

UNIVERSIDADE FEDERAL DO PAMPA

Jonnathan Riquelmo Lopes Frescura

**ERText: Uma Linguagem Específica de
Domínio para a Representação de Modelos
Conceituais de Bancos de Dados Relacionais**

Alegrete
2019

Jonnathan Riquelmo Lopes Frescura

**ERText: Uma Linguagem Específica de Domínio para
a Representação de Modelos Conceituais de Bancos
de Dados Relacionais**

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia de
Software da Universidade Federal do Pampa
como requisito parcial para a obtenção do tí-
tulo de Bacharel em Engenharia de Software.

Orientador: Prof. Dr. Maicon Bernardino da
Silveira

Coorientador: Prof. Dr. Fábio Paulo Basso

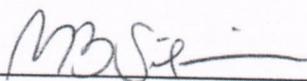
Alegrete
2019

Jonnathan Riquelmo Lopes Frescura

**ERText: Uma Linguagem Específica de Domínio para
a Representação de Modelos Conceituais de Bancos
de Dados Relacionais**

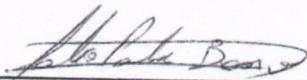
Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia de
Software da Universidade Federal do Pampa
como requisito parcial para a obtenção do tí-
tulo de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em 28 de Novembro de 2019
Banca examinadora:



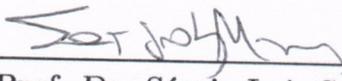
Prof. Dr. Maicon Bernardino da Silveira

Orientador
UNIPAMPA



Prof. Dr. Fábio Paulo Basso

Coorientador
UNIPAMPA



Prof. Dr. Sérgio Luis Sardi Mergen

UFSM



Prof. Dr. Elder de Macedo Rodrigues

UNIPAMPA

*Este trabalho é dedicado à minha família pelo apoio incondicional
em todos os momentos difíceis.*

AGRADECIMENTOS

Não é só com estudo que é possível concluir uma graduação, depende muito da nossa força de vontade. A luta para chegar até aqui foi grande, e certamente não conseguiria ter vencido tudo sozinho.

Eu nunca fui habilidoso quando o assunto é se expressar com palavras, e por isso acredito que não vou conseguir descrever o quanto sou grato a todas as pessoas que de alguma forma contribuíram para que eu alcançasse este momento. De toda a maneira, aqui fica o registro da minha tentativa.

Agradeço imensamente meus pais, Zilma e Aroldo, por todo o empenho para me proporcionar uma educação superior. Nada seria possível sem todo o suporte emocional e financeiro de vocês durante minha jornada. Obrigado por todo o incentivo, por terem lutado para que eu tivesse essa oportunidade, por me ensinarem valores. Dizem que bons pais corrigem erros, mas pais excepcionais ensinam a pensar. Obrigado por me conduzirem ao desenvolvimento de um pensamento crítico. Devo absolutamente TUDO a vocês. Obviamente, não posso deixar de agradecer as minhas irmãs, Suh e Fê, que da mesma forma me acompanharam.

Aos docentes da UNIPAMPA que contribuíram com minha formação - em especial aos professores Bernardino e Basso por aceitarem me guiar neste trabalho final. O apoio de vocês no planejamento e execução deste estudo foi decisivo, muito obrigado por me encorajarem e compartilharem conhecimento comigo.

A todos os colegas e amigos que me acompanharam nessa jornada, mesmo aqueles com quem perdi contato. Com vocês tive momentos de aprendizado e muitas risadas, assim como também tivemos momentos complicados (alô RP). Em especial, agradeço aos guris que me receberam na Colônia, Lukas e Marcus. Ao Alex pelas conversas e mates na 303, a Jaque por ser uma grande amiga, inclusive me auxiliando mais de uma vez com assuntos complexos (métodos formais que o diga haha), e ao Guilherme, agora mestrando, pelos conselhos ditos e suporte oferecido em minha pesquisa nestes últimos meses. Ainda em tempo, deixo a lembrança ao André, Naihara, Jean, Maurício, Paula, Luana, Karina e Filipe.

Ao Renato, meu irmão de coração, por sempre me receber para uma boa conversa quando retornava para casa, me ajudando a colocar as ideias no lugar. Você não tem noção de como te admiro como ser humano. Tenho orgulho de dizer que sou teu amigo. Obrigado por não deixar nossa amizade se perder mesmo com minha ausência.

Por fim, mas não menos importante, agradeço uma certa morena aquariana que me acompanhou nesses últimos dois anos e, mesmo que na maior parte do tempo estivesse distante fisicamente, me ajudou a superar tudo e ser uma pessoa melhor. Você me mostrou o verdadeiro significado de serendipidade.

*“Se enxerguei mais longe, foi porque
me apoiei em ombros de gigantes.”*

Isaac Newton

RESUMO

Com o avanço da tecnologia os bancos de dados passaram a ser elementos vitais na sociedade contemporânea. Os bancos de dados são conjuntos de dados armazenados para retratar algum sentido sobre um domínio específico. As informações armazenadas são consideradas bens de grande relevância nas organizações modernas. Dessa forma o uso eficaz de bancos de dados é de suma importância para a manutenção e o prosseguimento correto das suas atividades. Posto isto, a capacitação nessa área para profissionais oriundos da academia deve ser constante, sendo esse um ponto fundamental com o qual as instituições de ensino superior devem ter especial atenção. Contudo, a variedade de tecnologias de sistemas de banco de dados que se tornaram disponíveis nos últimos anos, sendo a grande maioria focada em abordagens gráficas, dificulta a escolha de ferramentas para modelagem de entidade-relacionamento (ER) na indústria e, conseqüentemente, no meio acadêmico. Objetivando contribuir com uma alternativa *open source* relevante, este Trabalho de Conclusão de Curso propõe uma Linguagem Específica de Domínio (*Domain Specific Language* - DSL) textual para apoiar o processo de ensino-aprendizagem da modelagem conceitual de banco de dados. O uso de DSLs fornece meios de especificar e modelar domínios de forma mais rápida e produtiva, pois são linguagens com expressividade limitada a domínios particulares, diferenciando-se assim das linguagens de propósito geral. Nesse sentido, foi executado uma investigação do estado da arte e da prática em projeto e modelagem de banco de dados utilizando DSLs. Um levantamento de inovações recentes foi realizado por meio de um mapeamento sistemático complementado por uma pesquisa na literatura cinza. Esse trabalho abrange um conjunto final de 10 estudos primários focados em DSLs e identifica 55 ferramentas já aplicadas na indústria e academia para modelagem ER em nível conceitual, lógico e físico. Em seguida, houve a seleção do *framework* Xtext para apoiar o desenvolvimento da linguagem de modelagem. Após o definição de duas versões da gramática, foi planejado e executado uma avaliação preliminar junto a um grupo focal composto por treze (13) participantes. Com o *feedback* recebido na dinâmica, chegou-se a uma versão final e, em seguida, ao *plugin* da solução ERText. Para avaliar a proposta foi conduzida uma avaliação empírica com vinte e sete (27) sujeitos. A intenção foi verificar o esforço (tempo), precisão, revocação, Medida-F, utilidade percebida e facilidade de uso da ferramenta com abordagem textual desenvolvida frente a uma ferramenta com abordagem gráfica. Os resultados obtidos apresentam evidências de que ao realizar tarefas de modelagem com ambas as abordagens há menor esforço associado à abordagem gráfica e um desempenho muito similar no que diz respeito a qualidade dos modelos feitos em ambas as ferramentas.

Palavras-chave: Banco de Dados. Projeto e Modelagem de Banco de Dados. Modelagem Conceitual. Linguagem Específica de Domínio.

ABSTRACT

With the advance of technology, databases have become vital elements in contemporary society. Databases are stored data sets to describe some meaning about a specific domain. The information stored is considered to be of great relevance in modern organizations. In this way, the effective use of databases have great importance for the maintenance and correct progress of their activities. That said, the formation in this area for professionals coming from academy must be constant, which is a fundamental point with which higher education institutions should pay special attention. However, the variety of database systems technologies that have become available in recent years, most of which are focused on graphical approaches, make it difficult to choose entity-relationship (ER) modeling tools in industry and, consequently, in the academy. In order to contribute with a relevant open source alternative, this Course Conclusion Work proposes a Textual Domain Specific Language (DSL) to support the teaching-learning process of conceptual database modeling. The use of DSLs provides means to specify and model domains more quickly and productively, since they are expressive languages limited to particular domains, thus differentiating themselves from general-purpose languages. In this sense, an investigation of the state of the art and the practice in database design and modeling using DSLs was performed. A survey of recent innovations was carried out through a systematic mapping complemented by a survey in the gray literature. This work covers a final set of 10 primary studies focused on DSLs and identifies 55 tools already applied in industry and academy for ER modeling at conceptual, logical and physical level. Then, the Xtext framework was selected to support the development of the modeling language. After defining two versions of the grammar, a preliminary assessment was planned and carried out with a focus group of thirteen (13) participants. With the feedback received in the dynamics, we arrived at a final version and then the plugin of the ERText solution. To evaluate the proposal, an empirical evaluation was conducted with twenty seven (27) subjects. The intention was to verify the effort (time), precision, recall, F-Measure, perceived utility and use of the textual approach tool compared to a graphical approach tool. The results show evidence that when performing modeling tasks with both approaches, there is less effort associated with the graphical approach and a very similar performance regarding the quality of the models made in both tools.

Key-words: Database. Database Design and Modeling. Conceptual Modeling. Domain Specific Language.

LISTA DE FIGURAS

Figura 1 – Classificação da Pesquisa.	30
Figura 2 – Desenho de pesquisa.	31
Figura 3 – Desenho de pesquisa (continuação).	32
Figura 4 – Fragmento de DER.	34
Figura 5 – Construtores do modelo Abordagem Entidade-Relacionamento (ER).	35
Figura 6 – Relacionamentos e cardinalidades.	36
Figura 7 – Exemplos de notações para modelagem ER.	37
Figura 8 – Exemplo de MLD para BD relacional.	38
Figura 9 – Estrutura de um modelo lógico descrita de forma textual.	38
Figura 10 – Modelo de dados relacional.	40
Figura 11 – Relação de MDE, MDD e MDA.	41
Figura 12 – Níveis de abstração do MDA.	43
Figura 13 – Dimensões de uma DSL.	45
Figura 14 – Representação dos níveis da Hierarquia de Chomsky.	48
Figura 15 – Exemplo de mini-linguagem definida com BNF.	49
Figura 16 – Processo de mapeamento sistemático.	51
Figura 17 – String genérica de busca.	53
Figura 18 – Estudos primários por biblioteca digital.	56
Figura 19 – Ciclos de seleção dos estudos primários.	57
Figura 20 – Diagrama de <i>Venn</i> dos modelos suportados nas ferramentas.	60
Figura 21 – Arquitetura geral do Xtext.	69
Figura 22 – Representação do modelo <i>Ecore</i> da 1º versão da DSL.	69
Figura 23 – Implementação da 1º versão da DSL.	71
Figura 24 – Fragmento de implementação da 2º versão da DSL.	72
Figura 25 – Exemplo de uso da 1º versão da DSL no RCP do Eclipse.	73
Figura 26 – Processo do Grupo Focal.	74
Figura 27 – Resultados da Discussão 1 - Grupo Focal.	76
Figura 28 – Resultados da Discussão 3 - Grupo Focal.	77
Figura 29 – Fragmento do <i>plugin</i> da solução instalado.	78
Figura 30 – Fragmento da solução sendo utilizada.	79
Figura 31 – Exemplo de modelo lógico gerado.	79
Figura 32 – Premissas adotadas para a transformação entre os modelos.	80
Figura 33 – <i>Design</i> do experimento.	88
Figura 34 – Distribuição amostral referente ao esforço.	94
Figura 35 – Esforços medidos nos tratamentos.	95
Figura 36 – Distribuição amostral referente a efetividade.	97
Figura 37 – <i>Boxplot</i> da Medida-F dos tratamentos.	98
Figura 38 – Avaliação dos atributos de qualidade dos tratamentos.	100

Figura 39 – Avaliação dos construtores da DSL. 101

LISTA DE TABELAS

Tabela 1 – Síntese do TCC.	32
Tabela 2 – Bibliotecas digitais utilizadas.	52
Tabela 3 – Termos e sinônimos utilizados.	53
Tabela 4 – Critérios de Avaliação de Qualidade.	55
Tabela 5 – Formulário de Extração de Dados.	55
Tabela 6 – Resultados da avaliação de qualidade.	58
Tabela 7 – Objetos de banco de dados representados.	59
Tabela 8 – Categorização das DSLs propostas.	59
Tabela 9 – Ferramentas para modelagem de bancos de dados.	61
Tabela 10 – Ferramentas para modelagem de bancos de dados (continuação).	62
Tabela 11 – Tempos de execução de cada abordagens.	94
Tabela 12 – Medidas relacionadas ao esforço médio.	95
Tabela 13 – Avaliação dos modelos produzidos no experimento.	96
Tabela 14 – Médias gerais da avaliação dos modelos produzidos no experimento.	98

LISTA DE SIGLAS E ABREVIATURAS

- AST** *Abstract Syntax Tree*
- BD** Banco de Dados
- BNF** *Backus-Naur Form*
- CE** Critério de Exclusão
- CI** Critério de Inclusão
- CIM** *Computation-Independent Model*
- CQ** Critério de Qualidade
- CSS** *Cascading Style Sheets*
- DDL** *Data Definition Language*
- DER** Diagrama Entidade-Relacionamento
- DML** *Data Manipulation Language*
- DOM** *Document Object Graph*
- DQL** *Data Query Language*
- DSL** *Domain Specific Language*
- DTL** *Data Transaction Language*
- EER** *Enhanced EntityRelationship*
- EMF** *Eclipse Modeling Framework*
- ER** Abordagem Entidade-Relacionamento
- ES** Engenharia de Software
- ESE** Engenharia de Software Experimental
- GLC** Gramática Livre de Contexto
- GPL** *General-Purpose Programming Language*
- IDE** *Integrated Development Environment*
- IHC** Interação Humano-Computador
- LN** Linguagem Natural

LW *Language Workbench*

MCD Modelo Conceitual de Dados

MDA *Model-Driven Architecture*

MDD *Model-Driven Development*

MDE *Model-Driven Engineering*

MDSE *Model-Driven Software Engineering*

MFD Modelo Físico de Dados

MLD Modelo Lógico de Dados

MLM *Multivocal Literature Mapping*

OMG *Object Management Group*

PIM *Platform-Independent Model*

PLN Processamento de Linguagem Natural

CIM *Platform-Specific Model*

QP Questão de Pesquisa

RCP *Rich Client Platform*

SGBD Sistema de Gerenciamento de Banco de Dados

SLM *Systematic Literature Mapping*

SQL *Structured Query Language*

SysML *Systems Modeling Language*

TCC Trabalho de Conclusão de Curso

TCLE Termo de Consentimento Livre e Esclarecido

UML *Unified Modeling Language*

VHDL *VHSIC Hardware Description Language*

XML *eXtensible Markup Language*

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Motivação	26
1.2	Objetivos	26
1.3	Questão de Pesquisa	27
1.4	Justificativa	27
1.5	Organização	28
2	METODOLOGIA DA PESQUISA	29
2.1	Classificação de Pesquisa	29
2.2	Desenho de Pesquisa	30
2.3	Lições do Capítulo	32
3	FUNDAMENTAÇÃO TEÓRICA	33
3.1	Projeto de Banco de Dados	33
3.1.1	Modelo Conceitual de Dados	34
3.1.1.1	Modelo Entidade-Relacionamento	34
3.1.1.2	Notações de Modelagem ER	37
3.1.2	Modelo Lógico de Dados	37
3.1.3	Modelo Físico de Dados	38
3.2	Linguagem SQL	39
3.2.1	Bancos de Dados Relacionais	40
3.3	<i>Model-Driven Engineering</i>	40
3.3.1	<i>Domain-Specific Language</i>	43
3.3.2	<i>Language Workbenches</i>	46
3.4	Gramáticas Livres de Contexto	47
3.4.1	Formalismo de Backus-Naur	48
3.5	Lições do Capítulo	49
4	MAPEAMENTO MULTIVOCAL DE LITERATURA	51
4.1	Protocolo	51
4.1.1	Questões de Pesquisa	52
4.1.2	Fontes de Busca	52
4.1.3	String de Busca	53
4.1.4	Critérios de Seleção	53
4.1.5	Avaliação de Qualidade	54
4.1.6	Estratégia de Extração de Dados	55
4.1.7	Pesquisa na Literatura Cinza	55
4.2	Execução do Mapeamento Multivocal	56

4.3	Resultados e Discussão	57
4.4	Ameaças à Validade	63
4.5	Trabalhos Relacionados	63
4.6	Lições do Capítulo	65
5	DSL PARA A REPRESENTAÇÃO DE MODELOS CON- CEITUAIS DE BANCO DE DADOS RELACIONAIS	67
5.1	Requisitos da Linguagem	67
5.2	Decisões de Projeto	68
5.3	Arquitetura	68
5.4	Protótipo	70
5.5	Avaliação Preliminar da Gramática	73
5.5.1	Planejamento	74
5.5.1.1	Preparação	74
5.5.2	Execução	74
5.5.3	Análise dos Resultados	75
5.5.3.1	Discussões	75
5.6	A Ferramenta ERText	77
5.6.1	Operação da ERText	78
5.7	Lições do Capítulo	81
6	AVALIAÇÃO EXPERIMENTAL	83
6.1	Definição da Avaliação	83
6.2	Escopo	83
6.3	Planejamento	84
6.3.1	Definição do Objetivo	84
6.3.2	Questões de Pesquisa	84
6.3.3	Contexto	84
6.3.4	Formulação de Hipótese	85
6.3.5	Seleção dos Participantes	87
6.3.6	<i>Design</i> da Avaliação	87
6.3.7	Instrumentação	88
6.3.8	Ameaças à Validade	89
6.3.8.1	Validade do Construto	89
6.3.8.2	Validade Interna	90
6.3.8.3	Validade Externa	90
6.3.8.4	Validade da Conclusão	91
6.4	Operação do Experimento	92
6.4.1	Preparação	92
6.4.2	Execução	92

6.5	Resultados	93
6.5.1	Esforço	93
6.5.2	Efetividade	95
6.5.3	Avaliação Qualitativa	98
6.6	Lições do Capítulo	101
7	CONSIDERAÇÕES FINAIS	103
7.1	Trabalhos Futuros	104
	REFERÊNCIAS	105
	 APÊNDICES	 111
	APÊNDICE A – DSL PROPOSTA	113
A.1	Versão Final da Gramática da DSL	113
	APÊNDICE B – GRUPO FOCAL	115
B.1	Termo de Consentimento	115
B.2	Glossário de Termos	116
B.3	Questionário de Perfil	117
B.4	Roteiro	118
B.5	Discussão 1	119
B.6	Discussão 2	120
B.7	Discussão 3	121
B.8	Modelos da DSL	122
	APÊNDICE C – EXPERIMENTO CONTROLADO	123
C.1	Termo de Consentimento	123
C.2	Glossário de Termos	124
C.3	Questionário de Perfil	125
C.4	Instruções BrModelo	126
C.5	Instruções ERDSL	127
C.6	Instrumento 1	128
C.7	Instrumento 2	129
C.8	Instrumento 3	130
C.9	Instrumento 4	131
	Índice	133

1 INTRODUÇÃO

É praticamente impossível administrar o mundo moderno sem software. As infra-estruturas de empresas nacionais e multinacionais são controladas por sistemas baseados em computadores e a maioria dos produtos elétricos inclui algum computador ou software de controle. A fabricação e distribuição industrial é totalmente informatizada, assim como o sistema financeiro. Entretenimento, incluindo a indústria criativa formada por jogos, música e cinema tem intensiva participação de software em suas atividades. Portanto, a Engenharia de Software (ES) é essencial para o funcionamento das sociedades (SOMMERVILLE, 2011).

Contudo, a ES é inviável sem a persistência e manipulação de dados. Nos primórdios do uso de computadores, a persistência de dados se dava em forma de arquivos de texto simples, inspirado inicialmente nas raízes de uma invenção muito antiga, denominada máquina de tabulação, criada por volta de 1860. Todavia, em pouco tempo, e em razão da evolução das tecnologias, o uso de arquivos de texto começou a se mostrar ineficiente (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

Uma alternativa ao problema relacionado aos arquivos de texto, surgida há mais de 40 anos atrás e ainda utilizada amplamente nos dias atuais, foi justamente a automatização do conceito das máquinas de tabulação. Isso foi realizado pelo cientista da computação Charles Bachman, pesquisador ligado a indústria e ganhador do prêmio ACM Alan Turing de 1973 pela sua fundamental contribuição para a área de Banco de Dados (BD) (KRISHNA, 1992).

Devido ao alto volume de informação que é produzido e manipulado, os BDs fundamentados inicialmente por Bachman são essenciais na sociedade contemporânea. Segundo Date e Warden (1990), uma base de dados é uma coleção de dados operacionais armazenados, usados pelos sistemas de aplicação de uma determinada organização. Para Elmasri e Navathe (2011), um BD pode ser definido como uma abstração do mundo real, também chamado de minimundo, uma vez que representa aspectos que, em conjunto, carregam um significado implícito.

Analisando de forma mais objetiva, as bases de dados podem ser consideradas os ativos organizacionais mais importantes atualmente. Isso se deve ao fato de que não armazenam apenas informações triviais mas, por exemplo, também dados de faturamento, pesquisas de mercado e outros aspectos que auxiliam na tomada de decisão. Contudo a sua importância não está focada apenas nas organizações, sendo também possível atestar que o uso de BD pode representar um papel crítico na vida dos usuários finais quando estes são analisados individualmente.

Com este cenário estabelecido, é notável que existe um dever crescente da academia em fornecer um bom nível de preparo para os futuros profissionais que vão ingressar em uma indústria cada vez mais exigente. Com frequência as instituições de ensino superior abordam a área de BD com disciplinas específicas e também em componentes curriculares

que são convergentes nos currículos de seus cursos. Neste contexto, a UNIPAMPA possui cursos que envolvem desenvolvimento de software, tais como Engenharia de Software e Ciência da Computação.

1.1 Motivação

Segundo Salgado e Medeiros (1995), o ensino na área de BD é parte essencial da formação de profissionais de computação. O foco no ensino em BD geralmente é dividido em quatro etapas: projeto e modelagem, sistemas de gerência de bases de dados, estudos comparativos entre estes sistemas e o desenvolvimento de aplicações.

Tendo como premissa que existe uma crescente busca por instrumentos que apoiem o processo de ensino-aprendizagem na academia, este estudo tem foco na primeira etapa. O ensino de projeto e modelagem de BD em geral é conduzido com a apresentação de tópicos essenciais e a posterior introdução ao uso de ferramentas de modelagem que utilizam abordagens geralmente gráficas. Este estudo tem como motivação oferecer um produto de software que dê apoio a esta fase. Este produto fará uso da abordagem textual, tendo em vista que possua uma gramática de fácil uso e compreensão.

1.2 Objetivos

O objetivo geral deste trabalho é propor a especificação e realizar a implementação de uma Linguagem Específica de Domínio, do inglês *Domain Specific Language* (DSL), para o projeto e modelagem de BDs relacionais. O foco da modelagem é em nível conceitual, mas deseja-se que a solução possa também realizar a transformação de modelos conceituais para o modelo lógico.

Para atingir o objetivo geral proposto, é fundamental que exista a divisão do problema nos seguintes objetivos específicos que precisam ser atingidos:

- Pesquisar a literatura visando encontrar propostas que façam proposições que forneçam apoio à modelagem de bases de dados;
- Investigar tecnologias que auxiliem no processo de criação de linguagens específicas de domínio;
- Compreender quais são os requisitos necessários para a criação da linguagem;
- Desenvolver uma gramática para modelagem conceitual de BDs que sigam o modelo relacional;
- Integrar a linguagem proposta em uma ferramenta *open source*;
- Implementar a transformação do modelo conceitual para o modelo lógico;

- Realizar a avaliação da solução proposta;
- Contribuir com uma ferramenta que auxilie no processo de ensino de projeto e modelagem de BDs.

1.3 Questão de Pesquisa

Visando a condução do restante deste Trabalho de Conclusão de Curso (TCC), foi definida uma Questão de Pesquisa (QP) central, descrita a seguir:

QP: Uma DSL textual pode auxiliar, em nível conceitual de modelagem, o ensino de projeto e modelagem de banco de dados relacionais?

1.4 Justificativa

Desde os primórdios os desenvolvedores utilizam texto para especificar produtos de software. As linguagens de programação aumentam o nível de abstração de maneira similar aos modelos. Logo, por consequência lógica, isso resulta em linguagens de modelagem textual.

Uma linguagem de modelagem textual é geralmente processada por mecanismos que transformam as informações expressas em formato textual para modelos. Esses mecanismos baseiam sua execução na estrutura sintática de uma linguagem de modelagem textual, que é formalizada em uma gramática. Uma gramática define palavras-chave de uma linguagem, o aninhamento de seus elementos e também a notação de suas propriedades. Dito isto, pode-se inferir que os modelos textuais podem trazer diversos benefícios:

- **Transmitir de muitos detalhes:** Quando se trata de elementos com inúmeras propriedades, a abordagem textual geralmente se destaca em relação aos gráficos. O mesmo pode ser dito quanto às estruturas formadas por um grande número de partes muito pequenas, como operações matemáticas ou sequências de instruções;
- **Aumentar coesão do modelo:** Um modelo textual geralmente especifica os elementos inteiramente em um só local. Se por um lado isso pode ser uma desvantagem para uma exibição em alto nível, em contrapartida pode facilitar a localização de definições de propriedades em baixo nível. Na proposta deste trabalho toda a modelagem conceitual é realizada em apenas um arquivo;
- **Realizar uma edição rápida:** Durante a criação e edição de modelos textuais não é necessário a recorrente alternância entre teclado e *mouse*. Logo, é provável que se gaste menos tempo formatando modelos textuais do que, por exemplo, refinando a posição, as ligações ou mesmo as bordas de elementos em diagramas;

- **Utilizar editores genéricos:** Não existe necessariamente a exigência de uma ferramenta específica para criar ou modificar modelos textuais, como é o caso de linguagens específicas de domínio com esta abordagem. Para alterações simples é possível o uso de qualquer editor de texto genérico. Entretanto, para tarefas maiores é melhor se ter algum suporte para a linguagem de modelagem. Na proposta desse trabalho está incluso a integração da linguagem definida com um editor Eclipse, fornecendo assim um alto nível de auxílio para a escrita.

1.5 Organização

Este trabalho está organizado da seguinte forma. No Capítulo 2 é apresentada a metodologia de pesquisa adotada. No Capítulo 3 é realizada a fundamentação teórica. Um mapeamento multivocal de literatura tem o protocolo e condução descritos no Capítulo 4. A seguir, ocorre a apresentação da proposta no Capítulo 5. O Capítulo 6 detalha o experimento controlado conduzido para a avaliação da proposta. Finalmente, as considerações finais são discutidas no Capítulo 7, onde também são apontados os trabalhos futuros.

2 METODOLOGIA DA PESQUISA

Este capítulo aborda a metodologia de pesquisa adotada para a execução deste estudo. Na Seção 2.1 são elencadas as classificações dos tipos de pesquisa relacionados as atividades realizadas. Os desenhos de pesquisa do TCC I e II utilizados para a execução da pesquisa são apresentados na Seção 2.2 e, em seguida, na Seção 2.3 as lições do capítulo são pontuadas.

2.1 Classificação de Pesquisa

Segundo Wazlawick (2017), as monografias geralmente possuem um capítulo para a elucidação da metodologia. Contudo, analisando-se semanticamente o termo, a metodologia seria o estudo dos métodos. Apesar do uso corrente, linguisticamente seria mais correto afirmar que um trabalho científico individualmente tem um método de pesquisa, ou desenho de pesquisa, utilizando abordagens, técnicas e procedimentos diversos combinados.

Em geral, para que um estudo possa ter maior confiança no que diz respeito ao rigor científico, é imprescindível que sejam identificadas as atividades e técnicas necessárias que possibilitem a se chegar no objetivo proposto (PEFFERS et al., 2007). A Figura 1 mostra a classificação desse estudo mediante sua natureza, abordagem, objetivos e procedimentos.

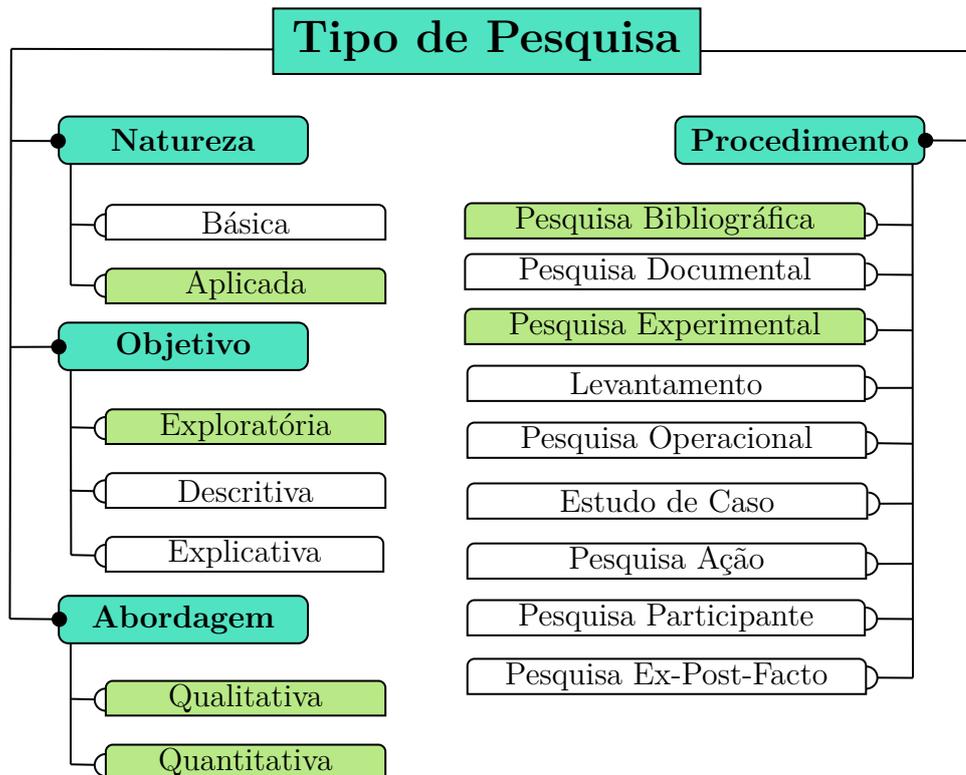
Para Prodanov e Freitas (2013), nenhum tipo de pesquisa é autossuficiente, sendo então necessário a mescla de diferentes tipos, tendo um ou outro ponto mais acentuado, para a obtenção de resultados satisfatórios. Os métodos escolhidos determinam os procedimentos que devem ser utilizados, tanto na coleta de dados e informações quanto na análise.

No que se refere a sua natureza, uma pesquisa pode ser básica ou aplicada. Este trabalho propõe uma DSL, gerando uma ferramenta prática para solucionar algo específico e, sendo assim, pode ser categorizada como uma pesquisa de natureza aplicada.

Do ponto de vista dos seus objetivos, uma pesquisa pode ser classificada como exploratória, descritiva ou explicativa. Este estudo tem como finalidade oferecer mais informações sobre o assunto que investiga, logo, é uma pesquisa exploratória.

A abordagem do problema pode classificar a pesquisa em quantitativa ou qualitativa. Este trabalho aplica os conceitos de ambas as categorias. Foi realizada uma avaliação preliminar da gramática da DSL, onde ocorreu a interpretação dos acontecimentos e a atribuição de significados aos resultados sem requerer necessariamente o uso de métodos e técnicas estatísticas, caracterizando assim essa atividade como uma pesquisa qualitativa. Por outro lado, na análise dos resultados da avaliação experimental da solução construída, foi necessário considerar todos os elementos passíveis de serem quantificados, o que significa explicar em números as opiniões e dados para classificar os resultados. Desta forma, é fundamental o uso de recursos e de técnicas estatísticas, definindo assim essa atividade como uma pesquisa quantitativa.

Figura 1 – Classificação da Pesquisa.



Fonte: Adaptado de Prodanov e Freitas (2013).

E por fim, com relação a seus procedimentos técnicos, uma pesquisa pode ser categorizada como pesquisa bibliográfica, documental, experimental, do tipo levantamento, também chamada de *survey*, pesquisa operacional, estudo de caso, pesquisa *ex-post-facto*, pesquisa-ação e pesquisa participante. Este trabalho realiza uma pesquisa bibliográfica nas atividades executadas para o levantamento de sua base teórica e, além disso, executa uma pesquisa experimental na etapa de avaliação da ferramenta produzida.

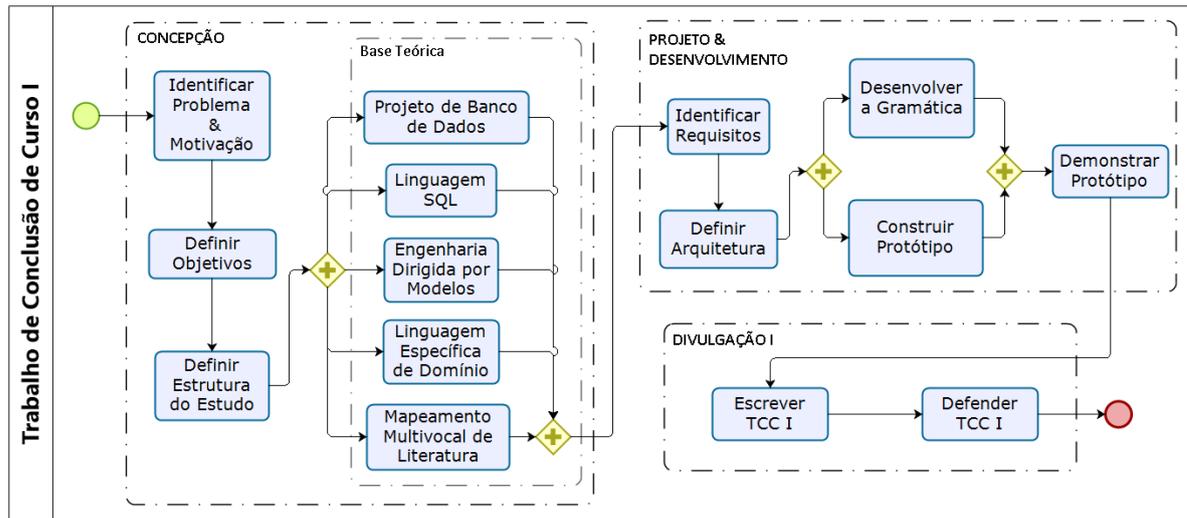
2.2 Desenho de Pesquisa

Para a condução deste estudo foi definido o desenho de pesquisa. Nesta atividade, três fases foram determinantes: a fase decisória, a fase construtiva e a fase redacional.

