FEDERAL UNIVERSITY OF PAMPA

Jonnathan Riquelmo Lopes Frescura

ERtext: an approach for designing relational databases based on a textual approach for conceptual modeling

Alegrete
2022

Jonnathan Riquelmo Lopes Frescura

ERtext: an approach for designing relational databases based on a textual approach for conceptual modeling

Master's thesis presented as partial requirement for obtaining the degree of Masters of Software Engineering at Federal University of Pampa (Unipampa).

Supervisor: Prof. PhD. Maicon Bernardino da Silveira

Co-supervisor: Prof. PhD. Fábio Paulo Basso

Alegrete
2022

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

F884e Frescura, Jonnathan Riquelmo Lopes
ERtext: an approach for designing relational databases based on a textual approach for conceptual modeling / Jonnathan Riquelmo Lopes Frescura.
135 p.

Dissertação(Mestrado)-- Universidade Federal do Pampa, MESTRADO EM ENGENHARIA DE SOFTWARE, 2022.
"Orientação: Maicon Bernardino da Silveira".

1. Database Design and Modeling. 2. Conceptual Modeling. 3. Domain-Specic Language. I. Título.

JONNATHAN RIQUELMO LOPES FRESCURA

ERtext: an approach for designing relational databases based on a textual approach for conceptual modeling

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Mestre em Engenharia de Software.

Dissertação defendida e aprovada em: 29 de abril de 2022.

Banca examinadora:

Prof. Dr. Maicon Bernardino da Silveira

Orientador

UNIPAMPA

Prof. Dr. Fábio Paulo Basso

Coorientador

UNIPAMPA

Prof. Dr. Ronaldo dos Santos Mello

UFSC

Prof. Dr. Juliano Varella de Carvalho

FEEVALE

Assinado eletronicamente por **FABIO PAULO BASSO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 04/05/2022, às 10:52, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **JULIANO VARELLA DE CARVALHO**, **Usuário Externo**, em 04/05/2022, às 11:12, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **Ronaldo dos Santos Mello**, **Usuário Externo**, em 04/05/2022, às 15:38, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0801348** e o código CRC **86185360**.

ACKNOWLEDGEMENTS

A realização deste mestrado foi certamente uma experiência de mudança de vida para mim e não teria sido possível sem o apoio e a orientação que recebi de muitas pessoas.

Primeiramente, gostaria de expressar a mais profunda gratidão ao meus pais, Zilma e Aroldo. Vocês apoiaram e incentivaram de forma incondicional como em todas as etapas da minha vida. Sem isso eu não seria capaz de alcançar mais essa conquista.

Agradeço todos os ensinamentos recebidos dos meus orientadores Bernardino e Basso. Vocês me inspiram como profissional e pesquisador.

Ao Programa de Pós-Graduação em Engenharia de Software (PPGES) e seu corpo docente de excelência. Sou extremamente grato pela oportunidade de continuar minha formação em uma universidade pública como a UNIPAMPA, gratuita e de tanta qualidade.

E finalmente, agradeço à Dani, *aka* morena aquariana que é minha definição de serendipidade há quase 5 anos. Sem sua compreensão e conselhos nos meus piores momentos tudo seria ainda mais díficil. Obrigado por viver todos os dias em meus pensamentos e palavras.

"Any fool can write code that a computer can understand.
Good programmers write code that humans can understand."
Martin Fowler

RESUMO

Os bancos de dados tornaram-se elementos vitais na sociedade moderna com o avanço da tecnologia. Bancos de dados são conjuntos de dados armazenados para dar sentido a um domínio específico. A informação armazenada é considerada um ativo de grande relevância nas organizações contemporâneas. Como a qualificação profissional exige a assimilação de conceitos fundamentais, incluindo modelagem conceitual, lógica e física de banco de dados, alternativas de ferramentas podem ser exploradas nas disciplinas de projeto e modelagem de banco de dados. As instituições de ensino superior devem prestar atenção às abordagens que melhor mapeiam as necessidades dos alunos de acordo com suas preferências para facilitar o aprendizado. No entanto, a variedade de tecnologias que se tornaram disponíveis nos últimos anos, com a grande maioria focada em abordagens gráficas, dificulta a escolha de ferramentas para modelagem entidade-relacionamento (ER) na indústria e, consequentemente, na academia. Assim, visando contribuir com uma alternativa relevante de código aberto, este estudo propôs dar continuidade a um projeto de linguagem de modelagem. O objetivo foi implementar uma ferramenta baseada em uma linguagem textual de domínio específico (DSL) para apoiar o processo de ensino-aprendizagem de modelagem conceitual de banco de dados. As DSLs permitem especificar e modelar domínios de forma mais rápida e produtiva. Elas se diferenciam das linguagens de uso geral pois sua expressividade é limitada para domínios específicos. Nesse sentido, realizamos uma investigação do estado da arte sobre os mecanismos de transformação de modelos de banco de dados. Esta investigação utilizou um protocolo de mapeamento sistemático da literatura, e compreendeu um conjunto final de 12 estudos primários que relatam as proposições das DSLs e os mecanismos que utilizam. Após estabelecer um protótipo da proposta, planejamos e realizamos uma avaliação empírica preliminar com um grupo composto por trinta e três (33) participantes. Com o feedback recebido no processo, chegamos a um arcabouço de conhecimento que suporta a evolução da proposta em uma versão estável da ferramenta, chamada ERtext. Assim, realizamos mais duas avaliações empíricas (25 e 15 participantes, respectivamente) semelhantes à avaliação preliminar. A intenção foi verificar o esforço (tempo), acurácia, *recall*, medida F, utilidade percebida e facilidade de uso de ambas as abordagens: uma ferramenta com base em texto (ERtext) comparada à uma ferramenta gráfica (brModelo). Além disso, avaliamos os artefatos gerados, bem como as novas funcionalidades adicionadas que ainda não haviam sido avaliadas. No final, os três experimentos controlados combinados tiveram setenta e três (73) participantes. Realizamos testes estatísticos detalhados de hipótese, bem como análises qualitativas. Em conclusão, no geral os resultados indicaram que não existem diferenças significativas *de facto* entre as abordagens, demonstrando assim que nossa proposta textual é viável por equiparar-se à já reconhecida e amplamente utilizada abordagem gráfica .

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

ABSTRACT

Databases have become vital elements in modern society with the advance in technology. Databases are sets of data stored to make sense of a specific domain. The stored information is considered a highly relevant asset in contemporary organizations. Since professional qualification requires the assimilation of fundamental concepts for database development, including conceptual, logical, and physical database modeling, tooling alternatives can explore in database design disciplines. Higher education institutions must pay attention to those approaches that best map student needs according to their preferences to facilitate learning from the students. However, the variety of database systems technologies that have become available in recent years, with the vast majority focused on graphical approaches, makes it hard to choose tools for entity-relationship (ER) modeling in the industry and, consequently, in academia. Hence, aiming to contribute with a relevant open-source alternative, this study proposed going on with a modeling language project. The goal was to implement a tool based on a textual domain-specific language (DSL) to support the teaching-learning process of conceptual database modeling. DSLs allow specifying and modeling domains faster and more productive way. They have been differentiating from the general-purpose languages as are expressiveness limited for particular domains. In this sense, we carried out a state-of-the-art investigation on database model transformation mechanisms. This survey that we have carried out uses a systematic literature mapping protocol. The execution of this protocol comprises a final set of 12 primary studies that report the propositions of DSLs and the mechanisms they use. After establishing a prototype of the proposed tool, we planned and carried out a replication of a preliminary empirical evaluation with a group consisting of thirty-three (33) participants. With the feedback received in the process, we reached a framework of knowledge that supports the evolution of the proposal in a stable version of the tool, called ERtext. Hence, we conducted two more empirical evaluations similar to the replication to evaluate this stable version. The intention was to verify the effort (time), accuracy, recall, F1-score, perceived usefulness, and ease of use of both approaches: a textual-based on our proposal tool compared to a graphical-based on an academic tool. Moreover, we evaluated the generated artifacts, as well as the new features added that had not yet been evaluated. In the end, the three combined controlled experiments had seventy-three (73) participants. We performed in-depth statistical tests of the hypothesis as well as qualitative analysis. Overall, in the end, the results indicated that there are no significant differences *de facto* between the approaches, thus demonstrating that our textual is viable as it equates to the recognized and widely used graphical one.

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

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AST Abstract Syntax Tree

BNF Backus-Naur Form

CASE Computer-Aided Software Engineering

CIM Plataform-Specific Model

CIM Computation-Independent Model

CSS Cascading Style Sheets

DB Database

DBMS Database Management System

DDL Data Definition Language

DML Data Manipulation Language

DOM Document Object Graph

DSL Domain-Specific Language

EBNF Extended Backus-Naur Form

EMF Eclipse Modeling Framework

ER Entity-Relationship

EX1 Experiment 1

EX2 Experiment 2

EX3 Experiment 3

GPL General-Purpose Programming Language

HTML HyperText Markup Language

IDE Integrated Development Environment

LSP Language Server Protocol

LW Language Workbench

MDA Model-Driven Architecture

MDD Model-Driven Development

MDE Model-Driven Engineering

MDSE Model-Driven Software Engineering

OMG *Object Management Group*

ORM Object-relational Mapping

PIM Plataform-Independent Model

RTE Round-trip Engineering

SQL Structured Query Language

SUS System Usability Scale

SysML Systems Modeling Language

UML Unified Modeling Language

VHDL VHSIC Hardware Description Language

XML eXtensible Markup Language

TABLE OF CONTENTS

1 INTRODUCTION

Nowadays, the use of databases (DBs), as well as the volume of manipulated data, increased exponentially. Its use is inserted in several areas of modern society, such as education, communication, infrastructure, healthcare, security, economy, and internet development in general. DBs and their associated technologies are a basis for the use, manipulation, and administration of the main assets of modern organizations: the information. With this established scenario, it is notable that there is a growing duty of the academy to provide a good level of preparation for future professionals who will enter an increasingly demanding industry.

The elementary principle that justifies the use of data modeling techniques for defining efficient storage structures is the ability to manipulate and retrieve data in a simple way. These data represent the minor portion of raw values, which together represent information, *i.e.* another set of already processed data that has a relevant meaning for the information domain.

According to (DATE; WARDEN, 1990), a DB defines as a collection of stored operational data used by the application systems of some organization. For (ELMASRI; NAVATHE, 2011), a DB determines as an abstraction of the real world, also called a mini-world, since it represents aspects that together carry an implicit meaning.

In general, the DB literature indicates three basic steps that compose the modeling and designing process:

- **Conceptual Modeling**: The first step aims at describing the domain at a high level and at detailing how to structure the DB. This phase usually uses the Entity-Relationship (ER) approach or its enhanced model (EER). The ER approach deals in short with objects and their relationships;

- **Logical Modeling**: The second phase maps these objects and the relationships described in the conceptual model to corresponding elements within the implementation data model. Furthermore, there are many models reported in the literature (relational, object-oriented, semi-structured, graph, key-value, etc), the most widespread has been the Relational Model (CODD, 1989; KARANIKOLAS, 2011; PAREDAENS et al., 2012; ROBINSON; WEBBER; EIFREM, 2015; PUANGSAIJAI; PUNTHEERANURAK, 2017);

- **Physical Modeling**: The third phase corresponds in transforming the logical model to DB code, *i.e.* Structured Query Language (SQL). This is a typically automated step that results in a SQL specification with the DB schema to be created in a Database Management System (DBMS).

This entire process classifies as part of Model-Driven Engineering (MDE), a software development approach in which models are the principal input artifacts in the development process. One of the main advantages of this approach is the power provided to express models using simpler concepts, which are independent of implementation details, *i.e.* resulting in a complexity abstraction (BRAMBILLA; CABOT; WIMMER, 2017).

The concepts of the MDE approach allow the creation of modeling tools that support the development of models, their transformation and integration at different stages of the software development process and, in the case of this study, DB modeling.

1.1 Motivation

According to (RASHID; AL-RADHY, 2014; SALGADO; MEDEIROS, 1995) teaching in the DB area is very important in higher education during the training of computer professionals. A teaching-learning process must integrate the theoretical and practical aspects that, gradually, explore abstraction levels independent and dependent from DBMS providers and their technologies. Besides, it is noteworthy that DB instructors and related courses should be concerned with how to teach their students wherefore they meet the needs and expectations due the demands of both the academic and the industry. Therefore, exercise domain analysis and their implementation with specific technologies is a need.

Starting from a premise that there is a growing search for instruments that support the teaching-learning process in academia, which is derived from several teaching classroom experiences, this research seeks to explore an alternative for database tool support scoping the modelling and designing process. Initially, the teaching of DB design and modeling in general includes the presentation of essential theoretical topics of conceptual modeling and model transformation rules. Afterward, there is an introduction to the modeling tools using mainly graphical approaches. Meanwhile, the state of the art misses textual approaches to support this process. Considering this limitation, this research is motivated to offer a software product that adopts a textual approach, with a grammar that is easy to use and understand, thus offering an alternative in the state of the art.

1.2 Objectives

The main goal of this study is to develop a tool for conceptual modeling, using a textual approach, to support the teaching-learning process of DB design modeling in higher education institutions. This tool was based on our first language proposal (LOPES, 2019), improved after preliminary evaluations, resulting in the current version.

Hence, to achieve the proposed main goal, it is essential to divide the problem into the following specific objectives:

1) Investigate technologies that support the process of creating Domain-Specific Languages (DSLs);

2) Define the development support technology and create a basic grammar;

3) Search the literature to find the transformation mechanisms of relational DB models;

4) Integrate the language, and a transformation engine for other technologies, into an open-source tool;

5) Conduct a preliminary evaluation of the proposed solution;

6) Make tool improvements;

7) Conduct a set of final evaluations of the proposed solution;

8) Contribute with a tool that helps in the process of teaching design and conceptual modeling of DBs.

1.3 Rationale

Since the beginning, developers have used text to specify software products. Programming languages increase the level of abstraction in a similar way to models. So, as a logical consequence, this results in textual modeling languages.

Hence, generally, mechanisms process a textual modeling language, which transforms the information expressed in textual format into models. These mechanisms base their execution on the syntactic structure of a textual modeling language, which formalizes in grammar. A grammar defines keywords of a language, the nesting of its elements, and the notation of its properties. That said, it can be inferred that textual models can bring several benefits:

- **Inform a lot of details**: When it comes to elements with numerous properties, the textual approach usually stands out over graphics. We said the same for structures formed by a large of smaller parts, such as mathematical operations or sequences of statements;

- **Increase model cohesion**: A textual model usually specifies elements entirely in one place. While this can be a disadvantage for a high-level view, it can make it easier to find low-level property definitions. In the proposal of this work, we carried out all the conceptual modeling in just one file;

- **Perform quick edits**: It is not necessary to recurrently switch between keyboard and mouse when creating or editing textual templates. Therefore, less time is likely to be spent when formatting textual models than, *e.g.* refining the position, links, or even edges of elements in diagrams;

- **Use of generic editors**: There is not necessarily a demand for a specific tool to create or modify textual models, as is the case for domain-specific languages with this approach. For simple changes, it is possible to use any generic text editor.

However, for larger tasks, it is better to have some support for the modeling language. In the proposal of this work, the integration of the defined language with an Eclipse editor is included, thus providing a high level of writing assistance.

However, in contrast to the graphical approach, textual modeling also has several disadvantages according to Moody's graphic communication theory (MOODY, 2007). Among them are absolute and relative discrimination, *i.e.* the ability to see elements separating them from the background beyond being able to differentiate them within a broad context, respectively. These requirements tend to be harder to control as the level of detail in the models expands. The author also mentions that this directly influences the cognitive limits of understanding and consequent abstraction, which can make it difficult for the user to assimilate the conceptual integration.

1.4 Contributions

Our main contributions are listed below, including but not limited to:

i. The definition of a language (syntax and semantics) for conceptual modeling for relational databases;

ii. The development of a tool that implements the language;

iii. The public and free of charge availability of the tool's solution;

iv. The associated set of knowledge for the elaboration and validation of the final proposal;

v. The availability of a detailed experimental protocol;

vi. The description of how to apply quantitative analysis methods through hypothesis testing;

vii. The description of how to apply qualitative analysis methods;

viii. The availability of the collected datasets in the execution of three (3) controlled experiments, when combined, added up to seventy-three (73) subjects.

1.5 Organization

We organized this document as follows:

- Chapter 2 details our research design;
- Chapter 3 describes the theoretical background of our study, addressing the main concepts related to ER modeling, MDE, DSLs, Language Workbenches (LWs), and the Xtext framework;

- Chapter 4 presents a systematic mapping performed for the investigation of the literature regarding the transformation mechanisms of relational database models;
- Chapter 5 presents the proposed modeling tool;
- Chapter 6 describes the experimental evaluations with the proposed tool;
- Finally, in Chapter 7, we present the conclusion and also our future perspectives.

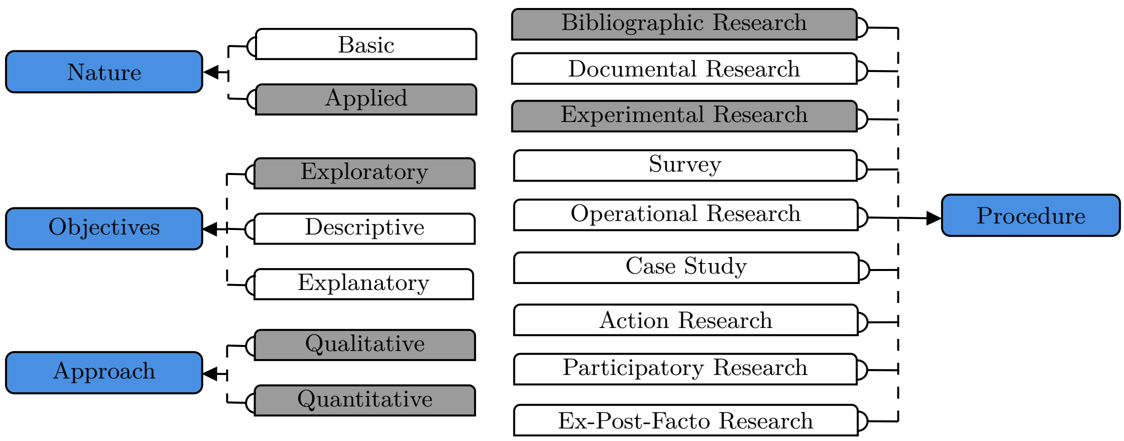
2 METHODOLOGY

This chapter discusses the research methodology adopted to carry out this study. In Section 2.1, we describe the classification of our research. Section 2.2 presents the research design.

2.1 Research Classification

In general, for a study to have greater confidence in terms of scientific rigor, it is essential to identify the necessary activities and techniques that make it possible to reach the proposed objective (PEFFERS et al., 2007; BAIRAGI; MUNOT, 2019). Figure 1 shows this study's classification according to its nature, approach, objectives, and procedures.

Figure 1 – Research classification.



Source – Adapted from (PRODANOV; FREITAS, 2013).

According to (PRODANOV; FREITAS, 2013), no type of research is self-sufficient, so it is necessary to mix different types with more accentuated points to obtain satisfactory results. The chosen methods determine the procedures that should be used, both in analysis and in data and information collection.

In terms of its nature, research can be basic or applied. This study proposes a tool that uses a DSL, generating a practical tool to solve something specific and, therefore, we can categorize it as applied research.

From the point of view of its objectives, research classifies as exploratory, descriptive, or explanatory. This study aims to provide more information on the subject it investigates. Hence, it is explanatory research.

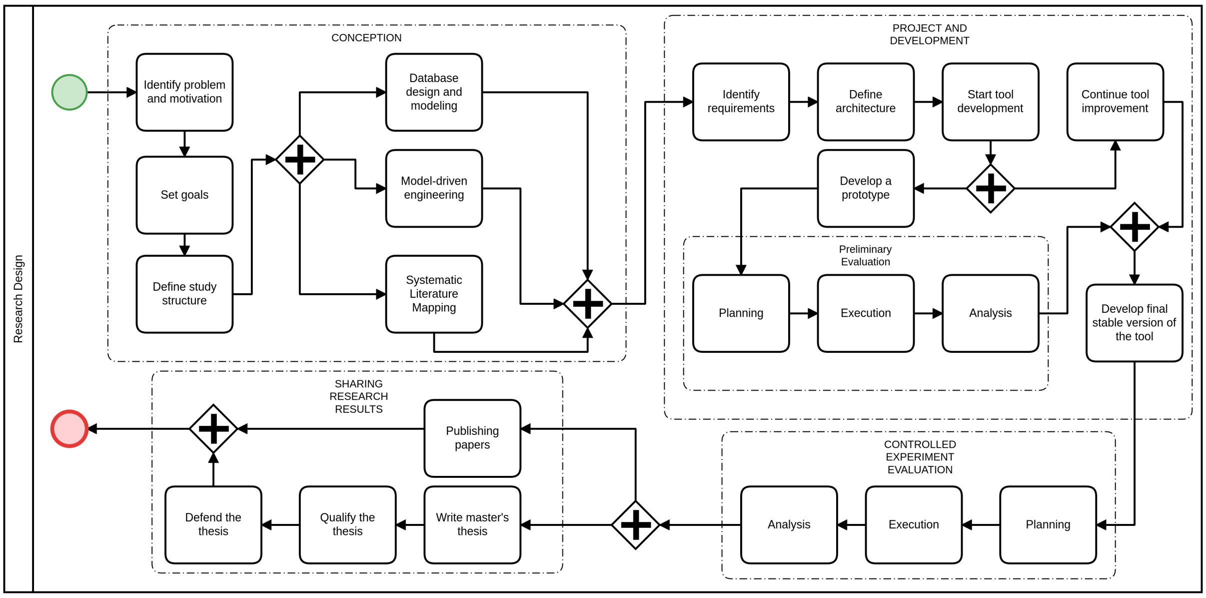
The problem approach can classify the research into quantitative or qualitative. This study applies the concepts of both categories. It occurs from experimental evaluations, both preliminary and those planned and carried out later, where it is necessary to make empirical assessment observations to interpret the participant experiences and also statistical methods to evaluate the models produced.

Finally, technical procedures group research categories as bibliographical, documentary, experimental, survey-type research, also called survey, operational research, case study, *ex-post-facto* research, action research and participant research. Hence, this study performs a bibliographical research, *i.e.* the activities performed to survey its theoretical framework. Besides, we carried out a series of experimental research for the evaluation stage of the proposed tool.

2.2 Research Design

In Figure 2 our research design is presented. We have divided this project into four (4) groups of activities.

Figure 2 – Research design.



Source – Author.

The first group corresponds to the design phase. This phase covers the identification of the problem and motivation, the establishment of objectives and the definition of the study structure. To create a theoretical framework, three activities were carried out in parallel, namely the study of DB design modeling, model-driven engineering and the execution of a systematic literature mapping seeking mechanisms for transforming relational DB models.

The second group covers the design and development stage of the tool. The requirements were identified, the architecture was defined and then the development of the tool started. In parallel, a prototype underwent a preliminary evaluation, described in

Chapter 6. After having analyzed the results from the preliminary evaluation, a stable version of the tool is then generated.

In the third group are the activities corresponding to the last two (2) controlled experiments to which this stable version was submitted, in order to evaluate all the functions developed. This also corresponds to the evaluation of the artifacts produced by the implemented generators. It is important to emphasize that none of these artifacts were initially considered in the first and second evaluations, but only in the last experiment:

- Conceptual model diagrams;
- Logical textual models in HyperText Markup Language (HTML) format;
- Data Definition Language (DDL) templates for specific DBMSs.

Finally, the fourth group corresponds to the dissemination of the results of the developed research. Among the activities, we highlight the writing of the master's thesis and the paper's publishing in scientific events.

3 BACKGROUND

This chapter presents the general concepts used to understand and implement this work in a comprehensive way. Section 3.1 introduces the entity-relationship model as well as the concepts of logical and physical models in Sections 3.2 and 3.3. Section 3.4 introduces Model-Driven Engineering and thus its associated notions. The concepts referring to Domain-Specific Languages are presented in Section 3.5, followed by Language Workbenches in Section 3.6. The Xtext Framework has an overview displayed in Section 3.7, and finally, we discuss the lessons learned in Section 3.8.

3.1 Entity-Relationship Model

The conceptual data model is the description of a DB independently of the implementation used in a DBMS. This model lists what data can occur in the DB but does not record how we stored this data at the DBMS level. The most widespread technique of conceptual data modeling is the ER approach of (CHEN, 1976). This approach was so well accepted that it is now considered a definitive reference for modeling relational DBs. It is primarily composed of a diagramming method and concepts that must be respected.

In the ER approach, the principal concept is the **entity**, representing a set of objects from the modeled domain (HEUSER, 2009). Rectangles symbolize the entities in the diagram. However, in this approach there are also other concepts that are essential and should be analyzed. For a better understanding of the conceptual modeling that uses this approach, Figure 3 presents some graphical representations of the constructors foreseen in the ER model.

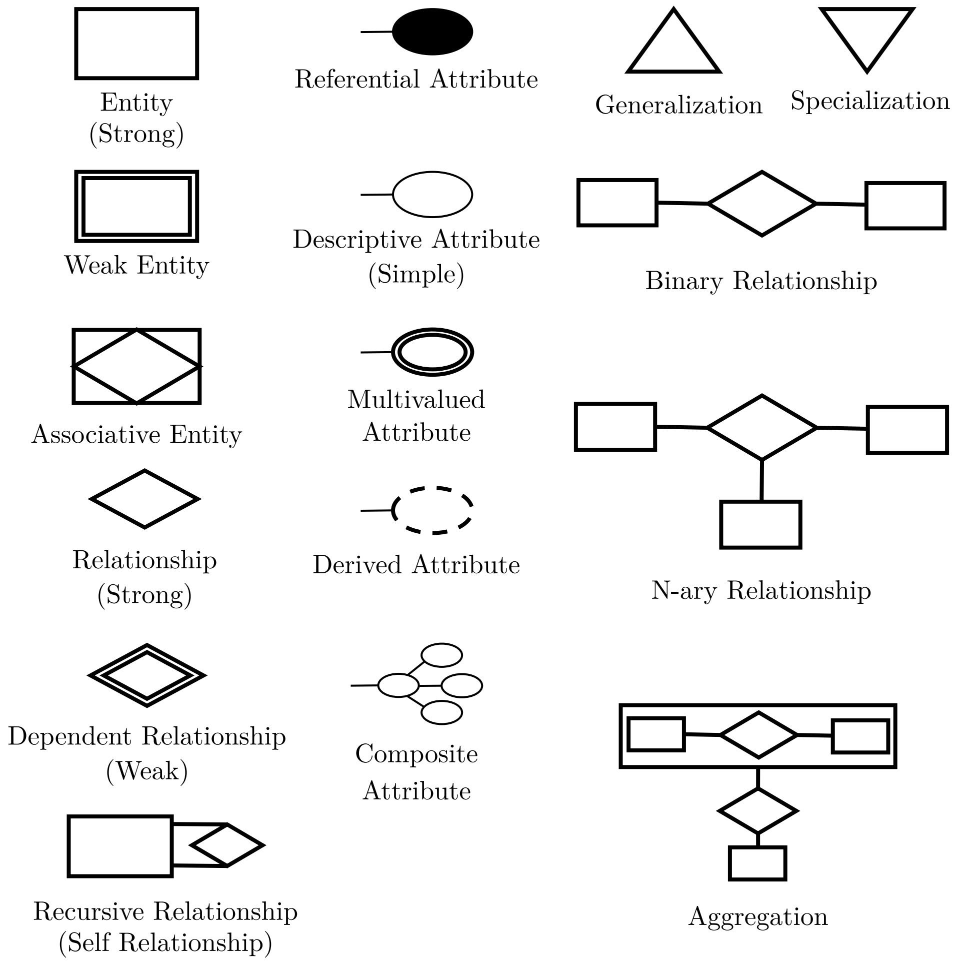
A **strong entity** is an entity that does not depend on the existence of other entity(ies) to exist. Conversely, a **weak entity** is an entity that has its existence or identification dependent on other entity(ies). An **associative entity** serves to give more semantic meaning to the model and to symbolize many-to-many relationships, containing references to columns of tables of the same or other DBs.

The **attributes** can be **referential**, that is, they identify the entity, and **descriptive** that describe it. Most attributes tend to be descriptive and are also called simple or single-valued. However, there are other types of these attributes such as **multi-valued**, which can take different values, **derived**, which are inferred from other attributes, and **composite** attributes, which can be divided into several parts with independent meanings.

Regarding the possible **relationships** between entities, we can mention the **binary** that involve two (2) entities and the **n-ary** that correspond to the link of three (3) or more entities. It is common for relations composed of three entities to be called **ternary** relations. Besides these types, it is still possible to have recursive relationships of an entity with itself, also called **self-relationship**.

The ER approach still supports concepts of **generalization/specialization**, which

Figure 3 – ER Model Constructors.



Source – Author.

define a hierarchical group of entities that share common attributes, and **aggregation** where the existence condition is that the principal relationship needs to be necessarily from many-to-many.

