

PUC

Série : Monografias em Ciência da Computação

No. 10/78

A ARQUITETURA DO REDAS
(UM SISTEMA DE GERÊNCIA DE BANCO DE DADOS)

por

Rubens N. Melo

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rua Marquês de São Vicente, 225 – CEP-22453

Rio de Janeiro – Brasil

Série : Monografias em Ciência da Computação

No. 10/78

Editor da Série: Michael Challis

Junho 1978

A ARQUITETURA DO REDAS
(UM SISTEMA DE GERÊNCIA DE BANCO DE DADOS)

por

Rubens N. Melo

* Este trabalho foi parcialmente financiado pela FINEP.

For copies contact:

Rosane T.L.Castilho
Head, Setor de Documentação e Informação
Depto. de Informática - PUC/RJ
Rua Marques de São Vicente, 209 - Gávea
22453 - Rio de Janeiro - RJ - Brasil

RESUMO

REDAS é um sistema de gerência de banco de dados baseado num modelo de entidades e relações apresentado em [3]. Este artigo resume os principais aspectos da arquitetura do sistema. Muitos dos conceitos e termos empregados vêm das recomendações do relatório ANSI/SPARC [5]. As linguagens do sistema são apresentadas porém os detalhes de implementação não são mostrados neste artigo.

Palavras chave: SGBD; Banco de Dados; Linguagem de Definição de Dados; Linguagem de Manipulação de Dados; Modelos de Dados; Modelo Conceitual; ANSI/SPARC.

ABSTRACT

REDAS is a database management system based on an entity-relationship model presented in [3]. This paper surveys the main aspects of the architecture of the system. Many of the concepts and terms employed come from the recommendations of the ANSI/SPARC report [5]. The languages provided by the system are introduced. The implementation details are not shown in this paper.

Keywords: DBMS; Data Base; Data Definition Language; DDL; CSL; SSL; Data Manipulation Language; DML; Data Model; Conceptual Model; ANSI/SPARC.

S U M A R I O

1 - INTRODUÇÃO.....	1
2 - ALGUNS ASPECTOS DA ARQUITETURA DO SISTEMA.....	2
2.1 - Facilidades de Definição de Dados.....	2
2.2 - Facilidades para Manipulação de Dados.....	2
2.3 - Aspectos de Segurança e Integridade dos Dados.....	3
3 - AS LINGUAGENS DE DEFINIÇÃO DOS DADOS.....	4
3.1 - A Linguagem CSL.....	4
3.2 - A Linguagem SSL.....	9
4 - A MANIPULAÇÃO DE DADOS.....	11
4.1 - A Linguagem FORTRAN-DML	11
4.2 - A Linguagem de Consultas REQUEL.....	16
5 - COMENTÁRIOS FINAIS.....	16
REFERÊNCIAS.....	20

1 - INTRODUÇÃO

REDAS é um sistema de gerência de Banco de Dados que teve sua primeira versão implementada no sistema IBM 1130 do Instituto Tecnológico da Aeronáutica. Esta implementação inicial foi em parte limitada pelo próprio sistema operacional hospedeiro, bem como por algumas simplificações naturais em versões iniciais.

O projeto do sistema REDAS, iniciado em meados de 1976, teve por finalidade desenvolver um sistema de Banco de Dados, possível de se implementar numa máquina pequena como o IBM 1130, mas portátil o suficiente, para permitir com algumas modificações, a implementação em computadores de maior porte. Neste artigo apresentamos alguns aspectos importantes da arquitetura do sistema.

Para o desenvolvimento do projeto foram considerados mais relevantes os seguintes aspectos:

a) Portabilidade

O sistema seria desenvolvido em FORTRAN e não seria orientado para nenhuma máquina em particular.

b) Representatividade

As estruturas conceituais do modelo de dados deveriam ser simples e permitir modelar diretamente situações reais.

c) Independência de dados

Os programas de aplicação seriam totalmente independentes das estruturas internas, suas organizações ou métodos de acesso.

d) Possibilidade de expansão

Ao menos para inclusão de novas estruturas de armazenamento e métodos de acessos, seria possível expandir o sistema.

e) Simplicidade

Para interagir com o sistema o usuário necessitaria tão somente qualificar as informações de interesse.

