

# *brModelo*: An Initiative for Aiding Database Design

Ronaldo dos Santos Mello<sup>1</sup>, Carlos Henrique Cândido<sup>2</sup>, Milton Bittencourt S. Neto<sup>3</sup>

<sup>1</sup> Universidade Federal de Santa Catarina (UFSC), Brazil  
r.mello@ufsc.br

<sup>2</sup> Universidade Federal do Mato Grosso (UFMT), Brazil  
chcandido@gmail.com

<sup>3</sup> Mercado Livre, Florianópolis, SC, Brazil  
miltonbst@gmail.com

**Abstract.** The *brModelo* tool is a initiative of the UFSC Database Group. Its first version was developed in 2005, and its main purpose is to help teaching of relational database design. Compared to similar tools, its main differentials are the support to all steps of the classical database design methodology, user interaction during the logical design step, as well as the support to all extended Entity-Relationship concepts. With more than fifteen years of existence, the *brModelo* was very well-accepted by the brazilian Database community, which motivated the development and release of several versions of the tool. This article presents the history of *brModelo*, including its available versions and their functionalities. Additionally, we detail its functionalities and compare it with popular related tools.

Categories and Subject Descriptors: H.2 [Database Management]: Logical Design

Keywords: database design, entity-relationship model, relational data model

## 1. INTRODUCTION

Database (DB) design is a common process in the context of development and maintenance of a computational infrastructure, giving that the resulting DB schema can be accessed by one or more business applications [Batini et al. 1992]. A DB modeling activity aims at providing a suitable abstraction of the business domain data, as well as to guarantee efficient storage and access to these data. Several problems may raise if a DB design methodology is not considered, like incorrect or partial understanding of the domain data, data redundancy and low access performance.

*brModelo* is a tool that supports the semiautomatic design of DBs whose main focus is relational DBs. It was developed and has been maintained by the UFSC DB Group (GBD/UFSC)<sup>1</sup>. After more than fifteen years of its creation and disclosure in the brazilian DB community, we see today its wide usage in Computer Science undergraduate and graduate DB courses, as well as technical DB design courses. One example of such a comprehensive usage are the tutorials produced by teachers and professionals that use the *brModelo*<sup>2</sup>. The main reason for the popularity of the *brModelo* is its differentials if compared to tools with the same purpose:

- (1) support to all traditional steps of the DB design methodology for relational and NoSQL DBs: conceptual design, logical design, and physical design;
- (2) it helps the designer to choose the most suitable and valid DB structures during the logical design;

<sup>1</sup><http://lisa.inf.ufsc.br/wiki/index.php/Main>

<sup>2</sup>[https://www.youtube.com/results?search\\_query=brmodelo](https://www.youtube.com/results?search_query=brmodelo)

- (3) support to all extended Entity-Relationship model (EER model) concepts, as proposed by Batini and Ceri [Batini et al. 1992], and adapted by a brazilian classical DB design literature [Heuser 2008].

The tool has been improved over time, which generated several versions whose main functionalities are presented in Section 3. Nevertheless, Section 2 tells a little bit about the history of the *brModelo* initiative, Section 4 discusses and compares related tools, and Section 5 is dedicated to the conclusion.

This article is an extension of a previous paper published at SBBD Demos 2020 as a distinguished demo [Mello et al. 2020]. We detail here the functionalities of the main *brModelo* versions, some design decisions, and we also provide a comparison of the tool against close related work.

## 2. SOME HISTORY

DB design is a required content in DB courses due to its great relevance. Some important literature for relational DB modeling are the book of Elmasri & Navathe [Elmasri and Navathe 2015], as well as the brazilian book of Carlos A. Heuser, whose first edition is 2001 [Heuser 2001]. At the time of the release of this edition of the Heuser's book, we realized that there was no DB design tool that could help in relational database design teaching, as well as a tool that could support the three traditional steps of a DB design methodology. These limitations had motivated the design and implementation of the first version of the *brModelo* tool by the GBD/UFSC in 2005 [Cândido 2005].

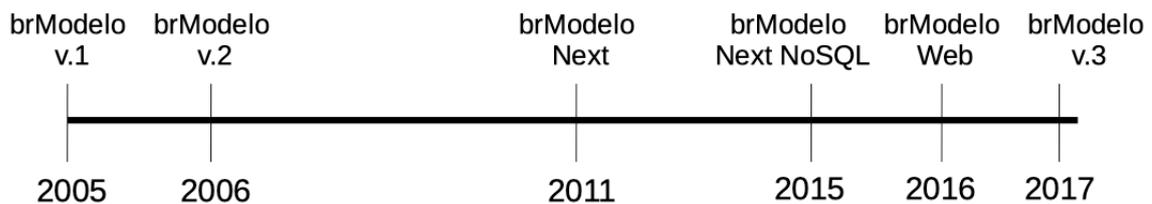


Fig. 1. *brModelo* tool timeline

Figure 1 shows a *timeline* with the main facts related to the history of *brModelo*. Its first version was developed using the *Delphi* programming language, being restricted to run into *Microsoft Windows* operational system. The second version, released at 2006, had maintained this restriction, but it corrected some usability bugs and some errors in the generation of a logical schema from an EER conceptual schema.

