

Caetano Traina Júnior - Jan Frans Willem Slaets  
ICMSC - USP - São Carlos - IFQSC - USP - São Carlos  
Av. Dr. Carlos Botelho, 1465 - São Carlos - SP

## RESUMO

Apresenta-se uma Máquina de Base de Dados para suportar um modelo de dados denominado Modelo de Representação de Objetos, um modelo com alta capacidade de representação semântica, concebido para dar apoio a Sistemas de Gerenciamento de Bases de Dados a serem usados em Sistemas de Apoio a Projetos de Engenharia e em Ambientes de Apoio ao Desenvolvimento de Software. No seu desenvolvimento foi levada em consideração a possibilidade de utilizar hardware específico para suportar o SGBD nele apoiado. A Máquina de Bases de Dados descrita segue uma arquitetura MIMD, com processamento em memória dedicada, e indexação a nível de páginas. A comunicação com os programas de aplicação é feita diretamente entre cada estação de trabalho e a Máquina de Base de Dados.

## ABSTRACT

A Database Machine to support a novel Database Model is presented. That model has a great semantic representation capacity, and was conceived to support the development of Computer-Aided Engineering and Software Engineering Environments. The possibility of to utilize hardware specific to support the Database Management System was considered in its development. The resultant Database Machine has a MIMD architecture, with dedicated memory in each processor, a page based index scheme and supports communication directly with each Workstation.

## 1. - INTRODUÇÃO

A crescente necessidade de criação e desenvolvimento de Bases de Dados muito grandes tem originado a busca de máquinas que suportem especialmente bem as necessidades de manipulação de dados dos grandes Sistemas de Gerenciamento de Bases de Dados - SGBDs. Já desde a década passada, várias pesquisas e desenvolvimento tem buscado solucionar determinados problemas críticos nesse tipo de software através de soluções de hardware, tais como Discos Associativos [1] [2] [3] [4] [5], ordenadores [6] [7], Filtros de Seleção de Dados [8] [9], etc.

Desde o início dos anos 70, a pesquisa em Sistemas de Bases de Dados tem dedicado-se principalmente ao Modelo Relacional [10], e isso tem refletido-se sobre o desenvolvimento de Máquinas de Bases de Dados, fazendo com que a maioria delas sejam voltadas exclusivamente ao modelo relacional [11] [12] [13] [14] [15] [16]. Por outro lado, muito trabalho tem sido feito também sobre modelos de dados de alta capacidade de representação semântica [17] [18]

[19]. Esses modelos em geral são desenvolvidos a nível conceitual, e para a maioria deles não existe uma implementação realizada que os suporte. Essas duas áreas de pesquisa em geral são abordadas separadamente, razão pela qual não se encontram máquinas construídas para modelos semânticos.

Neste trabalho utiliza-se um modelo de dados, denominado Modelo de Representação de Objetos - MRO [20], em cujo desenvolvimento foi levada em consideração a possibilidade de elaboração de hardware específico [21] para suportar o SGBD nele apoiado. Na seção 2 é feita a apresentação resumida do MRO. A seguir, na seção 3, é feita a descrição de uma Máquina de Bases de Dados voltada para apoiar as tarefas de processamento de SGBDs apoiados no MRO. De acordo com a classificação proposta por [22], essa máquina teria arquitetura MIMD, com processamento em memória dedicada, e indexação a nível de páginas. A comunicação com os programas de aplicação é feita com cada estação de trabalho comunicando-se diretamente com a Máquina de Base de Dados. Finalmente na seção 4 são apresentadas conclusões a respeito dessa Máquina.

## 2. - O MODELO DE REPRESENTAÇÃO DE OBJETOS

O MRO é um modelo de representação de dados que permite a modelagem de um alto grau de representação semântica, e foi concebido para dar apoio a Sistemas de Gerenciamento de Bases de Dados que devam ser usados em Sistemas de Apoio a Projetos de Engenharia e em Ambientes de Apoio ao Desenvolvimento de Software.