O modelo conceitual de dados adotado, baseado em classes de entidades, classes de relações e atributos sugeriu o nome do sistema: REDAS (Relation Entity Data base System) e também o nome da linguagem de consulta do sistema: REQUEL (Relation Entity QUery Language) [1]. Tal modelo conceitual inspirado no enfoque infológico para Banco de Dados propostos em [2] é extensamente discutido em [3].

2 - ALGUNS ASPECTOS DA ARQUITETURA DO SISTEMA

A arquitetura do sistema compreende os seguintes aspectos importantes que serão resumidos neste artigo:

2.1 - Facilidades de Definição de Dados

. No nível conceitual

Neste nível o sistema basicamente apresenta uma linguagem de especificação conceitual (CSL) que permite definir e caracterizar as classes de entidades e de relações, de interesse para um certo sistema de informação.

Informações sobre as classes, tais como, o nome, os atributos, a representação dos atributos e o identificador, são armazenadas em tabelas do sistema referenciados como esquema conceitual.

* - Na arquitetura do sistema, bem como na terminologia empregada, procuramos seguir as recomendações do relatório da ANSI/SPARC [5].

A linguagem de especificação conceitual é essencialmente utilizada pelo ABD (Administrador de Banco de Dados) * permitindo não apenas especificar o esquema conceitual, mas também a atualização, inserção e exclusão de classes de entidades e relações.

. No nível interno

No nível interno basicamente o sistema suporta uma linguagem de especificação estrutural (SSL), que permite associar às classes definidas conceitualmente, uma estrutura de armazenamento dos dados e caracterizá-la.

Analogamente ao esquema conceitual, as especificações da SSL são apresentadas em tabelas do sistema, referenciadas como esquema interno. A linguagem de especificação estrutural é de uso restrito ao ABD e tem como objetivo permitir a definição e a atualização das informações a nível interno.

. No nível externo

A definição de modelos externos quer como submodelos do modelo conceitual ou como outros modelos não foi considerada nesta versão, ficando para estudos posteriores.

2.2 - Facilidades para Manipulação de Dados

. FORTRAN/DML

Foi definido um pré-processador que estende o FORTRAN com comandos de manipulação de dados do Banco de Dados além de outras facilidades, tais como: tratamento de cadeias de caracteres e estruturas de controle adicionais. Os comandos de manipu-

* - Na realidade, de acordo com [5] a especificação do modelo conceitual é de responsabilidade do Administrador Empresarial, ficando com o ABD a responsabilidade pelo modelo interno.

lação de dados são traduzidos para chamadas de rotinas dos métodos de acesso disponíveis no sistema de acordo com a estrutura do Banco de Dados especificada em SSL pelo ABD. O nível em que os usuários interagem com o Banco de Dados com esta linguagem é o nível conceitual, isto é, em termos de Classes de Entidades, Classes de Relação e Atributos. O nível interno é transparente ao usuário.

REQUEL

Uma linguagem de consultas com sintaxe simples, sem envolvimento de operadores matemáticos complexos, que permite a formulação fácil de perguntas simples sendo porém bastante geral para permitir perguntas mais complexas. Esta linguagem que é apresentada em detalhes em [1] está em fase de implementação.

2.3 - Aspectos de Segurança e Integridade dos Dados

Um subsistema de segurança e integridade dos dados está sendo pesquisado e deverá ser relatado brevemente em [4].

3 - AS LINGUAGENS DE DEFINIÇÃO DOS DADOS

3.1 - A Linguagem CSL

A CSL desenvolvida para o sistema REDAS, embora apresente uma sintaxe bastante simples, permite ao DBA definir e manter convenientemente o esquema conceitual.

Os comandos desta linguagem resumidos na figura 1 serão descritos através de exemplos. Basicamente temos facilidades para:

1. Especificação de Classes:

ENTITY (especifica classes de entidades)

RELATION (especifica classe de relações)

2. Criação e Manutenção do Esquema Conceitual

OPEN (permite o acesso ao esquema conceitual)

INSERT (insere as definições de classes)

UPDATE (atualiza as definições de classes)

DELETE (exclui definições de classes)

CLOSE (finaliza as interações com o esquema conceitual)

COMMENT (usado para inserir comentários na especificação)

Especificação de classes

Permite declarar o nome de uma classe de entidades (ENTITY) ou de relações (RELATION) e de seus atributos, a representação destes atributos e qual o identificador da classe especificada. Uma classe de relações é identificada pelas classes de entidades associadas. No entanto, para uniformidade de representação e para realçar a associação, indicamos também o identificador da classe de relações pelos nomes das classes de entidades associadas.

