CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS – CAMPUS V

Relatório do Trabalho de Conclusão de Curso

Vitor Bonfim de Castro Lima

Divinópolis - MG

2016

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS – CAMPUS V

API-DER

Vitor Bonfim de Castro Lima

Orientador: Thiago Magela Rodrigues Dias

Co-Orientador: Michel Pires Silva

Relatório do Trabalho de Conclusão de Curso apresentado ao Curso Técnico de Informática do Centro Federal de Educação Tecnológica de Minas Gerais – Campus V como requisito parcial para a obtenção do título de Técnico em Informática.

Divinópolis

2016

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS – CAMPUS V

Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Técnico em Informática e aprovado pela banca composta pelos seguintes professores.

_____

Prof. Thiago Magela Rodrigues Dias – CEFET-MG (Orientador)

_____

Prof. Michel Pires da Silva - CEFET-MG (Co-orientador)

_____

Prof. Vinícius Larêdo Henrique Duarte - CEFET-MG

_____

Prof. Luís Augusto Mattos Mendes

Coordenador do Curso Técnico em Informática

RESUMO

Ferramentas que apresentam como objetivo o auxílio de aprendizagens em conteúdos didáticos vêm se apresentando, cada vez mais, uma opção para a dialética de professores. No entanto, em uma matéria específica da área da Informática, Banco de Dados, há muito poucos recursos voltados para este foco em particular. Tomando tal fato como ponto de partida, foi decidido desenvolver uma aplicação que agrega na gama de softwares relacionados a esse objetivo, e que venha ser uma ferramenta diferenciada em relação às outras já conhecidas. Desta maneira, a proposta aqui apresentada, expõe um software voltado para a construção de diagramas específicos, estudados na matéria de Banco de Dados, Diagramas de Entidade e Relacionamento, exibindo certas funcionalidades para desenhar tais diagramas com sucesso. Os modelos de diagramação criados pela aplicação seguem sempre uma notação padrão e específica já existente e criada por um especialista do ramo, diferentemente das outras ferramentas que seguem uma notação mais genérica.

Portanto, o objetivo proposto nesse projeto é suprir essa necessidade na área abordada, sendo mais uma alternativa na dialética de um professor, por exemplo.

Palavras-chaves: DER; Aplicação; Ferramenta;

SUMÁRIO

1. Introdução

Em um contexto didático, banco de dados é uma disciplina técnica da área da informática. Dentro dessa disciplina existe um conteúdo relacionado à modelagem de dados, onde se aprende, inicialmente, fazer modelos de banco, por meio de Diagramas de Entidade e Relacionamento (DER) (DATE, 2004). Para auxiliar na dialética dessa matéria específica, existem algumas ferramentas que são desenvolvidas com esse propósito. Nesse cenário, apresenta-se como tais ferramentas o brModelo ou o DataToadModeler, da Dell Inc., sendo o primeiro voltado mais para DER's, ou seja, mais conceitual; já o Toad, é um software de modelagem, mais prático e profissional. Nesse mesmo ambiente, que é o inserido pelo software da Dell, ainda há várias ferramentas alternativas para esse fim, como o Workbench SQL. Já para a didática de diagramas de entidades e relacionamentos, a carência de uma ferramenta para essa proposta ainda se apresenta mais evidente, visto que o brModelo contém vários bugs e erros em sua execução e não apresenta uma interação que auxilia a assimilação dos conceitos deste conteúdo.

Nesse projeto, é apresentado um software que foca em auxiliar a dialética do professor e contribuir para uma melhor compreensão dos fundamentos do conteúdo abordado. Desenvolvido na linguagem de programação Java que permite que a aplicação rode em qualquer plataforma, bem como se propõe uma ferramenta com fins em auxílio didáticos. O nome do projeto, API-DER, faz referência à dois elementos que seria implementados: API (Application Programming Interface); e os diagramas de entidade e relacionamento (DER). Porém, a ideia da elaboração de uma API foi descartada em função da complexidade do desenvolvimento em relação ao prazo de entrega do projeto, sendo produzido, então, um software normal. Quanto ao nome da ferramenta, continuou sendo API-DER por uma questão burocrática da instituição.

1.1. Referencial do Projeto

Após uma análise da grande área que o projeto se insere, foi evidenciada uma dificuldade relatada por professores que lecionam estas disciplinas na instituição. O projeto, portanto, apresenta como proposta de TCC, o desenvolvimento de uma aplicação para o uso didático por um professor, por exemplo, com fins para a criação e diagramação de modelos de DER de banco de dados.

1.1.1. Softwares Semelhantes

Abaixo temos exemplos de softwares semelhantes ao desenvolvido.

O brModelo, software já mencionado acima, é um programa que faz exatamente modelos DER's de acordo com uma notação específica. Apresenta mesmas funcionalidades, porém a notação representada por ele é baseada em outro autor.

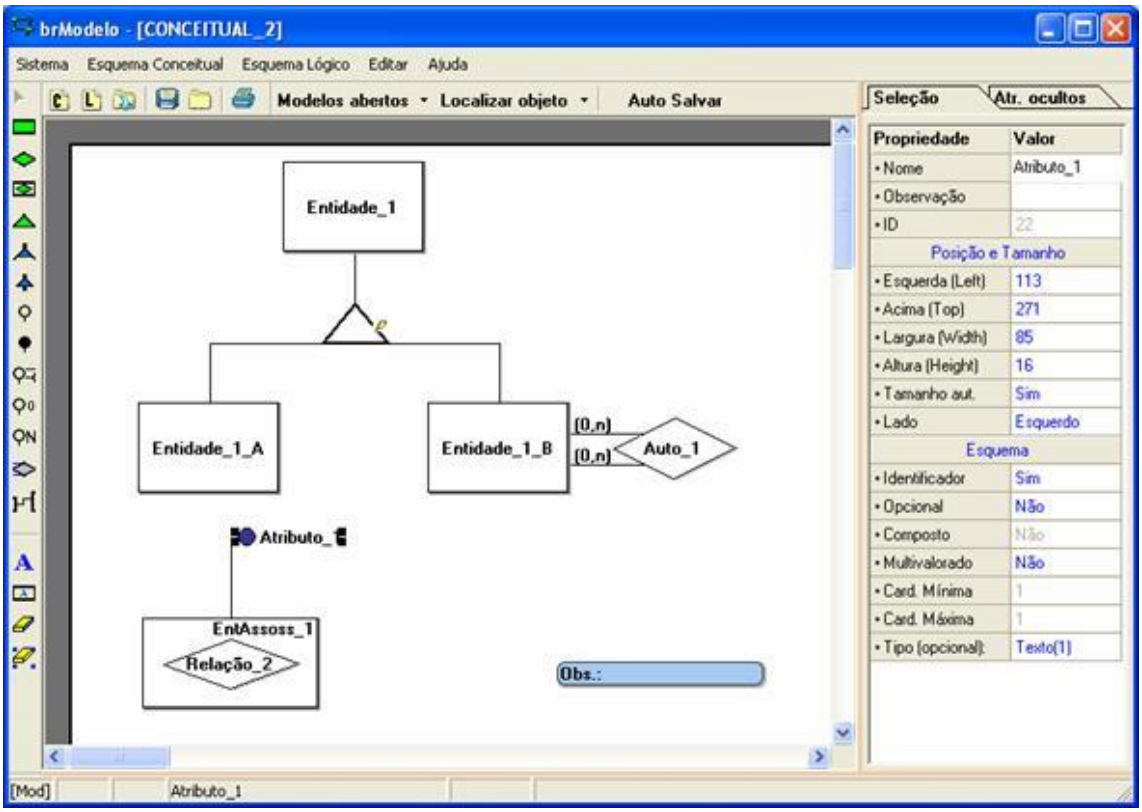


FIGURA 1. Exemplo do software brModelo

Este software apresenta vantagens na forma de variados recursos para a diagramação de entidade e relacionamento, como, permitir alterações estruturais no modelo a partir de novas decisões do usuário. Isso seria, por exemplo, converter um atributo em entidade ou relacionamento em entidade associativa.

Além disso, o programa conta com a possibilidade de adicionar atributos e definir seus tipos como Identificador, Multivalorado e Composto, como mostra a imagem seguinte de um DER reproduzido através do aplicativo.

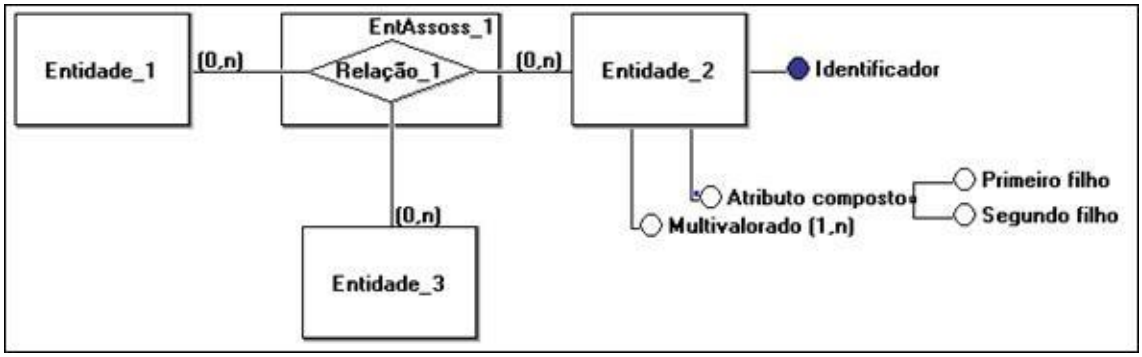


FIGURA 2. DER feito no brModelo com todos os tipos de atributo

Por esses motivos, o projeto tem fortemente como base este software.

