

71

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

preto
c/lista

88/202

INTERFACE DE USUÁRIO
PARA A
MANIPULAÇÃO DE
FORMULÁRIOS ELETRÔNICOS

por

DUNCAN DUBUGRAS ALCOBA RUIZ



Dissertação submetida como requisito parcial para
a obtenção do grau de Mestre em
Ciência da Computação

[Handwritten Signature]
Prof. José Palazzo Moreira de Oliveira
Orientador

Porto Alegre, junho de 1987.

UFRGS
BIBLIOTECA
CPD/PGC

CATALOGAÇÃO NA FONTE

+-----+
|
| Ruiz, Duncan Dubugras Alcoba |
| Interface de Usuário para Manipulação de |
| Formulários Eletrônicos. Porto Alegre, |
| PGCC da UFRGS, 1987. |
| 1 v. |
| Diss. (mestr. ci. comp.) UFRGS - PGCC, |
| Porto Alegre, BR-RS, 1987. |
| |
| Dissertação: |
| Automação de Escritórios, Interface de |
| Usuário, Formulários Eletrônicos, Objetos |
| |
+-----+

à Lourdes,
minha mulher

AGRADECIMENTOS

Ao Centro de Processamento de Dados e ao Curso de Pós-Graduação em Ciência da Computação, pelo apoio institucional.

Ao professor José Palazzo Moreira de Oliveira, pela competente orientação e ilimitada paciência.

Ao amigo e companheiro Sergio Alexandre "KF" Korndörfer, pelas acaloradas discussões e valiosas trocas de idéias.

SUMARIO

LISTAS DE ABREVIATURAS	07
LISTAS DE FIGURAS	08
RESUMO	09
ABSTRACT	10
1 INTRODUÇÃO	11
1.1 A Descrição da Realidade	12
1.2 Bancos de Dados orientados para Objetos ...	13
2 FORMULÁRIOS ELETRÔNICOS	19
2.1 O Sistema de Formulários Eletrônicos	20
2.1.1 A Estrutura em Níveis	20
2.1.2 O Modelo Sintático de Formulários	23
2.1.3 Arquitetura do Sistema de Formulários Eletrônicos	25
2.1.4 A Concepção do Sistema	27
2.1.5 A Interface de Manipulação	28
2.2 A Definição Sintática de Formulários	28
2.2.1 Variável Formulário	29
2.2.2 Tipo Formulário	30
2.2.3 Definição da Estrutura do Formulário	31
2.2.4 A Extensão da Definição de Tipos de Dados	35
2.2.5 A Especificação das Apresentações	36
2.2.6 A Especificação das Operações	41
3 O DICIONÁRIO DE DADOS	42
3.1 Estrutura do Dicionário de Dados	43
3.1.1 Entidade DDVAR	44
3.1.2 Entidade FORMS	45
3.1.3 Entidade CAMPOS	46
3.1.4 Entidade APRESENT	47
3.1.5 Entidade TRANSAS	47
3.1.6 Entidade NODOS	49
3.1.7 Entidade TIPOS	51

3.1.8	Rotinas de Manipulação de Formulários ...	54
3.1.9	Ações de Apresentação de Formulários	57
3.1.10	Entidade OPERS	59
3.1.11	Entidade ESTACOES	60
3.1.12	Entidade TABMENS	61
3.2	Primitivas de Acesso e Atualização	61
4	SEMÂNTICA DE APRESENTAÇÃO	64
4.1	A Apresentação como Semântica do Modelo ...	65
4.2	Mecanismos de Visões ("VIEWS")	71
4.3	Necessidades a Atender	75
4.4	Ações de Apresentação	77
5	INTERFACE DE MANIPULAÇÃO	86
5.1	Formatação e Restituição	86
5.2	Interface com o Usuário	87
5.2.1	Acesso ao Sistema pelos Menus	87
5.2.2	Apresentações de Formulários	91
5.3	Interface com o GBD	92
5.4	Interface com o Dicionário de Dados	98
5.5	Interface com as Ações de Apresentação	99
5.6	Características de Implementação	103
6	CONCLUSÕES	104
6.1	A Proposição Inicial do Projeto	104
6.2	O Modelo de Dados para Formulários: o Objeto Formulário	105
6.3	A Definição Sintática de Formulários	107
6.4	Semântica de Apresentação	108
6.5	Mecanismos de Visões ("VIEWS")	109
6.6	Ações de Apresentação	110
6.7	Arquitetura do Projeto e Estágio Atual dos seus Módulos.....	111
6.8	Interface de Manipulação	114
6.9	Futuros Tópicos para Estudo e Especificação	116
ANEXO 1: Extensão do PASCAL para Definição de Formulários.....		118
ANEXO 2: Exemplo de uma Aplicação baseada em Formulários		123
BIBLIOGRAFIA		135

LISTA DE ABREVIATURAS

AE	Automação de Escritórios
BD	Banco de Dados
CPD	Centro de Processamento de Dados
CPGCC	Curso de Pós-Graduação em Ciência da Computação
DD	Dicionário de Dados
DDL	Data Description Language
FD	File Description
GBD	Gerência do Banco de Dados para formulários
GEF	Geração da Estrutura Física
IDF	Interface de Definição de Formulários
IMF	Interface de Manipulação de Formulários
PED	Processamento Eletrônico de Dados
SGBD	Sistema de Gerência de Banco de Dados
SGBDG	Sistema de Gerência de Banco de Dados Generalizados
UFRGS	Universidade Federal do Rio Grande do Sul

LISTA DE FIGURAS

Figura 1	"GAP" de semântica entre a realidade e o BD	14
Figura 2	Níveis de semântica associados aos objetos	16
Figura 3	O modelo hierárquico para representação de formulários	21
Figura 4	Diagrama da arquitetura do sistema ...	26
Figura 5	Modelo Entidade-Relacionamento do DD .	44
Figura 6	Locação de Veículo	66
Figura 7	Devolução de Veículo	67
Figura 8	Cadastro de Cliente - Pessoa Física .	69
Figura 9	FORM2 e apresentação alternativa	70
Figura 10	Mecanismos de ajuda	82
Figura 11	Menu principal	88
Figura 12	Seleção por Formulários	89
Figura 13	Devolução de Veículo	89
Figura 14	Seleção por Transações	90
Figura 15	Conteúdo de uma Tela com um Formulário	92
Figura 16	Arquitetura do Sistema de Formulários em Redes.....	93
Figura 17	Comunicação IMF vs. GBD	94
Figura 18	Coordenadas para dimensionamento do projeto COBATEF	102

RESUMO

Este trabalho propõe um modelo de dados, na forma de objeto, para representação de formulários, com ênfase nos aspectos referentes à apresentação, e a especificação da interface de manipulação de formulários para as atividades estruturadas de escritórios.

A proposição do modelo é feita pela extensão da definição de tipos de dados da linguagem PASCAL para a elaboração de uma nova entidade, o tipo formulário, separada em três contextos: estrutura, apresentação e operações. Este trabalho está direcionado para o estudo do contexto de apresentações nos aspectos de caracterizar o que seja semântica de apresentação e as formas de agregar esta semântica ao objeto formulário. São mostradas as formas com que a interface de manipulação faz chegar ao usuário toda a semântica agregada, no objeto formulário, dentro do contexto de apresentações.

A interface de manipulação insere-se no projeto de automação das atividades estruturadas do escritório, baseada em formulários, num ambiente de microcomputadores interligados em rede. A especificação apresentada mostra as formas de comunicação entre a interface de manipulação e as outras peças de software do sistema: servidor de formulários e dicionário de dados. É apresentada a arquitetura do sistema de automação de escritórios baseada em formulários eletrônicos atualmente em desenvolvimento no Curso de Pós-Graduação em Ciência da Computação da UFRGS.

ABSTRACT

This work proposes an object-oriented data model to represent forms, with emphasis on presentation aspects and specification of the interface to manipulate forms for the structured activities of offices.

The model being proposed is obtained by extending the data type definition structure of Pascal language, in order to obtain a new entity, the form data type. This entity has three contexts: structure, presentation and operations. This work is directed to the study of presentation contexts about the aspects of characterization of what is presentation semantics and the ways to associate this semantics into the object forms. It is shown the ways how the interface of manipulation gets all the aggregated semantics of the object forms for the user, in a context of presentation.

The interface of manipulation inserts in the project of automation of structured activities of offices, based upon forms, in an environment of microcomputers network. The specification presented shows the communication ways between the manipulation interface and other programs of the system: forms file server and data dictionary. The architecture of offices automation system presented here is based upon electronic forms, now being developed in the Curso de Pós-Graduação em Ciência da Computação.

1 INTRODUÇÃO:

Os Sistemas de Informação tem sido utilizados, na maior parte das vezes, em atividades ligadas à administração tradicional. Estes sistemas são apoiados por SGBDs convencionais baseados em modelos que manipulam dados simples e pouco estruturados tais como inteiros, reais e cadeias de caracteres.

Recentemente tem sido desenvolvidos esforços para integrar a estes sistemas outros grupos de atividades com características particulares. Estas novas áreas de aplicação apresentam diferentes necessidades de manipulação de dados. Entre elas podem ser citadas: a Automação de Escritórios, que manipula documentos, formulários e o tempo, o CAD que trabalha com estruturas complexas de dados, a Economia com suas séries históricas e modelos econométricos, entre outras.

Como consequência, o desenvolvimento de novas pesquisas no campo de Banco de Dados tem-se direcionado, principalmente, para a inclusão de um maior parcela da semântica da realidade no modelo de dados. Surge, então, a necessidade de dispor de novos tipos de dados associada à extensão da semântica. Estes estudos levaram ao desenvolvimento de Sistemas de Gerência de Bancos de Dados Generalizados (SGBDG).

Uma tendência dentro dessa área é a representação da realidade por modelos de dados orientados a objetos. De acordo com a definição informal dada por [DIT 86], um sistema de banco de dados é orientado a objetos por prover a representação de uma entidade da realidade por exatamente um objeto do banco de dados e, por consequência, inclui ou habilita a definição de operadores apropriados para explorar a estrutura do objeto.

Portanto, em um SGBD orientado a Objetos a representação da realidade (entidade) é obtida pela associação com um elemento do modelo de dados (objeto). Diferente da modelagem tradicional onde

a entidade é modelada em diversas relações.

A AE tem sido uma das áreas que mais tem-se destacado nas aplicações de SGBD orientados para Objetos. Dentro da automação de escritórios, um dos aspectos mais importantes é o uso dos formulários como elementos de transferência de informação e de controle.

O formulário é o elemento privilegiado para a transferência de dados dentro de uma organização ([TSI 82], [BAT 84]). Os formulários tem, também, um papel preponderante na estrutura de controle das operações. Este controle é processado pelo desencadeamento de atividades pela chegada de um formulário a uma estação de trabalho satisfazendo certas condições pré-estabelecidas baseadas em seu conteúdo.

Dentro de uma visão conceitual, um formulário é um tipo particular de documento generalizado [BOG 83]. No presente projeto, entretanto, devido a sua importância particular o tipo formulário é considerado como um tipo independente, isto é, tendo definição própria, e é usado como elemento básico para desenvolvimento de um SGBD orientado a Objetos. Os conceitos desenvolvidos para documentos foram adaptados ao caso particular dos formulários.

1.1 A Descrição da Realidade:

A modelagem de sistemas de informação constitui-se em um dos problemas de solução mais complexa na área de processamento eletrônico de dados (PED). Isto é decorrência de ser a modelagem uma atividade interdisciplinar. É necessário o trabalho coordenado de especialistas em computação com os usuários. Os primeiros conhecem as soluções técnicas disponíveis e os segundos tem o conhecimento das necessidades das aplicações. O problema da comunicação é grave. Diversas técnicas foram desenvolvidas na área de análise de sistemas para procurar resolver a dificuldade

de comunicação [MAR 78].

Em PED a modelagem dos sistemas de informação está muito ligada ao modelo de dados utilizado pelo sistema de gerência de banco de dados (SGBD) escolhido. Um dos itens mais importantes para permitir uma boa comunicação entre os usuários e os técnicos em BD é a adequação dos conceitos existentes no modelo de dados às entidades a serem modeladas.

Um modelo de dados é um formalismo utilizado para representar as entidades que pertencem à parte da realidade correspondente a um grupo de aplicações. A semântica associada ao modelo corresponde às propriedades de estrutura, de apresentação e de comportamento das entidades selecionadas.

Como consequência, torna-se cada vez mais importante a possibilidade de uma representação mais fiel da realidade. Esta necessidade tem sido detetada em diversos trabalhos que propõem modelos ditos **semânticos** [COD 79], [HAM 81], [SAN 86]. Estes modelos procuram incorporar uma maior parcela da semântica da realidade no modelo de dados através de novos tipos e de operadores de tipo [TSI 82].

1.2 Banco de Dados orientados para Objetos:

Os atuais sistemas de banco de dados manipulam dados alfanuméricos, estruturados em registros ou tuplas. As aplicações precisam ser modeladas segundo o modelo de dados do SGBD, convertendo as especificações da análise numa descrição de dados orientada a registros. Em sistemas avançados como a Automação de Escritórios (AE) isto pode gerar banco de dados inadaptados, complexos, caros e com operações complicadas de recuperação e atualização. Isto é consequência das entidades existentes na realidade serem modeladas por diversas estruturas (tuplas) no banco de dados.

Normalmente os SGBDs admitem tipos de dados básicos (inteiro, real, etc) e compostos (arranjo, registro, etc). Admitem, também, operações (inserir, recuperar, excluir instâncias de uma estrutura) genéricas, no sentido de serem aplicáveis a todas as estruturas construídas com o mesmo modelo de dados. Admitem, por fim, consistências em relação a restrições de integridade implícitas e consistências em relação a restrições de integridade explícitas. É possível associar algumas restrições de integridade na definição da estrutura. Elas encontram-se associadas aos dados e são dependentes de transação como, por exemplo, valores válidos, campo obrigatório, valor inicial, etc. Admitem, também, operadores específicos para tipos de dados (funções para conversão de datas, diferença de datas, etc).

Por mais sofisticadas que sejam as capacidades dos SGBDs, sempre é necessário modelar a realidade de acordo com as estruturas permitidas no modelo. O que é observado é um "gap" de semântica entre a realidade e o modelo do banco de dados (figura 1). Este "gap" de semântica é coberto pelos programas de aplicação.

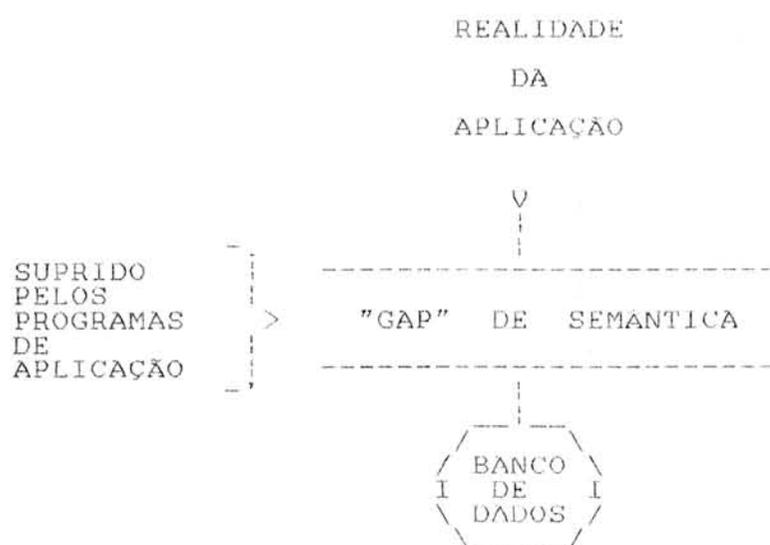


Figura 1. "GAP" de semântica entre a realidade e o BD.

Em sistemas desenvolvidos em linguagens de terceira geração, seja em execução "batch", seja em execução "on-line", toda a parte de validação dos dados, tanto a nível de preenchimento como a nível de fechamento com valores de outros campos, é feita no módulo de consistência dos programas de atualização. Isto é feito porque o próprio modelo de dados não possui mecanismos para agregar esta parte do conhecimento. A equipe de desenvolvimento de sistemas é obrigada a programar estas restrições para manter a qualidade dos dados.

Conforme [DIT 86], o pensamento atual é reduzir este "gap" de semântica, representando a maior parte desta no modelo de dados. Portanto, é necessário desenvolver modelos de dados mais complexos para melhor representar a realidade. Contrariamente aos dados alfanuméricos clássicos, esses dados, denominados **generalizados**, não são atômicos e possuem uma semântica e estrutura próprias. Em automação de escritórios estes dados são: os documentos estruturados, os formulários, as séries históricas e as planilhas. Esta nova classe de aplicações tem necessidades de **objetos** mais ricos que levem em consideração os aspectos semânticos dos mesmos.

Considerando, ainda, o ambiente de trabalho no qual os novos sistemas integrados de **AE** devem operar, verifica-se que as possibilidades de representação da realidade disponíveis nos atuais SGBDs não é suficiente para o desenvolvimento adequado dos sistemas pretendidos. O fator determinante desta insuficiência é a operação direta do sistema pelo usuário final, pois com os mecanismos de manipulação de dados existente hoje, a responsabilidade pela integridade dos dados fica na mão do próprio usuário.

Surge então o conceito de **banco de dados orientado a objeto** que modela a realidade com toda a sua estrutura e complexidade, permitindo a definição de operações apropriadas. Em consequência, através de **interfaces amigáveis** (user friendly interfaces), o usuário poderá utilizar todo o potencial dos objetos sem

necessidade de escrever programas.

São, então, resumidas as principais características desejadas nos novos SGBDs :

- * Tipos de dados estendidos
- * Melhor representação da realidade
- * Instrumentos mais evoluídos para a manipulação dos dados
- * Interfaces capazes de representar de forma intuitiva a semântica do modelo

Identificam-se quatro níveis de semântica associada aos objetos (figura 2) que precisam ser representados em um banco de dados orientado a objeto.

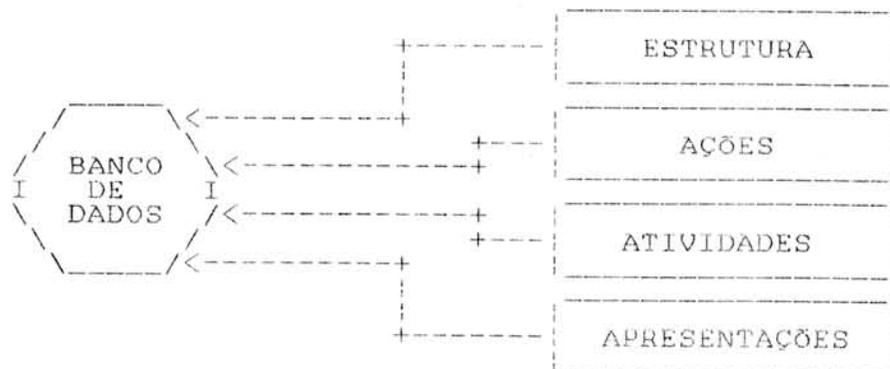


Figura 2. Níveis de semântica associados aos objetos.

Em relação à estrutura, o SGBD deve incluir facilidades para representar estruturas complexas de entidades juntamente com as operações para manipulação, conforme a classificação de orientação à objetos na forma estrutural, proposta em [DIT 86].

Em relação às ações, o SGBD deve ser flexível para permitir a definição de operações próprias do objeto e operadores

diferentes daqueles definidos para os tipos básicos ou compostos [TSI 82], em termos da manipulação dos objetos e dos componentes (atributos) do objeto.

Em relação às atividades, o SGBD deve permitir associar mecanismos de controle, fluxo de objetos e temporização, vinculando estruturas às ações, com o objetivo de viabilizar a manipulação das estruturas como um todo como, por exemplo, consolidar as informações constantes em uma entidade em um gráfico e produzir ocorrências de entidades a partir de outras.

Em relação às apresentações o SGBD deve ser versátil a fim de permitir a definição de diversas apresentações de um objeto para cada uma das etapas de sua manipulação (mecanismos de visões). As apresentações, por si, também fazem parte da semântica do objeto pois a forma do objeto faz parte da realidade a representar. Elas estão associadas ao fluxo de ações, hardware utilizado e interface com o usuário final.

Em resumo, um formulário é caracterizado da seguinte forma: formulário(identificador, estrutura, dados, operadores internos, operadores externos, apresentações). O identificador é uma variável declarada sobre um tipo formulário, que define a estrutura do mesmo. Os dados correspondem a uma instância (preenchimento) de um formulário; os operadores internos correspondem às ações de conteúdo vinculadas a tipos de dados, a campos e ao formulário em si. Os operadores externos correspondem às ações de conteúdo interformulários, ou seja, a ações que manipulam diversas ocorrências de um formulário, ocorrências de diversos formulários ou a combinação destas. As apresentações correspondem às diversas visões ("views") que o formulário possui para sua manipulação.

A primeira proposta desta dissertação é a de contribuir na definição de um modelo de dados para formulários eletrônicos, orientado à objeto, com ênfase nos aspectos referentes à apresentação. A proposição de mecanismos que possibilitem agregar

a semântica de apresentação do formulário no objeto assim como um estudo para caracterizar o que é a semântica de apresentação também faz parte deste trabalho. Finalmente, esta dissertação apresenta a especificação de uma interface para manipulação do objeto formulário pelo usuário no ambiente de escritórios, para as atividades ditas estruturadas (conhecidas de antemão) [PAN 84].

No capítulo 2, é apresentado o modelo, orientado à objetos, para representar formulários, realçando os mecanismos inseridos para agregar, ao modelo, a semântica da entidade nos aspectos de apresentação.

No capítulo 3, é apresentada uma proposta para a implementação de um dicionário ativo de dados capaz de atender as necessidades do projeto. É apresentado, no modelo E-R, a estrutura e vinculações das entidades do dicionário. São propostas definições, em PASCAL, para implementar as estruturas.

No capítulo 4, é feito um estudo do que seja semântica de apresentação, das necessidades a atender em termos de apresentações, e são mostradas as alternativas adotadas no contexto do projeto.

No capítulo 5, é feita a especificação completa da interface de manipulação de formulários. São apresentadas as soluções adotadas tanto para implementar o modelo e suas particularidades como os padrões de relacionamento entre a interface e as outras peças de software do projeto.

No anexo 1, é apresentada a definição completa, em BNF, do modelo de dados formulário.

No anexo 2, é apresentada a especificação de um sistema para uma auto-locadora em termos de formulários.

2 FORMULÁRIOS ELETRÔNICOS:

O objetivo de um sistema integrado de formulários é a eliminação completa, dentro do ambiente de uma organização, do formulário impresso sobre papel.

Os formulários são um elemento comum nas atividades cotidianas. Eles permitem o intercâmbio ordenado e não ambíguo de informações. Estas características são devidas, principalmente, a sua forma ao mesmo tempo estruturada e de fácil interpretação. Além disso, as convenções para o uso de formulários são bem conhecidas e de uso universal. Por estes motivos os formulários são largamente utilizados nos sistemas de aplicações de AE.

Em um sistema informatizado, os formulários existem e são armazenados de forma digital. Estes formulários são denominados **Formulários Eletrônicos**.

O presente projeto, proposto em [OLI 85], integra os trabalhos sobre Modelos de Dados com aqueles ligados a AE e visa o estudo dos diferentes aspectos envolvidos na utilização de formulários em um ambiente automatizado.

O primeiro objetivo consiste em garantir a integração dos conceitos desenvolvidos em SGBD orientados para Objetos com as atividades de escritório de forma a manter os procedimentos naturais de manipulação de formulários. Este é um fator importante pois da transição suave, sem rupturas dos procedimentos atualmente adotados, depende a aceitação de um sistema orientado ao usuário final.

O segundo consiste em desenvolver uma estrutura conceitual consistente para a definição dos diferentes níveis de especificação de formulários. Uma definição clara dos níveis de representação de formulários é essencial para assegurar a independência de dados e de formas de apresentação.

2.1 O Sistema de Formulários Eletrônicos:

Nesta seção é mostrado o sistema de formulários eletrônicos em desenvolvimento no Curso de Pós-Graduação em Ciência da Computação da UFRGS. É mostrada a sua estruturação, o modelo sintático proposto, a arquitetura do sistema, sua concepção e, em especial, sua interface de manipulação.

2.1.1 A Estruturação em Níveis:

O conceito básico utilizado para a especificação de um modelo conceitual para a definição e manipulação de formulários é a decomposição desta atividade em um conjunto de níveis. Cada nível é caracterizado por suas funções e permite a decomposição do problema de modelagem em diversos sub-conjuntos isolados.

Este modelo em níveis foi proposto, inicialmente, para a representação de documentos em [BOG 83]. O formulário, numa visão conceitual, é considerado um tipo particular de documento que possui importância particular o que justifica a modelagem independente ([BAT 84], [YAO 84], [TSI 82]). O referido modelo permite uma grande independência de dados e de dispositivos físicos na modelagem e desenvolvimento de sistemas de aplicação.

A figura 3 apresenta a estruturação em níveis e as etapas do processo de criação de um formulário. A geração de um formulário é composta pela descrição do tipo de formulário, de suas características de formatação, pela criação de um formulário eletrônico e, finalmente, pela reprodução de um formulário em um meio físico.

