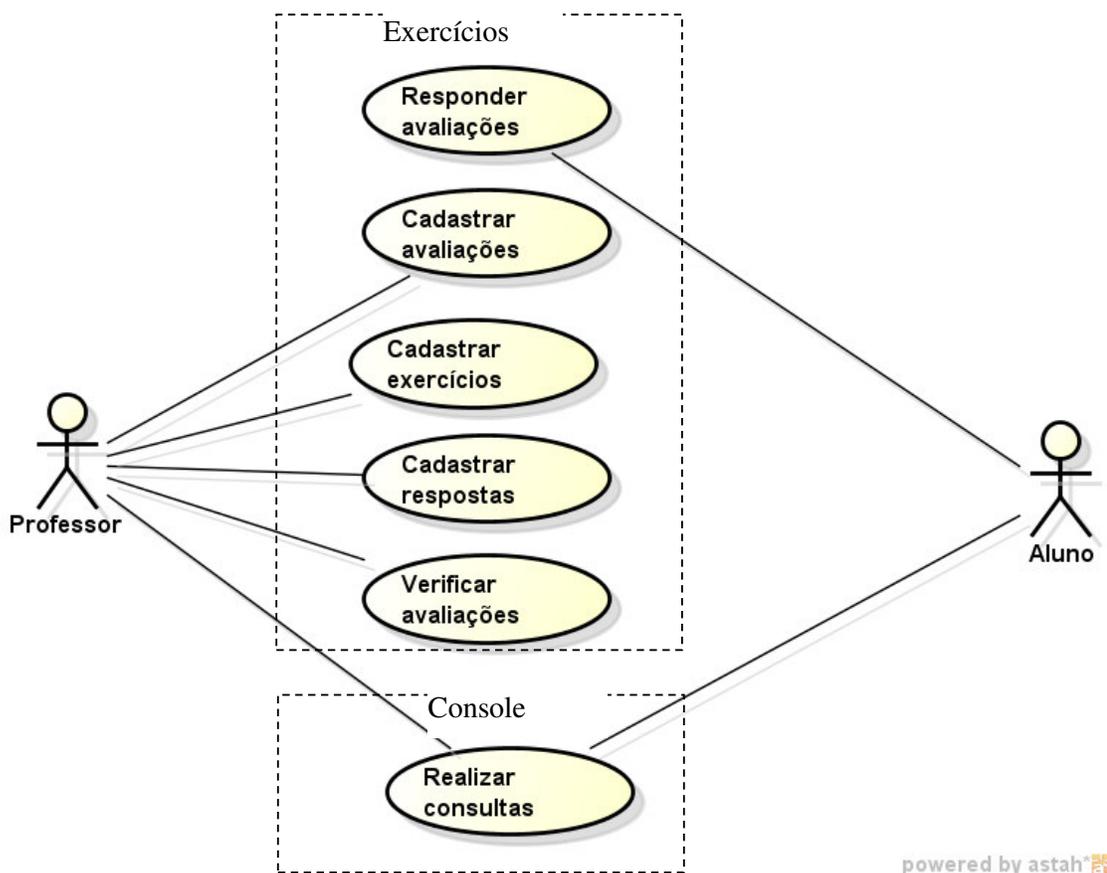


Figura 10 – Diagrama de casos de uso

O caso de uso compor avaliação é apresentado no Quadro 7. Nas descrições dos casos de uso apresentadas nos quadros a seguir, os fluxos alternativos estão vinculados à sequência de eventos. A identificação da linha se refere à qual evento (linha) da sequência de eventos o fluxo alternativo está vinculado.

<p>Caso de uso: Cadastrar Avaliação.</p> <p>Descrição: O usuário (professor) é responsável pela composição da avaliação. Uma avaliação é composta por exercícios e respostas. Esses exercícios são agrupados em categorias e a cada item pode estar associado a um peso específico.</p> <p>Evento Iniciador: Ator realiza o cadastro de uma avaliação.</p> <p>Atores: Usuário (Professor) responsável pelo tipo de atividade.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona formulário para cadastro de avaliações. 2. Sistema apresenta formulário. 3. Ator seleciona a turma que é vinculada a avaliação. 4. Ator preenche os campos do formulário para descrever enunciado e demais campos para compor a avaliação. 5. Sistema insere a avaliação no banco de dados. <p>Pós-Condição: Avaliação cadastrada no sistema.</p>
--

Nome do fluxo alternativo (extensão)	Descrição
Linha 5 da sequencia de eventos: Falta de informações	5.1 No momento de salvar, o sistema faz a verificação e constata que há dados faltantes. 5.2 É emitida mensagem de dados faltantes. 5.3 É retornado para o formulário de avaliação para correção do item faltante, sem executar o salvamento do mesmo.
Linha 5 da sequencia de eventos: Dados não são válidos.	5.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos. 5.2 É emitida mensagem de dados inválidos. 5.3 É retornado para o formulário de avaliação para correção do item inválido, sem executar o salvamento do mesmo.

Quadro 7 – Caso de uso compor avaliação

O caso de uso compor exercícios é apresentado no Quadro 8.

<p>Caso de uso: Cadastrar Exercícios</p> <p>Descrição: O usuário (professor) é responsável pela composição dos exercícios. Uma avaliação é composta por um ou mais exercícios. Cada exercício pode ter um peso específico associado.</p> <p>Evento Iniciador: Ator realiza o cadastro de exercícios.</p> <p>Atores: Usuário (professor) responsável pelo tipo de atividade.</p> <p>Pré-condição: Avaliação cadastrada.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona formulário para cadastro de avaliações. 2. Sistema apresenta formulário. 3. Ator seleciona a avaliação que é vinculada aos exercícios. 4. Ator preenche os campos do formulário para descrever enunciado dos exercícios
--

e demais campos para compor e complementar os exercícios como exemplos. 5. Sistema insere o exercício no banco de dados. Pós-Condição: Exercício cadastrado no sistema.	
Nome do fluxo alternativo (extensão)	Descrição
Linha 5 da sequência de eventos: Falta de informações	5.1 No momento de salvar, o sistema faz a verificação e constata que há dados faltantes. 5.2 É emitida mensagem de dados faltantes. 5.3 É retornado para o formulário de avaliação para correção do item faltante, sem executar o salvamento do mesmo.
Linha 5 da sequência de eventos: Dados não são válidos.	5.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos. 5.2 É emitida mensagem de dados inválidos. 5.3 É retornado para o formulário de exercícios para correção do item inválido, sem executar o salvamento do mesmo.

Quadro 8 – Caso de uso compor exercícios

O caso de uso compor respostas é apresentado no Quadro 9.

<p>Caso de uso: Cadastrar Respostas</p> <p>Descrição: O usuário (professor) é responsável pela composição das respostas. Essas respostas podem conter a descrição em código Álgebra Relacional e escrita formal.</p> <p>Evento Iniciador: Ator realiza o cadastro de respostas.</p> <p>Atores: Usuário (Professor) responsável pelo tipo de atividade.</p> <p>Pré-condição: Exercícios cadastrados.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona formulário para cadastro de respostas. 2. Sistema apresenta formulário. 3. Ator seleciona o exercício que é vinculado a resposta. 4. Ator preenche os campos do formulário para descrever a resposta e comentários ou exemplos. 5. Sistema insere a resposta no banco de dados. <p>Pós-Condição: Resposta cadastrada no sistema.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 5 da sequência de eventos: Falta de informações	5.1 No momento de salvar, o sistema faz a verificação e constata que há dados faltantes. 5.2 É emitida mensagem de dados faltantes. 5.3 É retornado para o formulário de respostas para correção do item faltante.

Quadro 9 – Caso de uso compor respostas

O caso de uso compor usuários é apresentado no Quadro 10.

<p>Caso de uso: Cadastrar Usuários</p> <p>Descrição: O usuário com login de administrador é o responsável pelo cadastro dos usuários. Um usuário é definido por um nível especial que define o acesso que ele terá no sistema.</p> <p>Evento Iniciador: Ator realiza o cadastro de usuários.</p> <p>Atores: Professor ou responsável pelo tipo de atividade.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona formulário para cadastro de usuários. 2. Sistema apresenta formulário. 3. Ator seleciona o tipo de usuário. 4. Ator preenche os dados do usuário como <i>login</i> e senha no formulário para compor o cabeçalho e as informações do aluno e acesso do mesmo ao sistema. 5. Sistema insere o usuário no banco de dados. <p>Pós-Condição: Usuário cadastrado no sistema.</p>

Nome do fluxo alternativo (extensão)	Descrição
Linha 5 da sequencia de eventos: Falta de informações	5.1 No momento de salvar, o sistema faz a verificação e constata que há dados faltantes. 5.2 É emitida mensagem de dados faltantes. 5.3 É retornado para o formulário de avaliação para correção do item faltante, sem executar o salvamento do mesmo.
Linha 5 da sequencia de eventos: Dados não são válidos.	5.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos. 5.2 É emitida mensagem de dados inválidos. 5.3 É retornado para o formulário de respostas para correção do item inválido, sem executar o salvamento do mesmo.

Quadro 10 – Caso de uso compor usuários

O caso de uso realizar avaliação é apresentado no Quadro 11.

<p>Caso de uso: Realizar Avaliação.</p> <p>Descrição: A realização da avaliação é realizada pelo usuário (professor) que define os critérios da avaliação como data e horário e repassa as instruções para realização da atividade.</p> <p>Evento Iniciador: Ator realiza a avaliação.</p> <p>Atores: Usuário (professor) pelo tipo de atividade e usuário (aluno) para responder a avaliação.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona formulário para realização de avaliação. 2. Sistema apresenta formulário. 3. Ator seleciona o tipo de usuário. 4. Ator define a avaliação a ser executada conforme a turma cadastrada. 5. Ator realiza a avaliação. 6. O sistema grava as questões respondidas até o momento. 7. O sistema finaliza a avaliação conforme o tempo definido ou término da avaliação. 8. O sistema armazena e gera relatórios que serão somente visualizados pelo
--

usuário (professor). Pós-Condição: Avaliação respondida e armazenada no sistema.	
Nome do fluxo alternativo (extensão)	Descrição
Linha 5 da sequencia de eventos: Falta de informações	5.1 No momento de salvar, o sistema faz a verificação e constata que há dados faltantes. 5.2 É emitida mensagem de dados faltantes. 5.3 É retornado para o formulário de avaliação para correção do item faltante, sem executar o salvamento do mesmo.
Linha 5 da sequencia de eventos: Dados não são válidos.	5.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos. 5.2 É emitida mensagem de dados inválidos. 5.3 É retornado para o formulário de realização de avaliação para correção do item inválido, sem executar o salvamento do mesmo.

