

SISTEMAS DE INFORMAÇÃO GERENCIAIS

Aluno: Luiza Cavalcanti Marques

Orientador: Silvio Hamacher

Introdução

A modelagem e a utilização de bancos de dados em atividades gerenciais têm sofrido um aumento significativo nos últimos tempos.

Dentro do contexto de rápidas transformações, a concorrência cresce constantemente. Como consequência deste mercado global, cresce também a pressão dos consumidores em relação ao preço e à qualidade de produtos e serviços. A tecnologia da informação assume, portanto, um papel de suma importância, ao permitir, de forma rápida e simples, a extração, organização e análise de dados, impactando de forma decisiva ações tomadas pelas empresas.

Dada a maior complexidade das situações reais modeladas, o correto armazenamento de dados pode ser considerado imprescindível para o sucesso da gestão da situação modelada. Percebe-se, então, uma oportunidade de atender esta necessidade através da criação de interfaces gráficas e ferramentas computacionais para o auxílio de usuários na elaboração de bancos de dados. De forma prática e automática, conseguiu-se uma ferramenta geradora de banco de dados correspondentes a situação modelada pelo usuário e retratada por este em diagramas. Com este procedimento, minimizam-se erros e garante-se uma melhor integridade entre os elementos presentes no banco.

Neste projeto, o objetivo principal foi criar um mecanismo de criação de bancos de dados íntegros baseados em modelos fornecidos pelo usuário. Para isto, no entanto, foi necessário, antes, desenvolver um ambiente gráfico para a entrada do modelo também realizada pelo usuário.

Desta forma, foi elaborado, inicialmente, em projeto de Iniciação Científica, um estudo sobre modelagem de dados baseado na abordagem do Modelo Entidade-Relacionamento (MER), proposto por Peter P. Chen, em 1976. Após a realização de estudos preliminares, desenvolveu-se uma biblioteca gráfica de objetos para representar, de uma maneira mais fácil e simplificada, modelos de situações reais por meio de diagramas. Em seguida, pôde-se desenvolver um algoritmo de programação para a geração de um banco de dados correspondente ao diagrama Diagrama Entidade-Relacionamento (DER) criado pelo usuário, garantindo o cumprimento das regras associadas a tal diagrama, bem como a integridade do banco de dados.

A seguir, serão apresentados, neste relatório, uma visão geral dos sistemas de informações gerenciais, uma breve revisão bibliográfica a respeito do tema e a metodologia empregada no projeto, onde serão detalhados os passos seguidos em cada etapa. Será, também, apresentada a interface gráfica desenvolvida para o projeto, bem como um exemplo de estrutura de banco de dados criado.

Referencial Teórico

Nesta seção, será apresentada uma visão geral acerca dos conceitos fundamentais dos Sistemas de Informação e de projetos de bancos de dados.

Segundo Rezende (2005), “Todo sistema, usando ou não recursos de tecnologia da informação, que guarda dados e gera informação pode ser genericamente considerado Sistema de Informação.”. Este mesmo autor define dado como “elemento da informação, um conjunto de letras, números ou dígitos, que tomado isoladamente não transmite nenhum conhecimento, ou seja, não contém um significado claro”. A partir deste conceito, define informação como

“todo dado trabalhado, útil, tratado, com valor significativo atribuído ou agregado a ele e com sentido natural e lógico para quem usa a informação”.

O volume destas informações, no entanto, cresce cada vez mais, exigindo uma seleção e organização para seu melhor uso. Nas organizações e empresas, são observados diversos problemas relacionados ao manejo de tais informações, principalmente em níveis estratégicos e gerenciais. Assim, os Sistemas de Informação surgem como uma alternativa viável e necessária para auxiliar a tomada de decisões e a solução de problemas organizacionais. Estes sistemas também se caracterizam pelo grande volume de dados e informações, pela complexidade dos processos, pela grande quantidade de usuários envolvidos e por serem mutáveis e dinâmicos.