Um grupo de formulários que possuam a mesma estrutura é identificado pelo correspondente tipo formulário. O conceito de tipo formulário é análogo ao de tipo de dados tal como empregado em linguagens de programação. A descrição do tipo formulário é realizada através da linguagem de definição de dados associada ao

Modelo Sintático.

Em analogia com sistemas convencionais, o modelo sintático corresponde a descrição do arquivo em uma FD de um programa COBOL ou a definição das estruturas feitas em uma DDL de um SGBD convencional. O tipo de dados, por si só, não é elemento suficiente para manipulação. As variáveis associadas ao tipo são os elementos manipuláveis pelo usuário.

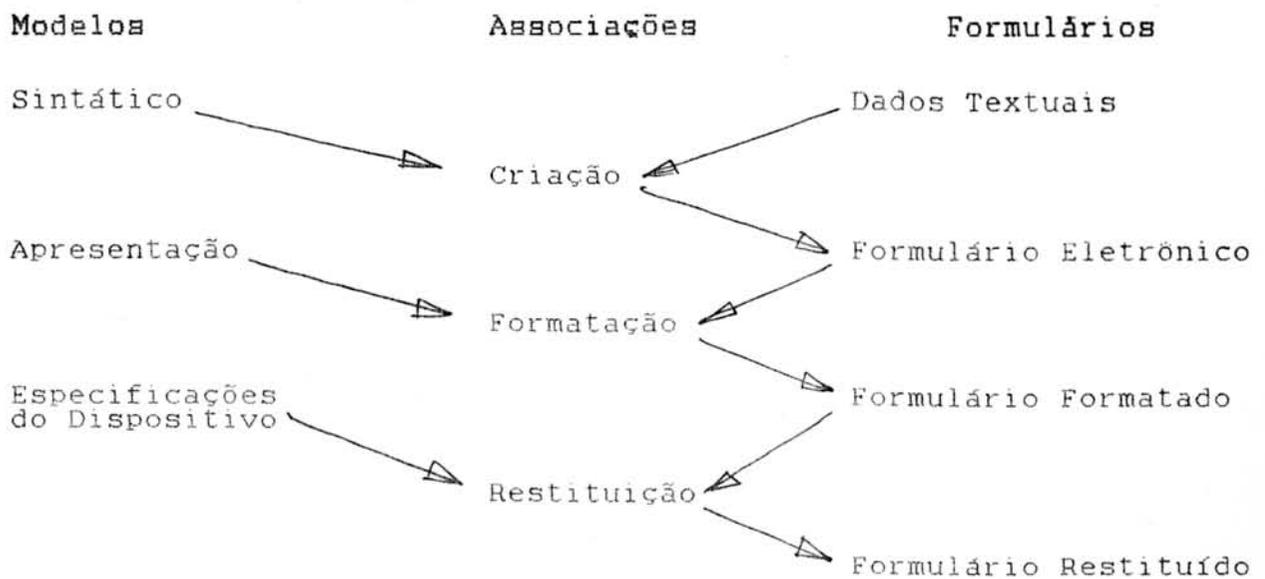


Figura 3. O modelo hierárquico para representação de formulários

A geração de uma ocorrência de um formulário corresponde à associação de um conjunto de dados (*Dados Textuais*) ao modelo sintático. Isto é representado, no banco de dados, pela inicialização de uma área (registro) e pelo armazenamento dos dados correspondentes à estrutura definida no modelo sintático (arquivo). Uma ocorrência de um conjunto de dados associada a um modelo sintático é denominada *Formulário Eletrônico*. Portanto, a geração de uma ocorrência de um formulário corresponde, em sistemas convencionais, à inclusão de um registro em um arquivo.

Um formulário possui características que não são associadas ao seu conteúdo mas ligadas à forma de apresentação. A associação dos formulários eletrônicos às regras de apresentação lógica denomina-se **formatação**. Estas regras de apresentação são por exemplo: realce de um campo ou representação de um grupo de campos associados. Esta característica não tem correspondência nos sistemas convencionais. As formas de apresentação das informações normalmente são implementadas pelos programas aplicativos ou, quando muito, algumas linguagens possuem extensões para definição mais facilitada. O presente projeto, neste aspecto, propõe um avanço significativo nesta área. As formas de apresentação dos formulários fazem parte do próprio modelo de dados.

As regras de apresentação podem ser divididas em dois grupos: as independentes das restrições impostas pelas dimensões lógicas da página e aquelas vinculadas ao formato da página ou janela de exibição. Por exemplo: um texto pode ser formatado em duas colunas (parte puramente lógica) e estas colunas devem ser representadas em uma página com 96 caracteres de largura.

As regras de formatação são lógicas e, portanto, independentes do sistema físico de restituição. Sua modificação não altera o conteúdo do formulário e, dependendo da aplicação, um mesmo formulário eletrônico pode ser associado a diferentes conjuntos de regras de formatação (mecanismos de visões).

Para ser percebido por um usuário o formulário formatado deve ser materializado em um dispositivo de saída. Esta atividade é denominada **restituição**. Um formulário é **restituído** de acordo com as especificações do equipamento (tipo de caracteres, vídeo inverso ou negrito, etc). A restituição consiste na interpretação das regras de apresentação em função das características físicas do equipamento utilizado, e esta atividade é comumente executada por um programa especializado às características do equipamento e, normalmente, referenciado como "driver" do equipamento.

2.1.2 O Modelo Sintático de Formulários:

A descrição do tipo de formulário é feita pelo **Modelo Sintático**. Um modelo sintático constitui-se na descrição de um tipo de formulário. Um formulário é composto por um **cabeçalho** que é utilizado para a descrição externa e pelo **corpo**.

O cabeçalho do formulário corresponde à descrição externa. Esta descrição engloba aspectos pertinentes sobre quem definiu o formulário, em que estágio da elaboração ele se encontra, em que versão ele está, etc. As informações pertinentes ao modelo externo do formulário estão estabelecidas pela norma ISO 646.

O corpo de um formulário é definido pela enumeração de suas propriedades. Cada propriedade é representada por meio de um campo definido sobre um tipo de dado semelhante aos da linguagem Pascal. Estes tipos correspondem ao conceito de **tipos simples**, **tipo renomeado** ou **tipo estruturado** [OLI 84]. Propostas para modelos de formulários são encontradas na bibliografia: [CZE 84], [TSI 82], [YAO 84].

A manipulação efetiva dos formulários a nível de sistema é feita pela declaração de variáveis sobre os tipos formulário, coerente com a conceituação de tipos e variáveis da linguagem PASCAL.

Os identificadores de tipos são incluídos em um dicionário de dados onde, além da definição correspondente, são armazenadas informações sobre o uso, origem e outros dados administrativos. Desta forma é possível manter a uniformidade de descrição dos campos dos diversos tipos de formulário.

O processo de definição dos formulários é facilitado pela existência de uma função de apoio (help) que permite, a qualquer momento, a obtenção de informações sobre os campos dos formulários. Além disso, definições de formulários já feitas podem ser usadas na construção de novos formulários.

Estas facilidades são supridas pela interface de definição de formulários que, de uma forma amigável, disponibiliza uma gama de recursos para utilização. Na presente proposta, a interface de definição de formulários disponibiliza a construção de formulários a partir de outros já definidos pela simples utilização das definições. É objetivo deste projeto que esta facilidade seja disponibilizada por operadores de tipos, tais como, especialização, agregação e generalização. Para tanto, pretende-se integrar o projeto de formulários eletrônicos com os trabalhos sendo desenvolvidos sobre o modelo E, proposto em [SPC 86].

O modelo sintático não limita-se à definição dos diversos campos constantes de um formulário. É definido também o modelo de apresentação do formulário que consiste nas diversas visões pelo qual o formulário é manipulado dentro do ambiente do escritório e, também, do modelo de operações que define as diversas restrições a serem satisfeitas na manipulação e as formas de funcionamento das transações sobre o modelo.

Estas características de apresentação e de operações do modelo constituem-se na agregação da realidade na estrutura de dados, ou seja, da semântica da entidade.

Portanto, no modelo formulário estão associadas duas formas de semântica: a de conteúdo e a de apresentação. Como é mostrado no capítulo 4 deste trabalho para o contexto de apresentações, a semântica deve ser agregada em todos os níveis na construção do modelo. Em consequência, a própria definição de tipos para efeitos do contexto de definição de formulários deve permitir associar especificações que reflitam necessidades a atender tanto em termos de conteúdo como em termos de apresentação. Assim, na definição do tipo de dados, na definição dos campos do formulário, na definição das diversas visões e na definição do formulário em si deve ser possível agregar semântica a nível de conteúdo e a nível de apresentação.

A semântica de apresentação varia de apresentação para apresentação de um mesmo formulário. A de conteúdo, entretanto, é única no contexto do formulário. Caso seja necessário ter semânticas de conteúdo diferentes para um mesmo formulário, este deve ser desdobrado em tantos formulários quantos forem necessários para satisfazer as necessidades.

2.1.3 Arquitetura do Sistema de Formulários Eletrônicos:

A figura 4 mostra a arquitetura básica para o SGBD orientado para formulários. Ela é semelhante (mais aperfeiçoada) à proposta para o projeto FORMANAGER [YAO 84], com o enfoque de tratamento de objetos.

O módulo de Interface de Definição de Formulários (IDF) destina-se a viabilizar a modelagem do sistema de escritório em formulários, através da análise do fluxo de informações e a elaboração das apresentações.

O módulo do Dicionário de Dados (DD) manipula e armazena informações relativas à estrutura, apresentações e operações de formulários. Mantém também as especificações de tipos derivados e seus operadores, bem como cataloga as ações interformulários.

O módulo de Interface de Manipulação de Formulários (IMF) atende as necessidades do usuário final no que diz respeito a interação do usuário com o sistema e nos mecanismos de apresentação.

O módulo de Geração da Estrutura Física (GEF) visa tornar dinâmica a criação de novos formulários no sistema. A sua função básica é criar os arquivos físicos do banco de dados, baseado nas informações previamente inseridas no dicionário de dados (DD). E, ainda, adiciona ao DD informações como arquivos criados e formato de registros assinalando o posicionamento dos campos dentro dos

registros.

O módulo de Gerência do Banco de Dados (GBD) tem como função principal manter, atualizar e recuperar as instâncias dos formulários armazenados no banco de dados. Deve manter e executar (na hora certa) as ações de conteúdo associadas a definição dos formulários, bem como resolver e repassar ao IMF os resultados de consultas e atualizações solicitadas pelo IMF.

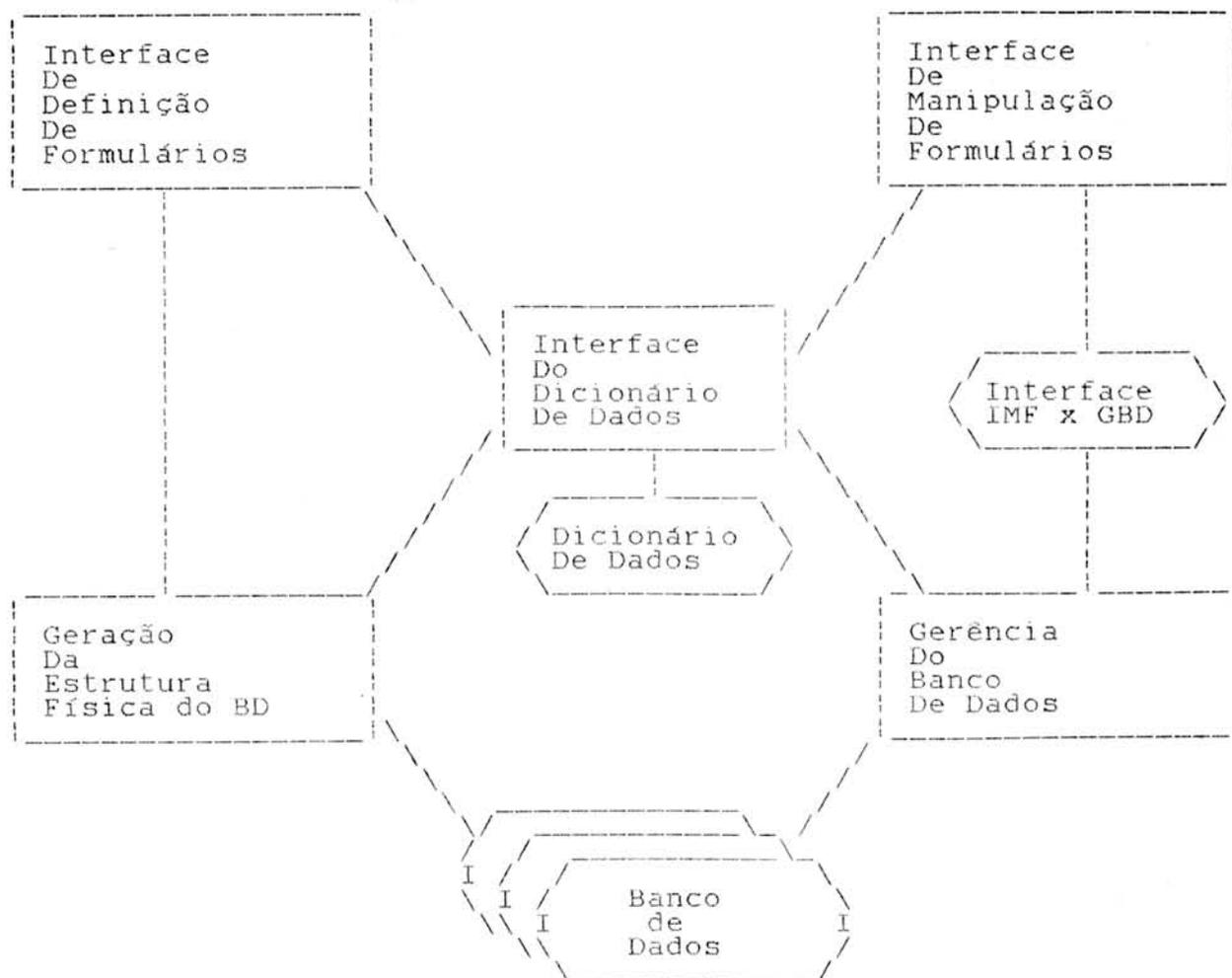


Figura 4 Diagrama da arquitetura do sistema.

2.1.4 A Concepção do Sistema:

A elaboração de um sistema de formulários para escritório principia pela análise do fluxo de informações. Como resultado, são definidos os diversos formulários, seu fluxo no escritório e as visões correspondentes a cada etapa do percurso. Em [TSI 82], é mencionado que esta atividade poderia ser modelada desta maneira e em [BAR 85] é proposta uma metodologia.

Independentemente da metodologia usada para modelar as atividades de escritório, outra área de pesquisa no contexto do projeto, o resultado obtido consiste na especificação dos formulários, seus campos e as condições a serem satisfeitas por cada formulário para transitar dentro do escritório. Cada formulário corresponde à consolidação dos campos manipulados pelas estações por onde transita. Entretanto, cada estação trabalha, na maioria dos casos, com um subconjunto dos campos do formulário: uma visão. A relação dos campos presentes em cada visão também é resultado da modelagem do sistema.

No entanto, a modelagem das apresentações para cada visão depende de aspectos nem sempre ligados ao processamento de dados e, em consequência, dissociados dos resultados da modelagem do sistema, tais como, padronização da empresa, estética e ergonomia. É necessário que mecanismos para elaboração de apresentações também estejam disponíveis além de ferramentas para a modelagem dos sistemas.

A interface de definição objetiva aglutinar estas ferramentas para uso pelo analista de formulários. É por seu intermédio que é modelado o sistema de escritório, definidos os tipos de dados para apoio na definição do formulário, inclusive no aspecto de definição de tipos derivados e dos operadores sobre estes tipos. Pela interface de definição, é feita a elaboração das diversas apresentações dos formulários. Por recursos gráficos, o analista tem condições de definir quais campos farão parte da apresentação sendo elaborada, melhor posicionar os

diversos campos do formulário no plano de apresentação e definir as melhores alternativas a serem usadas, em termos de formatação, para tornar sua manipulação, preenchimento e uso, mais fácil.

2.1.5 A Interface de Manipulação:

A interface de manipulação é a peça de software que faz chegar o sistema de formulários eletrônicos, ou seja, faz chegar ao usuário o tipo de dados associado à variáveis e a apresentação correspondente.

A semântica a nível de apresentações é garantida pela própria interface. Ao receber uma ocorrência, é consultada a descrição da apresentação no dicionário de dados (DD), é produzido o formulário formatado e, pelas características do dispositivo de saída, criada a restituição. O estudo mais aprofundado da IMF é realizado no capítulo 5 deste trabalho.

2.2 A Definição Sintática de Formulários:

Para a especificação do modelo sistático do formulário foi decidido utilizar a linguagem PASCAL e estendê-la pois a mesma possui estruturas de dados mais próximas da realidade a atender que outras linguagens.

A extensão da linguagem PASCAL visa criar mecanismos para agregar as diversas formas de semântica ao tipo de dados. Esta extensão foi efetivada em duas formas: agregação de mais elementos sintáticos na definição de tipos do PASCAL e criação de um novo tipo de dados, o tipo formulário, coerente com a filosofia e padrões sintáticos da linguagem. A especificação completa, em BNF, das extensões agregadas à linguagem estão no Anexo 1 deste trabalho e no Anexo 2 estão descritos, na extensão, os formulários e suas semânticas referentes a um sistema para uma autolocadora.

2.2.1 Variável Formulário:

A declaração de variáveis, no contexto da extensão da linguagem, mantém-se da mesma forma que na linguagem PASCAL, ou seja, pela associação de um identificador ao tipo de dados é possível manipular formulários.

Com isso, o analista de formulários pode definir uma entidade num tipo formulário e, para cada grupo de ocorrências, ter um arquivo apropriado. Por exemplo, se uma entidade Pessoas pertence a dois sistemas diferentes, sistema de pagamento e sistema acadêmico, o tipo de dados é o mesmo: Pessoas; mas são arquivos diferentes: Servidores para o sistema de pagamento e Alunos e Professores para o sistema acadêmico.

A declaração de variáveis de tipos formulários obedece a seguinte sintaxe:

```
<definicao-variaveis> ::= 'var' <declaracao-variaveis>
```

```
<declaracao-variaveis> ::= <declaracao-variavel> ;
      <declaracao-variavel> ';' <declaracao-variaveis>
```

```
<declaracao-variavel> ::= <rol-identificadores> ':'
      <identificador-tipo-formulario>
```

```
<rol-identificadores> ::= <identificador> ;
      <identificador> ',' <rol-identificadores>
```

Exemplo:

```
var LOCARAUT01, LOCALIZ01 : FORM1;
```


Exemplo:

```

type FORM1 (Cadastramento de Clientes: Pessoas Fisicas) =
  formulario
    estrutura
      <conjunto-estruturas>
    fim;
  apresentacao
    <conjunto-apresentacoes>
  fim;
  operacoes
    <conjunto-operacoes>
  fim
fim;

```

O <identificador> especifica o nome do formulário para tratamento dentro do sistema. O título é a forma pela qual o usuário identifica o formulário no escritório. Este é um dos resultados do projeto: o usuário não é obrigado a aceitar como forma de referenciar o formulário o nome (ou, muitas vezes, a sigla) que o sistema trata internamente o formulário. A interface de manipulação apresenta o conteúdo definido em <título> como identificação a ser selecionada pelo usuário. Este mecanismo está implementado nos diversos níveis de definição do formulário como forma de aproximar o formulário com a realidade do usuário.

Em todas as construções sintáticas para definição de tipos, o <identificador> é usado para associação única entre tipo que se está definindo e um nome único ao usuário da entidade definida.

2.2.3 Definição da Estrutura do Formulário:

Em <def-estrutura> são definidos os diversos campos do formulário:

```

<conjunto-estruturas> ::= <estrutura> | <estrutura> ';'
                        <conjunto-estruturas>

<estrutura> ::= <identificador> '(' <titulo> ')' :
                <elemento-estrutura>

<elemento-estrutura> ::= <identificador-tipo-formulario>
                        '.estrutura' | <definicao-campo> |
                        <definicao-lista>

<definicao-lista> ::= 'lista(' <min> ',' <max> ') de'
                    <declaracao-campo>

<declaracao-campo> ::= <identificador> '(' <titulo> ')' :
                    <definicao-campo>

<definicao-campo> ::= <campo> <def-chave> <def-oper-campo>

<def-chave> ::= <empty> | ', chave'

<def-oper-campo> ::= <empty> | ', ' <definicao-de-rotinas>

<campo> ::= <tipo-simples> | <identif-tipo>

<definicao-de-rotinas> ::= <definicao-de-rotina> |
                        <definicao-de-rotina> ', ' <definicao-de-rotinas>

<definicao-de-rotina> ::= <identificador> '('
                        <momentos-de-execucao> ')'

<momentos-de-execucao> ::= <momento-de-execucao> |
                        <momento-de-execucao> ', ' <momentos-de-execucao>

<momento-de-execucao> ::= 'ini' | 'inc' | 'exc' | 'alt' |
                        'sem' | 'fin'

```

Exemplo:

```
type UF = (AC,AL,AP,AM,BA,CE,DF,ES,FN,GO,MA,MG,MS,MT,PA,
          PB,PR,PE,PI,RJ,RN,RO,RR,RS,SC,SE,SP);
type CEP = inteiro:VERIFICA-CEP;
type VALOR = real,APRESENTA-VALOR;
type
```

FORM1(Cadastramento de Clientes: Pessoas Físicas)

estrutura

```
  CODIGO-CLP(Codigo):integer, chave, GERACODIGO(sem);
  NOME-CLP(Nome <peessoa física>):string[40];
  ENDER-CLP(Endereço <Rua,Av.,compl.>):string[40];
  BAIRR-CLP(Bairro):string[15];
  CEP-CLP(CEP):CEP;
  MUNIC-CLP(Município):string[15];
  UF-CLP(U.F.):UF;
  TELEF-CLP(Telefone):string[12];
  DATNASC-CLP(Data de Nascimento):tempo;
  CI-CLP(Carteira de Identidade):inteiro;
  CPF-CLP(CPF):inteiro, VERIFICACPF(SEM);
  PROF-CLP(Profissão):string[16];
  LOCTRAB-CLP(Local de Trabalho):string[40];
  ENDTRAB-CLP(Endereço):string[40];
  BAIRTRAB-CLP(Bairro):string[15];
  CEPTRAB-CLP(CEP):CEP;
  MUNICTRAB-CLP(Município):string[15];
  UFTRAB-CLP(U.F.):UF;
  TELETRAB-CLP(Telefone):string[12];
  RAMAL-CLP(Ramal):inteiro;
  CARGO-CLP(Cargo do Cliente):string[16];
  RENDIM-CLP(Rendimentos do Cliente):VALOR;
fim;
```

A definição da estrutura do formulário, em sua forma mais simples, resume-se na declaração dos diversos campos que o compõem. Na <declaracao-campo>, é feita a especificação dos atributos do formulário pela associação de um identificador. Na

mesma construção, é associado um título para traduzir a realidade em termos de especificação de campos dos formulários dentro do escritório e é especificado o tipo de dado que o campo representa.

Associada à definição do campo, a <definicao-de-rotinas> associa ações de conteúdo. Estas ações destinam-se a definir, a nível de modelo de dados, as restrições de integridade que o conteúdo do campo deve satisfazer. Além disso, a <def-chave> define que o campo é considerado chave para a estrutura e, por conseguinte, o servidor de formulários mantém uma estrutura de acesso classificada para a estrutura.

As outras formas de especificar <elemento-estrutura> são <identificador-tipo-formulario> 'estrutura' e <definicao-lista>. A primeira forma destina-se a usar uma definição de campos já escrita para outro formulário. Desta maneira, a interface de definição fornece ao analista de formulários a possibilidade de reutilizar definições anteriores. A segunda forma tem por objetivo criar uma lista com cardinalidade variável. Sendo do interesse do analista de formulários, é possível definir operações a nível de lista. Operações de inserção e remoção são maneiras naturais de tratar esta estrutura de dados.

A especificação do <momento-de-execucao> na associação de rotinas à definição de campos destina-se a vincular a execução dessas rotinas a uma ou mais operações básicas sobre formulários. A nível de apresentações é feita a associação da transação básica a executar no formulário com a transação do usuário.

A definição do tipo de dados de cada campo está restrita a <tipo-simples>. No entanto, isto não caracteriza uma limitação na especificação do modelo. A forma <identif-tipo> abre o espectro de definição de tipos aos limites da extensão proposta pelo projeto.

