

# **brModelo: FERRAMENTA DE MODELAGEM CONCEITUAL DE BANCO DE DADOS**

## **MONOGRAFIA**

Observação: As marcas aqui citadas são de propriedade de suas respectivas firmas. O nome brModelo é uma invenção livre (br: Brasil, Modelo: modelagem de dados), qualquer semelhança a qualquer marca comercial é mera coincidência.

## **SUMÁRIO**

### ÍNDICE DE FIGURAS

### RESUMO

### ABSTRACT

### 1. INTRODUÇÃO

#### 1.1. Apresentação

##### 1.1.1. Abordagem Relacional

##### 1.1.2. Notação de Peter Chen

#### 1.2. Problemática

#### 1.3. Limitações do tema

#### 1.4. Objetivo

#### 1.5. Justificativa

#### 1.6. Hipótese.

### 2. MODELAGEM DE DADOS

#### 2.1. Nível de abstração do modelo

#### 2.2. Modelo Conceitual.

#### 2.3. Modelo lógico

#### 2.4. Modelo físico

### 3. ABORDAGEM RELACIONAL.

#### 3.1. O Modelo entidade-relacionamento (MER) – P. P. Chen

#### 3.2. Diagrama de entidade-relacionamento – DER

### 4. FERRAMENTAS DE CONSTRUÇÃO DE MODELOS/ESQUEMAS

#### 4.1. A ferramenta CASE ideal

#### 4.2. Análise das ferramentas disponíveis no mercado.

##### 4.2.1. A análise

###### 4.2.1.1. Oracle – OracleDesigner

###### 4.2.1.2. Sybase – PowerDesigner

###### 4.2.1.3. CA - ERwin

###### 4.2.1.4. Freeware DBDesigner

###### 4.2.1.5. PyDesigner – Python Database Designer

###### 4.2.1.6. Microsoft Visio

#### 4.2.2. Conclusão da análise:

### 5. A APLICAÇÃO brModelo

#### 5.1. brModelo – Diferencial na construção do modelo conceitual.

#### 5.2. brModelo – Conversão para o modelo lógico.

#### 5.3. brModelo: Adicionais.

#### 5.4. brModelo: Para o futuro

### 6. CONCLUSÃO

### 7. REFERÊNCIAS BIBLIOGRÁFICAS

- [7.1. Livros](#)
- [7.2. Revistas](#)
- [7.3. Artigos e textos da internet](#)

## ÍNDICE DE FIGURAS

- [Figura 1. Modelo gerado no dbMain: software sugerido por Heuser. Fonte: \[www.dbmain.com\]\(http://www.dbmain.com\)](#)
- [Figura 2. Imagem editada para SQL Magazine. Fonte: Reinaldo V. Álvares](#)
- [Figura 3. Conceito de entidade e relacionamento - Peter Chen \[9\]](#)
- [Figura 4. Uma entidade: visão em um conjunto de registro](#)
- [Figura 5. Relacionamento <sup>7</sup>.](#)
- [Figura 6. Notação original do Dr. Chen \(1976\)\[10\] – Sem atributos.](#)
- [Figura 7. Notação original do Dr. Chen \(1976\)\[10\] – Com atributos.](#)
- [Figura 8. brModelo – Notação de Heuser \(2001\)\[3\] – Sem atributos](#)
- [Figura 9. brModelo – Notação de Heuser \(2001\)\[3\] – Com atributos](#)
- [Figura 10. Foras de modelar – Universidade da Califórnia, Los Angeles, Califórnia 90024.](#)
- [Figura 11. Notação de Elmasri & Navathe](#)
- [Figura 12. Imagem feita provavelmente à mão](#)
- [Figura 13. Tela do PowerDesigner - Área de trabalho](#)
- [Figura 14. Tela do PowerDesigner – Configuração de especialização em um modelo conceitual](#)
- [Figura 15. Tela inicial do ERwin – Não há opção de criação de modelo conceitual](#)
- [Figura 16. Tela do ERwin – Modelando a base de dados de uma biblioteca.](#)
- [Figura 17. Tela do ErWin – Edição de campos de uma tabela.](#)
- [Figura 18. Tela do ERwin – Relacionamentos: Definido a cardinalidade.](#)
- [Figura 19. Tela do ERwin – Gera esquemas de implementação para diversos bancos de dados.](#)
- [Figura 20. Tela do DBDesigner – Vários Bancos de dados.](#)
- [Figura 21. Tela do DBDesigner – Direto ao modelo lógico.](#)
- [Figura 22. Tela do DBDesigner – Trabalho sincronizado com banco de dados.](#)
- [Figura 23. Tela do DBDesigner – Export par a ERWin.](#)
- [Figura 24. Tela do DBDesigner – Modelo salvo em html/xml/xstl.](#)
- [Figura 25. Python Database Designer](#)
- [Figura 26. Tela inicial do MS-VISO](#)
- [Figura 27. Diagrama no VISIO – diagramas de modelo conceitual e lógico na mesma tela.](#)
- [Figura 28. Diagrama no VISIO – Notas/Dicionário de dados.](#)
- [Figura 29. A esquerda implementação do modelo do Dr. Heuser no brModelo. À direita notação original.](#)
- [Figura 30. brModelo – Implementa os diagramas com sutis diferenças se comparada a notação original.](#)
- [Figura 31. brModelo – Implementa os diagramas com sutis diferenças se comparada a notação original.](#)
- [Figura 32. brModelo – Resultado da conversão.](#)
- [Figura 33. brModelo – Atribudos.](#)
- [Figura 34. brModelo – Dicionário de dados.](#)
- [Figura 35. brModelo – Dicionário de dados – Reunião dos dicionários em um documento único.](#)
- [Figura 36. brModelo – Interação com o usuário no processo de conversão.](#)

[Figura 37. brModelo – processo de conversão \(A\).](#)

[Figura 38. brModelo – processo de conversão \(B1\).](#)

[Figura 39. brModelo – processo de conversão \(B2\).](#)

[Figura 40. brModelo – modelo lógico com cardinalidade.](#)

[Figura 41. Internet Explorer: brModelo – modelo lógico conforme Dr. Heuser.](#)

## **RESUMO**

## **ABSTRACT**

# **1. INTRODUÇÃO**

## **1.1. Apresentação**

A escolha do tema desta dissertação, *aprendizagem em banco de dados, implementação de ferramenta de modelagem ER*, foi motivada pela inexistência, até o momento, de uma ferramenta voltada para o ensino de modelagem de dados em banco relacional que implemente exatamente os conceitos de criação de modelos de uma forma didática, simples, clara e de fácil assimilação de forma independente o Sistema Gerenciador de Banco de Dados adotado.

Observa-se, por conta disso, que a base desta dissertação, são as demonstrações das implementações dos principais conceitos defendidos e publicados pelo Dr. Carlos Aberto Heuser em seu livro “Projeto de Banco de Dados” – 4.º edição, além de comparar os princípios e as notações presentes nas principais ferramentas usadas para modelar bancos de dados com a ferramenta desenvolvida - brModelo.

### **1.1.1. Abordagem Relacional**

Embora o desenvolvimento de software esteja fortemente tendenciado aos modelos orientado a objeto e, como consequência, exista a possibilidade de que os bancos de dados sigam o mesmo caminho. Não esperamos<sup>[1]</sup> uma mudança significativa nos próximos dez anos, o que me faz acreditar que os bancos relacionais continuarão dominando o mercado de forma esmagadora ainda por muito tempo.

Por isso a ferramenta desenvolvida implementa modelagem para bancos relacionais.

### **1.1.2. Notação de Peter Chen**

Entendo que um bom esquema deve apresentar uma notação simples, limpa e de fácil entendimento.

Ao observar as diversas notações utilizadas para criação de modelos relacionais, concluí que a forma mais adequada ao ensino é a que foi proposta pelo Dr. Peter Chen (1976) com as alterações introduzidas pelo Dr. Carlos Alberto Heuser.

## **1.2. Problemática**

A maior parte das ferramentas analisadas nesta pesquisa implementam seu próprio diagrama de desenho (notação) para a construção dos modelos conceitual e lógico.

Isto diminui sensivelmente o trabalho de desenvolvimento, já que formas de desenho mais complexas são evitadas.

No caso dos freewares analisados, observa-se que muitos, se quer, possibilitam a construção de modelos conceituais. Partem diretamente para o modelo lógico, descartando um importante passo: a conceitualização do problema; o que pode até ser admitido em sistemas pequenos, construídos por profissional com grande experiência em modelagem de dados, mas, que é inadmissível na fase de aprendizagem.

## **1.3. Limitações do tema**

Foge do escopo desta dissertação ensinar como modelar um banco de dados, bem como, não se especifica como a informação deve, fisicamente, ser armazenada em um SGBD.

Na comparação entre as principais ferramentas de modelagem do mercado, o item “facilidade de uso” não foi avaliado já que este aspecto é um tanto quanto cognitivo, dependendo diretamente da experiência do avaliador com a ferramenta, e é estritamente dependente da versão do produto avaliado.

A ferramenta desenvolvida ainda não gera código para implementação, limita-se a construção dos esquemas conceitual e lógico. Ela também não tem a ambição de ser melhor que ferramentas maduras como ERWin, OracleDesigner ou PowerDesigner mas apenas de ser mais fiel à notação proposta sendo, desta forma, mais didática.

## 1.4. Objetivo

O objetivo desta monografia é justificar a necessidade de desenvolvimento da ferramenta de modelagem (codinome brModelo) através da comparação de seus recursos de criação de esquemas conceitual e lógico, dentro da abordagem defendida pelo Dr. Carlos A. Heuser, e os disponíveis nas principais ferramentas de modelagem do mercado.

## 1.5. Justificativa

Não foi encontrada uma ferramenta para modelagem de dados que utilize a notação do Dr. Peter Chen com as modificações realizadas pelo Professor Dr. Observei que mesmo no site do Professor Heuser há a indicação de uma ferramenta CASE chamada dbMain (fig. 01) que não utiliza com precisão a notação do autor em questão.

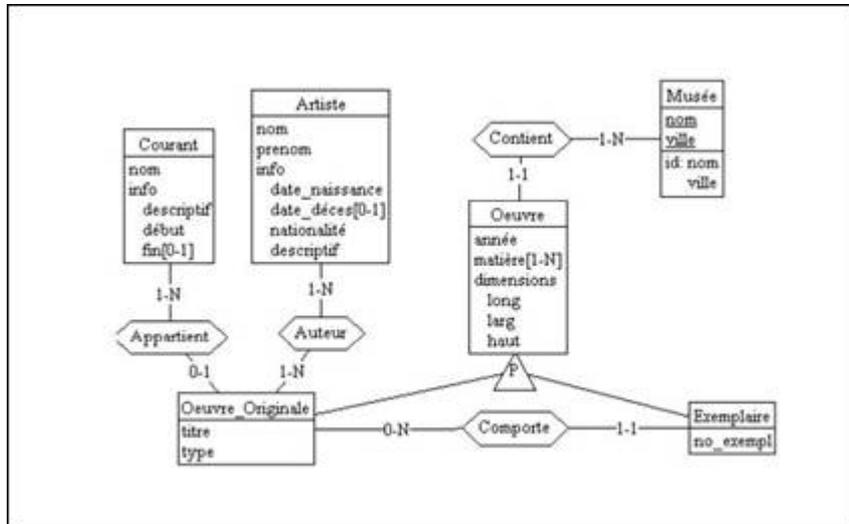
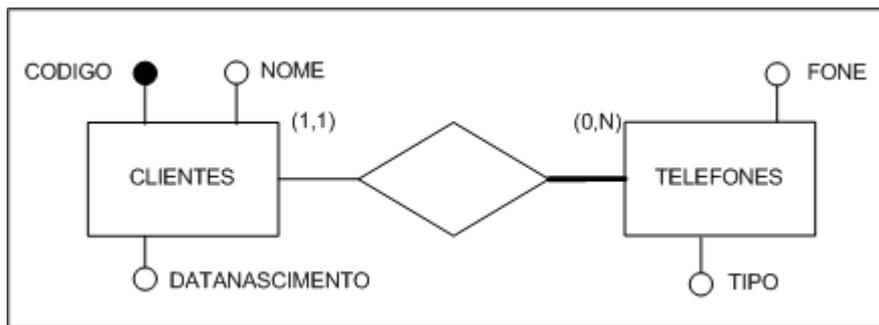


Figura 1. Modelo gerado no dbMain: software sugerido por Heuser.  
Fonte: [www.dbmain.com](http://www.dbmain.com)

Analisando as publicações sobre o assunto na revista SQL Magazine® encontrei artigos sobre modelagem de dados que se utilizam da notação do Dr. Heuser, porém, ao questionar sobre a forma ou ferramenta utilizada para a confecção dos diagramas, a resposta foi surpreendentemente o MS Word®.



**Figura 2.** Imagem editada para SQL Magazine.  
Fonte: Reinaldo V. Álvares

Vale lembrar que a revista em questão é uma das mais lidas por profissionais da área de banco de dados no país.

Observa-se, desta forma, a carência de uma ferramenta nos moldes do brModelo e, por conta disso, o desenvolvemos. E Como regra para conclusão e aprovação na especialização em banco de dados pleiteada, é que escrevo esta monografia.

## 1.6. Hipótese.

As alterações realizadas pelo Dr. Carlos A. Heuser no diagrama de entidade e relacionamento apresentado pelo Dr. P. P. Chen em 1976, tornaram a notação no modelo conceitual mais fácil de ser entendida e aplicável às mais recentes técnicas de modelagem de dados.