Quadro 11 – Caso de uso realizar avaliação

O caso de uso emitir relatórios é apresentado no Quadro 12.

<p>Caso de uso: Emitir Relatórios.</p> <p>Descrição: A emissão dos relatórios pode ser executada conforme a seleção do relatório.</p> <p>Evento Iniciador: O Ator usuário (professor) realiza a emissão de relatórios.</p> <p>Atores: Usuário (professor) e usuário (aluno) conforme os níveis de acesso.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona formulário para emissão de relatórios. 2. Sistema apresenta formulário. 3. Ator seleciona o tipo de relatório. 4. Ator visualiza o relatório.
--

Quadro 12 – Caso de uso emitir relatórios

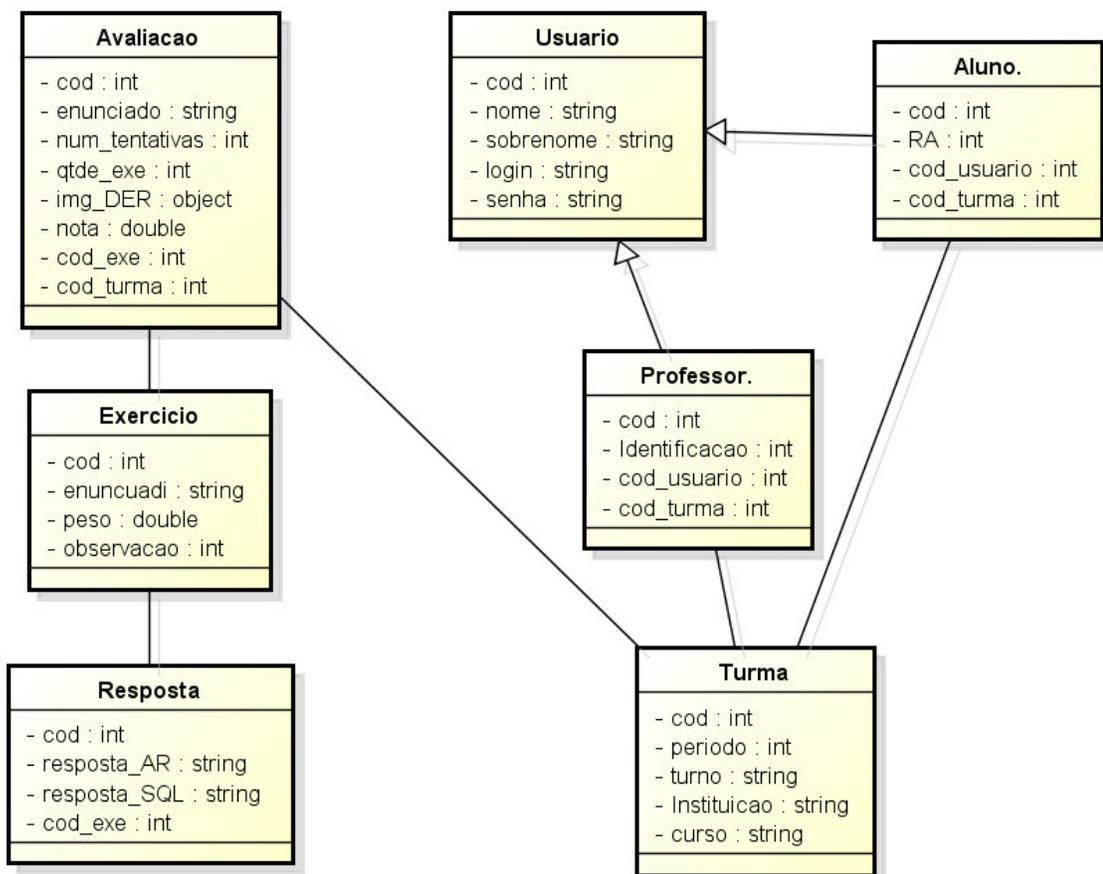
O caso de uso realizar consultas é apresentado no Quadro 13.

<p>Caso de uso: Realizar Consultas.</p> <p>Descrição: A realização das consultas pode ser realizada por ambos os usuários no console do sistema.</p> <p>Evento Iniciador: O Ator usuário (professor/aluno) realiza as consultas.</p> <p>Atores: Usuário (professor) e usuário (aluno) conforme os níveis de acesso.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona formulário para consulta. 2. Sistema apresenta formulário. 3. Ator insere a consulta. 4. Sistema retorna a consulta realizada. 	
Nome do fluxo alternativo (extensão)	Descrição
Linha 5 da sequencia de eventos: Falta de	5.1 No momento de executar, o sistema faz a

informações	verificação e constata que há dados faltantes. 5.2 É emitida mensagem de dados faltantes.
Linha 5 da sequencia de eventos: Dados não são válidos.	5.1 No momento de executar, o sistema faz a verificação e constata que há dados inválidos. 5.2 É emitida mensagem de dados inválidos.

Quadro 13 – Caso de uso realizar consultas

Na Figura 11 está o diagrama de classes de análise do sistema.



powered by astah®

Figura 11 – Diagrama de classes de análise do sistema

As classes apresentadas no diagrama da Figura 9 estão documentadas a seguir. Nessas classes, o atributo “cod” é necessário como o campo chave primária (*Primary Key*) criado no banco de dados em cada tabela como uma chave única incrementada automaticamente pelo comando *AUTOINCREMENT* do banco de dados.

No Quadro 14 está a apresentação da classe Avaliacao.

Identificação:	Avaliacao
Descrição:	Uma avaliação é composta por um conjunto de itens aos quais será atribuída nota, quantidade de exercícios, número de tentativas e imagem do diagrama proposto no exercício.
Requisitos:	RF01
Atributos:	cod (número): responsável pela chave primária de identificação na tabela. cod_exe (número): código do exercício vinculado a avaliação. cod_turma (número): código da turma vinculado a avaliação. enunciado (texto): descrição da avaliação. num_tentativas(numero)
Métodos:	boolean insert(); boolean update(id Integer); boolean delete(id Integer); void listaTodos(); T buscaPorId(id : Integer);

Quadro 14 – Descrição da classe avaliacao

No Quadro 15 está a apresentação da classe Exercicios.

Identificação:	Exercicios
Descrição:	Contém os exercícios cadastrados no sistema. Um exercício é composto por pesos que equivalem à nota atribuída à avaliação.
Requisitos:	RF01
Atributos:	cod (número): responsável pela chave primária de identificação na tabela. cod_avaliacao (número): código da avaliação vinculado aos exercícios.
Métodos:	boolean insert(); boolean update(id Integer); boolean delete(id Integer); void listaTodos(); T buscaPorId(id : Integer);

Quadro 15 – Descrição da classe Exercicios

No Quadro 16 está a apresentação da classe Respostas.

Identificação:	Respostas
Descrição:	Contém as respostas cadastradas no sistema. Uma resposta é composta por um conjunto por um ou mais exercícios.
Requisitos:	RF01
Atributos:	cod (número): responsável pela chave primária de identificação na tabela. cod_exe (número): código do exercício vinculado as respostas.
Métodos:	boolean insert(); boolean update(id Integer); boolean delete(id Integer); void listaTodos(); T buscaPorId(id : Integer);

Quadro 16 – Descrição da classe Respostas

No Quadro 17 está a apresentação da classe Usuarios.

Identificação:	Usuarios
Descrição:	Contém os usuários cadastrados no sistema.
Requisitos:	RF01
Atributos:	cod (número): responsável pela chave única de identificação na tabela. cod_exe (número): código do exercício vinculado a avaliação. cod_turma (número): código da turma vinculado a avaliação.
Métodos abstratos:	boolean insert(); boolean update(id Integer); boolean delete(id Integer); void listaTodos(); T buscaPorId(id : Integer);

Quadro 17 – Descrição da classe Usuarios

No Quadro 18 é apresentada a descrição da classe Aluno.

Identificação:	Aluno
Descrição:	Os alunos são de um dos tipos de usuário cadastrados no sistema.
Requisitos:	RF01
Atributos:	cod (número): responsável pela chave primária de identificação na tabela. cod_exe (número): código do exercício vinculado a avaliação. cod_turma (número): código da turma vinculado a avaliação.
Métodos Implementados:	boolean insert(); boolean update(id Integer); boolean delete(id Integer); void listaTodos(); T buscaPorId(id : Integer);

Quadro 18 – Descrição da classe Aluno

No Quadro 19 está a apresentação da classe Professor.

Identificação:	Professor
Descrição:	Os professores é um dos tipos de usuário cadastrado no sistema.
Requisitos:	RF01
Atributos:	cod (número): responsável pela chave primaria de identificação na tabela. cod_exe (número): código do exercício vinculado a avaliação. cod_turma (número): código da turma vinculado a avaliação.
Métodos Implementados:	boolean insert(); boolean update(id Integer); boolean delete(id Integer); void listaTodos(); T buscaPorId(id : Integer);

Quadro 19 – Descrição da classe Professor

No Quadro 20 está a apresentação da classe Turmas.

Identificação:	Turmas
Descrição:	Contém as turmas cadastradas no sistema.
Requisitos:	RF01
Atributos:	cod (número): responsável pela chave primária de identificação na

	tabela.
Métodos Implementados:	boolean insert(); boolean update(id Integer); boolean delete(id Integer); void listaTodos(); T buscaPorId(id : Integer);

Quadro 20 – Descrição da classe Turmas

Figura 12 apresenta o diagrama de entidades e relacionamentos (DER) que representam o banco de dados da aplicação.