A fundamentação matemática dos modelos de dados comercialmente disponíveis mais recentes, tais como o Modelo Relacional [10] e o Modelo Entidade-Relacionamento [23] é a Álgebra de Relações [24] [25], a qual é adequada para manipular estruturas de informação que possam ser representadas com naturalidade através de Tabelas. Porém no caso de um modelo de dados voltado a uma representação de maior conteúdo semântico, a Teoria dos Grafos torna-se mais atraente.

A primeira diferença entre uma modelagem usando o ME-R e uma modelagem usando o MRO, é que o ME-R representa entidades (que correspondem a objetos no MRO) e relacionamentos (que correspondem a associações sempre através dos valores dos atributos que os define, e dessa forma, o acesso aos dados é feito exclusivamente através desses valores, sem importar o que significam tais valores armazenados. No MRO, todos os objetos e suas associações correspondentes são classificadas através de diferentes tipos. A identificação de dados está sempre associada ao tipo de objeto que se tem em consideração. Além disso, as associações entre objetos são diretamente armazenadas, e não através de chaves de acesso, as quais podem mesmo ser abandonadas sem que a caracterização ou a integridade dos objetos fique comprometida.

A modelagem de objetos é mais robusta, no sentido de que a recuperação de dados pode ser feita de maneira mais "inteligente". Não é suficiente o reconhecimento de valores de atributos e a indicação de quais entidades ou objetos estão associados em um dado instante são esses valores, mas é importante distinguir os dados obtidos dessa forma, das informações realmente significativas para a operação em andamento. O MRO permite essa maior especificidade do processo de recuperação de dados.

Além disso, o MRO permite que cada objeto seja tratado independentemente dos demais, facilitando a implementação de sistemas em que os dados são mantidos em "Redes" de objetos, cada um com "vida" própria. Essa característica, mais a identificação dos objetos internamente pelo acompanhamento do foco de interesse da operação em andamento facilita o desenvolvimento de uma Máquina de Bases de Dados para suportar SGBDs apoiados no MRO.

## 3. - A ARQUITETURA DA MÁQUINA DE BASES DE DADOS PROPOSTA.

A Máquina de Bases de Dados proposta apresenta uma arquitetura não convencional entre as Máquinas de Bases de Dados existentes, aproximando-se das arquiteturas que empregam conceitos mais inovativos, tal como a máquina SiDBM [26], que não usa discos, mas mantém toda a informação em memória principal, e a máquina descrita em [27], que se apoia em um modelo fundamentado em dígrafos.

A principal diferença que deve ser notada entre as Máquinas de Bases de Dados existentes e a aqui sugerida, é que os modelos convencionais necessitam pesadamente de operações de procura de registros baseados no conteúdo desses registros, tal como é necessário nas operações de seleção e união no modelo relacional [28], ao passo que no Modelo de Representação de Objetos essa necessidade é bastante reduzida. Outra diferença é que operação de ordenação no MRO são necessárias apenas quando da colocação de novos objetos.

A figura 1 mostra a arquitetura lógica para a máquina proposta. Essa figura permite analisar os módulos funcionais da máquina, sem preocupação com a maneira como a máquina será fisicamente implementada. Pode-se notar que ela é composta por quatro tipos de processadores:

- PL - Processadores de Ligação;
- PM - Processadores de Manipulação de Dados;
- PP - Processador de Pedidos de acesso;
- CD - Controlador das Unidades de Disco.

Além disso, existem dois dutos internos que interligam os vários processadores: **Duto de Dados (DD)** e **Duto de Páginas (DP)**. Ambos são dutos síncronos paralelos, de alta velocidade. O Duto de Dados é usado para enviar aos Processadores de Manipulação (PM) os pedidos de acesso solicitados pelas Estações de Trabalho (ET), e para a troca de dados entre as ET e os PM. O Duto de Páginas é usado para carregar as memórias locais dos PM com os dados necessários à execução das solicitações de acesso que cada um estiver atendendo, bem como para a gravação em disco das páginas modificadas por essa execução.