Também é possível apresentar como um exemplo de um software semelhante, o DBDesigner, desenvolvido a partir de freewares com código fonte para a plataforma Windows ® e Linux (GNU GPL). Como uma ferramenta para desenhos e diagramação, apresenta recursos completos, no entanto, seu principal problema é o fato de não implementar um modelo conceitual, o que a torna quase sem utilidade como ferramenta didática.

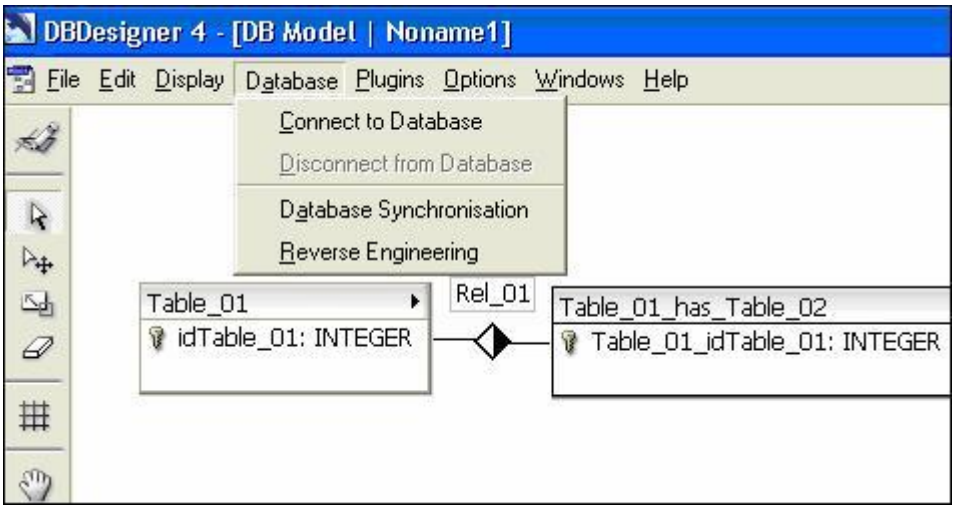


FIGURA 3. Tela do DBDesigner

Há mais softwares como esses, com finalidades parecidas, porém com suas particularidades. Um exemplo é o Microsoft Visio, voltada exclusivamente para desenhos, sem interação com banco de dados. Os seus esquemas de desenhos variam de acordo com a notação do diagrama carregado, diagramas estes que vem pré-definidos na aplicação.

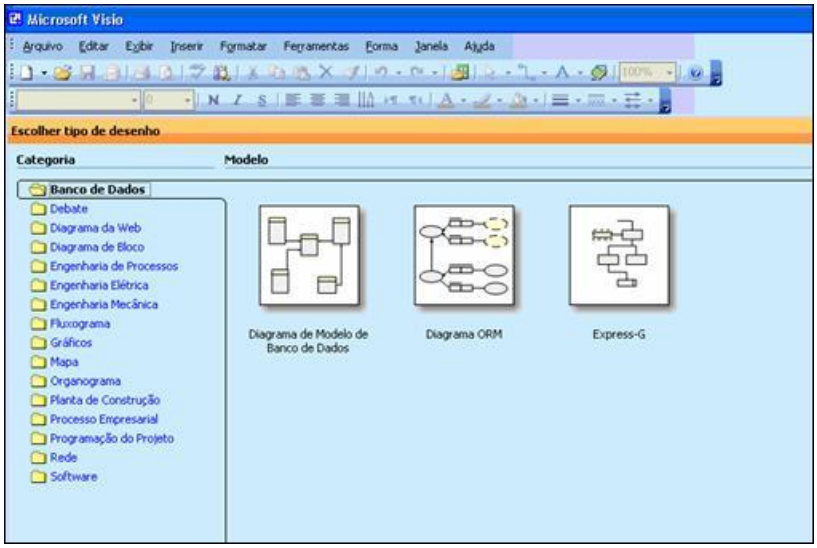


FIGURA 4. Tela inicial do Microsoft Visio

Apesar de a ferramenta lidar particularmente com diagramação e desenhos de modelagem de banco de dados, ela ainda não apresenta implementação dos conceitos de Diagrama de Entidade e Relacionamento, sem atuar de nenhuma maneira na construção destes.

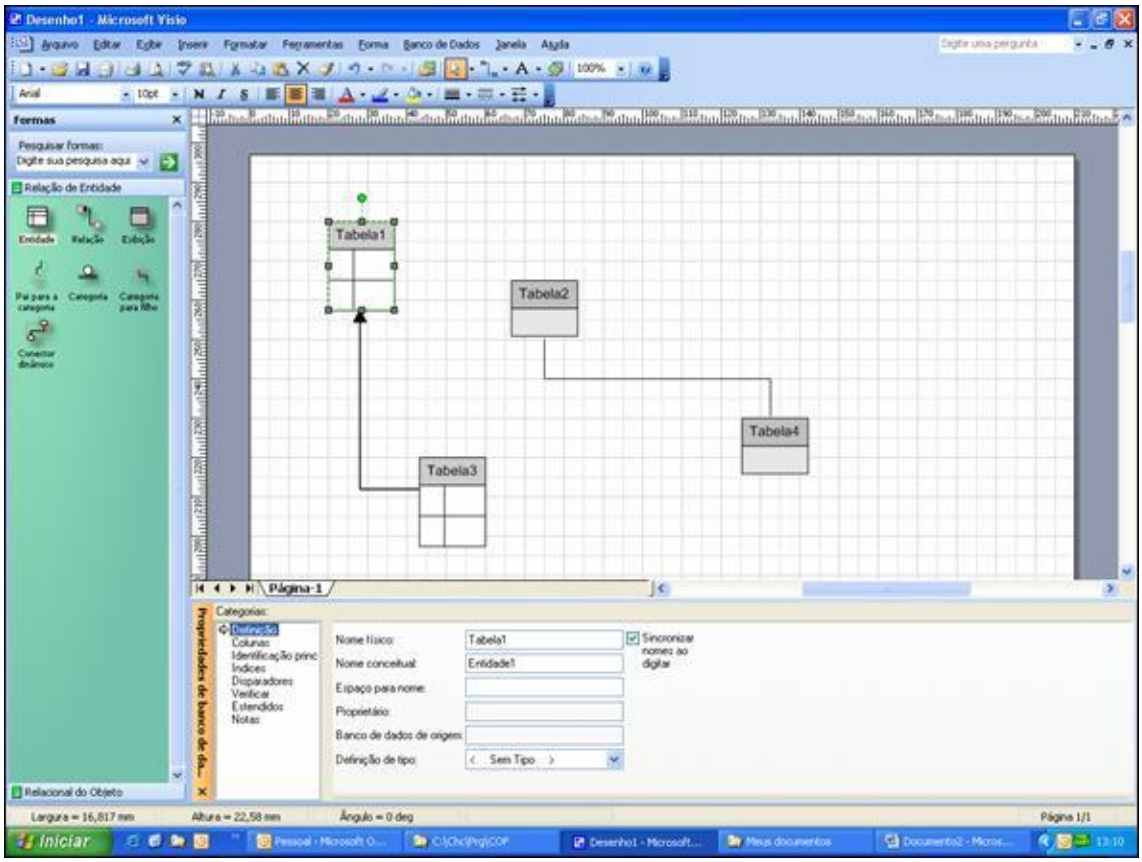


FIGURA 5. Diagramas conceitual e lógico no Microsoft Visio

1.2 Definição da Aplicação

A aplicação, aqui proposta, tem como objetivo fazer um DER por meios de recursos de desenho para a didática desta matéria da disciplina de Banco de Dados, que poderá ser usada por um professor e também por seus alunos. A aplicação, portanto, permite criar DER's de forma mais prática e simples possível, para o fácil e rápido entendimento dos conceitos da modelagem dos diagramas. Dessa forma, apresenta recursos de inserir atributos em entidades e seus relacionamentos, ou mesmo focando somente no relacionamento de entidades, descartando seus atributos. De tal maneira que se torna uma ferramenta perfeita para o cenário didático.

Essa aplicação terá como base a notação de DER proposta por Carlos Alberto Heuser.