The **relationships** represent the association among objects, and diamonds denote them. The **cardinality** of relationships records the number of occurrences that entities can associate with each other. Two cardinalities must be assigned: the minimum and the maximum. **Attributes** are features represented by small circles connected to entities.

3.2 Logical Model

A logical data model defines a model that represents objects, relationships, and characteristics according to implementation rules. This means that this model has a level of abstraction from the user's point of view of a DBMS. Still, the logical model is independent of the type of DBMS in which that implements it.

This type of model must necessarily respect concepts such as access keys, duplicate keys control, normalization, referential integrity, data redundancy control, among others. This model is intrinsically related to the design phase (COUGO, 2013). It is worth pointing out that this is a directed way to graphical aspect, but there are alternative ways to represent the same structurally model in a textual form (MARTELLI; FILHO; CABRAL, 2018).

3.3 Physical Model

A physical data model characterizes as a model in which the DB objects representation are already at a physical level of implementation of the occurrences (or instances) of the entities of relationships. Each DBMS can define different modes of physical implementation, involving characteristics and essential resources for the storage and manipulation of data structures. (COUGO, 2013).

In general, physical models have two well-represented aspects. First, there are the occurrences or instances, their relationships, and the basic arrangement of elements. The other aspect concerns the allocation in the different levels of possible groupings, such as tables, rows (records), columns (fields), and blocks (WEST, 2011). In short, it is the materialization of DB objects in an internal schema of a given DBMS from the execution of a logical sequence of instructions in SQL.

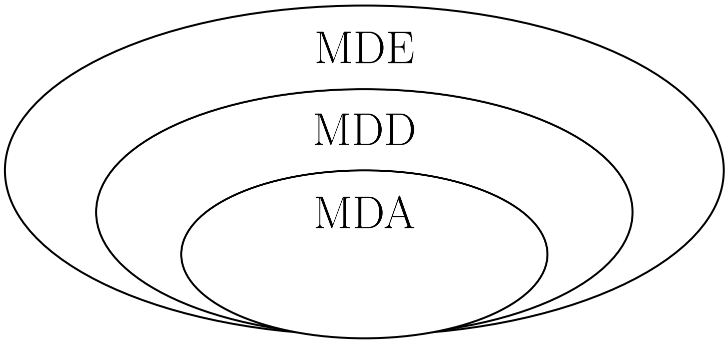
3.4 Model-Driven Engineering

A model is, conceptually, a representation, prototype or example that has the objective of reproducing or imitating something in some way. Model building is central and relevant in different scientific areas. In mathematics, physics, engineering, and chemistry areas, for example, the use of models is considered vital for theoretical and practical research in different fields of study (BAILER-JONES, 2009).

MDE, or Model-Driven Software Engineering (MDSE), is a field of Software Engineering that has its origins since the creation of the first Computer-Aided Software Engineering (CASE) tools that emerged in the 1980s. As a sub-area of Software Engineering, MDE is an approach that essentially calls for a development process to have models as the main inputs (or also outputs) of some activity. This approach aims to maximize productivity, compatibility between systems (through model reuse) and simplify the complexity involved in the software design process (BRAMBILLA; CABOT; WIMMER, 2017). Thus, this results in programs or computer activities executed in hardware or software that are automatically generated from models (SOMMERVILLE, 2015).

This area of study has been evolving considerably since the 1990s. In short, MDE has made contributions to leverage abstraction, thus aiming at automation, in almost all areas of development and analysis (BUCCHIARONE et al., 2020). These contributions identified in several domains where models are indispensable for success, *e.g.* software systems, transport systems, automotive industry, business processes, biotechnology, and genetics. Conceptually, MDE supports other concepts such as Model-Driven Development (MDD) and Model-Driven Architecture (MDA). Figure 4 illustrates the relationship among these concepts.

Figure 4 – Relationship among MDE, MDD and MDA.



Source – Adapted from (AMELLER, 2009)

MDD is a development paradigm that uses models as principal artifacts in a development process. In MDD, the implementation is usually generated automatically or semi-automatically from the models. Despite being seen as the same thing, the concepts established in the literature about MDE and MDD have direct origins in MDA, first formally proposed in 2001 by *Object Management Group* (OMG)[1].

According to (SOMMERVILLE, 2015), the differences are subtle since MDA focuses on the design and implementation stages of the software development process, being very similar to MDD, however implementing OMG-specific guidelines. Hence, this causes the MDD processes to use various abstractions to hide, or even remove, details that could eventually make a model more complex.

Once that is said, it follows that MDA is a subset of MDD. On the other hand, MDE can address many others topics in Software Engineering processes, including model-based requirements engineering, model-based software processes for development, or even model-based testing. MDE, as a methodology, helps to apply the advantages of modeling in Software Engineering activities. For (BRAMBILLA; CABOT; WIMMER, 2017), this approach takes into consideration four fundamental aspects listed below.

1. **Concepts**: the components that build the methodology, ranging from language artifacts to actors, and so forth;

2. **Notations**: The way concepts are represented, *i.e.* the languages used in the methodology;

[1] MDA Guide 2.0: <https://www.omg.org/cgi-bin/doc?ormsc/14-06-01.pdf>

3. **Processes and Rules**: The activities that lead to the elaboration of the final product, the rules for its administration and control, and the statements about the desired properties (correction, consistency, etc) of the products or of the process itself;

4. **Tools**: Applications that facilitate the execution of activities or their control, covering the production process, and supporting the developer in the use of notations.

The motivation behind MDE is increasing the abstraction level of the development process in general and then capturing processes or systems as a collection of reusable models. Therefore, it aims to reduce the difficulty associated with the development of software systems, which are generally more complex, through the modeling techniques implementation that supports the separation of interests and the automated generation of system artifacts from models (BATORY; AZANZA, 2015; KLEPPE et al., 2003).

Objectively, the MDA approach defines a structure to perform the MDE, or also the MDD. This approach defines three layers used as pillars for the whole process (FRANTZ, 2012) listed below. The conceptual relationship between these levels, using transformation mechanisms and transformation rules, is exemplified in Figure 5.

1. **Computation-Independent Models (CIMs)**: describe business objects and activities independently from support systems;

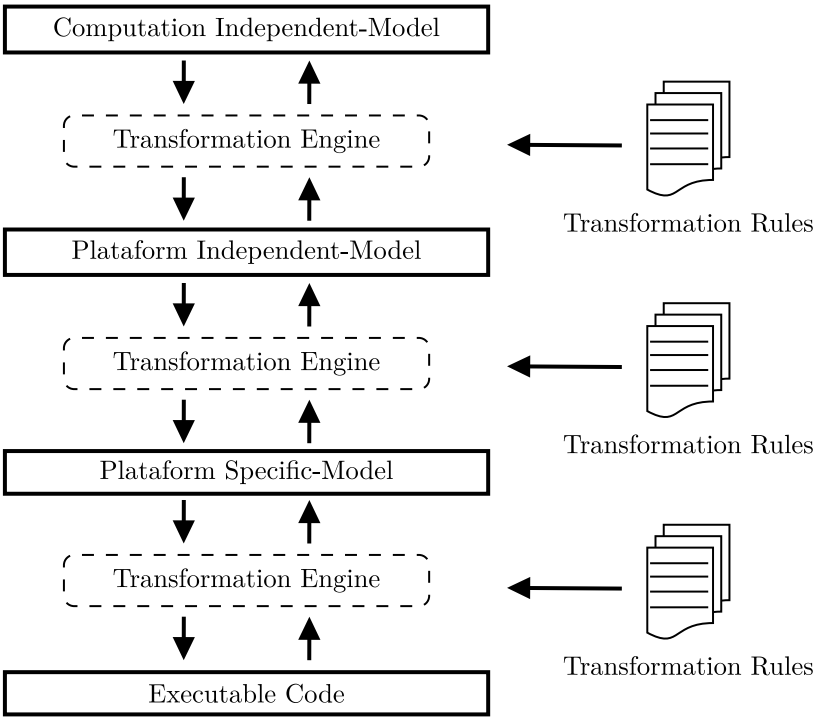
2. **Plataform-Independent Models (PIMs)**: describe how business processes are supported by systems, seen as functional black boxes, *i.e.* disregarding the restrictions associated with candidate technologies;

3. **Plataform-Specific Models (CIMs)**: describe system components as implemented by specific technologies.

For better clarification, we can understand the transformation mechanisms as generators that have model descriptions as input (HUTCHINSON; ROUNCEFIELD; WHITTLE, 2011). These generators must process such models, having a series of transformation rules implemented. For example, in the context of this study, an input artifact would be the model made using the implemented DSL, and a transformation mechanism is the several generators created. Therefore, these generators must have described all or part of the possible rules for transformation between models, *i.e.* they must carry out the equivalent mapping of these models.

The separation of interests of MDA is based, for example, on the exploration of different DSLs, each one providing constructs based on abstractions that are specific to the domain of a system. Because of this, DSLs can play a prominent role in MDE (SCHMIDT, 2006; FOWLER, 2010).

Figure 5 – MDA abstraction levels.



Source – Adapted from (FRANTZ, 2012)

3.5 Domain-Specific Languages

As stated by (DEURSEN; KLINT; VISSER, 2000), a DSL is a programming language or executable specification language. It offers expressive power focused through appropriate notations and abstractions and in general restricted to a specific problem domain. Like other languages, DSLs must present a set of sentences well defined by their syntax and semantics. According to (FOWLER, 2010), a DSL is defined as a computer programming language with limited expressiveness and focused on a particular domain. Among known examples of DSLs are:

- SQL, for databases;
- Cascading Style Sheets (CSS), for layout of web pages;
- eXtensible Markup Language (XML), for data encoding;
- Unified Modeling Language (UML), for software design;
- Systems Modeling Language (SysML), for system modeling;
- VHSIC Hardware Description Language (VHDL), for hardware design;
- LaTeX, for document typography.

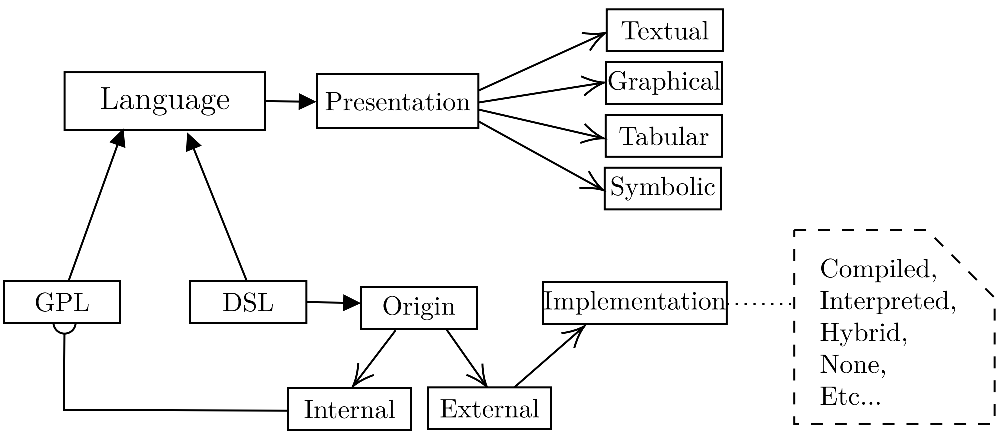
According to (FAVERI, 2013), although the term DSL can intuitively refer to a recent field of study, in fact, this is not a reality. For example, the APT is a DSL for numerically controlled machine programming developed in 1957 (ROSS, 1981). On the other hand, the Backus-Naur Form (BNF) syntax specification formalism, the most commonly used for the notation in programming languages nowadays, dates back to the late 1950s (BACKUS, 1959).

As a result, it is possible to find in the literature many studies that conceptually address DSLs but with different terminology. These include: *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).

The application of DSLs allows the development of the software more quickly and efficiently. The highest advantage in using DSLs is that the knowledge necessary for its applicability is abstracted to another level. In this way, domain experts can understand, validate and modify the code, adapting the model to their needs, making the impact of changes easier to understand. There is also a significant increase in productivity, reliability, ease of use and flexibility (GRONBACK, 2009; DEURSEN; KLINT; VISSER, 2000).

According to (MERNIK; HEERING; SLOANE, 2005), we can classify DSLs under three different dimensions: **origin**, **appearance**, and **implementation**. Figure 6 shows the classification dimensions of DSL. Regarding the **origin** of a DSL, the existing options are **internal** and **external** DSLs.

Figure 6 – DSL classification dimensions.



Source – Adapted from (FAVERI, 2013)

An **internal** DSL is designed from the syntactic and semantic rules of the grammar of an already existing language, which may be a General-Purpose Programming Language (GPL) or another DSL. Thus, for its correct functioning, an internal DSL ends up transferring all lexical, syntactic, semantic and code transformation verification activities to the host language compiler.

An **external** DSL is a language with distinct syntax and that depends on its infrastructure for lexical, syntactic, semantic, interpretation, compilation, optimization, and code generation. Compared to a GPL, an external DSL has similar features but restricted resources to the application domain for which the language is designed.

Regarding the dimension of **appearance**, a DSL can be classified as **textual**, **graphic**, **tabular**, and **symbolic**. When in textual format, DSLs expressing the domain with characters, which are then combined, generating words, expressions, sentences, and instructions that follow the grammatical rules previously established in the language. Non-textual DSLs follow the same logic but using graphical models to allow the user to express domain knowledge with a higher level of understanding, using symbols, tables, figures, and connectors.

And finally, regarding the dimension of **implementation**, we can classify the DSLs considering the perspective of its execution. These classifications comprise four groups: **(i)** well-defined execution DSLs (*e.g.* Excel Macro Language); **(ii)** DSLs which serve as input to application generators; **(iii)** not executable DSLs, but useful as input to application generators; **(iv)** DSLs not designed to be executed.

It is usual practice that the principal aspect taken into account for the development of DSLs should be its **origin**, as each approach has specific advantages and disadvantages inherent to each type (FOWLER, 2010). Although the external DSLs may have an effort associated with their creation many times greater than that of an internal DSL. Currently, some tools give great support to the development of DSLs. These tools are known as LWs and apply language-oriented programming concepts, providing a higher level of abstraction regarding complex infrastructure issues (FOWLER, 2005).

3.6 Language Workbenches

The development of a DSL is not a trivial task because, as they are programming languages, they have a syntax that is, as a logical consequence, defined by the grammar. Thus, it is necessary to use tools that support the definition of concepts for the new language (FOWLER, 2005).

LWs are tools that provide infrastructure mechanisms for implementing programming languages, thus making language creation more accessible (WACHSMUTH; KONAT; VISSER, 2014). Amongst the mechanisms provided in these environments are automatic formatting, validation based on the restrictions described in the grammar, syntax highlighting, and syntax completion. The following are some of the most well-known LWs of the present time:

- **Xtext:** released in 2006, Xtext is an open-source framework for developing textual programming languages with the Eclipse Integrated Development Environment (IDE). To specify a language, the developer describes a grammar. This grammar describes how an Ecore model derives from textual notation. From this definition, a code generator derives an ANTLR parser and classes for the object model. Xtext also has an editable Xtend generator, which gives the ability to generate code for any other grammar. Xtext includes features inherent to Eclipse IDE such as syntax highlighting, code completion, static analysis, source-code navigation and others. It is currently at version 2.25;

- **Sirius:** released in 2007 in an effort between Thales and Obeo, currently the Sirius framework is an open license project maintained by the Eclipse Foundation, whose objective is to allow the creation of modeling tools. With Sirius it is possible to specify visual DSLs and thus generate the graphical editor infrastructure. Sirius, like Xtext, has integration with specific features of the Eclipse environment. It is currently at version 6.5;

- **JetBrains MPS:** is a system developed by JetBrains, a company from the Czech Republic, which uses projective editing. This approach allows developers a better understanding, which sets them apart from other LWs. It also has common IDEs functions integrated into your development environment. It is currently at version 2021.2.1;

- **MetaEdit+:** MetaEdit+ is a proprietary LW developed by the Finnish company MetaCase to create and use DSLs. It has two versions namely MetaEdit+ Workbench and MetaEdit Modeler. Workbench includes tools for designing and using/testing modeling languages while Modeler includes tools for using modeling languages. Typically, MetaEdit+ Workbench is used by developers who design a domain DSL for a project. Then, this modeling language is used to develop final products with the support of MetaEdit+ Modeler. It is currently at version 5.5 SR1.

It is interesting to note that the ability to define cross-references provided by LWs is what makes them more attractive for the rapid development of DSLs. On a practical side note, this capability is even more notable in LWs like Xtext, as this platform provides enough abstraction to realize executable context-free grammars specification.

3.7 Xtext Framework

Xtext is an open-source framework for textual implementation of DSLs with full integration into the Eclipse IDE. The framework emerged in 2006 created by the German developer Sven Efftinge and is currently developed in the Eclipse Project as part of the

Eclipse Modeling Framework Project, *i.e.* its new versions respect the annual release cycle and are licensed under the Eclipse Public License.

The Xtext framework allows the developer to implement grammar quickly and covers all aspects of a complete infrastructure for the language. These aspects range from the parser, code generator, interpreter to full integration with all typical IDE features, such as editor with syntax highlighting, code completion, warning/error markers, file outline visualization, and automatic build infrastructure (BETTINI, 2016).

We can use the GPL Java to customize the implementation of a DSL as well as the entire infrastructure generated by the Xtext framework. However, Xtext provides the use of the GPL Xtend, a similar programming language that is completely interoperable with Java. In summary, Xtend has a more compact syntax, is easy to use and has advanced features like type inference, extension methods, complex templates in the form of string expressions, and even lambda expressions (BETTINI, 2016). Xtext also supports the generation of language servers that comply with the Language Server Protocol (LSP), as well as providing, since version 2.9, an interface for integrating text editors into web applications. These text editors are implemented in JavaScript and language-related services, such as automatic code completion, are performed via HTTP requests to a server-side component.

Although the entire fundamental infrastructure of a language generated in the Xtext framework from the definition of grammar is automatic, *i.e.* it has grammar-driven development (single source), we can modify all generated artifacts (OBEO; TYPEFOX, 2017). Among the artifacts that we can implement are scoping classes, validation classes, and templates for the defined language. The Framework still provides full support for JUnit 5 since its 2.14 release. Furthermore, it is also possible to change the behavior of the user interface and the structure of the IDE itself.

3.8 Chapter Lessons

In this chapter, we explore the main concepts of our work. The concepts of the ER approach, logical model, physical model, MDE, DSLs, LWs, and the Xtext are relevant for understanding our proposal. We realized that different LWs could be used for the development of our proposal tool. Through a previous study, we have chosen the Xtext framework because it provides the best means for the development of a textual language with the support of an integrated editor. The concept of DB modeling is relevant as we are designing a DSL for this purpose. Finally, from these knowledge domains, it is imperative to derive a metamodel that represents the data modeling that minimally follows the relational model.

4 SYSTEMATIC LITERATURE MAPPING

This chapter describes the procedures adopted for the investigation of the literature carried out for this study. For achieving this purpose, we have carried out the planning and execution of a Systematic Literature Mapping. For this, we have applied conceptually well-established guidelines based on proposals from several works (KITCHENHAM; CHARTERS, 2007; PETERSEN et al., 2008; NAKAGAWA et al., 2017).

4.1 Protocol

A Systematic Literature Mapping is helpful for researchers and practitioners as it provides comprehensive insights into the state-of-the-art in a specific area. In this study, we have conducted a mapping using the process defined by Petersen (PETERSEN et al., 2008).

4.1.1 Research Questions

Taking into account the general objective of this work, which is to develop a modeling tool for the conceptual modeling of DBs that implements a previously proposed DSL, we have formulated the following research questions to investigate the transformation mechanisms used by DSLs:

- **RQ1.** What is the state of the art of mechanisms for transforming conceptual models?
- **RQ2.** What are transformation mechanisms applied among conceptual models in the context of DSLs?
- **RQ3.** What representations of database resources do the transformation mechanisms support?

4.1.2 Research Sources

Digital libraries are the principal source of research in an systematic literature mapping (PETERSEN et al., 2008). To choose the research sources for this systematic literature mapping, we have considered three mandatory requirements that the databases must meet: **(i)** have search engine based on *Web*; **(ii)** be able to use keywords when searching, and; **(iii)** cover primary studies Computer Science area. Table 1 lists the digital libraries selected for this mapping.

4.1.3 Search String

The elaboration of the search string is not a trivial task. The identification of a terms combination that allow finding the greatest number of relevant primary studies

Table 1 – Digital libraries used.

Source	Address	Type
ACM Digital library	*dl.acm.org*	Hybrid
IEEE Xplore	*ieeexplore.ieee.org*	Bibliographic Database
Scopus	*scopus.com*	Search Engine
Google Scholar	*scholar.google.com*	Search Engine

Source – Adapted from (NAKAGAWA et al., 2017).

in an objective way requires, in most cases, experience and in-depth knowledge of the research area addressed.

For the search string definition, it is essential to define a set of words referring to the research topic as well as the synonyms are considered expressive. These terms should comprehensively represent the central theme of the study. For this work, we have established the terms and synonyms whose combination generated the generic string shown in Figure 7. Subsequently, Table 2 presents the strings that we have derived for fitting the constraints of each search engine.

Figure 7 – Generic search string.

```
("Domain-Specific Language" OR "Domain Specific Language" OR DSL OR
"Domain-Specific Modeling Language" OR "Domain Specific Modeling Language"
OR DSML OR "Domain-Specific Modeling" OR DSM)
AND
(Entity-Relationship OR ER OR "Enhanced Entity-Relationship" OR
EER OR "Entity Relationship Diagram" OR ERD)
AND
("Model Transformation" OR Model-to-Text OR M2T OR Model-to-Model OR
M2M OR "Code Generation")
```

Source – Author.

For the generation of strings as far as the entire mapping process, we have used the support of the Thoth[1] tool, a web tool that provides support for systematic review processes (MARCHEZAN et al., 2019). The string for the ACM was the base version generated by the tool, limiting it to *The ACM Full-Text Collection* set (with 606,408 records at the time) and limiting it to the abstract. In Google Scholar the string was limited to 256 characters, with hyphens making no difference. The derived string that was used was exactly 256 characters long and the results brought relevant studies. The string for IEEE needed to be limited to 20 terms. The result was obtained by searching full text and metadata of the studies. This was necessary because when only checking the abstracts, no results were returned.

4.1.4 Selection Criteria

According to (KITCHENHAM; CHARTERS, 2007), the inclusion criteria indicate by which parameters a study is included, that is, considered relevant. Likewise, the

[1] Thoth Tool: <http://lesse.com.br/tools/thoth/>

Table 2 – Search strings derived by sources.

Source	Search String

ACM	`Abstract:(("Domain-Specific Language" OR "Domain Specific Language" OR DSL OR "Domain-Specific Modeling Language" OR "Domain Specific Modeling Language" OR DSML OR "Domain-Specific Modeling" OR DSM)`
	`AND`
	`(Entity-Relationship OR ER OR "Enhanced Entity-Relationship" OR EER OR "Entity Relationship Diagram" OR ERD)`
	`AND`
	`("Model Transformation" OR Model-to-Text OR M2T OR Model-to-Model OR M2M OR "Code Generation")`

Google	`("Domain-Specific Language" OR "Domain-Specific Modeling Language" OR "Domain-Specific Modeling")`
	`AND`
	`(Entity-Relationship OR "Enhanced Entity-Relationship")`
	`AND`
	`("Model Transformation" OR Model-to-Text OR M2T OR Model-to-Model OR M2M OR "Code Generation")`

Scopus	`TITLE-ABS-KEY ( "Domain-Specific Language" OR "Domain Specific Language" OR dsl OR "Domain-Specific Modeling Language" OR "Domain Specific Modeling Language" OR dsm OR "Domain-Specific Modeling" OR dsm )`
	`AND`
	`( entity-relationship OR per OR "Enhanced Entity-Relationship" OR eer OR "Entity Relationship Diagram" OR erd )`
	`AND`
	`( "Model Transformation" OR model-to-text OR met OR model-to-model OR mcm OR "code generation")`

IEEE	`(("Full Text .AND. Metadata":"Domain-Specific Language" OR "Full Text .AND. Metadata":"Domain Specific Language" OR "Full Text .AND. Metadata":DSL OR "Full Text .AND. Metadata":"Domain-Specific Modeling Language" OR "Full Text .AND. Metadata":"Domain Specific Modeling Language" OR "Full Text .AND. Metadata":DSML OR "Full Text .AND. Metadata":"Domain-Specific Modeling" OR "Full Text .AND. Metadata"::DSM)` `AND` `("Full Text .AND. Metadata":"Entity-Relationship" OR "Full Text .AND. Metadata"::ER OR "Full Text .AND. Metadata"::"Enhanced Entity-Relationship" OR "Full Text .AND. Metadata"::EER OR "Full Text .AND. Metadata":"Entity Relationship Diagram" OR "Full Text .AND. Metadata":ERD)` `AND` `("Full Text .AND. Metadata":"Model Transformation" OR "Full Text .AND. Metadata":"Model-to-Text" OR "Full Text .AND. Metadata":M2T OR "Full Text .AND. Metadata":"Model-to-Model" OR "Full Text .AND. Metadata":M2M OR "Full Text .AND. Metadata":"Code Generation")`

exclusion criteria indicate by which parameters a study is excluded, that is, considered not relevant in the research. For this work, we have determined the selection criteria listed below.

- **Inclusion Criteria (IC)**
  - **IC1.** Primary study reporting mechanisms, guidelines, heuristics or strategies for transforming conceptual data models of relational databases.
- **Exclusion Criteria (EC)**
  - **EC1.** Duplicate primary studies;
  - **EC2.** Primary studies with less than 4 pages;
  - **EC3.** Primary study that is not written in English;
  - **EC4.** Secondary and tertiary studies;

- **EC5.** Primary study that does not provide full access;
- **EC6.** Primary study does not satisfy the inclusion criteria.

4.1.5 Quality Assessment Criteria

According to (NAKAGAWA et al., 2017), although we can find with some ease in the literature examples of quality assessment forms for the criteria elaboration and their weights definition, the researchers involved in each review or mapping must consider the particularities depending on the theme and the research questions. Therefore, there is complete freedom to define the number of quality assessment criteria and their related weights in this research type.

We have defined three (3) quality assessment criteria to rate primary studies approved after applying the selection criteria. The quality assessment criteria aim to quantify the relevance so that it is possible to realize a comparison among the selected studies. A score was also defined to be assigned based on the quality criteria. We have established acronyms to define the scores to represent them.

- **G: Good**, the study fully meets the quality assessment criteria;
- **A: Average**, the study partially meets the quality assessment criteria;
- **P: Poor**, the study does not meet the quality assessment criteria.

For all criteria, the acronym **P** (Poor) represents zero (0) and the acronym **G** (Good) depicts the maximum concept of value one (1). The acronym **A** (Average) refers to a middle ground, with a score of 0.5. By the way, the maximum possible score when we evaluated all criteria is three (3.0), and the minimum is zero (0). Table 3 shows the quality assessment criteria.

Table 3 – Quality assessment criteria.

ID	Description	Weight
QC1.	Does the study present refinement techniques, guidelines, mechanisms, or heuristics based on conceptual data models for relational databases?	1,0
QC2.	Does the study provide details of the model transformation mechanisms?	1,0
QC3.	Does the study present some evaluation way of the transformation mechanisms presented?	1,0

Source – Author.

4.1.6 Data Extraction Strategy

We have defined a data form for the extraction and analysis of relevant data of the selected primary studies. Table 4 presents the detailed description of this extraction form.

Table 4 – Data extraction form.

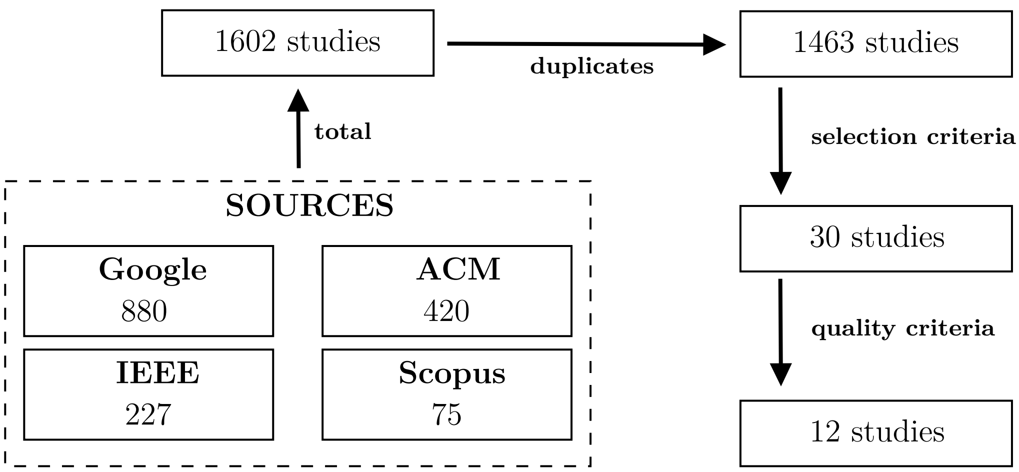
Data	Description
Solution Origin	Organization or University of the study authors
Year of Publication	Year of publication of the study
Solution Objective	Proposed Solution Description
Purpose of the paper	Description of the article's objectives
Reported strengths	Description of reported strengths
Reported shortcomings	Description of reported shortcomings
Database resources	Description of objects supported in the transformation process
DSL names	DSLs that apply transformation mechanisms among conceptual data models
Transformation mechanism type	Reported Guideline, Heuristic, Strategy or Technique

Source – Author.