Even with its increasing use in DB undergraduate courses around Brazil, the GBD/UFSC continued to receive several requests to convert the *brModelo* into a multiplatform tool. These requests were considered in 2011 with the release of the *brModeloNext* version. The *brModeloNext* was completely implemented in *Java* programming language, and could run in any computer with an installed *Java* virtual machine. It was introduced to the brazilian DB community in the Demos Session of the Brazilian DB Symposium [Menna et al. 2011]. Due to its new user interface, more friendly and modern, this new release was considered a new generation of the tool, which justified the inclusion of the "Next" term in its name. In 2015, the prototype of a new *brModeloNext* tool, called *brModeloNext NoSQL*, was made available. It is able to generate a logical schema called *aggregated schema*, which acts as a canonical abstraction for three NoSQL data models: key-value, column-oriented and document-oriented. A data schema based on aggregates for NoSQL DBs was defined in the literature [Sadallage and Fowler 2012], and a diagrammatic notation for it was proposed by the GBD/UFSC [Lima and Mello 2015]. The *brModeloNext NoSQL* tool version is still under development, but a test version is available (see Section 5).

All *brModelo* versions until 2015 are *desktop* tools, *i.e.*, they can only be downloaded and executed in a local machine. It changed in 2016 with the release of the *brModeloWeb*, the first *brModelo* version accessible through the Internet [Neto 2016]. Despite having less functionalities than the desktop versions, it provides an easy access by anyone, which benefit mainly DB modeling classes at Computer Science laboratories. The *brModeloWeb* tool was introduced to the brazilian DB community as a workshop at DB regional school (ERBD) in 2017<sup>3</sup>.

In the same event (ERBD 2017), the GBD/UFSC also released the third version of *brModelo*: *brModelo v.3* [Cândido and Mello 2017]. This is the actual desktop version of the tool, which was also redesigned in Java to become multiplatform. Besides the support to DB design, it also becomes a CASE tool, providing graphical editors to other diagrammatic notations useful for software engineering, link activity diagrams and data flow diagrams.

### 3. MAIN FUNCTIONALITIES AND DESIGN DECISIONS

All *brModelo* versions and branches introduced in the previous section share the following functionalities: (i) support to all steps of a traditional DB design (conceptual modeling, logical modeling and physical modeling); (ii) user intervention during the generation of the DB logical modeling; (iii) graphical user-friendly interface with several menu options and icons representing DB modeling concepts that can be manipulated in a drag-and-drop way.

The aforementioned functionalities are exemplified in the following for the *brModelo v.3* tool version. Figure 2 shows the design of a conceptual model through the EER notation of Heuser [Heuser 2008]. The icons in the right side represent the EER model concepts, which can be selected and moved to the central area of the user interface. In turn, Figure 3 shows an example of user interaction with the tool during the generation of the logical model for the conceptual model of Figure 2. In this situation, every time the tool detects more than one option for the mapping of a concept in the conceptual model to a relational schema, it presents these options to the user, and the user can choose one of them. The menu options and icons in the top of the user interface provide the common manipulation operations (*start new model*, *save model*, etc), as well as the shift to the next step the DB design process.

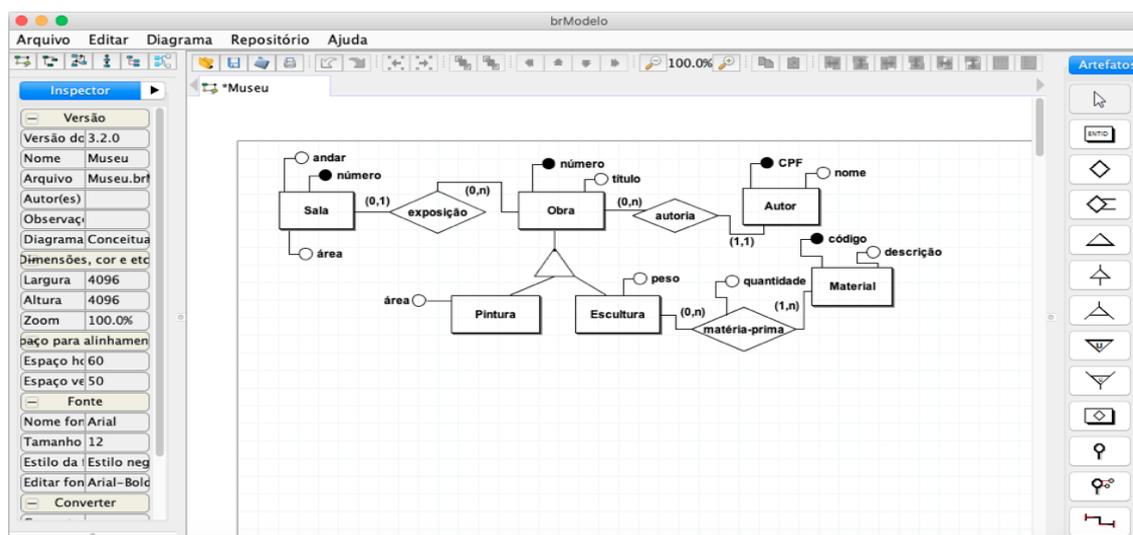


Fig. 2. *brModelo v.3*: user interface for conceptual modeling (in Portuguese)

