

# Banco de Dados Relacional: Modelagem Conceitual e Lógica de Banco de Dados Relacional

Gustavo dos Santos Sobrinho<sup>1</sup>

<sup>1</sup>Bacharelado em Engenharia de Computação – Centro Universitário de Anápolis  
(UniEVANGÉLICA) – Anápolis – GO

<sup>1</sup>gustavo-ssobrinho@hotmail.com

**Resumo.** *Com o avanço tecnológico, o armazenamento e controle de dados e informações históricas é extremamente necessário. O presente relatório tem como objetivo, apresentar conceitos sobre a modelagem de dados relacional e a execução de um projeto em um ambiente de estudo de caso: a Fábrica de Tecnologias Turing (FTT), que se apresenta como um instrumento para a prática inovadora no processo de aprendizagem na Engenharia de Software, bem como espaço propício para a geração de inovação tecnológica. Neste contexto, o estudo foi realizado utilizando de referências bibliográficas, orientação e prática em projeto real desenvolvido na FTT. Este trabalho destina-se ao público iniciante no processo de modelagem de dados relacionais e naqueles interessados em conhecer práticas inovadoras no processo de aprendizagem em Engenharia de Software.*

**Palavras-chave:** *Modelagem de Dados; Banco de Dados Relacional; Prática Inovadora; Inovação Tecnológica.*

## 1. Introdução

Este documento tem como foco um público iniciante na modelagem do banco de dados, bem como pessoas que se interessem por práticas inovadoras no processo de aprendizagem em Engenharia de Software. Será tratado sobre 2 (dois) tipos de modelagem de dados relacional, sendo eles: modelagem conceitual e a modelagem lógica. Serão abordadas, ainda, ferramentas de gerenciamento de banco de dados e de modelagem, de modo que ajude no trabalho com a mesma, compreensão sobre o tema e o controle para futuros analistas de banco de dados.

Conforme RICARDO (2012), os erros comuns em um projeto de banco de dados são: mal planejamento, falta de padrões de nomenclatura, falta de documentação, dentre outros. Com essas falhas e faltas, futuras manutenções ou até mesmo melhorias ficam difíceis, pois, o analista de banco de dados possivelmente não compreenderá como e o que está acontecendo na situação atual.

A Fábrica de Tecnologias Turing (FTT) gerou novas oportunidades para alunos dos cursos de Engenharia de Computação e Engenharia de Software do Centro Universitário de Anápolis (UniEVANGÉLICA). Ela se trata de um ambiente impulsionador de inovação tecnológica e de prática inovadora no ensino da computação, no qual acadêmicos de todos os períodos dos cursos de Computação, podem ter contato com o meio profissional de sua área, compondo equipes de desenvolvimento de projetos reais, compostas por professores orientadores, líder técnico, acadêmicos, empresas e clientes. Neste contexto, ocorre de forma exitosa o desenvolvimento de competências necessárias aos profissionais do século XXI, que vão além da formação tecnológica. Para a realização do estudo de caso apresentado neste relato, acadêmicos do primeiro e segundo períodos do curso de Engenharia de Software ingressaram no desenvolvimento do projeto SEINTEGRA. O SEINTEGRA trata-se de uma proposta de desenvolvimento de um sistema integrador, que abarcará as ideias de inovação tecnológica relacionadas aos processos didático, pedagógico e gerencial dos cursos da UniEVANGÉLICA.

De acordo com MIRANDA (2016), os SGBDs (Sistemas Gerenciadores de Banco de Dados) mais utilizados no mercado, utilizam o modelo relacional, que trabalham com princípios de entidades e relacionamentos que receberá enfoque no presente relatório.

Partindo-se desta breve introdução ao desenvolvimento, segue-se com a apresentação de conceitos para a modelagem de um banco de dados relacional, seja de um novo projeto ou para a evolução de um sistema em andamento, e a demonstração da experiência adquirida no projeto SEINTEGRA.

## 2. Desenvolvimento

### 2.1. Compartilhamento de Dados

Antes de começar a tratar a respeito da modelagem, é necessário entender alguns conceitos iniciais. O compartilhamento de dados pode ocorrer em sistemas simples, como o controle de mercadorias de uma mercearia do seu bairro até mesmo em uma empresa de tecnologias multinacional. Porém, esse compartilhamento de dados influencia diretamente no funcionamento de sua empresa, estabelecimento, dentre outros. Pode também, influenciar na fluidez do sistema como também no mal funcionamento do mesmo.