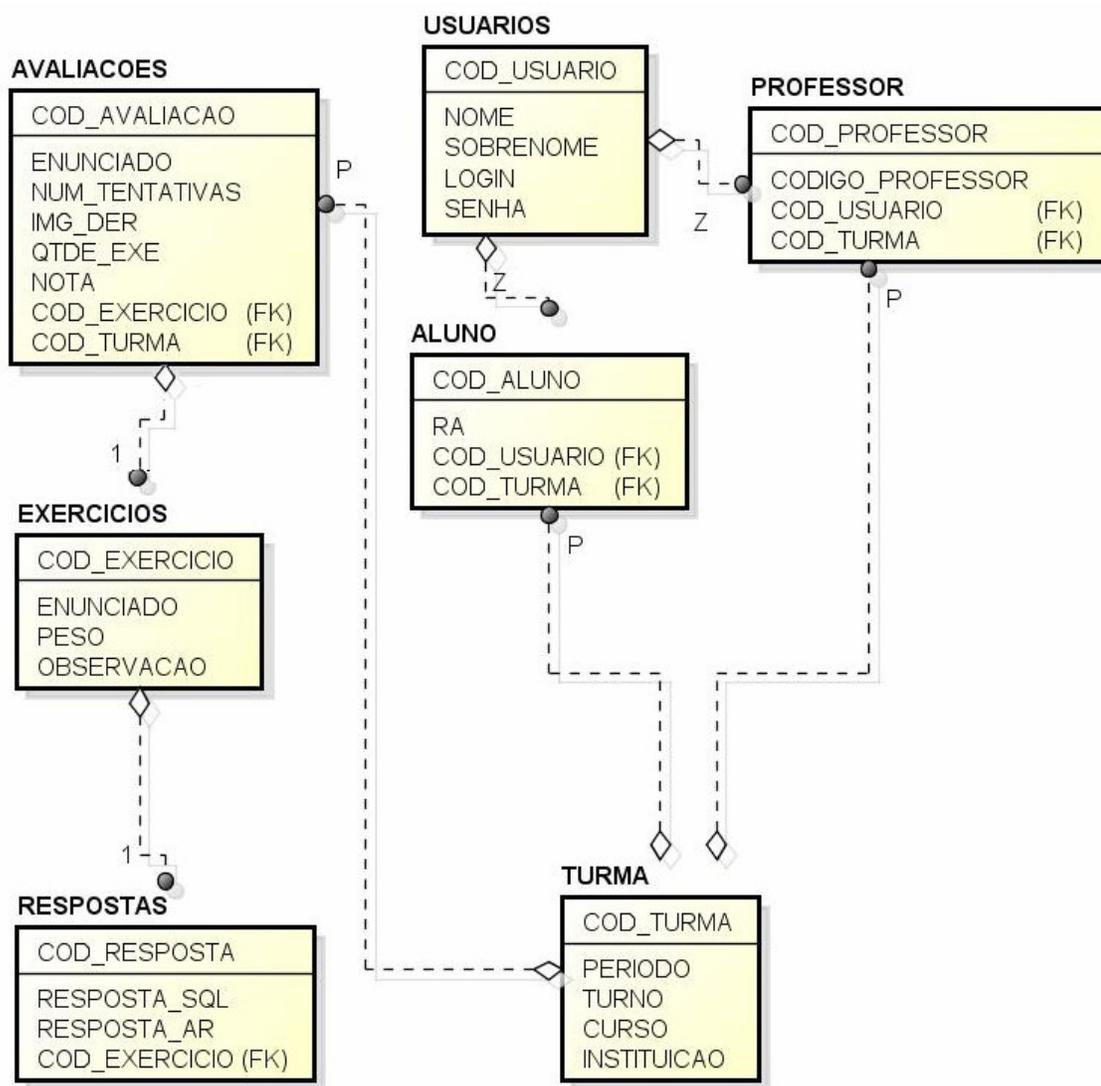


Figura 12 – Diagrama de entidades e relacionamentos do banco de dados

No Quadro 21 estão os campos da tabela de Avaliacao. Uma avaliação contém muitos exercícios para resolução pelos alunos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
cod_avaliacao	Numérico	Não	Sim	Não	
enunciado	Numérico	Não	Não	Não	
num_tentativas	Texto	Não	Não	Não	
img_der	Blob	Sim	Não	Não	Contém a imagem do DER
qtde_exe	Numérico	Não	Não	Não	Quantidade de exercícios da avaliação
nota	Ponto flutuante	Não	Não	Não	Nota total atribuída a avaliação
cod_exercicio	Numérico	Não	Não	Sim	
cod_turma	Numérico	Não	Não	Sim	

Quadro 21 – Campos da tabela avaliacao

No Quadro 22 estão os campos da tabela de exercícios. Um exercício contém muitas respostas e pode ser atribuído a muitas provas.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
cod_exercicio	Numérico	Não	Sim	Não	
observacao	Texto	Sim	Não	Não	
peso	Ponto flutuante	Não	Não	Não	Contém o peso de cada questão para fechar uma nota total na avaliação.

Quadro 22 – Campos da tabela exercicios

No Quadro 23 estão os campos da tabela de Respostas. Uma resposta pode conter um ou muitos exercícios vinculados a ela.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
cod_resposta	Numérico	Não	Sim	Não	
resposta_sql	Texto	Sim	Não	Não	
resposta_ar	Texto	Sim	Não	Não	
cod_exercicio	Numérico	Não	Não	Sim	

Quadro 23 – Campos da tabela respostas

No Quadro 24 estão os campos da tabela de Turmas. Uma turma é composta por várias avaliações. E cada avaliação pode ser aplicada para mais de uma turma.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
cod_turma	Numérico	Não	Sim	Não	
periodo	Numérico	Não	Não	Não	
turno	Texto	Não	Não	Não	
curso	Texto	Não	Não	Não	
instituicao	Texto	Não	Não	Sim	

Quadro 24 – Campos da tabela turmas

No Quadro 25 estão os campos da tabela de usuários. Um usuário pode ser cadastrado separadamente do seu tipo contendo para acesso as funcionalidades básicas do sistema como consultas. Para os tipos de usuários estão em tabelas diferentes que contém alunos e professores.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
cod_usuario	Numérico	Não	Sim	Não	
nome	Texto	Não	Não	Não	
sobrenome	Texto	Não	Não	Não	
login	Texto	Não	Não	Não	
senha	Texto	Não	Não	Não	

Quadro 25 – Campos da tabela usuarios

No Quadro 26 estão os campos da tabela de alunos. Os alunos são cadastrados pelo usuário que possui o nível especial de professor.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
cod_aluno	Numérico	Não	Sim	Não	
ra	Texto	Não	Não	Não	
cod_usuario	Numérico	Não	Não	Sim	
cod_turma	Numérico	Não	Não	Sim	

Quadro 26 – Campos da tabela alunos

No Quadro 27 estão os campos da tabela de professores. Os professores serão cadastrados por meio de uma senha padrão gerada na construção do sistema para o primeiro acesso.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
cod_professor	Numérico	Não	Sim	Não	
codigo_professor	Texto	Não	Não	Não	
cod_usuario	Numérico	Não	Não	Sim	
cod_turma	Numérico	Não	Não	Sim	

Quadro 27 – Campos da tabela professores

4.3 APRESENTAÇÃO DO SISTEMA

A apresentação do sistema está organizada em console e composição de avaliação.

4.3.1 Console

Na Figura 13 está a tela principal do console. Nessa tela há uma paleta de manipulação de documentos de texto contendo operações de salvar, imprimir, copiar e colar, entre outras. Na interface é disponibilizado um espaço livre para digitação das consultas com apoio de uma paleta contendo os operadores algébricos para utilização, juntamente um componente que apresenta o resultado obtido das consultas realizadas no formato de colunas e linhas. Facilitando, assim, a visualização dos resultados obtidos. A interface possibilita visualizar as tabelas e as colunas de um banco de dados em forma de árvore para ajudar na manipulação.

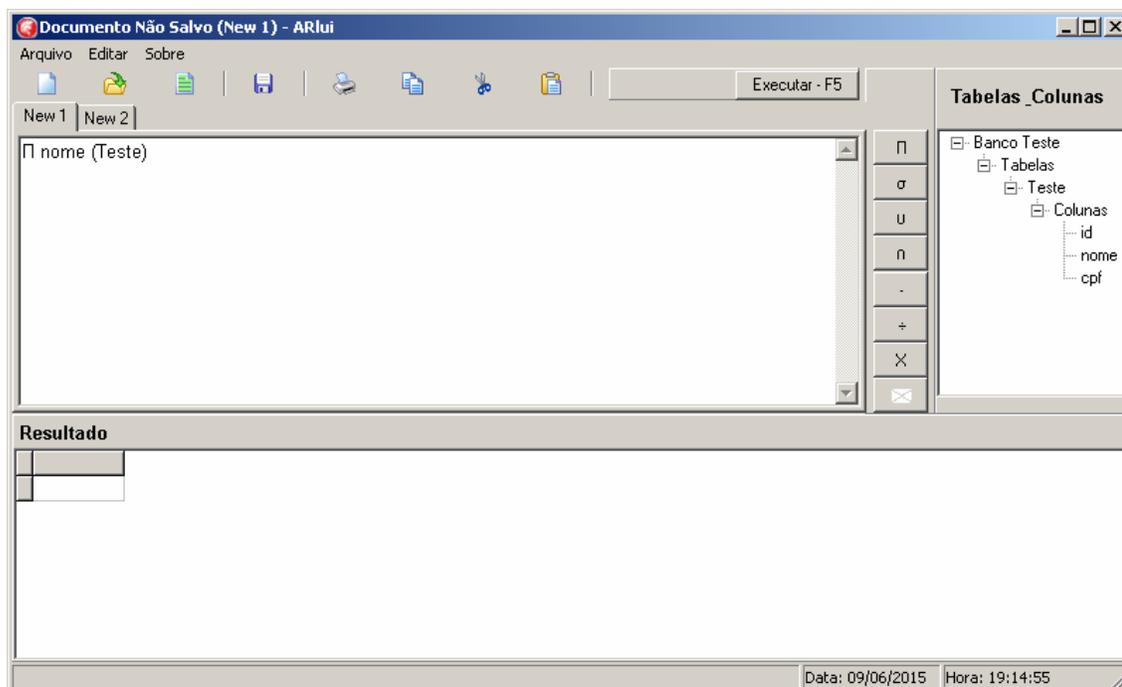


Figura 13 – Tela principal

Na Figura 14 está destacada, por um quadro, a paleta que permite a manipulação de arquivos textos gerados dentro do sistema. Essa paleta contém as operações básicas de manipulação como novo, abrir, salvar, selecionar tudo, imprimir, copiar, recortar e colar.