## 2. MODELAGEM DE DADOS

Um modelo de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados<sup>[2]</sup>.

Dentre as técnicas utilizadas para construção dos modelos utiliza-se “linguagem de modelagem de dados”, que podem ser gráficas ou textuais.

Até a década passada era comum o uso de linguagem de modelagem textual, hoje,

graças às ferramentas de edição, estes tipos de modelos não são mais comuns.

## **2.1. Nível de abstração do modelo**

A quantidade de representações abstraídas de um modelo é que define seu nível de abstração.

Usualmente, em banco de dados, trabalha-se com três níveis de abstração, denominados modelo conceitual, modelo lógico e modelo físico.

O três níveis de abstração estão hierarquicamente organizados da seguinte forma:

1. O modelo conceitual trata os conceitos fundamentais abstraídos do mundo real, por tanto, independe da arquitetura do banco de dados;
2. O modelo lógico está ligado ao tipo de banco de dados (objeto, relacional ou hierárquico, por exemplo);
3. O modelo físico está diretamente ligado ao banco de dados (Oracle, MySQL, Sybase).

## **2.2. Modelo Conceitual.**

A maior dificuldade para o aprendizado das técnicas de modelagem em banco de dados é entender um problema do mundo real e convertê-lo, criando uma solução.

O profissional da área de informática precisa entender o problema e conceituar o que será a solução e, para isso, duas coisas podem ser consideradas imprescindíveis:

- a) saber ouvir o cliente/usuário abstraindo da conversa o que é realmente útil para implementar a solução;
- b) conhecer as técnicas de modelagem a fim de representar o problema de forma conceitual antes de iniciar a implementação.

## **2.3. Modelo lógico**

O modelo lógico é o resultado ou produto da conversão de um modelo conceitual para um determinado tipo de banco de dados, ou conforme Heuser <sup>[3]</sup>, “Um modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do sistema gerenciador

de banco de dados”.

Por isso, nesta fase do processo de modelagem de dados, o projetista já deve ter conhecimento do tipo de banco de dados no qual o projeto será implementado (relacional, hierárquico, objeto - relacional, entre outros).

## 2.4. Modelo físico

É o modelo que descreve a forma como os dados são armazenados no SGBD, nesta fase, o modelo lógico é convertido, no caso dos bancos relacionais em linguagem DDL (data description language) e as regras descritas no modelo conceitual são convertidas em regras de integridade.

Dentre os tipos de bancos de dados, os mais comuns são os relacionais, embora seja crescente a quantidade dos bancos de dados orientados a objetos.

Neste contexto, nosso escopo limita-se aos hegemônicos bancos relacionais e a forma de modelá-los.

## 3. ABORDAGEM RELACIONAL.

Nosso mundo é guiado pela informação!

A frase acima retrata a grande importância dos bancos de dados, já que dados, nada mais são do que informações armazenadas nos conceitos da informática - ciência que estuda o tratamento automático e racional da informação<sup>[4]</sup>.

Um Banco de Dados é uma coleção de dados com algum significado e objetivo. Conforme Reinaldo Viana Álvares (2004)[7], um catálogo telefônico pode ser considerado um banco de dados.

A maneira com a informação encontra-se armazenada é que dita sua disponibilidade. Por tanto, o processo de armazenagem deve ser preciso a fim de garantir e facilitar o processo de recuperação.

Para garantir que uma informação armazenada possa ser recuperada devemos nos assegurar da eficiência do processo de armazenamento e recuperação. A **abordagem relacional (entidade-relacionamento)** apresentada pelo Dr. Peter Pin-Chan Chen (1976)[9] é hoje uma das formas comprovadamente segura e eficaz de armazenamento e recuperação da informação.

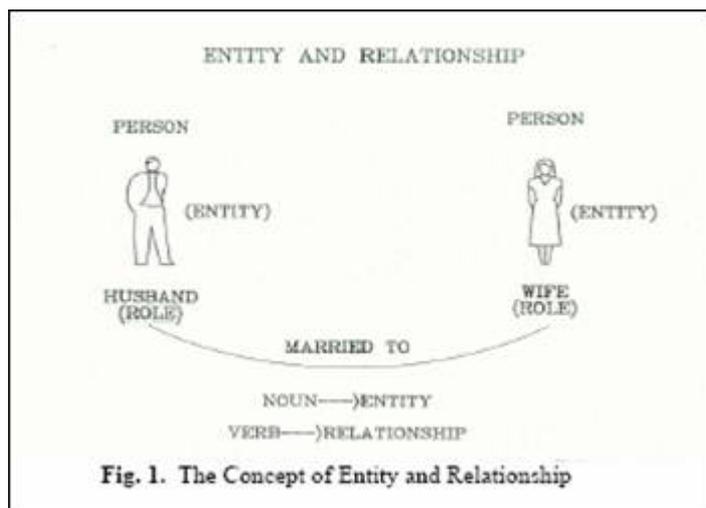


Figura 3. Conceito de entidade e relacionamento - Peter Chen [9]

### 3.1. O Modelo entidade-relacionamento (MER) – P. P. Chen

Neste modelo as informações são armazenadas em tabelas que podem ser interrelacionadas através de operações de conjunto.

The Entity-Relationship Model

ATTRIBUTE ENTITY SET AND VALUE SET	$f_1$	$f_2$	$f_3$	$f_4$
	(EMPLOYEE-NO)	(NAME)	(ALTERNATIVE-NAME)	(AGE)
$E_1$ (EMPLOYEE)	$v_1$ (EMPLOYEE-NO)	$v_2$ (FIRST-NAME)	$v_3$ (LAST-NAME)	$v_4$ (NO-OF-YEARS)
$e_1$	$v_{11}$ (2566)	$v_{21}$ (PETER)	$v_{31}$ (JONES)	$v_{41}$ (25)
$e_2$	$v_{12}$ (3378)	$v_{23}$ (MARY)	$v_{32}$ (CHEN)	$v_{42}$ (23)
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Figura 4. Uma entidade: visão em um conjunto de registro [5]

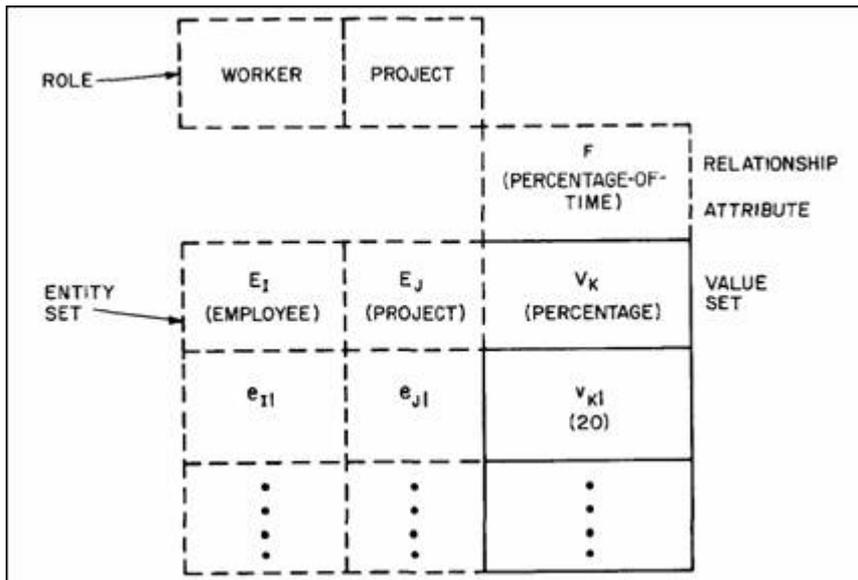


Figura 5. Relacionamento <sup>7</sup>.

Este verdadeiro padrão de modelagem de dados descreve as técnicas principais para criação de bases de dados relacionais.

Este modelo quando apresentado graficamente (diagrama), recebe o nome de Diagrama de Entidade e Relacionamento - DER.

### 3.2. Diagrama de entidade-relacionamento – DER

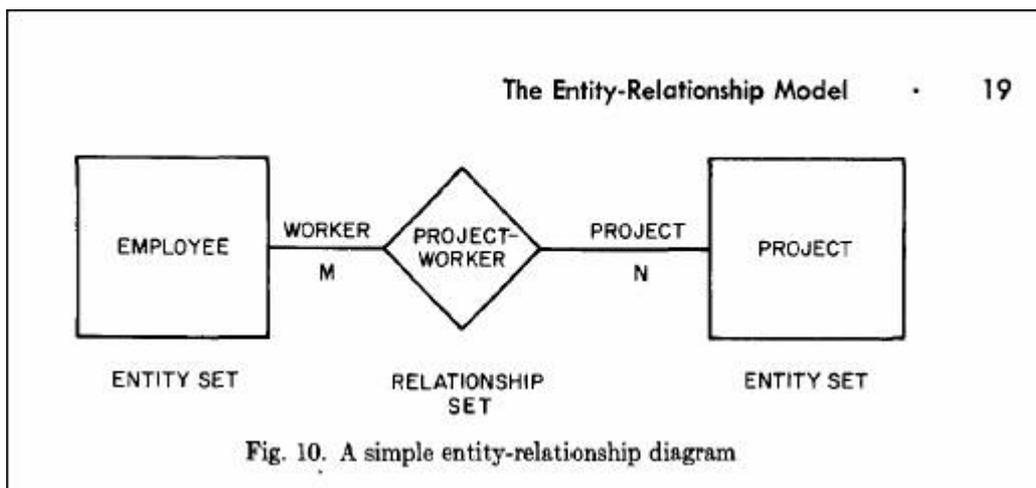


Figura 6. Notação original do Dr. Chen (1976)[10] – Sem atributos.

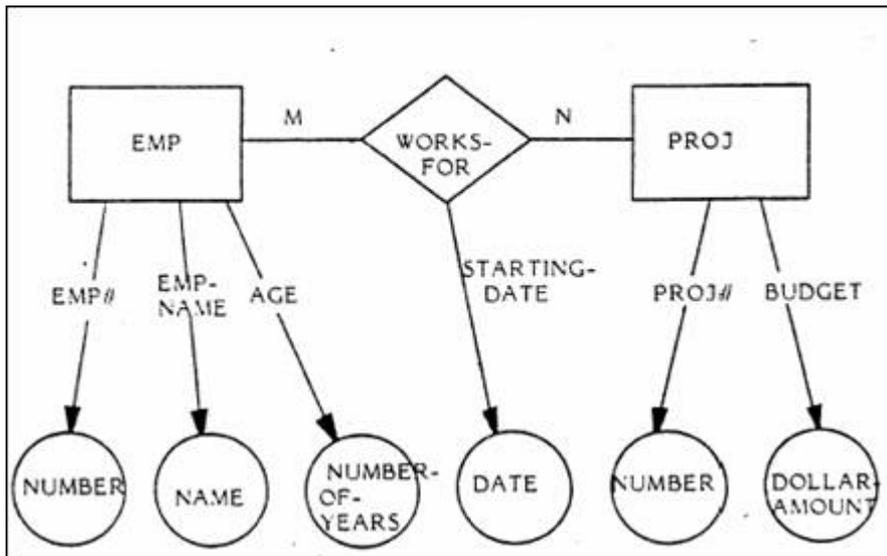


Figura 7. Notação original do Dr. Chen (1976)[10] – Com atributos.

A notação original do Dr. Peter Chen sofreu mudanças ao longo do tempo. Algumas facilitaram o entendimento e a clareza dos diagramas como a da figura abaixo:

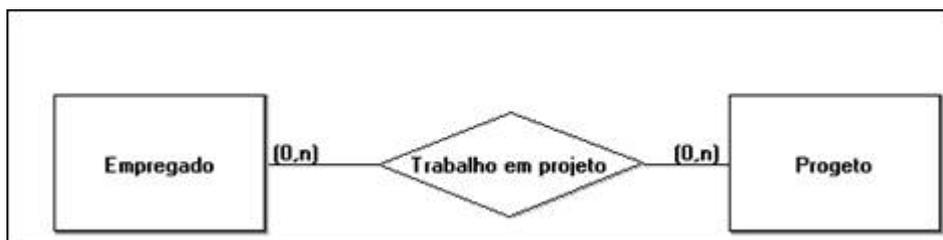


Figura 8. brModelo – Notação de Heuser (2001)[3] – Sem atributos

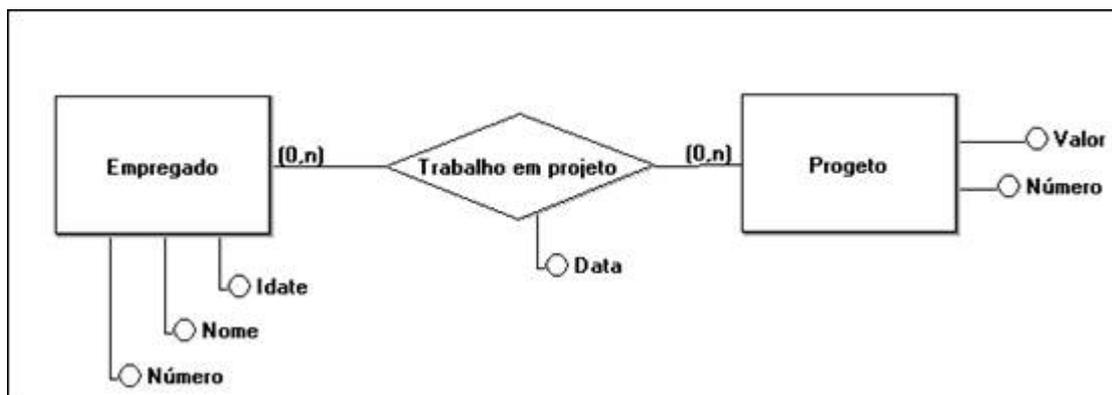


Figura 9. brModelo – Notação de Heuser (2001)[3] – Com atributos

Existem várias formas de representar graficamente o modelo relacional, algumas desenvolvidas em centros de pesquisa, outras, fruto de padrões estabelecidos por fabricantes de ferramentas e ainda uma terceira, que é o resultado do trabalho acadêmico somado ou modificado pelas diferentes formas de implementação adotadas pelos produtores das ferramentas de

modelagem.