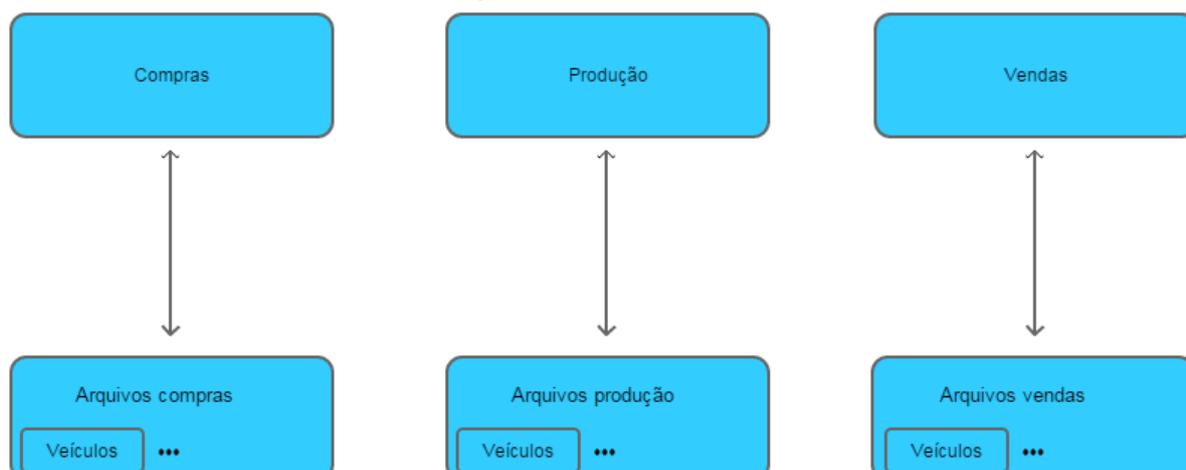
Formalmente, o compartilhamento de dados é o uso da mesma informação por diferentes usuários, sem influenciar nas ações de cada um deles (MACHADO, 2014)

Para ficar mais claro esse compartilhamento, vamos considerar uma indústria automobilística hipotética, na qual serão executadas três funções: venda, compra e produção.

- Venda: Esta função diz respeito as atividades da indústria relativa ao contato com as distribuidoras de automóveis.
- Produção: Esta função diz respeito as atividades relacionadas a montagem.
- Compra: Esta função diz respeito as atividades relacionadas as compras de peças.

No exemplo acima os dados de um veículo são usados em varias funções. Caso uma das atividades informadas acima sejam armazenadas separadamente gerando uma inconsistência dos dados (rapidamente falando, são informações diferentes em locais diferentes, gerando assim a inconsistência). Com isso, surge o problema da redundância de dados, sendo elas: a redundância controlada e a não controlada de dados.

Figura 1. Sistemas Isolados.

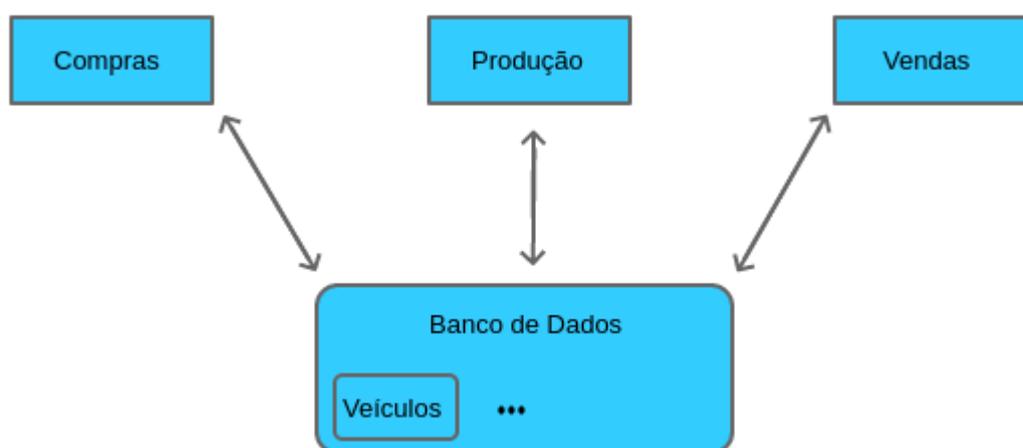


Fonte: Os autores.