Os **Processadores de Ligação (PL)** são usados para atuar como interface entre cada ET e o Duto de Dados. Não efetuam nenhum processamento lógico dos dados que manipulam, a menos da substituição ou preenchimento de alguns campos nos pedidos de acesso. Atuam principalmente como "buffers" e temporizadores para os dados que trafegam pelo Duto de Dados e que têm como origem e destino aquela específica ET. Os pedidos de acesso de cada ET consistem de conjuntos de parâmetros e dados que são introduzidos (ou recebidos) pelo PL no Duto de

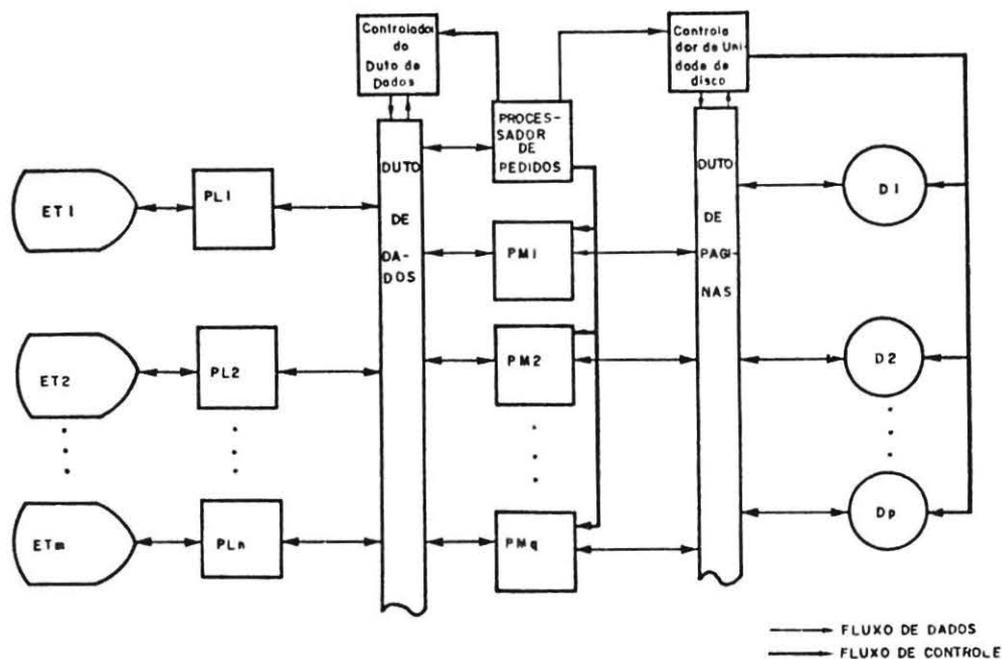


Figura 1 - Arquitetura da Máquina Proposta.

Dados como um "Pacote de Dados". O PL coloca em cada pacote de dados uma identificação de qual é o PL que o originou.

Uma vez tendo sido lançado no Duto de Dados cada Pacote é "ouvido" por todos os Processadores de Manipulação, que verificam se esse pacote deve ser tratado por ele, ou se não for indicado nenhum processador específico, se o pacote pode ser atendido pelos dados presentes na memória do Processador. Quando um PM pode atender ao pedido, ele envia ao final do pacote de dados um sinal indicando a aceitação do pacote.

O PM consulta os dados que tem disponíveis, e atende a todos os pedidos que tenha aceitado, em geral na ordem de chegada. Cada pedido pode ocasionar três tipos de resposta do PM que o atende:

- finalização do pedido. Nesse caso o PM coloca um novo pacote de dados no DD, informando como destino o PL que originou o pedido. Nesse pacote estão a indicação de estado de finalização do pedido, e dados que possam ter sido solicitados.

- solicitação de uma tarefa auxiliar em algum outro PM. Essas tarefas são destinadas a realizar algum processamento acionado automaticamente a partir de outras tarefas em execução. Não são prioritárias para a execução do pedido em andamento, e por isso são armazenadas para execução quando houver processador desocupado. Nesse caso o PM coloca um novo pacote de dados no duto, que mantém as

mesmas características dos que são originados pelos PL, informando os dados e parâmetros necessários, porém com a informação de que não é um pedido prioritário.