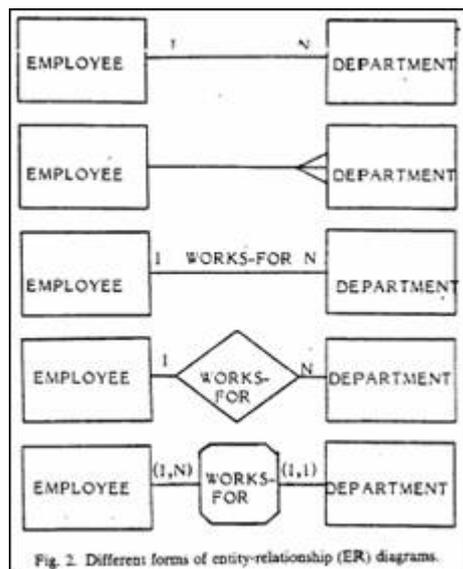


Figura 10. Foras de modelar – Universidade da Califórnia, Los Angeles, Califórnia 90024.

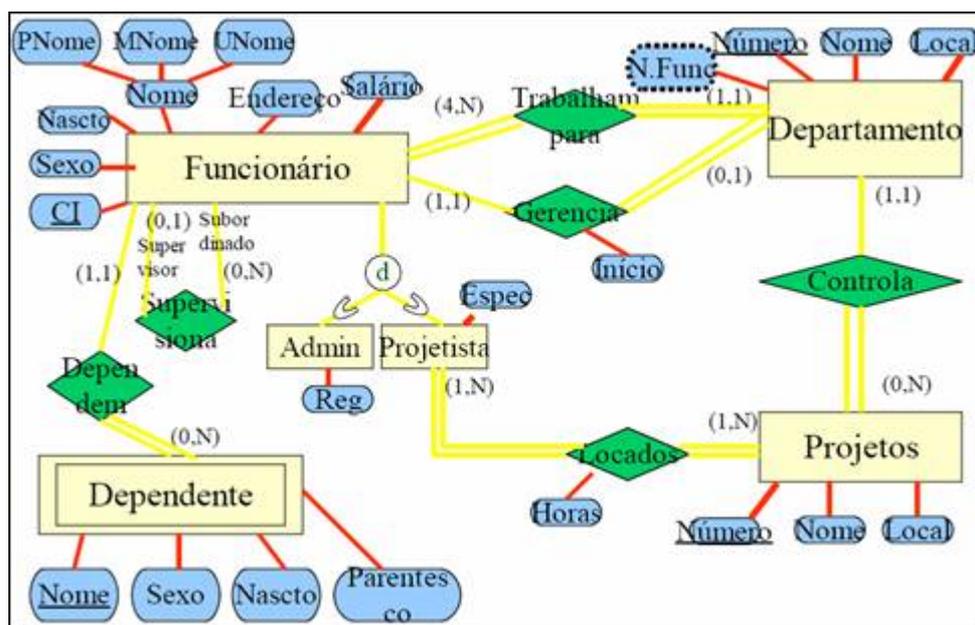


Figura 11. Notação de Elmasri & Navathe <sup>[6]</sup>

#### 4. FERRAMENTAS DE CONSTRUÇÃO DE MODELOS/ESQUEMAS

O ensino e a aprendizagem das técnicas de modelagem de banco de dados é perfeitamente possível mesmo sem o uso de uma ferramenta CASE (Computer Aided Software Engineering).

No entanto, é extremamente inconveniente o trabalho manual de desenho dos diagramas. Além disso, a conversão entre esquemas (esquema conceitual para esquema lógico), se

feito manualmente, implica no exaustivo e estressante trabalho de redesenho das formas.

Com o uso de uma ferramenta CASE pode-se observar as alterações do modelo à medida que as decisões forem tomadas – processo “causa-efeito”.

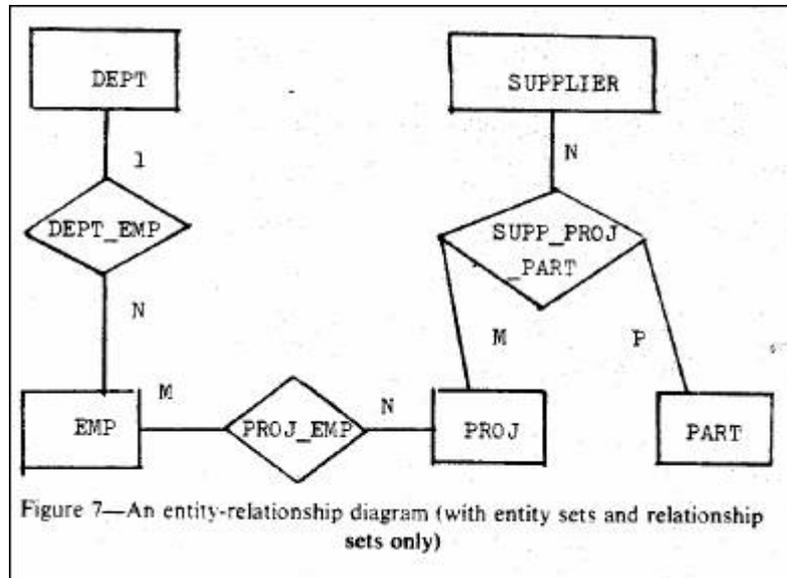


Figura 12. Imagem feita provavelmente à mão<sup>[7]</sup>

#### 4.1. A ferramenta CASE ideal

Segundo Carlos A. Heuser (2001) [3], uma ferramenta CASE deve ter a capacidade de edição diagramática, dicionário de dados e integração entre o diagrama ER e o dicionário de dados.

Na opinião do Dr. Ronaldo Mello, além das definições do Dr. Heuser, a ferramenta deve possibilitar um nível mínimo de interação com o analista/usuário no momento das tomadas de decisão.

Em conformidade com as opiniões acima, entendo que além do exposto, o sistema deve ser fiel a notação a que se propõe implementar.

É com base nestes conceitos que foi realizada a análise das principais ferramentas de modelagem de dados disponíveis no mercado e a comparação com a aplicação desenvolvida.

#### 4.2. Análise das ferramentas disponíveis no mercado.

Existe uma grande quantidade de ferramentas que auxiliam na modelagem de banco de dados. Algumas tão completas que fazem desde a modelagem conceitual até a implementação, tratando inclusive da integridade referencial.

O nível de maturidade destas ferramentas é altíssimo e muitas implementam, inclusive, técnicas de engenharia reversa como o ERWin, OracleDesign, PowerDesign e o DBDesign.

Neste mesmo contexto, algumas destas ferramentas são direcionadas a um único banco de dados como é o caso do aplicativo OracleDesign (Oracle ®).

Existem outras cuja finalidade é apenas documentar a construção do esquema e, como exemplo cito o aplicativo MS Visio.

Para análise, foram escolhidas as aplicações mais conhecidas entre os desenvolvedores<sup>[8]</sup>, agrupadas da seguinte forma:

- a) Uso exclusivo em um banco de dados específico: OracleDesigner™ (Oracle ®);
- b) Mais utilizadas no mercado: PowerDesigner™ (Sybase ®), e ERWin (CA ®);
- c) Freeware: DBDesigner e PyDesigner
- d) Desenho: VISIO™ (Microsoft ®)

Outras ferramentas como a UCASE2 e a PowerModel também foram analisadas, mas por falta de informação a respeito do funcionamento ou inexistência de versão demo não figuram nesta monografia.

Os quesitos de avaliação adotados foram:

- a) Notação utilizada;
- b) Capacidade de gerar modelo conceitual;
- c) Capacidade de conversão: modelo conceitual para modelo lógico;
- d) Nível de controle do usuário sobre o modelo e sobre o processo de conversão (de modelo conceitual para modelo lógico).

#### **4.2.1. A análise**

##### **4.2.1.1. Oracle – OracleDesigner**

- a) Quanto à notação:  
Utiliza notação própria.

b) Modelo conceitual:

Não consegui identificar o modelo conceitual. Informações obtidas na Oracle ® descrevem tal característica na versão 9.0 da ferramenta.

c) e d) Não há interação com o usuário na conversão do modelo.

e) Observação:

1 - É preciso utilizar um repositório de dados para trabalhar com a ferramenta, razão esta, da limitação dos testes realizados.

2 - Versão de avaliação disponível no site da empresa (apenas para usuários OTN)

– [www.oracle.com](http://www.oracle.com).

#### 4.2.1.2. Sybase – PowerDesigner

Uma das ferramentas mais utilizadas e completas do mercado.

Gera modelo conceitual, converte para modelo lógico e trabalha com todos os principais bancos de dados disponíveis empregando inclusive, técnicas de engenharia reversa e de integridade referencial.

Não interage com o usuário na conversão entre os esquemas lógico e conceitual.

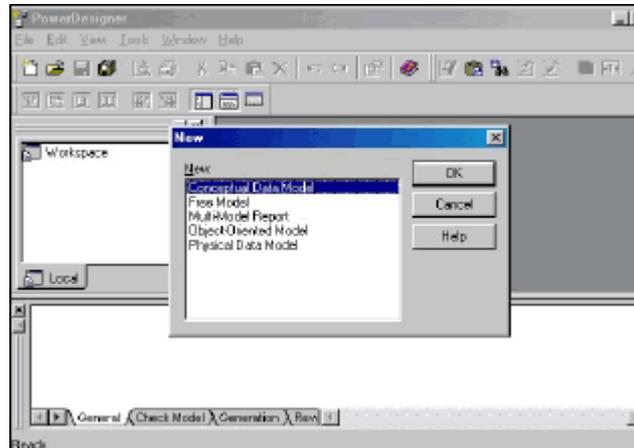
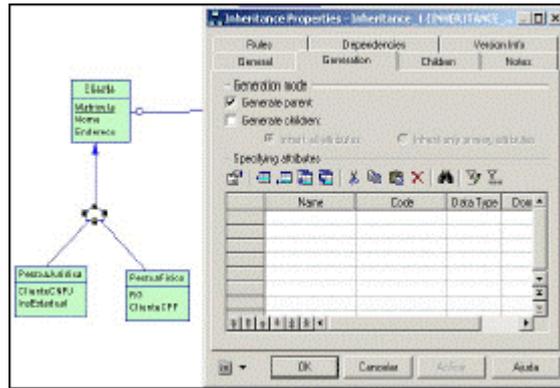


Figura 13. Tela do PowerDesigner - Área de trabalho<sup>[9]</sup>



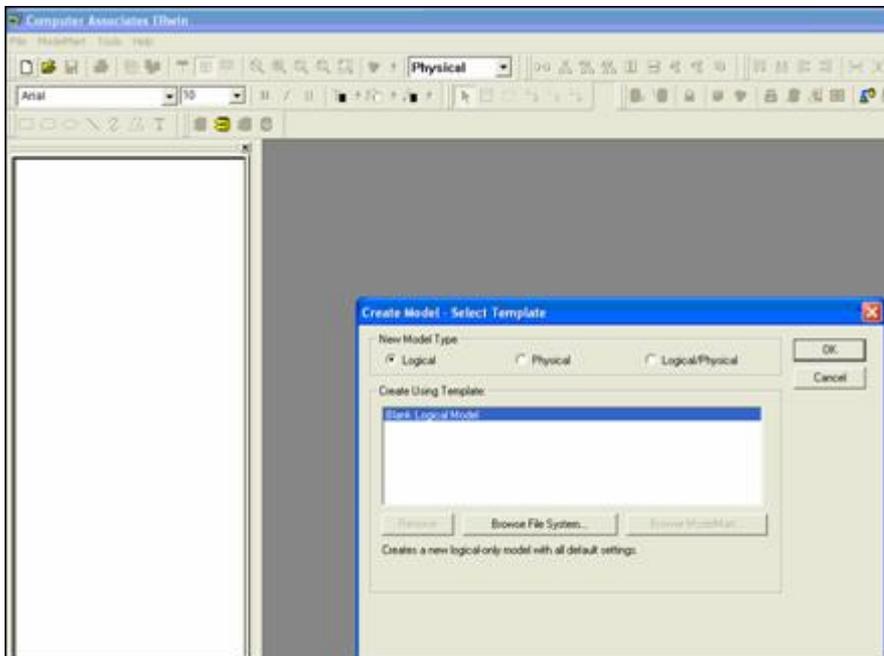
**Figura 14. Tela do PowerDesigner – Configuração de especialização em um modelo conceitual <sup>[10]</sup>**

Versão demo disponível no site: [www.sybase.com](http://www.sybase.com) (15 dias).

#### 4.2.1.3. CA - ERwin

Esta é a ferramenta mais utilizada no mercado, conforme informado no site do fabricante.

É direcionada para profissionais experientes e sua principal deficiência está no fato de não gerar modelo conceitual.