- Redundância controlada: Acontece quando o sistema tem conhecimento da múltipla representação da informação e garante a sincronia entre elas.
- Redundância não controlada: A redundância não controlada de dados acontece quando a responsabilidade pela manutenção da sincronia entre as diversas representações de uma informação está com o usuário e não com o software (HEUSER 2009).

Uma possível solução para evitar esse redundância não controlada é o compartilhamento de dados. Nesta forma de processamento, cada informação é armazenada uma única vez, sendo acessada pelos diferentes sistemas que a necessitam.

Figura 2. Sistema compartilhado.



Fonte: Os autores.

## 2.2. Entidade, Atributo e Relacionamento

Outros conceitos principais para a modelagem do banco de dados são os conceitos de entidade, atributos e relacionamento. Os SGBDs (Sistemas Gerenciadores de Banco de Dados) mais utilizados no mercado são chamados de SGBDs Relacionais onde trabalham com estes princípios.

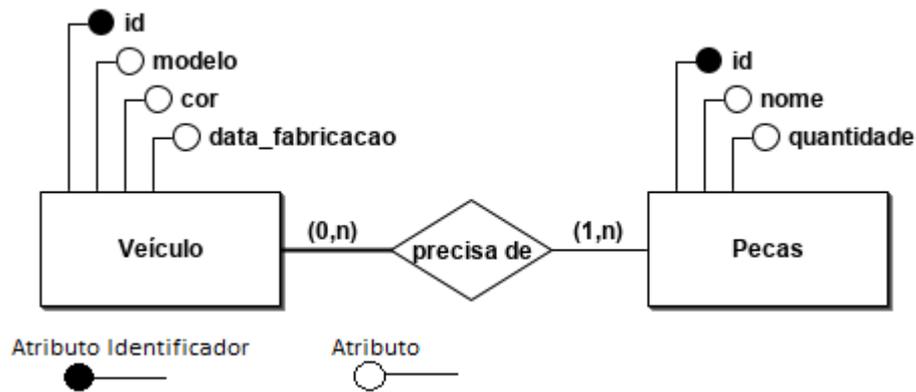
As entidades são conjuntos de objetos da realidade porém modelados na qual se deseja manter as informações no banco de dados. Relacionamentos são conjuntos de interações entre as entidades. Essas relações geram uma cardinalidade mínima e máxima entre a entidade e o relacionamento. Essa cardinalidade pode ser de três tipos: 0, 1 e n.

- 0 = Em uma definição informal, significa que uma entidade pode existir sem o relacionamento com a próxima, ou seja, opcional.
- 1 = Pode significar cardinalidade mínima e máxima, podendo ser interpretada como necessário no mínimo uma interação e quando utilizada como máxima, no máximo uma interação.
- n = Significa que se tem n (quantidade indefinida) interações entre elas.

As cardinalidades utilizadas na modelagem podem ser de quatro tipos: (1,1), (0,1), (1,n) e (0,n).

Já os atributos podem ser considerados como "características" das entidades gerando também, atributos identificadores, que quando implantado no modelo físico, facilitará a busca de valores através do identificador. A Figura 3 contém exemplos de cardinalidades e mostra, ainda, como os atributos podem ser graficamente representados.

Figura 3. Modelagem com atributos, relacionamento, entidades e cardinalidade.



Fonte: Os autores.

Na imagem acima a entidade Veículo está relacionada com a entidade Peças através do relacionamento chamado "precisa de". A cardinalidade utilizada significa que:

- Quando se considera o relacionamento entre Veículo e Peças, a cardinalidade que representa esta leitura é a cardinalidade (1,n), pois, o veículo, pra começar a ser produzido, precisa de no mínimo uma peça (vamos considerar a "carcaça" metálica do carro) logo após isso, ele pode receber n outras peças, desde, peças essenciais até peças de customização visual.
- Já o relacionamento entre Peças e Veículo, a cardinalidade que a representa é a cardinalidade (0,n), pois, a peça pode existir sem depender de um carro, pois ela pode fazer parte de um estoque, porém aquela peça também pode servir em outros veículos, como por exemplo, uma roda.