- Solicitação de dados não disponíveis para a execução de uma tarefa em andamento. Nesse caso o PM coloca um pedido de dados no DD, e passa o pedido em andamento para o final da fila de pedidos pendentes, aguardando a disponibilidade dos dados solicitados.

Os Processadores de Manipulação não são necessariamente todos iguais; podem especializar-se em alguns tipos, para executar com mais eficiência determinados tipos de processamento necessários numa base de Dados que utilize o MRO. Por exemplo, os processadores que atenderem a identificação de objetos podem se beneficiar de unidades ordenadoras.

O Processador de Pedidos (PP) monitora todos os pacotes que passam no DD e verifica os sinais de aceitação com que os PM respondem aos pacotes que podem atender. Dessa forma, o PP pode manter um registro de todas as operações que estão sendo atendidas em cada PM. Quando nenhum PM envia o sinal de aceitação para o pacote, o PP verifica qual PM tem espaço de memória com capacidade para receber os dados inicialmente necessários para o pedido, que esteja há mais tempo sem uso, e que esteja com baixa carga de trabalho. Localizado um PM nessas condições, o PP solicita o carregamento dos dados adequados ao Controlador das Unidades

de Disco, que efetua a transferência dos dados armazenados em uma unidade de disco na memória do PM escolhido. Ao final dessa transferência, o PP coloca novamente no DD o pacote que não havia sido aceito, tendo como endereço de destino explícito o PM cuja memória recebeu os dados adequados. O PP cuida também dos Pacotes enviados pelos PMs. Os pedidos desses processadores solicitando mais dados são tratados de forma semelhante ao anterior, com a diferença que a indicação para o Controlador das Unidades de Disco de qual é o processador destino de uma transferência já vem especificada. Além disso, ao final dessa transferência o PP coloca no DD apenas uma indicação de que os dados pedidos já podem ser acessados.

O **Controlador de Unidades de Disco (CD)** é apenas um processador capaz de fazer a transformação de pedidos de dados lógicos, que são manipulados por todos os demais processadores, em endereços físicos em disco, identificando os discos e trilhas que devem ser acessados. Esse processador poderá também fazer a busca escolhendo caminhos de acesso mais curtos, no caso de haver uma sucessão de pedidos de acesso.

Deve-se ressaltar que o PP, os PMs, e o CD utilizam processadores cujos programas são em parte permanentemente residentes, tais como aqueles que permitem a inicialização do sistema e o carregamento de outros programas, e outros são carregados de disco usando o mesmo duto de páginas usado para o carregamento dos dados da Base de Dados.

Essencialmente, cada PM consta de uma Unidade Central de Processamento (UCP), dois ou mais bancos de memória com possibilidade de operação de transferência de dados através de Acesso Direto à Memória, e interface para o Duto de Dados.

A memória acessada pelos PMs deve ser dividida em bancos, e o registro de qual é o conteúdo de cada banco deve poder ser reconhecido separadamente. Além disso, o PP pode travar o acesso a alguns blocos dessa memória para poder autorizar a transferência de dados ao CD. A estruturação em bancos de memória facilita a operação de travamento, fazendo com que o bloco travado corresponda a um banco.

Os pedidos de acesso de dados que se originam nas ET, após o tratamento nos PL se referenciam a objetos através de seus códigos. Esses códigos são usados pela Máquina de Base de Dados para derivar o índice da página de dados do arquivo de Objetos da base de dados. O número índice de cada página armazenada em um banco de dados do Processador é o dado principal usado para a aceitação de um pacote de dados. Sempre que houver uma referência a um objeto, verifica-se através de seu código se o

processador pode atender ao pedido ou não.

É atribuição do PP zelar para que não exista a mesma página de dados duplicada na memória de mais de um PM ao mesmo tempo, e assim, uma vez que um PM aceite um pedido, nenhum outro o terá feito. Esse fato, além de simplificar o controle do sinal de aceitação do pedido, garante a consistência das alterações efetuadas, eliminando a necessidade de se efetuar travas de consistência sobre os registros ou páginas que devem ser modificados.

