

Universidade Tecnológica Federal do Paraná  
Campus Cornélio Procópio  
Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas

Gabriel Guimarães da Silva

**LearnAlgTool – SISTEMA PARA AUXÍLIO DE APRENDIZAGEM DE  
ÁLGEBRA RELACIONAL**

TRABALHO DE CONCLUSÃO DE CURSO

Cornélio Procópio  
2014

Gabriel Guimarães da Silva

## **LearnAlgTool – SISTEMA PARA AUXÍLIO DE APRENDIZAGEM DE ÁLGEBRA RELACIONAL**

Trabalho de conclusão de curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Ms. Adriano Rivolli da Silva

Cornélio Procópio

2014

Dedico este trabalho a minha família, especialmente a meu pai Márion e minha mãe Aparecida, que sempre estiveram ao meu lado, me apoiando e dando forças, que me ensinaram a ser a pessoa que sou hoje, e nunca deixaram que nada me faltasse. Aos meus amigos e a todos aqueles que de alguma forma me ajudaram nesse longo caminho que me trouxe até aqui.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, pelo dom da vida, pela saúde e pelas pessoas que tive o prazer de conhecer, e por ter me dado a capacidade de adquirir tamanha sabedoria que me capacitou para o desenvolvimento deste trabalho.

A minha família e minha namorada, por todo o apoio, toda força e toda ajuda para superar todos os obstáculos que apareceram durante essa dura caminhada, e principalmente por entenderem os motivos de minhas faltas, minhas reclamações e minhas brigas.

Aos professores, principalmente aqueles que fizeram parte do meu curso, e tanto se esforçaram para me passar um pouco de seus conhecimentos. Agradeço em especial ao professor Adriano Rivolli da Silva, por seu enorme auxílio, preocupação, paciência e cobrança, e principalmente por acreditar em mim e sempre me incentivar, desde o momento em que aceitou participar deste trabalho.

Aos amigos de longa data, e os amigos que tive o prazer de conhecer durante o curso, todos sempre dispostos a contribuir e ajudar da forma que sabiam e podiam, sem eles todos os desafios seriam mais difíceis.

E por último, e nem por isso menos importante, agradeço aos professores da banca examinadora pela atenção e contribuição dedicadas a este trabalho.

Programadores são ferramentas para  
converter cafeína em código.

(Autor desconhecido).

Eu sou o caminho, a verdade e a vida.  
Ninguém vai ao pai se não por mim.

(Jesus Cristo).

## RESUMO

SILVA, Gabriel Guimarães da. LearnAlgTool: Sistema para auxílio de aprendizagem de álgebra relacional. 2014. 81 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

O presente documento tem por objetivo apresentar o desenvolvimento do sistema LearnAlgTool, cuja função é auxiliar no aprendizado das operações fundamentais da álgebra relacional. A álgebra relacional é uma linguagem de consulta procedural, que consiste em um conjunto de operações tendo como entrada uma ou duas relações e produzindo, como resultado, uma nova relação. A álgebra relacional provê um fundamento formal para operações no modelo relacional, e é utilizada como base para implementar e otimizar as consultas em sistemas de gerenciadores de bancos de dados relacional. A dificuldade em encontrar ferramentas que interpretem consultas utilizando álgebra relacional dificulta o aprendizado dos alunos nos cursos de computação que possuem disciplinas de banco de dados. Duas ferramentas encontradas foram analisadas, e os aspectos positivos e negativos de cada *software* ajudaram a tomar decisões sobre o que ainda poderia ser feito. O presente trabalho documenta a construção de um sistema que simula consultas utilizando a sintaxe da álgebra relacional, para auxiliar no aprendizado das operações fundamentais. O sistema foi desenvolvido, tendo como base, suprir as carências dos *softwares* existentes.

**Palavras-chave:** Álgebra relacional, Consulta, Modelo relacional, Aprendizado.

## ABSTRACT

SILVA, Gabriel Guimarães da. LearnAlgTool: System to aid the learning of relational algebra. 2014. 81 f. Course Conclusion Paper (Technology in Analysis and Systems Development) - Federal University of Technology from Parana. Cornélio Procópio, 2014.

This paper aims to present the development of LearnAlgTool system whose function is to assist in learning the basic operations of relational algebra. The relational algebra is a procedural query language, consisting of a set of operations having as input one or two relations and producing, as a result, a new relation. The relational algebra provides a formal foundation for operations in the relational model, and is used as a basis to implement and optimize queries in systems of managers of relational database. The difficulty in finding tools that interpret queries using relational algebra hinders student learning in undergraduate computing course that have database disciplines. Two tools found were analyzed, and the positive and negative aspects of each software helped to make decisions about what could be done. This paper documents the construction of a system that simulates queries using the syntax of the relational algebra, to assist in learning basic operations. The system was developed on the bases of supplying the needs of the existing software.

**Keywords:** Relational Algebra, Query, Relational Model, Learning.

## LISTA DE ILUSTRAÇÕES

FIGURA 1 - INTERFACE DO SISTEMA RELACIONAL .....	24
FIGURA 2 - INTERFACE DO SISTEMA LEAP .....	25
FIGURA 3 - MODELO INCREMENTAL .....	28
FIGURA 4 - MVC (MODELO, VISUALIZAÇÃO, CONTROLE).....	29
FIGURA 5 - CASOS DE USOS .....	30
FIGURA 6 - DIAGRAMA DE CLASSES .....	33
FIGURA 7 - MODELO RELACIONAL .....	35
FIGURA 8 - TELAS DE LOGON E REGISTRO .....	37
FIGURA 9 - TELA DE ATIVIDADES.....	38
FIGURA 10 – RESOLUÇÃO DE ATIVIDADES .....	38
FIGURA 11 – RELATÓRIOS DO SISTEMA.....	39



## LISTA DE TABELAS

TABELA 1 - RELAÇÃO EMPRÉSTIMO .....	18
TABELA 2 - RELAÇÃO DEVEDOR .....	18
TABELA 3 - RESULTADO DE $\Sigma$ NOME_AGENCIA = "PERRYRIDGE" (MPRÉSTIMO) .....	19
TABELA 4 - RESULTADO DE $\Pi$ NUMERO_EMPRESTIMO, TOTAL (EMPRESTIMO) .....	20
TABELA 5 - RESULTADO DE EMPRESTIMOX DEVEDOR .....	21
TABELA 6 - RELAÇÃO DEPOSITANTE .....	21
TABELA 7 - RESULTADO DE $\Pi$ NOME_CLIENTE (DEVEDOR) U $\Pi$ NOME_CLIENTE (DEPOSITANTE) .....	22
TABELA 8 - RESULTADO DE $\Pi$ NOME_CLIENTE (DEPOSITANTE) - $\Pi$ NOME_CLIENTE (DEVEDOR) .....	23
TABELA 9 - COMPARAÇÃO ENTRE OS SOFTWARES .....	25
TABELA 10 - FERRAMENTAS E TECNOLOGIAS .....	26
TABELA 11 - DESCRIÇÃO CASOS DE USOS GERAL .....	31
TABELA 12 - DESCRIÇÃO CASO DE USO GERENCIAR USUÁRIOS .....	31
TABELA 12 - DESCRIÇÃO CASO DE USO GERENCIAR USUÁRIOS .....	32
TABELA 13 - VALIDAÇÕES E TESTES .....	40
TABELA 14 - CRONOGRAMA FINAL.....	42
TABELA 15 - DESCRIÇÃO CASO DE USO GERENCIAR GRUPOS .....	47
TABELA 15 - DESCRIÇÃO CASO DE USO GERENCIAR GRUPOS .....	48
TABELA 16 - DESCRIÇÃO CASO DE USO GERENCIAR ESQUEMAS RELACIONAIS .....	48
TABELA 16 - DESCRIÇÃO CASO DE USO GERENCIAR ESQUEMAS RELACIONAIS .....	49
TABELA 17 - DESCRIÇÃO CASO DE USO GERENCIAR ATIVIDADES .....	49
TABELA 17 - DESCRIÇÃO CASO DE USO GERENCIAR ATIVIDADES .....	50
TABELA 18 - DESCRIÇÃO CASO DE USO SOLUCIONAR ATIVIDADES.....	50
TABELA 18 - DESCRIÇÃO CASO DE USO SOLUCIONAR ATIVIDADES.....	51
TABELA 19 - DESCRIÇÃO CASO DE USO ARMAZENAR HISTÓRICO DE ATIVIDADES .....	51
TABELA 19 - DESCRIÇÃO CASO DE USO ARMAZENAR HISTÓRICO DE ATIVIDADES .....	52
TABELA 20 - DESCRIÇÃO CASO DE USO ANALISAR ATIVIDADES EXECUTADAS .....	52
TABELA 20 - DESCRIÇÃO CASO DE USO ANALISAR ATIVIDADES EXECUTADAS .....	52
TABELA 20 - DESCRIÇÃO CASO DE USO ANALISAR ATIVIDADES EXECUTADAS .....	52

## LISTA DE SIGLAS E ABREVIATURAS

CSS – *Cascading Style Sheets* (Folhas de Estilo em Cascata).

HTML – *HyperText Markup Language* (Linguagem de Marcação de Hipertexto).

IDE– *Integrated Development Environment* (Ambiente Integrado de Desenvolvimento).

MVC – *Model, View, Controller* (Modelo, Visualização e Controle).

SQL – *Structured Query Language* (Linguagem de Consulta Estruturada).

SGBD – Sistema Gerenciador de Banco de Dados.

UML – *Unified Modeling Language* (Linguagem de Modelagem Unificada).

## LISTA DE SÍMBOLOS

- $\pi$  – Operador de projeção em álgebra relacional (Pi).
- $\sigma$  – Operador de seleção em álgebra relacional (Sigma).
- $\times$  – Produto cartesiano em álgebra relacional.
- $\rho$  – Renomear em álgebra relacional (Rho).
- $\cup$  – União em álgebra relacional.

## SUMÁRIO

1. INTRODUÇÃO.....	13
1.1. OBJETIVOS .....	14
1.1.1. Objetivos gerais.....	14
1.1.2. Objetivos específicos.....	14
1.2. PROBLEMAS E JUSTIFICATIVA .....	14
1.3. ORGANIZAÇÃO DO TEXTO .....	15
2. FUNDAMENTAÇÃO TEÓRICA .....	16
2.1. MODELO RELACIONAL.....	16
2.2. ÁLGEBRA RELACIONAL.....	18
2.3. TRABALHOS RELACIONADOS .....	23
2.3.1. Relational.....	23
2.3.2. LEAP .....	24
2.3.3. LearnAlgTool e os softwares relacionados .....	25
3. METODOLOGIA .....	26
3.1. FERRAMENTAS E TECNOLOGIAS .....	26
3.2. PROCESSO DE DESENVOLVIMENTO.....	27
3.3. ARQUITETURA DO SISTEMA .....	28
4. RESULTADOS .....	30
4.1. CASOS DE USOS.....	30
4.2. DIAGRAMA DE CLASSES.....	32
4.3. MODELO LÓGICO RELACIONAL .....	34
4.4. INTERFACES GRÁFICAS DO SISTEMA .....	36
4.5. VALIDAÇÕES E TESTES.....	39
5. CRONOGRAMA .....	41
6. CONSIDERAÇÕES FINAIS.....	43
6.1. DIFICULDADES ENCONTRADAS.....	43
6.2. TRABALHOS FUTUROS .....	44
7. REFERÊNCIAS .....	45
ANEXO A – Diagrama de casos de usos e especificações.....	47
ANEXO B – Proposta do trabalho de Diplomação .....	57

## 1. INTRODUÇÃO

Um sistema de gerenciamento de banco de dados (SGBD) pode ser definido com um conjunto de dados, associado a um conjunto de programas que permita gerenciar tais dados. Um SGBD busca proporcionar um ambiente mais amigável e eficiente para gerenciamento das informações armazenadas em um banco de dados.

Os sistemas de banco de dados são projetados para gerir um grande volume de informações, definindo as estruturas de armazenamento dessas informações e os mecanismos para sua manipulação, além de serem os responsáveis por garantir a segurança das informações armazenadas contra tentativas de acesso não autorizadas, anomalia de informações, ou contra eventuais problemas com o sistema.

Historicamente, o modelo relacional estabeleceu-se como o primeiro modelo de dados para aplicações comerciais (KORTH, 1997). O modelo de dados é definido como um conjunto de ferramentas conceituais usadas para a descrição de dados e o relacionamento entre eles, e para definir suas regras de consistência.

O modelo relacional representa os dados e a relação entre eles utilizando um conjunto de relações, onde cada relação possui um ou vários atributos, e cada atributo possui um nome único. Cada relação possui um atributo (ou atributos) chamado chave primária que permite identificar os conjuntos de valores presentes. Para se efetuar uma consulta, utiliza-se uma linguagem de consulta, processada pelo SGBD.

O conjunto básico de operações para o modelo relacional é a álgebra relacional, uma linguagem de consulta procedural, que consiste em um conjunto de operações tendo como entrada uma ou duas relações e produzindo, como resultado, uma nova relação. A álgebra relacional é utilizada como base para implementar e otimizar as consultas em sistemas de gerenciadores de bancos de dados relacional.

## **1.1.OBJETIVOS**

### **1.1.1. Objetivos gerais**

O objetivo geral deste trabalho consiste em documentar e apresentar o desenvolvimento de uma aplicação web, denominada LearnAlgTool, que permite realizar operações de consultas em bancos de dados relacionais utilizando a sintaxe de álgebra relacional. Espera-se que os professores possam gerar as relações e dados, propondo questionários de consultas, e que os alunos possam resolver as atividades utilizando os conceitos de álgebra relacional, permitindo que os professores possam controlar a evolução e desempenho dos alunos, e estes passam a obter resposta imediata para seus exercícios.

### **1.1.2. Objetivos específicos**

- Analisar os softwares existentes, que permitam a realização de consulta utilizando álgebra relacional.
- Estudar e conhecer as operações de álgebra relacional.
- Fornecer uma ferramenta prática para a elaboração de consultas utilizando álgebra relacional
- Permitir que os professores de banco de dados possam acompanhar o desempenho de seus alunos, quanto aos conceitos de álgebra relacional

## **1.2. PROBLEMAS E JUSTIFICATIVA**

O estudo e a compreensão do modelo de dados relacional e de álgebra relacional são fundamentais para estudantes dos cursos de computação que contenham disciplinas de banco de dados. A álgebra relacional, por se tratar de uma matéria teórica, apresenta na falta de atividades práticas a principal dificuldade para sua compreensão e aprendizado, tornando-os mais complicados e demorados. Uma forma de facilitar seu aprendizado é proporcionar um ambiente para testes e consultas, utilizando a álgebra relacional.

Com a crescente utilização de banco de dados relacionais, a necessidade de pessoas que gerenciem essa base de dados cresce em mesma proporção, o ensino de banco de dados relacionais e álgebra relacional se tornam mais fundamental, e devido ao curto tempo disponível para aprendizagem, ferramentas que facilitem o aprendizado e diminuam a curva de tempo são de extrema importância.

A dificuldade em encontrar ferramentas relacionadas, e os problemas existentes nas poucas ferramentas encontradas, motivaram o desenvolvimento desse sistema.

### **1.3. ORGANIZAÇÃO DO TEXTO**

No capítulo 2 é apresentada uma breve revisão dos conceitos sobre álgebra relacional, principais formas de utilização, as vantagens do seu uso, alguns exemplos práticos e, por fim, uma comparação entre os sistemas relacionados encontrados.

No capítulo 3 é apresentada toda a metodologia de desenvolvimento do sistema, apresentando e descrevendo as principais tecnologias e ferramentas utilizadas, o processo de desenvolvimento seguido, e o padrão que o sistema foi arquitetado.