4.1.7 Selection Process

We also have performed all selection steps with the help of the Thoth tool, and Figure 8 depicts an overview. Initially, we identified a total of 1602 studies extracted from the four (4) sources consulted. The next step involved the verification and exclusion of duplicate studies. This step resulted in the exclusion of 139 studies, leaving 1463 studies to apply the inclusion and exclusion criteria step. At this point, we have specifically applied the criteria based on titles, keywords, and abstracts. After this task, we completed the selection process, where 30 studies remained.

Figure 8 – Selection process steps.



Source – Author.

Then, we have submitted these papers to the quality assessment criteria step, which consisted of reading the full texts. All papers were evaluated, with those that obtained at least one (1) point in the overall classification score being considered accepted in the final set. Thus, considering the minimum score restriction, we have reached the final number of twelve (12) studies, which then underwent data extraction to formulate considerations related to the research questions.

4.2 Results and Discussion

We have evaluated the primary studies based on the quality criteria defined in Section 4.1.5. The lowest score of accepted papers was 1.0 with three (3) articles, while only one study reached 3.0, the maximum score. Table 5 summarizes the results obtained in the quality assessment criteria of studies from the academic literature.

Table 5 – Quality assessment results.

Studies	Quality Criteria			Score
Reference	**QC1**	**QC2**	**QC3**	**Total**
DIMITRIESKI et al. (2015)	G	G	G	3
DIMITRIESKI et al. (2014)	G	G	P	2
KUNG and TUNG (2010)	G	G	P	2
HARTMANN and LINK (2007)	G	G	P	2
DEY, STOREY and BARRON (1999)	G	G	P	2
ROSENTHAL and REINER (1994)	G	G	P	2
TEOREY, YANG and FRY (1986)	G	G	P	2
SUBAHI and SIMONS (2011)	A	G	P	1.5
SOUSA and CURA (2018)	A	A	A	1.5
RISTIĆ et al. (2016)	A	A	P	1
VARA et al. (2007)	A	A	P	1
GOGOLLA (2005)	G	G	P	1

Source – Author.

Regarding **RQ1** (state of the art of mechanisms for transforming conceptual models), we identified as the most significant results three studies that are related (RISTIĆ et al., 2016; DIMITRIESKI et al., 2015; DIMITRIESKI et al., 2014). These studies present research that generated the *Multi-paradigm System Modeling Tool* (MIST), which uses bidirectional DSL for conceptual modeling of DBs. Furthermore, the proposal also argues using heuristics for transformations and describes them at a high level. It is relevant to point out that the public repository with the project code[2] presents the metamodel of the DSL language as far as the Java implementation, but does not have the code for the proposed inverse transformation (from graphical models or SQL to the DSL). As the main drawback pointed out, we highlighted the framework change initially used during development due to the project discontinuation that maintained it. For this reason, the authors claim to have adopted Xtext as the final implementation framework.

We also identified the study of (HARTMANN; LINK, 2007), which presents a proposition of four (4) types of collections for the ER approach, along with an overview for a process of transforming extended ER schemes into relational databases schemes. However, this work reports a lack of a query language (extended SQL) that covers the proposed collections.

As more significant works in the final set (score >= 2.0), we cite two older works, from (ROSENTHAL; REINER, 1994; TEOREY; YANG; FRY, 1986). The first presents a

[2] MIST repository: <https://github.com/vdimitrieski/mist>

prototype that is actually a toolchain. There is a description of the steps for transforming relational database models. The principal lack indicated refers to the underlying unified data, which can be redundant in some cases. The second defines a design methodology for large-scale relational databases using an EER approach, but the notation used was not implemented. Despite being focused on large-scale databases, the authors point out that the data normalization stage proposed in the methodology can be extensive and complex for coding.

As works that are a little beyond the scope of this research but are still relevant, we can mention the study by (SOUSA; CURA, 2018), which presents a process to transform a conceptual model into a representation of the NoSQL graphical data model. In the case of this study, the solution only supports graphical databases but the authors state that they intend to extend it to other NoSQL databases types and even the classic relational model. At all events, the authors also presented pseudo-algorithms of mapping for elements of the ER approach.

Finally, we can also cite the study by (DEY; STOREY; BARRON, 1999) that presents a generic framework for analyzing relationships in data models. Unlike other approaches, it focuses only on relationships, relegating entities. It introduces a generic notation for representing relationships and provides guidelines that identify which modeling constructs to use in different situations. It also provides explicit rules for evaluating the degree of a relationship as well as understanding recursive relationships. The authors also analyze, using special semantics, the cases of weak relationships. Finally, the study describes the design rules for transforming these ER constructs into a relational model, along with the associated integrity rules that can be derived from the conceptual design.

Regarding **RQ2** (transformation mechanisms applied among conceptual models in the context of DSLs), we can mention that all the studies evaluated present transformation mechanisms at different levels of abstraction (algorithms and pseudo-algorithms). Although there are similarities between practically all reported mechanisms, there are five (41.7%) that report explicitly naming them as strategies, three (25%) studies describe them as guidelines, and four (33.3%) indicate them as heuristics.

Regarding **RQ3** (representations of database resources supported in transformation mechanisms), the analyzed studies, when explicitly mentioning the database resources, in general, converge in relation to the most common ones such as tables and relations, with exceptions (DEY; STOREY; BARRON, 1999). Other database resources are also covered such as triggers, views, and stored procedures. We highlighted again the studies that refer to MIST, as they claim that there is support for database functions as well.

We can also mention that few DSLs report with a proper name. In fact, among the twelve (12) studies evaluated, only two (2) explicitly name their languages, namely EERDSL (RISTIĆ et al., 2016; DIMITRIESKI et al., 2015; DIMITRIESKI et al., 2014) and ReMoDeL DBQ (SUBAHI; SIMONS, 2011). Finally, on the methods used to evaluate the proposals, there is only one preliminary study that presents an experimental validation of the proposal (DIMITRIESKI et al., 2015) using sixteen (16) participants, two (2) IHC experts, three (3) experts in systems modeling and eleven (11) students. Among the students, six (6) were masters in the area of DBs and five (5) doctoral students with experience in modeling. In general, all other studies indicate the lack of an evaluation of their propositions as possible future work.

4.3 Related Work

In general, the textual approach has been discussed in several works that have investigated database modeling for many years. For example, we can cite the work of (MALHOTRA et al., 1989) which presents a syntax for an integrated ER programming language. Problems that arise when a query language is incorporated into a general-purpose programming language are also discussed, although the implementation of the language is not necessarily presented.

Another more recent case is the study by (HOSSAIN; RAJAN; SCHWITTER, 2019), where a controlled natural language for specifying and verbalizing entity relationship (ER) models is presented. The idea is not only to solve the verbalization problem for these models, but also to provide the benefits of automatic verification and validation and semantic round-trip to make the communication process transparent between domain experts and knowledge engineers.

However, although a grammar specification that defines the controlled natural language made in Prolog is presented, further implementations of usage are unknown. Having said that, we start with a more practical comparison with a set of tools already implemented and that use similar approaches to ours.

Considering solutions and projects cited in secondary studies, and similar to our proposal, we can mention the survey presented in the work of (RIQUELMO et al., 2021). This study is a Multivocal Literature Review, *i.e.* a form of Systematic Literature Mapping that includes gray literature.

This study covers a final set of ten (10) primary studies focused on DSLs, which identifies fifty-five (55) tools already applied in industry and academia for modeling ER at a conceptual, logical, and physical level. Table 6 presents an overview of the most relevant solutions of the study when compared to our proposal. Furthermore, we also added the *bigER* tool, a recently released plugin for *VS Code* that uses the Xtext framework basis.

The University of Novi Sad in Serbia conceived the *Multi-paradigm System Modeling Tool* (MIST). The conducted systematic review selected publications that address the development of this tool, where its description was previously made.

Table 6 – Related modeling tools.

	MODELING TOOLS					
Features	**ERtext**	**MIST**	**dbdiagram.io**	**QuickDBD**	**RelaX**	**bigER**
IDE Integration	Eclipse	Eclipse				VS Code
Web-based			✓	✓	✓	
LSP Implementation	✓	✓				✓
Open Source	✓	✓			✓	✓
Free of Charge	✓	✓	±	±	✓	✓
Textual Editor	✓	✓	✓	✓	✓	✓
Model Validation	✓					✓
Diagram View	✓		✓	✓	✓	✓
Code Generation	✓		✓	✓		✓
Experimental Evaluation	✓	✓				

Legend: IDE = Integrated Development Environment, LSP = Language Server Protocol, ± = Partially.
Source – Author.

dbdiagram.io[3] is a free web tool for drawing ER diagrams, developed by a Singaporean company with a textual approach that implements its DSL. This DSL uses a model that is very close to the logical one. The tool's differential is its fast learning curve and, in addition, the presentation of a graphical representation of what is being modeled. The user can freely organize the presentation of diagram elements in real-time. However, it is relevant to point out that all the modeling is a textual-based approach. The tool even offers automatic SQL code generation.

Likewise, *QuickDBD*[4], developed by a company in Ireland, is a web tool with the same operating mode as *dbdiagram.io*, also implementing a textual DSL for modeling DBs. However, as it is a proprietary tool, it has an associated cost and an industry focus. Both tools are very similar in generating graphical representations of the modeling and present several arguments for their adoption, such as the quick understanding of their DSLs, the perspective of performing fluent work, access from any platform, and the sharing of the models with other users.

RelaX (Relational Algebra Calculator)[5] is a free web tool. It is a tool developed at the University of Innsbruck, Austria, and aimed at teaching relational algebra by performing operations on relational databases. It has a textual approach, using a DSL called RelAlg, and includes two operating perspectives: RelAlg instructions and SQL instructions. RelaX uses a physical-level model approach to operations, such as DDL and Data Manipulation Language (DML). Despite its functionalities, RelaX has not intended to be a professional tool for DB design and modeling, but its use is restricted to teaching within the academy.

Finally, the *bigER* tool offers features for flexibly specifying and visualizing ER conceptual data models. Within the *VS Code* text editor, this solution allows modeling

3 <https://dbdiagram.io/>
4 <https://quickdatabasediagrams.com/>
5 <https://dbis-uibk.github.io/relax/>

through its DSL to specify ER elements. It also offers an interactive visualization for displaying ER diagrams and manages to generate SQL code in a generic format. This solution still stands out for being the only tool to incorporate the LSP implementation use *de facto*. Furthermore, being the only one distributed through the widely known *VS Code* ecosystem.

Overall we can verify that our proposal meets the relevant functionalities when comparing all the tools. Perhaps the main positive difference concerning most of the other competing proposals is that we carried out experimental evaluations to assess the conceived idea of the project. In addition, the grammar is relatively simple, and the SQL code generation is specific platforms directed. The main weakness currently observed is the lack of a web-based alternative, *i.e.* a version of ERtext that can run directly in web browsers. However, thanks to the LSP implementation provided by Xtext, migrating to a web application will be a relatively easier task than starting from scratch.

4.4 Chapter Lessons

Every year, several contributions to the ER modeling are published. Database modeling is a fundamental area in computing, and unfortunately, in the literature not trivially found DSLs that support this activity. To follow the evolution and trends of various DBs systems, a search for alternatives to design is essential. This chapter provides an overview of the DSLs used for transforming data models through systematic literature mapping.

This mapping process covered 1602 articles to investigate primary studies that made proposals that showed transformation mechanisms for modeling DBs. The systematic mapping protocol was detailed as well as its conduction and subsequent analysis of the results obtained. In the end, we selected 12 primary studies to be analyzed quantitatively and qualitatively. Among the results of the related works, the study of (DIMITRIESKI et al., 2015) stands out, which presents a bidirectional modeling tool that applies its DSL based on the EER approach and describes it in detail the transformation mechanisms between the models.

In general, these works served as guidelines in the way we structured and subsequently implemented our proposal. Overall we believe that we have managed to cover at least the most basic elements expected in a DSL for conceptual modeling of DBs. However, the studies evaluated also present a range of other possibilities for representing DB objects that we do not support. Therefore, we can highlight that there is a previous planning that is already being executed and it is related to the evolution of our proposal, initially with the possibility of mapping weak entities and automatic generation of stored procedures that cover the CRUD operations (Create, Read, Update, Delete).

Finally, we can also highlight the table that presented proposals and tools from academia and industry similar to our solution developed. In general, we can say that

ERtext achieves most of the main characteristics expected from solutions of this type, with the differential of having been extensively evaluated with the execution of empirical experiments. Still analyzing these related work, we infer that a natural step of our project expansion will be supporting not only the plugin distributed into the Eclipse ecosystem albeit could implement the migration to the *VS Code* ecosystem and web-based systems.

5 ERTEXT MODELING TOOL PROPOSAL

This chapter presents the main proposal of this study. Aspects about the definition of DSL are detailed, as well as the modeling tool built.

5.1 Development

The development of the modeling tool proposal took place from the work extension of (LOPES, 2019), restarting from June 2020. In that study, we have defined language requirements, design decisions, and derived grammar. We use all these concepts to implement the new editor infrastructure that supports this DSL.

5.1.1 Language Requirements

This section lists the requirements defined based on the literature used in this work and even the prior knowledge of the researchers involved in conducting the study. These requirements are directly related to design decisions.

- ***RQ1. The DSL must be made available under an open-source license.*** As the focus of the proposal is on the teaching process, the language must be open-source. The advantage that this requirement provides is the further evolution and collaborative maintenance with the involvement of other developers.

- ***RQ2. The DSL must allow the textual representation of conceptual DB models.*** As it is an objective that the solution is another option concerning graphical approaches, this requirement is justified. This allows focusing on understanding the domain and developing the DSL.

- ***RQ3. The conceptual models must support the definition of entities, attributes, relationships, and cardinalities.*** The tool used to develop the language needs to allow implementation of the domain concepts that govern the traditional ER diagram structure.

- ***RQ4. The conceptual models must support the definition of identifying attributes, generalization/specialization, self-relationships, and ternary relationships.*** The language should allow more sophisticated domain concepts to be defined.

- ***RQ5. The implementation of DSL must perform the transformation from the conceptual to the logical model.*** The solution must perform the transformation from conceptual to logical, displaying the generated result to the user.

- ***RQ6. The implementation of DSL must generate equivalent SQL statements based on the conceptual or logical model.*** The solution needs to generate SQL statements for different DBMSs.

- ***RQ7. The implementation of DSL should generate some form of graphical representation.*** The solution needs to perform diagram generation based on the conceptual model.

5.1.2 Design Decisions

This section describes the Design Decisions (DD) to create the textual DSL that supports all the requirements presented in Section 5.1.1.

For each design decision, we have indicated its associated requirements.

- ***DD1. The solution should adopt an open-source LW to help implement the textual DSL (RQ1, RQ2).*** Through the investigation conducted in the previously cited study (LOPES, 2019), we have selected LW Xtext for the development of the proposal as it is an open-source framework focused on the development of textual DSLs, providing all the necessary infrastructure. Furthermore, Xtext is a tool with a high level of maturity, detailed documentation and an extremely active community.

- ***DD2. The DSL must provide a textual representation equivalent to the commonly used graphical ER models (RQ3, RQ4).*** For the requirements covered by this design decision, we have adopted the strategy of performing an analysis in the tools verified in the mapping described in (LOPES, 2019), as well as in the reference book by (HEUSER, 2009).

- ***DD3. The solution must perform the transformation between models (RQ5).*** Xtext uses Eclipse Modeling Framework (EMF) templates as the in-memory representation of any parsed text file. This in-memory object graph is called an Abstract Syntax Tree (AST). These concepts are also called Document Object Graph (DOM), semantic model, or simply model. Thus, there is a representation of the grammar model in the form of a kernel metamodel in the core of EMF, called the Ecore model. Since having the proposed DSL Ecore as a representation, it is possible to apply transformation rules, thus generating other models.

- ***DD4. The solution must provide the integration between the DSL and other technologies (RQ6).*** The solution should allow the export of the built models to an SQL statement format, thus representing the physical model. Initially we perform this integration for PostgreSQL and MySQL

- ***DD5. The solution should provide the generation and visualization of diagrams of the models built with DSL (RQ7).*** The solution must allow the visualization of data models built using a graphical format and representing the conceptual model. Initially we perform this integration using PlantUML. PlantUML is an open-source tool that allows the creation of diagrams from a simple text language.

5.1.3 DSL Grammar

In the current phase of the work, the language at the conceptual level is meeting the language requirements defined in Section 5.1.1. The definition of DSL using a variation of the Extended Backus-Naur Form (EBNF) metasyntax notation is displayed in Figure Figure 9.

Figure 9 – Definition of implemented DSL grammar.

```
grammar org.xtext.unipampa.erdsl.ErDsl with org.eclipse.xtext.common.Terminals
generate erDsl "http://www.xtext.org/unipampa/erdsl/ErDsl"

ERModel:
        ("Generate" targetGenerator= ("LogicalSchema" | "PostgreSQL" |
                                      "MySQL" | "Diagram" | "All") ";")?
        domain=Domain ";"
        ("Entities" "{") entities+=Entity+ ("}" ";")
        ("Relationships" "{") relations+=Relation* ("}" ";");

Domain:
        "Domain" name=ID;

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

Entity:
        name=ID ("is" generalization=("total/disjoint"  | "total/overlapped" |
                                      "partial/disjoint" | "partial/overlapped")
        is=[Entity])?
        ("{" attributes+=Attribute
        ("," attributes+=Attribute)* "}")?;

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

RelationSideLeft:
        target=[Entity] | target=[Relation]
        cardinality=("(0:1)" | "(1:1)" | "(0:N)" | "(1:N)") ;

RelationSideRight:
        cardinality=("(0:1)" | "(1:1)" | "(0:N)" | "(1:N)")
        target=[Entity] | target=[Relation];

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

Source – Author.

There are topics related to scope validation that we analyzed and then implemented, as in the case of handling unwanted cross-references and other restrictions inherent to the ER model. However, we recognize that it is possible to evolve other points regarding scope validation.

The `grammar` command specifies the name of the DSL, while the `with` statement declares an inheritance from another language. In the case of the proposed grammar, a standard Xtext grammar called `Terminals` is used, which provides some predefined rules, such as the `ID` rule for identifiers. The `generate` command is the statement that produces the AST of the language.

The first rule, called the input rule, is `ERModel` and defines the general structure of the language. Words and symbols enclosed in double or single quotes indicate reserved words (keywords). For example, the `Entities` object must be preceded by `"Entities{"`. This object represents a kind of *container*, which is indicated by the `+=` assignment operator. It is an object that can contain other objects, in this case, one or more `Entity` (entities). It is established that each DSL file must also be composed of a `Domain` (domain) and zero or more `Relation` (relations). In addition, a file optionally contains a `Generate` command, which indicates which file(s) should be generated.

For a better understanding, it should be clear that the multiplicity is indicated by `*` (zero or many), `+` (one or many), or `?` (zero or one). By not placing any of these operators, the grammar implicitly expects only one occurrence. Regarding assignments, when only one `=` is specified, it means that the left object expects only one record. So, for `+=` then zero is expected, one or more occurrences, and so forth.

The `Domain` object precedes by a reserved word with the same name, followed by an identifier. The entity is defined by the word `Entity` and a specific identifier name for this object. Setting an inheritance is optional via the `is` keyword. When specifying an inheritance, it is necessary to indicate the type among the four (4) possible ones to then point to the inherited class, *e.g. EntityB is total/disjoint EntityA*. After defining the name, a curly bracket opens a snippet to specify the entity attributes. An entity must contain at least one attribute, but it does not need to be an identifier due to the possible existence of weak entities.

Composed rules are only performed due to the possibility of grouping expressions using parentheses, in addition to the possibility of using other rules through cross-references. The square brackets in the `is` attribute of the `Entity` rule are meant to indicate that we aim to use only the `name` attribute that identifies the referenced object. If this detail is not specified then the language compiler would interpret that it is necessary to apply the whole `Entity` rule again, that is, the mandatory definition of another entity would be started, thus entering in a loop. Entity attributes are defined by a name, inheriting the `ID` rule from `Terminals`, as well as by a data type. Optionally an attribute can have the keyword `isIdentifier` to symbolize primary keys.

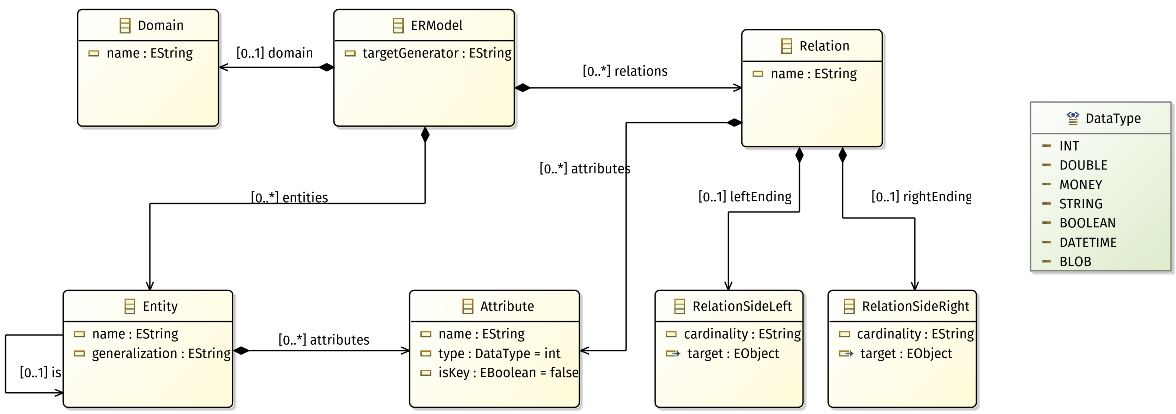
A relationship is defined, already inside the body of the `Relationships` block, with a mandatory declaration of its identification. Then square brackets are opened and we must specify two elements that represent the sides of the relationship. These elements are the `RelationSideLeft` and `RelationSideRight`. These objects must be separated by the `relates` expression. Both elements have two attributes, one indicating cardinality and the other referring to the related object (an entity or other relationship). A relationship can still have attributes declared between curly braces, and for these cases, the `Attribute` rule is applied.

Attribute types are contained in an enumerated list called `DataType`, on the left is the representation within the Ecore model and on the right is the keyword available to the user. The conditional symbol `|` means the logical operator *OR* and serves to separate each `<key> = <value>` definition as an option within the list.

5.1.4 The Metamodel

Metamodels (or semantic models) in the context of MDA are expressed using MOF (Meta-Object Facility), an OMG standard for MDE. A metamodel specifies the abstract syntax of a modeling language, and as such is a special type of model. It can be understood as the specification of the set of all possible models expressed in a given modeling language. Each model must conform to its metamodel(s), that is, it must follow the standards of syntax, semantics, rules and relationships between concepts. As any DSL can be metamodeled, and the grammar we implement has the metamodel shown in Figure 10.

Figure 10 – DSL grammar metamodel.



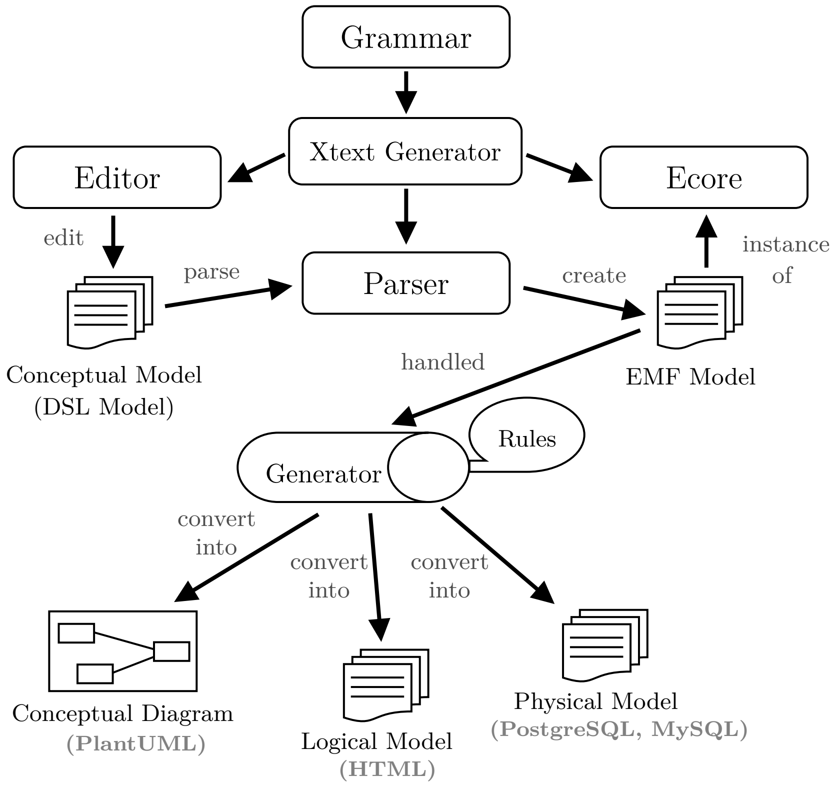
Source – Author.

The language metamodel is relevant in the approach used in Xtext as it allowed the DSL and generators to be built separately.

5.1.5 Tool Architecture

The Xtext framework generates much of the infrastructure for languages based fundamentally on defined grammar. Figure 11 provides an abstract level overview of the final ERtext architecture, which we have modified by adding the SQL generators and diagram generator with PlantUML.

Figure 11 – ERtext architecture.



Source – Author.

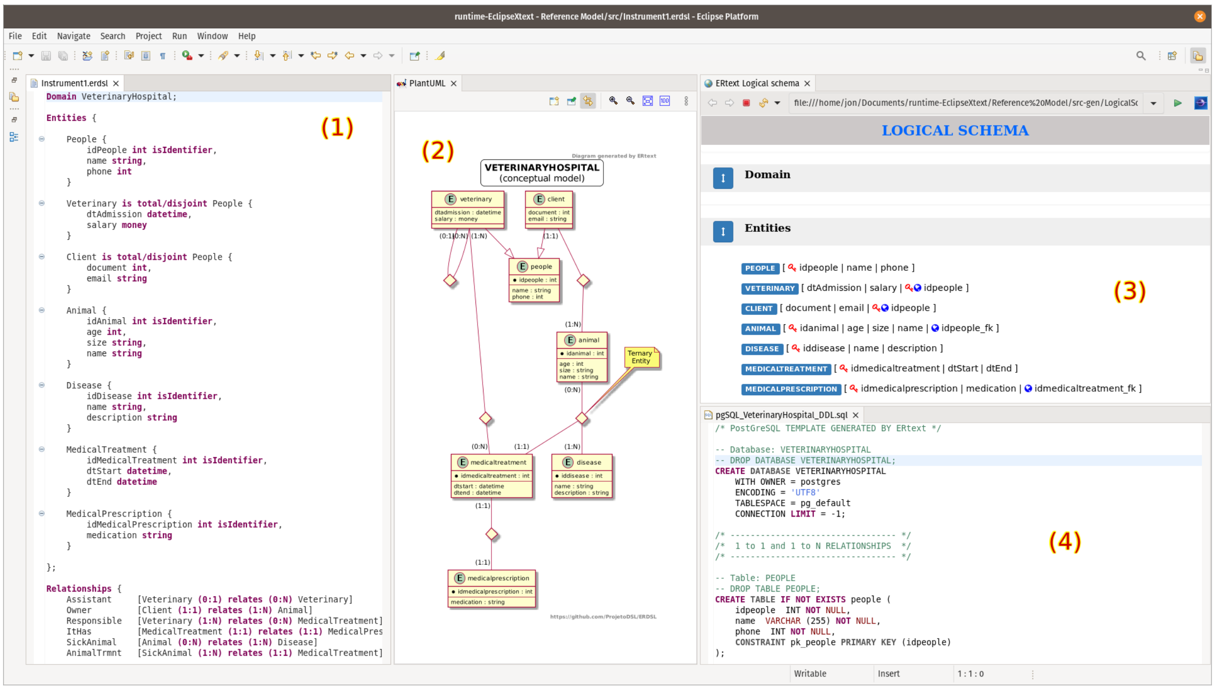
We say that Xtext generates a majority part, but not entirely, as several aspects must be implemented according to the requirements and objectives of the language. For example, we have manually needed and implemented the following aspects, all of the scope-based validation, bug marking and feedback, snippet suggestion, template creation, and language-specific project and file creation wizards.

5.1.6 Stable Version

The source code of the stable version of the produced modeling tool is available in a public repository on GitHub[1]. This version gone to the last experimental evaluation which was designed to evaluate not only the DSL but also the artifacts that can be produced by it, as well as its usability. Figure 12 presents a visual fragment of the tool in use.

[1] Repository: <https://github.com/ProjetoDSL/ERDSL>

Figure 12 – ERtext tool stable version.



Source – Author.

The caption (1) in the figure demonstrates the editor with a model described in the implemented DSL notation. Caption (2) presents a graphical representation (diagram) using the integration with PlantUML. For this, the model is transformed from our DSL to the necessary notation for PlantUML to generate the static diagram using Graphviz libraries. Caption (3) shows the model transformed to a logical notation, already solving the relations, *i.e.* the primary and foreign keys. Finally, in caption (4) we present the generated SQL DDL code for DBMS PostgreSQL.

For the transformations of the models, we have evaluated the transformation rule options described by (HEUSER, 2009). Regarding 1:1 type relationships, we have adopted the column addition strategy for the three (3) possible cases. For the 1:N relationships, we have followed the same transformation premise for the four (4) possible cases. For the transformation of N:N relationships, we have chosen the only recommended transformation rule, *i.e.* the rule for creating own tables for the three (3) cases.

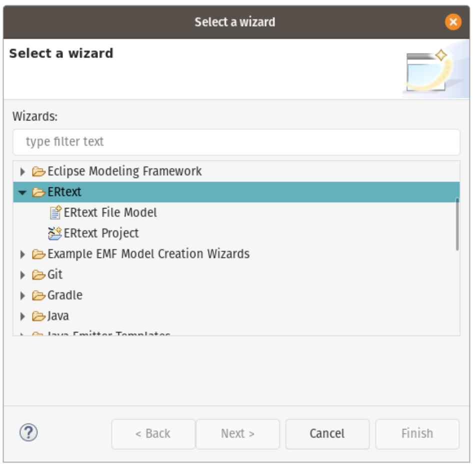
Basically, during the modeling activity a plain text file is created with the language extension (.erdsl) on the system running the tool. Each time this file is saved, an Ecore model is created or updated and then called to the generators that handle the artifacts that must be created. It happens through the `IFileSystemAccess2` interface that provides the means to iterate through the created model and apply as transformation rules in a standalone scenario. This interface provides an abstraction for file system operations, enabling logical path mapping (output artifact). This interface composes all the extension interfaces for `IFileSystemAccess` with other improvements implemented.

For the sake of the record, our specialized generators use the Template Method design pattern (GAMMA et al., 1995) in their implementation. The Template Method is a behavioral design pattern that defines a structure of an algorithm in a superclass, letting the subclasses override specific steps of the algorithm without modifying its structure.

This design pattern recommends that the algorithm be modularized into a series of steps, transforming these steps into methods and placing a series of calls to these methods within a single default method. In our solution, we defined the steps of this transformation in an abstract class. Thus, our specialized generators implement all these abstract steps and eventually override some optional ones when necessary, but never the default method itself.

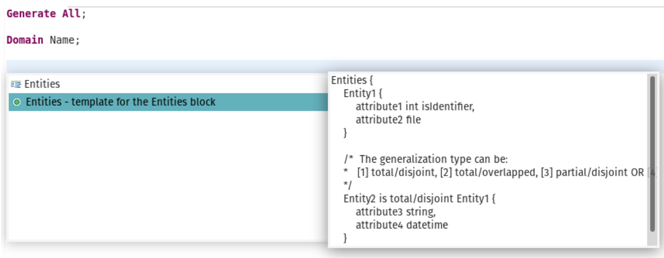
The stable version also has the implementation of other usability features. Among the highlights are the project and file creation wizards (Figure 13) for DSL and code template proposals (Figure 14) during use. Last but not least, we could not have been implemented these features without the help of the Xtext developer community[2].

Figure 13 – Project wizard.



Source – Author.

Figure 14 – Template proposals.



Source – Author.

5.1.7 Toy Example

To demonstrate a toy example, let's imagine a hypothetical model of a simple `social network`. In this example, there are seven (7) entities. A `User` entity, which specializes in the `People` and `Organization` entities. Users can share `Posts` that may contain associated `Photos`. Users can have friendly relationships with other users as well as participate in `Groups`. Finally, for each relationship of Users that belong to Groups, we model `Roles` (*e.g.* owner, moderator, participant, etc) entity. As the relationship of Users belonging to Groups is an N:N relationship, the addition of Role generates a

[2] Xtext community forum: <https://www.eclipse.org/forums/eclipse.modeling.tmf>

ternary (associative) relationship between the derived entity and role occurrences. Figure 15 shows the description.

Figure 15 – Description of a social network example.