O PM executa os pedidos aceitos sequencialmente, e sempre que estiver usando um banco de dados, deve travá-lo para evitar o seu travamento pelo PP. Além disso, sempre que aceita um pacote, analisa os índices das páginas implicadas por esse pedido, e mesmo que não o processe imediatamente, indica que a página correspondente não deve ser retirada de sua memória. O número de pedidos de acesso pendentes que necessitam de cada página disponível deve ser contado, apesar de apenas a informação de estar ela prendendo uma página ser informado ao PP. Quando o final do processamento de um pedido resultar em não mais serem necessários os dados de uma determinada página para nenhum dos pedidos pendentes, essa informação é transmitida ao PP, que a utiliza para avaliar se esse processador pode ter essa página substituída de modo a poder atender a um pacote de dados enviado por uma ET que não foi aceito por nenhum outro PM.

### 3.1. - O Formato de um Pacote de Dados.

O formato de cada pacote é mostrado na Figura 2. O primeiro byte indica o código de grupo da função solicitada por esse pacote. O Código de Grupo de Função serve para desativar as Interfaces que estão conectadas a Processadores que não podem responder a esse Pacote. A seguir vem o código da própria função solicitada, e depois a informação sobre o Processador Destino do pacote. Se esse campo é usado, o processador destino deve obrigatoriamente aceitar o pacote de dados.

A seguir vem o índice da página que é necessariamente referenciada por essa solicitação, tal como o código do objeto se a solicitação corresponder a um pedido de dados sobre um objeto. A seguir vem a informação de qual é o processador origem desse pacote, para que o Processador que aceitar o pacote possa designá-lo diretamente em um eventual pacote de resposta. Essa informação não é modificada pelo Processador de Pedidos se ele for retransmitir essa mensagem, permanecendo a informação do processador origem que primeiro transmitiu o pacote.

Depois do oitavo byte, cada pacote contém um

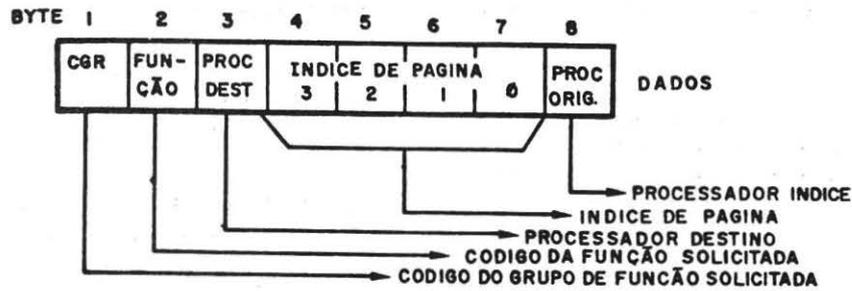


Figura 2 - Formato de um Pacote de Dados

número variável de bytes que contém os parâmetros e dados da função solicitada. Esse número deve ter um valor mínimo e máximo a ser determinado. A princípio, um número mínimo de 4 bytes deve ser considerado, para permitir a inicialização do processador que aceitar o pacote, e um número máximo de 120 bytes seria suficiente para os dados necessários para qualquer das funções do MRO.

O final de um pacote de dados deve ser sempre seguido por um sinal de aceitação de um processador que tenha recolhido e aceito o pacote. Esse final é indicado pelo processador origem da mensagem através de um sinal de controle do duto de dados, e informa a todos os demais processadores o final desse pacote, com a indicação de liberação do duto. O sinal de aceitação de um pacote é dado usando um período a mais do duto para a transmissão do sinal de aceitação do processador destino, o qual é acompanhado de um Byte contendo o número do processador que o enviou. Para a transmissão desse byte, não é necessária a solicitação de uso do duto, uma vez que somente um processador terá condições de enviar esse sinal.