No capítulo 4 são apresentados os resultados obtidos com o desenvolvimento do sistema, apresentando e descrevendo os principais diagramas gerados, a modelagem final das relações no banco de dados, as principais telas do sistema e suas principais funcionalidades, e o processo de validações e testes do sistema.

No capítulo 5 é apresentado o cronograma final do projeto, apresentando e descrevendo como as fases foram divididas, e as datas de início e encerramento de cada fase.

No capítulo 6 são apresentadas as considerações finais, as dificuldades encontradas e as expectativas para trabalhos futuros.

## 2. FUNDAMENTAÇÃO TEÓRICA

Esse capítulo destina-se a apresentar, segundo a terceira edição do livro *Sistemas de Bancos de Dados*, de Abraham Silberschatz, à definição do modelo relacional e da álgebra relacional, além de suas formas mais comuns de utilização, e alguns exemplos utilizando seus operadores fundamentais.

### 2.1. MODELO RELACIONAL

O modelo relacional teve sua introdução na década de 70 após um artigo clássico (*Relational Model of Data for Large Shared Data Banks*, 1970), publicado por Ted Codd, da IBM Research. O modelo usa o conceito de uma relação matemática como seu bloco de construção básica e tem sua base teórica na teoria dos conjuntos e na lógica de predicados de primeira ordem.

O modelo relacional estabeleceu-se como primeiro modelo de dados para aplicações comerciais.

Um banco de dados relacional é constituído de uma coleção de tabela, cada uma com um nome único. Uma linha em uma tabela representa um relacionamento entre um conjunto de valores. Uma vez que essa tabela é uma coleção de tais relacionamentos, há uma estreita correspondência entre o conceito de tabela e o conceito matemático de relação, a partir dos quais se origina o nome desse modelo de dados.

Na terminologia do modelo relacional, os nomes das colunas são conhecidos como atributos e para cada atributo existe um conjunto de valores permitido, chamado domínio do atributo em questão.

Uma relação é matematicamente definida como um subconjunto de um produto cartesiano de uma lista de atributos, o que corresponde a quase exatamente a definição de uma tabela. A única diferença é que designamos nomes aos atributos, enquanto, matematicamente, usam-se apenas números inteiros, utilizando 1 para denotar o atributo cujo domínio aparece em primeiro lugar na lista dos domínios, o 2 para denotar o segundo atributo e assim sucessivamente. Como as tabelas em essência são relações, utilizamos os termos matemáticos relação e tupla no lugar de tabela e linha.

É possível, que alguns atributos, possuam o mesmo domínio, e o valor de domínio que pertença a qualquer domínio possível é o valor nulo, que indica que o valor é desconhecido ou não existe.



O conceito de chave nos permite especificar como as entidades dentro de um conjunto de entidades e os relacionamentos dentro de um conjunto de relacionamentos podem ser identificados. As noções de super chave, chave candidata e chave primária são aplicadas no modelo relacional.

Uma super chave é um conjunto de um ou mais atributos, que tomados coletivamente, servem para distinguir uma tupla das demais, em uma relação. Uma chave candidata é uma super chave da qual nenhum subconjunto possa ser uma super chave. É possível que vários conjuntos diferentes de atributos possam servir como super chave. O termo chave primária é usado para caracterizar a chave candidata escolhida para a identificação única das tuplas dentro de uma relação, geralmente a chave candidata de menor tamanho.

Quando se incluiu, entre os atributos de uma relação, uma chave primária de outra relação, essa chave é chamada de chave estrangeira. A chave estrangeira é o mecanismo que permite a implementação de relacionamentos em um banco de dados relacional. A utilização de chave estrangeira impõe restrições, que devem ser garantidas, ao executar diversas operações no banco de dados relacional, visto que ela referencia a chave primária de alguma relação. Caso seja necessário uma chave estrangeira pode referenciar a chave primária da própria relação.

Por meio de uma linguagem de consulta, os usuários podem obter informações do banco de dados. Essas linguagens são, tipicamente, de nível mais alto que as linguagens de programação tradicionais, e podem ser caracterizadas em procedurais e não procedurais.

Em uma linguagem não procedural, o usuário especifica a informação que deseja obter, sem fornecer um procedimento específico para a obtenção dessas informações, já em uma linguagem procedural, o usuário deve especificar ao sistema uma sequência de operações no banco de dados para se obter o resultado desejado.

O conjunto básico de operações para o modelo relacional é a álgebra relacional. A álgebra relacional provê um fundamento formal para operações no modelo relacional, e é utilizada como base para implementar e otimizar as consultas em sistemas de gerenciadores de bancos de dados relacional.

## 2.2. ÁLGEBRA RELACIONAL

A álgebra relacional é uma linguagem de consultas procedural, que consiste em um conjunto de operações que recebe como entrada uma ou duas relações, e produz, como resultado, uma nova relação. A álgebra relacional é composta por seis operações fundamentais, que são: seleção, projeção, união, diferença, produto cartesiano e renomear. Além das operações fundamentais, existem outras operações - adicionais e estendidas – que não serão descritas neste capítulo.

As operações seleção, projeção e renomear operam uma única relação, e por isso são denominadas como operações primárias. As outras operações fundamentais são chamadas de operações binárias, pois operam um par de relações.

Consideremos a relação empréstimo, conforme mostrada na Tabela 1, e a relação devedor, conforme mostrado na Tabela 2.

**Tabela 1 - Relação Empréstimo**

nome_agencia	numero_emprestimo	Total
Downtown	L-17	1000
Redwood	L-23	2000
Perryridge	L-15	1500
Downtown	L-14	1500
Mianus	L-93	500
Round Hill	L-11	900
Perryridge	L-16	300

**Fonte: Silberschatz, 1997, p. 66.**

**Tabela 2 - Relação Devedor**

nome_cliente	numero_emprestimo
Jones	L-17
Smith	L-23
Hayes	L-15
Jackson	L-14
Curry	L-93
Smith	L-11
Williams	L-17
Adams	L-16

**Fonte: Silberschatz, 1997, p. 66.**

A operação de seleção, como o nome sugere, seleciona tuplas que satisfaçam um determinado predicado. Ela é denotada pela letra grega minúscula sigma ( $\sigma$ ). O predicado aparece subscrito a letra que denomina a operação, e o argumento da relação é dado entre parênteses, seguindo a mesma letra. Em geral podemos usar comparações do tipo =,  $\neq$ , <,  $\leq$ , > e  $\geq$  no predicado das operações. Além do mais, podemos combinar vários predicados, em um único predicado, utilizando os conectivos e ( $\wedge$ ) e ou ( $\vee$ ). A seleção pode ainda comparar dois ou mais atributos.

Para selecionar as tuplas, da relação empréstimo, cuja agência é “Perryridge”, escrevemos:

$$\sigma_{\text{nome\_agencia} = \text{“Perryridge”}}(\text{emprestimo})$$

Se a relação empréstimo é apresentada na Tabela 1, então a relação resultante da consulta de seleção é mostrada na Tabela 3.

**Tabela 3 - Resultado de  $\sigma_{\text{nome\_agencia} = \text{“Perryridge”}}$  (empréstimo)**

nome_agencia	numero_emprestimo	Total
Perryridge	L-15	1500
Perryridge	L-16	300

Fonte: Silberschatz, 1997, p. 69.

A operação projeção, denotada pela letra grega pi ( $\pi$ ), retorna o argumento da relação, descartando certos atributos. Na operação de projeção, quaisquer linhas em duplicidade são eliminadas. Listamos, subscritos a letra pi, os atributos que desejamos no resultado, e os atributos que não estão listados são descartados pela operação. O argumento da relação vem entre parênteses, a seguir.

Suponha que desejamos listar todos os números de empréstimos e todos os totais correspondentes, porém o nome das agências envolvidas não interessa. Assim a consulta pode ser escrita da seguinte forma:

$$\pi_{\text{numero\_emprestimo, total}}(\text{emprestimo})$$

Considerando a relação empréstimo é apresentada na Tabela 1, então a relação resultante da projeção é mostrada na Tabela 4.

Tabela 4 - Resultado de  $\pi_{\text{numero\_emprestimo, total}}(\text{emprestimo})$ 

numero_emprestimo	Total
L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-93	500
L-11	900
L-16	300

Fonte: Silberschatz, 1997, p. 70.

A operação renomear, denotada pela letra grega rho ( $\rho$ ), permite-nos dar um nome ao resultado de uma expressão em álgebra relacional, visto que o resultado de uma expressão em álgebra relacional não possui um nome que nos permita referenciá-la. Podemos também aplicar a operação renomear à uma relação já existente, para obter a mesma relação sobre um novo nome.

Dada a expressão em álgebra relacional E, a expressão:

$$\rho_x(E)$$

tem por resultado a expressão E sob o nome x.

Assumindo que a expressão E em álgebra relacional seja de ordem primária, se torna possível alterar os nomes dos atributos de uma relação utilizando a operação renomear:

$$\rho_{x(\text{atributo}_1, \text{atributo}_2)}(E)$$

A operação produto cartesiano, denotada por um ( $\times$ ), permite que combinemos as informações de duas relações quaisquer. O produto das relações r1 e r2 é representado por r1xr2. Esta operação constrói uma nova relação a partir das duas relações especificadas. A nova relação que será construída é constituída de todas as possibilidades de pares de tuplas concatenadas, de cada uma das relações.

Suponhamos que desejamos utilizar a operação de produto cartesiano, com as relações empréstimo e devedor. Assim a consulta pode ser escrita dessa forma:

$$\text{emprestimo} \times \text{devedor}$$

Considerando a relação empréstimo é apresentada na Tabela 1, e a relação devedor na Tabela 2, então a relação resultante da consulta de produto cartesiano é mostrada na Tabela 5.

**Tabela 5 - Resultado de empréstimos devedor**

emprestimo.nome_ agencia	emprestimo.numero_ emprestimo	emprestimo.t otal	devedor.nome_ cliente	devedor.numero_ emprestimo
Downtown	L-17	1000	Jones	L-17
...	...	...	...	...
Downtown	L-17	1000	Jackson	L-14
Downtown	L-17	1000	Curry	L-93
...	...	...	...	...
Downtown	L-17	1000	Smith	L-11
Redwood	L-23	2000	Smith	L-11
...	...	...	...	...
Downtown	L-17	1000	Williams	L-17
Redwood	L-23	2000	Williams	L-17
...	...	...	...	...
...	....	...	....	....
Mianus	L-93	500	Adams	L-16
Round Hill	L-11	900	Adams	L-16
Perryridge	L-16	300	Adams	L-16

Fonte: Silberschatz, 1997, p. 74.

A operação de união, denotada por (U), constrói uma nova relação consistindo em todas as tuplas que aparecem em ambas as relações especificadas. Para uma operação de união r U s ser válida são necessárias duas condições: As relações devem possuir o mesmo número de atributos e os domínios do i-ésimo atributo de r e o i-ésimo atributo de s devem ser os mesmos para todo i.

Suponhamos que, agora também possuímos a relação depositante, conforme mostrado na Tabela 6.

**Tabela 6 - Relação Depositante**

nome_cliente	numero_conta
Johnson	A-101
Smith	A-215
Hayes	A-102
Turner	A-305
Johnson	A-201
Jones	A-217
Lindsay	A-222

Fonte: Silberschatz, 1997, p. 70.

Considere que desejamos encontrar os nomes dos clientes do banco que possuam uma conta, um empréstimo, ou possuam ambos. Para encontrar os nomes dos clientes que possuem um empréstimo, devemos projetar o atributo nome\_cliente, da relação devedor, e para encontrar os nomes dos clientes que possuem uma conta no banco devemos projetar o atributo nome\_cliente da relação depositante.

Para encontrarmos os nomes dos clientes que possuem tanto uma conta quanto um empréstimo, precisamos de uma união desses dois conjuntos, para projetarmos os nomes dos clientes que aparecem em ambas as relações, e como as relações são conjuntos, valores duplicados são eliminados.

A consulta pode ser expressa da seguinte forma:

$$\pi_{\text{nome\_cliente}}(\text{devedor}) \cup \pi_{\text{nome\_cliente}}(\text{depositante})$$

Considerando a relação devedor é apresentada na Tabela 2, e a relação depositante na Tabela 6, então a relação resultante da união é mostrada na Tabela 7.

**Tabela 7 - Resultado de  $\pi_{\text{nome\_cliente}}(\text{devedor}) \cup \pi_{\text{nome\_cliente}}(\text{depositante})$**

nome_cliente
Johnson
Smith
Hayes
Turner
Jones
Lindsay
Jackson
Curry
Williams
Adams

**Fonte: Silberschatz, 1997, p. 72.**

A operação diferença, denotada por (-), permite-nos encontrar as tuplas que estão em uma relação, mas não em outra. Esta operação constrói uma nova relação consistindo em todas as tuplas que aparecem na primeira, mas não na segunda, do par de relações especificadas.

Suponha que desejamos encontrar todos os clientes que do banco que possuem contas, mas não realizaram empréstimo. A nova relação com os dados desejados pode ser gerada utilizando a seguinte consulta:

$$\pi_{\text{nome\_cliente}}(\text{depositante}) - \pi_{\text{nome\_cliente}}(\text{devedor})$$

Se a relação devedor é apresentada na Tabela 2, e a relação depositante na Tabela 6, então a relação resultante da operação de diferença é mostrada na Tabela 8.

**Tabela 8 - Resultado de  $\pi_{\text{nome\_cliente}}(\text{depositante}) - \pi_{\text{nome\_cliente}}(\text{devedor})$**

nome_cliente
Johnson
Turner
Lindsay

Fonte: Silberschatz, 1997, p. 73.

## 2.3. TRABALHOS RELACIONADOS

Esta seção tem como objetivo apresentar uma breve análise sobre os softwares de consultas de álgebra relacional, encontrados. Foram analisados dois softwares, o primeiro, RELATIONAL (2007), desenvolvido pela *FREE SOFTWARE FOUNDATION, INC.* O segundo o LEAP, foi desenvolvido na Escócia. Os dois *softwares*, analisados apresentaram pontos positivos e negativos, quando analisados segundo suas funcionalidades e sua usabilidade.

### 2.3.1. Relational

O *software RELATIONAL* (2007) suporta todas as operações fundamentais da álgebra relacional, seleção, projeção, união, intersecção, produto cartesiano e renomeação, diferença, junção e junção natural. Além destas operações de consulta, o *RELATIONAL* permite criação, edição e remoção das relações.

A interface do programa é mostrada na Figura 1. Os operadores são separados em botões, e adotam a sintaxe e os símbolos da álgebra relacional, utilizada na maioria dos livros de bancos de dados. Através da interface também é possível acessar, criar e excluir novas relações. O *software* exibe os atributos das relações carregadas, e exibe um histórico com as consultas realizadas. Por fim, o sistema permite acesso a outras telas, que apresentam o histórico do programa, e uma ajuda, muito eficiente e explicativa, que ensina os conceitos básicos da álgebra e da utilização do *software*.

Apesar da interface bem elaborada, intuitiva e auto-explicativa, alguns de seus componentes não funcionam, ou não produzem o resultado desejado, como os botões de união, divisão, e junção natural, por exemplo. Toda a ferramenta também é produzida em inglês, o que faz com que usuários que não compreendam a língua, encontrem dificuldades em utilizá-la.