A utilização e a gestão da informação nos níveis estratégico, tático e operacional facilitam a tomada de decisões e soluções corporativas, além de satisfazerem a clientela. Um Sistema de Informação eficiente pode ter um grande impacto na estratégia corporativa e no sucesso da empresa. Este impacto pode beneficiar a empresa, os clientes ou usuários e qualquer indivíduo ou grupo que interagir com os Sistemas de Informação (OLIVEIRA,1998;STAIR,1998).

Os Sistemas de Informações incluem diversos tipos de sistemas, inclusive o Sistema de Informações Gerenciais (SIG), o qual será tratado, com ênfase, neste documento. Esta categoria tem por principal característica o processamento de dados de operações, transformando estes em informações agrupadas para o auxílio da gestão nas tomadas de decisão.

Segundo Heuser (1998), “sistemas de gerência de banco de dados (SGBD) surgiram no início da década de 70 com o objetivo de facilitar a programação de aplicações de banco de dados”. Com o aumento de investimentos em pesquisas na área, no entanto, desenvolveu-se um sistema de gerência de banco de dados relacional, amplamente difundido desde então. Para Heuser(1998), “o desenvolvimento de sistemas de informação ocorre hoje quase que exclusivamente sobre banco de dados, com uso de SGBD relacional”.

Este autor sugere a divisão de um projeto de banco de dados em três etapas: a modelagem conceitual, o projeto lógico e o projeto físico. Na fase de modelagem conceitual, constrói-se um modelo conceitual, na forma de um Diagrama Entidade-Relacionamento (DER) para capturar as necessidades da organização em termos de armazenamento de dados. Na fase seguinte, é desenvolvido o projeto lógico a partir de uma transformação do modelo conceitual obtido anteriormente. O modelo lógico tem por objetivo definir como o banco de dados será implementado. Na última fase, o projeto físico, são definidos os parâmetros físicos de acesso ao banco de dados.

Metodologia

Para a execução da modelagem de dados, segundo a lógica do Modelo Entidade-Relacionamento (MER) proposto por Peter P. Chen, foram necessários, inicialmente, o entendimento dos conceitos, a representação gráfica dos objetos e a verificação de fidelidade e coerência para, então, considerar o modelo válido.

A metodologia do projeto se baseou, então, primeiramente, em analisar as características e regras gerais do Modelo Entidade-Relacionamento (MER). Posteriormente, um ambiente de visualização gráfica foi criado, segundo este modelo, no Microsoft Office Visio, programa do pacote Office de diagramação e ferramentas de desenho de fácil utilização. Depois desta fase, foi elaborado um algoritmo de programação para estruturar as informações expostas no diagrama em um banco de dados íntegro.

Solução

Para a compreensão dos conceitos fundamentais do Modelo Entidade-Relacionamento, foi estudada a técnica de visualização gráfica, que, de forma simples, representa entidades, entidades fracas, relacionamentos, atributos e generalizações por retângulos, retângulos de linha dupla, losangos, elipses e triângulos, respectivamente. Esta técnica de representação gráfica associada ao Modelo Entidade-Relacionamento (MER) facilita a compreensão das situações reais modeladas pelos usuários e evita possíveis erros, além de ser mais intuitiva e simples em relação aos usuários. Esta técnica pode ser chamada de Diagrama de Entidade-Relacionamento (DER).

Dentro da fase de estudos preliminares, também foram pesquisadas as características inerentes a cada uma destas classificações. As entidades são definidas como objetos existentes no mundo real com identificação e um significado próprio. As entidades fracas, por não possuírem atributos suficientes para caracterizá-las, são, portanto, dependentes de uma entidade forte. Os relacionamentos representam a associação entre entidades e os atributos, as propriedades das entidades e/ou dos relacionamentos. A generalização mostra que entidades com atributos em comum podem ser generalizadas por uma superentidade. É importante ressaltar que todas as entidades se relacionam com as demais segundo um grau chamado de cardinalidade (também chamada de restrição de mapeamento), representado por uma proporção que indica o número de elementos que se relacionam e são relacionados com uma dada entidade.