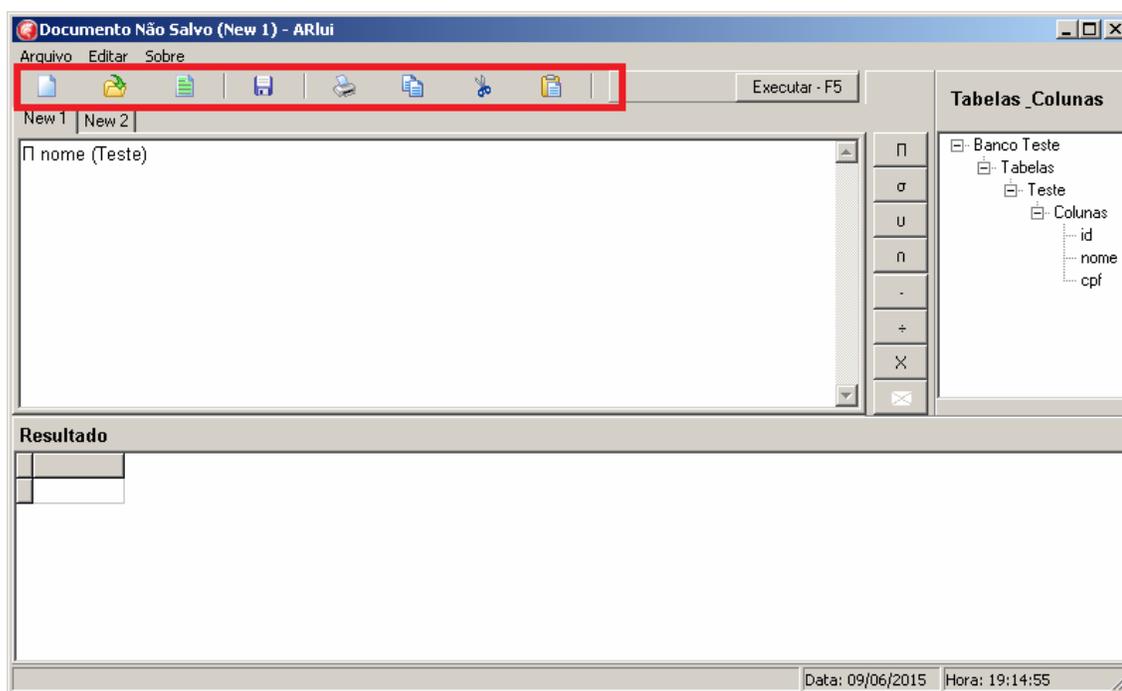


Figura 14 – Paleta de manipulação de arquivos texto

Na Figura 15 está destacada, por um quadro, a lista de tabelas e colunas contidas no banco de dados selecionado. As informações são meramente ilustrativas para auxiliar na montagem das consultas.

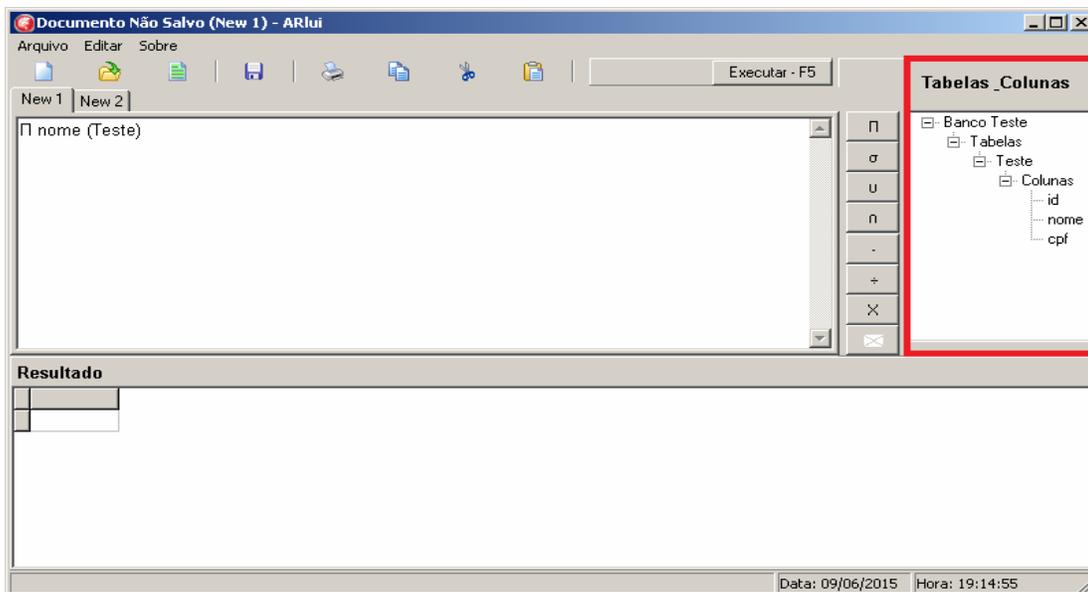


Figura 15 – Tabelas e colunas

Na Figura 16 está destacado o ambiente livre para montagem das consultas em Álgebra Relacional, com opção de criar e manipular várias abas para executar as consultas.

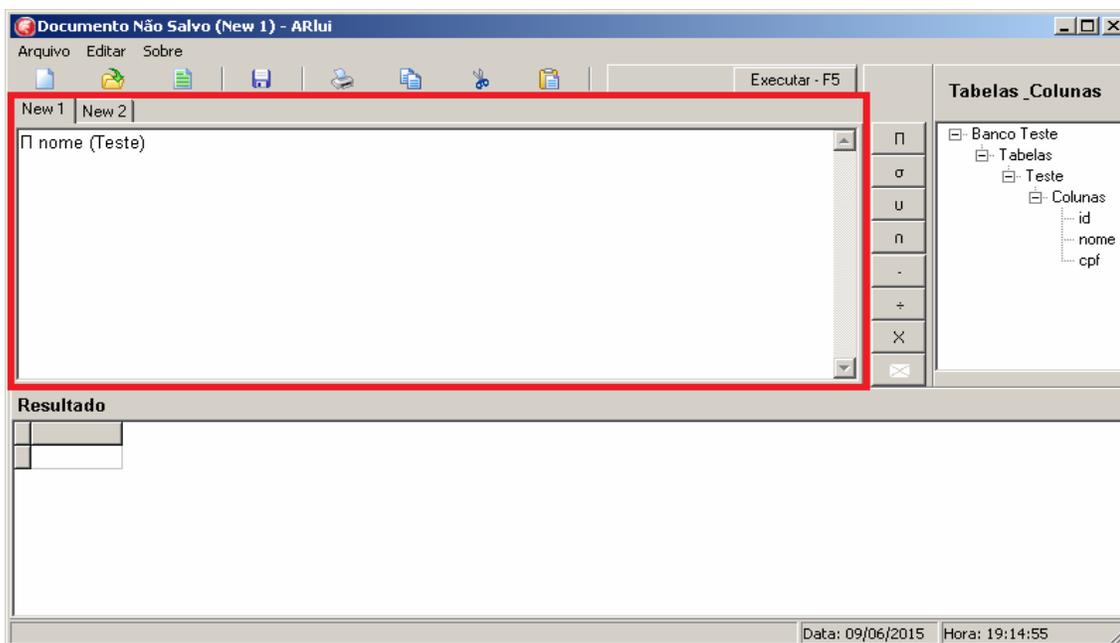


Figura 16 - Área de construção das consultas

Na Figura 17 está destacada a paleta para o auxílio na construção de comandos em Álgebra Relacional. Devido à dificuldade em digitação de alguns operadores algébricos foi planejada essa solução: os operadores são incluídos no ambiente de construção e com um *click* em cima do operador é realizada a escolha/seleção do mesmo.

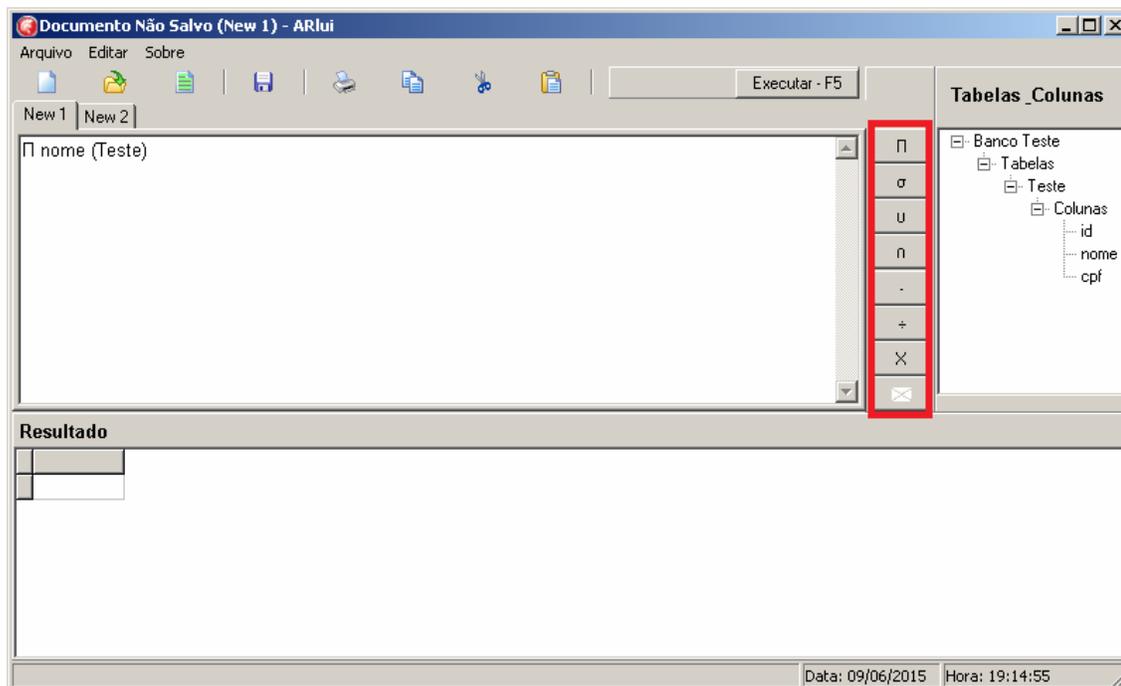


Figura 17 – Paleta de operadores de Álgebra Relacional

Na Figura 18 está destacado o ambiente que retorna o resultado das consultas realizadas. Nesse ambiente, os resultados podem ser visualizados em colunas e linha. Essa forma de apresentação segue o modelo que muitas ferramentas de manipulação de bancos de dados utilizam para mostrar resultados de consultas.