```
Domain Social_Network;
Entities {
    User {
        idUser int isIdentifier,
        loginEmail string,
        loginPassword string
    }

    People is total/disjoint User {
        name string,
        birthday datetime
    }

    Organization is total/disjoint User {
        fantasyName string
    }

    Posts {
        idPost int isIdentifier,
        contents string,
        dateCreated datetime,
        dateModified datetime
    }

    Photos {
        idPhoto int isIdentifier,
        image file
    }

    Groups {
        idGroup int isIdentifier,
        title string,
        description string
    }

    Roles {
        idRole int isIdentifier,
        description string,
        permissionLevel int

    }
};
Relationships {
        Friendship      [User (0:N) relates (1:N) User] {date datetime}
        PostSharing     [User (1:1) relates (0:N) Posts]
        PhotoPost       [Posts(1:1) relates (0:N) Photos]
        BelongsToGroup  [User (0:N) relates (0:N) Groups]
        RoleInTheGroup  [Roles(1:1) relates (1:N) BelongsToGroup]
};
```

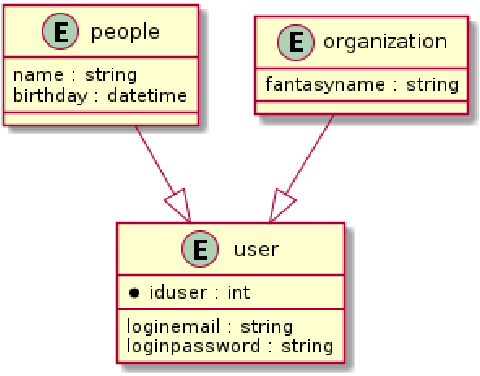
Source – Author.

As stated earlier, the tool calls the generators each time that saves the model. If the tool considers the model valid according to the DSL syntax, it dispatches the in-memory Ecore model to the main generator. Then forwards it to the other specialized generators, thus producing the diagram (conceptual model), HTML (logical model), and the SQL files (physical model). In these generators, the tool generates the target files in a folder called "`src-gen`" at the root of the project folder.

Figure 16 display the representation of the inheritance between the `User` entity and the `People` and `Organization` entities. The tool uses the UML notation for class diagrams when using PlantUML. Figure 17 shows the equivalent generated in the logical model.

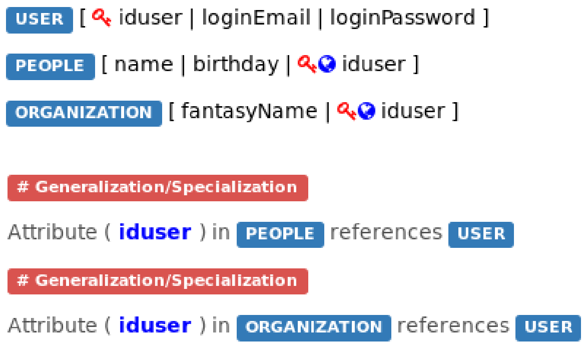
Figure 18 shows the generated SQL model for the PostgreSQL platform, also equivalent to the previous models. The same logic is used for binary relationships and ternary entities in both logical and physical models, that is, entities are created with the references being resolved (foreign keys). The relationships have their mappings being indicated right after the structure of the entities. In Figures 19 and 20, the models are presented with three (3) binary relations (PostSharing, PhotoPost, and FriendShip). In

Figure 16 – Diagram snippet of a generalization.



Source – Author.

Figure 17 – Logical model snippet of a generalization.



Source – Author.

Figure 18 – SQL model snippet for generalizations.

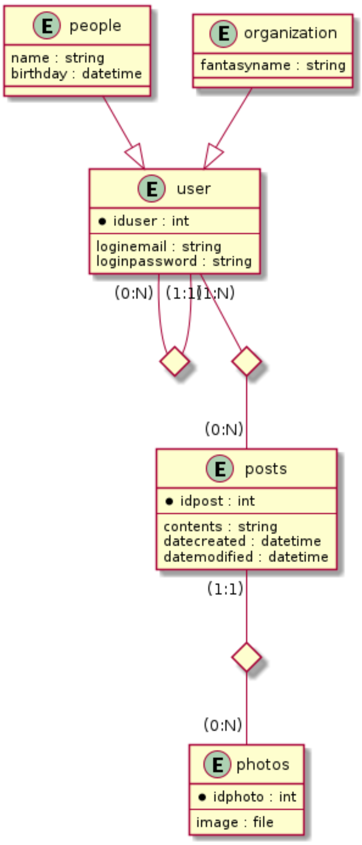
```sql
-- Table: USER
-- DROP TABLE USER;
CREATE TABLE IF NOT EXISTS user (
    iduser INTEGER NOT NULL,
    loginemail VARCHAR (255) NOT NULL,
    loginpassword VARCHAR (255) NOT NULL,
    CONSTRAINT pk_user PRIMARY KEY (iduser)
);

-- Table: PEOPLE
-- Generalization/Specialization TOTAL/DISJOINT from table USER
-- DROP TABLE PEOPLE;
CREATE TABLE IF NOT EXISTS people (
    iduser INTEGER NOT NULL,
    name VARCHAR (255) NOT NULL,
    birthday DATE NOT NULL,
    CONSTRAINT pk_people PRIMARY KEY (iduser)
);

-- Table: ORGANIZATION
-- Generalization/Specialization TOTAL/DISJOINT from table USER
-- DROP TABLE ORGANIZATION;
CREATE TABLE IF NOT EXISTS organization (
    iduser INTEGER NOT NULL,
    fantasyname VARCHAR (255) NOT NULL,
    CONSTRAINT pk_organization PRIMARY KEY (iduser)
);

-- GENERALIZATION
ALTER TABLE ONLY public.people ADD CONSTRAINT fk_gen_people_user FOREIGN KEY (idUser) REFERENCES public.user (idUser);
-- GENERALIZATION
ALTER TABLE ONLY public.organization ADD CONSTRAINT fk_gen_organization_user FOREIGN KEY (idUser) REFERENCES public.user (idUser);
```

Source – Author.

the conceptual model the N:N Friendship self-relation of the User entity is presented omitting the entity that is derived. However, in the logical model, it is shown together with the other relationships, both in structure and in the detected mapping list. For knowledge, the red key symbols represent the primary keys in the logic model. Blue globe symbols represent foreign keys. When an attribute is both key types, both symbols are used. Figure 21 shows the equivalent SQL model generated by the tool.

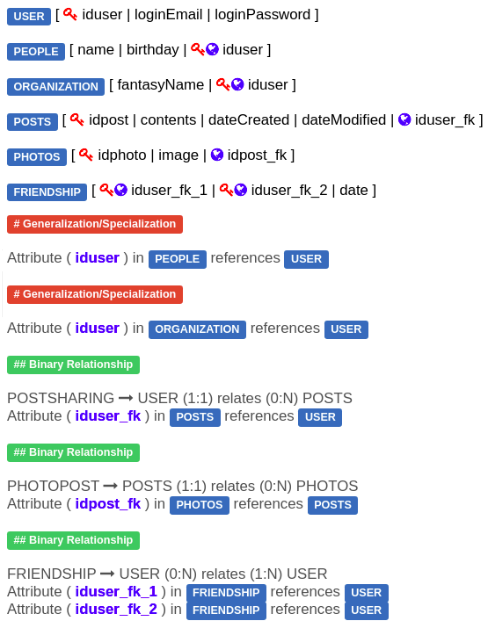
Concerning ternary relationships, when it generates a diagram, there is an indication of where they occur in the image with the help of an annotation. It is important to note that the language does not yet support n-ary relationships. In the logical model, the

Figure 19 – Diagram snippet with binary relatioships.



Source – Author.

Figure 20 – Logical model snippet with binary relationships.



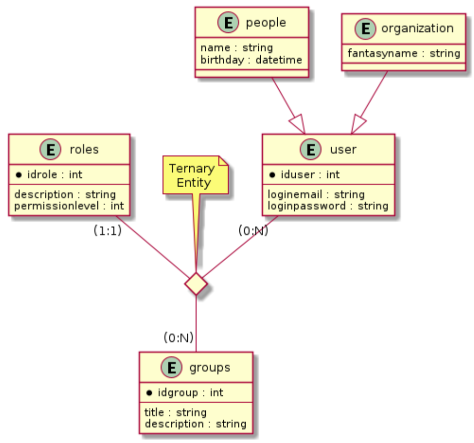
Source – Author.

Figure 21 – SQL model snippet with binary relationships.

```sql
-- Table: POSTS
-- DROP TABLE POSTS;
CREATE TABLE IF NOT EXISTS posts (
    idpost INTEGER NOT NULL,
    contents VARCHAR (255) NOT NULL,
    datecreated DATE NOT NULL,
    datemodified DATE NOT NULL,
    postsharing_iduser INTEGER NOT NULL,
    CONSTRAINT pk_posts PRIMARY KEY (idpost)
);

-- Table: PHOTOS
-- DROP TABLE PHOTOS;
CREATE TABLE IF NOT EXISTS photos (
    idphoto INTEGER NOT NULL,
    image BYTEA NOT NULL,
    photopost_idpost INTEGER NOT NULL,
    CONSTRAINT pk_photos PRIMARY KEY (idphoto)
);

-- Table: FRIENDSHIP
-- DROP TABLE FRIENDSHIP;
CREATE TABLE IF NOT EXISTS friendship (
    user_iduser_1 INTEGER NOT NULL,
    user_iduser_2 INTEGER NOT NULL,
    date_ DATE NOT NULL,
    CONSTRAINT pk_friendship PRIMARY KEY (user_iduser_1, user_iduser_2)
);

-- GENERALIZATION
ALTER TABLE ONLY public.people ADD CONSTRAINT fk_gen_people_user FOREIGN KEY (idUser) REFERENCES public.user (idUser);
-- GENERALIZATION
ALTER TABLE ONLY public.organization ADD CONSTRAINT fk_gen_organization_user FOREIGN KEY (idUser) REFERENCES public.user (idUser);
-- [1toN]: Relationship in which BOTH ENTITIES have NO GENERALIZATION
ALTER TABLE ONLY public.posts ADD CONSTRAINT fk_postsharing FOREIGN KEY (postsharing_iduser) REFERENCES public.user (iduser);
-- [1toN]: Relationship in which BOTH ENTITIES have NO GENERALIZATION
ALTER TABLE ONLY public.photos ADD CONSTRAINT fk_photopost FOREIGN KEY (photopost_idpost) REFERENCES public.posts (idpost);
-- [NtoN]: Self-relation WITHOUT generalization
ALTER TABLE ONLY public.friendship ADD CONSTRAINT fk_friendship_user_1 FOREIGN KEY (user_iduser_1) REFERENCES public.user (iduser);
ALTER TABLE ONLY public.friendship ADD CONSTRAINT fk_friendship_user_2 FOREIGN KEY (user_iduser_2) REFERENCES public.user (iduser);
```

entity derived from this type of relationship is displayed. Basically, there is an association between an N:N relationship with another entity. For better user understanding, we keep the same presentation pattern using colored labels that indicate the mapping detected type. Figures 22 and 23 show a fragment of this specific relationship type mapped in our tool based on the model described at the beginning of this section.

Figure 22 – Diagram snippet with a ternary relationship.



Source – Author.

Figure 23 – Logical model snippet with a ternary relationship.

Source – Author.

Figure 24 – SQL model snippet with a ternary relationship.

```sql
-- Table: GROUPS
-- DROP TABLE GROUPS;
CREATE TABLE IF NOT EXISTS groups (
    idgroup INTEGER NOT NULL,
    title VARCHAR (255) NOT NULL,
    description VARCHAR (255) NOT NULL,
    CONSTRAINT pk_groups PRIMARY KEY (idgroup)
);

-- Table: ROLES
-- DROP TABLE ROLES;
CREATE TABLE IF NOT EXISTS roles (
    idrole INTEGER NOT NULL,
    description VARCHAR (255) NOT NULL,
    permissionlevel INTEGER NOT NULL,
    CONSTRAINT pk_roles PRIMARY KEY (idrole)
);

-- Table: BELONGSTOGROUP
-- DROP TABLE BELONGSTOGROUP;
CREATE TABLE IF NOT EXISTS belongstogroup (
    user_iduser INTEGER NOT NULL,
    groups_idgroup INTEGER NOT NULL,
    CONSTRAINT pk_belongstogroup PRIMARY KEY (user_iduser, groups_idgroup)
);


-- Table: ROLEINTHEGROUP
-- DROP TABLE ROLEINTHEGROUP;
CREATE TABLE IF NOT EXISTS roleinthegroup (
    idrole INTEGER NOT NULL,
    user_iduser INTEGER NOT NULL,
    groups_idgroup INTEGER NOT NULL,
    CONSTRAINT pk_roleinthegroup PRIMARY KEY (idrole, user_iduser, groups_idgroup)
);