2.2.4 A Extensão da Definição de Tipos de Dados:

A complementação da definição da estrutura é feita pela extensão da definição de tipos da linguagem:

```

<declaracao-tipo-dados> ::= '=' <tipo-dados> <acao-apresentacao>
                               <apresentacao-textos>

<tipo-dados> ::= <tipo-basico> | <tipo-estruturado> | <tipo-tempo>

<tipo-basico> ::= <tipo-simples> <derivacao> | <tipo-restrito>

<tipo-simples> ::= 'inteiro' | 'real' | 'booleano' | 'caracter' |
                  'string([' <inteiro-sem-sinal> '])'

<derivacao> ::= <empty> | ':' <ident-rotina>

<tipo-restrito> ::= <tipo-escalar> | <tipo-intervalo> |
                  <identif-tipo>

<tipo-estruturado> ::= <tipo-arranjo> | <tipo-registro>

<tipo-arranjo> ::= 'arranjo([' <tipo-restrito> ']) de '
                  <tipo-basico>

<tipo-registro> ::= 'registro' <conjunto-tipos-basicos> 'fim'

<conjunto-tipos-basicos> ::= <declaracao-tipo-basico> |
                             <declaracao-tipo-basico> ';' <conjunto-tipos-basicos>

<declaracao-tipo-basico> ::= <identificador> '(' <titulo> '):'
                             <tipo-basico>

<tipo-tempo> ::= 'tempo' | 'periodo' | 'duracao'

```



```

<construtor-apresentacao> ::= 'transacoes'
    <conj-transacoes-apresent> 'fim; globais'
        <conjunto-apresentacoes-globais> 'fim;'
            <conjunto-apresentacoes-nodos>

<conj-transacoes-apresent> ::= <transacao-apresent> |
    <transacao-apresent> ';' <conj-transacoes-apresent>

<transacao-apresent> ::= <identificador> '(' <titulo> ')

<acao-apresentacao> ::= <empty> | ',' <identificador>

<conjunto-apresentacoes-globais> ::= <apresentacao-global> |
    <apresentacao-global> ';' <conjunto-apresentacoes-globais>

<apresentacao-global> ::= <apresentacao-cabecalhos>
    <apresentacao-textos>

<apresentacao-cabecalhos> ::= <empty> | 'cabecalho(' <texto> ')
    <acao-apresentacao> ';'

<apresentacao-textos> ::= <empty> | <conteudo-texto>
    <acao-apresentacao>

<conjunto-apresentacoes-nodos> ::= <apresentacao-nodo> |
    <apresentacao-nodo> ';' <conjunto-apresentacoes-nodos>

<apresentacao-nodo> ::= <nodo> | 'grupo' <acao-apresentacao>
    <apresentacao-textos> ';'
    <conjunto-apresentacoes-nodos> 'fim'

<nodo> ::= 'nodo' <nome-nodo> <acao-apresentacao>
    <apresentacao-textos>

<conteudo-texto> ::= '#' <texto> '#'

```

Exemplo:

```

type SIMNAO = (S,N);
type OPCSEG = (1,2,3);
type
  FORM4(Locacoes) =
    formulario
      estrutura
        CODVEI-LOC(Codigo Veiculo):inteiro,chave,VERIFCODVEICULO(sem);
        CODCLI-LOC(Codigo Cliente):inteiro,chave,VERIFCODCLIENTE(sem);
        DATASAI-LOC(Data de Saida):tempo;
        ODOMSAI-LOC(Odometro de Saida):inteiro;
        OPCSEG-LOC(Opcao Seguro):OPCSEG;
        DANOSPESS-LOC(Danos Pessoais Seguro):SIMNAO;
        DEVPREV-LOC(Data Devolucao Prevista):tempo;
        DEVOL-LOC(Data Devolucao):tempo;
        ODOMCHEG-LOC(Odometro de Chegada):inteiro;
        LITRCOMB-LOC(Litros de Combustivel):real;
        VALCOMB-LOC(Valor Combustivel):VALOR,CALCVCOMB-LOC(sem);
        KMROD-LOC(Km Rodados):inteiro,CALCKM-LOC(sem);
        VALKMROD-LOC(Valor Km Rodados):VALOR,CALCVKMROD-LOC(sem);
        NDIAS-LOC(Numero de Dias Locados):inteiro,CALCNDIAS-LOC(sem);
        VALNDIAS-LOC(Valor das Diarias):VALOR,CALCVDIAR-LOC(sem);
        TOTAL-LOC(Valor a Pagar):VALOR,CALCVTOT-LOC(sem);
      fim;
    apresentacao
      SAIDA-VEICULO:
        visao,
          APRESENTA-SAIDA;
        transacoes
          SAIDA-VEIC(Saida de Veiculo com Cliente);
          ALT-SAIDA-VEIC(Correcao das Informacoes sobre Saida de
Veiculo)
        fim;
      globais
        cabecalho(Locacao de Veiculo),APRESENTA-CABEC;

```

#Este Formulário serve para Registrar a Saída de um
Veículo Locado#

fim;

grupo, APRESENT-MOLDURA;

nodo CODVEI-LOC;

nodo CODCLI-LOC;

fim;

grupo, APRESENT-MOLDURA;

nodo DATASAI-LOC, APRESENTA-TEMPO;

nodo ODOMSAI-LOC;

nodo OPCSEG-LOC;

nodo DANOSPES-LOC;

nodo DEVPREV-LOC, APRESENTA-TEMPO;

nodo CAUCAO-LOC

fim;

fim;

DEVOLUCAO-VEICULO:

visao,

APRESENTA-DEVOLUCAO;

transacoes

DEVOL-VEIC(Devolução de Veículo pelo Cliente e Cobrança);

ALT-DEVOL-VEIC(Correção das Informações sobre Devolução de

Veículo)

fim;

globais

cabecalho(Devolução de Veículo), APRESENTA-CABEC;

#Este Formulário serve para Registrar a Devolução de um
Veículo e para a Cobrança do Valor da Locação#

fim;

grupo, APRESENT-MOLDURA;

nodo CODVEI-LOC;

nodo CODCLI-LOC;

fim;

grupo, APRES-MOLDURA-LEIT;

nodo DATASAI-LOC, APRESENTA-TEMPO;

nodo ODOMSAI-LOC;

nodo OPCSEG-LOC;

```

    nodo DANOSPESS-LOC;
    nodo DEVPREV-LOC, APRESENTA-TEMPO;
    nodo CAUCAO-LOC
fim;
grupo, APRESENTA-MOLDURA;
    nodo DEVOL-LOC, APRESENTA-TEMPO;
    nodo ODOMCHEG-LOC;
    nodo LITRCOMB-LOC;
    nodo VALCOMB-LOC;
    nodo KMROD-LOC;
    nodo VALKMROD-LOC;
    nodo NDIAS-LOC;
    nodo VALNDIAS-LOC;
    nodo TOTAL-LOC;
    nodo TOTLIQ-LOC
    fim;
fim;
operacoes
    <def-operacoes>
    fim;
fim;

```

Um formulário pode ter diversas apresentações. Cada apresentação representa uma etapa do preenchimento do formulário.

Para cada apresentação podem ser associadas diferentes regras de formatação de forma que o preenchimento na etapa em questão seja facilitada. A possibilidade de existir mais de uma apresentação para um mesmo formulário foi sugerida, também, em [TSI 82].

Para manipulação pelo sistema, cada apresentação tem um identificador único. As atividades que podem ser realizadas pela apresentação são definidas em <conj-transacoes-apresent>. O <identificador> associa a transação com atividades básicas a nível de funções do servidor de formulários. O <titulo> vincula a

transação a nível de sistema com o nome ou expressão pelo qual o usuário identifica a transação a nível do ambiente do escritório.

O <conjunto-apresentacoes-globais> define os conteúdos dos cabeçalhos (<apresentacao-cabecalhos>) e dos textos de ajuda a nível de formulário (<apresentacao-textos>). A relação dos campos que fazem parte da apresentação corrente é feita em <conjunto-apresentacoes-nodos>. É necessário relacionar os campos associados a cada apresentação pois é mais comum que apenas um subconjunto dos campos dos formulários sejam manipulados em cada estação de trabalho. Os mecanismos de visões baseiam-se nessa característica. Apenas nos casos em que o formulário possua uma única apresentação todos os campos são relacionados.

O agrupamento de campos na apresentação é uma característica do modelo proposto e objetiva permitir associar ações de apresentação a nível de agrupamento de campos. Esta necessidade fica ilustrada no anexo 2 e o agrupamento de campos para apresentação também foi uma preocupação da metodologia apresentada em [BAT 84].

Como visto na sintaxe, a <acao-apresentacao> pode ser associada nos diversos níveis de definição da apresentação. As prioridades e sobreposições entre as diversas ações semânticas de apresentação além dos valores assumidos por omissão são estudados com mais profundidade no capítulo 4 deste trabalho.

2.2.6 A Especificação das Operações:

Em <conjunto-operacoes> são especificadas as diversas ações semânticas de operações a nível do formulário como um todo. Neste contexto são comumente especificadas as rotinas de fechamento intercampos do formulário. O estudo completo dos diversos aspectos de definição e manipulação de conteúdos de formulários é um dos objetos da dissertação sobre servidor de arquivos para formulários.

3 O DICIONÁRIO DE DADOS:

Neste capítulo, é mostrado como foi estruturado o dicionário de dados do projeto de formulários eletrônicos.

No contexto deste projeto, o dicionário de dados centraliza todas as descrições definidas no sistema. Tanto as interfaces, de definição e de manipulação, como o servidor de formulários são peças de software interpretativas, ou seja, são independentes de qualquer particular conjunto de definições de formulários. Quando necessário, as descrições são buscadas no dicionário de dados (DD).

Em sistemas de gerenciamento de dados compilativos, a descrição da base de dados é necessária apenas no momento da compilação das diversas peças de software, ou seja, no próprio código dos programas de aplicação ficam armazenadas todas as características e restrições a respeito da estrutura de dados.

Nos sistemas interpretativos a descrição da base de dados é um componente atuante em todas as fases. Desde a definição até a manipulação dependem dessa descrição. No DD esta atuação é consubstanciada na definição de ações nos diversos níveis de especificação do formulário: a nível de tipo de dados, a nível de campo, a nível de grupo de campos, a nível de apresentações, a nível de formulário e a nível de grupo de formulários.

Dessa forma, o DD é um elemento ativo na operação do sistema como um todo. As primitivas (ou as rotinas utilizando as primitivas) devem poder consultá-lo para verificações de integridade, mecanismos de ajuda, gerência de comunicação com o usuário (origem dos dados apresentados nos menus), etc.

Este dicionário tem por objetivo principal guardar a definição dos formulários, ou seja, a definição completa da estrutura dos formulários, as definições das diversas

apresentações dos formulários e a definição das operações pertinentes a cada formulário. Enfim, o que é representado no DD são os objetos formulários.

As diversas peças de software do projeto interagem com o dicionário de dados. Na fase atual do projeto, foi implementado um reconhecedor para sintaxe de linguagens escritas em BNF que, a partir dessa definição, analisa sintaticamente a especificação de formulários e produz chamadas às primitivas do dicionário para inclusão das especificações.

3.1 Estrutura do Dicionário de Dados:

Foram especificadas dezessete entidades para manter as informações do dicionário de dados:

- DDVAR,
- FORMS,
- CAMPOS,
- APRESENT,
- TRANSAS,
- NODOS,
- TIPOS,
- ROTSIS,
- ROTFORMS,
- ROTCAMPS,
- ROTTIPOS,
- APRAPRES,
- APRNODOS,
- APRTIPOS,
- OPERS,
- ESTACOES e
- TABMENS.

A figura 5 mostra o relacionamento entre as diversas entidades.

Na implementação, todos os relacionamentos do tipo "1 para n" foram implementados inserindo a chave da estrutura do lado "1" do relacionamento na estrutura do lado "n" do relacionamento.

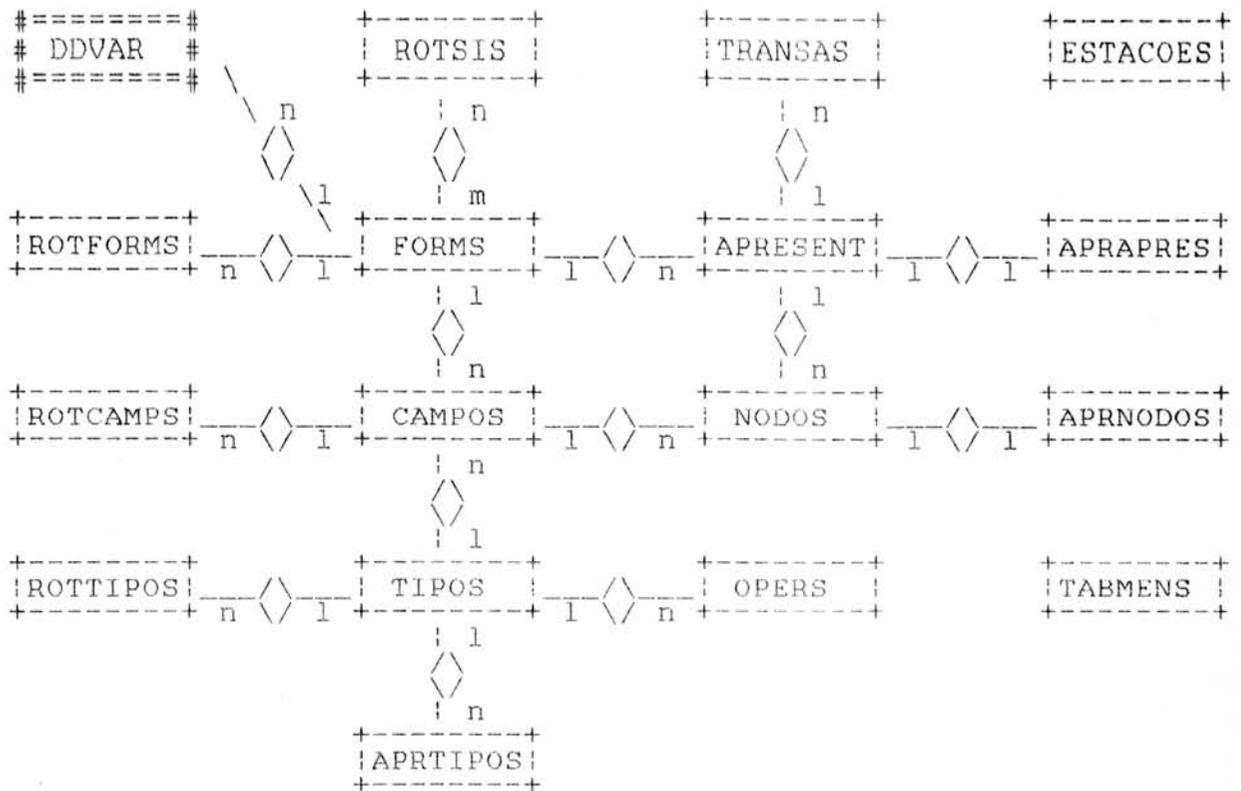


Figura 5 Modelo Entidade-Relacionamento do DD.

3.1.1 Entidade DDVAR:

A entidade DDVAR destina-se a catalogar as variáveis definidas sobre tipos formulários. Serve como estrutura base para acesso das descrições contidas no DD.

A partir de uma variável formulário, facilmente é construída a descrição completa do tipo.

O formato PASCAL para representar uma entrada na entidade

DDVAR é mostrado abaixo.

```
TYPE
  TIPODDVAR
    RECORD
      VARIAVEL      : STRING[20];
      IDENTFORMS    : STRING[20];
    END;
```

Exemplo:

```
VARIAVEL = LOCALIZA1
IDENTFORMS = FORM1
```

3.1.2 Entidade FORMS:

A entidade FORMS destina-se a catalogar os formulários definidos no sistema.

As informações armazenadas na estrutura são um identificador de formulário para servir como elemento de vinculação entre as diversas estruturas e um título que é a forma como o formulário é visto (ou chamado) dentro do ambiente do escritório.

O formato PASCAL para representar uma entrada da entidade FORMS é mostrado abaixo.

```
TYPE
  TIPOFORMS =
    RECORD
      IDENTIFICADOR : STRING[20];
      TITULO         : STRING[80];
    END;
```

Exemplo:

```
IDENTIFICADOR = FORM1
TITULO = Cadastramento de Clientes: Pessoas Físicas
```

3.1.3 Entidade CAMPOS:

A entidade CAMPOS mantém a descrição dos atributos de um formulário. É composto do identificador do campo, do título, do tipo de dados, do identificador do formulário a que pertence e se é chave deste. Além disso, contém coordenadas para localização do campo no registro de armazenamento de instâncias do formulário.

O formato PASCAL para representar uma entrada na entidade CAMPOS é mostrada a seguir.

```
TYPE
TIPOCAMPOS =
    RECORD
        IDENTFORMSCAMPOS : STRING[20];
        IDENTIFICADOR    : STRING[20];
        TITULO            : STRING[80];
        IDENTTIPOCAMPO   : STRING[20];
        CHAVE             : BOOLEAN;
        POSINIC          : INTEGER;
        NROPOS           : INTEGER;
        FORMATO          : CHAR;
    END;
```

Exemplo:

```
IDENTFORMSCAMPOS = FORM1
IDENTIFICADOR = CODIGO-CLP
TITULO = Codigo
IDENTTIPOCAMPO = Integer
CHAVE = true
```

POSINIC = 1

NROPOS = 5

FORMATO = E

3.1.4 Entidade APRESENT:

A entidade APRESENT contém uma entrada para as apresentações de cada formulário. E' composta pelo identificador do formulário a que pertence, do identificador para a apresentação e do nome da ação de apresentação associada, caso presente.

O formato PASCAL abaixo representa uma entrada na entidade APRESENT.

TYPE

TIPOAPRESENT =

RECORD

IDENTFORMSAPRESENT : STRING[20];

NOMEAPRESENT : STRING[20];

NOMEACAOAPRESENT : STRING[20];

END;

Exemplo:

IDENTFORMSAPRESENT = FORM1

NOMEAPRESENT = SAIDA-VEICULO

NOMEACAOAPRESENT = APRESENTA-SAIDA

3.1.5 Entidade TRANSAS:

A entidade TRANSAS cataloga as transações que podem ser executadas a nível de cada apresentação. As transações são identificadas por um nome significativo (ou um pequeno número de palavras), representando a ação executada. Por exemplo, no formulário de locação de automóveis, existe a transação "saída de veículo" e esta será mapeada em primitivas do servidor de

formulários.

Ela contém o identificador da apresentação a qual pertence, do identificador da transação para manipulação pelo sistema, do título da transação para manipulação pelo usuário e do código da primitiva do servidor de formulários correspondente à transação.

Um aspecto importante a realçar a respeito das transações associadas às apresentações é que uma transação chamada preenchimento, presente nas duas apresentações de um mesmo formulário, pode significar execuções de primitivas diferentes, no contexto do servidor de formulários. No exemplo da autolocadora, no FORM4, as transações de saída de veículo e de devolução de veículo podem ser vistas como sendo dois preenchimentos quando, para efeitos do servidor, são uma inclusão e uma alteração.

O formato PASCAL abaixo representa uma entrada na entidade TRANSAS:

```

TYPE TIPOTRANSAS =
    RECORD
        NOMEAPRESENT : STRING[20];
        NOMETRANSACAO : STRING[20];
        TITRANSACAO : STRING[80];
        PRIMITIVAGBD : STRING[03];
    END;

```

Exemplos:

```

NOMEAPRESENT = SAIDA-VEICULO
NOMETRANSACAO = SAIDA-VEIC
TITRANSACAO = Saída de Veículo com Cliente
PRIMITIVAGBD = inc
NOMEAPRESENT = DEVOLUCAO-VEICULO
NOMETRANSACAO = DEVOL-VEIC
TITRANSACAO = Devolucao de Veiculo pelo Cliente e
                Cobranca
PRIMITIVAGBD = alt

```

3.1.6 Entidade NODOS:

A entidade NODOS contém as apresentações globais dos formulários, os textos para os mecanismos de help e as características de formatação dos campos de um determinado formulário.

E' composto do identificador do formulário de que faz parte, do identificador da apresentação a que pertence; caso represente um nodo contém, também, o identificador do campo ao qual está associado e o identificador da ação de apresentação associada.

O registro de nodos tem sua descrição variável dependente do tipo de nodo. Se fo um cabeçalho, seu conteúdo; se for um texto para os mecanismos de ajuda, seu conteúdo e a identificação do nível a que está vinculado: nível de apresentação, nível de grupo de campos, nível de campo ou nível de tipo; se for um nodo, o posicionamento do campo no plano de apresentação (como usado em [PEE 85] e, também, sugerido em [RIC 81]), e o dimensionamento do campo, largura, altura e profundidade, de acordo com a proposta do projeto COBATEF [PEE 85] na caracterização dos elementos para formatação automática de textos. Se o registro representar um grupo de nodos, o nome do nodo é gerado pelo sistema pois a nível de definição do formulário não é especificado um identificador para cada grupo de campos.

O formato PASCAL abaixo representa uma entrada na entidade NODOS.

```

TYPE
  TIPOREGNODOS = (GLOBAL,TEXTO,GRUPO,NODO);
  TIPOELEMASOC = (TXAPRESENT, TXGRUPO, TXNODO, TXTIPO);
  TIPOUSO = (ES, SAIDA);
TYPE
  TIPONODOS =
    RECORD
      IDENTFORMSNODO : STRING[20];

```

```

NOMEAPRESENTNODO : STRING[20];
NOMENODO          : STRING[20];
NOMEGRUPONODO    : STRING[20];
NOMEACAOAPRESENT : STRING[20];
CASE TIPOREGNODO : TIPOREGNODOS OF
GLOBAL: ( GLOBALNODO      : STRING[80]);
TEXTO : ( ELEMASSOC       : TIPOELEMASSOC;
          TEXTONODO       : STRING[80]);
GRUPO : ( USOGRUPO        : TIPOUSO);
NODO  : ( USONODO         : TIPOUSO;
          POSHORIZ        : INTEGER;
          POSVERT         : INTEGER;
          LARGURA        : INTEGER;
          ALTURA         : INTEGER;
          PROFUNDIDADE    : INTEGER);
END;

```

Exemplos:

```

IDENTFORMSNODO = FORM4
NOMEAPRESENTNODO = SAIDA-VEICULO
NOMENODO = --
NOMEGRUPONODO = --
NOMEACAOAPRESENT = APRESENTA-CABEC
TIPOREGNODO = GLOBAL
GLOBALNODO = Locacao de Veiculo

IDENTFORMSNODO = FORM4
NOMEAPRESENTNODO = SAIDA-VEICULO
NOMENODO = --
NOMEGRUPONODO = --
NOMEACAOAPRESENT = --
TIPOREGNODO = TEXTO
ELEMASSOC = TXAPRESENT
TEXTONODO = Este Formulário serve para Registrar a
           Saida de um Veiculo Locado

```

```
IDENTFORMSNODO = FORM4
NOMEAPRESENTNODO = SAIDA-VEICULO
NOMENODO = FORM4g2
NOMEGRUPONODO = --
NOMEACAOAPRESENT = APRESENT-MOLDURA
TIPOREGNODO = GRUPO
USOGRUPO = ES
```

```
IDENTFORMSNODO = FORM4
NOMEAPRESENTNODO = SAIDA-VEICULO
NOMENODO = DATASAI-LOC
NOMEGRUPONODO = FORM4g2
NOMEACAOAPRESENT = APRESENTA-TEMPO
TIPOREGNODO = NODO
USONODO = ES
POSHORIZ = 00008
POSVERT = 00006
LARGURA = 00019
ALTURA = 00001
PROFUNDIDADE = 00001
```

3.1.7 Entidade TIPOS:

A entidade TIPOS armazena as definições dos tipos de dados usados nos diversos formulários. Ela é implementada em duas estruturas: TIPOS e TIPOS2. A estrutura TIPOS possui uma entrada para cada tipo de dado e a estrutura TIPOS2 é uma área para colocar a complementação de alguns tipos de dados cuja entrada na entidade é ou muito longa ou de tamanho imprevisível, tais como RECORD, ARRAY, enumerações, etc.

E' armazenado o identificador de tipo e, conforme o tipo derivador, informações pertinentes à definição do tipo. Além disso, é possível associar uma ação semântica de apresentação a nível de tipo de dados para que fique uniformizada a formatação para o referido tipo.

O formato PASCAL para representar uma entrada nas estruturas TIPOS e TIPOS2 é mostrado a seguir.