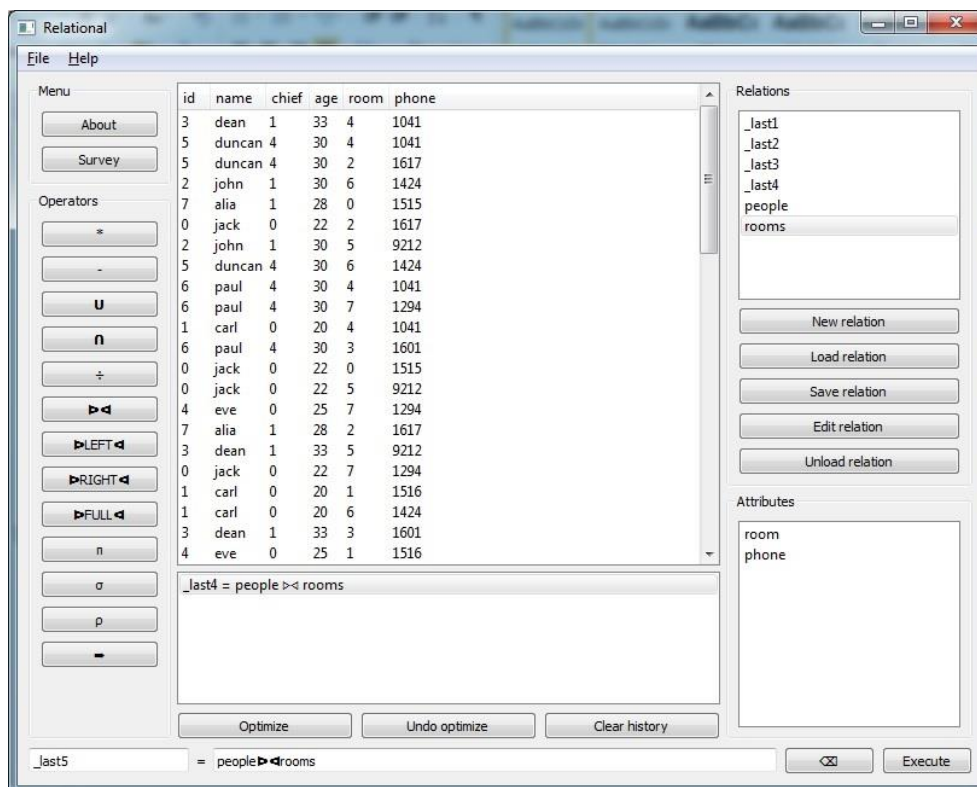


Figura 1 - Interface do sistema RELATIONAL

Fonte: RELATIONAL, 2007

### 2.3.2. LEAP

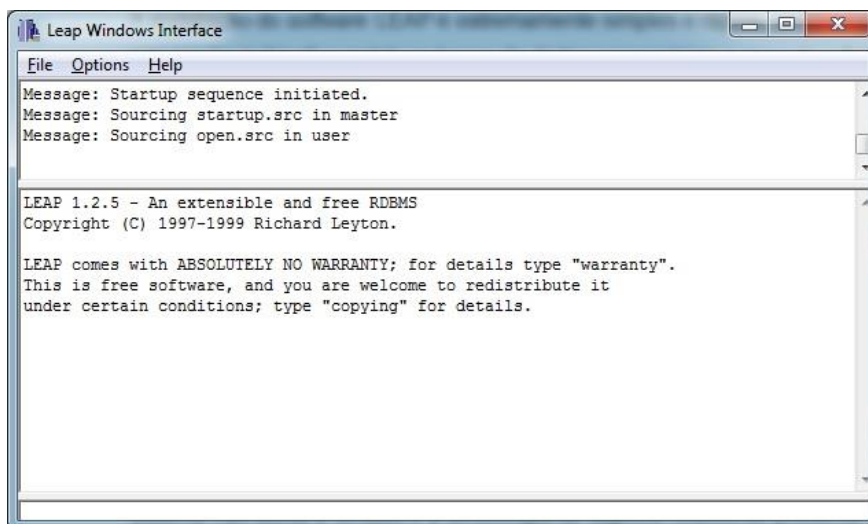
O *software LEAP* suporta as operações fundamentais da álgebra relacional, diferença, junção e junção natural. Além destas operações de consulta, ele permite também a criação, edição e remoção de tabelas e bancos de dados.

A interface do LEAP é totalmente textual, conforme ilustrado na Figura 2, o que faz com que o usuário seja obrigado a ler a documentação do sistema antes de utilizá-lo, para que se conheça qual o formato de entrada do *software*, e quais operações ele suporta. O sistema não adota a sintaxe e a simbologia da álgebra relacional, presente na maioria dos livros de banco de dados.



Porém, após a leitura da documentação a utilização da ferramenta é mais simples. O software possui comandos para listar o conteúdo de determinada tabela, listar as tabelas presentes no banco, realizar consultas em álgebra relacional, e alterações nas tabelas.

A ferramenta também permite ao usuário algumas alterações em suas funcionalidades, para que ele possa, por exemplo, acompanhar o que está acontecendo durante a consulta.



**Figura 2 - Interface do Sistema LEAP**

Fonte: LEAP, 1995.

### 2.3.3. LearnAlgTool e os softwares relacionados

Na Tabela 9 é apresentada uma comparação entre as principais características, desejáveis no software proposto e presente nos softwares Relational e LEAP.

**Tabela 9 - Comparação entre os softwares**

	Relational	LEAP	LearnAlgTool
Não necessita Instalação			X
Armazena Historico de Consultas	X	X	X
Simbologia e Sintaxe Comum	X		X
Suporte a Operações Básicas	X	X	X
Interface Clara e Intuitiva	X		X
Gerenciar Relações	X	X	X
Avaliação e Relatório das Consultas Realizadas			X

Fonte: Autoria própria.

### 3. METODOLOGIA

Esse capítulo destina-se a apresentar a metodologia de desenvolvimento do sistema LearnAlgTool. Serão apresentadas as tecnologias e ferramentas utilizadas, a especificação do processo de desenvolvimento seguido, e a forma que o sistema foi arquitetado.

#### 3.1.FERRAMENTAS E TECNOLOGIAS

Na Tabela 10, são apresentadas algumas das tecnologias e ferramentas que possibilitaram o desenvolvimento do sistema LearnAlgTool.

**Tabela 10 - Ferramentas e Tecnologias**

Tecnologia	Descrição
PHP	Linguagem interpretada que realizará o processamento no lado do servidor.
Kohana <sup>1</sup>	Framework PHP
HTML	Linguagem de marcação utilizada para produzir páginas web.
Java Script	Linguagem de programação interpretada.
CSS	Linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como por exemplo, o HTML.
UML	Linguagem visual utilizada para modelar <i>softwares</i> baseados no paradigma a orientação a objetos.
Astah	<i>Software</i> utilizado para geração de todos os diagramas da UML.
NetBeans	IDE de desenvolvimento.
MySQL	Servidor de banco de dados relacional.
Apache	Servidor Web

**Fonte: Autoria própria.**

---

<sup>1</sup> O Framework Kohana pode ser baixado diretamente de seu site oficial, disponível no seguinte link: <http://kohanaframework.org/>

### 3.2. PROCESSO DE DESENVOLVIMENTO

O processo de desenvolvimento seguido para a implementação desse projeto foi o modelo incremental (Figura 3 - **Modelo Incremental**).

O modelo incremental, busca combinar os elementos do modelo cascata de forma interativa, dividindo o processo de desenvolvimento em etapas (incrementos), e realizando todo o ciclo do desenvolvimento de software.

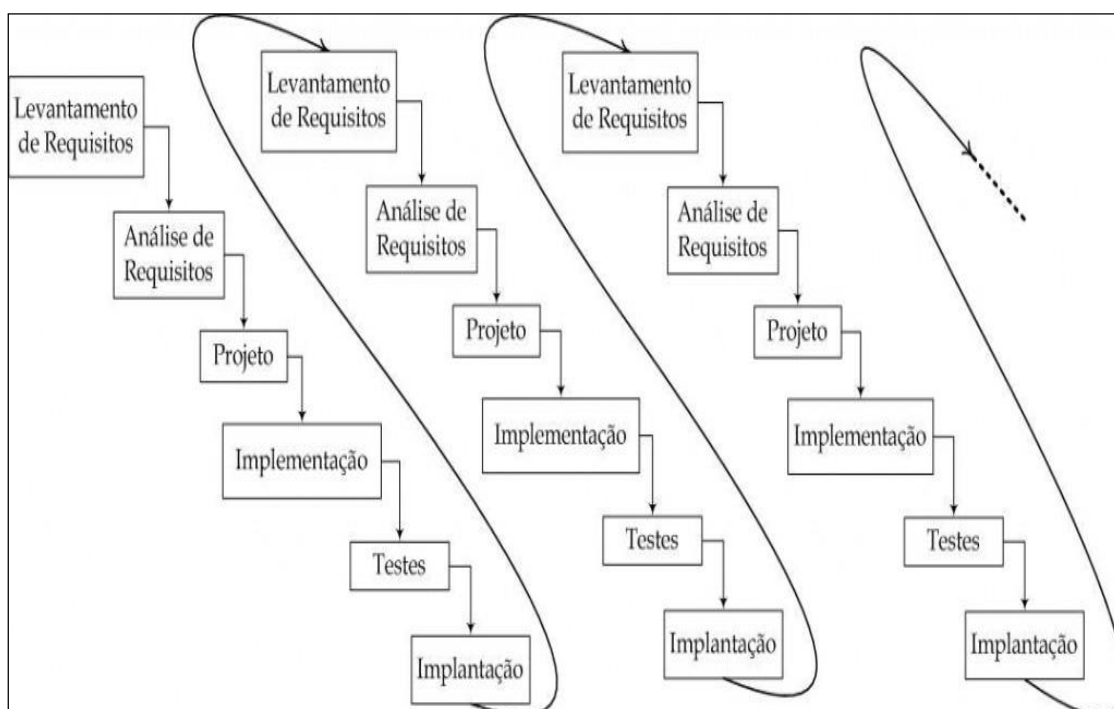
A construção de sistemas menores (incrementos) se torna menos arriscada que a construção de um sistema grande, pois caso algum erro seja cometido, apenas o último incremento é descartado.

O sistema LearnAlgTool é composto pela construção de oito incrementos, conforme descrito na seguinte lista;

- O primeiro incremento consiste nas atividades relacionadas as telas de login e cadastro dos usuários, tais telas permitem o acesso dos usuários as funcionalidades do sistema.
- O segundo incremento, visa permitir ao usuário de nível professor, gerenciar os grupos de alunos aos quais serão aplicadas as atividades.
- O terceiro incremento engloba todas as atividades referentes ao gerenciamento de atividades e itens de atividades que serão aplicados ao grupo de alunos.
- O quarto incremento engloba as atividades realizadas para possibilitar que os professores gerenciem as relações que serão utilizadas na aplicação das atividades.
- O quinto incremento engloba as atividades realizadas para que seja possível a correção das atividades, após serem propostas as soluções pelos alunos matriculados no grupo.
- O sexto incremento engloba as atividades necessárias para que seja possível a geração dos relatórios, após a correção das atividades propostas.
- O sétimo incremento, faz parte das atividades voltadas ao nível de acesso aluno, e busca possibilitar que os usuários desse nível de acesso possam visualizar e se matricular no grupo, e após efetuar sua

matricula, possam visualizar as atividades propostas ao grupo que se matriculou.

- O oitavo incremento consiste da parte mais importante do sistema, pois este incremento engloba todas as atividades necessárias para que os alunos possam executar a consulta e enviar a solução proposta para avaliação dos professores.
- Para concluir este projeto, podemos acrescentar uma nova fase, esta fase consiste na elaboração deste documento.



**Figura 3 - Modelo Incremental**

Fonte: [wiki.sj.ifsc.edu.br/wiki/index.php/Ciclo\\_de\\_Vida\\_Iterativo\\_e\\_Incremental](http://wiki.sj.ifsc.edu.br/wiki/index.php/Ciclo_de_Vida_Iterativo_e_Incremental)

### 3.3. ARQUITETURA DO SISTEMA

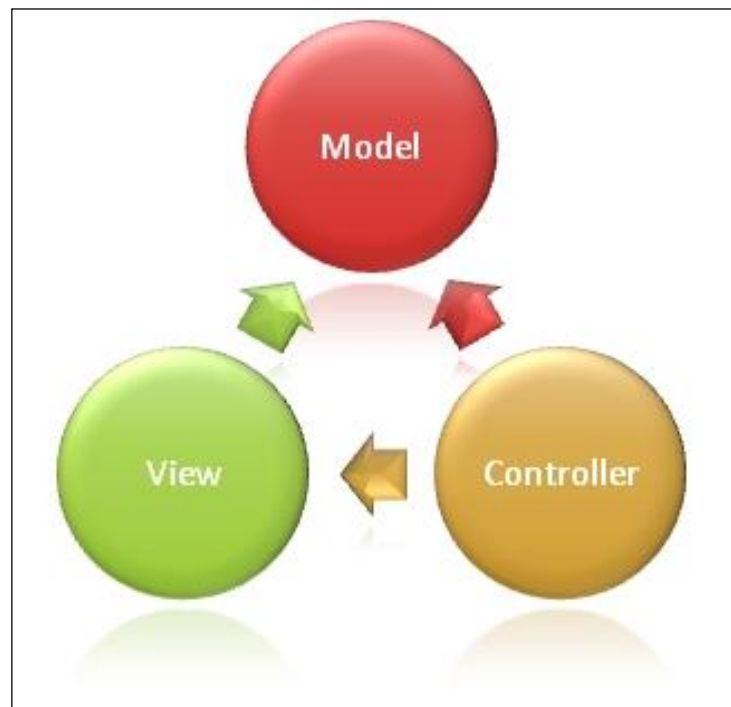
O desenvolvimento do sistema LearnAlgTool seguiu o modelo de arquitetura de *software* MVC (Modelo, Visualização e Controle).

A abordagem MVC é composta por três tipos de objetos. O modelo é o objeto de aplicação, a visão é a apresentação na tela, e o controlador é o que defini a maneira como a interface do usuário reage às entradas do mesmo. (GAMMA, 2000).

Essa abordagem consiste em separar as visões e modelos, estabelecendo um protocolo de inserção/notificação entre eles, sempre que os

dados do modelo mudam, o modelo notifica as visões que dependem dele, conforme podemos visualizar na explicação mostrada na figura 4.

Separação clara entre persistência e interface, maior facilidade de manutenção, reaproveitamento de código e melhor visibilidade da camada de negócios do sistema, são algumas das principais vantagens da utilização do padrão de arquitetura MVC.



**Figura 4 - MVC (Modelo, Visualização, Controle)**

Fonte: [www.w3schools.com/aspnet/mvc\\_intro.asp](http://www.w3schools.com/aspnet/mvc_intro.asp)

## 4. RESULTADOS

Esse capítulo destina-se a apresentar os resultados obtidos com o desenvolvimento do sistema LearnAlgTool. Serão apresentados os principais diagramas, as principais telas do sistema e suas funcionalidades, e a maneira como foram realizados os testes e as validações.

### 4.1. CASOS DE USOS

O diagrama de casos de uso possibilita a compreensão do comportamento externo do sistema por intermédio de uma perspectiva do usuário. Trata-se de um diagrama UML mais abstrato, que representa os papéis desempenhados pelos diversos usuários do sistema.

Na figura 5, são apresentados os principais casos de uso do sistema LearnAlgTool e nas tabelas seguintes, suas especificações. Estes casos de uso apresentam os atores e os recursos que ele pode utilizar.

