

## AVALIAÇÃO DOS PROCESSOS DE AUTOMAÇÃO EM BIBLIOTECAS UNIVERSITÁRIAS

Luiz Fernando Sayão \*

Carlos Henrique Marcondes \*

Carlos Cesar Fernandes \*\*

Ligia Polycarpo M. Medeiros \*\*\*

### RESUMO

Apresenta-se análise dos resultados do projeto "Avaliação de Processos de Automação em Bibliotecas Universitárias Brasileiras" (PNBU/CNPq). A análise diz respeito ao perfil dos softwares desenvolvidos e/ou utilizados pelas IESs brasileiras nos processos de automação de suas bibliotecas. Analisam-se os resultados principalmente sob dois aspectos: a) adequação das ferramentas de software utilizadas para o desenvolvimento de sistemas de automação de bibliotecas; b) porte/complexidade de um projeto de desenvolvimento de um software bibliográfico. Relacionam-se os requisitos técnicos desejáveis em um software bibliográfico e avalia-se até que ponto as ferramentas de software utilizadas nos casos analisados podem atender estes requisitos. Apresentam-se também estatísticas dos dados apurados. Finalmente propõe-se o desenvolvimento de um software padrão para a automação das bibliotecas das IESs brasileiras.

**Unitermos:** *Bibliotecas Universitárias – Automação.*

### 1. INTRODUÇÃO

Este é o primeiro Relatório do Projeto de Pesquisa "Avaliação dos Processos de Automação em Bibliotecas Universitárias Brasileiras", do PNBU/SESu, com o apoio do CNPq/CAPES/FINEP. O Relatório se concentra exclusivamente na parte relativa aos software existentes nas IESs, tendo como alvo a busca de softwares "portáteis", que pudessem ser repassados a outras IESs e, eventualmente, tornarem-se um padrão nacional.

O instrumento da pesquisa foi um questionário, distribuído à quase totalidade das bibliotecas das IESs brasileiras no segundo semestre de 1988, onde se procurava coletar dados que permitissem uma avaliação glo-

\* IBICT/UFRJ.

\*\* CIN/CNEN.

\*\*\* PRÓ-INFO.

bal dos processos de automação das bibliotecas universitárias brasileiras, concluídos, em andamento ou simplesmente em planejamento. Como sub-produto deste trabalho, uma parte do questionário, denominada "Formulário de Descrição de Software", procurava coletar dados sobre os softwares existentes, suas características técnicas, sua funcionalidade, a possibilidade de serem repassados a outras IESs, etc. Estes dados vieram a constituir a Base de Dados "Guia de Software de Automação de Bibliotecas", do PNUB, no sentido de disseminar o parque de softwares da IESs e permitir sua reutilização. São estes os dados, o objeto deste relatório.

Os resultados da pesquisa mostram um esforço significativo de várias bibliotecas de IESs no sentido de desenvolverem softwares bibliográficos, visando automatizar o registro e processamento de informações bibliográficas. No nosso entender, no entanto, estas iniciativas podem e devem ser criticadas, já que, por um lado, o esforço de automatizar os acervos das bibliotecas universitárias do país necessariamente resultará em maiores facilidades de acesso e intercâmbio de informações bibliográficas, com reflexos imediatos no desenvolvimento da ciência e tecnologia no Brasil, e, por outro lado, este é um processo recém iniciado, o que permite redirecionar as iniciativas e evitar alguns equívocos. Nossas críticas se desdobram em dois aspectos: as ferramentas de software utilizadas e a noção de porte/complexidade de um projeto como este.

O Relatório está organizado da seguinte maneira: a parte 1 apresenta uma estatística dos resultados do questionário, destacando alguns pontos importantes para a portabilidade do software, tais como a linguagem com a qual foi desenvolvido, se o software dispõe ou não de uma interface em formato de intercâmbio bibliográfico, quais os pacotes customizados (especiais para aplicações bibliográficas). A parte 2 é uma avaliação destes resultados à luz de dois parâmetros: as ferramentas de software (linguagens, pacotes) utilizadas no seu desenvolvimento e as dimensões de um projeto de desenvolvimento de um software bibliográfico para uma biblioteca de uma IES. A parte 3 apresenta algumas conclusões e a parte 4 procura delinear algumas propostas. Seguem-se Notas, Bibliografia e um Anexo, onde está delineada a proposta de um software bibliográfico padrão para as bibliotecas das IESs brasileiras.

### 1.1. DADOS APURADOS A PARTIR DO QUESTIONÁRIO

A maioria dos softwares pesquisados (48,21%) estavam desenvolvidos, utilizando ferramentas do tipo SISTEMAS GERENCIADORES DE BANCOS DE DADOS (SGBD)/LINGUAGENS DE 4ª GERAÇÃO (L4G) comerciais (FIG. 1). Uma percentagem de 28,5% utiliza linguagens de 3ª Geração; 16,10% utilizam pacotes (a grande maioria deste grupo usa o ISIS – micro e mini, e 1 o STAIRS).



Dos softwares pesquisados, somente 7 diferentes do ISIS possuem algum tipo de interface padrão e destes apenas 4 possuem interface em formato de intercâmbio bibliográfico – 3 com IBICT, sendo 1 ainda em desenvolvimento, e 1 com MARC (FIG. 2).

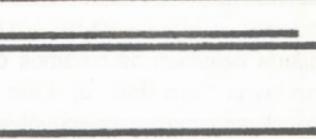
## 2. AVALIAÇÃO

### 2.1. FERRAMENTAS UTILIZADAS NO DESENVOLVIMENTO DESTES SOFTWARES

É significativo o fato de que se a maioria dos softwares de automação, identificados pela pesquisa, esteja desenvolvido, utilizando as facilidades providas por Sistemas Gerenciadores de Bancos de Dados/Linguagens de 4ª Geração comerciais: os exemplos vão desde o SUPRA, passando pelo DMS II/LINK, até o DBASE III/CLIPPER. Parece claro que a matriz de automação de bibliotecas de IESs passa pelo uso destas ferramentas; este fato têm implicações importantes, tanto do ponto de vista do software em si, das características técnicas/funcionalidades desejáveis em um software bibliográfico, como do ponto de vista de uma integração sistêmica das bibliotecas das IESs brasileiras. Com o objetivo de compreender melhor esta situação, neste item passaremos a discutir a) as características gerais destas ferramentas e b) sua adequação ao desenvolvimento de software bibliográfico.