A fase decisória se refere à escolha do tema e à delimitação dos problemas de pesquisa. A fase construtiva foi onde ocorreu o planejamento das atividades que deveriam ser realizadas. A fase redacional se refere à escrita do trabalho e à análise de dados do andamento do estudo. O desenho de pesquisa foi dividido em duas fases, sendo o TCC I apresentado na Figura 2 e o TCC II na Figura 3.

O projeto foi desenvolvido no TCC I, possuindo atividades bem definidas. Primeiramente aconteceu o processo de concepção, identificando o problema e a motivação

Figura 2 – Desenho de pesquisa.



Fonte: O autor.

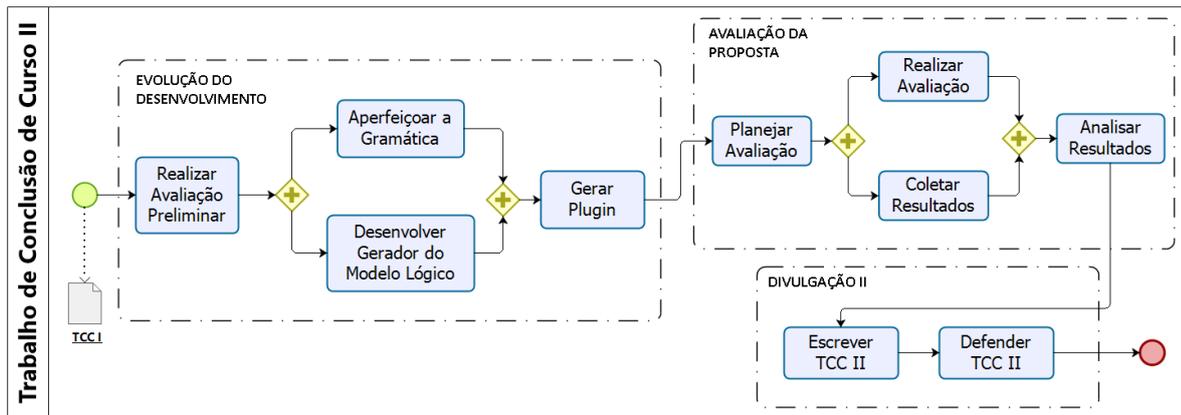
que guiariam todo o estudo. Após, foram definidos os objetivos que este trabalho procura atingir. Então, houve a elaboração da estrutura geral do estudo em que percebeu-se que era necessário criar uma base teórica bem fundamentada. Desta forma, foram caracterizadas as principais áreas que deveriam ser investigadas para o entendimento do domínio do problema. Em paralelo, foi realizado um mapeamento multivocal de literatura com o objetivo de encontrar trabalhos similares a esta proposta.

Com a etapa teórica de concepção realizada, partiu-se então para o projeto inicial da proposta de DSL, visando identificar seus requisitos essenciais e sua arquitetura básica. Feito isto, deu-se início ao desenvolvimento da descrição de gramática da linguagem e sua implementação de fato. Ao fim, um protótipo da ferramenta foi construído. A demonstração de uso do protótipo está descrita no Capítulo 5. Como última etapa do TCC I, foi realizada a especificação do projeto para a apresentação da proposta.

O TCC II teve como entrada os resultados do TCC I e iniciou-se com uma avaliação preliminar da gramática criada para a DSL. Após, ocorreu a evolução do desenvolvimento, com o aperfeiçoamento da gramática e a construção do gerador que realiza o mapeamento entre o modelo conceitual e lógico. Com essas atividades realizadas, um *plugin* foi gerado e incorporado em um *Rich Client Platform* (RCP) Eclipse.

Com o desenvolvimento finalizado, foi realizada a execução de uma avaliação da ferramenta. Essa fase envolveu o planejamento e aplicação de um experimento controlado para coleta de dados. Finalmente, houve a análise dos resultados, a escrita do TCC II e sua posterior defesa. A Tabela 1 apresenta o resumo geral deste estudo.

Figura 3 – Desenho de pesquisa (continuação).



Fonte: O autor.

Tabela 1 – Síntese do TCC.

Assunto	Banco de Dados
Tópico	Projeto e Modelagem de Banco de Dados
Questão de Pesquisa	Uma DSL textual pode auxiliar, em nível conceitual de modelagem, o ensino de projeto e modelagem de banco de dados relacionais?
Objetivo Principal	Propor a especificação e realizar a implementação de uma DSL para o projeto e modelagem de banco de dados relacionais.

Fonte: O autor.

2.3 Lições do Capítulo

Este capítulo forneceu uma visão abrangente sobre a metodologia adotada, bem como de que forma a pesquisa deste trabalho pode ser classificada. Além disso, foi apresentado o desenho estabelecido para a pesquisa, proporcionando assim o entendimento de quais processos foram executados até sua finalização e defesa.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo ocorre a apresentação, de forma abrangente, dos domínios abordados para a realização e compreensão deste trabalho. A Seção 3.1 expõe o que é um projeto de BDs e suas fases de modelagem para a construção de esquemas de BDs. A Seção 3.2 elucida a aplicação da Linguagem de Consulta Estruturada, do inglês *Structured Query Language* (SQL). A Seção 3.3 aborda a Engenharia Dirigida por Modelos, do inglês *Model-Driven Engineering* (MDE), e suas aplicações com as DSLs. A Seção 3.4 trata de conceitos relacionados as Gramáticas Livres de Contexto (GLCs), assim como do formalismo envolvido para o seu desenvolvimento. Por fim, na Seção 3.5 são apresentadas algumas lições aprendidas, bem como os principais tópicos discutidos.

3.1 Projeto de Banco de Dados

Segundo Date (2004) uma estrutura de BD é fundamentalmente um sistema computacional para a manutenção de registros. Sistemas desse tipo têm a finalidade de armazenar dados de forma persistente, bem como permitir que usuários definam, busquem e atualizem esses dados para gerar informações pertinentes quando necessário. Um BD pode ser representado por um modelo de dados, expressado em diferentes níveis e com diferentes técnicas.

Normalmente durante o ciclo de vida do desenvolvimento de software os modelos de dados passam por níveis distintos de transformações. Inicialmente não existia um padrão ou recomendação difundida amplamente na indústria, ou mesmo na academia, para o processo de modelagem de dados. A estratégia para a utilização de diferentes níveis de projeto e representação tem suas origens com o grupo de estudos em Sistemas de Gerenciamento de Bancos de Dados (SGBDs) intitulado *ANSI/X3/SPARK*, ainda na década de 1970 (ANSI, 1975).

Na abordagem proposta, o padrão de definição e especificação de parâmetros e elementos que compreendiam um BD levavam em consideração aspectos conceituais, lógicos e físicos. Esses aspectos eram chamados genericamente de esquemas (do inglês, *schemas*). Esses esquemas na realidade eram fragmentos que serviam, quando em conjunto, para todo o mapeamento da estrutura de um BD. Esses mesmos conceitos continuam em aplicação até os dias de hoje na implementação de BDs em SGBDs.

De acordo com Cougo (2013), as dificuldades existentes antes do estabelecimento da arquitetura de três níveis estava essencialmente em um ponto. Um mesmo modelo de dados concebido para uma aplicação necessitava de diferentes implementações quando aplicados aos SGBDs primitivos da época anterior a proposta de três níveis, incluindo modificações significativas no próprio modelo original. Isso ocasionava como resultado esquemas bastante particulares e reflexos significativos no modelo de dados final.

Tendo essa realidade como fato, um mesmo modelo de dados gerado para uma única aplicação poderia necessitar de um grande número de diferentes esquemas para

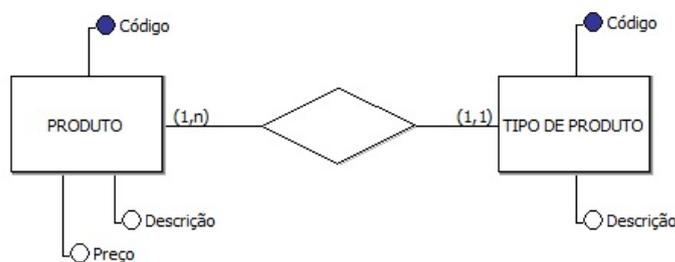
abranger as variações de modos de implementação e de visões externas a serem disponibilizadas aos usuários. As dificuldades provenientes da administração e manutenção de toda essa variedade de modelos levaram o grupo *ANSI/X3/SPARK* a propor o padrão que tem como ideia central a definição de níveis de esquemas relacionados a um modelo de dados (COUGO, 2013). Esse padrão acabou por influenciar a proposta de modelagem conceitual de dados concebida por Chen (1976). Sendo assim, os BDs relacionais até os dias atuais continuam levando em consideração estes conceitos.

A construção de um BD é baseada em um modelo de BD, o qual é uma descrição detalhada dos tipos de informações que devem ser armazenadas. O projeto de BD acontece em três fases distintas de modelagem, onde são gerados o Modelo Conceitual de Dados (MCD), Modelo Lógico de Dados (MLD) e o Modelo Físico de Dados (MFD) (HEUSER, 2009). Para a elaboração de modelos de dados deve-se usar uma linguagem de modelagem de dados. Existem linguagens gráficas e textuais capazes de descrever os modelos em diferentes níveis de detalhamento e abstração.

3.1.1 Modelo Conceitual de Dados

O MCD é a descrição do BD de forma independente da implementação utilizada em um SGBD. Este modelo lista quais dados podem ocorrer no BD, mas não registra como estes dados estão armazenados no nível de SGBD. A técnica mais difundida de MCD é a ER de Chen (1976). Esta abordagem foi tão bem aceita que passou a ser considerada uma referência definitiva para a modelagem de BDs relacionais. É composta basicamente por um método de diagramação e de conceitos que devem ser respeitados. Sendo assim, com esta abordagem os MCDs são representados com Diagramas Entidade-Relacionamento (DERs), como pode ser visto na Figura 4.

Figura 4 – Fragmento de DER.



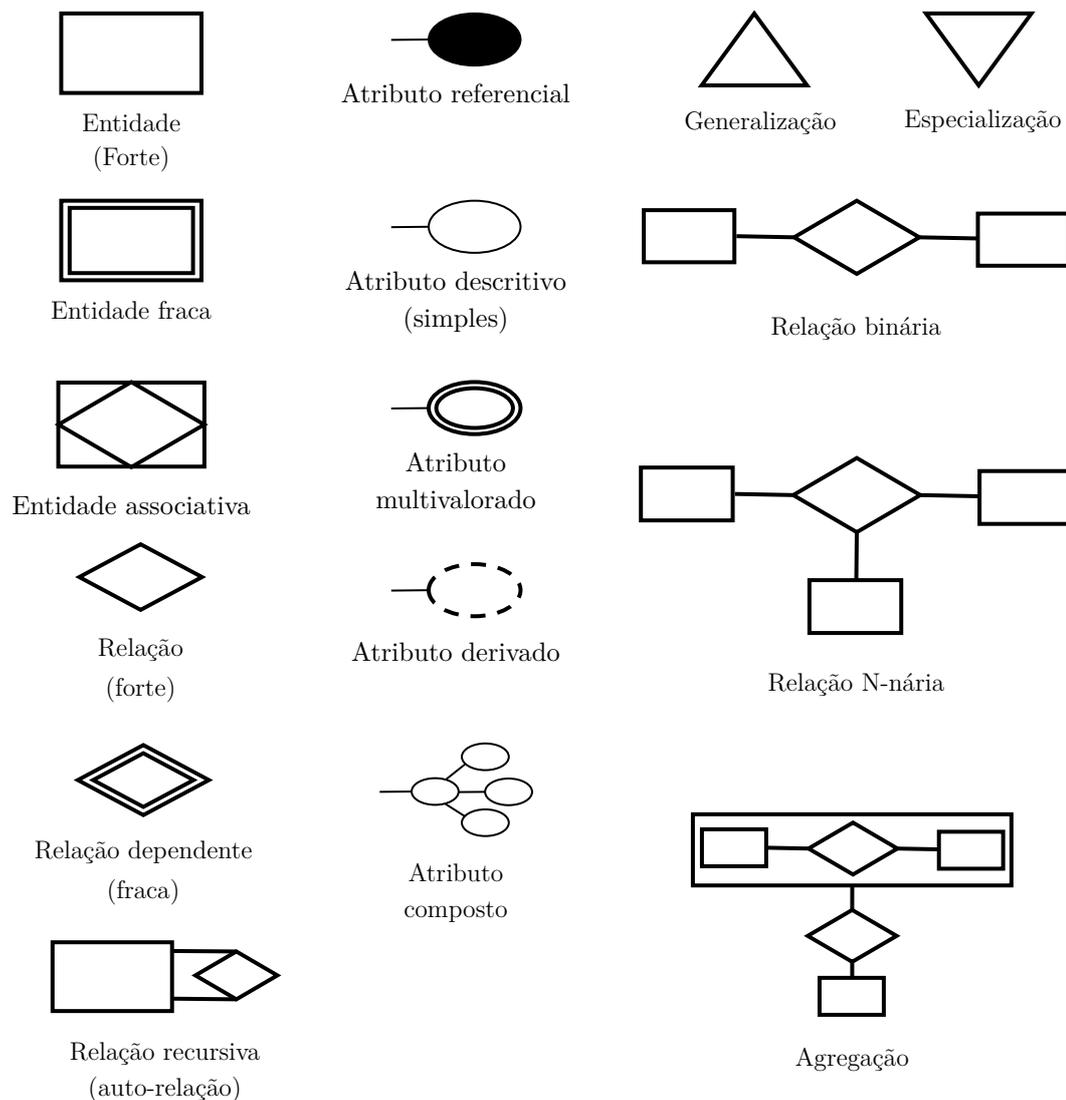
Fonte: Heuser (2009).

3.1.1.1 Modelo Entidade-Relacionamento

Na abordagem ER o conceito principal é o de **entidade**, o qual é uma representação de um conjunto de objetos do domínio modelado. As entidades são simbolizadas

por retângulos. Contudo, nesta abordagem ainda existem também outros conceitos que são essenciais e devem ser analisados. Para melhor compreensão da modelagem conceitual que utiliza esta abordagem, a seguir são apresentados na Figura 5 algumas representações gráficas dos construtores previstos no modelo ER.

Figura 5 – Construtores do modelo ER.



Fonte: O autor.

Uma **Entidade Forte** é uma entidade que para existir não depende da existência de outra(s) entidades(s). De forma inversa, uma **Entidade Fraca** é uma entidade que possui sua existência ou identificação dependente de outra(s) entidade(s). A representação de **Entidade Associativa** serve para dar mais sentido semântico ao modelo, e pode ser usada para simbolizar relações muitos para muitos, contendo referências a colunas de tabelas do mesmo ou de outros BDs.

Os **atributos** podem ser **referenciais**, ou seja, que identificam a entidade, e **des-**

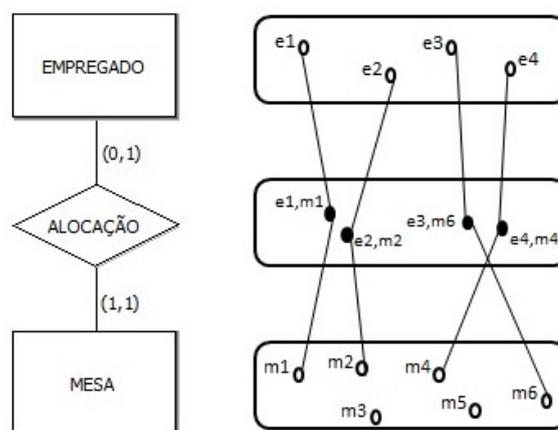
critivos que a descrevem. A maioria dos atributos tendem a ser descritivos e também são chamados de simples ou monovalorados. Contudo, existem outros tipos destes atributos como os **multivalorados**, que podem assumir diversos valores, os **derivados**, que são inferidos a partir de outros atributos, e os **compostos**, que podem ser divididos em várias partes com significados independentes.

No que diz respeito aos **relacionamentos** possíveis entre entidades pode-se citar os **binários**, aqueles que envolvem duas (2) entidades, e os **n-ários** que correspondem a ligação de três (3) ou mais entidades. É comum que as relações compostas especificamente por três entidades sejam chamadas de relações **ternárias**. Além destes tipos ainda é possível haver relações recursivas de uma entidade com ela mesma, também chamado de **auto-relacionamento**.

A abordagem ER ainda suporta conceitos de **generalização/especialização**, que definem um grupo hierárquico de entidades que compartilham atributos em comum, e **agregação** onde a condição de existência é que o relacionamento principal necessita ser necessariamente de muitos para muitos.

Os **relacionamentos** representam a associação entre os objetos e são sinalizados por losangos. A **cardinalidade** de relacionamentos registra o número de ocorrências com que as entidades podem se associar. Existem duas cardinalidades que devem ser atribuídas: a mínima e a máxima. Os **atributos** são características representadas por pequenos círculos conectados as entidades. Na esquerda da Figura 6 é ilustrado um exemplo de DER simples e, na direita, uma representação hipotética do conjunto de ocorrências que podem acontecer entre suas entidades.

Figura 6 – Relacionamentos e cardinalidades.

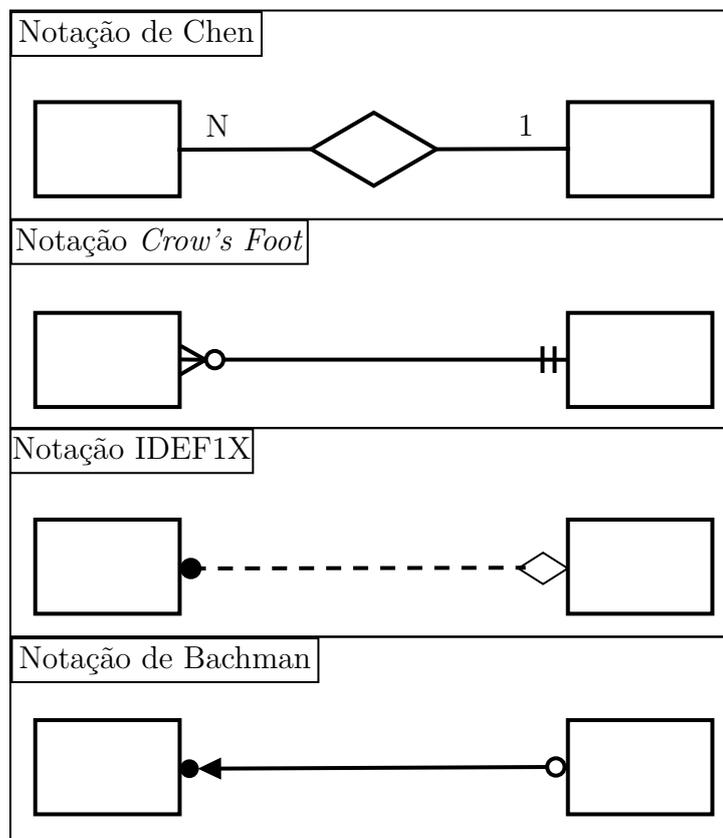


Fonte: Heuser (2009).

3.1.1.2 Notações de Modelagem ER

A abordagem ER teve uma notação estabelecida no trabalho original de Chen, porém muitas outras notações foram propostas com o passar do tempo. Todas essas convenções de diagramação possíveis objetivam o mesmo propósito de modelar BDs, e eventualmente podem ser utilizadas para diagramar até mesmo modelos em nível lógico como o caso da notação *Crow's Foot*, conhecida pelo seu alto nível de aceitação pela indústria. A Figura 7 ilustra como uma relação binária simples com cardinalidade *um para muitos* pode ser representada em quatro (4) diferentes notações.

Figura 7 – Exemplos de notações para modelagem ER.

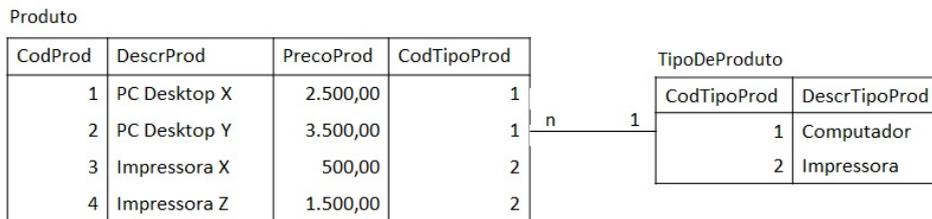


Fonte: O autor.

3.1.2 Modelo Lógico de Dados

Um MLD é definido como um modelo que possui a representação dos objetos, relacionamentos e características de acordo com regras de implementação. Isso significa que esse modelo tem um nível de abstração do ponto de vista do usuário de um SGBD. Ainda assim, o modelo lógico é independente do tipo de SGBD em que é implementado. Na Figura 8 é apresentado um MLD gráfico baseado no modelo da Figura 4.

Figura 8 – Exemplo de MLD para BD relacional.



Fonte: Heuser (2009).

Esse tipo de modelo deve necessariamente respeitar conceitos tais como chaves de acesso, controle de chaves duplicadas, normalização, integridade referencial, controle de redundância de dados, entre outros. Este modelo é intrinsecamente relacionado a fase de projeto (COUGO, 2013). É importante salientar que essa é uma forma direcionada a um aspecto gráfico, porém existem meios alternativos de se representar estruturalmente o mesmo modelo de forma textual (MARTELLI; FILHO; CABRAL, 2018). Isso é mostrado na Figura 9 que apresenta a definição da mesma estrutura descrita na Figura 8.

Figura 9 – Estrutura de um modelo lógico descrita de forma textual.

TipoDeProduto (<u>CodTipoProd</u> , DescrTipoProd) Produto (<u>CodProd</u> , DescrProd, PreçoProd, CodTipoProd) CodTipoProd referencia TipoDeProduto
--

Fonte: Heuser (2009).

A obtenção de um MLD se dá mediante a aplicação de regras de derivação sobre um MCD já construído. Entretanto, não é raro que desenvolvedores e analistas experientes comecem diretamente pelo processo de modelagem lógica, ignorando a modelagem conceitual. Isso ocorre pois esse modelo não se preocupa somente com a representação dos objetos observados no domínio analisado, mas também com outros elementos como chaves, métodos de acesso, formatos de campo, etc. Isso implica, em uma última observação, que do ponto de vista formal da definição da abordagem ER, esse modelo não se enquadra fielmente como um modelo ER (WEST, 2011).

3.1.3 Modelo Físico de Dados

Um MFD caracteriza-se como um modelo em que a representação dos objetos de BD já estão em um nível físico de implementação das ocorrências, ou instâncias, das entidades de relacionamentos. Cada SGBD pode definir diferentes modos de implementação física das características e recursos indispensáveis para o armazenamento e manipulação das estruturas de dados (COUGO, 2013).

Em geral os modelos físicos apresentam dois aspectos bem representados. Primeiramente, existem as ocorrências ou instâncias, seus relacionamentos e a disposição básica dos elementos. O outro aspecto diz respeito a alocação nos diversos níveis de agrupamentos possíveis, como as tabelas, linhas (registros), colunas (campos) e blocos (WEST, 2011). Em suma, é a materialização dos objetos de BD em um esquema interno de um dado SGBD a partir da execução de uma sequência lógica de instruções em SQL.

3.2 Linguagem SQL

Para a manipulação dos dados a linguagem SQL é o padrão utilizado por sistemas de BDs relacionais disponíveis no mercado. A SQL teve sua gênese originalmente nos laboratórios da IBM Research, na década de 1970, com o nome inicial de SEQUEL (CHAMBERLIN; BOYCE, 1974). A SQL é uma linguagem com a versão estável mais recente lançada em 2016 e denominada SQL:2016, possuindo um total de mais de 2000 páginas de especificação^{1,2,3,4}. Entretanto, é importante salientar que ao mesmo tempo em que a maioria dos produtos do mercado trabalham com a SQL, estas soluções também deixam de oferecer suporte a determinados aspectos ou ainda os implementa de uma forma diferente da especificação oficial.

A SQL é uma única linguagem, mas comumente é categorizada conforme a funcionalidade das suas instruções. A primeira categoria é chamada Linguagem de Definição de Dados, do inglês *Data Definition Language* (DDL). Entre os comandos dessa categoria estão o CREATE, ALTER, DROP e TRUNCATE. A segunda categoria é nomeada de Linguagem de Definição de Dados, do inglês *Data Manipulation Language* (DML). Entre os comandos categorizados como DML estão o INSERT, UPDATE e DELETE. A terceira categoria é chamada de Linguagem de Consulta de Dados, do inglês *Data Query Language* (DQL), a qual possui apenas o comando SELECT. A quarta categoria é denominada Linguagem de Transação de Dados, do inglês *Data Transaction Language* (DTL), a qual detém comandos como COMMIT e ROLLBACK.

A SQL ainda utiliza uma série de cláusulas (*e.g.* FROM, WHERE, GROUP BY, ORDER BY, HAVING, DISTINCT, UNION), operadores lógicos (*e.g.* AND, OR, NOT), operadores relacionais (*e.g.* <, >, <=, >=, =, <>) e funções de agregação (*e.g.* AVG, SUM, COUNT, MAX, MIN). A aplicação destes termos e palavras reservadas, associado a características inspiradas na álgebra relacional, fundamenta a base da SQL utilizada pelos SGBDs.

¹ <https://iso.org/standard/63555.html>

² https://standards.iso.org/ittf/PubliclyAvailableStandards/c065143_1SO_1EC_T_R_19075-5_2016.zip

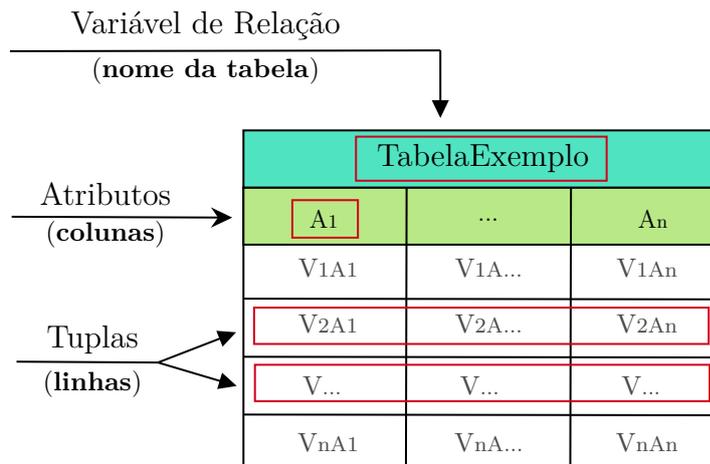
³ https://standards.iso.org/ittf/PubliclyAvailableStandards/c067367_1SO_1EC_T_R_19075-6_2017.zip

⁴ https://standards.iso.org/ittf/PubliclyAvailableStandards/c069776_1SO_1EC_T_R_19075-7_2017.zip

3.2.1 Bancos de Dados Relacionais

Os SGBDs relacionais suportam o modelo de dados relacional. O modelo de dados relacional foi proposto por Codd (1970) e tem como premissa a modelagem orientada a tabelas. Um ponto importante a ser citado é o fato de praticamente todos os SGBDs relacionais do mercado utilizarem SQL para a criação e manipulação dos dados.

Figura 10 – Modelo de dados relacional.



Fonte: O autor.

A Figura 10 mostra o esquema estrutural de um modelo relacional. Nele uma tabela, também chamada de entidade, é definida por um nome e um número fixo de atributos com seus tipos de dados indicados. Cada tabela deve ter um ou mais atributos identificadores, chamados de chaves, os quais auxiliam a manter a integridade referencial dos dados. Um registro, também chamado de ocorrência ou tupla, corresponde a uma linha na tabela e consiste nos valores de cada atributo. Uma relação, portanto, consiste em um conjunto de registros associados (referenciados) entre tabelas através das variáveis de relação (RAMAKRISHNAN; GEHRKE, 2003).

3.3 Model-Driven Engineering

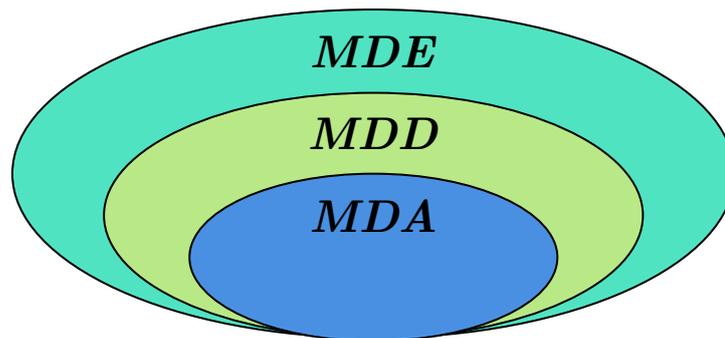
Conceitualmente um modelo é uma representação, protótipo ou exemplo que se tem por objetivo reproduzir ou imitar de alguma forma. A construção de modelos são pontos centrais e importantes em diferentes áreas científicas. Na matemática, física e química, por exemplo, o emprego de modelos é tido como vital para a investigação teórica e prática em diferentes campos de estudo (BAILER-JONES, 2009).

Em uma análise mais profunda, Brambilla, Cabot e Wimmer (2017) discutem que, considerando-se a premissa de que um observador e suas observações alteram a própria realidade, é possível se concluir que tudo na percepção de um indivíduo é um modelo,

já que absolutamente nada pode ser processado pela mente humana sem ser modelado. Em resumo, a criação de modelos é uma tarefa de abstração de domínios e conceitos do mundo real. Sendo assim, não é de surpreender que os modelos tenham se tornado cruciais e amplamente adotados também em áreas técnicas como mecânica, engenharia civil e, por fim, na ciência da computação, engenharia da computação e ES.

A MDE, também chamada de *Model-Driven Software Engineering* (MDSE), é uma abordagem da ES para desenvolvimento de *software* que tem essencialmente modelos como saídas principais de algum processo. Essa abordagem resulta em programas ou atividades de computador executados em *hardware* ou *software* que são gerados automaticamente a partir de modelos (SOMMERVILLE, 2011). Conceitualmente, a MDE fornece apoio a outros conceitos, como o *Model-Driven Development* (MDD) e a *Model-Driven Architecture* (MDA). Na Figura 11 a relação entre estes conceitos é ilustrada.

Figura 11 – Relação de MDE, MDD e MDA.



Fonte: Adaptado de Ameller (2009).

O MDD é um paradigma de desenvolvimento que usa modelos como o principal artefato do processo de desenvolvimento. No MDD geralmente a implementação é gerada de forma semiautomática a partir dos modelos. Apesar de serem vistas como a mesma coisa, o conceito da MDE tem origem na MDA, proposta em 2001 pelo *Object Management Group* (OMG). Com diferenças sutis, Sommerville (2011) afirma que a MDA concentra-se nos estágios de projeto e implementação do processo de desenvolvimento de software, sendo muito similar ao MDD, porém implementando diretrizes específicas da OMG. Desta forma, conclui-se que a MDA é um subconjunto do MDD. Por outro lado, a MDE pode abordar muitos outros tópicos do processo de ES, entre eles a engenharia de requisitos baseada em modelos, processos de software para desenvolvimento baseado em modelos, ou ainda, testes baseados em modelos.

A MDE, como uma metodologia, auxilia a aplicação das vantagens da modelagem nas atividades de ES. Para (BRAMBILLA; CABOT; WIMMER, 2017) essa abordagem leva em consideração quatro aspectos fundamentais, listados a seguir.

1. **Conceitos:** os componentes que constroem a metodologia, abrangendo desde artefatos de linguagem até atores, e assim por diante;
2. **Notações:** A maneira como os conceitos são representados, ou seja, as linguagens usadas na metodologia;
3. **Processos e Regras:** As atividades que levam à elaboração do produto final, as regras para sua administração e controle, e as afirmações sobre as propriedades desejadas (correção, consistência, etc) dos produtos ou do próprio processo;
4. **Ferramentas:** Aplicações que facilitam a execução de atividades ou seu controle, abrangendo o processo de produção e apoiando o desenvolvedor no uso das notações.

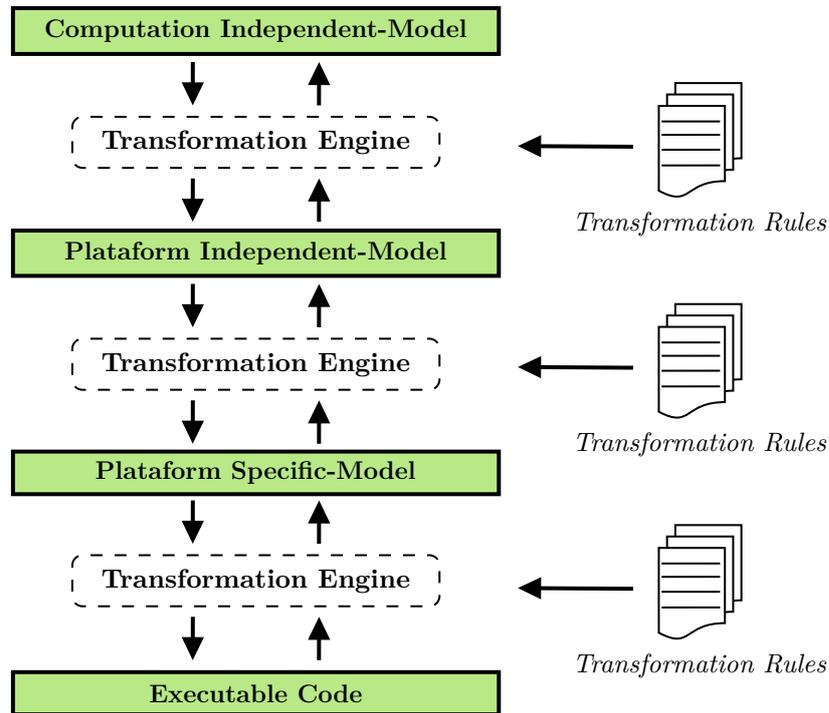
A motivação por trás da MDE é a ideia de se aumentar o nível de abstração do processo de desenvolvimento em geral, para então assim capturar sistemas ou processos como uma coleção de modelos reutilizáveis. Logo, ela visa reduzir a dificuldade associada ao desenvolvimento de sistemas de software, em geral mais complexos, por meio do uso de técnicas de modelagem que suportam a separação de interesses e geração automatizada de artefatos de sistemas a partir de modelos (KLEPPE et al., 2003).

