

# Avaliação Preliminar de uma Linguagem para a Representação Textual de Modelos Conceituais de Bancos de Dados Relacionais

Jonnathan Riquelmo<sup>1</sup>, Maicon Bernardino<sup>1</sup>,  
Fábio Paulo Basso<sup>1</sup>, Elder Macedo Rodrigues<sup>1</sup>

<sup>1</sup>Laboratory of Empirical Studies in Software Engineering (LESSE),  
Engenharia de Software, Universidade Federal do Pampa (UNIPAMPA)  
Av. Tiarajú, 810 - Bairro Ibirapuitã - Alegrete, RS

{jonnathan.riquelmo, fabiopbasso, eldermr}@gmail.com, bernardino@acm.org

**Abstract.** *The selection of a conceptual database modeling approach (DBs), among other things, depends on the domain of the problem, knowledge and preference of the developer. To evaluate a textual language with such scope, this article reports an evaluation study of two initial grammars using the focus group technique. This preliminary assessment was carried out with thirteen (13) participants, providing important feedbacks for the evolution of grammar. Subsequently, information about the developed Eclipse plugin is presented, including modeling and model transformation resources.*

**Resumo.** *A seleção de uma abordagem de modelagem conceitual de bancos de dados (BDs), entre outras coisas, depende do domínio do problema, do conhecimento e da preferência do desenvolvedor. Para avaliar uma linguagem textual com tal escopo, este artigo relata um estudo de avaliação de duas gramáticas iniciais usando a técnica de grupo focal. Essa avaliação preliminar foi realizada com treze (13) participantes, fornecendo feedbacks importantes para a evolução da gramática. Posteriormente, são apresentadas informações sobre o plugin Eclipse desenvolvido, incluindo recursos de modelagem e transformação de modelo.*

## 1. Introdução

Nas últimas décadas surgiram várias abordagens de desenvolvimento de Banco de Dados (BD) para sistemas de informação. Alguns dos métodos em uso, entre eles o de modelo relacional, tem como base a proposta de modelagem conceitual de dados com abordagem gráfica concebida por [Chen 1976]. Essa é uma das abordagens mais bem aceitas pela comunidade de desenvolvedores ainda nos dias de hoje, chamada de Modelo Entidade-Relacionamento (ER).

Na modelagem ER a construção de um BD é estruturada 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, em que são gerados o modelo conceitual, lógico e o físico [Heuser 2009]. A modelagem ER tem foco no modelo conceitual, cujo os conceitos principais são o de entidades, que são uma representação de um conjunto de objetos do domínio modelado, e suas relações.

Posto isso, esse estudo apresenta a continuação da pesquisa evidenciada no trabalho de [Riquelmo et al. 2019], o qual propôs a definição de uma linguagem específica de domínio (*Domain Specific Language - DSL*), com abordagem textual, para a representação de modelos conceituais de BDs relacionais, uma vez que a grande maioria das soluções existentes tem abordagem puramente gráfica. Partindo da premissa de que desenvolvedores e projetistas de software possuem diferentes preferências por abordagens representacionais de modelos de BDs, este estudo contribui para a concepção de uma nova DSL mapeada para necessidades de ensino, para alunos com maior afinidade com linguagens de programação num perfil de desenvolvedor.

Para [van Deursen et al. 2000] e [Fowler 2010] 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. Entre exemplos conhecidos de DSLs estão: **(i)** SQL, para bancos de dados; **(ii)** CSS, para *layout* de páginas *Web*; **(iii)** XML, para codificação de dados; **(iv)** UML, para projeto de software; e **(v)** L<sup>A</sup>T<sub>E</sub>X, para tipografia de documentos.

Este estudo é o relato de uma avaliação preliminar com a utilização de um grupo focal para as duas variações da gramática proposta, exibidas na Figura 1, bem como exposição da estrutura e operação da ferramenta implementada posteriormente.

**Figura 1. Exemplos de Uso das duas Variações da Gramática da DSL.**

```

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]
};

```

```

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]
};

```

Este artigo está organizado da seguinte maneira. A Seção 3 apresenta o processo de planejamento, preparação e execução do grupo focal, bem como a análise dos resultados e suas ameaças à validade. A seguir, ocorre a demonstração da ferramenta construída a partir da proposta de DSL, na Seção 4. Finalmente, as considerações preliminares são discutidas na Seção 5, em que ainda são indicados os planos de trabalhos futuros.