TYPE

TIPOREGTIPOS = 1..8;

TYPE

TIPOTIPOS =

RECORD

NOMETIPO : STRING[20];

CONTADOR : INTEGER;

NOMEACAOAPRESENT : STRING[20];

CASE TIPOTIPO : TIPOREGTIPOS OF

1:(LIMINFTIPO1 : INTEGER; { INTEIRO }
LIMSUPTIPO1 : INTEGER);

2:(LIMINFTIPO2 : REAL; { REAL }
LIMSUPTIPO2 : REAL);

3:(); { CHAR }

4:(NROPOSTIPO4 : INTEGER); { STRING }

5:(); { ESCALAR }

6:(LIMINFTIPO6 : REAL; { INTERVALO }
LIMSUPTIPO6 : REAL);

7:(); { ARRANJO }

8:(); { REGISTRO }

END;

TYPE

TIPOTIPOS2 =

RECORD

NOMETIPO2 : STRING[20];

TEXTOTIPO : STRING[80];

CASE TIPOTIPO2:TIPOREGTIPOS OF

1:(); 2:(); 3:(); 4:();

5:(ELEMESCALAR : STRING[20]); { ESCALAR }

6:();

7:(NOMETIPODIM : STRING[20]; { ARRANJO }

NOMEELEMARRANJO : STRING[20]);

8:(NOMEELEMREG : STRING[20]); { REGISTRO }

END;

Exemplos:

NOMETIPO = inteiro
NOMEACAOAPRESENT = --
TIPOTIPO = 1
LIMINFTIPO1 = -32767
LIMSUPTIPO1 = 32767

NOMETIPO = CEP
NOMEACAOAPRESENT = --
TIPOTIPO = 6
LIMINFTIPO6 = 01000
LIMSUPTIPO6 = 99999

NOMETIPO = NOME
NOMEACAOAPRESENT = --
TIPOTIPO = 4
NROPOSTIPO4 = 40

NOMETIPO = VALOR
NOMEACAOAPRESENT = APRESENTA-VALOR
TIPOTIPO = 2
LIMINFTIPO2 = -9.999.999.999,99
LIMSUPTIPO2 = 9.999.999.999,99

NOMETIPO = UF
TIPOTIPO = 5
NOMETIPO2 = UF
TIPOTIPO2 = 5
ELEMESCALAR = AC
NOMETIPO2 = UF
TIPOTIPO2 = 5
ELEMESCALAR = AL

...
NOMETIPO2 = UF
TIPOTIPO2 = 5
ELEMESCALAR = SP

3.1.8 Rotinas de Manipulação de Formulários:

A entidade ROTSIS cataloga rotinas que manipulam diversos formulários: instâncias ou tipos. Através da interface, o usuário seleciona quais rotinas estão disponíveis para um determinado formulário e vice-versa. Existem três contextos para execução de rotinas interformulários: na primeira abertura de sessão no sistema (ativação do servidor), sempre que uma determinada condição se verifique e por decisão do usuário. É importante realçar que o preenchimento de um formulário cujas informações são resultado da consolidação de dados presentes em outros formulários é possível justamente pela existência destas rotinas interformulários.

O formato PASCAL abaixo representa uma entrada na entidade ROTSIS.

```

TYPE
  TIPOROTSIS =
    RECORD
      NOMEROTSIS : STRING[20];
      TITROTSIS  : STRING[80];
    END;

```

Exemplo:

```

NOMEROTSIS = CRIA-C-CHEQUES
TITROTSIS  = Consolida e Gera Contra-Cheques

```

A entidade ROTFORMS cataloga as rotinas que executam algum processamento a nível de formulário. A associação de rotinas ao tipo formulário acresce semântica ao formulário como um todo. Neste contexto insere-se os procedimentos para consistência cruzada entre os conteúdos dos diversos campos (rotina de fechamento) e o controle da vinculação lógica da atualização de vários campos calculados. Explicando melhor: se há uma modificação de informações internas para a instância corrente,

tal como, número de dependentes, os campos de imposto retido na fonte e salário líquido devem ser recalculados; uma das formas de efetivar o recálculo é ter uma rotina (por exemplo RECALCIMPELIQ) que vai ativar para execução as respectivas rotinas de atualização dos campos.

A definição de rotinas para cálculo do valor de um campo manipulam, somente, campos não calculados. Caso uma rotina de cálculo pudesse utilizar um campo calculado por outra rotina, a responsabilidade de evitar que uma dependesse do resultado da outra ficaria com o analista de formulários.

A estrutura contém o identificador do formulário ao qual está associado, o identificador da rotina no contexto do sistema e a especificação de em qual transação ela deve ser executada.

O formato PASCAL abaixo representa uma entrada na entidade ROTFORMS.

```

TYPE
  TIPOROTFORMS =
    RECORD
      IDENTFORMSROTFORMS : STRING[20];
      NOMEROTFORMS       : STRING[20];
      TIPOTRANSACAO      : STRING[03];
    END;

```

Exemplo:

```

IDENTFORMSROTFORMS = FORM3
NOMEROTFORMS       = VERIFICA-COR
TIPOTRANSACAO     = inc

```

A entidade ROTCAMPS cataloga as rotinas que agem sobre um campo. O recurso de associar uma ação a um campo torna mais inteligível a definição do objeto formulario na medida em que o analista de formulários tem condições de localizar no campo o procedimento a ser executado. Isto vem de encontro com as novas

tendências em modelo de dados e objetos pois se a ação age somente sobre um campo, nada mais natural que estar associada diretamente ao campo.

Portanto, a associação de rotinas a nível de campo é uma das formas de acrescer semântica a uma determinada informação. Este acréscimo pode ser feito para cada tipo de transação: para inicialização, valores constantes. Por exemplo, o campo salário que armazena o salário base do empregado pode ter, como valor inicial, o salário mínimo. De forma semelhante, podem ser associadas rotinas para as outras operações, tais como, inclusão, alteração, exclusão e consulta.

O formato PASCAL abaixo representa uma entrada na entidade ROTCAMPS.

```

TYPE
  TIPOROTCAMPS =
    RECORD
      IDENTFORMSROTCAMPS : STRING[20];
      IDENTCAMPOROTCAMPS : STRING[20];
      NOMEROTCAMPS       : STRING[20];
      TIPOTRANSACAO      : STRING[03];
    END;

```

Exemplo:

```

IDENTFORMSROTCAMPS = FORM1
IDENTCAMPOROTCAMPS = CODIGO-CLP
NOMEROTCAMPS = GERACODIGO
TIPOTRANSACAO = sem

```

A entidade ROTTIPOS cataloga as rotinas associadas a tipo de dados, ou seja, as diversas restrições de integridade que o tipo de dados deve satisfazer ou regras de geração de valores a ser utilizada. Neste contexto, a semântica associada caracteriza-se por ser independente da transação sendo efetuada e do formulário sendo manipulado. Com isso, é garantida a integridade

da informação que se encontra armazenada em diferentes formulários: os campos são diferentes para cada formulário mas são definidos sobre o mesmo tipo de dados.

O formato PASCAL que representa uma entrada na entidade ROTTIPOS é mostrada abaixo.

```

TYPE
  TIPOROTTIPOS =
    RECORD
      IDENTTIPOROTTIPOS : STRING[20];
      NOMEROTTIPOS      : STRING[20];
    END;

```

Exemplo:

```

  IDENTTIPOROTTIPOS = CEP
  NOMEROTTIPOS = VERIFICA-CEP

```

3.1.9 Ações de Apresentação de Formulários:

A entidade APRAPRES cataloga as ações de apresentação a nível de formulário, ou seja, as formatações a serem executadas sobre todo o formulário.

O formato PASCAL que representa uma entrada na entidade APRAPRES é mostrada abaixo.

```

TYPE
  TIPOAPRAPRES =
    RECORD
      NOMEACAOAPRES : STRING[20];
      TITROTACAOAPRES : STRING[20];
    END;

```

Exemplo:

```
NOMEACAOAPRES = APRESENTA-CADASTRAM
TITROTACAOAPRES = B:APRESCAD.COM
```

A entidade APRNODOS cataloga as ações de apresentação a nível de nodos, ou seja, a nível de cabeçalhos, de textos, de grupos de campos e de campos de uma apresentação.

O formato PASCAL abaixo representa uma entrada na entidade APRNODOS:

```
TYPE
TIPOAPRNODOS =
    RECORD
        NOMEACAONODO : STRING[20];
        TITROTACAONODO : STRING[20];
    END;
```

Exemplo:

```
NOMEACAONODO = APRESENTA-CPF
TITROTACAONODO = B:APRESCPF.COM
```

A entidade APRTIPOS cataloga as ações de apresentação a nível de tipo de dados. Esta associação traz ao modelo de dados uma característica extremamente vantajosa em sua construção. O analista de formulários tem condições de especificar que um determinado tipo (ou tipos) de dados possua um mesmo conjunto de regras de formatação independente do formulário a que pertençam. Para valores monetários, exemplo típico, pode ser especificado que a formatação desejada seja com ponto entre milhares, vírgula decimal, duas casas decimais, dez dígitos para a parte inteira, cifrão flutuante sobre os zeros não significativos e sinal à esquerda do campo. Independente dos formulários a que pertençam, a apresentação de valores monetários sempre obedecerá esse padrão.

O formato PASCAL abaixo ilustra uma entrada na estrutura APRTIPOS.

```

TYPE
  TIPOAPRTIPOS =
    RECORD
      NOMEACAOTIPO = STRING[20];
      TITROTACAOTIPO = STRING[20];
    END;

```

Exemplo:

```

  NOMEACAOTIPO = APRESENTA-VALOR
  TITROTACAOTIPO = B:APRESVAL.COM

```

3.1.10 Entidade OPERS:

A entidade OPERS cataloga as operações próprias de cada tipo. Estes operadores tem singular importancia na medida em que para o tipo definido devem ser usados operadores próprios para manipulação. Por exemplo, na definição de um tipo de dados latitude, a soma de latitudes não pode ser executada pelo operador soma normal para inteiros pois o sistema de representação de latitudes não é o decimal e sim o sexagesimal. Portanto, para ser possível efetuar a soma de duas latitudes é necessário definir uma função específica.

A definição de operadores de tipos derivados é feita no contexto da IDF (interface de definição de formulários), de forma conversacional. A programação dessa função fica a cargo do analista de formulários, se já não estiver pronta no conjunto inicial de tipos do sistema. O DD mantém o catálogo das diversas operações e a associação delas com os tipos de dados.

O formato PASCAL que representa uma entrada na entidade OPERS é mostrada abaixo.

TYPE

```

TIPOOPERS =
  RECORD
    IDENTTIPOOPERS : STRING[20];
    SIMBOLOOPERS   : STRING[3];
    NOMEROTINAOPERS : STRING[20];
  END;

```

Exemplo:

```

IDENTTIPOOPERS = LATITUDE
SIMBOLOOPERS = +
NOMEROTINAOPERS = SOMALATITUDES

```

3.1.11 Entidade ESTACOES:

A entidade ESTACOES contém a definição da estação de trabalho do usuário para o sistema. Contém o modelo da estação, o nome com o qual é referenciado dentro do sistema e o nome do "driver" para efetivação da restituição, ou seja, para cada tipo de equipamento haverá um programa especializado em converter as especificações de formatação nos comandos de controle que as implementam no equipamento.

O formato PASCAL abaixo representa uma entrada na entidade ESTACOES.

```

TYPE TIPOESTACOES =
  RECORD
    NOMEESTACAO : STRING[20];
    MODELOESTACAO : STRING[20];
    NOMEDRIVER : STRING[20];
    DIMHORIZ : INTEGER;
    DIMVERT : INTEGER;
  END;

```

3.1.12 Entidade TABMENS:

A entidade TABMENS serve para manter as mensagens de erro que as diversas rotinas, a interface de manipulação e o servidor de arquivos, emitem em caso de detecção de erro.

O formato PASCAL abaixo representa uma entrada na entidade TABMENS.

```

TYPE
  TIPOTABMENS =
    RECORD
      CODMENS      : INTEGER;
      MENSAGEM     : STRING[80];
    END;

```

Exemplo:

```

CODMENS = 27
MENSAGEM = Digito de Controle inválido no CPF

```

3.2 Primitivas de Acesso e Atualização:

Para acesso e atualização das estruturas do dicionário de dados existem primitivas que obedecem ao seguinte padrão de construção:

- para inclusão:


```
DDINCL(<parametro>,<estado>)
```
- para exclusão:


```
DDEXCL(<identif-form>,<estado>);
```
- para consulta:


```
DDCONS(<forma-acesso>,<parametro>,<estado>);
```

onde:

<estado> é uma variável inteira que serve para retornar se a operação foi executada corretamente e, em caso contrário, qual o erro acontecido;

<identif-form> é uma variável string contendo o nome do formulário a ser excluído;

<forma-acesso> é uma variável inteira que define como deve ser buscado o registro da entidade;

<parametro> é um record de PASCAL que contém a descrição dos registros das diversas entidades do dicionário de dados e é apresentado a seguir.

TYPE

```
TIPOREGPARAMETRO = (DDVAR, FORMS, CAMPOS, APRESENT,
                    TRANSAS, NODOS, TIPOS, TIPOS2, ROTSYS,
                    ROTFORMS, ROTCAMPS, ROTTIPOS, APRAPRES,
                    APRNODOS, APRTIPOS, OPERS, ESTACOES,
                    TABMENS);
```

TYPE

```
PARAMETROPRIMITIVA =
    RECORD
        CASE TIPOREG : TIPOREGPARAMETRO OF
            DDVAR      : ( REGDDVAR      : TIPODDVAR      );
            FORMS      : ( REGFORMS      : TIPOFORMS      );
            CAMPOS      : ( REGCAMPOS      : TIPOCAMPOS      );
            APRESENT    : ( REGAPRESENT    : TIPOAPRESENT    );
            TRANSAS     : ( REGTRANSAS     : TIPOTRANSAS     );
            NODOS       : ( REGNODOS       : TIPONODOS       );
            TIPOS        : ( REGTIPOS        : TIPOTIPOS        );
            TIPOS2       : ( REGTIPOS2       : TIPOTIPOS2       );
            ROTSYS       : ( REGROTSYS       : TIPOROTSYS       );
            ROTFORMS    : ( REGROTFORMS    : TIPOROTFORMS    );
            ROTCAMPS     : ( REGROTCAMPS     : TIPOROTCAMPS     );
            ROTTIPOS     : ( REGROTTIPOS     : TIPOROTTIPOS     );
```

```

APRAPRES : ( REGAPRAPRES : TIPOAPRAPRES );
APRNODOS : ( REGAPRNODOS : TIPOAPRNODOS );
APRTIPOS : ( REGAPRTIPOS : TIPOAPRTIPOS );
OPERS    : ( REGOPERS    : TIPOOPERS    );
ESTACOES : ( REGESTACOES : TIPOESTACOES );
TABMENS  : ( REGTABMENS  : TIPOTABMENS  );

```

```

END;

```

A primitiva de inclusão serve para incluir uma entrada na entidade selecionada. Os programas que fizerem uso desta primitiva devem obedecer as descrições de registros de cada, ou seja, uma definição de tipo poderá gerar uma série de chamadas na primitiva de inclusão: a definição de um tipo enumeração faz com que fique na entidade TIPOS o identificador do tipo e, na estrutura TIPOS2, uma entrada para cada elemento da enumeração.

A primitiva de exclusão exclui o formulário e todas as definições associadas univocamente. Portanto, a chamada desta primitiva fará com que também sejam excluídas as entradas em CAMPOS, em APRESENT, em ROTFORMS e em NODOS. Não serão excluídas as entradas em TIPOS e as entidades vinculadas.

A primitiva de consulta permite acessar ocorrências nas diversas estruturas. A <forma-acesso> define como será buscado o registro: se valer 1, significa que deve ser buscado o registro que tenha os campos que vieram preenchidos no <parametro> iguais aos do registro; se valer 2, deve ser buscado o primeiro registro a seguir na sequência daquele que possui os mesmos valores nos campos que vieram preenchidos.

As entidades ROTSYS, ROTFORMS, ROTCAMPS, ROTTIPOS e OPERS dizem respeito à dissertação sobre o servidor de formulários, assim como a definição do posicionamento do dado no registro do formulário na estrutura CAMPOS. Foram tratados nesta dissertação para apresentar uma visão de conjunto do dicionário de dados.

4 SEMÂNTICA DE APRESENTAÇÃO:

A especificação das formas de associar semântica de apresentação no objeto formulário e disponibilizá-las a nível de interface de manipulação é o principal objetivo deste trabalho.

Nos sistemas atuais de bancos de dados (orientados a registros), a semântica da realidade está incluída nos programas de aplicação, quer quanto a conteúdo dos campos, quer quanto à apresentação. Isto é devido ao modelo de dados que os SGBDs atuais se baseiam. Normalmente, os tipos de dados disponíveis são inteiros, reais e cadeias de caracteres. Em alguns casos são disponibilizados mecanismos para verificação de conteúdos de campos e fechamento de valores entre campos. A complementação dos mecanismos, necessária para manter a fidelidade na representação da realidade pelo modelo, é suprida pelos aplicativos.

O termo semântica tem um sentido muito amplo quando usado isoladamente. A conotação de semântica que é usada nesta dissertação é a da associação ao objeto formulário da parcela de conhecimento presente no escritório. Como consequência, o objetivo proposto é que o modelo de dados para formulário reflita da forma mais fiel possível a realidade do escritório e, para tanto, foi proposto um modelo de dados orientado à objeto com mecanismos para integrar ao modelo uma parcela da realidade a ser representada.

O modelo de formulário é composto por quatro níveis de semântica: estrutura, apresentação, ações e atividades. Para o contexto da formatação é relevante a estrutura do formulário e a apresentação. Isto é uma consequência de nela serem definidos os atributos do formulário que serão manipulados pelo gerenciador de apresentação para os diversos "layouts" do formulário e as diretivas de apresentação necessárias.

Cada "layout" corresponde a uma visão que, normalmente, está associada a uma etapa do percurso do formulário dentro do

escritório. Por exemplo, a atividade de locar um veículo em uma autolocadora pode ser dividida em duas etapas:

- locação de veículo (saída do carro) e
- devolução do veículo (retorno do carro e cobrança da locação).

O formulário é um só, associado à atividade de locar um veículo. As visões são: locação e devolução. As figuras 6 e 7 ilustram estas visões. As duas apresentações, se fossem realizadas em papel, estariam superpostas e seriam preenchidas com carbono entre elas. A primeira via seria o documento de saída do veículo e serviria ao locatário, para efeitos legais, como o comprovante de locação. A segunda via, seria o controle da locadora para veículos que estão locados e o documento para cobrança da locação. Na devolução do veículo, a locadora preencheria os dados referentes à devolução, efetuaria a cobrança e passaria recibo do valor pago na via do locatário.

4.1 A Apresentação como Semântica do Modelo:

Nos sistemas convencionais, a forma de impressão do formulário reveste-se de grande importância pois objetiva o fácil entendimento e rapidez no preenchimento. Seu "layout" é definido a partir dos dados a serem manipulados e é orientado por uma série de regras adotadas pelas organizações junto com regras estéticas que levam a uma aparência agradável e que propiciam uma utilização facilitada.

A elaboração desse "layout" é dividida em duas partes: a formatação interna ao campo e a formatação do formulário. A formatação do campo depende do tipo de informação que o mesmo deve representar. Os campos inteiros possuem, normalmente, a indicação do número de posições; os reais, além do número de posições, possuem a indicação da posição da vírgula decimal; os campos alfanuméricos, a indicação do número de posições.

Locação de Veículo

Código Veículo Código Cliente

Data de Saída / / : : Odômetro de Saída

Opção Seguro Dados Pessoais Seguro

Data Devolução Prevista / / : : Caução

Figura 6 LOCAÇÃO DE VEÍCULO

Devolução de Veículo	
Código Veículo <input type="text"/>	Código Cliente <input type="text"/>
Data de Saída <input type="text" value=" / / : :"/>	Odômetro de Saída <input type="text"/>
Opção Seguro <input type="checkbox"/>	Dados Pessoais Seguro <input type="checkbox"/>
Data Devolução Prevista <input type="text" value=" / / : :"/>	Caução <input type="text" value=" . . . ,"/>
Data Devolução <input type="text" value=" / / : :"/>	Odômetro de Chegada <input type="text"/>
Litros de Combustível <input type="text"/>	Valor Combustível <input type="text" value=" . . . ,"/>
Km Rodados <input type="text"/>	Valor Km Rodados <input type="text" value=" . . . ,"/>
Número de Dias Locados <input type="text"/>	Valor das Diárias <input type="text" value=" . . . ,"/>
	Valor a Pagar <input type="text" value=" . . . ,"/>
	Valor Líquido <input type="text" value=" + . . . ,"/>

Figura 7 DEVOLUÇÃO DE VEÍCULO

Exceto no caso de códigos com número fixo de posições, os campos numéricos são sistematicamente preenchidos com alinhamento pela direita. Os campos alfanuméricos não possuem um padrão fixo sobre o alinhamento dos valores. Os valores podem estar alinhados pela esquerda, direita ou centralizado no espaço disponível. Além disso, campos de maior importância podem ter cores diferentes, serem hachurados ou serem de proporções maiores aos restantes.

A nível de formulário, as considerações quanto à formatação, em uma folha de papel, são mais abrangentes. Associado a cada campo está um título que, em uma palavra ou em uma expressão, indica o tipo de informação usada para preenchê-lo. Eventualmente, se o campo é codificado, o significado da codificação é indicado (ou ao lado do campo ou no rodapé do formulário).

De acordo com sua vinculação lógica, os campos são dispostos no plano de apresentação (folha de papel). Para realçar prioridades, indicar obrigatoriedade e vinculações no preenchimento, são usados mecanismos como cores, molduras, diferentes tamanhos, disposições especiais no plano, etc.

Estas características não podem ser dissociadas do formulário. Como mostrado na figura 8, a falta de formatação torna impraticável o manuseio dos formulários. Na figura (8-a), o formulário é visto com toda sua riqueza de informações proporcionadas pelos mecanismos de representação. Na figura (8-b), é mostrado o armazenamento dos campos em um arquivo, ou seja, o formulário é a imagem da forma de armazenamento dos campos no banco de dados.

Outro aspecto importante é realçado na figura 9 onde em (9-a) é representado um conjunto de campos com um "layout" e em (9-b) é representado o mesmo conjunto de campos com um "layout" bem diverso. A disposição dos campos no banco de dados é igual pois possuem a mesma ordenação, os tamanhos e os tipos dos campos são equivalentes.

Cadastro de Cliente - Pessoa Física			
Código	<input type="text"/>	Nome (pessoa física)	<input type="text"/>
Data de Nascimento	<input type="text"/> / <input type="text"/> / <input type="text"/>		
Carteira de Identidade	<input type="text"/>	CPF	<input type="text"/> - <input type="text"/>
Endereço (Rua, Av., compl.)	<input type="text"/>		
Bairro	<input type="text"/>	CEP	<input type="text"/>
		Município	<input type="text"/>
U.F.	<input type="checkbox"/>	Telefone	<input type="text"/>
Profissão	<input type="text"/>	Rendimentos do Cliente	\$ <input type="text"/>
Local de Trabalho	<input type="text"/>		
Endereço	<input type="text"/>		
Bairro	<input type="text"/>	CEP	<input type="text"/>
		Município	<input type="text"/>
		U.F.	<input type="checkbox"/>
Telefone	<input type="text"/>	Ramal	<input type="text"/>
		Cargo do Cliente	<input type="text"/>

a) Apresentação PESSOA-FÍSICA do formulário FORM1

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	69
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	140
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	196
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	236
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	313
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	346

b) Disposição dos campos em um registro do formulário FORM1

Figura 8 CADASTRO DE CLIENTE - PESSOA FÍSICA

Cadastro de Pessoas Jurídicas			
Código	<input type="text"/>		
Nome (pessoa jurídica)	<input type="text"/>		
Razão Social	<input type="text"/>		
Diretor Responsável	<input type="text"/>		
Endereço (Rua, Av., compl.)	<input type="text"/>		
Bairro	<input type="text"/>	CEP	<input type="text"/>
		Município	<input type="text"/>
			U.F. <input type="checkbox"/>
Telefone	<input type="text"/>	CGC	<input type="text"/> / <input type="text"/> - <input type="text"/>

a) Apresentação PESSOA - JURÍDICA do formulário FORM2

Dados Cadastrais do Servidor			
Código	<input type="text"/>		
Nome do Servidor	<input type="text"/>		
Filiação - Pai	<input type="text"/>		
- Mãe	<input type="text"/>		
Endereço (Rua, Av., compl.)	<input type="text"/>		
Bairro	<input type="text"/>	CEP	<input type="text"/>
		Município	<input type="text"/>
			U.F. <input type="checkbox"/>
Telefone	<input type="text"/>	Data Nasc.	<input type="text"/> / <input type="text"/> / <input type="text"/> : <input type="text"/> : <input type="text"/>

b) Possível apresentação alternativa de FORM2

Figura 9 FORM2 E APRESENTAÇÃO ALTERNATIVA

Portanto, elaborando a mesma representação da figura 8 para (9-a) e (9-b) é obtido o mesmo resultado. A diferença está justamente na forma de apresentar os campos a quem vai manipulá-los.

Estes exemplos demonstram que os campos de um formulário não podem ser dissociados dos mecanismos de apresentação definidos. Diferentes definições de apresentação podem implicar em interpretações e formas de preenchimento completamente diversas.

Na figura (9-b), o terceiro campo está titulado com "Filiacao - Pai" enquanto que na figura (9-a), o campo correspondente está titulado com "Razao Social". Em (9-b), é mais natural preencher o campo com o nome de uma pessoa, possivelmente com o do pai de um funcionário de alguma empresa. Em (9-a), é mais natural preencher o campo com o nome de uma empresa, provavelmente uma prestadora de serviços ou cliente.

O número de posições do campo é o mesmo: 40 caracteres. E' evidente, portanto, que informações diferenciadas tendem a uma interpretação diferenciada no contexto da apresentação do formulário. Mesmo o usuário sabendo que o formulário é usado para cadastrar empresas, o fato da apresentação do mesmo ser a da figura 8 - b certamente fará com que, no mínimo, o usuário se questione se está utilizando o formulário correto.