De uma forma objetiva, a abordagem MDA, ou ainda a MDD, é a forma de se realizar a MDE. Essa abordagem define três camadas que devem ser usadas como pilares para todo o processo, listados a seguir. A relação conceitual entre esses níveis, com o uso de mecanismos de transformação e regras de transformação, é exemplificado na Figura 12.

1. **Computation-Independent Models (CIMs):** descrevem objetos de negócio e as atividades independentemente de sistemas de suporte;
2. **Platform-Independent Models (PIMs):** descrevem como os processos de negócio são suportados por sistemas, vistos como caixas-pretas funcionais, ou seja, desconsiderando as restrições associadas às tecnologias candidatas;
3. **Platform-Specific Models (CIMs):** descrevem os componentes do sistema conforme implementados por tecnologias específicas.

Para melhor esclarecimento, os mecanismos de transformação podem ser entendidos como geradores que tem como entrada a descrição de modelos. Esses geradores devem processar tais modelos, tendo em si implementados uma série de regras de transformação. Por exemplo, no contexto deste estudo um artefato de entrada seria o modelo feito utilizando a DSL da proposta, e um mecanismo de transformação seria o gerador do modelo lógico. Para tanto, este gerador deve ter descritas ao todas ou parte das regras possíveis de transformação do modelo conceitual para o modelo lógico, ou seja, deve realizar o mapeamento do modelo da DSL para um modelo relacional equivalente. É importante salientar que estes mecanismos de transformação podem estar dispostos em vários níveis.

Figura 12 – Níveis de abstração do MDA.



Fonte: Adaptado de Frantz (2012).

Sendo assim, no contexto exemplificado é possível haver até dois (2) mecanismos de transformação: um que faz a transição do modelo conceitual para o lógico e outro que gera código SQL. Este último gerador ainda pode realizar esta tarefa a partir do modelo lógico previamente gerado ou mesmo diretamente do modelo conceitual.

A separação de interesses da MDA baseia-se, por exemplo, na exploração de diferentes DSLs, cada uma fornecendo construções baseadas em abstrações que são específicas do domínio de um sistema. Por conta disto, as DSLs podem desempenhar um papel de destaque na MDE (SCHMIDT, 2006).

3.3.1 Domain-Specific Language

Para Deursen, Klint e Visser (2000) uma DSL é uma linguagem de programação ou linguagem de especificação executável que oferece, por meio de notações e abstrações apropriadas, poder expressivo focado e, geralmente, restrito a um domínio de problema específico. Assim como outras linguagens, as DSLs devem apresentar um conjunto de sentenças bem definidas por uma sintaxe e semântica própria. Para Fowler (2010) uma DSL é definida como uma linguagem de programação de computadores com expressividade limitada e focada em um domínio particular. Entre exemplos conhecidos de DSLs estão:

- SQL, para bancos de dados;

- *Cascading Style Sheets* (CSS), para *layout* de páginas *Web*;
- *eXtensible Markup Language* (XML), para codificação de dados;
- *Unified Modeling Language* (UML), para projeto de software;
- *Systems Modeling Language* (SysML), para modelagem de sistemas;
- *VHSIC Hardware Description Language* (VHDL), para projeto de hardware;
- \LaTeX , para tipografia de documentos.

Segundo Faveri (2013), apesar do termo DSL poder intuitivamente remeter para um campo de estudos recente, de fato isso não é uma realidade. Por exemplo, a APT é uma DSL para programação de máquinas controladas numericamente que foi desenvolvida por dois anos a partir de 1957 (ROSS, 1981), enquanto o formalismo de especificação de sintaxe *Backus-Naur Form* (BNF), o mais usado para notação das linguagens de programação nos dias de hoje, remonta o final da década de 1950 (BACKUS, 1959).

Em razão disso é possível encontrar na literatura muitos estudos que abordam conceitualmente DSLs, porém com diferentes terminologias. Entre estas, pode-se citar: *Languages for specialized application* (SAMMET, 1972); *Special-purpose languages* (WEXELBLAT, 1981); *Application Languages* (MARTIN, 1982); *Task-specific programming languages* (NARDI, 1993); *Specialized languages* (BERGIN JR.; GIBSON JR., 1996).

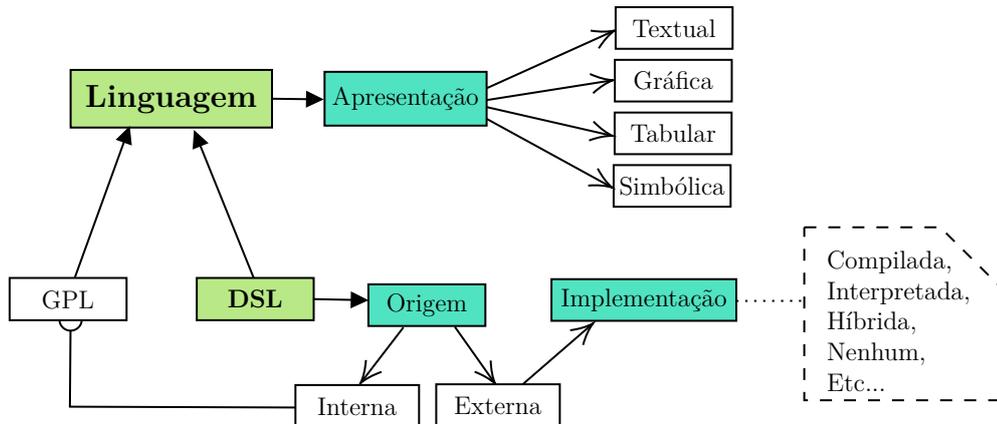
A aplicação de DSLs permite que softwares sejam desenvolvidos de forma mais rápida e eficaz. A maior vantagem observada no uso de DSLs é que o conhecimento necessário para a sua aplicabilidade é abstraído para outro nível. Desta forma, especialistas do domínio podem entender, validar e modificar o código, adaptando o modelo as suas necessidades, tornando o impacto das mudanças mais fácil de ser compreendido. Também existe um aumento significativo na produtividade, confiabilidade, facilidade de uso e flexibilidade (DEURSEN; KLINT; VISSER, 2000).

Segundo Mernik, Heering e Sloane (2005) as DSLs podem ser classificadas sob três dimensões diferentes: **origem**, **aparência** e **implementação**. As dimensões de classificação de DSL são exibidas na Figura 13. Em relação à origem de uma DSL, as opções existentes são as DSLs **internas** e **externas**.

Uma DSL **interna** é projetada a partir das regras sintáticas e semânticas da gramática de uma linguagem já existente, podendo ser essa uma linguagem de propósito geral, do inglês *General-Purpose Programming Language* (GPL), ou outra DSL. Sendo assim, para seu funcionamento correto uma DSL interna acaba transferindo todas as atividades de verificação léxica, sintática, semântica e de transformação de código ao compilador da linguagem hospedeira.

Uma DSL **externa** é uma linguagem com sintaxe distinta e que depende de uma infraestrutura própria para a análise léxica, sintática, semântica, interpretação, compilação, otimização e geração de código. Se comparada a uma GPL, uma DSL externa possui

Figura 13 – Dimensões de uma DSL.



Fonte: Adaptado de Favari (2013).

especificidades similares, porém seus recursos são restritos ao domínio de aplicação para o qual a linguagem é projetada.

No que diz respeito à dimensão de **aparência**, uma DSL pode ser classificada como **textual**, **gráfica**, **tabular** e **simbólica**. Quando no formato textual as DSLs permitem que o domínio seja expressado com caracteres, os quais são então combinados gerando palavras, expressões, sentenças e instruções que seguem as regras gramaticais previamente estabelecidas na linguagem. As DSLs não textuais seguem a mesma lógica, mas utilizando-se de modelos gráficos para permitir que o usuário possa expressar conhecimento de domínio com um maior nível de compreensão e empregando para tal o uso de símbolos, tabelas, figuras e conectores.

E finalmente, no que se refere a dimensão de **implementação**, as DSLs podem ser classificadas tendo em vista a perspectiva de sua execução. Essas classificações formam quatro grupos: (i) DSLs de execução bem definidas (*e.g.* Excel Macro Language); (ii) DSLs que servem de entrada para geradores de aplicação; (iii) DSLs não executáveis mas úteis como entrada de geradores de aplicação; (iv) DSLs não projetadas para serem executadas.

É prática usual que o principal aspecto levado em consideração para a construção de uma DSLs deve ser a sua **origem**, pois cada abordagem apresenta vantagens e desvantagens específicas que são inerentes a cada tipo (FOWLER, 2010). Apesar das DSLs externas poderem ter um esforço associado a sua construção muitas vezes maior do que o de uma DSL interna, atualmente existem ferramentas que dão grande suporte a construção de DSLs. Estas ferramentas são conhecidas como *Language Workbenches* (LWs) e aplicam conceitos de programação orientada a linguagens, fornecendo um nível de abstração maior no que diz respeito as questões complexas de infraestrutura (FOWLER, 2005).

3.3.2 *Language Workbenches*

O desenvolvimento de uma DSL não é tarefa trivial pois, como são linguagens de programação, possuem uma sintaxe que é, por consequência lógica, definida por uma gramática. Desta forma, se faz necessária a utilização de ferramentas que suportem a definição dos conceitos para a nova linguagem (FOWLER, 2005).

Os LWs são ferramentas que fornecem mecanismos de infraestrutura para a implementação de linguagens de programação, tornando assim a criação de linguagens mais acessível (Wachsmuth; Konat; Visser, 2014). Entre os mecanismos fornecidos nesses ambientes está a formatação automática, validação com base nas restrições descritas na gramática, *syntax highlighting*⁵ e *syntax completion*⁶. A seguir são citados alguns dos mais conhecidos LWs da atualidade:

- **Xtext:** lançado em 2006, o Xtext é um *framework* de código aberto para o desenvolvimento linguagens de programação textuais, com integração com o ambiente de desenvolvimento integrado, do inglês *Integrated Development Environment* (IDE), Eclipse. Para especificar uma linguagem, o desenvolvedor descreve uma gramática no Xtext. Essa gramática descreve como um modelo *Ecore* deve ser derivado de uma notação textual. A partir dessa definição, um gerador de código deriva um analisador ANTLR e as classes para o modelo de objetos. O Xtext também tem um gerador Xtend editável, o que dá a capacidade de se gerar código para qualquer outra gramática. O Xtext inclui recursos inerentes ao IDE Eclipse como *syntax highlighting*, *code completion*, *static analysis*, *source-code navigation* e outros. Atualmente está na versão 2.18.0.
- **Sirius:** lançado em 2007 em um esforço entre as empresas Thales e Obeo, atualmente o *framework* Sirius é um projeto de licença aberta mantido pela Eclipse Foundation, cujo objetivo é permitir a criação de ferramentas de modelagem. Com o Sirius é possível especificar DSLs visuais e gerar assim a infraestrutura de editores gráficos. O Sirius, assim como o Xtext, possui integração com recursos específicos do ambiente Eclipse. Atualmente está na versão 6.2.1.
- **JetBrains MPS:** o JetBrains MPS é um sistema desenvolvido pela JetBrains, empresa da República Tcheca, que usa edição projetiva. Essa abordagem permite aos desenvolvedores uma melhor compreensão, o que a diferencia de outros LWs. Também possui funções comuns de IDEs integrado a seu ambiente de desenvolvimento. Está atualmente na versão 2019.1.1 sob a licença Apache 2.0.
- **MetaEdit+:** o MetaEdit+ é LW proprietário desenvolvido pela companhia finlandesa MetaCase para criar e utilizar DSLs. Possui duas versões nomeadamente

⁵ Realce de código-fonte com cor, negrito, etc. Serve para indicar sua estrutura sintática.

⁶ Uma função, como em um mecanismo de busca, que fornece uma ou mais opções de palavras reservadas previstas na gramática a partir dos caracteres que um usuário já inseriu.

MetaEdit+ Workbench e *MetaEdit Modeler*. O *Workbench* inclui ferramentas para projetar e usar/testar linguagens de modelagem enquanto o *Modeler* inclui ferramentas para se utilizar linguagens de modelagem. Normalmente, o *MetaEdit+ Workbench* é usado pelos desenvolvedores que projetam uma DSL do domínio para um projeto. Em seguida, essa linguagem de modelagem é usada para desenvolver produtos finais com o apoio do *MetaEdit+ Modeler*. Atualmente está na versão 5.5 SR1.

É interessante salientar que a capacidade de definir referências cruzadas fornecida por LWs é o que os torna mais atrativos para o desenvolvimento rápido de DSLs. Observando por um lado prático, essa capacidade é ainda mais notável em LWs como o Xtext, uma vez que essa plataforma proporciona abstração suficiente para a realizar a especificação de GLCs executáveis.

3.4 Gramáticas Livres de Contexto

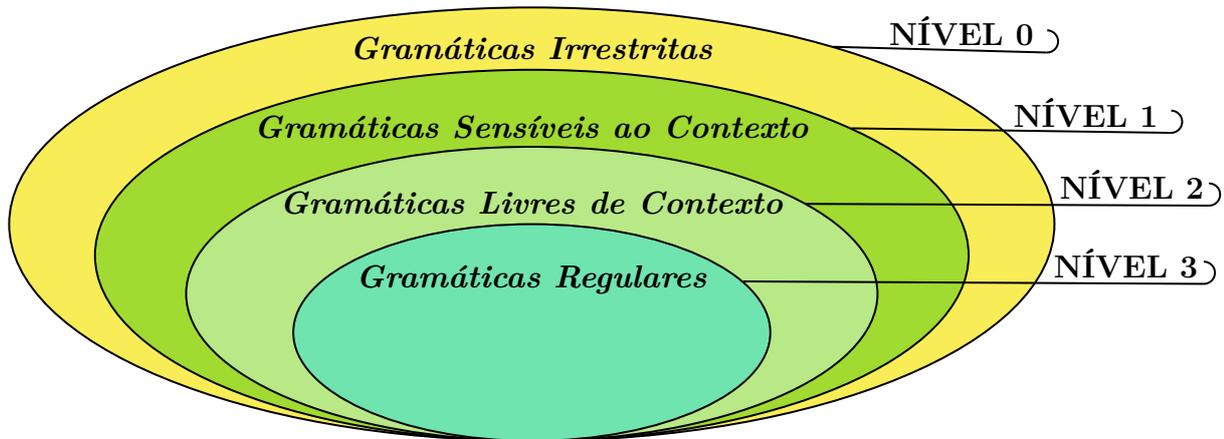
Na matemática, ciência da computação e linguística, uma linguagem formal consiste em palavras cujas letras são retiradas de um alfabeto e são bem formadas de acordo com um conjunto específico de regras. Existem diversas abordagens computacionais que tem por objetivo proporcionar que os computadores possam compreender a Linguagem Natural (LN) falada e/ou escrita por humanos. Compreender neste cenário significa, no mínimo, reconhecer o contexto, fazer análise sintática, semântica, léxica e morfológica. Esse conjunto de técnicas computacionais compõem uma área denominada Processamento de Linguagem Natural (PLN) (JURAFSKY; MARTIN, 2009).

Contudo, mesmo com a evolução tecnológica proporcionada pelo esforço de diversos pesquisadores, a comunicação entre homem e máquina através de LN continua sendo um desafio. Ainda assim, por ser uma área vasta o PLN engloba diversos espectros onde as linguagens podem ser classificadas conforme as regras de formação e sua expressividade. Neste sentido a figura do filósofo, linguista e escritor norte-americano Noan Chomsky é um expoente, uma vez que ele estabeleceu uma teoria em 1959 que descreve um conjunto de quatro (4) níveis para a classificação de qualquer linguagem. A teoria mostra que há quatro (4) classes de gramáticas capazes de gerar diferentes linguagens (LINZ, 2016).

Posteriormente o modelo proposto foi refinado por outros pesquisadores, e conta atualmente com versões de até sete (7) níveis. Em razão do foco deste estudo ser uma GLC, será abordado o nível dois (2) da definição clássica de Chomsky, expressada na Figura 14.

Nesta hierarquia toda a classe de gramática é um subconjunto da classe do nível imediatamente acima. Por exemplo, toda linguagem livre de contexto também é sensível ao contexto, porém a situação inversa não é sempre verdadeira. As GLCs são poderosas

Figura 14 – Representação dos níveis da Hierarquia de Chomsky.



Fonte: O autor.

o suficiente para descrever a sintaxe da maioria das linguagens de programação utilizadas na atualidade.

Segundo Linz (2016), as gramáticas consistem de uma coleção de regras de substituição, também chamadas de produções. Cada regra aparece como uma linha na gramática, compreendendo um símbolo e uma cadeia separados por uma seta. Estas regras descrevem como as cadeias podem ser reconhecidas ou formadas. Uma GLC é uma gramática formal em que cada regra de produção é da forma $V \rightarrow w$, onde V é um único símbolo não-terminal (variável) e w é uma cadeia de terminais e/ou variáveis, podendo ainda ser também uma cadeia vazia (palavra vazia). O termo "livre de contexto" expressa basicamente o fato de que um V não terminal sempre pode ser substituído por w , independentemente do contexto em que ocorre.

Demonstrando em um exemplo simples, a linguagem $L = \{a^n b^n : n \geq 1\}$ define uma estrutura onde todas as cadeias de caracteres são não vazias, de tamanho par e tem a primeira metade preenchida por "a"s e a segunda metade sempre por "b"s. Algumas palavras que essa linguagem aceita são ab , $aabb$ e $aaabbb$. Um exemplo de GLC que consegue gerá-las é $S \rightarrow aSb \mid ab$, contudo é importante salientar que dependendo da linguagem diferentes gramáticas podem gerá-las.

3.4.1 Formalismo de Backus-Naur

A primeira representação da gramática de uma linguagem de programação foi apresentada por John Backus, em 1959, para expressar a linguagem Algol. Esta notação seguia o mesmo padrão de GLCs estudado por linguistas como Chomsky, dando origem a notação BNF (EDELWEISS; LIVI, 2014).

A notação BNF é utilizada para expressar GLCs e, em geral, é amplamente empregada para a especificação de GPLs. Porém é uma notação que não se restringe apenas

as GLPs, sendo usada também para expressar protocolos de comunicação, formato de dados e outros tipos de linguagens como as específicas de domínio. Em geral não é usado em língua natural porque essa notação não é adequada para gramáticas que dependem de contexto.

Na notação BNF é simples e muito similar ao modelo utilizado no campo da linguística para definição de GLCs pois nela: **(I)** as variáveis são palavras entre os símbolos $\langle \text{ e } \rangle$; **(II)** as palavras não delimitadas são terminais; **(III)** a regra geral de produção $S \rightarrow \alpha$ é representada como $S ::= \alpha$, ou ainda, $\langle SNT \rangle ::= \langle \alpha T \mid \alpha T + SNT \rangle$.

Para compreensão, o " $::=$ " indica "*é definido como*", o \mid indica "*ou*" e os colchetes angulares ($\langle \rangle$) distinguem nomes de regras de sintaxe, chamados de símbolos não terminais (SNT), dos símbolos de terminais (αT) que são escritos exatamente como devem ser representados, sendo estes últimos também conhecidos como palavras reservadas. Um exemplo da aplicação da notação BNF para definir a estrutura de uma mini-linguagem de programação pode ser expressada conforme a Figura 15.

Figura 15 – Exemplo de mini-linguagem definida com BNF.

```

<program> ::=
    program
        <declaration_seq>
    begin
        <statements_seq>
    end ;

```

Esse modelo estabelece a estrutura para um programa genérico de forma abstrata, em alto nível. Ele consiste de uma regra chamada **program** que inicia com a palavra-chave **program** (essa redundância caracteriza a linguagem não ser sensível ao contexto). Em seguida vem a regra de **declaration_seq**, e após a palavra-chave **begin** e a sequência de instruções definidas na regra **statements_seq**. Finalmente, a especificação da palavra-chave **end** seguida de um ponto e vírgula (;).

3.5 Lições do Capítulo

Os conceitos mais importantes para este trabalho foram apresentados neste capítulo. Foi necessário investigar dois grandes domínios, sendo eles: (i) projeto e modelagem de BD; (ii) MDE. Dentre os temas abordados destaca-se a Seção 3.3.1, a qual apresenta definições importantes para a compreensão do que é de fato uma DSL, bem como são definidas as GLCs para representá-la. A Seção 3.3.2 também merece destaque pois cita alguns LWs, dentre eles o Xtext que acabou por ser a ferramenta selecionada para a construção da solução proposta neste estudo.

4 MAPEAMENTO MULTIVOCAL DE LITERATURA

Esse capítulo descreve os procedimentos adotados para a investigação da literatura realizada para este estudo. Para este propósito ser alcançado foi realizado o planejamento e a execução de um Mapeamento Multivocal da Literatura, do inglês *Multivocal Literature Mapping* (MLM). Para isto foram aplicadas diretrizes bem estabelecidas conceitualmente por diversos trabalhos (KITCHENHAM; CHARTERS, 2007; PETERSEN et al., 2008; NAKAGAWA et al., 2017).

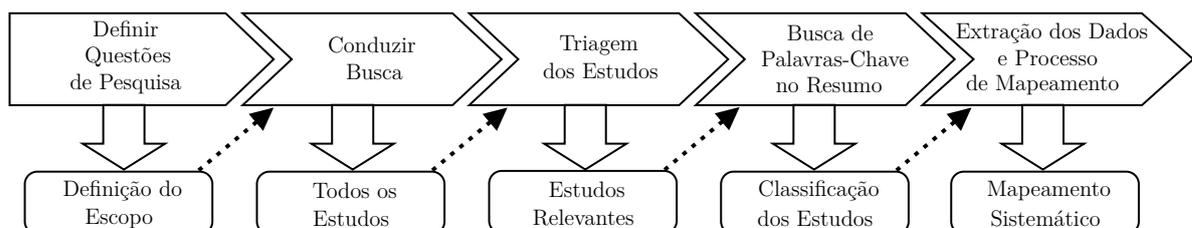
O protocolo seguido é descrito de forma detalhada na Seção 4.1. Na Seção 4.2 é relatado todo o processo de execução do mapeamento multivocal. Os resultados provenientes da execução são analisados e discutidos na Seção 4.3. As ameaças ao MLM são debatidas na Seção 4.4 e, por fim, são pontuadas algumas lições do capítulo na Seção 4.6.

4.1 Protocolo

Um MLM é uma forma de Mapeamento Sistemático de Literatura, do inglês *Systematic Literature Mapping* (SLM), que inclui a literatura cinza. Os MLMs são úteis para pesquisadores e para profissionais pois fornecem visões abrangentes sobre o estado da arte e da prática em uma determinada área. Neste estudo conduziu-se um MLM utilizando o processo de SLM definido por Petersen (PETERSEN et al., 2008) e as diretrizes propostas por Garousi, V. and Felderer, M. and Mäntylä (2019) para pesquisa na literatura cinza.

Um SLM envolve uma busca para determinar que tipos de estudos abordam as Questões de Pesquisa (QPs) que se objetivam investigar, além de possibilitar a classificação e extração de dados que possam gerar informações relevantes (Bailey et al., 2007). Com o propósito de ter a obtenção de resultados confiáveis e reproduzíveis em um SLM, é vital que uma série de atividades bem definidas e estruturadas sejam seguidas (NAKAGAWA et al., 2017). Na Figura 16 é apresentado o processo de SLM realizado neste estudo, com base na proposição realizada por Petersen et al. (2008).

Figura 16 – Processo de mapeamento sistemático.



Fonte: Adaptado de Petersen et al. (2008).

4.1.1 Questões de Pesquisa

Levando-se em consideração o objetivo geral deste trabalho, o qual é desenvolver uma DSL para a modelagem conceitual de BDs, foram formulados os seguintes questionamentos que serviram de guia para o restante do MLM:

- **QP1.:** Qual o estado da arte do desenvolvimento de DSLs para transformação de modelos em ER?
- **QP1.1.:** Quais são as metodologias, técnicas e propostas de refinamento (desenho automatizado) baseado em modelos de dados?
- **QP1.2.:** Qual é o ferramental utilizado como apoio ao desenvolvimento dessas DSL?
- **QP1.3.:** Quais são as representações de objetos de BD adotadas ou sugeridas nas DSL propostas?
- **QP2** Quais os métodos de avaliação utilizados nos estudos?
- **QP2.1.:** Quais os pontos positivos e negativos observados na execução dos estudos?
- **QP2.2.:** Quais desafios são apontados após execução dos estudos empíricos?
- **QP3.:** Quais são as ferramentas para modelagem conceitual de BDs?
- **QP3.1.:** Quais são as notações que estas ferramentas usam?
- **QP3.2.:** Quais são os níveis de modelagem de BD (Conceitual, Lógico, Físico) que estas ferramentas suportam?

4.1.2 Fontes de Busca

As bibliotecas digitais são a principal fonte de busca em um SLM (PETERSEN et al., 2008). Para a escolha das fontes de busca desse SLM são considerados três requisitos obrigatórios que as bases devem contemplar: **(i)** possuir mecanismo de pesquisa baseado na *Web*; **(ii)** ser capaz de usar palavras-chave durante a pesquisa, e; **(iii)** abranger estudos primários da grande área da Ciência da Computação. Na Tabela 2 são listadas as bibliotecas digitais selecionadas para este SLM.

Tabela 2 – Bibliotecas digitais utilizadas.

Fonte	Endereço	Tipo
ACM Digital library	<i>dl.acm.org</i>	Híbrida
IEEE Xplore	<i>ieeexplore.ieee.org</i>	Base Bibliográfica
ScienceDirect	<i>sciencedirect.com</i>	Base Bibliográfica
Scopus	<i>scopus.com</i>	Motor de Busca
SpringerLink	<i>link.springer.com</i>	Base Bibliográfica

Fonte: Adaptado de Nakagawa et al. (2017).

4.1.3 String de Busca

A elaboração da *string* de busca não é uma tarefa trivial. A identificação de uma combinação de termos que permitam encontrar o maior número de estudos primários relevantes de forma objetiva necessita, na maioria dos casos, de experiência e profundo conhecimento sobre a área de pesquisa abordada.

Tabela 3 – Termos e sinônimos utilizados.

Termos	Sinônimos
Domain Specific Language	DSL, Domain-Specific Language, Domain-Specific-Language, DSLM, Domain Specific Modeling Language, Domain-Specific Modeling Language, Domain-Specific-Modeling-Language, Query Language
Entity-Relationship	ER, Enhanced Entity-Relationship, EER, Database

Fonte: O autor.

Para a formação da *string* de busca é fundamental definir um conjunto de palavras referentes ao tema de pesquisa, bem como os sinônimos considerados expressivos. Estes termos devem representar de forma abrangente o tema central do estudo. Para este trabalho foram estabelecidos os termos e sinônimos da Tabela 3, sendo que sua combinação gerou a *string* genérica da Figura 17.

Figura 17 – String genérica de busca.

```
(DSL OR Domain Specific Language OR Domain-Specific Language OR
Domain-Specific-Language OR DSLM OR Domain Specific Modeling Language
OR Domain-Specific Modeling Language OR
Domain-Specific-Modeling-Language OR Query Language) AND (ER OR
Entity-Relationship OR Enhanced Entity-Relationship OR Extended
Entity-Relationship OR Database)
```

Fonte: O autor.

4.1.4 Critérios de Seleção

Segundo Kitchenham e Charters (2007), os Critérios de Inclusão (CIs) indicam por qual ou quais parâmetros um estudo é incluído no SLM, ou seja, considerado relevante. Da mesma forma, os Critérios de Exclusão (CEs) indicam por qual ou quais parâmetros um estudo é excluído, ou seja, considerado não relevante na pesquisa realizada. Para o este trabalho foram determinados os critérios de seleção listados a seguir.

- **Critérios de Inclusão (CI)**

- CI1: Estudo que propõe alguma técnica, método, abordagem ou ferramenta para a representação e transformação de modelos de BD utilizando DSLs.

- **Crítérios de Exclusão (CE)**

- CE1: Estudo com menos de 4 páginas;
- CE2: Estudo que não esteja escrito em inglês;
- CE3: Estudo duplicado;
- CE4: Estudo que não fornece acesso completo ao seu conteúdo;
- CE5: Estudo que não atende o CI1.

4.1.5 Avaliação de Qualidade

Segundo Nakagawa et al. (2017), embora exemplos de formulários de avaliação de qualidade possam ser encontrados com certa facilidade na literatura, para a elaboração de critérios e definição de seus pesos os pesquisadores envolvidos em cada revisão ou mapeamento devem levar em consideração as particularidades conforme o tema e as QPs. Portanto, em pesquisas desse tipo existe total liberdade para definição o número de CQ e seus pesos relacionados.

Foram definidos sete (7) CQs para a avaliação dos estudos primários aprovados após a aplicação dos critérios de seleção. Os CQs visam quantificar a relevância para que seja possível realizar uma comparação entre os estudos selecionados. Foi definido também uma pontuação para ser atribuída a partir das CQs. Para a definição das pontuações foram estabelecidas siglas para representar a pontuação dos CQs.

- **T: Total**, contemplando de forma integral o critério de qualidade avaliado;
- **P: Parcial**, dependendo do peso total do CQ), contemplando parcialmente o critério de qualidade avaliado;
- **N: Negativo**, não contempla de forma nenhuma o critério de qualidade avaliado.

A pontuação máxima possível, avaliados todos os critérios, é dez (10.0) e a mínima zero (0). Cada CQ possui um peso específico (1 → 1.5 → 2) dependendo da sua importância considerada para este estudo. Na Tabela 4 são listadas os CQs e seus respectivos pesos. Estes CQs basearam-se em aspectos que foram considerados relevantes para o SLM, sendo eles **relato** (QA1, QA4, QA5), **rigor** (QA2, QA3), **credibilidade** (QA2, QA3) e **relevância** (QA1, QA6, QA7).

Para todos os CQs a sigla **N** (Negativo) representa zero (0) e a sigla **T** (Total) representa o conceito máximo (1 → 1.5 → 2). Por outro lado é importante se salientar que apenas os CQs que possuem pontuação um (1) e dois (2) a sigla **P** (Parcial) representa 50%. Nos CQs com peso 1.5 o **P** representa 60% (0.9) da pontuação. Logo após essa definição detalhada dos CQs foi, então, possível dar início a fase de execução do SLM.

Tabela 4 – Critérios de Avaliação de Qualidade.

ID	Descrição	Peso
CQ1.	O estudo apresenta alguma contribuição para a área de modelagem de BD?	1,5
CQ2.	O estudo apresenta metodologias, técnicas ou propostas de refinamento baseado em modelos de dados?	1,5
CQ3.	O estudo apresenta alguma forma de avaliação empírica?	1,5
CQ4.	O estudo apresenta as características do processo de criação da DSL?	1,5
CQ5.	O estudo caracteriza as atividades de transformação do modelo para diferentes diferentes tecnologias de BDs?	2,0
CQ6.	O estudo apresenta pontos positivos e negativos observados em sua execução?	1,0
CQ7.	O estudo aponta desafios decorrentes da sua execução?	1,0

Fonte: O autor.

4.1.6 Estratégia de Extração de Dados

Definiu-se um formulário de dados para a extração e análise dos dados relevantes contidos nos estudos primários selecionados. A descrição detalhada deste formulário de extração é apresentado na Tabela 5.

Tabela 5 – Formulário de Extração de Dados.

Dado	Descrição
Origem da Solução	Organização ou Universidade dos autores do estudo
Ano de Publicação	Ano de publicação do estudo
Objetivo da Solução	Descrição da solução proposta
Ferramenta Citad	A ferramenta(s) citada(s) no estudo
Objetos de BD	Os objetos de BD adotados ou sugeridos pelas DSLs
Avaliação do Estudo	Se houve alguma avaliação, qual?
Ambiente de Avaliação	Academia ou indústria
Desafios	Quais são os desafios/trabalhos futuros apontados no estudo?
Pontos Positivos	Quais são os pontos positivos observados na implementação do estudo?
Pontos Negativos	Quais são os pontos positivos observados na execução do estudo?

Fonte: O autor.

4.1.7 Pesquisa na Literatura Cinza

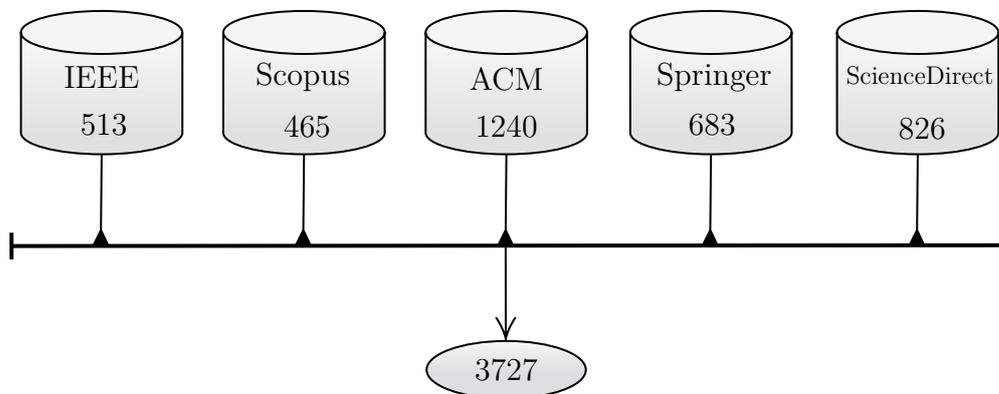
Para a pesquisa por ferramentas que dessem apoio a modelagem ER na literatura cinza foi utilizado o motor de busca da Google. Para isto, uma série de *keywords* foram definidas a partir da compreensão da *string* de busca genérica utilizada no mapeamento e na sua adequação para o contexto da busca na *Web*, sendo: *ERD Moldelling Tool*, *ERD Design Tool*, *Conceptual Design of Database*, *Conceptual Modelling of Database*, *Conceptual Modelling*, *Database Modelling Tool* e *Database Design Tool*.