## 2. Trabalhos Relacionados

Anteriormente, após um extenso estudo composto por um Mapeamento Multivocal de Literatura (MLM, do inglês *Multivocal Literature Mapping*)<sup>1</sup>, foram selecionadas ferramen-

<sup>1</sup>DOI: <https://doi.org/10.5281/zenodo.3950586>

tas que mais se aproximam da DSL proposta, totalizando quatro (4) soluções. Contudo, este estudo tem também como cerne uma avaliação utilizando um grupo focal. Sendo assim, realizou-se uma investigação *ad-hoc* buscando estudos que relatassem avaliações com grupos focais no contexto de DSLs.

O trabalho de Poltronieri [Poltronieri et al. 2018] apresenta um *framework* de avaliação de usabilidade para DSLs. Este estudo relata um grupo focal para avaliação da proposta feita. Mesmo não sendo a avaliação de uma DSL propriamente dita, mas sim uma abordagem para avaliação de DSLs, o uso da mesma metodologia de avaliação o torna um trabalho interessante para análise. Esta avaliação contou com sete (7) participantes, entre eles especialistas em Engenharia de Software, Interação Humano-Computador e Teste de Desempenho.

Houve ainda previamente um teste piloto com apenas dois participantes para verificar a adequação do protocolo criado. Todo o processo é descrito dividido nas fases de planejamento, preparação, moderação e análise de dados. O resultado final confirmou que o *framework* proposto ajuda na avaliação de usabilidade de DSLs.

### 3. Grupo Focal

Esta seção descreve a avaliação preliminar conduzida para analisar as duas alternativas de DSLs propostas, visando assim o seu aperfeiçoamento em uma versão final que seja aderente ao processo ensino-aprendizagem no ambiente acadêmico. Para tanto, foi estabelecido o uso de um grupo focal, um método de pesquisa qualitativa para 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 nessa etapa, expressado na Figura 2, teve como base as diretrizes estabelecidas no trabalho de [Kontio et al. 2008], as quais cobrem a aplicação desse método no contexto da Engenharia de Software.

Figura 2. Processo do Grupo Focal.



#### 3.1. Planejamento

Durante o planejamento foi definido um protocolo que deveria ser seguido. Nesse 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 disponíveis em um repositório<sup>2</sup> público.

<sup>2</sup><https://github.com/JonnathanRiquelmo/Focus-Group-Protocol>

### 3.2. 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 et al. 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, treze (13) participantes colaboraram, todos da área da Engenharia de Software. 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 DSL, 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.

### 3.3. Execução

O grupo focal foi realizado no segundo semestre de 2019, nas dependências de uma Instituição de Ensino Superior (IES), e teve duração 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*<sup>3</sup> previstas em grupos focais.

### 3.4. Análise dos Resultados

Segundo [Kontio et al. 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 desses materiais e das respostas dos participantes para cada instrumento de discussão, foi possível avaliar os resultados do grupo focal executado.

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), sendo um (1) indica discordância total e cinco (5) concordância total:

---

<sup>3</sup>*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.

***“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. 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 3a exibe a distribuição das respostas dos participantes para a primeira discussão. Após, passou-se para a execução da discussão do segundo instrumento. A atividade era composta da seguinte pergunta:

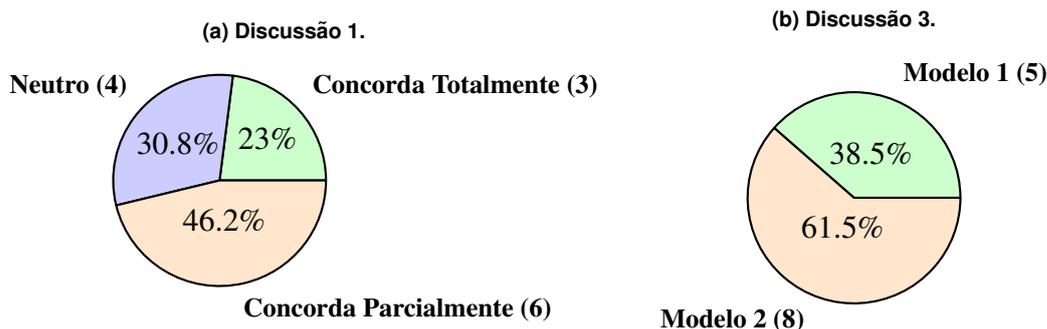
***“Considerando que um modelo conceitual de banco de dados deve definir ao menos as entidades de domínio, seus atributos e o número de ocorrências (cardinalidade) possíveis de associações (relacionamento), como você definiria uma gramática básica (DSL) para a sua representação?”***

Foi informado que os participantes poderiam conversar livremente durante toda a realização deste instrumento. Após cada um sugerir a sua sintaxe, houve debate e troca de informações sobre como estruturar melhor as informações. Neste momento foi possível avaliar que o perfil de alguns acabaria por influenciar no terceiro elemento, pois alguns fizeram sugestões para tornar a linguagem mais sucinta, enquanto outros defenderam que a gramática poderia utilizar mais elementos. A discussão que se seguiu 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 já que alguns participantes acabaram 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 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



**Figura 3. Resultados das Discussões - Grupo Focal.**

negativos observados em cada modelo.