### 3.2. - Implementação da Máquina de Bases de Dados.

A descrição do comportamento da máquina mostra que a implementação física não depende de circuitos especiais para desempenhar funções específicas, mas apenas um conjunto de processadores independentes ligados através de dutos de alta velocidade de transferência de dados, e com memória não compartilhada.

O modelo de dados utilizado tem uma implementação feita para operar em mono-processador, tendo sido utilizada em IBM-PC e VAX. Nessa implementação, algumas medidas têm sido efetuadas, e está em andamento a elaboração de um modelo estocástico [29] que permita apontar números adequados para o tamanho da memória de cada processador, tamanho

de cada bloco nessa memória, número de processadores em função da carga de trabalho da máquina, etc.

### 4. - CONCLUSÕES

Neste trabalho foi apresentada a arquitetura de uma máquina voltada para o processamento das tarefas de Gerenciamento de uma Base de Dados apoiada no Modelo de Representação de Objetos. Esse modelo foi desenvolvido levando em consideração as necessidades de um Sistema de Apoio ao Projeto e Desenvolvimento para Engenharia, e a possibilidade de ter suas operações suportadas por um Hardware específico.

As considerações para a proposta desta máquina surgiram do desenvolvimento do Modelo de Representação de Objetos, e da constatação que muitas Estações de Trabalho deveriam poder acessar a base de dados simultaneamente.

Um ponto importante a ser destacado é que a Unidade Central de Processamento (Micro-processador) utilizada em cada Processador dessa máquina não é crítico, podendo mesmo existir Processadores que usam diferentes Micro-processadores num mesmo sistema. O problema causado pela existência de vários Micro-processadores diferentes seria apenas a necessidade de se manter cópias dos mesmos programas mantidos para as diferentes linguagens de máquina de cada Micro-processador. Esse ponto é importante em um sistema com a complexidade da Máquina de Base de Dados aqui proposta, pois permite que seu desenvolvimento possa acompanhar por muito tempo o rápido desenvolvimento que hoje existe nessa área.

### BIBLIOGRAFIA

- [1] Slotnik, D. L. - "Logic per Track Devices", in *Advancing in Computers*, Vol 10. F. Alt.