A partir desta análise, foi elaborada no programa Microsoft Office Visio uma biblioteca gráfica de objetos (estêncil). Esta biblioteca criada permite a representação de entidades, entidades fracas, relacionamentos, atributos e generalizações, além de ligações entre estes, em uma folha de trabalho do próprio programa Microsoft Office Visio. A Figura 1 representa o estêncil criado.

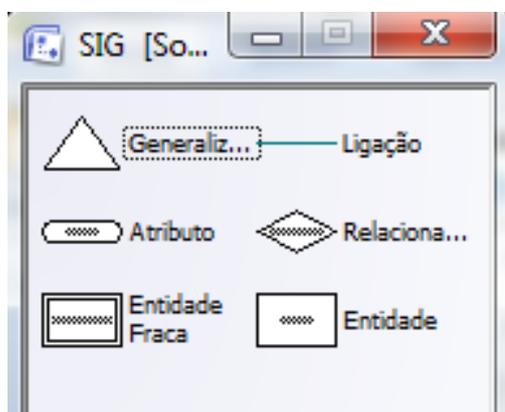


Figura 1 - Estêncil

Para a criação de um diagrama Entidade-Relacionamento, é necessário apenas mover os objetos desejados da biblioteca (estêncil) para a folha de trabalho do Microsoft Office Visio, preencher as informações pertinentes no campo de texto de cada um dos itens, ao dar um duplo clique, e fazer as devidas ligações entre objetos. Um possível exemplo pode ser conferido na Figura 2.