O segundo modelo acabou sendo escolhido pela maioria, como pode ser observado na Figura 3b. 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 as cardinalidades, eram os mais adequados para a aplicação no ensino, indicando assim a necessidade de uma fusão de ambas as versões.

### 3.5. Ameaças à Validade

Para a listagem das possíveis ameaças tiveram como base os tópicos levantados no trabalho de Wohlin [Wohlin et al. 2012].

**Validade do Construto** - (i) *Explicação pré-operacional inadequada*: Esta ameaça está relacionada com o fato do grupo focal não ter o objetivo dos artefatos suficientemente definidos antes de serem traduzidos em medidas ou tratamentos. (ii) *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. Todos os sujeitos fizeram apenas esse grupo focal na época de sua realização.

**Validade Interna** - (i) *História*: Há o risco de algum período temporal específico ter influência na realização do experimento. Para mitigar esta ameaça, e em razão do grupo focal ser realizado em ambiente acadêmico, todo o processo foi executado mediante aviso dos participantes quanto a um período em que todos não estivessem necessariamente sobrecarregados com atividades acadêmicas *e.g.* provas e trabalhos. (ii) *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 cada participante. (iii) *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.

**Validade Externa** - (i) *Sujeitos*: Os sujeitos selecionados para o grupo focal

podem não representar um grupo significativo para a área de estudo. Buscando tentar mitigar essa ameaça foi realizado com participantes da área de Engenharia de Software e Ciência da Computação, e logo, inseridos no contexto de uso da modelagem conceitual de BD relacionais. Porém, o fato da amostra ser de apenas um grupo focal é uma ameaça indicada na literatura. Não foi possível mitigar este fato. (ii) **Interação dos sujeitos com os artefatos de avaliação**: Essa é a ameaça relacionada a aplicação dos artefatos de avaliação do grupo focal com os sujeitos. Dependendo do momento isto pode afetar os resultados. Se, por exemplo, um questionário for realizado alguns dias após a execução do grupo focal, as pessoas tendem a responder de maneira diferente do que fariam momentos após as atividades. Todos os instrumentos foram realizados na mesma sessão.

**Validade da Conclusão - (i) Confiabilidade dos resultados**: A confiabilidade dos resultados obtidos tem impacto direto na validade do grupo focal como um todo. Por ser uma avaliação com maior foco qualitativo, esta ameaça não pôde ser mitigada em razão da subjetividade inerente às resposta dos sujeitos em tais avaliações.

## 4. A Ferramenta

Esta seção apresenta a ferramenta Eclipse com o *plugin* final da solução proposta. Acabou se optando inicialmente por um *plugin* por conta de sua integração com a plataforma Eclipse, bem como pela vantagem de poder ser insumo para um aplicativo independente (*standalone application*).

A principal 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. Em ambos os casos é possível que haja a distribuição como uma ferramenta livre desde que seguidas as diretrizes da licença de software EPL-2.0<sup>4</sup>.

A arquitetura da solução é exposta na Figura 4 e foi construída com o auxílio do Xtext, um *framework* de código aberto para o desenvolvimento linguagens de programação textuais. Nele, a partir da gramática resultante do grupo focal conduzido, gerou-se a maior parte da infraestrutura do editor, do *parser* e modelo Ecore. Para entendimento, o Ecore é uma representação na memória em tempo de execução do modelo criado utilizando a DSL. Após, ocorreram então alguns ajustes manuais necessários e o acréscimo da escrita de um gerador para a conversão dos modelos conceituais para os modelos lógicos.