A maioria dos SGBDs/L4Gs está baseada no famoso MODELO RELACIONAL, desenvolvido por E. F. Codd nos laboratórios da IBM em Santa Mônica, Califórnia, no início da década de 70. Um modelo de dados, como o MODELO RELACIONAL, é uma forma de estruturar informações e seus relacionamentos para armazená-las em computador. Uma das motivações mais fortes para o desenvolvimento do MODELO RELACIONAL era implementar o conceito de "independência de dados", visando prover o usuário de Banco de Dados de um formalismo de estruturação dos dados que o liberasse de preocupações acerca da forma e dos mecanismos de como os dados eram armazenados e relacionados fisicamente na memória de massa do computador, ou seja, que separasse a estrutura lógica dos dados, a maneira como o usuário os enxergava, da sua estruturação física no computador. Isso representava um grande avanço na tecnologia dos sistemas de Bancos de Dados, já que nos sistemas anteriores o usuário, fosse ele um programador ou um usuário final, que estivesse interessado somente em fazer consultas esporádicas ao BD, tinha que conhecer detalhes da estruturação física dos dados, como apontadores, hierarquias, disposições dos dados no meio de armazenamento, etc., para poder manipulá-los; a manipulação dos dados ficava condicionada por características do seu armazenamento físico, o que obrigava o usuário a levar em conta detalhes de implementação totalmente irrelevante para a sua aplicação.

FIG. 2 - INTERFACE PADRÃO

%	7,1	1,8	1,8	1,8	1,8	1,8	12,5	73,2
Nº	4	1	1	1	1	1	7	41
								

No MODELO RELACIONAL proposto por Codd, um Banco de Dados é representado simplesmente por uma RELAÇÃO, que pode ser visualizada como uma TABELA em que cada linha constitui uma entidade ou registro (chamada "tupla") e cujas colunas representam atributos ou campos destas entidades. A representação em forma de tabela implicava em que todos os registros tivessem os mesmos campos, com os mesmos tamanhos, ou seja, um lay-out fixo (FIG. 3). Esta característica vai ter sérias implicações quando utilizada para o desenvolvimento de sistemas bibliográficos, como veremos adiante. (Para uma definição formal do MODELO RELACIONAL, veja NOTA 1).

O MODELO RELACIONAL provê ainda o usuário de formalismos matemáticos de manipulação, a ÁLGEBRA RELACIONAL e o CÁLCULO RELACIONAL, baseados na teoria dos conjuntos e na teoria das relações, através dos quais o usuário simplesmente especifica operações com tabelas cujos resultados também são tabelas, ou especifica as características da tabela desejada, sem se preocupar em especificar um procedimento ou algoritmo para efetuar as operações ou obter as tabelas desejadas. Este tipo de manipulação chama-se "não-procedural", em oposição às linguagens de manipulação algorítmicas, também chamadas linguagens de 3ª GERAÇÃO, como COBOL, PASCAL, "C", etc., em que tem-se que especificar o procedimento (algoritmo) para realizar a função desejada, em oposição a especificar-se somente o "resultado" desejado. O CÁLCULO RELACIONAL e a ÁLGEBRA RELACIONAL foram o ponto de partida do desenvolvimento das chamadas Linguagens de 4ª Geração, chamadas também "Linguagens Não-Procedurais". É como se o usuário estivesse um nível "acima" do nível de interação com o sistema de um programador, que estivesse desenvolvendo uma aplicação utilizando uma linguagem de 3ª Geração. Isto retrata uma tendência geral no desenvolvimento de linguagens de programação, desde o ASSEMBLER (interagindo diretamente com o hardware), passando pelas linguagens de 3ª Geração (algorítmicas, procedurais), até as Linguagens de 4ª Geração (não-procedurais), de afastar o usuário de detalhes de máquina, de como realizar suas aplicações, permitindo-lhe preocupar-se somente em "declarar" suas aplicações e concentrar-se em especificar "o que" o sistema deve fazer (FIG. 4).

Embutido no MODELO RELACIONAL vinha também uma chamada "teoria de normalização", ou seja, de como projetar relações ditas "normalizadas", de modo a garantir ao máximo a independência de dados. As operações de normalização visavam produzir lay-outs de registros que, em última instância, garantissem que cada relação representasse uma e somente uma entidade do mundo real. O objetivo com isto era evitar custosas atualizações, tão frequentes em um ambiente de aplicações comerciais, de diversos registros (tuplas) onde uma mesma entidade estava representada, como no caso de se representar um periódico e seu fornecedor em uma mesma

FIG. 3 - TABELA "REFERÊNCIAS"

## REFERÊNCIAS

IBBN	TÍTULO	AUTOR	VOL	NUM	EDITOR	ENDED	FORNEC
23832	MICROISIS NA BIBLIOTECA	MACEDO, L.F.	18	1	PRO-INFO	RIO	J. CABRAL
5489357	CONSTITUIÇÃO DE TERAUDO	ESPANHA, N.	3	5	UFF	MITERÓI	DIB CAPITAL
348494	LINCE; PADRÃO NACIONAL	FERNANDES, C.C.	12	15	OROV/CIN	RIO	P&B
997568	POLÍTICA DE AQUISIÇÃO	QUELIDOS, G.	3	2	IM	S. PAULO	PALADIUM