-- GENERALIZATION
ALTER TABLE ONLY public.people ADD CONSTRAINT fk_gen_people_user FOREIGN KEY (idUser) REFERENCES public.user (idUser);
-- GENERALIZATION
ALTER TABLE ONLY public.organization ADD CONSTRAINT fk_gen_organization_user FOREIGN KEY (idUser) REFERENCES public.user (idUser);
-- [NtoN]: Relationship in which BOTH ENTITIES have NO GENERALIZATION
ALTER TABLE ONLY public.belongstogroup ADD CONSTRAINT fk_belongstogroup_user FOREIGN KEY (user_iduser) REFERENCES public.user (iduser);
-- [NtoN]: Relationship in which BOTH ENTITIES have NO GENERALIZATION
ALTER TABLE ONLY public.belongstogroup ADD CONSTRAINT fk_belongstogroup_groups FOREIGN KEY (groups_idgroup) REFERENCES public.groups (idgroup);
-- [Ternary]: Associative relationship
ALTER TABLE ONLY public.roleinthegroup ADD CONSTRAINT fk_roleinthegroup_roles FOREIGN KEY (idrole) REFERENCES public.roles (idrole);
ALTER TABLE ONLY public.roleinthegroup ADD CONSTRAINT fk_roleinthegroup_belongstogroup FOREIGN KEY (user_iduser, groups_idgroup) REFERENCES public.belongstogroup (user_iduser, groups_idgroup);
```

Source – Author.

Finally, Figure 24 shows the mapping to the SQL file also follows the same pattern, generating the `Alter Table` commands after defining the structure of the entities. Something relevant to report is that firstly the SQL models were being created with the foreign key constraints being declared already in the bodies of the tables. It was done because we believed that facilitates the readability and the understanding of DDL. However, it was necessary to re-evaluate this strategy since the generated SQL codes were not being executed successfully on the target platforms. This happened because it became difficult to predict when a restriction indicated a relationship with some attribute of an entity not yet created. We tried to implement ways to generate the correct order, despite due to the complexity that presented itself, we adhered to the creation of restrictions via `Alter Table` commands at the end of the file, thus making them executable in the DBMSs.

5.2 Chapter Lessons

In this chapter, we have laid out the requirements, design decisions, and architecture for the developed solution. We also presented an overview of the stable version that was tested in subsequent experimental evaluations. Furthermore, we described an example of use with the respective artifacts produced by the generators.

Again, we reinforce the importance of the Xtext community formed by a group of developers who helped, directly and indirectly, the implementation of a large part of the presented tool. Through interactions with these developers, our proposal matured, even as we acquired a range of technical and personal skills (soft and hard skills) that decisively helped in the optimization process of the DSL and the infrastructure code provided by the framework.

We hope that the developed modeling tool can collaborate with the teaching-learning process, providing an easy-to-understand grammar and ease of use facilitated by the Eclipse IDE integration. Finally, to achieve this objective, the experimental evaluations were duly planned, executed and analyzed, taking into account the characteristics that we proposed to offer.

6 EXPERIMENTAL EVALUATIONS

In this chapter, we present the research questions and hypotheses investigated during the evaluation of our proposal. In addition, we present the experimental design adopted and its conduction. Section 6.1 presents the first evaluation of ERText, which focuses on the tool features assessment. Section 6.2 describes the execution of two other experiments carried out in December 2021 and February 2022, which also included the evaluation of artifacts generated by the tools and learning aspects. In the end, Section 6.3 draws the chapter's lessons.

6.1 Tool Features - Experiment 1

This section presents all the planning, conduction, the results obtained discussion, threats to validity of the study, and the conclusions regarding the preliminary evaluation carried out to evaluate the first release of a stable prototype of the developed tool.

6.1.1 Planning

We carried out this preliminary evaluation from the replication of the experimental protocol performed in a previous study (LOPES, 2019), which aimed to obtain evidence of the comparison from two approaches for modeling relational databases, one graphical and the other textual. We specified the following treatments:

i. Control treatment: the brModelo tool (graphical approach) and;

ii. Experimental treatment: the ERtext tool (textual approach).

This replication goal is to evaluate the feasibility of using a textual approach to support the teaching-learning process of conceptual modeling relational databases.

Context: We characterized the context of the experiment according to four dimensions:

i. Process: Process: it used an *in-vitro* approach since we performed the tasks under controlled conditions;

ii. Subjects: undergrad students in Computer Science (CS) and Software Engineering (SE) programs;

iii. Reality: the experiment addressed a real problem, *i.e.* the difference in the effort spent by subjects in the conceptual modeling of relational databases;

iv. Generality: we inserted this evaluation in a specific context involving database modeling students. However, we can replicate the general ideas of this experiment in another set of subjects, approaches, or DSLs that support database designing.

Research Questions (RQs): Regards the discussion of controlled experiment results, we decided to formulate four RQs related to the activities performed in the protocol execution.

- **RQ1.** Which approach requires the most effort spent on average during the modeling activity?
- **RQ2.** What is the quality level of the models produced using the graphical and textual approaches?
- **RQ3.** What is the subject's perception regarding the Perceived Ease of Use (PEoU) and Perceived Usefulness (PU) of the proposed DSL?
- **RQ4.** What is the subject's assessment concerning the representation of the ER modeling builders supported in the proposed DSL?

Hypotheses Formulation: We have taken into account the first two RQs. Regarding **RQ1.** the average effort spent required using each approach, our scientific hypotheses are as follows:

- **Null hypothesis:** $H_0 : \mu Time_T = \mu Time_G$: There is no difference in the measure of average effort spent between textual and graphical approaches during conceptual modeling.
- **Alternative hypothesis:** $H_1 : \mu Time_T \neq \mu Time_G$: There is a significant difference in the measure of average effort spent between textual and graphical approaches during conceptual modeling.

Regarding **RQ2.** in the same way, we stated two-sided hypotheses that measure the modeling effectiveness between textual and graphical approaches during conceptual modeling.

- **Null hypothesis:** $H_0 : \mu Effectiveness_T = \mu Effectiveness_G$: There is no difference in effectiveness measures between textual and graphical approaches during conceptual modeling.
- **Alternative hypothesis:** $H_1 : \mu Effectiveness_T \neq \mu Effectiveness_G$: There is a significant difference in effectiveness measures between textual and graphical approaches during conceptual modeling.

Statistical Methods: Unlike the first experiment, this replication included a change in the statistical methods adopted. Previously, we have used the Shapiro-Wilk normality test and the paired T-test for dependent samples. Hence, this happened because the sample was smaller (27) than 30 elements. However, the literature recommends alternative tests for samples equal to or greater than this quantity (TRIOLA, 2018).

For the effort, we chose the Kolmogorov-Smirnov test to verify normality and the Wilcoxon Signed-Rank test for paired samples to investigate the hypotheses using the time spent metric. In short, the Kolmogorov-Smirnov two-sample test is used to test the agreement between two cumulative distributions. The null hypothesis in this method states that there is no difference between the two distributions. A simplified version of the method is given in the Equation 6.1:

$$D = Max|F_{n1}(X) - F_{n2}(X)| \tag{6.1}$$

Where *n1* and *n2* are the values observed in the distributions, respectively, and the ***D*** represents the large maximum deviation indicating towards a difference between the two sample distributions.

Furthermore, the Wilcoxon Signed-Rank test for dependent samples compares a sample median against a hypothetical median. Initially, the hypothetical mean, μ_0 , is subtracted from each data value. Then, we classified the values according to their absolute values. Next, we calculated the sum of the positive ranks Sp and the sum of the negative ranks Sn. In the statistical test, W_R is the minimum of Sp and Sn. With this, it is possible to calculate the mean and standard deviation of W_R using the simplified version of the formulas shown below in Equations 6.2 and 6.3, where t represents the number of times the i^th value occurs. Then, we finally calculated the *z-value* using Equation 6.4.

$$^{\mu}W_R = \frac{n(n+1)}{4} \tag{6.2}$$

$$^{\sigma}W_R = \frac{n(n+1)(2n+1)}{24} - \frac{\Sigma\, t^3 \; - \; \Sigma\, t}{48} \tag{6.3}$$

$$Z_W = \frac{W_R \; - \; ^{\mu}W_R}{^{\sigma}W_R} \tag{6.4}$$

For the effectiveness tests, we have adopted the same statistical methods despite instead of using the time spent metric, another measure was necessary. Thus, we have performed the F1 calculations, deriving from the harmonic mean of precision and recall metrics, for each model produced in both approaches. The F1 calculation (DERCZYNSKI, 2016) takes into account variables known as *True Positives*, *False Positives*, and *False Negatives.*

True Positives (TP): Number of elements correctly modeled using the approach.

False Positives (FP): Number of elements incorrectly modeled using the approach.

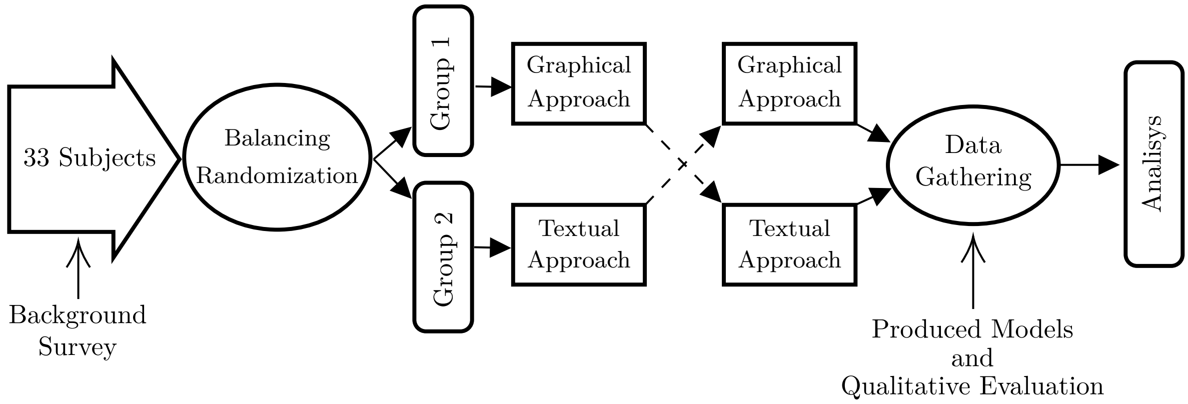
False Negatives (FN): Number of elements not modeled using the approach.

From the identification of the variables, it is then possible to calculate the *Precision*, *Recall*, and *F1* of each model according to these formulas:

Precision (PR): $\frac{TP}{TP + FP}$ ***Recall (RE)***: $\frac{TP}{TP + FN}$ ***F1-Score (F1)***: $\frac{2 * (PR * RE)}{PR + RE}$

Experiment Design: Finally, Figure 25 presents the design of the controlled experiment performed. We followed the design of one factor with two treatments, where we blocked, balanced, and randomized the subjects, which carried out both treatments, featuring a paired comparison design.

Figure 25 – Experiment design.



Source – Author.

6.1.2 Conduction

Preparation: Initially, we held remote meetings between the researchers involved to define the planning and the adopted operation mode in response to the current exception scenario due to the worldwide pandemic. As a result, we defined activities adapted from the first experiment conducted in person. Hence, to capture a significant sample for the object of study, we decided to contact the professors responsible for teaching, in the first half of 2021, two courses of different undergraduate courses: Database (SE) and Database I (CS). Once the initial objectives aligned, the collaborating teachers disclosed the profile questionnaires (Google Forms) to the subjects.

We reused the four (4) instruments from the original experiment. The first two were modeling problems with similar levels of complexity, while the last two were qualitative assessments. For respecting the health security protocols required (social distancing), we decided to carry out the activities remotely. For this purpose, we prepared a virtual machine with the tools installed, even as the instruments and supporting materials. This virtual machine had accessed on the university's computers by the subjects using their institutional credentials.

Execution: The profile form also served as a term of participation since the presence in the experiment was voluntary. With this information, we have randomized the subjects to define the groups. For the record, randomization was performed with a form with six (6) questions with associated weights. From these answers, the score of each participant was calculated and then, the random distribution into two sets took place, trying to make both groups as homogeneous as possible, based on the overall score. In the end, we did not find wide discrepancies among the subjects' knowledge levels, demonstrating a homogeneous sample in general. On the experiment day, the first activity that we performed was a brief initial presentation. Then, the training phase for the participants began. During this phase, we have presented the two database modeling tools used in the experiment, providing an overview of the operation and answering possible questions that arose. The training phase included watching videos to demonstrate both tools: brModelo, and our proposal, respectively.

Then, we start the modeling phase of the proposed problems. All subjects accessed the virtual machines with the problems provided in PDF documents. When starting Instrument 1, we informed which tool each participant should develop the solution, thus respecting the group of which one was part. We asked the subjects to write down the start and end times of the tasks for each instrument they performed. We stipulated no time limit for completion according to the subjects who completed the modeling task. We asked them to comply with the guidelines included in the support material for saving the generated artifacts. With the models saved, we informed the subjects that they should move on to the next task described in Instrument 2, although it was necessary to use the reverse approach to the one they had initially used.

Since the subjects concluded the instruments of the data modeling problems, we delivered the qualitative assessment instruments. As the subjects had completed the tasks, we had thanked and released them. With the conclusion of the experiment by 33 subjects, we closed the evaluation and performed the stage of result analysis.

6.1.3 Results Analysis

We have performed all Kolmogorov-Smirnov and Wilcoxon Signed-Rank calculations with the support of the R language and the Gnumeric software. In parallel with the validation of a specialist in the statistics field and the aid of literature (TRIOLA, 2018).

Effort: To answer **RQ1.**, we have extracted the execution times from the instruments. From the gross amount of the execution times, we have calculated the difference to perform the Kolmogorov-Smirnov normality test. Because it is a statistical test, this technique has the product of measuring the p-value. For this test, we adopted a significance level of $\alpha = 5\%$. It means that the p-value is less than 5%, then we rejected the hypothesis that the distribution is normal.

After calculations with the set of time differences, we reached a p-value of 0.26218. As p-value $> \alpha$, we do not reject the null hypothesis, thus concluding that the data are normally distributed. In other words, the difference between the data sample and the normal distribution is not large enough to be statistically significant. It is important to note that the higher the p-value, the more it supports a null hypothesis. In the case of the result obtained, the chance of type 1 error (rejecting a correct null hypothesis) is very high and translates into 26.21% (0.26218). Once we performed the normality tests on the sample, we carried out the hypothesis test of the average effort regarding **RQ1.** In the Wilcoxon Signed-Rank test for dependent samples, we used a significance level of $\alpha = 5\%$, with which we reached a measure of 0.77948 for the p-value. Because it is a two-tailed test, *i.e.*, it includes equality in its null hypothesis, this p-value shows not enough evidence to guarantee the rejection of the statement of $H_0 : \mu Time_G = \mu Time_T$. Therefore, we do not reject the null hypothesis that the approaches have no difference in average efforts since, according to the test, this difference is not statistically significant. Figure 26 displays a box plot with the variation of data observed through these data.

Effectiveness: To answer **RQ2.**, regarding the approaches' use effectiveness, we evaluated the artifacts produced by the subjects according to the established reference models[1]. The indicated repository contains all reference models and models produced by the subjects. In any case, in Appendix D of this study, the instruments, reference models used, and two examples modeled by a participant are shown as an example. For the record, the reference models basically had a certain number of elements (entities, attributes, relationships, etc.), which were compared to the models produced by the subjects. In this evaluation, we used the F1 applied to the pattern recognition and the information retrieval areas. F1 represents the combination of the observed accuracy and the recallability of a result concerning a reference. By definition, this combination refers to Precision and Recall metrics. On the one hand, Precision is the ratio of the recovered instances by relevance. On the other hand, Recall is the ratio of the relevant instances to the recovered.

In addition, we performed the Kolmogorov-Smirnov normality test to F1 for each model. After calculations with the set of differences in F1 for each model, we reached a p-value of 0.45459. With this test result, the chance of type 1 error (rejecting a correct null hypothesis) can be very high, and hence we can translate it into 45.45% (0.45459). As the p-value $> \alpha$, we do not reject the null hypothesis, thus realizing that the data are a normal distribution. After we tested the sample for normality, we tested the second hypothesis defined in this experiment.

This time, in the Wilcoxon Signed-Rank test for dependent samples, we used again a significance level of $\alpha = 5\%$, with which we reached a measure of 0.00197 for the p-value. By the original statement including equality, also characterizing this test as two-tailed, we concluded that the calculated p-value demonstrated enough evidence to guarantee the statement rejection of the primary null hypothesis, denoted as $H_0 : \mu Effectiveness_G = \mu Effectiveness_T$. Therefore, we rejected the null hypothesis that the approaches have equal effectiveness since, according to the statistical test, the average difference of F1 among treatments is statistically significant. Table 7 shows average measures of the

[1] Available at: <https://doi.org/10.5281/zenodo.5454378>

evaluated values and also provides the possibility to carry out a dispersion analysis. Based on these data, it was possible to verify that the textual approach has an advantage on average.

Table 7 – Measures of the conceptual data models produced in the experiment.

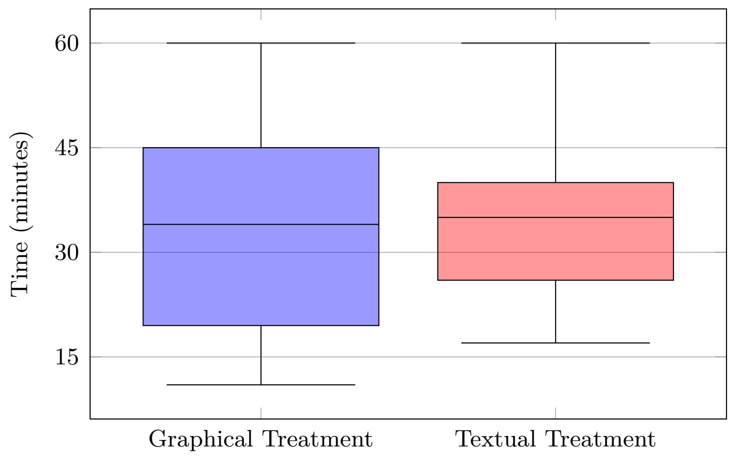
	Graphical Treatment					Textual Treatment				
Measure	**MI**	**RI**	**P(%)**	**R(%)**	**F1(%)**	**MI**	**RI**	**P(%)**	**R(%)**	**F1(%)**
Maximum	47.00	36.00	96.67	92.31	88.00	56.00	46.00	97.22	97.87	91.36
3°Quartile	31.00	28.00	92.31	63.04	76.32	34.00	31.00	94.74	75.00	82.86
Median	26.00	24.00	88.89	56.41	68.85	30.00	29.00	90.63	65.96	74.63
Average	27.52	24.12	87.69	57.96	69.13	30.88	27.45	89.49	63.65	73.16
1°Quartile	23.00	20.00	84.21	50.00	62.50	26.00	23.00	87.88	51.06	63.01
Minimum	18.00	16.00	72.73	41.03	52.46	19.00	15.00	72.73	31.91	45.45
Variance	35.58	28.65	42.87	143.35	78.93	63.32	39.76	42.59	259.85	133.58
SD	5.97	5.35	6.55	11.97	8.88	7.96	6.31	6.53	16.12	11.56

Legend: MI = Modeled Items; RI = Relevant Items; P = Precision; R = Recall; F1 = F1-Score; SD = Standard Deviation.

Source – Author.

Figure 27 box plot graph displays the F1-Score for each treatment applied. Based on this graph, it is possible to verify the result obtained in the hypothesis test because the data dispersion does present a significant difference between the approaches.

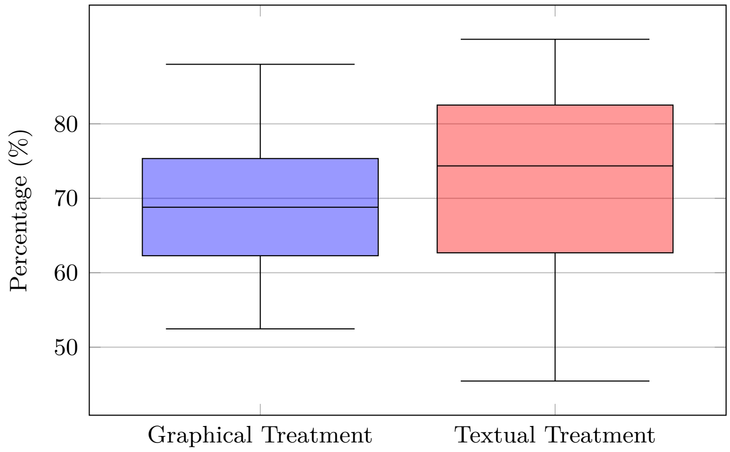
Figure 26 – Box-plot - Effort per treatments.



Source – Author.

Qualitative Evaluation: We took place with the analysis of the two instruments applied after the modeling tasks. We have used the first instrument according to the TAM model (DAVIS, 1989; PERSICO; MANCA; POZZI, 2014) to answer the **RQ3.** regarding the PEoU and PU. It occurred through the quality attributes selection described in ISO/IEC 25010. For this to happen, we established a Likert (LIKERT, 1985) scale from one to six points to measure the subjects' agreement level in the face of the statements exposed in the form. We have chosen an even number of alternatives to avoid possible

Figure 27 – Box-plot - F1 per treatments.



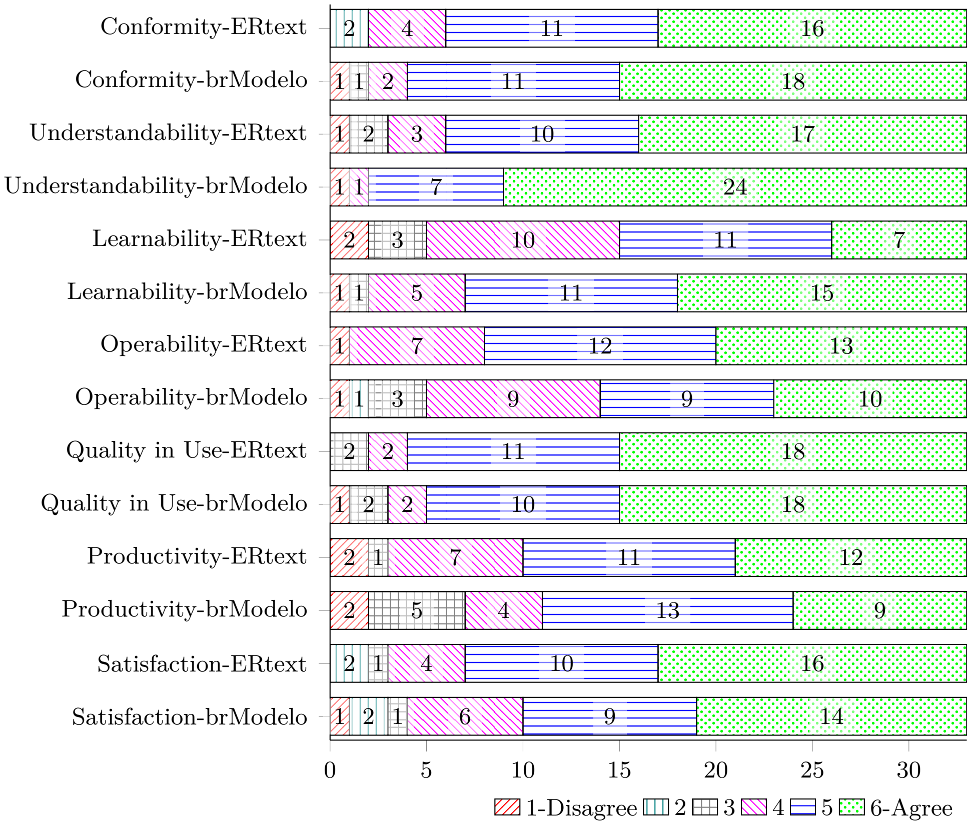
Source – Author.

neutral responses. Thus, we grouped the seven (7) quality attributes into three (3) defined categories as follows:

- **Functionality**
  - *Conformity*: ability level to which the software achieves specified goals with functional completeness, correctness, and appropriateness related to their functionalities.
- **Usability**
  - *Understandability*: ability level to which users can recognize whether the software is appropriate for their needs;
  - *Learnability*: ability level to which the software enables the user to learn how to use it with effectiveness, efficiency in emergencies;
  - *Operability*: ability level to which the software is easy to operate and control, being also appropriate to use.
- **Quality in Use**
  - *Quality in Use*: ability level to which the software achieves specified goals with effectiveness and efficiency with their users when used in specified contexts;
  - *Productivity*: ability level to which the software to achieve specified goals with time-behavior, resources utilization, and capacity when performing its functions to meet requirements;
  - *Satisfaction*: ability level to which the software achieves specified goals with usefulness, trust, pleasure, and comfort with their users when used in specified contexts.

After summarizing the results, we observed a good acceptance by the subjects for the ERtext tool developed in this work. Figure 28 synthesizes the responses received for each quality attribute, showing a certain degree of similarity in the subjects' perception during the application of the treatments. It is worth emphasizing a point that is the positive responses set concerning the *Productivity* and *Operability* since the hypothesis test related to the effort the treatments demonstrated a similar need for execution time. According to the evaluations received, the most evidence about disadvantages of ERtext was mainly concerning the quality attributes of *Understandability* and *Learnability*.

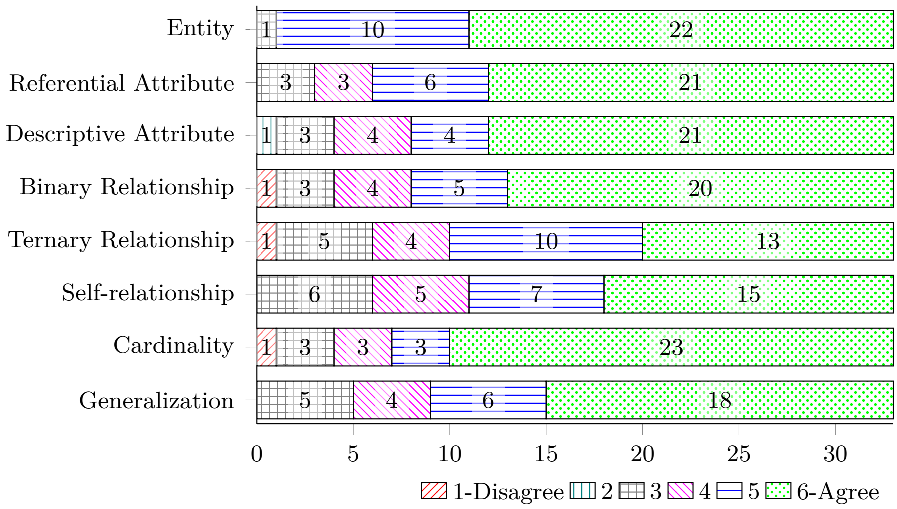
Figure 28 – Quality attributes per treatments.



Source – Author.

Concerning **RQ4.** on the assessment of DSL designers, we analyzed the artifacts of the 2nd qualitative assessment instrument. This instrument listed the 8 ER modeling builders covered by DSL, arranged with a Likert (LIKERT, 1985) scale from one to six points. Again, we have chosen an even number on the scale to avoid neutral responses that could lead to a more subjective bias. Figure 29 compiles all the responses received. The builders related to Entities, Referential Attributes, Descriptive Attributes, and Cardinality were the best evaluated. In contrast, all builders obtained at least one disagreement, in which we can highlight the most disagreeing evaluations related to the current representations of ternary relationship and self-relationship, unfortunately. All data collected and used for statistical tests can be accessed in a public repository available on the Zenodo[2] platform.

Figure 29 – Evaluation of DSL designers.



Source – Author.

6.1.4 Threats to Validity

In empirical studies, it is necessary to analyze and discuss the threats to validity also the strategies used to mitigate them. For this list of possible threats, we adopted the classification schema proposed by Cook and Campbell (COOK; CAMPBELL, 1979). These threats followed the proposed pattern, and we divided them into four (4) categories: construct validity, internal validity, external validity, and conclusion validity.

Construct Validity: Threats to construct validity concerns the experiment design and social factors.

i. *Inappropriate Pre-operational Explanation*: is related to the fact that the experiment did not have the objective of the artifacts sufficiently defined before translation into measures or treatments. To mitigate this threat, we compared the effort of each approach as well as their effectiveness carried out according to the Precision and Recall metrics;

ii. *Interaction of Different Treatments*: if the subjects are involved in one more study, the controls of the different studies can change and reverberate in the final results. To avoid this, we followed a paired design and, therefore, all subjects performed both treatments. However, we have not observed learning issues during the execution of activities. We can verify this through the normality of the analyzed distributions

[2] Available at: <https://doi.org/10.5281/zenodo.5454378>

of both samples: effort and effectiveness, demonstrating that the results remained similar as a whole with a low variation;

iii. *Hypothesis Prediction*: when subjects participate in an experiment, they may try to find out what the objective or intended experiment result is. This threat can bias the behavior, positively or negatively, depending on the anticipated hypothesis. To mitigate this threat, we do not inform the subjects about further details of the experiment;

iv. We do not inform the subjects about further details of the experiment to mitigate the bias of the behavior, which can affect positively or negatively, depending on the anticipated hypothesis.

Internal Validity: Threats to internal validity are the influences that can affect independent variables in correlation to causality without the researcher's knowledge.

i. *History*: there is a risk when a specific period influences the experiment's performance. Due to the coronavirus pandemic, we conducted the entire evaluation in a controlled environment providing remote access for the subjects. In addition, to soothe this threat, we carried out the entire process in March when, in general, students are not necessarily overwhelmed with academic activities;

ii. *Maturation*: occurs concerning the subjects react in different ways over time. Examples are when subjects are affected negatively (tiredness or boredom) or positively (learning) during the experiment execution. To alleviate this threat, we informed subjects from the beginning that they could terminate their participation at any time, without any penalty;

iii. *Tests*: if we have repeated the tests, the subjects can respond differently at other times, *i.e.*, they know how to perform the test. If there is a need to become familiar with the tests, we do not return the test results to the participant to not subsidize intentional learning. There was no need for repetition of activities since they were performed once per participant in each treatment;

iv. *Instrumentation*: is related to the artifacts used to carry out the experiment study, such as data collection forms. If the design of instruments is poor, then it negatively affects the experience. To combat this threat, we minimally adapted all artifacts due to the design of the experiment (remote) and previously verified and validated in meetings between the researchers involved in this work. In addition, we already had the first study with results counting 27 participants to validate the planned protocol for this experiment replication;

v. We adapted all artifacts due to the design of the experiment (remote), which we verified and validated previously in meetings among the researchers involved in this work to avoid the instrumentation of artifacts that can affect the execution. In addition, we already had the first study with results counting 27 participants to validate the planned protocol for this experiment replication.

External Validity:

i. We carried out the experimental study with undergrad students of SE and CS programs and soon inserted in the context of using the conceptual database modeling seeking to mitigate the selection of subjects from a significant group for a study area;

ii. We asked the participants to answer the questionnaires post-experiment and, before thanking them, we checked the submission of the instruments to avoid the subjects' interaction with the evaluation artifacts can be affected the preliminary experimental results;

iii. We used the documentation based on templates and traditional models found in database teaching material to soften an unrepresentative configuration and material.

Conclusion Validity:

i. The participants' prior knowledge can also be a threat to the validity of the experiment. For this, we adopted a pre-experiment questionnaire for the classification and subsequent randomization of the subjects within the groups that would perform the modeling activities, seeking to make the two sets of subjects as homogeneous as possible.

ii. Controlling the time of activities is also a threat, since it was necessary for the subjects themselves to inform this due to the pandemic context (remote activities). Trying to mitigate this, we tried to make it clear before and during the execution of the modeling activities that the subjects should inevitably write down the start and end times to submit to us at the end.

iii. We adopted some statistical methods such as the Kolmogorov-Smirnov normality test and the Wilcoxon Signed-Rank Test as a hypothesis test for dependent samples. We use them to try to mitigate the low statistical power of our preliminary experimental results;

iv. We adopted objective measurements that did not depend on subjective judgment (effort spent, measured in time, and F1) to mitigate the measurement reliability threat. On the other hand, the metrics used for the qualitative evaluation still served as a complementary input in the results obtained discussion, alongside the indication of possible points of improvement in our proposal;

v. We created copies of a virtual machine that all subjects accessed to guarantee that we experimented in a controlled environment.

6.2 Learning and Tool Features - Experiment 2 and 3

The goal of the final empirical evaluations was to perform dynamics similar to those described in Section 6.1. However, this time the goal was to evaluate beyond the language, the quality of artifacts produced, the usability, and the subjects' experience. Therefore, the intention was to compare the approaches (textual vs graphical) in deeper evaluations of the new features developed in our tool. Hence, this implied the creation and inclusion of new instruments to verify the generated artifacts.

There was prior planning for using a group composed of academics from a database design and modeling class at UNIPAMPA. This group started classes in November 2021 and ended in March 2022, initially having fifty (50) students enrolled. This section describes the highlight of the two experiments performed - Experiment 2 (EX2) and Experiment 3 (EX3) - their execution, results obtained, and discussion.

6.2.1 Experiment 2

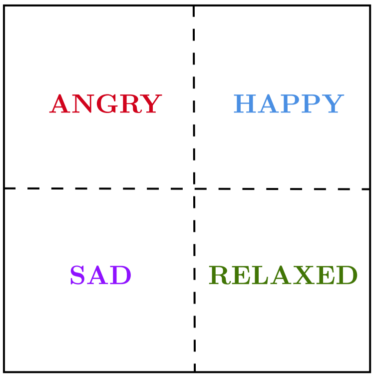
In general, the performed EX2 replicated the previously described protocol again, that is, the conduction underwent a preparation almost *ipsis litteris* as described in Section 6.1.2. We used the same treatments, and the same research questions and hypotheses. However, it was necessary to adopt different statistical methods due to the size of the samples.

We made the invitations to a Software Engineering class and, given the opportunity, it was extended to another, this one from the Computer Science course, totaling fifty-four (54) respondents. Hence, in this new execution, we obtained less than half of the subjects who agreed to participate, i.e., totaling twenty-five (25) respondents. These subjects performed the tasks remotely in an environment previously prepared by the researchers involved.

In addition to reusing the four (4) previous assessment instruments, we added a new fifth instrument containing a non-verbal method for participants to report their emotions. This method, called Emocards, is based on the concept of Russell's Circumplex of Affection (DESMET; OVERBEEKE; TAX, 2001). While Russell's Circumplex divides human emotions into different quadrants, the Emocards method extends it by adding sixteen (16) graphical representations of human faces. These representations are male/female pairs, which are then associated with each of the octants contained in Russell's Circumplex quadrants.