Exemplo de Especificação de Entidade:

ENTITY FORNECEDOR (NUMERO, NOME, CIDADE)

NUMERO INTEGER,

NOME CHAR (30),

CIDADE CHAR (10),

IDENTIFIER (NUMERO).

```

OPEN SCHEMA nome do esquema

COMMENT UM RESUMO DA CSL

INSERT

ENTITY nome da classe (at1 , ... , atk )
    at1 representação
    :
    :
    atk representação
IDENTIFIER (at1 , ... , atj ).
RELATION nome da classe (e1 , ... , ek , at1 , ... , atj )
    at1 representação
    .
    :
    atj representação
IDENTIFIER (e1 , ... , ek ).

UPDATE

(novas especificações de classes para substituir
as de mesmo nome)

DELETE nome-de-classe1 , ... , nome-de-classek .
CLOSE SCHEMA nome do esquema

```

Figura 1 - Resumo da CSL

Exemplo de Especificação de Relação:

```
RELATION FORNECE (FORNECEDOR, ITEM, QUANTIDADE)
    QUANTIDADE REAL,
IDENTIFIER (FORNECEDOR, ITEM).
```

Obs.: FORNECEDOR e ITEM já ~~de~~ devem ter sido declarados como classes de entidades.

Convém observar que FORNECEDOR e ITEM são nomes / das classes de entidades relacionadas. Uma ocorrência de FORNECE só existirá se existirem, no Banco de Dados, as entidades (ocorrências) de FORNECE DOR e ITEM respectivamente relacionadas. A exclu são de uma destas entidades implica na exclusão da correspondente ocorrência de FORNECE.

O exemplo abaixo embora simples, mostra o uso dos co - mandos da CSL para a criação do esquema conceitual de um pequeno Banco de Dados.

```
OPEN SCHEMA EMPRESA
```

```
    COMMENT EXEMPLO DA UTILIZAÇÃO DOS COMANDOS
        DA CSL.
```

```
    COMMENT CLASSE DE ENTIDADES FORNECEDOR.
```

```
INSERT
```

```
    ENTITY FORNECEDOR (NUMERO, NOME, CIDADE)
        NUMERO    INTEGER,
        NOME      CHAR(30),
        CIDADE    CHAR(10),
        IDENTIFIER (NUMERO).
    COMMENT CLASSE DE ENTIDADES ITEM.
```

ENTITY ITEM (NUMERO, NOME, COR, PESO)

NUMERO INTEGER,

NOME CHAR(25),

COR CHAR(8),

PESO REAL,

IDENTIFIER (NUMERO).

COMMENT CLASSE DE RELACOES FORNECE

RELATION FORNECE (FORNECEDOR, ITEM, QUANTIDADE)

QUANTIDADE REAL,

IDENTIFIER (FORNECEDOR, ITEM).

CLOSE SCHEMA EMPRESA.

Mais tarde este esquema poderia ser modificado utilizando-se os comandos UPDATE, DELETE, etc. Por exemplo:

OPEN SCHEMA EMPRESA

UPDATE

ENTITY FORNECEDOR (NOME, CIDADE)

NOME CHAR(40),

CIDADE CHAR(10),

IDENTIFIER NOME.

RELATION FORNECE (FORNECEDOR, ITEM)

IDENTIFIER (FORNECEDOR, ITEM).

DELETE ITEM.

INSERT ITEM(NUMERO, PESO)

NUMERO INTEGER,

PESO REAL,

IDENTIFIER (NUMERO).

CLOSE SCHEMA EMPRESA .

É óbvio que as modificações no esquema conceitual vão até um certo limite sem que as aplicações tenham que ser reprogramadas. Comentamos sobre isso nas próximas seções.

3.2 - A Linguagem SSL

O modelo interno deve conter informações sobre o modo como as classes de entidades e de relações serão implementadas internamente. Nesten nível, o sistema apresenta uma linguagem de especificação interna (SSL), que permite:

- estabelecer características destes arquivos.
- especificar a organização de dados escolhida.
- especificar os atributos escolhidos como índices secundários.

Estas informações são armazenadas em tabelas que constituem o esquema interno do Banco de Dados.

A SSL, além de definir o esquema interno, possibilita ainda a sua modificação, permitindo que o ABD:

- crie ou destrua índices secundários.
- modifique características físicas de arquivos como seu tamanho, o tamanho dos registros, etc.
- modifique a organização dos arquivos de dados.