**Figura 15. Tela inicial do ERwin – Não há opção de criação de modelo conceitual**

Devido à liderança desta ferramenta foi dada atenção especial nesta avaliação conforme figuras abaixo:

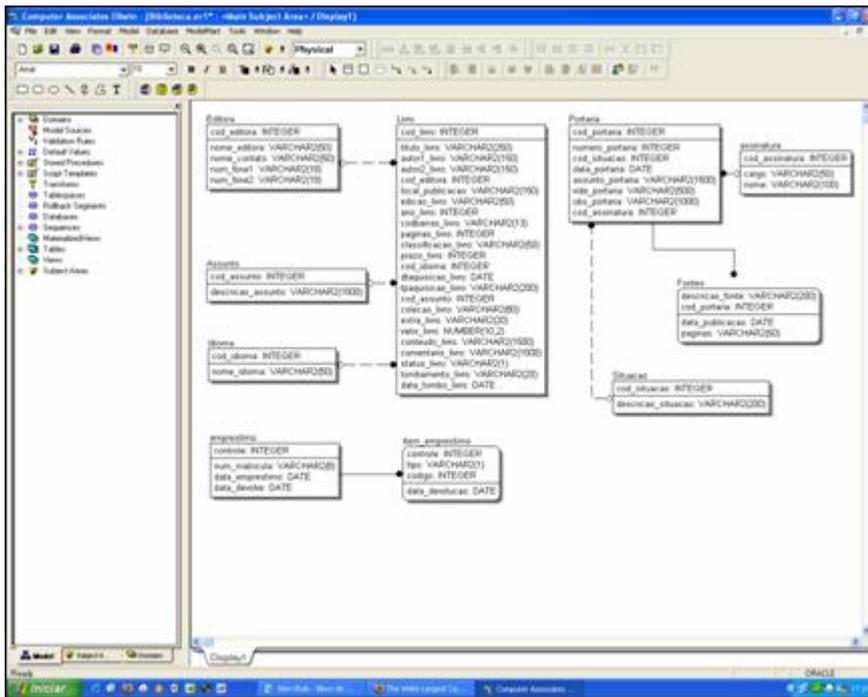


Figura 16. Tela do Erwin – Modelando a base de dados de uma biblioteca.

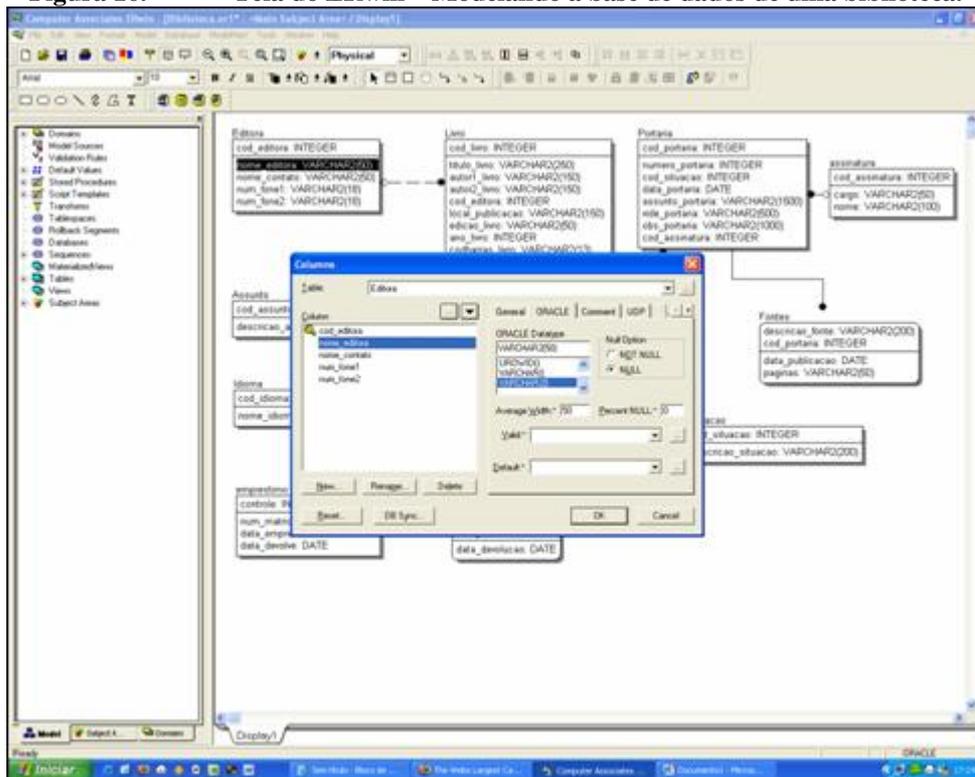


Figura 17. Tela do Erwin – Edição de campos de uma tabela.

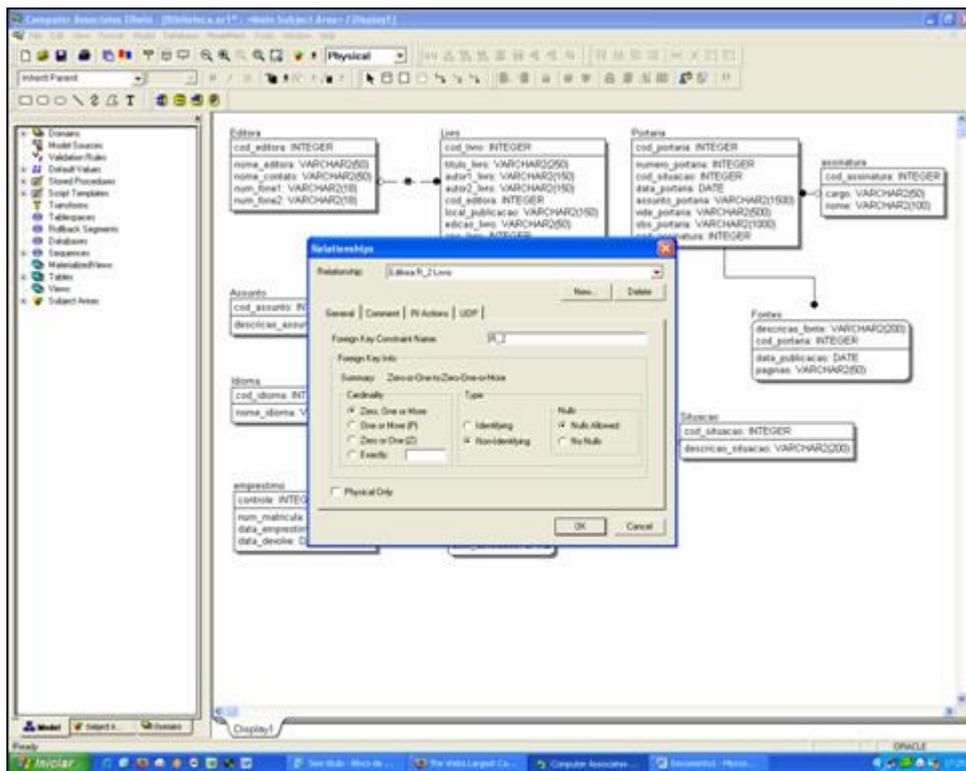


Figura 18. Tela do ERwin – Relacionamentos: Definido a cardinalidade.

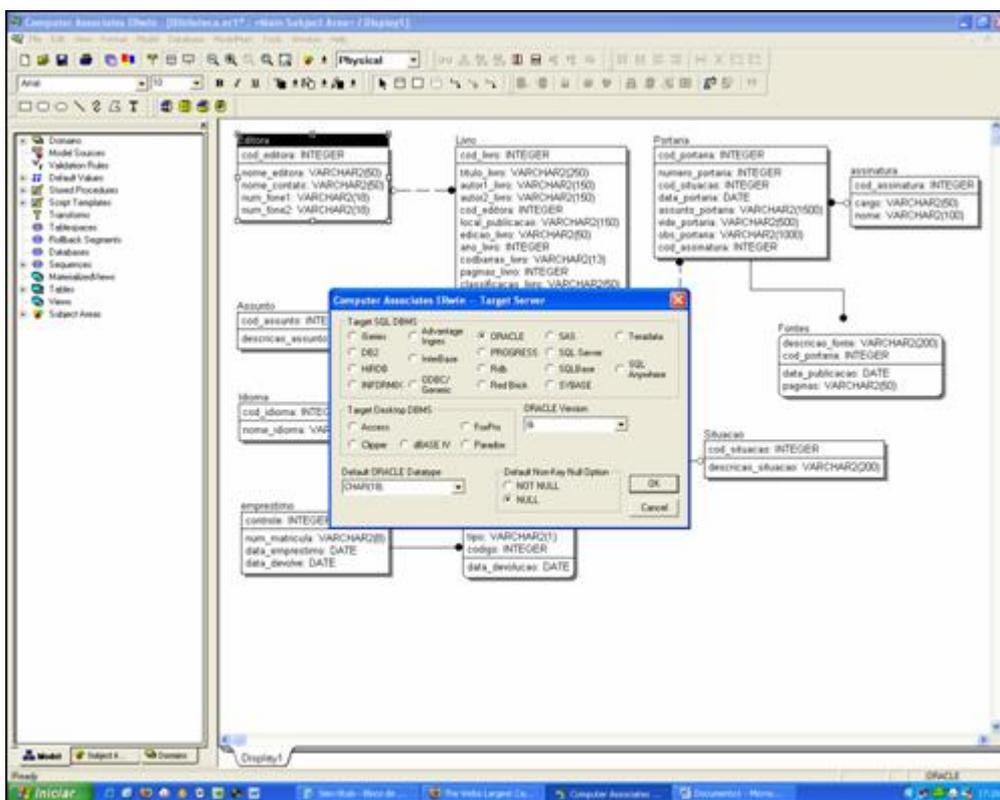


Figura 19. Tela do ERwin – Gera esquemas de implementação para diversos bancos de dados.

O ERwin gera modelos para todos os principais bancos de dados empregando

inclusive, técnicas de engenharia reversa e de integridade referencial.

A notação utilizada pelo ERwin não descende da notação original do Dr. Chen, ela implementa diagramas parecidos com os utilizados na engenharia de software.

#### 4.2.1.4. Freeware DBDesigner

No mundo dos freewares com código fonte para a plataforma Windows ® e Linux (GNU GPL), a ferramenta pesquisada foi a DBDesigner.

Alguns dos conceitos de desenho da ferramenta brModelo foram expirados nesta ferramenta já que ambas foram desenvolvidas em Borland® Delphi™.

Seu principal problema é o fato de não implementar modelo conceitual, o que a torna quase sem utilidade como ferramenta didática.

Contudo, trata-se de uma ferramenta quase completa para a área afim. Ela gera modelo lógico com diagramas próprios além de gerar modelo físico para diversos bancos de dados implementando regras de integridade e até mesmo técnicas de engenharia reversa (MySQL).

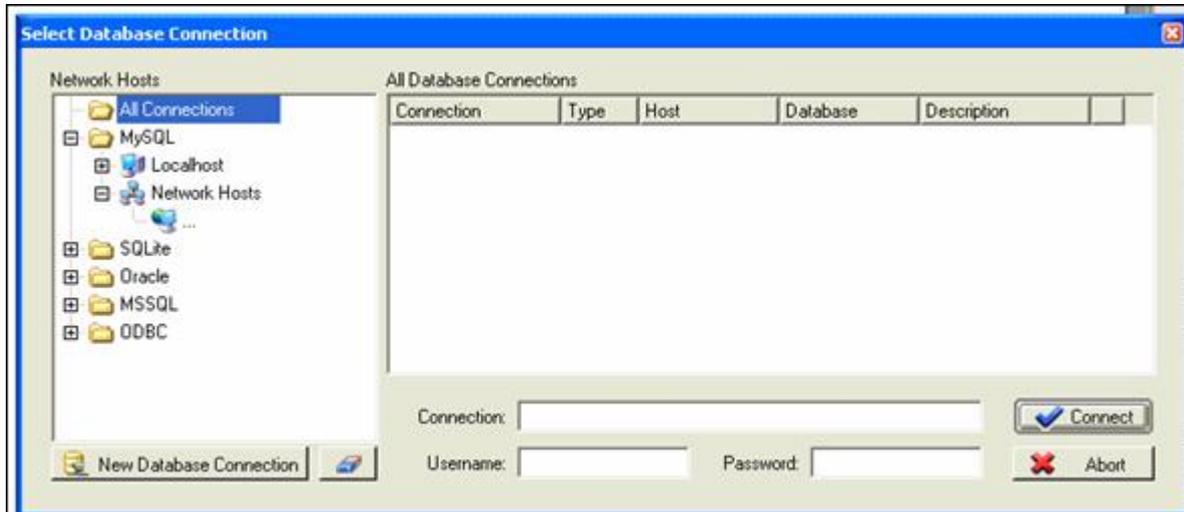


Figura 20. Tela do DBDesigner – Vários Bancos de dados.