Portanto, a apresentação faz parte da semântica do campo e da semântica do formulário como um todo. Em consequência, o modelo de dados deve prover mecanismos para incorporar esta semântica nos diversos níveis de representação e o modelo proposto neste projeto tem por objetivo principal disponibilizar estes mecanismos.

4.2 Mecanismos de Visões ("VIEWS"):

Como foi apresentado no início deste trabalho, a simples

transposição de formulários para microcomputadores (que, num primeiro momento, satisfaz grande parte da comunidade de usuários) leva a um pequeno ganho comparado às possibilidades de aproveitamento das novas potencialidades.

A engenharia de software muito tem contribuído para que a migração dos sistemas para equipamentos de PED não se atenham somente às vantagens intrínsecas dos computadores como alta capacidade de armazenamento e processamento. As formas de conceber os sistemas tem se tornado cada vez mais acessíveis pela proposição e conseqüente implementação de ferramentas automatizadas para análise.

Além disso, pelas próprias necessidades de ferramentas de software que garantam a qualidade dos dados contra falhas do equipamento, o usuário tem à disposição uma série de produtos que atendem grandes parcelas do seu sistema em particular. Ou seja, o usuário tem um ambiente aparelhado de desenvolvimento de sistemas necessitando, somente, de elaborar seu fluxo de informações dentro do ambiente e das formas de interação do sistema com o usuário.

Em sistemas de formulários esta necessidade torna-se mais evidente. Nos sistemas convencionais, o formulário é usado como elemento de entrada de informações ao sistema, quer seja ele manual, quer seja ele automatizado.

Num sistema integrado de formulários, o formulário passa a ser o elemento único de tratamento dentro do sistema. É sobre o mesmo que será feito o trabalho de transformação e interpretação das informações. Portanto, o formulário passa a ter um significado mais amplo no ambiente de escritórios.

Fazendo analogia com sistemas convencionais, o formulário em si passa a ser o depositário de informações do sistema, ou seja, passa a também abranger a função de arquivo de informações. Isto é uma conseqüência da orientação para objeto do servidor de banco

de dados.

Sendo o formulário o elemento único de entrada e tratamento de informações, este passará por diversas etapas de manipulação. O formulário recebe um conjunto inicial de dados, fornecidos por um funcionário do escritório. A este momento chama-se etapa inicial e a tarefa realizada é a de inicialização de uma instância de um formulário. Nas etapas seguintes, o formulário servirá de mecanismo para transformação de informações, elemento para apropriação de mais dados ou como objeto fornecedor de informações para outros formulários ou pessoas.

As atividades de escritório são distribuídas entre as diversas pessoas que as executam. Isto significa que cada pessoa tem um determinado conjunto de tarefas a realizar. Portanto, é extremamente desejável que cada pessoa manipule somente o conjunto de campos, do formulário, que lhe diz respeito. Como consequência, é necessário um "layout" próprio para cada etapa do percurso do formulário dentro do escritório.

A vantagem advinda deste enfoque é permitir a cada usuário uma apresentação voltada às suas necessidades. O sistema, como um todo, terá, assim, um mecanismo de segurança eficiente para garantir a qualidade dos dados.

Cada formulário, por consequência, tem associada uma apresentação para cada etapa de sua manipulação, ou seja, um mesmo formulário tem uma semântica de apresentação vinculada a cada usuário.

É importante realçar que um usuário não é necessariamente uma pessoa do escritório. O usuário pode ser visto como sendo a pessoa que efetiva o agrupamento de tarefas a serem realizadas por um mesmo "tipo funcionário" (secretário, agente administrativo, técnico em contabilidade, etc). Distinção semelhante foi proposta em [SAN 86] e [TSI 82].

Este "tipo funcionário" caracteriza a pessoa ou grupo de pessoas que deve realizar as tarefas. Dependendo do volume de trabalho, importância, organograma da empresa e outros critérios, um "tipo funcionário" pode estar associado a uma pessoa, a um conjunto de pessoas, ou até uma pessoa pode acumular as atividades de vários "tipos funcionários".

A caracterização das atividades e dos tipos funcionários é feita na elaboração do sistema. Nesta fase é definido o modelo lógico do escritório que é composto do fluxo de informações, a caracterização das diversas atividades e dos tipos funcionários, ou seja, a definição do perfil dos diversos usuários do sistema no ambiente do escritório.

Portanto, cada apresentação disponível a um usuário é uma visão ("view") de um formulário e, como consequência, possui uma semântica própria de apresentação, ou seja, possui um padrão próprio de uso das técnicas de formatação para direcionar o preenchimento para a forma mais adequada.

Os aspectos mais relevantes que podem ser obtidos pelos mecanismos de apresentação são: a variação dos mecanismos de vinculação entre os campos e realces e a possibilidade de limitar o acesso a determinados campos em algumas apresentações. Esta limitação é disponibilizada de duas formas: campos não pertencerem a uma apresentação ou campos estarem presentes na apresentação sem que possam ter seus conteúdos modificados (apenas para consulta).

A definição de quais visões estarão disponíveis para quais usuários é obtida na análise do fluxo de informações do escritório (análise do sistema) e é discutido seu contexto no capítulo 2.

4.3 Necessidades a Atender:

Tendo sido caracterizado que apresentação faz parte da semântica do objeto, um estudo das necessidades a atender em termos de apresentação do formulário se faz necessário.

O estado da arte em tecnologias para microcomputadores e monitores de vídeo tem se caracterizado por uma diversidade muito grande em termos de possibilidades para gerar imagens. A resolução nos monitores tem sido um parâmetro que evoluiu muito. Hoje é possível ver-se imagens produzidas por estações gráficas com perfeição de imagem comparável às geradas pelas emissoras de televisão.

Isto influencia, e muito, o trabalho de caracterização das necessidades a atender em formatação de formulários porque, a continuar a evolução nesta área, em pouco tempo equipamentos com essas características estarão acessíveis à comunidade em geral.

Portanto, o objeto formulário deve ser versátil o suficiente para não tornar-se obsoleto frente aos novos desenvolvimentos tecnológicos, ou seja, deve suprir mecanismos capazes de permitir a incorporação de novos recursos de formatação às definições de formatação dos formulários.

Até bem pouco tempo atrás, os monitores de vídeo estavam limitados à representação de caracteres em termos de 12 x 40 (linhas vs. colunas), 24 x 40 e 24 (ou 25) x 80. Para gráficos, em torno de 20.000 pontos. Hoje tem-se disponível monitores coloridos com mais de 100.000 pontos e com representação de caracteres gráficos.

A limitação principal não se encontra na tecnologia. A limitação se encontra na capacidade de percepção e tratamento, por parte do ser humano, das imagens. Em consequência, a representação de caracteres em monitores que permita uma manipulação rápida e facilitada por parte do usuário não supera

em muito as tradicionais 24 linhas por 80 colunas. Ou seja, a apresentação de formulários muito extensos em uma só tela pode não ser a melhor solução.

Tomando, por exemplo, uma folha de jornal, mesmo que seja no formato tablóide, inteira em uma tela de monitor comum (12 polegadas), mesmo que a resolução do equipamento permitisse reproduzir com fidelidade o conteúdo da folha, a pessoa que estiver querendo "ler" a folha irá certamente precisar de, pelo menos, uma lupa já que uma folha tem por volta de 100 linhas por 160 colunas.

Em contrapartida, o crescimento da resolução e a implementação de novos recursos tais como novos tipos de caracteres, tamanhos diferenciados de campos em uma mesma tela, cores de fundo e de caracteres, intensidades diferenciadas, molduras em alta resolução entre outras, abre um leque de alternativas extremamente amplo para a elaboração das apresentações.

Como consequência, a interface de definição deve ser versátil para possibilitar a agregação de novas características e torná-las disponíveis para o analista de formulários.

Outra necessidade importante a atender em termos de apresentação é a vinculação hierárquica entre os campos. Numa abordagem inicial pode parecer que esta vinculação pertence à definição da estrutura do formulário. No entanto, como a presença de todos os campos nas diversas apresentações não é obrigatória, nas formatações ficariam descaracterizadas essas vinculações. Portanto, um mecanismo de hierarquização de campos se faz necessário na especificação das apresentações [BDL 84].

Por mais versátil que seja a ferramenta de software para definição, armazenamento e manipulação de formulários sempre haverá alguma parcela de semântica não passível de ser representada diretamente no objeto. É importante dispor de

mecanismos de ajuda (help) para permitir ao analista a redação de orientações adicionais ao usuário, associadas aos campos ou associadas ao formulário como um todo.

4.4 Ações de Apresentação:

De forma semelhante à associação de semântica de conteúdo ao objeto formulário, a associação de semântica de apresentação é feita em diversos níveis.

Foi proposta uma extensão da definição de tipos do PASCAL como forma de definir o objeto formulário. Esta extensão é dividida em dois contextos: definição de tipos de dados e definição do tipo de objeto formulário.

Na definição dos tipos de dados do PASCAL, foi agregado um mecanismo para associar um identificador que corresponde à uma ação semântica de apresentação que é usada, pela interface de manipulação, para a formatação.

Exemplo:

```
type VALOR = real, APRESENTA-VALOR; <== ação de apresentação a
                                     nível de tipo de dados
```

Na definição do tipo formulário, a semântica de apresentação pode ser associada em três níveis: nível de campo, nível de grupo de campos e nível de apresentações do formulário. Diferentemente da ação de apresentação associada a nível de tipo de dados, a ação de apresentação associada ao tipo formulário está vinculada à operação sendo executada. Para cada apresentação, existe um rol de campos associados com suas respectivas ações de apresentação, e um conjunto de transações passíveis de serem executadas pelo usuário. Estas transações, identificadas na análise do fluxo de informações do escritório, são mapeadas nas operações básicas do servidor de formulários.

Para um determinado tipo formulário, pode ser definido um conjunto de apresentações. Cada uma delas é associada a um identificador e a um rol de ações de apresentação. Para cada apresentação são relacionados os campos que dela fazem parte (com suas ações de apresentação convenientes) e, caso conveniente, definidas as vinculações hierárquicas entre os campos através do mecanismo "grupo ... fim". Cada agrupamento de campos também pode ter associadas ações de apresentação.

A nível de campo, a ação de apresentação é definida a seguir do identificador do campo. A nível de grupo de campos, a ação é definida após a palavra reservada "grupo". A nível de apresentação, as ações são definidas logo após o identificador da apresentação.

Exemplo:

```

type SIMNAO = (S,N);
type OPCSEG = (1,2,3);
type
  FORM4(Locacoes) =
    formulario
      estrutura
        CODVEI-LOC(Codigo Veiculo):inteiro,chave,VERIFCODVEICULO(sem);
        CODCLI-LOC(Codigo Cliente):inteiro,chave,VERIFCODCLIENTE(sem);
        { - ações semânticas de conteúdo.
          São executadas no contexto definido:
            'sem' }
        DATASAI-LOC(Data de Saida):tempo;
        ODOMSAI-LOC(Odometro de Saida):inteiro;
        OPCSEG-LOC(Opcao Seguro):OPCSEG;
        DANOSPESS-LOC(Danos Pessoais Seguro):SIMNAO;
        DEVPREV-LOC(Data Devolucao Prevista):tempo;
        DEVOL-LOC(Data Devolucao):tempo;
        ODOMCHEG-LOC(Odometro de Chegada):inteiro;
        LITRCOMB-LOC(Litros de Combustivel):real;
        VALCOMB-LOC(Valor Combustivel):VALOR,CALCVCOMB-LOC(sem);

```

```

KMROD-LOC(Km Rodados):inteiro,CALCKM-LOC(sem);
VALKMROD-LOC(Valor Km Rodados):VALOR,CALCVKMROD-LOC(sem);
NDIAS-LOC(Numero de Dias Locados):inteiro,CALCNDIAS-LOC(sem);
VALNDIAS-LOC(Valor das Diarias):VALOR,CALCVDIAR-LOC(sem);
TOTAL-LOC(Valor a Pagar):VALOR,CALCVTOT-LOC(sem);
fim;
apresentacao
SAIDA-VEICULO:
visao,
    APRESENTA-SAIDA; <== ação de apresentação a nível de
                        formulário
transacoes
    SAIDA-VEIC(Saida de Veiculo com Cliente);
    ALT-SAIDA-VEIC(Correcao das Informacoes sobre Saida de
Veiculo)
    fim;
    globais
        cabecalho(Locacao de Veiculo),APRESENTA-CABEC;
                                <== ação de apresentação
                                a nível de cabeçalho
        #Este Formulario serve para Registrar a Saida de um
Veiculo Locado#
    fim;
    grupo, APRESENT-MOLDURA; <==+ ação de apresentação a
        nodo CODVEI-LOC;                | nível de grupo de campos
        nodo CODCLI-LOC;                |
    fim;                                |
    grupo, APRESENT-MOLDURA; <==+
        nodo DATASAI-LOC,APRESENTA-TEMPO; <== ação de apresentação
        nodo ODOMSAI-LOC;                a nível de campos
        nodo OPCSEG-LOC;
        nodo DANOSPESS-LOC;
        nodo DEVPREV-LOC,APRESENTA-TEMPO;
        nodo CAUCAO-LOC
    fim;
fim;
DEVOLUCAO-VEICULO:

```

```

visao,
  APRESENTA-DEVOLUCAO;
transacoes
  DEVOL-VEIC(Devolucao de Veiculo pelo Cliente e Cobranca);
  ALT-DEVOL-VEIC(Correcao das Informacoes sobre Devolucao de
Veiculo)

```

```

{ - as transações são associadas às
apresentações pois estas é que
chegam para manipulação pelo
usuário. A manipulação do formulário
pode ser obtida selecionando, pela
interface, a transação desejada. }

```

```

fim;
globais
  cabecalho(Devolucao de Veiculo),APRESENTA-CABEC;
  #Este Formulario serve para Registrar a Devolucao de um
Veiculo e para a Cobranca do Valor da Locacao#

```

```

{ - este texto de ajuda vinculado ao
cabecalho é apresentado pela
interface de manipulação ao ser
solicitada ajuda com o cursor
posicionado dentro da área de
cabecalho do formulário. }

```

```

fim;

```

```

grupo, APRESENT-MOLDURA;
  nodo CODVEI-LOC;
  nodo CODCLI-LOC;
fim;
grupo, APRES-MOLDURA-LEIT;
  nodo DATASAI-LOC,APRESENTA-TEMPO;
  nodo ODOMSAI-LOC;
  nodo OPCSEG-LOC;
  nodo DANOSPESS-LOC;
  nodo DEVPREV-LOC,APRESENTA-TEMPO;
  nodo CAUCAO-LOC

```

```

{ - para esta apresentação está

```

associada uma ação semântica de conteúdo que, além de definir moldura para este grupo de campos, especifica que eles poderão somente serem consultados. }

```

fim;
grupo, APRESENTA-MOLDURA;
  nodo DEVOL-LOC,APRESENTA-TEMPO;
  nodo ODOMCHEG-LOC;
  nodo LITRCOMB-LOC;
  nodo VALCOMB-LOC;
  nodo KMROD-LOC;
  nodo VALKMROD-LOC;
  nodo NDIAS-LOC;
  nodo VALNDIAS-LOC;
  nodo TOTAL-LOC;
  nodo TOTLIQ-LOC
  fim;
fim;
fim;
operacoes
  <def-operacoes>
  fim;
fim;

```

As ações de apresentação, como exemplificado, podem ser associadas nos diversos níveis do objeto: nível de campo, nível de grupo de campos e nível de formulário. A referência aos formulários e aos campos, pela interface de manipulação, não é feita pelos identificadores definidos: palavras ou expressões que caracterizam as informações sendo manipuladas no ambiente do escritório são vinculadas aos respectivos identificadores no objeto na sua definição.

Finalmente, a nível de campos e de formulários, podem ser associados textos para serem usados nos mecanismos de ajuda. A

interface permite ao usuário três formas de ajuda: a restituição da definição do campo, a apresentação do texto associado ao campo e, caso o campo seja chave de outro formulário, mecanismos para pesquisar valores já existentes. De forma semelhante é feito no contexto do formulário (figura 10). O mecanismo para pesquisa de valores é o sugerido em [KOV 86].

```

+-----+
| Locacao de Veiculo                                     |
+-----+
|
|   Codigo Veiculo _____   Codigo Cliente _____   |
|
+-----+
|   Data de Saida __/__/__ __:__:__   Odometro de Saida _____   |
|   Opcao Seguro #           Danos Pessoais Seguro _           |
|   Data Devolucao Prevista __/__/__ __:__:__           |
|   Caucao $____.____.____,____           |
|
+-----+
|
|>Ação: Ajuda Definicao                                     <
|>   OPCSEG-LOC(Opcao Seguro):OPCSEG;                       <<
|>   OPCSEG = (1,2,3);                                       <<<

```

figura 10-a Restituição da Definição

```

+-----+
| Locacao de Veiculo                                     |
+-----+
|
|   Codigo Veiculo _____   Codigo Cliente _____   |
|
+-----+
|   Data de Saida __/__/__ __:__:__   Odometro de Saida _____   |
|   Opcao Seguro #           Danos Pessoais Seguro _           |
|   Data Devolucao Prevista __/__/__ __:__:__           |
|   Caucao $____.____.____,____           |
|
+-----+
|
|>Ação: Ajuda Texto                                     <
|>=> 1 - Acidentes, 2 - Roubo, Incendio, 3 - Total          <==<

```

figura 10-b Texto Associado ao Campo

de cores, seja de fundo, seja do próprio conteúdo do campo pode não ser compatível com a definição de intensificado pois para o dispositivo, intensificado significa usar uma cor diferente.

Entretanto, grande parte das especificações de formatação não se sobrepõem. Um campo pode ser definido como tendo cor de fundo azul marinho, caracteres em amarelo, o tipo dos caracteres serem em itálicos, o campo ser emoldurado com a moldura piscante. Neste caso, foram usadas cinco características de formatação sem que uma interferisse na ação da outra.

Caso não seja encontrada especificação para alguma característica é assumido um valor pré-definido. Somente a nível de apresentação de formulários tem-se valores pré-definidos. A não especificação nos outros níveis não implica em assumir algum valor pré-definido.

Ou seja, a interface de manipulação, no momento de produzir a apresentação do formulário executa as diversas ações semânticas de apresentação sendo a especificação de tipo a que possui maior prioridade. Caso a ação de tipo não tenha especificado alguns dos tipos de formatação possíveis, a interface busca esta especificação na ação vinculada ao nodo, ao grupo de nodos e ao formulário. Se não houve especificação em nenhum dos níveis para uma determinada característica, a interface assume uma especificação pré-definida.

Sintaticamente, a descrição das visões de um formulário é feita na seção de **apresentação** no modelo formulário. Para cada definição de formatação há um identificador e uma lista de campos associados à formatação.

No exemplo abaixo, a <acao-apresentacao> é um identificador que associa a ação de apresentação ao elemento a ser formatado: campo, grupos de campos e formulário. Pode também ser associada ao tipo de dado.

```

type LOCACAO(Locacao de veiculos da autolocadora)=
  formulario
    estrutura
      COD-VEICULO(Codigo Veiculo):inteiro;
      COD-CLIENTE(Codigo Cliente):inteiro;
      ...
    fim;
  apresentacao
    LOCA-VEICULO:
      globais
        cabecalho(Locacao de Veiculo), <acao-apresentacao>;
      fim;
      nodo COD-VEICULO, <acao-apresentacao>;
      nodo COD-CLIENTE, <acao-apresentacao>;
      ...
    fim;
  ...
fim;

```

A parcela semântica não agregada ao tipo formulário é passível de ser especificada nos textos associados aos mecanismos de ajuda (help), ou seja, o analista de formulários redige textos explicativos para orientar o usuário no tratamento dos dados.

Concluindo, a semântica de apresentação especifica de que forma o formulário deve ser formatado. Esta especificação aborda vários aspectos, seja a nível de campo, seja a nível de formulário: o posicionamento no plano de representação, o tamanho, cores (dos dados, do fundo), tipos de caracteres (itálicos, letras góticas, etc), edições (vírgula decimal, ponto entre milhares, substituição de zeros não significativos por brancos, etc), tipo de acesso (só leitura ou não), títulos, textos para os mecanismos de ajuda, etc.

5 INTERFACE DE MANIPULAÇÃO:

A interface de manipulação é o resultado da ação do software que faz chegar o formulário ao usuário. Este componente é o responsável por fazer chegar, ao usuário, o tipo de dados com a semântica associada. Os mecanismos necessários para que o sistema garanta o acesso limitado ao usuário são supridos pela interface.

5.1 Formatação e Restituição:

Como foi mostrado anteriormente, as atividades de formatação e restituição pertencem a contextos diferentes. A formatação preocupa-se em obter a descrição do formulário do dicionário, selecionar os campos que fazem parte da apresentação a ser formatada e aplica as ações semânticas de apresentação de acordo como definido para obter a formatação ideal para a apresentação. A restituição preocupa-se em obter as características do dispositivo onde será apresentado o formulário do dicionário de dados e adaptar a imagem ideal produzida pela formatação às limitações próprias do equipamento.

A semântica a nível de apresentações é garantida pela própria interface. Ao receber uma ocorrência, é consultada a descrição da apresentação no dicionário de dados (DD), é produzido o formulário formatado e, pelas características do dispositivo de saída, criada a restituição.

As ações de apresentação, em caso de especificações conflitantes, são combinadas priorizando as ações de tipo, após as de campo, as de grupo de campos e, por fim, as de apresentação de formulário. Desta forma, o analista de formulários tem condições de definir uma apresentação única para um tipo. Independente dos formulários a que pertença e das visões onde apareça, o usuário irá "ver" o dado sempre da mesma forma. Este mecanismo é abordado com mais profundidade no capítulo anterior.

As transações que o usuário pode executar sobre uma visão de

um formulário são definidas no DD. As mesmas são mapeadas em operações básicas do servidor de formulários e são especificadas na fase de análise do fluxo de informações.

5.2 Interface com o Usuário:

A interface de manipulação chega ao usuário de duas formas: menus para seleção de atividades e apresentações de formulários. Os menus para seleção de atividades servem para ajudar o usuário a selecionar os formulários que deseja manipular, seja através das transações catalogadas pelo sistema, seja pela seleção do formulário. A manipulação de apresentações de formulários é o objetivo principal desta especificação e é o aspecto de maior interesse neste capítulo.

5.2.1 Acesso ao Sistema pelos Menus:

Como implementado no projeto FORMANAGER [YAO 84], a sistemática de utilização do sistema é baseada em menus e parte de um menu principal para seleção da atividade desejada. O acesso ao sistema deve, necessariamente, ser controlado por mecanismos de códigos de usuário e senhas para que somente pessoal autorizado possa acessar as informações. É comum os próprios equipamentos possuírem este mecanismo.

A proposição do projeto feita em [OLI 85] prevê a centralização do armazenamento das informações, dentro de um sistema em rede, em um servidor de arquivos. O projeto tem proposto que o servidor de formulários garanta o acesso autorizado dos formulários.

O menu principal do sistema é ilustrado na figura 11. As opções disponíveis são: formulários, transações e atividades sobre formulários.

A opção formulários permite selecionar a apresentação do

formulário desejada para manipular. No contexto do projeto, aparecem como passíveis de serem manipulados apenas os formulários que, na modelagem do sistema, foram especificados para estarem disponíveis numa dada estação para um determinado usuário (figura 12). Esta seleção é feita por código de usuário e por estação. Outro aspecto importante é que o usuário manipula, efetivamente, apresentações de formulários, apesar da analogia com sistemas convencionais levar a entender que o que é apresentado em si seja o formulário.