1.3 A Notação de Heuser

A notação de Heuser proposta para DER, é uma notação simples e prática, por esse motivo foi adotada para esse projeto, com o auxílio dos orientadores. Historicamente, essa notação de Diagramas de Entidade e Relacionamento, se baseou em uma notação já criada por um dos autores pioneiros na criação de uma notação padrão para essa modelagem (HEUSER, 2001). A notação de Heuser, portanto, é baseada nos conceitos básicos de diagramação de entidade e relacionamento, numa tentativa de aperfeiçoar e melhorar a obra do Dr. Chen (HEUSER, 2001). Abaixo tem-se um exemplo dessa notação.

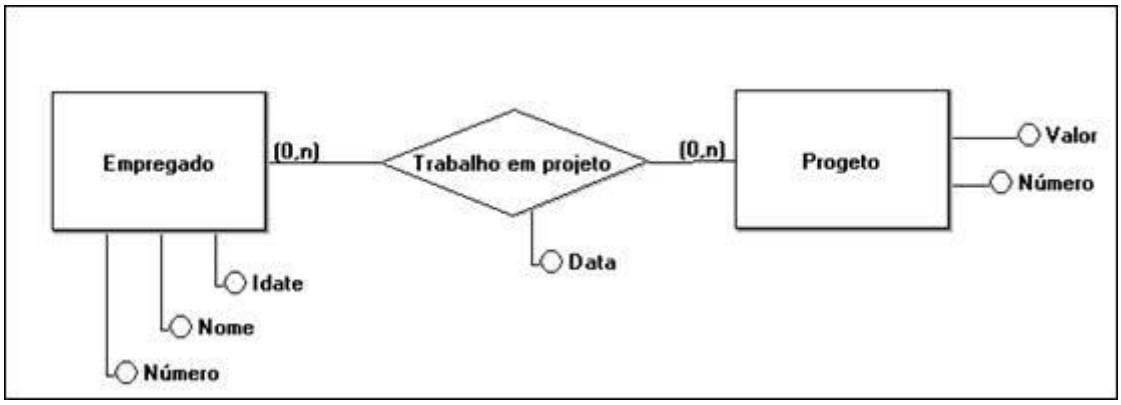


FIGURA 6. Notação de Heuser com atributos - Adaptada (2001)

Percebe-se, diante da notação que a representação dos conceitos do DER é representado da seguinte maneira:

- Entidade: Retângulo (Empregado e Projeto)

- Relacionamento: Ligado por uma aresta das entidades relacionadas à um losango que pode ser uma entidade fraca.
- Cardinalidade: A cardinalidade é notada pelos números que cada entidade participa do relacionamento. Exemplo: (0,1), (0,n), (1,1), (1,n) ou (n,n).
- Atributos: Os atributos são pequenos círculos ligados por arestas às entidades que elas pertencem.

1.4 Outras Notações Estudadas

Ao longo dos estudos e pesquisas para a realização do projeto apresentado, também foram analisadas outras notações de modelos de diagramas de entidade e relacionamento. Foram estudas, portanto, notações de outros autores, como as do Navathe e Elmasri. A Figura 7, traz um exemplo de um DER construído com base nessas notações.

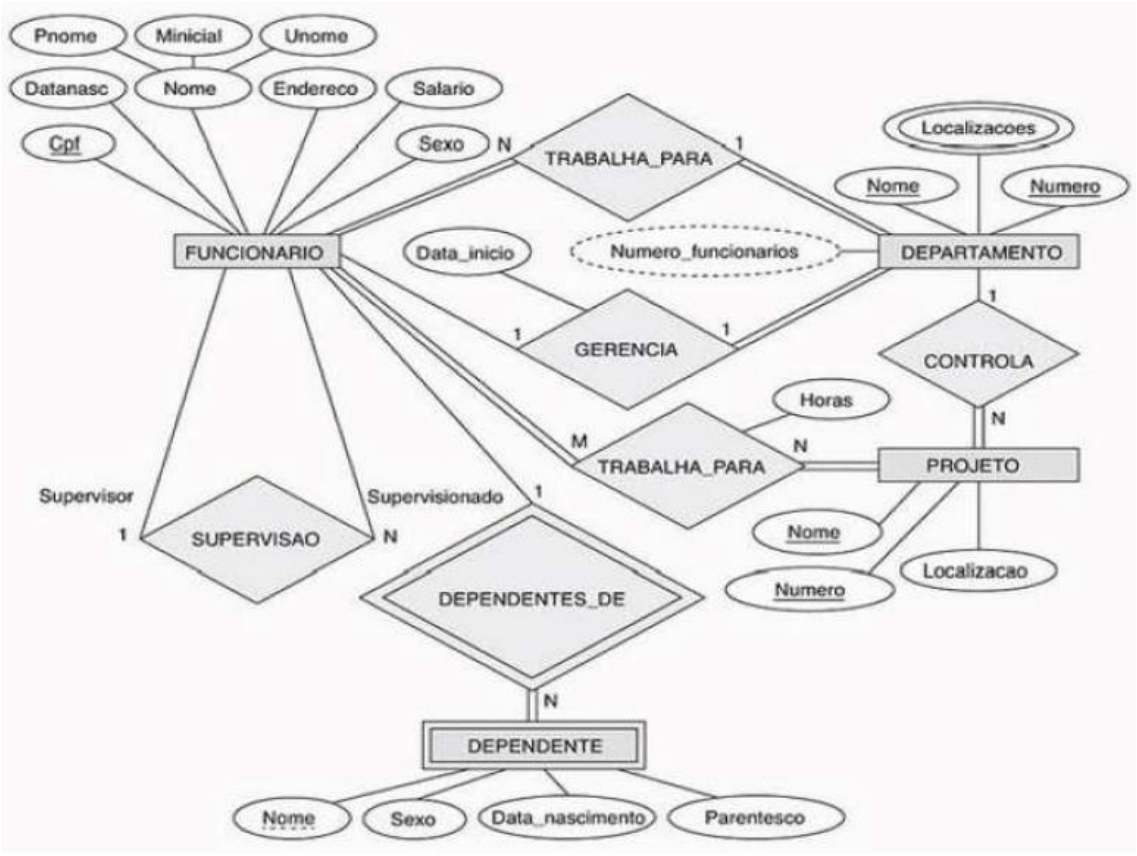


FIGURA 7. Exemplo de um DER com a notação de Navathe e Elmasri, 2011

Ao comparar essa notação, com a notação de Heuser, percebe notáveis diferenças. Por exemplo, os atributos são demonstrados através de uma elipse com seu nome dentro, nessa notação (ELMASRI, 2011), já na do Heuser, apresenta apenas um pequeno círculo na

extremidade da aresta que sai da entidade à qual o atributo pertence (HEUSER, 2001). Essa notação, também apresenta algumas particularidades, como as representações dos tipos de atributos, como atributos multivalorados, derivados, chaves primárias e parciais (ELMASRI, 2011). A Figura 8 demonstra o significado correspondente à cada símbolo da notação.