FIG. 4 - OPERAÇÃO EM ÁLGEBRA RELACIONAL

## SELECCIONE referências tal que ENDED=RIO DE JANEIRO

## ENDED=RIO

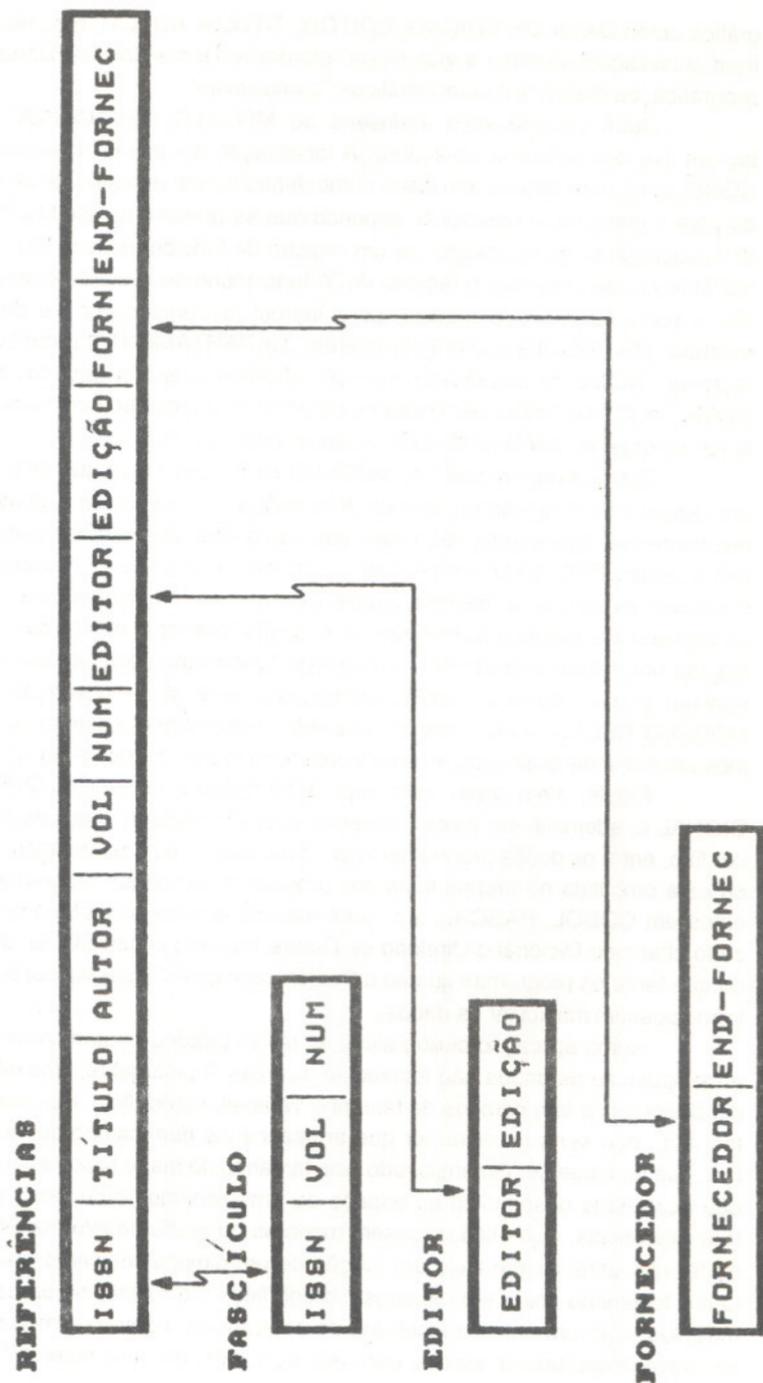
IBBN	TÍTULO	AUTOR	VOL	NUM	EDITOR	ENDED	FORNEC
1 23832	MICROISIS NA BIBLIOTECA	MACEDO, L.F.	18	1	PRO-INFO	RIO	J. CABRAL
2 348494	LINCE; PADRÃO NACIONAL	FERNANDES, C.C.	12	15	OROV/CIN	RIO	P&B

relação; se o endereço do fornecedor for alterado, vão ter que ser alterados todos os registros correspondentes a periódicos fornecidos por este fornecedor; dependendo do número de periódicos fornecidos pelo fornecedor que teve seu endereço alterado, esta operação pode se tornar bastante cara em termos de custo de processamento; a teoria de normalização manda que "se removam as dependências funcionais", ou seja, que se represente cada entidade em uma relação distinta: uma relação para periódicos e outra relação distinta para fornecedores (FIG. 5).

O uso de relações não normalizadas pode acarretar também as chamadas "anomalias" de inserção ou remoção de registros: se os dados de um fornecedor estão representados na mesma relação que os pedidos de fornecimento de um periódico, só se pode registrar os dados de um fornecedor quando houver um pedido de fornecimento ao mesmo. Da mesma forma, ao se remover do Bando de Dados um pedido de fornecimento, se só houver este pedido a um dado fornecedor, perdem-se os dados deste fornecedor ao se removerem os dados do pedido. Tudo isto porque fornecedor constitui uma entidade e tem existência independente de seus pedidos de fornecimento.

Os conceitos embutidos no MODELO RELACIONAL, como "independência de dados", formalismos de manipulação do tipo "Linguagens não-procedurais", ênfase no projeto prévio de relações visando otimizar as frequentes atualizações, tão comuns em um ambiente de aplicações comerciais (normalização), significaram um grande avanço em termos de tecnologia de Bancos de Dados, tornando esta tecnologia mais fácil de ser operada tanto pelo usuário especializado, o programador que vai desenvolver uma aplicação usando as facilidades providas pelo SGBD/L4G, como para o usuário final, aquele que vai consultar ou atualizar o BD sem precisar construir um programa para isso, que pode, através destes formalismos, manipular diretamente o BD, sem se preocupar em "COMO" (procedimentos computacionais, algoritmos) o sistema fará para realizá-los. Além disso, o MODELO RELACIONAL permitiu que o tempo gasto no desenvolvimento de novas aplicações fosse consideravelmente reduzido, provendo um ganho de produtividade para analistas e programadores. O MODELO RELACIONAL influenciou o desenvolvimento de toda uma geração de produtos de software, que vão desde o DBSE III, o DATAFLEX, o PARADOX, para micros, até o DB2 (IBM), SUPRA (CINCON), ORACLE, etc., para mainframes. No entanto, estas características, tão importantes num ambiente de BD comercial, não contemplam facilidades que seriam desejáveis num ambiente de aplicações bibliográficas. Em primeiro lugar, num ambiente de BD comercial, os dados são muito mais voláteis, ou seja, sofrem atualizações constantes, daí as técnicas de normalização visando justamente otimizar as atualizações. Num ambiente bibliográfico, os dados são muito menos voláteis, sofrendo nenhuma ou pouquíssimas atualizações: os dados de uma referência biblio-

FIG. 5 - NORMALIZAÇÃO DA TABELA "REFERÊNCIAS"



gráfica como DATA DE EDIÇÃO, EDITOR, TÍTULO, AUTOR, etc., não sofrem atualizações durante a vida desta referência. Numa Base de Dados bibliográfica, os dados têm características "cumulativas".