Caso seja solicitado pelo usuário, a interface produz a apresentação do formulário para que este possa ter certeza de que está selecionando o correto (figura 13). Esta restituição feita é apenas em termos de cabeçalhos, títulos de campos e espaço ocupado pelos campos com as respectivas ações de apresentação aplicadas. O espaço alocado para o conteúdo do campo fica vazio não implicando que este "vazio" corresponda ao valor assumido por omissão. Somente ao ser solicitada a inicialização de uma instância é que os campos vão aparecer com os conteúdos iniciais.

```
+-----+
|                                     |
|           Sistema de Formularios   |
|           Eletronicos              |
|                                     |
|           Menu Principal            |
|                                     |
|           - Formularios            |
|                                     |
|           - Transacoes             |
|                                     |
|           - Atividades             |
|                                     |
|           - Log Off                |
|                                     |
+-----+
```

Figura 11 Menu Principal

<p style="text-align: center;">Sistema de Formularios Eletronicos</p> <p style="text-align: center;">Selecao por Formularios</p> <p style="text-align: center;">_ Cadastro de Cliente - Pessoa Fisica</p> <p style="text-align: center;">_ Cadastro de Pessoa Juridica</p> <p style="text-align: center;">_ Cadastro de Veiculos</p> <p style="text-align: center;">_ Locacao de Veiculo</p> <p style="text-align: center;">_ Devolucao de Veiculo</p>
--

Figura 12 Seleção por Formulários

<p style="text-align: center;">Devolucao de Veiculo</p>												
<table border="1"> <tr> <td style="text-align: center;">Codigo Veiculo _____</td> <td style="text-align: center;">Codigo Cliente _____</td> </tr> </table>	Codigo Veiculo _____	Codigo Cliente _____										
Codigo Veiculo _____	Codigo Cliente _____											
<p>Data de Saida ___/___/___ __:__:__ Odometro de Saida _____</p> <p>Opcao Seguro _ Danos Pessoais Seguro _</p> <p style="text-align: center;">Data Devolucao Prevista ___/___/___ __:__:__</p> <p style="text-align: center;">Caucao \$_____</p>												
<table border="1"> <tr> <td style="text-align: center;">Data Devolucao ___/___/___ __:__:__</td> <td style="text-align: center;">Odometro Chegada _____</td> </tr> <tr> <td style="text-align: center;">Litros Combustivel _____</td> <td style="text-align: center;">Valor Combustivel \$_____</td> </tr> <tr> <td style="text-align: center;">Km Rodados _____</td> <td style="text-align: center;">Valor Km Rodados \$_____</td> </tr> <tr> <td style="text-align: center;">Nro Dias Locados _____</td> <td style="text-align: center;">Valor das Diarias \$_____</td> </tr> <tr> <td></td> <td style="text-align: center;">Valor a Pagar \$_____</td> </tr> <tr> <td></td> <td style="text-align: center;">Valor Liquido \$_____</td> </tr> </table>	Data Devolucao ___/___/___ __:__:__	Odometro Chegada _____	Litros Combustivel _____	Valor Combustivel \$_____	Km Rodados _____	Valor Km Rodados \$_____	Nro Dias Locados _____	Valor das Diarias \$_____		Valor a Pagar \$_____		Valor Liquido \$_____
Data Devolucao ___/___/___ __:__:__	Odometro Chegada _____											
Litros Combustivel _____	Valor Combustivel \$_____											
Km Rodados _____	Valor Km Rodados \$_____											
Nro Dias Locados _____	Valor das Diarias \$_____											
	Valor a Pagar \$_____											
	Valor Liquido \$_____											

Figura 13 Devolução de Veículo

A seleção por transações visa permitir ao usuário definir a atividade a realizar pesquisando as transações possíveis de serem feitas por ele, independente de formulários (figura 14). Pela transação selecionada, a interface recupera a apresentação do formulário associada.

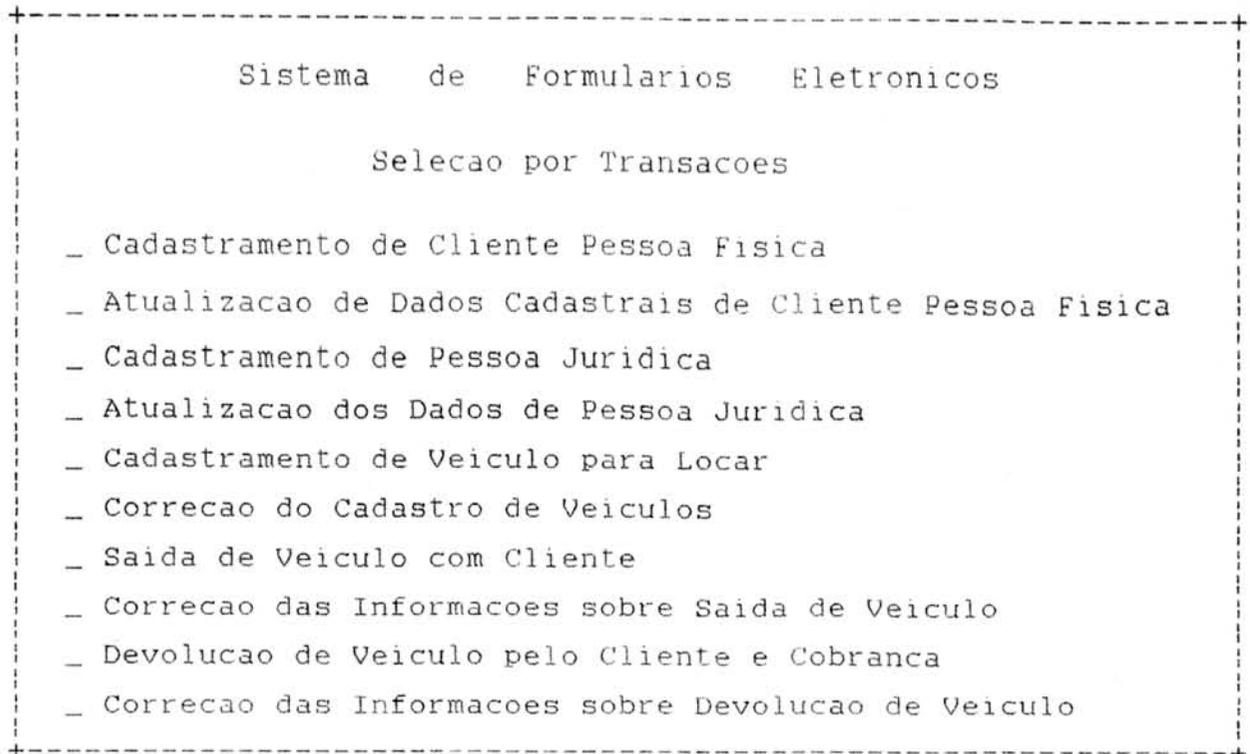


Figura 14 Seleção por Transações

A seleção por atividades sobre formulários permite o usuário selecionar as ações interformulários (corresponde as ocorrências da entidade ROTSIS do DD). No enfoque do projeto, a seleção de alguma das ações interformulários provoca a execução do programa correspondente que trabalha no contexto de todas as instâncias de todos os formulários. No entanto, o usuário não interage com este programa; ele manipula os produtos da execução acessando os formulários correspondentes. Ou seja, diferentemente da seleção por transações, a seleção por atividades não apresenta um formulário como resultado da opção escolhida e, inclusive, não bloqueia a estação para outras atividades. O servidor de formulários, ao término da execução da ação interformulários, comunica o seu término à estação que a acionou. Neste meio tempo, o usuário pode interagir com a interface para execução de outras tarefas locais.

Esta é uma característica resultante do fato que o projeto propõe-se a implementar as atividades estruturadas do escritório, ou seja, aquelas conhecidas de antemão. O acesso conversacional aos formulários será objeto de trabalhos posteriores.

5.2.2 Apresentações de formulários:

Após o usuário percorrer os menus para chegar ao formulário desejado, a interface produz a apresentação escolhida. O espectro que realmente aparece é definido na especificação do sistema. Dependendo do tamanho da apresentação do formulário e do dimensionamento de janela adotada, o formulário aparecerá completo ou não. Caso não apareça completo, a interface disponibiliza comandos para fazer com que a janela "percorra" o formulário.

O conteúdo de uma tela com formulário é ilustrado na figura 15. É dividida em dois contextos: área de apresentação e área de diálogo. Padrão semelhante foi implementado no projeto SPECDOQ divulgado em [KIT 84].

A área de apresentação é a zona onde a interface coloca a apresentação. A qualidade da apresentação depende das características do equipamento, tais como, restituição gráfica, colorida, com intensidades diferenciadas e outros mecanismos de restituição.

A área de diálogo fica localizada no rodapé da tela e, para este projeto, ocupa as quatro últimas linhas. Ela serve para comunicação entre a interface e o usuário: comandos de navegação entre os diversos menus, acionamento dos mecanismos de ajuda, transações disponíveis sobre a apresentação, comandos de controle para manipulação de formulários maiores que a janela, etc.

Devolucao de Veiculo	
Codigo Veiculo _____	Codigo Cliente _____
Data de Saida __/__/__ __:__:__	Odometro de Saida _____
Opcao Seguro _	Danos Pessoais Seguro _
Data Devolucao Prevista __/__/__ __:__:__	
Caucao \$_____	
Data Devolucao __/__/__ __:__:__	Odometro Chegada _____
Litros Combustivel _____	Valor Combustivel \$_____
Km Rodados _____	Valor Km Rodados \$_____
Nro Dias Locados _____	Valor das Diarias \$_____
	Valor a Pagar \$_____
	Valor Liquido \$_____

>>>>> AREA DE DIALOGO <<<<<<

Figura 15 Conteúdo de uma Tela com um Formulário

5.3 Interface com o GRD:

A comunicação entre a interface de manipulação e o servidor de formulários é um dos aspectos mais sensíveis dentro do trabalho de implementação. Esta comunicação, na fase atual de implementação, não está considerando os aspectos de comunicação entre módulos em redes. A implementação sugerida em [OLI 85] prevê diversas estações de trabalho com um servidor de arquivos e um servidor de impressão (figura 16).

Em cada estação de trabalho executa uma cópia da interface de manipulação com as autorizações e poderes definidos na fase de definição do sistema. Os formulários ficam centralizados no servidor junto com o dicionário de dados. Considerando que novas

definições não devem implicar na interrupção do uso dos formulários pelos usuários, a interface de definição garante a integridade do dicionário com mecanismos de versões de formulários e do próprio dicionário para manter a compatibilidade do sistema.

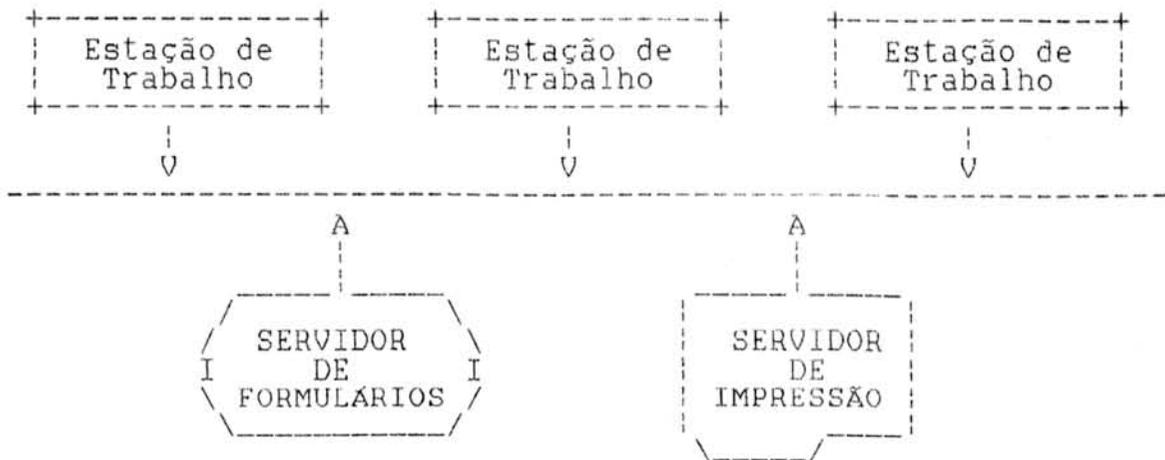


Figura 16 Arquitetura do Sistema de Formulários em Redes

A centralização e controle dos formulários e do DD no servidor de arquivos segue as considerações apresentadas em [MOR 86], em relação à segurança e integridade de dados em uma rede de computadores. Estas considerações, implementadas no projeto ANDREW, levam em conta a diversidade de equipamentos conectados e a falta de garantia quanto a integridade e confiabilidade do equipamento e do usuário. A estação, na visão do projeto, é passível de ser corrompida por um usuário quanto ao seu conteúdo e funcionamento. Por consequência, é necessário que somente o estritamente necessário seja enviado para a estação do servidor de arquivos e que deve ser verificada a integridade das mensagens recebidas das estações.

A integridade das instâncias dos diversos formulários é garantida pela centralização no servidor do armazenamento. Portanto, mecanismos de chaveamento e acesso exclusivo à instância, estrutura ou conjunto de estruturas é garantida pelo servidor, de acordo com o tipo de atividade sendo executada pelo usuário. Por exemplo, na alteração de uma instância, apenas esta deve ficar inacessível ao resto da rede. Na execução de ações interformulários, o servidor verifica o contexto da execução da

ação e chaveia as estruturas acessadas.

A rede local adquirida para implementação do projeto tem como forma de comunicação entre as diversas estações a transferência de arquivos. Portanto, o mecanismo usado na atual proposta do sistema é a comunicação por arquivos (figura 17).

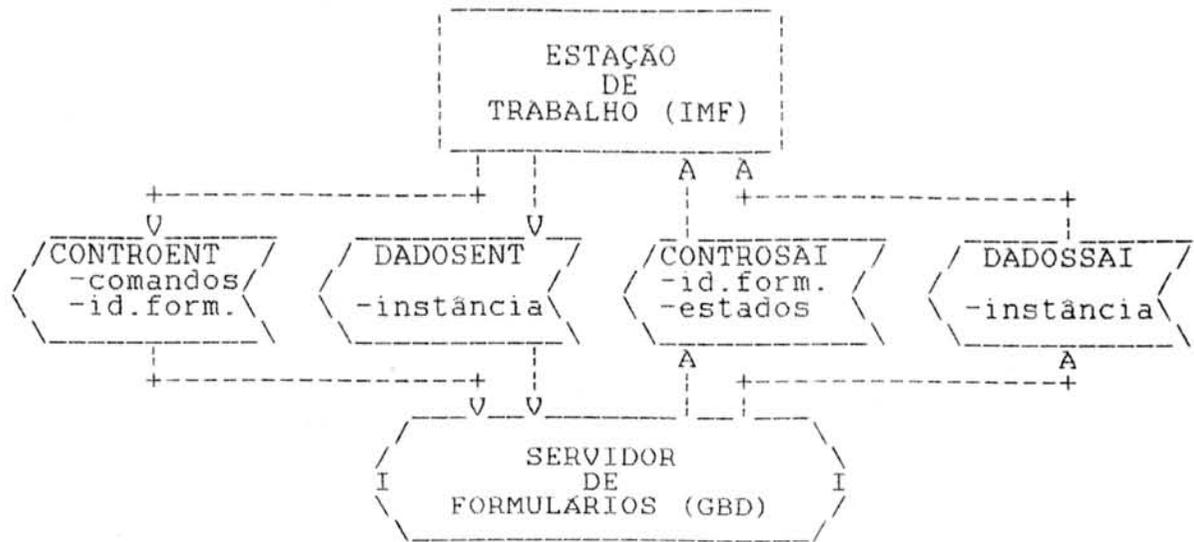


Figura 17 Comunicação IMF vs. GBD

O envio de mensagens da interface de manipulação para o servidor é feita por dois arquivos: um para comandos e outro para dados. O primeiro, chamado CONTROENT, contém os comandos da interface para o servidor e a identificação do formulário sendo manipulado. O segundo, chamado DADOSENT, contém uma instância do formulário para complementação dos comandos enviados.

No arquivo DADOSENT, a interface coloca uma instância do formulário sendo manipulado, de acordo com a distribuição de campos armazenada no DD, referente a posicionamento dentro de um registro, e o seu surrogate.

O envio de mensagens do servidor para a interface de manipulação também é feita por dois arquivos: CONTROSAI e DADOSSAI. No arquivo CONTROSAI o servidor coloca o identificador de qual formulário está manipulando e códigos de estados para

indicar se a operação solicitada foi executada corretamente e, caso negativo, códigos de erros para o caso de que campos tenham sido preenchidos com conteúdos inválidos ou outras inconsistências detectadas. No arquivo DADOSSAI, o servidor coloca uma instância de um formulário, caso a resposta do comando enviado assim o estipule, junto com seu surrogate.

Para o futuro funcionamento do sistema em rede, em todos os arquivos de comunicação os registros possuem o número da estação originária e um número sequencial de transação da estação.

Nem todas as mensagens que a interface envia para o servidor compõem-se de todas as informações. De acordo com o comando enviado, é utilizado o identificador do formulário e, conforme o caso, é enviada uma instância.

Os tipos de comandos que a interface envia para o servidor são: comandos de operação em uma ocorrência de um formulário, comandos para consulta, folheio e classificação de formulários, comandos para execução de ações interformulários e comandos de "logon" e "logoff".

Os comandos de operação em uma ocorrência de um formulário são os definidos no próprio objeto formulário. Na definição de um formulário, podem ser associadas operações a nível de formulário. O contexto de execução dessas operações são: "ini" para a inicialização de uma instância, "inc" para inclusão, na estrutura, de uma nova instância, "exc" para exclusão de uma instância, "alt" para modificação de uma instância, e "fin" para finalização de uma instância. A nível de objeto, na apresentação, é feita uma associação entre uma dessas operações básicas do servidor com transações do usuário.

Portanto, o usuário seleciona uma transação a ser efetuada em um formulário, a interface verifica em qual das operações a transação está mapeada e passa uma mensagem ao servidor com a operação, o identificador do formulário sendo manipulado e a instância a ser tratada pelo servidor, se for o caso.

Um aspecto importante no contexto dos comandos de operação é que o usuário, de alguma forma, deve localizar a instância sobre a qual deseja trabalhar. Em sistemas não integrados, a operação de inclusão sempre é efetivada a partir de um formulário em branco. Neste projeto, isto não é obrigatório. Se o usuário, por exemplo, necessitar incluir quinze veículos marca Chevrolet, modelo Chevette, ano 1987, cujas diferenças encontram-se nas cores, placas e números de chassis, é mais racional ser aproveitada a apresentação do modelo anterior, preenchendo os campos que diferem, para efetivar as inclusões.

Portanto, para efetivar uma inclusão não é necessária uma prévia inicialização de uma instância. Na alteração, entretanto, é necessária a localização de uma instância a manipular. Como consequência, antes da etapa de execução de comandos de operação sobre formulários há uma etapa de localização da instância desejada ou a inicialização de uma instância do formulário correspondente.

O surrogate da instância é gerado na sua inclusão. A interface assinala com valor nulo o surrogate para a inclusão. Na alteração, a interface exige uma instância associada à estação.

Em suma, a interface de manipulação mantém sempre a referência da última instância referenciada pelo usuário, mesmo que seja apenas uma inicialização. Isto é necessário porque o usuário, inicialmente, segue uma rota de menus para definir a instância a ser manipulada e, após, migra para outra rota para decidir qual operação deseja executar.

Os comandos para consulta, folheio e classificação de formulários tem por objetivo suprir as necessidades do usuário em termos de obter informações do sistema para sua decisão, seja a nível operacional, seja a nível gerencial. Para tanto, a interface permite ao usuário formas de selecionar instâncias a partir do próprio formulário, ou seja, através do preenchimento parcial de campos e da seleção da forma de pesquisa apropriada (tipo ocorrências que possuam valores iguais aos preenchidos ou

que possuam valores maiores que os preenchidos) a interface envia mensagem contendo a instância parcialmente preenchida e a forma de acesso desejada e o servidor cria uma tabela de ponteiros para as ocorrências que satisfazem a condição estabelecida. O mecanismo de acesso à instâncias pelo preenchimento parcial de formulários foi sugerido em [TSI 82].

Caso o usuário necessite de uma seleção por intervalo de valores de campos, a interface permite, via comando, este mecanismo de acesso pela construção de uma expressão booleana que traduza a consulta do usuário. Esta execução também produz a tabela de ponteiros para as instâncias que satisfazem a condição.

A partir daí, o usuário tem ao seu dispor comandos para "folhear" esta tabela: busque o primeiro, busque o último, busque o próximo e busque o anterior. O servidor mantém informações a respeito de qual estação pertence qual tabela e qual a ocorrência corrente.

A pesquisa na tabela também pode ser feita da forma de seleção de ocorrências. No comando de seleção, neste caso, é especificado que a seleção é feita no arquivo de ponteiros.

O usuário tem à disposição um comando para classificar uma estrutura, ou o arquivo de chaves. O resultado da classificação é a tabela de ponteiros na ordem desejada. A estrutura, na realidade, não é organizada na nova ordenação.

Para cada um dos campos especificados como 'chave' do formulário, o servidor mantém uma estrutura classificada para acesso das instâncias. Esta classificação faz com que a a tabela de ponteiros seja carregada com o índice correspondente, caso a classificação solicitada seja sobre uma estrutura e por um campo chave.

Os comandos de ações interformulários são enviados para o servidor pela seleção da ação desejada no menu de ações disponíveis. Como isso não implica na transferência da execução

da estação para o programa correspondente, o retorno dado pelo servidor é de finalização da execução solicitada. A partir daí, o usuário tem condições de usar os comandos de consulta e folheio de formulários para tomar conhecimento dos resultados da execução.

Os comandos de "logon" e "logoff" seguem os padrões comumente adotados para sistemas interativos. O servidor concentra a função de aceitar ou não o início de atividades com o sistema. E' objetivo do projeto que, no DD, esteja definido que tipo de uso pode ser feito pelo usuário ao sistema nos aspectos de quais apresentações de formulários pode manipular, quais ações inter-formulários pode disparar e se pode efetuar consultas aos formulários. Na atual proposta, isto não está sendo feito pois o módulo de definição e controle de fluxo de formulários dentro do escritório está em estudo.

5.4 Interface com o Dicionário de Dados:

A proposta de localização do DD no sistema distribuído em uma rede é de uma única cópia localizada junto ao servidor de arquivos. As estações que executam a interface de manipulação, acessam o DD apenas para consulta das descrições.

O acesso ao DD é feito usando as primitivas documentadas no capítulo 3 deste trabalho. Como a atividade é apenas consulta, é usada a primitiva DDCONS. A interface de manipulação, para poder produzir uma apresentação, acessa o DD e carrega a descrição do formulário e da apresentação em uso para uma área de memória. A apresentação é montada de acordo com a especificação das posições dos diversos campos no plano de representação e, após, são executadas as ações de apresentação associadas a apresentação selecionada do objeto. O mecanismo de atuação dessas ações foi comentado no capítulo 4. A execução dessas ações é feita considerando, as mesmas, subrotinas da interface. No DD, estão catalogados os nomes das rotinas no diretório em disco.

5.5 Interface com as Ações de Apresentação:

Na atual fase de desenvolvimento do projeto, a comunicação entre as ações de apresentação e a interface de manipulação foi implementada por chamada de subrotinas.

A ação de apresentação é uma rotina em PASCAL que possui três parâmetros: string com o conteúdo do campo no BD, string com o conteúdo editado pela rotina e string com as especificações de apresentação produzidas.

As especificações de apresentação estão colocadas dentro de um string na distribuição mostrada a seguir.

Tipo Informação	Posição Inicial	Tamanho	Significado
largura	001	003	largura final do campo no plano de representação; dependente do dispositivo ser em alta resolução; unidade em caracteres (baixa resolução) ou pontos (alta resolução).
altura	004	003	altura do campo a partir do ponto de referência; também depende da resolução e do dispositivo.
profundidade	007	003	profundidade do campo a partir do ponto de referência; também depende da resolução e do dispositivo.

tipo apresentação	010	001	0 - sem valor 1 - normal 2 - secreto 3 - enfatizado
moldura	011	001	0 - sem valor 1 - com moldura 2 - sem moldura
tipo caracteres	012	001	0 - sem valor 1 - ASCII 2 - letras 3 - números 4 - alfanumericos
minúsculas	013	001	0 - sem valor 1 - com minúsculas 2 - sem minúsculas
com branco	014	001	0 - sem valor 1 - com branco 2 - sem branco
valor substituto	015	001	0 - sem valor substituto 1 - substit. quando zeros 2 - substit. quando brancos
caracter	016	001	caracter substituto para valor substituto.
vírgula decimal	017	001	0 - sem valor 1 - vírgula decimal
sinal	018	001	' ' - sem valor '+' - sinais '+' e '-' '-' - sinal '-' apenas.

ponto em milhares	019	001	0 - sem valor 1 - ponto em milhares
caracter flutuante	020	001	caracter flutuante à esquerda de campos numéricos para zeros não significativos.
cor fundo	021	003	cor de fundo do campo.
cor dos símbolos	024	003	cor dos símbolos no campo.
tipo alfabeto	025	002	00 - sem valor 01 - courier 02 - itálico 03 - delegate 04 - grego 05 - gótico 06 - cirílico

No primeiro parâmetro, é colocado o conteúdo do campo como ele se encontra no banco de dados. Caso a ação de apresentação não esteja associada a campos ou a tipos, então este parâmetro não é utilizado. A ação de apresentação detecta que não há conteúdo válido no primeiro parâmetro caso seja um string vazio.

Caso a ação de apresentação seja chamada para agir sobre um campo, é no segundo parâmetro que estará o resultado da edição feita pela ação. Em muitos casos a ação de apresentação apenas faz a cópia do primeiro parâmetro no segundo. Por exemplo, nomes de pessoas são armazenados, comumente, na mesma forma como são apresentados.

No terceiro parâmetro a ação de apresentação coloca as orientações de formatação nela programadas. As orientações dos diversos níveis são combinadas para produzir a formatação

idealizada pelo analista de formulários. A combinação é feita na forma apresentada no capítulo 4 deste trabalho.

As especificações de largura, altura e profundidade são as mesmas apresentadas no projeto COBATEF [PEE 85]. Nesse projeto, o mecanismo de formatação foi desenvolvido para formatar documentos. A formatação é feita considerando que cada elemento é uma "caixa" com três coordenadas para dimensionamento. Esta forma é ilustrada na figura 18.