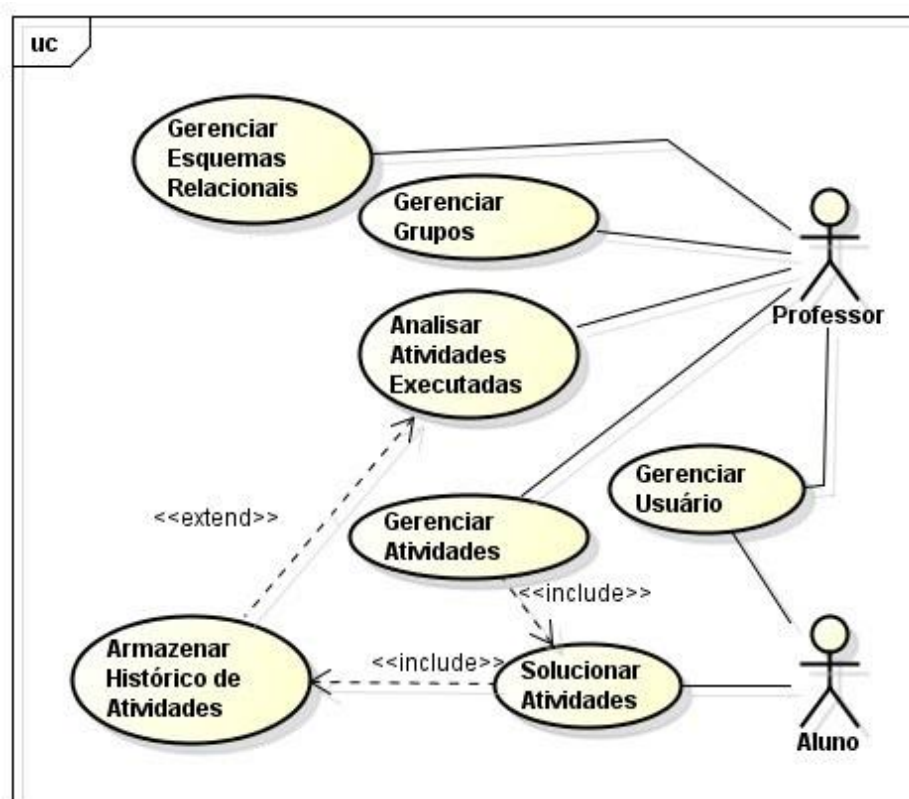


Figura 5 - Casos de Usos

Fonte: A autoria própria

Na tabela 11 é apresentada a descrição geral do diagrama de casos de usos, os dois atores que interagem com o sistema são apresentados e descritos e suas responsabilidades são atribuídas conforme o nível de acesso que possuem.

Na tabela 12 é descrito o principal caso de uso do sistema, gerenciar usuários, os demais casos de usos são descritos e apresentados no anexo A deste documento.

**Tabela 11 - Descrição casos de usos geral**

Nome	Descrição	Responsabilidades
Professor	Professor da disciplina, interessado em avaliar e auxiliar o processo de aprendizagem dos alunos.	Gerenciar atividades e grupos, corrigir e avaliar as atividades solucionadas.
Aluno	Aluno da disciplina, interessado em solucionar as atividades a serem avaliadas.	Solucionar as atividades propostas.

**Fonte: Autoria própria**

**Tabela 12 - Descrição caso de uso gerenciar usuários**

**(continua)**

Nome do caso de uso	Gerenciar usuários
Ator principal	Aluno
Ator secundário	
Resumo	Este caso de uso descreve as etapas necessárias para gerenciar os dados dos usuários do sistema
Pré-condições	É necessário preencher todos os dados solicitados para cadastro, atualizar as informações quando preciso, e encerrar todas as pendências para exclusão
Pós-condições	É necessário se cadastrar no grupo de usuários e realizar as atividades propostas para serem avaliadas.
Fluxo principal	
Ações do ator	Ações do sistema

Tabela 13 - Descrição caso de uso gerenciar usuários

(conclusão)

1- Entrar na página de cadastro e preencher os dados solicitados/ alterar os dados previamente cadastrados/ solicitar sua exclusão do sistema	
	2- Validar os dados inseridos e confirmar solicitação
3- Confirmar seu cadastro clicando no link enviado para o e-mail solicitado/ conferir e corrigir, caso necessário, seu novo cadastro/ confirmar a exclusão.	
	4- Redirecionar o usuário para a página inicial, e permitir seu acesso as funcionalidades do sistema/ atualizar os dados armazenados no sistema/ realizar a exclusão do usuário
Restrições/Validações	1- Para se cadastrar é necessário estar matriculado no curso e possuir um número de registro acadêmico 2- Para que a exclusão seja permitida, não podem haver pendências em aberto
Fluxo alternativo	
Ações do ator	Ações do sistema
	1- Recusar a solicitação e informar ao usuário sobre os motivos do problema ocorrido
2- Cancelar solicitação, ou ignorar o e-mail de confirmação de cadastro	

Fonte: Autoria própria

## 4.2. DIAGRAMA DE CLASSES

O Diagrama de Classes é uma modelagem muito útil para o sistema, pois define todas as classes que o sistema necessita possuir e é a base para a construção dos próximos diagramas utilizados na UML.

A estrutura de classes e suas relações para o sistema LearnAlgTool está definida no diagrama da figura 6.



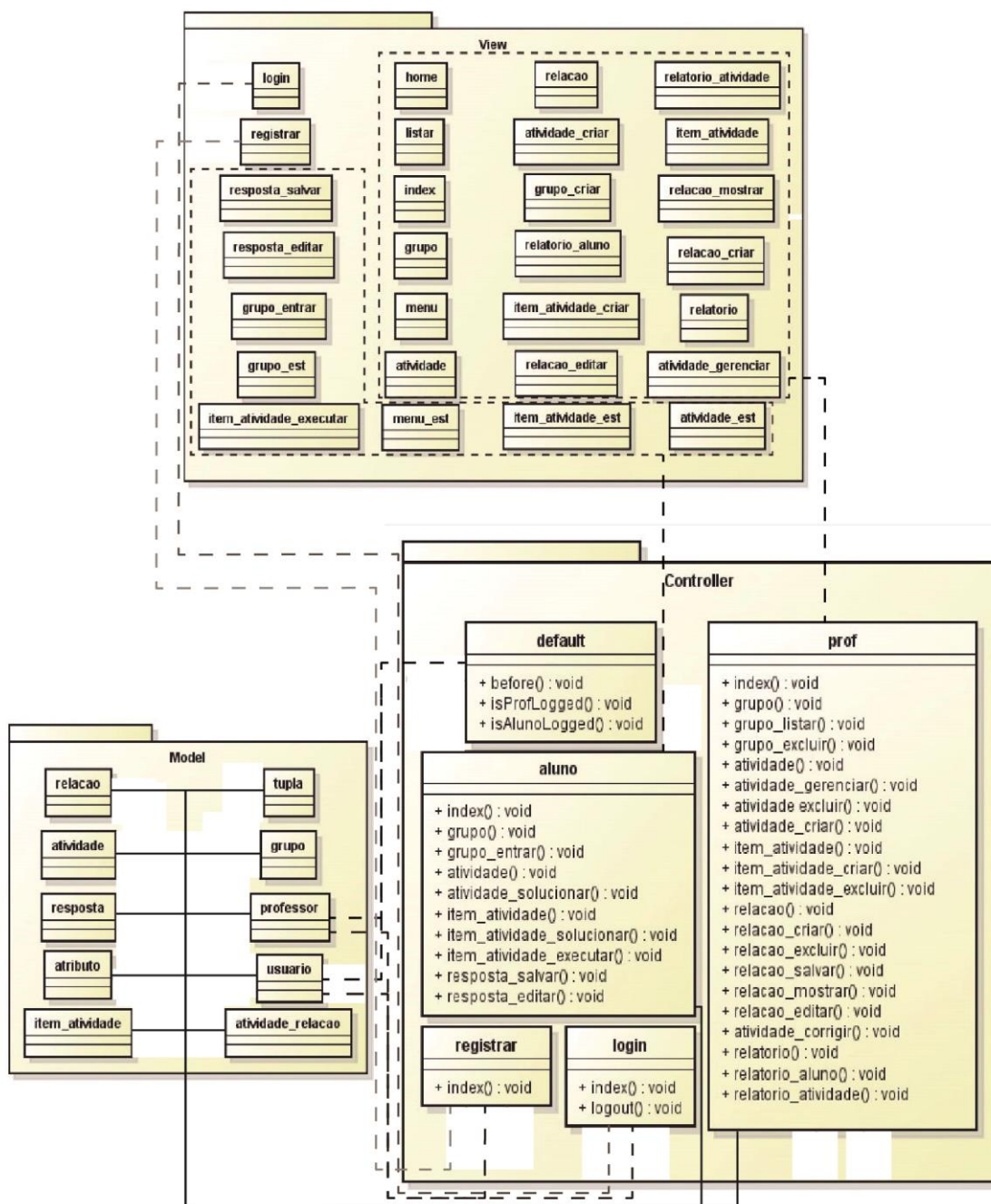


Figura 6 - Diagrama de classes

Fonte: Autoria própria

Acessando a página inicial do Kohana, *framework* utilizado na construção do sistema LearnAlgTool, é possível observar que é utilizado o conceito de divisão de classes definido como MVC, portanto, a construção do sistema segue a mesma linha de pensamento, conforme descrito em seção anterior deste documento.

O *framework* define que para cada tabela existente no banco de dados, é necessária uma única classe correspondente no grupo modelo, e não é

necessário que haja qualquer método ou atributo nessas classes, elas apenas devem se estender a classe ORM, e o próprio *framework* faz as configurações e os ajustes necessários para estabelecer a conexão entre a aplicação e a base de dados.

Há também a definição da rota a ser seguida por meio das classes implementadas no grupo controlador, essas classes enviam as informações necessárias quando os dados precisam ser mudados ou carregados. Quando o controlador é chamado, o *framework* define como rota o nome da ação apresentada na chamada do controlador, caso nenhuma chamada seja passada, o sistema entende que deverá executar a ação index. No sistema LearnAlgTool, foi implementa um controlador para cada nível de acesso, o controlador prof implementa os métodos necessários para gerenciar os dados e definir a rota de redirecionamento a ser seguida pelos usuários do tipo professor, o controlador aluno possui a mesma finalidade do controlador prof, porém defini a regra a ser seguida para usuários com nível de acesso aluno.

Os controladores prof e aluno se estendem ao controlador default. O controle default implementa os métodos utilizados em ambos controladores, além de implementar todos os métodos utilizados em várias partes do mesmo controlador.

Os controladores login e registrar implementam as funcionalidades disponíveis nas classes do grupo de visualização que utilizam os mesmos nomes de seu respectivo controlador. Cada controlador possui relação com duas visualizações diferentes, o menu, de acordo com o nível de acesso do usuário, e o conteúdo da página, de acordo com a ação selecionada no controlador, há também casos em que dois ou mais controladores utilizam a mesma visualização, como, por exemplo nas ações que envolvem criar e editar.

As visualizações implementadas, definem apenas a forma de apresentação dos dados, elas não implementam qualquer função, validação ou cálculo com os dados inseridos, apenas definem a interface gráfica do sistema e sua interação com o usuário final.

### 4.3. MODELO LÓGICO RELACIONAL

O modelo lógico relacional é definido como um modelo baseado na percepção do mundo real, que consiste em um conjunto de objetos básicos chamados entidades e nos relacionamentos entre esses objetos.

Na figura 7 é apresentada o modelo lógico relacional do sistema LearnAlgTool, de forma abstrata e simplificada, seu principal objetivo é facilitar o projeto de banco de dados, possibilitando a especificação da estrutura lógica do banco de dados.

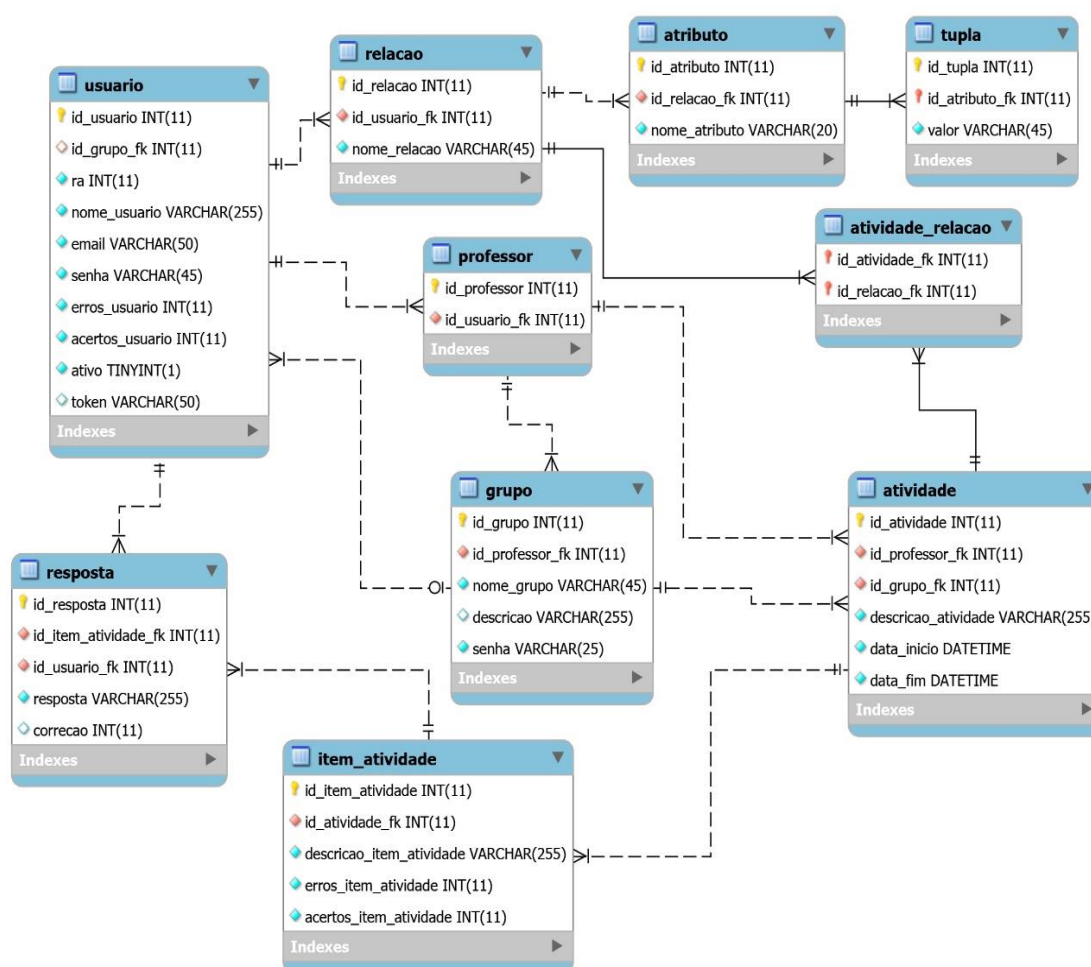


Figura 7 - Modelo relacional

Fonte: Autoria própria

O modelo lógico relacional do sistema LearnAlgTool é composto por dez tabelas, e nelas são armazenados todos os dados necessários para o correto funcionamento do sistema.

Quando um novo usuário realizar seu cadastro, seus dados serão mantidos na tabela “usuario”, que também será responsável por armazenar suas estatísticas de erros e acertos na resolução de atividades. A tabela “professor”, consiste de uma especialização da tabela “usuario”, onde, professor é um usuário responsável por gerenciar atividades e relações, e avaliar o desempenho dos alunos na resolução das atividades propostas. A tabela “grupo” consiste em uma junção de dois ou mais usuários, sendo que um deles deve obrigatoriamente ser um usuário do tipo professor, pois este será responsável por gerenciar o grupo e aplicar as atividades que serão avaliadas.

As tabelas “relacao”, “atributo” e “tupla”, juntas formam as relações que serão utilizadas nas atividades propostas pelos professores, cada uma dessas tabelas, possui um usuário proprietário, responsável por manter e atualizar os dados necessários para a aplicação da atividade.

A tabela “atividade” é responsável pelo armazenamento de todos os dados referentes as atividades que serão aplicadas aos grupos de alunos, cada atividade, terá de ser solucionada dentro de um prazo, pré-definido pelo usuário professor responsável, e cada atividade fará uso de uma ou mais relações, como as relações podem servir para várias atividades propostas, gerou-se um relacionamento N para N, e se fez necessário a criação de uma tabela intermediária “atividade\_relacao”.