### 2.3. Modelagem Conceitual e Modelagem Lógica

Com tais conceitos apresentados, como isto será utilizado na modelagem?

Como já dito anteriormente, estes conceitos são essenciais para o entendimento e modelagem de um banco de dados, principalmente na modelagem conceitual pois a técnica mais difundida e menos complexa é a abordagem entidade-relacionamento. Essa abordagem é uma abordagem formal, precisa e não ambígua. Isto significa que diferentes leitores de um mesmo modelo ER (Entidade Relacionamento) devem sempre entender o mesmo, sendo subutilizado em certas organizações como ferramenta para apresentação informal de ideias. Mas, este tipo de modelagem tem um poder de expressão limitado, não representando todas as propriedades de um banco de dados.

Já o modelo lógico, leva em conta algumas limitações do modelo conceitual e disponibiliza alguns recursos para a "melhoria" do mesmo. O modelo lógico implementa, a nível de SGBD relacional, os dados apresentados abstratamente no modelo conceitual. O termo "implementação" significa que ocorre uma transformação de um modelo abstrato para um modelo mais detalhado para a implementação física.

#### 2.3.1. Regras do Modelo Lógico

Estas regras foram definidas tendo em vista dois objetivos do projeto de BD (Banco de Dados):

- Obter um banco que obtenha boa performance;

- Obter um banco de dados que simplifique o desenvolvimento e a manutenção de aplicações;

Estas regras buscam obter um banco que ocupe pouco espaço em disco. A fim de alcançar estes objetivos as regras foram baseadas nos seguintes princípios:

- Evitar junções - Ter os dados necessários a uma consulta em uma só linha. Um SGBD relacional normalmente armazena os dados de uma linha em uma tabela;
- Diminuir o número de chaves;
- Evitar campos opcionais - Campos opcionais são campos que podem assumir o valor vazio. Os SGBDs relacionais usualmente não desperdiçam espaço ao armazenar um campo vazio.

Para a implementação dos relacionamentos, existem três alternativas básicas de tradução, sendo elas:

- Tabela própria = O relacionamento é implementado através de uma única tabela contendo duas colunas: coluna correspondente aos identificadores das tabelas relacionadas e a coluna correspondente aos atributos (caso haja algum). Caso o relacionamento sendo traduzido possua a cardinalidade n:n (n para n), a chave primária desta tabela é formada pela coluna correspondente aos identificadores das entidades.;
- Adição de colunas = O relacionamento é a adição de outra coluna em uma das tabelas correspondentes a entidades participantes do relacionamento, gerando assim, chaves estrangeiras;
- Fusão de tabelas de entidades = Esta tradução se aplica somente a relacionamentos 1:1 (1 pra 1), consta em implementar em uma única entidade todos os atributos de ambas entidades relacionadas.

A alternativa específica para a implementação depende da cardinalidade mínima e máxima das entidades envolvidas.

#### **2.4. Informações Adjacentes**

Padrões devem ser adotados para deixar o modelo livre de redundâncias. Exemplos:

- Se possível, manter o nome dos atributos identificadores o mesmo;
- Caso o atributo tenha mais de uma palavra, separar por underline (data\_fabricacao);
- Evitar o uso de caracteres especiais (+, -, ., \*, &, %, \$, #, @, !, =, §, dentre outros);
- Evitar o uso de acentos;
- Caso a entidade tenha mais de uma palavra, utilize o Camelcase (Utilização de letra maiúscula no começo da segunda palavra).

A modelagem, dependendo do ambiente, deve refletir o aspecto temporal, como por exemplo, em uma faculdade onde necessitam manter o histórico de alunos, histórico de transações, histórico de funcionários, dentre outros. O modelo de um banco de dados que armazena somente informações atuais é diferente do modelo que armazena o histórico da informação.