A seguir descreveremos os comandos existentes e como são utilizados, através de exemplos. Um resumo destes comandos é apresentado na figura 2.

Exemplo: OPEN SCHEMA ORG-INTERNA.

```
COMMENT ESTRUTURA DE ARMAZENAMENTO DAS CLASSES
ASSIGN
    (FORNECEDOR) TO DATASET (FORNE)
    DSD = (descrição do arquivo)
    (ITEM) TO DATASET (ITEM)
    DSD = (descrição do arquivo)
    (FORNECE) TO DATASET (SUPLY)
    DSD = (descrição do arquivo).
COMMENT INDICES SECUNDARIOS.
CREATE INDEX ON
    (CIDADE) OF (FORNECEDOR)
    DSD = (descrição do arquivo)
    (FORNECEDOR) OF (FORNECE)
    DSD = (descrição do arquivo)
    (ITEM) OF (FORNECE)
    DSD = (descrição do arquivo).
COMMENT FINAL DE EXEMPLO UTILIZANDO A SSL.
CLOSE SCHEMA ORG-INTERNA.
```

Um exemplo de descrição de arquivo seria:

```
DSD = (ORG = HASH, FSIZE = a, BSIZE = b, OBINT = c)
```

onde ORG, FSIZE, BSIZE e OBINT significam respectivamente a organização do arquivo, o tamanho do arquivo, o tamanho do "bucket" (bloco de registros) e o intervalo entre "buckets" de overflow / (área de overflow distribuída).


```
OPEN SCHEMA nome do esquema interno.

COMMENT ASPECTO DE UMA ESPECIFICAÇÃO INTERNA EM SSL.

ASSIGN
  (nome da classe) TO DATASET (nome do arquivo)
    DSD = (descrição do arquivo)
      :
      :

DELETE
  (nome da classe),
    :
    :

UPDATE
  (nome da classe) |TO DATASET (nome do arquivo)|
    |DSD = (nova descrição do arquivo)|,
      :
      :

CREATE INDEX ON
  (nome do atributo) OF (nome da classe)
    DSD = (descrição do arquivo),
      :
      :

DESTROY INDEX ON
  (nome do atributo) OF (nome da classe),
    :
    :

MODIFY INDEX ON
  (nome do atributo) OF (nome da classe)
    DSD = (nova descrição do arquivo),
      :
      :

COMMENT FIM DO RESUMO.

CLOSE SCHEMA nome do esquema interno.
```

Figura 2 - Resumo dos Comandos da SSL

4 - A MANIPULAÇÃO DE DADOS

4.1 - A Linguagem FORTRAN-DML

A DML proposta para o sistema REDAS utiliza como linguagem hospedeira o FORTRAN. Embora esta linguagem seja bastante limitada, para manipulação de alguns tipos de dados (por exemplo, "STRING" de caracteres), a escolha recaiu sobre ela basicamente pelas mesmas razões pela qual foi utilizada para implementação de todo o sistema: a sua grande portabilidade, pois praticamente todos os computadores / tem compilador FORTRAN.

Algumas das limitações foram resolvidas pelo desenvolvimento de facilidades de apoio (por exemplo, comandos especiais para manipulação de "string"). Outras extensões tais como estruturas de controle para programação estruturada em FORTRAN serão ainda adicionadas.

A linguagem é destituída de qualquer referência a estruturas internas, como nomes de arquivos e suas estruturas assegurando / assim um bom nível de independência de dados do sistema.

O esquema adotado, de referências lógicas aos arquivos através dos vetores de referência com as características de nível interno que só são inicializados em tempo de execução, garante esta independência. Desta forma, em nenhum comando da DML desenvolvida o usuário necessita especificar detalhes sobre os arquivos do sistema.

Uma vez que nem todas as operações sobre um Banco de Dados são executadas com sucesso (por exemplo, achar um elemento de uma classe que ainda não tenha sido armazenado), é desejável que o usuário tenha meios de interpretar mensagens do sistema comunicando se a operação teve sucesso ou não.