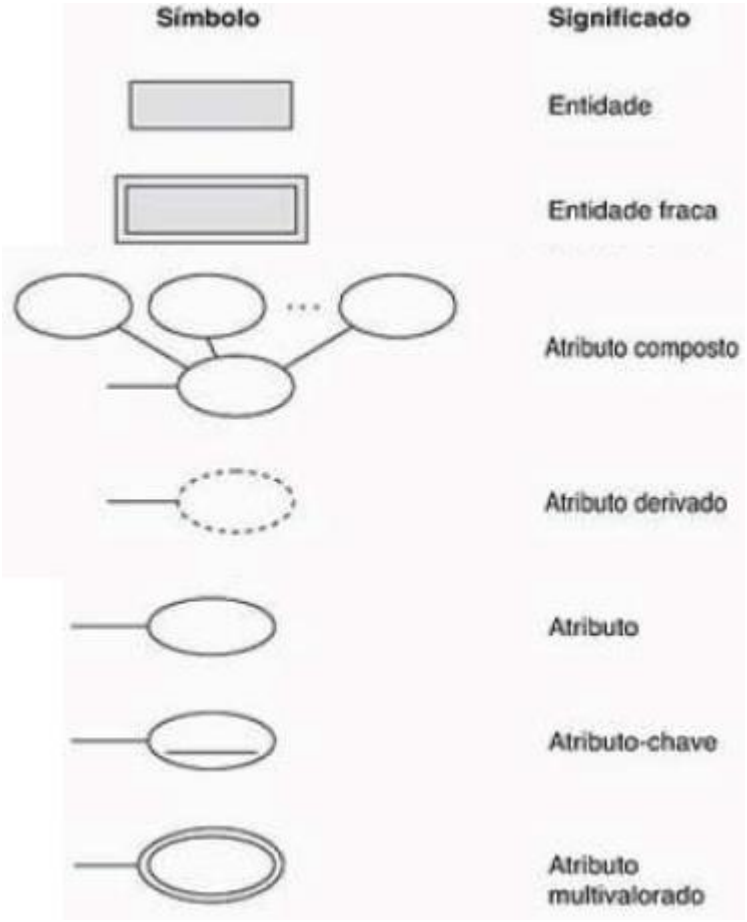
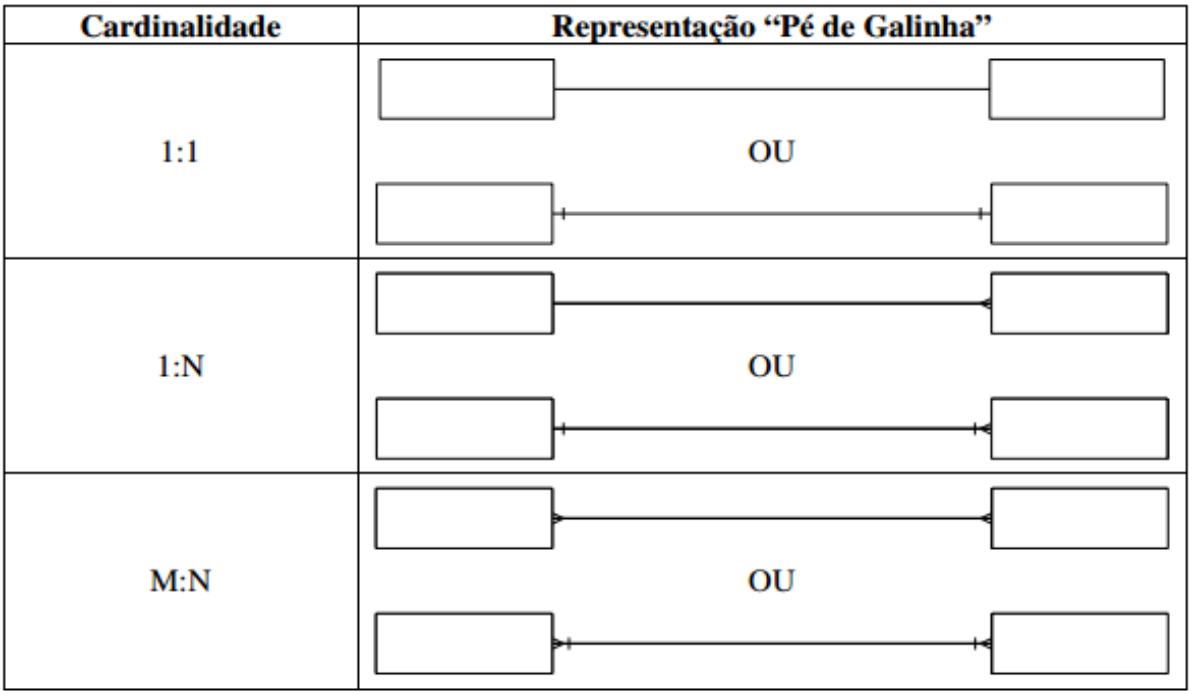


FIGURA 8. Significado dos símbolos da notação DER de Navathe

Além dessa notação, também foi estudada a notação do autor James Martin. Esta, por sua vez, se diferencia das outras, pela forma de representar o relacionamento entre as entidades, o que pode ser observado na Figura 9. Fazendo a análise da figura e comparando com as outras notações, percebe que a forma com que James Martin sugere a representação dos relacionamentos se dá, apenas, por meio de linhas, excluindo o losango, como representado nas notações anteriores (MARTIN, 1990). Para, então, demonstrar, a cardinalidade da relação, as linhas que representa o relacionamento conta com umas alterações em suas extremidades, determinando o tipo da cardinalidade (MARTIN, 1990). Tal forma de representação, deu nome à esta notação por pé de galinha (MARTIN, 1990).



Notação 'Pé-de-Galinha' para cardinalidade

FIGURA 9. Representação dos relacionamentos dos DER's de James Martin, 1990

Além dessas notações, também foi estudada a notação de Peter Chen, na qual se baseia a de Heuser, (HEUSER, 2001). Pode observar, pela Figura 10, certa semelhança entre elas, ao analisarmos um DER construído utilizando sua notação, como na busca por uma representação do modelo lógico do banco simples e com seus componentes bem dispostos (CHEN, 1990).

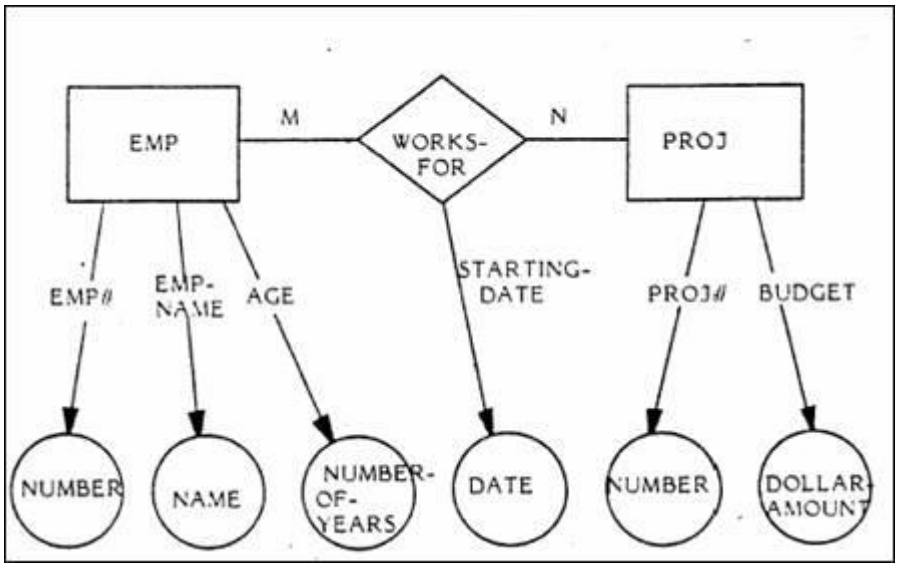


FIGURA 10. Diagrama construído utilizando notação original de Peter Chen, 1976

A forma como dispõe as entidades e o relacionamento nessa notação são muito parecidas com a notação de Heuser. No entanto, a maneira de representar os atributos da entidade recebem modificações consideráveis.

A partir das análises de todas as notações estudadas, foi determinado o desenvolvimento da aplicação baseado na notação de Heuser, devido a sua simplicidade e objetividade quanto aos conceitos gerais da modelagem de banco de dados. De modo que ao fim da elaboração do sistema, a aplicação estaria o mais próximo possível dos objetivos propostos desse projeto.

1.5 Definição das Funcionalidades

A partir do sistema desenvolvido, apresenta algumas funcionalidades, como:

- A criação de entidades no modelo DER tomando como base a notação já descrita
- A adição de atributos para cada entidade
- A ligação entre entidades, formando os relacionamentos de tipos diferentes (relacionamento 1:n, relacionamento 1:1, relacionamento n:n, por exemplo)
- A cardinalidade de cada relacionamento envolvido no diagrama.

Essas funcionalidades permite o usuário a criação do DER da forma que ele preferir. Pode ser criadas entidades com ou sem atributos, conforme o objetivo do analista. O relacionamento depende da existência de entidades, porém estas não precisam ter atributos inseridos. A disposição das entidades pode ser definida também conforme desejar o usuário.

Também é válido notar que as cardinalidades de um relacionamento são inseridas automaticamente de acordo com o tipo de relacionamento escolhido para aquela situação, assim como o tipo de atributo.

1.6. Ferramentas Utilizadas

Para o desenvolvimento desse projeto, foram utilizadas algumas ferramentas que serão mencionadas nesse tópico.

Na parte da programação da aplicação, a linguagem Java foi escolhida pela praticidade e facilidade dos desenvolvedores e também por ser uma linguagem orientada à objeto. A linguagem de programação Java também se mostra mais interessante para o desenvolvimento desse projeto por suas características como a portabilidade, ou seja, apresenta-se independente de plataforma, possui sintaxe similar com a linguagem C/C++, suporta caracteres Unicode e,

principalmente, dispor de facilidades para criação de programas multitarefas e distribuída com um vasto conjunto de bibliotecas.

Para codificação, foi usado o NetBeans, devido à familiaridade com o programa ao longo do curso. Além disso, oferece recursos necessários para a criação de aplicativos profissionais. Entre esses recursos, o NetBeans se destaca apresentando editor de código fonte integrado, com aplicações visuais com Swing que é uma aplicação; visualizador de classes integrado a interface da IDE; suporte ao Java Enterprise Edition; e pulg-ins para UML (Unifield Modeling Lenguage, em português Linguagem de Modelagem Unificada).

Nas partes de diagramação foi utilizada UML com ferramentas como o StarUML e Visual Paradigm. O StarUML é um ferramenta desenvolvida pela MKLab na linguagem Java/Eclipse, voltada com um foco principal e exclusivo em diagramas de classe, apresentando recursos que facilita essa diagramação específica. Portanto, os diagramas de classe foram desenvolvidos a partir dessa ferramenta. Já o Visual Paradigm é uma ferramenta mais completa e abrangente, que, ao longo do desenvolvimento desse projeto, foi a ferramenta mais utilizada para esta parte de modelagem e diagramação. Este software conta com ótimos recursos para seus usuários, como a geração de relatórios e capacidade de engenharia de códigos, incluindo, até mesmo, a geração de código. Ele ainda é capaz de fazer engenharia reversa de diagramas para várias linguagens de programação e trabalha com modelagem de vários tipos de diagrama (Diagrama de Classes, Diagrama de Caso de Uso, Diagrama de Sequência, Diagrama de Comunicação, Diagrama de Estado de Máquinas, Diagrama de Atividades e outros).