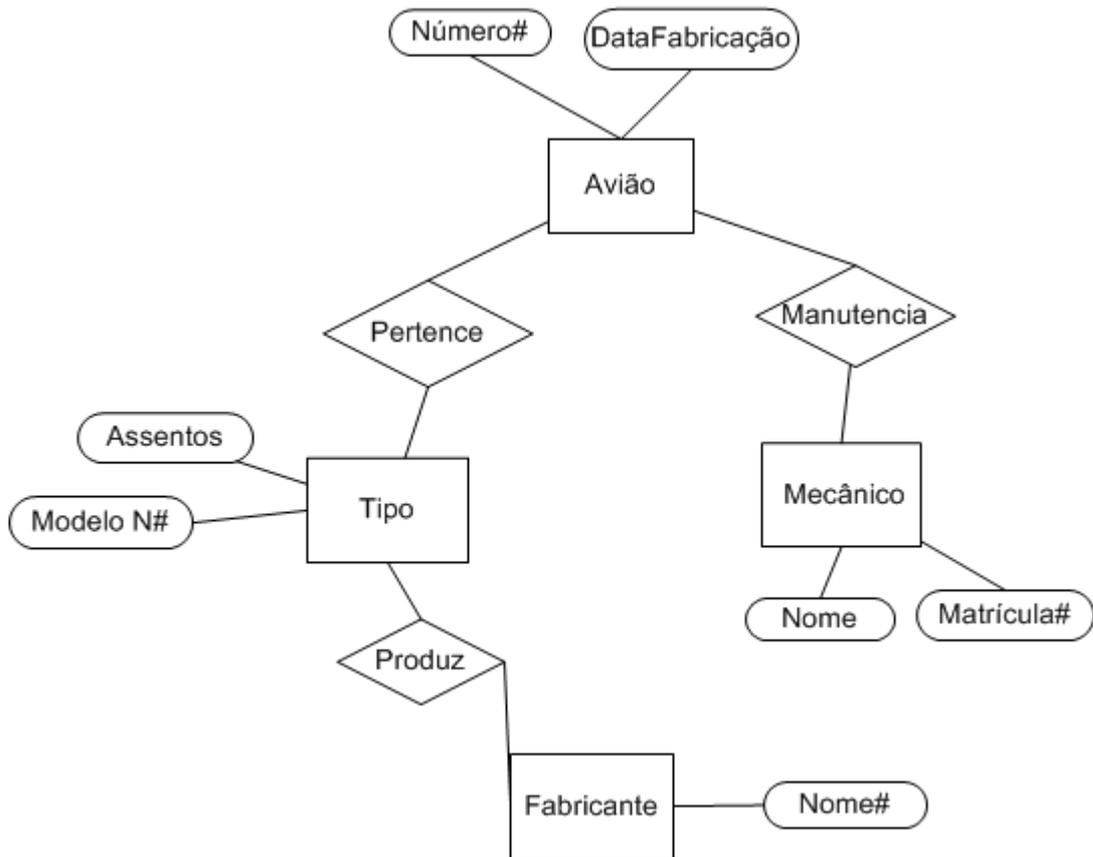


Figura 2 - Exemplo de Diagrama Entidade-Relacionamento (DER)

Além da modelagem esquemática da situação via Diagrama Entidade-Relacionamento (DER), foi também desenvolvido, em um segundo momento, um mecanismo de atribuição de cardinalidades às ligações entre entidades e relacionamentos. Através de programação desenvolvida em Visual Basic no próprio Microsoft Office Visio, foi criada uma caixa de opções, que é acionada com um duplo clique na ligação onde se deseja atribuir uma cardinalidade. Com isto, é possível escolher um valor dentre as opções oferecidas (0:1 , 0:N , 1:1 , 1:N e N:N) ou inserir o valor de cardinalidade desejado para a ligação em questão. Este mecanismo de inserção de cardinalidades às ligações é mostrado nas Figuras 3 e 4.

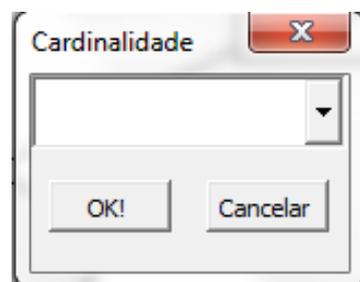


Figura 3 - Atribuição de cardinalidade a ligação

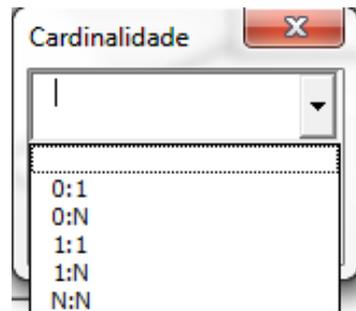


Figura 4 - Atribuição de cardinalidade a ligação

De posse de um modelo conceitual, representado graficamente, tornou-se viável a transferência de informações contidas no diagrama a um banco de dados. Este, por sua vez, deve retratar fielmente as necessidades do usuário, possuindo, portanto, integridade.

A geração de bancos de dados, no entanto, deve respeitar certas regras, segundo o Modelo Relacional, proposto por Codd (1970). Cada banco de dados deve expressar suas relações por meio de tabelas intituladas pelos nomes de cada entidade, salvo casos especiais, os quais possuem outras regras. Cada uma destas tabelas, por sua vez, deve possuir campos nomeados pelos atributos correspondentes à entidade em questão. Além de tal fato, toda tabela deve possuir pelo menos um campo definido como chave, isto é, um campo de dados com informações únicas, sem repetição, que definem a tabela. Estas podem ser divididas em primárias, quando representam unicamente a tabela em questão, ou estrangeiras, quando representam informações únicas de outra tabela relacionada à estudada. Tais campos são fundamentais, pois, dada a unicidade de informações, torna-se mais prática e rápida a consulta de tabelas e a busca de informações nestas. Deve-se, no entanto, garantir que chaves primárias e estrangeiras relacionadas em diferentes tabelas contenham as mesmas informações, conferindo, assim, integridade ao banco.