Comandos de Recuperação

- 1 - FIND classe (id₁ = valor, ..., id_n = valor) | SET cursor;
- 2 - GET (lista de variáveis) classe (lista de atributos)
|GIVEN cursor|
- 3 - GET (lista de variáveis) classe de entidades
(lista de atributos) FROM classe de relações
|GIVEN cursor|
- 4 - FOR classe atributo op valor
-
- lista de comandos
-
ENDFOR
- 5 - FOR ALL classe
-
- lista de comandos
-
ENDFOR

Comandos de Modificação

- 1 - STORE classe (atributo₁ = valor ... atributo_n = valor)
- 2 - UPDATE classe (id₁ = valor, ..., id_n = valor, atributo₁
valor ... atributo_k = valor)
- 3 - DELETE classe (id₁ = valor, ..., id_n = valor)

Comandos Auxiliares

- 1 - OPEN DATABASE nome do banco de dados
- 2 - CLOSE DATABASE nome do banco de dados
- 3 - DECLARE
-
- declarações de especificação
-
- 4 - ASSIGN variável = valor

FIGURA 3 - Resumo dos comandos da DML

Na DML do sistema REDAS, o usuário tem acesso a um indicador do "STATUS" da operação denominado ERROR.

Basicamente os comandos existentes podem ser divididos em / dois grupos:

- Comando de recuperação e modificação de dados;
- Comandos auxiliares.

Os exemplos a seguir procuram mostrar o uso destes comandos que estão resumidos na Figura 3. O Banco de Dados da referência é o do exemplo da seção 3.

Exemplo 1 - "Obter o nome dos fornecedores e o nome dos í- tens que fornecem, quando a quantidade fornecida for maior que 15".

Esta pergunta será expressa como:

DECLARE

INTEGER X(30), Y(25)

OPEN DATABASE EMPRESA

FOR FORNECE.QUANTIDADE > 15

GET(X) FORNECEDOR(NOME) FROM FORNECE

GET(Y) ITEM(NOME) FROM FORNECE

WRITE(IMP,400) X, Y

400 FORMAT(1X,30A1, 10X, 25A1)

END FOR

CLOSE DATABASE EMPRESA

Consideremos outra pergunta:

Exemplo 2 - "Obter o nome dos fornecedores que fornecem itens cujo peso seja maior que o peso de todos os itens de cor verme - lha".

O modo que esta pergunta é expressa utilizando os comandos da DML não é único. Desta forma apresentaremos um programa sem a preocupação de otimizá-lo.

```
DECLARE
    INTEGER X(30)
    REAL Y,Z
OPEN DATABASE EMPRESA
    FOR ALL FORNECE
        GET(X) FORNECEDOR(NOME) FROM FORNECE
        GET(Y) ITEM(PESO) FROM FORNECE
        FOR ITEM.COR = "VERMELHO"
            GET(Z) ITEM(PESO)
            IF(Y - Z) 300, 300, 100
100    ENDFOR
        WRITE(IMP,200) X
200    FORMAT(1X, 30A1)
300    ENDFOR
CLOSE DATABASE EMPRESA
```

Este dois exemplos nos dão uma idéia da utilização dos comandos da DML, misturados aos comandos da linguagem hospedeira e nos permitem observar:

- a ausência de referências ao nível interno.
- a importância de detalhes a nível de representação (tipo de das variáveis).
- a simplicidade da sintaxe dos comandos.

4.2 - A Linguagem de Consultas REQUEL

Detalhes sobre esta linguagem são apresentados em [1] e alguns de seus aspectos importantes serão relatados em outro artigo. Apenas como ilustração, os dois programas do exemplo anterior seriam expressos pelas seguintes consultas em REQUEL:

```
Exemplo 1: PRINT (FORNECEDOR.NOME, ITEM.NOME)
           WHERE((FORNECEDOR,ITEM) ARE RELATED BY FORNE-
           CE AND FORNECE.QUANTIDADE > 15)
```

```
Exemplo 2: PRINT (FORNECEDOR.NOME)
           WHERE((FORNECEDOR,ITEM) ARE RELATED BY FORNE-
           CE AND ITEM.PESO > ALL (ITEM.PESO)
           WHERE(ITEM.COR = "VERMELHO"))
```

5 - COMENTÁRIOS FINAIS

Embora a implementação inicial seja orientada para computadores, pouco versáteis em relação a manipulação de arquivos, a arquitetura proposta inicialmente para o sistema "REDAS", não obstante sua simplicidade, apresenta características importantes como a facilidade de incorporação ao sistema de novas estruturas de armazenamento e novos métodos de acesso.