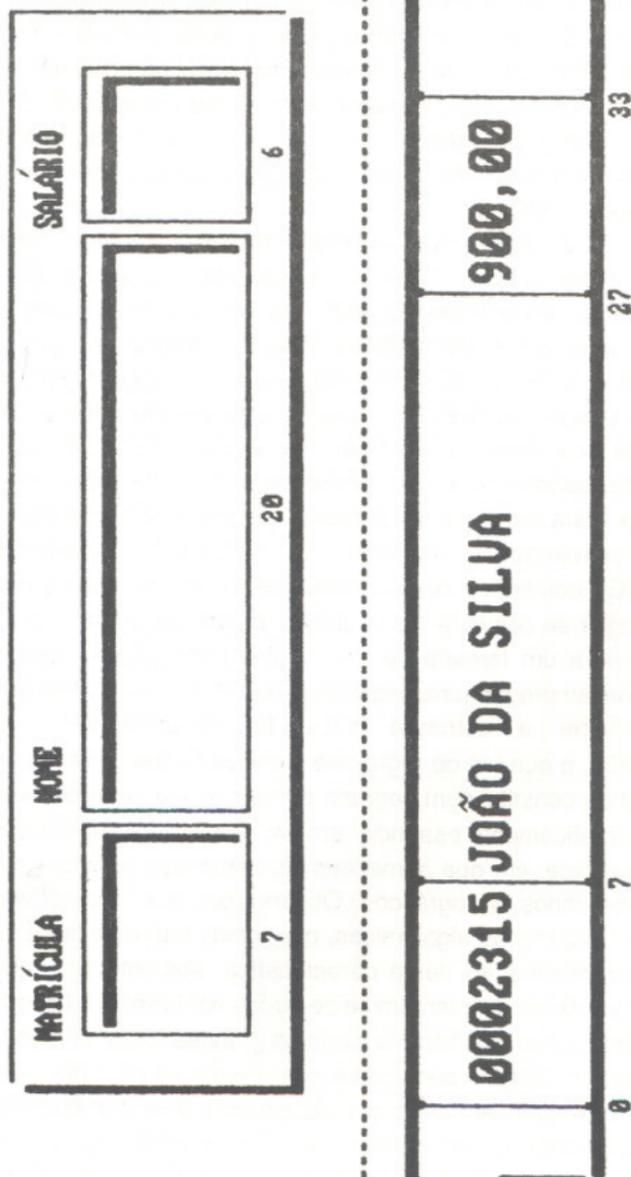
Outra característica intrínseca ao MODELO RELACIONAL é o lay-out fixo dos registros (relações). A localização de uma informação pelo SGBD, tanto num arquivo em disco como dentro de um registro que foi trazido para a memória, é posicional: supondo que se queira o campo NOME do 3º funcionário de um cadastro; se um registro de funcionário tem 100 bytes (caracteres) de tamanho, o registro do 3º funcionário se iniciará no carácter 201 a partir do início do arquivo; se o lay-out do registro constar de, por exemplo, MATRÍCULA (com 5 caracteres), DEPARTAMENTO (com 20 caracteres), NOME do funcionário (com 30 caracteres), etc., o valor do campo NOME do 3º funcionário se iniciará no carácter 26 do registro correspondente (ou do carácter 226 a partir do início do arquivo) – FIG. 6.

Outra característica do MODELO RELACIONAL é que se houver um campo para o projeto em que um funcionário esteja alocado e momentaneamente este funcionário não esteja alocado a nenhum projeto, mesmo assim o campo PROJETO fica previsto no lay-out do registro do funcionário e é previsto espaço de armazenamento para o mesmo; neste caso este espaço tem que ser preenchido com um valor igual a "brancos" ou "zeros", significando um projeto inexistente, ou que este funcionário não está alocado a nenhum projeto; ou seja, somos obrigados a usar uma convenção extra MODELO RELACIONAL, porque para este, estritamente, neste caso teríamos um funcionário alocado ao projeto identificado por "brancos" ou "zeros".

Existe, além disso, uma separação rigorosa no MODELO RELACIONAL e, ademais, em todos os sistemas que manipulam arquivos de lay-out fixo, entre os dados propriamente ditos e a descrição dos mesmos. Esta, ou está embutida no próprio texto dos programas, como em programas escritos em COBOL, PASCAL, etc., ou é mantida em um depósito centralizado chamado Dicionário/Diretório de Dados, no caso de um SGBD, de modo que tanto os programas quanto o SGBD "conheçam" este lay-out e desta forma possam manipular os dados.

Numa aplicação bibliográfica, as necessidades de armazenamento e manipulação de dados são totalmente distintas. Típicamente, uma referência bibliográfica tem campos de tamanho variável, como título, resumos, notas, etc.; não seria razoável ter que armazená-los num campo de tamanho fixo, que teria que ser dimensionado pelo tamanho do maior título esperado, o que acarretaria desperdício no espaço de armazenamento de todas as outras referências, cujos títulos fossem menores ou perda de informações por se ter que abreviar um título em função de um dimensionamento mal feito. Outro fenômeno típico em registros bibliográficos é a existência de campos múltiplos, com um número indefinido de ocorrências, como o campo de autor: deveríamos prever espaço para um, dois, três, ou mais autores? Outra