<sup>3</sup><https://www.upf.br/erbd2017>

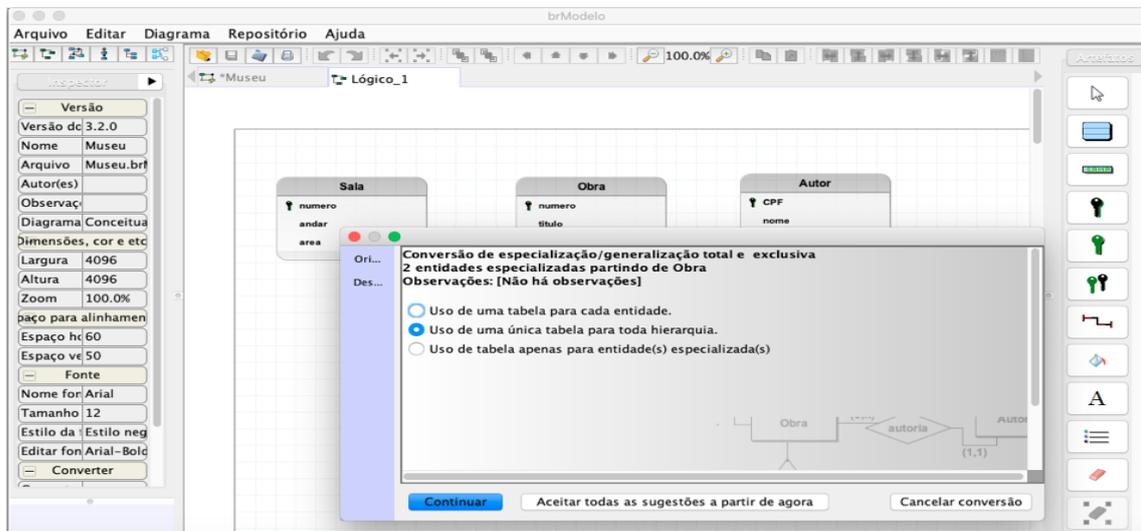


Fig. 3. brModelo v.3: user interface for logical modeling (in Portuguese)

The detection of conceptual-to-logical mapping options mentioned before is one of the main contributions of *brModelo*. It makes the tool a powerful DB design-aided tool. Currently, the *brModelo* versions provide such an assistance during a relational DB design. The user is able to decide for the best mapping option for the following EER concepts:

- (1) *inheritance hierarchies*: it is possible to fusion the generic entity and its specializations into a single table, to define tables for each entity in the hierarchy, or to create tables only for the specialized entities through the inheritance of the generic entity attributes and relationships. If we have an inheritance hierarchy with more than one level, this mapping process is accomplished in a *bottom-up* way. This is the situation illustrated in Figure 3.
- (2) *multivalued attributes*: an specific table can be created to accommodate the attribute values. This table also have a foreign key to the table that represents the entity  $E_i$  that holds the attribute. Another option is to generate  $N$  columns ( $N$  is set by the user) in table that represents  $E_i$ , considering a multivalued attribute with  $(1, N)$  values. This option is suitable when  $N$  is small.
- (3) *one-to-one optional relationships*: a foreign key or a specific table  $T_j$  for the relationship are the possible options. The second case is suggested when few entity occurrences participate of this relationship. The  $T_j$  primary key is the identifier of one of the tables that represent the involved entities. Sometimes is also possible to fusion the related entities, as well as the relationship, into a single table. This situation is recommended when only one of the entities holds an optional participation but, in most of the cases, its occurrences are related to the other entity.
- (4) *one-to-many optional relationships*: the same options of the *one-to-one optional relationships*. In this case, if  $T_j$  is created, its primary key is the identifier of the table that represent the entity with maximal cardinality equal to one. This is the situation illustrated in Figure 8.
- (5) *N-ary relationships*: this case deals with relationships involving more than two entities. Some possibilities raise for defining the primary key ( $pk_r$ ) of the table that represents the relationship, and the user must choose one of them. If all the maximal cardinalities are higher than 1, for example, the  $pk_r$  must be the composition of the primary keys of all tables that correspond to the related entities. If one maximal cardinality (with an entity  $e_n$ ) is 1, then  $pk_r$  is the composition of the primary keys of all tables that correspond to the related entities  $e_1$  to  $e_{n-1}$ . Finally, in a extreme scenario where all maximal cardinalities are 1, any combination of the identifiers of  $n-1$  related entities can serve as  $pk_r$ , and the other combinations must be unique.

Another relevant automatic design decision provided by *brModelo* is the treatment of an associative entity, *i.e.*, a relationship  $R_k$  that is promoted to an entity in the conceptual model in order to be further related to one or more entities through a set of relationships  $\{R_1, \dots, R_x\}$ . In this case, the mapping process must first treat  $R_k$  to decide for the most suitable corresponding logical construct. Once decided the mapping of  $R_k$ , it is possible to provide the mapping of  $\{R_1, \dots, R_x\}$ .