. Estrutura do Sistema

A figura 4 permite visualizar os aspectos mais importantes da estrutura proposta inicialmente para o sistema e que compreende quatro tipos de arquivos gerenciados pelo sistema.

a) Catálogo do sistema

O catálogo do sistema contém uma descrição de cada banco de dados criado no sistema, em particular a definição e

o dimensionamento de cada banco de dados em termos do número de classes de entidades e de relações, número de índices secundários, número de atributos, etc.

b) Arquivo de definição de banco de dados

Correspondem aos esquemas conceitual e interno descritos anteriormente. A partir desses arquivos é possível obter todas as informações sobre a estrutura e a organização dos arquivos associados às classes de entidades ou relações definidas no banco de dados.

c) Arquivos de classes

Contém os registros associados às entidades e relações de uma determinada classe. Através do sistema podem ser acessados e manipulados por qualquer usuário autorizado.

d) Arquivos de índices secundários

Para cada atributo que possui índice secundário, esse arquivo contém uma lista de ponteiros para as ocorrências de um particular valor do atributo nos arquivos de classes. São utilizados apenas pelo sistema, não são visíveis ao usuário, possuem organização e método de acesso definidos pelo ABD através da SSL.

Alguns comentários ainda se fazem necessários com respeito aos arquivos do sistema:

Cabe ao ABD criar e inicializar os arquivos do sistema.

Embora desejável, os esquemas conceitual e interno não possuem a mesma estrutura dos arquivos de classes e de índices secundários. Isto pode ser justificado tendo em conta a simplicidade da estrutura proposta para estes arquivos e que,

principalmente devido as suas dimensões, na implementação inicial, são utilizados como tabelas permanentes na memória. Convém notar que essas tabelas não são usadas correntemente e que para um banco de dados com 20 classes de entidades ou relações e 50 atributos, a memória ocupada em cada caso corresponde aproximadamente a 1K palavras do 1130.

- . Os arquivos de índices secundários foram inicialmente mapeados cada um num arquivo físico. A modificação para mapear todos os índices num só arquivo está sendo estudada.
- . Além das linguagens de especificação conceitual e interna propostas anteriormente, facilidades para criação e caracterização do banco de dados a nível do catálogo do sistema precisam ser desenvolvidas.

Estruturas de Armazenamento (Organização de Arquivos)

As estruturas de armazenamento propostas são comuns a maioria dos sistemas já implementados, ou seja, HASH, SEQUENCIAL, SEQUENCIAL INDEXADO, sendo disponível, no momento, apenas a estrutura HASH e SEQUENCIAL.

Métodos de Acesso

Um dos objetivos principais da implementação inicial consistia em assegurar a flexibilidade do sistema em relação a inclusão de novos métodos de acesso. Para atingir esse objetivo, consideramos dois níveis de acesso, o lógico e o físico. Após um processamento inicial, as interações com os arquivos do sistema são realizadas por um conjunto de rotinas (FIND, GET, STORE,