2. Projeto Conceitual

Neste tópico serão apresentados os documentos conceituais desenvolvidos, com o intuito de ilustrar a disposição do sistema e facilitar o entendimento do projeto. Para essa ilustração, foi usada a diagramação de caso de uso, e em sequência a explicação de cada parte deste.

2.1. Diagrama de Caso de Uso

Por meio da imagem abaixo, é apresentado o Diagrama de Caso de Uso do sistema desenvolvido. O diagrama busca especificar os atores responsáveis pela utilização do sistema e suas respectivas funcionalidades.

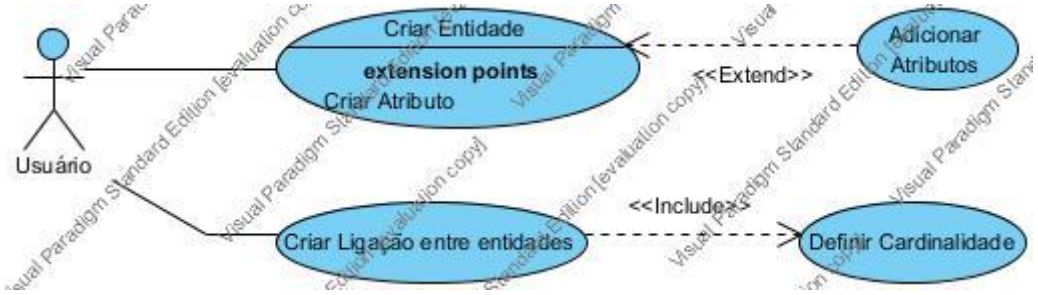


FIGURA 11. Diagrama de Caso de Uso

2.1.1. Documentação dos Autores

No diagrama de caso de uso, têm-se presentes os autores do sistema, que representam diversos usuários que irão existir dentro deste.

Nesse projeto, o ator em questão pode ser, tanto um professor, como um aluno, já que inserimos o sistema em um contexto didático, porém este ator poderia ser qualquer pessoa usuário do sistema. Apesar de termos dois possíveis atores, eles não se extinguem diante do digrama de caso de uso, pois praticam exatamente as mesmas funcionalidades.

O ator único do projeto, então, será responsável por todas as funcionalidades do sistema, que seria todas as etapas de criação e montagem dos desenhos do diagrama de entidade e relacionamento.

2.1.2. Descrição Detalhada das Funcionalidades

Para demonstrar o processo desenvolvido pelo autor, nesse tópico foram detalhadas todas as funcionalidades com suas particularidades.

2.1.2.1. Funcionalidade 1

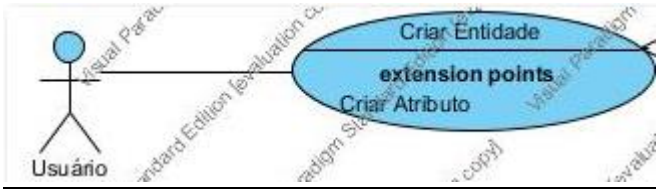


FIGURA 12. Funcionalidade 1 - Criar Entidades

Nome do Caso de Uso	
Criação de Entidades	
Ator Primário	
Professor/Aluno	
Resumo	
Nesse caso de uso, o usuário cria as entidades do modelo DER e determinará sua posição e seu nome.	
Pré-condições	
Para que ocorra essa criação, o usuário precisa apenas ter instanciado um processo da aplicação.	
Pós-condições	
O próprio Ator faria a verificação das entidades de acordo com suas preferências	
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Determina, via parâmetros, a posição das entidades e seus nomes.	2- Desenha a entidade na tela de acordo com as especificações enviadas pelo Ator, via parâmetro.

Responsável pela definição	Data da criação
Vitor Lima	09/10/2015

2.1.2.2. Funcionalidade 2

FIGURA 13. Funcionalidade 2 - Adicionar Atributos nas Entidades

Nome do Caso de Uso	
Adição de Atributos nas Entidades	
Ator Primário	
Professor/Aluno	
Resumo	
Nesse caso de uso, será possível a adição de atributos nas tabelas respectivas às entidades correspondentes, de acordo com as preferências do Ator.	
Pré-condições	
Deverá haver ao menos uma entidade criada, para que possa ser adicionado à ela o respectivo atributo.	
Pós-condições	
O atributo será adicionado à entidade.	
Fluxo Principal	
Ações do Ator	**Ações do Sistema**

1- O Ator adicionará os atributos nas tabelas das entidades já criadas na funcionalidade anterior, também, via parâmetro.	2- O Sistema desenha na tela, puxando vértices das entidades e indicando os atributos respectivos, recebidos pelo Ator via parâmetro, e representado pelo círculo de acordo com a notação padronizada e já mencionada.
Responsável pela Definição	**Data da criação**
Vitor Lima	09/10/2015

2.1.2.3. Funcionalidade 3

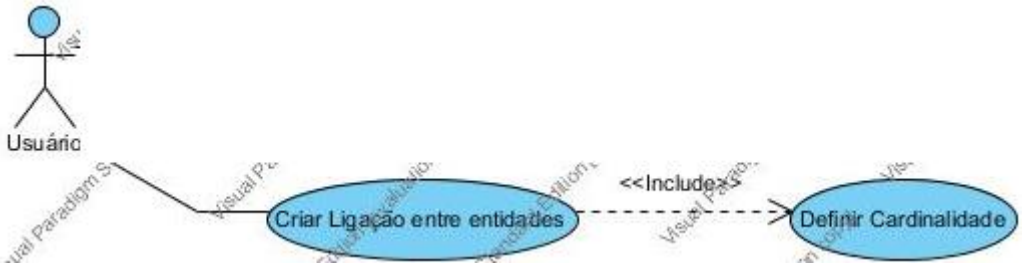


FIGURA 14. Funcionalidade 3 - Relacionamento de Entidades

Nome do Caso de Uso
Relacionamento de Entidades
Ator Primário
Aluno/Professor
Resumo
Nesse caso de uso, será estabelecido os relacionamentos entre as entidades do DER.

Pré-condições	
Para que ocorram os relacionamentos, é necessário que as entidades já estejam instanciadas.	
Pós-condições	
O relacionamento será desenhado na tela, dentro das notações do DER, a partir de vértices que saem das entidades relacionadas e encontram-se num losango que poderá também ser uma entidade, dependendo do tipo de relacionamento.	
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Inicialmente, o Ator deve identificar o tipo de relacionamento que será usado. 2- Após a identificação do tipo de relacionamento, o Ator deve chama-lo e como parâmetros, as entidades relacionadas para que ocorram as devidas ligações.	3- O Sistema, por sua vez, ligará as entidades por meio de um vértice, que em seu centro terá um losango que determinará o relacionamento, com este podendo ser uma nova entidade, dependendo do caso. 4- O Sistema ainda indicará, automaticamente a cardinalidade deste relacionamento, de acordo com o tipo de relacionamento escolhido.
Fluxo Alternativo I – Caso do Relacionamento ser uma Nova Entidade	
Ações do Ator	**Ações do Sistema**
1- No caso do Relacionamento ser uma	

nova entidade, o Ator deve determinar, se quiser, os atributos adicionais dessa entidade.	2- O Sistema, assim como na Funcionalidade 2, ligará os atributos nas entidades por meio de um vértice com um círculo nas pontas.
Responsável pela definição	**Data da criação**
Vitor Lima	09/10/2015

3. Modelagem UML

A estrutura do projeto físico é modelada pelo diagrama de classes, que representa todas as classes usadas no programa. Juntamente com seus métodos e atributos, o diagrama busca

exibir uma melhor codificação através de um planejamento de projetos e programação de softwares.

3.1. Diagrama de Classes

O diagrama de classes busca mostrar todas as classes usadas no projeto, e como elas se relacionam. O diagrama apresentado neste tópico foi feito fundamentado no diagrama que foi gerado baseado no código do programa. No entanto, ele está incompleto, já que não apresenta seus atributos e métodos, pois, devido ao tamanho do diagrama completo, ficaria totalmente ilegível a imagem. Portanto, as classes completas, com seus atributos e métodos, se encontram na sessão de anexos.