Cada atividade proposta será composta por um ou vários itens de atividades, esses serão armazenados na tabela “item\_atividade” juntamente com suas estáticas de erro e acerto, possibilitando ao professor analisar em qual parte do conteúdo está concentrada a maior dificuldade do grupo de alunos.

A cada usuário que propuser uma solução para o item de atividade, será gerada uma resposta, que será armazenada na tabela “resposta” e, posteriormente, possibilitará a avaliação do usuário professor responsável pelo grupo.

#### 4.4. INTERFACES GRÁFICAS DO SISTEMA

Esta seção destina-se a apresentar e descrever as principais telas do sistema LearnAlgTool, e como elas interagem com os dados e as outras interfaces.

Quando o usuário solicita o acesso ao sistema, a primeira tela a ser apresentada é a tela de entrada (Figura 8), onde ele deverá digitar seu e-mail e sua senha de acesso ao sistema, caso as informações estejam corretas, o usuário será redirecionado a tela principal. Caso o usuário não possua um e-mail e uma senha para acesso ao sistema, ele poderá solicitar a abertura da tela de cadastro, e inserir os dados que deseja utilizar para obter o acesso a tela principal.

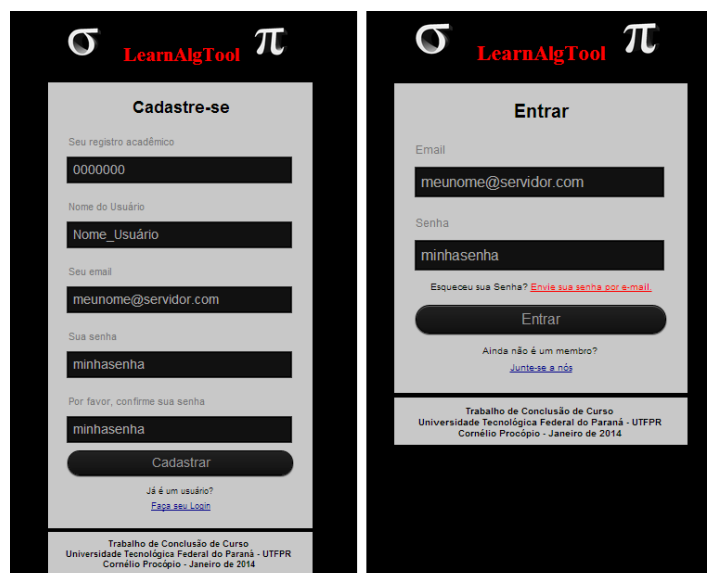


Figura 8 - Telas de logon e registro

Fonte: Sistema LearnAlgTool

Após o usuário entrar no sistema, utilizando seu e-mail e sua senha previamente cadastrada, ele será redirecionado para a tela principal do sistema. A tela principal do sistema não apresenta funcionalidades, sua finalidade é apenas possibilitar ao usuário acesso rápido as funções desejadas, de acordo com o seu nível de acesso permitido. Com a utilização do menu, o usuário pode escolher a funcionalidade que deseja, e essas funcionalidades variam conforme o nível de acesso do usuário, que pode ser aluno ou professor.

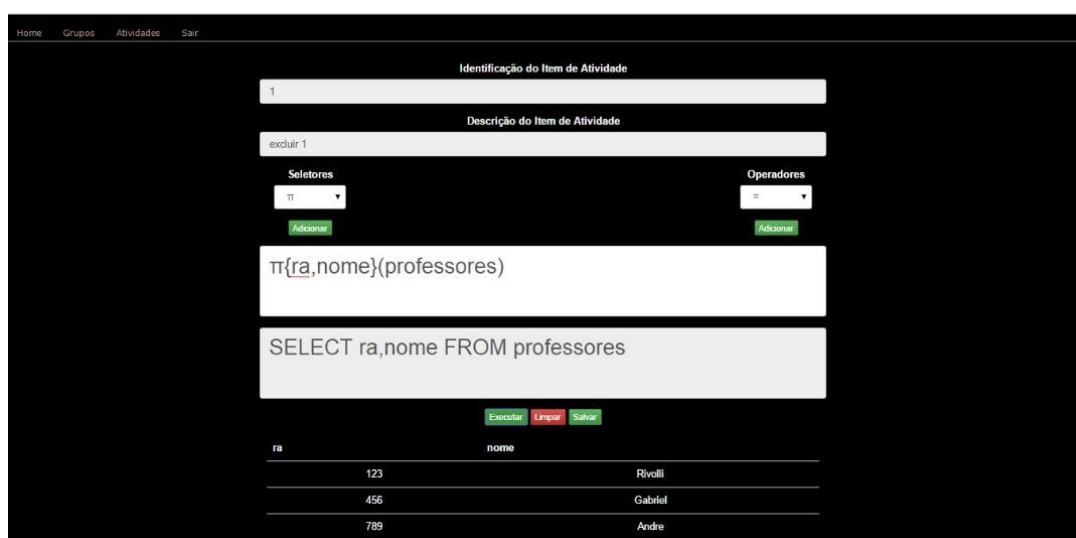
Acessando o menu “Atividades”, os usuários terão acesso a lista de atividades propostas pelo professor responsável por seu grupo (Figura 9). Caso a data de acesso esteja entre a data de início e fim da atividade proposta, os alunos poderão solucionar as atividades e enviar suas respostas para o professor responsável avaliá-las. Essas atividades serão exibidas em uma das telas do sistema, juntamente com as relações necessárias para sua realização, além de um espaço destinado a escrita da solução proposta, conforme mostrado na figura 10, e uma relação resultado, que exibirá a nova relação gerada com base na solução proposta.



Atividade	Data de início	Data final	Solucionar
teste	23/11/2014 - 20:00	30/11/2014 - 03:33	Solucionar
teste	16/11/2014 - 15:40	30/11/2014 - 23:59	Solucionar
teste de criação	15/11/2014 - 22:22	15/11/2014 - 22:23	Solucionar
atividade ok	15/11/2014 - 19:35	30/11/2014 - 00:00	Solucionar

Figura 9 - Tela de atividades

Fonte: Sistema LearnAlgTool



Identificação do Item de Atividade

1

Descrição do Item de Atividade

excluir 1

Seletores

TI

Operadores

=

Adicionar

Adicionar

$\pi\{ra, nome\}(professores)$

SELECT ra,nome FROM professores


Executar Limpar Salvar

ra	nome
123	Rivelli
456	Gabriel
789	Andre

Figura 10 – Resoluções de atividades

Fonte: Sistema LearnAlgTool


Após a data final definida para que os alunos possam solucionar as atividades propostas, o usuário, com nível de acesso professor, irá corrigir as soluções propostas pelos alunos de seu grupo, podendo atribuir certo e aumentar a estatística de acerto do aluno, ou errado e aumentar a estatística de erros do aluno. Após a correção de todas as soluções propostas, o usuário professor poderá gerar um relatório de avaliação de desempenho dos alunos, conforme mostrado na figura 11, possibilitando que o professor analise quais são os alunos que estão apresentando maior dificuldade no processo de aprendizagem.



**LearnAlgTool - Relatórios**

**Grupo - "teste2"**

**Relatório de Alunos**



RA	Nome	E-mail	Erros	Acertos
987654321	teste123	tcc@gmail.com	0	1

Página 1/1

**Figura 11 - Relatórios do Sistema**

Fonte: Sistema LearnAlgTool

## 4.5. VALIDAÇÕES E TESTES

As validações buscam avaliar se os requisitos propostos no início do projeto estão funcionando adequadamente e se estão atendendo as necessidades do usuário conforme o esperado.

Desde o início da implementação, a cada implemento concluído, todas as funcionalidades desenvolvidas durante o incremento eram validadas e testadas, alguns dados eram inseridos e salvos na base dados, depois alterados e excluídos, cada linha de código que o sistema utilizava era analisada e corrigida, caso fosse necessário.

O sistema LearnAlgTool passou por inúmeras validações e testes, e todas as atividades desenvolvidas com esta finalidade geraram resultados satisfatórios.

Em um dos processos de testes aplicado na ferramenta, definido como testes de caixa preta, o analista busca analisar o comportamento de um objeto ignorando totalmente sua construção interna. Este tipo de teste, também

chamado de teste funcional, o analista não possui acesso ao código fonte e desconhece a estrutura interna do sistema.

A cada conclusão de incremento do sistema, afim de realizar os testes de caixa preta, dados eram inseridos, alterados e eliminados, para verificar como o sistema se comportaria durante o processo, nesse caso, sem preocupação com as atividades internas do sistema, era analisado somente o resultado final do processamento.

No primeiro incremento foi necessário realizar a correção no cadastro, evitando que alunos se cadastrassem utilizando o mesmo número de registro acadêmico. No segundo incremento, foi necessário corrigir as exclusões dos grupos, para que não gerarem problemas de restrições nas bases de dados, e o gerenciamento das informações dos grupos. No terceiro incremento, foi necessário corrigir as exclusões das atividades, e corrigir o gerenciamento dos vínculos entre as atividades e as relações. No quarto incremento foram necessárias quatro correções diferentes na parte de gerenciamento das relações, desde a inserção e exclusão de dados, até a exibição e alteração dos dados inseridos. No quinto incremento foi necessário corrigir os somatórios de erros e acertos dos usuários e de suas atividades, para que caso mudasse o status de uma correção, que já havia sido corrigida anteriormente, o sistema calculasse os erros e acertos de forma correta. No sexto e no sétimo incremento, devido ao trabalho realizado nos incrementos anteriores não foram encontrados erros. No oitavo incremento, todos os erros encontrados e corrigidos se referem a tradução da consulta em álgebra relacional para SQL.

Os resultados dos testes realizados são exibidos na tabela 13.

**Tabela 14 - Validações e testes**

Incremento	Quantidade de erros encontrados e corrigidos
Incremento 01	1
Incremento 02	2
Incremento 03	2
Incremento 04	3
Incremento 05	1
Incremento 06	0
Incremento 07	0
Incremento 08	6

**Fonte: Autoria Própria**



## 5. CRONOGRAMA

Este capítulo apresenta o cronograma final do sistema LearnAlgTool, (Tabela 15 - **Cronograma Final**), no qual, são especificadas as datas, de início e fim, das fases do projeto.

As fases finais do sistema LearnAlgTool foram divididas em:

- **Concepção** – Fase que compreende a discussão com o professor orientador, a fim de analisar a viabilidade do projeto;
- **Análise** – Fase que compreende a análise dos requisitos para o desenvolvimento do sistema;
- **Projeto** – Fase que compreende o projeto das telas do sistema, do SGBD;
- **Desenvolvimento** – Fase que compreende o desenvolvimento das funcionalidades do sistema proposto e de sua interface, juntamente com os testes unitários;
- **Avaliação** – Fase que compreende a resolução de erros e problemas encontrados no sistema, além da manutenção de suas funcionalidades, a fim de que o sistema continue executando corretamente, e conforme as especificações.

As atividades foram iniciadas em junho de 2014, e encerradas em novembro de 2014. Algumas fases ocorrem simultaneamente, buscando acelerar o desenvolvimento, e permitir o cumprimento do cronograma proposto.

Com exceção da primeira fase do projeto, definida como fase de concepção, e da última fase, a elaboração deste documento, todas as fases são cumpridas a cada incremento do sistema que está sendo desenvolvido, buscando a criação de um projeto, livre de erros ou falhas, e que respeite o prazo definido em seu cronograma.

Tabela 15 - Cronograma Final

Ano	2014																											
Meses	Junho				Julho				Agosto				Setembro				Outubro				Novembro							
Semanas	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Fases																												
Concepção	X	X	X																									
Incremento 1																												
• Análise				X																								
• Projeto				X																								
• Desenv.					X																							
• Avaliação					X																							
Incremento 2																												
• Análise						X	X																					
• Projeto						X	X																					
• Desenv.							X	X	X																			
• Avaliação							X	X	X																			
Incremento 3																												
• Análise									X																			
• Projeto									X																			
• Desenv.									X	X	X																	
• Avaliação									X	X	X																	
Incremento 4																												
• Análise										X																		
• Projeto										X	X																	
• Desenv.											X	X	X															
• Avaliação											X	X	X															
Incremento 5																												
• Análise													X	X														
• Projeto													X	X														
• Desenv.														X	X													
• Avaliação														X	X													
Incremento 6																												
• Análise														X														
• Projeto														X														
• Desenv.															X	X												
• Avaliação																X												
Incremento 7																												
• Análise																	X											
• Projeto																	X											
• Desenv.																		X										
• Avaliação																		X										
Incremento 8																												
• Análise																	X	X										
• Projeto																	X	X	X									
• Desenv.																		X	X	X	X							
• Avaliação																			X	X	X							
Documento																												
• Conc.																	X	X	X	X	X	X	X	X	X	X	X	X

Fonte: Autoria Própria

## 6. CONSIDERAÇÕES FINAIS

A análise das ferramentas que permitem realizar consultas utilizando álgebra relacional, permitiu constatar que uma nova ferramenta poderia ser implementada. As ferramentas analisadas LEAP e RELATIONAL possuem características que dificultam sua utilização, como interface não intuitiva, sintaxe e simbologia diferente da utilizada pelos principais autores dos livros de banco de dados, idioma em inglês e alguns componentes da interface que não funcionam ou não produzem o resultado desejado.

A ferramenta LearnAlgTool foi proposta, tendo como objetivos principais a criação de uma ferramenta web, de fácil utilização, que auxilie no aprendizado de álgebra relacional. A segunda característica da ferramenta que merece ser destacada está no fato que ela utiliza simbologia e sintaxe comum, aos principais livros de banco de dados, e interface intuitiva, que facilita sua utilização. Por fim, como terceira característica, vale destacar que por se tratar de um sistema web, a ferramenta LearnAlgTool não necessita de instalação na máquina do usuário, o que diminui a resistência dos usuários em utilizá-la. Espera-se que estas características e vantagens façam do LearnAlgTool uma ferramenta empregada para auxiliar no ensino e aprendizagem de álgebra relacional, nos cursos de computação.

### 6.1. DIFICULDADES ENCONTRADAS

As dificuldades encontradas para a conclusão deste trabalho foram diversas, entretanto, duas dessas dificuldades merecem maior destaque. A primeira dificuldade surgiu com a necessidade em aprender novos conceitos e tecnologias, que seriam necessários para o desenvolvimento de todo o trabalho.

A segunda e mais significativa dificuldade encontrada, consistia em estudar e implementar uma maneira, de garantir que os usuários do sistema, possam escrever suas consultas, utilizando a sintaxe da álgebra relacional, com o menor número de restrições possíveis, e garantir a correção no resultado apresentado após a execução da consulta.

## 6.2. TRABALHOS FUTUROS

Após a finalização da implementação e a realização de vários testes bem sucedidos com as operações suportadas pelo sistema, tornou-se possível a análise dos resultados e a comparação entre as funcionalidades dos programas relacionados e o sistema desenvolvido. Com base nesses resultados conclui-se que o sistema LearnAlgTool atingiu com sucesso seu principal objetivo, e está apta a ser utilizada no auxílio do aprendizado da álgebra relacional.

Entretanto, surgiram novos requisitos e oportunidades. Sugere-se para trabalhos futuros, que seja implementado o suporte as operações adicionais e estendidas, permitindo assim que os usuários realizem essas consultas, utilizando a sintaxe da álgebra relacional, e não se limitem apenas as operações fundamentais.

Permitir também, que sejam gerados relatórios mais completos, de acordo com as necessidades do usuário. Além de avaliar o desempenho dos alunos nos itens de atividade, a possibilidade de gerar um novo relatório, por atividades, grupos ou por outros filtros, possibilitaria aos professores terem um maior foco nas atividades que estão as maiores dificuldades de aprendizado de um determinado grupo. Outro avanço será permitir que os usuários gerenciem as relações quando estiverem executando consultas livres. Com esse recurso, os usuários poderiam criar, deletar e alterar as relações do sistema, o que facilitaria o seu aprendizado e entendimento, antes de realizar as atividades propostas por seus professores, para serem avaliados.