A verificação dos resultados limitou-se a 10 (dez) páginas no motor de busca para cada *keyword*. As análises das ferramentas no MLM obedeceu a um processo de seleção com base em alguns requisitos. Primeiramente, a ferramenta deveria oferecer acesso para alguma forma de uso (incluindo versão de demonstração para ferramentas comerciais), necessitaria dar suporte a algum nível de modelagem de BD e precisaria ter *interface* em inglês ou português. Após, as ferramentas incluídas deveriam ter as notações e níveis de modelagem extraídos, juntamente com outras informações relevantes, e então categorizadas.

4.2 Execução do Mapeamento Multivocal

Para a execução do SLM foi necessário adaptar a sintaxe da *string* genérica para gerar outras versões, buscando assim adequá-la às peculiaridades de parametrização das diferentes bases utilizadas. Em seguida, foi realizada a busca dos estudos nas bases de dados. A Figura 18 mostra os resultados por biblioteca digital, bem como o total de estudos recuperados.

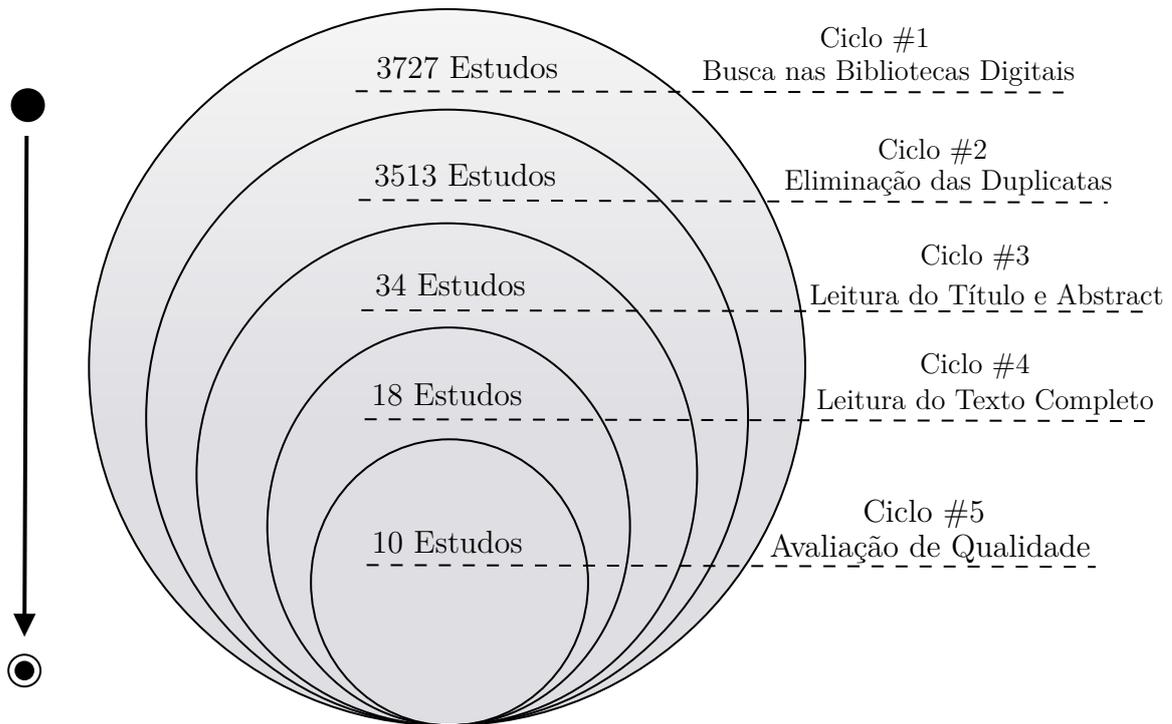
Figura 18 – Estudos primários por biblioteca digital.



Fonte: O autor.

Com o conjunto inicial de 3727 estudos primários identificados, foram definidos cinco (5) ciclos de seleção, apresentados na Figura 19. Nestas iterações houve a exclusão de estudos duplicados (restando 3513), a seleção de estudos baseado em título e *abstract* (restando 34), seleção baseada em texto completo (restando 18) e a seleção baseada na avaliação de qualidade (restando 10). Cada iteração tinha o objetivo de eliminar estudos que estavam fora do escopo da pesquisa ou considerados não relevantes. Na última iteração, 18 estudos tiveram sua qualidade analisada. Foi estabelecido que apenas estudos com pontuação acima de 5 (cinco) seriam aceitos. Assim, após a aplicação dos CQ foram excluídos 8 (oito) trabalhos. O conjunto final de dez (10) estudos aprovados na SLM seguiu para a etapa de extração de dados. Entre estes trabalhos a menor pontuação foi de

Figura 19 – Ciclos de seleção dos estudos primários.



Fonte: O autor.

5.3, enquanto o maior alcançou 10. A Tabela 6 resume os resultados obtidos na avaliação da qualidade.

A execução do protocolo de busca na *grey literature* retornou um total de 132 ferramentas. Após sumarização foi realizado a exclusão de duplicatas (restando 67). Durante esse processo houve doze (12) ferramentas que não puderam ser avaliadas *e.g.* impossibilidade de instalação ou incompatibilidade com o ambiente utilizado. Ao fim, foi possível executar testes de uso com um total de 55 ferramentas, e conseqüentemente, a extração dos dados relevantes ao estudo. Durante o uso das ferramentas foram realizados modelos simples de BDs, em que se procurou observar o suporte aos níveis de modelagem e as notações e linguagens usadas.

4.3 Resultados e Discussão

Os estudos primários foram avaliados com base nos critérios de qualidade definidos na subseção 4.1.5. A menor pontuação foi de 5,3 enquanto o maior alcançou 10. A Tabela 6 resume os resultados obtidos na avaliação da qualidade.

Em relação ao estado da arte (**QP1**) do desenvolvimento de DSL aplicado na transformação de modelos de dados, identificou-se um estudo que apresenta a *System Modeling Tool* (MIST), o qual utiliza uma DSL bidirecional para modelagem conceitual

Tabela 6 – Resultados da avaliação de qualidade.

Estudos	Critérios de Qualidade							Pontuação
Referência	CQ1	CQ2	CQ3	CQ4	CQ5	CQ6	CQ7	Total
Ayadi, Bouslimi e Akaichi (2016)	P	P	T	P	N	T	P	5,7
Celikovic et al. (2014)	T	T	N	T	P	P	T	7,0
Dimitrieski et al. (2015)	T	T	T	T	T	T	T	10,0
Hammer e Leod (1981)	T	T	N	T	N	P	P	5,5
Jagannathan et al. (1988)	T	T	N	T	P	N	N	5,5
Kersten et al. (2011)	P	T	T	P	N	N	P	5,3
Litwin et al. (1989)	P	T	N	T	N	P	T	5,4
Mazairac e Beetz (2013)	T	T	P	T	N	N	P	5,9
Shipman (1981)	T	T	N	T	P	N	P	6,0
Tian et al. (2006)	T	P	N	T	P	P	T	6,4

Fonte: O autor.

de BDs, chamado *EERDSL*.

Quanto às metodologias, técnicas ou propostas de refinamento (**QP1.1**) baseadas em modelos de dados, apenas os dois estudos mencionados aplicam conceitos para refinamento, sendo que tais conceitos são apoiados na normalização de BDs para auxiliar os desenvolvedores que utilizam sua solução.

No que se refere às tecnologias usadas como suporte para o desenvolvimento de DSLs (**QP1.2**), foram registradas Xtext, Xtextend, Sirius e Eugenia (CELIKOVIC et al., 2014; DIMITRIESKI et al., 2015), StarUML (AYADI; BOUSLIMI; AKAICHI, 2016), IfcDoc Tool e ViewEdit Tool (MAZAIRAC; BEETZ, 2013), MonetDB (KERSTEN et al., 2011), Java, JFlex e JCup (TIAN et al., 2006). No entanto, estudos mais antigos eram geralmente especificações de DSLs, não apresentando qualquer forma de implementação ou ferramenta usada (SHIPMAN, 1981; JAGANNATHAN et al., 1988; LITWIN et al., 1989).

As representações de BD adotadas (**QP1.3**) possuem **Tables** e **Functions** em todos os estudos primários analisados. Há também referências explícitas à definição de **Stored Procedures**, **Triggers** e **Views** em outros estudos. A Tabela 7 resume esses dados recuperados de cada um dos estudos primários. Quanto ao uso de DSL para transformação de modelos de dados em casos reais (**QP2**, **RQ2.1**), nenhuma referência foi encontrada nos estudos avaliados.

Sobre os métodos usados para avaliar as DSLs (**QP2.2**) existe apenas um estudo preliminar que apresenta a validação da proposta (DIMITRIESKI et al., 2015) usando dezesseis (16) participantes, sendo dois (2) especialistas em Interação Humano-Computador (IHC), três (3) especialistas em modelagem de sistemas e onze (11) estudantes. Entre os estudantes seis (6) eram mestrandos na área de BDs e cinco (5) doutorandos com experiência em modelagem. Em geral, os outros estudos indicam a falta de uma avaliação de suas proposições como um possível trabalho futuro.

Entre os aspectos positivos e negativos observados (**RQ2.3**) nos estudos, houve

Tabela 7 – Objetos de banco de dados representados.

Referência	Estudo Primário		Objetos de BD			
	DSL	Tables	SP	Functions	Triggers	Views
Ayadi, Bouslimi e Akaichi (2016)	Ayadi's Notation	✓				
Celikovic et al. (2014)	EERDSL v.1	✓		✓	✓	
Dimitrieski et al. (2015)	EERDSL v.2	✓	✓	✓	✓	✓
Hammer e Leod (1981)	SDM	✓			✓	
Jagannathan et al. (1988)	SDM	✓		✓		
Kersten et al. (2011)	SciSQL	✓		✓		
Litwin et al. (1989)	MSQL	✓	✓	✓	✓	✓
Mazairac e Beetz (2013)	BIMQL	✓	✓	✓		
Shipman (1981)	DAPLEX	✓		✓		
Tian et al. (2006)	NeuroQL	✓		✓		

Legenda: SP = *Stored Procedures*.

Fonte: O autor.

observações favoráveis à facilidade de compreensão, a modelagem intuitiva e independência de plataformas específicas (TIAN et al., 2006; MAZAIRAC; BEETZ, 2013). Pontos negativos foram a falta de geração automática de SQL para sistemas de BD (AYADI; BOUSLIMI; AKAICHI, 2016) ou uma limitação neste item (DIMITRIESKI et al., 2015). Ainda há um registro da falta de implementação real de DSLs até o momento em que o estudo foi realizado, havendo apenas especificações (HAMMER; LEOD, 1981; JAGANNATHAN et al., 1988; TIAN et al., 2006; KERSTEN et al., 2011; AYADI; BOUSLIMI; AKAICHI, 2016). Os principais desafios identificados pelos estudos (**RQ2.4**), em geral, são as avaliações das abordagens, bem como a evolução e/ou simplificação das propostas. Finalmente, na Tabela 8 as DSLs são apresentadas em relação ao seu tipo. No entanto, é importante observar que os estudos que marcam a coluna **bidirecional** (CELIKOVIC et al., 2014; DIMITRIESKI et al., 2015) são versões diferentes da mesma implementação de DSL, enquanto o (HAMMER; LEOD, 1981; JAGANNATHAN et al., 1988) são uma especificação de DSL e implementação com base nesta especificação, respectivamente.

Tabela 8 – Categorização das DSLs propostas.

Referência	Estudos Primários		Tipo de DSL		
	DSL	Textual	Gráfica	Bidirecional	
Ayadi, Bouslimi e Akaichi (2016)	Ayadi's Notation		✓		
Celikovic et al. (2014)	EERDSL v.1			✓	
Dimitrieski et al. (2015)	EERDSL v.2			✓	
Hammer e Leod (1981)	SDM	✓			
Jagannathan et al. (1988)	SDM	✓			
Kersten et al. (2011)	SciQL	✓			
Litwin et al. (1989)	MSQL	✓			
Mazairac e Beetz (2013)	BIMQL	✓			
Shipman (1981)	DAPLEX	✓			
Tian et al. (2006)	NeuroQL	✓			

Fonte: O autor.

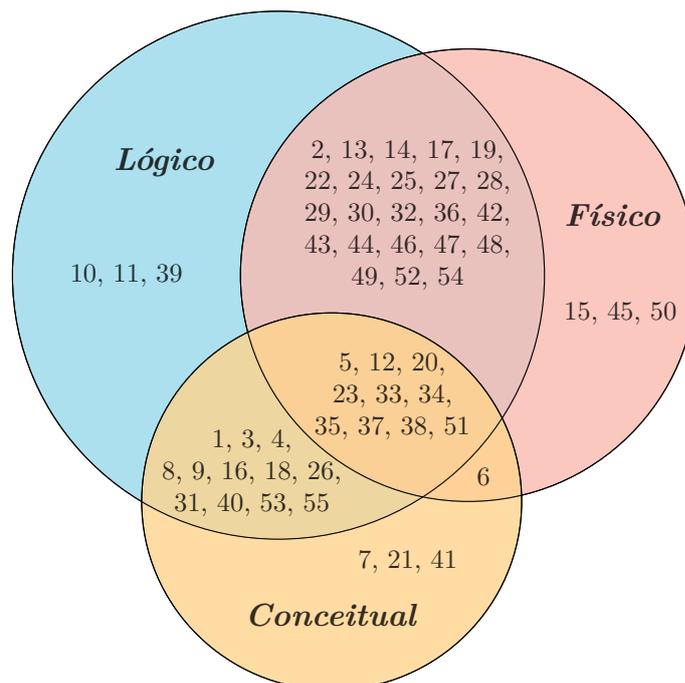
Quanto ao estado da prática (**QP3**) das ferramentas utilizadas na modelagem de BD, foram mapeadas 55 ferramentas. Houve a classificação quanto ao seu tipo, sendo

29 exclusivas de modelagem BD (*Data Modeling*), 13 de modelagem que ainda oferecem conexão com BD e execução de consultas (*Full IDE*), 10 com suporte a diagramação de diversos tipos de modelos (*Diagramming*) e 3 ferramentas projetadas para grandes empresas, podendo diagramar inúmeros tipos de documentos e processos (*Enterprise Modeling*).

Em relação às notações utilizadas nas ferramentas (**QP3.1**) foram identificados mais de 10 (dez) variedades de notações, com destaque a notação *Crow's Foot* com 35 ocorrências e da notação IDEF1X com 23 registros. Esses e outros dados estão listados nas Tabelas 9 e 10.

Finalmente, no que diz respeito aos modelos suportados pelas ferramentas (**QP3.2**) foi constatado que individualmente 26 ferramentas oferecem suporte a modelagem conceitual, 48 ferramentas à modelagem lógica e 37 à modelagem física. O conjunto representando as intersecções do suporte aos modelos quanto às ferramentas é exibido na Figura 20.

Figura 20 – Diagrama de *Venn* dos modelos suportados nas ferramentas.



Fonte: O autor.

Tabela 9 – Ferramentas para modelagem de bancos de dados.

#	Ferramenta	Tipo				Modelos			Notações Suportadas								Ambiente		Licença	
		DM	FIDE	DG	EM	C	L	F	CF	IDEF1x	CN	MN	UML	BN	AN	ON	D	W	C	G
1	AnalyseSI	✓				✓	✓			✓							✓			✓
2	Aqua Data Studio ER Modeler		✓				✓	✓	✓	✓							✓			✓
3	Astah			✓		✓	✓		✓	✓							✓			✓
4	brModelo			✓		✓	✓			✓	✓								✓	
5	Creately			✓		✓	✓		✓										✓	
6	Database Deployment Manager		✓			✓	✓	✓			✓						✓			✓
7	Database Workbench		✓			✓		✓	✓								✓			✓
8	DB Designer	✓				✓											✓			✓
9	DB-Main	✓				✓	✓						✓				✓			✓
10	DBDesigner 4	✓				✓	✓		✓	✓			✓				✓			✓
11	DBDesigner.net	✓					✓											✓		✓
12	dbdiagram.io	✓					✓											✓		✓
13	dbDiffo	✓				✓	✓	✓	✓						✓			✓		✓
14	dbForge Studio for MySQL		✓				✓	✓	✓	✓							✓			✓
15	DBSchema		✓				✓	✓	✓	✓				✓			✓			✓
16	DBVisualizer		✓					✓									✓			✓
17	DbWrench	✓				✓	✓		✓								✓			✓
18	DeZign for Databases	✓					✓	✓	✓	✓							✓			✓
19	Dia			✓		✓	✓	✓	✓	✓	✓		✓	✓	✓		✓			✓
20	dModelAid	✓					✓	✓	✓									✓		✓
21	Enterprise Architect				✓	✓	✓	✓	✓	✓							✓			✓
22	ER-Assistant	✓				✓			✓								✓			✓
23	ER/Builder	✓					✓	✓									✓			✓
24	ER/Studio Data Architect	✓				✓	✓	✓	✓	✓							✓			✓
25	ERD Concepts		✓				✓	✓	✓	✓							✓			✓
26	ERDesigner NG	✓					✓	✓	✓	✓							✓			✓
27	ERDPlus	✓				✓	✓		✓									✓		✓
28	Erwin Data Modeler	✓					✓	✓	✓	✓							✓			✓

Legenda: DM (*Data Modeling*) FIDE (*Full IDE*) DG (*Diagramming*) EM (*Enterprise Modeling*) | C (Conceitual) L (Lógico) F (Físico) | CF (*Crow's Foot*) CN (*Chen's Notation*) MN (*Merise Notation*) BN (*Barker's Notation*) AN (*Arrow Notation*) ON (*Other Notation*) | D (*Desktop*) W (Web) C (Comercial) G (*Gratuita*)

Fonte: O autor.

Tabela 10 – Ferramentas para modelagem de bancos de dados (continuação).

#	Ferramenta	Tipo				Modelos			Notações Suportadas							Ambiente		Licença	
		DM	FIDE	DG	EM	C	L	F	CF	IDEF1x	CN	MN	UML	BN	AN	ON	D	W	C
29	GenMyModel RDS			✓			✓	✓	✓									✓	
30	InfoSphere Data Architect				✓		✓	✓	✓	✓							✓		✓
31	Jeddict			✓			✓	✓	✓										✓
32	ModelRight	✓				✓	✓			✓							✓		✓
33	MySQL Workbench		✓				✓	✓	✓	✓				✓		✓			✓
34	Navicat Data Modeler	✓				✓	✓	✓	✓	✓							✓		✓
35	Navicat Data Modeler	✓				✓	✓	✓	✓	✓			✓				✓		✓
36	Open ModelSphere	✓				✓	✓	✓	✓		✓	✓					✓		✓
37	Oracle SQL Developer Data Modeler		✓				✓	✓	✓	✓							✓		✓
38	pgModeler	✓				✓	✓	✓									✓		✓
39	PowerDesigner				✓	✓	✓	✓	✓	✓	✓		✓				✓		✓
40	QuickDBD	✓				✓	✓										✓	✓	✓
41	RISE			✓		✓	✓		✓								✓		✓
42	Software Ideas Modeler			✓		✓			✓	✓	✓						✓		✓
43	SQL Database Modeler	✓					✓	✓		✓							✓	✓	✓
44	SQL Maestro		✓				✓	✓		✓							✓		✓
45	SQL Power Architect	✓					✓	✓	✓								✓		✓
46	SQL Server Management Studio		✓					✓									✓		✓
47	SQLDBM	✓					✓	✓		✓							✓		✓
48	SQLyog		✓				✓	✓									✓		✓
49	Toad Data Modeler	✓					✓	✓	✓	✓	✓						✓		✓
50	Valentina Studio		✓				✓	✓	✓	✓							✓		✓
51	Vertabelo	✓						✓	✓	✓							✓		✓
52	Visual Paradigm			✓		✓	✓	✓	✓								✓		✓
53	Win A&D			✓			✓	✓	✓	✓							✓		✓
54	WWW SQL Designer	✓				✓	✓	✓									✓	✓	✓
55	xCase	✓					✓	✓	✓								✓		✓

Legenda: DM (*Data Modeling*) FIDE (*Full IDE*) DG (*Diagramming*) EM (*Enterprise Modeling*) | C (Conceitual) L (Lógico) F (Físico) | CF (*Crow's Foot*) CN (*Chen's Notation*) MN (*Merise Notation*) BN (*Barker's Notation*) AN (*Arrow Notation*) ON (*Other Notation*) | D (*Desktop*) W (Web) C (Comercial) G (*Gratuita*)

Fonte: O autor.

4.4 Ameaças à Validade

Ameaças ao resultado do estudo foram identificadas no MLM realizado, e então categorizadas nos seguintes tipos: validade de construto, validade interna, validade externa e validade de conclusão (COOK; CAMPBELL, 1979; WOHLIN et al., 2012).

Validade do Construto: Aborda a possibilidade de que as QPs ou os termos de pesquisa que estruturam a *string* de busca sejam inadequados ou incompletos. Para mitigar essas ameaças, pesquisadores da área de DSL e modelagem de dados foram consultados. Além disso, foi realizada uma pesquisa piloto para avaliar a consistência de nossa *string* de pesquisa. Outra ameaça é a qualidade do material publicado que foi coletado na literatura cinza.

Validade Interna: Algumas possíveis ameaças são o uso de métodos incorretos de busca, o que pode levar a exclusão de estudos relevantes, uma aplicação de estratégia de extração de dados precária, a ocorrência de vieses na seleção ou no conteúdo dos estudos primários. Na tentativa de mitigar esses riscos, um protocolo foi definido com base em modelos de referência já bem estabelecidos na literatura.

Validade Externa: Ameaças externas geralmente abordam se as descobertas de um estudo podem ser generalizadas para outro domínio. Uma razão para esta ameaça seria a ocorrência da seleção de estudos de primários contendo informações incompletas. Contudo é provável que, por se tratar de uma área de intersecção entre modelagem de dados e DSLs, os resultados não podem ser generalizados para outros tópicos de pesquisa, reduzindo assim esse risco naturalmente.

Validade da Conclusão: Uma possível ameaça é o viés na extração de dados, o que leva a erros de conclusão. Para atenuar esse problema foi realizado uma leitura criteriosa e, assim como os testes de uso das ferramentas, houve a síntese de dados em planilha eletrônica¹ para uma melhor análise.

4.5 Trabalhos Relacionados

Essa seção descreve os trabalhos de maior representatividade para o objeto deste estudo. Uma vez que a proposta envolve a construção de uma ferramenta que implemente uma DSL textual, e após a pesquisa descrita neste capítulo, selecionou-se propostas e ferramentas que mais se aproximam do objetivo final deste trabalho.

O trabalho de Dimitrieski et al. (2015), desenvolvido na universidade de Novi Sad na Sérvia, apresenta uma ferramenta chamada *System Modeling Tool* (MIST). Essa ferramenta utiliza uma DSL chamada EERDSL, uma linguagem com base no modelo aprimorado de entidade-relacionamento, do inglês *Enhanced EntityRelationship* (EER). O EER inclui todos os conceitos introduzidos pelo modelo ER original proposto por Chen, sendo assim uma extensão do mesmo. Além disso, inclui os conceitos de subclasses e su-

¹ <http://bit.ly/2Vs0oYN>

perclasses (*Is-a*), juntamente com os conceitos de especialização e generalização. A MIST apresenta uma abordagem de modelagem bidirecional (gráfica e textual) de modelagem de BDs. O autor discute que tal decisão tem como motivo o entendimento de que a preferência sobre a abordagem de modelagem utilizada pode depender do domínio do problema, do conhecimento e das preferências pessoais de um projetista de BD. Apresenta também uma experiência anterior, onde foi construída uma ferramenta de modelagem com uma abordagem baseada em formulários. A partir dos resultados obtidos nesta experiência, foi concebida a ideia da MIST. O propósito da ferramenta é a aplicação tanto no mercado profissional quanto para o ensino de projeto e modelagem de BD no meio acadêmico. A MIST foi desenvolvida com o auxílio do *framework* Xtext para a notação da DSL textual e inicialmente utilizava o *framework* Eugene, um projeto que foi descontinuado, para a sua versão gráfica. Posteriormente em razão disso o Eugene foi substituído pelo *framework* Sirius. A MIST ainda oferece suporte à geração de código SQL.

O *dbdiagram.io*² é uma ferramenta *Web* gratuita para o desenho de DERs, desenvolvida por uma empresa de Singapura, com uma abordagem textual que implementa uma DSL própria. Esta DSL utiliza um modelo muito próximo do lógico. O diferencial da ferramenta é sua rápida curva de aprendizagem e, além disso, a apresentação de uma representação gráfica do que está sendo modelado. A apresentação dos elementos do diagrama pode ser organizada livremente pelo usuário em tempo real. Entretanto é importante se salientar que toda a modelagem de fato é feita de modo textual. A ferramenta ainda oferece a geração automática de código SQL.

Da mesma forma, a *QuickDBD*³, desenvolvida por uma empresa na Irlanda, é uma ferramenta *Web* com exatamente o mesmo modo operacional que a *dbdiagram.io*, também implementando uma DSL textual própria para modelagem de BDs. Contudo é uma ferramenta proprietária, ou seja, paga e com o foco declaradamente na indústria. Ambas as ferramentas são muito similares também quanto a geração de representações gráficas da modelagem e apresentam diversos argumentos para sua adoção, como a rápida compreensão de suas DSLs, a perspectiva de realização de trabalhos fluídos, o acesso de qualquer plataforma e o compartilhamento dos modelos com outros usuários.

Finalmente, pode-se citar a ferramenta *Web* gratuita *RelaX (Relational Algebra Calculator)*⁴. Esta ferramenta não foi encontrada no mapeamento, mas indicada por um pesquisador da área de DSLs e BDs. Trata-se de uma ferramenta desenvolvida na universidade de Innsbruck, na Áustria, e voltada ao ensino de álgebra relacional fazendo operações sobre bases de dados relacionais. Tem uma abordagem textual, utilizando uma DSL chamada RelAlg, e apresentando inclusive duas perspectivas de operação: instruções de RelAlg e instruções em SQL. A RelaX utiliza uma abordagem de modelo já em nível físico para operações, como as DDLs de construção e DMLs para as consultas. Apesar de

² <https://dbdiagram.io/>

³ <https://quickdatabasediagrams.com/>

⁴ <https://dbis-uibk.github.io/relax/>

suas funcionalidades, a Relax não se propõe a ser uma ferramenta de projeto e modelagem de BD, mas de uso restrito ao ensino dentro da academia.

4.6 Lições do Capítulo

Todos os anos, várias contribuições para a modelagem ER são publicadas. A modelagem de BD é uma área essencial na ES e as DSLs que suportam essa atividade não são encontradas trivialmente na literatura. A fim de acompanhar a evolução e as tendências de vários sistemas de BDs, uma pesquisa de alternativas para o *design* é essencial. Neste capítulo é fornecida uma visão geral sobre as DSLs usadas pela modelagem de ER por meio de um MLM.

Este mapeamento abrangeu 3727 artigos com a intenção de investigar estudos primários que fizeram propostas de DSLs para modelagem de BD. Da mesma forma, foi levantado um conjunto de 132 ferramentas que dão suporte a modelagem de BDs, procurando mapear as notações e modelos suportados pelas mesmas. O protocolo do MLM foi detalhado, assim como sua condução e subsequente análise dos resultados obtidos. Ao final, 10 estudos primários e 55 ferramentas foram selecionadas para serem analisadas de forma quantitativa e qualitativa. Como resultado, classificou-se apenas as DSLs atualmente usadas para dar suporte à modelagem de ER e ferramentas que são utilizadas para projetar BDs. Entre os resultados destaca-se, entre os trabalhos relacionados, o estudo de Dimitrieski et al. (2015), o qual apresenta uma ferramenta de modelagem bidirecional que aplica sua própria DSL com base na abordagem EER.

Este capítulo forneceu algumas evidências de que, a cada ano, um número significativo de trabalhos apresentando diferentes tipos de notações é publicado. Isso é de certa forma surpreendente, devido ao fato de que as notações de relacionamento entre entidade usadas hoje pela indústria e pela academia, como as de Chen (1976) e Barker (1990), não são propostas recentes. Portanto, conclui-se que a modelagem de ER continua um amplo campo de pesquisa com algumas lacunas a serem exploradas *e.g.*: No que diz respeito a abordagens gráficas, qual a notação de mais fácil aprendizado? A abordagem textual pode ser uma alternativa viável para a modelagem de BDs em diferentes níveis? Qual o grau de aderência das abordagens entre os usuários conforme o seu perfil (analista ou desenvolvedor)?

5 DSL PARA A REPRESENTAÇÃO DE MODELOS CONCEITUAIS DE BANCO DE DADOS RELACIONAIS

Este capítulo apresenta a proposta central deste trabalho. A Seção 5.1 aponta os requisitos levantados para a construção da DSL. A Seção 5.2 descreve as decisões de projeto referentes aos requisitos. A arquitetura da implementação da proposta é detalhada na Seção 5.3. A demonstração do protótipo construído ocorre na Seção 5.4. Em seguida a avaliação preliminar conduzida com um grupo focal é apresentado na Seção 5.5, bem como os resultados obtidos. A Seção 5.6 detalha a ferramenta ERText, produto do desenvolvimento da DSL e, por fim, as lições do capítulo são pontuadas na Seção 5.7.

5.1 Requisitos da Linguagem

Esta seção lista os requisitos que foram definidos com base na literatura utilizada neste trabalho, bem como no conhecimento prévio dos pesquisadores envolvidos na condução do estudo. Estes requisitos são relacionados diretamente com as decisões de projeto.

- ***RQ1. A DSL precisa ser disponibilizada sob uma licença open source.*** Como o foco da proposta é no processo de ensino é fundamental que a linguagem seja de código aberto. A vantagem que este requisito proporciona é a posterior evolução e manutenção colaborativa com o envolvimento de outros desenvolvedores.
- ***RQ2. A DSL deve permitir representar textualmente modelos conceituais de BDs.*** Como é um objetivo que a solução seja outra opção em relação às abordagens gráficas, esse requisito se justifica. Isso permite o foco na compreensão do domínio e no desenvolvimento da DSL.
- ***RQ3. Os modelos conceituais devem dar suporte à definição de entidades, atributos, relações e cardinalidades.*** As ferramentas utilizadas para o desenvolvimento da linguagem precisam permitir que sejam implementados os conceitos de domínio que regem a estrutura de DER tradicional.
- ***RQ4. Os modelos conceituais devem dar suporte a definição de atributos identificadores, generalização/especialização, auto-relacionamentos e relacionamentos ternários.*** A linguagem deve permitir que conceitos mais sofisticados dos domínios sejam definidos.
- ***RQ5. A implementação da DSL deve realizar a transformação do modelo conceitual para o lógico.*** A solução deve realizar a transformação do conceitual para o lógico, exibindo o resultado gerado ao usuário.

5.2 Decisões de Projeto

Nesta seção são descritas as decisões de projeto para criar a DSL textual que suporte todos os requisitos discutidos na Seção 5.1. Para cada decisão de projeto são indicados os seus requisitos associados.

- **DP1. A solução deve adotar um LW open source no auxílio da implementação da DSL textual (RQ1, RQ2).** Mediante a investigação conduzida durante este estudo foi selecionado o LW Xtext para o desenvolvimento da proposta por ser um *framework open source* focado no desenvolvimento de DSLs textuais, fornecendo toda a infraestrutura necessária. Além disto, o Xtext é uma ferramenta com alto nível de maturidade, documentação detalhada e uma comunidade ativa.
- **DP2. A DSL deve fornecer uma representação textual que seja equivalente ao modelo ER gráfico usualmente utilizado (RQ3, RQ4).** Para os requisitos cobertos por esta decisão de projeto foi adotada a estratégia de se realizar uma análise nas ferramentas averiguadas no mapeamento descrito no Capítulo 4, bem como no livro referência de Heuser (2009).
- **DP3. A solução deve realizar a transformação entre os modelos (RQ5).** O Xtext usa modelos do *Eclipse Modeling Framework* (EMF) como a representação na memória de qualquer arquivo de texto analisado. Esse grafo de objetos na memória é chamado de árvore sintática abstrata, do inglês *Abstract Syntax Tree* (AST). Esses conceitos também são chamados de gráficos de objeto de documento, do inglês *Document Object Graph* (DOM), modelo semântico ou simplesmente modelo. Desta forma, existe a representação do modelo da gramática na forma de um metamodelo central no núcleo do EMF, chamado de modelo *Ecore*. Tendo o *Ecore* da DSL proposta como uma representação, é possível então aplicar regras de transformação, gerando assim outros modelos.

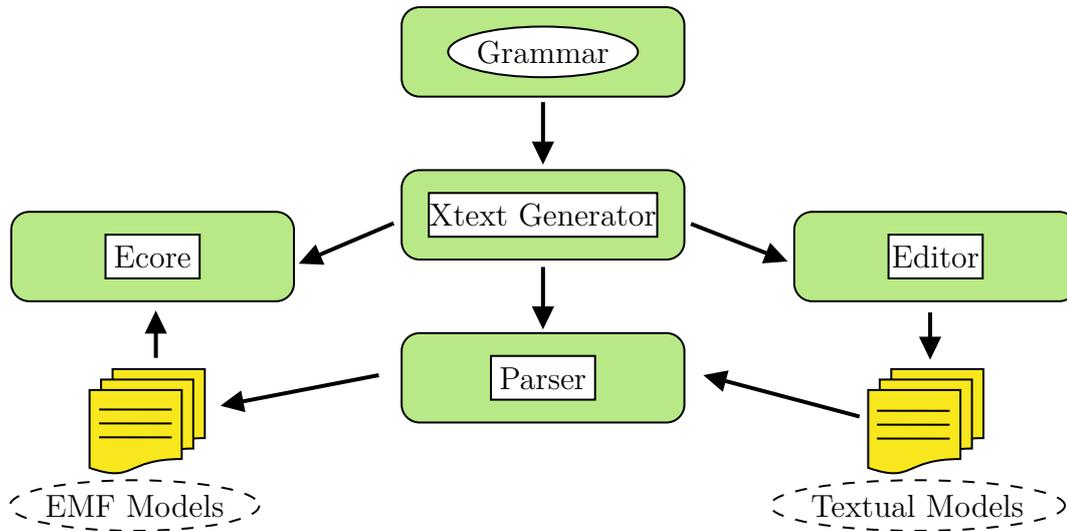
5.3 Arquitetura

O *framework* Xtext gera toda a infraestrutura para a criação de linguagens com base fundamentalmente nas gramáticas definidas. A Figura 21 fornece uma visão geral em um nível abstrato da arquitetura do Xtext.