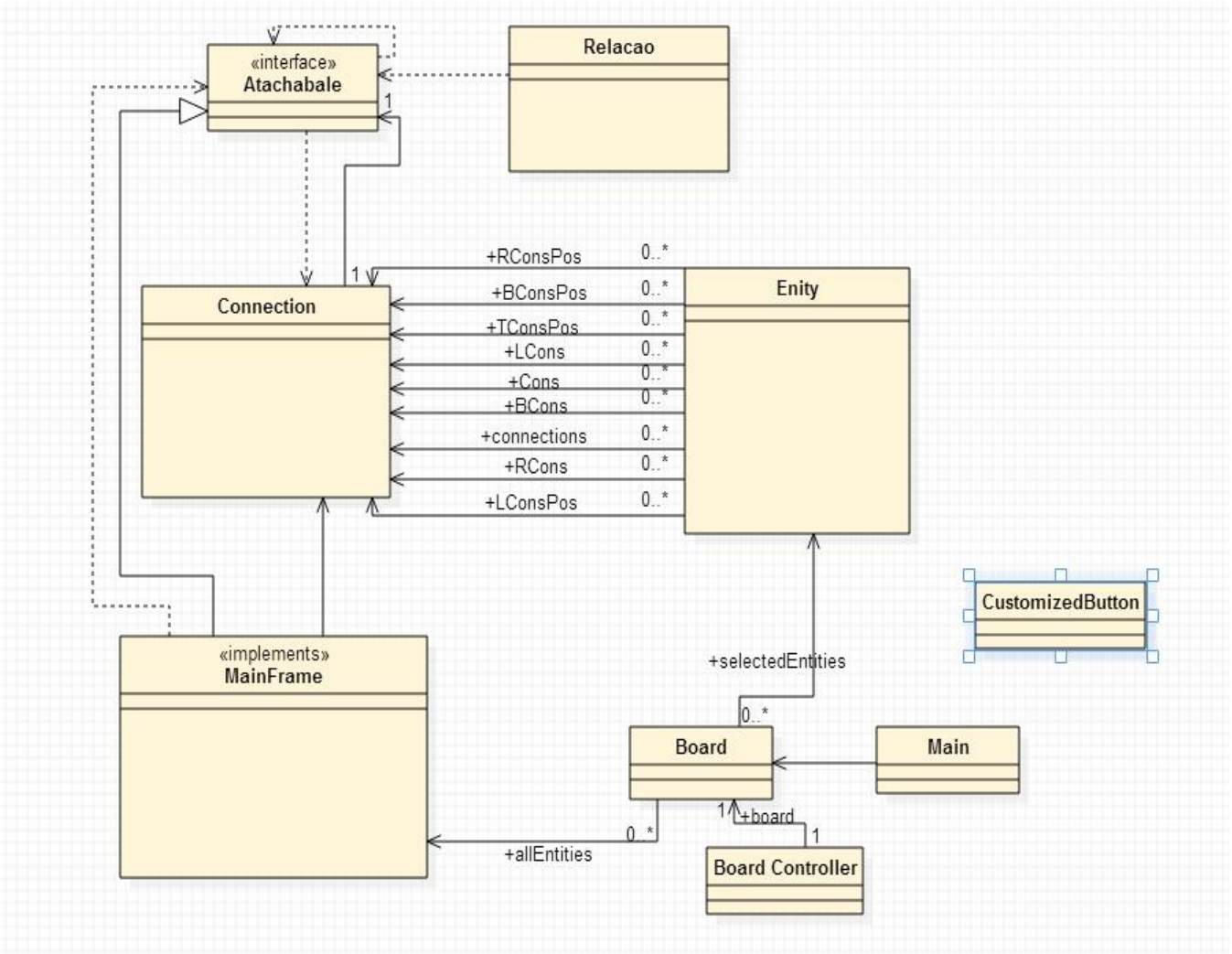


FIGURA 15. Diagrama de Classe - Gerado pelo código do programa

4. Resultados Obtidos

Após todo o processo de análise sobre ferramentas de modelagem de DER e estudo conceitual e técnico destes diagramas, seguiu o desenvolvimento da aplicação proposta.

A aplicação desenvolvida é capaz de criar diagramas de entidade e relacionamento conforme a notação especificada que se baseia os diagramas, o que pode inseri-la no ambiente didático.

Para demonstrar o funcionamento do software, será construído um diagrama de entidade e relacionamento, através da ferramenta, e será detalhado cada funcionalidade do programa.

4.1. Funcionalidade do Cursor

A primeira funcionalidade do software desenvolvida apresentada é a funcionalidade do cursor do mouse. É o primeiro botão do menu da aplicação. É a funcionalidade que, quando selecionada pelo seu botão do menu, permite a seleção dos componentes do diagrama, edição de seus nomes e a possibilidade de arrastar os elementos selecionados.

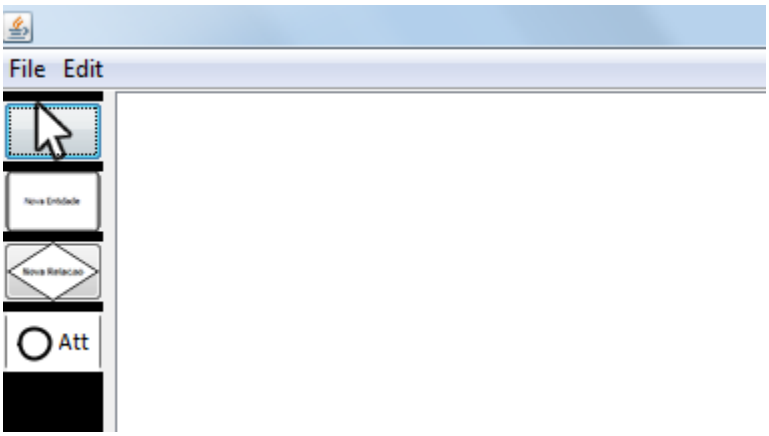


FIGURA 16. Menu de opções do software e botão do Cursor selecionado

4.2. Funcionalidade de Nova Entidade

A funcionalidade de criar novas entidades é referente ao segundo botão do menu de opções, o botão de Nova Entidade. E é responsável por criar cada instância de entidades desejadas.

O processo de criação de entidades é dado por um clique com botão esquerdo do mouse na opção de Nova Entidade e, em seguida, na tela onde ela será criada. As entidades criadas na tela, portanto, poderá ser livremente arrastadas, usando a funcionalidade do Cursor.

Assim que uma entidade é criada, ela recebe o nome padrão de Abstract Component. Porém a aplicação permite a edição deste nome, para que o DER fique de acordo com as preferências de seu usuário. A edição do nome da entidade se dá, selecionando o botão do cursor, e clicando duplamente com o botão esquerdo do mouse em seu nome, aparecendo assim, o campo de digitação.

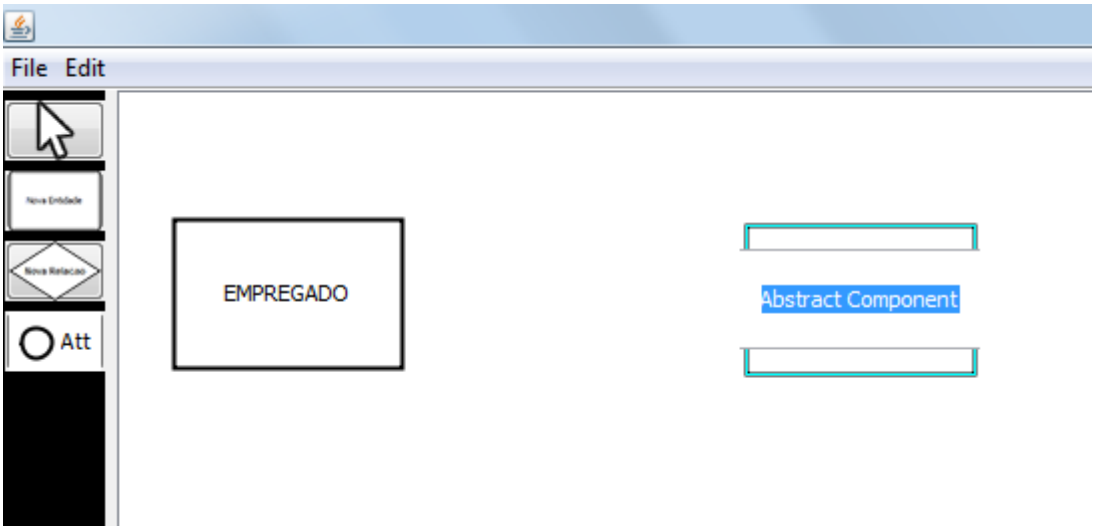


FIGURA 17. Funcionalidades de Nova Entidade

A Figura 17 exibe as funcionalidades do botão Nova Entidade, com duas instâncias de entidade criadas, uma com o nome editado, e outra com o campo de digitação aberto, para a edição do nome padrão.

4.3. Funcionalidade de Nova Relação

O botão de Nova Relacao é responsável por criar cada instância de relacionamento. No entanto, este botão só funciona quando há duas instâncias de entidades criadas na tela, já que, logicamente, só há um relacionamento quando existem, no mínimo, duas entidades criadas.

O processo de criação de um relacionamento é dado por um clique com o botão esquerdo do mouse na opção Nova Relação e, em seguida, com um clique na entidade participante da relação e outro clique na outra entidade que também participa.