## 7. REFERÊNCIAS

Date. C. J. **Introdução a Sistemas de Banco de Dados**. 4. ed. Rio de Janeiro: Campus, 1990.

Elmasri. Ramez. **Sistema de Banco de Dados**. 4. ed. São Paulo: Pearson Education do Brasil, 2005.

Gamma. Erich. **Padrões de projeto, soluções reutilizáveis de software orientado a objetos**. Porto Alegre: Bookman, 2000.

Heuser. Carlos Alberto. **Projeto de Banco de Dados**. 4. ed. Porto Alegre: Sagra Luzzatto, 1999.

Kroenke. David M. **Banco de Dados – Fundamentos, Projeto e Implementação**. 6. ed. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 1999.

RELATIONAL. **Um compilador de álgebra relacional em python**. Disponível em:<<http://galileo.dmi.unict.it/wiki/relational/doku.php>> Acesso em: 07 abr. 2013.

LEAP. **Um sistema de gerenciamento de banco de dados relacional educacional**. Disponível em:<<http://leap.sourceforge.net/index.html>> Acesso em: 07 abr. 2013.

Silberschatz. Abraham; Korth, Henry F. **Sistema de Banco de Dados**. 3. ed. São Paulo: Makron Books Ltda, 1997.

\_\_\_\_\_. **Sistema de Banco de Dados**. 5. ed. Rio de Janeiro: Campus, 2006.

**ANEXO A – Diagrama de casos de usos e especificações**

Conforme descrito na seção 4.1 do capítulo 4 deste documento, a especificação dos demais casos de usos serão especificadas nas tabelas deste arquivo anexo.

Na tabela 15 é descrito o caso de uso gerenciar grupos.

**Tabela 16 - Descrição caso de uso gerenciar grupos**

**(continua)**

Nome do caso de uso	Gerenciar grupos
Ator principal	Professor
Ator secundário	Aluno
Resumo	Este caso de uso descreve as etapas necessárias para gerenciar os dados dos grupos de usuários do sistema
Pré-condições	É necessário que os usuários estejam cadastrados no sistema antes de serem inseridos no grupo
Pós-condições	É necessário realizar as atividades propostas para serem avaliadas.
Fluxo principal	
Ações do ator	Ações do sistema
1- O professor deve criar e manter novos grupos de usuários conforme necessário	
2- Os usuários devem entrar na página de cadastro e solicitar sua inserção no grupo, digitando a senha solicitada, ou pedindo sua exclusão do grupo, quando já cadastrado	
	2- Validar os dados inseridos e solicitar a confirmação da solicitação
3- Confirmar a solicitação	
	4- Caso os dados inseridos estejam corretos, e não haja pendências, executar a solicitação do usuário
	1- Para se cadastrar no grupo é necessário estar previamente cadastrado no sistema 2- O usuário não poderá realizar um novo cadastro, sem antes deixar de fazer parte do grupo ao qual está cadastrado

Tabela 17 - Descrição caso de uso gerenciar grupos

(conclusão)

Restrições/Validações	1- Para se cadastrar no grupo é necessário estar previamente cadastrado no sistema 2- Caso já esteja cadastrado em algum grupo, o usuário não poderá realizar um novo cadastro, sem antes deixar de fazer parte do grupo ao qual está cadastrado
Fluxo alternativo	
Ações do ator	Ações do sistema
Fluxo de exceção	
Ações do ator	Ações do sistema
	1- Recusar a solicitação e informar ao usuário sobre os motivos do problema ocorrido
2- Cancelar solicitação, corrigir as informações necessárias e enviar nova solicitação	

Fonte: Autoria própria

Na tabela 16 é descrito o caso de uso gerenciar esquemas relacionais.

Tabela 18 - Descrição caso de uso gerenciar esquemas relacionais

(continua)

Nome do caso de uso	Gerenciar esquemas relacionais
Ator principal	Professor
Ator secundário	
Resumo	Este caso de uso descreve as etapas necessárias para gerenciar os esquemas relacionais que serão utilizados no sistema
Pré-condições	É necessário estar logado no sistema e possuir acesso ao nível "professor"
Pós-condições	
Fluxo principal	
Ações do ator	Ações do sistema



**Tabela 19 - Descrição caso de uso gerenciar esquemas relacionais****(conclusão)**

1- O professor deve criar e manter novas relações que serão utilizadas nas atividades	
	2- Validar os dados inseridos e pedir a confirmação da solicitação
3- Confirmar a solicitação	
	4- Executar a solicitação do cliente e atualizar a lista de relações disponíveis a serem usadas nas atividades
Restrições/Validações	1-As relações que estiverem sendo utilizadas em atividades ainda em aberto não poderão ser exclusas ou atualizadas
Fluxo alternativo	
Ações do ator	Ações do sistema
Fluxo de exceção	
Ações do ator	Ações do sistema
	1- Recusar a solicitação e informar ao usuário sobre os motivos do problema ocorrido
2- Cancelar solicitação, corrigir as informações necessárias e enviar nova solicitação	

**Fonte: Autoria própria**

Na tabela 17 é descrito o caso de uso gerenciar atividades.

**Tabela 17 - Descrição caso de uso gerenciar atividades****(continua)**

Nome do caso de uso	Gerenciar atividades
Ator principal	Professor
Ator secundário	
Resumo	Este caso de uso descreve as etapas necessárias para gerenciar as atividades que serão propostas para a avaliação do desempenho dos alunos
Pré-condições	É necessário estar logado no sistema e possuir acesso ao nível "professor"

Tabela 17 - Descrição caso de uso gerenciar atividades

(conclusão)

Pós-condições	
Fluxo principal	
Ações do ator	Ações do sistema
1- O professor deverá solicitar a criação, e manter as atividades que serão utilizadas na avaliação de desempenho dos alunos	
	2- Conferir e validar os dados inseridos e solicitar a confirmação da solicitação
3- Confirmar a solicitação	
	4- Executar a solicitação, e disponibilizar a atividade para ser solucionada dentro do prazo definido
Restrições/Validações	1-Após o prazo de encerramento de uma atividade ela não poderá ser editada
Fluxo alternativo	
Ações do ator	Ações do sistema
Fluxo de exceção	
Ações do ator	Ações do sistema
	1- Recusar a solicitação e informar ao usuário sobre os motivos do problema ocorrido
2- Cancelar solicitação, corrigir as informações necessárias e enviar nova solicitação	

Fonte: Autoria própria

Na tabela 18 é descrito o caso de uso solucionar atividades.

Tabela 18 - Descrição caso de uso solucionar atividades

(continua)

Nome do caso de uso	Solucionar atividades
Ator principal	Aluno
Ator secundário	
Resumo	Este caso de uso descreve as etapas necessárias para solucionar as atividades propostas para a avaliação do desempenho

**Tabela 18 - Descrição caso de uso solucionar atividades****(conclusão)**

Pré-condições	É necessário estar logado no sistema e possuir acesso ao nível “aluno”
Pós-condições	É necessário avaliar as soluções apresentadas pelos alunos
Fluxo principal	
Ações do ator	Ações do sistema
1-Inserir a consulta, utilizando a sintaxe da álgebra relacional, no campo indicado, e solicitar a execução da consulta proposta	
	2- Executar a consulta inserida e exibir o resultado da execução na interface gráfica
3- Analisar o resultado da consulta, e enviá-la para avaliação	
	4- Executar o caso de uso armazenar histórico de atividades
Restrições/Validações	1- Após o envio da solução proposta para validação ela não poderá ser editada
Fluxo alternativo	
Ações do ator	Ações do sistema
Fluxo de exceção	
Ações do ator	Ações do sistema
1- Analisar o resultado da consulta e corrigi-la caso o resultado não seja o esperado	
	2- Voltar ao segundo passo do fluxo principal deste mesmo caso de uso

**Fonte: Autoria própria**

Na tabela 19 é descrito o caso de uso armazenar histórico atividades.

**Tabela 19 - Descrição caso de uso armazenar histórico de atividades****(continua)**

Nome do caso de uso	Armazenar histórico de atividades
Ator principal	
Ator secundário	

**Tabela 19 - Descrição caso de uso armazenar histórico de atividades****(conclusão)**

Resumo	Este caso de uso descreve as etapas necessárias para armazenar o histórico das atividades solucionadas pelo usuário
Pré-condições	É necessário que o caso de uso solucionar atividades já tenha sido executado
Pós-condições	É necessário executar o caso de uso analisar atividades executadas
Fluxo principal	
Ações do ator	Ações do sistema
1- O usuário realizará o caso de uso solucionar atividades, e solicitará o envio da resposta por ele proposta para avaliação	
	2- Armazenar no banco de dados a solução proposta e disponibiliza-la para avaliação
Restrições/Validações	1-Após o envio da resposta para avaliação, não será permitido, pelo mesmo usuário, a execução do caso de uso solucionar atividades, para a mesma atividade proposta
Fluxo alternativo	
Ações do ator	Ações do sistema
Fluxo de exceção	
Ações do ator	Ações do sistema
	1- Recusar a solicitação e informar ao usuário sobre os motivos do problema ocorrido
2- Cancelar solicitação, corrigir as informações necessárias e enviar nova solicitação	

**Fonte: Autoria própria**

Na tabela 20 é descrito o caso de uso analisar atividades executadas.

**Tabela 20 - Descrição caso de uso analisar atividades executadas****(continua)**

Nome do caso de uso	Analisar as atividades executadas
Ator principal	Professor

Tabela 20 - Descrição caso de uso analisar atividades executadas

(continua)

Ator secundário	
Resumo	Este caso de uso descreve as etapas necessárias para analisar as atividades solucionadas pelo usuário
Pré-condições	É necessário estar logado no sistema e possuir acesso ao nível “professor” O caso de uso armazenar histórico de atividades já deve ter sido executado A data da correção deve ser maior que o prazo final para solucionar as atividades
Pós-condições	É possível gerar relatórios buscando facilitar a avaliação final do desempenho dos alunos
Fluxo principal	
Ações do ator	Ações do sistema
1-Solicitar o início da correção das soluções propostas	
	2- Exibir as soluções propostas pelos alunos, uma a uma, e solicitar o resultado, certo ou errado
3- Analisar a solução proposta, e atribuir certo, caso a consulta gere o resultado esperado, ou errado, caso a consulta gere um resultado diferente do esperado	
	4- Atribuir a resposta a avaliação de desempenho do aluno, e inclui-la na geração de dados do relatório de avaliação
Restrições/Validações	1- Somente serão avaliados os usuários que propuserem uma solução a ser avaliada, os que não propuserem uma solução não terão sua avaliação afetada
Fluxo alternativo	
Ações do ator	Ações do sistema
1- Caso necessário, o usuário “professor” poderá solicitar ao sistema a execução da consulta	
	2- Executar a consulta, e exibir a relação resultado na interface gráfica

**Tabela 20 - Descrição caso de uso analisar atividades executadas****(conclusão)**

3- Analisar a solução proposta, e atribuir certo, caso a consulta gere o resultado esperado, ou errado, caso a consulta gere um resultado diferente do esperado	
	4- Atribuir a resposta a avaliação de desempenho do aluno, e inclui-la na geração de dados do relatório de avaliação
Fluxo de exceção	
Ações do ator	Ações do sistema

**Fonte: Autoria própria**

**ANEXO B – Proposta do trabalho de Diplomação**

Universidade Tecnológica Federal do Paraná  
Campus Cornélio Procópio  
Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas

Gabriel Guimarães da Silva

**LearnAlgTool – SISTEMA PARA AUXÍLIO DE APRENDIZAGEM  
DE ÁLGEBRA RELACIONAL**

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

Cornélio Procópio  
2013



Gabriel Guimarães da Silva

## **LearnAlgTool – SISTEMA PARA AUXÍLIO DE APRENDIZAGEM DE ÁLGEBRA RELACIONAL**

Proposta para trabalho de conclusão de curso em Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Ms. Adriano Rivolli  
da Silva

Cornélio Procópio

2013

## Resumo

Um banco de dados relacional consiste em uma coleção de relações, cada uma das quais com um nome único. Uma tupla em uma relação representa um relacionamento entre um conjunto de valores. Este modelo atraiu imediatamente a atenção, após ser proposto em um artigo clássico (Codd, p. 11), em virtude de sua simplicidade e sua base matemática. O modelo usa o conceito de uma relação matemática como seu bloco de construção básica e tem sua base teórica na teoria dos conjuntos e na lógica de predicados de primeira ordem. O conjunto básico de operações para o modelo relacional é a álgebra relacional. A álgebra relacional é uma linguagem de consulta procedural, que consiste em um conjunto de operações tendo como entrada uma ou duas relações e produzindo, como resultado, uma nova relação. A álgebra relacional provê um fundamento formal para operações no modelo relacional, e é utilizada como base para implementar e otimizar as consultas em sistemas de gerenciadores de bancos de dados relacional. A dificuldade em encontrar ferramentas que interpretem consultas utilizando álgebra relacional dificulta o aprendizado dos alunos nos cursos de computação que possuem disciplinas de banco de dados. Duas ferramentas encontradas foram analisadas, e os aspectos positivos e negativos de cada *software* ajudaram a tomar decisões sobre o que ainda pode ser feito. O presente trabalho propõe a construção de um sistema que simule consultas utilizando a sintaxe da álgebra relacional, para auxiliar no aprendizado das operações fundamentais. O sistema foi proposto, tendo como base, suprir as carências dos *softwares* existentes.

**Palavras-chave:** Álgebra relacional, Consulta, Modelo relacional, aprendizado.

**Lista de Ilustrações**

FIGURA 1 - INTERFACE DO SISTEMA RELATIONAL .....	24
FIGURA 2 - INTERFACE DO SISTEMA LEAP .....	25
FIGURA 3 - CASOS DE USOS DO SISTEMA LEARNALGTOOL .....	74
FIGURA 4 - MODELO ENTIDADE E RELACIONAMENTO .....	76

**Lista de Tabelas**

TABELA 1 - COMPARAÇÃO ENTRE OS <i>SOFTWARES</i> .....	73
TABELA 2 - TECNOLOGIAS UTILIZADAS .....	77
TABELA 3 - CRONOGRAMA INICIAL .....	78

## Lista de Siglas e Abreviaturas

SQL – *Structured Query Language* (Linguagem de Consulta Estruturada).

SGBD – Sistema Gerenciador de Banco de Dados.

HTML – *Hypertext Markup Language* (Linguagem de Marcação de Hipertexto).

CSS – *Cascading Style Sheets* (Folhas de Estilo em Cascata).

UML – *Unified Modeling Language* (Linguagem de Modelagem Unificada).

IDE – *Integrated Development Environment* (Ambiente Integrado de Desenvolvimento).

### Lista de Símbolos

- $\pi$  – Operador de projeção em álgebra relacional (PI).
- $\sigma$  – Operador de seleção em álgebra relacional (Sigma).
- $\times$  – Produto cartesiano em álgebra relacional.
- $\rho$  – Renomear em álgebra relacional (Rho).
- $\cup$  – União em álgebra relacional.