FIG. 6 - LAY-OUT PARA UM REGISTRO DE TAMANHO FIXO



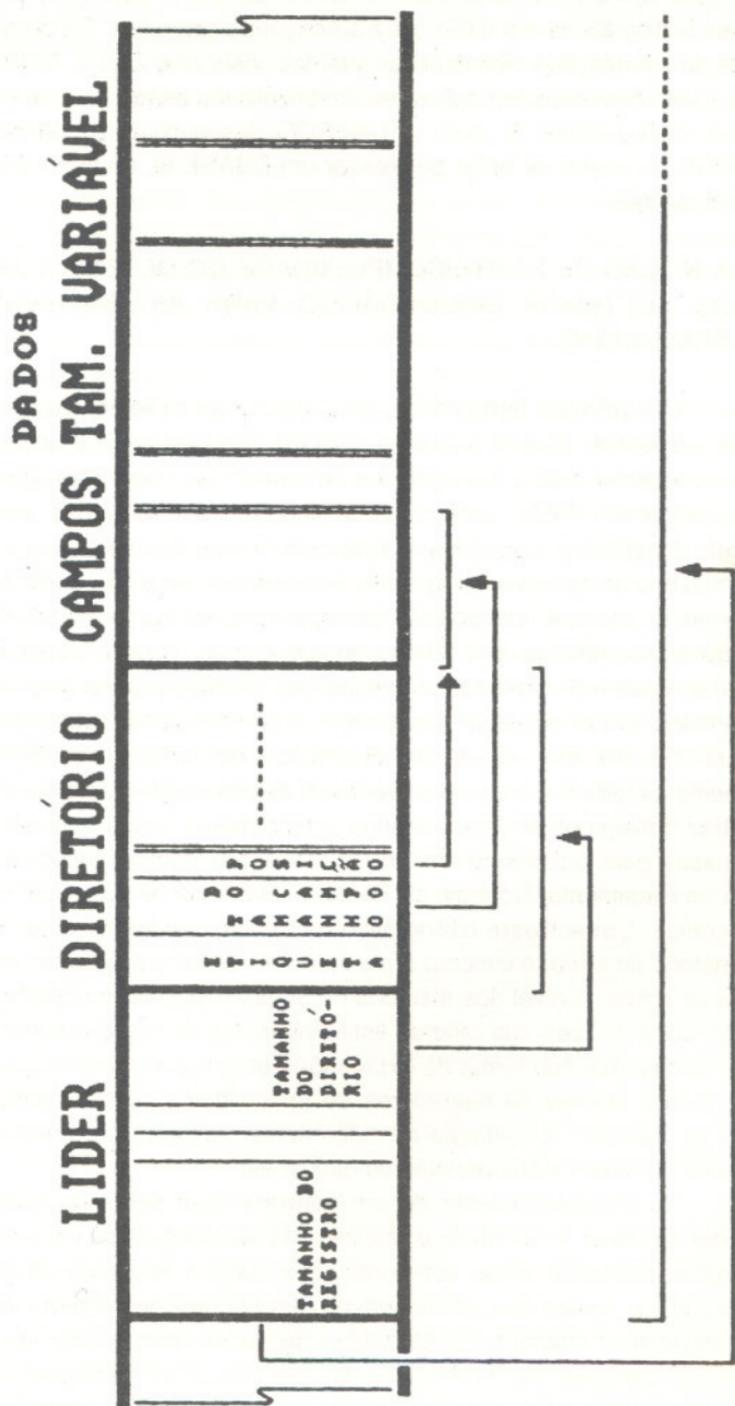
característica típica de aplicações bibliográficas é a existência de campos opcionais, aqueles cuja existência na descrição de uma referência não é obrigatória, como por exemplo, tradutor (nem todas as referências são traduzidas), ilustrador, notas para casos especiais, dados de uma conferência, etc .

Na verdade, em uma aplicação bibliográfica temos tipicamente campos de tamanho variável, campos múltiplos, campos opcionais, o que implica em praticamente um lay-out único e individual para cada referência. Daí a impossibilidade de manter este lay-out (que não é comum) centralizado. Um lay-out de um registro bibliográfico contém, além dos dados propriamente ditos, informações que permitem processar os próprios dados de cada registro de uma referência bibliográfica, como identificadores de campos (conhecidos como "parágrafos" ou "etiquetas"), tamanhos de cada campo, número de ocorrências de campos múltiplos, indicadores, separadores de subcampos, etc. (FIG. 7).

Isso não é nenhuma novidade, basta consultar o manual de um formato bibliográfico típico. Todas estas características não são praticamente atendidas por um SGBD/L4G comercial, embora sejam essenciais para o tratamento de registros bibliográficos. Vinculado também às características e limitações dos SGBDs/L4Gs comerciais, em que um grande número de aplicações em bibliotecas das IESs brasileiras foi desenvolvido, está o fato significativo de que todos os sistemas que empregaram esta tecnologia não dispõem de facilidades de importação/exportação de dados em formato de intercâmbio. Esta tarefa se torna bastante complexa se os dados já se encontram estruturados segundo o MODELO RELACIONAL. Os SGBDs/L4Gs comerciais não oferecem nenhuma facilidade a este respeito. Para conseguir-se converter os dados do formato de armazenamento interno do SGBD para um formato de intercâmbio bibliográfico, necessariamente ter-se-ia que empregar uma linguagem de 3ª Geração algorítmica e além disso conhecer a estrutura física de armazenamento interno dos SGBDs/L4Gs, o que via de regra não é colocado disponível aos seus usuários porque se constitui num segredo comercial. No entanto, esta é uma característica praticamente essencial em um software bibliográfico destinado a uma universidade, em que aumentam as demandas para um crescente intercâmbio de dados bibliográficos. Observamos que, dos softwares desenvolvidos em linguagens algorítmicas, que, pelos fatores citados acima facilitariam a implementação desta característica, somente um deles declarou dispor de facilidades de intercâmbio de dados em formato bibliográfico.

Outro dado significativo é que os grandes fornecedores de software não confundem SGBDs comerciais com Sistemas de Recuperação de Informações Bibliográficas; cada qual desenvolve produtos distintos para aplicações comerciais e para aplicações bibliográficas. Por exemplo, a IBM possui um produto: o STAIRS, para aplicações bibliográficas e o DB2 e o

FIG. 7 - LAY-OUT PARA UM REGISTRO BIBLIOGRÁFICO



SQL para aplicações comerciais; a BULL possui o MISTRAL para aplicações bibliográficas e o IDS-II para aplicações comerciais. Os grandes sistemas de informações bibliográficas internacionais como OCLC, ORBIT, DIALOG, estão baseados em softwares desenvolvidos especialmente para aplicações bibliográficas. E mais, a UNESCO desenvolveu o conhecido MICROISIS, ao invés de optar por utilizar um DBASE III, ou outro SGBD comercial qualquer.