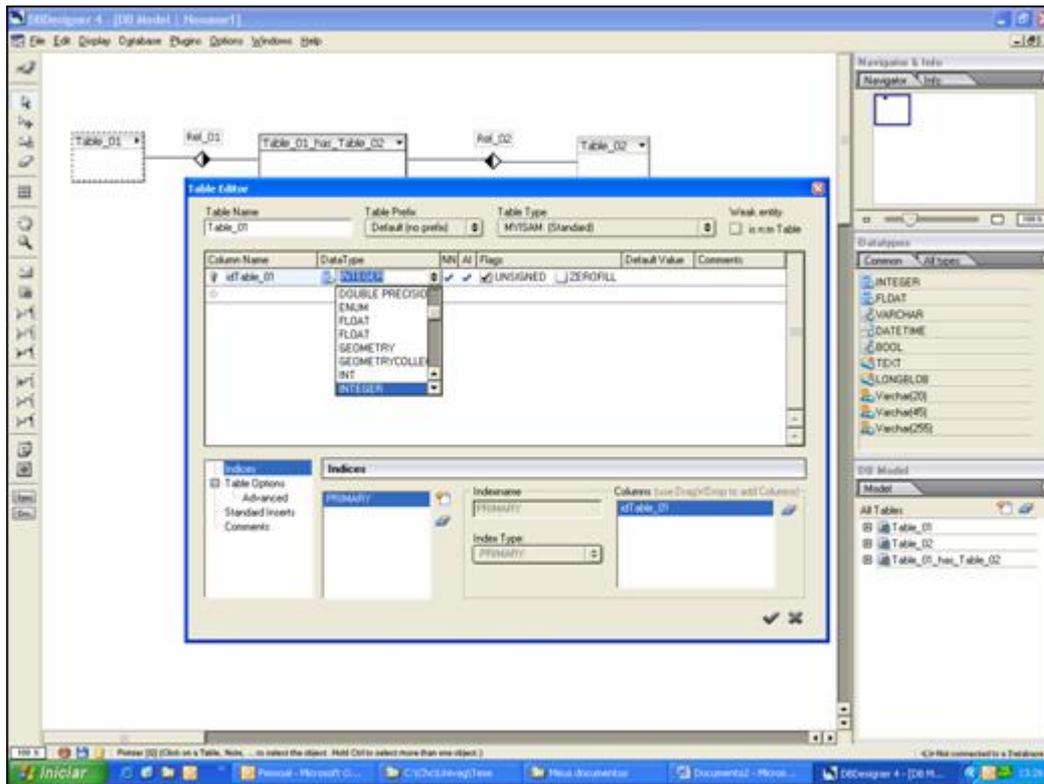


Figura 21. Tela do DBDesigner – Direto ao modelo lógico.

Esta ferramenta permite o trabalho sincronizado com o banco de dados.

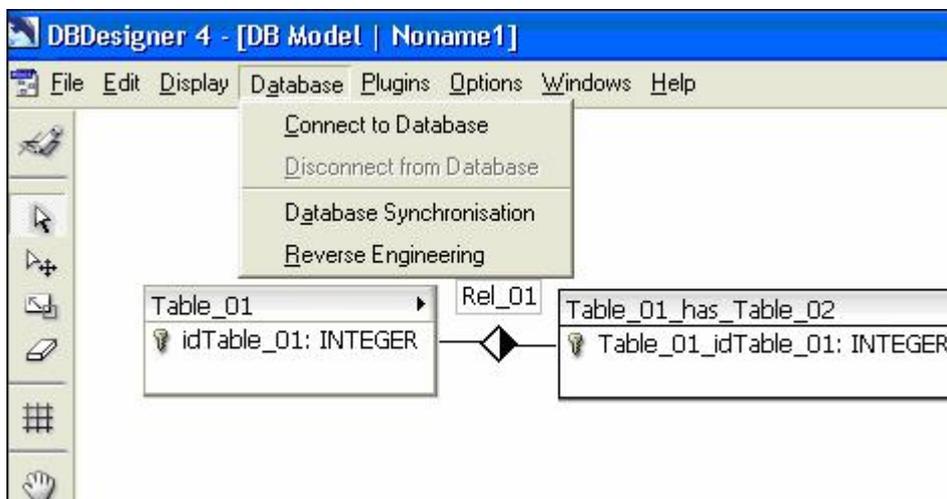


Figura 22. Tela do DBDesigner – Trabalho sincronizado com banco de dados.

O DBDesigner não é uma ferramenta adequada para o ensino de técnicas de modelagem de banco de dados por não implementar seus principais conceitos, mas é uma excelente aplicação para desenvolvimento devido a sua grande quantidade de recursos dentre os quais destaca-se a capacidade de importar modelos do ERwin.

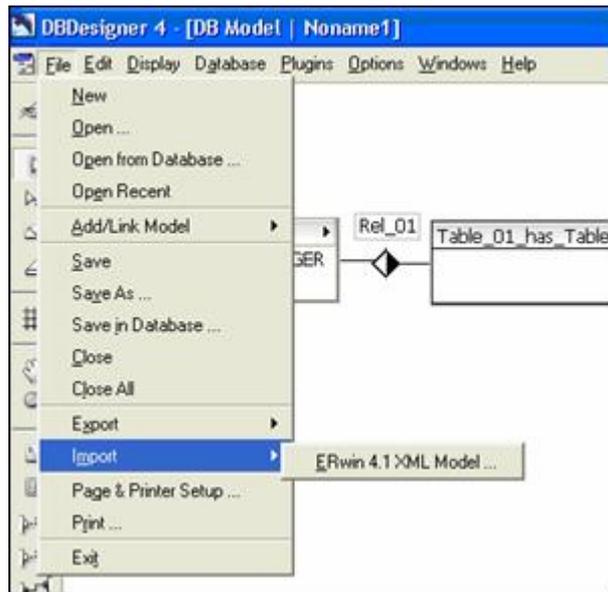


Figura 23. Tela do DBDesigner – Export par a ERWin.

A capacidade de exportar o modelo em vários formatos também é um de seus destaques.

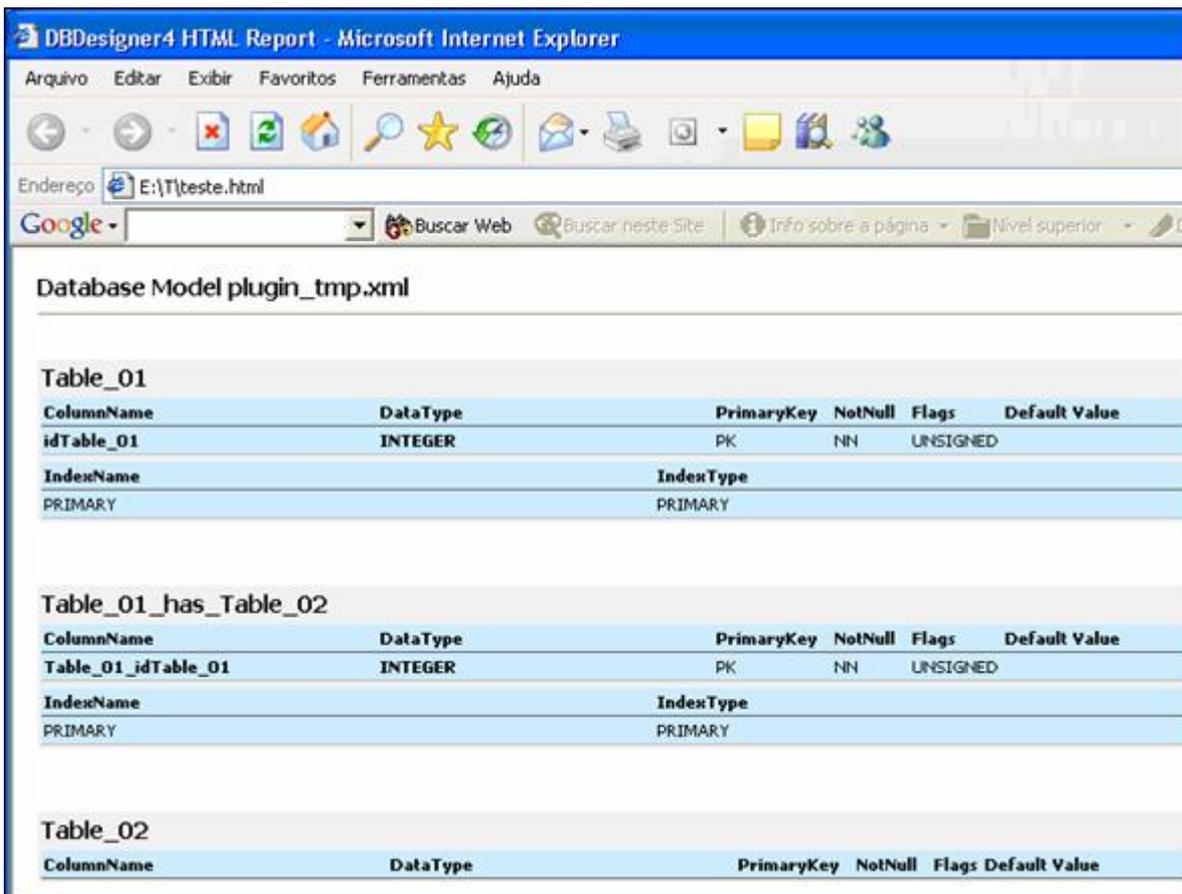


Figura 24. Tela do DBDesigner – Modelo salvo em html/xml/xstl.

Fabricante: <http://www.fabforce.net/>

#### 4.2.1.5. PyDesigner – Python Database Designer

Não consegui informações suficientes para realizar uma avaliação completa desta ferramenta. Decidi citá-la aqui apenas para constar, já que a mesma está disponível para a plataforma Linux e é distribuída sob as regras de licença GPL.

Uma das informações obtidas no site do grupo que a mantém, diz que a mesma não gera modelo conceitual (suficiente para esta avaliação).

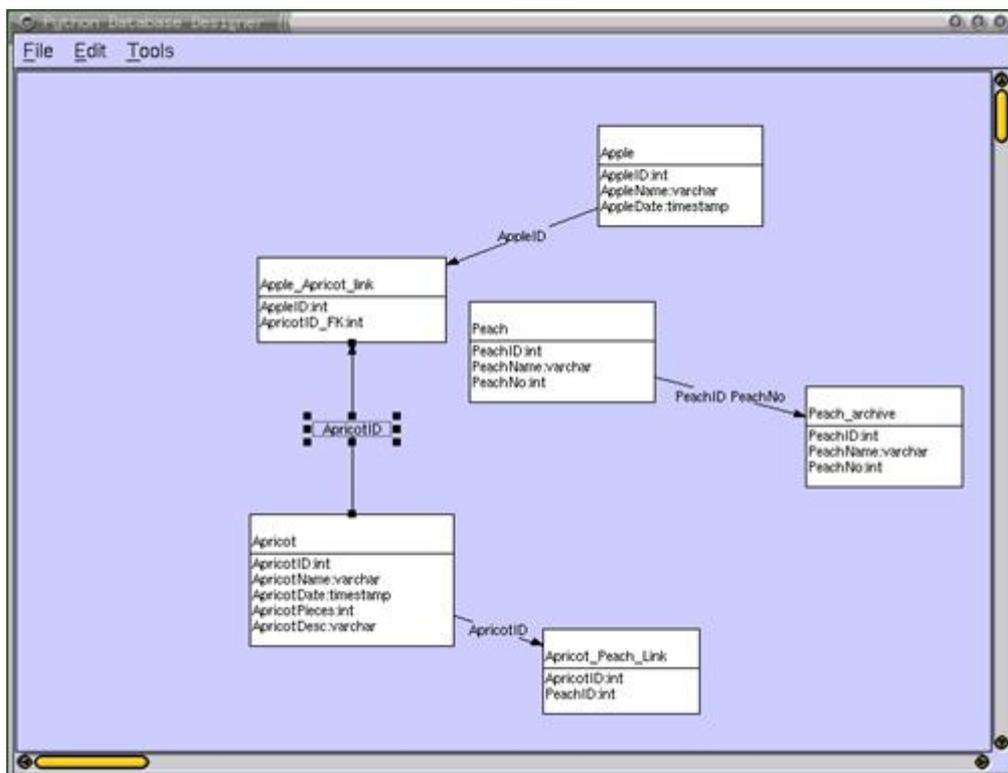


Figura 25. Python Database Designer

#### 4.2.1.6. Microsoft Visio

O VISIO é uma ferramenta direcionada exclusivamente para desenho, não há interação com banco de dados propriamente dito. Seu esquema de desenho depende do diagrama carregado – existem, na internet, diagramas com a notação original de Peter P. Chen para download.

Seu diagrama padrão não implementa os conceitos de modelagem conceitual e a ferramenta auxilia apenas na diagramação (desenho), não atuando de nenhuma forma sob os

conceitos da construção de modelos de entidade e relacionamento.