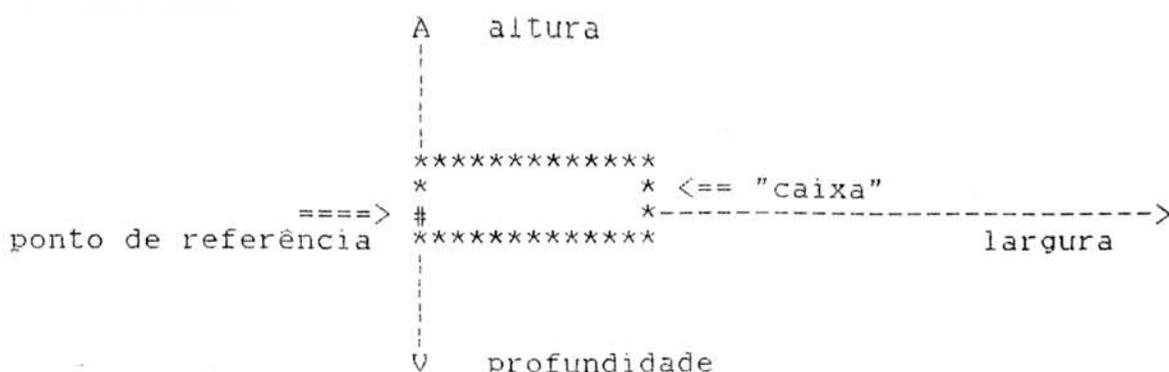


Figura 18 Coordenadas para dimensionamento do projeto COBATEF

Os elementos formatados no projeto COBATEF são palavras, frases, parágrafos, etc. No contexto de formulários, esta mesma idéia foi aproveitada para campos e grupos de campos. O ponto de referência é a origem deste sistema de coordenadas e identifica o ponto de localização do elemento a ser formatado. Qualquer que seja o tamanho desejado do campo, dado pela combinação de largura, altura e profundidade, o ponto de referência não muda. O analista de formulários, ao elaborar as diversas apresentações, decide quais os tamanhos dos diversos componentes das apresentações. Esta decisão fica armazenada nas ações de apresentação produzidas.

As especificações de tipo de apresentação, moldura e cores (de fundo e dos símbolos) são repassadas para o "driver" de restituição. Não implicam em maiores cuidados pela interface.

As especificações de tipo de caracteres, minúsculas e a presença de branco como caracter válido são usadas pela interface

para controlar os conteúdos sendo preenchidos nos respectivos campos. A interface, no exato momento da digitação do valor, verifica se o mesmo pertence ao conjunto de caracteres válidos especificado.

A especificação de valor substituto é usada no momento que está sendo produzida a formatação com os valores vindos pela instância no banco de dados e no momento que o usuário preenche um campo com somente zeros ou brancos.

As especificações de vírgula decimal, sinal, ponto em milhares e caracter flutuante são usadas no preenchimento de campos numéricos. A interface atua no momento do preenchimento de cada caracter do campo não permitindo a digitação de caracteres diferentes de dígitos e efetivando a edição no momento que assim se fizer necessário.

A especificação de tipo de alfabeto permite usar padrões de representação de caracteres diferentes. Os valores definidos são para exemplificar. O "driver" de restituição deve reconhecer o padrão definido; caso não reconheça, é usado o alfabeto padrão do equipamento.

5.6 Características da Implementação:

A implementação corrente está sendo ultimada em equipamentos compatíveis com o microcomputador IBM PC e com Apple II Plus, usando a linguagem PASCAL comercializada pela MicroSoft Corp.

Além disso, a rede utilizada é uma rede Cetus com nodos para os equipamentos citados. Num dos IBM PC, está localizado um disco Winchester de 20 MB para uso pelo servidor de formulários e pelo DD.

6 CONCLUSÕES:

Neste capítulo é feita uma retrospectiva desta dissertação consolidando o trabalho realizado, caracterizando as necessidades e mostrando o que foi feito. São descritos, também, tópicos para futuros estudos no contexto do projeto.

6.1 A Proposição Inicial do Projeto:

A origem deste projeto foi a proposição de automatizar os serviços administrativos do Centro de Processamento de Dados da UFRGS, ou seja, trabalhar em automação de escritórios. Neste contexto foi criado um grupo de trabalho para propor as atividades para efetivar esta tarefa. Estudos foram feitos e a proposição inicial do projeto está apresentada em [OLI 85]. Neste trabalho fica caracterizada a opção para o desenvolvimento de um sistema integrado de formulários.

Os formulários dentro do escritório possuem importância particular por ser o instrumento básico para o fluxo de informações. A modelagem, em computador, das atividades de escritório por formulários pode levar a uma migração, sem rupturas, do sistema manual de tratamento de dados para um sistema automatizado.

Outro aspecto da proposta para o projeto é a de modelar o escritório nas suas atividades estruturadas que correspondem a uma grande parte das atividades realizadas [PAN 84]. As atividades estruturadas são aquelas conhecidas de antemão, ou seja, as tarefas realizadas normalmente pelo pessoal do escritório, tais como, preenchimento de notas fiscais, elaboração da tabela de preços, fechamento de caixa do mes, etc.

A proposta para esta dissertação foi de participar na elaboração de um modelo de dados para formulários e de especificar uma interface de manipulação de formulários

eletrônicos, para operar em um ambiente integrado de escritórios, baseado em uma rede de microcomputadores.

Em paralelo a este trabalho, encontra-se em elaboração uma dissertação cuja proposta foi também de participar da elaboração de um modelo de dados para formulários e de especificar um servidor de arquivos para formulários. Além disso, alguns trabalhos individuais de alunos de graduação e pós-graduação têm complementado o desenvolvimento do projeto.

6.2 O Modelo de Dados para Formulários: o Objeto Formulário:

O ponto de partida para a proposição de um modelo de dados para formulários foram os trabalhos desenvolvidos no projeto TIGRE no INPG (Institut National Polytechnique de Grenoble), França: [BOG 83] e [OLI 84]. Estes trabalhos, dentre outros, propuseram um modelo de dados para documentos e analisaram diversos aspectos vinculados à elaboração e manipulação de documentos.

Numa visão conceitual, o formulário é considerado um tipo particular de documento e, para o contexto do projeto, os conceitos desenvolvidos para documentos foram adaptados ao caso particular dos formulários. Além disso, para a especificação do modelo conceitual de formulário, foi adotada a proposição feita em [BOG 83] de decompõe esta atividade em níveis de especificação. São eles: o modelo sintático, o modelo de apresentação e as especificações do dispositivo.

Um aspecto importante desse contexto é o fato das pesquisas mais recentes em banco de dados terem se voltado para estruturas de dados mais complexas e com mais semântica associada. De fato, uma das aplicações que necessitam de modelos complexos é a automação de escritórios.

Em recente encontro internacional realizado em Asilomar,

California, USA, em setembro de 1986, foi discutido, sob diversos aspectos, esta agregação da realidade no modelo de dados na forma de **objetos**. Um resumo das idéias principais e dos projetos em andamento em SGBDs orientados à Objetos é reportado em [DIT 86]. Segundo o mesmo autor, um objeto é um modelo de dados com estruturas mais complexas para representar a realidade e com mecanismos para agregar a semântica da realidade no modelo de dados.

A influência dessas pesquisas no projeto foi de dar respaldo conceitual ao trabalho sendo desenvolvido para proposição do modelo de dados para formulários. A primeira preocupação nesta proposição foi de respeitar a separação entre estrutura e apresentação no objeto. Com isso é mantida a independência entre os dois contextos e viabilizado o trabalho em separado da especificação da interface de manipulação e do servidor de formulários.

O objeto formulário é composto de um cabeçalho e de um corpo. O cabeçalho do formulário corresponde à descrição externa. Esta descrição engloba aspectos pertinentes sobre quem definiu o formulário, em que estágio da elaboração ele se encontra, em que versão ele está, etc. As informações pertinentes ao modelo externo do formulário estão estabelecidas na norma ISO 646. Um protótipo para armazenar o modelo externo de formulários foi construído no ambiente desta universidade e documentado em [OLI 85].

O corpo do formulário é definido pela enumeração de suas propriedades. Cada propriedade é representada por meio de um campo definido sobre um tipo de dados semelhante aos da linguagem Pascal. A manipulação do formulário é feita pela declaração de uma variável sobre o tipo.

6.3 A Definição Sintática de Formulários:

Para a especificação do objeto formulário optou-se por utilizar a linguagem Pascal e estendê-la. A extensão foi efetivada de duas formas: agregação de mais elementos sintáticos na definição de tipos do Pascal e criação de um novo tipo de dados, o tipo formulário, coerente com os padrões sintáticos da linguagem. A extensão da linguagem visa criar mecanismos para agregar as diversas formas de semântica ao tipo de dados. Esta semântica, no contexto do projeto, foi separada em dois tipos: semântica de conteúdo e semântica de apresentação.

A semântica de conteúdo visa garantir a integridade do tipo, ou seja, garantir que os valores que um campo declarado sobre o tipo venha a possuir sempre obedeça as restrições estipuladas na elaboração do tipo.

A semântica de apresentação visa agregar regras de formatação ao tipo de dados. Um formulário é visto de diferentes formas dentro do escritório, ou seja, de acordo com a atividade a ser realizada, o usuário recebe uma visão apropriada. A forma de apresentação faz parte da semântica do tipo de dados e isto é mostrado no item 6.4. deste capítulo.

A definição sintática de formulários é dividida em três contextos: estrutura, apresentação e operações. No contexto da estrutura, é feita a definição dos campos que fazem parte do formulário e a associação de ações de conteúdo a nível de campo. No contexto da apresentação são especificadas as diversas apresentações que o formulário possui, relacionando os campos que pertencem a cada uma das apresentações, e associando as ações de apresentação aos campos e às visões (apresentações) que o formulário possui. No contexto de operações são associadas ao tipo formulário as ações de conteúdo que visam agregar ao tipo a semântica da realidade em relação aos valores armazenados.

6.4 Semântica de Apresentação:

A forma de apresentação de formulários é de grande importância pois visa o fácil entendimento e rápido preenchimento. A elaboração de formulários em papel é orientada por uma série de regras adotadas pelas organizações junto com regras estéticas que levam a uma aparência agradável e que propiciam uma utilização facilitada.

A elaboração das apresentações dos formulários eletrônicos segue os mesmos padrões dos formulários em papel. O posicionamento dos campos no plano de apresentação associado a mecanismos de formatação, tais como, cores, molduras e diferentes tamanhos, são recursos utilizados para realçar prioridades, indicar obrigatoriedade e vinculações no preenchimento.

A apresentação complementa a elaboração do formulário. Uma apresentação mal feita leva o usuário a um preenchimento errado do formulário.

A apresentação faz parte da realidade a ser representada e, por conseguinte, da semântica do formulário. O objeto deve, portanto, prover mecanismos para incorporar esta semântica nos diversos níveis de representação. O objeto formulário proposto neste projeto tem como um dos objetivos principais disponibilizar estes mecanismos.

A associação de um nome ou expressão ao identificador do campo ilustra uma das características do objeto proposto. Em todos os níveis, nomes ou expressões podem ser associados a identificadores para permitir que, na manipulação, o usuário não trabalhe com códigos ou siglas que normalmente caracterizam a definição de identificadores. O nome pelo qual o dado é conhecido no ambiente do escritório é a forma de manipulá-lo no sistema de formulários eletrônicos.

Além da associação de nomes aos identificadores dos campos

para aproximar o usuário do formulário, a apresentação é composta de um cabeçalho para identificar a função do formulário no sistema, de transações que é a forma de mapear as ações do usuário sobre o formulário em primitivas do servidor de formulários e de textos para os mecanismos de ajuda ao usuário da interface de manipulação.

6.5 Mecanismos de Visões ("Views"):

Num sistema integrado de formulários, o formulário eletrônico passa a ser o elemento único de tratamento de informações dentro do sistema. É sobre o mesmo que é feito o trabalho de transformação e interpretação das informações.

Por conseguinte, o formulário passa por diversas etapas de manipulação. Um formulário é inicializado e incluído, com dados iniciais, e passa a percorrer as diversas etapas de sua manipulação, ora recebendo novos dados ou alterando os já presentes, ora servindo como fonte de informações para o usuário ou para novos formulários.

Como consequência, é importante que haja uma apresentação para cada etapa da manipulação do formulário para permitir que cada usuário tenha uma apresentação voltada às suas necessidades com o subconjunto de campos de seu interesse formatados convenientemente para ter uma aparência agradável e para ser fácil seu entendimento e seu preenchimento. Estas diversas formas diferentes de visualizar o mesmo formulário vem exatamente de encontro à conceituação de visões em banco de dados. O objeto aqui proposto tem um contexto próprio para definição das apresentações onde são especificadas as diversas visões de cada formulário.

As atividades de escritório são distribuídas entre as diversas pessoas que as executam. Estas atividades, pelo grau de dificuldade, responsabilidade e tipo de informação manipulada,

são classificadas e associadas a um "tipo funcionário" que caracteriza o perfil da pessoa que as deve realizar (secretário, agente administrativo, técnico em contabilidade, etc.). Caracterização semelhante foi proposta em [SAN 86] e [TSI 82]. É importante realçar que não há uma vinculação biunívoca entre um "tipo funcionário" e uma pessoa do escritório. As atividades de um "tipo funcionário" podem ser efetivadas por uma pessoa, um grupo de pessoas e, até, uma pessoa pode aglutinar um grupo de "tipos funcionários".

6.6 Ações de Apresentação:

A semântica de apresentação é a agregação dos mecanismos de apresentação ao objeto. Ela pode ser agregada em quatro contextos: tipo de dados, campo do formulário, grupo de campos do formulário e tipo formulário.

A associação de ação de apresentação a nível de tipo de dados permite definir a formatação do tipo de dados. Todos os campos de formulários que forem definidos sobre um tipo de dados com ação de apresentação associada tem o mesmo padrão de formatação. Isto significa que independente de quais campos estejam definidos sobre o tipo e de que formulários eles pertençam a formatação desses campos é a mesma.

A associação de ação de apresentação a nível de campo permite definir a formatação do campo numa particular apresentação (visão) do formulário, ou seja, só é válida para aquela formatação em particular onde ela foi vinculada ao campo.

A associação de ação de apresentação a nível de grupo de campos permite definir uma formatação conjunta para um grupo de campos de uma apresentação. O mecanismo de agrupamento de campos, na proposição deste projeto, é feita a nível de apresentação (visão) de formulário. Este mecanismo não é permitido a nível de definição dos campos do formulário porque as apresentações não

garantem a presença de todos os campos do formulário e esta vinculação poderia ficar descaracterizada nas apresentações.

A associação de ação de apresentação a nível de formulário permite definir a formatação a nível do formulário como um todo. Estas especificações não entram em conflito com as dos campos ou grupos de campos constantes da apresentação. Elas agem sobre o cabeçalho do formulário, sobre as transações e sobre os textos de ajuda associados a nível de apresentação.

A combinação dessas diversas ações é feita pela interface de manipulação. As ações de maior prioridade são às associadas a tipos de dados, ou seja, se sobrepõem sobre as dos outros níveis. As ações a nível de campos se sobrepõem às de nível de grupos de campos e às de formulários e as de grupo de campos se sobrepõem às de formulários.

Exceto a nível de ações de apresentação associadas a formulários, não são assumidos valores por omissão na aplicação das ações de apresentação.

Todas as especificações de formatação são lógicas e visam a construção de uma apresentação ideal do formulário. A apresentação que, efetivamente, é vista pelo usuário é o resultado da adaptação da apresentação ideal feita pelo "driver" do equipamento. Este "driver" tem por objetivo adaptar as especificações lógicas de formatação nas reais capacidades que o equipamento possui. Este nível de independência permite um grau de liberdade grande na definição de formatações e desvincula, da interface, o conhecimento das características do equipamento. Portanto, para cada tipo de equipamento, há um "driver" apropriado para adaptar as formatações ao equipamento.

6.7 Arquitetura do Projeto e Estágio Atual dos seus Módulos:

A arquitetura do projeto é composta de quatro módulos

principais: o Dicionário de Dados (DD), o Servidor de Arquivos para Formulários (GBD), a Interface de Definição de Formulários (IDF) e a Interface de Manipulação de Formulários (IMF).

O DD armazena as definições completas dos diversos tipos formulários, dos tipos de dados definidos no contexto do sistema e das variáveis declaradas sobre os tipos formulários. Armazena, também, as diversas apresentações dos formulários com a relação de campos pertencentes a cada uma delas. Armazena, finalmente, os nomes das diversas ações semânticas, de conteúdo e de apresentação, associadas aos diversos componentes da descrição do sistema.

Em termos funcionais, o DD é um elemento ativo na operação do sistema. É nele que os outros componentes do sistema buscam as descrições dos formulários e, conforme o caso, as ações semânticas associadas às descrições. Os mecanismos de ajuda ("help") também acessam o DD para recuperar descrições ou os textos associados aos componentes armazenados.

No estágio atual do projeto, o DD está especificado para ser implementado em Pascal. Esta especificação encontra-se no capítulo 3 deste trabalho.

O GBD armazena as diversas instâncias dos formulários do sistema garantido sua integridade e qualidade dos dados. No contexto do GBD são executadas as ações de conteúdo, seja a nível de campos, seja a nível do formulário. O acesso ao GBD pelos outros componentes do sistema é feito por primitivas de consulta e de atualização.

Na concepção do sistema, para funcionamento em rede, há um único servidor na rede enquanto há várias cópias da IMF em execução nos microcomputadores interagindo com o GBD. Para tanto, o GBD garante a integridade dos dados quanto ao acesso concorrente.

A especificação completa do GBD é um dos objetos da dissertação sobre servidor de arquivos para formulários e nela são analisados os formulários eletrônicos no contexto de armazenamento e manipulação.

A IDF é o componente que habilita o analista de formulários a definir os diversos formulários do escritório, suas apresentações e modelar o fluxo de formulários dentro do ambiente de escritório. Independente da metodologia usada para modelar o escritório, ao final, é obtido o conjunto de dados a serem manipulados, as regras de fluxo dos formulário dentro do escritório e os campos tratados em cada estação. Portanto, a IDF supre, com ferramentas de apoio, o desenvolvimento do fluxo de dados do escritório.

A IDF possui um outro contexto na interação com o analista de formulários que é a tarefa de elaborar as apresentações dos formulários, tarefa semelhante à elaboração de formulários em papel realizada atualmente nos escritórios. A elaboração das apresentações à parte da tarefa de modelagem do sistema se justifica por ser essa um processo com ingredientes estéticos resultantes da padronização da empresa, experiência do analista na elaboração e aceitação de outros formulários e ergonomia.

No estágio atual do projeto, não foi adotada uma metodologia para modelagem de sistemas de escritórios baseados em formulários. Estudos tem sido realizados para a modelagem baseada em redes de Petri. Como consequência, não foi feita a especificação da interface de definição para modelagem do fluxo de formulários.

No aspecto de definição de formulários e de suas apresentações, foi especificado e implementado um reconhecedor para definição de formulários escrita na extensão proposta da linguagem Pascal. Este reconhecedor não é interativo. O lote de definições é processado e produzidas chamadas das primitivas do DD para armazenar as descrições dos formulários.

A IMF tem por objetivo habilitar a manipulação dos formulários pelo usuário de forma interativa. Por ser um dos objetivos principais deste trabalho, a IMF é tratada em um item próprio, a seguir.

6.8 Interface de Manipulação:

A interface de manipulação de formulários (IMF) é o resultado da ação do software que faz chegar o formulário ao usuário, com sua estrutura e semântica associadas.

A semântica a nível de apresentações é garantida pela própria interface. Ao receber uma ocorrência do GBD, é consultada a descrição da apresentação no DD, é produzido o formulário formatado e, pelas características do dispositivo de saída, criada a restituição.

A IMF é percebida pelo usuário de duas formas: menus para seleção de atividades e apresentações de formulários. Os menus para seleção de atividades servem para auxiliá-lo a selecionar o formulário que deseja manipular, seja através das transações catalogadas pelo sistema, seja pela seleção do formulário em si.

Estão especificadas duas formas de selecionar o formulário a ser manipulado: pelo nome do formulário e pela transação a ser executada. A forma de utilização do sistema por menus parte de um menu principal para a seleção do tipo de atividade desejada, semelhante à implementação do projeto FORMANAGER [YAO 84].

Após o usuário selecionar o formulário desejado, a IMF produz a formatação do mesmo. Dependendo do tamanho da apresentação do formulário e do dimensionamento da janela adotada, o formulário aparece completo ou não. A IMF disponibiliza comandos para "percorrer" a janela sobre o formulário, caso seja necessário.

O formato da tela é dividida em dois contextos: área de apresentação e área de diálogo. Padrão semelhante foi implementado no projeto SPECDOQ divulgado em [KIT 84]. A área de apresentação é a zona onde a interface coloca a formatação do formulário. A área de diálogo, para este projeto, fica nas últimas 4 linhas da tela e serve para comunicação entre a interface e usuário: comandos de navegação entre os diversos menus, acionamento dos mecanismos de ajuda, transações disponíveis sobre a apresentação, comando de controle para manipulação de formulários maiores que a janela, comandos de consulta, folheio e classificação de formulários, etc.

A atuação da IMF em relação ao conteúdo dos campos limita-se às especificações de conjunto de caracteres válidos e edições para campos numéricos. Esta atuação é a nível de cada caracter digitado pelo usuário. A verificação das restrições de integridade a nível de campos e a nível de formulários é feita pelo GBD e, por consequência, após o término da manipulação da instância corrente e transmissão da mesma para o GBD.

A comunicação entre a IMF e o GBD é feita por arquivos devido a ser este o mecanismo de interação entre os nodos da rede adquirida. Esta comunicação é feita por dois pares de arquivos, um par para solicitações e outro par para respostas. Em cada par, um arquivo contém as especificações de controle: no envio, comandos e a identificação de qual formulário está sendo manipulado; no retorno, informações sobre a execução correta ou não dos comandos e do identificador do formulário em questão. O outro arquivo do par contém, conforme o comando, uma instância do formulário.

A comunicação entre a IMF e o DD restringe-se a consultas da IMF ao DD sobre descrições de formulários. O DD fica armazenado junto com o servidor de arquivos. A IMF busca somente a descrição corrente do formulário sendo manipulado e da apresentação correspondente à etapa sendo efetivada. Esta restrição de descrições presentes vem ao encontro às características

recomendadas no projeto ANDREW documentado em [MOR 86].

A interação da IMF com as ações de apresentação são feitas por chamadas normais de subrotinas em Pascal. A ação de apresentação é ativada e recebe, conforme o caso, o conteúdo a editar, edita-o e o devolve junto com um conjunto de especificações de apresentação. A definição do que é feito por cada ação de apresentação pertence ao contexto da IDF e fica a cargo do analista de formulários.

No estágio atual, a IMF encontra-se especificada para funcionamento em um ambiente em redes. As ações de apresentação são programas escritos em Pascal e a forma de interagir com a IMF é por chamada de rotinas e está mostrada no capítulo 5.

6.9 Futuras Táticas para Estudo e Especificação:

Segundo a classificação mostrada em [DIT 86] para SGBDs orientados à Objetos, o projeto como atualmente está especificado enquadra-se como um sistema orientado à objetos de forma estrutural ("structural object-orientation"). Para a continuação do trabalho visando agregar as características de sistemas orientados à objetos de forma também operacional e assim se enquadrar na classificação mais ampla proposta por Dittrich ([DIT 86]) está sendo realizada a integração do projeto com os trabalhos sobre o modelo E proposto em [SAN 86] e cujas primitivas de manipulação já encontram-se implementadas.

Esta integração, a princípio, está sendo conduzida pela representação do objeto formulário nos conceitos e primitivas do modelo E e conseqüentemente do dicionário de dados, e a integração do servidor de formulários com as primitivas de manipulação para armazenamento das instâncias de formulários.

A especificação da IDF (Interface de Definição de Formulários) é, sem dúvida, uma área a desenvolver. E'

subdividida em dois contextos: definição de uma metodologia para modelagem de escritórios em formulários e especificação de uma interface de definição de apresentações de formulários.

A metodologia, num primeiro momento, pode partir da modelagem de atividades por redes de Petri. Vários trabalhos apresentam como conveniente o uso dessa técnica: [TSI 82] e [RIC 81]. A modelagem de formulários em si deve habilitar mecanismos para, entre outras coisas, tornar operacional a definição de operadores sobre tipos derivados.

As formas de habilitar o uso do sistema pelo usuário para atividades não estruturadas é outro tópico a abordar. O uso de álgebra relacional para esta manipulação é uma alternativa.