## Sumário

1. INTRODUÇÃO.....	13
1.1. OBJETIVOS .....	14
1.1.1. Objetivos gerais.....	14
1.1.2. Objetivos específicos.....	14
1.2. PROBLEMAS E JUSTIFICATIVA .....	14
1.3. ORGANIZAÇÃO DO TEXTO .....	15
2. FUNDAMENTAÇÃO TEÓRICA .....	16
2.1. MODELO RELACIONAL.....	16
2.2. ÁLGEBRA RELACIONAL.....	18
2.3. TRABALHOS RELACIONADOS .....	23
2.3.1. Relational.....	23
2.3.2. LEAP .....	24
2.3.3. Comparação entre o LearnAlgTool e os softwares relacionados.....	25
3. METODOLOGIA .....	26
3.1. FERRAMENTAS E TECNOLOGIAS.....	26
3.2. PROCESSO DE DESENVOLVIMENTO.....	27
3.3. ARQUITETURA DO SISTEMA .....	28
4. RESULTADOS .....	30
4.1. CASOS DE USOS.....	30
4.2. DIAGRAMA DE CLASSES.....	32
4.3. MODELO RELACIONAL.....	35
4.4. INTERFACES GRÁFICAS DO SISTEMA .....	37
4.5. VALIDAÇÕES E TESTES.....	39
5. CRONOGRAMA .....	41
6. CONSIDERAÇÕES FINAIS.....	43
6.1. DIFICULDADES ENCONTRADAS.....	43
6.2. TRABALHOS FUTUROS .....	44
7. REFERÊNCIAS .....	45

## **1. Introdução**

A forma como os dados eram armazenados em períodos anteriores ao computador, se tornou inviável, por não suprir as necessidades, gerar alto custo e um grande volume de trabalho. Então, na década de 60, as empresas passam a utilizar computadores para armazenar seus dados e duas décadas depois o modelo Entidade-Relacionamento se populariza devido a facilidade de utilização, e a Linguagem de Consulta Estruturada (SQL) se torna um padrão mundial, utilizada até os dias de hoje (REZENDE 2007).

Atualmente são utilizados sistemas gerenciadores de banco de dados (SGBD) que gerenciam e acessam os dados em bancos relacionais (BDR). Os bancos relacionais armazenam os dados na forma de relações e cada relação possui um atributo (ou atributos) chamado chave primaria que permite identificar os conjuntos de valores presentes. Para se efetuar uma consulta, utiliza-se uma linguagem de consulta, processada pelo SGBD.

O conjunto básico de operações para o modelo relacional é a álgebra relacional, uma linguagem de consulta procedural, que consiste em um conjunto de operações tendo como entrada uma ou duas relações e produzindo, como resultado, uma nova relação. A álgebra relacional é utilizada como base para implementar e otimizar as consultas em sistemas de gerenciadores de bancos de dados relacional.

### **1.1. Objetivos**

#### **1.1.1. Objetivos Gerais**

O objetivo geral deste trabalho consiste em desenvolver uma aplicação web que permita realizar operações de consultas em bancos de dados relacionais utilizando a sintaxe de álgebra relacional. Espera-se que os professores possam gerar as relações e dados, propondo questionários de consultas, e que os alunos possam resolver as atividades utilizando os conceitos de álgebra relacional, permitindo que os professores possam controlar a evolução e desempenho dos alunos, e estes passam a obter resposta imediata para seus exercícios.



### 1.1.2. Objetivos Específicos

- Analisar os softwares existentes, que permitam a realização de consulta utilizando álgebra relacional.
- Estudar e conhecer as operações de álgebra relacional.
- Fornecer uma ferramenta prática para a elaboração de consultas utilizando álgebra relacional
- Permitir que os professores de banco de dados possam acompanhar o desempenho de seus alunos, quanto aos conceitos de álgebra relacional

### 1.2. Justificativa

O estudo e a compreensão do modelo de dados relacional e de álgebra relacional são fundamentais para estudantes dos cursos de computação que contenham disciplinas de banco de dados. A álgebra relacional, por se tratar de uma matéria teórica, apresenta na falta de atividades práticas a principal dificuldade para sua compreensão e aprendizado, tornando-os mais complicados e demorados. Uma forma de facilitar seu aprendizado é proporcionar um ambiente agradável para testes e consultas, utilizando a álgebra relacional.

Com a crescente utilização de banco de dados relacionais, a necessidade de pessoas que gerenciem essa base de dados cresce em mesma proporção, o ensino de banco de dados relacionais e álgebra relacional se tornam mais fundamental, e o tempo para aprendizagem se torna curto, ferramentas que facilitem o aprendizado e diminuam a curva de tempo são de extrema importância.

A dificuldade em encontrar ferramentas relacionadas, e os problemas existentes nas poucas ferramentas encontradas, torna este um espaço vago e carente, que necessita de ferramentas que atendam aos requisitos especificados, e motiva o desenvolvimento desse sistema.

### **1.3. Organização do texto**

No capítulo 2 é apresentada uma breve revisão dos conceitos sobre álgebra relacional, principais formas de utilização, as vantagens do seu uso, e por fim, exemplos práticos de sua utilização.

No capítulo 3 é apresentado um breve resumo dos trabalhos relacionados encontrados, uma descrição sobre as ferramentas, suas características principais, além das vantagens e desvantagens em sua utilização.

No capítulo 4 é apresentada a definição inicial do sistema LearnAlgTool, o diagrama de casos de uso inicial, os requisitos funcionais e não funcionais, o projeto de banco de dados, e as tecnologias utilizadas para desenvolvimento.

No capítulo 5 é apresentado o cronograma inicial do projeto, com as datas previstas para o início e o encerramento de cada uma de suas fases.

No capítulo 6 são apresentadas as considerações finais.

## 2. Fundamentação Teórica

Esse capítulo destina-se a apresentar a definição do modelo relacional e da álgebra relacional, as formas mais comuns de sua utilização, e apresentar alguns exemplos de utilização e seus operadores fundamentais.

### 2.1. Modelo Relacional

Um banco de dados relacional é constituído de uma coleção de tabelas, cada uma com um nome único. Uma linha em uma tabela representa um relacionamento entre um conjunto de valores. Uma vez que essa tabela é uma coleção de tais relacionamentos, há uma estreita correspondência entre o conceito de tabela e o conceito matemático de relação, a partir dos quais se origina o nome desse modelo de dados.

O modelo relacional teve sua introdução na década de 70 após um artigo clássico (Codd, 1970), publicado por Ted Codd, da IBM Research. O modelo usa o conceito de uma relação matemática como seu bloco de construção básica e tem sua base teórica na teoria dos conjuntos e na lógica de predicados de primeira ordem. Imediatamente após sua aplicação o modelo atraiu a atenção em virtude de sua simplicidade e sua base matemática.

Na terminologia do modelo relacional formal, uma linha é chamada de tupla, um cabeçalho de coluna é conhecido como atributo, e a tabela é chamada de relação.

Uma relação é definida como um conjunto de tuplas. Por definição, todos os elementos de um conjunto são distintos, por isso, todas as tuplas da relação também devem ser distintas.

No modelo relacional, cada tupla na relação representa um fato que corresponde a uma entidade ou relacionamento no mundo real. O nome da relação e os nomes dos atributos são usados para ajudar na interpretação do significado dos valores em cada tupla.

O conceito básico para identificar tuplas e estabelecer relações entre as tuplas das relações de um banco de dados relacional é o da chave. Em um banco de dados relacional, há ao menos três tipos de chaves a considerar: a chave primária, a chave alternativa, e a chave estrangeira.

Uma chave primária é um atributo, ou uma combinação de dois ou mais atributos, cujos valores servem para distinguir uma tupla das demais tuplas dentro da relação. Segundo as definições formais de chave primária, esta chave deve ser mínima, pois exige-se que todos os seus atributos sejam efetivamente necessários para garantir o requisito de unicidade de valores que compõem a chave.

Uma chave estrangeira é um atributo, ou uma combinação de dois ou mais atributos, cujos valores são necessariamente chave primária de alguma relação. A chave estrangeira é o mecanismo que permite a implementação de relacionamentos em um banco de dados relacional. A utilização de chave estrangeira impõe restrições, que devem ser garantidas, ao executar diversas operações no banco de dados relacional, visto que ela referencia a chave primária de alguma relação. Caso seja necessário uma chave estrangeira pode referenciar a chave primária da própria relação.

Em alguns casos, mais de um atributo ou combinações de dois ou mais atributos podem servir para distinguir uma tupla das demais. Um dos atributos é escolhido como chave primária, os demais atributos ou combinações de atributos são denominados chaves alternativas.

O conjunto básico de operações para o modelo relacional é a álgebra relacional. A álgebra relacional provê um fundamento formal para operações no modelo relacional, e é utilizada como base para implementar e otimizar as consultas em sistemas de gerenciadores de bancos de dados relacional.

## **2.2. Álgebra Relacional**

A álgebra relacional é uma linguagem de consulta procedural, que consiste em um conjunto de operações que recebe como entrada uma ou duas relações, e produz, como resultado, uma nova relação. A álgebra relacional é composta por seis operações fundamentais, que são: seleção, projeção, união, diferença, produto cartesiano e renomear. Além das operações fundamentais, existem outras operações - adicionais e estendidas – que não serão descritas neste capítulo.

As operações seleção, projeção e renomear operam uma única relação, e por isso são denominadas como operações primárias. As outras

operações fundamentais são chamadas de operações binárias, pois operam um par de relações (SILBERSCHATS, 1997).

A operação de seleção, como o nome sugere, seleciona tuplas que satisfaçam um determinado predicado. Ela é denotada pela letra grega minúscula sigma ( $\sigma$ ). O predicado aparece subscrito a letra que denomina a operação, e o argumento da relação é dado entre parênteses, seguindo a mesma letra. Em geral podemos usar comparações do tipo  $=$ ,  $\neq$ ,  $<$ ,  $\leq$ ,  $>$  e  $\geq$  no predicado das operações. Além do mais, podemos combinar vários predicados, em um único predicado, utilizando os conectivos e ( $\wedge$ ) e ou ( $\vee$ ). A seleção pode ainda comparar dois ou mais atributos.

A operação projeção, denotada pela letra grega pi ( $\pi$ ), retorna o argumento da relação, descartando certos atributos. Na operação de projeção, quaisquer linhas em duplicidade são eliminadas. Listamos, subscritos a letra pi, os atributos que desejamos no resultado, e os atributos que não listados são descartados pela operação. O argumento da relação vem entre parênteses, a seguir.

A operação renomear, denotada pela letra grega rho ( $\rho$ ), permite-nos dar um nome ao resultado de uma expressão em álgebra relacional, visto que o resultado de uma expressão em álgebra relacional não possui um nome que nos permita referenciá-la.

A operação produto cartesiano, denotada por um ( $\times$ ), permite que combinemos as informações de duas relações quaisquer. O produto das relações  $r_1$  e  $r_2$  é representado por  $r_1 \times r_2$ . Esta operação constrói uma nova relação a partir das duas relações especificadas. A nova relação que será construída é constituída de todas as possibilidades de pares de tuplas concatenadas, de cada uma das relações.

A operação de união, denotada por ( $\cup$ ), constrói uma nova relação consistindo em todas as tuplas que aparecem em ambas as relações especificadas. Para uma operação de união  $r \cup s$  ser válida são necessárias duas condições: As relações devem possuir o mesmo número de atributos e os domínios do  $i$ -ésimo atributo de  $r$  e o  $i$ -ésimo atributo de  $s$  devem ser os mesmo para todo  $i$ .

A operação diferença, denotada por ( $-$ ), permite-nos encontrar as tuplas que estão em uma relação, mas não em outra. Esta operação constrói uma

nova relação consistindo em todas as tuplas que aparecem na primeira, mas não na segunda, do par de relações especificadas.

### 3. Trabalhos Relacionados

Este capítulo tem como objetivo apresentar uma breve análise sobre os softwares de consultas de álgebra relacional, encontrados. Foram analisados dois softwares, o primeiro, RELATIONAL (2007), desenvolvido pela *FREE SOFTWARE FOUNDATION, INC.* O segundo o LEAP (LEYTON; HILL, 1995), foi desenvolvido na Escócia. Os dois *softwares*, analisados apresentaram pontos positivos e negativos, quando analisados segundo suas funcionalidades e sua usabilidade.

#### 3.1. RELATIONAL

O *software RELATIONAL* (2007) suporta todas as operações fundamentais da álgebra relacional (seleção, projeção, união, intersecção, produto cartesiano e renomeação), diferença, junção e junção natural. Além destas operações de consulta, o *RELATIONAL* permite criação, edição e remoção das relações.

A interface do programa é bem simples e intuitiva, conforme mostrado na Figura 1, os operadores são separados em botões, e adotam a sintaxe e os símbolos da álgebra relacional, utilizada na maioria dos livros de bancos de dados. Através da interface também é possível acessar, e criar ou excluir novas relações. O *software* exibe os atributos das relações carregadas, e exibe um histórico com as consultas realizadas. Por fim, o sistema permite acesso a outras telas, que apresentam o histórico do programa, e uma ajuda, muito eficiente e explicativa, que ensina os conceitos básicos da álgebra e da utilização do *software*.

Apesar da interface bem elaborada, intuitiva e auto-explicativa, alguns de seus componentes não funcionam, ou não produzem o resultado desejado. Toda a ferramenta também é produzida em inglês, o que faz com que usuários que não compreendam a língua, encontrem dificuldades em utilizá-la.

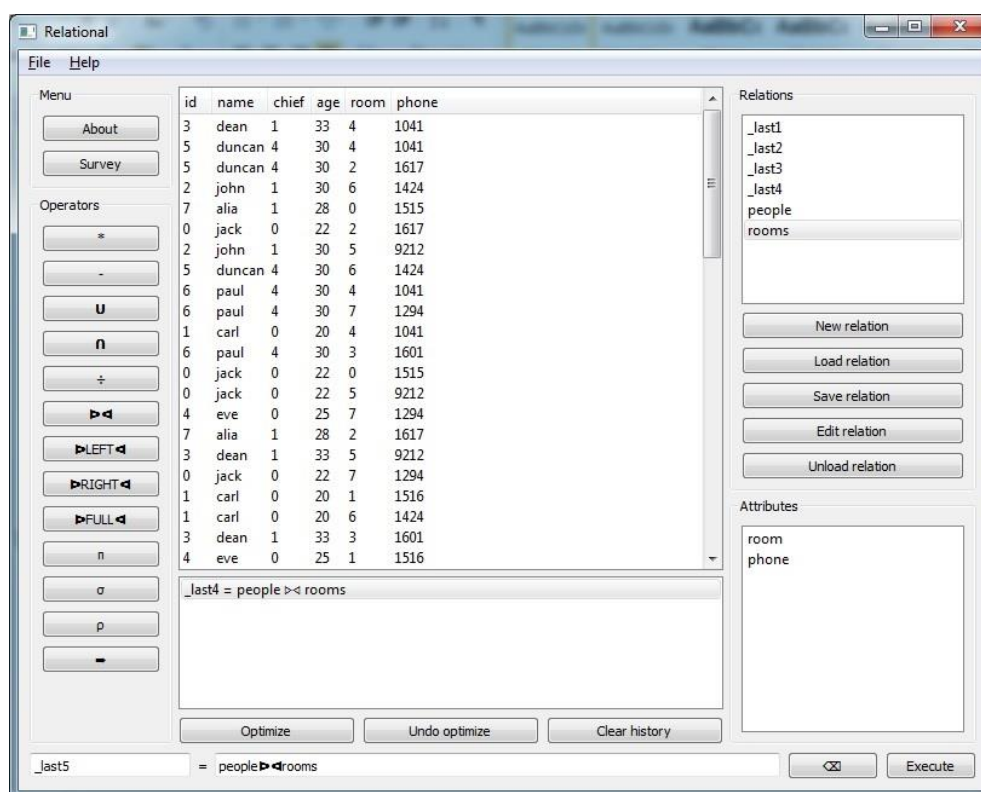


Figura 12 - Interface do sistema RELATIONAL

Fonte: RELATIONAL, 2007

### 3.2. LEAP

O *software LEAP* suporta as operações fundamentais da álgebra relacional, diferença, junção e junção natural. Além destas operações de consulta, ele permite também a criação, edição e remoção de tabelas e bancos de dados.