Foi elaborado um algoritmo de programação na linguagem Visual Basic dentro do próprio programa utilizado, Microsoft Office Visio, com o objetivo de transmitir as informações contidas no Diagrama Entidade-Relacionamento criado pelo usuário a um banco de dados associado no Microsoft Office Access. Este algoritmo, bem como sua lógica de programação, serão detalhados a seguir.

Em um primeiro momento, este algoritmo percorre todas as ligações presentes no diagrama fornecido, de acordo com a biblioteca de objetos criada. Para cada ligação percorrida, este armazena as informações sobre os objetos conectados por esta ligação em uma tabela, de uso interno do programa. Nesta tabela, encontram-se informações dos objetos ligados, que possuem extrema importância para a busca de informações em futuras consultas, como nome, código de identificação, classificação do tipo de objeto, entre outras. Tais consultas são desenvolvidas dentro do próprio código na linguagem de banco de dados SQL.

Com esta tabela preenchida a partir do diagrama, as demais tabelas são criadas via código, de acordo com as regras presentes no Modelo Entidade-Relacionamento (MER). Para isso, o código percorre a tabela de uso interno, linha a linha, tratando uma entidade de cada vez. Ao ler a primeira entidade, será criada uma tabela intitulada por seu nome, o qual está presente no campo de texto da entidade analisada. Após este procedimento, será realizada uma consulta à tabela geral do programa de forma a identificar os objetos ligados a tal entidade. De acordo com o tipo de objeto ligado, uma diferente metodologia é aplicada.

Caso o objeto ligado seja um atributo, a tabela criada receberá uma nova coluna de dados intitulada pelo campo de texto associado a tal atributo.

Caso o objeto ligado seja um atributo considerado chave, este seguirá o mesmo procedimento de um atributo, porém a coluna da tabela será classificada como chave.

Entretanto, caso o objeto ligado seja um relacionamento, o procedimento a ser seguido pelo código é outro. Neste caso, é realizada uma consulta para descobrir o(s) outro(s) item(ns) ligados ao relacionamento, bem como as cardinalidades das ligações entre estes. Contudo, é necessário apenas observar os máximos valores da proporção representada pela cardinalidade e não ela como um todo. Estes valores de máximo geram outra proporção, que pode assumir valores 1:1 , 1:N , N:1 e N:N, onde N pode representar qualquer número maior que um.

Assim, no caso em que esta relação entre máximos for 1:1, a entidade analisada deverá receber, como atributo considerado chave, o atributo chave da entidade ligada a esta.

Quando a relação é de 1:N, nenhuma medida é tomada, pois este mesmo relacionamento voltará a ser analisado na forma N:1, quando a entidade ligada estiver sendo estudada.

No caso da forma N:1, no entanto, o procedimento realizado é similar ao primeiro caso. É feita uma consulta para descobrir o atributo chave da entidade relacionada e este é também inserido na tabela como chave.

O último caso, quando a relação entre máximos é N:N, é um pouco mais complicado. Neste caso, as informações são armazenadas em uma tabela própria do relacionamento, intitulada pelos nomes das entidades relacionadas por ele. Esta tabela recebe, como atributos chaves, os atributos chaves das entidades ligadas ao relacionamento e, como atributos, os eventuais itens esquematizados pelo usuário no diagrama.

Ao percorrer todas as ligações de determinada entidade, criando sua tabela principal, preenchendo esta com atributos e tratando seus relacionamentos, o código passa então a analisar a próxima entidade armazenada na tabela de uso interno do programa.

Outros casos mais específicos também são tratados como o relacionamento de mesma entidade e o relacionamento ternário.