## 2.2 . A NOÇÃO DE PORTE/COMPLEXIDADE DO DESENVOLVIMENTO DE SOFTWARE BIBLIOGRÁFICO PARA AS UNIVERSIDADES BRASILEIRAS

Um software bibliográfico, para atender às bibliotecas das universidades brasileiras, deveria ter como requisito essencial a capacidade de importar e exportar dados bibliográficos no formato de intercâmbio padrão nacional, o formato IBICT. Uma característica não essencial, mas também altamente desejável por suas implicações em termos de economia de espaço de armazenamento, seria a capacidade de armazenar e manipular registros de tamanho variável, campos de tamanho variável, campos múltiplos com número de ocorrências variável e campos opcionais. Deveria também permitir que uma série de produtos pudessem ser gerados a partir dos dados armazenados, como catálogos (impressos e on-line), fichas, etiquetas, serviços de DSI, BR, etc. . O uso de ferramentas de software tipo SGBD/L4G realmente simplifica e barateia o processo de desenvolvimento de software, mas traz consigo os prejuízos citados anteriormente; estas ferramentas são adequadas para o desenvolvimento de aplicações comerciais típicas, como Folha de Pagamento, Controle de Estoque, etc., mas não para um software bibliográfico. Um software bibliográfico tem então que implementar seu próprio método de armazenamento e manipulação capaz de suportar as características acima. A nível dos métodos de acesso/busca são desejáveis facilidades como índices que utilizam listas invertidas de modo a suportar consultas formuladas sob forma de expressões em Lógica Booleana, busca em texto, busca através de operadores de proximidade, índices multicampos, lógica de patamar, ordenação de referências, segundo um critério de relevância, técnicas de compressão de dados, etc. .

No desenvolvimento de um software com estas características, torna-se da maior importância o domínio das técnicas e dos algoritmos que possam implementar estas estruturas, que, dado o seu nível de detalhe e complexidade, teriam que ser desenvolvidas em uma linguagem que permitisse especificar algoritmos e estruturas de dados complexas, uma linguagem de 3ª Geração como PASCAL, "C", ALGOL, PL-1 ou mesmo COBOL. Um projeto como este compara-se em complexidade praticamente ao desenvolvimento de qualquer software básico, como são os Sistemas Opera-

cionais, Compiladores, Editores de Texto e mesmo SGBDs comerciais (FIG. 8), ou seja, trata-se de um projeto de grande porte, caro, demorado, que exige pessoal altamente especializado.

### 3. CONCLUSÕES

Não negamos as facilidades providas pela tecnologia de SGBD/L4G comerciais no desenvolvimento de uma série de pequenas aplicações bibliográficas, realmente úteis e interessantes em várias bibliotecas de IESs pelo Brasil afora. So queremos deixar claro os limites desta tecnologia, quando se pretende produtos mais ambiciosos. Neste sentido, muitos dos softwares, analisados na pesquisa e constantes da base de dados "GUIA DE SOFTWARE DE AUTOMAÇÃO DE BIBLIOTECAS", não podem ser enquadrados como softwares bibliográficos completos, por não abrange-rem as funções básicas de uma biblioteca. Um indicativo deste fato é que muitos deles rodam em micros, com pequena capacidade de disco para conter todo o acervo de uma biblioteca; um número significativo deles dá suporte a funções administrativas; neste caso, é razoável que eles tenham sido desenvolvidos utilizando-se a tecnologia dos SGBDs/L4Gs comerciais, devido aos ganhos de produtividade e facilidades providas por estas ferramentas. No entanto, o desenvolvimento de um software bibliográfico completo, que dê suporte às funções básicas de catalogação e recuperação de dados bibliográficos e que permita ainda o intercâmbio destes dados em formato padrão, é uma questão ainda por ser resolvida.

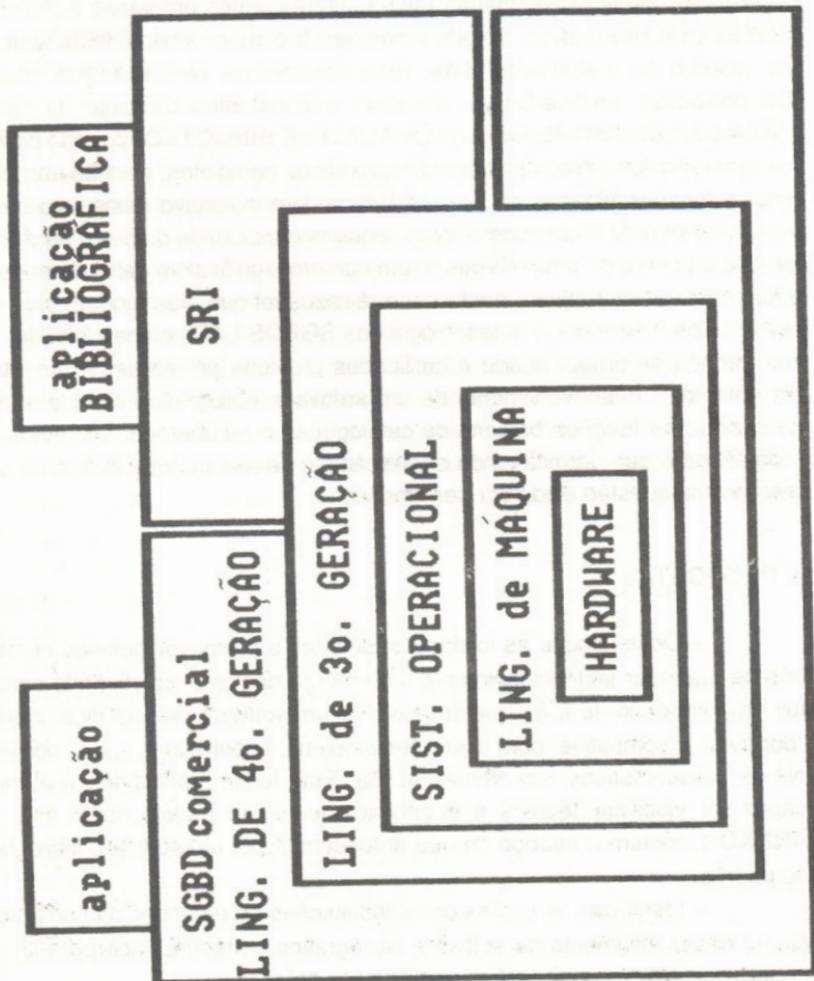
### 4. PROPOSTAS

– Desestimular as iniciativas isoladas e promover gestões no sentido de viabilizar institucionalmente um esforço nacional, objetivando constituir um consórcio de IESs que desenvolvesse um software bibliográfico padrão, "portável" e compatível com diferentes ambientes operacionais, que contemple as características ressaltadas acima. Esta forma institucional é a única capaz de viabilizar técnica e economicamente um projeto como este. O ANEXO 1 contém o esboço do que entendemos por um software bibliográfico padrão.