Após a criação de um relacionamento, assim como a entidade, ele receberá o nome padrão Relationship. Porém, este nome também pode ser editado pelo usuário, da mesma maneira que se edita o nome de uma entidade. A nova relação criada, também recebe cardinalidade padrão de 1 para 1, que também poderá ser editada pelo usuário conforme suas preferências, no mesmo momento que se edita o nome.

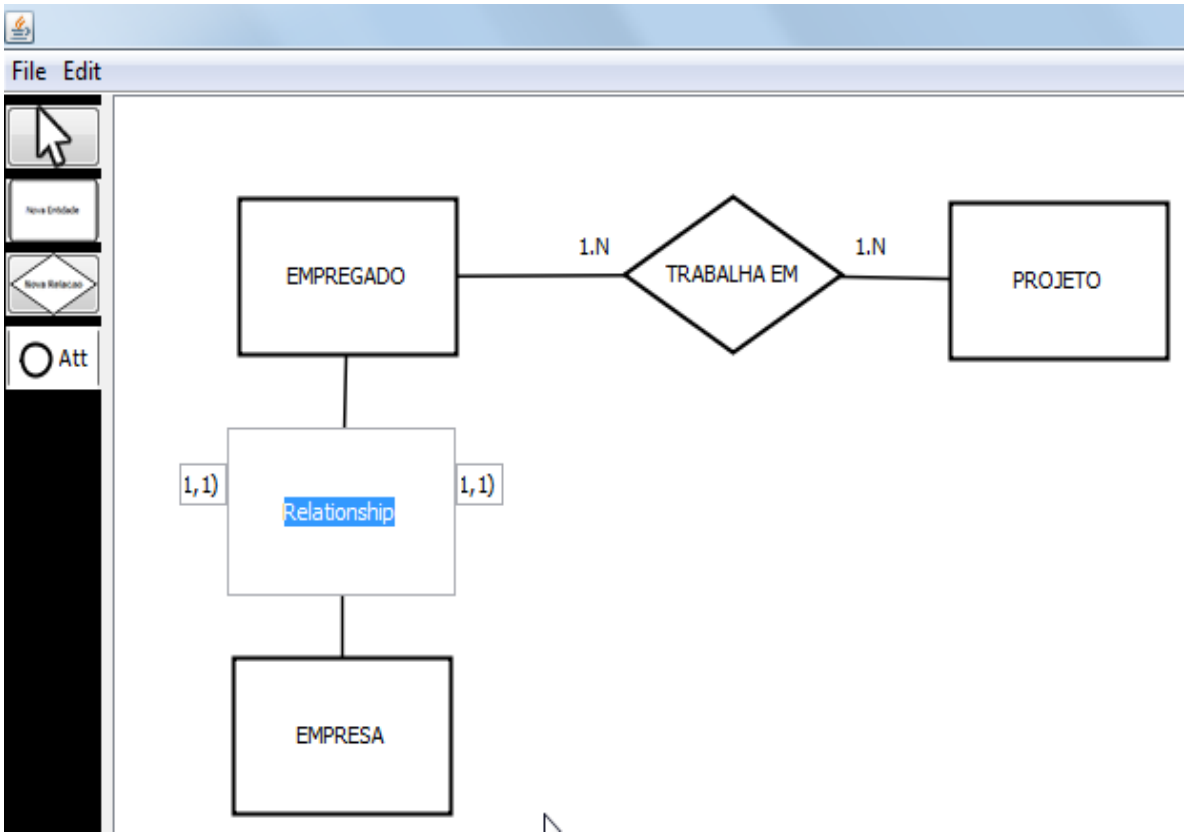


FIGURA 18. Funcionalidades de Nova Relação

Na Figura 18 é mostrado dois relacionamentos criados pelo programa, um com o nome e cardinalidade já editados (relacionamento "TRABALHA EM"), e outro com o campo de digitação aberto para ser editado.

4.4. Funcionalidade dos Atributos

O botão "Att" no menu da aplicação é referente à adição de atributos em entidades. Assim como o relacionamento, atributos só existem quando já existem suas entidades respectivas. Portanto, esse recurso só será disponível quando houver uma entidade para ser atribuída a ele.

A adição de atributos nas entidades é feita a partir da seleção do botão "Att" e, após a seleção feita, é adicionado na entidade referente ao atributo com um clique simples sobre a mesma. Quando adicionado o atributo na entidade, ele receberá o nome padrão de Abstract Component, o qual poderá ser editado da mesma maneira que os outros elementos.

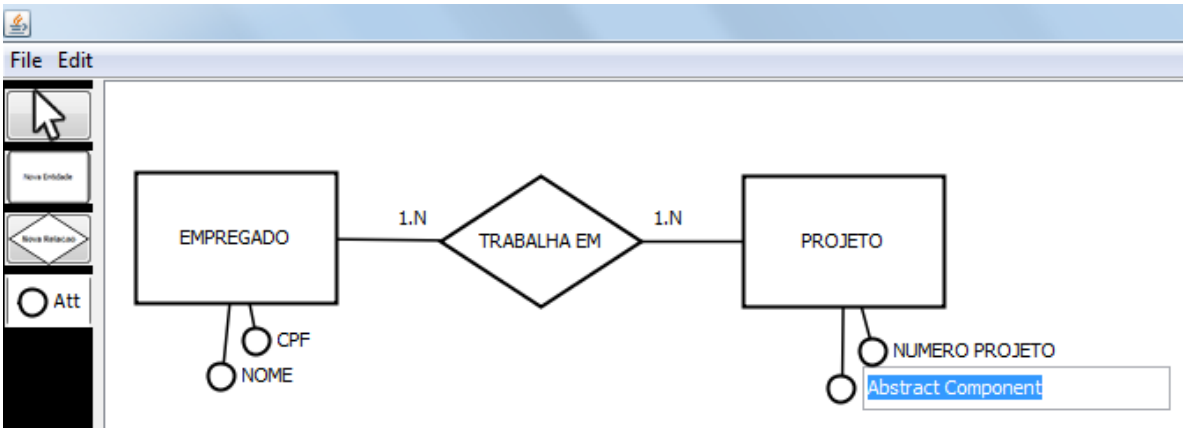


FIGURA 19. Funcionalidades dos Atributos

Analisando a Figura 19, observa-se dois atributos adicionados à entidade "Empregado", ambos com seus nomes editados, e mais dois atributos adicionados à entidade "Projeto", sendo um com seu nome modificado, e outro com o campo de digitação aberto para que seja alterado do padrão.

4.5. Funcionalidade do botão Del

Essa função do programa é responsável por apagar qualquer componente do diagrama selecionado, tanto entidades, como relacionamentos e atributos. Para isso basta, com o Cursor selecionado, fazer a seleção dos elementos que se deseja deletar, e apertar o botão Del do teclado, após a seleção dos componentes feita.

4.6. Funcionalidade de Salvar

O software também conta com um recurso de salvar o diagrama construído. No canto superior esquerdo da janela, há a opção "File" que, quando selecionada, exibe a opção "Salvar".

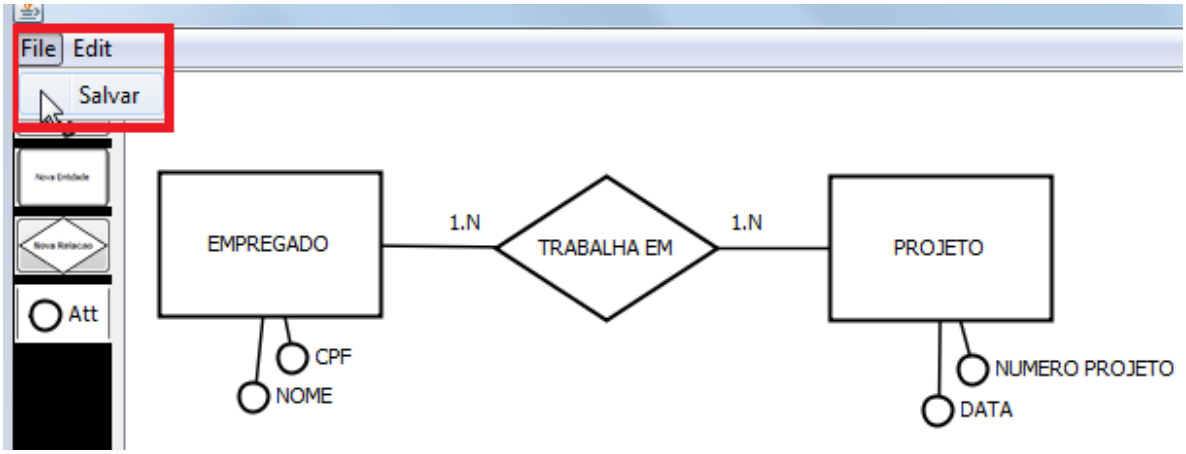


FIGURA 20. Funcionalidade de Salvar

Como mostrado na Figura 20, a aplicação, portanto, é capaz de salvar o diagrama construído. O arquivo salvado é um arquivo de imagem do tipo jpg; e o local que será salvo poderá ser escolhido assim, que a opção “Salvar” for selecionada.