- Ed. New York: Academic, 1970, pp. 291, 296.
- [2] Coulouris, G. H.; Evans, J. M.; Mitchell, R. W. - "Towards Content Addressing in Databases", Computer Journal, Vol 15., Nro. 2, May 72, pp. 95-98.
- [3] Canaday, R. H. et alli - "A Backend Computer for Database Management", Commun. ACM, Vol. 17, Nro. 10, October 74, pp. 575-582.
- [4] Langdon, G. G. - "A Note on Associative Processors for Data Management", ICM TODS, Vol3, June 1978, pp. 148-158.
- [5] Hawthorn, P. B.; DeWitt, D. J. - "Performance Analysis of alternative Database Machine Architecture", IEEE Trans. on Software Engineering, Vol SE-8, Nro.1, January 1982, pp. 61-75.
- [6] Kitsuregawa, M.; Fushimi, S.; Tanaka, H.; Moto-oka, M. - "Memory Management Algorithms in Pipeline Merge Sorter", in Proc. 4th Int. Workshop on Database Machines, March 1985, pp. 208-232.
- [7] Shibayama, S. et alli - "Query Processing Flow on RDBM Delta's Functionally-Distributed Architecture" in Proc. Int. Conference on 5th Generation Computer Systems, 1984, pp. 427-435.
- [8] Gamerman, S.; Scholl, M. - "Hardware versus Software Data Filtering: The VERSO Experience", in Proc. 4th Int. Workshop on Database Machines, March 1985, pp. 112-136.
- [9] Faudemay, P.; Valduriez, P. - "Design and Analysis of a Direct Filter Using Parallel Comparators", in Proc. 4th Int. Workshop on Database Machines, March 1985, pp. 137-150.
- [10] Codd, E. F. - "A Relational Model for Large Shared Data Banks", Commun. ACM, Vol 13, Nro. 6, June 70, pp. 377-387.
- [11] Babb, E. - "Implementing a Relational Database by Means of Specialized Hardware", ACM TODS, Vol. 4, March 1979, pp. 1-29.
- [12] Valduriez, P.; Gardarin, G. - "Join and Semijoin Algorithms for a Multiprocessor Database Machine", ACM Trans. on Database Systems, Vol. 9, Nro. 1, March 84, pp. 133-161.
- [13] Hillyer, B. K.; Shaw, D. E.; Nigam, A. - "NON-VON's Performance on Certain Database Benchmarks", IEEE Trans. on Software Engineering, Vol SE-12, Nro. 4, April 1986, pp. 577-583.
- [14] Taki, K. et alli - "Hardware Design and Implementation of the Personal Sequential Inference Machine (PSI)", in Proc. Int. Conference on 5th Generation Computer Systems, 1984, pp. 398-409.
- [15] Sakai, H. et alli - "Design and Implementation of the Relational Database Engine", in Proc. Int. Conference on 5th Generation Computer Systems, 1984, pp. 419-426.
- [16] Kakuta, T.; Miyasaki, N.; Shibayama, S.; Yokota, H.; Murakami, K. - "The design and Implementation of Relational Database Machine Delta", in Proc. 4th Int. Workshop on Database Machines, March 1985, pp.
- [17] Hammer, M.; McLeod, D. - "Database Description with SDM: A Semantic Database Model", ACM Trans. Database Systems, Vol 6, Nro. 3, September 1981, pp. 351-386.
- [18] Abiteboul, S.; Hull, R. - "IFO: A Formal Semantic Database Model", ACM TODS, Vol. 12, Nro. 4, December 1987, pp. 525-565.
- [19] Su, S. Y. W. - "Modeling Integrated Manufacturing Data with SAM\*" IEEE Computer, Vol. 19, Nro. 1, January 86, pp. 34-49.
- [20] Traina Jr., C.; Slaets, J. F. W. - "Um Modelo de Representação de Objetos", in Anais do 3 Simposio Brasileiro de Banco de Dados, Recife, Março de 1988, pp. 227-242.
- [21] Traina Jr., C. - "Máquina e Modelo de Dados Dedicados para Aplicações de Engenharia", Tese apresentada ao IFQSC-USP para obtenção do Título de Doutor, Dezembro 86.

- [22] Qadah, G. Z.; Irani, K. B. - "A Database Machine for Very Large Relational Databases", IEEE Trans. on Computers, Vol. C-34, Nro. 11, November 1985, pp. 1015-1026.
- [23] Chen, P. P. - "The Entity-Relationship Model - Toward a Unified View of Data" ACM Trans. Database Systems, Vol. 1, Nro. 1, march 76, pp. 9-36.
- [24] Ng, A. P. - "Further analysis of the Entity-Relationship approach to Database Design" IEEE Trans. Software Engineering Vol. SE-7 Nro. 1, January 81, pp. 85-99.
- [25] Parent, C.; Spaccapietra, S. - "An Algebra for a General Entity-Relationship Model" IEEE Trans. Software Engineering, Vol. SE-11 Nro. 7, July 85, pp. 634-643.
- [26] Leland, M. D. P.; Roome, W. D. - "The Silicon Database Machine", in Proc. 4th Int. Workshop on Database Machines, March 1985, pp. 169-189.
- [27] Bic, L.; Hartmann, R. L. - "Hither Hundreds of Processors in a Database Machine", 153-168.
- [28] Bitton, D.; Boral, H.; DeWitt, D. J.; Wilkinson, W. K. - "Paralell Algoritms for the Execution of Relational Database Operations", ACM Trans. on Database Systems, Vol. 8, Nro. 3, September 83, pp. 324-353.
- [29] Sales, M. C.; Traina Jr. C. - "Filas  $M/GI/n+1/0$ , com Realimentação Bernoulli: Um Modelo Protótipo para uso de Processadores Paralelos para uma Máquina de Base de Dados", a ser publicado nos anais do 8 Simpósio Nacional de Probabilidade e Estatística, Rio de Janeiro, Julho de 1988.