The concept of the circumplex and its divisions is that every two (2) octants form a quadrant that defines a state of mind. The set of octants 1 and 2 forms a quadrant that concerns feelings of happiness. The set of octants 3 and 4 defines a quadrant related to feelings of relaxation. The set of octants 5 and 6 forms a quadrant that contains feelings of sadness. Finally, octants 7 and 8 constitute the quadrant that encompasses feelings of irritability. Figure 30 shows the four (4) states of mind predicted in Russel's Circumplex.

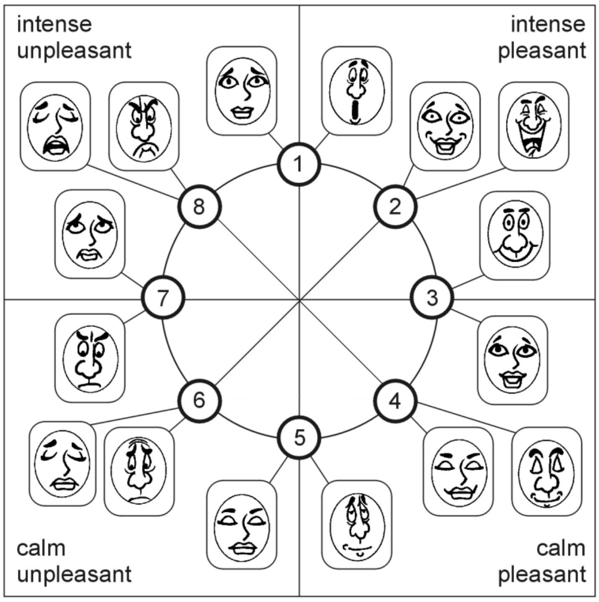
Figure 30 – The four (4) quadrants with the states of mind of Russell's Circumplex.



Source – Author.

Figure 31 presents the Emocards method inserted in Russell's Circumplex of Affection, showing the eight (8) octants distributed in the four (4) states of mind and with the sixteen (16) graphic representations associated with them.

Figure 31 – Emocards method and its eight categories within Russell's Circumplex.



Source – (REIJNEVELD et al., 2003).

The execution took place in December 2021, still without offering access to the participants so that they could evaluate the artifacts produced by the generators implemented in both tools. On the one hand, regarding the ERtext generators, we deactivated them within the controlled environment that we distributed. On the other hand, for the brModelo generators, we only advised that it was not required to transform the conceptual model into a logical or physical one. Hence, this has happened because we tried to replicate the previous experiment to compare the results. Also, given the limited time available, this could add another threat to the results, as it would require more effort to perform the additional tasks.

6.2.1.1 Results Analysis

This section presents the results obtained from the execution of EX2. The data used is available in a public repository[3] on the Zenodo platform. All statistical tests were performed with the support of the R language with the IDE RStudio. The scripts for this experiment are publicly available in a repository[4]. Hence, anyone can run it locally and verify their results beyond also querying all the raw data.

We bore in mind the same previous hypotheses related to the effort (time) needed and effectiveness (quality) achieved by the models created. For the evaluation regarding the effort, once the sample had less than thirty (30) elements, we used the Shapiro-Wilk normality test. Otherside the paired T-Test for dependent samples, when we considered the effort spent collected during the execution of the modeling activities.

The Shapiro-Wilk method tests the null hypothesis by calculating whether is a normal distribution revealed by the value W. Then we can verify in the test table[5] whether the hypothesis is rejected or accepted. The calculation of the Shapiro-Wilk method is given according to the formula of Equation 6.5:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n} (x_i - \overline{x})} \tag{6.5}$$

A simplified form of the Paired T-Test for dependent samples is given according to the formula of Equation 6.6:

$$t = \frac{m}{s/\sqrt{n}} \tag{6.6}$$

In this formula, $\boldsymbol{m}$ and $\boldsymbol{s}$ are the mean and standard deviation of the difference ($\boldsymbol{d}$), respectively. The $\boldsymbol{n}$ corresponds to the size of $\boldsymbol{d}$, that is, the size of the sample. We used this hypothesis test to compare the means of two related samples, *i.e.*, when you have two values (or pairs of values) for the same sample. However, to compare the means of the two paired data sets, the differences between all pairs had to be calculated first. The alpha significance level (α) used was five percent (5%). We adopted the same statistical methods for the effectiveness tests, but instead of using the effort spent (time) measure on activities, we used another magnitude. Therefore, we performed calculations again for an F1 measure, as described previously in Section 6.1.1.

3 Available at: <https://doi.org/10.5281/zenodo.6416356>.
4 Example available at: <https://github.com/JonnathanRiquelmo/SE-Masters-Thesis/blob/main/hypothesis-tests.r>.
5 Available at: <http://www.uel.br/projects/experimental/pages/arquivos/probabilities_Shapiro.pdf>

From the raw time values, the difference was calculated to be able to perform the Shapiro-Wilk normality test. As it is a statistical test, this technique has as its product the measure of the p-value. A significance level of $\alpha = 5\%$ was adopted for this test. Hence, through this, if the p-value is less than 5% ($p < 0.05$), we must reject the null hypothesis since the distribution is not normal.

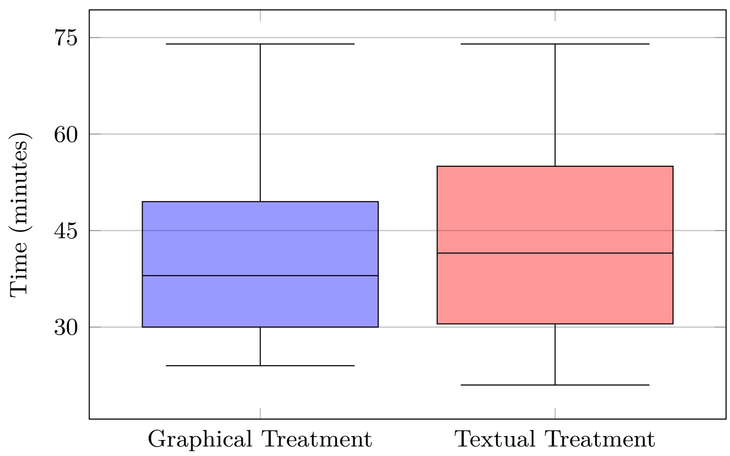
After the calculations with the set of time differences, we reached a p-value of 0.5991. As p-value $>\alpha$, we accepted the null hypothesis, thus concluding that the data are a normal distribution. In other words, the difference between the sample data and the normal distribution is not large enough to be statistically significant.

It is important to note that the higher the p-value, the more it supports a null hypothesis. In the case of the result obtained, the chance of a type 1 error (rejecting a correct null hypothesis) is very high, which we can translate into 59.91% (0.5991). Still, about the normality test, we calculated the value of W resulting 0.968, being within the range of the critical value accepted of 95%. So this means that there is a 95% chance that the sample comes from a population considered normal.

Once we tested the sample for normality, it was possible to test the first hypothesis established in this experiment. In the paired T-test for dependent samples, we used a significance level of $\alpha = 5\%$, which reached a measure of 0.3492 for the p-value.

As it is a two-tailed test, that is, it includes an equality in its null hypothesis, this p-value does not show enough evidence to guarantee the rejection of the statement of $H_0 : \mu Time_G = \mu Time_T$. In short, we considered the average of the values of the graphical approach (brModelo) similar to the textual one (ERtext). In other words, the difference between the brModelo and ERtext means is not large enough to be statistically significant. Figure 32 shows a box-plot with the observed variation of the data obtained.

Figure 32 – Box-plot - Effort per treatments in EX2.



Source – Author.

To assess the hypothesis regarding the effectiveness of using the approaches, we analyzed the artifacts produced by the subjects according to the previously established reference models. As mentioned, we used in this evaluation the F1 measure, this measure from the area of pattern recognition and information retrieval. This measure represents the combination of the accuracy observed and the recallability of a result against a reference.

After obtaining the F1 values of each model, we performed the Shapiro-Wilk normality test. After the calculations with the set of differences of the F1 measure of each model, we reached a p-value of 0.5166. With this result obtained, the chance of a type 1 error (rejecting a correct null hypothesis) can be very high, which we can translate into 51.66% (0.5166).

As the value-$p > \alpha$, we accepted the null hypothesis, thus noting that the data are a normal distribution, *i.e.*, the difference between the sample of data and normal distribution is not large enough to be statistically significant.

After we tested the sample for normality, we performed the test of the second hypothesis defined concerning the effectiveness (quality) of the approaches. This time, in the paired T-test for dependent samples, we used a significance level $\alpha = 5\%$ again, with which we reached a measure of 0.2147 for the p-value.

By the original statement including equality, also characterizing this test as two-tailed, we concluded that the calculated p-value demonstrates that there is not enough evidence to guarantee the rejection of the statement of the original null hypothesis, denoted as $H_0 : \mu Effectiveness_G = \mu Effectiveness_T$. Therefore, we accepted the null hypothesis that the approaches have equal effectiveness because, in accord with the test, the mean difference of the F1 measure between the treatments is not statistically significant. Table 8 presents the basic statistical measures of the values evaluated, providing means for dispersion analysis.

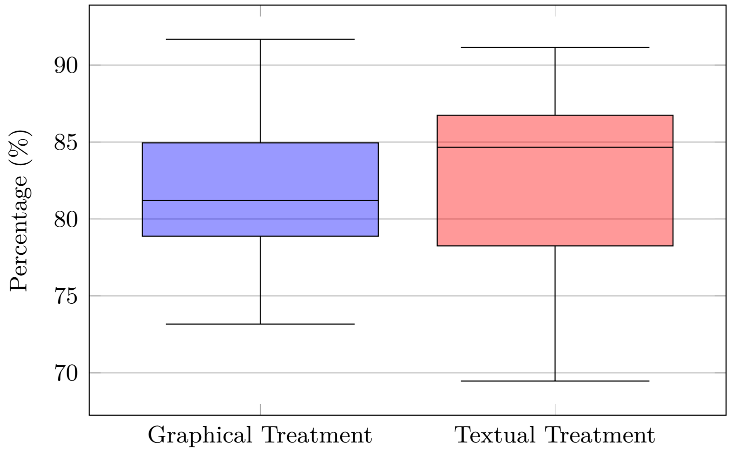
Table 8 – Measures of the conceptual data models produced in EX2.

	Graphical Treatment					**Textual Treatment**				
Measure	**MI**	**RI**	**P(%)**	**R(%)**	**F1(%)**	**MI**	**RI**	**P(%)**	**R(%)**	**F1(%)**
Maximum	49.00	44.00	93.75	92.50	91.67	65.00	45.00	94.74	92.68	91.14
3°Quartile	46.00	39.00	88.89	85.42	86.02	47.00	38.00	89.19	87.80	87.06
Average	41.08	35.20	85.88	78.57	81.74	44.64	36.48	83.24	83.35	82.81
Median	43.00	36.00	86.05	77.50	81.32	44.00	36.00	85.37	83.67	84.71
1°Quartile	36.00	31.00	84.78	75.00	79.12	40.00	34.00	81.82	79.59	78.72
Minimum	27.00	23.00	71.43	57.50	68.66	34.00	32.00	58.46	69.39	69.47
Variance	34.63	23.84	20.21	72.57	26.91	59.59	10.25	102.43	34.72	37.18
SD	5.89	4.88	4.50	8.52	5.19	7.72	3.20	10.12	5.89	6.10

Legend: MI = Modeled Items; RI = Relevant Items; P = Precision;
R = Recall; F1 = F1-Score; SD = Standard Deviation.
Source – Author.

The box-plot in Figure 33 shows a visual representation of the F1 measurement of the treatments applied. Based on this graph, it is possible to verify the result obtained in the hypothesis test, as the data dispersion does not present a high difference between the approaches.

Figure 33 – Box-plot - F1 per treatments in EX2.



Source – Author.

Figure 34 shows the evaluation regarding the quality attributes of the tools. In general, the evaluation results made by the participants were proportionally similar to that obtained in the previous experiment. The highlights this time was the Satisfaction attribute for the textual approach, while the Conformity and Understanding attributes performed better in the graphical one. Another point to highlight is the slightly smaller difference in the Productivity attribute, but still in favor of the textual approach.

The evaluation results referring to the DSL builders of the textual approach, displayed in Figure 35, also followed the same proportion observed in the first experiment. Then leads us to conclude that it is necessary, mainly, to re-execute an evaluation of the representation of the ternary relationships to make them easier to learn, interpret, and use.

Finally, we come to the added instruments trying to verify the state of mind or the emotions caused by the execution of the activities with each approach. We arranged the results of the evaluations in Russell's Circumplex. It is possible to observe that regarding the use of the textual approach with the ERtext tool, there was a greater concentration in the quadrants that concern feelings of happiness (Octants 1 and 2) and relaxation (Octants 3 and 4).

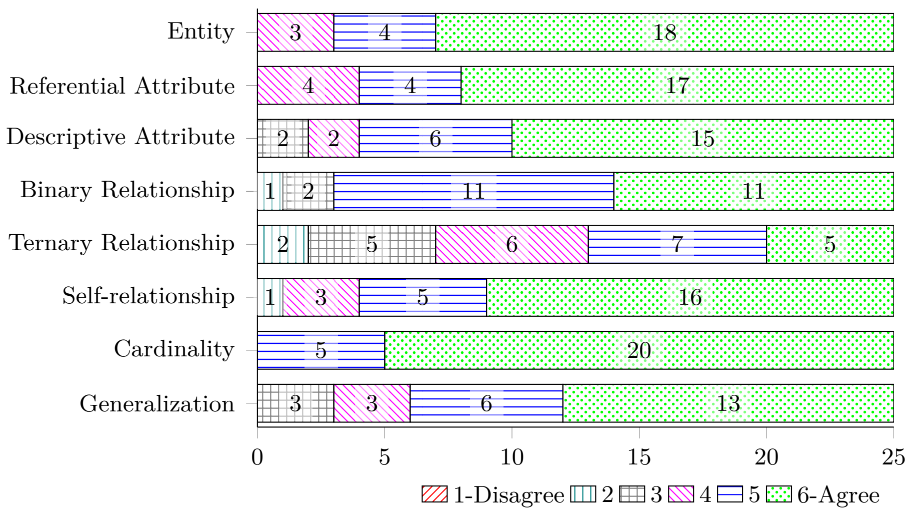
These two quadrants concentrated a total of twenty (20) respondents. Also, there were a distribution of three (3) participants who expressed feelings related to sadness (Octants 5 and 6) and two (2) subjects who demonstrated identification with the Emocards related to the quadrant that concerns emotions of irritability (Octants 7 and 8). Figure 36 presents the distribution of subjects for each octant predicted in the Emocards method, which used the textual approach.

The evaluation of the graphical approach, using the brModelo tool, showed a sim-

Figure 34 – Quality attributes per treatments in EX2.

Source – Author.

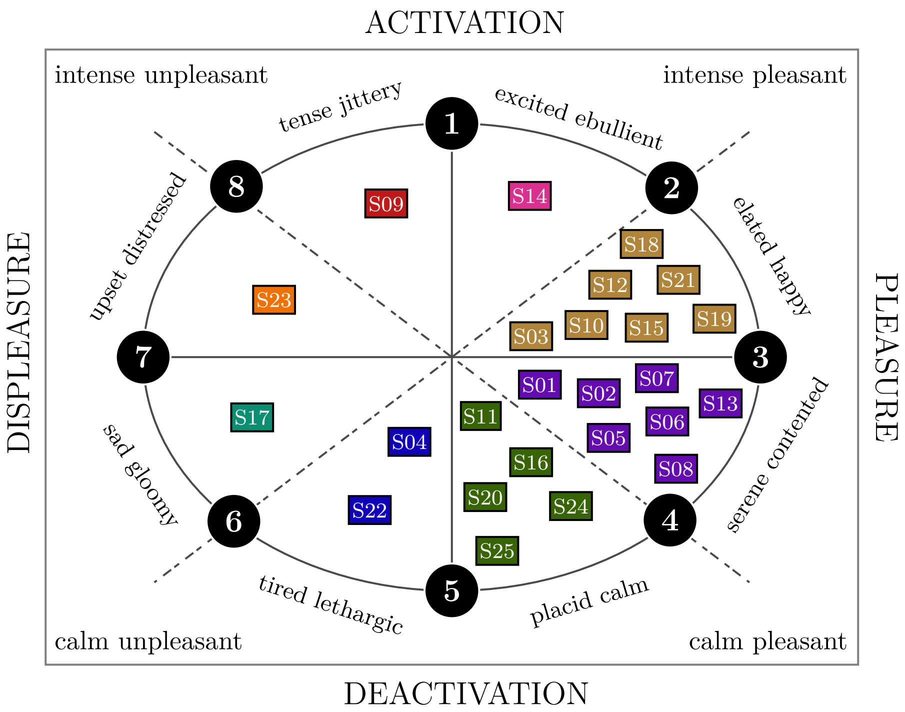
Figure 35 – Evaluation of DSL designers in EX2.



Source – Author.

ilar distribution. For the quadrants that encompass feelings of happiness and relaxation, there were nineteen (19) subjects. In the quadrant that represents feelings related to

Figure 36 – EX2 Emocards - ERtext.



Source – Author.

sadness, there were four (4) subjects, while in the quadrant corresponding to feelings of irritability, there were only two (2) participants. Figure 37 shows the distribution of subjects for each octant predicted in the Emocards method, which used the graphical approach.

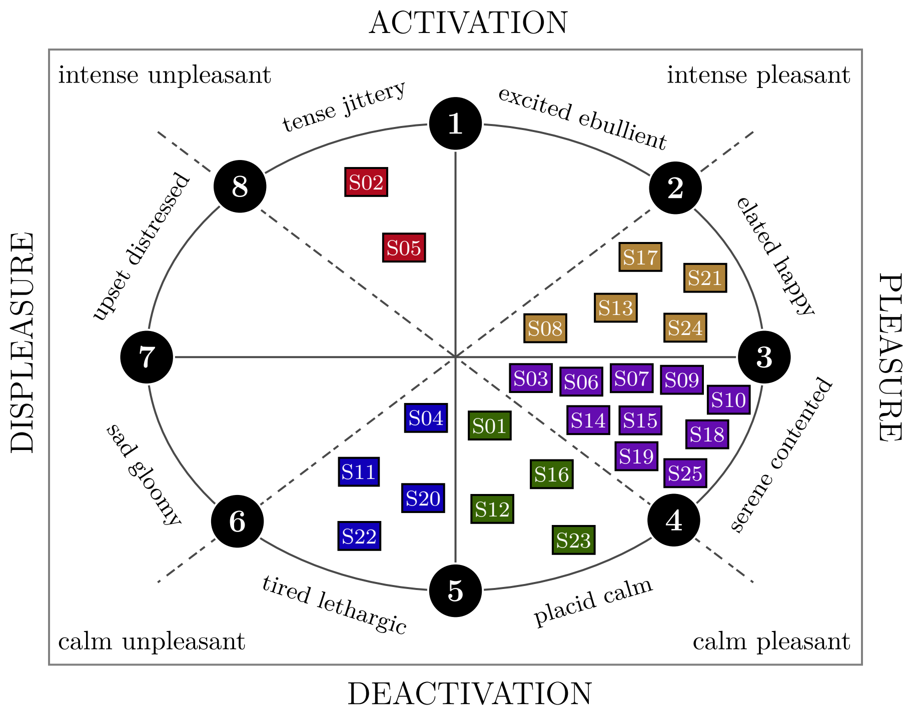
Seeing that these are approaches that, in essence, are opposed, we believe that the similarity of the observed results may be a good sign. Firstly, this indicates that our proposal achieves a near level of satisfaction as the other mature and widely established tool. However, further to check this better, we carried out a third experiment to evaluate the implemented generators, this time qualitatively, since we already have the statistically relevant material that supports our conclusions concerning the hypotheses treated.

6.2.2 Experiment 3

We carried out EX3 in the second half of February 2022. We again invited twenty-five (25) participants from EX2 to participate. There were the acceptance of fifteen (15) subjects. This time, we modified the original protocol again. In this execution, we do not seek to measure the time or quality of the models produced. Likewise, we required the participants to model two (2) simple problems, one with each approach.

The controlled environment was no longer in a virtual machine since a complete plugin for Eclipse was generated and made publicly available. This plugin only depended

Figure 37 – EX2 Emocards - brModelo.



Source – Author.

on the Eclipse IDE and Java installation, regardless of the operating system utilized. In any case, the activities took place remotely when as in the other experiments, the researchers followed the subjects online.

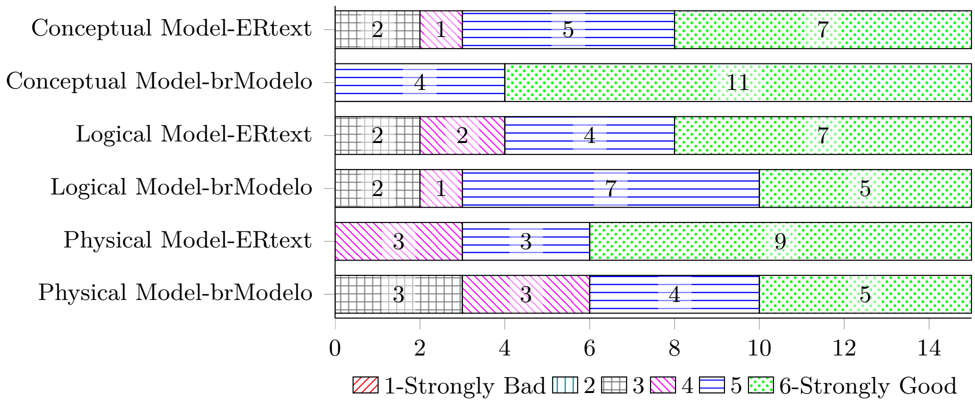
The instruments were also different in this execution since we omitted the builders' assessment instruments and the quality attributes. Our intention was for the subjects to evaluate the artifacts generated by the tools that supported the approaches at the different levels. For this, we used a Likert (LIKERT, 1985) scale to obtain the evaluation of these artifacts generated in each of the modeling levels (conceptual, logical, and physical).

The Emocards method described previously was also reused to understand the state of mind that utilization of the full features in the tools could cause. Beyond, we also added a series of open-ended questions and, finally, a form with a well-known usability scale utilized in systems engineering named System Usability Scale (SUS) (BROOKE, 1995).

6.2.2.1 Results Analysis

Regarding the artifacts evaluation related to the modeling levels, Figure 38 presents a brief of the values obtained. Concerning the classification of the conceptual models produced, there was a better acceptance of the graphical approach in the brModelo tool. In contrast to the textual approach, the proposed ERtext solution received eleven (11) evaluations with a maximum score. In addition to having fewer maximum scores, the textual approach, which totaled seven (7), still had evaluations with a score of two (2), *i.e.*, low considerations. We applied the Likert (LIKERT, 1985) scale ranging from one (1 - the worst rating) to six (6 - the best rating). It is worth noting that while the conceptual model of the graphical approach was just the diagram, the textual one has represented by the textual model and accompanied by a graphic equivalent that used UML elements.

Figure 38 – Evaluation of produced artifacts per treatments.



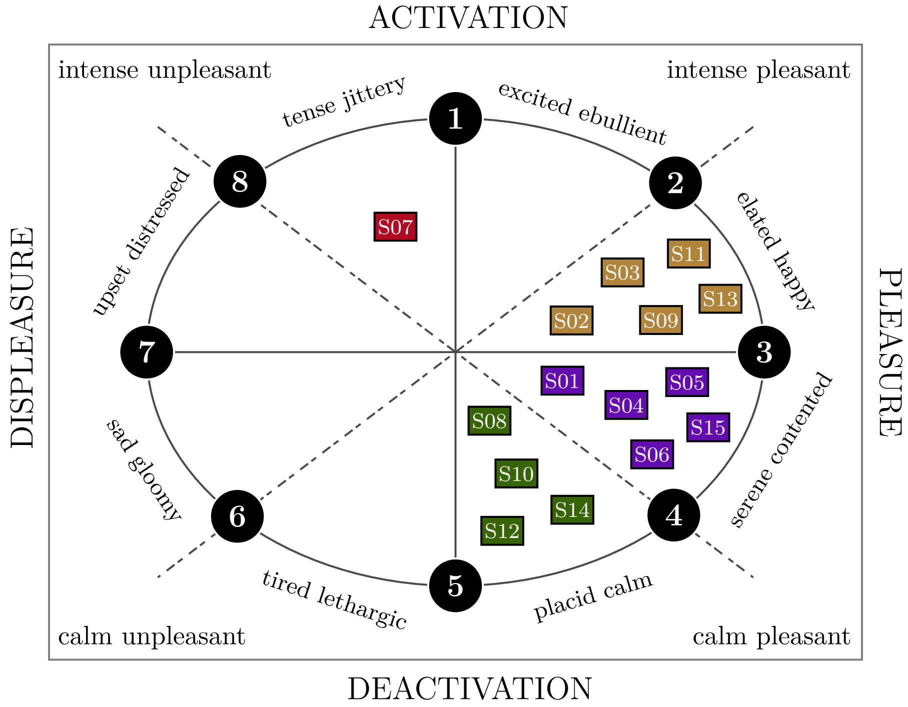
Source – Author.

Respecting the evaluation of the logical models generated from the conceptual models, we recognize the most equivalency but with fewer advantages for the textual approach. Regarding the evaluation of the physical models, we requested that the subjects run the SQL codes generated in a DBMS, specifically PostgreSQL since it has been used recurrently in the Databases' classes.

The results of the physical models showed a slight advantage for the textual approach, and the graphical one also received evaluations with a score (2), that is, considering the models with low quality. It is relevant to point out that at each evaluation of the artifacts, it was still openly asked if there were criticisms, compliments, doubts, or suggestions for improvement regarding the observed representations. Then the collected responses would be used for a qualitative analysis using qualitative coding. As for the evaluation using the Emocards method, Figures 39 and 40 show the results. Again, there were very similar results, demonstrating that both tools can arouse similar feelings and states of mind.

In the Emocards related to the textual approach (Figure 39), using ERtext, we received five (5) responses from subjects who expressed being in the quadrant that encompasses feelings of happiness. From another point of view, another nine (9) responses from subjects showed being in the quadrant that involves feelings related to relaxation. Ending, only one (1) participant chose a graphic representation of the human face classified in the quadrant that comprises feelings of irritability.

Figure 39 – EX3 Emocards - ERtext.



Source – Author.

In the Emocards regarding the graphical approach (Figure 40), using the brModelo, we received four (4) responses from subjects who declared themselves in the quadrant regarding feelings of happiness. Another eight (8) subjects demonstrated to be in the quadrant that involves feelings related to relaxation. Also, two (2) participants chose Emocards related to the quadrant that expresses feelings of sadness and only one (1) to the quadrant that includes feelings of irritability.

The SUS form was applied based on its classic format, with ten (10) statements with an associated Likert (LIKERT, 1985) scale. This scale goes from one (1) to five (5). The ten (10) statements that the subjects rated were those shown in the Table 9.

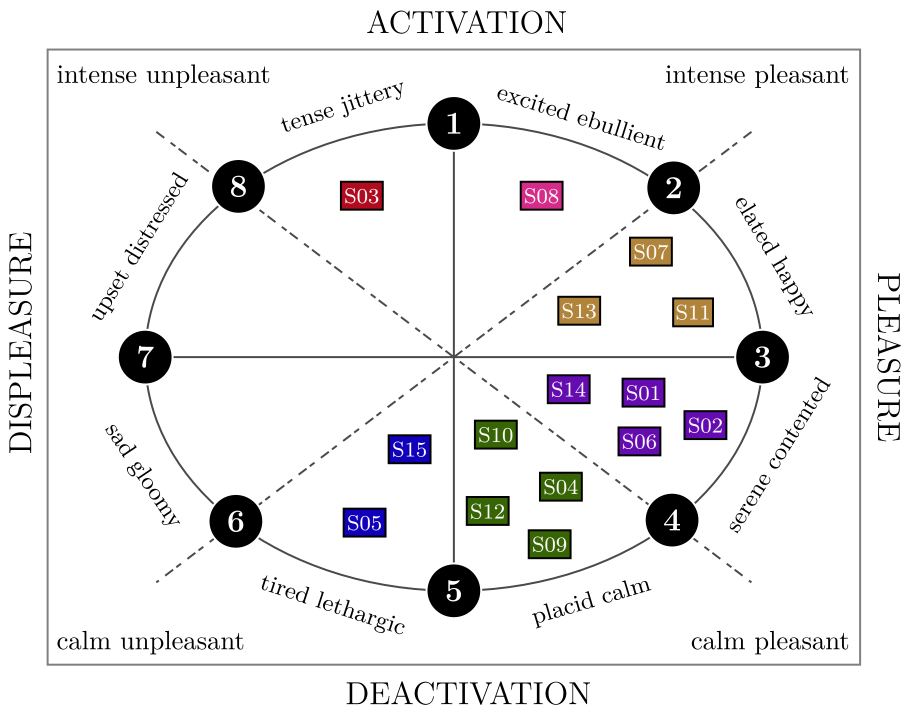
Table 9 – SUS statements.

#	SUS STATEMENTS
01	I think that I would like to use this solution frequently.
02	I found the solution unnecessarily complex.
03	I thought the solution was easy to use.
04	I think that I would need the support of a technical person to be able to use this solution.
05	I found that the various functions in this solution were well integrated.
06	I thought there was too much inconsistency in this solution.
07	I would imagine that most people would learn to use this solution very quickly.
08	I found the solution very cumbersome to use.
09	I felt very confident using the solution.
10	I needed to learn a lot of things before I could get going with this solution.

Source – (BROOKE, 1995).

We can use this method to evaluate products, services, hardware, software, web-

Figure 40 – EX3 Emocards - brModelo.



Source – Author.

sites, applications, and any other type of interface. The criteria this method helps to assess cover are: (i) Effectiveness (can users complete their objectives?); (ii) Efficiency (how much effort and resources are needed for this?); and (iii) Satisfaction (was the experience satisfactory?).

We calculated the score for each statement using a simple equation defined in the method. The odd statements (1, 3, 5, 7, 9) need to be subtracted 1 (one) from the Likert (LIKERT, 1985) scale value chosen by the participant. For example, statement 5 evaluated as 4, so 4−1=3. For even statements (2, 4, 6, 8, 10), the chosen value on the Likert scale must be subtracted from 5. For example, let's say that in statement 4, the value chosen is 3. So we have 5−3=2. With the individual values of each statement calculated, add them all together and multiply by 2.5. The values obtained must then be added and divided by 10, thus arriving at the corresponding final general value.

Table 10 presents the final values obtained. It can see that both approaches reached a maximum of 100, and a minimum of 37.50 for the graphical, while the textual one had 50.00. The total average values were very close, whose values are 76.83 and 78.50, respectively, for the graphical and textual approaches. Likewise, the standard deviation proved to be very close in both approaches as well.

Table 11 presents the table with adjectives that can be assigned to the evaluated artifacts using the SUS method. Figure 41 shows the box-plot of the final results, including two outliers from the graphical approach.

Table 10 – Measures of the SUS values in EX3.

	Graphical Treatment	**Textual Treatment**
Measure	**SUS Final Score**	**SUS Final Score**
Maximum	100.00	100.00
3°Quartile	85.00	92.50
Average	76.83	78.50
Median	80.00	75.00
1°Quartile	73.75	67.50
Minimum	37.50	50.00
Variance	248.63	241.79
SD	15.77	15.55

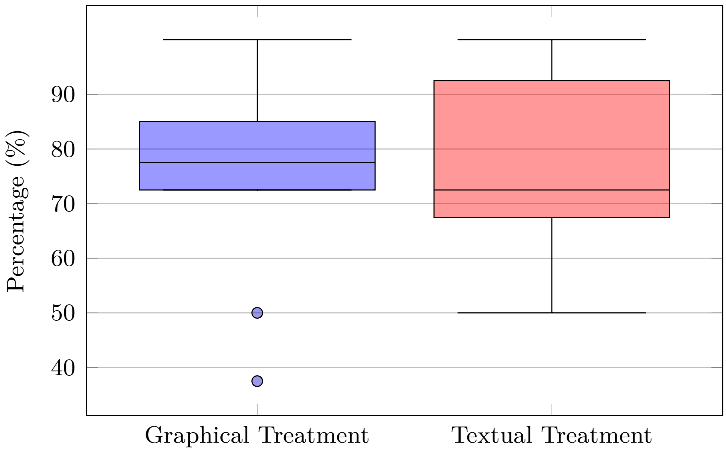
Legend: SD = Standard Deviation.
Source – Author.

Table 11 – SUS classification.

SUS Score	**Grade**	**Adjective Rating**
> 80.3	A	Excellent
68 - 80.3	B	Good
68	C	Okay
51 - 68	D	Poor
< 51	E	Awful

Source – Adapted from (BROOKE, 1995).

Figure 41 – Box-plot - SUS score per treatments in EX3.



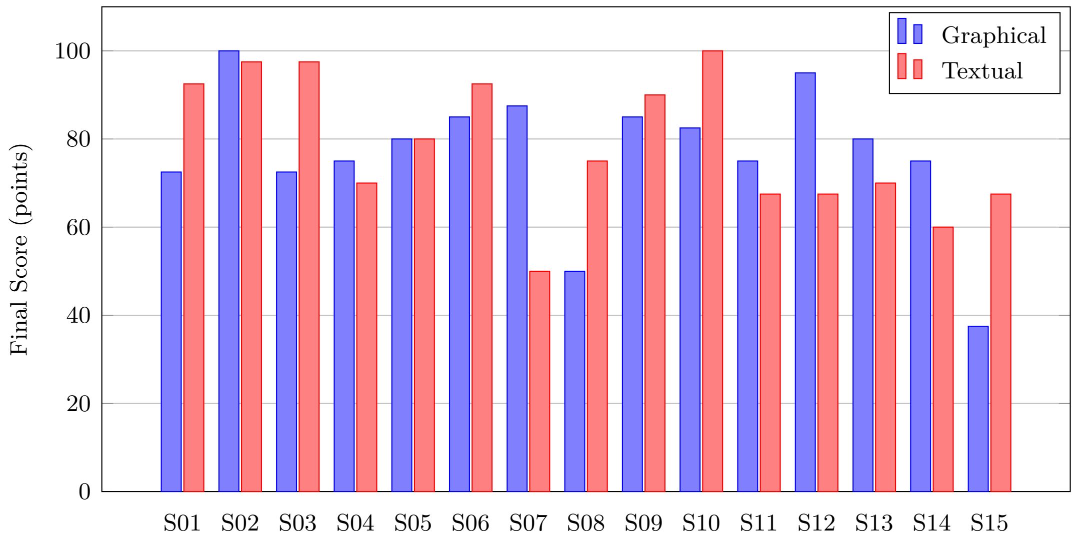
Source – Author.

Figure 42 shows a chart of grouped bars and presents the values obtained by each one of fifteen (15) individuals for both approaches. It is ambitious to evaluate a cause and effect relationship equivalent to the values obtained in the SUS scores and the Emocards, but it is possible to make some inferences.

For example, subject S03 demonstrated that he felt a feeling of irritability while using the brModelo tool, while he was in an aspect of emotion linked to happiness in the use of the ERtext tool. We translated this evidence in the SUS evaluation since the subject better evaluated the textual approach when compared to the graphical one.

This pattern happens more often, but there is no such equivalence in other instances. For instance, the case of subject S07, who declared that he was feeling emotions linked to irritability during the textual approach utilization while feeling on the spectrum of happy emotion in the graphical one. In a contradictory way, in the value SUS calculated for this subject, there is a much better evaluation for the graphical approach using the brModelo tool, while the textual one using the proposed solution ERtext received one of the lowest values of SUS. However, in general, the values obtained from most other participants correspond to the sensation reported for the state of mind they were feeling at the time.

Figure 42 – Grouped Bar Chart - SUS score per subjects and treatments.



Source – Author.

For the evaluation of open-ended questions, there was a process of coding the answers given in the evaluation instruments. At this moment, we used the qualitative analysis tool QAnubis[6]. This web tool aims to support the qualitative analysis process by creating codings and annotations directly on PDF files, generating reports capable of summarizing the captured concepts.

There were about seventy (70) short answers in an open-ended format since each of the fifteen (15) participants could give their impressions about the artifacts generated in the three (3) modeling levels for each of the two (2) approaches.

6 QAnubis tool: <https://dspace.unipampa.edu.br/bitstream/riu/6891/1/Giliardi_Schmidt_2022.pdf> (in portuguese)

The codes defined were *Positive Feedback*, *Negative Feedback*, *Neutral Feedback*, *Improvements*, and *Feature Idea*. As aforementioned, when participants found it interesting to report for criticism, compliments, questions, or suggestions for improvement, they answered the open-ended questions.

We categorized compliments and reports of satisfactory experience as Positive Feedback. Further, we classified as Negative Feedback some criticisms when they did not mention improvements or pointed out malfunctions of some features of the tools. Furthermore, we grouped Neutral Feedback responses that indicated that there were no suggestions, criticisms, or compliments, only reporting vague or ambiguous opinions.

Moreover, we classified as Improvements responses that indicated problems with functionality that did not operate correctly, as expected by the user. Finally, we categorized as Feature Ideas reports of possible non-existent features that could implement.

Table 12 shows the number of occurrences of each coding in the analyzed responses, according to the approach to which they belonged.

Table 12 – Occurrences of codings per treatments in EX3.

Approach	Positive F.	Neutral F.	Negative F.	Improvements	Feature Idea
Graphical	14	4	17	7	2
Textual	16	2	5	2	4

Legend: F = Feedback.
Source – Author.

We mapped respondents as was done for the responses referring to Emocards and the SUS form. Therefore, for example, it is possible to mention that subject S06 considered the textual approach to conceptual modeling efficient and faster than the graphical one. This subject still mentions having found it a bit confusing to understand the logical textual model generated at first, but yet have liked the final result.

Another example is subject S12, who recommends that it would be interesting to have a functionality that could offer customization of the colors of the conceptual diagram and textual logical model in the approach used in the ERtext tool, something he mentions being present in brModelo.

On the one hand, subject S03 reports that the standardization using elements from UML for the conceptual diagram produced by ERtext was unexpected for him as even the impossibility of reorganizing it. On the other hand, he mentioned that despite being more familiar with learning process in brModelo diagram, its presented some problems with the automatic organization of the elements, making him spend more time than expected.

The main criticisms generating classifications in Negative Feedback were concerning the physical models. We related the highest number to the SQL code generated by the brModelo tool. Many of the respondents needed minor adjustments to execute the required DBMS. We aim at obtaining enough shreds of evidence because the instructions are not focused on a specific DBMS platform since, in general, it has a more generic template.

Subjects S04, S07, and S14 still reported problems in the foreign keys generation, which had to be analyzed and then modified for correct execution. However, we have not reported the same problems in the textual approach using ERtext. All raw data analyzed in EX3 is also available in a public repository[7] on the Zenodo platform.

6.3 Chapter Lessons

This chapter presents the replication of a preliminary controlled experiment, called Experiment 1 (EX1), with a sample of 33 subjects evaluating the ERtext proposal, a text-based approach tool using DSL for conceptual modeling of DBs . We compared ERtext with brModelo, a graphical-based approach tool well known in Computing classes.

Then, we carried out two (2) more experiments. EX2 was a new replication with some modifications in the protocol, specifically the addition of new instruments, which has had twenty-five (25) participants this time. The objective was to carry out a new analysis of the effort and effectiveness of the approaches with each tool. We also sought to make a deeper assessment of the quality of use by applying concepts of Russell's circumplex combined with the Emocards method.

Finally, EX3 was another new implementation of the protocol, now with the addition of the SUS form and a qualitative coding analysis. This last experiment had fifteen (15) participants and aimed to evaluate the various artifacts (conceptual models, diagrams, logical models and SQL instructions) that could be generated by both tools.

In this way, we reached a set of three (3) empirical evaluations with a combined total of seventy-three (73) participants. With these results obtained, we could answer the four RQs of the experiment as many as their associated hypotheses. In this sense, we can say that we investigated three main aspects of our DSL: effort, effectiveness, and quality in use. From the analysis of all datasets obtained, we highlight the following aspects:

Effort: The average difference calculated in the experiments concludes that there are no differences between the approaches, *i.e.*, one is not faster than the other;

Efficacy: Through the statistical test analysis in EX1, we have rejected the null hypothesis that the approaches are equally effective. When comparing the box-plot plots produced, we observed a fewer advantages for the textual approach of around 5% on average. However, we observed there were no statistically significant differences concerning the artifacts modeled in EX2. Indeed, the average quality had a difference of about a little over 1% on EX2. Somehow, this demonstrates that it is possible to sustain that both approaches are similar in terms of the quality of the

[7] Available at: <https://doi.org/10.5281/zenodo.6416391>.

models produced, given the low percentage difference appreciated. We also do not discard possible influences caused by the fact that the experiments were conducted remotely due to the pandemic context;

Qualitative comparison between treatments: There was a certain balance between treatments, but with a positive evaluation for ERtext regarding the attributes "Productivity" and "Operability" in EX1. The main advantage observed for brModelo occurred in EX2 concerning the "Learnability" attribute;

We also collected **qualitative feedback**, including in the form of **open-ended responses** from subjects who participated in the experiments: (i) We need to review some improvements regarding the design of DSL, in particular in the ternary relationships (EX1 and EX2) and self-relationships (EX1). (ii) Concerning the coding of the responses to open-ended questions received in EX3, we were able to map some interesting criticisms and suggestions for planning the evolution of ERtext, especially regarding the dynamic creation and customization of diagrams equivalent to textual descriptions.

Since we expect to continue the development of ERtext DSL, the execution of a possible refactoring, as many as the implementation of new ER builders, is a natural step towards the evolution of the solution. From the experimental results obtained, we conclude that there is feasibility and good prospects for our solution proposed as an alternative tool to support the teaching-learning process.

7 FINAL REMARKS

This master's thesis has proposed to execute a study that addresses the area of design and conceptual modeling of relational DBs. Therefore, it was necessary to carry out a research design that guided all the activities necessary to achieve the specific objectives. The project also involved a bibliographical research for the theoretical basis, conducting a systematic mapping, finding related studies, developing the proposed solution and three (3) empirical evaluations for its evaluation.

We know that the set of tools that support the design and modeling of relational DBs is relatively wide. We do not expect to reinvent the wheel or just increase this fragmentation made up of several solutions in an already mature area. In general, this work is about an experimentation. We are looking to verify assumptions and exploring an alternative to the most used approach in this area, and therefore somewhat challenging the established *status quo*. As a result we hope to contribute with a tool that is potentially useful for the teaching-learning process.

In this attempt to push the boundaries and explore an alternative path, which we believe should be encouraged within academia, we propose a DSL and implement a complete modeling tool. This was only possible through the Xtext framework, a known LW for creating textual approach languages.

We performed the replication of an empirical experiment with a set of 33 subjects to verify the feasibility of the proposal and the preliminary results indicate that there may be relevance in the project built in our study. We are aware that even if the results ended up ultimately showing that, after the last planned experiment, there was no significant difference from our proposal for the existing tools, still our collective body of knowledge will remain relevant with regard to development and evolution of modeling languages for specific domains.

We confirmed this in the other two (2) empirical experiment executions. In the end, the three (3) combined experiments had seventy-three (73) subjects. We were then able to quantitatively and qualitatively evaluate a series of assumptions. Our main goal was to assess whether our textual approach could perform the user acceptance as effectively or at least very close to the graphical one. We concluded that both approaches are similar, without a statistical difference or quality of use that proves significant among the subjects participating in the controlled experiments.