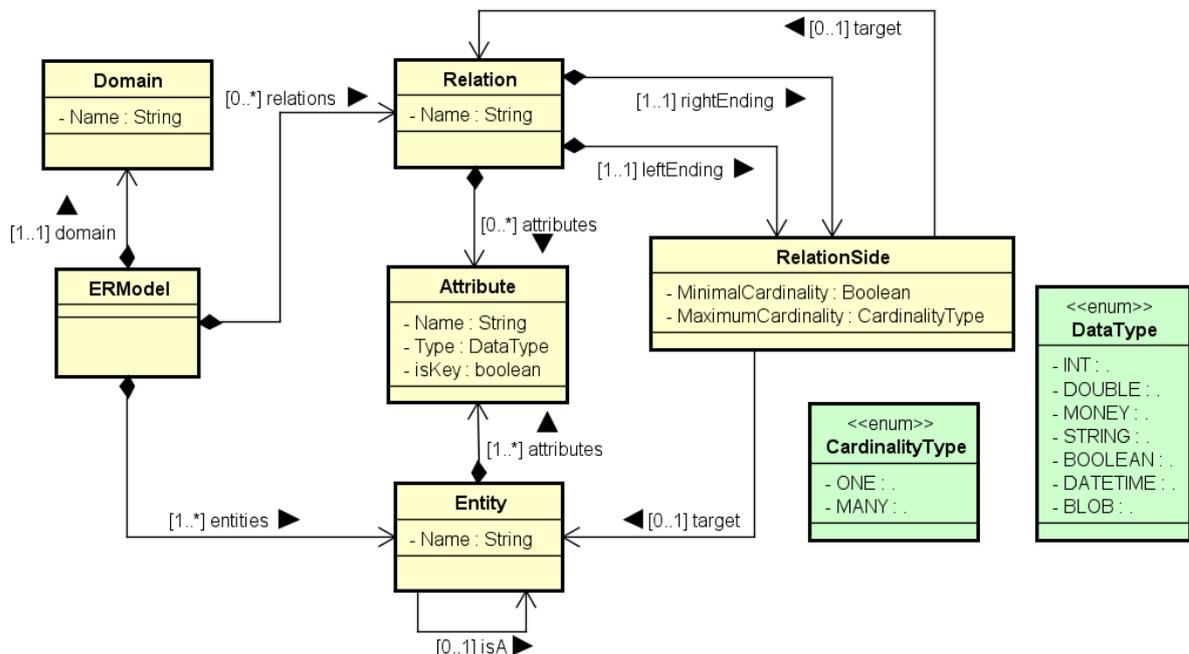
Foram implementadas duas versões da gramática no protótipo da DSL. A seguir as arquiteturas dessas implementações são descritas e pontua-se as diferenças entre elas. Essa abordagem foi definida tendo em vista que será realizada uma avaliação preliminar junto a um grupo focal. A partir dos resultados dessa avaliação, objetiva-se gerar uma versão final da gramática e, conseqüentemente, da arquitetura.

A Figura 22 mostra um diagrama de classes para representar o modelo *Ecore* da primeira versão criada para esta proposta. O elemento central do modelo é a classe

Figura 21 – Arquitetura geral do Xtext.



Fonte: Adaptado de Obeo e TypeFox (2017).

Figura 22 – Representação do modelo *Ecore* da 1ª versão da DSL.

Fonte: O autor.

`ERModel`, a qual corresponde por uma composição com outros elementos. O `ERModel` deve possuir um `Domain` associado, simbolizando o nome da base de dados modelada.

Um `ERModel` também deve conter um ou mais elementos `Entity`. Um elemento `Entity` pode se relacionar com outro elemento `Entity`, cobrindo assim o conceito de generalização/especialização. Um `Entity` é também uma composição de uma ou mais

classes `Attribute`. Definiu-se os tipos de dados em uma lista enumerada no elemento `DataType`.

Em seguida, um `ERModel` pode ser composto de uma ou mais relações, retratado como a classe `Relation`. Uma `Relation` por sua vez é formada por duas classes `RelationSide`, os quais são as cardinalidades da esquerda e da direita em uma relação. Estas duas relações possuem uma referência chamada `Target`, que pode ser uma `Entity` ou uma `Relation`. A inclusão da possibilidade de se referenciar uma `Relation` se faz necessário para cobrir a modelagem dos relacionamentos ternários.

Finalmente, tem-se o conceito das cardinalidades. Nele as cardinalidades possíveis, mantidas em atributos da `RelationSide`, são os tipos enumerados em `CardinalityType`, sendo `One` para um e `Many` para muitos. Por meio das regras da gramática garante-se que a cardinalidade mínima é implícita, tendo a palavra reservada `zero` para simbolizar quando uma cardinalidade mínima pode ser nula. Isto significa que, em uma modelagem onde uma cardinalidade mínima `zero` é omitida, assume-se que ela automaticamente é igual a um.

Estruturalmente, o modelo *Ecore* da segunda versão DSL não muda significativamente. A única diferença com maior impacto é a opção pela definição da cardinalidade utilizando-se as quatro combinações possíveis como termos reservados, armazenados diretamente no atributo `Cardinality` de `RelationSide`.

5.4 Protótipo

Na fase atual do trabalho a linguagem a nível conceitual não se encontra totalmente finalizada. Existem tópicos relativos a validação de escopo, como no caso do tratamento de referências cruzadas indesejadas e outras restrições inerentes ao modelo ER que devem ser analisadas e então implementadas. A definição da DSL criada é exibida na Figura 23.

O comando `grammar` especifica o nome da DSL, enquanto a instrução `with` declara uma herança de outra linguagem. No caso da gramática proposta, será utilizada uma gramática padrão do Xtext, chamada `Terminals`, que fornece algumas regras predefinidas como, por exemplo, a regra `ID` para identificadores ou `INT` para inteiros. O comando `generate` é a instrução que produz a AST da linguagem.

A primeira regra, chamada de regra de entrada, define como é a estrutura geral da linguagem. Palavras e símbolos entre aspas duplas ou simples indicam as palavras reservadas. Por exemplo, o objeto `Entities` é obrigatoriamente precedido de `"Entities{"`. Este objeto representa uma espécie de *container*, sendo isto indicado por meio do operador de atribuição `+=`. Ele é um objeto que pode conter outros objetos, no caso um ou mais `Entity` (entidades). É estabelecido que cada arquivo da DSL deve também ser composto de um `Domain` (domínio) e zero ou mais `Relation` (relações).

Para melhor entendimento, deve-se deixar claro que a multiplicidade é indicada por `*` (zero ou muitos), `+` (um ou muitos) ou `?` (zero ou um). Ao não se colocar nenhum

Figura 23 – Implementação da 1ª versão da DSL.

```

grammar org.xtext.unipampa.lesse.erdsl with org.eclipse.xtext.common.Terminals

generate erdsl "http://www.xtext.org/unipampa/lesse/erdsl/"

ERModel:
    domain=Domain ";"
    ("Entities{"} entities+=Entity+ ("};")
    ("Relationships{"} relations+=Relation* ("};");

Domain:
    "Domain" name=ID;

Entity:
    name=ID ("isA" isA+=[Entity])*
    ("{" attributes+=Attribute ("," attributes+=Attribute)* "}");

Attribute:
    name=ID ":" type=DataType (isKey?="isIdentifier")?;

Relation:
    (name=ID)?
    ("[" leftEnding=RelationSide
    "isRelatedWith"
    rightEnding=RelationSide "]" )
    ("{" attributes+=Attribute
    ("," attributes+=Attribute)* "}");

RelationSide:
    ((minimalCardinality?="zero")?) maximumCardinality=CardinalityType
    target=[Entity | target=[Relation];

enum DataType:
    INT="int" | DOUBLE="double" | MONEY="money" | STRING="string" |
    BOOLEAN="boolean" | DATETIME="datetime" | BLOB="file";

enum CardinalityType:
    One="one" | Many="many";

```

Fonte: O autor.

desses operadores, implicitamente espera-se então apenas uma ocorrência. Em relação às atribuições, quando apenas um = for especificado significa que o objeto da esquerda espera apenas um registro. Logo, para += espera-se então zero, uma ou mais ocorrências.

O objeto **Domain** é precedido de uma palavra reservada com o mesmo nome, seguido de um identificador. A entidade é definida pela palavra **Entity** e um nome identificador específico para este objeto. A definição de uma herança é opcional por meio da palavra reservada **isA**. Após a definição do nome, abre-se um corpo de chaves em que são especificados os atributos da entidade. Uma entidade deve conter ao menos um atributo, mas ele não precisa ser identificador por conta da possível existência de entidades fracas.

As regras compostas só são realizadas devido à possibilidade de se agrupar expressões com o uso de parênteses, além da possibilidade de se utilizar outras regras por meio de referências cruzadas. Os colchetes entre a regra **Entity** servem para indicar que se almeja usar apenas o atributo **name** que identifica o objeto. Os atributos das entidades são definidos por um nome, herdando a regra **ID** de **Terminals**, e atributos **isIdentifier**

opcionais para simbolizar chaves primárias.

A relação é definida, já dentro do corpo do bloco `Relationships`, com uma declaração opcional de sua identificação. Em seguida, são abertos colchetes e deve-se especificar dois elementos `RelationSide` como referência aos atributos `leftEnding` e `rightEnding`. Estes atributos representam os lados de uma relação. Estes objetos devem ser separados pela expressão `isRelatedWith`. Os lados da relação são definidos na regra `RelationSide`, composta de dois atributos. O atributo `minimalCardinality` é opcional, indicado pelo operador `?`, e aceita apenas a palavra reservada `zero`. O atributo `maximumCardinality` aceita um objeto `CardinalityType`.

Os tipos de atributo estão contidos em uma lista enumerada chamada `DataType`, na esquerda fica a representação no modelo *Ecore*. Na direita está a palavra reservada que é usada na linguagem. O símbolo condicional `|` significa o operador lógico *OR* (ou) e serve para separar cada definição `<chave> = <valor>` como uma opção dentro da lista. Das três cardinalidades possíveis, duas estão definidas em outra lista enumerada chamada `CardinalityType`. Na Figura 24 exibe a principal mudança entre as versões da DSL proposta, a qual diz respeito a cardinalidade explícita.

Figura 24 – Fragmento de implementação da 2ª versão da DSL.

```

Relation:
  (name=ID)? ("[" leftEnding=RelationSide "relates"
  rightEnding=RelationSide "]" ) ("{" attributes+=Attribute
  ("," attributes+=Attribute)* "}")*);

RelationSide:
  Cardinality=("0,1" | "1,1" | "0,N" | "1,N")
  target=[Entity] | target=[Relation];

```

Fonte: O autor

Na Figura 25 mostra-se um exemplo de uso com um pequeno modelo que aborda uma universidade como domínio. Neste protótipo pode-se observar a modelagem de generalização/especialização, auto-relacionamento e relacionamento ternário.

É importante destacar que neste exemplo são modeladas seis entidades e quatro relacionamentos. Não obstante, no processo de transformação para o modelo lógico espera-se que seja gerado um esquema textual com mais três entidades, inferindo-se isso por meio de, por exemplo, relacionamentos *muitos para muitos*. Isso ocorreria, nesta amostra, no relacionamento sem identificação, atribuindo automaticamente a nova entidade o nome resultante da concatenação das duas entidades que se relacionam no conceitual. Também seriam criadas novas entidades a partir da derivação dos relacionamentos nomeados `TeacherClass` e `ClassSchedule`, sendo que o último caracteriza um relacionamento ternário. Por fim, o auto-relacionamento `Supervisor` de `OutSrcEmployee` implicaria na adição de um novo atributo na entidade no novo modelo.

Com base nos modelos *Ecore* gerados em tempo real, começou-se a implementar

Figura 25 – Exemplo de uso da 1ª versão da DSL no RCP do Eclipse.

```

Domain University;

Entities{
    Person{
        PID: int isIdentifier ,
        Name: string
    }
    Teacher isA Person{
        Phone: int ,
        Salary: money
    }
    Student isA Person{
        Course: string
    }
    OutSrcEmployee isA Person{
        Company: string
    }
    Class{
        ClassID: int isIdentifier ,
        Course: string ,
        Semester: string
    }
    Classroom {
        ClassroomID: int isIdentifier ,
        Capacity: int
    }
};

Relationships{
    [many Student isRelatedWith many Class]
    TeacherClass [many Teacher isRelatedWith many Class]
    ClassSchedule [many TeacherClass isRelatedWith many Classroom]
        {CSID: int isIdentifier , DayOfWeek: datetime , Discipline: string}
    Supervisor [one OutSrcEmployee isRelatedWith many OutSrcEmployee]
};

```

Fonte: O autor.

a transformação preliminar do modelo conceitual para o lógico. A implementação desta transformação foi realizada utilizando-se a Xtend, uma GPL baseada em Java.

5.5 Avaliação Preliminar da Gramática

Esta seção descreve a avaliação preliminar conduzida para analisar as duas alternativas de gramática propostas, visando assim o seu aperfeiçoamento em uma versão final na solução.

Para tanto, foi estabelecido a utilização de um grupo focal, um método de pesquisa qualitativa que objetiva gerar *feedback* de um conjunto de pessoas em relação a um tema específico. Essa abordagem é muito utilizada como uma atividade para pesquisa de mercado em diversas áreas, uma vez que pode cumprir papel importante apoiando pesquisas exploratórias.

O processo executado nesta etapa, expressado na Figura 26, teve como base as diretrizes estabelecidas no trabalho de Kontio, Bragge e Lehtola (2008), as quais cobrem a aplicação desse método no contexto da ES.

Figura 26 – Processo do Grupo Focal.



Fonte: O autor.

5.5.1 Planejamento

Durante o planejamento foi definido um protocolo que deveria ser seguido. Neste protocolo, motivado pelo problema que era gerar uma versão definitiva da gramática da DSL proposta, foram criados os documentos necessários para sua execução: **(i)** Termo de Consentimento Livre e Esclarecido (TCLE); **(ii)** Glossário de Termos; **(iii)** Questionário de Perfil; **(iv)** Instrumentos da Discussão 1, 2 e 3; **(v)** Modelos das Gramáticas Avaliadas; **(vi)** Roteiro de Apresentação. Todos os modelos dos documentos produzidos estão incluídos no Apêndice B.

5.5.1.1 Preparação

Tipicamente as avaliações que utilizam grupos focais devem ser constituídas de quatro (4) a seis (6) grupos focais individuais para que o rigor científico seja considerado verdadeiramente alto. O tamanho de cada grupo focal pode variar de três (3) a até doze (12) elementos, sendo mais comum esse número ficar entre quatro (4) e oito (8) participantes (KONTIO; BRAGGE; LEHTOLA, 2008).

Por questões de viabilidade de tempo e recursos humanos, para o presente estudo foi possível ser executado um (1) grupo focal. Após a realização do convite houve a colaboração de treze (13) participantes, todos da área da ES. Desse total, três (3) participantes eram alunos de graduação, nove (9) eram mestrandos e um (1) doutorando.

Foi então aplicado o Questionário de Perfil, em que foi possível identificar que havia um nível equilibrado de conhecimento entre os participantes. Isso foi constatado pois todos já possuíam contato com DSLs, tendo utilizado esse tipo de linguagem, ao menos, durante a graduação. Ainda foi informado que todo o processo seria gravado em áudio, fato com o qual todos concordaram.

5.5.2 Execução

O grupo focal foi realizado em 08/2019, nas dependências da UNIPAMPA, e teve duração de duas (2) horas. Iniciando com a exposição do roteiro que deveria ser seguido,

onde houve a apresentação do objetivo do grupo focal e os conceitos básicos envolvidos, aconteceu o pedido para que os participantes realizassem a leitura e assinatura do TCLE. Com o documento preenchido, foi dada continuidade ao roteiro previsto,

A cada Instrumento de Discussão disponibilizado esperou-se até que os participantes respondessem de forma individual. Após, aconteceu uma discussão em grupo sobre o tema levantado. Todo o processo foi gravado em áudio e teve o suporte dos três (3) pesquisadores envolvidos neste estudo. Foi realizada também a transcrição das observações levantadas pelo debate que se seguiu, caracterizando assim as práticas de *brainstorming*¹ previstas em grupos focais.

5.5.3 Análise dos Resultados

Segundo Kontio, Bragge e Lehtola (2008), a fase de análise e interpretação dos dados gerados constitui parte importante da pesquisa qualitativa, considerando o contexto, o comportamento e a percepção dos sujeitos. Para a fase de análise dos dados, o áudio foi analisado paralelamente as anotações realizadas. De posse destes materiais e das respostas dos participantes para cada instrumento de discussão, foi possível avaliar os resultados do grupo focal executado.

5.5.3.1 Discussões

Após a apresentação do roteiro preparado para o grupo focal, deu-se início as discussões dos três (3) instrumentos criados para a dinâmica. O primeiro instrumento continha a seguinte afirmação associada a uma escala Likert composta de níveis de concordância dispostos de um (1) a cinco (5):

“Linguagens específicas de domínio com abordagem textual podem ser aplicadas na modelagem conceitual posto que conseguem descrever de forma rápida e concisa determinadas propriedades. Sendo assim, essas soluções podem ser utilizadas ou mesmo adaptadas de uma forma efetiva no que diz respeito a representação do domínio que modelam.”

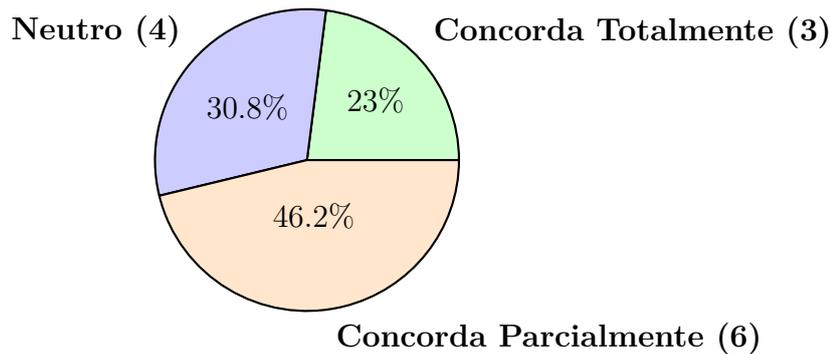
Após todos os participantes responderem o instrumento, foi aberto um momento de discussão entre todos. Por não terem visto o modelo proposto de DSL, algumas dúvidas surgiram e os pesquisadores envolvidos procuraram sanar todas de forma a não influenciar as discussões seguintes.

O debate prosseguiu com os participantes levantando possíveis vantagens de um modelo textual. Alguns citaram acreditar que essa abordagem poderia ser de mais fácil entendimento, porém que isso dependeria do usuário. Esta suposição incluiu dois prováveis tipos de perfis: analistas e desenvolvedores.

¹ *Brainstorming* é uma técnica utilizada para propor soluções a um problema específico. Consiste em uma reunião, também chamada de tempestade de ideias, na qual os participantes devem ter liberdade de expor suas sugestões e debater sobre as contribuições dos colegas.

O grupo chegou a conclusão que a abordagem poderia ser vista como mais produtiva por usuários de perfil desenvolvedor, mas menos proveitosa por aqueles que tivessem um viés mais analista em razão do seu nível de abstração em relação às abordagens gráficas. A Figura 27 exhibe a distribuição das respostas dos participantes para a primeira discussão.

Figura 27 – Resultados da Discussão 1 - Grupo Focal.



Fonte: O autor.

Após, passou-se para a execução da discussão do segundo instrumento. O atividade era composta da seguinte pergunta:

“Considerando que um modelo conceitual de banco de dados deve definir ao menos as entidades de domínio, seus atributos e o número de ocorrências (cardinalidade) possíveis de associações (relacionamento), como você definiria uma gramática básica (DSL) para a sua representação?”

Foi informado que os participantes poderiam conversar livremente durante toda a realização deste exercício. Após cada um sugerir a sua sintaxe, houve debate e troca de informações sobre como estruturar melhor as informações. A discussão foi focada principalmente em como representar as relações do modelo ER em uma sintaxe textual.

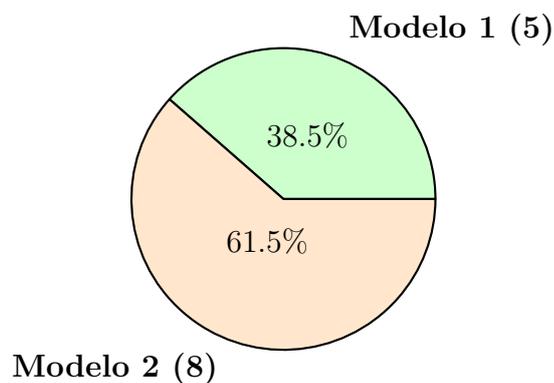
A maior dificuldade se mostrou em como definir uma ordem. Outro ponto que merece destaque foi quanto à cardinalidade, onde no geral seguiu-se a nomenclatura utilizada no diagrama original de Chen (*e.g.* 0,1). Entretanto, ao fim do instrumento houveram opiniões muito divergentes em relação à representação ideal. Aparentemente, cada um teve uma visão distinta, alguns optando por incluir as relações dentro das entidades, e outros fora.

Aconteceram também sugestões quanto às palavras-chave possíveis, como **Element**, **ElementFather**, **ElementSource**, **Type**, e **Referential**. Ainda, seis (6) participantes sugeriram o uso de ponto e vírgula (;) para separação das declarações de elementos, e todos os treze (13) preconizaram a utilização de símbolos como parênteses, colchetes e/ou chaves para agrupar conjuntos similares de elementos.

Finalizada a dinâmica proposta, chegou-se ao último instrumento do grupo focal. Este artefato era composto de um exemplo de cada versão da gramática (Apêndice B.8) produzida preliminarmente para este trabalho. De posse das versões, foi pedido que os participantes realizassem a escolha entre as opções, apontando assim qual avaliavam como mais viável para modelagem ER. Também foi solicitado que fossem indicados os pontos positivos e negativos observados em cada modelo.

O segundo modelo acabou por ser escolhido pela maioria, como pode ser observado na Figura 28. Porém, ao final das discussões obteve-se um consenso de que a forma de definição de entidades do primeiro modelo e a disposição dos relacionamentos do segundo, em especial o uso das convenções nas cardinalidades, eram os mais adequados para a aplicação no ensino, indicando assim a necessidade de uma fusão de ambas as versões.

Figura 28 – Resultados da Discussão 3 - Grupo Focal.



Fonte: O autor.

5.6 A Ferramenta ERText

Esta seção apresenta a ferramenta com o *plugin* final da solução proposta. Este *plugin* pode tanto ser integrado com a RCP Eclipse, como pode ser insumo para um aplicativo independente (*standalone application*).

A diferença é que quando usado como um *plugin* do Eclipse o editor pode fornecer, além das funcionalidades da gramática, suporte a outras linguagens *e.g.* Java, PHP. Um produto independente, por outro lado, fornece toda a infraestrutura voltada unicamente a linguagem desenvolvida, e pode ser distribuído como uma ferramenta livre desde que seguidas as diretrizes da licença de software EPL-2.0².

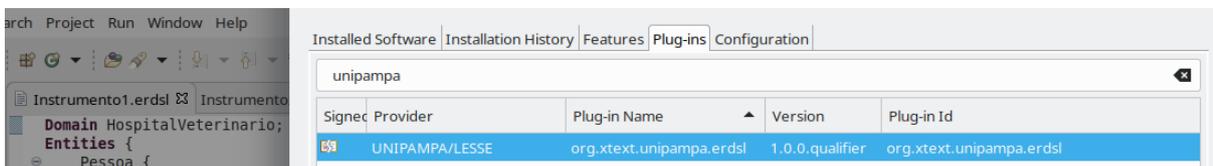
² <https://www.eclipse.org/legal/epl-2.0/>

5.6.1 Operação da ERText

Como dito anteriormente, a versão final da gramática resultou da fusão entre os duas versões previamente definidas e avaliadas com a execução do grupo focal. Esta versão está no Apêndice A.1. As principais mudanças foram as modificações das palavras reservadas `isA` e `isRelatedWith` por `is` e `relates`, respectivamente, além da adoção da convenção (0:1) (1:1) (0:N) (1:N) para as cardinalidades.

Após o *plugin* ser integrado no RCP, é possível verificar sua instalação no ambiente conforme exposto na Figura 29.

Figura 29 – Fragmento do *plugin* da solução instalado.



Fonte: O autor.

A Figura 30 apresenta a ferramenta em funcionamento, onde é possível ser feita a criação de arquivos de modelagem utilizando a DSL. Neste exemplo há a modelagem de sete (7) entidades e seis (6) relacionamentos, incluindo um autorelacionamento e um relacionamento ternário.

A modelagem na ferramenta ganha validação em tempo real, *syntax highlighting*, que indica erros de sintaxe em tempo de escrita, autocomplemento de código e *hovering*, uma funcionalidade que exibe informações sobre um item quando o cursor do *mouse* é colocado sobre ele.

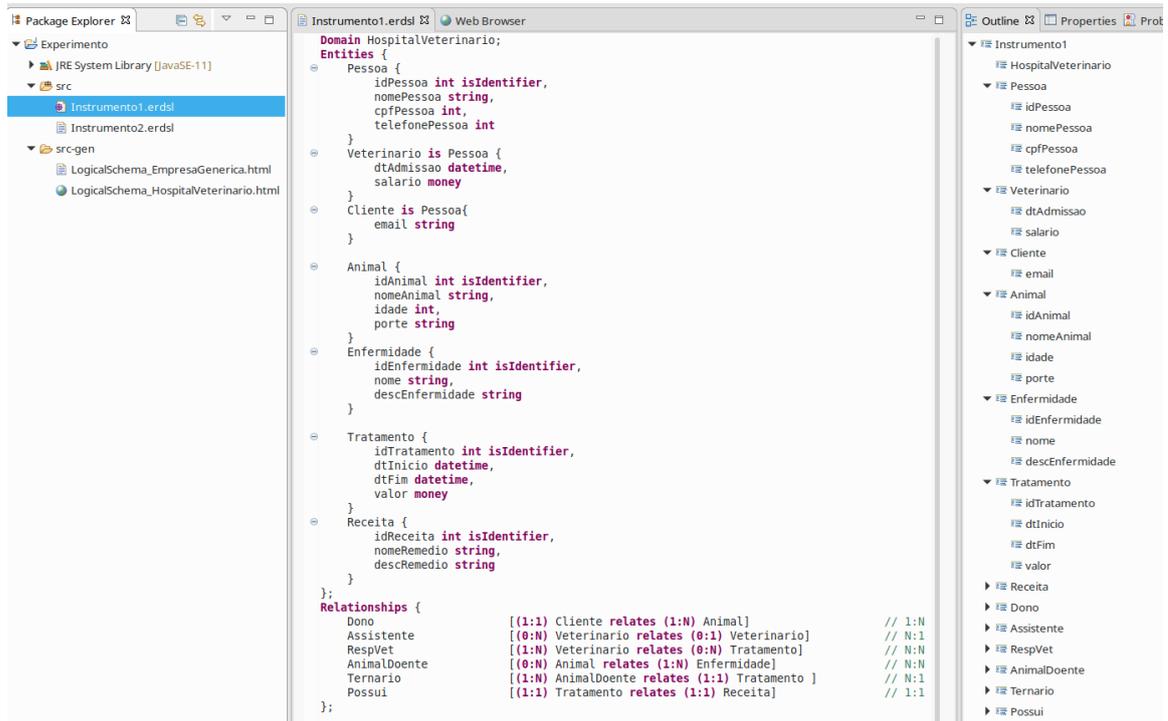
Se sabe que a definição dos tipos de dados não é prevista no modelo conceitual clássico mas, por questões relacionados a pretensão futura de realizar a geração de instruções SQL, foi decidido pela manutenção dessa escolha de projeto.

A função para o mapeamento e geração do modelo lógico é executada automaticamente toda a vez que uma modelo é salvo. Na figura anterior pode-se observar os arquivos *.html* na estrutura de diretórios na guia da esquerda, dentro da pasta *src-gen*. Foram gerados arquivos com esse formato para uma melhor visualização a partir de marcações no texto.

Estas marcações podem ser renderizadas por qualquer navegador de Internet, ou mesmo dentro do próprio ambiente, aumentando o poder de compreensão por parte do usuário. O modelo lógico derivado pelo gerador que mapeia o modelo conceitual é apresentado na Figura 31.

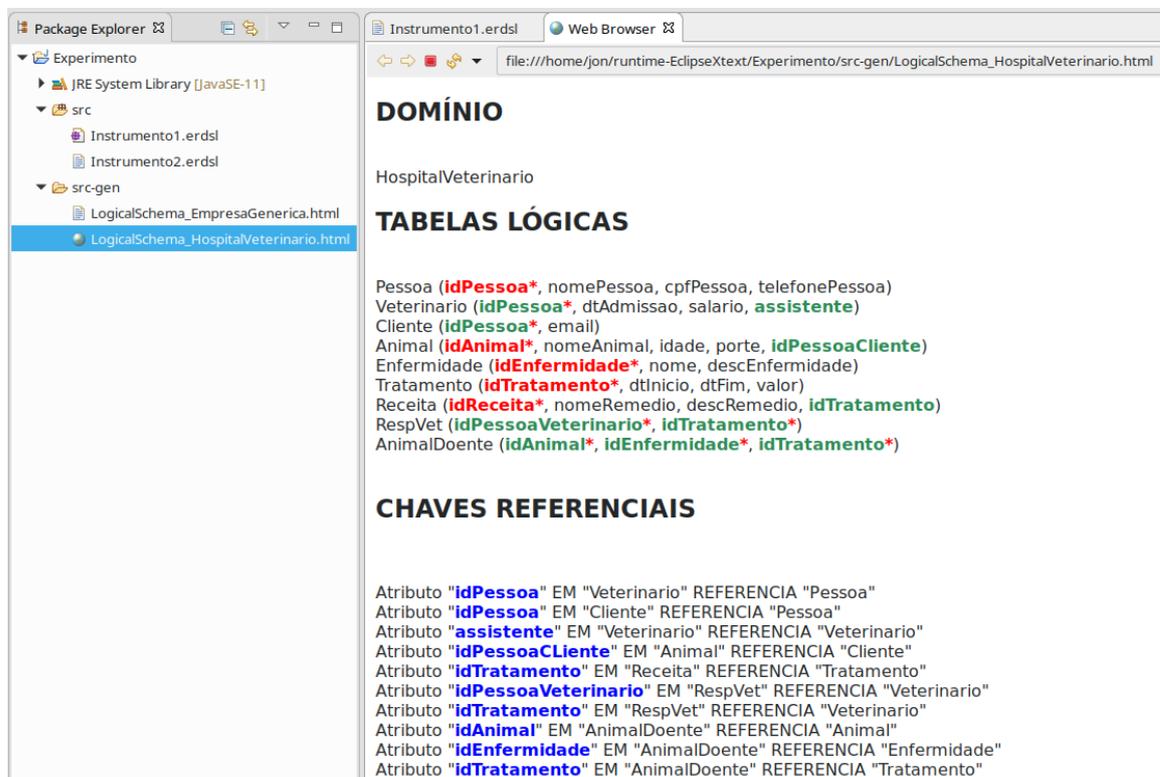
Mediante as premissas assumidas para a realização da transformação, o modelo original resulta em um novo modelo composto de nove (9) entidades, e também já possuindo suas integridade referenciais inferidas, ou seja, os registros que apontam para outros registros já são estabelecidos.

Figura 30 – Fragmento da solução sendo utilizada.



Fonte: O autor.

Figura 31 – Exemplo de modelo lógico gerado.



Fonte: O autor.

É importante salientar que as premissas foram escolhidas previamente com base no livro referência de Heuser (2009). A Figura 32 expõe as alternativas de mapeamento entre os modelos, sendo que as assinaladas são as que foram implementadas até o momento da entrega deste estudo. Em relação ao conceito de generalização/especialização, também foi necessário assumir uma ideia inicial que teria que ser tomada como verdade.

Figura 32 – Premissas adotadas para a transformação entre os modelos.

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
Relacionamentos 1:1			
	±	✓	×
	×	±	✓
	×	±	✓
Relacionamentos 1:n			
	±	✓	×
	±	✓	×
	×	✓	×
	×	✓	×
Relacionamentos n:n			
	✓	×	×
	✓	×	×
	✓	×	×

✓ Alternativa preferida ± Pode ser usada × Não usar

Fonte: Heuser (2009).

Segundo Heuser (2009), para estes casos existe duas alternativas passíveis de serem derivadas. A primeira recomenda o uso de uma única tabela para toda a hierarquia de entidades, ou seja, recomenda a fusão das tabelas. A segunda recomenda o uso de uma tabela por entidade modelada, desde que respeitado a integridade referencial, ou seja, as chaves primárias das entidades filhas devem apontar necessariamente para a chave primária da entidade pai. No caso da ferramenta resultante neste trabalho, optou-se pela segunda alternativa.

O gerador do modelo lógico foi desenvolvido com a GPL Xtend, e atualmente conta com cerca de quatrocentas linhas de código para realizar a transformação do modelo conceitual para o modelo lógico.

5.7 Lições do Capítulo

Neste capítulo foram expostos os requisitos, as decisões de projeto e a arquitetura da DSL proposta para a solução desenvolvida. Também foi demonstrado como o protótipo da versão preliminar da linguagem foi definido, um exemplo de uso e o grupo focal conduzido para a avaliação preliminar.

Com base nos resultados do grupo focal, foi possível chegar na versão final da DSL proposta, realizando então a sua implementação como um *plugin* integrado em um RCP Eclipse, gerando assim a ferramenta ERText. Desta forma o processo de modelagem com a nova linguagem criada ganhou recursos nativos como autocomplemento de código, formatação, validação com base nas restrições descritas na gramática e *syntax highlighting*.