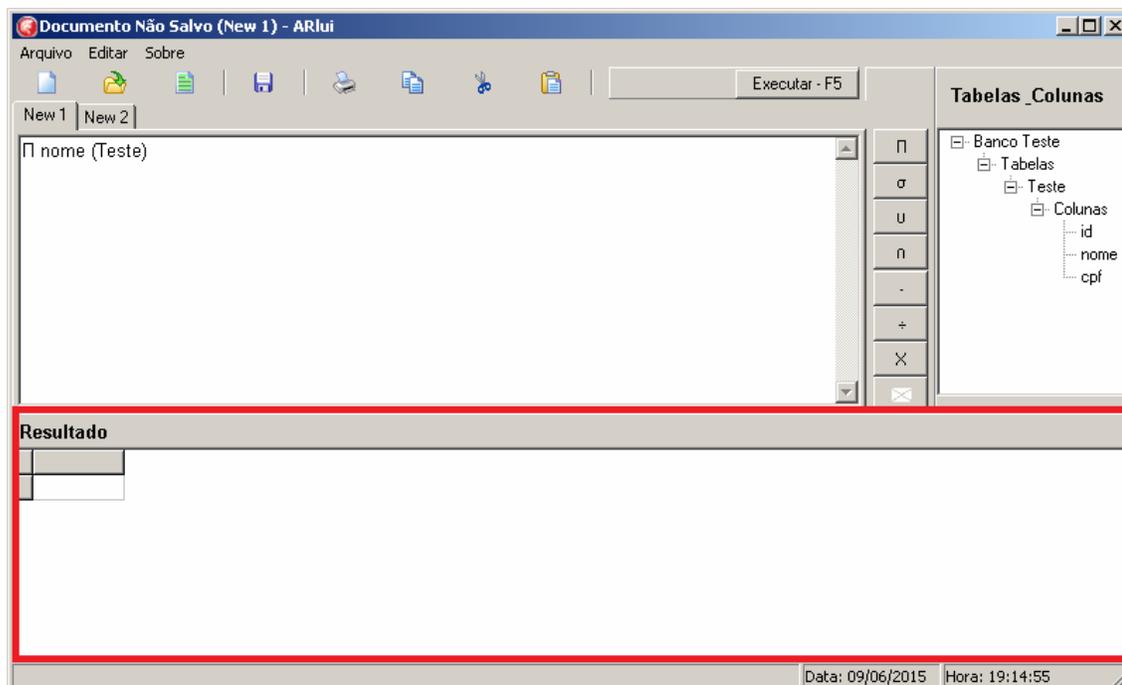


Figura 18 – Visualização de resultados

Na Figura 19 está o ambiente no qual é configurado o banco de dados (*Data Bases Configuration*) a ser utilizado pelo sistema. Nessa configuração é inserido o caminho do banco, *user name* e *password* de conexão com o banco de dados e também, a opção de criação do banco de dados.

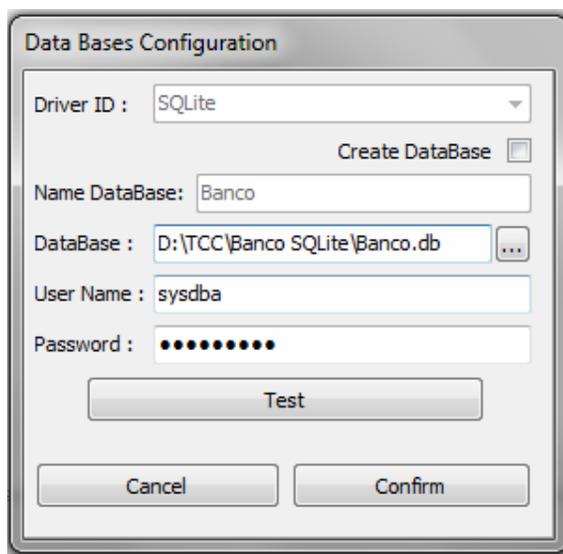


Figura 19 – Configuração de bases de dados

O console auxilia na apresentação de instrução trazendo uma “hint” (mensagem que aparece automaticamente quando posicionamos o cursor do mouse em cima da funcionalidade) em cada botão conforme a funcionalidade. A área selecionada da Figura 20 apresenta o *hint* nos botões, no caso os operadores de Álgebra Relacional.

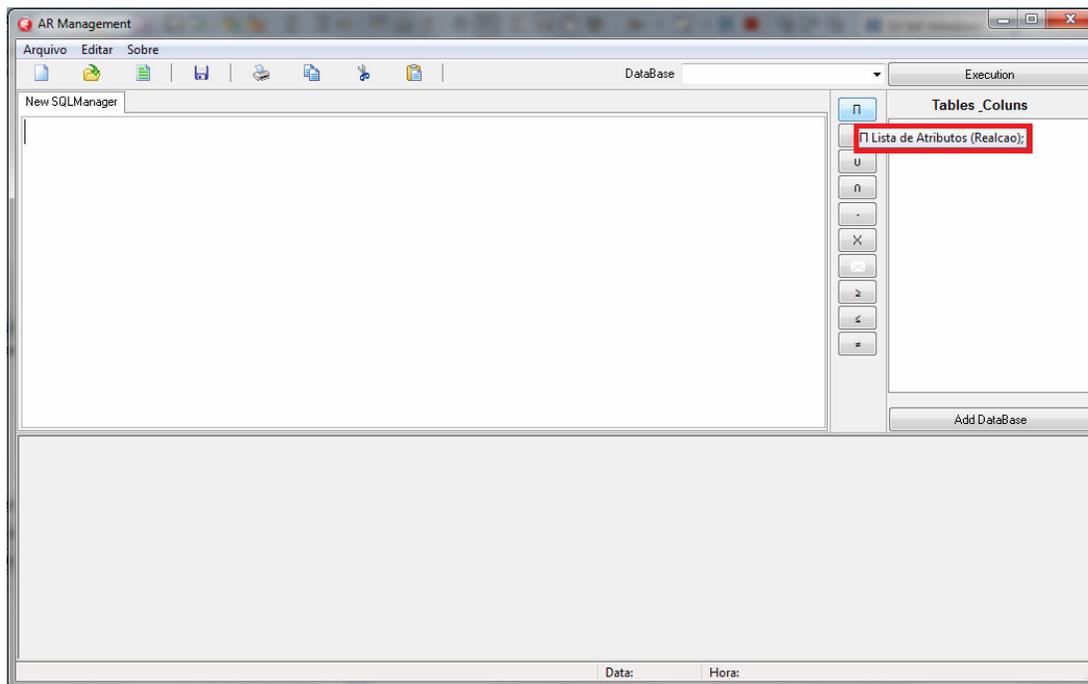


Figura 20 – Hint nos botões

É necessário a configuração de uma base de dados para execução das consultas (Figura 19). Após a configuração da base de dados é apresentada uma lista, como mostrada na Figura 21, contendo a base de dados e as tabelas e campos relacionados a cada tabela, bem como o respectivo tipo de dado.

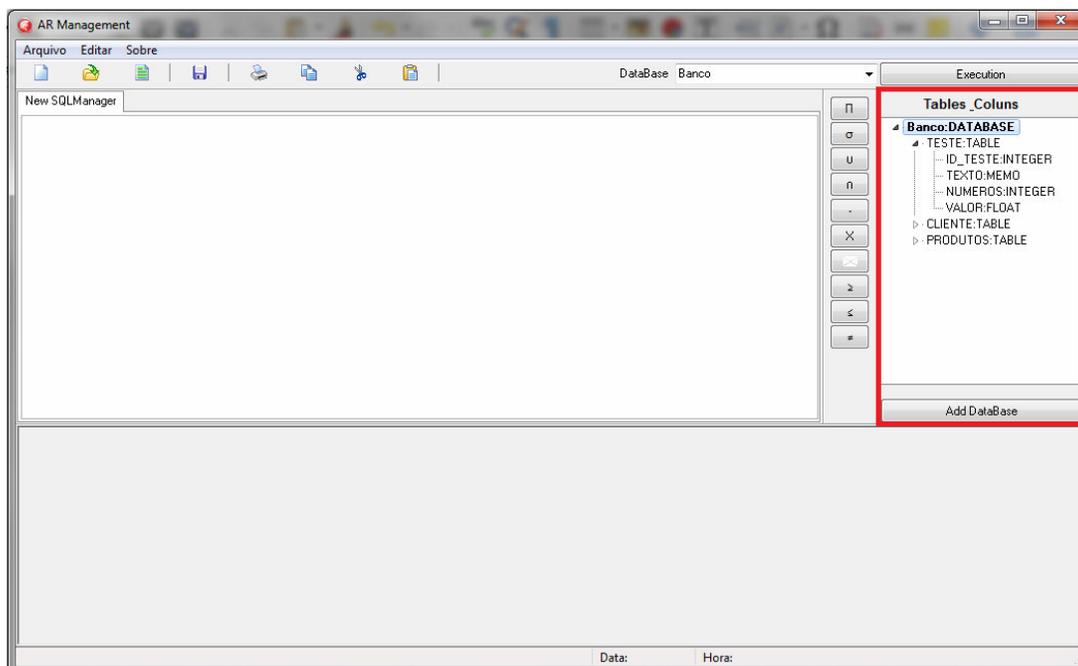


Figura 21 – Lista de bases de dados, tabela e campos

A execução das consultas é composta pelo usuário utilizando os campos das tabelas da base de dados selecionada e os operadores apresentados. A composição das consultas (Figura 22) é feita com o auxílio de uma paleta contendo os caracteres específicos da álgebra relacional, apresentados na Figura 17.

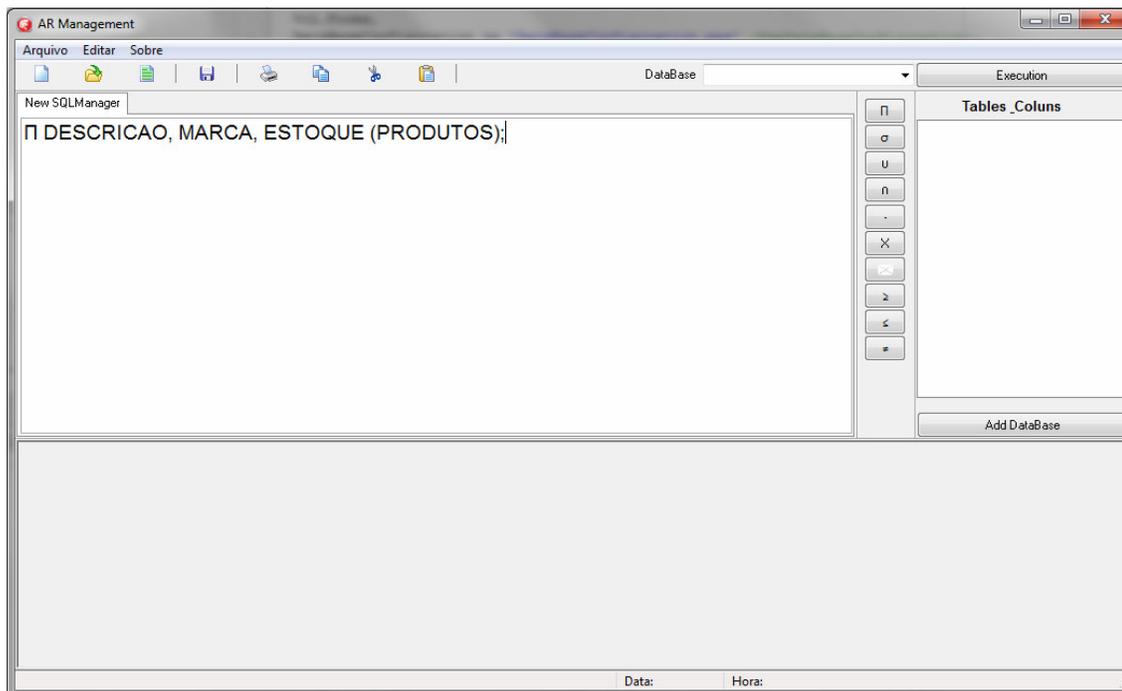


Figura 22 – Montagem da consulta

Após a elaboração das consultas é necessário clicar no botão “Execution” e visualizar o resultado na área de apresentação do resultado, em formato de linhas e coluna ou uma mensagem se a execução é mal sucedida. A Figura 23 apresenta o resultado de uma execução bem sucedida.