No primeiro, o procedimento segue, em princípio, a mesma lógica de relacionamentos entre diferentes entidades, porém, por relacionar uma entidade com si mesma, referencia os atributos relacionados acrescentando “Bis” para diferenciá-los dos demais. Já o relacionamento ternário apresenta as informações das três entidades relacionadas entre si, todas apresentando um grau máximo de cardinalidade maior que um, na forma de tabela também inserida no banco de dados.

É válido lembrar que todas as tabelas criadas via código são armazenadas em um banco de dados íntegro, conferindo qualidade às informações neste inseridas.

A seguir, na Figura 5, é representado um exemplo de Diagrama Entidade-Relacionamento com suas respectivas cardinalidades, modelando uma Companhia de Aviação.

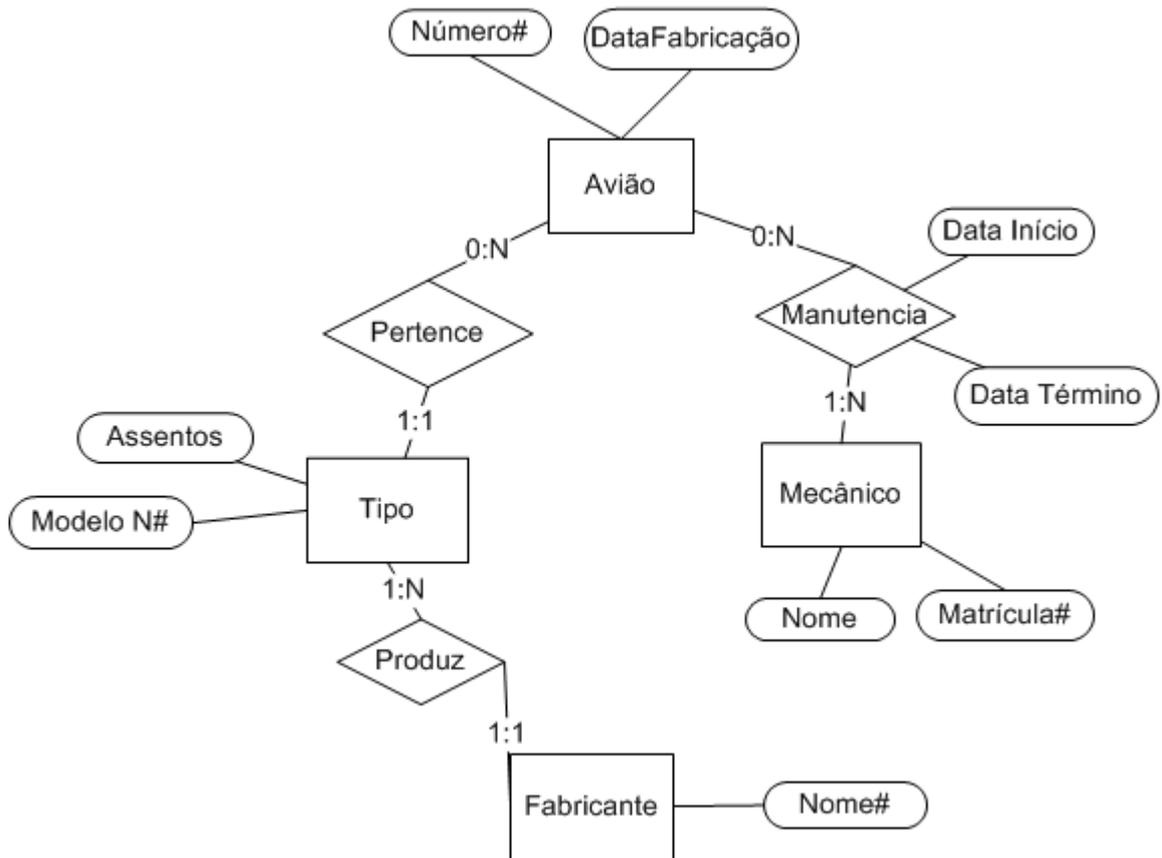


Figura 5 - Exemplo de DER (Companhia de Aviação)

Associado a este diagrama e com auxílio do algoritmo de programação, pode ser gerado, em Access, um banco de dados, como mostrado na Figura 6.