Por fim, o projeto desta solução está disponível publicamente, sob a licença EPL-2.0, no *ProjetoDSL*³, pertencente ao grupo de pesquisa *Laboratory of Empirical Studies in Software Engineering* (LESSE) da UNIPAMPA.

³ Repositório ERDSL: <<https://github.com/ProjetoDSL/ERDSL>>

6 AVALIAÇÃO EXPERIMENTAL

Este capítulo apresenta o experimento controlado que foi executado com a finalidade de verificar a viabilidade de uso da solução desenvolvida neste trabalho. Foram realizadas atividades que envolveram a execução de práticas de modelagem comparando os resultados da proposta deste estudo com a abordagem gráfica do BrModelo, uma ferramenta reconhecidamente utilizada no ensino de BDs nos cursos de graduação.

Sendo assim, foram realizadas análises quantitativas e qualitativas dos resultados coletados do experimento. Isso foi realizado a partir de métricas referentes ao esforço (tempo), precisão, revocação, Medida-F, bem como em questionários qualitativos para avaliação de itens referentes à utilidade percebida e facilidade de uso de ambas as abordagens.

A Seção 6.1 apresenta a definição da avaliação, seguida de seu escopo na Seção 6.2 e do planejamento na Seção 6.3. As atividades conduzidas na operação do experimento são relatadas na Seção 6.4. Finalmente, os resultados obtidos são apresentados na Seção 6.5 e os lições do capítulo pontudas na Seção 6.6.

6.1 Definição da Avaliação

Segundo Wohlin et al. (2012), a Engenharia de Software Experimental (ESE) trata da utilização de abordagens científicas para o desenvolvimento, evolução, manutenção e validação de software. A ESE usa métodos científicos para fazer pesquisa e para também tomar decisões sobre mudanças na forma de desenvolver software.

Na ESE é possível realizar medições controladas, objetivas e orientadas a verificação, caracterizando assim a análise quantitativa. Da mesma forma, também é possível fazer observações naturalísticas, entrevistas e questionários, sendo essa uma abordagem orientada a descoberta e caracterizada como análise qualitativa.

As diferentes técnicas que a ESE engloba podem ser classificadas em dois grandes grupos: estudos experimentais e observacionais. Enquanto nas técnicas observacionais o pesquisador conduz a investigação sobre um objeto de estudo sem qualquer ação sobre ele, nas técnicas experimentais existe essa intervenção pela inclusão, exclusão ou modificação de algum fator estudado, procurando comparar o comportamento dos conjuntos de amostras perante algum cenário predefinido.

6.2 Escopo

Houve a escolha por uma abordagem experimental para avaliação da proposta, a qual foi executada com base na definição de um planejamento prévio. Tendo como base o modelo proposto por Wohlin et al. (2012), esta etapa do estudo contemplou as seguintes atividades: **(i)** Planejamento (Objetivo, questões de pesquisa, etc); **(ii)** Operação (Pre-

paração, Execução); **(iii)** Coleta dos Dados; **(iv)** Análise dos Dados e **(v)** Relatório e discussão dos resultados.

6.3 Planejamento

Nesta seção será apresentado o objetivo, as questões de pesquisa, o contexto, a hipótese, a seleção dos participantes, os instrumentos utilizados, o *design* experimental executado e as ameaças à validade.

6.3.1 Definição do Objetivo

O experimento tem por objetivo obter evidências a partir da comparação de duas abordagens para a modelagem de BDs relacionais, uma de forma gráfica e outra textual. Os intitulados tratamentos, foram: **(i)** Tratamento controle: a ferramentas BrModelo, de abordagem gráfica, e **(ii)** Tratamento experimental: ferramenta ERText, de abordagem textual e resultado da proposta deste trabalho.

O propósito do experimento é avaliar a viabilidade do uso de uma abordagem textual para apoio no processo de ensino-aprendizagem de modelagem conceitual de BD relacionais.

6.3.2 Questões de Pesquisa

Para a discussão dos resultados obtidos neste experimento optou-se pela formulação de quatro (4) QPs que tivessem relação com as atividades executadas no experimento, bem como na percepção de uso dos participantes.

- **QP1.** Qual abordagem demanda mais esforço médio durante a atividade de modelagem conceitual?
- **QP2.** Qual o nível da qualidade dos modelos produzidos utilizando a abordagem gráfica e a textual?
- **QP3.** Qual é a percepção dos sujeitos quanto a utilidade e facilidade de uso da DSL proposta?
- **QP4.** Qual a avaliação dos sujeitos em relação à representação dos construtores ER suportados na DSL proposta?

6.3.3 Contexto

O contexto do experimento é caracterizado conforme quatro (4) dimensões:

- **Processo:** Foi utilizada uma abordagem *in-vitro*, uma vez que as tarefas foram realizadas em laboratório sob condições controladas e sem atividades *online*.

- **Participantes:** Os participantes foram estudantes de graduação e pós-graduação em cursos de computação.
- **Realidade:** O experimento abordou um problema real, ou seja, a diferença no esforço dos indivíduos na modelagem conceitual de BDs relacionais, a qualidade dos artefatos produzidos e a percepção dos sujeitos usando duas abordagens distintas.
- **Generalidade:** Esta avaliação está inserida em um contexto específico, envolvendo alunos de computação e modelagem de BD. Contudo, as ideias gerais deste experimento podem ser replicadas em outro conjunto de participantes, abordagens ou DSLs que suportem modelagem de BDs.

6.3.4 Formulação de Hipótese

Geralmente a análise das hipóteses de um experimento controlado tem como base métricas e teste estatísticos. Com base nos dados coletados durante a execução de um experimento as hipóteses são testadas para verificar se é possível aceitar uma alternativa à hipótese nula (WOHLIN et al., 2012).

Para a formulação das hipóteses foram levadas em consideração as duas primeiras QPs. Em relação a QP1, sobre o esforço médio necessário utilizando cada abordagem, as hipóteses são as seguintes:

Hipótese Nula: $H_0 : \mu Tempo_G = \mu Tempo_T$: Não há diferença de esforço médio entre as abordagens textual e gráfica durante a modelagem conceitual de BDs relacionais.

Hipótese Alternativa: $H_1 : \mu Tempo_T \neq Tempo_G$: Existe diferença de esforço médio entre as abordagens textual e gráfica durante a modelagem conceitual de BDs relacionais.

Em relação a QP2, sobre a efetividade de modelagem utilizando cada abordagem, as hipóteses são as seguintes:

Hipótese Nula: $H_0 : \mu Efetividade_G = \mu Efetividade_T$: Não há diferença de efetividade entre as abordagens textual e gráfica durante a modelagem conceitual de BDs relacionais.

Hipótese Alternativa: $H_1 : \mu Efetividade_T \neq \mu Efetividade_G$: Há diferença de efetividade entre as abordagens textual e gráfica durante a modelagem conceitual de BDs relacionais.

Para a avaliação referente ao esforço foram utilizados o teste de normalidade Shapiro-Wilk e o Teste T pareado para amostras dependentes, em que foi levado em consideração os tempos coletados durante a execução das atividades de modelagem do experimento. O método Shapiro-Wilk testa a hipótese nula de que uma distribuição é normal, mediante o cálculo do valor W , onde após é então verificado na tabela do teste¹

¹ <http://www.uel.br/projetos/experimental/pages/arquivos/Probabilidades_Shapiro.pdf>

se ocorre a rejeição ou aceitação da hipótese. O cálculo do método Shapiro-Wilk é dado conforme a fórmula da Equação 6.1:

$$W = \frac{\left(\sum_{i=1}^n a_i x_{(i)}\right)^2}{\sum_{i=1}^n (x_i - \bar{x})} \quad (6.1)$$

Uma forma simplificada do Teste T pareado para amostras dependentes se dá conforme a fórmula da Equação 6.2:

$$t = \frac{m}{s/\sqrt{n}} \quad (6.2)$$

Nesta fórmula o m e o s são a média e o desvio padrão da diferença (d), respectivamente. O n corresponde ao tamanho de d , ou seja, o tamanho da amostra. Este teste de hipótese é usado para comparar as médias de duas amostras relacionadas, ou seja, quando se possui dois valores (par de valores) para uma mesma amostra. Contudo, para comparar as médias dos dois conjuntos de dados emparelhados, as diferenças entre todos os pares devem ser calculadas primeiro. O nível de significância alfa (α) utilizado foi de cinco por cento (5%). Os cálculos Shapiro-Wilk e Teste T pareado foram realizados com o suporte da linguagem R^{2,3}.

Para os testes da efetividade foram adotados os mesmos métodos estatísticos, porém ao invés do uso da métrica de tempo foi necessário outra grandeza. Sendo assim, foram realizados os cálculos da Medida-F, a qual é derivada a partir dos valores de *Precisão* e *Revocação*, para cada um dos modelos produzidos nas abordagens. O cálculo da Medida-F leva em consideração variáveis conhecidas como *Verdadeiros Positivos*, *Falsos Positivos* e *Falsos Negativos*.

- **Verdadeiros Positivos (VP)**: Quantidade de elementos corretamente modelados com a utilização da abordagem.
- **Falsos Positivos (FP)**: Quantidade de de elementos incorretamente modelados com a abordagem.
- **Falsos Negativos (FN)**: Quantidade de elementos não modelados com a abordagem.

A partir da identificação das variáveis é possível então calcular a *Precisão*, *Revocação* e *Medida-F* de cada modelo conforme as fórmulas:

- **Precisão (P)**: $\frac{VP}{VP + FP}$
- **Revocação (R)**: $\frac{VP}{VP + FN}$
- **Medida-F (F)**: $\frac{2 * (P * R)}{P + R}$

² <https://www.rdocumentation.org/packages/stats/versions/3.6.1/topics/shapiro.test>

³ <https://www.rdocumentation.org/packages/stats/versions/3.6.1/topics/t.test>

6.3.5 Seleção dos Participantes

Os sujeitos do experimento foram selecionados por amostragem não probabilística, indicada para estudos exploratórios. Esse tipo de amostragem se caracteriza a partir da escolha proposital de indivíduos com uma ou mais características que interessam ao objeto de estudo.

Neste experimento houve a participação de vinte e sete (27) alunos de graduação (Ciência da Computação e Engenharia de Software) e pós-graduação (Engenharia de Software) da Universidade Federal do Pampa - Campus Alegrete. Entre os participantes haviam catorze (14) alunos matriculados em disciplinas de BD.

Depois de identificar os potenciais participantes, aconteceu a definição da data para a execução do experimento, o que incluía o treinamento nas duas abordagens. Além disso, antes do treinamento seria solicitado aos participantes que respondessem a um questionário de perfil para nivelamento.

A partir dos dados extraídos dos questionários houve a distribuição aleatória dos sujeitos em dois grupos. Os grupos foram compostos por treze (13) e catorze (14) sujeitos em razão do número total de participantes ser ímpar, e também procurou-se manter um nível equilibrado de habilidades entre os conjuntos dos sujeitos.

6.3.6 Design da Avaliação

Segundo Wohlin et al. (2012), um experimento deve atender alguns conceitos fundamentais: tipo de projeto padrão, bloqueio, balanceamento e randomização.

Tipo de Projeto Padrão: Existem alguns tipos possíveis para o projeto padrão em um experimento, e este estudo adotou o de um (1) *Fator* com dois (2) *Tratamentos*. O *Fator* foi a modelagem de BDs relacionais, e os *Tratamentos* foram as duas (2) abordagens utilizadas (gráfica e textual).

Bloqueio: Este item se refere ao fato dos participantes do experimento poderem possuir diferentes níveis de experiência na área de modelagem e projeto de BDs. Em razão disso foi aplicado um questionário de perfil para o nivelamento dos sujeitos.

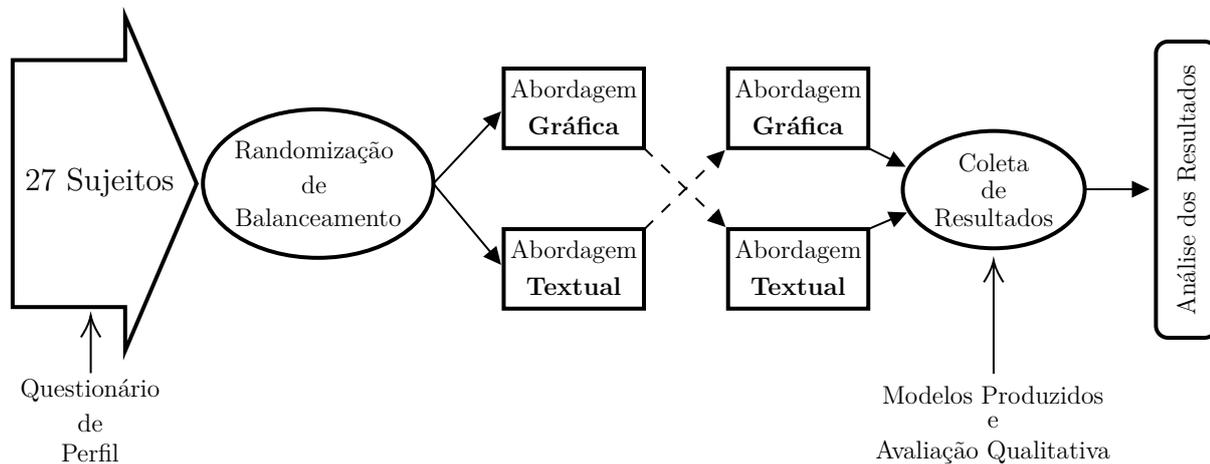
Balanceamento: Os participantes foram separados em dois grupos com níveis semelhantes de conhecimento. Desta forma, ambas as abordagens foram executadas por grupos homogêneos.

Randomização: Os sujeitos foram alocados aleatoriamente para cada grupo e abordagem. Os sujeitos executaram os dois tratamentos, caracterizando um *design* pareado. A ordem de execução das abordagens para cada grupo também foi definida aleatoriamente.

A Figura 33 exibe o desenho experimental da avaliação. Após todas as atividades realizadas pelo sujeitos utilizando os tratamentos, ocorre a coleta dos resultados. Esta etapa consiste na avaliação qualitativa feita pelos sujeitos e no salvamento dos modelos

produzidos pela aplicação dos tratamentos. Estes modelos servem posteriormente para uma avaliação qualitativa. Por fim, é realizada a etapa de análise, onde os dados são compilados e examinados com o objetivo de se extrair conclusões.

Figura 33 – *Design* do experimento.



Fonte: Adaptado de Silveira (2016).

6.3.7 Instrumentação

Como a participação no experimento foi voluntária, preparou-se um TCLE aos participantes para o registro da concordância de todos os sujeitos na execução das atividades, apresentado no Apêndice C.1. O questionário de perfil criado e aplicado para o balanceamento dos grupos e está presente no Apêndice C.3. Também foi elaborado um glossário para conceitos que foram utilizados na apresentação de abertura do experimento e durante os treinamentos, exposto no Apêndice C.2.

Com a finalidade de fornecer o suporte aos participantes da avaliação, foram providenciados instrumentos descrevendo o passo a passo com exemplos, exibidos nos Apêndices C.4 e C.5, para o uso das ferramentas utilizadas no experimento (BrModelo e ERTText). Além disso, foi realizado um treinamento que incluía vídeos com tutoriais de uso das abordagens em cada ferramenta. Os vídeos demonstravam como se deveria iniciar a modelagem, exemplos dos construtores previstos nos problemas que eles teriam que resolver e as recomendações para o salvamento dos artefatos produzidos.

Foram elaborados dois (2) instrumentos que continham um problema cada, com níveis similares de complexidade, os quais deveriam ser modelados e também anotados os horários de começo e término da atividade. Estes instrumentos estão nos Apêndices C.6 e C.7. Para a posterior avaliação por parte dos sujeitos foram criados outros dois (2) instrumentos. O primeiro apresentava sete (7) atributos de qualidade com base na ISO/IEC

25010⁴, uma norma para qualidade de produtos de software, e serviu para avaliar as abordagens das duas ferramentas do ponto de vista dos sujeitos que executaram as atividades de modelagem ER. Este instrumento é apresentado no Apêndice C.8. O segundo instrumento serviu para a avaliação da representação dos construtores da solução proposta neste estudo, e está disposto no Apêndice C.9. Os dados gerados por estes instrumentos serviram para responder as QPs qualitativas definidas para este experimento.

Ademais, foi também gerado um ambiente de trabalho igual para todos os sujeitos do experimento. Para tanto, criou-se uma máquina virtual que utilizava o sistema operacional Xubuntu⁵, uma distribuição Linux derivada do Ubuntu que utiliza o ambiente gráfico XFCE. Neste ambiente foram disponibilizados os materiais de apoio citados anteriormente, bem como as ferramentas de modelagem ER. Para evitar possíveis influências externas as máquinas virtuais não tinham acesso com a Internet, garantindo assim que os sujeitos não poderiam realizar consultas à conteúdos externos. Dessa forma, foi possível assegurar que todos os sujeitos realizaram as mesmas tarefas, e nas mesmas condições.

6.3.8 Ameaças à Validade

Objetivando que a avaliação tenha um resultado válido para o experimento proposto, foi necessário analisar e discutir as ameaças à validade, bem como as estratégias utilizadas para mitigá-las. Para a listagem das possíveis ameaças foram utilizadas os tópicos e recomendações levantadas no trabalho de Wohlin et al. (2012). Essas ameaças seguiram o padrão proposto e foram divididas em quatro (4) categorias, sendo: validade do construto, validade interna, validade externa e validade de conclusão.

6.3.8.1 Validade do Construto

A validade do construto diz respeito ao *design* do experimento e a fatores sociais.

Explicação pré-operacional inadequada: Esta ameaça está relacionada com o fato do experimento não ter o objetivo dos artefatos suficientemente definidos antes de serem traduzidos em medidas ou tratamentos. Para mitigar essa ameaça foi comparado o esforço em tempo de cada abordagem, bem como foi realizada a verificação da efetividade de acordo com os conceitos de *Precisão* e *Revocação* da *Medida-F*.

Interação de diferentes tratamentos. Se os sujeitos estiverem envolvidos em mais de um estudo, os tratamentos dos diferentes estudos poderão interagir e reverberar nos resultados finais. Como este experimento seguiu um *design* pareado, todos os sujeitos realizaram os dois tratamentos. Contudo, não foi identificado aprendizado entre a execução das atividades. Isto pode ser verificado por meio da normalidade das distribuições analisadas das amostras de esforço e efetividade, as quais demonstram que os resultados mantiveram-se semelhantes como um todo.

⁴ <https://www.iso.org/standard/35733.html>

⁵ <https://xubuntu.org/>

Predição das hipóteses: Quando os sujeitos participam de um experimento é possível que tentem descobrir qual é o objetivo ou o resultado pretendido do experimento. Este fato pode trazer viés ao comportamento, positivamente ou negativamente, dependendo da hipótese antecipada. Para mitigar essa ameaça os sujeitos não foram informados sobre maiores detalhes do experimento *e.g.* perguntas de pesquisa, hipóteses, objetivos e propósito.

6.3.8.2 Validade Interna

Ameaças à validade interna são influências que podem afetar as variáveis independentes em relação à causalidade, sem o conhecimento do pesquisador.

História: Há o risco de algum período temporal específico ter influência na realização do experimento. Para mitigar esta ameaça, e em razão do experimento ser realizado em ambiente acadêmico, todo o processo foi executado no mês de agosto, em que no geral os alunos não estão necessariamente sobrecarregados com atividades acadêmicas *e.g.* provas e trabalhos.

Maturação: Essa é a ameaça que ocorre em relação aos sujeitos reagirem de maneiras diferentes com o passar do tempo. Exemplos são quando os sujeitos são afetados negativamente (cansaço ou tédio) ou positivamente (aprendizado) durante o curso do experimento. Para mitigar essa ameaça foi comunicado desde o início aos sujeitos que poderiam encerrar a sua participação no momento que quisessem, sem qualquer tipo de penalização.

Testes: Se os testes forem repetidos, os sujeitos podem responder de maneira diferente em momentos diferentes, pois sabem como o teste é realizado. Se houver necessidade de familiarização com os testes, é importante que os resultados do teste não sejam devolvidos ao sujeito, para assim não apoiar o aprendizado não intencional. Não houve necessidade de repetição das atividades, uma vez que as mesmas foram executadas uma vez por participante em cada tratamento.

Instrumentação: Essa ameaça está relacionada aos artefatos usados para a execução da experiência, como formulários de coleta de dados, etc. Se estes foram mal projetados, a experiência é afetada negativamente. Para combater essa ameaça, todos os artefatos foram verificados e validados previamente em reuniões entre os pesquisadores envolvidos neste trabalho. Além disso, foi realizado um estudo piloto para validar o protocolo planejado para o experimento.

6.3.8.3 Validade Externa

As ameaças à validade externa são as condições relacionadas a replicação do experimento.

Sujeitos do experimento: Os sujeitos selecionados para o experimento podem não representar um grupo significativo para a área de estudo. Buscando tentar mitigar

essa ameaça o experimento foi realizado com estudantes de Engenharia de Software e Ciência da Computação, e logo, inseridos no contexto de uso da modelagem conceitual de BDs relacionais. Porém, o fato da amostra ter menos de trinta (30) elementos é uma ameaça estatística na área analisada, e não foi possível mitigar este fato.

Interação dos sujeitos com os artefatos de avaliação: Essa é a ameaça relacionada a aplicação dos artefatos de avaliação do experimento com os sujeitos. Dependendo do momento isto pode afetar os resultados. Se, por exemplo, um questionário for realizado alguns dias após a execução do experimento, as pessoas tendem a responder de maneira diferente do que fariam momentos após as atividades.

Interação de configuração e tratamento: São ameaças relacionadas ao fato de se usar uma configuração ou material não representativo. Para mitigar essa ameaça, foi utilizada uma documentação com base em *templates* e modelos tradicionais encontrados em material de ensino de BDs. Além disso, os artefatos foram validados com dois especialistas da área de Engenharia de Software.

Um ponto importante quanto a validade externa é que as ameaças podem ser reduzidas tornando o ambiente experimental o mais realista possível. Por outro lado, a realidade nem sempre é homogênea. Porém, mais importante é definir e relatar as características do ambiente, como experiência da equipe, ferramentas, métodos de avaliação e a aplicabilidade em um contexto específico (WOHLIN et al., 2012).

6.3.8.4 Validade da Conclusão

As ameaças à validade da conclusão estão relacionadas com questões que afetam a capacidade de se inferir uma conclusão correta sobre as relações entre os tratamentos e o resultado de um experimento.

Baixo poder estatístico: Uma possível ameaça nesta categoria é o baixo poder estatístico. Para tentar mitigar esta ameaça foi adotado alguns métodos estatísticos como o teste de normalidade Shapiro-Wilk, o Teste T pareado como teste de hipótese para amostras dependentes, e a Medida-F para análise qualitativa dos modelos produzidos.

Confiabilidade das medições: A confiabilidade das medições utilizadas tem impacto direto na validade do experimento como um todo. Para mitigar essa ameaça foram realizadas medições objetivas que não dependiam de julgamento subjetivo (esforço, medido em tempo, e Medida-F). Por outro lado, as métricas utilizadas para a avaliação qualitativa ainda serviram como insumo complementar na discussão dos resultados obtidos.

Ambiente experimental: O experimento deve ser realizado em um ambiente controlado, procurando evitar influências externas. Para mitigar esta possível ameaça os participantes foram orientados que não poderia ocorrer conversas durante todo o tempo das atividades, nem saídas do ambiente ou acesso a dispositivos eletrônicos.

6.4 Operação do Experimento

A operação do experimento pode ser dividida em duas (2) etapas: preparação e execução. Esta seção descreve de forma geral as ações realizadas em ambas as etapas.

6.4.1 Preparação

Inicialmente ocorreram reuniões entre os pesquisadores envolvidos para definição geral do planejamento e do modo de operação que deveria ser adotado. Buscando captar uma amostra significativa para o objeto do estudo, optou-se pelo contato com a docente responsável por ministrar as disciplinas de Banco de Dados (Engenharia de Software) e Banco de Dados I (Ciência da Computação) em 2019/2. Após reuniões para explicar os objetivos do trabalho, houve a concordância quanto a viabilidade do experimento. Com os objetivos iniciais alinhavados, a docente que colaborou realizou a divulgação dos questionários de perfil via plataforma Moodle para os alunos matriculados. Com os alunos do mestrado não foi possível o envio antecipado dos questionários, e por conta disso o mesmo foi aplicado no dia do experimento.

Com as respostas daqueles que retornaram o formulário de nivelamento, tantos os alunos de Engenharia de Software quanto de Ciência da Computação, começou-se a realizar com antecedência o nivelamento dos possíveis candidatos a participar do experimento. Após a elaboração de todos os artefatos que seriam utilizados no experimento, os mesmos foram analisados e validados em conjunto pelos pesquisadores envolvidos neste estudo, havendo ainda nesse percurso a necessidade de adequação de alguns instrumentos quanto as sugestões e eventuais correções necessárias.

6.4.2 Execução

No dia do experimento a primeira atividade realizada foi uma breve apresentação inicial onde era informado que o experimento era de caráter não avaliativo. Com isto esclarecido, era então disponibilizado o TCLE (Apêndice C.1) para todos os participantes. Após a assinatura do TCLE, foram distribuídos os questionários de perfil para aqueles participantes que ainda não haviam o preenchido anteriormente. Foi constatado que não havia fortes discrepâncias entre os níveis de conhecimento dos sujeitos, demonstrando assim que no geral era uma amostra homogênea.

Antes da divisão randômica dos grupos foi realizada então a fase de treinamento. Durante essa etapa foram apresentadas ambas as ferramentas que seriam utilizadas, passando uma visão geral de operação e respondendo a possíveis dúvidas que pudessem surgir. O treinamento incluía a exibição de vídeos demonstrando as ferramentas, primeiramente do BrModelo e após do editor Eclipse com o *plugin* da DSL proposta, respectivamente, com cerca de nove (9) e onze (11) minutos.

Então, deu-se início a fase de modelagem dos problemas propostos. Todos os participantes receberam o Instrumento 1 (Apêndice C.6) e foram informados com qual ferramenta deveriam desenvolver a solução. Foi solicitado que cada sujeito anotasse no instrumento sua identificação e o horário de início da tarefa. Não foi estipulado nenhum tempo limite para o término e, conforme os sujeitos finalizavam a modelagem, era pedido que cumprissem as orientações inclusas no material de apoio para o salvamento dos artefatos gerados. Com os modelos salvos, os instrumentos eram recolhidos e passava-se para a próxima tarefa descrita no Instrumento 2 (Apêndice C.7), porém sendo necessário o uso da abordagem inversa a que haviam utilizado inicialmente.

Ao término dos instrumentos que continham os problemas de modelagem, foi entregue os instrumentos de avaliação qualitativa (Apêndice C.8 e C.9). Conforme os participantes finalizavam o preenchimento da avaliação, eram então liberados. Com a conclusão do experimento por parte dos vinte e sete (27) sujeitos, encerrou-se o experimento e partiu-se para a etapa de tabulação e análise dos resultados.

6.5 Resultados

Esta seção relata a análise dos resultados obtidos a partir dos material coletado no experimento. É importante salientar que toda a análise estatística foi realizada com apoio da Linguagem R⁶, bem como nas recomendações do livro referência de Triola (2017).

6.5.1 Esforço

Para responder a QP1 referente ao esforço de uso das abordagens, os tempos foram extraídos dos instrumentos. A Tabela 11 exibe os valores associados aos sujeitos, bem como a ordem de execução das abordagens.

A partir dos valores brutos dos tempos foi calculada a diferença para ser possível realizar o teste de normalidade Shapiro-Wilk. Por ser um teste estatístico, esta técnica tem como produto a medida do valor- p . Para este teste foi adotado um nível de significância $\alpha = 5\%$. Isso significa que se o valor- p for menor que 5% ($p < 0.05$), a hipótese nula de que a distribuição é normal deve ser rejeitada.

Após os cálculos com o conjunto das diferenças dos tempos chegou-se a um valor- p de 0.606530. Como valor- $p > \alpha$, a hipótese nula foi aceita, concluindo assim que os dados são normalmente distribuídos. Em outras palavras, a diferença entre a amostra de dados e a distribuição normal não é grande o suficiente para ser estatisticamente significativa.

É importante ressaltar que quanto maior o valor- p , mais ele suporta uma hipótese nula. No caso do resultado obtido a chance de erro do tipo 1 (rejeitar uma hipótese nula que é correta) é muito alta, podendo ser traduzida em 60,65% (0.606530).

⁶ RStudio: <<https://rstudio.com>>

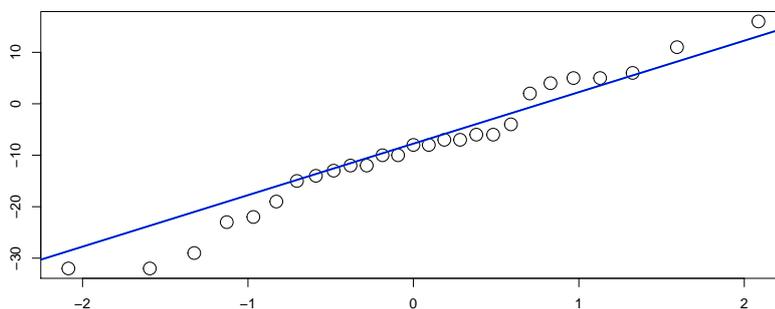
Tabela 11 – Tempos de execução de cada abordagens.

Sujeito	Abordagem (minutos)	Abordagem (minutos)	Abordagem (minutos)	
01	Gráfica	25	Textual	48
02	Gráfica	26	Textual	45
03	Gráfica	28	Textual	38
04	Gráfica	21	Textual	28
05	Gráfica	31	Textual	20
06	Gráfica	42	Textual	26
07	Gráfica	60	Textual	58
08	Gráfica	22	Textual	36
09	Gráfica	22	Textual	28
10	Gráfica	22	Textual	30
11	Gráfica	20	Textual	28
12	Gráfica	18	Textual	30
13	Gráfica	21	Textual	15
14	Gráfica	12	Textual	16
15	Textual	60	Gráfica	28
16	Textual	18	Gráfica	22
17	Textual	56	Gráfica	27
18	Textual	38	Gráfica	28
19	Textual	60	Gráfica	28
20	Textual	41	Gráfica	46
21	Textual	33	Gráfica	20
22	Textual	57	Gráfica	35
23	Textual	32	Gráfica	17
24	Textual	35	Gráfica	28
25	Textual	38	Gráfica	26
26	Textual	20	Gráfica	25
27	Textual	18	Gráfica	12

Fonte: O autor.

Ainda em relação ao teste de normalidade, o valor de W calculado foi de 0.970178, estando dentro do intervalo aceito do valor crítico de 95% (0.9242 : 1.0000). Isto significa que existe 95% de chances da amostra ter origem em uma população normal. A Figura 34 apresenta uma representação visual da distribuição analisada.

Figura 34 – Distribuição amostral referente ao esforço.



Fonte: O autor.

Tendo sido a amostra testada quanto à sua normalidade, foi possível realizar o teste da primeira hipótese estabelecida neste experimento. No teste T pareado para amostras dependentes foi utilizado um nível de significância $\alpha = 5\%$, com o qual se chegou a uma medida de 0.000962084 para o valor- p . Por ser um teste bicaudal, ou seja, que inclui uma

igualdade na sua hipótese nula, esse valor- p mostra evidências suficientes para garantir a rejeição da afirmativa de $H_0 : \mu Tempo_G = \mu Tempo_T$. Logo, a hipótese alternativa de que as abordagens possuem esforços diferentes é aceita, pois segundo o teste essa diferença é estatisticamente significativa.

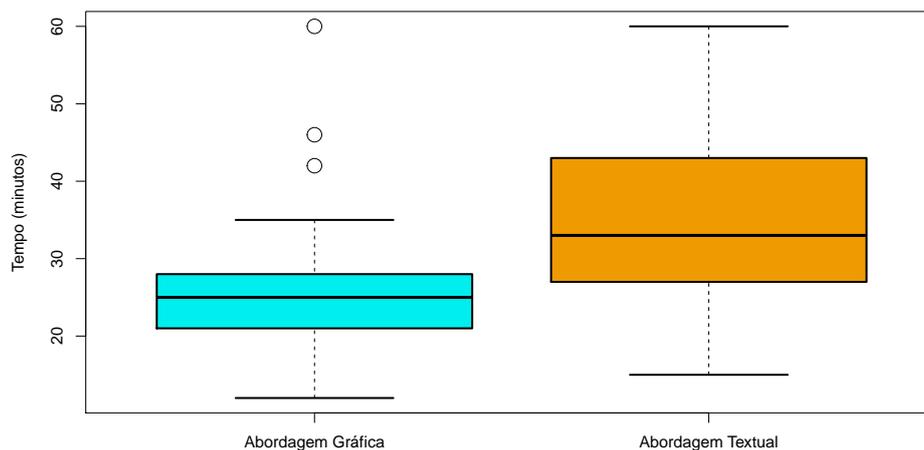
A Tabela 12 mostra os valores que possibilitam uma análise de dispersão. A Figura 35 exibe um *boxplot* com a variação de dados observados por meio destes dados. Com base nestes dados e na sua representação visual, é possível verificar que, em média, a abordagem gráfica leva vantagem sobre a abordagem textual.

Tabela 12 – Medidas relacionadas ao esforço médio.

	Abordagem	
	Gráfica	Textual
Limite superior	60.00	60.00
3° Quartil	28.00	43.00
Mediana	25.00	33.00
Média	26.37	35.26
1° Quartil	21.00	27.00
Limite Inferior	12.00	15.00
Desvio Padrão	10.12	14.03

Fonte: O autor.

Figura 35 – Esforços medidos nos tratamentos.



Fonte: O autor.

6.5.2 Efetividade

Para responder a QP2 referente a efetividade do uso das abordagens, os artefatos produzidos pelos sujeitos foram avaliados conforme os modelos de referência estabelecidos.

Tabela 13 – Avaliação dos modelos produzidos no experimento.