'sem' ! 'fim'

<declaracao-tipo-dados> ::= '=' <tipo-dados> <acao-apresentacao>
 <apresentacao-textos>

<tipo-dados> ::= <tipo-basico> ! <tipo-estruturado> ! <tipo-tempo>

<tipo-basico> ::= <tipo-simples> <derivacao> ! <tipo-restrito>

<tipo-simples> ::= 'inteiro' ! 'real' ! 'booleano' ! 'caracter' !
 'string[' <inteiro-sem-sinal> ']'

<derivacao> ::= <empty> ! ':' <ident-rotina>

<tipo-restrito> ::= <tipo-escalar> ! <tipo-intervalo> !
 <identif-tipo>

<tipo-escalar> ::= '(' <conjunto-elementos-escalares> ')'

<conjunto-elementos-escalares> ::= <elemento-escalar> !
 <elemento-escalar> ',' <conjunto-elementos-escalares>

<elemento-escalar> ::= <identificador>

<tipo-intervalo> ::= <constante-numerica> '..'
 <constante-numerica>

<tipo-estruturado> ::= <tipo-arranjo> ! <tipo-registro>

<tipo-arranjo> ::= 'arranjo[' <tipo-restrito> ']' de '
 <tipo-basico>

<tipo-registro> ::= 'registro' <conjunto-tipos-basicos> 'fim'

<conjunto-tipos-basicos> ::= <declaracao-tipo-basico> !
 <declaracao-tipo-basico> ';' <conjunto-tipos-basicos>


```

<conjunto-apresentacoes-nodos> ::= <apresentacao-nodo> |
    <apresentacao-nodo> ';' <conjunto-apresentacoes-nodos>

<apresentacao-nodo> ::= <nodo> | 'grupo' <acao-apresentacao>
    <apresentacao-textos> ';'
    <conjunto-apresentacoes-nodos> 'fim'

<nodo> ::= 'nodo' <nome-nodo> <acao-apresentacao>
    <apresentacao-textos>

<conteudo-texto> ::= '#' <texto> '#'

<constante-numerica> ::= <sinal> <inteiro-sem-sinal>

<sinal> ::= <empty> | '+' | '-'

<identif-tipo> ::= <identificador>

<ident-rotina> ::= <identificador>

<identificador-tipo-formulario> ::= <identificador>

<nome-nodo> ::= <identificador>

<titulo> ::= <texto>

<max> ::= <inteiro-sem-sinal>

<min> ::= <inteiro-sem-sinal>

<inteiro-sem-sinal> ::= <digito> | <digito> <inteiro-sem-sinal>

<digito> ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' |
    '9'

```

Anexo 2: Exemplo de uma aplicação baseada em formulários

Sistema de locação de veículos

Este sistema prevê os seguintes formulários :

1. Formulário para cadastramento de clientes (pessoas).
2. Formulário para cadastramento de clientes (empresas).
3. Formulário para cadastramento de veículos.
4. Formulário para locação de veículo.

O Formulário para cadastramento de clientes (pessoas) possui as seguintes informações:

. Código do cliente - Um número sequencial atribuído ao cliente quando da inclusão deste no cadastro. Pode ser atualizado automaticamente pelo sistema.

. Nome do cliente -

. Endereço do cliente - Rua, Avenida, complemento.

. Bairro - Bairro referente ao endereço

. Cep - Código de endereçamento postal

. Município -

. U.F - Sigla do estado ou território

. Telefone - Composto por prefixo e número.

. Data de nascimento -

. Carteira de identidade - Número da carteira de identidade e local onde foi emitida.

. CPF - Número do cadastro de pessoas físicas da receita federal.

. Profissão do cliente -

. Local de trabalho do cliente -

. Endereço do local de trabalho - Rua, Avenida, complemento.

. Bairro do endereço do local de trabalho -

. Cep -

. Município -

- . U.F. -
- . Telefone -
- . Ramal -
- . Cargo que o cliente ocupa -
- . Rendimentos do cliente -

Sintaxe da definição do formulário de cadastramento de clientes:

```
type UF = (AC,AL,AP,AM,BA,CE,DF,ES,FN,GO,MA,MG,MS,MT,PA,
          PB,PR,PE,PI,RJ,RN,RO,RR,RS,SC,SE,SP);
```

```
type CEP = inteiro:VERIFICA-CEP;
```

```
type VALOR = real,APRESENTA-VALOR;
```

```
type
```

```
FORM1(Cadastramento de Clientes: Pessoas Físicas)
```

```
formulario
```

```
estrutura
```

```
  CODIGO-CLP(Codigo):integer, chave, GERACODIGO(sem);
```

```
  NOME-CLP(Nome <peessoa física>):string[40];
```

```
  ENDER-CLP(Endereço <Rua,Av.,compl.>):string[40];
```

```
  BAIRR-CLP(Bairro):string[15];
```

```
  CEP-CLP(CEP):CEP;
```

```
  MUNIC-CLP(Município):string[15];
```

```
  UF-CLP(U.F.):UF;
```

```
  TELEF-CLP(Telefone):string[12];
```

```
  DATNASC-CLP(Data de Nascimento):tempo;
```

```
  CI-CLP(Carteira de Identidade):inteiro;
```

```
  CPF-CLP(CPF):inteiro, VERIFICACPF(SEM);
```

```
  PROF-CLP(Profissao):string[16];
```

```
  LOCTRAB-CLP(Local de Trabalho):string[40];
```

```
  ENDTRAB-CLP(Endereço):string[40];
```

```
  BAIRTRAB-CLP(Bairro):string[15];
```

```
  CEPTRAB-CLP(CEP):CEP;
```

```
  MUNICTRAB-CLP(Município):string[15];
```

```
  UFTRAB-CLP(U.F.):UF;
```

```
  TELETRAB-CLP(Telefone):string[12];
```

```
  RAMAL-CLP(Ramal):inteiro;
```

```
  CARGO-CLP(Cargo do Cliente):string[16];
```

```
    RENDIM-CLP(Rendimentos do Cliente):VALOR;
fim;
apresentacao
PESSOA-FISICA:
visao,
    APRESENTA-CADASTRAM;
transacoes
    INC-CLIENTE(Cadastramento de Cliente Pessoa Fisica);
    ATU-CLIENTE(Atualizacao de Dados Cadastrais de Cliente
Pessoa Fisica)
    fim;
    globais
        cabecalho(Cadastro de Cliente - Pessoa Fisica),APRESENTA-CABEC;
        #Este Formulario serve para Manter o Cadastro de Clientes -
Pessoa Fisica#
    fim;
    grupo, APRESENT-MOLDURA;
        nodo CODIGO-CLP;
        nodo NOME-CLP;
        nodo DATNASC-CLP,APRESENTA-TEMPO;
        nodo CI-CLP;
        nodo CPF-CLP,APRESENTA-CPF;
    fim;
    grupo, APRESENT-MOLDURA;
        nodo ENDER-CLP;
        nodo BAIRR-CLP;
        nodo CEP-CLP;
        nodo MUNIC-CLP;
        nodo UF-CLP;
        nodo TELEF-CLP;
    fim;
    grupo, APRESENT-MOLDURA;
        nodo PROF-CLP;
        nodo RENDIM-CLP;
        nodo LOCTRAB-CLP;
        nodo ENDTRAB-CLP;
        nodo BAIRTRAB-CLP;
```

```

nodo CEPTRAB-CLP;
nodo MUNICTRAB-CLP;
nodo UFTRAB-CLP;
nodo TELETRAB-CLP;
nodo RAMAL-CLP;
nodo CARGO-CLP
fim;
fim;
operacões
    <def-operacoes>
fim;
fim;

```

O Formulário para cadastramento de clientes (empresas) possui as seguintes informações:

- . Código do cliente - Um número sequencial atribuído ao cliente quando da inclusão deste no cadastro. Pode ser atualizado automaticamente pelo sistema.
- . Nome da empresa -
- . Razão social da empresa -
- . Endereço da empresa - Rua, Avenida, complemento.
- . Bairro - Bairro referente ao endereço
- . Cep - Código de endereçamento postal
- . Município -
- . U.F - Sigla do estado ou território
- . Telefone - Composto por prefixo e número.
- . CGC - Número do CGC

Sintaxe da definição do formulário de cadastramento de empresas:

```

type
FORM2(Cadastramento de Empresas: Pessoas Jurídicas)
formulario
estrutura
    CODIGO-CLE(Codigo):inteiro, chave, GERACODIGO(SEM);
    NOME-CLE(Nome <peessoa jurídica>):string[40];
    RAZAOSOC-CLE(Razao Social):string[40];

```

```

DIRET-RESP-CLE(Diretor Responsavel):string[40];
ENDER-CLE(Endereco <Rua,Av.,compl.>):string[40];
BAIRR-CLE(Bairro):string[15];
CEP-CLE(CEP):CEP;
MUNIC-CLE(Municipio):string[15];
UF-CLE(U.F.):UF;
TELEF-CLE(Telefone):string[12];
CGC-CLE(CGC):inteiro, VERIFICACGC(SEM);
fim;
apresentacao
PESSOA-JURIDICA:
visao,
  APRESENTA-CADASTRAM;
transacoes
  INC-PESJURID(Cadastramento de Pessoa Juridica);
  ALT-PESJURID(Atualizacao dos Dados de Pessoa Juridica)
fim;
globais
  cabecalho(Cadastro de Pessoas Juridicas),APRESENTA-CABEC;
  #Este Formulario serve para Manter Atualizado o Cadastro de
Pessoas Juridicas#
  fim;
grupo, APRESENT-MOLDURA;
  nodo CODIGO-CLE;
  nodo NOME-CLE;
  nodo RAZAOSOC-CLE;
  nodo DIRET-RESP-CLE;
  nodo ENDER-CLE;
  nodo BAIRR-CLE;
  nodo CEP-CLE;
  nodo MUNIC-CLE;
  nodo UF-CLE;
  nodo TELEF-CLE;
  nodo CGC-CLE,APRESENTA-CGC
  fim;
fim;

```

```

operacoes
    <def-operacoes>
fim;
fim;

```

O Formulário para cadastramento de veículos possui as seguintes informações:

- . Código do veículo - este campo será calculado pelo sistema como um valor numérico sequencial com a finalidade de identificar univocamente um veículo.

- . Nome do fabricante -
- . Marca -
- . Modelo -
- . Ano de fabricação -
- . Data da compra -
- . Valor da compra -
- . Cor -
- . Capacidade de passageiros -
- . Capacidade de carga -
- . Tipo de veículo - poderá ser automóvel, caminhoneta, furgão, trailer.

- . Categoria do veículo - Este atributo determina a posição da tabela de preços para o veículo.

- . Número do registro DUT - Documento Único de Trânsito.
- . Placas -
- . Combustível do veículo - Pode ser gasolina, álcool, diesel.

Sintaxe da definição do formulário de cadastramento de veículos.

```

type TIPOVEICULO = (AUTOMOVEL,CAMIONETA,FURGAO,TRAILER);
type CATEG-VEICULO = (A,B,C,D,E,F,G,H,I,J,K);
type COMBUSTIVEL = (GASOLINA, ALCOOL, DIESEL);
type ANOFABR = 1975..1990;
type

```

```

FORM3(Cadastramento de Veiculos)
formulario
estrutura
  CODIGO-VEI(Codigo):inteiro, chave, GERACODIGO(SEM);
  FABR-VEI(Nome do Fabricante):string[30];
  MARC-VEI(Marca do Veiculo):string[12];
  MODEL-VEI(Modelo do Veiculo):string[20];
  ANO-VEI(Ano de Fabricacao):ANOFABR;
  VALOR-VEI(Valor de Compra de Veiculo):VALOR;
  DATCOMPR-VEI(Data da Compra):tempo;
  COR-VEI(COR):string[12];
  CAPPESS-VEI(Capacidade Pessoas):inteiro;
  CAPCARG-VEI(Capacidade Carga):inteiro;
  TIPO-VEI(Tipo):TIPOVEICULO;
  CATEG-VEI(Categoria):CATEG-VEICULO;
  REGDUT-VEI(Registro DUT):inteiro;
  PLACAS-VEI(Placas):string[7];
  COMBUST-VEI(Combustivel):COMBUSTIVEL;
fim;
apresentacao
  CADAST-VEICULOS:
  visao,
  APRESENTA-SAIDA;
  transacoes
    INC-VEIC(Cadastramento de Veiculo para Locar);
    ALT-VEIC(Correcao do Cadastro de Veiculos)
  fim;
  globais
    cabecalho(Cadastro de Veiculos),APRESENTA-CABEC;
    #Este Formulario serve para Manter o Cadastro de Veiculos
da Autolocadora#
  fim;
  grupo, APRESENT-MOLDURA:
  nodo CODIGO-VEI;
  nodo FABR-VEI;
  nodo MARC-VEI;
  nodo MODEL-VEI;

```

```

nodo ANO-VEI;
nodo VALOR-VEI;
nodo DATCOMPR-VEI, APRESENTA-TEMPO;
nodo COR-VEI;
nodo CAPPESS-VEI;
nodo CAPCARG-VEI;
nodo TIPO-VEI;
nodo CATEG-VEI;
nodo REGDUT-VEI;
nodo PLACAS-VEI;
nodo COMBUST-VEI
fim;
fim;
operacoes
    <def-operacoes>
fim;
fim;

```

O Formulário para locação de veículos possui as seguintes informações:

- . Códico do veículo - este campo corresponde ao código do veículo gerado automaticamente pelo sistema.
- . Código do cliente - Este campo corresponde ao código do cliente que está locando o veículo e pode ser referente à pessoa física ou jurídica.
- . Data da locação -
- . Odômetro de saída - kilometragem que consta no odômetro do veículo na data de locação.
- . Opção seguro - O cliente pode optar por seguro total com franquia ou total sem franquia.
- . Danos pessoais - O cliente pode optar (sim/não) pelo seguro contra terceiros.
- . Data prevista para devolução -
- . Data da devolução -
- . Odômetro de chegada - Kilometragem que consta no

odômetro do veículo na data de devolução.

. Combustível consumido - Quando da entrega do veículo o tanque de combustível é completado. Este campo será preenchido com o valor em litros de combustível utilizado.

. Valor do combustível - Este campo será calculado automaticamente pelo sistema com base no tipo de combustível do veículo e preço do litro.

. valor do seguro - É calculado pelo sistema com base na escolha do seguro feita pelo locatário.

. Kilômetros rodados - Representa a quilometragem feita pelo locatário. É calculado automaticamente como a diferença entre odômetro de chegada e odômetro de saída.

. Valor em função da quilometragem - É calculado com base no valor do quilômetro rodado e categoria de veículo.

. Número de dias - É obtido pela diferença entre as datas de chegada e de saída.

. Valor das diárias - É obtido levando-se em conta o valor da diária associada a categoria do veículo e o número de diárias.

. Total a pagar - Este campo será preenchido automaticamente com a soma dos seguintes campos já calculados: valor do combustível, valor do seguro, valor da quilometragem, valor das diárias.

Sintaxe da definição do formulário de locação de veículos.

```

type SIMNAO = (S,N);
type OPCSEG = (1,2,3) # 1 - acidentes, 2 - Roubo, Incendio, 3 -
Total #;
type
  FORM4(Locacoes) =
    formulario
      estrutura
        CODVEI-LOC(Codigo Veiculo):inteiro,chave,VERIFCODVEICULO(sem);
        CODCLI-LOC(Codigo Cliente):inteiro,chave,VERIFCODCLIENTE(sem);
        DATASAI-LOC(Data de Saida):tempo;
        ODOMSAI-LOC(Odometro de Saida):inteiro;

```

```

OPCSEG-LOC(Opcao Seguro):OPCSEG;
DANOSPESH-LOC(Danos Pessoais Seguro):SIMNAO;
DEVPREV-LOC(Data Devolucao Prevista):tempo;
DEVOL-LOC(Data Devolucao):tempo;
ODOMCHEG-LOC(Odometro de Chegada):inteiro;
LITRCOMB-LOC(Litros de Combustivel):real;
VALCOMB-LOC(Valor Combustivel):VALOR,CALCVCOMB-LOC(sem);
KMROD-LOC(Km Rodados):inteiro,CALCKM-LOC(sem);
VALKMROD-LOC(Valor Km Rodados):VALOR,CALCVKMROD-LOC(sem);
NDIAS-LOC(Numero de Dias Locados):inteiro,CALCNDIAS-LOC(sem);
VALNDIAS-LOC(Valor das Diarias):VALOR,CALCVDIAR-LOC(sem);
TOTAL-LOC(Valor a Pagar):VALOR,CALCVTOT-LOC(sem);
fim;
apresentacao
SAIDA-VEICULO:
visao,
  APRESENTA-SAIDA;
transacoes
  SAIDA-VEIC(Saida de Veiculo com Cliente);
  ALT-SAIDA-VEIC(Correcao das Informacoes sobre Saida de
Veiculo)
  fim;
globais
  cabecalho(Locacao de Veiculo),APRESENTA-CABEC;
  #Este Formulario serve para Registrar a Saida de um
Veiculo Locado#
  fim;
grupo, APRESENT-MOLDURA;
  nodo CODVEI-LOC;
  nodo CODCLI-LOC;
fim;
grupo, APRESENT-MOLDURA;
  nodo DATASAI-LOC,APRESENTA-TEMPO;
  nodo ODOMSAI-LOC;
  nodo OPCSEG-LOC;
  nodo DANOSPESH-LOC;
  nodo DEVPREV-LOC,APRESENTA-TEMPO;

```

```

    nodo CAUCAO-LOC
  fim;
fim;
DEVOLUCAO-VEICULO:
visao,
  APRESENTA-DEVOLUCAO;
  transacoes
    DEVOL-VEIC(Devolucao de Veiculo pelo Cliente e Cobranca);
    ALT-DEVOL-VEIC(Correcao das Informacoes sobre Devolucao de
Veiculo)
  fim;
  globais
    cabecalhoDevolucao de Veiculo),APRESENTA-CABEC;
    #Este Formulario serve para Registrar a Devolucao de um
Veiculo e para a Cobranca do Valor da Locacao#
  fim;
  grupo, APRESENT-MOLDURA;
    nodo CODVEI-LOC;
    nodo CODCLI-LOC;
  fim;
  grupo, APRES-MOLDURA-LEIT;
    nodo DATASAI-LOC,APRESENTA-TEMPO;
    nodo ODOMSAI-LOC;
    nodo OPCSEG-LOC;
    nodo DANOSPES-LOC;
    nodo DEVPREV-LOC,APRESENTA-TEMPO;
    nodo CAUCAO-LOC
  fim;
  grupo, APRESENTA-MOLDURA;
    nodo DEVOL-LOC,APRESENTA-TEMPO;
    nodo ODOMCHEG-LOC;
    nodo LITRCOMB-LOC;
    nodo VALCOMB-LOC;
    nodo KMROD-LOC;
    nodo VALKMROD-LOC;
    nodo NDIAS-LOC;
    nodo VALNDIAS-LOC;

```

```
nodo TOTAL-LOC;
nodo TOTLIQ-LOC
fim;
fim;
fim;
operacoes
    <def-operacoes>
fim;
fim;
```

BIBLIOGRAFIA

- [ALB 85] ALBANO, A.; CARDELI, L.; ORSINI, R. Galileo: a Strongly Typed, Interactive Conceptual Language. ACM Transactions on Data Base Systems, New York, 10(2):230-60, June 1985.
- [ADI 85] ADIBA, M.; QUANG, B. N.; OLIVEIRA, J. P. M. Time Concept in Generalized Data Bases. In: ACM ANNUAL CONFERENCE, Denver, Oct 14-16, 1985. Proceedings. New York, ACM, 1985. p. 214-23.
- [BAR 85] BARBIC, F. et alii. Modeling and Integrating Procedures in Office Information Systems Design. Information Systems, Oxford, 10(2):149-68, July 1985.
- [BAT 84] BATINI, C.; DEMO, R.; DI LEVA, A. A Methodology for Conceptual Design of Office Data Bases. Information Systems, Oxford, 9(3/4):251-63, April 1984.
- [BOG 83] BOGO, G.; RICHY, H.; VATTON, I. Proposition de Modele pour la Normalisation des Documents. Grenoble, Institut National Polytechnique de Grenoble, 1983. (Relatório de Pesquisa TIGRE n. 3)
- [CZE 84] CZEJDO, B. & EMBLEY, D. W. Office Form Definition and Processing using a Relational Data Model. SIGOA BULLETIN, New York, 5(1/2):123-31, Feb. 1984.
- [COD 79] CODD, E. F. Extending the Relational Database Model to Capture more Meaning. ACM Transactions on Database Systems, New York, 4(4):397-434, Dec. 1979.
- [COL 85] COLLET, C. Les Formulaire Multi-Media. In: SEMINAIRE INFORSID 85, Luchon, France, 5-15 May 1985. Proceedings. p. 231-46.

- [DIT 86] DITTRICH, K. R. Object-Oriented Database Systems. In: INTERNATIONAL CONFERENCE ON THE ENTITY-RELATIONSHIP APPROACH, 5., Dijon, Oct. 13-18, 1986. Proceedings. Amsterdam, North Holland, 1986. p.115-34.
- [HAM 81] HAMMER, M. & MCLEOD, D. Database Description with SMD: a Semantic Database Model. ACM Transactions on Database Systems, New York, 6(3):351-84, Sept. 81.
- [KIT 84] KITAGAWA, H.; GOTOH, M.; MISAKI, S.; AZUMA, M. Form Document Management System SPECDOQ - its Architecture and Implementation. SIGQA BULLETIN, New York, 5(1/2):132-42, Jan. 1984.
- [KOV 86] KOVED, L. & SCHNEIDERMAN, B. Embedded Menus: Selecting Items in Context. Communications of ACM, New York, 29(4):117-45, Apr. 1 1986.
- [KOW 82] KOWALSKI, I. & LOPEZ, M. The Document Concept in a Database. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, Orlando, June 2-4, 1982. Proceedings. New York, ACM, 1982. p.276-83.
- [MAR 78] MARCO, T. Structured Analysis and System Specification. New York, Yourdon, 1978.
- [MOR 86] MORRIS, J. H. et alii. ANDREW: a Distributed Personal Computing Environment. Communications of the ACM, New York, 29(3):184-201, Mar. 1986.
- [OLI 85] OLIVEIRA, J. P. M.; AMARAL, K. E. F.; SILVA, O. L. Formulários Eletrônicos: um novo conceito em Automação de Escritórios. In: CONGRESSO NACIONAL DE INFORMATICA, 18., São Paulo, Out. 8-12, 1985. Anais. São Paulo, Sucusu, 1985. p. 815-20.

- [OLI 84] OLIVEIRA, J. P. M. Le Modèle de Données et sa Représentation Relationnelle dans un Système de Gestion de Bases de Données Généralisées, Grenoble, Institut National Polytechnique de Grenoble, 1984. (Tese de Doutorado)
- [OLI 86] OLIVEIRA, J. P. M. Electronic Forms, project description. Porto Alegre, CPGCC DA UFRGS, 1986. (Relatório de Pesquisa CPGCC/UFRGS n. 47)
- [PAN 84] PANKO, R. R. 38 Offices: Analyzing Needs in Individual Offices. ACM Transactions on Office Information Systems, New York, 2(3):226-34, July 1984.
- [PEE 85] PEELS, A; JANSEN, N.; NAWIJN, W. Document Architecture and Text Formatting. ACM Transactions on Office Information Systems, New York, 2(4):347-69, Oct. 1985.
- [RIC 81] RICHTER, G. IML - Inscribed Nets for Modeling Text Processing and Data(base) Management System. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 7., Cannes, 1981. Proceedings. Cannes, IEEE, 1981. p.363-75.
- [SAN 80] SANTOS, C. S.; NEUHOLD, E. J.; FURTADO, A. L. A Data Type Approach to the Entity-Relationship Model. In: CHEN, P. P., ed. Entity-Relationship Approach to System Analysis and Design. Amsterdam, North-Holland, 1980.
- [SAN 86] SANTOS, S. A.; OLIVEIRA, J. P. M.; TAZZA, M. Projeto Físico de Sistemas Distribuídos. In: CONGRESSO NACIONAL DE INFORMÁTICA, 19., Rio de Janeiro, Ago. 14-28, 1986. Anais. Rio de Janeiro, Sucesu, 1986. p. 315-32.

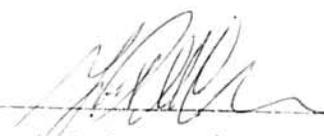
- [TSI 82] TSICHRITZIS, D. Form Management. Communications of the ACM, New York, 25(7):453-78, July 1982.
- [YAO 84] YAO, S. B.; HEVNER, A. R.; SHI, Z. Formanager: an Office Forms Management System. ACM Transactions on Office Information Systems, New York, 2(3):235-62, July 1984.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

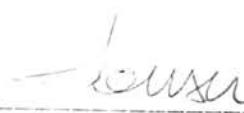
Pós-Graduação em Ciência da Computação

Interface de Usuário para a
Manipulação de Formulários
Eletrônicos

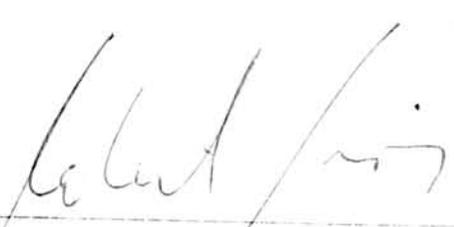
Dissertação apresentada aos Srs.


Prof. Dr. José Palazzo Moreira de Oliveira - Orientador


Prof. Dr. Clesio Saraiva dos Santos


Prof. Dr. Carlos Alberto Heuser

Visto e permitida a impressão
Porto Alegre, 10 / 11 / 87


Prof. Dr. Roberto Tom Price
Coordenador do Curso de Pós-Graduação
em Ciência da Computação