The *brModeloNext* tool, in turn, had introduced several improvements in terms of user interaction, like the possibility of working with multiple windows, as shown in Figure 4. In the example, the user can visualize his/her conceptual, logical and physical modelings at the same time, being able to manipulate each one of them.

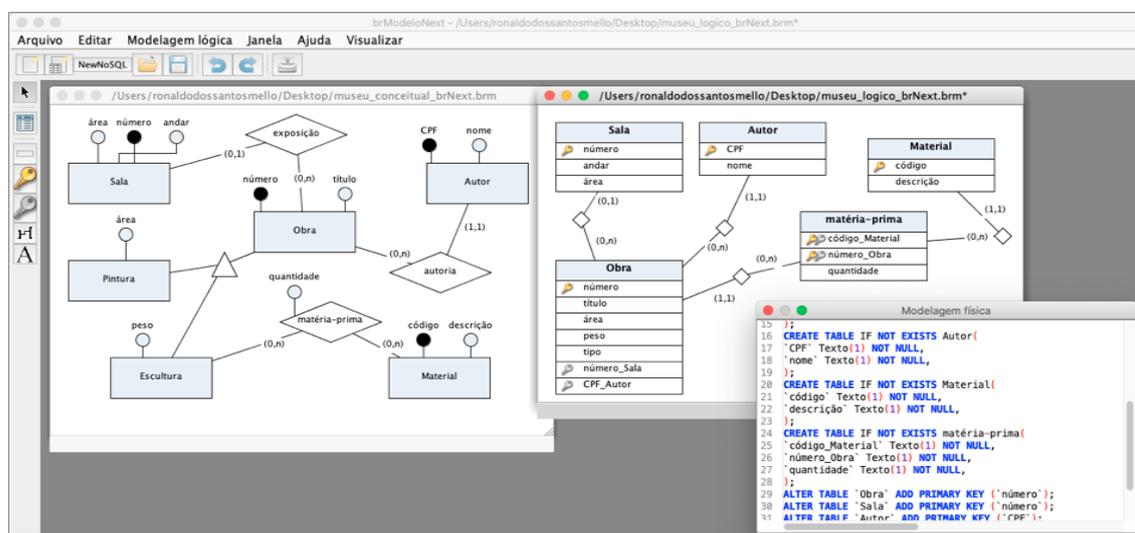


Fig. 4. *brModeloNext* (in Portuguese)

Figure 5 shows the user interface of the *brModeloNext NoSQL* branch. As stated before, this tool allows the generation of an aggregated-based logical model from an EER conceptual model. An aggregate is a complex object schema, *i.e.*, a schema that can aggregate other object schemas, as is the case of the *Obra* (in English Artwork) object schema, which encapsulates the *Pintura* (in English Painting) and *Escultura* (in English Sculpture) schemas besides its attributes.

Basically, the conversion process starts by analyzing the specialization hierarchies and suggesting three possible mapping alternatives: (i) one aggregate object type for the overall hierarchy, and the specialized entities become inner object types, as shown in Figure 5 for *Obra*; (ii) one aggregate object type for the generic entity that comprises the attributes and relationships (properties) of the specialized entities; and (iii) aggregate object types for the specialized entities, so these object types inherit the properties of the generic entity. The user must choose one of these options. The first one is indicated when all of the entities hold several particular properties. The second one is recommended when the generic entity holds more of the hierarchy properties, and the third one is suitable when the specializations are the more relevant entities.

In the following, the conversion process deals with the relationships. Relationship conversion is based on the reasoning for relational DB design. One-to-one relationships generates an aggregate object type that fusion the involved entities or makes one of them an inner object type of the other. One-to-many ones may generate an aggregation of entities, *i.e.*, an entity  $e_1$  that is associated to several occurrences of the other entity  $e_2$  generates an aggregate object type  $e_1$  that encapsulates  $e_2$ , or reference attributes are created. This second option is adopted in Figure 5 for the conversion of the

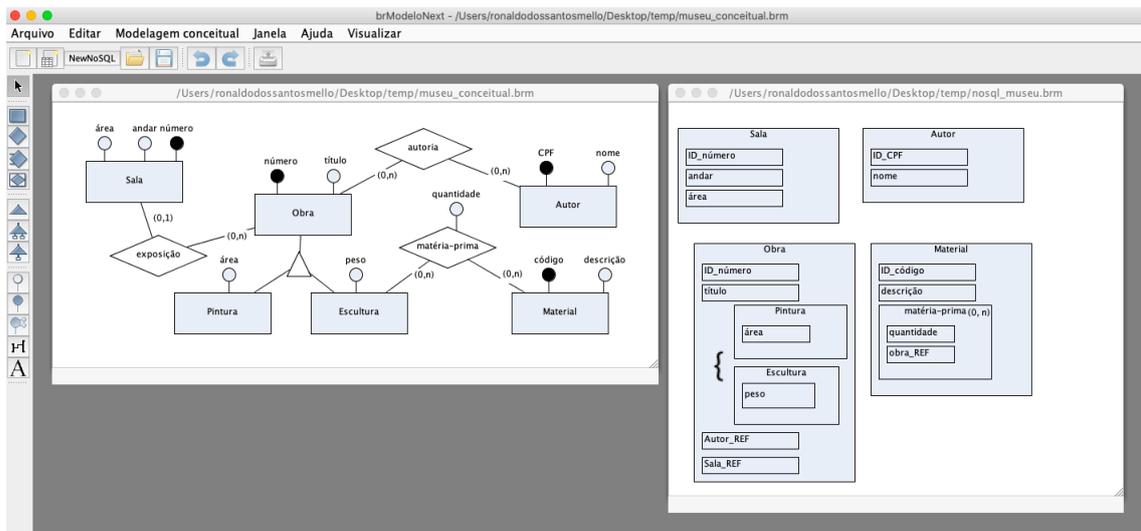


Fig. 5. brModeloNextNoSQL (in Portuguese)