– Identificar os profissionais/instituições de reconhecida competência no desenvolvimento de software bibliográfico e procurar incorporá-los ao projeto de software bibliográfico padrão nacional.

– Apoiar a criação de linhas de pesquisa em sistemas bibliográficos a nível de pós-graduação – Mestrado/Doutorado, em Informática, a exemplo do que existia no Programa de Pós-Graduação em Engenharia de Sistemas do Instituto Militar de Engenharia – IME/RJ.

FIG. 8 - NÍVEIS DE SOFTWARE



– Promover cursos de aperfeiçoamento para profissionais de informática em desenvolvimento de sistemas bibliográficos e cursos de gerência de processos de automação para profissionais bibliotecários.

## NOTAS

[01] O MODELO RELACIONAL baseia-se no conceito matemático de "relação": toma-se uma coleção de conjuntos, não necessariamente distintos  $D_1, D_2, D_3$ , etc., e seu produto cartesiano  $C = D_1 \times D_2 \times D_3$ , etc. Formam-se desta maneira todos os possíveis conjuntos  $R$  formados por elementos  $d_1, d_2, d_3$ , etc., tais que  $d_1$  pertence ao domínio  $D_1$ ,  $d_2$  a  $D_2$ ,  $d_3$  a  $D_3$ , e assim por diante; a função que seleciona elementos  $d_n$  de um domínio  $D_n$  para formar a relação  $R$  chama-se ATRIBUTO e representa o "uso" dos elementos do domínio  $D_n$  na relação  $R$ . Qualquer subconjunto de  $C$  com estas características é uma Relação.

## BIBLIOGRAFIA

- CASWELL, Jerry V. Normalization: a method for structured file design **Information Technology and Libraries**. 3(3):293-6, September 1984.
- CRAWFORD, Richard. The Relational Model in Information Retrieval. **JASIS**. 32(1): 51-64, January, 1981.
- DATAPRO Reserch Corp. Linguagens de Quarta Geração – Um Modelo Funcional. **MIS – Relatório de Gerenciamento da Informação**. 10:5-24, novembro 1988
- DATE, C J. **Introdução a sistemas de Bancos de Dados**. Ed. Campus, 1986. 513 p.
- INSTITUTO BRASILEIRO DE INFORMAÇÃO EM CIÊNCIA E TECNOLOGIA. **Formato IBICT**: formato de intercâmbio bibliográfico e catalográfico. Brasília, IBICT, 1987 400 p.
- KOENIG, M. E. D. Data relationships: bibliographic information Retrieval Systems and Database Management Systems. **Information Technology and Libraries**. 4(3):247-72, September 1985.
- POLLARD, Richard. Bibliographic Data Management with Dbase: a Study of Secondary Key Retrieval on Multivalued Data Items. **Information Technology and Libraries**. 7(1):56-66, March 1988.

## ABSTRACT

SAYÃO, L.F., MARCONDES, C.H., FERNANDES, C.C., MEDEIROS, L.P.M. Evaluation of Automation Process in Brazilian University Libraries. **Trans-in- formação**, 1(2), maio/ago 1989.

An analysis of the results of the project "Evaluation of Automation Process in Brazilian University Libraries" (PNBU/CNPq) is presented. The analysis relates to the design of software developed and/or used by Brazilian Universities in the process of library automation. The results are analysed under the aspects (a) adequacy of software tools used for the development of systems, and (b) development of bibliographical software. Statistical data are presented and the development of a model for Brazilian Universities Libraries is also introduced.

## ANEXO 1

# PROPOSTA DE UM SOFTWARE PADRÃO DE MANIPULAÇÃO DE REGISTROS BIBLIOGRÁFICOS

### 1. INTRODUÇÃO

A padronização de software bibliográfico no Brasil se insere na questão mais ampla de racionalização e otimização dos poucos recursos disponíveis no país para que o mesmo consiga superar a barreira do subdesenvolvimento. A automação das bibliotecas universitárias é um meio de tornar disponíveis à comunidade de C&T do país mais rapidamente informações essenciais a este esforço. Ela trará também benefícios paralelos que se inserem na mesma prioridade, como a melhor utilização do acervo, a racionalização das aquisições e a otimização do uso do material bibliográfico.

No entanto, a questão é bastante complicada. O que significa um software "padrão"? Será que existe um padrão de funcionamento nas diferentes bibliotecas das IES brasileiras? Será que um software "padrão" não incorporará características que são específicas de uma biblioteca universitária, tornando-se uma camisa de força para outras instituições que venham a utilizá-lo? É indiscutível a necessidade de um software portátil e reutilizável, já que as bibliotecas universitárias dispõem de poucos recursos para desenvolverem softwares customizados para suas necessidades.

Nossa proposta, ainda um esboço bastante superficial, restrito às características funcionais do software, sem entrar em detalhes de algoritmos, mecanismos de armazenamento e demais características técnicas, etc., tenta incorporar estas preocupações, no sentido de produzir um software que seja portátil para vários equipamentos e ambientes operacionais e, ao mesmo tempo, não seja uma camisa de força para as instituições que venham a utilizá-lo, dando-lhes margem para que possam customizar aplicações específicas, de acordo com suas características e necessidades, tendo como suporte o software padrão.

### 2. CONCEPÇÃO

O software padrão seria composto de dois subsistemas distintos e autocontidos: um subsistema de entrada de dados rodando em microcomputador PC-compatível sob sistema operacional DOS, que geraria os dados bibliográficos em formato de intercâmbio. Além de entrada de dados, tal programa permitiria também a verificação e correção dos dados digitados.