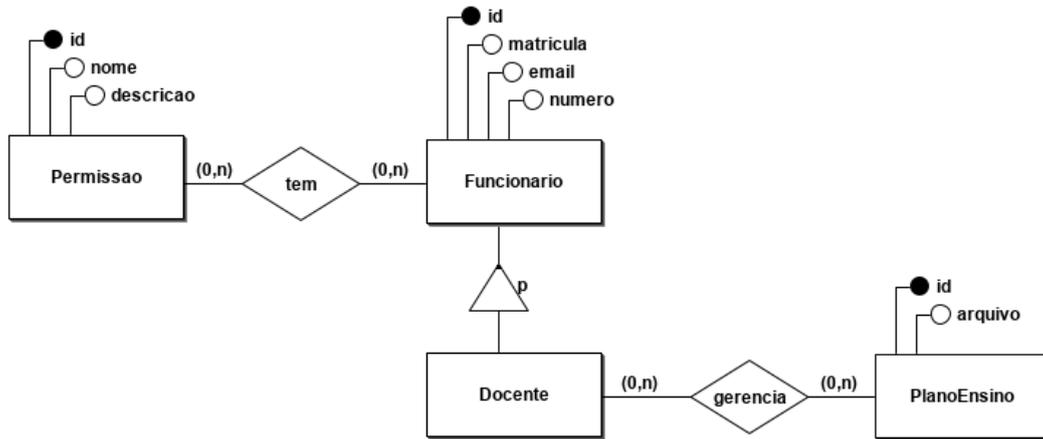
#### **2.5. Estudo de Caso: Modelagem de Dados no Seintegra**

Conforme apresentado na Introdução deste relato, o estudo de caso que irá demonstrar o desenvolvimento prático dos conceitos apresentados é o projeto SEINTEGRA, desenvolvido na FTT. O SEINTEGRA trata-se de um sistema integrador, que abará as ideias de inovação tecnológica relacionadas aos processos didático, pedagógico e gerencial dos cursos da

UniEVANGÉLICA. A princípio, o objetivo era desenvolver um sistema gestor de planos de ensino. Com as reuniões entre o cliente, a equipe do projeto e outros membros da FTT, essa ideia evoluiu para o desenvolvimento da proposta de um sistema integrador, capaz de abarcar sistematicamente subsistemas relacionados às demandas acima.

Com as ideias iniciais e o pouco conhecimento a respeito de modelagem, desenvolveu-se a modelo inicial, conforme é apresentada parcialmente na Figura 4.

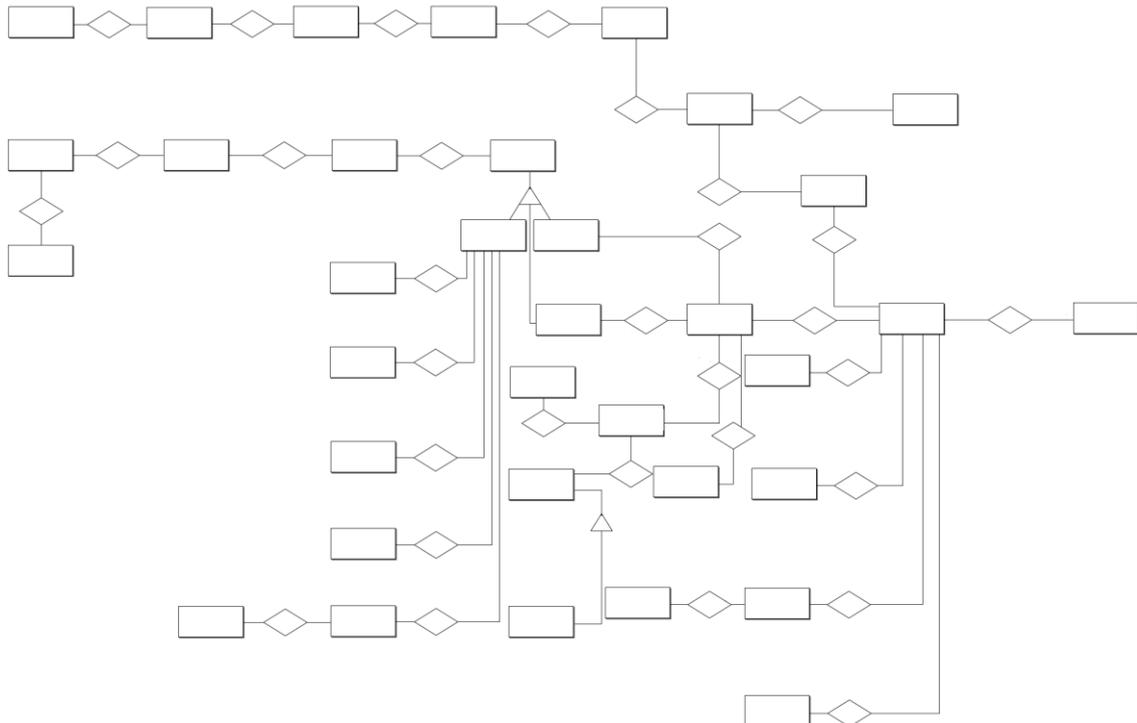
Figura 4. Modelagem inicial do gestor de plano de ensino.