*exposição* (in English exhibition) and *autoria* (in English authorship) relationships. At last, many-to-many relationships may be converted to multivalued references or an inner object type representing the relationship. This last option is exemplified for the conversion of the *matéria-prima* (in English raw material) relationship.

Currently, the brModeloNext NoSQL is able to generate and export DB schemas for the document-oriented DBMS MongoDB and the column-oriented DBMS Cassandra from an aggregated-based logical schema. In the first case, a *JSON schema* specification is produced (a MongoDB Collection). In the second case, the tool produces a *CQL schema* (a Cassandra Keyspace). Examples for both cases are shown in Figure 6 (a) and Figure 6 (b), respectively. The examples highlight the generation of the schema for the *Obra* aggregate object.

```

db.createCollection("Obra", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      properties: {
        ID número Obra: { bsonType: "objectId"},
        título: { bsonType: "string"},
        Pintura: {
          bsonType: "object",
          properties: {
            área: { bsonType: "number"},
          },
          required: ["área"],
          additionalProperties : false,
        },
        Escultura: {
          bsonType: "object",
          properties: {
            ...
          }
        },
        Autor_REF: { bsonType: "objectId"},
        Sala_REF: { bsonType: "objectId"},
        required: ["ID número Obra", "título",
          "Autor_REF"],
        oneOf : [
          {required : ["Pintura"]},
          {required : ["Escultura"]}, ],
      } ... }
}

```

(a)

```

CREATE KEYSPACE IF NOT EXISTS NewDB WITH
replication = {
  'class': 'SimpleStrategy',
  'replication_factor': '3'
};
use NewDB;

CREATE TYPE PINTURA (área FLOAT);
CREATE TYPE ESCULTURA ( ... );
CREATE TYPE AUTOR ( ... );
CREATE TYPE SALA ( ... );
...
CREATE TABLE OBRA (
  ID número Obra UUID,
  título TEXT,
  Pintura PINTURA,
  Escultura ESCULTURA,
  Autor REF AUTOR,
  Sala REF SALA,
  PRIMARY KEY (id)
);
...

```

(b)

Fig. 6. brModeloNextNoSQL: MongoDB (a) and Cassandra (b) Schemas

In short, an aggregate object type  $ao_i$  becomes an object definition in JSON Schema, and  $ao_i$  inner

object types generate inner JSON Schema object definitions, while  $ao_i$  attributes produce JSON schema simple properties. The JSON Schema *ObjectID* type allows the definition of object references and the *oneOf* constraint specifies inner aggregate object types disjuncture. For Cassandra, types are defined for each aggregate object types, and its possible to nest object into a table definition, as shown in Figure 6 (b). Different from MongoDB, Cassandra is not able to specify disjuncture constraints.

Finally, the *brModeloWeb* tool is a complete reconstruction of the *brModelo* desktop version as a service in the cloud, based mainly on the *JavaScript* programming language. It maintains all the main *brModelo* fundamentals, such as the user interface layout and modeling elements, as shown in Figure 7, and the assisted conversion between conceptual to logical modeling steps (Figure 8).

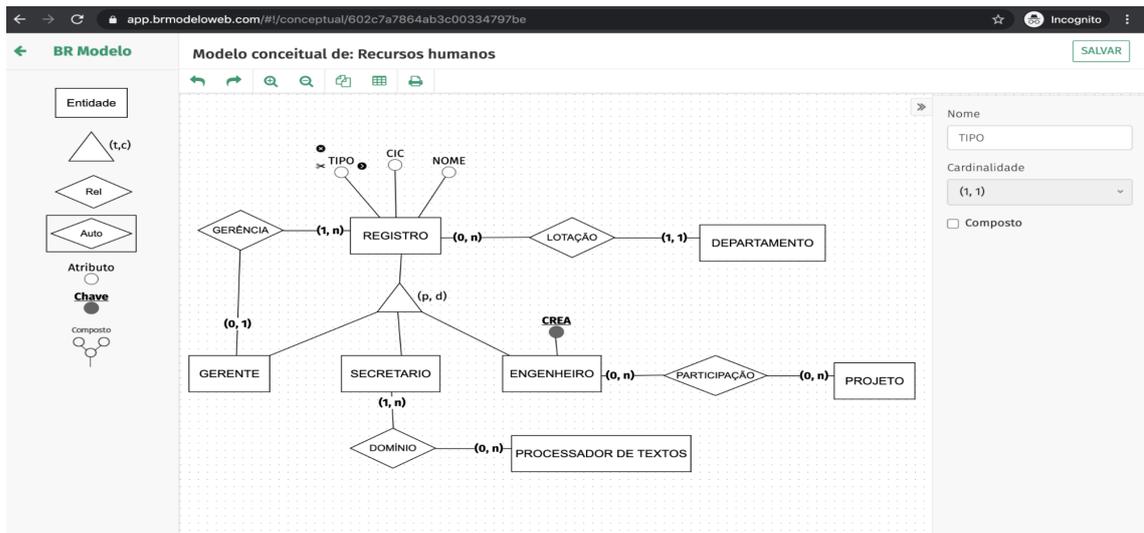


Fig. 7. brModeloWeb: conceptual modeling workspace interface (in Portuguese)