O outro componente do sistema seria um subsistema de armazenamento e manipulação de registros bibliográficos, que permitiria a leitura de dados em formato de intercâmbio em disquetes gerados pelo primeiro subsistema e seu armazenamento em dispositivo de acesso direto (disco

magnético), bem como a recuperação dos registros bibliográficos e sua manipulação. Este subsistema teria a forma de uma biblioteca de rotinas de manipulação de registros bibliográficos, que estariam disponíveis para que se desenvolvessem aplicações mais ou menos sofisticadas em cima deste suporte básico. Desta forma, o subsistema de manipulação de registros bibliográficos ficaria livre de incorporar características específicas, políticas, normas, etc., que estariam embutidas nas aplicações construídas sobre o subsistema. Tal partido garantiria as especificidades das diferentes aplicações e forneceria um suporte básico à parte mais complexa de um software bibliográfico, que é a manipulação de registros/campos de tamanho variável, a manipulação de campos opcionais, a manipulação de campos múltiplos, compressão de dados, etc. .

Para garantir maior independência das aplicações a serem desenvolvidas do subsistema de manipulação de registros bibliográficos, a cada referência armazenada, o sistema atribuiria um número de referência (NREF) e proveria um índice que relacionaria os NREFs com os endereços físicos em disco de armazenamento dos registros bibliográficos. Desta forma, os registros bibliográficos seriam acessados através de um endereço simbólico, podendo ser reorganizados fisicamente sem que tal fato afete às aplicações construídas sobre o subsistema.

Como garantia maior de portabilidade, o subsistema de manipulação de registros bibliográficos seria escrito em linguagem de alto nível (possivelmente em "C") e, uma vez compilado, poderia rodar em qualquer ambiente operacional que dispusesse desta linguagem, como micros PC-DOS, supermicros PS/OS II, supermicros UNIX, superminis UNIX.

### 3. BIBLIOTECA DE ROTINAS DE MANIPULAÇÃO DE REGISTROS BIBLIOGRÁFICOS – DESCRIÇÃO.

#### 3.1. ARMAZENAMENTO

Os registros bibliográficos, de tamanho variável, serão armazenados em blocos físicos de um arquivo de acesso direto em disco magnético, provido de um único índice que associará NREFs a endereços físicos em disco dos registros.

#### 3.2. ROTINA DE LEITURA DE DISQUETES EM FORMATO DE INTERCÂMBIO

Lê um registro bibliográfico armazenado em disquete.

#### 3.3. ROTINA DE INCLUSÃO DE REGISTROS BIBLIOGRÁFICOS

Recebe um registro bibliográfico e o armazena no arquivo em disco, devolvendo seu NREF, através do qual o registro pode ser acessado.

### 3.4 . ROTINA DE EXCLUSÃO DE REGISTROS BIBLIOGRÁFICOS

Recebe um NREF e exclui o registro bibliográfico correspondente.

### 3.5 . ROTINA DE ALTERAÇÃO DE REGISTROS BIBLIOGRÁFICOS

Recebe um NREF correspondente a um registro bibliográfico, uma tabela com os parágrafos a serem alterados e os novos valores a serem assumidos, e altera o registro bibliográfico correspondente ao NREF, substituindo os valores dos parágrafos especificados.

### 3.6 . ROTINA DE RECUPERAÇÃO DE REGISTROS BIBLIOGRÁFICOS

Recebe um NREF correspondente a um registro bibliográfico e, opcionalmente, uma tabela com parágrafo do registro especificado e recupera toda a referência ou simplesmente os parágrafos especificados.

### 3.7. ROTINA DE ADMINISTRAÇÃO DO SISTEMA

Permite definir os parágrafos de um formato que vão ser lidos e os parágrafos que vão ser armazenados, além de monitorar o funcionamento do sistema.

### 3.8. ROTINAS UTILITÁRIAS

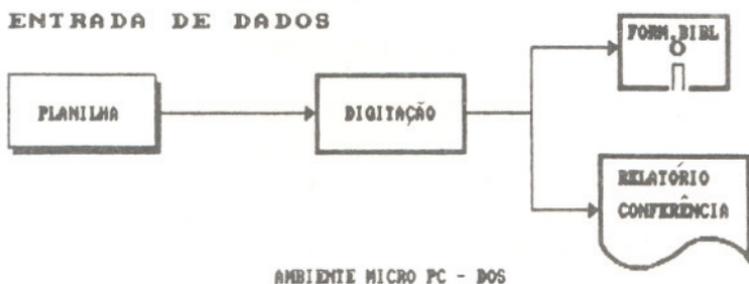
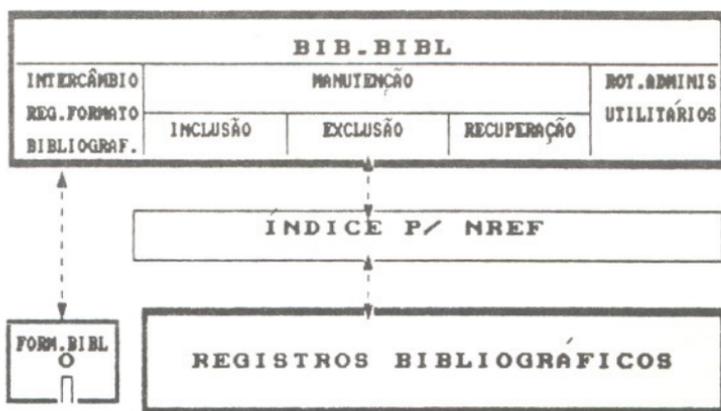
Realizam funções diversas, como estatísticas de disponibilidade de espaço no arquivo de referências bibliográficas, reorganização deste arquivo, copia/back-up do arquivo de referências bibliográficas, etc. .

## OBSERVAÇÃO

A figura 1 procura ilustrar os elementos do Sistema de Manipulação de registros Bibliográficos: o subsistema de entrada de dados e a biblioteca de rotinas bibliográficas. As figuras 2 e 3 procuram ilustrar exemplos de aplicações bibliográficas que poderiam ser desenvolvidas sobre a biblioteca de rotinas bibliográficas, no caso a emissão de catálogos ou fichas (figura 2) e um sistema disseminação seletiva de informações (figura 3).

Esperamos que a nossa proposta ajude a esclarecer dúvidas e auxilie o PNBu na formulação de políticas para a automação das bibliotecas universitárias brasileiras.

FIG. 1 (anexo) – BIBLIOTECA DE ROTINAS BIBLIOGRÁFICAS

**BIBLIOTECA DE ROTINAS BIBLIOGRÁFICAS**

AMBIENTE MICRO PC-DOS/SUPERMICRO, SUPERMINI - UNIX

FIG.1 - BIBLIOTECA DE ROTINAS BIBLIOGRAFICAS