A interface do LEAP é totalmente textual, conforme ilustrado na Figura 2, o que faz com que o usuário seja obrigado a ler a documentação do sistema antes de utilizá-lo, para que se conheça qual o formato de entrada do *software*, e quais operações ele suporta. O sistema não adota a sintaxe e a simbologia da álgebra relacional, presente na maioria dos livros de banco de dados. Porém após a leitura da documentação a utilização da ferramenta é mais simples. O software possui comandos para listar o conteúdo de determinada tabela, listar as tabelas presentes no banco, realizar consultas em álgebra relacional, e alterações nas tabelas.



A ferramenta também permite ao usuário algumas alterações em suas funcionalidades, para que ele possa, por exemplo, acompanhar o que está acontecendo durante a consulta.

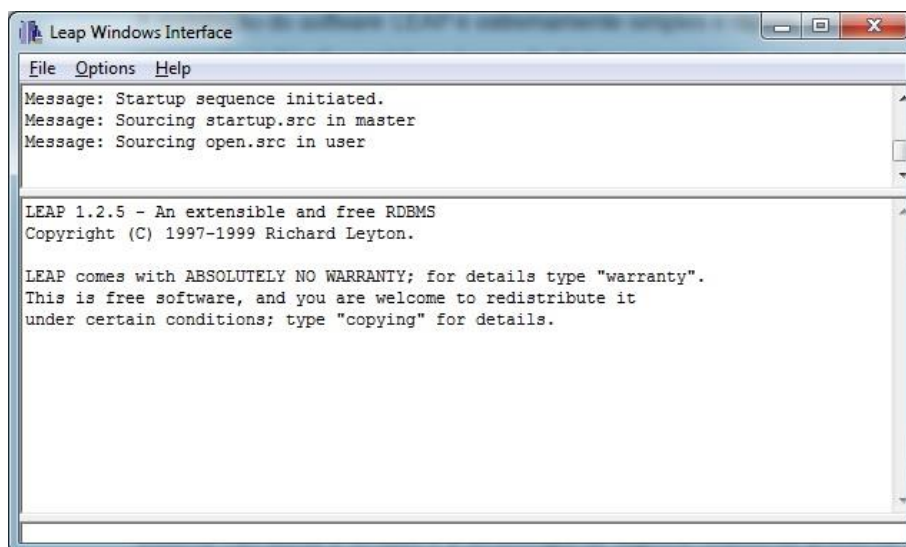


Figura 13 - Interface do Sistema LEAP

Fonte: LEAP, 1995.

### 3.3. Comparação entre o LearnAlgTool e os softwares relacionados

Na Tabela 20 é apresentada uma comparação entre as principais características, desejáveis no software proposto e presente nos softwares Relational e LEAP.

Tabela 20 - Comparação entre os softwares

	Relational	LEAP	LearnAlgTool
Não necessita Instalação			X
Armazena Historico de Consultas	X	X	X
Simbologia e Sintaxe Comum	X		X
Suporte a Operações Básicas	X	X	X
Interface Clara e Intuitiva	X		X
Gerenciar Relações	X	X	X
Avaliação e Relatório das Consultas Realizadas			X

## 4. Projeto LearnAlgTool

Este capítulo tem como objetivo apresentar os conceitos iniciais da ferramenta LearnAlgTool. Neste capítulo é apresentado o diagrama de casos de uso inicial juntamente com os requisitos, além do projeto inicial de banco de dados e das tecnologias utilizadas no desenvolvimento da ferramenta.

### 4.1. Diagrama de Casos de Uso

O diagrama de casos de uso possibilita a compreensão do comportamento externo do sistema por intermédio de uma perspectiva do usuário. Trata-se de um diagrama UML mais abstrato, que representa os papéis desempenhados pelos diversos usuários do sistema.

Na Figura 14, são apresentados os principais casos de uso do sistema LearnAlgTool. Estes casos de uso apresentam os atores e os recursos que ele pode utilizar.

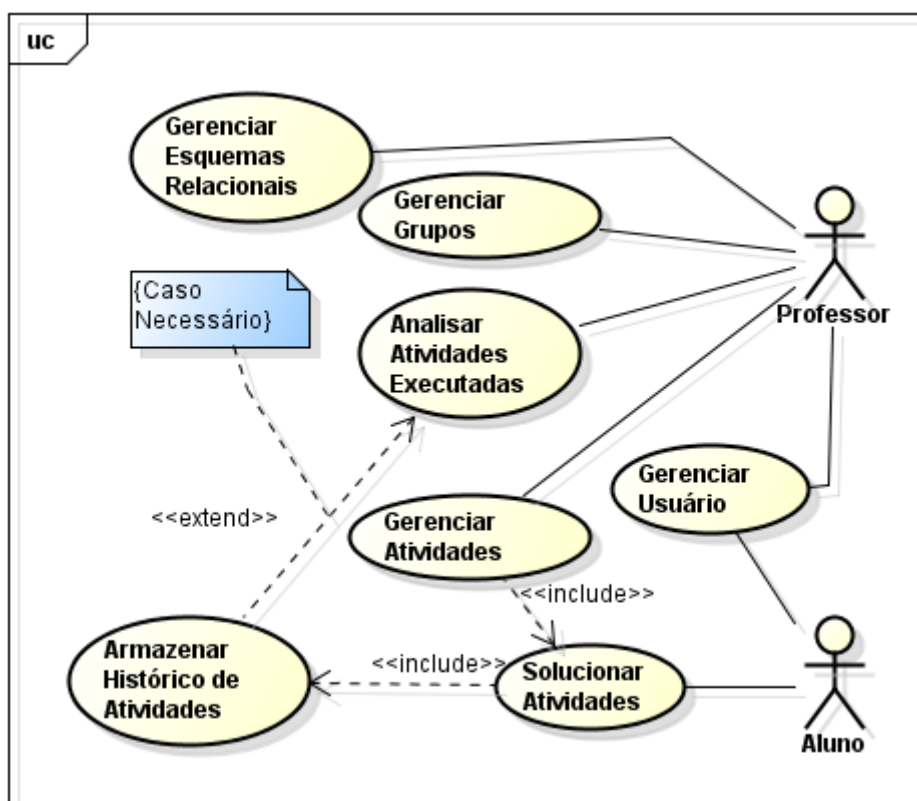


Figura 14 - Casos de Usos do sistema LearnAlgTool

## 4.2. Requisitos

Os requisitos do sistema LearnAlgTool, fornecem as informações necessárias para o projeto e o desenvolvimento, ditando quais são as funcionalidades do sistema. Através deles também são realizados os testes, visando sempre buscar o seu correto funcionamento.

Os requisitos funcionais indicam como o sistema deve operar, especificam as entradas e as saídas do sistema e o relacionamento comportamental entre elas, assim como a iteração com o usuário.

Os requisitos não funcionais são aqueles que impõem as restrições no produto a ser desenvolvido e algumas vezes no processo de desenvolvimento, também especificam as restrições externas que o produto deve atender.

### 4.2.1. Requisitos Funcionais

RF01 – O sistema deverá permitir o gerenciamento de usuários;

RF02 – O sistema deverá permitir a organização dos alunos em grupos;

RF03 – O sistema deverá permitir a geração de esquemas relacionais (relações) e a inserção de dados nestas relações;

RF04 – O sistema deverá permitir o gerenciamento de atividades;

RF05 – O sistema deverá permitir a resolução das atividades;

RF06 – O sistema deverá permitir a realização de consultas utilizando álgebra relacional;

RF07 – O sistema deverá armazenar um histórico de consultas realizadas, e atividades solucionadas;

RF08 – O sistema deverá exibir as estatísticas de erros e acertos dos usuários, e das questões propostas;

RF09 – O sistema deverá permitir a autenticação dos usuários;

RF10 – O sistema deverá permitir a realização de consultas livres (sem estar associada a atividades do sistema);

### 4.2.2. Requisitos não funcionais

RNF01 – O sistema deve possuir interface clara e intuitiva.

RNF02 – O sistema deve ser *Web*.

RNF03 – O sistema deve funcionar nos principais navegadores modernos

RNF04 – O sistema deve ser rápido e fornecer resposta imediata a para o usuário

### 4.3. Projeto de Banco

O modelo entidade e relacionamento é definido como um modelo baseado na percepção do mundo real, que consiste em um conjunto de objetos básicos chamados entidades e nos relacionamentos entre esses objetos.

Na Figura 15 é apresentada o modelo entidade e relacionamento do sistema LearnAlgTool, desenvolvido utilizando o *software* Astah, de forma abstrata e simplificada, seu principal objetivo é facilitar o projeto de banco de dados, possibilitando a especificação da estrutura lógica do banco de dados.

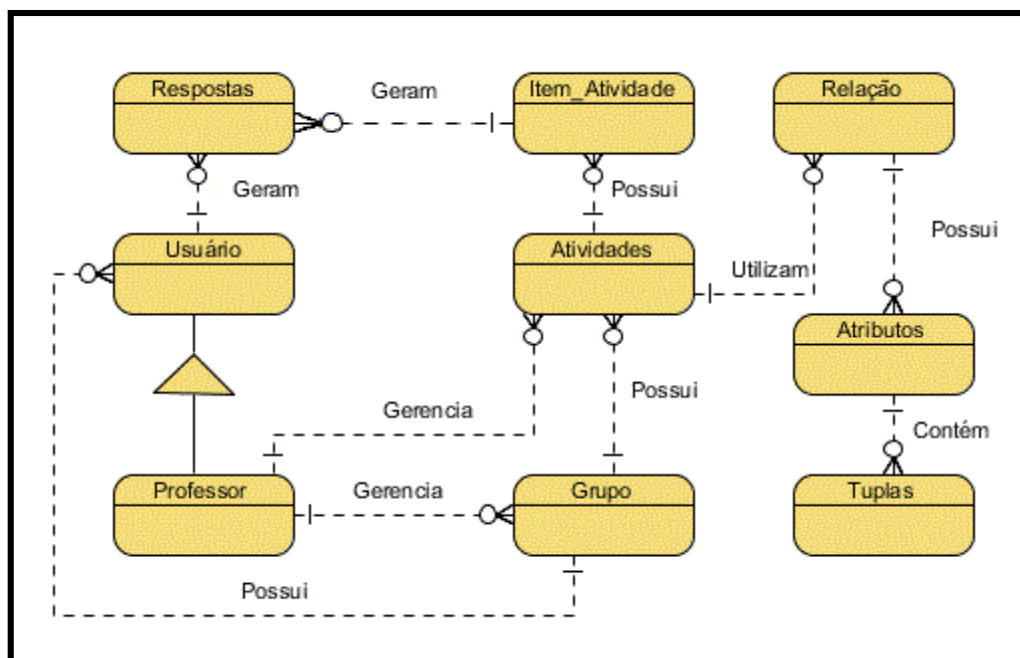


Figura 15 - Modelo Entidade e Relacionamento

#### 4.4. Tecnologias Utilizadas

Neste capítulo são apresentadas as tecnologias e ferramentas que serão utilizadas durante o desenvolvimento do sistema LearnAlgTool, conforme apresentado na Tabela 21. Espera-se que sua utilização facilite o desenvolvimento e ajude a satisfazer os requisitos propostos.

Tabela 21 - Tecnologias Utilizadas

Tecnologia	Descrição
PHP	Linguagem interpretada que realizará o processamento no lado do servidor.
HTML	Linguagem de marcação utilizada pra produzir páginas web.
Java Script	Linguagem de programação interpretada.
CSS	Linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como por exemplo, o HTML.
UML	Linguagem visual utilizada para modelar <i>softwares</i> baseados no paradigma a orientação a objetos.
Astah	<i>Software</i> utilizado para geração de todos os diagramas da UML.
NetBeans	IDE de desenvolvimento.
Postgresql	Servidor de banco de dados relacional.
Apache	Servidor Web

Fonte: Autoria Própria

## 5. Cronograma

Este capítulo apresenta o cronograma inicial do sistema LearnAlgTool, (Tabela 22), no qual, são especificadas os meses, para o início e fim, das fases do projeto.

As fases iniciais do LearnAlgTool foram divididas em:

- Concepção – Fase que compreende a discussão com o professor orientador, a fim de analisar a viabilidade do projeto;
- Análise– Fase que compreende a análise dos requisitos para o desenvolvimento do sistema;
- Projeto – Fase que compreende o projeto das telas do sistema, do SGBD;
- Desenvolvimento – Fase que compreende o desenvolvimento das funcionalidades do sistema proposto e de sua interface, juntamente com os testes unitários;
- Avaliação – Fase que compreende a resolução de erros e problemas encontrados no sistema, além da manutenção de suas funcionalidades, afim de que o sistema continue executando corretamente, e conforme as especificações.

Tabela 22 - Cronograma inicial

FASES	2013				2014					
	Set.	Out.	Nov.	Dez.	Jan.	Fev.	Mar.	Abr.	Mai.	Jun.
Concepção										
Análise										
Projeto										
Desenvolvimento										
Avaliação										

Fonte: Autoria Própria

As atividades serão iniciadas em setembro de 2013, e espera-se que sejam encerradas em Junho de 2014. Algumas fases ocorrem simultaneamente, buscando acelerar o desenvolvimento, e permitir o cumprimento do cronograma proposto.

## 6. Considerações Finais

A análise das ferramentas que permitem a consulta utilizando álgebra relacional, constatou que uma nova ferramenta necessita ser implementada. As ferramentas analisadas LEAP e RELATIONAL possuem características que dificultam sua utilização, como interface não intuitiva, sintaxe e simbologia diferente da utilizada pelos principais autores dos livros de banco de dados, tratamento de erros inadequado, idioma em inglês, componentes da interface que não funcionam ou não produzem o resultado desejado e um tratamento inadequado dos erros.

A ferramenta LearnAlgTool é proposta, tendo como objetivos principais a criação de uma ferramenta web, de fácil utilização, que auxilie no aprendizado de álgebra relacional. A segunda característica da ferramenta que merece ser destacada está no fato que ela utiliza simbologia e sintaxe comum, aos principais livros de banco de dados, e interface intuitiva, que facilita sua utilização. Por fim, como terceira característica, vale destacar que por se tratar de um sistema web, a ferramenta LearnAlgTool não necessita de instalação na máquina do usuário, o que diminui a resistência dos usuários em utilizá-la. Espera-se que estas características e vantagens façam do LearnAlgTool uma ferramenta empregada para auxiliar no ensino e aprendizagem de álgebra relacional, nos cursos de computação.

## 7. Referências Bibliográficas

Date. C. J. **Introdução a Sistemas de Banco de Dados**. 4. ed. Rio de Janeiro: Campus, 1990.

Silberschatz. Abraham. **Sistema de Banco de Dados**. 3. ed. São Paulo: Makron Books Ltda, 1997.

\_\_\_\_\_. **Sistema de Banco de Dados**. 5. ed. Rio de Janeiro: Campus, 2006.

Elmasri. Ramez. **Sistema de Banco de Dados**. 4. ed. São Paulo: Pearson Education do Brasil, 2005.

Kroenke. David M. **Banco de Dados – Fundamentos, Projeto e Implementação**. 6. ed. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 1999.

Heuser. Carlos Alberto. **Projeto de Banco de Dados**. 4. ed. Porto Alegre: Sagra Luzzatto, 1999.

RELATIONAL. **Um compilador de álgebra relacional em python**. Disponível em:

<<http://galileo.dmi.unict.it/wiki/relational/doku.php>> Acesso em: 07 abr. 2013.

NetBeans IDE. **A maneira mais inteligente e rápida para o código**. Disponível em:

<<http://netbeans.org/>> Acesso em: 07 abr. 2013.

PHP. **O que é PHP**. Disponível em: <<http://php.net/>> Acesso em: 07 abr. 2013.

Universidade Tecnológica Federal do Paraná. **Normas para elaboração de trabalhos acadêmicos**. Curitiba: Editora da UTFPR, 2008.