P	Abordagem Gráfica					Abordagem Textual				
	IM	IR	Precisão (%)	Revocação (%)	Medida-F (%)	IM	IR	Precisão (%)	Revocação (%)	Medida-F (%)
01	23	17	73.91	36.96	49.28	29	24	82.76	60.00	69.57
02	41	33	80.49	84.62	82.50	67	41	61.19	87.23	71.93
03	38	28	73.68	71.79	72.73	50	42	84.00	89.36	86.60
04	36	25	69.44	64.10	66.67	44	32	72.73	68.09	70.33
05	32	26	81.25	66.67	73.24	31	28	90.32	59.57	71.79
06	32	24	75.00	61.54	67.61	38	36	94.74	76.60	84.71
07	37	32	86.49	69.57	77.11	23	21	91.30	52.50	66.67
08	41	34	82.93	87.18	85.00	44	39	88.64	97.50	92.86
09	36	31	86.11	79.49	82.67	36	30	83.33	63.83	72.29
10	38	33	86.84	84.62	85.71	20	19	95.00	40.43	56.72
11	40	35	87.50	76.09	81.40	29	25	86.21	62.50	72.46
12	53	33	62.26	84.62	71.74	53	42	79.25	89.36	84.00
13	37	34	91.89	73.91	81.93	48	35	72.92	87.50	79.55
14	46	33	71.74	84.62	77.65	36	29	80.56	61.70	69.88
15	29	27	93.10	69.23	79.41	41	37	90.24	78.72	84.09
16	33	25	75.76	54.35	63.29	34	30	88.24	75.00	81.08
17	39	26	66.67	66.67	66.67	45	37	82.22	78.72	80.43
18	40	35	87.50	89.74	88.61	45	43	95.56	91.49	93.48
19	27	24	88.89	52.17	65.75	51	32	62.75	80.00	70.33
20	25	24	96.00	52.17	67.61	34	25	73.53	62.50	67.57
21	36	33	91.67	71.74	80.49	35	30	85.71	75.00	80.00
22	34	29	85.29	74.36	79.45	44	38	86.36	80.85	83.52
23	42	32	76.19	82.05	79.01	41	27	65.85	57.45	61.36
24	28	19	67.86	41.30	51.35	26	19	73.08	47.50	57.58
25	36	32	88.89	82.05	85.33	38	35	92.11	74.47	82.35
26	34	13	38.24	33.33	35.62	31	18	58.06	38.30	46.15
27	31	26	83.87	56.52	67.53	37	31	83.78	77.50	80.52

Legenda: IM (*Itens Modelados*) IR (*Itens Relevantes*)

Fonte: O autor.

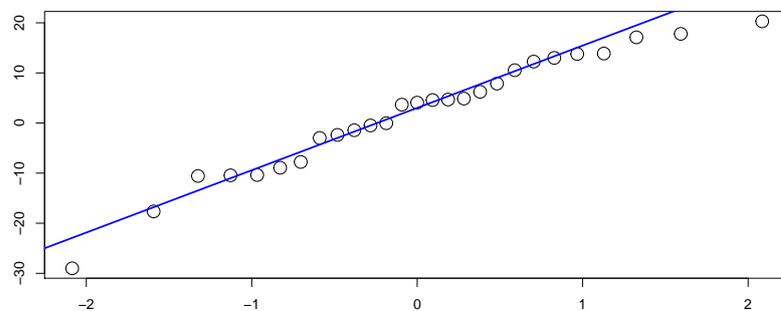
Nesta avaliação foi utilizada a Medida-F, uma medida proveniente da área de reconhecimento de padrões e recuperação de informação. A Medida-F representa a combinação da precisão e revocabilidade observada de um resultado em relação à uma referência.

Por definição, essa combinação se refere à fração de instâncias recuperadas que são relevantes (precisão) e a fração de instâncias relevantes que são recuperadas (revocabilidade). A Tabela 13 exibe estes valores associados obtidos pelos sujeitos, separados pelos tratamentos realizados.

Após a obtenção dos valores da Medida-F de cada modelo, foi realizado o teste de normalidade Shapiro-Wilk. Após os cálculos com o conjunto das diferenças da Medida-F de cada modelo, chegou-se a um valor- p de 0.404455. Com este resultado obtido a chance de erro do tipo 1 (rejeitar uma hipótese nula que é correta) pode ser muito alta, podendo ser traduzida em 40,45% (0.404455).

Como o valor- $p > \alpha$, a hipótese nula foi aceita, constatando assim que os dados são normalmente distribuídos, isto é, a diferença entre a amostra de dados e a distribuição normal não é grande o suficiente para ser estatisticamente significativa. A Figura 36 apresenta a representação visual da distribuição do conjunto de diferenças da Medida-F dos modelos.

Figura 36 – Distribuição amostral referente a efetividade.



Fonte: O autor.

Após a amostra ser testada quanto à sua normalidade, foi realizado o teste da segunda hipótese definida neste experimento. Desta vez, no teste T pareado para amostra dependentes, novamente foi utilizado um nível de significância $\alpha = 5\%$, com o qual se chegou a uma medida de 0.396468 para o valor- p .

Pela afirmativa original incluir uma igualdade, caracterizando também este teste como bicaudal, chegou-se a conclusão que o valor- p calculado demonstra que não há evidências suficientes para garantir a rejeição da afirmativa da hipótese nula original, denotada como $H_0 : \mu E_{fetividade_G} = \mu E_{fetividade_T}$. Logo, a hipótese nula de que as abordagens possuem efetividades iguais é aceita, pois segundo o teste a diferença média

da Medida-F entre os tratamentos não é estatisticamente significativa.

A Tabela 14 mostra medidas médias dos valores avaliados, e também fornecem a possibilidade para a realização de uma análise de dispersão.

Tabela 14 – Médias gerais da avaliação dos modelos produzidos no experimento.

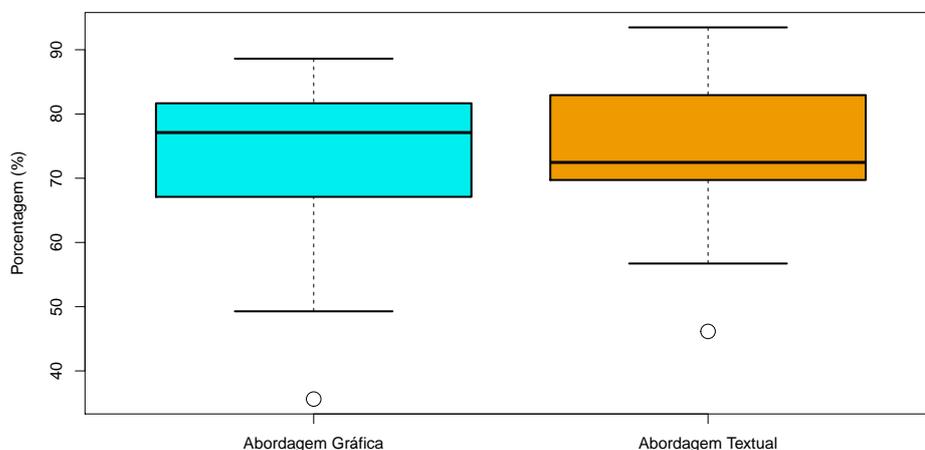
	Abordagem Gráfica					Abordagem Textual				
	IM	IR	P (%)	R (%)	MF (%)	IM	IR	P (%)	R (%)	MF (%)
Limite Superior	53.00	35.00	96.00	89.74	88.61	67.00	43.00	95.56	97.50	93.48
3° Quartil	39.50	33.00	87.50	82.05	81.66	44.50	37.00	89.44	80.43	82.93
Mediana	36.00	29.00	82.93	71.74	77.11	38.00	31.00	83.78	75.00	72.46
Média	35.70	28.26	79.61	68.57	72.79	38.89	31.30	81.50	70.88	74.73
1° Quartil	32.00	25.00	73.80	59.03	67.10	32.50	26.00	73.30	60.85	69.72
Limite Inferior	23.00	13.00	38.24	33.33	35.62	20.00	18.00	58.06	38.30	46.15
Desvio Padrão	6.33	5.63	11.94	15.39	12.16	9.97	7.30	10.44	15.36	10.94

Legenda: IM (*Itens Modelados*) IR (*Itens Relevantes*) P (*Precisão*) R (*Revocação*) MF (*Medida-F*)

Fonte: O autor.

O *boxplot* da Figura 37 exibe uma representação visual da Medida-F dos tratamentos aplicados. Com base neste gráfico é possível verificar o resultado obtido no teste de hipótese pois a dispersão dos dados não apresenta grande diferença entre as abordagens.

Figura 37 – *Boxplot* da Medida-F dos tratamentos.



Fonte: O autor.

6.5.3 Avaliação Qualitativa

A avaliação qualitativa se deu com a análise dos instrumentos aplicados após as tarefas de modelagem. O primeiro destes, presente no Apêndice C.8, foi utilizado para responder a QP3 referente a percepção de utilidade e facilidade de uso dos tratamentos. Isto ocorreu mediante a seleção de atributos de qualidade descritos na ISO/IEC 25010.

Para tanto, foi estabelecido uma escala Likert de um (1) até seis (6). Esta escala serviu para medir o nível de concordância dos sujeitos perante as afirmativas expostas no formulário. Optou-se por um número par de alternativas para evitar possíveis respostas neutras. Os sete (7) atributos de qualidade estão agrupados nas categorias de **Funcionalidade**, **Usabilidade** e **Qualidade em Uso**, sendo definidos a seguir.

- **Funcionalidade**

- *Conformidade*: é capacidade do software de estar de acordo com normas, convenções e/ou prescrições similares relacionadas as funcionalidades.

- **Usabilidade**

- *Inteligibilidade*: é capacidade do software possibilitar ao usuário compreender se o software é apropriado e como ele pode ser usado para tarefas e condições de uso específicas;
- *Apreensibilidade*: é a capacidade do software de possibilitar ao usuário aprender sua aplicação;
- *Operacionalidade*: é a capacidade do software de possibilitar ao usuário operá-lo e controlá-lo.

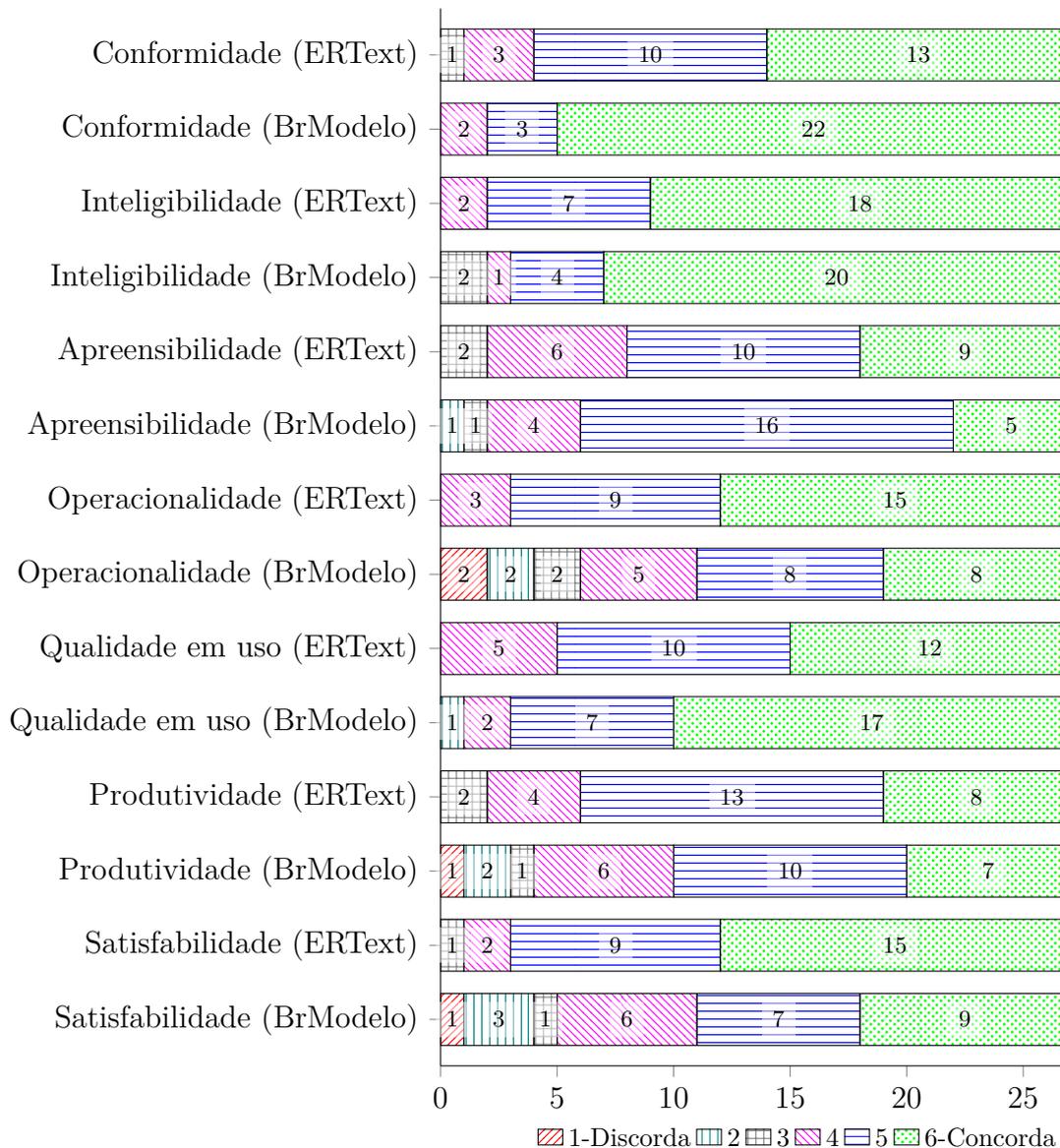
- **Qualidade em Uso**

- *Qualidade em Uso*: é a capacidade do software de permitir que usuários atinjam metas especificadas com eficácia, produtividade, segurança e satisfação em contextos de uso especificados;
- *Produtividade*: é a capacidade do software de permitir que seus usuários empreguem quantidade apropriada de recursos em relação à eficácia obtida, em um contexto de uso especificado;
- *Satisfação*: é a capacidade do software de satisfazer usuários, em um contexto de uso específico.

Após a sumarização dos resultados foi observada a boa aceitação por parte dos sujeitos para a ferramenta ERText, desenvolvida neste trabalho. A Figura 38 sintetiza as respostas recebidas para cada item, evidenciando um certo grau de similaridade de percepção dos sujeitos durante a aplicação dos tratamentos.

Um ponto que pode ser ressaltado é o conjunto de respostas positivas em relação ao item *Produtividade*, uma vez que no teste de hipótese relativo ao esforço o tratamento utilizando o BrModelo demonstrou uma menor necessidade de tempo para operação. Na análise dos comentários abertos deste formulário de avaliação, que solicitavam que os participantes indicassem pontos positivos e negativos das ferramentas, foi relatado de

Figura 38 – Avaliação dos atributos de qualidade dos tratamentos.



Fonte: O autor.

forma explícita que a característica de autocomplemento de código proporcionou uma sensação de agilidade no processo de modelagem.

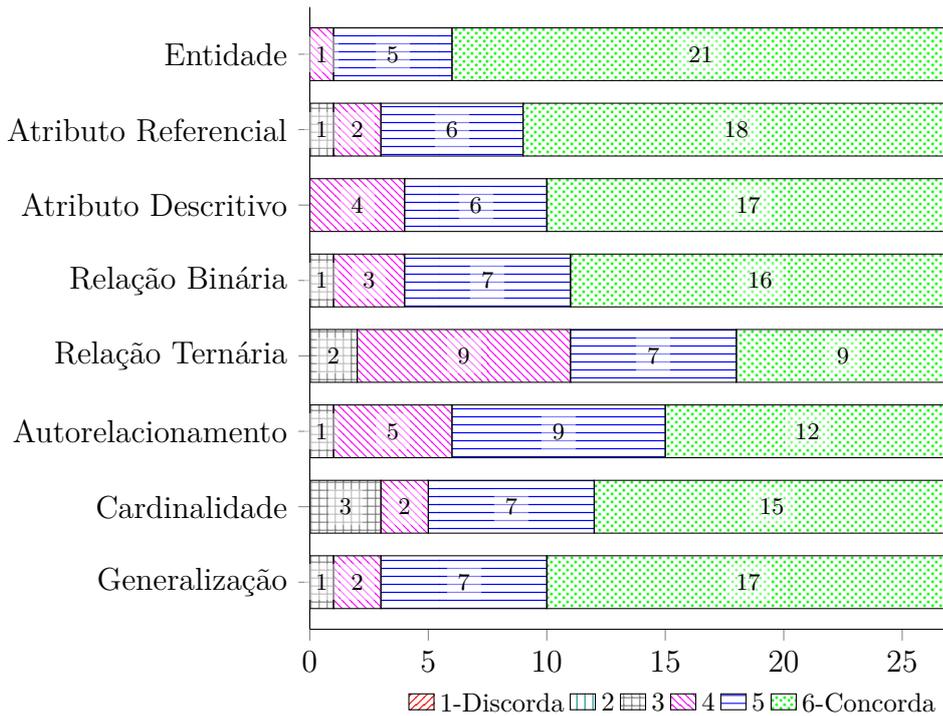
No que diz respeito a QP4, sobre a avaliação dos construtores da DSL, foram analisados os artefatos do instrumento presente no Apêndice C.9. Este instrumento listava os oito (8) construtores de modelagem ER cobertos pela DSL, dispostos com uma escala Likert de um (1) até seis (6). Novamente, optou-se pelo número par na escala para evitar respostas neutras que poderiam levar a uma interpretação mais subjetiva.

Como pode ser observado na Figura 39, que compila todas as respostas recebidas, os construtores relacionados as Entidades e Atributos Descritivos foram os melhores

avaliados, com todos os vinte sete (27) concordando com a sua atual representação.

Em contrapartida, todos os outros seis (6) obtiveram ao menos uma avaliação de discordância. Nesse sentido, destaca-se os construtores dos *Relacionamentos Ternários* e das *Cardinalidades*, com duas (2) e três (3) avaliações discordando das suas atuais representações respectivamente.

Figura 39 – Avaliação dos construtores da DSL.



Fonte: O autor.

6.6 Lições do Capítulo

Neste capítulo foi apresentado o protocolo e a execução do experimento controlado que foi conduzido para avaliação da proposta deste trabalho. Para tanto, foram observados fatores como esforço, efetividade, utilidade e facilidade de uso.

Com as respostas obtidas foi possível responder as quatro (4) QPs do experimento, bem como as duas (2) hipóteses associadas. A partir da análise feita é possível destacar os seguintes temas:

- *Esforço*: a abordagem gráfica para modelagem ER mostrou necessitar de menos esforço associado para realizar as tarefas solicitadas, porém considera-se que essa diferença pode ser diminuída com melhorias futuras na ERTText;

- *Efetividade*: não foi identificado uma diferença média que justifique afirmar que uma abordagem foi melhor do que a outra no que diz respeito a efetividade, entretanto observa-se que existe a necessidade de serem realizados testes que envolvam problemas de complexidades maiores para uma melhor avaliação.
- *Comparação qualitativa entre os tratamentos*: foi possível observar que há certo equilíbrio entre os tratamentos, mas destaca-se a avaliação positiva que o atributo *Produtividade* da ERText recebeu dos participantes do experimento. Em razão de ter sido a primeira vez que os sujeitos tiveram contato com a gramática, bem como da ferramenta estar em sua primeira versão, este é um ponto que chama a atenção e merece destaque;
- *Avaliação da linguagem proposta*: em decorrência da avaliação dos construtores da DSL proposta, existem algumas lacunas quanto ao *design* da linguagem que precisam ser revistos, em especial atenção as cardinalidades e relacionamentos ternários implementados. Uma vez que está previsto o prosseguimento do desenvolvimento da linguagem, a execução de uma possível refatoração, e também a implementação de novos construtores, se mostra um passo natural para sua evolução.

A partir dos resultados é possível concluir que existe viabilidade para o uso da abordagem textual para modelagem ER, uma vez que os resultados não obtiveram diferenças severas no que diz respeito as atividades executadas. Contudo, com as respostas alcançadas neste experimento verifica-se a necessidade da realização de melhorias, tanto na implementação da ERText quanto no planejamento e execução de um novo experimento controlado.

7 CONSIDERAÇÕES FINAIS

Este TCC se propôs a realizar um estudo que abordou a área de projeto e modelagem conceitual de BDs relacionais. Para tanto, foi necessário realizar um desenho de pesquisa que orientou toda a execução das atividades necessárias para atingir os objetivos específicos. O projeto envolveu a pesquisa bibliográfica para o embasamento teórico, a condução de um MLM visando encontrar estudos relacionados, o desenvolvimento da solução proposta e um experimento controlado para sua avaliação.

Através dos estudos realizados, assim como por meio dos resultados obtidos com o MLM e do experimento controlado, é possível concluir que o objetivo principal deste trabalho foi alcançado. Isto significa que se ofereceu uma alternativa ferramental para uso de uma abordagem textual para apoiar o processo de ensino-aprendizagem de projeto de modelagem de BD relacionais. Além disso, mediante uma possível exposição contínua da gramática aos alunos, apresentou-se uma perspectiva de melhora em relação ao esforço associado.

Além disso, destaca-se como principais contribuições deste TCC:

- A investigação da literatura científica e o relato de experiência, adquirida principalmente durante a etapa de projeto, acerca da forma de desenvolvimento de uma DSL textual utilizando um LW;
- A definição da gramática de uma DSL voltada à modelagem conceitual de BDs relacionais;
- A avaliação preliminar do protótipo e sua evolução;
- O uso de tecnologias *open source*;
- A Avaliação empírica do produto desenvolvido;
 - Definição de um protocolo experimental;
 - Execução;
 - Coleta de dados;
 - Análise quantitativa e qualitativa.
- A disponibilização do *plugin* da linguagem desenvolvida, com suporte à modelagem conceitual, bem como mapeamento e transformação para o modelo lógico.

Inegavelmente houveram diversos desafios durante o andamento deste TCC. A definição do protocolo de MLM e do experimento não foram atividades triviais, pois exigem experiência para que se consiga obter resultados confiáveis. O desenvolvimento da solução também se mostrou dificultosa em certos momentos, principalmente na implementação do gerador que faz o mapeamento do modelo conceitual para o modelo lógico. Entretanto,

por meio do uso de metodologias apoiadas na literatura, e também de protocolos bem definidos, foi possível superar as adversidades enfrentadas.

Outro ponto que merece destaque foi o esforço dos pesquisadores envolvidos neste estudo para a realização de três artigos. O primeiro abordou o SLM e foi submetido ao *ER - International Conference on Conceptual Modeling* (ER'2019). Outro abrangeu toda o MLM e foi submetido ao Simpósio Brasileiro de Bancos de Dados (SBBD'2019). E finalmente, um artigo descrevendo o protótipo desenvolvido, o qual foi submetido à Escola Regional de Engenharia de Software (ERES'2019), aprovado e apresentado.

Existe a intenção de que seja realizada a escrita de outro artigo, desta vez compilando os resultados de todo o estudo, a ser submetido à um *journal* inserido na área deste trabalho.

7.1 Trabalhos Futuros

Como sugestão de trabalhos futuros é possível indicar que através dos resultados do experimento executado foi identificada a necessidade de evolução da DSL desenvolvida. A evolução pode ser tanto em cima dos construtores atualmente suportados, quanto na definição de novos termos para cobrir outros conceitos importantes do modelo ER.

Também pode-se citar que existe espaço para a aplicação das regras de transformação entre modelos ainda não desenvolvidas, bem como a geração de *scripts* SQL que para tecnologias específicas *e.g.* PostgreSQL, MySQL e SQL Server.

REFERÊNCIAS

- AMELLER, D. **Considering Non-Functional Requirements in Model-Driven Engineering**. Dissertação (Mestrado) — Llenguatges i Sistemes Informàtics (LSI), June 2009. Disponível em: <<http://upcommons.upc.edu/pfc/handle/2099.1/7192>>. Citado na página 41.
- ANSI, A. N. S. I. Interim report: ANSI/X3/SPARC study group on data base management systems 75-02-08. **FDT - Bulletin of ACM SIGMOD**, v. 7, n. 2, p. 1–140, 1975. Citado na página 33.
- AYADI, M. G.; BOUSLIMI, R.; AKAICHI, J. A framework for medical and health care databases and data warehouses conceptual modeling support. **NetMAHIB**, v. 5, n. 1, p. 13, 2016. Citado 2 vezes nas páginas 58 e 59.
- BACKUS, J. W. The syntax and semantics of the proposed international algebraic language of the zurich acm-gamm conference. In: **IFIP Congress**. International Business Machines Corp., 1959. p. 125–131. Disponível em: <<http://dblp.uni-trier.de/db/conf/ifip/ifip1959.html#Backus59>>. Citado na página 44.
- BAILER-JONES, D. **Scientific Models in Philosophy of Science**. University of Pittsburgh Press, 2009. Disponível em: <<https://books.google.com.br/books?id=avur50J7UecC>>. Citado na página 40.
- Bailey, J. et al. Evidence relating to object-oriented software design: A survey. In: **First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)**. Madrid, Spain: IEEE, 2007. p. 482–484. Citado na página 51.
- BARKER, R. **CASE method - entity relationship modelling**. Wokingham, England: Addison-Wesley, 1990. (Computer aided systems engineering). Citado na página 65.
- BERGIN JR., T. J.; GIBSON JR., R. G. (Ed.). **History of Programming languages—II**. New York, NY, USA: ACM, 1996. Citado na página 44.
- BRAMBILLA, M.; CABOT, J.; WIMMER, M. **Model-Driven Software Engineering in Practice, Second Edition**. San Rafael, California (USA): Morgan & Claypool Publishers, 2017. (Synthesis Lectures on Software Engineering). Citado 2 vezes nas páginas 40 e 41.
- CELIKOVIC, M. et al. A DSL for EER data model specification. In: **Information Systems Development: Transforming Organisations and Society through Information Systems - Proceedings of the 23rd International Conference on Information Systems Development, ISD 2014, Varaždin, Croatia, September 2-4, 2014**. Varaždin, Croatia: Springer, 2014. Citado 2 vezes nas páginas 58 e 59.
- CHAMBERLIN, D. D.; BOYCE, R. F. Sequel: A structured english query language. In: **Proceedings of the 1974 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control**. New York, NY, USA: ACM, 1974. (SIGFIDET '74), p. 249–264. Citado na página 39.

- CHEN, P. P.-S. The entity-relationship model—toward a unified view of data. **ACM Trans. Database Syst.**, ACM, New York, NY, USA, v. 1, n. 1, p. 9–36, mar. 1976. Citado 2 vezes nas páginas 34 e 65.
- CODD, E. F. A relational model of data for large shared data banks. **Commun. ACM**, ACM, New York, NY, USA, v. 13, n. 6, p. 377–387, jun. 1970. ISSN 0001-0782. Citado na página 40.
- COOK, T.; CAMPBELL, D. **Quasi-Experimentation: Design and Analysis Issues for Field Settings**. Chicago, IL, USA: Houghton Mifflin, 1979. Citado na página 63.
- COUGO, P. **Modelagem conceitual e projeto de banco de dados**. Rio de Janeiro, RJ, Brasil: Elsevier Editora Ltda., 2013. Citado 3 vezes nas páginas 33, 34 e 38.
- DATE, C. **Introdução a Sistemas de Bancos de Dados**. Rio de Janeiro, RJ, Brasil: Elsevier Editora Ltda., 2004. Citado na página 33.
- DATE, C.; WARDEN, A. **Relational database writings, 1985-1989**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., 1990. (Relational database / C.J. Date, v. 1). ISBN 9780201508819. Citado na página 25.
- DEURSEN, A. van; KLINT, P.; VISSER, J. Domain-specific languages: An annotated bibliography. **SIGPLAN Not.**, ACM, New York, NY, USA, v. 35, n. 6, p. 26–36, jun. 2000. Disponível em: <<http://doi.acm.org/10.1145/352029.352035>>. Citado 2 vezes nas páginas 43 e 44.
- DIMITRIESKI, V. et al. Concepts and evaluation of the extended entity-relationship approach to database design in a multi-paradigm information system modeling tool. **Comput. Lang. Syst. Struct.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, v. 44, n. PC, p. 299–318, dez. 2015. Citado 4 vezes nas páginas 58, 59, 63 e 65.
- EDELWEISS, N.; LIVI, M. **Algoritmos e Programação com Exemplos em Pascal e C: Série Livros Didáticos UFRGS - Volume 23**. Bookman Editora, 2014. (23). ISBN 9788582601907. Disponível em: <<https://books.google.com.br/books?id=nTOpAwAAQBAJ>>. Citado na página 48.
- ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados**. Franca, SP, Brasil: Pearson Universidades, 2011. ISBN 9788579360855. Citado na página 25.
- FAVERI, C. de. **Uma Linguagem Específica de Domínio para Consulta em Código Orientado a Aspectos**. Dissertação (Mestrado) — Universidade Federal de Santa Maria, aug 2013. Citado 2 vezes nas páginas 44 e 45.
- FOWLER, M. Language workbenches: The killer-app for domain specific languages? 2005. Disponível em: <<http://www.martinfowler.com/articles/languageWorkbench.html>>. Citado 2 vezes nas páginas 45 e 46.
- FOWLER, M. **Domain Specific Languages**. 1st. ed. London, England: Addison-Wesley Professional, 2010. Citado 2 vezes nas páginas 43 e 45.

FRANTZ, R. Z. **Enterprise Application Integration: An Easy-to-Maintain Model-Driven Engineering Approach**. Tese (Doutorado) — Universidad de Sevilla. Departamento de Lenguajes y Sistemas Informáticos, 2012. Doctoral Thesis. Citado na página 43.

Garousi, V. and Felderer, M. and Mäntylä, M. V. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. **Information & Software Technology**, v. 106, p. 101–121, 2019. Disponível em: <<https://doi.org/10.1016/j.infsof.2018.09.006>>. Citado na página 51.

HAMMER, M.; LEOD, D. M. Database description with sdm: A semantic database model. **ACM Trans. Database Syst.**, ACM, New York, NY, USA, v. 6, n. 3, p. 351–386, set. 1981. Citado 2 vezes nas páginas 58 e 59.

HEUSER, C. **Projeto de banco de dados : Volume 4 da Série Livros didáticos informática UFRGS**. Porto Alegre, RS, Brasil: Bookman, 2009. (Livros didáticos informática UFRGS). Citado 5 vezes nas páginas 34, 36, 38, 68 e 80.

JAGANNATHAN, D. et al. Sim: A database system based on the semantic data model. **SIGMOD Rec.**, ACM, New York, NY, USA, v. 17, n. 3, p. 46–55, jun. 1988. Citado 2 vezes nas páginas 58 e 59.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing (2Nd Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009. ISBN 0131873210. Citado na página 47.

KERSTEN, M. et al. Sciq1, a query language for science applications. In: **Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases**. New York, NY, USA: ACM, 2011. (AD '11), p. 1–12. Citado 2 vezes nas páginas 58 e 59.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. Keele/Durham, England, 2007. Citado 2 vezes nas páginas 51 e 53.

KLEPPE, A. et al. **MDA Explained: The Model Driven Architecture : Practice and Promise**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. (Object technology). Citado na página 42.

KONTIO, J.; BRAGGE, J.; LEHTOLA, L. The focus group method as an empirical tool in software engineering. In: _____. **Guide to Advanced Empirical Software Engineering**. London: Springer London, 2008. p. 93–116. ISBN 978-1-84800-044-5. Disponível em: <https://doi.org/10.1007/978-1-84800-044-5_4>. Citado 3 vezes nas páginas 73, 74 e 75.

KRISHNA, S. **Introduction to Database and Knowledge-base Systems**. Singapore: World Scientific, 1992. (Computer Science Series). ISBN 9789810206192. Citado na página 25.

LINZ, P. **An Introduction to Formal Languages and Automata**. Jones & Bartlett Learning, 2016. (G - Reference, Information and Interdisciplinary Subjects Series). ISBN 9781284077247. Disponível em: <<https://books.google.com.br/books?id=pDaUCwAAQBAJ>>. Citado 2 vezes nas páginas 47 e 48.

- LITWIN, W. et al. Msq: A multidatabase language. **Inf. Sci.**, Elsevier Science Inc., New York, NY, USA, v. 49, n. 1-3, p. 59–101, out. 1989. Citado 2 vezes nas páginas 58 e 59.
- MARTELLI, R.; FILHO, O.; CABRAL, A. de L. **Modelagem e banco de dados**. São Paulo, SP, Brasil: Editora Senac São Paulo, 2018. (Informática). Citado na página 38.
- MARTIN, J. **Application Development Without Programmers**. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1982. Citado na página 44.
- MAZAIRAC, W.; BEETZ, J. Bimql - an open query language for building information models. **Adv. Eng. Inform.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 27, n. 4, p. 444–456, out. 2013. Disponível em: <<http://dx.doi.org/10.1016/j.aei.2013.06.001>>. Citado 2 vezes nas páginas 58 e 59.
- MERNIK, M.; HEERING, J.; SLOANE, A. M. When and how to develop domain-specific languages. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 37, n. 4, p. 316–344, dez. 2005. Disponível em: <<http://doi.acm.org/10.1145/1118890.1118892>>. Citado na página 44.
- NAKAGAWA, E. et al. **Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática**. Rio de Janeiro, RJ, Brasil: Elsevier Editora Ltda., 2017. Citado 3 vezes nas páginas 51, 52 e 54.
- NARDI, B. A. **A Small Matter of Programming: Perspectives on End User Computing**. 1st. ed. Cambridge, MA, USA: MIT Press, 1993. Citado na página 44.
- OBE0; TYPEFOX. **Xtext/Sirius - Integration The Main Use-Cases**. [S.l.], 2017. Disponível em: <https://www.obeodesigner.com/resource/white-paper/WhitePaper_XtextSirius_EN.pdf>. Citado na página 69.
- PEFFERS, K. et al. A design science research methodology for information systems research. **J. Manage. Inf. Syst.**, M. E. Sharpe, Inc., Armonk, NY, USA, v. 24, n. 3, p. 45–77, dez. 2007. Citado na página 29.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: **EASE'08**. Swindon, UK: BCS Learning Development Ltd., 2008. v. 8, p. 68–77. Citado 2 vezes nas páginas 51 e 52.
- PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico - 2ª Edição**. Novo Hamburgo, RS, Brasil: Editora Feevale, 2013. Citado 2 vezes nas páginas 29 e 30.
- RAMAKRISHNAN, R.; GEHRKE, J. **Database Management Systems**. 3. ed. New York, NY, USA: McGraw-Hill, Inc., 2003. ISBN 0072465638, 9780072465631. Citado na página 40.
- ROSS, D. T. History of programming languages. In: WEXELBLAT, R. L. (Ed.). New York, NY, USA: ACM, 1981. cap. Origins of the APT Language for Automatically Programmed Tools, p. 279–338. Disponível em: <<http://doi.acm.org/10.1145/800025.1198374>>. Citado na página 44.
- SALGADO, A. C.; MEDEIROS, C. B. Ensino de banco de dados. In: **Anais do Workshop de Informática na Escola**. [S.l.: s.n.], 1995. Citado na página 26.