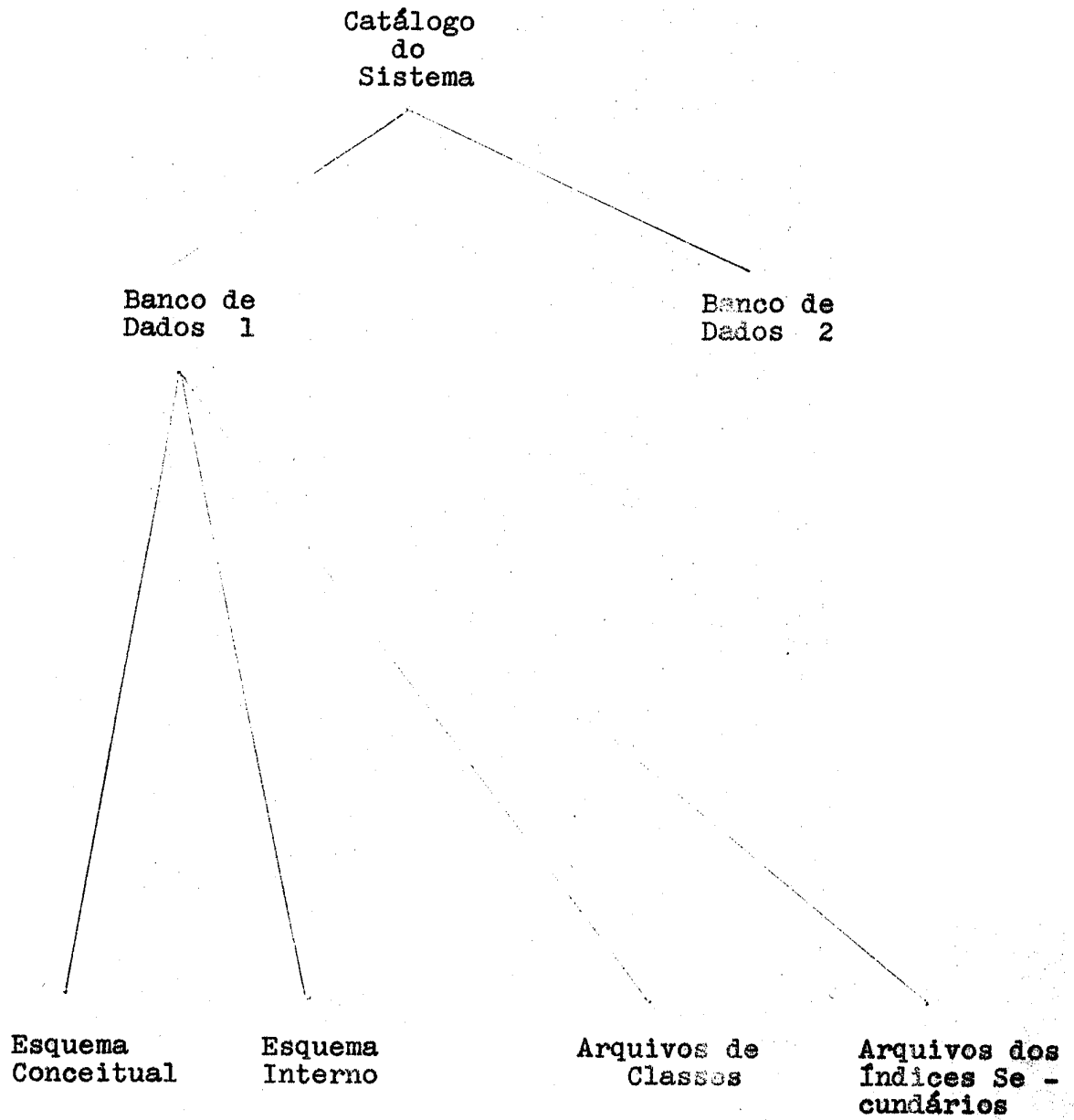


Figura 4 - Estrutura Inicial do Sistema

DELETE, UPDAT, FOR) que utilizam como parâmetros os vetores de referência NVET e NDESC (informações sobre estrutura do registro, existência de índices secundários, etc) e um vetor DESC (vetor que contém os descritores dos arquivos do sistema). Essas rotinas de acesso lógico, com poucas exceções (algumas operações no arquivo de índice secundários) se utilizam da rotina FIND para localizar / um registro e das informações em NVET e NDESC para operar sobre o registro. Por sua vez, a rotina FIND consulta o descritor do arquivo correspondente e a partir das informações obtidas utiliza as rotinas de acesso físico para localizar propriamente o registro.

Portanto, para adicionar um novo método de acesso e provavelmente uma nova estrutura de armazenamento ao sistema é suficiente estender a rotina FIND para aceitar esse novo método. Esta flexibilidade, justifica em parte, não terem sido implementadas até o momento as outras estruturas propostas.

Muitas características desejáveis, tais como, acesso concorrente, e alguns aspectos de segurança que dependem de certas ferramentas não existentes no sistema operacional hospedeiro, ficaram obviamente fora deste projeto inicial.

REFERÊNCIAS

- [1] - Silva, C. S. A. - "REQUEL: Uma linguagem de Consultas para o Sistema REDAS", Tese de Mestrado, ITA - 1977
- [2] - Sundgren, B. - "An Infological Approach to Data Bases" Phd Thesis URVAL, No. 7
- [3] - Gomes, A. F. C. - "O Modelo de Dados do Sistema REDAS", Tese de Mestrado, ITA - 1977
- [4] - Silva, F. A., - "Um Subsistema de Segurança e Integridade para o Sistema REDAS", Tese de Mestrado - ITA
(em desenvolvimento)
- [5] - ANSI/X3/SPARC - "Interim Report ANSI/X3/SPARC Study Group on Data Base Management Systems".