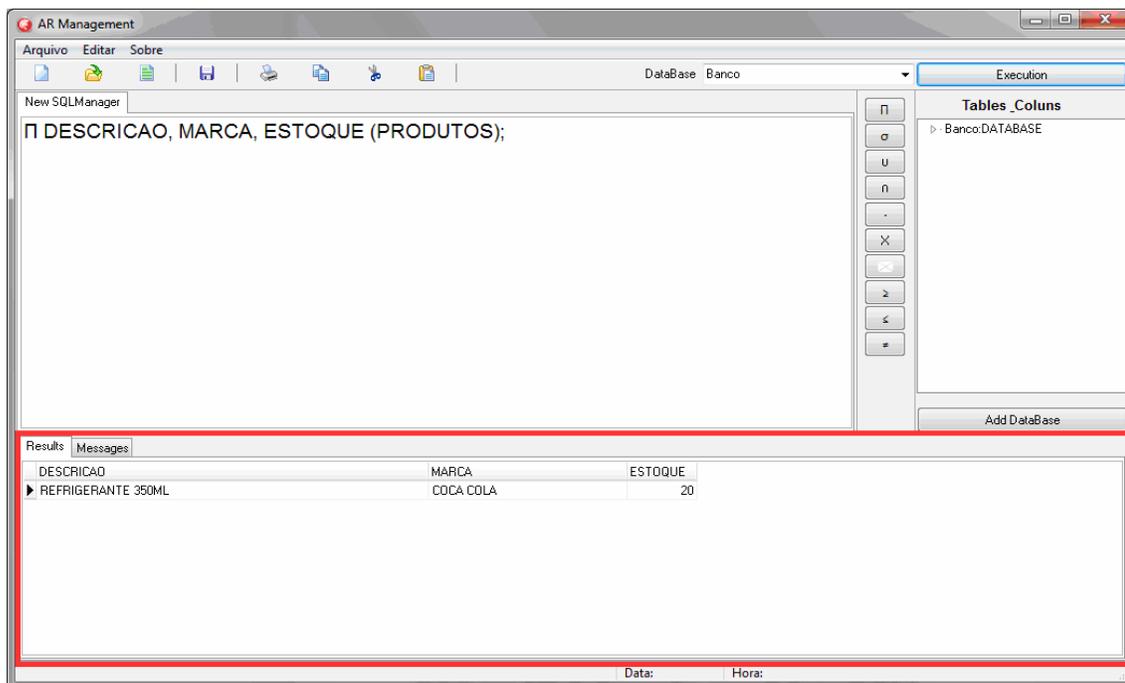


Figura 23 – Resultado da execução de uma consulta

Se a execução não foi bem-sucedida é apresentado, na forma de texto informativo, o que resultou a não execução. Na Figura 24 foi simulado um erro de sintaxe, atributo DESCRICAO da relação PRODUTOS está escrito DESCRICAOKKK e por esse motivo não foi possível executar a consulta.

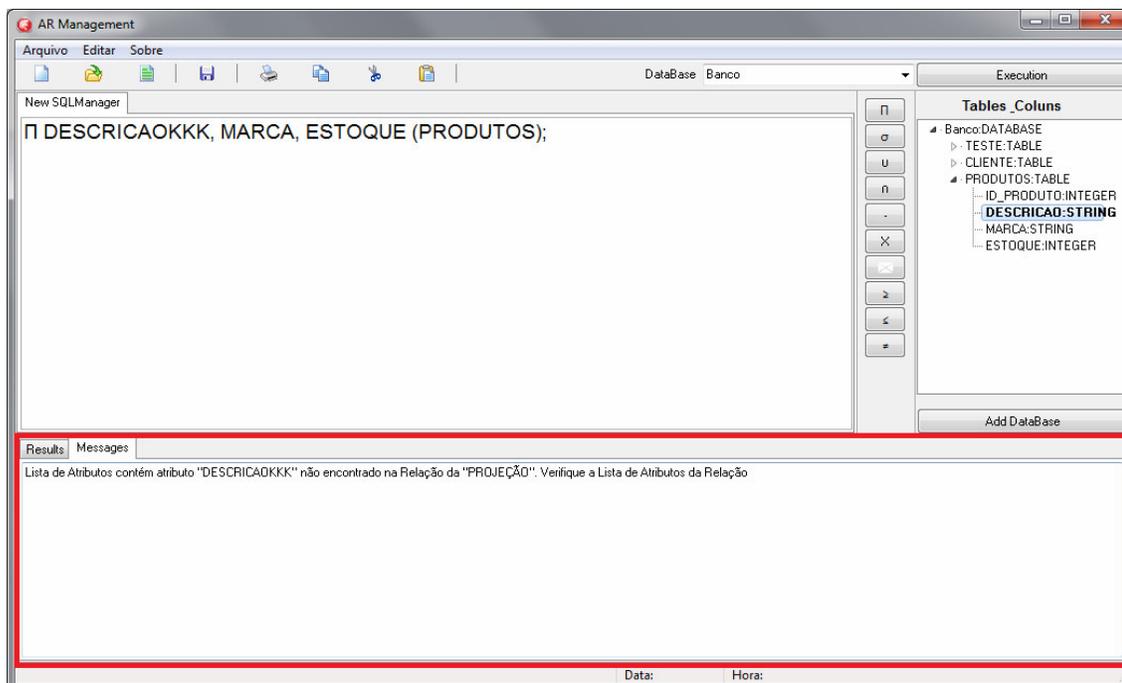


Figura 24 – Mensagem de erro de sintaxe

O mesmo ocorre quando há outro tipo de consulta, como, por exemplo, seleção. A Figura 25 contém uma seleção na tabela teste na qual não é especificado a forma de comparação dos valores.

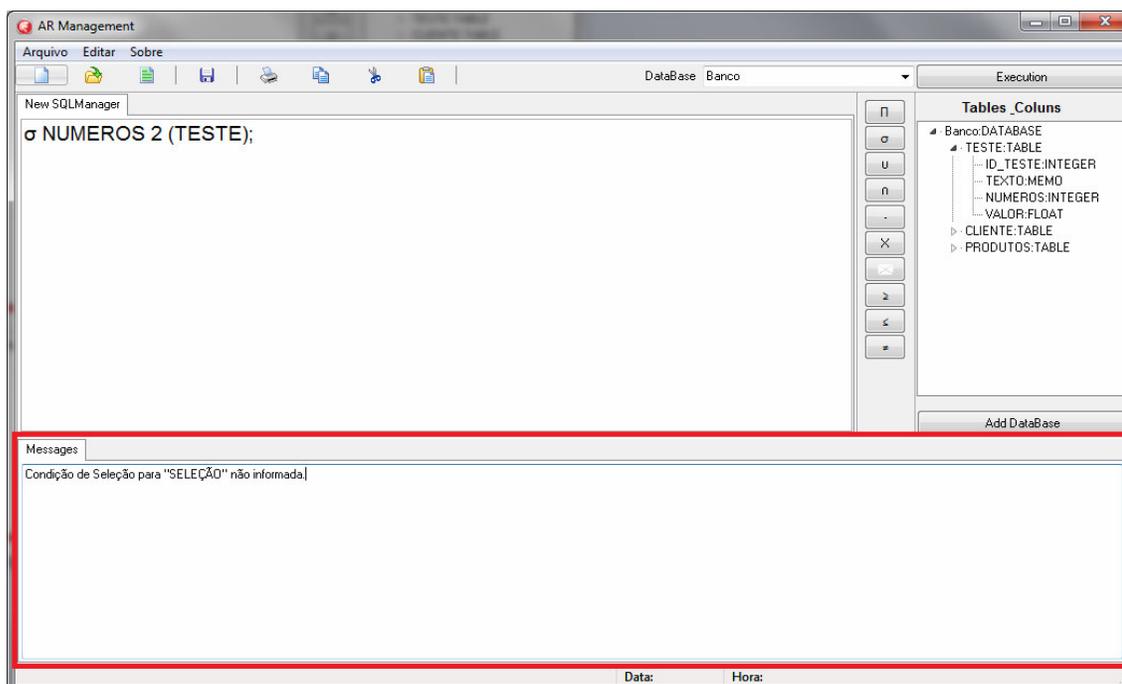


Figura 25 – Mensagem de erro na montagem da seleção

E erros na estrutura da consulta também são mostrados em um texto informativo, como apresentado no Figura 26.