- SAMMET, J. E. Programming languages: History and future. **Commun. ACM**, ACM, New York, NY, USA, v. 15, n. 7, p. 601–610, jul. 1972. Disponível em: <<http://doi.acm.org/10.1145/361454.361485>>. Citado na página 44.
- SCHMIDT, D. C. Guest editor’s introduction: Model-driven engineering. **Computer**, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 39, n. 2, p. 25–31, fev. 2006. Citado na página 43.
- SHIPMAN, D. W. The functional data model and the data languages dplex. **ACM Trans. Database Syst.**, ACM, New York, NY, USA, v. 6, n. 1, p. 140–173, mar. 1981. Disponível em: <<http://doi.acm.org/10.1145/319540.319561>>. Citado 2 vezes nas páginas 58 e 59.
- SILBERSCHATZ, A.; KORTH, H.; SUDARSHAN, S. **Sistema de bancos de dados**. São Paulo, SP, Brasil: Makron Books, 1999. ISBN 9788534610735. Citado na página 25.
- SILVEIRA, M. B. da. **Canopus : a domain-specific language for modeling performance testing**. Tese (Doutorado), 2016. Pontifícia Universidade Católica do Rio Grande do Sul. Disponível em: <<http://tede2.pucrs.br/tede2/handle/tede/6861>>. Citado na página 88.
- SOMMERVILLE, I. **Engenharia de software**. Franca, SP, Brasil: Pearson, 2011. Citado 2 vezes nas páginas 25 e 41.
- TIAN, H. et al. Neuroql: A domain-specific query language for neuroscience data. In: **Proceedings of the 2006 International Conference on Current Trends in Database Technology**. Berlin, Heidelberg, Germany: Springer-Verlag, 2006. (EDBT’06), p. 613–624. Disponível em: <http://dx.doi.org/10.1007/11896548_46>. Citado 2 vezes nas páginas 58 e 59.
- TRIOLA, M. **Introdução à Estatística**. São Paulo, SP, Brasil: GEN - LTC, 2017. ISBN 9788521633747. Citado na página 93.
- Wachsmuth, G. H.; Konat, G. D. P.; Visser, E. Language design with the spoofax language workbench. **IEEE Software**, v. 31, n. 5, p. 35–43, Sep. 2014. Citado na página 46.
- WAZLAWICK, R. **Metodologia de Pesquisa para Ciência da Computação**. [S.l.]: Elsevier Editora Ltda., 2017. Citado na página 29.
- WEST, M. **Developing High Quality Data Models**. Amsterdam, Netherlands: Elsevier Science, 2011. Citado 2 vezes nas páginas 38 e 39.
- WEXELBLAT, R. L. (Ed.). **History of Programming Languages**. New York, NY, USA: ACM, 1981. Citado na página 44.
- WOHLIN, C. et al. **Experimentation in Software Engineering**. London, England: Springer Publishing Company, Incorporated, 2012. Citado 6 vezes nas páginas 63, 83, 85, 87, 89 e 91.

Apêndices

APÊNDICE A – DSL PROPOSTA

A.1 Versão Final da Gramática da DSL

```

1 grammar org.xtext.unipampa.erdsl.ErDsl
2 with org.eclipse.xtext.common.Terminals
3
4 generate erDsl "http://www.xtext.org/unipampa/erdsl/ErDsl"
5
6 ERModel:
7     domain=Domain ';'
8     ('Entities' '{') entities+=Entity+ ('}' ';')
9     ('Relationships' '{') relations+=Relation* ('}' ';');
10
11 Domain:
12     'Domain' name=ID;
13
14 Attribute:
15     name=ID type=DataType (isKey?='isIdentifier')?;
16
17 Entity:
18     name=ID ('is' is+=[Entity])*
19     ('{' attributes+=Attribute
20     (',' attributes+=Attribute)* '}'?);
21
22 Relation:
23     (name=ID)? ('[' leftEnding=RelationSide
24     'relates'
25     rightEnding=RelationSide ']')
26     ('{' attributes+=Attribute
27     (',' attributes+=Attribute)* '}'?)*;
28
29
30 RelationSide:
31     Cardinality=('(0:1)' | '(1:1)' | '(0:N)' | '(1:N)')
32     target=[Entity] | target=[Relation];
33
34 enum DataType:
35     INT='int'           | DOUBLE='double'       |
36     MONEY='money'      | STRING='string'       |
37     BOOLEAN='boolean'  | DATETIME='datetime'  |
38     BLOB='file';

```


APÊNDICE B – GRUPO FOCAL

B.1 Termo de Consentimento



Termo de Consentimento Livre e Esclarecido

Você está sendo convidado(a) para participar, como voluntário(a), de um estudo para avaliação da gramática de uma linguagem para modelagem conceitual de bancos de dados relacionais. Este estudo é parte integrante da pesquisa chamada “Uma Linguagem Específica de Domínio para a Representação de Modelos Conceituais de Bancos de Dados Relacionais”, coordenada pelo Prof. Dr. Maicon Bernardino da Silveira e pelo Prof. Dr. Fábio Paulo Basso.

Por meio deste documento, a qualquer momento, você poderá solicitar esclarecimentos adicionais sobre o estudo ou sobre a pesquisa. Também poderá retirar seu consentimento ou interromper a participação a qualquer momento, sem sofrer qualquer tipo de penalidade ou prejuízo. Ao participar deste estudo você não terá nenhum custo e nem receberá qualquer vantagem financeira. Seus dados pessoais serão mantidos em sigilo. Os resultados deste estudo serão armazenados pelo pesquisador responsável e poderão ser divulgados em publicações científicas.

Para participar deste estudo, você precisará: ouvir as explicações, ler o material fornecido, participar das discussões e reportar sugestões de melhorias e críticas. Ao participar deste estudo você corre o risco de frustrar-se por não conseguir realizar as atividades solicitadas. Esperamos que os resultados deste estudo contribuam para a minimização da complexidade da modelagem conceitual de bancos de dados relacionais.

Após esses esclarecimentos, solicitamos o seu consentimento de forma livre para que participe deste estudo. Para tanto, preencha os itens a seguir.

Eu, _____, tendo em vista as informações acima apresentadas, de forma livre e esclarecida, aceito participar deste estudo.

_____, ____ de _____ de _____.

Assinatura do Participante

B.2 Glossário de Termos



Glossário de Termos

Linguagem de Domínio Específico: É uma linguagem de programação ou linguagem de especificação executável que oferece, por meio de notações e abstrações apropriadas, poder expressivo focado e restrito a um domínio de problema específico [5, 3].

Modelo Conceitual: É a descrição de um banco de dados de forma independente da implementação usada em sistemas de gerenciamento de banco de dados (SGBDs) [4].

Abordagem ER: A abordagem entidade-relacionamento é a técnica [1] mais difundida para criação de modelos conceituais de bancos de dados. Esta abordagem foi tão bem aceita que passou a ser considerada uma referência definitiva para a modelagem de banco de dados relacionais. É composta basicamente por um método de diagramação e de conceitos que devem ser respeitados [2].

Entidade: É uma representação de um conjunto de objetos do domínio modelado [4].

Atributo: São as características próprias das entidades [4].

Relacionamento: É a representação da associação entre os objetos modelados [4].

Cardinalidade: Registra o número de ocorrências, possuindo um número mínimo e máximo, com que as entidades podem se associar [4].

Generalização/Especialização: Associado a este conceito está a ideia de herança de propriedades, ou seja, a atribuição de propriedades particulares a um subconjunto de ocorrências de uma entidade genérica [4].

Referências

- [1] P. P.-S. Chen. The entity-relationship model—toward a unified view of data. *ACM Trans. Database Syst.*, 1(1):9–36, Mar. 1976.
- [2] P. Cougo. *Modelagem conceitual e projeto de banco de dados*. Elsevier Editora Ltda., 2013.
- [3] M. Fowler. *Domain Specific Languages*. Addison-Wesley Professional, 1st edition, 2010.
- [4] C. Heuser. *Projeto de banco de dados : Volume 4 da Série Livros didáticos informática UFRGS*. Livros didáticos informática UFRGS. Bookman, 2009.
- [5] A. van Deursen, P. Klint, and J. Visser. Domain-specific languages: An annotated bibliography. *SIGPLAN Not.*, 35(6):26–36, June 2000.

B.3 Questionário de Perfil

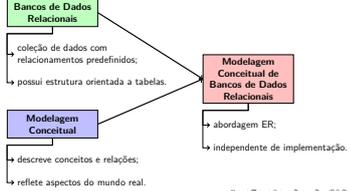
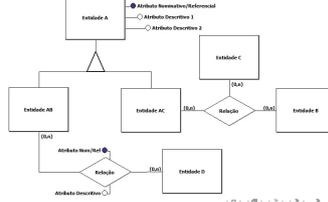
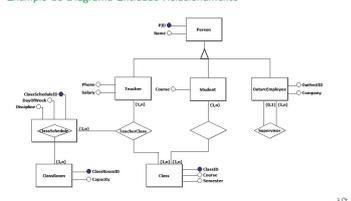


Questionário de Perfil do Participante

Nome: _____
Local: _____ Data: _____

1. Qual o seu nível de formação?
 - Graduação.
 - Especialização.
 - Mestrado.
 - Doutorado.
2. Qual a sua experiência com desenvolvimento de software?
 - ≤ 2 anos.
 - > 2 anos e ≤ 4 anos.
 - > 4 anos e ≤ 7 anos.
 - > 7 anos e ≤ 10 anos.
 - > 10 anos.
3. Qual a sua experiência com modelagem conceitual de bancos de dados relacionais?
 - Nenhuma.
 - Eu já li sobre.
 - Eu usei só na faculdade.
 - Eu aplico em meus projetos.
 - Eu desenvolvo pesquisa sobre o tema.
4. Com que frequência você usa abordagens (e.g. Modelagem ER) de modelagem conceitual?
 - Eu nunca uso.
 - Eu usei só na faculdade.
 - Eu eventualmente uso em meus projetos.
 - Eu frequentemente uso em meus projetos.
 - Eu sempre uso em meus projetos.
5. Qual o seu contato com linguagens de domínio específico?
 - Eu nunca ouvi falar.
 - Eu já li sobre elas.
 - Eu usei só na faculdade.
 - Eu sou usuário deste tipo de linguagem.
 - Eu desenvolvo este tipo de linguagem.

B.4 Roteiro

 <p>Uma Linguagem Específica de Domínio para a Representação de Modelos Conceituais de Bancos de Dados Relacionais</p> <p>Jonathan Riquelmo Prof. Dr. Maicon Bernardino Prof. Dr. Fábio Basso Universidade Federal do Pampa Laboratory of Empirical Studies in Software Engineering (LESSE)</p>	<h3>Roteiro</h3> <ol style="list-style-type: none"> 1 Abertura 2 Fundamentação 3 Discussões 4 Fechamento 	<h3>Roteiro</h3> <ol style="list-style-type: none"> 1 Abertura 2 Fundamentação 3 Discussões 4 Fechamento 	<h3>Preliminares</h3> <p>Obrigado por participar!</p> <p>O que é um grupo focal?</p> <p>Um grupo focal é um método de pesquisa que reúne pessoas em uma sala para prover feedback sobre algo.</p> <p>Como funciona um grupo focal?</p> 
<h3>Objetivo Principal</h3> <p>Discutir sobre a gramática de uma linguagem de domínio específico (Domain Specific Language - DSL) para modelagem conceitual de bancos de dados relacionais.</p>	<h3>Primeira Tarefa</h3> <ol style="list-style-type: none"> 1. Ler o Termo de Consentimento Livre e Esclarecido. <ul style="list-style-type: none"> Se concordar, por favor, complete-o com o nome e assine o documento. 2. Responder o Questionário de Perfil do Participante. 	<h3>Roteiro</h3> <ol style="list-style-type: none"> 1 Abertura 2 Fundamentação 3 Discussões 4 Fechamento 	<h3>Conceitos Principais</h3>  <p>Bancos de Dados Relacionais</p> <ul style="list-style-type: none"> coleção de dados com relacionamentos predefinidos; possui estrutura orientada a tabelas. <p>Modelagem Conceitual</p> <ul style="list-style-type: none"> descreve conceitos e relações; reflete aspectos do mundo real. <p>Modelagem Conceitual de Bancos de Dados Relacionais</p> <ul style="list-style-type: none"> abordagem ER; independente de implementação.
<h3>Conceitos Relacionados</h3> <p>Entidade: é o conjunto de objetos da realidade modelada.</p> <ul style="list-style-type: none"> ► Pessoa, Departamento, Produto <p>Relacionamento: é o conjunto de associações entre entidades.</p> <ul style="list-style-type: none"> ► Nome da Pessoa ► Preço do Produto <p>Atributo: é um dado associado a cada ocorrência de uma entidade ou de um relacionamento.</p> <ul style="list-style-type: none"> ► Muitas Pessoas estão lotadas em um departamento ► Um ou muitos Produtos pertencem a uma Marca 	<h3>Conceitos Relacionados</h3> <p>Cardinalidade são as ocorrências de uma entidade que podem estar associadas através de um relacionamento.</p> <ul style="list-style-type: none"> ► Muitas (1,n) Pessoas estão lotadas em um (1,1) departamento ► Um ou muitos (1,n) Produtos pertencem a uma (1,1) Marca ► Uma Pessoa (1,1) possui nenhum ou muitos (0,n) Dependentes <p>Generalização/Especialização é a identificação de subconjuntos de entidades que compartilham características em comum.</p> <ul style="list-style-type: none"> ► Elemento de caracterização semântica ► Conceito de herança 	<h3>Diagrama Entidade-Relacionamento</h3> 	<h3>Requisitos para Modelagem</h3> <p>A modelagem conceitual de um BD relacional deve:</p> <p>Req-1: identificar as entidades relevantes para o domínio.</p> <p>Req-2: identificar os atributos que definem as características das entidades ou relacionamentos.</p> <p>Req-3: identificar as associações entre as entidades.</p> <p>Req-4: identificar o número de ocorrências possíveis entre as entidades.</p>
<h3>Roteiro</h3> <ol style="list-style-type: none"> 1 Abertura 2 Fundamentação 3 Discussões 4 Fechamento 	<h3>Discussão 1</h3> <p>Considerando que:</p> <ul style="list-style-type: none"> ► modelos conceituais de BDs mapeiam conceitos e relações de um domínio; ► uma DSL com abordagem textual apresenta um conjunto de sentenças bem definidas por uma sintaxe e semântica própria. <p>É plausível aplicar uma DSL na modelagem conceitual de BDs relacionais?</p>	<h3>Discussão 2</h3> <p>Considerando que:</p> <ul style="list-style-type: none"> ► sim, é plausível aplicar uma DSL na modelagem conceitual de BDs relacionais; ► modelos conceituais de BDs relacionais devem especificar: <ul style="list-style-type: none"> ► as entidades do domínio; ► os atributos das entidades; ► os relacionamentos entre as entidades; ► as cardinalidades dos relacionamentos. <p>Como deve ser a gramática aplicada para a modelagem conceitual de BDs relacionais?</p>	<h3>Discussão 2</h3> <p>Exemplo de Diagrama Entidade-Relacionamento</p> 
<h3>Discussão 3</h3> <p>Considerando:</p> <ul style="list-style-type: none"> ► as versões da gramática de modelagem proposta (ver impresso). <p>Qual gramática para a DSL de modelagem proposta é mais adequada para modelagem conceitual de BDs relacionais?</p> 	<h3>Roteiro</h3> <ol style="list-style-type: none"> 1 Abertura 2 Fundamentação 3 Discussões 4 Fechamento 	<h3>Contribuições Finais</h3> <p>Fiquem à vontade para contribuir.</p> <p>Obrigado por participar!</p>	 <p>Uma Linguagem Específica de Domínio para a Representação de Modelos Conceituais de Bancos de Dados Relacionais</p> <p>Jonathan Riquelmo Prof. Dr. Maicon Bernardino Prof. Dr. Fábio Basso Universidade Federal do Pampa Laboratory of Empirical Studies in Software Engineering (LESSE)</p> <p>2019/2</p>

B.5 Discussão 1



Discussão 1

Nome: _____
Local: _____ Data: _____

1. Linguagens específicas de domínio com abordagem textual podem ser aplicadas na modelagem conceitual posto que conseguem descrever de forma rápida e concisa determinadas propriedades. Sendo assim, essas soluções podem ser utilizadas ou mesmo adaptadas de uma forma efetiva no que diz respeito a representação do domínio que modelam.

- 1 - Discordo totalmente.
- 2 - Discordo parcialmente.
- 3 - Indiferente.
- 4 - Concordo parcialmente.
- 5 - Concordo totalmente.

Justifique sua resposta.

B.6 Discussão 2



Times New Roman

Discussão 2

Nome: _____

Local: _____ Data: _____

1. Considerando que um modelo conceitual de banco de dados deve definir ao menos as entidades de domínio, seus atributos e o número de ocorrências (cardinalidade) possíveis de associações (relacionamento), como você definiria uma gramática básica (DSL) para a sua representação?

B.7 Discussão 3



Times New Roman

Discussão 3

Nome: _____
Local: _____ Data: _____

1. Levando em consideração os modelos apresentados, na sua visão qual é o mais adequado para a modelagem conceitual de bancos de dados relacionais?

- Modelo 1.
 Modelo 2.

Justifique sua resposta indicando críticas e sugestões para cada um dos modelos.

Modelo 1

Modelo 2

B.8 Modelos da DSL

Exemplo de uso da Gramática DSL - Modelo 1

```

Domain University;

Entities{
  Person{
    PID: int isIdentifier,
    Name: string
  }
  Teacher isA Person{
    Phone: int,
    Salary: money
  }
  Student isA Person{
    Course: string
  }
  OutsrcEmployee isA Person{
    OutsourcedEID: int,
    Company: string
  }
  Class{
    ClassID: int isIdentifier,
    Course: string,
    Semester: string
  }
  Classroom {
    ClassroomID: int isIdentifier,
    Capacity: int
  }
};

Relationships{
  [many Student isRelatedWith many Class]
  TeacherClass [many Teacher isRelatedWith many Class]
  ClassSchedule [many TeacherClass isRelatedWith many Classroom]
  {ClassScheduleID: int, DayOfWeek: datetime, Discipline: string}
  Supervisor [zero one OutsrcEmployee isRelatedWith many OutsrcEmployee]
};

```

Exemplo de uso da Gramática DSL - Modelo 2

```

Domain University

Entities{
  Person{
    * PID int,
    Name string
  }
  Teacher is Person{
    Phone int,
    Salary money
  }
  Student is Person{
    Course string
  }
  OutsrcEmployee is Person{
    OutsourcedEID int,
    Company string
  }
  Class{
    * ClassID int,
    Course string,
    Semester string
  }
  Classroom {
    * ClassroomID int,
    Capacity int
  }
}

Relationships{
  [(1,N) Student relates (1,N) Class]
  TeacherClass [(1,N) Teacher relates (1,N) Class]
  ClassSchedule [(1,N) TeacherClass relates (1,N) Classroom]
  {ClassScheduleID int, DayOfWeek datetime, Discipline string}
  Supervisor [(0,1) OutsrcEmployee relates (1,N) OutsrcEmployee]
}

```

APÊNDICE C – EXPERIMENTO CONTROLADO

C.1 Termo de Consentimento



Termo de Consentimento Livre e Esclarecido

Você está sendo convidado(a) para participar, como voluntário(a), de um experimento para uma linguagem para modelagem conceitual de bancos de dados relacionais. Este estudo é parte integrante da pesquisa chamada “Uma Linguagem de Domínio Específico para a Representação de Modelos Conceituais de Bancos de Dados Relacionais”, realizada pelo aluno Jonnathan Riquelmo Lopes Frescura e coordenada pelo Prof. Dr. Maicon Bernardino da Silveira e Prof. Dr. Fábio Paulo Basso.

Por meio deste documento, a qualquer momento, você poderá solicitar esclarecimentos adicionais sobre o estudo ou sobre a pesquisa. Também poderá retirar seu consentimento ou interromper a participação a qualquer momento, sem sofrer qualquer tipo de penalidade ou prejuízo. Ao participar deste estudo você não terá nenhum custo e nem receberá qualquer vantagem financeira. Seus dados pessoais serão mantidos em sigilo. Os resultados deste estudo serão armazenados pelo pesquisador responsável e poderão ser divulgados em publicações científicas.

Para participar deste estudo, você precisará: ouvir as explicações, ler o material fornecido, executar as atividades solicitadas e reportar sugestões de melhorias e críticas. Ao participar deste estudo você corre o risco de frustrar-se por não conseguir realizar as atividades solicitadas. Esperamos que os resultados deste estudo contribuam para a minimização da complexidade da modelagem conceitual de bancos de dados relacionais.

Após esses esclarecimentos, solicitamos o seu consentimento de forma livre para que participe deste estudo. Para tanto, preencha os itens a seguir.

Eu, _____, tendo em vista as informações acima apresentadas, de forma livre e esclarecida, aceito participar deste estudo.

Alegrete, ____ de setembro de 2019.

Assinatura do Participante

C.2 Glossário de Termos



Glossário de Termos

Linguagem de Domínio Específico: É uma linguagem de programação ou linguagem de especificação executável que oferece, por meio de notações e abstrações apropriadas, poder expressivo focado e restrito a um domínio de problema específico [?, ?].

Modelo Conceitual: É a descrição de um banco de dados de forma independente da implementação usada em sistemas de gerenciamento de banco de dados (SGBDs) [?].

Abordagem ER: A abordagem entidade-relacionamento é a técnica [?] mais difundida para criação de modelos conceituais de bancos de dados. Esta abordagem foi tão bem aceita que passou a ser considerada uma referência definitiva para a modelagem de banco de dados relacionais. É composta basicamente por um método de diagramação e de conceitos que devem ser respeitados [?].

Entidade: É uma representação de um conjunto de objetos do domínio modelado [?].

Atributo: São as características próprias das entidades [?].

Relacionamento: É a representação da associação entre os objetos modelados [?].

Cardinalidade: Registra o número de ocorrências, possuindo um número mínimo e máximo, com que as entidades podem se associar [?].

Generalização/Especialização: Associado a este conceito está a ideia de herança de propriedades, ou seja, a atribuição de propriedades particulares a um subconjunto de ocorrências de uma entidade genérica [?].

Referências

[1] M. Shell. (2007) IEEETran homepage. [Online]. Available: <http://www.michaelshell.org/tex/ieetran/>

C.3 Questionário de Perfil



Questionário de Perfil

Nome: _____ Matrícula: _____
Curso: _____ Data: _____

Este questionário tem como objetivo verificar seu nível de conhecimento em Projeto e Modelagem de Banco de Dados. Este instrumento não é uma ferramenta avaliativa e os seus dados não serão divulgados. Por favor, responda às seguintes perguntas:

1. Qual a sua experiência com desenvolvimento de software?
 - ≤ 1 ano.
 - > 1 ano e ≤ 3 anos.
 - > 3 anos e ≤ 5 anos.
 - > 5 anos.
2. Como você classifica seu conhecimento em Projeto e Modelagem de Banco de Dados?
 - Nível 1: nunca ouvi falar em Projeto e Modelagem de Banco de Dados.
 - Nível 2: meu conhecimento sobre Projeto e Modelagem de Banco de Dados foi adquirido em disciplinas de graduação e/ou pós-graduação.
 - Nível 3: além de ter aprendido no meio acadêmico, já trabalhei com Projeto e Modelagem de Banco de Dados de forma ativa na área acadêmica ou no mercado de trabalho.
3. Qual a sua experiência com modelagem conceitual de bancos de dados relacionais?
 - Nenhuma.
 - Eu já li sobre.
 - Eu usei só na faculdade.
 - Eu aplico em meus projetos.
 - Eu desenvolvo pesquisa sobre o tema.
4. Com que frequência você usa abordagens (e.g. Modelagem ER) de modelagem conceitual?
 - Eu nunca uso.
 - Eu usei só na faculdade.
 - Eu eventualmente uso em meus projetos.
 - Eu frequentemente uso em meus projetos.
 - Eu sempre uso em meus projetos.
5. Qual o seu contato com linguagens de domínio específico?
 - Eu nunca ouvi falar.
 - Eu já li sobre elas.
 - Eu usei só na faculdade.
 - Eu sou usuário deste tipo de linguagem.
 - Eu desenvolvo este tipo de linguagem.

Se utiliza ou já utilizou alguma, qual(is)?

C.4 Instruções BrModelo



Instruções (brModelo)

1. Acesse a pasta **brModelo2.0**.
2. Execute o arquivo **brModelo.exe**.
3. Anote na folha do instrumento que está sendo feito o **horário de início** da atividade.
4. Com a interface aberta, comece a realizar a modelagem.
 - (a) Na **lateral esquerda** da tela existe uma **paleta de elementos**.
 - (b) Entidades são representadas por retângulos, relações por losangos e assim por diante.
 - (c) **Dica:** em caso de dúvida, basta passar o cursor sobre os elementos para ter uma breve descrição exibida.
 - (d) Na **lateral direita** está um menu de propriedades que é modificado conforme o elemento selecionado. Utilize-o quando for necessário editar características *e.g.* atributos e cardinalidades.
5. Ao término da sua atividade de modelagem vá até o menu superior e acesse **Sistema**;
6. Selecione a opção **Salvar Como...**
7. Selecione a pasta **brModelo2.0** para armazenar o arquivo;
 - (a) Por favor, siga o formato descrito para identificar o modelo **<SeuPrimeiroNome><Matrícula>.brM**
e.g. **Jonnathan151150231.brM**
8. Feche a aplicação.
9. Anote na folha do instrumento que está sendo feito o **horário de fim** da atividade.

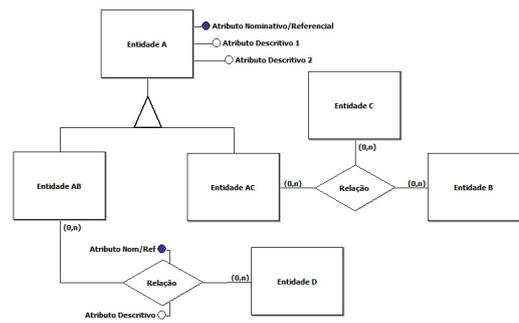


Figura 1: Exemplo de Modelo produzido no BrModelo

C.5 Instruções ERDSL



Instruções (ERDSL)

1. Acesse a pasta **eclipse**.
2. Execute o arquivo **eclipse.exe**.
3. Anote na folha do instrumento que está sendo feito o **horário de início** da atividade.
4. Na tela de seleção de diretório como área de trabalho, clique em **browse**.
5. Vá até o diretório onde está a pasta **eclipse** e procure por pela pasta **eclipse_workspace**.
6. Entre nesta pasta e clique no botão **Selecionar Pasta**.
7. De volta a tela de seleção anterior, clique em **Launch**.
8. Com a aplicação aberta, feche a aba de boas vindas.
9. Vá no menu superior e acesse **File > New > Java Project**.
10. Dê o nome de sua preferência ao projeto e clique em **Finish**
11. Vá até a aba esquerda **Package Explorer**, e clique com o botão esquerdo sobre a pasta **src**.
12. No menu de contexto que surge, vá até **New > File**.
13. Por favor, identifique o arquivo no formato `<SeuPrimeiroNome><Matricula>.erdsl`
e.g. Jonnathan151150231.erdsl
14. No arquivo criado, vá com o cursor e clique com o botão direito e selecione **Open With > ErDsl Editor**.
15. Com o editor ativo, comece a realizar a modelagem.
16. Ao término da sua atividade de modelagem vá até o menu superior e acesse **File > Save All**;
17. Feche a aplicação.
18. Anote na folha do instrumento que está sendo feito o **horário de fim** da atividade.



```
Domain <Identificador> ;
Entities {
  <nomeEntidade1> {
    <nomeAtributo1> isIdentifier,
    <nomeAtributo2>
  }
  <nomeEntidade2> is <nomeEntidade1> {
    <nomeAtributo3>
  };
}
Relationships {
  <relacao1> [(0:1) <entidade> relates (0:1) <entidade>]
  [(1:1) <entidade> relates (0:N) <entidade>]
  <relacao3> [(1:N) <entidade> relates (1:N) <entidade>]
  {<AtribRelacao1>, <AtribRelacao2>
};
```

Modelo da Gramática

C.6 Instrumento 1



Instrumento 1

Nome: _____ Data: _____
Curso: _____ Hora Início: _____
Matrícula: _____ Hora Fim: _____

1. O objetivo é desenvolver um modelo de dados para um hospital veterinário. Clientes e veterinários são pessoas que precisam ter seus dados persistidos. Os clientes têm informações de contato como telefone e e-mail. Os veterinários tem o registro de seu ingresso na clínica, bem como seu salário. Cada cliente pode possuir um ou vários animais, e estes podem estar tratamento. Para a realização de cada tratamento é necessário saber a idade e o porte do animal. O tratamento tem uma duração prevista, um valor específico e envolve também o uso de remédios que são anotados nas receitas. Cada animal pode estar sofrendo de uma ou várias enfermidades, com os mais variados nomes e possíveis causas. As ocorrências mais simples de tratamentos são resolvidas, geralmente, por um único veterinário, entretanto, podem ocorrer situações em que um animal é atendido por mais de um veterinário. Eventualmente, um veterinário pode ter ainda um assistente que, por via de regra, é um companheiro de profissão.

C.7 Instrumento 2



Instrumento 2

Nome: _____ Data: _____
Curso: _____ Hora Início: _____
Matrícula: _____ Hora Fim: _____

1. O objetivo é desenvolver um modelo de dados para uma empresa genérica. Essa empresa é composta de diversos departamentos de vários tipos e com diferentes características, tendo empregados que trabalham diretamente lotados nos departamentos. Os empregados são diferenciados por aspectos inerentes a sua área de atuação (e.g. área de atuação e número do CREA para engenheiros, registro CRC para contadores, número da CNH e veículo para Motorista, etc). Empregados podem ainda supervisionar qualquer outro de seus pares. Neste ambiente são realizados diversos projetos, que são controlados por um departamento. Os projetos possuem descrição e possivelmente outros dados sensíveis. Os projetos são desenvolvidos por vários empregados. Além disso, os projetos necessitam de materiais, que devem possuir uma descrição detalhada e, se for o caso, o registro de validade. Os materiais utilizados pelos projetos são provenientes de fornecedores, que tem cnpj, nome fantasia e telefone de contato. É importante que cada fornecimento possua uma data de realização e um valor associado, mantendo assim o histórico das compras.

C.9 Instrumento 4



Instrumento 4 (Avaliação)

Nome: _____ Data: _____
Curso: _____ Matrícula: _____

Na solução proposta ERDSL, utilizando uma abordagem textual, é possível representar explicitamente/implicitamente oito dos construtores previstos na modelagem ER. Em relação a representação desses construtores, avalie:

- | | Discorda | Concorda |
|--|--------------------------|--------------------------|
| 1. As entidades podem ser representadas adequadamente. | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. Os atributos referenciais/nominativos podem ser representados adequadamente. | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. Os atributos descritivos podem ser representados adequadamente. | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. As relações binárias podem ser representadas adequadamente. | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. As relações ternárias podem ser representadas adequadamente. | <input type="checkbox"/> | <input type="checkbox"/> |
| 6. O autorelacionamento pode ser representado adequadamente. | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. As cardinalidades podem ser representadas adequadamente. | <input type="checkbox"/> | <input type="checkbox"/> |
| 8. A generalização/especialização pode ser representada adequadamente. | <input type="checkbox"/> | <input type="checkbox"/> |

ÍNDICE

- AST, 68, 70
- BD, 25–27, 33–35, 37–39, 49, 52, 53, 55–60, 64, 65, 67, 83–85, 87, 91, 103
- BNF, 44, 48, 49
- CE, 53, 54
- CI, 53
- CIM, 42
- CQ, 54, 56
- CSS, 44
- DDL, 39, 64
- DER, 34, 36, 64, 67
- DML, 39, 64
- DOM, 68
- DQL, 39
- DSL, 26, 27, 29, 31–33, 42–47, 49, 52, 53, 55, 57–59, 63–65, 67, 68, 70, 72, 74, 75, 78, 81, 84, 85, 92, 100, 102–104
- DTL, 39
- EER, 63, 65
- EMF, 68
- ER, 15, 34–38, 63, 65, 68, 70, 76, 77, 84, 89, 100–102, 104
- ES, 25, 41, 65, 73, 74
- ESE, 83
- GLC, 33, 47–49
- GPL, 44, 48, 49, 73, 80
- IDE, 46
- IHC, 58
- LN, 47
- LW, 45–47, 49, 68, 103
- MCD, 34, 38
- MDA, 41–43
- MDD, 41, 42
- MDE, 33, 41–43, 49
- MDSE, 41
- MFD, 34, 38
- MLD, 34, 37, 38
- MLM, 51, 52, 56, 65, 103, 104
- OMG, 41
- PIM, 42
- PLN, 47
- CIM, 42
- QP, 51, 54, 84, 85, 89, 93, 95, 98, 100, 101
- RCP, 31, 77, 78, 81
- SGBD, 33, 34, 37–40
- SLM, 51–54, 56, 104
- SQL, 33, 39, 40, 43, 59, 64, 78, 104
- SysML, 44
- TCC, 27, 29–31, 103
- TCLE, 74, 75, 88, 92
- UML, 44
- VHDL, 44
- XML, 44