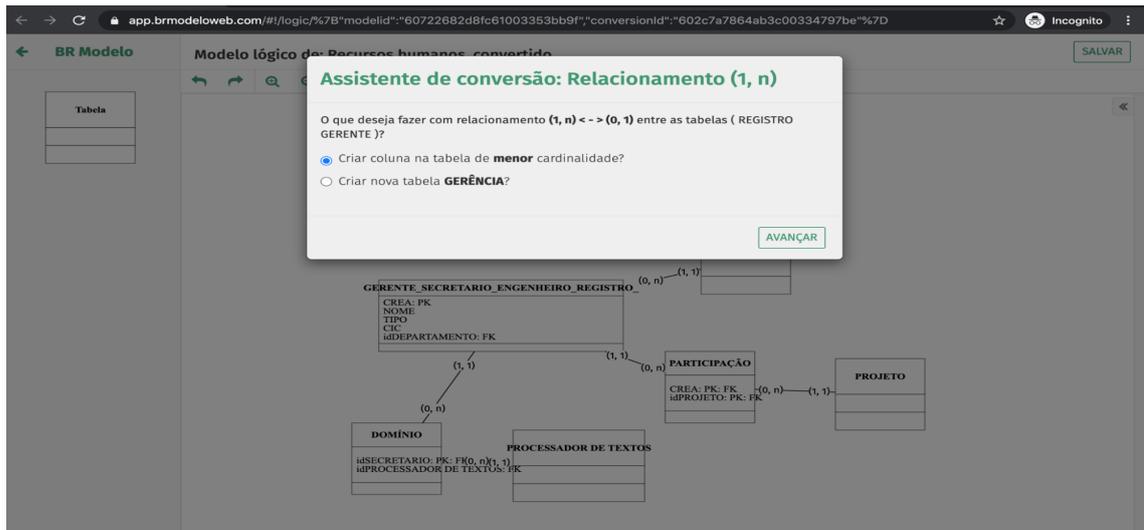


Fig. 8. brModeloWeb: create account interface (in Portuguese)

One of the main motivations for re-implementing *brModelo* into a Web environment is to democratize the teaching and learning concepts related to DB modeling, as the tool is hosted in the cloud. Any user, free of charge, is able to create an account in few seconds, start a modeling and access it later from any computer with Internet connection. It allows students without their own computers to access their work from a laboratory, library or any computer account they have access to. Figure 9 shows the account page, and Figure 10 displays the designer's working area, which shows his/her created and saved models in the cloud. It is also possible to create new conceptual or logical models in this screen.



Fig. 9. brModeloWeb: create account interface (in Portuguese)

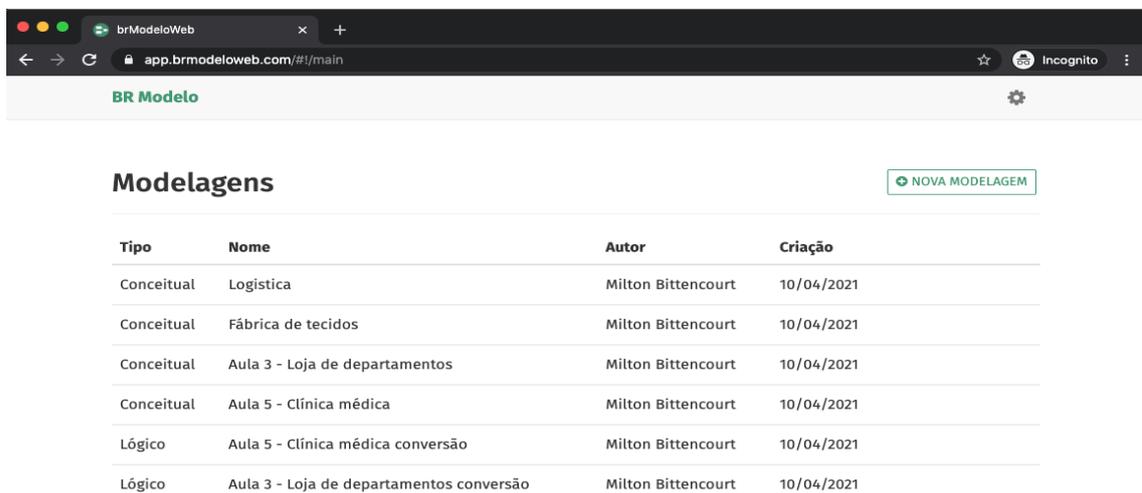


Fig. 10. brModeloWeb: workspace interface (in Portuguese)

The *brModeloWeb* lacks some functionalities available at older tool versions, like the simultaneous visualization of the conceptual and logical modelings he/she is working on. However, this tool version is attracting the attention of the developer's community and enthusiasts of DB modeling, which are

constantly making contributions to the application repository. It is ensuring that the *brModeloWeb* be in constant maintenance and evolution.