### 4.1. Operação

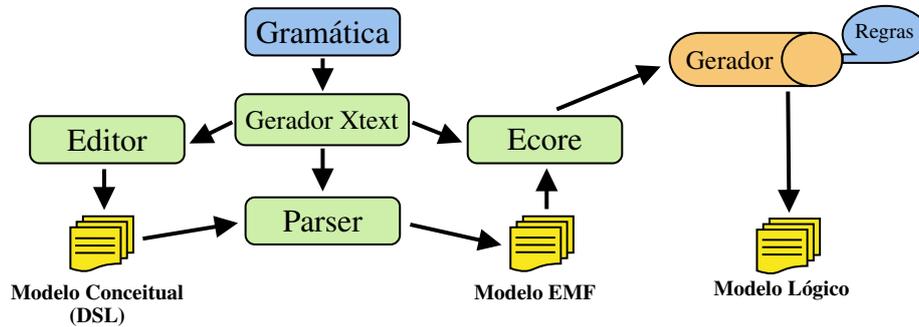
Como dito anteriormente, a versão final da gramática resultou da fusão entre as duas versões previamente definidas e avaliadas com a execução do grupo focal. As principais mudanças foram as modificações das palavras reservadas *isA* e *isRelatedWith* por *is* e *relates*, respectivamente, além da adoção da convenção (0:1) (1:1) (0:N) (1:N) para as cardinalidades.

A Figura 5 apresenta a ferramenta em funcionamento, após o *plugin* ser integrado no Eclipse, onde é possível ser feita a criação de arquivos de modelagem utilizando a

---

<sup>4</sup><https://www.eclipse.org/legal/epl-2.0/>

Figura 4. Arquitetura geral da ferramenta.



DSL. Nesse 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 com base nos dois blocos previstos na gramática (entidades e relacionamentos), *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. Sabe-se que a definição dos tipos de dados não é prevista no modelo conceitual clássico mas, por questões relacionadas a pretensão futura de realizar a geração de instruções SQL, foi decidido pela manutenção dessa escolha de projeto.

Figura 5. Fragmento da solução sendo utilizada.

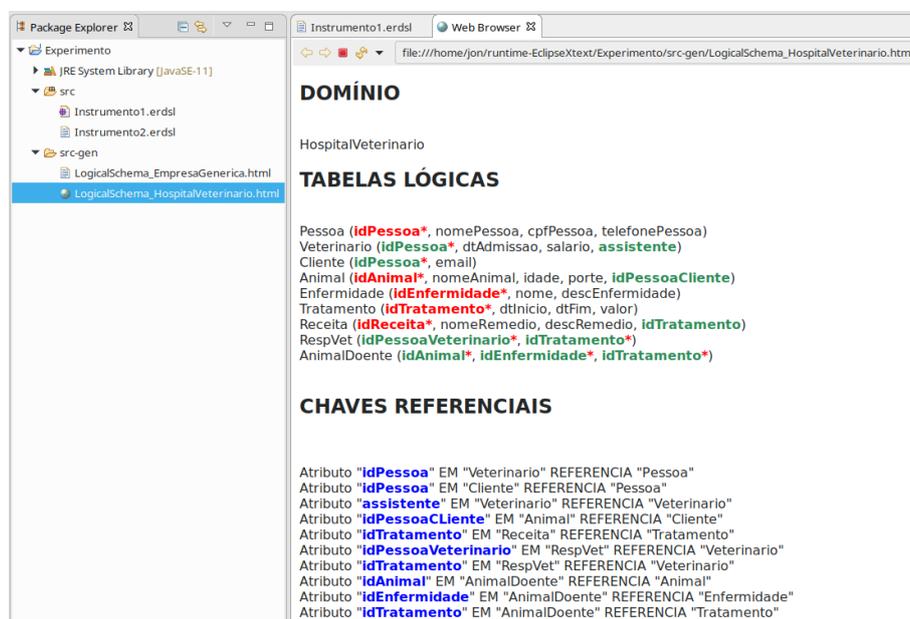
```

Domain HospitalVeterinario;
Entities {
    Pessoa {
        idPessoa int isIdentifier,
        nomePessoa string,
        cpfPessoa int,
        telefonePessoa int
    }
    Veterinario is Pessoa {
        dtAdmissao datettime,
        salario money
    }
    Cliente is Pessoa{
        email string
    }
    Animal {
        idAnimal int isIdentifier,
        nomeAnimal string,
        idade int,
        porte string
    }
    Enfermidade {
        idEnfermidade int isIdentifier,
        nome string,
        descEnfermidade string
    }
    Tratamento {
        idTratamento int isIdentifier,
        dtInicio datettime,
        dtFim datettime,
        valor money
    }
    Receita {
        idReceita int isIdentifier,
        nomeRemedio string,
        descRemedio string
    }
};
Relationships {
    Dono [(1:1) Cliente relates (1:N) Animal] // 1:N
    Assistente [(0:N) Veterinario relates (0:1) Veterinario] // N:1
    RespVet [(1:N) Veterinario relates (0:N) Tratamento] // N:N
    AnimalDoente [(0:N) Animal relates (1:N) Enfermidade] // N:N
    Ternario [(1:N) AnimalDoente relates (1:1) Tratamento ] // N:1
    Possui [(1:1) Tratamento relates (1:1) Receita] // 1:1
};
    
```