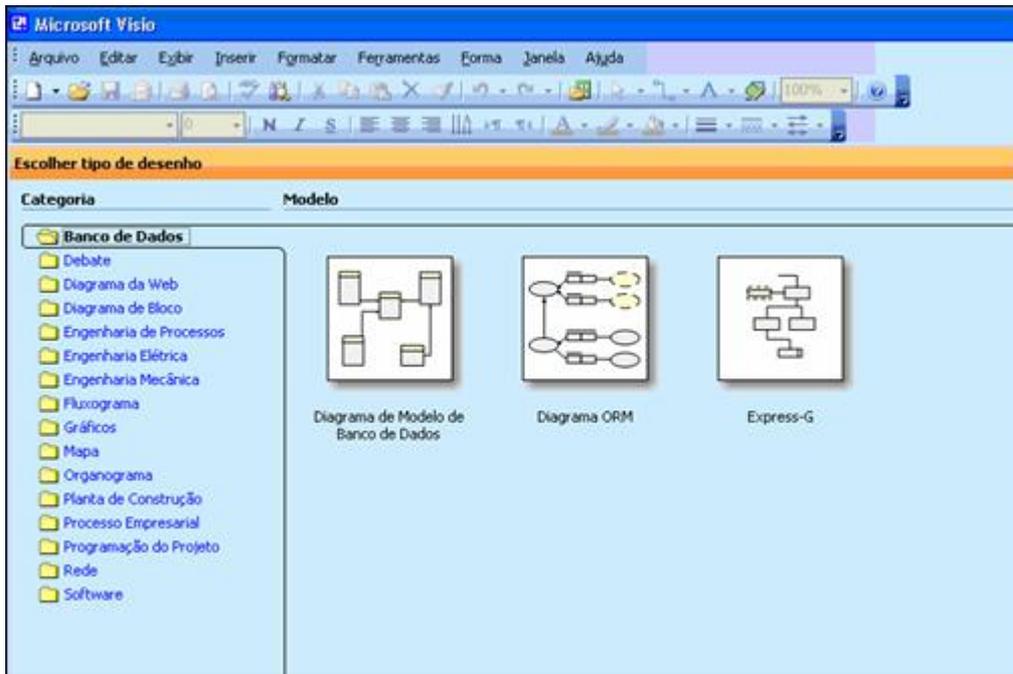


Figura 26. Tela inicial do MS-VISO

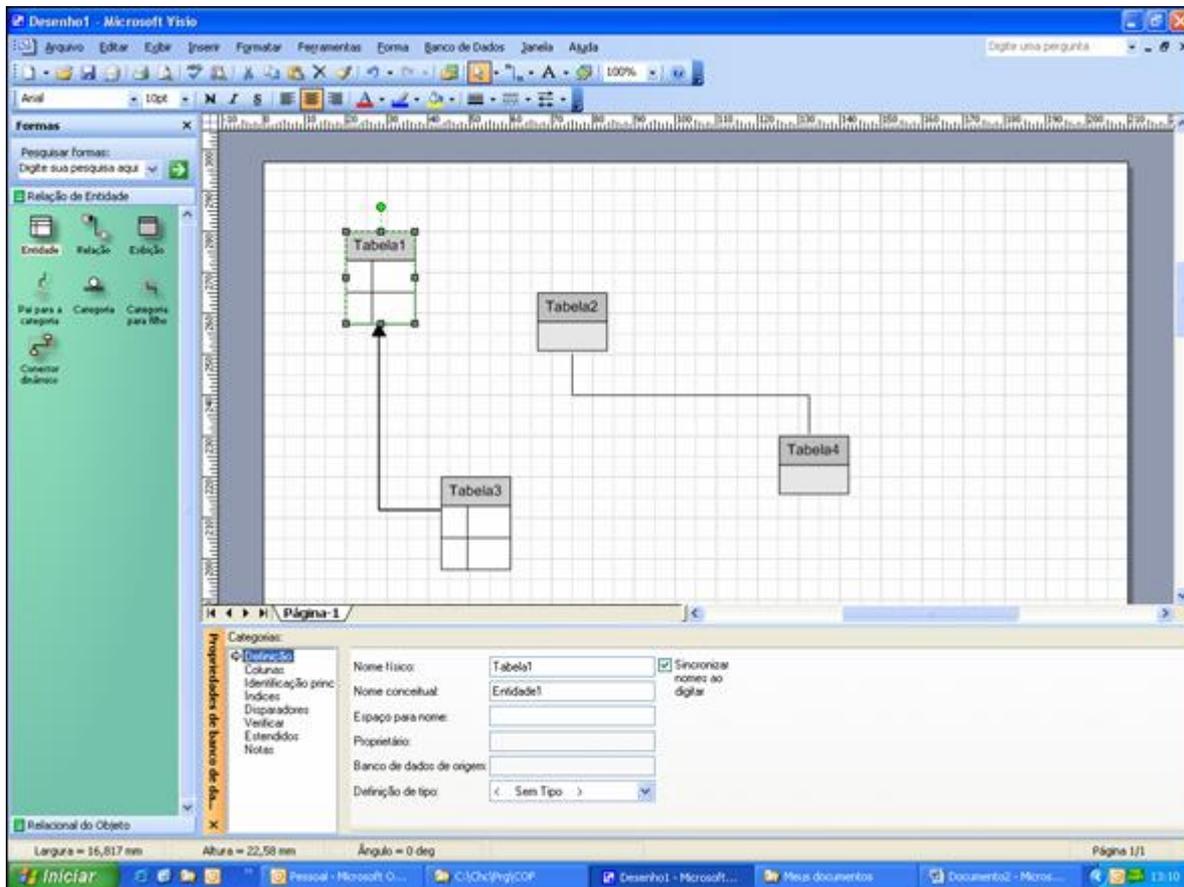


Figura 27. Diagrama no VISIO – diagramas de modelo conceitual e lógico na mesma tela.

O VISIO permite a paginação do modelo além de anotações para os objetos, o que pode ser encarado como uma espécie de dicionário de dados.

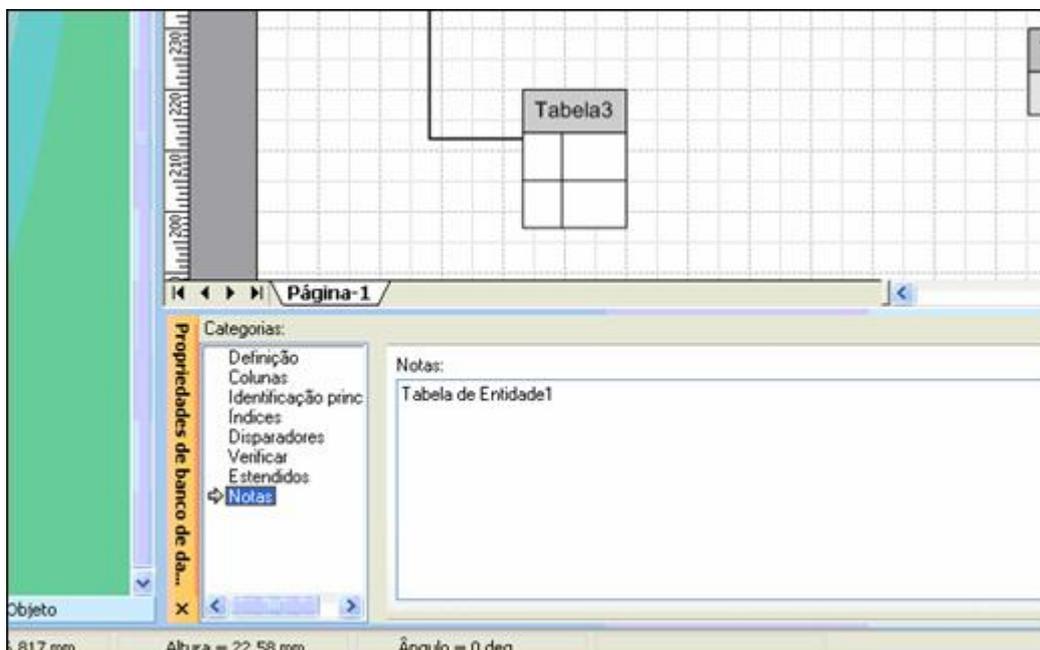


Figura 28. Diagrama no VISIO – Notas/Dicionário de dados.

#### 4.2.2. Conclusão da análise:

Nenhuma das ferramentas analisadas foi considerada ideal para o ensino das técnicas de modelagem de dados.

Tal afirmação se justifica pelo fato de que as ferramentas pesquisadas utilizam notações voltadas basicamente para a implementação do esquema gerado no SGBD sendo que algumas, se quer, implementam modelo conceitual.

Dentre os programas analisados, o PowerDesigner é a ferramenta cuja implementação mais se assemelha aos padrões de modelagem de dados defendidos por Peter P. Chen e Carlos A. Heuser no tocante à notação e construção dos modelos. Uma única falha identificada foi o fato de não permitir a interação do usuário no momento da conversão do modelo lógico para o modelo conceitual.

### 5. A APLICAÇÃO brModelo

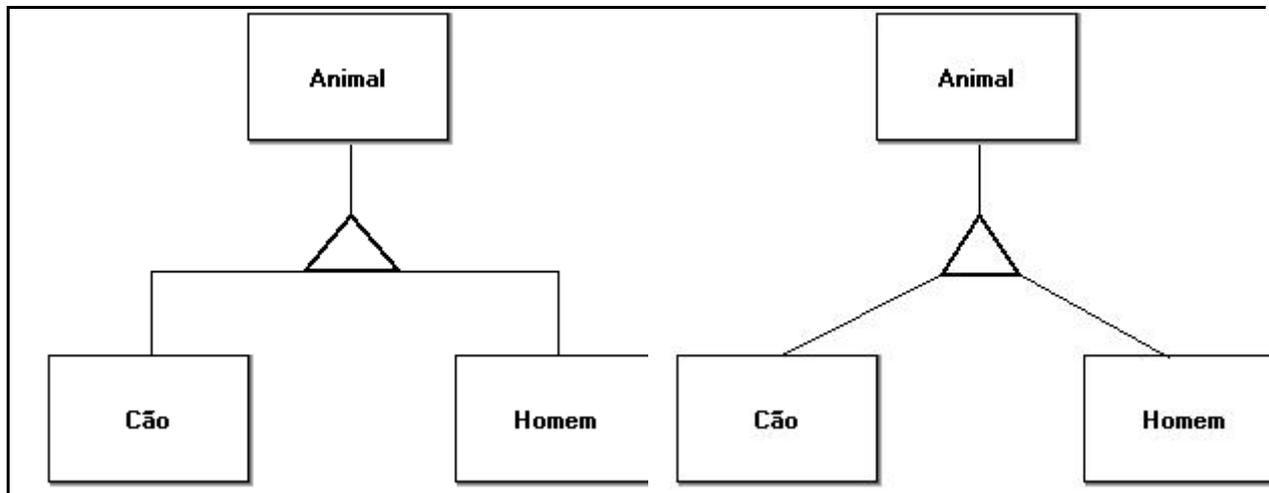
Após a análise das ferramentas mais comuns (capítulo 3) e das técnicas de

modelagem, decidi pela criação de uma ferramenta de modelagem utilizando a abordagem adotada pelo Dr. Heuser, tanto para a criação dos diagramas quanto para a conversão do modelo conceitual para o modelo lógico.

Decidi pela notação do Prof. Heuser e não a notação original do Dr. Peter P. Chen, por entender com significantes as contribuições dadas pelo primeiro ao trabalho do segundo, principalmente em se tratando do tratamento dado as cardinalidades das entidades e dos atributos.

**A idéia principal por trás da aplicação é ser uma ferramenta voltada para o ensino das técnicas de modelagem de dados.**

Para facilitar o trabalho de programação, algumas mudanças sutis foram inseridas na notação original do Dr. Heuser.



**Figura 29.** A esquerda implementação do modelo do Dr. Heuser no brModelo. À direita notação original.

A ferramenta ainda é um beta do que será defendido, se aceita no congresso brasileiro para aplicações demos em bancos de dados (outubro de 2005).

### **5.1. brModelo – Diferencial na construção do modelo conceitual.**

O modelo conceitual é o foco desta aplicação ao contrário das principais ferramentas disponíveis no mercado.

O brModelo está fortemente acoplado aos conceitos de construção de modelos/esquemas adotados pelo Dr. Carlos A. Heuser.

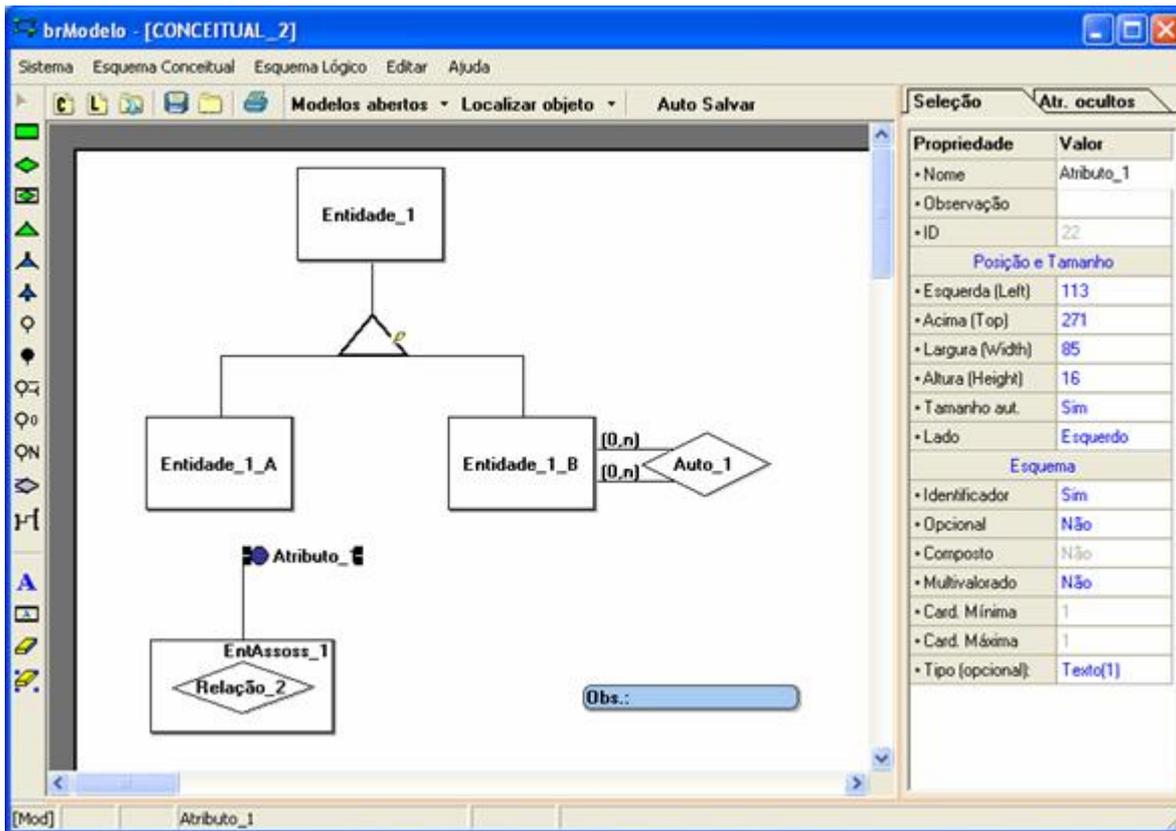


Figura 30. brModelo – Implementa os diagramas com sutis diferenças se comparada a notação original.