We still intend to conduct a better analysis based on the possible inclusion of the future solution in the activities carried out while teaching database lessons, thus performing the activities continuously in the same way used in the graphical tools.

In this way, as we initially observed (Section 1.3), it will be possible to recheck some advantages and assess whether others not previously described could be added to this benefits set. Among the initial assumptions, we can assert that the textual approach can: bring good advantages in modeling speed when users are more familiar with the language;

provide easier integration with other languages; offer superior formatting quality; better version control, and; platform independence (concerning operating systems).

We know that the proposed solution uses a plugin perspective at the moment. This is due to the need for the agile construction of a complete infrastructure that includes from parser, linker, and type-checker to the compiler of the created language. There is currently no Xtext-like framework that makes this available directly in a specialized web context. However, despite the limitation imposed by an Eclipse plugin, *i.e.* works only locally, Xtext provides full LSP support, which opens up the real possibility of the future migration of our solution to a web platform able to run directly in a browser.

7.1 Publications

Furthermore, we emphasize that during the period study, we have submitted the partial results to several events aiming at evaluations of expert peers about our work. As follows, we list some publications that originated in the context of our research.

Published

i. Lopes, J., Bernardino, M., Basso, F., Rodrigues. Avaliação preliminar de uma linguagem para a representação textual de modelos conceituais de bancos de dados. In *Anais da IV Escola Regional de Engenharia de Software.* Porto Alegre, RS, Brasil: SBC, 2020. p. 306-315. (LOPES et al., 2020).

ii. Lopes, J., Bernardino, M., Basso, F., Rodrigues. Multivocal Literature Mapping on DSLs and Tools for Entity-Relationship Modeling. In *Anais da V Escola Regional de Engenharia de Software.* Porto Alegre, RS, Brasil: SBC, 2021. p. 149–158. (RIQUELMO et al., 2021)

iii. Lopes, J., Bernardino, M., Basso, F., Rodrigues, E. Textual approach for designing database conceptual models: A focus group. In *Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development - MODELSWARD*: SciTePress, 2021. p. 171-178. (LOPES. et al., 2021).

iv. Lopes, J., Bernardino, M., Basso, F., Rodrigues, E. Empirical evaluation of a textual approach to database design in a modeling tool. In *Proceedings of the 23th International Conference on Enterprise Information Systems (ICEIS)*: SciTePress, 2021. p. 208–215. (LOPES et al., 2021).

v. Lopes, J., Bernardino, M., Basso, F., Rodrigues, E. Textual-based DSL for Conceptual Database Modeling: A Controlled Experiment. In *Anais do XXXVI Simpósio Brasileiro de Bancos de Dados (SBBD)*: SBC, 2021. (LOPES et al., 2021)

vi. Lopes, J., Bernardino, M., Basso, F., Rodrigues, E. Entity-Relationship Modeling Tools and DSLs: is it still possible to advance the state of the art from observations in practice? In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS)*: SciTePress, 2022. p. 8. (LOPES et al., 2022).

Under Review

i. Journal of Information Data Management (JIDM), 2022.

To be Submitted

i. We intend to write a complete paper and submit it to **Software and System Modeling (SoSyM)**. The general idea will condense all previously published works reporting the research story and comparing the results, including the last controlled experiment.

7.2 Future Work

There are other improvements that we are developing. We intend to implement them as the project continues, making them public in the repository. These resources seek to improve the user experience and increase the spectrum of artifacts generated by the solution. Among the main features we intend to add are:

i. more **templates proposals**, especially for full templates (currently, there is only one available);

ii. **improve scope-based validation**, especially for ternary entities (currently, this validation is partial);

iii. create DDL **generators** for **stored procedures** covering CRUD operations;

iv. carry on implementing **code mining support**[1] using features released in the Eclipse Photon project (currently, this functionality is partially implemented but needs more software testing to be made available in the plugin). The objective will be to show the user summaries of the number of entities, including those derived from N:N relationships, as well as the total of relationships;

v. create **more SQL generators**, thus increasing the set of platforms that can target (currently, we support PostgreSQL and MySQL);

[1] Code Mining: <https://www.eclipse.org/eclipse/news/4.8/platform_isv.php>

vi. implement **customized coloring features** of the automatically generated conceptual (diagram with PlantUML), logical (.html), and physical (SQL) models. Currently, the coloring is defined statically in the developed generators, but we perceived that this option is something as a direct suggestion of the subjects of the experiments;

vii. provide automatic generation of **occurrence diagrams** for indicated relationships in the models;

viii. the generation of **dynamic diagrams** enables the modeling in a graphical way that generates our DSL automatically, thus making our solution a **bidirectional modeling tool**. We currently have the design decision to use Sprotty[2] as a diagramming framework, replacing PlantUML in related projects that we intend to continue. It happens because Sprotty has full support for Xtext, allowing its use as a backend (with requests via LSP) and Typescript (a Javascript superset with support for static typing and OO) on the frontend;

ix. implement Round-trip Engineering (RTE) techniques, *i.e.*, **reverse engineering** methods looking to generate artifacts (conceptual, logical, and physical models) from SQL instructions;

x. investigate the possible **integration** with **Object-relational Mapping (ORM) frameworks** and compatible programming languages;

xi. Provide a semi-automatic process for transforming between models, allowing users to determine adjustments;

xii. Generators for NoSQL database models.

[2] Sprotty Framework: <https://github.com/eclipse/sprotty>

REFERENCES

AMELLER, D. **Considering Non-Functional Requirements in Model-Driven Engineering**. Dissertação (Mestrado) — Llenguatges i Sistemes Informàtics (LSI), June 2009. Available on: <http://upcommons.upc.edu/pfc/handle/2099.1/7192>. Cited at page 38.

BACKUS, J. W. The syntax and semantics of the proposed international algebraic language of the zurich acm-gamm conference. In: **IFIP Congress**. Int. Business Machines Corp., 1959. p. 125–131. Available on: <http://dblp.uni-trier.de/db/conf/ifip/ifip1959.html#Backus59>. Cited at page 41.

BAILER-JONES, D. **Scientific Models in Philosophy of Science**. University of Pittsburgh Press, 2009. Available on: <https://books.google.com.br/books?id=avur50J7UecC>. Cited at page 37.

BAIRAGI, V.; MUNOT, M. V. (Ed.). **Research Methodology**. Chapman and Hall/CRC, 2019. Available on: <https://doi.org/10.1201/9781351013277>. Cited at page 31.

BATORY, D.; AZANZA, M. Teaching model-driven engineering from a relational database perspective. Springer Science and Business Media LLC, v. 16, n. 2, p. 443–467, ago. 2015. Cited at page 39.

BERGIN JR., T. J.; GIBSON JR., R. G. (Ed.). **History of Programming languages—II**. New York, NY, USA: ACM, 1996. Cited at page 41.

BETTINI, L. **Implementing Domain Specific Languages with Xtext and Xtend - Second Edition**. 2nd. ed. : Packt Publishing, 2016. ISBN 1786464969. Cited at page 44.

BRAMBILLA, M.; CABOT, J.; WIMMER, M. **Model-Driven Software Engineering in Practice, Second Edition**. : Morgan & Claypool, 2017. (Synthesis Lectures on Software Engineering). Cited 3 times at pages 26, 37, and 38.

BROOKE, J. Sus: A quick and dirty usability scale. **Usability Eval. Ind.**, v. 189, 11 1995. Cited 3 times at pages 91, 93, and 95.

BUCCHIARONE, A. et al. Grand challenges in model-driven engineering: an analysis of the state of the research. Springer Science and Business Media LLC, v. 19, n. 1, p. 5–13, jan. 2020. Available on: <https://doi.org/10.1007/s10270-019-00773-6>. Cited at page 37.

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. Cited at page 35.

CODD, E. Relational database: A practical foundation for productivity. In: Mylopolous, J.; Brodie, M. (Ed.). **Readings in Artificial Intelligence and Databases**. San Francisco (CA): Morgan Kaufmann, 1989. p. 60–68. ISBN 978-0-934613-53-8. Available on: <https://www.sciencedirect.com/science/article/pii/B9780934613538500091>. Cited at page 25.

COOK, T. D.; CAMPBELL, D. T. **Quasi-Experimentation: Design and Analysis Issues for Field Settings**. Chicago, IL, USA: Houghton Mifflin, 1979. Cited at page 80.

COUGO, P. **Modelagem conceitual e projeto de banco de dados**. Rio de Janeiro, RJ, Brasil: Elsevier Editora Ltda., 2013. Cited at page 37.

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. Cited at page 25.

DAVIS, F. D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. **Management Inf. Systems Quarterly**, v. 13, n. 3, p. 319–340, 1989. ISSN 02767783. Cited at page 77.

DERCZYNSKI, L. Complementarity, f-score, and NLP evaluation. In: **10th Int. Conf. on Language Resources and Evaluation**. : ELRA, 2016. p. 261–266. Cited at page 73.

DESMET, P.; OVERBEEKE, K.; TAX, S. Designing products with added emotional value: Development and appllcation of an approach for research through design. **The Design Journal**, Routledge, v. 4, n. 1, p. 32–47, 2001. Available on: <https://doi.org/10.2752/146069201789378496>. Cited at page 83.

DEURSEN, A. V.; 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. Available on: <http://doi.acm.org/10.1145/352029.352035>. Cited 2 times at pages 40 and 41.

DEY, D.; STOREY, V.; BARRON, T. Improving database design through the analysis of relationships. **ACM Trans. Database Syst.**, v. 24, p. 453–486, 12 1999. Cited 2 times at pages 50 and 51.

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, v. 44, p. 299–318, 2015. Cited 4 times at pages 50, 51, 52, and 54.

DIMITRIESKI, V. et al. Extended entity-relationship approach in a multi-paradigm information system modeling tool. In: **2014 Federated Conference on Computer Science and Information Systems**. 2014. p. 1611–1620. Cited 2 times at pages 50 and 51.

ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados**. Franca, SP, Brasil: Pearson Universidades, 2011. ISBN 9788579360855. Cited at page 25.

FAVERI, C. D. **Uma Linguagem Específica de Domínio para Consulta em Código Orientado a Aspectos**. Dissertação (Mestrado) — Universidade Federal de Santa Maria, aug 2013. Cited at page 41.

FOWLER, M. Language workbenches: The killer-app for domain specific languages? 2005. Available on: <http://www.martinfowler.com/articles/languageWorkbench.html>. Cited at page 42.

FOWLER, M. **Domain Specific Languages**. 1st. ed. London,England: Addison-Wesley Professional, 2010. Cited 3 times at pages 39, 40, and 42.

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. Cited 2 times at pages 39 and 40.

GAMMA, E. et al. **Design patterns: elements of reusable object-oriented software**. : Addison-Wesley Professional, 1995. Cited at page 64.

GOGOLLA, M. Exploring er and re syntax and semantics with metamodel object diagrams. In: **Proceedings of the 2005 Symposia on Metainformatics**. New York, NY, USA: Association for Computing Machinery, 2005. (MIS '05), p. 6es. ISBN 9781595937193. Available on: <https://doi.org/10.1145/1234324.1234330>. Cited at page 50.

GRONBACK, R. **Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit**. : Pearson Education, 2009. (Eclipse Series). ISBN 9780321635198. Cited at page 41.

HARTMANN, S.; LINK, S. Collection type constructors in entity-relationship modeling. In: . 2007. p. 307–322. ISBN 978-3-540-75562-3. Cited at page 50.

HEUSER, C. A. **Projeto de Banco de Dados**. Porto Alegre, RS, Brasil: Bookman, 2009. Cited 3 times at pages 35, 58, and 63.

HOSSAIN, B.; RAJAN, G.; SCHWITTER, R. Cnler: a controlled natural language for specifying and verbalising entity relationship models. In: _____. **Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association**. : Australasian Language Technology Association, 2019. p. 126135. Cited at page 52.

HUTCHINSON, J.; ROUNCEFIELD, M.; WHITTLE, J. Model-driven engineering practices in industry. In: . : ACM, 2011. Cited at page 39.

KARANIKOLAS, N. Database design revisited. In: . 2011. Cited at page 25.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. Keele/Durham, England, 2007. Cited 2 times at pages 45 and 46.

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). Cited at page 39.

KUNG, H.-J.; TUNG, H.-L. A web-based tool for teaching data modeling. **Journal of Computing Sciences in Colleges**, v. 26, p. 231–237, 12 2010. Cited at page 50.

LIKERT, R. **A technique for the measurement of attitudes / by Rensis Likert.** New York: [s.n.], 1985. (Archives of psychology ; no. 140). Cited 6 times at pages 77, 79, 91, 92, 93, and 94.

LOPES, J. et al. Avaliação preliminar de uma linguagem para a representação textual de modelos conceituais de bancos de dados. In: **Anais da IV Escola Regional de Engenharia de Software**. Porto Alegre, RS, Brasil: SBC, 2020. p. 306–315. ISSN 0000-0000. Available on: <https://sol.sbc.org.br/index.php/eres/article/view/13741>. Cited at page 102.

LOPES, J. et al. Empirical evaluation of a textual approach to database design in a modeling tool. In: **Proc. of the 23th International Conference on Enterprise Information Systems (ICEIS)**. : SciTePress, 2021. p. 8. Cited at page 102.

LOPES., J. et al. Textual approach for designing database conceptual models: A focus group. In: INSTICC. **Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development - MODELSWARD,**. : SciTePress, 2021. p. 171–178. ISBN 978-989-758-487-9. ISSN 2184-4348. Cited at page 102.

LOPES, J. et al. Textual-based dsl for conceptual database modeling: A controlled experiment. In: **Anais do XXXVI Simpósio Brasileiro de Bancos de Dados**. Porto Alegre, RS, Brasil: SBC, 2021. p. 169–180. ISSN 2763-8979. Cited at page 102.

LOPES, J. et al. Entity-relationship modeling tools and dsls: is it still possible to advance the state of the art from observations in practice? In: **Proc. of the 24th International Conference on Enterprise Information Systems (ICEIS)**. : SciTePress, 2022. p. 8. Cited at page 103.

LOPES, J. R. **ERtext: uma linguagem específica de domínio para a representação de modelos conceituais de bancos de dados relacionais (in portuguese)**. 135 51–65. Monografia (TCC) — Universidade Federal do Pampa, Rio Grande do Sul, 2019. Cited 4 times at pages 26, 57, 58, and 71.

MALHOTRA, A. et al. An entity-relationship programming language. **IEEE Transactions on Software Engineering**, v. 15, n. 9, p. 1120–1130, 1989. Cited at page 52.

MARCHEZAN, L. et al. Thoth: A web-based tool to support systematic reviews. In: **2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)**. 2019. p. 1–6. Cited at page 46.

MARTELLI, R.; FILHO, O.; CABRAL, A. D. L. **Modelagem e banco de dados**. São Paulo, SP, Brasil: Editora Senac São Paulo, 2018. (Informática). Cited at page 37.

MARTIN, J. **Application Development Without Programmers**. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1982. Cited at page 41.

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. Available on: <http://doi.acm.org/10.1145/1118890.1118892>. Cited at page 41.

MOODY, D. What makes a good diagram? improving the cognitive effectiveness of diagrams in is development. In: ______. **Advances in Information Systems Development: New Methods and Practice for the Networked Society**.

Netherlands: Springer, 2007. (Advances in Information Systems Development, 2), p. 481–492. ISBN 978-0-387-70801-0. Cited at page 28.

NAKAGAWA, E. et al. **Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática**. Rio de Janeiro, RJ, Brasil: Elsevier Editora Ltda., 2017. Cited 3 times at pages 45, 46, and 48.

NARDI, B. A. **A Small Matter of Programming: Perspectives on End User Computing**. 1st. ed. Cambridge, MA, USA: MIT Press, 1993. Cited at page 41.

OBEO; TYPEFOX. **Xtext/Sirius - Integration The Main Use-Cases**. 2017. Available on: <https://www.obeodesigner.com/resource/white-paper/WhitePaper_XtextSirius_EN.pdf>. Cited at page 44.

PAREDAENS, J. et al. **The structure of the relational database model**. : Springer Science & Business Media, 2012. v. 17. Cited at page 25.

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. Cited at page 31.

PERSICO, D.; MANCA, S.; POZZI, F. Adapting the technology acceptance model to evaluate the innovative potential of e-learning systems. **Computers in Human Behavior**, Elsevier, v. 30, p. 614–622, 2014. Cited at page 77.

PETERSEN, K. et al. Systematic mapping studies in software engineering. In: **EASE'08: Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering**. Swindon, UK: BCS Learning Development Ltd., 2008. v. 8, p. 68–77. Cited at page 45.

PRODANOV, C. C.; FREITAS, E. C. D. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico - 2ª Edição**. Novo Hamburgo, RS, Brasil: Editora Feevale, 2013. Cited at page 31.

PUANGSAIJAI, W.; PUNTHEERANURAK, S. A comparative study of relational database and key-value database for big data applications. In: IEEE. **2017 International Electrical Engineering Congress (iEECON)**. 2017. p. 1–4. Cited at page 25.

RASHID, T.; AL-RADHY, R. Transformations to issues in teaching, learning, and assessing methods in databases courses. In: **IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)**. IEEE, 2014. Available on: <https://ieeexplore.ieee.org/abstract/document/7062635/>. Cited at page 26.

REIJNEVELD, K. et al. Measuring the emotions elicited by office chairs. In: **Proceedings of the 2003 International Conference on Designing Pleasurable Products and Interfaces**. New York, NY, USA: Association for Computing Machinery, 2003. (DPPI '03), p. 610. ISBN 1581136528. Available on: <https://doi.org/10.1145/782896.782899>. Cited at page 84.

RIQUELMO, J. et al. Multivocal literature mapping on dsls and tools for entity-relationship modeling. In: **Anais da V Escola Regional de Engenharia de Software**. Porto Alegre, RS, Brasil: SBC, 2021. p. 149–158. ISSN 0000-0000. Available on: <https://sol.sbc.org.br/index.php/eres/article/view/18460>. Cited 2 times at pages 52 and 102.

RISTIĆ, S. et al. A model-to-model transformation of a generic relational database schema into a form type data model. **2016 Federated Conference on Computer Science and Information Systems (FedCSIS)**, p. 1577–1580, 2016. Cited 2 times at pages 50 and 51.

ROBINSON, I.; WEBBER, J.; EIFREM, E. **Graph databases: new opportunities for connected data**. : " O'Reilly Media, Inc.", 2015. Cited at page 25.

ROSENTHAL, A.; REINER, D. Tools and transformations – rigorous and otherwise – for practical database design. **ACM Transactions on Database Systems**, v. 19, 06 1994. Cited at page 50.

ROSS, D. T. History of programming languages. In: Wexelblat, R. L. (Ed.). New York, NY, USA: ACM, 1981. Chap. Origins of the APT Language for Automatically Programmed Tools, p. 279–338. Available on: <http://doi.acm.org/10.1145/800025.1198374>. Cited at page 41.

SALGADO, A. C.; MEDEIROS, C. B. Ensino de banco de dados. In: **Anais do Workshop de Informática na Escola**. 1995. Cited at page 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. Available on: <http://doi.acm.org/10.1145/361454.361485>. Cited at page 41.

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. Cited at page 39.

SOMMERVILLE, I. **Software Engineering**. 10th. ed. London, UK: Pearson, 2015. ISBN 0133943038. Cited 2 times at pages 37 and 38.

SOUSA, V. M. D.; CURA, L. M. D. V. Logical design of graph databases from an entity-relationship conceptual model. In: **iiWAS2018: International Conference on Information Integration and Web-based Applications Services**. New York, NY, USA: Association for Computing Machinery, 2018. p. 183189. ISBN 9781450364799. Available on: <https://doi.org/10.1145/3282373.3282375>. Cited 2 times at pages 50 and 51.

SUBAHI, A. F.; SIMONS, A. J. H. A model transformation approach for translating conceptual database schemas into executable database systems. In: **Semantic Scholar**. : Semantic Scholar, 2011. Cited 2 times at pages 50 and 52.

TEOREY, T. J.; YANG, D.; FRY, J. P. A logical design methodology for relational databases using the extended entity-relationship model. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 18, n. 2, p. 197222, jun. 1986. ISSN 0360-0300. Available on: <https://doi.org/10.1145/7474.7475>. Cited at page 50.

TRIOLA, M. **Elementary Statistics**. : Pearson, 2018. ISBN 9780134462455. Cited 2 times at pages 72 and 75.

VARA, J. M. et al. Model transformation for object-relational database development. In: **Proceedings of the 2007 ACM Symposium on Applied Computing**. New York, NY, USA: Association for Computing Machinery, 2007. (SAC '07), p. 10121019. ISBN 1595934804. Available on: <https://doi.org/10.1145/1244002.1244222>. Cited at page 50.

WACHSMUTH, G. H.; KONAT, G. D. P.; VISSER, E. Language design with the spoofax language workbench. In: **IEEE Software**. : IEEE, 2014. v. 31, n. 5, p. 35–43. Cited at page 42.

WEST, M. **Developing High Quality Data Models**. Amsterdam, Netherlands: Elsevier Science, 2011. Cited at page 37.

WEXELBLAT, R. L. (Ed.). **History of Programming Languages**. New York, NY, USA: ACM, 1981. Cited at page 41.

Appendix

APPENDIX A – EXPERIMENT 1 (EX1)

A.1 F-score - EX1

	Graphical Approach					Textual Approach					
Subject	Retrieved Items	Relevant Items	Precision (%)	Recall (%)	F-Measure (%)	Retrieved Items	Relevant Items	Precision (%)	Recall (%)	F-Measure (%)	DIF
1	32	30	93.75	76.92	84.51	36	35	97.22	74.47	84.34	-0.17
2	20	18	90.00	46.15	61.02	28	26	92.86	55.32	69.33	8.32
3	26	24	92.31	61.54	73.85	32	31	96.88	65.96	78.48	4.63
4	23	20	86.96	51.28	64.52	27	24	88.89	51.06	64.86	0.35
5	30	29	96.67	63.04	76.32	31	30	96.77	75.00	84.51	8.19
6	22	20	90.91	51.28	65.57	44	32	72.73	68.09	70.33	4.76
7	34	32	94.12	69.57	80.00	33	32	96.97	80.00	87.67	7.67
8	18	16	88.89	41.03	56.14	21	19	90.48	40.43	55.88	-0.26
9	34	28	82.35	60.87	70.00	49	36	73.47	90.00	80.90	10.90
10	22	21	95.45	53.85	68.85	36	32	88.89	68.09	77.11	8.26
11	32	29	90.63	63.04	74.36	35	32	91.43	80.00	85.33	10.97
12	19	18	94.74	46.15	62.07	28	24	85.71	51.06	64.00	1.93
13	47	36	76.60	92.31	83.72	56	46	82.14	97.87	89.32	5.60
14	25	22	88.00	56.41	68.75	19	18	94.74	38.30	54.55	-14.20
15	22	17	77.27	43.59	55.74	25	20	80.00	42.55	55.56	-0.18
16	30	29	96.67	63.04	76.32	33	30	90.91	75.00	82.19	5.88
17	26	19	73.08	48.72	58.46	24	22	91.67	46.81	61.97	3.51
18	25	22	88.00	47.83	61.97	25	23	92.00	57.50	70.77	8.80
19	22	16	72.73	41.03	52.46	26	23	88.46	48.94	63.01	10.55
20	33	28	84.85	60.87	70.89	32	29	90.63	72.50	80.56	9.67
21	28	27	96.43	58.70	72.97	34	30	88.24	75.00	81.08	8.11
22	27	24	88.89	61.54	72.73	33	29	87.88	72.50	79.45	6.72
23	31	28	90.32	71.79	80.00	34	30	88.24	63.83	74.07	-5.93
24	26	23	88.46	50.00	63.89	27	25	92.59	62.50	74.63	10.74
25	36	33	91.67	84.62	88.00	41	37	90.24	92.50	91.36	3.36
26	30	27	90.00	58.70	71.05	30	29	96.67	72.50	82.86	11.80
27	26	24	92.31	52.17	66.67	22	19	86.36	47.50	61.29	-5.38
28	31	27	87.10	69.23	77.14	31	30	96.77	75.00	84.51	7.36
29	25	21	84.00	53.85	65.63	22	21	95.45	44.68	60.87	-4.76
30	25	20	80.00	51.28	62.50	30	24	80.00	51.06	62.34	-0.16
31	35	31	88.57	67.39	76.54	30	29	96.67	72.50	82.86	6.31
32	27	21	77.78	53.85	63.64	26	24	92.31	60.00	72.73	9.09
33	19	16	84.21	41.03	55.17	19	15	78.95	31.91	45.45	-9.72

Reference Models (Instruments)

Graphical	Model1	Model2
Entities	7	9
Binary relationship	3	4
Ternary relationship	1	1
Self-relationship	1	1
Cardinalities	6	7
Attributes	19	21
Specialization	2	3
TOTAL	39	46

Textual	Model1	Model2
Entities	7	9
Binary relationship	3	4
Ternary relationship	1	1
Self-relationship	1	1
Cardinalities	6	7
Attributes	19	21
Specialization	2	3
Domain	1	1
TOTAL	40	47

General	Retrieved Items	Relevant Items	Precision (%)	Recall (%)	F-Measure	Retrieved Items	Relevant Items	Precision (%)	Recall (%)	F-Measure
Minimum	18.00	16.00	72.73	41.03	52.46	19.00	15.00	72.73	31.91	45.45
1º Quartile	23.00	20.00	84.21	50.00	62.50	26.00	23.00	87.88	51.06	63.01
Average	27.52	24.12	87.69	57.96	69.13	30.88	27.45	89.49	63.65	73.16
Median	26.00	24.00	88.89	56.41	68.85	30.00	29.00	90.63	65.96	74.63
3º Quartile	31.00	28.00	92.31	63.04	76.32	34.00	31.00	94.74	75.00	82.86
Maximum	47.00	36.00	96.67	92.31	88.00	56.00	46.00	97.22	97.87	91.36
Variance	35.58	28.65	42.87	143.35	78.93	63.32	39.76	42.59	259.85	133.58
Std Deviation	5.97	5.35	6.55	11.97	8.88	7.96	6.31	6.53	16.12	11.56

A.2 Times - EX1

INSTRUMENT 1	
BRMODELO	ERTEXT (15)
19	51
51	20
39	60
48	32
35	33
30	46
45	15
17	11
12	42
19	51
17	45
35	35
32	60
32	37
30	21
43	
50	
14	

INSTRUMENT 2	
BRMODELO	ERTEXT (18)
51	17
21	38
30	39
31	46
38	26
39	44
20	48
60	26
24	40
27	23
34	21
38	40
50	26
47	36
27	30
	49
	40
	23

GENERAL		
BRMODELO	ERTEXT	DIF
19	17	2
51	38	13
39	40	-1
48	46	2
35	26	9
30	44	-14
45	48	-3
17	26	-9
12	40	-28
19	23	-4
17	21	-4
35	40	-5
32	26	6
32	36	-4
30	31	-1
43	49	-6
50	40	10
14	23	-9
51	52	-1
20	21	-1
60	30	30
32	31	1
33	38	-5
46	39	7
15	20	-5
11	60	-49
42	24	18
51	27	24
45	34	11
35	38	-3
60	50	10
37	47	-10
21	27	-6

APPENDIX B – EXPERIMENT 2 (EX2)

B.1 F-score - EX2

	Graphical Approach					Textual Approach					
Subject	Retrieved Items	Relevant Items	Precision (%)	Recall (%)	F-Measure (%)	Retrieved Items	Relevant Items	Precision (%)	Recall (%)	F-Measure (%)	DIF
1	44	37	84.09	77.08	80.43	44	37	84.09	90.24	87.06	6.62
2	41	35	85.37	72.92	78.65	65	38	58.46	92.68	71.70	-6.95
3	32	30	93.75	75.00	83.33	49	43	87.76	87.76	87.76	4.42
4	46	41	89.13	85.42	87.23	37	35	94.59	85.37	89.74	2.51
5	43	37	86.05	92.50	89.16	44	39	88.64	79.59	83.87	-5.29
6	34	30	88.24	62.50	73.17	42	32	76.19	78.05	77.11	3.94
7	49	42	85.71	87.50	86.60	36	34	94.44	82.93	88.31	1.71
8	45	39	86.67	81.25	83.87	34	32	94.12	78.05	85.33	1.46
9	45	40	88.89	83.33	86.02	37	33	89.19	80.49	84.62	-1.41
10	34	30	88.24	75.00	81.08	48	40	83.33	81.63	82.47	1.39
11	43	36	83.72	75.00	79.12	38	34	89.47	82.93	86.08	6.96
12	46	41	89.13	85.42	87.23	38	36	94.74	87.80	91.14	3.91
13	42	36	85.71	90.00	87.80	46	41	89.13	83.67	86.32	-1.49
14	38	30	78.95	75.00	76.92	45	37	82.22	75.51	78.72	1.80
15	46	39	84.78	81.25	82.98	44	36	81.82	87.80	84.71	1.73
16	43	37	86.05	77.08	81.32	40	35	87.50	85.37	86.42	5.10
17	36	31	86.11	77.50	81.58	41	35	85.37	71.43	77.78	-3.80
18	49	39	79.59	81.25	80.41	41	35	85.37	85.37	85.37	4.95
19	27	23	85.19	57.50	68.66	54	33	61.11	80.49	69.47	0.82
20	36	31	86.11	64.58	73.81	60	36	60.00	87.80	71.29	-2.52
21	39	35	89.74	72.92	80.46	40	36	90.00	87.80	88.89	8.43
22	34	31	91.18	77.50	83.78	47	39	82.98	79.59	81.25	-2.53
23	48	44	91.67	91.67	91.67	42	37	88.10	90.24	89.16	-2.51
24	38	31	81.58	77.50	79.49	44	34	77.27	69.39	73.12	-6.37
25	49	35	71.43	87.50	78.65	60	45	75.00	91.84	82.57	3.92

General	Retrieved Items	Relevant Items	Precision (%)	Recall (%)	F-Measure	Retrieved Items	Relevant Items	Precision (%)	Recall (%)	F-Measure
Minimum	27.00	23.00	71.43	57.50	68.66	34.00	32.00	58.46	69.39	69.47
1º Quartile	36.00	31.00	84.78	75.00	79.12	40.00	34.00	81.82	79.59	78.72
Average	41.08	35.20	85.88	78.57	81.74	44.64	36.48	83.24	83.35	82.81
Median	43.00	36.00	86.05	77.50	81.32	44.00	36.00	85.37	83.67	84.71
3º Quartile	46.00	39.00	88.89	85.42	86.02	47.00	38.00	89.19	87.80	87.06
Maximum	49.00	44.00	93.75	92.50	91.67	65.00	45.00	94.74	92.68	91.14
Variance	34.63	23.84	20.21	72.57	26.91	59.59	10.25	102.43	34.72	37.18
Std Deviation	5.89	4.88	4.50	8.52	5.19	7.72	3.20	10.12	5.89	6.10

Reference Models (Instruments)

Graphical	Model1	Model2
Entities	7	9
Binary relationship	3	4
Ternary relationship	1	1
Self-relationship	1	1
Cardinalities	6	7
Attributes	20	23
Specialization	2	3
TOTAL	40	48
Textual	**Model1**	**Model2**
Entities	7	9
Binary relationship	3	4
Ternary relationship	1	1
Self-relationship	1	1
Cardinalities	6	7
Attributes	20	23
Specialization	2	3
Domain	1	1
TOTAL	41	49

B.2 Times - EX2

	INSTRUMENT 1		INSTRUMENT 2		GENERAL		
	BRMODELO	ERTEXT	BRMODELO	ERTEXT	BRMODELO	ERTEXT	DIF
1	28			50	28	50	-22
2	70			30	70	30	40
3	35			30	35	30	5
4	30			32	30	32	-2
5	30			55	30	55	-25
6	44			74	44	74	-30
7	46			43	46	43	3
8	33			40	33	40	-7
9	45			25	45	25	20
10	28			31	28	31	-3
11	46			29	46	29	17
12	102			171	102	171	-69
1		44	90		90	44	46
2		47	38		38	47	-9
3		55	46		46	55	-9
4		21	30		30	21	9
5		35	53		53	35	18
6		55	74		74	55	19
7		23	24		24	23	1
8		36	61		61	36	25
9		40	31		31	40	-9
10		50	38		38	50	-12
11		67	24		24	67	-43
12		93	58		58	93	-35
13		93	29		29	93	-64

B.3 Emocards - EX2

INSTRUMENT 1		INSTRUMENT 2	
ERtext	brModelo	ERtext	brModelo
3	3	2	4
3	8	3	8
5	3	3	5
3	3	2	3
3	2	3	2
8	3	1	3
4	2	6	5
2	3	2	4
2	2	2	3
4	5	5	4
2	2	4	3
4	3	4	5
7			4

ERtext	brModelo
3	3
3	8
5	3
3	3
3	2
8	3
4	2
2	3
2	2
4	5
2	2
4	3
7	4
2	8
3	5
3	3
2	2
3	3
1	5
6	4
2	3
2	4
5	3
4	5
4	4

	COUNT			
	ERtext	%	brModelo	%
	1	4.00%	0	0.00%
	7	28.00%	5	20.00%
	7	28.00%	10	40.00%
	5	20.00%	4	16.00%
	2	8.00%	4	16.00%
	1	4.00%	0	0.00%
	1	4.00%	0	0.00%
	1	4.00%	2	8.00%
SUM	25	100.00%	25	100.00%

APPENDIX C – EXPERIMENT 3 (EX3)

C.1 Emocards - brModelo - EX3

brModelo (15)
3
3
8
4
5
3
2
1
4
4
2
4
2
3
5

	COUNT	
	brModelo	%
	1	6,67%
	3	20,00%
	4	26,67%
	4	26,67%
	2	13,33%
	0	0,00%
	0	0,00%
	1	6,67%
SUM	15	100,00%

C.2 Emocards - ERtext - EX3

ERtext (15)		COUNT	
3		ERtext	%
2		0	0,00%
2		5	33,33%
3		5	33,33%
3		4	26,67%
3		0	0,00%
8		0	0,00%
4		0	0,00%
2		1	6,67%
4	SUM	15	100,00%
2			
4			
2			
4			
3			

C.3 SUS - brModelo - EX3

Question	1	2	3	4	5	6	7	8	9	10	
Subject											SUS FINAL SCORE
1	4	2	4	2	4	2	4	1	3	3	72,5
2	5	1	5	1	5	1	5	1	5	1	100
3	3	1	4	1	3	2	4	2	3	2	72,5
4	4	2	3	2	4	1	4	2	4	2	75
5	3	1	5	1	3	3	3	1	5	1	80
6	3	1	5	1	3	1	5	1	4	2	85
7	5	2	5	3	5	2	5	1	4	1	87,5
8	3	2	3	4	4	5	4	2	2	3	50
9	4	2	5	2	4	2	5	1	5	2	85
10	5	2	4	2	4	1	4	1	4	2	82,5
11	5	1	4	4	4	2	4	1	4	3	75
12	5	1	5	1	5	1	4	1	5	2	95
13	5	1	5	4	5	1	4	2	5	4	80
14	5	1	5	2	3	3	4	2	3	2	75
15	3	4	2	3	3	4	2	4	4	4	37,5

SUS SCORE	GRADE	Adjective Rating
> 80.3	A	Excellent
68 - 80.3	B	Good
68	C	Okay
51 - 68	D	Poor
< 51	E	Awful

General	
Minimum	37,50
1º Quartile	73,75
Average	76,83
Median	80,00
3º Quartile	85,00
Maximum	100,00
Variance	248,63
Std Deviation	15,77

For every odd-numbered question, subtract 1 from the score (X-1)

For every even-numbered question, subtract the score from 5 (5-X)

1	0	8	0	5	0	6	0	9	0	3
2	0	6	1	5	0	5	1	5	1	7
3	5	0	2	2	5	2	1	0	3	3
4	3	1	4	3	6	1	9	1	6	2
5	7	0	8	0	4	1	4	0	5	0
SUM	15	15	15	15	15	15	15	15	15	15

C.4 SUS - ERtext - EX3

Question	1	2	3	4	5	6	7	8	9	10	
Subject											SUS FINAL SCORE
1	4	3	5	1	5	1	5	1	5	1	92,5
2	5	1	5	1	5	1	4	1	5	1	97,5
3	5	1	5	2	5	1	5	1	5	1	97,5
4	4	2	4	3	4	1	3	1	4	4	70
5	4	3	5	2	4	1	5	1	4	3	80
6	4	1	5	1	5	2	5	1	4	1	92,5
7	3	3	3	3	3	3	3	3	3	3	50
8	4	3	4	2	4	2	4	1	4	2	75
9	5	1	5	3	5	1	5	1	5	3	90
10	5	1	5	1	5	1	5	1	5	1	100
11	4	1	4	4	3	1	4	3	4	3	67,5
12	5	2	3	4	4	1	4	2	4	4	67,5
13	5	4	5	4	5	1	4	2	4	4	70
14	4	2	4	3	4	4	4	2	2	3	60
15	5	3	4	2	3	2	4	2	5	5	67,5

SUS SCORE	GRADE	Adjective Rating
> 80.3	A	Excellent
68 - 80.3	B	Good
68	C	Okay
51 - 68	D	Poor
< 51	E	Awful

General	
Minimum	50,00
1º Quartile	67,50
Average	78,50
Median	75,00
3º Quartile	92,50
Maximum	100,00
Variance	241,79
Std Deviation	15,55

For every odd-numbered question, subtract 1 from the score (X-1)

For every even-numbered question, subtract the score from 5 (5-X)

1	0	6	0	4	0	10	0	9	0	5
2	0	3	0	4	0	3	0	4	1	1
3	1	5	2	4	3	1	2	2	1	5
4	7	1	5	3	5	1	7	0	7	3
5	7	0	8	0	7	0	6	0	6	1
SUM	15	15	15	15	15	15	15	15	15	15

D REFERENCE MODELS AND EXAMPLES

D.1 Instrument 1 (portuguese description)

unipampa

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.

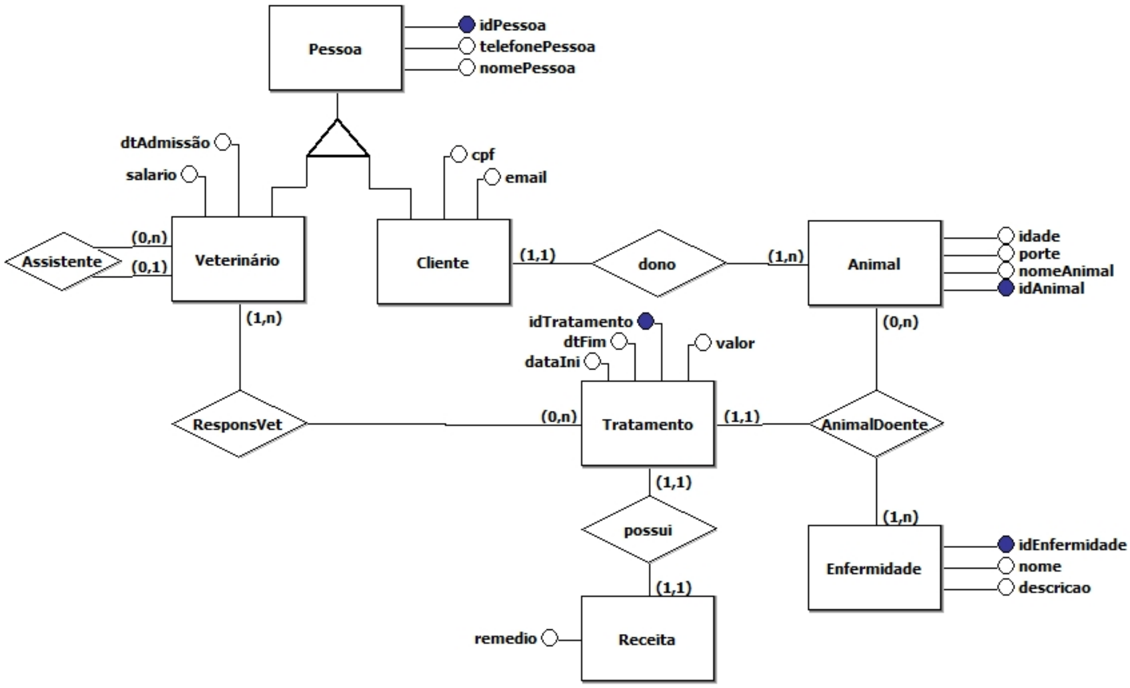
D.2 Instrument 2 (portuguese description)

Instrumento 2

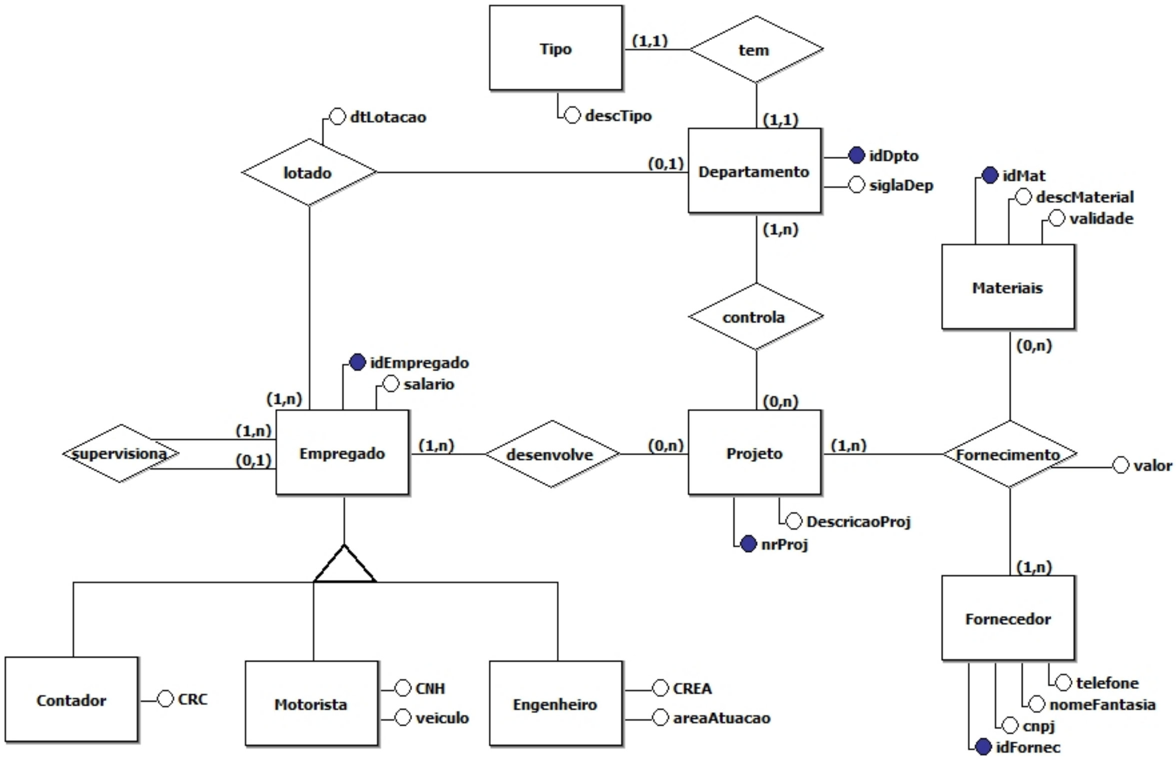
Nome: ____________________ Data: __________
Curso: ____________________ Hora Início: __________
Matrícula: ____________________ Hora Fim: __________

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.

D.3 Reference model - brModelo (Instrument 1)



D.4 Reference model - brModelo (Instrument 2)



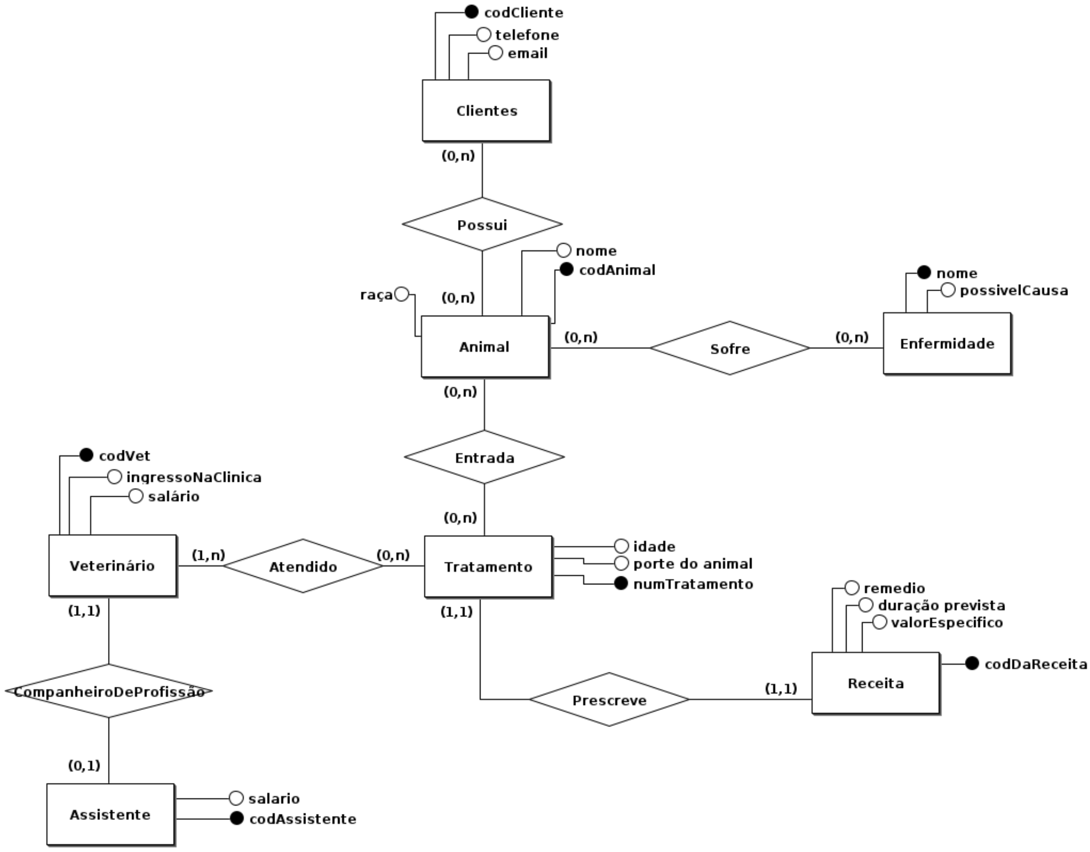
D.5 Reference model - ERtext (Instrument 1)