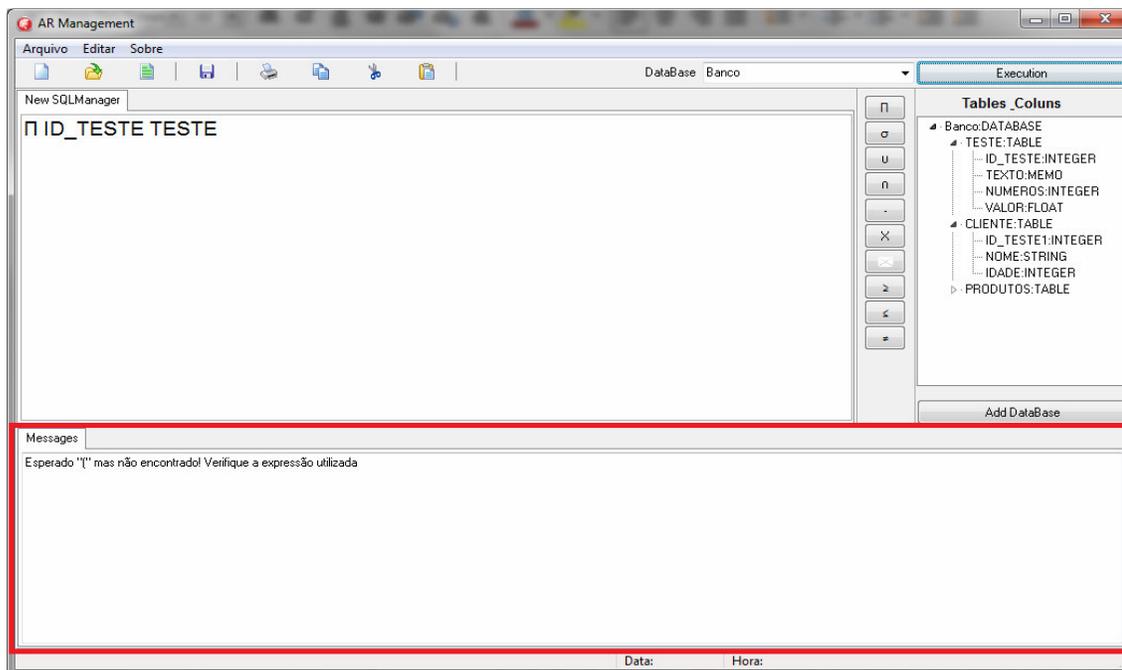


Figura 26 – Erro de estrutura da consulta

4.3.2 Composição de avaliações

Para compor uma avaliação é necessário realizar o cadastro de avaliações. A Figura 27 apresenta a tela de cadastro de avaliações.

Avaliações

Manutenção de Dados:

Cód. Avaliação Nro. Tentativas Modelo (DER)

Qtde. Exercícios Nota

Título da Avaliação

tcc2

Carregar Imagem DER Visualizar DER

Exercícios Prova

Enunciado Avaliação	Enunciado Avaliação
1	tcc2
2	avalicao 1

Figura 27 – Cadastro de avaliações

Cada avaliação contém uma ou mais turmas vinculados a ela. Assim, é necessário compor o cadastro de turmas, conforme apresenta a tela da Figura 28.

Cadastro de Turmas

Manutenção de Dados:

Cód. Turma

Período

Curso

Instituição

Consulta de Dados:

◀ ◁ ▷ ▶ +

Cód. Turma	Curso	Período	Instituição
<input type="text"/>			

Avaliação

Cód. Turma	Cód. Avaliação
<input type="text"/>	<input type="text"/>

Figura 28 – Cadastro de turmas

E para cada avaliação é necessário compor a avaliação com exercícios. O formulário para a composição dos exercícios é apresentado na Figura 29.

Manutenção de Dados:

Cód. Exercício Nota do Exercício

Enunciado (Descrição do Exercício)

sfsdfs

Observação

dsfdfs

Consulta de Dados:

◀ ◁ ▷ ▶ +

Cód. Exercício	Nota	Enunciado Exercício	Obs Exercício
1	1	sfsdfs	dsfdfs
2	5	meu exercicio 5	ghjhfg

Avaliação

Cód. Exercício	Cód. Avaliação
<input type="text"/>	<input type="text"/>

Figura 29 – Cadastro de exercícios

E para cada exercício podem ser vinculadas respostas. No formulário de respostas é possível elaborar a resposta em Álgebra Relacional e em SQL, como apresentado na Figura 30.

Manutenção de Dados:

Cód. Exercício Nota do Exercício

Resposta em AR

Respostas em SQL

Consulta de Dados:

Cód. Resposta	Nota

Exercícios das Respostas

Cód. Resposta	Resposta em AR	Cód. Exe	Enunciado

Figura 30 – Cadastro de respostas para exercícios

Após prontos os cadastros para a composição dos exercícios. As avaliações podem ser realizadas. Na Figura 31 está o ambiente no qual são realizadas as avaliações cadastradas. A montagem da tela consiste nos exercícios cadastrados e no cabeçalho atribuído a cada avaliação. Na tela de avaliações há botões para

auxílio na inserção de operadores lógicos utilizados na montagem de consultas em Álgebra Relacional.

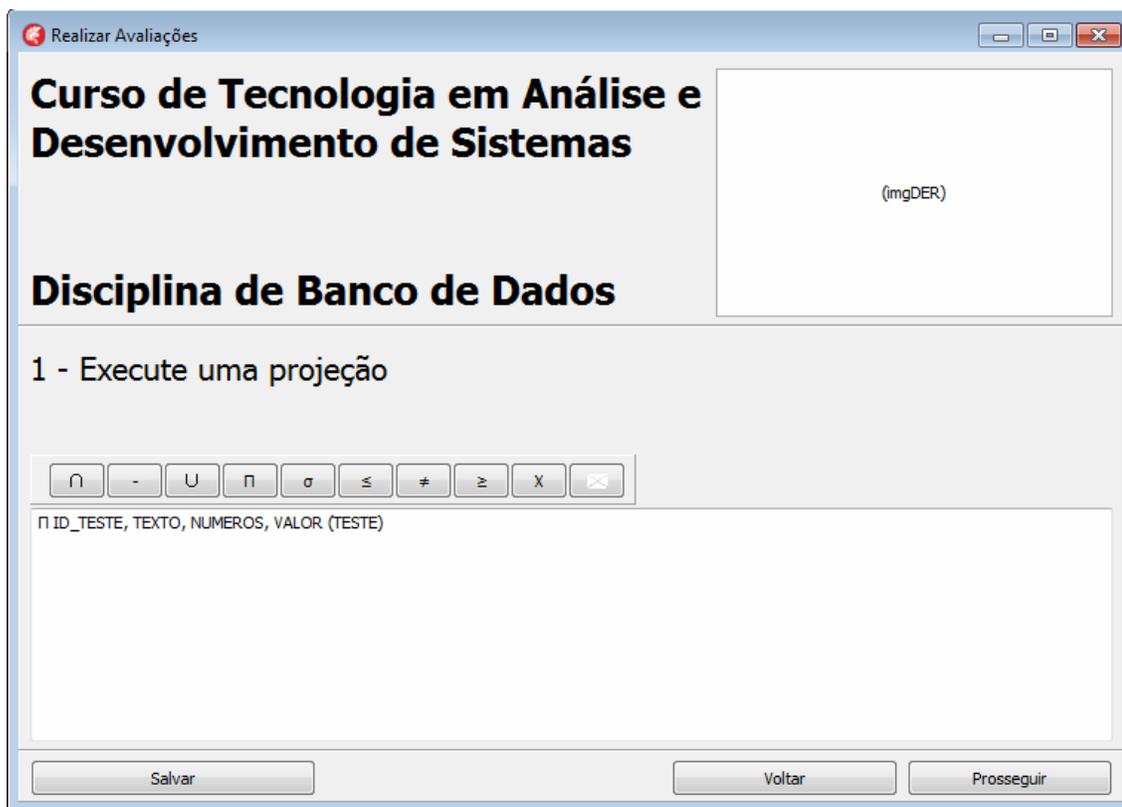


Figura 31 – Realização de avaliação

4.4 IMPLEMENTAÇÃO DO SISTEMA

A seguir são apresentados os principais códigos utilizados para o desenvolvimento desta ferramenta. O sistema interpretador de consultas contém uma interface dinâmica encontrada em editores de texto. Esse trabalho com editores de texto envolve certa complexidade referente à criação de componentes dinâmicos em tempo de execução (*run time*). Como exemplo principal está a criação da área de texto para digitação das consultas, apresentada na Listagem 1.

```

procedure CreateTextArea();
var
  Rich : TRichEdit;
  TabSheet : TTabSheet;
begin
  try
    iPageCount := iPageCount + 1;
    TabSheet := TtabSheet.Create(FrmArManagement.pgControlEditorText);
    TabSheet.Caption := 'New SQLManager';
    TabSheet.Name := 'Aba' + IntToStr(iPageCount);
    TabSheet.PageControl := FrmArManagement.pgControlEditorText;
    FrmArManagement.pgControlEditorText.ActivePage := TabSheet;

    Rich := TRichEdit.Create(TabSheet);
    with Rich do
      begin
        Parent := TabSheet;
        Visible := True;
        Align := alClient;
        Name := 'Rich' + IntToStr(iPageCount);
        ScrollBars := ssBoth;
        Clear;
        SetFocus;
        PopupMenu := FrmArManagement.PopupMenuRichEdit;
        OnChange := FrmArManagement.TextAreaOnChange;
      end;

      AddArrayTextAreaProperties(iPageCount, False, TabSheet.Name, Rich.Name,
        '');
    except on E: Exception do
      ErrorException(sUnitName, 'CreateTextArea', e );
    end;
  end;
end;

```

Listagem 1 – Código para criação da área de texto para digitação das consultas

No código da Listagem 1 está a criação de um PageControl e um TabSheet que fazem parte de um componente para a criação de abas dentro do sistema. Para cada TabSheet criado há um RichEdit que é um componente para digitação de texto e após isto uma lista de propriedades que é específica para cada área de texto que é adicionado a cada vez que é criada uma nova área de texto executado pela função AddArrayTextAreaProperties. Nesta função ocorre a criação de registros dinâmicos para inserção de parâmetros referentes à área de texto como mostrado no código da Listagem 2.

```

  SetLength(TextArea, iIndex);
  with TextArea[iIndex] do
    begin
      Save := bSave;
      NameTab := sNameTab;
      NameRich := sNameRich;
      DirectorySave := sDirectory;
    end;

```

Listagem 2 – Criação de registros dinâmicos para a inserção de parâmetros

Por meio da lista de propriedades de cada área de texto é possível saber se foram salvas ou não. Dentro dessas implementações de código, considerando somente da área de texto, há muitas outras implementações que consistem em salvar o documento, validar se foi salvo, abrir um documento existente.