Suas vantagens em relação às ferramentas avaliadas, em síntese, são:

1. Permitir alterações estruturais no modelo diante de novas decisões do analista (figura abaixo)
  - a. Conversão de atributo em entidade.
  - b. Conversão de relacionamento em entidade associativa
  - c. Conversão de especialização de restrita para opcional ou vice-versa

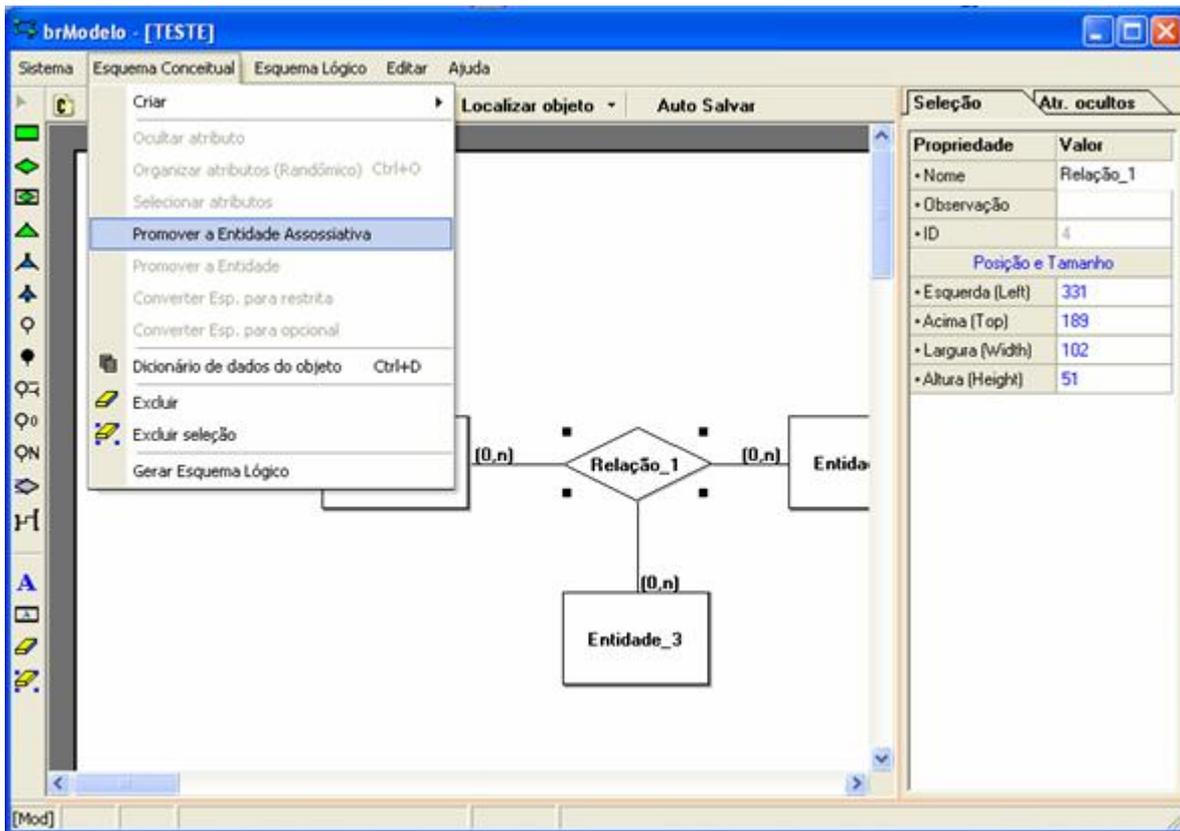


Figura 31. brModelo – Implementa os diagramas com sutis diferenças se comparada a notação original.

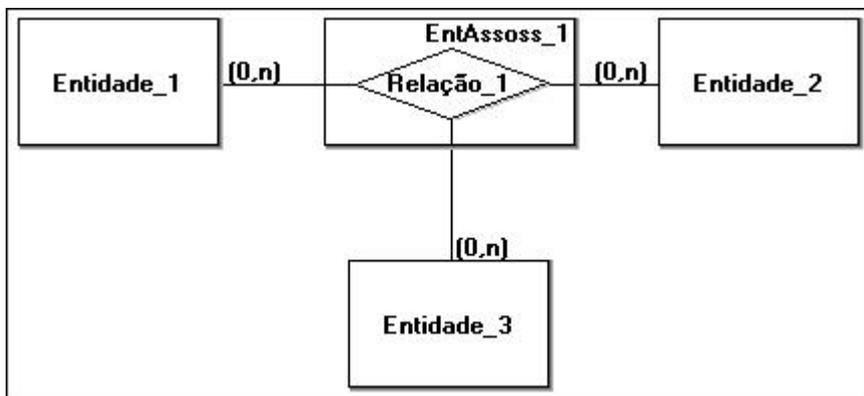


Figura 32. brModelo – Resultado da conversão.

2. Atenção especial dispensada aos atributos e todas as suas especificações:
  - a. Atributo opcional
  - b. Atributo multivalorado
  - c. Atributo composto
  - d. Atributo identificador

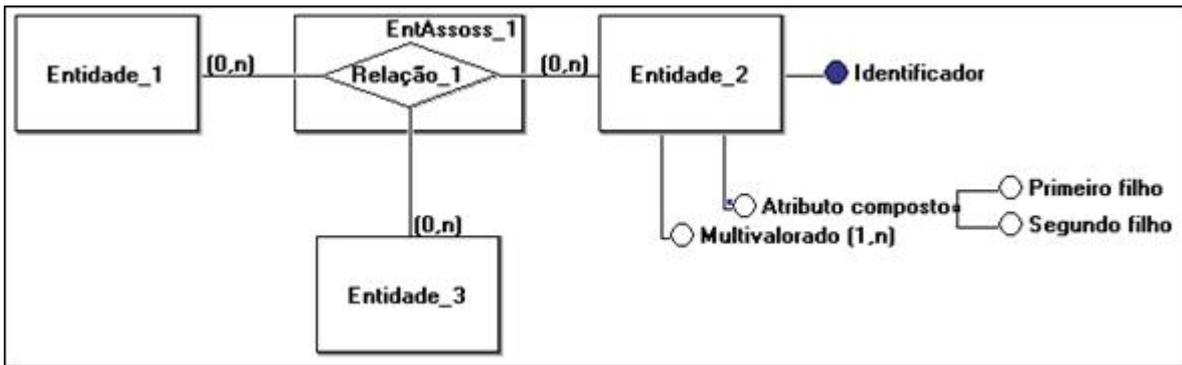


Figura 33. brModelo – Atribudos.

3. Possibilitar a “despoluição” do esquema ao ocultar atributos que não tenham significância no modelo conceitual, más, que poderão ser relevantes ao modelo lógico.
4. Dicionário de dados completos, específico para cada objeto do esquema e com capacidade de reunião em um único documento.

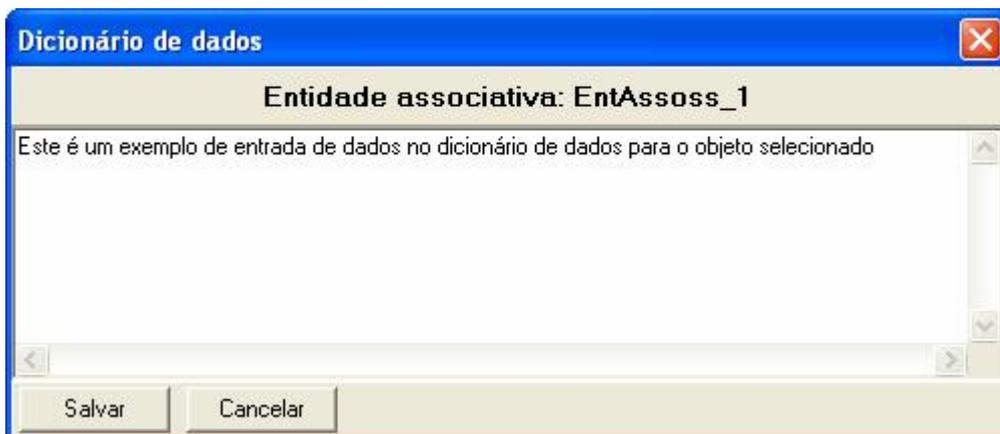


Figura 34. brModelo – Dicionário de dados.



Figura 35. brModelo – Dicionário de dados – Reunião dos dicionários em um documento único.

## 5.2. brModelo – Conversão para o modelo lógico.

Neste quesito, o diferencial é a interação com o usuário no momento da conversão (conceitual para lógico).

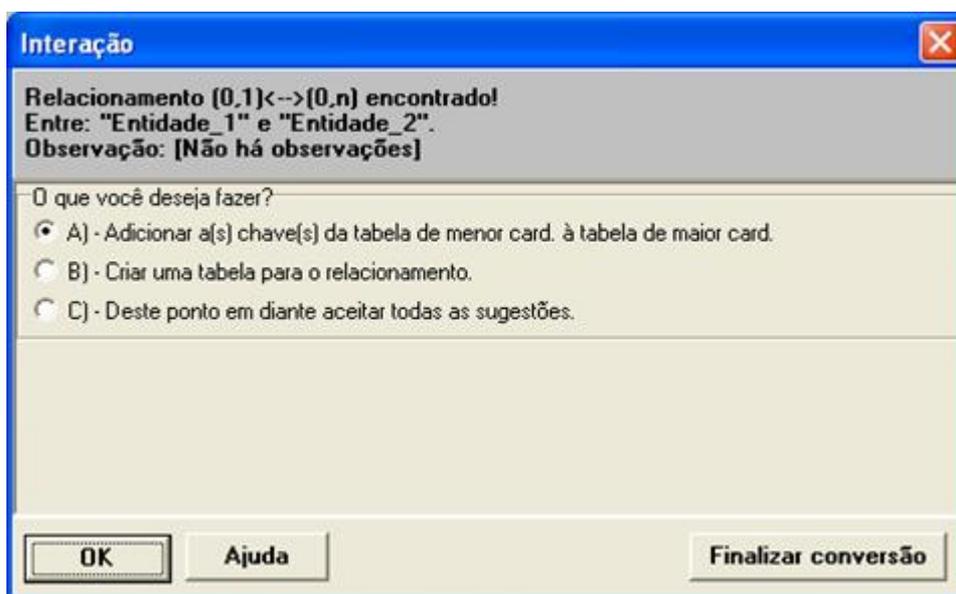


Figura 36. brModelo – Interação com o usuário no processo de conversão.

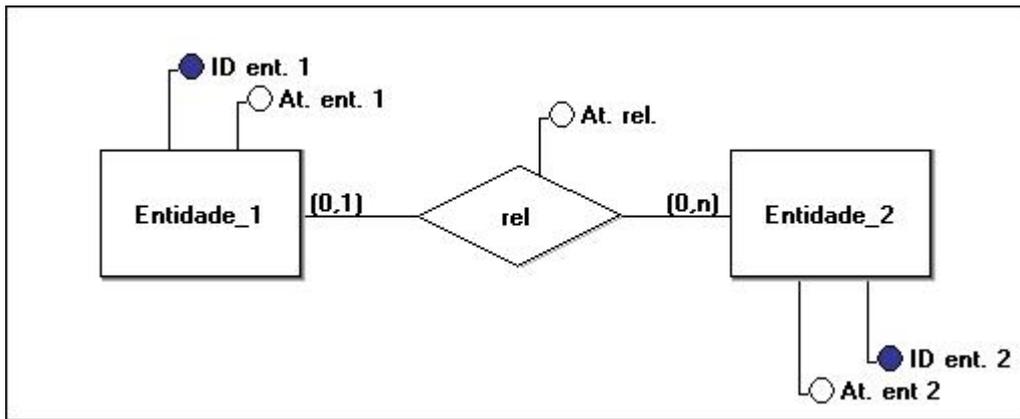


Figura 37. brModelo – processo de conversão (A).

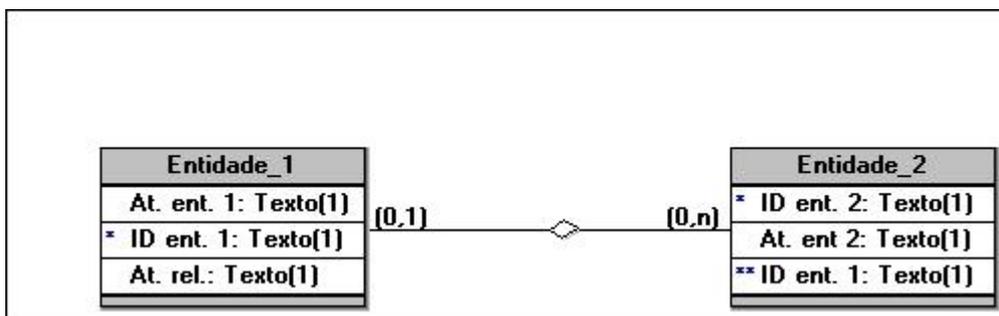


Figura 38. brModelo – processo de conversão (B1).