5. Conclusão

O DER construído pelo software segue a notação a que se propôs, de Heuser, buscando a praticidade, tanto do manuseio do aplicativo, quanto da assimilação dos conceitos de modelagem de banco de dados, já que este é o principal foco de todo o projeto.

Desta maneira, o API-DER apresenta as funcionalidades propostas pelo projeto, sendo mais uma sugestão de ferramentas contribuindo para o auxílio da didática nas disciplinas de banco de dados. No entanto, o software apresenta aberturas para uma implementação de recursos que foram observadas após os resultados finais obtidos. Como exemplos desses recursos apresenta-se uma forma de identificar atributos que são chaves, auto relacionamento, um método de salvar o DER feito e continuar sua construção em um momento posterior e etc.

Portanto, o produto final também se mostra como uma base para um futuro desenvolvimento de uma aplicação mais sofisticada e completa; ou até a implantação de otimizações na próprio API-DER, tornando-o até mesmo uma API, como se pensou inicialmente.

6. Referências

CHEN, Peter. Modelagem de Dados: A Abordagem Entidade Relacionamento para Projeto Lógico; Tradução Cecília Camargo Bartalotti São Paulo, McGraw-Hill, 1990.

Chen, Peter P. The entity-relationship model – Toward a Unified View of Data, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1976.

DATE, C.J. Introdução a sistemas de banco de dados. 8º ed, Campus, 2004.

ELMASRI, Ramez, NAVATHE, Shamkant. Sistemas de Banco de Dados, São Paulo: 6ª ed. São Paulo, Pearson, 2011.

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 4ª ed. Sagra Luzzatto, 2001.

MARTIN, James. Information Engineering: Books I, II & III. Englewood Cliffs: Prentice-Hall, 1990.

MACHADO, Felipe, ABREU, Mauricio. Projeto de Banco de Dados – Uma Visão Prática. 7ª ed. Érica, 2009.

7. Anexos

Nesta sessão, estão inseridos todos os anexos necessários para uma compreensão mais aprofundado do desenvolvimento do projeto.

Da Figura 17 em diante, serão apresentadas as classes do diagrama de classes completas, ou seja, com seus métodos e atributos.

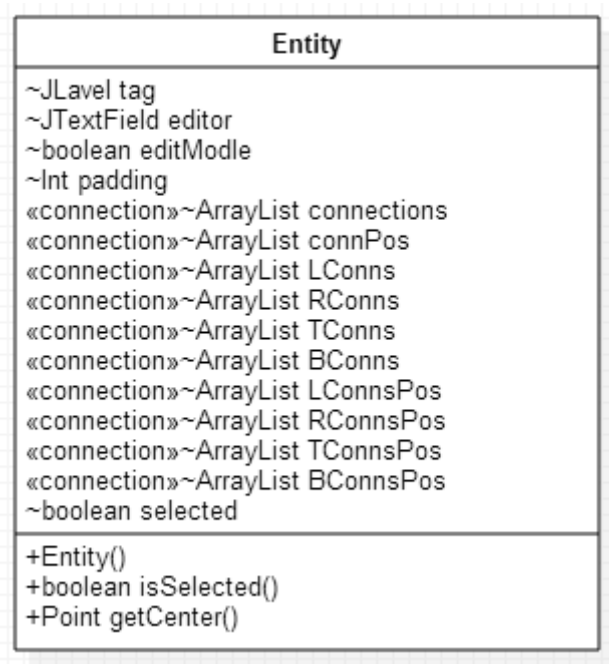


FIGURA 21. Classe Entity, com seus métodos e atributos

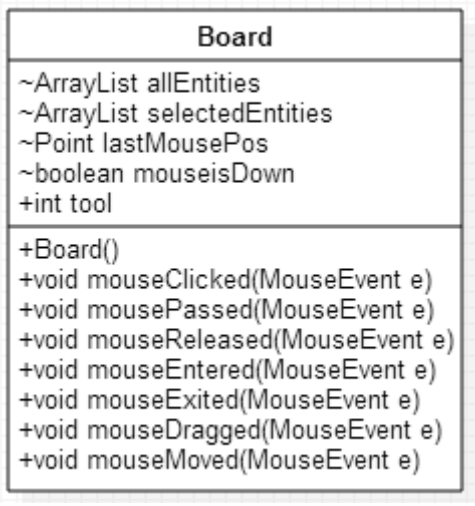


FIGURA 22. Classe Board, com seus métodos e atributos

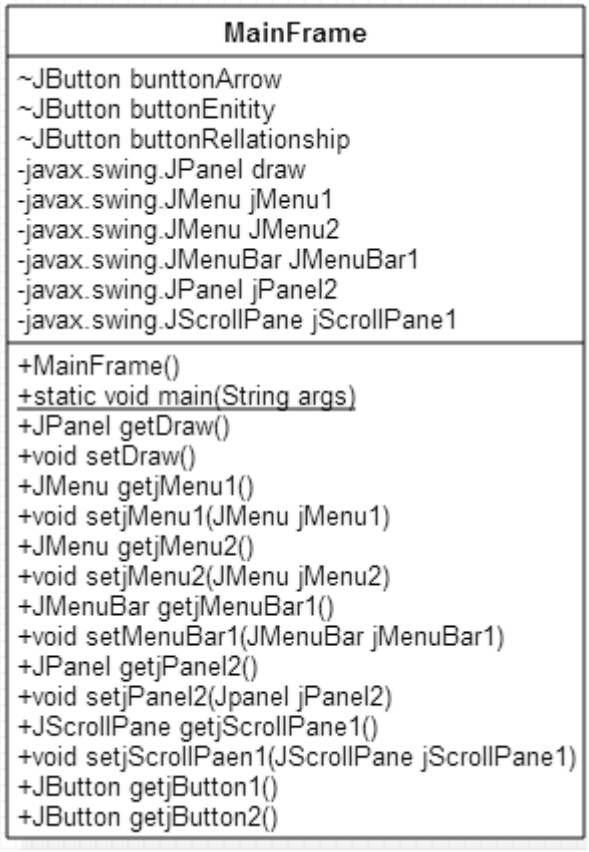


FIGURA 23. Classe MainFrame, com seus métodos e atributos

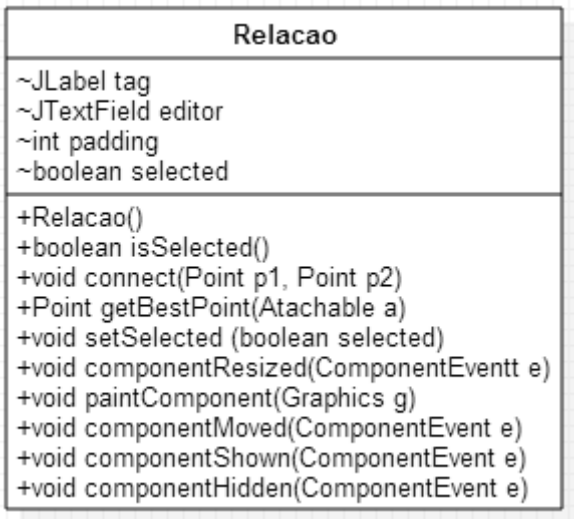


FIGURA 24. Classe Relacao, com seus métodos e atributos

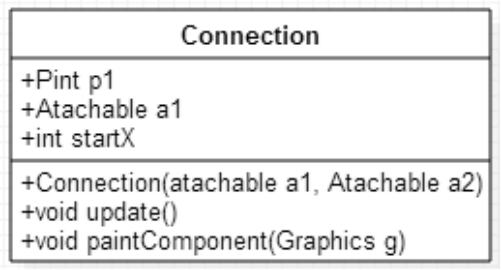


FIGURA 25. Classe Connection, com seus métodos e atributos

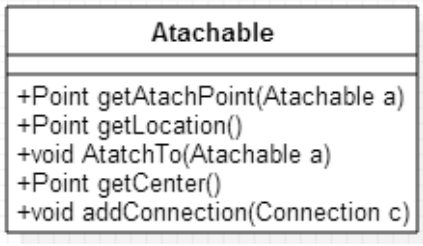


FIGURA 26. Classe Atachable, com seus métodos e atributos

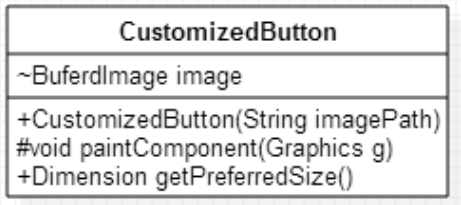


FIGURA 27. Classe CustomizedButton, com seus métodos e atributos

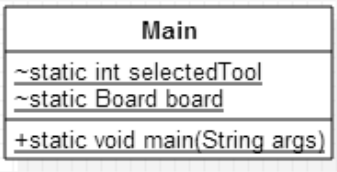


FIGURA 28. Classe Main, com seus métodos e atributos

FIGURA 29. Classe BoardController, com seus métodos e atributos