Para a conexão com o banco de dados foi montada uma tela para inserções de informações referentes à conexão como pode ser visualizado na Figura 17. Essas informações são transportadas entre as telas até o componente de conexão utilizando um *record*, que permite que seja criado um registro com vários campos de diferentes tipos como no exemplo apresentado na Listagem 3.

```

if not VerificationExistsDataBase(edtNameDataBase.Text) then
begin
  if AddDefinationsDataBase then
  begin
    DataBaseParams.Update(edtNameDataBase.Text,
                          EdtDataBase.Text,
                          EdtUserName.Text,
                          edtPassword.Text);

    ModalResult := mrOk;
  end;
end
else
  Application.MessageBox('Database already exists. check list data bases
!', PChar(Application.Title), MB_ICONINFORMATION + MB_OK)

```

Listagem 3 – Código para criação da registro com campos de tipos distintos

Primeiramente é realizada uma verificação da existência de um banco de dados com o mesmo nome e posteriormente são atribuídas as configurações da conexão ao registro. Para auxiliar na construção de consultas é realizado um mapeamento do banco de dados, que foi selecionado para a execução de consultas. O mapeamento consiste em verificar todas as tabelas e campos, com seus determinados tipos de dados e montar uma lista contendo todas essas informações como mostra a Figura 13. No código da Listagem 4 é exemplificado esse mapeamento.

```

sNameDataBase := ChangeFileExt(ExtractFileName(Connect.Params.Database),
'');
qryTables := TFDQuery.Create(nil);
qryTables.Name := 'qryTables';
qryTables.Connection := Connect;

qryFields := TFDQuery.Create(nil);
qryFields.Name := 'qryFields';
qryFields.Connection := Connect;

TreeView := SQLManager.FrmSQLManager.twTableAndColuns;

try

```

```

tnDataBase := TreeView.Items.Add(nil, sNameDataBase + ':DATABASE');
tnDataBase.Selected := True;
qryTables.Close;
qryFields.Close;
qryTables.SQL.Clear;
qryTables.SQL.Add(' SELECT NAME FROM SQLITE_MASTER');
qryTables.Open();

```

Listagem 4 – Código de mapeamento do banco de dados

No código da Listagem 5 é exemplificado o uso de um TreeView que monta uma árvore de Informações, contendo itens e seus subitens. Essa composição apresenta como são formadas as tabelas e seus campos em um banco de dados. Para a montagem de tabelas e campos há uma execução de uma consulta na tabela master do banco que é utilizado. No caso da aplicação desenvolvida o banco de dados é o SQLite, então essa tabela é a SQLITE_MASTER que retorna todas as tabelas do sistema e seus campos.

Como regra principal está o mapeamento da consulta escrita em álgebra relacional com caracteres algébricos. Para esse sistema foi utilizado o mapeamento posicional para identificar o que foi utilizado em cada estrutura da consulta.

```

if (Pos('Π', sTexto) > 0) then
begin
  sCampos := Copy(sTexto, Pos('Π', sTexto) + 1, Pos('(', sTexto) - 1 );
  sTabela := Copy(sTexto, Pos('(', sTexto) + 1, Pos(')', sTexto) - 1 );

  if (Pos('σ', sTabela) > 0) then
  begin
    sWhere := ReturnWhere( sTabela );
    sConsulta := 'SELECT ' + sCampos + 'FROM ' + sTabela + sWhere;
  end
  else
    sConsulta := 'SELECT ' + sCampos + 'FROM ' + sTabela;

  ExecutionSQL(sConsulta);
end
else
  Application.MessageBox('Error Syntax', PChar(Application.Title),
MB_ICONERROR);

```

Listagem 5 – Mapeamento da consulta com os caracteres em álgebra relacional

A estrutura apresentada na Listagem 5 consiste em mapear a consulta de com base no posicionamento dos caracteres de acordo com suas posições padrões para a conversão em SQL, para posteriormente serem executadas no banco de dados selecionado.

Para o cadastro das consultas foi implementado o uso de componentes FIREDAC disponibilizados de forma nativa a partir da versão XE5 do Delphi. Na composição dos cadastros existe um cadastro principal do qual são herdados os demais cadastros, como o de exercícios, respostas, turmas e usuários. No código da Listagem 6 está o tratamento de operações com o procedimento SetOperacao.

```

procedure TfrmPadraoDB.SetOperacao(const Value: TOperacoes);
begin
  FOperacao := Value;
  //Habilita botões
  btnNovo.Enabled := (FOperacao in [topCancelar, topExcluir]);
  btnAlterar.Enabled := btnNovo.Enabled;
  btnExcluir.Enabled := (FOperacao = topCancelar);
  grbConsulta.Enabled := btnExcluir.Enabled;
  btnSalvar.Enabled := (FOperacao in [topAlterar, topIncluir]);
  btnCancelar.Enabled := btnSalvar.Enabled;
  grbManutencao.Enabled := btnSalvar.Enabled;
  //Ajusta o caption do formulário
  if Pos(' - [' , Self.Caption) > 0 then
    Self.Caption := Copy(Self.Caption,1,Pos(' - [' , Self.Caption)-1);
  case FOperacao of
    topIncluir: Self.Caption := Self.Caption + ' - [Inclusão]';
    topAlterar: Self.Caption := Self.Caption + ' - [Alteração]';
    topExcluir: Self.Caption := Self.Caption + ' - [Exclusão]';
  end;
end;
end;

```

Listagem 6 – Código para tratamento de operações com o procedimento SetOperacao

O procedimento apresentado na Listagem 6 setOperação ajusta os componentes do formulário conforme a operação que está sendo realizada, desabilitando botões e apresentando mensagens sobre qual operação está sendo realizada.

5 CONCLUSÃO

A necessidade da implementação de uma ferramenta para auxiliar o aprendizado de Álgebra Relacional é justificada pela análise realizada de aplicativos existentes para essa funcionalidade e por conversas com professores e alunos de disciplinas relacionadas ao aprendizado de banco de dados. As ferramentas analisadas, *SimAlg*, *DBTools 2000*, *EnsinAR* e Sistema de Aprendizagem em AR, não apresentam algumas características consideradas importantes como ferramenta didática. A ferramenta que foi desenvolvida como resultado da modelagem apresentada neste trabalho visa apresentar essas características, como definido na modelagem apresentada no Capítulo 4 deste texto.

Como resultado deste trabalho foi desenvolvido um aplicativo para auxílio na aprendizagem de conceitos de Álgebra Relacional. O sistema foi implementado visando apresentar uma interface gráfica contendo simbologia e sintaxe tradicional, não necessitar de instalação e disponibilizar um ambiente de auxílio para aplicação de tarefas e avaliações propostas em sala de aula. Espera-se que esses diferenciais sirvam de incentivo para a adesão de alunos e professores para o uso da ferramenta em disciplinas de banco de dados que trabalhem conteúdo de Álgebra Relacional.

Sugere-se, para trabalhos futuros, a implementação de cálculo relacional para geração de consultas. Outra melhoria a ser realizada é a sincronização de dados entre o servidor e a aplicação onde estão contidos os cadastros de alunos e professores. Com esse recurso os usuários professores terão acesso melhorado a construção de avaliações e vinculações de alunos as avaliações.

E também para trabalhos futuros sugere-se a implementação de técnicas de Inteligência Artificial, como lógica fuzzy, para auxiliar na correção das consultas e composição de listas em decorrência dos erros e/ou acertos nas respostas. Deste modo, a ferramenta não será de uso restrito a usuários que saibam manipular álgebra relacional, e sim, usuários que estão iniciando no aprendizado da linguagem.

REFERÊNCIAS

CODD, Edgar. F. **Relational completeness of data base sublanguages**, 1972. Disponível em: <<http://www.iai.uni-bonn.de/III/lehre/vorlesungen/Informationssysteme/WS05/materialien/Codd72a.pdf>>. Acesso em: 04 abr. 2015.

DAMAS, Luís. **SQL: structured query language**. Rio de Janeiro: LTC, 2007.

DATE, Christopher J. **Bancos de dados: fundamentos**. Rio de Janeiro: Câmpus, 1985.

DATE, Christopher J. **Introdução a sistemas de banco de dados.**, Rio de Janeiro: Câmpus, 2000.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6. ed. São Paulo, SP: Pearson Addison-Wesley, 2011.

KINAIPP, Renato V. **Sistema para apoio a aprendizagem de álgebra relacional**. Monografia (trabalho de conclusão de curso de graduação). Ciência da Computação. Universidade do Vale do Itajaí. 2009. Disponível em: <<http://siaibib01.univali.br/pdf/Renato%20Vidal%20Kinaipp.pdf>>. Acesso em: 20 mai. 2015.

KINAIPP, Renato V.; ALVES, Adriana G. **Instrumento pedagógico para apoio a aprendizagem de álgebra relacional**. v. 9, n. 1, p. 41-52, Nov. 2010. Disponível em: <<http://www.pgss.com.br/revistacientifica/index.php/exatas/article/view/1146>>. Acesso em: 06 abr. 2015.

LAUTER, Larissa R. **Implementação de um simulador de consultas em álgebra relacional**. Monografia (trabalho de conclusão de curso). Curso de Ciência da Computação. Universidade Federal de Santa Maria, 2010.

MACORATTI, José Carlos. **SQL - álgebra relacional - operações fundamentais - conceitos básicos**. Disponível em: <http://www.macoratti.net/13/06/sql_arcb.htm>. Acesso em: 07 mar. 2015.

OLIVEIRA, Celso H. P de. **SQL: curso prático**. São Paulo: Novatec, 2002.

PAES, Ederson Luiz. **EnsinAR: ferramenta didática para o ensino de álgebra relacional**. Universidade Federal de Santa Catarina - Centro Tecnológico. Departamento de Informática e Estatística. Florianópolis. 2004. p. 1-14. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_295/ArtigoTCCEderson.pdf>. Acesso em: 04 mar. 2015.

REZENDE, Ricardo. **Álgebra relacional – parte I**. DevMedia. Disponível em: <<http://www.devmedia.com.br/algebra-relacional-parte-i/2663>>. Acesso em: 07 mar. 2015.

REZENDE, Ricardo. **Linguagem de consulta formal - álgebra relacional – parte II**. 2012. Disponível em: <<http://www.devmedia.com.br/linguagem-de-consulta-formal-algebra-relacional-parte-ii/20123>>. Acesso em: 10 abr. 2015.

TAKAI, Osvaldo K. **Projeto lógico de BD relacional: notas didáticas**. Disponível em: <http://www.ifpiparnaiba.edu.br/index.php?option=com_docman%26task%3Ddoc_download%26gid%3D256%26Itemid%3D79&ei=99wfVfbQKsOpNqCdgYgl&usg=AFQjCNGneuHWqpl8o1cVCFhzRAjWq2EtQQ&sig2=s72lZd1HMHfiySSLqbkYA&bv=bv.89947451,d.eXY>. Acesso em: 4 abr. 2015.