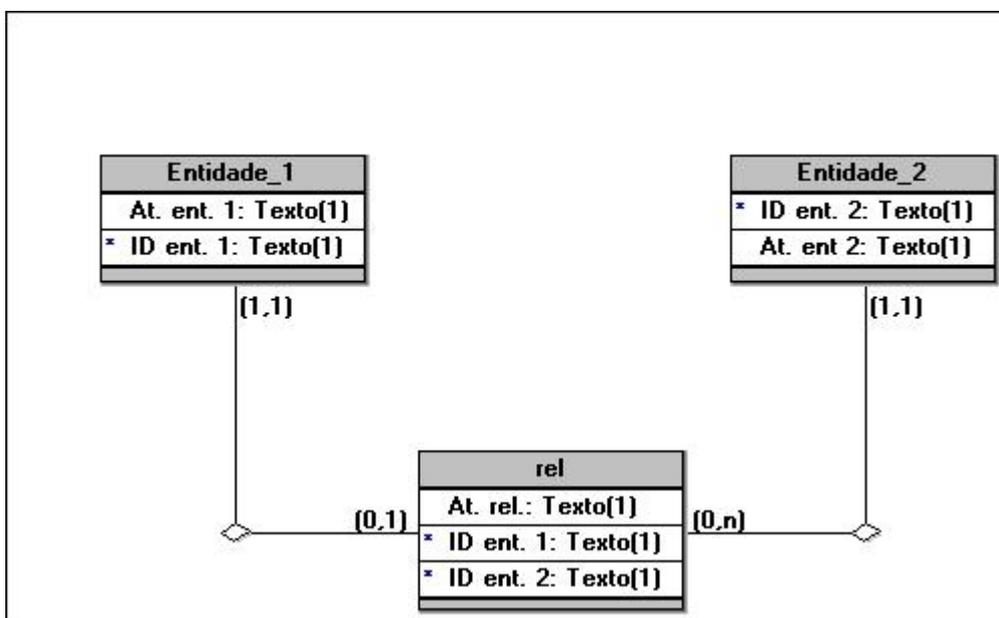


Figura 39. brModelo – processo de conversão (B2).

As figuras anteriores mostram o resultado da conversão de um modelo conceitual (fig. 37) com base nas opções do usuário (fig. 38: opção 1 e fig. 39 opção 2).

### 5.3. brModelo: Adicionais.

Mesmo na edição do modelo lógico, o brModelo continua apresentando algumas das características derivadas da notação adotada pelo Prof. Heuser. Um exemplo é a maneira como estão representadas as cardinalidades dos relacionamentos.

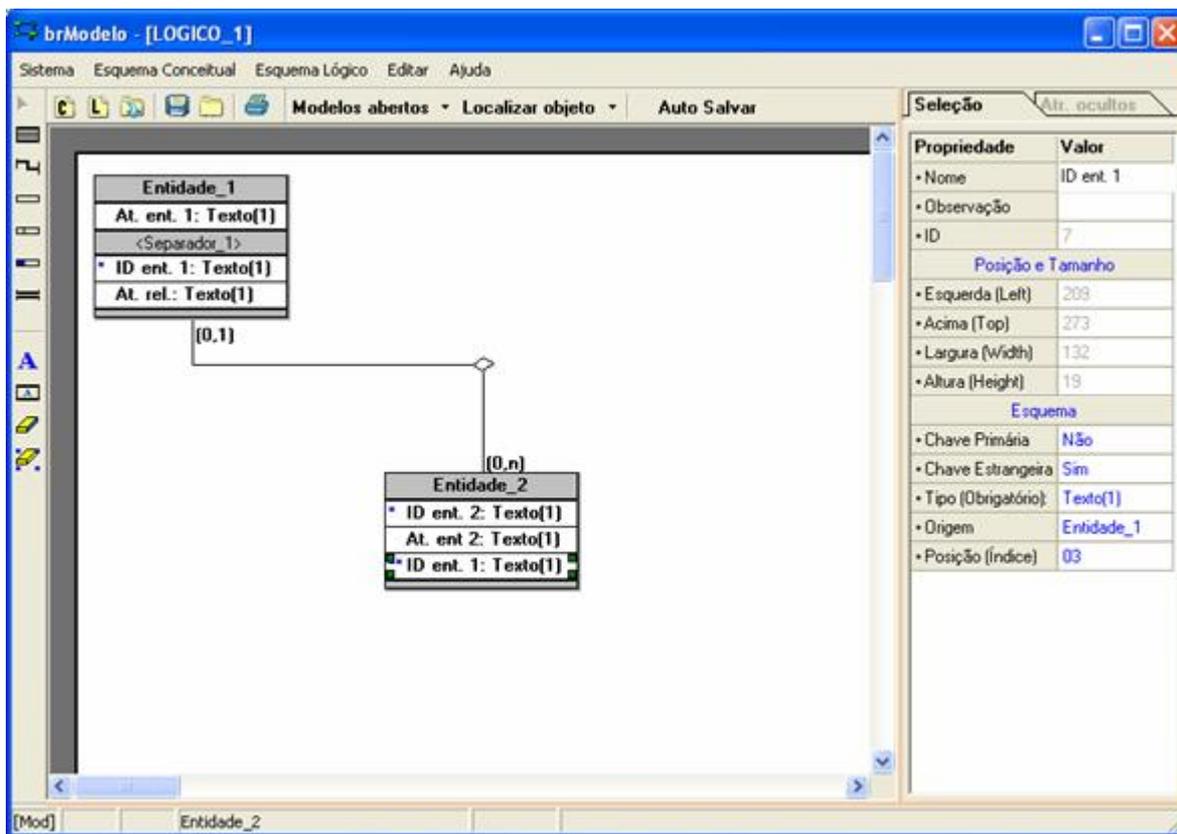


Figura 40. brModelo – modelo lógico com cardinalidade.

Uma outra importante funcionalidade é a capacidade de exibir seus modelos lógicos (salvos em XML) na mesma notação adotada pelo professor Heuser através do uso de XSLT.

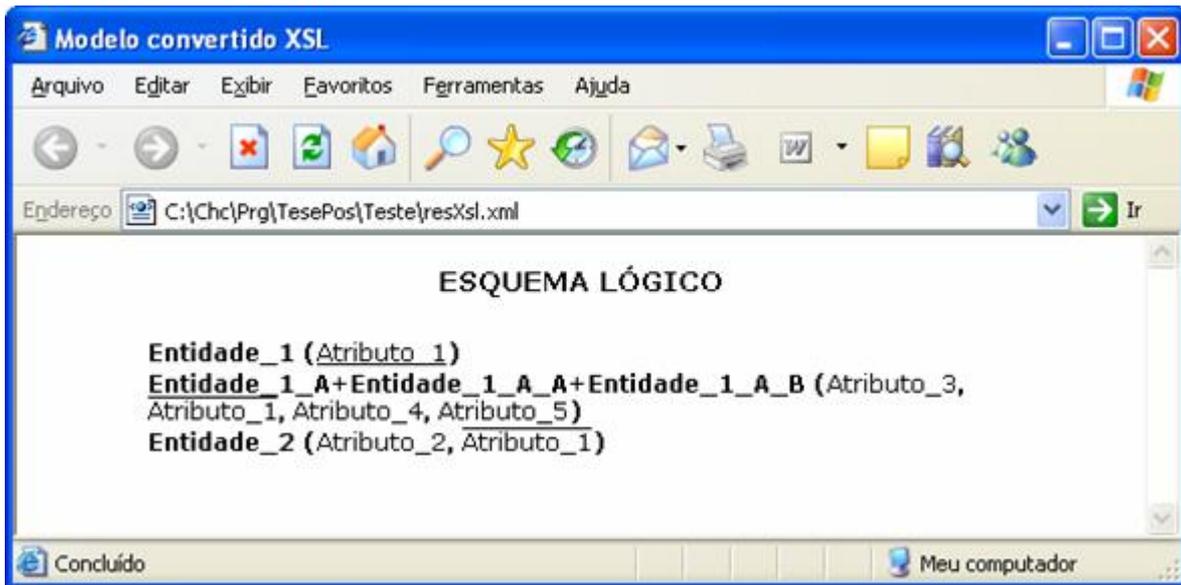


Figura 41. Internet Explorer: brModelo – modelo lógico conforme Dr. Heuser.

#### 5.4. brModelo: Para o futuro

1. Geração de código (modelo físico) em SQL ANSI.
2. Implementar regras de integridade.

**Mais detalhes copie a aplicação no CD em anexo para um diretório onde a mesma tenha permissão de escrita e a execute.**

## 6. CONCLUSÃO

A ferramenta desenvolvida está longe do nível de qualidade das aplicações de modelagem de dados do mercado, principalmente em se tratando da qualidade dos gráficos gerados por aquelas, contudo, não é conhecida nenhuma ferramenta com uma implementação tão fiel a um modelo acadêmico quanto o brModelo.

Seu desenvolvimento, ao longo dos últimos seis meses, revelou a grande dificuldade em converter os conceitos defendidos em livros e artigos científicos, mas revelou que é possível.

O sistema ainda não foi posto à prova. Após a defesa desta monografia, a aplicação entrará em fase de teste experimental com vista para o congresso de banco de dados que ocorrerá

de 03 à 10 de outubro do corrente ano.

## **7. REFERÊNCIAS BIBLIOGRÁFICAS**

### **7.1. Livros**

[1] ALCALDE Lancharro, Eduardo Garcia Lopez, Miguel Peñuelas Fernandez, Salvador. Informática Básica; São Paulo, Makron Books, 2004.

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

[3] HEUSER, Carlos Alberto. Projeto de Banco de Dados,  
Porto Alegre: Instituto de informática da UFRGS, Sagra Luzzato, 2001  
Série livros didáticos n.º 4.

### **7.2. Revistas**

[4] SQL MAGAZINE, Grajaú-RJ, SevMedia Group, 2004, Edição 13, 14, 15, 16,  
ISSN 1677918-5.

[5] SQL MAGAZINE, Grajaú-RJ, SevMedia Group, 2005, Edição 16, 22, ISSN  
1677918-5.

[6] INFO EXAME, Edição 230, Ano XX

### **7.3. Artigos e textos da internet**

[7] Álvares, Reinaldo Viana. Tecnologias de Banco de Dados e Modelagem de Dados Parte 1, Sql Magazine, Grajaú-RJ, Edição 12, pág. 31.

Álvares, Reinaldo Viana. Tecnologias de Banco de Dados e Modelagem de Dados Parte 2, Sql Magazine, Grajaú-RJ, Edição 12, pág. 37.

[8] Carvalho, Eric Barbosa de. Método Computacional, Dedicado à Modelagem e Tratamento de Informações em Enfermagem, Eric Barbosa de Carvalho, Rosane Beatriz Oliveira Severo, Mônica Ribeiro Ventura, Nilsa Sumie Yamashita Wadt, Josefa Olivinha Souza Oliveira, Centro Universitário Nove de Julho (UNINOVE), Brasil.

[9] Chen, Peter P. Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned, Computer Science Department Louisiana State University Baton Rouge, LA 70803, USA (\*)

[10] Chen, Peter P. English Sentence Structure and Entity-Relationship Diagrams, Elsevier Science Publishing Co., Inc. 52 Vanderbilt Ave, New York, NY 10017 (\*).

[11] Chen, Peter P. Future Directions of Conceptual Modeling, Bernhard Thalheim and Leah Y. Wong,, ,Computer Science Department Louisiana State University, Baton Rouge, LA 70803, USA (\*),

[12] Chen, Peter P. The entity-relationship model – A basis for the enterprise view of data, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1980 (\*).

[13] Chen, Peter P. The entity-relationship model – Toward a Unified View of Data, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1976 (\*).

\* Fonte: Research Papers On line : <http://bit.csc.lsu.edu/~chen/pdf/>

[14] Ding, Guoli. Generating r-regular graphs, Louisiana State University, Baton Rouge, USA, 2002.

**Texto na internet:**

[15] Heemann, Vivian: Avaliação de base de dados, dentro de um enfoque ergonômico.

**Texto na internet retirado do livro:**

[16] Database Management Systems 3ed, R. Ramakrishnan and J. Gehrke, McGraw-Hill, 2003.

---

[1] Chen, Peter P., Future Directions of Conceptual Modeling, Bernhard Thalheim, and Leah Y. Wong

[2] Heuser (2001) [3]

[3] Heuser, Dr. Carlos A., Projeto de Banco de Dados, 4.ª edição Heuser, pág 07

[4] O termo foi criado na França, em 1962, e provém da contração das palavras: **Information automatique (Informação Automática)**, conforme Alcalde (2004)[1].

[5] e <sup>7</sup> Peter P. Chen [XXX]

[6] UNICAMP/IC/MO410/2003-4 - Slides do livro Database Management Systems 3ed, R. Ramakrishnan and J. Gehrke, McGraw-Hill, 2003.

[7] National Computer Conference, 1977. Peter P. Chen (1976)[11]

[8] Claudete Moscardini – Sql Magazine Ed. 12 Ano 1.

[9]

[10] Sql Magazine (2004)[4]