#### 4. RELATED WORK COMPARISON

Several commercial softwares that support relational DB design are available today, like *ERwin*<sup>4</sup>, *DBDesigner*<sup>5</sup>, *Enterprise Architect*<sup>6</sup>, *RISE Editor*<sup>7</sup>, *DeZign*<sup>8</sup>, *Power Designer*<sup>9</sup>, *Oracle Designer*<sup>10</sup>, *ER/Studio*<sup>11</sup>, *Toad Data Modeler*<sup>12</sup>, *Open ModelSphere*<sup>13</sup> and *DB-Main*<sup>14</sup>. We also have academic tools, like *TerraER* [Rocha and Terra 2010] e *EERCASE* [Alves et al. 2014] for Portuguese language.

We compare the desktop version (*brModelo v.3*) against some related tools in order to demonstrate the level of maturity of the *brModelo* initiative. Two criteria were considered to select the related desktop tools for analysis: (i) gratuity; and (ii) conceptual modeling support. We argue that these criteria are fundamental when using this kind of tool for teaching purposes. Given that, the selected tools were EERCASE, TerraER, Open ModelSphere, and RISE Editor. We also consider the following comparison criteria:

- *Logical modeling*: creation of a logical model from scratch;
- *Conceptual-logical conversion (C-L conversion)*: ability to generate a logical model from a conceptual model;
- *Guided conversion*: user intervention is allowed during the conceptual-to-logical conversion;
- *SQL generation*: ability to generate a physical model (an SQL DDL script) from a logical model;
- *NoSQL support*: ability to generate a logical model for NoSQL DBs, as well as physical models for at least one of the NoSQL DBs data models;
- *Open source*: the tool is an open source initiative;
- *Active project*: the tool is continuously updated;
- *Multiplatform*: the tool runs over several computer platforms.

Table I presents the comparison of the selected tools. The check mark denotes that the tool holds the criterium. It is worth mentioning that all tools were downloaded, installed and tested in a local machine in order to verify their capabilities.

Open ModelSphere is a complete tool in terms of support to all steps of a DB design methodology. However, it provides such a support only to relational DB design. This tool, as well as RISE editor, are not compatible to modern operational systems. In order to test them, we have to create a virtual machine with old Java releases (Java 6), which makes difficult their usage.

On considering the academic tools (EERCASE and TerraER), they cover the conceptual modeling step, and only EERCASE is able to generate an SQL script from the conceptual model, lacking the support to the logical modeling, which is an important step in terms of defining suitable relational structures for the application domain.

<sup>4</sup><https://www.erwin.com>

<sup>5</sup><https://www.dbdesigner.net>

<sup>6</sup><https://www.sparxsystems.eu>

<sup>7</sup><http://www.risetobloome.com/>

<sup>8</sup><https://www.datanamic.com/dezign/>

<sup>9</sup><https://sybase-powerdesigner.informer.com>

<sup>10</sup><https://www.oracle.com/database/technologies/developer-tools/designer.html>

<sup>11</sup><https://er-studio.software.informer.com>

<sup>12</sup><https://www.toadworld.com/products/toad-data-modeler>

<sup>13</sup><http://www.modelsphere.com/>

<sup>14</sup><https://db-main.software.informer.com/>

Table I. Comparison of desktop tools.

Feature	EERCASE	TerraER	Open ModelSphere	RISE Editor	brModelo 3.0
Logical modeling			✓	✓	✓
C-L conversion			✓		✓
Guided conversion					✓
SQL generation	✓		✓	✓	✓
NoSQL support					✓
Open source	✓	✓			✓
Active project		✓			✓
Multiplatform	✓	✓			✓

From Table I, it is possible to see that only *brModelo v.3* holds all the considered features. It is the only one that offer a guided conceptual-to-logical model conversion, as well as support to NoSQL DB design.

We also compare some Web tools for DB design against the Web version of *brModelo* (*brModeloWeb*), as shown in Table II. Once again, the gratuity and conceptual modeling functionality were used as selection criteria, and the same comparison criteria were also considered, except the *Multiplatform* criterion, which had been replaced by the *Cross browser* one, *i.e.*, the capability to run over several Web browsers. The selected tools were *ERDPlus*, *Lucidchart*, *dbdiagram.io* and *SQL Designer*.

Table II. Comparison of Web tools.

Feature	ERDPlus	Lucidchart	dbdiagram.io	SQLDesigner	brModeloWeb
Logical modeling	✓	✓	✓		✓
C-L conversion	✓				✓
Guided conversion					✓
SQL generation	✓		✓	✓	✓
NoSQL support					
Open source				✓	✓
Active project	✓	✓	✓	✓	✓
Cross browser	✓	✓	✓	✓	✓

From the analyzed tools, we highlight ERPlus. It also supports all the three DB design steps, it is able to convert a conceptual model to a logical one, and makes available several tutorials. Some bugs during the model conversions were detected, even for simple binary conceptual relationships, which results in not connected tables in the relational logical model.