Fonte: Os autores.

Atualmente, depois de desenvolver melhor os requisitos e amadurecer as ideias, o modelo desenvolvido possui mais de 60 entidades como mostra a Figura 5, podendo ser alterada ao longo do projeto, com o fim de atender às necessidades de atualização, bem como de maturidade a partir do domínio do contexto do sistema.

Figura 5. Modelagem SEINTEGRA (adaptado).



Fonte: Os autores.

Fora também realizada a Modelagem Lógica, conforme demonstrada parcialmente a seguir:

Permissoes (**id\_perm**, **nome**, **descricao**)

Cargo (**id\_cargo**, **nome**, **descricao**)

Tem (**id\_perm**, **id\_cargo**)

**id\_perm referencia** Permissoes

**id\_cargo referencia** Cargo

Cargo (**id\_cargo**, **nome**, **descricao**)

Funcionario (**id\_func**, **nome**, **matricula**, **foto**, **telefone**, **email**, **data\_admissao**)

Gera (**id\_fargo**, **id\_func**)

**id\_cargo referencia** Cargo

**id\_func referencia** Funcionario

### 3. Considerações Finais

Conforme ELMASRI (2010, 20), "Os bancos de dados e sua tecnologia tem um impacto importante sobre o uso crescente dos computadores". Ao primeiro contato com a modelagem de banco de dados, tudo se parece muito complexo. Dedicção aos estudos, orientação dos docentes e disposição de outras pessoas que possuem conhecimento nesta área, foi essencial para o entendimento e desenvolvimento da modelagem relacional de novos projetos na FTT.

Pode-se perceber que existem algumas convenções e padrões que demonstram boas práticas de projeto, cujo uso permitem tanto facilitar o entendimento entre a equipe e melhorar a produtividade, como aumentar as oportunidades de inserção dos envolvidos no mercado de trabalho. Contudo, cada empresa possui padrões específicos a serem seguidos e os padrões apresentados neste relato são baseados no modelo seguido pela FTT. Como lição aprendida, pode-se afirmar que uma boa prática a ser realizada ao ingressar em uma nova empresa é procurar se inteirar dos padrões adotados por ela.

Por fim, conclui-se que a Fábrica de Tecnologias Turing proporciona de forma inovadora, oportunidades de aprendizagem efetivo e contextualizado nas diversas áreas da Engenharia de Software, aos acadêmicos que desejam se inserir neste ambiente. Com a participação no projeto SEINTEGRA, percebe-se a evolução adquirida no que diz respeito à compreensão e aplicação de conhecimentos em Banco de Dados.

### Referências

HEUSER, C. A. *Projeto de Banco de Dados*. 6ª edição. Porto Alegre: Bookman, 2009.

MACHADO, F. N. R. *Projeto e implementação de banco de dados*. 3ª edição. São Paulo : Érica, 2014.

MIRANDA, W. OS 5 Bancos de dados mais utilizados do mercado. Disponível em: <<http://aprendaplsql.com/oracle/os-5-bancos-de-dados-mais-utilizados-do-mercado/>>. Acesso em: 14 nov. 2018.

NAVATHE, S. B.; ELMASRI, R. *Sistemas de Banco de Dados*. São Paulo: Pearson, 2011.

RICARDO. *Os dez erros comuns em projetos de banco de dados*. Disponível em: <<https://www.devmedia.com.br/os-dez-erros-comuns-em-projetos-de-banco-de-dados-revista-sql-magazine-101-parte-1/25049>>. Acesso em: 15 nov. 2018