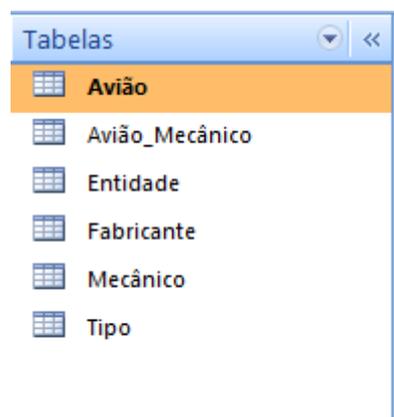


Figura 6 - Tabelas contidas no Banco de Dados

Estas tabelas criadas seguem as regras propostas por Codd (1970) e detalhadas ao longo deste documento. Exemplos das tabelas “Avião” e “Avião_Mecânico” podem ser conferidos nas Figuras 7 e 8.

Número	DataFabricação	Modelo N_Tipo	Adicionar Novo Campo
* (Novo)			

Figura 7 - Tabela "Avião"

A Figura 7 mostra a tabela “Avião”, que referencia a entidade Avião, possuindo como campos de atributos Número (classificado também como chave), DataFabricação e Modelo N_Tipo. Este último corresponde ao campo chave da entidade Tipo e deve ser anexado a esta tabela, já que o relacionamento entre Avião e Tipo ocorre numa proporção de máximos de N:1.

Data Início	Data Término	Número_Avião	Matrícula_Mecânico	Adicionar Novo Campo
		* (Novo)		

Figura 8 - Tabela "Avião_Relacionamento"

A Figura 8, no entanto, retrata uma tabela de relacionamento entre entidades chamada “Avião_Mecânico”. Este relacionamento do tipo N:N liga a entidade Avião a entidade Mecânico e possui, por isso, os campos de atributos Data Início, Data Término, Número_Avião (classificado como chave) e Matrícula_Mecânico (também classificado como chave). Estes últimos correspondem às chaves das entidades relacionadas, que devem ser adicionadas à tabela de relacionamento neste caso, como explicado anteriormente.

Conclusões

A criação de uma ferramenta de representação de diagramas, segundo o Modelo Entidade-Relacionamento (MER), permitiu um melhor entendimento de sistemas reais de forma mais abstrata e simplificada.

O desenvolvimento posterior de um algoritmo de programação, em Visual Basic, tornou possível a transferência de informações contidas no diagrama construído pelo usuário para um banco de dados associado a tal esquema. Este banco foi construído de forma a garantir a sua integridade e de acordo com as regras advindas do MER.

A ferramenta geradora de bancos de dados bem como o esquema gráfico desenvolvido podem ser utilizados de forma fácil pelo usuário. A geração de bancos via código, além de ser simples, reduz o tempo de desenvolvimento e os erros de construção dos mesmos, além de melhorar a segurança dos dados inseridos.

A ferramenta desenvolvida será utilizada a partir de 2011.2 no curso ENG 1518 – Sistemas de Informação Gerenciais, para cerca de 150 alunos por ano.

Referências

1 - COUGO, P. **Modelagem conceitual e projeto de banco de dados**. Ed. Campus. Rio de Janeiro. 1997.

2- CODD, E.F. **A Relational Model of Data for Large Shared Data Banks**. Communications of the ACM, Vol. 13, Nº. 6, p. 377-387, Jun. 1970.

3 – REZENDE, D.A. **Engenharia de software e sistemas de informação**. Ed. Brasport. 3ª edição. Rio de Janeiro. 2005.

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

5 - STAIR, R.A. **Princípios de Sistemas de Informação: Uma Abordagem Gerencial**. Ed. LTC - Livros Técnicos e Científicos. 2ª edição. Rio de Janeiro. 1998.