Lucidchart is a very user-friendly tool, which focus on conceptual and logical modeling. However, it does not provide an automatic process for the conversion between these two models, and it has limitations that restrict its usage as a free tool. In turn, dbdiagram.io presents a proprietary notation for data modeling, focusing on the usage of the JSON format to build the conceptual and logical models.

Finally, SQLDesigner is an open source tool with an emphasis on the conceptual modeling and further generation of SQL code from a conceptual model. A good point of all of the analyzed Web tools is that they are active projects and run over several Web browsers.

Similar to the comparison against desktop tools, the *brModeloWeb* is the only one that provides a guided conversion. Currently, it does not provide NoSQL DB modeling, but this is a support we intend to release in a near new version. We also highlight that few ones are open source projects and also offer a conceptual-to-logical conversion, as *brModeloWeb* does.

It is worth mentioning that there are other tools that support the creation of ER diagrams, like *draw.io*<sup>15</sup>, but these tools are not devoted for designing DBs.

## 5. CONCLUSION

The *brModelo* initiative had begun in 2005 and continues along these sixteen years due to the research and development joint effort of the GBD/UFSC and external collaborators. All of these efforts is motivated by the great acceptance that the *brModelo* tools had obtained mainly in terms of DB design teaching in undergraduate courses in Brazil as well as DB design training in the Brazilian industry.

A large number of video classes about the *brModelo* tool is available at *YouTube*, with hundreds of thousands of visualizations and downloads, maybe more. In 2017, when *brModelo v.3* was released, the number of tool downloads was higher than 500 thousand. Such a number had probably increased until now. Such a great interest in the tool was not expected by our research group, so we must go on! The main desktop versions of *brModelo* are available for download at the GBD/UFSC Wiki<sup>16</sup>. Regarding *brModeloWeb*, it is available at <https://www.brmodeloweb.com>. It is an open-source project and contributions are welcome at <https://github.com/brmodeloweb/brmodelo-app>.

As future works, several functionality improvements are under development in some *brModelo* versions thanks mainly to feedback and suggestions we had received over the years. Some examples of ongoing improvements are the tool internationalization, as well as the support to schema constraints design and logical modeling of other post-relational DBs besides NoSQL, like object-oriented and XML ones.

## REFERENCES

- ALVES, E., FRANCO, N., NASCIMENTO, A., AND FIDALGO, R. EERCASE: Uma Ferramenta para Apoiar o Estudo do Projeto Conceitual de Banco de Dados. In *Workshops do III Congresso Brasileiro de Informática na Educação (CBIE)*, 2014.
- BATINI, C., CERI, S., AND NAVATHE, S. B. *Conceptual Database Design: An Entity-Relationship Approach*. Benjamin/Cummings, 1992.
- CÂNDIDO, C. H. *Aprendizagem em Banco de Dados: Implementação de Ferramenta de Modelagem E.R.*, 2005. Monografia de Especialização. Universidade Federal de Santa Catarina. 44p.
- CÂNDIDO, C. H. AND MELLO, R. Ferramenta de Modelagem de Banco de Dados Relacionais *brModelo v3*. In *XIII Escola Regional de Banco de Dados (ERBD)*, 2017.
- ELMASRI, R. AND NAVATHE, S. B. *Fundamentals of Database Systems*. Pearson Higher Education, 2015.
- HEUSER, C. A. *Projeto de Banco de Dados*. Sagra Luzzatto, 2001.
- HEUSER, C. A. *Projeto de Banco de Dados*. Bookman, 2008.
- LIMA, C. AND MELLO, R. A Workload-Driven Logical Design Approach for NoSQL Document Databases. In *XVII International Conference on Information Integration and Web-based Applications & Services (iiWAS)*. ACM, 2015.
- MELLO, R., CÂNDIDO, C. H., AND NETO, M. B. S. Ferramenta *brModelo*: Quinze Anos! In *SBBDD 2020 Demonstrations and Applications (SBBDD Demos)*. Brazilian Computer Society, 2020.
- MENNA, O. S., RAMOS, L. A., AND MELLO, R. *brModeloNext*: a Nova Versão de uma Ferramenta para Modelagem de Bancos de Dados Relacionais. In *VI Sessão de Demos do Simpósio Brasileiro de Banco de Dados (SBBDD Demos)*, 2011.
- NETO, M. B. *brModeloWeb*: Ferramenta Web para Ensino e Modelagem de Banco de Dados, 2016. Trabalho de Conclusão de Curso em Ciência da Computação. Universidade Federal de Santa Catarina. 70p.
- ROCHA, H. S. C. AND TERRA, R. *TerraER*: Uma Ferramenta voltada ao Ensino do Modelo de Entidade-Relacionamento. In *VI Escola Regional de Banco de Dados (ERBD)*, 2010.
- SADALAGE, P. J. AND FOWLER, M. *NoSQL Distilled : A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley, 2012.

<sup>15</sup><https://drawio-app.com>

<sup>16</sup><http://lisa.inf.ufsc.br/wiki/index.php/Projects>