A função para o mapeamento e geração do modelo lógico é executada automaticamente toda a vez que uma modelo é salvo. Na Figura 5 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 6.

Figura 6. Exemplo de modelo lógico gerado.



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.

É importante salientar que as premissas foram escolhidas previamente com base no livro referência de [Heuser 2009] para o processo de mapeamento e transformação de um modelo conceitual em uma representação lógica do banco de dados. As premissas implementadas até agora podem ser resumidas em: (i) adição de coluna para relacionamentos 1:1; (ii) adição de coluna para relacionamentos 1:N, e; (iii) criação de tabela própria para relacionamentos N:N. Em relação ao conceito de generalização/especialização, também foi necessário assumir uma ideia inicial que teria que ser tomada como verdade.

Segundo [Heuser 2009], para esses casos existem 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 linguagem de propósito geral (*General-purpose Language* - GPL) Xtend, uma linguagem que tem suas raízes no Java.

## 5. Considerações Preliminares

Com a alegação anterior em que afirmou-se que profissionais que trabalham com software possuem diferentes predileções nas abordagens para representação de BDs, este estudo

colaborou para a formulação de uma DSL que possa atender atividades de ensino, visando estudantes com perfis direcionados para as atividades de desenvolvimento.

Os resultados preliminares apontaram para uma aceitação da proposta, sendo que, a partir de dois modelos de gramática analisados, houveram recomendações para a adoção de expressões mais concisas no bloco de entidades e relações, assim como de convenções no que diz respeito as cardinalidades. Dessarte, a construção de um *plugin* capaz de realizar a modelagem e transformação entre modelos conceituais e lógicos foi alcançado e então disponibilizado como software de código aberto <sup>5</sup>.

Este trabalho tem por objetivo contribuir com uma ferramenta que auxilie no processo de projeto e modelagem de BDs utilizando uma DSL textual de fácil uso e compreensão. Pretende-se que a proposta final não apenas realize modelagem, mas sim a transformação dos modelos conceituais gerados em *scripts* SQL para diferentes tecnologias SGBDs, e.g. PostgreSQL, MySQL e SQL Server.

Já estão sendo desenvolvidos os geradores para *scripts* SQL. Pretende-se ainda que seja realizada uma avaliação empírica do produto desenvolvido. Esse experimento, por exemplo, poderá servir para a avaliação da abordagem textual e sua viabilidade frente a abordagem gráfica. Outra possibilidade de evolução futura é a criação de uma versão gráfica desta DSL e sua integração com a atual, tornando a solução final uma ferramenta de abordagem bidirecional (textual e gráfica).

## Referências

- Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. *ACM Trans. Database Syst.*, 1(1):9–36.
- Fowler, M. (2010). *Domain Specific Languages*. Addison-Wesley Professional, London, England, 1st edition.
- Heuser, C. A. (2009). *Projeto de Banco de Dados*. Bookman, Porto Alegre, BR.
- Kontio, J., Bragge, J., and Lehtola, L. (2008). *The Focus Group Method as an Empirical Tool in Software Engineering*, pages 93–116. Springer London, London.
- Poltronieri, I., Zorzo, A. F., Bernardino, M., and de Borba Campos, M. (2018). Usa-dsl: usability evaluation framework for domain-specific languages. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 2013–2021.
- Riquelmo, J., Bernardino, M., Basso, F. P., and Rodrigues, E. M. (2019). Uma linguagem específica de domínio para a representação de modelos conceituais de bancos de dados relacionais. In *Anais da III Escola Regional de Engenharia de Software*, pages 89–96, Porto Alegre, RS, Brasil. SBC.
- van Deursen, A., Klint, P., and Visser, J. (2000). Domain-specific languages: An annotated bibliography. *SIGPLAN Not.*, 35(6):26–36.
- Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B., and Wessln, A. (2012). *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, London, England.

---

<sup>5</sup>Repositório: <https://github.com/ProjetoDSL/ERDSL>