```
Domain HospitalVeterinario;

Entities {
    Pessoa {
        idPessoa int isIdentifier,
        telefone int,
        nome string
    }
    Veterinario is total/disjoint Pessoa {
        dtAdmissao datetime,
        salario money
    }
    Cliente is total/disjoint Pessoa {
        cpf int,
        email string
    }
    Animal {
        idAnimal int isIdentifier,
        idade int,
        porte string,
        nome string
    }
    Enfermidade {
        idEnfermidade int isIdentifier,
        nome string,
        descricao string
    }
    Tratamento {
        idTratamento int isIdentifier,
        dtInicio datetime,
        dtFim datetime
    }
    Receita {
        idReceita int isIdentifier,
        remedio string
    }
};

Relationships {
    Assistente   [Veterinario (0:1) relates (0:N) Veterinario]
    Dono         [Cliente (1:1) relates (1:N) Animal]
    Reponsavel   [Veterinario (1:N) relates (0:N) Tratamento]
    Possui       [Tratamento (1:1) relates (1:1) Receita]
    AnimalDoente [Animal (0:N) relates (1:N) Enfermidade]
    Tratamento   [AnimalDoente (1:N) relates (1:1) Tratamento]
};
```

D.6 Reference model - ERtext (Instrument 2)

```
Domain EmpresaGenerica;

Entities {
    Empregado {
        idEmpregado int isIdentifier,
        salario money
    }
    Contador is total/disjoint Empregado {
        crc int
    }
    Motorista is total/disjoint Empregado {
        cnh int,
        veiculo string
    }
    Engenheiro is total/disjoint Empregado {
            crea int,
            areaAtuacao string
    }
    Departamento {
        idDept int isIdentifier,
        sigla string
    }
    Tipo {
        idTipo int isIdentifier,
        descricao string
    }
    Projeto {
        idProjeto int isIdentifier,
        descricao string
    }
    Materiais {
        idMaterial int isIdentifier,
        descricao string,
        validade string
    }
    Fornecedor {
        idFornecedor int isIdentifier,
        cnpj int,
        nomeFantasia string,
        telefone int
    }
};

Relationships {
    Supervisiona [Empregado (0:1) relates (1:N) Empregado]
    Desenvolve   [Empregado (1:N) relates (0:N) Projeto]
    Lotado       [Empregado (1:N) relates (0:1) Departamento]
    Tem          [Departamento (1:1) relates (1:1) Tipo]
    Controla     [Departamento (1:N) relates (0:N) Projeto]
    Fornecimento [Materiais (0:N) relates (1:N) Fornecedor]
    ProjFornec   [Projeto (1:N) relates (1:N) Fornecimento]
};
```

D.7 Modeled examples by a subject - brModelo



D.8 Modeled examples by a subject - ERtext

```
Domain EmpresaGenerica;

Entities{
    Empregado{
        codigo int isIdentifier,
        nome string,
        idade int,
        sexo string
    }

    Motorista is partial/overlapped Empregado{
        CNH int
    }

    Engenheiro is partial/overlapped Empregado{
        CREA int
    }

    Contador is partial/overlapped Empregado{
        CRC int
    }

    Departamento{
        numero int isIdentifier
    }

    Fornecedor{
        nome string,
        CNPJ int isIdentifier,
        telefone int
    }

    Projeto{
        codigo int isIdentifier,
        descricao string,
        dadosProjeto string
    }

    Material{
        codigo int isIdentifier,
        descricao string,
        registroValidade string
    }
};
Relationships{

    lotacao [Departamento (1:1) relates (0:N) Empregado]
    supervisao [Empregado (1:N) relates (1:1) Empregado]
    desenvolvimento [Empregado (1:N) relates (1:1) Projeto]
    possui [Projeto (0:N) relates (0:N) Material]
    provem [Material (0:N) relates (1:1) Fornecedor]
};
```

INDEX